

前言

本参考手册面向应用开发人员，提供有关使用 SPD1121 微控制器储存器与外设的完整信息。

有关 SPD1121 器件的机械与电气特性，请参见该器件的数据手册。

有关带 FPU 的 ARM® Cortex-M4F 内核的信息，请参见相应的 ARM®技术参考手册。

目录

1	存储器和总线架构	32
1.1	系统架构	32
1.2	存储器映射	33
1.2.1	引导 ROM	35
1.2.2	嵌入式 Flash 存储器	36
1.2.3	片上 SRAM	36
1.3	启动引导配置	37
2	电源	38
2.1	单电源供电	38
2.2	过压/欠压检测	39
2.3	睡眠模式	40
2.4	寄存器	41
2.4.1	Power 寄存器列表	41
2.4.2	Power 寄存器	42
3	时钟与复位	54
3.1	复位架构	54
3.2	引脚复位	54
3.3	上电复位	54
3.4	时钟结构	55
3.5	高频振荡器	56
3.6	PLL 时钟	56
3.6.1	输入参考和分频	56
3.6.2	输出分频	56
3.6.3	环路分频	57
3.6.4	PLL 失锁检测	57
3.7	时钟平滑选择器	57
3.8	时钟安全	58
3.9	寄存器	59
3.9.1	Clock 寄存器列表	59
3.9.2	Clock 寄存器	60
4	通用 I/O (GPIO)	82
4.1	概述	82
4.2	引脚复用 (PINMUX)	83

4.2.1	施密特输入模式	84
4.2.2	消抖控制	84
4.2.3	I/O 可选功能	85
4.2.4	调试模式	85
4.3	GPIO 控制模块	86
4.3.1	方向和电平控制	86
4.3.2	边沿触发的中断	87
4.3.3	电平触发的中断	88
4.4	寄存器	91
4.4.1	PINMUX 寄存器列表	91
4.4.2	PINMUX 寄存器	92
4.4.3	GPIO 寄存器列表	120
4.4.4	GPIO 寄存器	122
5	NVIC	145
5.1	概述	145
5.2	中断向量表	146
5.3	中断使能	147
5.4	中断优先级	148
5.5	中断仲裁和挂起	149
5.6	应用场景	150
5.6.1	同优先级和低优先级挂起	150
5.6.2	高优先级抢占	152
6	循环冗余校验模块 (CRC)	154
6.1	概述	154
6.2	CRC 工作流程	155
6.3	寄存器	156
6.3.1	CRC 寄存器列表	156
6.3.2	CRC 寄存器	157
7	通用定时器	163
7.1	概述	163
7.2	功能描述	163
7.3	初始化流程	164
7.4	寄存器	165
7.4.1	TIMER 寄存器列表	165
7.4.2	TIMER 寄存器	166
8	看门狗定时器 (WDT)	171

8.1	概述	171
8.2	功能描述	171
8.3	初始化流程	172
8.4	寄存器	173
8.4.1	WDT 寄存器列表	173
8.4.2	WDT 寄存器	174
9	PWM	178
9.1	PWM 概述	178
9.2	时基产生 (TB) 子模块	181
9.3	计数比较 (CC) 子模块	184
9.4	行为限定 (AQ) 子模块	185
9.5	死区控制 (DB) 子模块	187
9.6	信号封锁 (TZ) 子模块	189
9.7	数字比较 (DC) 子模块	192
9.8	事件触发 (ET) 子模块	194
9.9	寄存器	195
9.9.1	PWM 寄存器列表	195
9.9.2	PWM 寄存器	197
9.9.3	PWMCFG 寄存器列表	262
9.9.4	PWMCFG 寄存器	263
10	增强的捕获模块 (ECAP)	276
10.1	概述	276
10.2	功能描述	276
10.3	捕获模式和辅助 APWM 模式	277
10.4	捕获模式详述	277
10.4.1	事件预分频	278
10.4.2	边沿选择和排序编号	279
10.4.3	单引脚捕获和双引脚捕获	279
10.4.4	持续捕获和一次性捕获控制	279
10.4.5	32 比特计数器和相位控制	280
10.4.6	CAP0-CAP3 寄存器	280
10.4.7	中断控制	280
10.4.8	影子寄存及其控制	281
10.5	APWM 模式详述	282
10.6	ECAP 的应用	283
10.6.1	上升沿触发的绝对时间戳捕获	283
10.6.2	由绝对时间戳捕获周期和占空	284

10.6.3	通过计数器复位捕获相对时间差	285
10.6.4	相对事件模式捕获周期和占空	286
10.6.5	基于计数器复位的双引脚捕获	287
10.7	寄存器	288
10.7.1	ECAP 寄存器列表.....	288
10.7.2	ECAP 寄存器	289
11	模数转换器（ADC）	305
11.1	ADC 概述	305
11.2	ADC 架构	306
11.3	通道 CH 工作原理	307
11.3.1	请求信号	307
11.3.2	ADC 采样模式和通道选择	308
11.3.3	转换优先级	309
11.3.4	平均化处理	309
11.3.5	ADC 采样和转换窗口	310
11.4	转换结束和产生中断原理.....	310
11.5	ADC 时序	312
11.6	ADC 校准	313
11.7	ADC 结果	313
11.7.1	满幅度测量范围	313
11.7.2	ADC 输出码值计算	313
11.7.3	通道码值计算	314
11.8	ADC 安全特性.....	314
11.8.1	输入浮空检测	314
11.9	寄存器	317
11.9.1	ADC 寄存器列表	317
11.9.2	ADC 寄存器	318
12	温度传感器（T-sensor）	342
12.1	概述	342
12.2	精度提升	342
12.3	寄存器	342
13	差分可编程增益放大器（DPGA）	343
13.1	DPGA 概述	343
13.2	输入及输入电压范围.....	343
13.3	应用	343
13.4	偏置校准	345

13.5	寄存器	346
13.5.1	DPGA 寄存器列表	346
13.5.2	DPGA 寄存器	347
14	差分转单端缓冲器（D2S）	349
14.1	概述	349
14.2	功能描述	349
15	模拟比较器（COMP）	350
15.1	模拟比较器概述	350
15.2	模拟比较器结构	350
15.3	模拟比较器输入选择	351
15.4	数字滤波器	351
15.5	数模转换器（DAC）	352
15.6	寄存器	353
15.6.1	COMP 寄存器列表	353
15.6.2	COMP 寄存器	354
16	通用异步收发器（UART）	373
16.1	UART 概述	373
16.2	UART 主要特性	373
16.3	UART 信号描述	374
16.4	UART 操作	375
16.4.1	Reset 复位	376
16.4.2	FIFO 操作	376
16.4.3	自动波特率检测	377
16.4.4	可编程的波特率产生器	378
16.5	寄存器	379
16.5.1	UART 寄存器列表	379
16.5.2	UART 寄存器	380
17	内部集成电路总线（I2C）	393
17.1	概述	393
17.2	I2C 端口信号	393
17.3	I2C 协议	394
17.3.1	字节格式	394
17.3.2	开始信号与停止信号	394
17.3.3	地址模式	394
17.3.4	应答与非应答	395
17.4	I2C 时序参数	396

17.4.1	电平时间配置	396
17.4.2	滤波配置	396
17.4.3	数据保持时间配置	396
17.4.4	数据建立时间配置	396
17.5	I2C 主机模式	397
17.5.1	主机发送数据	397
17.5.2	主机接收数据	397
17.5.3	主机组合传输	397
17.6	I2C 从机模式	398
17.6.1	地址匹配	398
17.6.2	从机发送数据	398
17.6.3	从机接收数据	398
17.7	I2C 发送中止	398
17.8	寄存器	399
17.8.1	I2C 寄存器列表	399
17.8.2	I2C 寄存器	401
18	串行外围设备接口 SPI	439
18.1	SPI 概述	439
18.2	SPI 主要特性	440
18.3	SPI 信号描述	440
18.4	SPI 功能描述	440
18.4.1	功能操作介绍	440
18.4.2	波特率产生	441
18.4.3	FIFO 操作	441
18.4.4	数据帧格式	443
18.4.5	SPI 中断	445
18.5	寄存器	446
18.5.1	SPI 寄存器列表	446
18.5.2	SPI 寄存器	447
19	嵌入式 Flash 存储器 (FLASH)	455
19.1	Flash 主要特性	455
19.2	Flash 架构	456
19.3	读操作	457
19.3.1	XIP 接口读操作	457
19.3.2	Flash 控制器接口读操作	457
19.4	编程和擦除操作	458
19.4.1	使能 Flash 控制器	458

19.4.2	Flash 存储器编程操作	459
19.4.3	Flash 存储器擦除操作	460
19.5	保护机制	463
19.5.1	写保护	463
19.6	Flash 时序设置	463
19.7	配置字（Configuration Words）	463
19.8	寄存器	464
19.8.1	Flash 寄存器列表	464
19.8.2	Flash 寄存器	465
20	系统控制模块	484
20.1	概述	484
20.2	64 位产品识别号	484
20.3	64 位唯一设备识别号	484
20.4	存储器错误中断控制寄存器	485
20.5	存储器 ECC 使能寄存器	485
20.6	存储器错误地址	485
20.7	存储器锁定状态寄存器	485
20.8	复位事件控制寄存器	485
20.9	系统引导信息	485
20.10	寄存器	486
20.10.1	系统寄存器列表	486
20.10.2	系统寄存器	487
21	调试行为（Debug Behavior）	507
21.1	看门狗定时器	507
21.2	通用定时器	507
21.3	PWM	507
21.4	ECAP	507
21.5	UART	508
21.6	SPI	508
21.7	I2C	509
22	降压开关电源（DC-DC）	510
22.1	概述	510
22.2	DC-DC 模块控制	511
22.2.1	DC-DC 模块初始化	511
22.2.2	MCU 和 DC-DC 模块的通信格式	512
22.2.3	通信协议和命令	513

22.3	DC-DC 模块特性	513
22.3.1	可编程的开关频率	513
22.3.2	PWM 和 PFM 模式切换	513
22.3.3	输出短路保护	514
22.3.4	频率抖动选项	514
22.4	寄存器	516
22.4.1	EPWR 寄存器列表	516
22.4.2	EPWR 寄存器	517
22.4.3	Buck 模块寄存器列表	526
22.4.4	Buck 模块寄存器	527
23	预驱模块（高压模块）	532
23.1	概述	532
23.2	低压模块和高压模块的通信	532

图片列表

图 1-1: SPD1121 系统框图	32
图 1-2: SPD1121 存储空间地址映射	33
图 1-3: ROM 存储空间划分	35
图 2-1: 电源系统总览	38
图 2-2: 3.3V 额定电压域检测电路框图	39
图 2-3: 1.2V 额定电压域检测电路框图	39
图 2-4: DVDD 阈值	40
图 2-5: VCAP12 阈值	40
图 3-1: 复位信号框图	54
图 3-2: 上电复位时序图	55
图 3-3: SPD1121 时钟树	55
图 3-4: PLL 结构总览	56
图 3-5: PLL 失锁检测时序图示例 ($f_{PFD} > f_{DIV}$)	57
图 3-6: 时钟交叉检测	58
图 4-1: I/O 控制结构框图	82
图 4-2: PINMUX 结构框图	83
图 4-3: 施密特输入模式	84
图 4-4: 输入消抖	84
图 4-5: GPIO 框图	86
图 4-6: GPIO 边沿检测	87
图 4-7: GPIO 边沿触发的中断	87
图 4-8: GPIO 电平检测	89
图 4-9: GPIO 电平触发的中断	89
图 5-1: 中断框图	145
图 5-2: 中断仲裁和挂起	149
图 5-3: 中断执行过程中触发低优先级中断	150
图 5-4: 入栈过程中触发低优先级中断	150
图 5-5: 出栈过程中触发低优先级中断	151
图 5-6: 中断执行过程中触发高优先级中断	152
图 5-7: 出栈过程中触发高优先级中断	152
图 5-8: 入栈过程中触发高优先级中断	153
图 6-1: CRC 模块框图	154
图 7-1: 通用定时器结构	163
图 7-2: 通用定时器运行流程图	164
图 8-1: 看门狗工作流程	171
图 9-1: PWM 系统总览	178
图 9-2: PWM 内部结构	179
图 9-3: PWM 时基产生子模块信号链	181
图 9-4: 时基计数波形	182
图 9-5: 时基相位同步波形	183
图 9-6: 计数比较子模块结构	184
图 9-7: 计数比较子模块波形	184

图 9-8: 行为限定子模块输入输出	185
图 9-9: 上下计数时的 PWM 双边沿对称波形	186
图 9-10: 上下计数时的 PWM 双边沿非对称波形	186
图 9-11: 死区控制子模块结构	187
图 9-12: 死区控制典型波形	188
图 9-13: 双边沿延迟死区控制波形	188
图 9-14: 信号封锁子模块结构	189
图 9-15: 信号封锁中断产生	191
图 9-16: 数字比较子模块结构	192
图 9-17: 事件处理	192
图 9-18: 事件触发子模块结构	194
图 10-1: ECAP 基本工作方式	276
图 10-2: 捕获模式和 APWM 模式	277
图 10-3: 捕获模式功能框图	277
图 10-4: 事件预分频控制	278
图 10-5: 事件预分频波形	278
图 10-6: 持续捕获和一次性捕获逻辑	279
图 10-7: 时间戳计数器及其同步	280
图 10-8: ECAP 模块中断	281
图 10-9: APWM 模式下 PRD 和 CMP 的有效值和影子值	282
图 10-10: APWM 模式波形（极性为高电平有效）	282
图 10-11: 上升沿检测和绝对时间戳捕获	283
图 10-12: 通过绝对事件戳捕获计算周期和占空	284
图 10-13: 通过计数器复位捕获周期	285
图 10-14: 通过计数器复位捕获周期和占空	286
图 10-15: 基于计数器复位的双引脚捕获	287
图 11-1: ADC 内部模块框图	306
图 11-2: 以 CH0 为例配置同步采样的信号流	309
图 11-3: ADC 转换结果的平均化处理	309
图 11-4: ADC 中断结构框图	310
图 11-5: ADC 中断	311
图 11-6: 顺序转换模式时序	312
图 11-7: 同时采样模式时序	312
图 11-8: ADC 转换曲线	313
图 11-9: 输入引脚悬空的检测（通过节点放电的故障注入）	314
图 11-10: 输入引脚悬空的检测（通过节点充电的故障注入）	315
图 13-1: DPGA 应用框图	344
图 13-2: DPGA 输入范围的要求	344
图 13-3: DPGA 偏置校准的路径	345
图 14-1: 差分转单端缓冲器应用框图	349
图 15-1: 比较器 COMP0 框图	350
图 15-2: 数字滤波器行为描述	351
图 15-3: DAC 结构框图	352
图 16-1: UART 模块框图	374
图 16-2: UART 数据帧格式示例	375
图 16-3: NRZ 编码示例 - 0b0100_1011	375

图 17-1: I2C 总线连接	393
图 17-2: I2C 协议基本格式	394
图 17-3: 开始与结束信号	394
图 17-4: 7 位地址格式	395
图 17-5: 10 位地址格式	395
图 17-6: SDA 数据保持时间	396
图 17-7: SDA 数据建立时间	396
图 18-1: SPI 模块框图	439
图 18-2: SPI 主机-从机连接示意图	441
图 18-3: FIFO 压缩和非压缩模式	442
图 18-4: SPI 时钟格式 0 (PHS = 0)	443
图 18-5: SPI 时钟格式 1 (PHS = 1)	444
图 18-6: 多帧传输 (Motorola)	445
图 19-1: Flash 存储器访问接口	455
图 19-2: Flash 模块逻辑结构	456
图 19-3: Flash 存储器读操作流程	458
图 19-4: Flash 存储器编程操作流程	459
图 19-5: Flash 存储器扇区擦除操作流程	460
图 19-6: Flash 存储器块擦除操作流程	461
图 19-7: Flash 存储器整片擦除操作流程	462
图 22-1: 典型应用框图	510
图 22-2: MCU 和 DC-DC 模块连接示意图 (QFN38)	511
图 22-3: MCU 向 DC-DC 发送数据 (PHA=0)	512
图 22-4: MCU 从 DC-DC 接受数据 (PHA=0)	512
图 22-5: 电流限制	514
图 22-6: 频率抖动行为	515
图 23-1: MCU 和预驱模块的连接关系	532

表格列表

表 1-1: 外设模块基地址	34
表 1-2: 固化的函数库入口地址向量表	35
表 1-3: 启动模式	37
表 2-1: POWER 模块基地址	41
表 2-2: POWER 寄存器列表	41
表 2-3: 电源状态寄存器 (PWRSTS) 位段定义	42
表 2-4: 电源状态寄存器 (PWRSTS) 位段描述	42
表 2-5: LDO 控制寄存器 (LDOCTL) 位段定义	43
表 2-6: LDO 控制寄存器 (LDOCTL) 位段描述	43
表 2-7: 电源中断事件标志寄存器 (PWRIEVTF) 位段定义	44
表 2-8: 电源中断事件标志寄存器 (PWRIEVTF) 位段描述	44
表 2-9: POWER 控制寄存器 (PWRCTL) 位段定义	45
表 2-10: POWER 控制寄存器 (PWRCTL) 位段描述	45
表 2-11: POWER 中断标志寄存器 (PWRIF) 位段定义	46
表 2-12: POWER 中断标志寄存器 (PWRIF) 位段描述	46
表 2-13: POWER 中断清除寄存器 (PWRIC) 位段定义	47
表 2-14: POWER 中断清除寄存器 (PWRIC) 位段描述	47
表 2-15: Power 中断使能寄存器 (PWRIE) 位段定义	48
表 2-16: Power 中断使能寄存器 (PWRIE) 位段描述	48
表 2-17: 3.3V BOD 控制寄存器 (BOD33CTL) 位段定义	49
表 2-18: 3.3V BOD 控制寄存器 (BOD33CTL) 位段描述	49
表 2-19: 1.2V BOD 控制寄存器 (BOD12CTL) 位段定义	51
表 2-20: 1.2V BOD 控制寄存器 (BOD12CTL) 位段描述	51
表 2-21: 睡眠模式使能寄存器 (SLEEPKEY) 位段定义	53
表 2-22: 睡眠模式使能寄存器 (SLEEPKEY) 位段描述	53
表 2-23: POWER 模块写使能寄存器 (PWRREGKEY) 位段定义	53
表 2-24: POWER 模块写使能寄存器 (PWRREGKEY) 位段描述	53
表 3-1: 输入分频比设置	56
表 3-2: 输出时钟频率和分频比	57
表 3-3: CLOCK 模块基地址	59
表 3-4: CLOCK 寄存器列表	59
表 3-5: 时钟状态寄存器 (CLKSTS) 位段定义	60
表 3-6: 时钟状态寄存器 (CLKSTS) 位段描述	60
表 3-7: RCO 控制寄存器 (RCOCTL) 位段定义	61
表 3-8: RCO 控制寄存器 (RCOCTL) 位段描述	61
表 3-9: PLL 控制寄存器 0 (PLLCTL0) 位段定义	62
表 3-10: PLL 控制寄存器 0 (PLLCTL0) 位段描述	62
表 3-11: PLL 控制寄存器 1 (PLLCTL1) 位段定义	63
表 3-12: PLL 控制寄存器 1 (PLLCTL1) 位段描述	63
表 3-13: 时钟检测控制寄存器 (CLKDETCTL) 位段定义	64
表 3-14: 时钟检测控制寄存器 (CLKDETCTL) 位段描述	64
表 3-15: 时钟检测计数阈值寄存器 (CLKDETCTH) 位段定义	65
表 3-16: 时钟检测计数阈值寄存器 (CLKDETCTH) 位段描述	65

表 3-17: 时钟检测计数值寄存器 (CLKDETCNT) 位段定义	66
表 3-18: 时钟检测计数值寄存器 (CLKDETCNT) 位段描述	66
表 3-19: 时钟中断标志寄存器 (CLKIF) 位段定义	67
表 3-20: 时钟中断标志寄存器 (CLKIF) 位段描述	67
表 3-21: 时钟中断清除寄存器 (CLKIC) 位段定义	68
表 3-22: 时钟中断清除寄存器 (CLKIC) 位段描述	68
表 3-23: 时钟中断使能寄存器 (CLKIE) 位段定义	69
表 3-24: 时钟中断使能寄存器 (CLKIE) 位段描述	69
表 3-25: 时钟封锁事件使能寄存器 (CLKTZE) 位段定义	70
表 3-26: 时钟封锁事件使能寄存器 (CLKTZE) 位段描述	70
表 3-27: SYSCLK0 控制寄存器 (SYSCLK0CTL) 位段定义	71
表 3-28: SYSCLK0 控制寄存器 (SYSCLK0CTL) 位段描述	71
表 3-29: HCLK 控制寄存器 (HCLKCTL) 位段定义	71
表 3-30: HCLK 控制寄存器 (HCLKCTL) 位段描述	71
表 3-31: ADC 时钟控制寄存器 (ADCCLKCTL) 位段定义	72
表 3-32: ADC 时钟控制寄存器 (ADCCLKCTL) 位段描述	72
表 3-33: PWM 时钟控制寄存器 (PWMCLKCTL) 位段定义	72
表 3-34: PWM 时钟控制寄存器 (PWMCLKCTL) 位段描述	72
表 3-35: ECAP 时钟控制寄存器 (ECAPCLKCTL) 位段定义	73
表 3-36: ECAP 时钟控制寄存器 (ECAPCLKCTL) 位段描述	73
表 3-37: Timer 0 时钟控制寄存器 (TMR0CLKCTL) 位段定义	73
表 3-38: Timer 0 时钟控制寄存器 (TMR0CLKCTL) 位段描述	73
表 3-39: Timer 1 时钟控制寄存器 (TMR1CLKCTL) 位段定义	74
表 3-40: Timer 1 时钟控制寄存器 (TMR1CLKCTL) 位段描述	74
表 3-41: SYSCLK1 控制寄存器 (SYSCLK1CTL) 位段定义	75
表 3-42: SYSCLK1 控制寄存器 (SYSCLK1CTL) 位段描述	75
表 3-43: PCLK 控制寄存器 (PCLKCTL) 位段定义	75
表 3-44: PCLK 控制寄存器 (PCLKCTL) 位段描述	75
表 3-45: 消抖时钟控制寄存器 (DGCLKCTL) 位段定义	76
表 3-46: 消抖时钟控制寄存器 (DGCLKCTL) 位段描述	76
表 3-47: UART 时钟控制寄存器 (UARTCLKCTL) 位段定义	77
表 3-48: UART 时钟控制寄存器 (UARTCLKCTL) 位段描述	77
表 3-49: SPI 时钟控制寄存器 (SPICLKCTL) 位段定义	77
表 3-50: SPI 时钟控制寄存器 (SPICLKCTL) 位段描述	77
表 3-51: I2C 时钟控制寄存器 (I2CCLKCTL) 位段定义	78
表 3-52: I2C 时钟控制寄存器 (I2CCLKCTL) 位段描述	78
表 3-53: WDT0 时钟控制寄存器 (WDT0CLKCTL) 位段定义	79
表 3-54: WDT0 时钟控制寄存器 (WDT0CLKCTL) 位段描述	79
表 3-55: WDT1 时钟控制寄存器 (WDT1CLKCTL) 位段定义	80
表 3-56: WDT1 时钟控制寄存器 (WDT1CLKCTL) 位段描述	80
表 3-57: CLOCK 模块写使能寄存器 (CLKREGKEY) 位段定义	81
表 3-58: CLOCK 模块写使能寄存器 (CLKREGKEY) 位段描述	81
表 4-1: I/O 引脚功能通道定义	85
表 4-2: PINMUX 模块基地址	91
表 4-3: PINMUX 寄存器列表	91
表 4-4: GPIO0 引脚控制寄存器 (GPIO0) 位段定义	92

表 4-5: GPIO0 引脚控制寄存器 (GPIO0) 位段描述.....	92
表 4-6: GPIO1 引脚控制寄存器 (GPIO1) 位段定义.....	94
表 4-7: GPIO1 引脚控制寄存器 (GPIO1) 位段描述.....	94
表 4-8: GPIO2 引脚控制寄存器 (GPIO2) 位段定义.....	96
表 4-9: GPIO2 引脚控制寄存器 (GPIO2) 位段描述.....	96
表 4-10: GPIO3 引脚控制寄存器 (GPIO3) 位段定义.....	98
表 4-11: GPIO3 引脚控制寄存器 (GPIO3) 位段描述.....	98
表 4-12: GPIO4 引脚控制寄存器 (GPIO4) 位段定义.....	100
表 4-13: GPIO4 引脚控制寄存器 (GPIO4) 位段描述.....	100
表 4-14: GPIO10 引脚控制寄存器 (GPIO10) 位段定义.....	102
表 4-15: GPIO10 引脚控制寄存器 (GPIO10) 位段描述.....	102
表 4-16: GPIO11 引脚控制寄存器 (GPIO11) 位段定义.....	104
表 4-17: GPIO11 引脚控制寄存器 (GPIO11) 位段描述.....	104
表 4-18: GPIO16 引脚控制寄存器 (GPIO16) 位段定义.....	106
表 4-19: GPIO16 引脚控制寄存器 (GPIO16) 位段描述.....	106
表 4-20: GPIO17 引脚控制寄存器 (GPIO17) 位段定义.....	108
表 4-21: GPIO17 引脚控制寄存器 (GPIO17) 位段描述.....	108
表 4-22: GPIO34 引脚控制寄存器 (GPIO34) 位段定义.....	110
表 4-23: GPIO34 引脚控制寄存器 (GPIO34) 位段描述.....	110
表 4-24: GPIO35 引脚控制寄存器 (GPIO35) 位段定义.....	112
表 4-25: GPIO35 引脚控制寄存器 (GPIO35) 位段描述.....	112
表 4-26: GPIO38 引脚控制寄存器 (GPIO38) 位段定义.....	114
表 4-27: GPIO38 引脚控制寄存器 (GPIO38) 位段描述.....	114
表 4-28: GPIO39 引脚控制寄存器 (GPIO39) 位段定义.....	116
表 4-29: GPIO39 引脚控制寄存器 (GPIO39) 位段描述.....	116
表 4-30: GPIO40 引脚控制寄存器 (GPIO40) 位段定义.....	118
表 4-31: GPIO40 引脚控制寄存器 (GPIO40) 位段描述.....	118
表 4-32: PINMUX 模块写使能寄存器 (PINMUXREGKEY) 位段定义.....	119
表 4-33: PINMUX 模块写使能寄存器 (PINMUXREGKEY) 位段描述.....	119
表 4-34: GPIO 模块基地址.....	120
表 4-35: GPIO 寄存器列表.....	120
表 4-36: GPIO 引脚电平寄存器 0 (GPLR0) 位段定义.....	122
表 4-37: GPIO 引脚电平寄存器 0 (GPLR0) 位段描述.....	122
表 4-38: GPIO 引脚电平寄存器 1 (GPLR1) 位段定义.....	122
表 4-39: GPIO 引脚电平寄存器 1 (GPLR1) 位段描述.....	122
表 4-40: GPIO 引脚方向寄存器 0 (GPDR0) 位段定义.....	123
表 4-41: GPIO 引脚方向寄存器 0 (GPDR0) 位段描述.....	123
表 4-42: GPIO 引脚方向寄存器 1 (GPDR1) 位段定义.....	123
表 4-43: GPIO 引脚方向寄存器 1 (GPDR1) 位段描述.....	123
表 4-44: GPIO 引脚输出置 1 寄存器 0 (GSLR0) 位段定义.....	124
表 4-45: GPIO 引脚输出置 1 寄存器 0 (GSLR0) 位段描述.....	124
表 4-46: GPIO 引脚输出置 1 寄存器 1 (GSLR1) 位段定义.....	124
表 4-47: GPIO 引脚输出置 1 寄存器 1 (GSLR1) 位段描述.....	124
表 4-48: GPIO 引脚输出清除寄存器 0 (GCLR0) 位段定义.....	125
表 4-49: GPIO 引脚输出清除寄存器 0 (GCLR0) 位段描述.....	125
表 4-50: GPIO 引脚输出清除寄存器 1 (GCLR1) 位段定义.....	125

表 4-51: GPIO 引脚输出清除寄存器 1 (GCLR1) 位段描述	125
表 4-52: GPIO 上升沿检测使能寄存器 0 (GRER0) 位段定义	126
表 4-53: GPIO 上升沿检测使能寄存器 0 (GRER0) 位段描述	126
表 4-54: GPIO 上升沿检测使能寄存器 1 (GRER1) 位段定义	126
表 4-55: GPIO 上升沿检测使能寄存器 1 (GRER1) 位段描述	126
表 4-56: GPIO 下降沿检测使能寄存器 0 (GFER0) 位段定义	127
表 4-57: GPIO 下降沿检测使能寄存器 0 (GFER0) 位段描述	127
表 4-58: GPIO 下降沿检测使能寄存器 1 (GFER1) 位段定义	127
表 4-59: GPIO 下降沿检测使能寄存器 1 (GFER1) 位段描述	127
表 4-60: GPIO 边沿检测状态寄存器 0 (GEDR0) 位段定义	128
表 4-61: GPIO 边沿检测状态寄存器 0 (GEDR0) 位段描述	128
表 4-62: GPIO 边沿检测状态寄存器 1 (GEDR1) 位段定义	128
表 4-63: GPIO 边沿检测状态寄存器 1 (GEDR1) 位段描述	128
表 4-64: GPIO 引脚按位设置输出寄存器 0 (GSDR0) 位段定义	129
表 4-65: GPIO 引脚按位设置输出寄存器 0 (GSDR0) 位段描述	129
表 4-66: GPIO 引脚按位设置输出寄存器 1 (GSDR1) 位段定义	129
表 4-67: GPIO 引脚按位设置输出寄存器 1 (GSDR1) 位段描述	129
表 4-68: GPIO 引脚按位设置输入寄存器 0 (GCDR0) 位段定义	130
表 4-69: GPIO 引脚按位设置输入寄存器 0 (GCDR0) 位段描述	130
表 4-70: GPIO 引脚按位设置输入寄存器 1 (GCDR1) 位段定义	130
表 4-71: GPIO 引脚按位设置输入寄存器 1 (GCDR1) 位段描述	130
表 4-72: GPIO 上升沿检测按位使能寄存器 0 (GSRER0) 位段定义	131
表 4-73: GPIO 上升沿检测按位使能寄存器 0 (GSRER0) 位段描述	131
表 4-74: GPIO 上升沿检测按位使能寄存器 1 (GSRER1) 位段定义	131
表 4-75: GPIO 上升沿检测按位使能寄存器 1 (GSRER1) 位段描述	131
表 4-76: GPIO 上升沿检测按位关闭寄存器 0 (GCRER0) 位段定义	132
表 4-77: GPIO 上升沿检测按位关闭寄存器 0 (GCRER0) 位段描述	132
表 4-78: GPIO 上升沿检测按位关闭寄存器 1 (GCRER1) 位段定义	132
表 4-79: GPIO 上升沿检测按位关闭寄存器 1 (GCRER1) 位段描述	132
表 4-80: GPIO 下降沿检测按位使能寄存器 0 (GSFER0) 位段定义	133
表 4-81: GPIO 下降沿检测按位使能寄存器 0 (GSFER0) 位段描述	133
表 4-82: GPIO 下降沿检测按位使能寄存器 1 (GSFER1) 位段定义	133
表 4-83: GPIO 下降沿检测按位使能寄存器 1 (GSFER1) 位段描述	133
表 4-84: GPIO 下降沿检测按位关闭寄存器 0 (GCFER0) 位段定义	134
表 4-85: GPIO 下降沿检测按位关闭寄存器 0 (GCFER0) 位段描述	134
表 4-86: GPIO 下降沿检测按位关闭寄存器 1 (GCFER1) 位段定义	134
表 4-87: GPIO 下降沿检测按位关闭寄存器 1 (GCFER1) 位段描述	134
表 4-88: GPIO 边沿中断使能寄存器 0 (GEIE0) 位段定义	135
表 4-89: GPIO 边沿中断使能寄存器 0 (GEIE0) 位段描述	135
表 4-90: GPIO 边沿中断使能寄存器 1 (GEIE1) 位段定义	135
表 4-91: GPIO 边沿中断使能寄存器 1 (GEIE1) 位段描述	135
表 4-92: GPIO 电平中断标志寄存器 0 (GLIF0) 位段定义	136
表 4-93: GPIO 电平中断标志寄存器 0 (GLIF0) 位段描述	136
表 4-94: GPIO 电平中断标志寄存器 1 (GLIF1) 位段定义	136
表 4-95: GPIO 电平中断标志寄存器 1 (GLIF1) 位段描述	136
表 4-96: GPIO 电平中断使能寄存器 0 (GLIE0) 位段定义	137

表 4-97: GPIO 电平中断使能寄存器 0 (GLIE0) 位段描述.....	137
表 4-98: GPIO 电平中断使能寄存器 1 (GLIE1) 位段定义.....	137
表 4-99: GPIO 电平中断使能寄存器 1 (GLIE1) 位段描述.....	137
表 4-100: GPIO 电平中断清除寄存器 0 (GLIC0) 位段定义.....	138
表 4-101: GPIO 电平中断清除寄存器 0 (GLIC0) 位段描述.....	138
表 4-102: GPIO 电平中断清除寄存器 1 (GLIC1) 位段定义.....	138
表 4-103: GPIO 电平中断清除寄存器 1 (GLIC1) 位段描述.....	138
表 4-104: GPIO 电平中断强制寄存器 0 (GLIFRC0) 位段定义.....	139
表 4-105: GPIO 电平中断强制寄存器 0 (GLIFRC0) 位段描述.....	139
表 4-106: GPIO 电平中断强制寄存器 1 (GLIFRC1) 位段定义.....	139
表 4-107: GPIO 电平中断强制寄存器 1 (GLIFRC1) 位段描述.....	139
表 4-108: GPIO 电平中断极性寄存器 0 (GLIPOL0) 位段定义	140
表 4-109: GPIO 电平中断极性寄存器 0 (GLIPOL0) 位段描述	140
表 4-110: GPIO 电平中断极性寄存器 1 (GLIPOL1) 位段定义	140
表 4-111: GPIO 电平中断极性寄存器 1 (GLIPOL1) 位段描述	140
表 4-112: GPIO 全局中断标志寄存器 (GIF) 位段定义.....	141
表 4-113: GPIO 全局中断标志寄存器 (GIF) 位段描述.....	141
表 4-114: GPIO 全局中断清除寄存器 (GIC) 位段定义	142
表 4-115: GPIO 全局中断清除寄存器 (GIC) 位段描述	142
表 4-116: GPIO 模块写使能寄存器 (GPIOREGKEY) 位段定义	142
表 4-117: GPIO 模块写使能寄存器 (GPIOREGKEY) 位段描述	142
表 4-118: GPIO 上升沿中断标志寄存器 0 (GREIF0) 位段定义	143
表 4-119: GPIO 上升沿中断标志寄存器 0 (GREIF0) 位段描述	143
表 4-120: GPIO 上升沿中断标志寄存器 1 (GREIF1) 位段定义	143
表 4-121: GPIO 上升沿中断标志寄存器 1 (GREIF1) 位段描述	143
表 4-122: GPIO 下降沿中断标志寄存器 0 (GFEIF0) 位段定义.....	144
表 4-123: GPIO 下降沿中断标志寄存器 0 (GFEIF0) 位段描述.....	144
表 4-124: GPIO 下降沿中断标志寄存器 1 (GFEIF1) 位段定义.....	144
表 4-125: GPIO 下降沿中断标志寄存器 1 (GFEIF1) 位段描述.....	144
表 5-1: SPD1121 的中断向量表.....	146
表 5-2: 中断优先级寄存器 (IP/SHP)	148
表 5-3: 优先级分组寄存器 (AIRCR. PRIGROUP)	148
表 5-4: 配置中断优先级寄存器 (IP/SHP)	148
表 6-1: CRC 模块基地址.....	156
表 6-2: CRC 寄存器列表.....	156
表 6-3: CRC 中断标志寄存器 (CRCIF) 位段定义.....	157
表 6-4: CRC 中断标志寄存器 (CRCIF) 位段描述.....	157
表 6-5: CRC 中断事件标志寄存器 (CRCIEVTF) 位段定义	158
表 6-6: CRC 中断事件标志寄存器 (CRCIEVTF) 位段描述	158
表 6-7: CRC 中断清除寄存器 (CRCIC) 位段定义	158
表 6-8: CRC 中断清除寄存器 (CRCIC) 位段描述	158
表 6-9: CRC 中断使能寄存器 (CRCIE) 位段定义	159
表 6-10: CRC 中断使能寄存器 (CRCIE) 位段描述	159
表 6-11: CRC 控制寄存器 (CRCCTL) 位段定义.....	160
表 6-12: CRC 控制寄存器 (CRCCTL) 位段描述.....	160
表 6-13: CRC 数据流长度寄存器 (CRCSTRLEN) 位段定义	161

表 6-14: CRC 数据流长度寄存器 (CRCSTRLEN) 位段描述	161
表 6-15: CRC 数据流输入寄存器 (CRCSTRIN) 位段定义	161
表 6-16: CRC 数据流输入寄存器 (CRCSTRIN) 位段描述	161
表 6-17: CRC 结果寄存器 (CRCRESULT) 位段定义	162
表 6-18: CRC 结果寄存器 (CRCRESULT) 位段描述	162
表 7-1: TIMER 模块基地址	165
表 7-2: TIMER 寄存器列表	165
表 7-3: 定时器控制寄存器 (TMRCTL) 位段定义	166
表 7-4: 定时器控制寄存器 (TMRCTL) 位段描述	166
表 7-5: 定时器计数器值寄存器 (TMRCNT) 位段定义	167
表 7-6: 定时器计数器值寄存器 (TMRCNT) 位段描述	167
表 7-7: 定时器超时重载寄存器 (TMRRELOAD) 位段定义	167
表 7-8: 定时器超时重载寄存器 (TMRRELOAD) 位段描述	167
表 7-9: 定时器中断标志寄存器 (TMRIF) 位段定义	168
表 7-10: 定时器中断标志寄存器 (TMRIF) 位段描述	168
表 7-11: 定时器中断事件标志寄存器 (TMRIEVTF) 位段定义	168
表 7-12: 定时器中断事件标志寄存器 (TMRIEVTF) 位段描述	168
表 7-13: 定时器中断使能寄存器 (TMRIE) 位段定义	169
表 7-14: 定时器中断使能寄存器 (TMRIE) 位段描述	169
表 7-15: 定时器中断强制寄存器 (TMRIFRC) 位段定义	169
表 7-16: 定时器中断强制寄存器 (TMRIFRC) 位段描述	169
表 7-17: 定时器中断清除寄存器 (TMRIC) 位段定义	170
表 7-18: 定时器中断清除寄存器 (TMRIC) 位段描述	170
表 8-1: WDT 模块基地址	173
表 8-2: WDT 寄存器列表	173
表 8-3: 看门狗定时器超时加载寄存器 (WDTLOAD) 位段定义	174
表 8-4: 看门狗定时器超时加载寄存器 (WDTLOAD) 位段描述	174
表 8-5: 看门狗定时器计数器当前值寄存器 (WDCNT) 位段定义	174
表 8-6: 看门狗定时器计数器当前值寄存器 (WDCNT) 位段描述	174
表 8-7: 看门狗定时器控制寄存器 (WDTCTL) 位段定义	175
表 8-8: 看门狗定时器控制寄存器 (WDTCTL) 位段描述	175
表 8-9: 看门狗定时器中断清除寄存器 (WDTIC) 位段定义	176
表 8-10: 看门狗定时器中断清除寄存器 (WDTIC) 位段描述	176
表 8-11: 看门狗定时器中断事件标志寄存器 (WDTIEVTF) 位段定义	176
表 8-12: 看门狗定时器中断事件标志寄存器 (WDTIEVTF) 位段描述	176
表 8-13: 看门狗定时器中断标志寄存器 (WDTIF) 位段定义	177
表 8-14: 看门狗定时器中断标志寄存器 (WDTIF) 位段描述	177
表 8-15: 看门狗定时器写使能寄存器 (WDTREGKEY) 位段定义	177
表 8-16: 看门狗定时器写使能寄存器 (WDTREGKEY) 位段描述	177
表 9-1: 向上计数时的行为限定事件优先级	185
表 9-2: 向下计数时的行为限定事件优先级	186
表 9-3: 上下计数时的行为限定事件优先级	186
表 9-4: 典型的死区控制工作模式	187
表 9-5: 影响 PWM 最终输出波形的事件优先级	191
表 9-6: PWM 模块基地址	195
表 9-7: PWM 寄存器列表	195

表 9-8: 影子状态寄存器 (SHADOWSTS) 位段定义	197
表 9-9: 影子状态寄存器 (SHADOWSTS) 位段描述	197
表 9-10: 时基控制寄存器 (TBCTL) 位段定义	199
表 9-11: 时基控制寄存器 (TBCTL) 位段描述	199
表 9-12: 时基周期寄存器 (TBPRDS) 位段定义	201
表 9-13: 时基周期寄存器 (TBPRDS) 位段描述	201
表 9-14: 时基周期有效值寄存器 (TBPRD) 位段定义	202
表 9-15: 时基周期有效值寄存器 (TBPRD) 位段描述	202
表 9-16: 时基相位寄存器 (TBPHS) 位段定义	203
表 9-17: 时基相位寄存器 (TBPHS) 位段描述	203
表 9-18: 时基计数值寄存器 (TBCNT) 位段定义	203
表 9-19: 时基计数值寄存器 (TBCNT) 位段描述	203
表 9-20: 时基状态寄存器 (TBFLG) 位段定义	204
表 9-21: 时基状态寄存器 (TBFLG) 位段描述	204
表 9-22: 时基状态清除寄存器 (TBCLR) 位段定义	205
表 9-23: 时基状态清除寄存器 (TBCLR) 位段描述	205
表 9-24: 计数比较控制寄存器 (CMPCTL) 位段定义	206
表 9-25: 计数比较控制寄存器 (CMPCTL) 位段描述	206
表 9-26: 计数比较阈值 A 寄存器 (CMPAS) 位段定义	208
表 9-27: 计数比较阈值 A 寄存器 (CMPAS) 位段描述	208
表 9-28: 计数比较阈值 A 有效值寄存器 (CMPA) 位段定义	208
表 9-29: 计数比较阈值 A 有效值寄存器 (CMPA) 位段描述	208
表 9-30: 计数比较阈值 B 寄存器 (CMPBS) 位段定义	209
表 9-31: 计数比较阈值 B 寄存器 (CMPBS) 位段描述	209
表 9-32: 计数比较阈值 B 寄存器 (CMPB) 位段定义	209
表 9-33: 计数比较阈值 B 寄存器 (CMPB) 位段描述	209
表 9-34: 计数比较阈值 C 寄存器 (CMPCS) 位段定义	210
表 9-35: 计数比较阈值 C 寄存器 (CMPCS) 位段描述	210
表 9-36: 计数比较阈值 C 有效值寄存器 (CMPC) 位段定义	210
表 9-37: 计数比较阈值 C 有效值寄存器 (CMPC) 位段描述	210
表 9-38: 行为限定控制寄存器 (AQCTL) 位段定义	211
表 9-39: 行为限定控制寄存器 (AQCTL) 位段描述	211
表 9-40: 行为限定 A 路输出控制寄存器 (AQCTLAS) 位段定义	213
表 9-41: 行为限定 A 路输出控制寄存器 (AQCTLAS) 位段描述	213
表 9-42: 行为限定 A 路输出控制有效值寄存器 (AQCTLA) 位段定义	215
表 9-43: 行为限定 A 路输出控制有效值寄存器 (AQCTLA) 位段描述	215
表 9-44: 行为限定 B 路输出控制寄存器 (AQCTLBS) 位段定义	217
表 9-45: 行为限定 B 路输出控制寄存器 (AQCTLBS) 位段描述	217
表 9-46: 行为限定 B 路输出控制有效值寄存器 (AQCTLB) 位段定义	219
表 9-47: 行为限定 B 路输出控制有效值寄存器 (AQCTLB) 位段描述	219
表 9-48: 行为限定软件强制寄存器 (AQSFRC) 位段定义	220
表 9-49: 行为限定软件强制寄存器 (AQSFRC) 位段描述	220
表 9-50: 行为限定软件持续强制寄存器 (AQCSFRCS) 位段定义	221
表 9-51: 行为限定软件持续强制寄存器 (AQCSFRCS) 位段描述	221
表 9-52: 行为限定软件持续强制有效值寄存器 (AQCSFRC) 位段定义	222
表 9-53: 行为限定软件持续强制有效值寄存器 (AQCSFRC) 位段描述	222

表 9-54: 死区产生控制寄存器 (DBCTLS) 位段定义	223
表 9-55: 死区产生控制寄存器 (DBCTLS) 位段描述	223
表 9-56: 死区产生控制有效值寄存器 (DBCTL) 位段定义	226
表 9-57: 死区产生控制有效值寄存器 (DBCTL) 位段描述	226
表 9-58: 死区产生上升沿延迟量寄存器 (DBREDS) 位段定义	228
表 9-59: 死区产生上升沿延迟量寄存器 (DBREDS) 位段描述	228
表 9-60: 死区产生上升沿延迟量有效值寄存器 (DBRED) 位段定义	228
表 9-61: 死区产生上升沿延迟量有效值寄存器 (DBRED) 位段描述	228
表 9-62: 死区产生下降沿延迟量寄存器 (DBFEDS) 位段定义	229
表 9-63: 死区产生下降沿延迟量寄存器 (DBFEDS) 位段描述	229
表 9-64: 死区产生下降沿延迟量有效值寄存器 (DBFED) 位段定义	229
表 9-65: 死区产生下降沿延迟量有效值寄存器 (DBFED) 位段描述	229
表 9-66: 封锁事件选择寄存器 (TZSEL) 位段定义	230
表 9-67: 封锁事件选择寄存器 (TZSEL) 位段描述	230
表 9-68: 封锁状态寄存器 (TZFLG) 位段定义	232
表 9-69: 封锁状态寄存器 (TZFLG) 位段描述	232
表 9-70: 封锁状态清除寄存器 (TZCLR) 位段定义	234
表 9-71: 封锁状态清除寄存器 (TZCLR) 位段描述	234
表 9-72: 数字比较封锁事件选择寄存器 (TZDCSEL) 位段定义	236
表 9-73: 数字比较封锁事件选择寄存器 (TZDCSEL) 位段描述	236
表 9-74: 封锁 A 路输出控制寄存器 (TZACTL) 位段定义	237
表 9-75: 封锁 A 路输出控制寄存器 (TZACTL) 位段描述	237
表 9-76: 封锁 B 路输出控制寄存器 (TZBCTL) 位段定义	239
表 9-77: 封锁 B 路输出控制寄存器 (TZBCTL) 位段描述	239
表 9-78: 封锁中断标志寄存器 (TZIF) 位段定义	240
表 9-79: 封锁中断标志寄存器 (TZIF) 位段描述	240
表 9-80: 封锁中断清除寄存器 (TZIC) 位段定义	242
表 9-81: 封锁中断清除寄存器 (TZIC) 位段描述	242
表 9-82: 封锁中断使能寄存器 (TZIE) 位段定义	243
表 9-83: 封锁中断使能寄存器 (TZIE) 位段描述	243
表 9-84: 封锁中断强制寄存器 (TZFRC) 位段定义	244
表 9-85: 封锁中断强制寄存器 (TZFRC) 位段描述	244
表 9-86: 数字比较 DCAL 事件选择寄存器 (DCALTRIPSEL) 位段定义	245
表 9-87: 数字比较 DCAL 事件选择寄存器 (DCALTRIPSEL) 位段描述	245
表 9-88: 数字比较 DCAH 事件选择寄存器 (DCAHTRIPSEL) 位段定义	247
表 9-89: 数字比较 DCAH 事件选择寄存器 (DCAHTRIPSEL) 位段描述	247
表 9-90: 数字比较 DCBL 事件选择寄存器 (DCBLTRIPSEL) 位段定义	249
表 9-91: 数字比较 DCBL 事件选择寄存器 (DCBLTRIPSEL) 位段描述	249
表 9-92: 数字比较 DCBH 事件选择寄存器 (DCBHTRIPSEL) 位段定义	251
表 9-93: 数字比较 DCBH 事件选择寄存器 (DCBHTRIPSEL) 位段描述	251
表 9-94: 数字比较 A 控制寄存器 (DCACTL) 位段定义	252
表 9-95: 数字比较 A 控制寄存器 (DCACTL) 位段描述	252
表 9-96: 数字比较 B 控制寄存器 (DCBCTL) 位段定义	253
表 9-97: 数字比较 B 控制寄存器 (DCBCTL) 位段描述	253
表 9-98: 事件触发控制寄存器 (ETCTL) 位段定义	254
表 9-99: 事件触发控制寄存器 (ETCTL) 位段描述	254

表 9-100: 事件触发预分频寄存器 (ETPS) 位段定义	256
表 9-101: 事件触发预分频寄存器 (ETPS) 位段描述	256
表 9-102: 事件触发标志寄存器 (ETFLG) 位段定义	258
表 9-103: 事件触发标志寄存器 (ETFLG) 位段描述	258
表 9-104: 事件触发标志清除寄存器 (ETCLR) 位段定义	259
表 9-105: 事件触发标志清除寄存器 (ETCLR) 位段描述	259
表 9-106: 事件触发软件强制寄存器 (ETFRC) 位段定义	260
表 9-107: 事件触发软件强制寄存器 (ETFRC) 位段描述	260
表 9-108: PWM 模块写使能寄存器 (PWMREGKEY) 位段定义	261
表 9-109: PWM 模块写使能寄存器 (PWMREGKEY) 位段描述	261
表 9-110: PWMCFG 模块基地址	262
表 9-111: PWMCFG 寄存器列表	262
表 9-112: TZ0 信号源控制寄存器 (TZ0SRCCTL) 位段定义	263
表 9-113: TZ0 信号源控制寄存器 (TZ0SRCCTL) 位段描述	263
表 9-114: TZ1 信号源控制寄存器 (TZ1SRCCTL) 位段定义	264
表 9-115: TZ1 信号源控制寄存器 (TZ1SRCCTL) 位段描述	264
表 9-116: TZ2 信号源控制寄存器 (TZ2SRCCTL) 位段定义	264
表 9-117: TZ2 信号源控制寄存器 (TZ2SRCCTL) 位段描述	264
表 9-118: 全局 PWM 软件强制同步寄存器 (FRCSYNC) 位段定义	265
表 9-119: 全局 PWM 软件强制同步寄存器 (FRCSYNC) 位段描述	265
表 9-120: GPIO 触发 PWMSYNCI 信号源控制寄存器 (GPIOSYNCCTL) 位段定义	266
表 9-121: GPIO 触发 PWMSYNCI 信号源控制寄存器 (GPIOSYNCCTL) 位段描述	266
表 9-122: GPIO 触发 PWMSYNCI 使能寄存器 (GPIOSYNCEN) 位段定义	267
表 9-123: GPIO 触发 PWMSYNCI 使能寄存器 (GPIOSYNCEN) 位段描述	267
表 9-124: Timer 0 触发 PWMSYNCI 使能寄存器 (TMR0SYNCEN) 位段定义	268
表 9-125: Timer 0 触发 PWMSYNCI 使能寄存器 (TMR0SYNCEN) 位段描述	268
表 9-126: Timer 1 触发 PWMSYNCI 使能寄存器 (TMR1SYNCEN) 位段定义	269
表 9-127: Timer 1 触发 PWMSYNCI 使能寄存器 (TMR1SYNCEN) 位段描述	269
表 9-128: ECAP 强制 PWMSYNCI 使能寄存器 (ECAPSYNCEN) 位段定义	270
表 9-129: ECAP 强制 PWMSYNCI 使能寄存器 (ECAPSYNCEN) 位段描述	270
表 9-130: PWMSYNCO 输出控制寄存器 (SYNCOCTL) 位段定义	271
表 9-131: PWMSYNCO 输出控制寄存器 (SYNCOCTL) 位段描述	271
表 9-132: PWMREQA 输出控制寄存器 (REQAOCTL) 位段定义	272
表 9-133: PWMREQA 输出控制寄存器 (REQAOCTL) 位段描述	272
表 9-134: PWMREQB 输出控制寄存器 (REQBOCTL) 位段定义	273
表 9-135: PWMREQB 输出控制寄存器 (REQBOCTL) 位段描述	273
表 9-136: PWMREQC 输出控制寄存器 (REQCOCTL) 位段定义	274
表 9-137: PWMREQC 输出控制寄存器 (REQCOCTL) 位段描述	274
表 9-138: PWMCFG 模块写使能寄存器 (PWMCFGREGKEY) 位段定义	275
表 9-139: PWMCFG 模块写使能寄存器 (PWMCFGREGKEY) 位段描述	275
表 10-1: ECAP 模块基地址	288
表 10-2: ECAP 寄存器列表	288
表 10-3: 捕获信号输入控制寄存器 (CAPSRCCTL) 位段定义	289
表 10-4: 捕获信号输入控制寄存器 (CAPSRCCTL) 位段描述	289
表 10-5: 同步信号输入控制寄存器 (CAPSYNCICTL) 位段定义	290
表 10-6: 同步信号输入控制寄存器 (CAPSYNCICTL) 位段描述	290

表 10-7: 时间戳计数寄存器 (CAPTSCNT) 位段定义	290
表 10-8: 时间戳计数寄存器 (CAPTSCNT) 位段描述	290
表 10-9: 时间戳相位寄存器 (CAPCNTPHS) 位段定义	291
表 10-10: 时间戳相位寄存器 (CAPCNTPHS) 位段描述	291
表 10-11: 捕获值 0 寄存器 (CAP0) 位段定义	291
表 10-12: 捕获值 0 寄存器 (CAP0) 位段描述	291
表 10-13: 捕获值 1 寄存器 (CAP1) 位段定义	292
表 10-14: 捕获值 1 寄存器 (CAP1) 位段描述	292
表 10-15: 捕获值 2 寄存器 (CAP2) 位段定义	293
表 10-16: 捕获值 2 寄存器 (CAP2) 位段描述	293
表 10-17: 捕获值 3 寄存器 (CAP3) 位段定义	294
表 10-18: 捕获值 3 寄存器 (CAP3) 位段描述	294
表 10-19: 捕获控制寄存器 (CAPCTL) 位段定义	295
表 10-20: 捕获控制寄存器 (CAPCTL) 位段描述	295
表 10-21: 捕获中断标志寄存器 (CAPIF) 位段定义	298
表 10-22: 捕获中断标志寄存器 (CAPIF) 位段描述	298
表 10-23: 捕获中断使能寄存器 (CAPIE) 位段定义	300
表 10-24: 捕获中断使能寄存器 (CAPIE) 位段描述	300
表 10-25: 捕获中断清除寄存器 (CAPIC) 位段定义	301
表 10-26: 捕获中断清除寄存器 (CAPIC) 位段描述	301
表 10-27: 捕获中断强制寄存器 (CAPIFRC) 位段定义	302
表 10-28: 捕获中断强制寄存器 (CAPIFRC) 位段描述	302
表 10-29: ECAP 模块写保护寄存器 (CAPREGKEY) 位段定义	303
表 10-30: ECAP 模块写保护寄存器 (CAPREGKEY) 位段描述	303
表 10-31: 捕获事件标志寄存器 (CAPIEVTF) 位段定义	304
表 10-32: 捕获事件标志寄存器 (CAPIEVTF) 位段描述	304
表 11-1: 请求信号选择	307
表 11-2: 采样模式描述	308
表 11-3: ADC 通道选择	308
表 11-4: 平均化次数选择	310
表 11-5: ADC 模块基地址	317
表 11-6: ADC 寄存器列表	317
表 11-7: ADC 中断标志寄存器 (CHIF) 位段定义	318
表 11-8: ADC 中断标志寄存器 (CHIF) 位段描述	318
表 11-9: ADC 中断清除寄存器 (CHIC) 位段定义	319
表 11-10: ADC 中断清除寄存器 (CHIC) 位段描述	319
表 11-11: ADC 中断溢出标志寄存器 (CHIOVF) 位段定义	320
表 11-12: ADC 中断溢出标志寄存器 (CHIOVF) 位段描述	320
表 11-13: ADC 中断溢出清除寄存器 (CHIOVFC) 位段定义	321
表 11-14: ADC 中断溢出清除寄存器 (CHIOVFC) 位段描述	321
表 11-15: ADC 中断使能寄存器 (CHIE) 位段定义	322
表 11-16: ADC 中断使能寄存器 (CHIE) 位段描述	322
表 11-17: ADC 通道请求标志寄存器 (CHREQFLG) 位段定义	323
表 11-18: ADC 通道请求标志寄存器 (CHREQFLG) 位段描述	323
表 11-19: ADC 通道请求强制寄存器 (CHREQFRC) 位段定义	325
表 11-20: ADC 通道请求强制寄存器 (CHREQFRC) 位段描述	325

表 11-21: ADC REQ 溢出寄存器 (CHREQOVF) 位段定义	326
表 11-22: ADC REQ 溢出寄存器 (CHREQOVF) 位段描述	326
表 11-23: ADC REQ 溢出清除寄存器 (CHREQOVFC) 位段定义	327
表 11-24: ADC REQ 溢出清除寄存器 (CHREQOVFC) 位段描述	327
表 11-25: ADC 通道连接使能寄存器 (CHLINKEN) 位段定义	328
表 11-26: ADC 通道连接使能寄存器 (CHLINKEN) 位段描述	328
表 11-27: ADC 通道连接寄存器 (CHLINK) 位段定义	329
表 11-28: ADC 通道连接寄存器 (CHLINK) 位段描述	329
表 11-29: ADC 外部 REQ 控制寄存器 (EXTREQCTL) 位段定义	330
表 11-30: ADC 外部 REQ 控制寄存器 (EXTREQCTL) 位段描述	330
表 11-31: ADC 通道 x (x=[0:7]) 控制寄存器 (CHCTL) 位段定义	331
表 11-32: ADC 通道 x (x=[0:7]) 控制寄存器 (CHCTL) 位段描述	331
表 11-33: ADC 通道 x (x=[0:7]) 偏移校准寄存器 (CHOFFSET) 位段定义	333
表 11-34: ADC 通道 x (x=[0:7]) 偏移校准寄存器 (CHOFFSET) 位段描述	333
表 11-35: ADC 通道 x (x=[0:7]) 增益校准寄存器 (CHGAIN) 位段定义	333
表 11-36: ADC 通道 x (x=[0:7]) 增益校准寄存器 (CHGAIN) 位段描述	333
表 11-37: ADC SHx (x=[0:1]) 偏移校准寄存器 (SHOFFSET) 位段定义	334
表 11-38: ADC SHx (x=[0:1]) 偏移校准寄存器 (SHOFFSET) 位段描述	334
表 11-39: ADC SHx (x=[0:1]) 增益校准寄存器 (SHGAIN) 位段定义	334
表 11-40: ADC SHx (x=[0:1]) 增益校准寄存器 (SHGAIN) 位段描述	334
表 11-41: ADC 状态寄存器 (SHFLG) 位段定义	335
表 11-42: ADC 状态寄存器 (SHFLG) 位段描述	335
表 11-43: ADC 状态清除寄存器 (SHFLGCLR) 位段定义	336
表 11-44: ADC 状态清除寄存器 (SHFLGCLR) 位段描述	336
表 11-45: ADC 控制寄存器 (ADCCTL) 位段定义	337
表 11-46: ADC 控制寄存器 (ADCCTL) 位段描述	337
表 11-47: ADC SHx (x=[0:1]) 原始码值寄存器 (SHRESULT) 位段定义	339
表 11-48: ADC SHx (x=[0:1]) 原始码值寄存器 (SHRESULT) 位段描述	339
表 11-49: ADC 通道 x (x=[0:7]) 转换结果寄存器 (CHRESULT) 位段定义	339
表 11-50: ADC 通道 x (x=[0:7]) 转换结果寄存器 (CHRESULT) 位段描述	339
表 11-51: 温度传感器控制寄存器 (TSENSCTL) 位段定义	340
表 11-52: 温度传感器控制寄存器 (TSENSCTL) 位段描述	340
表 11-53: ADC 模块写使能寄存器 (ADCREGKEY) 位段定义	341
表 11-54: ADC 模块写使能寄存器 (ADCREGKEY) 位段描述	341
表 13-1: 增益及差分输入/输出范围	343
表 13-2: DPGA 模块基地址	346
表 13-3: DPGA 寄存器列表	346
表 13-4: DPGA 控制寄存器 (DPGACTL) 位段定义	347
表 13-5: DPGA 控制寄存器 (DPGACTL) 位段描述	347
表 13-6: PGA 模块写使能寄存器 (PGAREGKEY) 位段定义	348
表 13-7: PGA 模块写使能寄存器 (PGAREGKEY) 位段描述	348
表 15-1: 比较器输入通道选择	351
表 15-2: COMP 模块基地址	353
表 15-3: COMP 寄存器列表	353
表 15-4: 比较器滤波输出寄存器 (COMPFLT) 位段定义	354
表 15-5: 比较器滤波输出寄存器 (COMPFLT) 位段描述	354

表 15-6: 比较器状态寄存器 (COMPFLG) 位段定义	355
表 15-7: 比较器状态寄存器 (COMPFLG) 位段描述	355
表 15-8: 比较器状态清除寄存器 (COMPCLR) 位段定义	356
表 15-9: 比较器状态清除寄存器 (COMPCLR) 位段描述	356
表 15-10: 比较器 0 控制寄存器 (COMP0CTL) 位段定义	357
表 15-11: 比较器 0 控制寄存器 (COMP0CTL) 位段描述	357
表 15-12: COMP0L 控制寄存器 (COMP0LCTL) 位段定义	359
表 15-13: COMP0L 控制寄存器 (COMP0LCTL) 位段描述	359
表 15-14: COMP0H 控制寄存器 (COMP0HCTL) 位段定义	361
表 15-15: COMP0H 控制寄存器 (COMP0HCTL) 位段描述	361
表 15-16: 比较器 1 控制寄存器 (COMP1CTL) 位段定义	363
表 15-17: 比较器 1 控制寄存器 (COMP1CTL) 位段描述	363
表 15-18: COMP1H 控制寄存器 (COMP1HCTL) 位段定义	364
表 15-19: COMP1H 控制寄存器 (COMP1HCTL) 位段描述	364
表 15-20: DAC0 控制寄存器 (DAC0CTL) 位段定义	366
表 15-21: DAC0 控制寄存器 (DAC0CTL) 位段描述	366
表 15-22: DAC0 码值影子寄存器 (DAC0CODES) 位段定义	367
表 15-23: DAC0 码值影子寄存器 (DAC0CODES) 位段描述	367
表 15-24: DAC0 有效码值寄存器 (DAC0CODE) 位段定义	367
表 15-25: DAC0 有效码值寄存器 (DAC0CODE) 位段描述	367
表 15-26: DAC1 控制寄存器 (DAC1CTL) 位段定义	368
表 15-27: DAC1 控制寄存器 (DAC1CTL) 位段描述	368
表 15-28: DAC1 码值影子寄存器 (DAC1CODES) 位段定义	369
表 15-29: DAC1 码值影子寄存器 (DAC1CODES) 位段描述	369
表 15-30: DAC1 有效码值寄存器 (DAC1CODE) 位段定义	369
表 15-31: DAC1 有效码值寄存器 (DAC1CODE) 位段描述	369
表 15-32: DAC2 控制寄存器 (DAC2CTL) 位段定义	370
表 15-33: DAC2 控制寄存器 (DAC2CTL) 位段描述	370
表 15-34: DAC2 码值影子寄存器 (DAC2CODES) 位段定义	371
表 15-35: DAC2 码值影子寄存器 (DAC2CODES) 位段描述	371
表 15-36: DAC2 有效码值寄存器 (DAC2CODE) 位段定义	371
表 15-37: DAC2 有效码值寄存器 (DAC2CODE) 位段描述	371
表 15-38: COMP 模块写使能寄存器 (COMPREGKEY) 位段定义	372
表 15-39: COMP 模块写使能寄存器 (COMPREGKEY) 位段描述	372
表 16-1: UART 信号描述	374
表 16-2: 推荐的波特率	378
表 16-3: UART 模块基地址	379
表 16-4: UART 寄存器列表	379
表 16-5: UART 接收缓存寄存器 (UARTBR) 位段定义	380
表 16-6: UART 接收缓存寄存器 (UARTBR) 位段描述	380
表 16-7: UART 发送保持寄存器 (UARTTHR) 位段定义	380
表 16-8: UART 发送保持寄存器 (UARTTHR) 位段描述	380
表 16-9: UART 分频系数低字节寄存器 (UARTDLL) 位段定义	381
表 16-10: UART 分频系数低字节寄存器 (UARTDLL) 位段描述	381
表 16-11: UART 分频系数高字节寄存器 (UARTDLH) 位段定义	381
表 16-12: UART 分频系数高字节寄存器 (UARTDLH) 位段描述	381

表 16-13: UART 中断使能寄存器 (UARTIER) 位段定义	382
表 16-14: UART 中断使能寄存器 (UARTIER) 位段描述	382
表 16-15: UART 中断识别寄存器 (UARTIIR) 位段定义	383
表 16-16: UART 中断识别寄存器 (UARTIIR) 位段描述	383
表 16-17: UART FIFO 控制寄存器 (UARTFCR) 位段定义	384
表 16-18: UART FIFO 控制寄存器 (UARTFCR) 位段描述	384
表 16-19: UART 线路控制寄存器 (UARTLCR) 位段定义	386
表 16-20: UART 线路控制寄存器 (UARTLCR) 位段描述	386
表 16-21: UART 调制解调器控制寄存器 (UARTMCR) 位段定义	387
表 16-22: UART 调制解调器控制寄存器 (UARTMCR) 位段描述	387
表 16-23: UART 线路状态寄存器 (UARTLSR) 位段定义	388
表 16-24: UART 线路状态寄存器 (UARTLSR) 位段描述	388
表 16-25: UART 接收 FIFO 占用寄存器 (UARTFOR) 位段定义	390
表 16-26: UART 接收 FIFO 占用寄存器 (UARTFOR) 位段描述	390
表 16-27: UART 自动波特率控制寄存器 (UARTABR) 位段定义	391
表 16-28: UART 自动波特率控制寄存器 (UARTABR) 位段描述	391
表 16-29: UART 自动波特率计数寄存器 (UARTACR) 位段定义	392
表 16-30: UART 自动波特率计数寄存器 (UARTACR) 位段描述	392
表 17-1: I2C 端口信号	393
表 17-2: I2C 各速度模式 SCL 电平时间	396
表 17-3: I2C 各速度模式滤波时间	396
表 18-1: SPI 信号描述	440
表 18-2: SPI 模块基地址	446
表 18-3: SPI 寄存器列表	446
表 18-4: SPI 控制寄存器 0 (SPICTL0) 位段定义	447
表 18-5: SPI 控制寄存器 0 (SPICTL0) 位段描述	447
表 18-6: SPI 控制寄存器 1 (SPICTL1) 位段定义	449
表 18-7: SPI 控制寄存器 1 (SPICTL1) 位段描述	449
表 18-8: SPI 状态寄存器 (SPISTS) 位段定义	451
表 18-9: SPI 状态寄存器 (SPISTS) 位段描述	451
表 18-10: SPI 中断强制寄存器 (SPIFRC) 位段定义	453
表 18-11: SPI 中断强制寄存器 (SPIFRC) 位段描述	453
表 18-12: SPI 数据寄存器 (SPIDATA) 位段定义	454
表 18-13: SPI 数据寄存器 (SPIDATA) 位段描述	454
表 18-14: SPI 超时时间寄存器 (SPITO) 位段定义	454
表 18-15: SPI 超时时间寄存器 (SPITO) 位段描述	454
表 19-1: Flash 模块构成	457
表 19-2: 配置字的构成及其描述	463
表 19-3: Flash 控制器模块基地址	464
表 19-4: Flash 控制器寄存器列表	464
表 19-5: Flash 控制寄存器 (FLASHCTL) 位段定义	465
表 19-6: Flash 控制寄存器 (FLASHCTL) 位段描述	465
表 19-7: Flash 地址寄存器 (FLASHADDR) 位段定义	466
表 19-8: Flash 地址寄存器 (FLASHADDR) 位段描述	466
表 19-9: Flash 数据输入寄存器 (FLASHDIN) 位段定义	466
表 19-10: Flash 数据输入寄存器 (FLASHDIN) 位段描述	466

表 19-11: Flash 数据输出寄存器 (FLASHDOUT) 位段定义	467
表 19-12: Flash 数据输出寄存器 (FLASHDOUT) 位段描述	467
表 19-13: Flash 定时寄存器 0 (FLASHTIMING0) 位段定义	467
表 19-14: Flash 定时寄存器 0 (FLASHTIMING0) 位段描述	467
表 19-15: Flash 定时寄存器 1 (FLASHTIMING1) 位段定义	468
表 19-16: Flash 定时寄存器 1 (FLASHTIMING1) 位段描述	468
表 19-17: Flash 定时寄存器 2 (FLASHTIMING2) 位段定义	468
表 19-18: Flash 定时寄存器 2 (FLASHTIMING2) 位段描述	468
表 19-19: Flash 定时寄存器 3 (FLASHTIMING3) 位段定义	469
表 19-20: Flash 定时寄存器 3 (FLASHTIMING3) 位段描述	469
表 19-21: Flash 定时寄存器 4 (FLASHTIMING4) 位段定义	469
表 19-22: Flash 定时寄存器 4 (FLASHTIMING4) 位段描述	469
表 19-23: Flash 定时寄存器 5 (FLASHTIMING5) 位段定义	470
表 19-24: Flash 定时寄存器 5 (FLASHTIMING5) 位段描述	470
表 19-25: Flash 写保护寄存器 0 (FLASHWP0) 位段定义	471
表 19-26: Flash 写保护寄存器 0 (FLASHWP0) 位段描述	471
表 19-27: Flash 写保护寄存器 1 (FLASHWP1) 位段定义	474
表 19-28: Flash 写保护寄存器 1 (FLASHWP1) 位段描述	474
表 19-29: Flash 写保护寄存器 2 (FLASHWP2) 位段定义	477
表 19-30: Flash 写保护寄存器 2 (FLASHWP2) 位段描述	477
表 19-31: Flash 写保护寄存器 3 (FLASHWP3) 位段定义	480
表 19-32: Flash 写保护寄存器 3 (FLASHWP3) 位段描述	480
表 19-33: Flash 控制器写使能寄存器 (FLASHREGKEY) 位段定义	483
表 19-34: Flash 控制器写使能寄存器 (FLASHREGKEY) 位段描述	483
表 20-1: 64 位产品识别号定义	484
表 20-2: 64 位设备唯一识别号定义	485
表 20-3: SYSTEM 模块基地址	486
表 20-4: SYSTEM 寄存器列表	486
表 20-5: 产品识别号寄存器 0 (CID0) 位段定义	487
表 20-6: 产品识别号寄存器 0 (CID0) 位段描述	487
表 20-7: 产品识别号寄存器 1 (CID1) 位段定义	487
表 20-8: 产品识别号寄存器 1 (CID1) 位段描述	487
表 20-9: 唯一设备识别号寄存器 0 (UID0) 位段定义	488
表 20-10: 唯一设备识别号寄存器 0 (UID0) 位段描述	488
表 20-11: 唯一设备识别号寄存器 1 (UID1) 位段定义	488
表 20-12: 唯一设备识别号寄存器 1 (UID1) 位段描述	488
表 20-13: 随机数寄存器 0 (RND0) 位段定义	489
表 20-14: 随机数寄存器 0 (RND0) 位段描述	489
表 20-15: 随机数寄存器 1 (RND1) 位段定义	489
表 20-16: 随机数寄存器 1 (RND1) 位段描述	489
表 20-17: 版本信息寄存器 0 (REV0) 位段定义	490
表 20-18: 版本信息寄存器 0 (REV0) 位段描述	490
表 20-19: 版本信息寄存器 1 (REV1) 位段定义	490
表 20-20: 版本信息寄存器 1 (REV1) 位段描述	490
表 20-21: 存储器错误中断标志寄存器 (MEMIF) 位段定义	491
表 20-22: 存储器错误中断标志寄存器 (MEMIF) 位段描述	491

表 20-23: 存储器错误中断清除寄存器 (MEMIC) 位段定义	493
表 20-24: 存储器错误中断清除寄存器 (MEMIC) 位段描述	493
表 20-25: 存储器错误中断使能寄存器 (MEMIE) 位段定义	494
表 20-26: 存储器错误中断使能寄存器 (MEMIE) 位段描述	494
表 20-27: 存储器 ECC 使能寄存器 (MEMECCEN) 位段定义	495
表 20-28: 存储器 ECC 使能寄存器 (MEMECCEN) 位段描述	495
表 20-29: 存储器锁定状态寄存器 (MEMLOCKSTS) 位段定义	496
表 20-30: 存储器锁定状态寄存器 (MEMLOCKSTS) 位段描述	496
表 20-31: 复位事件状态寄存器 (RSTEVTF LG) 位段定义	497
表 20-32: 复位事件状态寄存器 (RSTEVTF LG) 位段描述	497
表 20-33: 复位事件状态清除寄存器 (RSTEVTF CLR) 位段定义	499
表 20-34: 复位事件状态清除寄存器 (RSTEVTF CLR) 位段描述	499
表 20-35: 复位事件使能寄存器 (RSTEVTF EN) 位段定义	501
表 20-36: 复位事件使能寄存器 (RSTEVTF EN) 位段描述	501
表 20-37: SYSTEM 模块写使能寄存器 (SYSREGKEY) 位段定义	502
表 20-38: SYSTEM 模块写使能寄存器 (SYSREGKEY) 位段描述	502
表 20-39: ROM 读取 ECC 错误地址寄存器 (ROMERRADDR) 位段定义	503
表 20-40: ROM 读取 ECC 错误地址寄存器 (ROMERRADDR) 位段描述	503
表 20-41: Flash 读取 ECC 错误地址寄存器 (FLASHERRADDR) 位段定义	503
表 20-42: Flash 读取 ECC 错误地址寄存器 (FLASHERRADDR) 位段描述	503
表 20-43: RAM0 读取奇偶校验错误地址寄存器 (RAM0ERRADDR) 位段定义	504
表 20-44: RAM0 读取奇偶校验错误地址寄存器 (RAM0ERRADDR) 位段描述	504
表 20-45: 读取奇偶校验错误地址寄存器 (RAM1ERRADDR) 位段定义	504
表 20-46: 读取奇偶校验错误地址寄存器 (RAM1ERRADDR) 位段描述	504
表 20-47: 非易失寄存器 0 (NVREG0) 位段定义	505
表 20-48: 非易失寄存器 0 (NVREG0) 位段描述	505
表 20-49: 非易失寄存器 1 (NVREG1) 位段定义	505
表 20-50: 非易失寄存器 1 (NVREG1) 位段描述	505
表 20-51: 启动信息寄存器 (BOOTINFO) 位段定义	506
表 20-52: 启动信息寄存器 (BOOTINFO) 位段描述	506
表 21-1: CPU HALT 模式下 WDT 调试行为	507
表 21-2: CPU LOCKUP 模式下 WDT 调试行为	507
表 21-3: PWM 调试行为	507
表 21-4: ECAP 调试行为	507
表 21-5: UART 调试行为	508
表 21-6: SPI 调试行为	508
表 21-7: I2C 调试行为	509
表 22-1: MCU 通道与 DC-DC 模块的连接关系	511
表 22-2: EPWR 模块基地址	516
表 22-3: EPWR 寄存器列表	516
表 22-4: ePower 滤波后的封锁事件寄存器 (EPWRTZFLT) 位段定义	517
表 22-5: ePower 滤波后的封锁事件寄存器 (EPWRTZFLT) 位段描述	517
表 22-6: ePower 封锁状态寄存器 (EPWRTZFLG) 位段定义	518
表 22-7: ePower 封锁状态寄存器 (EPWRTZFLG) 位段描述	518
表 22-8: ePower 封锁状态清除寄存器 (EPWRTZCLR) 位段定义	519
表 22-9: ePower 封锁状态清除寄存器 (EPWRTZCLR) 位段描述	519

表 22-10: ePower 封锁事件 0 控制寄存器 (EPWRTZ0CTL) 位段定义	520
表 22-11: ePower 封锁事件 0 控制寄存器 (EPWRTZ0CTL) 位段描述	520
表 22-12: ePower 封锁事件 1 控制寄存器 (EPWRTZ1CTL) 位段定义	522
表 22-13: ePower 封锁事件 1 控制寄存器 (EPWRTZ1CTL) 位段描述	522
表 22-14: ePower 控制寄存器 (EPWRCTL) 位段定义	524
表 22-15: ePower 控制寄存器 (EPWRCTL) 位段描述	524
表 22-16: ePower 模块写使能寄存器 (EPWRREGKEY) 位段定义	525
表 22-17: ePower 模块写使能寄存器 (EPWRREGKEY) 位段描述	525
表 22-18: Buck 模块寄存器列表	526
表 22-19: 全局状态寄存器 (GLBSTS) 位段定义	527
表 22-20: 全局状态寄存器 (GLBSTS) 位段描述	527
表 22-21: 全局状态清除寄存器 (GLBSTSCLR) 位段定义	528
表 22-22: 全局状态清除寄存器 (GLBSTSCLR) 位段描述	528
表 22-23: 封锁事件控制寄存器 (TZCTL) 位段定义	529
表 22-24: 封锁事件控制寄存器 (TZCTL) 位段描述	529
表 22-25: Buck 控制寄存器 0 (BUCKCTL0) 位段定义	530
表 22-26: Buck 控制寄存器 0 (BUCKCTL0) 位段描述	530
表 22-27: Buck 控制寄存器 1 (BUCKCTL1) 位段定义	531
表 22-28: Buck 控制寄存器 1 (BUCKCTL1) 位段描述	531
表 23-1: MCU 和预驱模块的连接接口	532

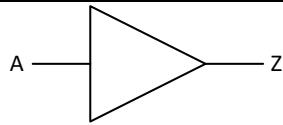
修订历史

版本	日期	作者	状态	变更
C/0	2025-05-28	J. Zhou	已发布	1. 首次发布。

术语或缩写

术语或缩写	描述
MCU	Microcontroller Unit, 微控制器单元
SWD	Serial Wire Debug, 串行线调试
AHB	Advanced High Performance Bus, 先进高性能总线
XIP	Execution In Place, 就地执行
PLL	Phase Locked Loop, 锁相环
BOD	Brownout Detector, 过压/欠压检测器
PFD	Phase Frequency Detector, 鉴频鉴相器
NVIC	Nested Vectored Interrupt Controller, 嵌套向量中断控制器
UART	Universal Asynchronous Receiver-Transmitter, 通用异步收发器
ADC	Analog-to-Digital Converter, 模数转换器
DAC	Digital-to-Analog Converter, 数模转换器
DPGA	Differential Programmable-Gain Amplifier, 差分可编程增益放大器
CRC	Cyclic Redundancy Check, 循环冗余校验
RW	Read-Write, 寄存器可读写
RO	Read-Only, 寄存器只读
WO	Write-Only, 寄存器只写
RC	Read-Clear, 读操作自动清除寄存器位
W1C	Write-1-to-Clear, 写 1 清除寄存器位, 写 0 无效
W1S	Write-1-to-Set, 写 1 置位, 写 0 无效

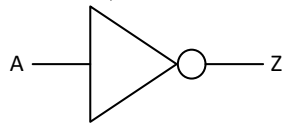
逻辑符号



Buffer

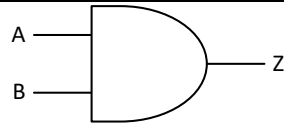
A	Z = A
0	0
1	1

○ Can be simplified as a single circle



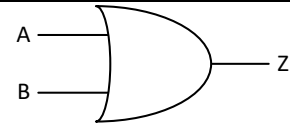
Inverter

A	Z = not A
0	1
1	0



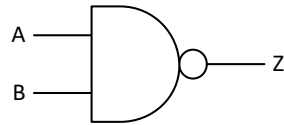
AND

A	B	Z = A and B
0	0	0
0	1	0
1	0	0
1	1	1



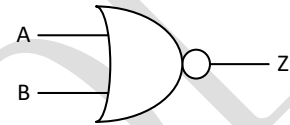
OR

A	B	Z = A or B
0	0	0
0	1	1
1	0	1
1	1	1



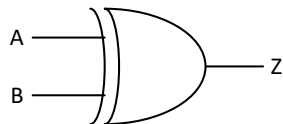
NAND

A	B	Z = not (A and B)
0	0	1
0	1	1
1	0	1
1	1	0



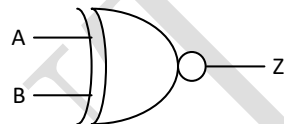
NOR

A	B	Z = not (A or B)
0	0	1
0	1	0
1	0	0
1	1	0



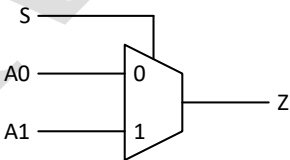
XOR (Exclusive OR)

A	B	Z = (A and not B) or (B and not A)
0	0	0
0	1	1
1	0	1
1	1	0



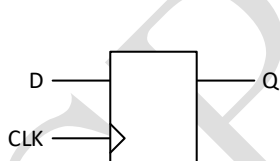
XNOR (Exclusive NOR)

A	B	Z = (A and B) or (not A and not B)
0	0	1
0	1	0
1	0	0
1	1	1



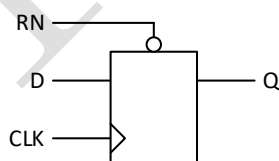
MUX

A0	A1	S	Z = A0 if S = 0 A1 if S = 1
0	any	0	0
1	any	0	1
any	0	1	0
any	1	1	1



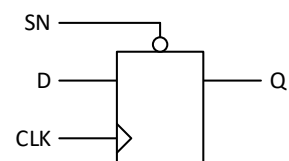
DFF (D Flip-flop)

D	CLK	Q
any	0	keep it's old value
any	1	keep it's old value
any	↓	keep it's old value
0	↑	0
1	↑	1



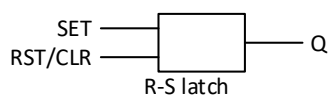
DFF with asynchronous reset

D	RN	CLK	Q
any	0	any	0
any	1	0	keep it's old value
any	1	1	keep it's old value
any	1	↓	keep it's old value
0	1	↑	0
1	1	↑	1



DFF with asynchronous reset

D	SN	CLK	Q
any	0	any	1
any	1	0	keep it's old value
any	1	1	keep it's old value
any	1	↓	keep it's old value
0	1	↑	0
1	1	↑	1



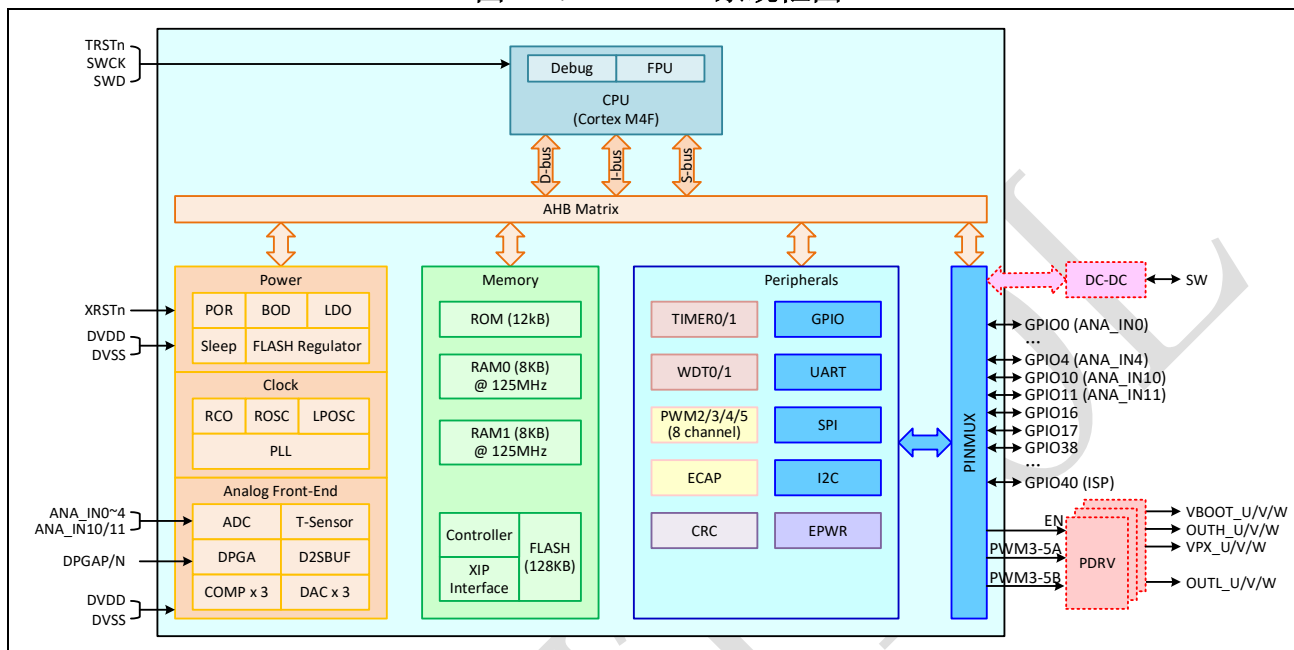
R-S latch

SET	RST	Q
1	0	1
0	1	0
0	0	keep it's old value
1	1	Invalid input

1 存储器和总线架构

1.1 系统架构

图 1-1: SPD1121 系统框图



SPD1121 为基于 ARM® Cortex-M4F 的系统级封装（SiP），如图 1-1 所示，通过低延时的 AHB 总线架构互联如下模块：

- 3 个总线主机
 - Cortex-M4F 主核的 I 总线、D 总线以及系统总线。
- 总线从机
 - ROM。
 - 2 块 RAM。
 - FLASH 控制器以及就地执行（XIP）接口。
 - 电源和时钟配置寄存器。
 - AHB 外设：CRC、AFE（Analog Front End）、PWM、ECAP、WDT、TIMER、GPIO、UART、SPI、I2C 以及 EPWR 等。

模拟电路部分包含如下子系统：

- 电源
 - 支持两种供电方式：通过 DVDD 提供 3.3V 供电或通过 VDDG，由合封的 DC-DC 供电。
 - 提供上电复位（POR）、1.2V 片上稳压器（LDO）、欠压过压检测（BOD）等。
- 时钟
 - 提供 4 个基础时钟源。
- 模拟前端信号链
 - 提供模数转换器（ADC）、差分可编程增益放大器（DPGA）、数模转换器（DAC）、温度传感器（T-sensor）、模拟比较器（COMP）等。
- 集成三相 600V 预驱

1.2 存储器映射

SPD1121 的存储空间地址映射如图 1-2 所示。未分配给任何存储单元或者外设寄存器的地址视作“未使用的”空间，对这类地址的访问将产生总线错误。

图 1-2: SPD1121 存储空间地址映射

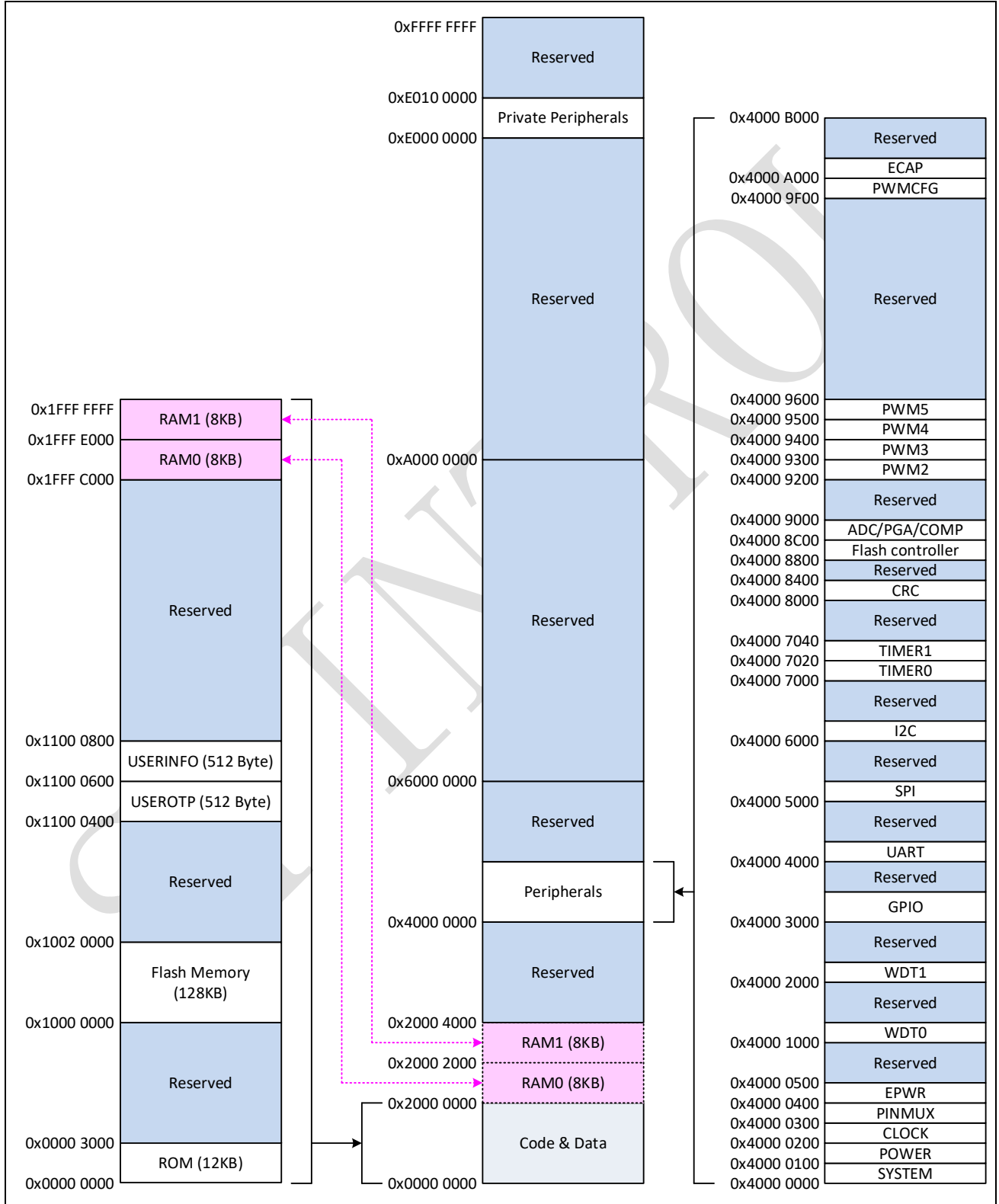


表 1-1 给出了 SPD1121 各个外设模块的基地址。

表 1-1: 外设模块基地址

外设模块	基地址	寄存器列表
SYSTEM	0x4000 0000	SYSTEM 寄存器表
POWER	0x4000 0100	POWER 寄存器表
CLOCK	0x4000 0200	CLOCK 寄存器表
GPIO	0x4000 3000	GPIO 寄存器表
PINMUX	0x4000 0300	PINMUX 寄存器表
EPWR	0x4000 0400	EPWR 寄存器表
CRC	0x4000 8000	CRC 寄存器表
TIMER0	0x4000 7000	TIMERx 寄存器表
TIMER1	0x4000 7020	
WDT0	0x4000 1000	WDTx 寄存器表
WDT1	0x4000 2000	
PWM2	0x4000 9200	PWMx 寄存器表
PWM3	0x4000 9300	
PWM4	0x4000 9400	
PWM5	0x4000 9500	
PWMCFG	0x4000 9F00	PWMCFG 寄存器表
ECAP	0x4000 A000	ECAP 寄存器表
ADC	0x4000 8C00	ADC 寄存器表
DPGA		DPGA 寄存器表
COMP		COMP 寄存器表
UART	0x4000 4000	UART 寄存器表
I2C	0x4000 6000	I2C 寄存器表
SPI	0x4000 5000	SPI 寄存器表
FLASH	0x4000 8800	FLASH 寄存器表

1.2.1 引导 ROM

SPD1121 集成 12KB 的片上 ROM 用于存放启动引导程序、与系统和 Flash 管理相关的函数库以及用户代码中经常会用到的数学库，支持 ECC 保护，可以纠正单比特错误和检测多比特错误。ECC 错误可配置为触发复位或中断，发生 ECC 错误的地址记录在 ROMERRADDR 寄存器中用于诊断。

如图 1-3 所示，为了方便用户进行系统管理、Flash 操作和三角函数等运算，ROM 中固化了函数库，其入口地址向量表存放的起始地址为 0x0000 2F60。表 1-2 罗列了函数库的细节。

为了保护代码，用户只允许读取入口地址向量表的内容，以及执行函数库代码，而无法读取函数的具体内容。

图 1-3: ROM 存储空间划分

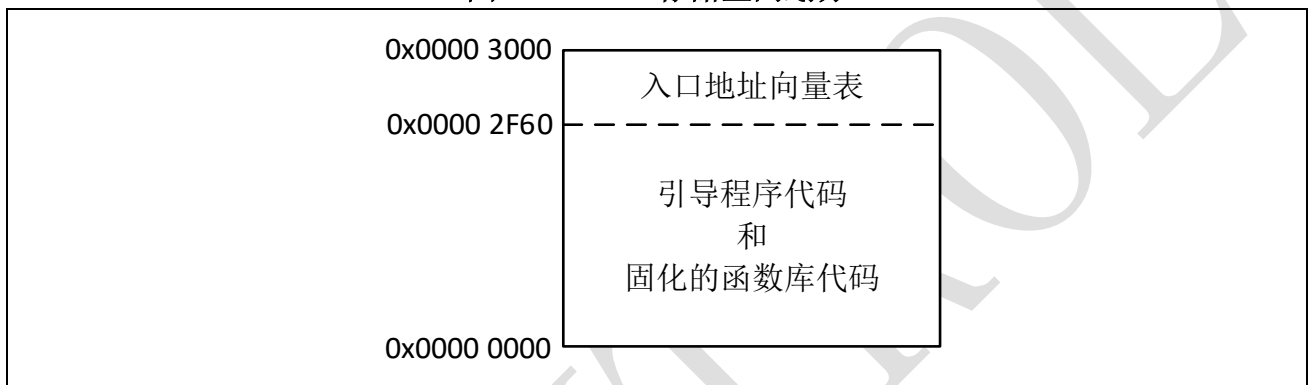


表 1-2: 固化的函数库入口地址向量表

函数名	向量地址	说明
FLASH_SetTiming	0x0000 2F60	根据时钟频率配置 Flash 时序参数
FLASH_Init	0x0000 2F64	初始化 Flash 配置
FLASH_Read	0x0000 2F68	读取 Flash 内容（常规 0/1 判别阈值）
FLASH_VerifyProgramRead	0x0000 2F6C	读取 Flash 内容用于验证烧录是否正确（采用偏于更难读到 0 的阈值）
FLASH_VerifyEraseRead	0x0000 2F70	读取 Flash 内容用于验证擦除是否正确（采用偏于更难读到 1 的阈值）
FLASH_VerifyErase	0x0000 2F74	检查指定区域的 Flash 是否被成功擦除
FLASH_Erase	0x0000 2F78	按照指定方式擦除 Flash 指定的地址（本函数不对客户开放）
FLASH_EraseSector	0x0000 2F7C	擦除 Flash 指定的扇区
FLASH_EraseBlock	0x0000 2F80	擦除 Flash 指定的块
FLASH_EraseChip	0x0000 2F84	擦除 Flash 整个主分区
FLASH_ProgramWord	0x0000 2F88	烧录一个字（32 位）到 Flash 指定地址
FLASH_Program	0x0000 2F8C	烧录一组数据到 Flash 指定地址
FLASH_WriteProtect	0x0000 2F90	使能 Flash 主分区指定区域的写保护

函数名	向量地址	说明
FLASH_WriteUnProtect	0x0000 2F94	关闭 Flash 主分区指定区域的写保护
SYSTEM_Sleep	0x0000 2F98	配置唤醒条件并让系统进入睡眠模式
AtanF32	0x0000 2F9C	由应用团队提供的函数库
ThetaRadWrap	0x0000 2FA0	
SAT0	0x0000 2FA4	
SVPWM	0x0000 2FA8	
ClarkeF32	0x0000 2FAC	
InvParkF32	0x0000 2FB0	
ParkF32	0x0000 2FB4	
BubbleSortForCurrent	0x0000 2FB8	
Motor_1PhaseCurrentReconstruct	0x0000 2FBC	
Motor_1PhaseCurrentSamplingShift	0x0000 2FC0	
arm_sin_f32	0x0000 2FC4	
arm_cos_f32	0x0000 2FC8	

1.2.2 嵌入式 Flash 存储器

SPD1121 内嵌 Flash 存储器，可提供最大 128 KB 空间用于存放程序的代码和数据。如图 1-1 所示，Flash 存储器可以通过总线寄存器访问，由状态机发送命令来擦写 Flash 存储器，还支持通过特别设计的就地执行（XIP）接口读取 Flash 存储器的内容。因此，当 SPD1121 启动时，Flash 存储器中的代码可以直接就地执行，也可以被加载到 RAM 后，再在 RAM 中执行。

出于数据安全性考虑，Flash 存储器提供了 ECC 功能，可以修正单比特错误和检测多比特错误。ECC 错误可以配置为触发复位或者中断。发生 ECC 错误的地址记录在 FLASHERRADDR 寄存器中用于诊断。

1.2.3 片上 SRAM

SPD1121 提供 16KB 的片上 RAM 用于存储代码和数据，支持最高 125MHz 速率零等待的读写操作，分为如下两个 8KB 分区：

- RAM0：通过地址 0x1FFF C000 到 0x1FFF DFFF 或 0x2000 0000 到 0x2000 1FFF 访问。
- RAM1：通过地址 0x1FFF E000 到 0x1FFF FFFF 或 0x2000 2000 到 0x2000 3FFF 访问。

所有 RAM 均支持偶校验保护，校验错误可配置为触发复位或者中断，发生错误的地址记录在 RAM0ERRADDR 和 RAM1ERRADDR 寄存器中用于诊断。

1.3 启动引导配置

SPD1121 复位后，执行位于 ROM 上的引导程序，并支持如表 1-3 所示的两种启动模式。

- 当上电复位或者 XRSTn 引脚复位后，启动时检测到 GPIO40 为低，则进入 ISP 启动模式，启动加载器通过 UART 对 Flash 或者 RAM 进行重新编程。在这个过程中，GPIO34 被配置为 UART_TXD 功能；GPIO35 被配置为 UART_RXD 功能。
- 其余情况均进入正常启动模式，启动加载器跳转至 Flash 起始地址开始执行。

表 1-3: 启动模式

GPIO40	复位原因	启动模式
0	上电复位	ISP 模式： 通过 UART 对 Flash 或者 RAM 进行重新编程
0	XRSTn 复位	
0	ARM® CPU 系统复位请求	正常启动模式： 从 Flash 存储器开始运行程序
0	电源过压/欠压	
0	锁相环失锁/时钟检测电路报错	
0	ROM/RAM/Flash 存储器错误	
0	看门狗定时器超时	
1	任何复位	

2 电源

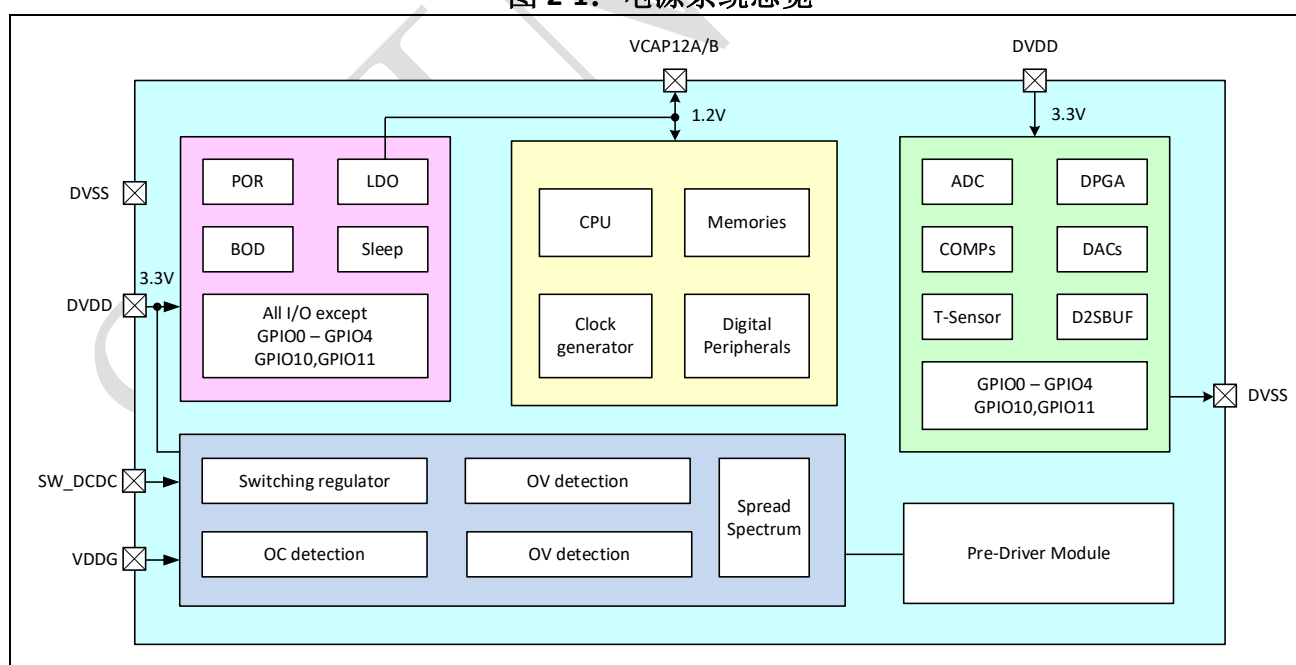
2.1 单电源供电

SPD1121 由 11V~20V 的单电源 VDDG 供电。片上 DC-DC 开关电源基于 VDDG 产生额定 3.3V 的 DVDD。当关闭 DC-DC 开关电源时，也可以通过外部 3.3V 电源给 DVDD 供电。上电顺序无特殊要求，但电源升压速率应小于 0.015 V/us。片上 LDO 稳压器基于 DVDD 产生额定 1.2V 电压，并输出到 VCAP12 引脚。电源系统具体如下：

- DVDD (3.3V)
 - 对模拟前端电路、电源子模块以及所有 I/O 引脚供电。
 - 对应的地是 DVSS。
 - 设计电路板时，需要注意在尽量靠近 DVDD 和 DVSS 引脚的地方放置去耦电容。
- LDO 产生的 1.2V 电源
 - 对 CPU、存储器、时钟产生电路和数字外设供电。
 - 支持过流保护。
 - 对应的地是 DVSS。
 - 设计电路板时，需要注意在尽量靠近 VCAP12 和 DVSS 引脚的地方放置去耦电容。
- VDDG (11~20V)
 - 对 DC-DC 开关电源和预驱供电。
 - 对应的地是 DVSS。

设计电路板时，需要注意在尽量靠近 VDDG 和 DVSS 引脚的地方放置去耦电容。

图 2-1：电源系统总览



2.2 过压/欠压检测

BOD (Brown-out Detectors) 模块用于将 3.3V/1.2V 额定电压域实际电压值与预设阈值加以比较, 实现过压或者欠压的检测, 检测结果存放于 PWRIEVTF 寄存器中, 并可以根据 PWRIE 寄存器配置决定是否产生中断。

BOD33CTL 和 BOD12CTL 寄存器用于设定 3.3V/1.2V 额定电压域的过压欠压检测阈值。

为了提高额外的灵活度, 1.2V 电压域提供了两个 BOD 欠压检测。用户可以实现诸如低于某个值时产生中断报警、进一步低于某个值时产生复位的保护功能。

预驱部分检测 VDDG 欠压, 欠压时预驱输出低电平。

图 2-2 和图 2-3 给出了 3.3V/1.2V 额定电压域检测电路框图。

图 2-2: 3.3V 额定电压域检测电路框图

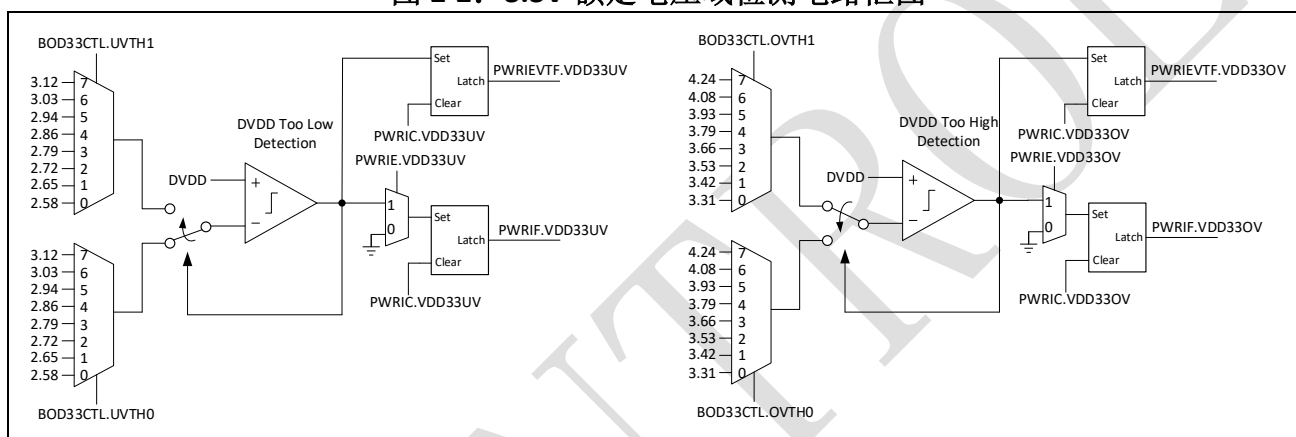
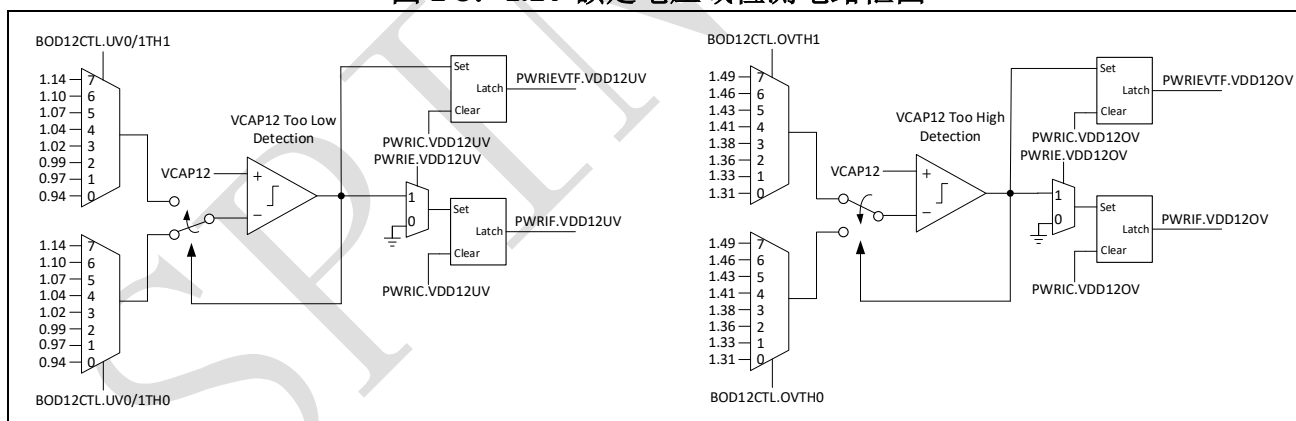
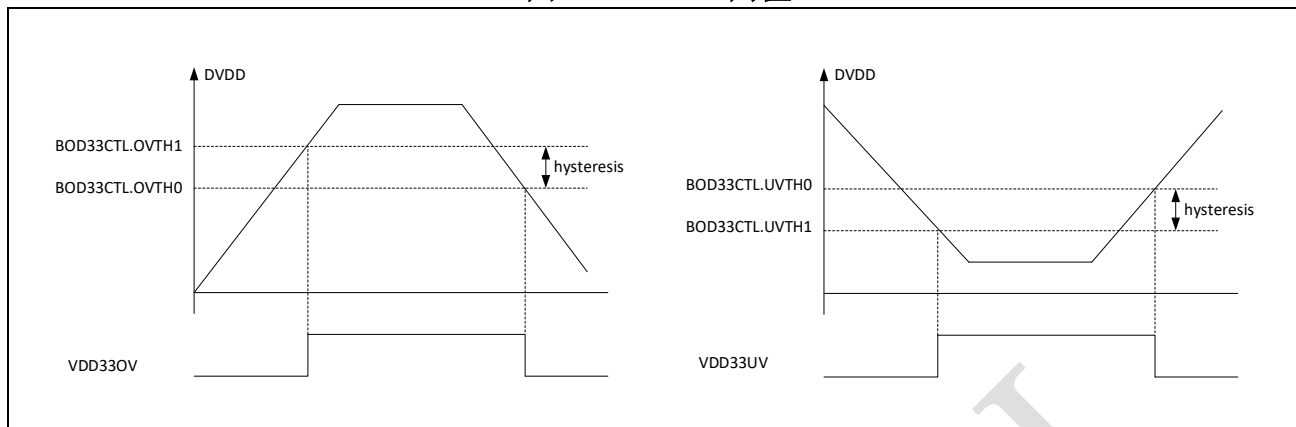


图 2-3: 1.2V 额定电压域检测电路框图



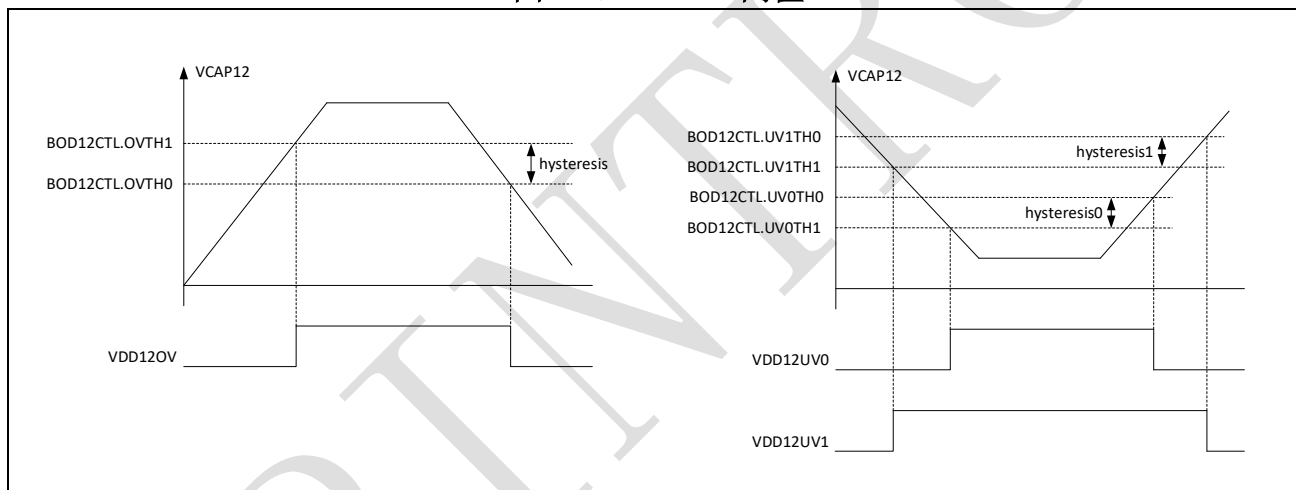
3.3V 额定电压域可以在 DVDD 电压上升并高于过压阈值 1 时产生过压事件, 之后电压降低恢复至低于过压阈值 0 时撤销过压事件, 也可以在 DVDD 电压下降并低于欠压阈值时 1 产生欠压事件, 之后电压升高恢复至高于欠压阈值 0 时撤销欠压事件。

图 2-4: DVDD 阈值



1.2V 额定电压域可以在 VCAP12 电压上升并高于过压阈值 1 时产生过压事件，之后电压降低恢复至低于过压阈值 0 时撤销过压事件。也可以在 VCAP12 电压下降并低于欠压阈值 1 时产生欠压事件，之后电压上升恢复至高于欠压阈值 0 时撤销事件。

图 2-5: VCAP12 阈值



注意： 如图 2-4 和图 2-5 所示按照如下说明来正确使用 BOD：

1. 过压检测时，
 - $BOD33CTL.OVTH1 \geq BOD33CTL.OVTH0$
 - $BOD12CTL.OVTH1 \geq BOD12CTL.OVTH0$
2. 在欠压检测时，
 - $BOD33CTL.UVTH1 \leq BOD33CTL.UVTH0$
 - $BOD12CTL.UV0TH1 \leq BOD12CTL.UV0TH0$
 - $BOD12CTL.UV1TH1 \leq BOD12CTL.UV1TH0$

2.3 睡眠模式

SPD1121 的片上 1.2V LDO 支持睡眠模式。用户可以通过向 SLEEPKEY 寄存器写 0x51EE9 来关闭 1.2V LDO，并使得整个系统进入睡眠模式，之后可以通过 GPIO10 或 GPIO11 来唤醒系统。

2.4 寄存器

2.4.1 Power 寄存器列表

表 2-1: POWER 模块基地址

外设模块	基地址
POWER	0x4000 0100

表 2-2: POWER 寄存器列表

寄存器	偏移地址	描述	复位值
PWRSTS	0x00	电源状态寄存器	0x00000001
LDOCTL*	0x04	LDO 控制寄存器	0x00000007
PWRIEVTF	0x08	电源中断事件标志寄存器	0x00000000
PWRCTL*	0x0C	电源控制寄存器	0x00000103
PWRIF	0x10	电源中断标志寄存器	0x00000000
PWRIC	0x14	电源中断清除寄存器	0x00000000
PWRIE*	0x18	电源中断使能寄存器	0x00000000
BOD33CTL*	0x20	3.3V BOD 控制寄存器	0x00006F11
BOD12CTL*	0x24	1.2V BOD 控制寄存器	0x006F1111
SLEEPKEY*	0x28	睡眠模式使能寄存器	0x00000000
PWRREGKEY	0x2C	电源寄存器允许写入使能寄存器	0x1ACCE551

注意： 由*标记的寄存器仅当 PWRREGKEY=0x1ACCE551 才可以改写。

2.4.2 Power 寄存器

表 2-3 到表 2-24 给出了 Power 模块各寄存器的定义。

2.4.2.1 电源状态寄存器（PWRSTS）

表 2-3：电源状态寄存器（PWRSTS）位段定义

PWRSTS (Power Status Register)							
Access: POWER->PWRSTS Offset: 0x0000 Default: 0x00000001							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							FLASHREGRDY

表 2-4：电源状态寄存器（PWRSTS）位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED_31_1	RO	0x0	保留
0	FLASHREGRDY	RO	0x1	Flash 电源就绪标志 0: 未就绪 1: 就绪

2.4.2.2 LDO 控制寄存器 (LDOCTL)

表 2-5: LDO 控制寄存器 (LDOCTL) 位段定义

LDOCTL (LDO Control Register) Access: POWER->LDOCTL Offset: 0x0004 Default: 0x00000007							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED				VOUT			

表 2-6: LDO 控制寄存器 (LDOCTL) 位段描述

位段	位段名	属性	复位值	描述
31:4	RESERVED_31_4	RO	0x0	保留
3:0	VOUT	RW	0x7	1.2V LDO 输出电压 3: 1.12V 4: 1.14V 5: 1.16V 6: 1.18V 7: 1.20V 8: 1.22V 9: 1.24V 10: 1.26V 11: 1.28V 12: 1.30V

2.4.2.3 电源中断事件标志寄存器（PWRIEVTF）

表 2-7：电源中断事件标志寄存器（PWRIEVTF）位段定义

PWRIEVTF (Power Interrupt Event Flag Register)							
Access: POWER->PWRIEVTF Offset: 0x0008 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED			VDD33OV	VDD33UV	VDD12OV	VDD12UV1	VDD12UV0

表 2-8：电源中断事件标志寄存器（PWRIEVTF）位段描述

位段	位段名	属性	复位值	描述
31:5	RESERVED_31_5	RO	0x0	保留
4	VDD33OV	RO	0x0	3.3V 电源过压事件标志 0: 事件未发生 1: 事件发生
3	VDD33UV	RO	0x0	3.3V 电源欠压事件标志 0: 事件未发生 1: 事件发生
2	VDD12OV	RO	0x0	1.2V 电源过压事件标志 0: 事件未发生 1: 事件发生
1	VDD12UV1	RO	0x0	1.2V 电源欠压事件 1 标志 0: 事件未发生 1: 事件发生
0	VDD12UV0	RO	0x0	1.2V 电源欠压事件 0 标志 0: 事件未发生 1: 事件发生

2.4.2.4 POWER 控制寄存器（PWRCTL）

表 2-9: POWER 控制寄存器（PWRCTL）位段定义

PWRCTL (Power Control Register) Access: POWER->PWRCTL Offset: 0x000C Default: 0x00000103							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED						WKUPIO	WKUPPOL
7	6	5	4	3	2	1	0
RESERVED						XRSTFILT	

表 2-10: POWER 控制寄存器（PWRCTL）位段描述

位段	位段名	属性	复位值	描述
31:10	RESERVED_31_10	RO	0x0	保留
9	WKUPIO	RW	0x0	从休眠状态唤醒的 GPIO 引脚 0: 被 GPIO10 唤醒 1: 被 GPIO11 唤醒
8	WKUPPOL	RW	0x1	休眠状态唤醒的 GPIO 电平 0: 引脚低电平唤醒 1: 引脚高电平唤醒
7:2	RESERVED_7_2	RO	0x0	保留
1:0	XRSTFILT	RW	0x3	外部复位输入消抖滤波窗口 0: 500us 1: 1ms 2: 2ms 3: 4ms

2.4.2.5 POWER 中断标志寄存器 (PWRIF)

表 2-11: POWER 中断标志寄存器 (PWRIF) 位段定义

PWRIF (Power Interrupt Flag Register) Access: POWER->PWRIF Offset: 0x0010 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED		BODINT	VDD33OV	VDD33UV	VDD12OV	VDD12UV1	VDD12UV0

表 2-12: POWER 中断标志寄存器 (PWRIF) 位段描述

位段	位段名	属性	复位值	描述
31:6	RESERVED_31_6	RO	0x0	保留
5	BODINT	RO	0x0	电源全局中断标志 0: 中断未发生 1: 中断曾发生 该位被清零前新事件不会触发中断
4	VDD33OV	RO	0x0	3.3V 电源过压中断标志 0: 中断未发生 1: 中断曾发生 该位被清零前新事件不会触发中断
3	VDD33UV	RO	0x0	3.3V 电源欠压中断标志 0: 中断未发生 1: 中断曾发生 该位被清零前新事件不会触发中断
2	VDD12OV	RO	0x0	1.2V 电源过压中断标志 0: 中断未发生 1: 中断曾发生 该位被清零前新事件不会触发中断
1	VDD12UV1	RO	0x0	1.2V 电源欠压 1 中断标志 0: 中断未发生 1: 中断曾发生 该位被清零前新事件不会触发中断
0	VDD12UV0	RO	0x0	1.2V 电源欠压 0 中断标志 0: 中断未发生 1: 中断曾发生 该位被清零前新事件不会触发中断

2.4.2.6 POWER 中断清除寄存器 (PWRIC)

表 2-13: POWER 中断清除寄存器 (PWRIC) 位段定义

PWRIC (Power Interrupt Clear Register)							
Access: POWER->PWRIC		Offset: 0x0014		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED		BODINT	VDD33OV	VDD33UV	VDD12OV	VDD12UV1	VDD12UV0

表 2-14: POWER 中断清除寄存器 (PWRIC) 位段描述

位段	位段名	属性	复位值	描述
31:6	RESERVED_31_6	RO	0x0	保留
5	BODINT	W1C	0x0	电源全局中断清除 0: 写 0 无效, 总是回读 0 1: 写 1 清除标志位, 本位段自动清零
4	VDD33OV	W1C	0x0	3.3V 电源过压中断清除 0: 写 0 无效, 总是回读 0 1: 写 1 清除标志位, 本位段自动清零
3	VDD33UV	W1C	0x0	3.3V 电源欠压中断清除 0: 写 0 无效, 总是回读 0 1: 写 1 清除标志位, 本位段自动清零
2	VDD12OV	W1C	0x0	1.2V 电源过压中断清除 0: 写 0 无效, 总是回读 0 1: 写 1 清除标志位, 本位段自动清零
1	VDD12UV1	W1C	0x0	1.2V 电源欠压 1 中断清除 0: 写 0 无效, 总是回读 0 1: 写 1 清除标志位, 本位段自动清零
0	VDD12UV0	W1C	0x0	1.2V 电源欠压 0 中断清除 0: 写 0 无效, 总是回读 0 1: 写 1 清除标志位, 本位段自动清零

2.4.2.7 Power 中断使能寄存器 (PWRIE)

表 2-15: Power 中断使能寄存器 (PWRIE) 位段定义

PWRIE (Power Interrupt Enable Register)							
Access: POWER->PWRIE Offset: 0x0018 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED			VDD33OV	VDD33UV	VDD12OV	VDD12UV1	VDD12UV0

表 2-16: Power 中断使能寄存器 (PWRIE) 位段描述

位段	位段名	属性	复位值	描述
31:5	RESERVED_31_5	RO	0x0	保留
4	VDD33OV	RW	0x0	3.3V 电源过压中断使能 0: 关闭 1: 使能
3	VDD33UV	RW	0x0	3.3V 电源欠压中断使能 0: 关闭 1: 使能
2	VDD12OV	RW	0x0	1.2V 电源过压中断使能 0: 关闭 1: 使能
1	VDD12UV1	RW	0x0	1.2V 电源欠压 1 中断使能 0: 关闭 1: 使能
0	VDD12UV0	RW	0x0	1.2V 电源欠压 0 中断使能 0: 关闭 1: 使能

2.4.2.8 3.3V BOD 控制寄存器 (BOD33CTL)

表 2-17: 3.3V BOD 控制寄存器 (BOD33CTL) 位段定义

BOD33CTL (3.3V BOD Control Register) Access: POWER->BOD33CTL Offset: 0x0020 Default: 0x00006F11							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15 RESERVED	14	13	12	11	10	9	8 RESERVED
		OVTH0		OVTH1			
7 RESERVED	6	5	4	3	2	1	0 RESERVED
		UVTH0		UVTH1			

表 2-18: 3.3V BOD 控制寄存器 (BOD33CTL) 位段描述

位段	位段名	属性	复位值	描述
31:15	RESERVED_31_15	RO	0x0	保留
14:12	OVTH0	RW	0x6	3.3V 电源下降时撤销 VDD33OV 的阈值 0: 3.31V 1: 3.42V 2: 3.53V 3: 3.66V 4: 3.79V 5: 3.93V 6: 4.08V 7: 4.24V
11:9	OVTH1	RW	0x7	3.3V 电源上升时触发 VDD33OV 的阈值 0: 3.31V 1: 3.42V 2: 3.53V 3: 3.66V 4: 3.79V 5: 3.93V 6: 4.08V 7: 4.24V
8:7	RESERVED_8_7	RO	0x2	保留
6:4	UVTH0	RW	0x1	3.3V 电源上升时撤销 VDD33UV 的阈值 0: 2.58V 1: 2.65V 2: 2.72V 3: 2.79V 4: 2.86V 5: 2.94V 6: 3.03V 7: 3.12V

位段	位段名	属性	复位值	描述
3:1	UVTH1	RW	0x0	3.3V 电源下降时触发 VDD33UV 的阈值 0: 2.58V 1: 2.65V 2: 2.72V 3: 2.79V 4: 2.86V 5: 2.94V 6: 3.03V 7: 3.12V
0	RESERVED_0	RO	0x1	保留

2.4.2.9 1.2V BOD 控制寄存器（BOD12CTL）

表 2-19: 1.2V BOD 控制寄存器（BOD12CTL）位段定义

BOD12CTL (1.2V BOD Control Register) Access: POWER->BOD12CTL Offset: 0x0024 Default: 0x006F1111							
31	30	29	28	27	26	25	24
RESERVED							
23 RESERVED	22	21	20	19	18	17	16 RESERVED
OVTH0				OVTH1			
15 RESERVED	14	13	12	11	10	9	8 RESERVED
UV1TH0				UV1TH1			
7 RESERVED	6	5	4	3	2	1	0 RESERVED
UV0TH0				UV0TH1			

表 2-20: 1.2V BOD 控制寄存器（BOD12CTL）位段描述

位段	位段名	属性	复位值	描述
31:23	RESERVED_31_23	RO	0x0	保留
22:20	OVTH0	RW	0x6	1.2V 电源下降时撤销 VDD12OV 的阈值 0: 1.31V 1: 1.33V 2: 1.36V 3: 1.38V 4: 1.41V 5: 1.43V 6: 1.46V 7: 1.49V
19:17	OVTH1	RW	0x7	1.2V 电源上升时触发 VDD12OV 的阈值 0: 1.31V 1: 1.33V 2: 1.36V 3: 1.38V 4: 1.41V 5: 1.43V 6: 1.46V 7: 1.49V
16:15	RESERVED_16_15	RO	0x2	保留
14:12	UV1TH0	RW	0x1	1.2V 电源上升时撤销 VDD12UV1 的阈值 0: 0.94V 1: 0.97V 2: 0.99V 3: 1.02V 4: 1.04V 5: 1.07V 6: 1.10V 7: 1.14V

位段	位段名	属性	复位值	描述
11:9	UV1TH1	RW	0x0	1.2V 电源下降时触发 VDD12UV1 的阈值 0: 0.94V 1: 0.97V 2: 0.99V 3: 1.02V 4: 1.04V 5: 1.07V 6: 1.10V 7: 1.14V
8:7	RESERVED_8_7	RO	0x2	保留
6:4	UV0TH0	RW	0x1	1.2V 电源上升时撤销 VDD12UV0 的阈值 0: 0.94V 1: 0.97V 2: 0.99V 3: 1.02V 4: 1.04V 5: 1.07V 6: 1.10V 7: 1.14V
3:1	UV0TH1	RW	0x0	1.2V 电源下降时触发 VDD12UV0 的阈值 0: 0.94V 1: 0.97V 2: 0.99V 3: 1.02V 4: 1.04V 5: 1.07V 6: 1.10V 7: 1.14V
0	RESERVED_0	RW	0x1	保留

2.4.2.10 睡眠模式使能寄存器 (SLEEPKEY)

表 2-21: 睡眠模式使能寄存器 (SLEEPKEY) 位段定义

SLEEPKEY (Sleep Enable Key Register)							
Access: POWER->SLEEPKEY		Offset: 0x0028		Default: 0x00000000			
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

表 2-22: 睡眠模式使能寄存器 (SLEEPKEY) 位段描述

位段	位段名	属性	复位值	描述
31:0	KEY	RW	0x0	写 0x51ee9 令系统进入睡眠模式。

2.4.2.11 POWER 模块写使能寄存器 (PWRREGKEY)

表 2-23: POWER 模块写使能寄存器 (PWRREGKEY) 位段定义

PWRREGKEY (Power Register Write-Allow Key Register)							
Access: POWER->PWRREGKEY		Offset: 0x002C		Default: 0x1ACCE551			
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

表 2-24: POWER 模块写使能寄存器 (PWRREGKEY) 位段描述

位段	位段名	属性	复位值	描述
31:0	KEY	RW	0x1ACCE551	写入 0x1ACCE551 解锁受保护的 POWER 寄存器。

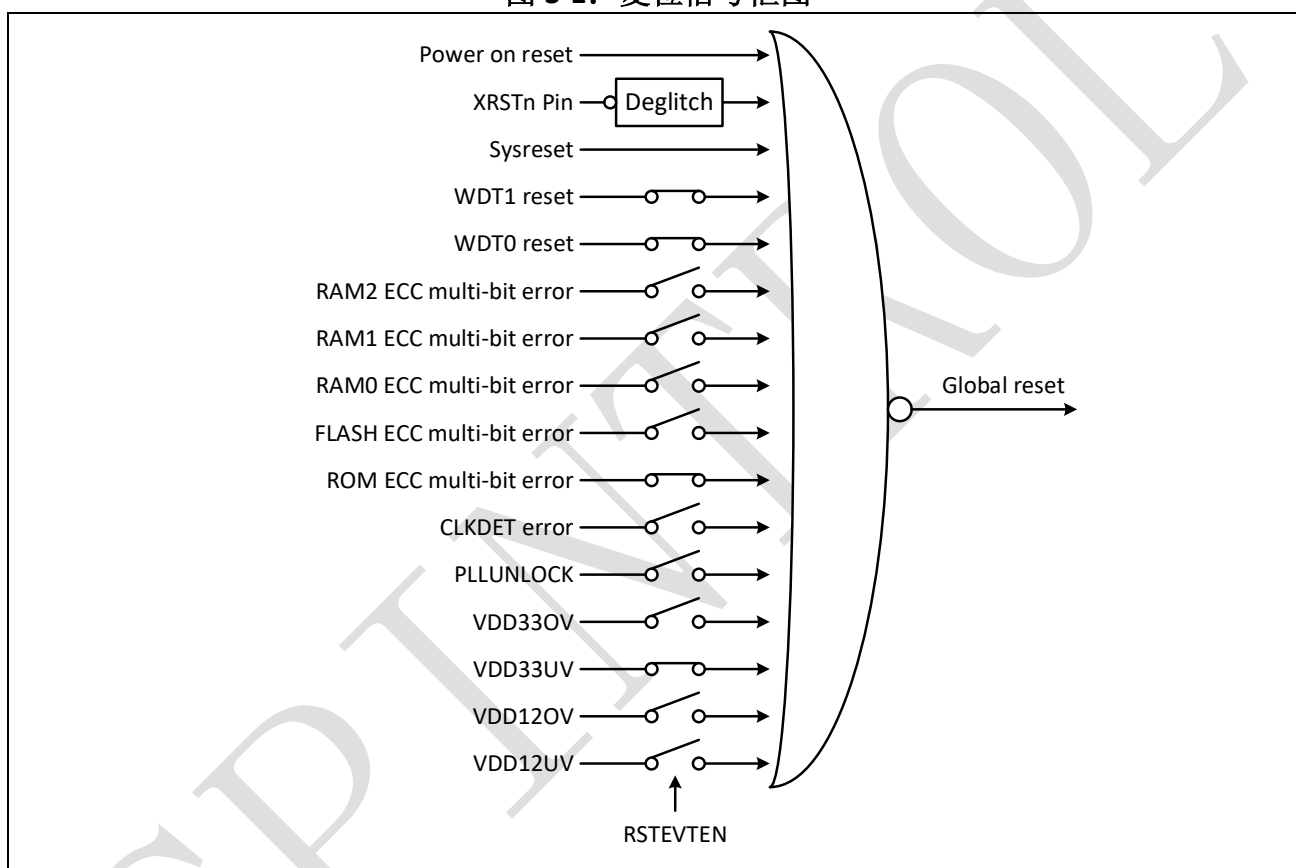
3 时钟与复位

3.1 复位架构

如图 3-1 所示，SPD1121 全局复位信号可以由 POR 上电复位请求、XRSTn 引脚复位请求以及功能模块复位事件触发。功能模块复位事件包括存储器 ECC 多位错误/奇偶校验错误、过压欠压事件、时钟错误事件、看门狗定时器超时复位、ARM® Cortex-M4F 系统复位请求。

功能复位源可通过 RSTEV TEN 寄存器对应位选择是否使能。为便于系统调试分析，复位发生时其触发事件将被锁存在 RSTEV TFLG 寄存器内，并可以通过 RSTEV TCLR 寄存器对应位清除。

图 3-1: 复位信号框图



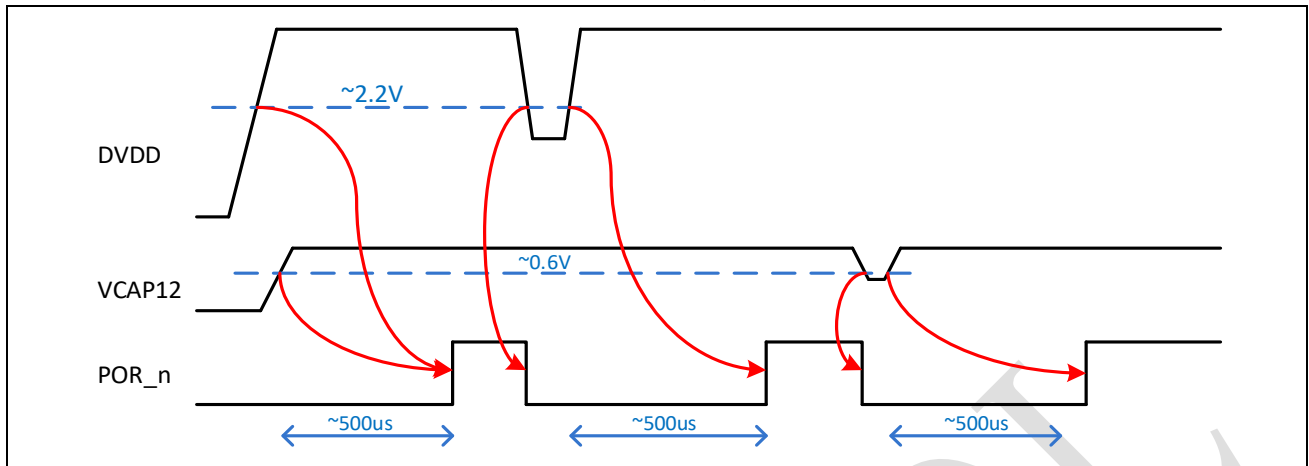
3.2 引脚复位

SPD1121 的 XRSTn 引脚提供了所有模块的全局逻辑复位，该引脚内部有上拉以确保芯片工作于正常状态。如图 3-1 所示，来自 XRSTn 引脚的信号会经过一个阈值可调整的除颤电路来消抖，仅当 XRSTn 保持为低的时间超过设定的阈值时才会触发有效的复位。默认的阈值为 500us。此外，若 XRSTn 保持为低持续时间超过约 8.2s，将进一步触发对电源系统的复位，产生与上电复位类似的效果。

3.3 上电复位

如图 3-2 所示，SPD1121 内建 POR 上电复位电路，确保芯片上电过程中时序无误，使芯片更易于使用。该复位在上电后会保持约 500us 才撤销，并且在 DVDD 低于 2.2V 或者 VCAP12 低于 0.6V 时会再次触发。

图 3-2: 上电复位时序图

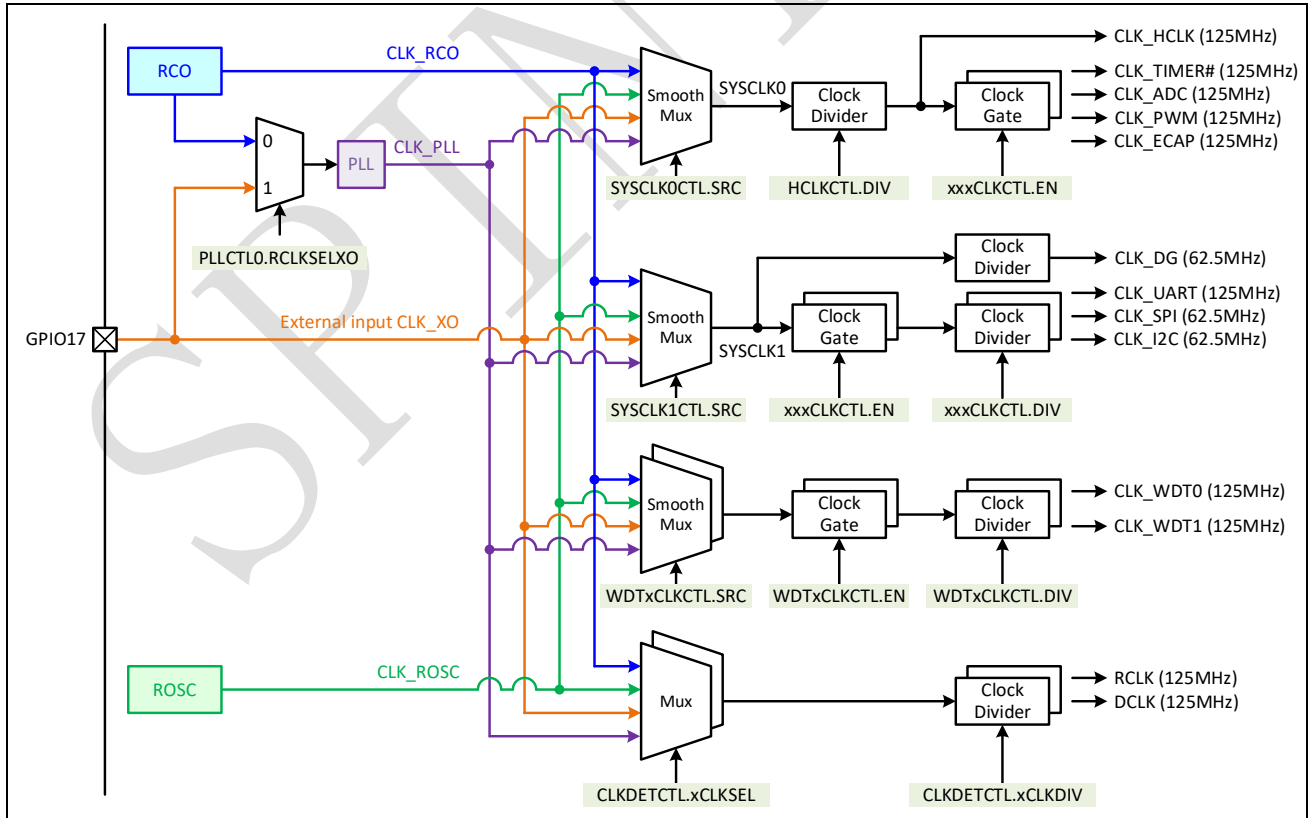


3.4 时钟结构

如图 3-3 所示，SPD1121 有如下 4 个时钟源：

- RCO：校准为 32MHz。
- ROSC：用于安全备份的 32MHz 环形振荡器，不校准。
- PLL：提供 25MHz ~ 125MHz 的灵活时钟，其参考时钟可选为 RCO 或者外部晶振时钟。
- 外部时钟：通过 GPIO17 输入。

图 3-3: SPD1121 时钟树



随后的平滑选择器可以实现无毛刺的时钟切换。配合分频器和使能控制，SPD1121 可以非常灵活地实现对每个模块时钟的配置。图 3-3 中以 CLK_HCLK 作为时钟的模块包括：CPU 及总线、ROM、RAM、FLASH、GPIO 模块、CRC 等。

注意： 可以通过 GPIO11/GPIO40 来观测时钟，具体参见 [IO 引脚功能通道定义](#)。

3.5 高频振荡器

SPD1121 内集成了两个高频振荡器。其中一个 RC 振荡器（RCO）在出厂时校准到 32MHz；另一个环形振荡器（ROSC）典型频率也为 32MHz，但不做校准，用作安全备用时钟。

3.6 PLL 时钟

图 3-4： PLL 结构总览

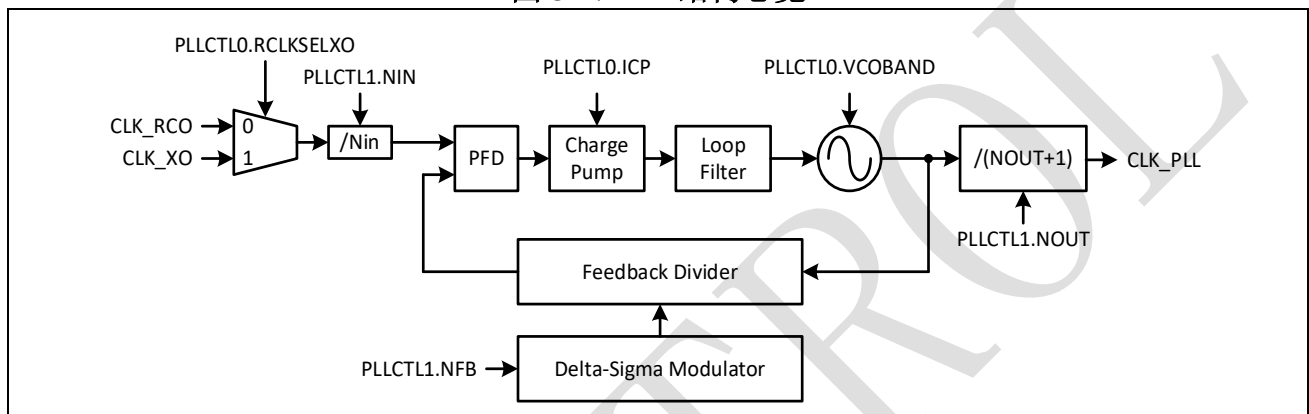


图 3-4 给出了 SPD1121 中所用 PLL（Phase-Locked Loop）的结构。它可以选择来自 RCO 或者外部晶振的时钟作为输入参考。当环路锁定时，VCO 频率在 400MHz 到 600MHz 范围内，随后经过分频器进一步降至 25MHz 到 125MHz 范围。具体输出频率计算公式为：

$$f_{PLL} = \frac{f_{REF}}{NIN} \times \frac{NFB}{65536} \times \frac{1}{NOUT + 1}$$

3.6.1 输入参考和分频

PLL 的输入参考时钟（ f_{REF} ）为 4MHz 到 56MHz，最终送至鉴频鉴相器 PFD 的输入（ f_{PFD} ）为从 4MHz 到 8MHz。因此，输入分频比（ f_{REF}/f_{PFD} ）应当如表 3-1 所示配置。此处的 NIN 是用户配置 PLL 时需要计算的第一个参数。

表 3-1： 输入分频比设置

输入频率（ f_{REF} ）		输入分频比（ $NIN = f_{REF} / f_{PFD}$ ）
4MHz	~ 8MHz	1
8MHz	~ 16MHz	2
16MHz	~ 24MHz	3
24MHz	~ 32MHz	4
32MHz	~ 40MHz	5
40MHz	~ 48MHz	6
48MHz	~ 56MHz	7

3.6.2 输出分频

VCO 的调频范围为 400MHz 到 600MHz，结合输出分频，最终时钟 CLK_PLL 的频率如表 3-2 所示。此处的 NOUT 是用户需要计算的第二个参数。

表 3-2: 输出时钟频率和分频比

输出频率 (f_{PLL})			分频比 ($f_{VCO}/f_{PLL} = N_{OUT} + 1$)
400MHz	~	600MHz	1
200MHz	~	300MHz	2
150MHz	~	200MHz	3
100MHz	~	150MHz	4
75MHz	~	100MHz	6
50MHz	~	75MHz	8
37.5MHz	~	50MHz	12
25MHz	~	37.5MHz	16

3.6.3 环路分频

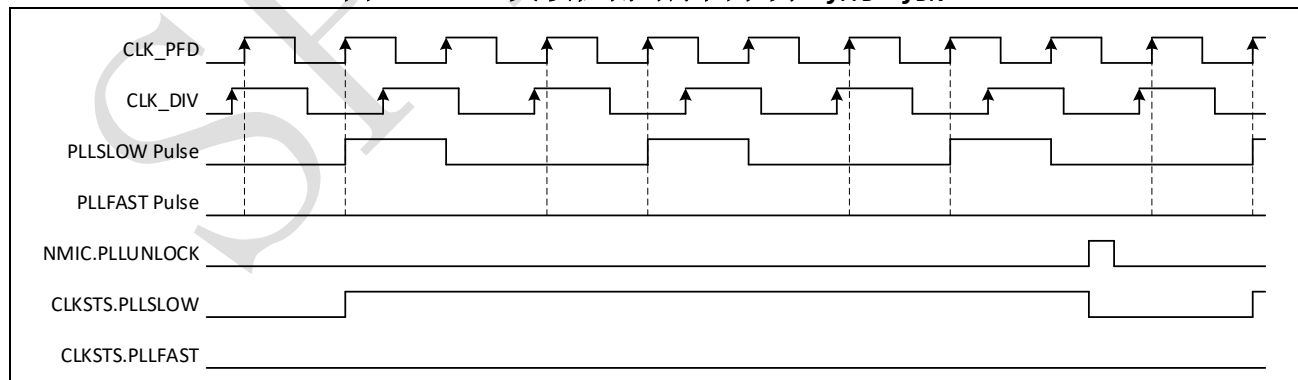
基于由表 3-1 和表 3-2 得到的 N_{IN} 和 N_{OUT} ，用户可以根据下面公式计算第三个参数：

$$NFB = \frac{f_{PLL} \times (N_{OUT} + 1)}{f_{REF} / N_{IN}} \times 65536$$

3.6.4 PLL 失锁检测

SPD1121 的 PLL 在 PFD 中内嵌了环路失锁检测电路，以提供环路状态侦测信息。具体原理为：对于 PFD 的两个输入，如果在某一个时钟相邻上升沿之间，另一个时钟出现了两个及以上的上升沿，则产生 PLLFAST 或 PLLSLOW 脉冲，并更新 CLKSTS 寄存器。用户可以通过对 CLKIC.PLLUNLOCK 写 1 来清除这些标志位。图 3-5 给出了 PFD 输入的参考时钟快于分频器输出时钟并使得 CLKSTS.PLLSLOW 置位的例子。

在 PLL 初始建立或者环路分频有很大改变的时候， f_{PFD} 和 f_{DIV} 有较大差别，导致有不可避免的 PLLFAST 或者 PLLSLOW 事件被锁存到 CLKSTS 寄存器中。为了避免误报，使用时需要等待约 10us 以待环路锁定，随后对 CLKIC.PLLUNLOCK 写 1 以清除建立过程中可能误报的失锁，然后才能根据需要通过 CLKIE.PLLUNLOCK 来使能 PLL 失锁事件中断。一旦 PLL 锁定，将不应当再出现 CLKSTS.PLLFAST 或 CLKSTS.PLLSLOW 事件。

图 3-5: PLL 失锁检测时序图示例 ($f_{PFD} > f_{DIV}$)

3.7 时钟平滑选择器

平滑选择器用于实现无毛刺的动态时钟切换。为了避免误切换到一个未开启的时钟并导致 CPU 卡死，直至目标时钟已经完全建立后，平滑选择器才会根据用户配置选择新的时钟，否则将一直选择当前时钟。如果当前选择的时钟源是 RCO 且 RCO 时钟失效，平滑选择器会选择 ROSC 作为当前时钟源，从而避免 CPU 卡死；与此同时，如果使能了时钟检测功能，将会

产生一个时钟检测错误事件。该事件可以根据 CLKIE 和 CLKTZE 寄存器相应位的配置来触发 CPU 中断或者 PWM 锁定事件。

3.8 时钟安全

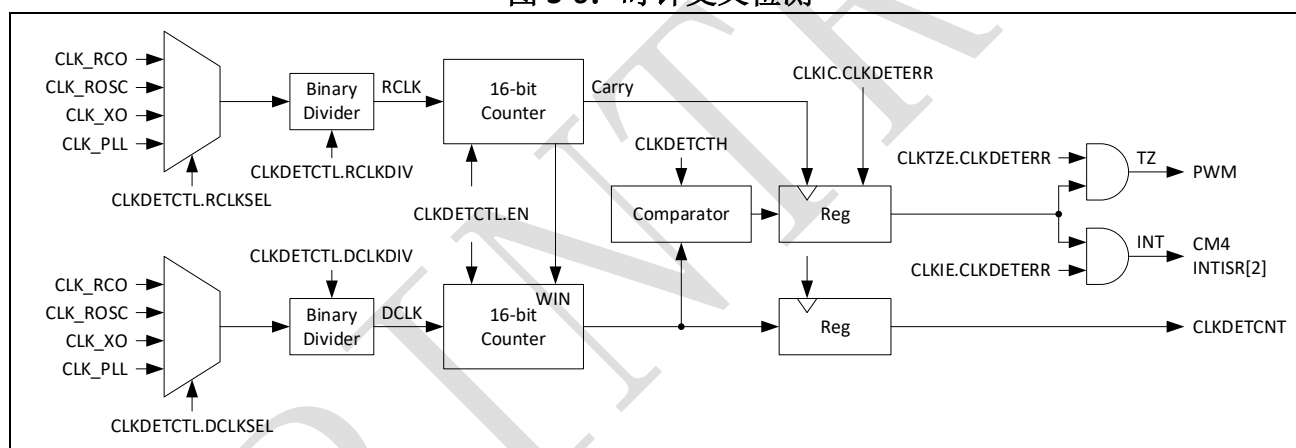
为了时钟安全，SPD1121 提供了时钟交叉检测模块。如图 3-6 所示，参考时钟和待检测时钟可以独立选择，随后各自经过一个二进制分频器得到 RCLK 和 DCLK。RCLK 用于产生观测窗口，在该窗口内 DCLK 保持计数。待观测窗口关闭后，DCLK 的最终计数值被锁存到 CLKDETCNT 寄存器，并与 CLKDETCTH 寄存器中的高低边界阈值进行比较，若计数越界则产生 CLKIF.CLKDETERR 标志位。计数值越界还可以根据 CLKIE 和 CLKTZE 寄存器相应位的配置来触发 CPU 中断或者 PWM 锁定事件。DCLK 可以通过配置 GPIO11.MUXSEL=4 或者 GPIO40.MUXSEL=3 后，在 GPIO11/GPIO40 引脚观测。具体参见 [IO 引脚功能通道定义](#)。

CLKDETCNT 的值可通过下面的公式计算：

$$\text{CLKDETCNT} = \frac{f_{\text{DCLK}}}{f_{\text{RCLK}}} \times 65536$$

其中， f_{DCLK} 和 f_{RCLK} 分别是 DCLK 和 RCLK 的频率，如图 3-6 所示。

图 3-6: 时钟交叉检测



3.9 寄存器

3.9.1 Clock 寄存器列表

表 3-3: CLOCK 模块基地址

外设模块	基地址
CLOCK	0x4000 0200

表 3-4: CLOCK 寄存器列表

寄存器	偏移地址	描述	复位值
CLKSTS	0x0000	时钟状态寄存器	0x00000003
RCOCTL*	0x0008	RCO 控制寄存器	0x00000001
PLLCTL0*	0x0014	PLL 控制寄存器 0	0x00000C10
PLLCTL1*	0x0018	PLL 控制寄存器 1	0x1C3E8000
CLKDECTL*	0x001C	Clock 检测控制寄存器	0x00000000
CLKDECTH*	0x0020	时钟检测计数阈值寄存器	0x00000000
CLKDETCNT	0x0024	时钟检测计数值寄存器	0x00000000
CLKIF	0x0028	时钟中断标志寄存器	0x00000000
CLKIC	0x002C	时钟中断清除寄存器	0x00000000
CLKIE*	0x0030	时钟中断使能寄存器	0x00000000
CLKTZE*	0x0034	时钟封锁事件使能寄存器	0x00000000
SYSCLK0CTL*	0x0038	SYSCLK0 控制寄存器	0x00000000
HCLKCTL*	0x003C	HCLK 时钟控制寄存器	0x00000000
ADCCLKCTL*	0x0040	ADC 时钟控制寄存器	0x00000000
PWMCLKCTL*	0x0044	PWM 时钟控制寄存器	0x00000000
ECAPCLKCTL*	0x0048	ECAP 时钟控制寄存器	0x00000000
TMR0CLKCTL*	0x004C	Timer 0 时钟控制寄存器	0x00000000
TMR1CLKCTL*	0x0050	Timer 1 时钟控制寄存器	0x00000000
SYSCLK1CTL*	0x005C	SYSCLK1 时钟控制寄存器	0x00000000
PCLKCTL*	0x0060	PCLK 时钟控制寄存器	0x00000000
DGCLKCTL*	0x0064	消抖时钟控制寄存器	0x00000200
UARTCLKCTL*	0x0068	UART 时钟控制寄存器	0x00000000
SPICLKCTL*	0x006C	SPI 时钟控制寄存器	0x00000000
I2CCLKCTL*	0x0070	I2C 时钟控制寄存器	0x00000000
WDT0CLKCTL*	0x0074	WDT0 时钟控制寄存器	0x00000100
WDT1CLKCTL*	0x0078	WDT1 时钟控制寄存器	0x00000300
CLKREGKEY	0x007C	Clock 模块写使能寄存器	0x1ACCE551

注意： 由*标记的寄存器仅当 CLKREGKEY=0x1ACCE551 才可以改写。

3.9.2 Clock 寄存器

表 3-5 到 3-58 给出了 CLOCK 模块各寄存器的定义。

3.9.2.1 时钟状态寄存器（CLKSTS）

表 3-5：时钟状态寄存器（CLKSTS）位段定义

CLKSTS (Clock Status Register)							
Access: CLOCK->CLKSTS Offset: 0x0000 Default: 0x00000003							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							CLKDETERR
7	6	5	4	3	2	1	0
PLLSLOW	PLLFAST	PLLFCALACK	PLLFCALHIT	PLLRDY	XORDY	ROSCRDY	RCORDY

表 3-6：时钟状态寄存器（CLKSTS）位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8	CLKDETERR	RO	0x0	CLKDET 检测到时钟错误 0: 事件未发生 1: 事件已发生
7	PLLSLOW	RO	0x0	PLL 失锁且频率偏慢标志，可能的原因包括： - $f_{REF}/NIN*NFB/65536$ 高于 VCO 频率上限 - VCO 未能起振 - 环路分频器异常，导致分频器没有输出或者输出频率低于预期 0: 事件未发生 1: 事件已发生
6	PLLFAST	RO	0x0	PLL 失锁且频率偏慢标志，可能的原因包括： - $f_{REF}/NIN*NFB/65536$ 低于 VCO 频率下限 - 参考时钟丢失 - 环路分频器异常，导致分频器输出频率高于预期 0: 事件未发生 1: 事件已发生
5	VCOFREQVLD	RO	0x0	VCOFREQ 位段读数有效标志 0: VCOFREQ 位段读数无效 1: VCOFREQ 位段读数有效
4	VCOFREQ	RO	0x0	VCO 频率标志位（用于校准）

位段	位段名	属性	复位值	描述
				0: VCO 频率偏慢 1: VCO 频率偏快
3	PLLRDY	RO	0x0	PLL 时钟就绪标志 0: 未就绪 1: 就绪
2	XORDY	RO	0x0	XO 时钟就绪标志 0: 未就绪 1: 就绪
1	ROSCRDY	RO	0x1	ROSC 时钟就绪标志 0: 未就绪 1: 就绪
0	RCORDY	RO	0x1	RCO 时钟就绪标志 0: 未就绪 1: 就绪

3.9.2.2 RCO 控制寄存器 (RCOCTL)

表 3-7: RCO 控制寄存器 (RCOCTL) 位段定义

RCOCTL (RCO Control Register) Access: CLOCK->RCOCTL Offset: 0x0008 Default: 0x00000001							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							EN

表 3-8: RCO 控制寄存器 (RCOCTL) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED_31_1	RO	0x0	保留
0	EN	RW	0x1	RCO 使能 0: 关闭 1: 使能

3.9.2.3 PLL 控制寄存器 0 (PLLCTL0)

表 3-9: PLL 控制寄存器 0 (PLLCTL0) 位段定义

PLLCTL0 (PLL Control Register 0)							
Access: CLOCK->PLLCTL0 Offset: 0x0014 Default: 0x00000C10							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED		ICP					RESERVED
7	6	5	4	3	2	1	0
RCLKSELXO	RESERVED	VCOBAND		FCALREQ	FCALEN	RESERVED	EN

表 3-10: PLL 控制寄存器 0 (PLLCTL0) 位段描述

位段	位段名	属性	复位值	描述
31:14	RESERVED_31_14	RO	0x0	保留
13:9	ICP	RW	0x6	PLL 电荷泵电流设置
8	RESERVED_8	RO	0x0	保留
7	RCLKSELXO	RW	0x0	PLL 参考时钟选择 0: CLK_RCO (RC 振荡器) 1: CLK_XO (外部时钟)
6	RESERVED_6	RO	0x0	保留
5:4	VCOBAND	RW	0x1	PLL VCO 频段选择 0: 大约 350 ~ 450MHz 1: 大约 450 ~ 550MHz 2: 大约 550 ~ 650MHz 3: 大约 600 ~ 700MHz
3	FCALREQ	RW	0x0	频率校准单轮测试请求 对每一轮, 将其设置为 1 两次, 直到 CLKSTS.PLLFCALACK=1, 本位段自动清零。 0: 时间窗关闭 1: 时间窗开启
2	FCALEN	RW	0x0	锁相环频率校准使能 设置为 1 启用校准。在找到目标码值后, 需要将其清除为 0 以关闭校准。 0: 关闭 1: 使能
1	RESERVED_1	RO	0x0	保留
0	EN	RW	0x0	PLL 使能 0: 关闭 1: 使能

3.9.2.4 PLL 控制寄存器 1 (PLLCTL1)

表 3-11: PLL 控制寄存器 1 (PLLCTL1) 位段定义

PLLCTL1 (PLL Control Register 1) Access: CLOCK->PLLCTL1 Offset: 0x0018 Default: 0x1C3E8000							
31 RESERVED	30	29	28	27	26	25	24
NOUT				NIN			
23	22	21	20	19	18	17	16
NFB							
15	14	13	12	11	10	9	8
NFB							
7	6	5	4	3	2	1	0
NFB							

表 3-12: PLL 控制寄存器 1 (PLLCTL1) 位段描述

位段	位段名	属性	复位值	描述
31	RESERVED_31	RO	0x0	保留
30:27	NOUT	RW	0x3	设置 VCO 到 PLL 输出的分频比(f_{VCO}/f_{PLL})为 NOUT+1
26:24	NIN	RW	0x4	设置输入参考时钟的分频比(f_{REF}/f_{PFD})为 NIN 1: 若 f_{REF} 在[4MHz, 8MHz]区间, 则设置为 1 2: 若 f_{REF} 在(8MHz, 16MHz]区间, 则设置为 2 3: 若 f_{REF} 在(16MHz, 24MHz]区间, 则设置为 3 4: 若 f_{REF} 在(24MHz, 32MHz]区间, 则设置为 4 5: 若 f_{REF} 在(32MHz, 40MHz]区间, 则设置为 5 6: 若 f_{REF} 在(40MHz, 48MHz]区间, 则设置为 6 7: 若 f_{REF} 在(48MHz, 56MHz]区间, 则设置为 7
23:0	NFB	RW	0x3E8000	设置环路反馈的分频比(f_{VCO}/f_{PFD})为 NFB/65536

3.9.2.5 时钟检测控制寄存器（CLKDETCTL）

表 3-13: 时钟检测控制寄存器（CLKDETCTL）位段定义

CLKDETCTL (Clock Detection Control Register) Access: CLOCK->CLKDETCTL Offset: 0x001C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED					EN	DCLKDIV	
7	6	5	4	3	2	1	0
DCLKDIV	DCLKSEL		RCLKDIV			RCLKSEL	

表 3-14: 时钟检测控制寄存器（CLKDETCTL）位段描述

位段	位段名	属性	复位值	描述
31:11	RESERVED_31_11	RO	0x0	保留
10	EN	RW	0x0	时钟检测使能 关闭时钟检测时，之前的错误事件自动清零 0: 关闭 1: 使能
9:7	DCLKDIV	RW	0x0	待检测时钟的分频比 0: 1 分频（不分频） 1: 2 分频 2: 4 分频 3: 8 分频 4: 16 分频 5: 32 分频 6: 64 分频 7: 128 分频
6:5	DCLKSEL	RW	0x0	待检测时钟选择 0: RCO 时钟 1: ROSC 时钟 2: XO 外部时钟输入 3: PLL 时钟
4:2	RCLKDIV	RW	0x0	参考时钟的分频比 0: 1 分频（不分频） 1: 2 分频 2: 4 分频 3: 8 分频 4: 16 分频 5: 32 分频

位段	位段名	属性	复位值	描述
				6: 64 分频 7: 128 分频
1:0	RCLKSEL	RW	0x0	参考时钟选择 0: RCO 时钟 1: ROSC 时钟 2: XO 外部时钟输入 3: PLL 时钟

3.9.2.6 时钟检测计数阈值寄存器（CLKDETCTH）

表 3-15: 时钟检测计数阈值寄存器（CLKDETCTH）位段定义

CLKDETCTH (Clock Detection Counter Threshold Register) Access: CLOCK->CLKDETCTH Offset: 0x0020 Default: 0x00000000							
31	30	29	28	27	26	25	24
HI							
23	22	21	20	19	18	17	16
HI							
15	14	13	12	11	10	9	8
LO							
7	6	5	4	3	2	1	0
LO							

表 3-16: 时钟检测计数阈值寄存器（CLKDETCTH）位段描述

位段	位段名	属性	复位值	描述
31:16	HI	RW	0x0	时钟检测计数值上边界 计数值高于上边界时，则可以触发 PWM 封锁事件、时钟中断和芯片复位
15:0	LO	RW	0x0	时钟检测计数值下边界 计数值低于下边界时，则可以触发 PWM 封锁事件、时钟中断和芯片复位

3.9.2.7 时钟检测计数值寄存器（CLKDETCNT）

表 3-17: 时钟检测计数值寄存器（CLKDETCNT）位段定义

CLKDETCNT (Clock Detection Counter Register)							
Access: CLOCK->CLKDETCNT Offset: 0x0024 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 3-18: 时钟检测计数值寄存器（CLKDETCNT）位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	VAL	RO	0x0	时钟检测最终计数值

3.9.2.8 时钟中断标志寄存器 (CLKIF)

表 3-19: 时钟中断标志寄存器 (CLKIF) 位段定义

CLKIF (Clock Interrupt Flag Register)							
Access: CLOCK->CLKIF		Offset: 0x0028		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED					CLKINT	CLKDETERR	PLLUNLOCK

表 3-20: 时钟中断标志寄存器 (CLKIF) 位段描述

位段	位段名	属性	复位值	描述
31:3	RESERVED_31_3	RO	0x0	保留
2	CLKINT	RO	0x0	全局时钟中断标志 0: 时钟中断未发生 1: 时钟中断曾发生 该位被清零前不会产生新的中断。
1	CLKDETERR	RO	0x0	CLKDET 模块检测到时钟错误标志 0: CLKDETCNT 在 CLKDETCTH 指定的范围内 1: CLKDETCNT 超出 CLKDETCTH 指定的范围 该位段清零前, 相同事件不会再触发中断。
0	PLLUNLOCK	RO	0x0	PLL 失锁标志, 细节可查询 CLKSTS.PLLFAST 和 CLKSTS.PLLSLOW 0: PLL 锁定 1: PLL 失锁 该位段清零前, PLL 失锁不会再触发中断。

3.9.2.9 时钟中断清除寄存器（CLKIC）

表 3-21: 时钟中断清除寄存器（CLKIC）位段定义

CLKIC (Clock Interrupt Clear Register)							
Access: CLOCK->CLKIC Offset: 0x002C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED					CLKINT	CLKDETERR	PLLUNLOCK

表 3-22: 时钟中断清除寄存器（CLKIC）位段描述

位段	位段名	属性	复位值	描述
31:3	RESERVED_31_3	RO	0x0	保留
2	CLKINT	W1C	0x0	全局时钟中断清除 0: 写 0 无效，总是读回 0 1: 写 1 清除标志位，本位段自动清零
1	CLKDETERR	W1C	0x0	CLKDET 错误中断清除 0: 写 0 无效，总是读回 0 1: 写 1 清除标志位，本位段自动清零
0	PLLUNLOCK	W1C	0x0	PLL 失锁中断清除 0: 写 0 无效，总是读回 0 1: 写 1 清除标志位和 CLKSTS 中的 PLLFAST、PLLSLOW 状态位，本位段自动清零

3.9.2.10 时钟中断使能寄存器（CLKIE）

表 3-23: 时钟中断使能寄存器（CLKIE）位段定义

CLKIE (Clock Interrupt Enable Register)							
Access: CLOCK->CLKIE		Offset: 0x0030		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED						CLKDETERR	PLLUNLOCK

表 3-24: 时钟中断使能寄存器（CLKIE）位段描述

位段	位段名	属性	复位值	描述
31:2	RESERVED_31_2	RO	0x0	保留
1	CLKDETERR	RW	0x0	CLKDET 错误中断使能 0: CLKDET 错误不触发中断 1: CLKDET 错误触发中断
0	PLLUNLOCK	RW	0x0	PLL 失锁中断使能 0: PLL 失锁不触发中断 1: PLL 失锁触发中断

3.9.2.11 时钟封锁事件使能寄存器（CLKTZE）

表 3-25: 时钟封锁事件使能寄存器（CLKTZE）位段定义

CLKTZE (Clock Trip-zone Event Enable Register)							
Access: CLOCK->CLKTZE Offset: 0x0034 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED						CLKDETERR	PLLUNLOCK

表 3-26: 时钟封锁事件使能寄存器（CLKTZE）位段描述

位段	位段名	属性	复位值	描述
31:2	RESERVED_31_2	RO	0x0	保留
1	CLKDETERR	RW	0x0	CLKDET 错误触发 PWM 封锁事件使能 0: 关闭 1: 使能
0	PLLUNLOCK	RW	0x0	PLL 失锁触发 PWM 封锁事件使能 0: 关闭 1: 使能

3.9.2.12 SYCLK0 控制寄存器 (SYCLK0CTL)

表 3-27: SYCLK0 控制寄存器 (SYCLK0CTL) 位段定义

SYCLK0CTL (SYCLK0 Control Register) Access: CLOCK->SYCLK0CTL Offset: 0x0038 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED					SRC		RESERVED
7	6	5	4	3	2	1	0
RESERVED							

表 3-28: SYCLK0 控制寄存器 (SYCLK0CTL) 位段描述

位段	位段名	属性	复位值	描述
31:11	RESERVED_31_11	RO	0x0	保留
10:9	SRC	RW	0x0	时钟源选择 0: RCO 时钟 1: ROSC 时钟 2: XO 外部时钟输入 3: PLL 时钟
8:0	RESERVED_8_0	RO	0x0	保留

3.9.2.13 HCLK 控制寄存器 (HCLKCTL)

表 3-29: HCLK 控制寄存器 (HCLKCTL) 位段定义

HCLKCTL (HCLK Control Register) Access: CLOCK->HCLKCTL Offset: 0x003C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
DIV							

表 3-30: HCLK 控制寄存器 (HCLKCTL) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7:0	DIV	RW	0x0	设定相对于 SYCLK0 的分频比为(DIV + 1)

3.9.2.14 ADC 时钟控制寄存器（ADCCLKCTL）

表 3-31: ADC 时钟控制寄存器（ADCCLKCTL）位段定义

ADCCLKCTL (ADC Clock Control Register)							
Access: CLOCK->ADCCLKCTL Offset: 0x0040 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							EN
7	6	5	4	3	2	1	0
DIV							

表 3-32: ADC 时钟控制寄存器（ADCCLKCTL）位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8	EN	RW	0x0	时钟输出使能 0: 关闭 1: 使能
7:0	DIV	RO	0x0	回读值与 HCLKCTL.DIV 相同

3.9.2.15 PWM 时钟控制寄存器（PWMCLKCTL）

表 3-33: PWM 时钟控制寄存器（PWMCLKCTL）位段定义

PWMCLKCTL (PWM Clock Control Register)							
Access: CLOCK->PWMCLKCTL Offset: 0x0044 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							EN
7	6	5	4	3	2	1	0
DIV							

表 3-34: PWM 时钟控制寄存器（PWMCLKCTL）位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8	EN	RW	0x0	时钟输出使能 0: 关闭 1: 使能
7:0	DIV	RO	0x0	回读值与 HCLKCTL.DIV 相同

3.9.2.16 ECAP 时钟控制寄存器 (ECAPCLKCTL)

表 3-35: ECAP 时钟控制寄存器 (ECAPCLKCTL) 位段定义

ECAPCLKCTL (ECAP Clock Control Register)							
Access: CLOCK->ECAPCLKCTL				Offset: 0x0048		Default: 0x00000000	
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							EN
7	6	5	4	3	2	1	0
DIV							

表 3-36: ECAP 时钟控制寄存器 (ECAPCLKCTL) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8	EN	RW	0x0	时钟输出使能 0: 关闭 1: 使能
7:0	DIV	RO	0x0	回读值与 HCLKCTL.DIV 相同

3.9.2.17 Timer 0 时钟控制寄存器 (TMR0CLKCTL)

表 3-37: Timer 0 时钟控制寄存器 (TMR0CLKCTL) 位段定义

TMR0CLKCTL (Timer 0 Clock Control Register)							
Access: CLOCK->TMR0CLKCTL				Offset: 0x004C		Default: 0x00000000	
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							EN
7	6	5	4	3	2	1	0
DIV							

表 3-38: Timer 0 时钟控制寄存器 (TMR0CLKCTL) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8	EN	RW	0x0	时钟输出使能 0: 关闭 1: 使能
7:0	DIV	RO	0x0	回读值与 HCLKCTL.DIV 相同

3.9.2.18 Timer 1 时钟控制寄存器 (TMR1CLKCTL)

表 3-39: Timer 1 时钟控制寄存器 (TMR1CLKCTL) 位段定义

TMR1CLKCTL (Timer 1 Clock Control Register)							
Access: CLOCK->TMR1CLKCTL Offset: 0x0050 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							EN
7	6	5	4	3	2	1	0
DIV							

表 3-40: Timer 1 时钟控制寄存器 (TMR1CLKCTL) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8	EN	RW	0x0	时钟输出使能 0: 关闭 1: 使能
7:0	DIV	RO	0x0	回读值与 HCLKCTL.DIV 相同

3.9.2.19 SYCLK1 控制寄存器 (SYCLK1CTL)

表 3-41: SYCLK1 控制寄存器 (SYCLK1CTL) 位段定义

SYCLK1CTL (SYCLK1 Control Register) Access: CLOCK->SYCLK1CTL Offset: 0x005C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED					SRC		RESERVED
7	6	5	4	3	2	1	0
RESERVED							

表 3-42: SYCLK1 控制寄存器 (SYCLK1CTL) 位段描述

位段	位段名	属性	复位值	描述
31:11	RESERVED_31_11	RO	0x0	保留
10:9	SRC	RW	0x0	时钟源选择 0: RCO 时钟 1: ROSC 时钟 2: XO 外部时钟输入 3: PLL 时钟
8:0	RESERVED_8_0	RO	0x0	保留

3.9.2.20 PCLK 控制寄存器 (PCLKCTL)

表 3-43: PCLK 控制寄存器 (PCLKCTL) 位段定义

PCLKCTL (PCLK Control Register) Access: CLOCK->PCLKCTL Offset: 0x0060 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
DIV							

表 3-44: PCLK 控制寄存器 (PCLKCTL) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7:0	DIV	RW	0x0	设定相对于 SYCLK1 的分频比为(DIV + 1)

3.9.2.21 消抖时钟控制寄存器（DGCLKCTL）

表 3-45：消抖时钟控制寄存器（DGCLKCTL）位段定义

DGCLKCTL (Deglitch Clock Control Register)							
Access: CLOCK->DGCLKCTL Offset: 0x0064 Default: 0x00000200							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED					PDIV		
7	6	5	4	3	2	1	0
DIV							

表 3-46：消抖时钟控制寄存器（DGCLKCTL）位段描述

位段	位段名	属性	复位值	描述
31:11	RESERVED_31_11	RO	0x0	保留
10:8	PDIV	RW	0x2	相对 SYSCLK1 的预分频比 0: 不分频 1: 2 分频 2: 4 分频 3: 8 分频 4: 16 分频 5: 32 分频 6: 64 分频 7: 128 分频
7:0	DIV	RW	0x0	时钟分频比 整体分频比为(DIV + 1) x 2^PDIV

3.9.2.22 UART 时钟控制寄存器 (UARTCLKCTL)

表 3-47: UART 时钟控制寄存器 (UARTCLKCTL) 位段定义

UARTCLKCTL (UART Clock Control Register)							
Access: CLOCK->UARTCLKCTL				Offset: 0x0068		Default: 0x00000000	
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							EN
7	6	5	4	3	2	1	0
DIV							

表 3-48: UART 时钟控制寄存器 (UARTCLKCTL) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8	EN	RW	0x0	时钟输出使能 0: 关闭 1: 使能
7:0	DIV	RW	0x0	设定相对于 SYSCLK1 的分频比为(DIV + 1)

3.9.2.23 SPI 时钟控制寄存器 (SPICLKCTL)

表 3-49: SPI 时钟控制寄存器 (SPICLKCTL) 位段定义

SPICLKCTL (SPI Clock Control Register)							
Access: CLOCK->SPICLKCTL				Offset: 0x006C		Default: 0x00000000	
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							EN
15	14	13	12	11	10	9	8
DIV							
7	6	5	4	3	2	1	0
DIV							

表 3-50: SPI 时钟控制寄存器 (SPICLKCTL) 位段描述

位段	位段名	属性	复位值	描述
31:17	RESERVED_31_17	RO	0x0	保留
16	EN	RW	0x0	时钟输出使能 0: 关闭 1: 使能
15:0	DIV	RW	0x0	设定相对于 SYSCLK1 的分频比为(DIV + 1)

3.9.2.24 I2C 时钟控制寄存器 (I2CCLKCTL)

表 3-51: I2C 时钟控制寄存器 (I2CCLKCTL) 位段定义

I2CCLKCTL (I2C Clock Control Register)							
Access: CLOCK->I2CCLKCTL Offset: 0x0070 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							EN
7	6	5	4	3	2	1	0
DIV							

表 3-52: I2C 时钟控制寄存器 (I2CCLKCTL) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8	EN	RW	0x0	时钟输出使能 0: 关闭 1: 使能
7:0	DIV	RW	0x0	设定相对于 SYSCLK1 的分频比为(DIV + 1)

3.9.2.25 WDT0 时钟控制寄存器 (WDT0CLKCTL)

表 3-53: WDT0 时钟控制寄存器 (WDT0CLKCTL) 位段定义

WDT0CLKCTL (WDT0 Clock Control Register)							
Access: CLOCK->WDT0CLKCTL Offset: 0x0074 Default: 0x00000100							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED					SRC		EN
7	6	5	4	3	2	1	0
DIV							

表 3-54: WDT0 时钟控制寄存器 (WDT0CLKCTL) 位段描述

位段	位段名	属性	复位值	描述
31:11	RESERVED_31_11	RO	0x0	保留
10:9	SRC	RW	0x0	时钟源选择 0: RCO 时钟 1: ROSC 时钟 2: XO 外部时钟输入 3: PLL 时钟
8	EN	RW	0x1	时钟输出使能 0: 关闭 1: 使能
7:0	DIV	RW	0x0	设定相对于时钟源的分频比为(DIV + 1)

3.9.2.26 WDT1 时钟控制寄存器 (WDT1CLKCTL)

表 3-55: WDT1 时钟控制寄存器 (WDT1CLKCTL) 位段定义

WDT1CLKCTL (WDT1 Clock Control Register)							
Access: CLOCK->WDT1CLKCTL Offset: 0x0078 Default: 0x00000300							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED				SRC		EN	
7	6	5	4	3	2	1	0
DIV							

表 3-56: WDT1 时钟控制寄存器 (WDT1CLKCTL) 位段描述

位段	位段名	属性	复位值	描述
31:11	RESERVED_31_11	RO	0x0	保留
10:9	SRC	RW	0x1	时钟源选择 0: RCO 时钟 1: ROSC 时钟 2: XO 外部时钟输入 3: PLL 时钟
8	EN	RW	0x1	时钟输出使能 0: 关闭 1: 使能
7:0	DIV	RW	0x0	设定相对于时钟源的分频比为(DIV + 1)

3.9.2.27 CLOCK 模块写使能寄存器（CLKREGKEY）

表 3-57: CLOCK 模块写使能寄存器（CLKREGKEY）位段定义

CLKREGKEY (Clock Register Write-Allow Key Register)							
Access: CLOCK->CLKREGKEY Offset: 0x007C Default: 0x1ACCE551							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

表 3-58: CLOCK 模块写使能寄存器（CLKREGKEY）位段描述

位段	位段名	属性	复位值	描述
31:0	KEY	RW	0x1ACCE551	写入 0x1ACCE551 解锁受保护的 CLOCK 寄存器

4 通用 I/O（GPIO）

4.1 概述

SPD1121 具有 14 个 I/O 引脚（在芯片启动后 ISP 引脚也可以用作 GPIO40）。这些 I/O 引脚是多路复用的，因而可以被配置为 GPIO 功能或者任意一个可选功能，每个 I/O 引脚都可以独立控制，分为如下两类：

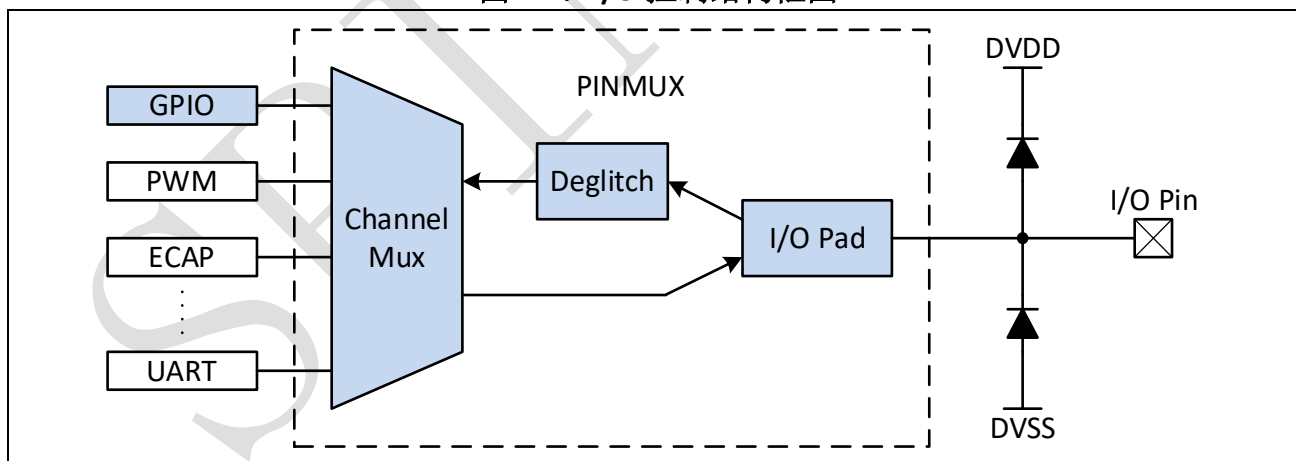
- 3.3V I/O，与模拟输入功能复用
 - GPIO0~GPIO4、GPIO10~GPIO11：可以用作模拟输入通道。
 - GPIO17：可以用作外部时钟输入。
- 3.3V I/O，纯数字功能通道复用
 - GPIO16、GPIO34~GPIO35、GPIO38~GPIO40。

本章节详细描述 I/O 引脚的控制，包括引脚复用和 GPIO 相关的功能。I/O 控制的结构框图如图 4-1 所示，由两部分组成：

- PINMUX 模块
 - I/O Pad：上拉和下拉控制，输入/输出控制，施密特输入以及输出驱动强度控制。
 - Deglitch：输入消抖控制。
 - MUX：引脚多路复用。
- GPIO 模块
 - GPIO 功能 IP 控制。

每个 I/O 引脚都可以独立配置和使用，且均有一个从 I/O 到 DVDD 和一个从 DVSS 到 I/O 的二极管用作 ESD 保护

图 4-1：I/O 控制结构框图

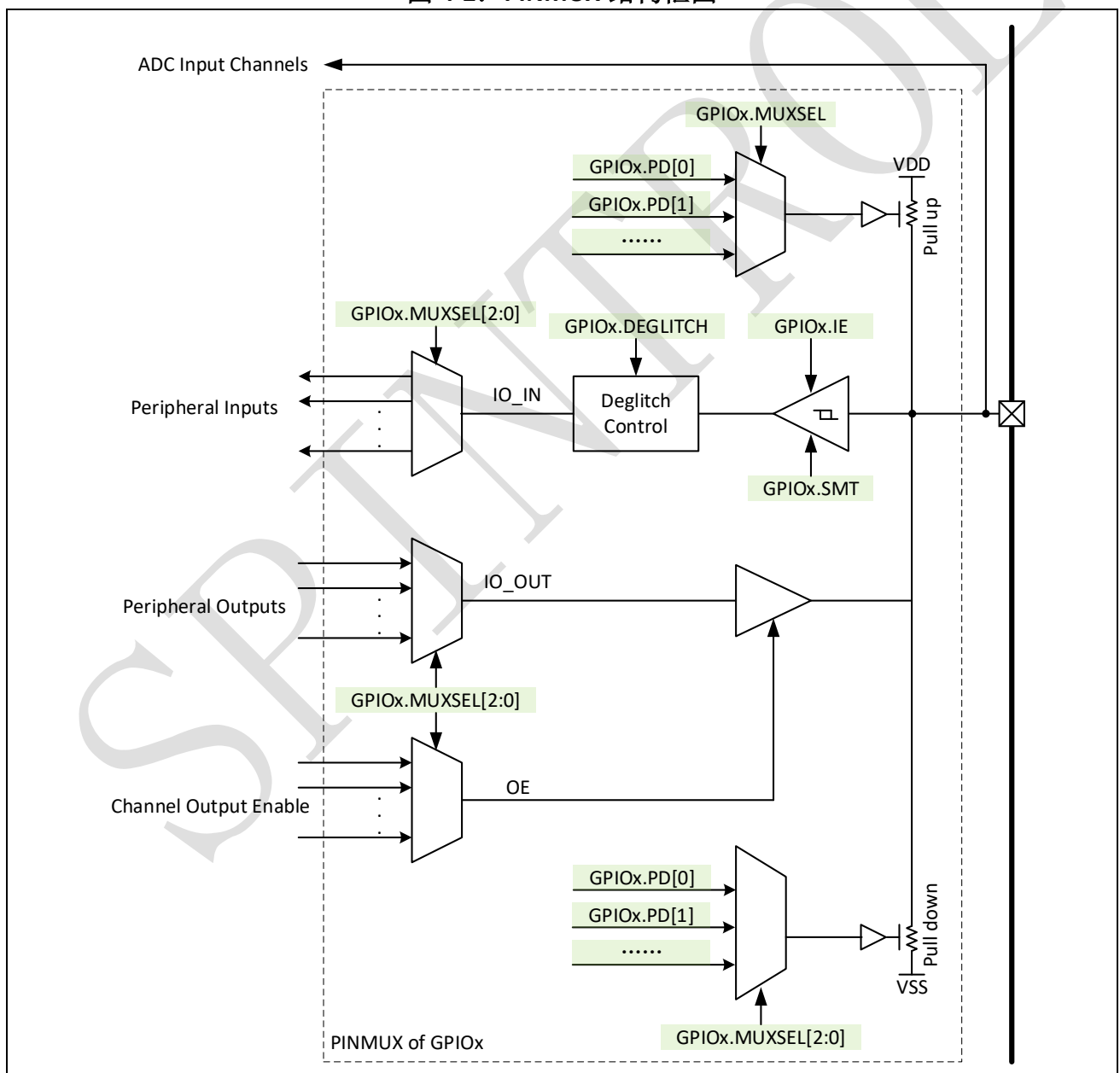


4.2 引脚复用 (PINMUX)

PINMUX 的结构框图如图 4-2 所示。每个 I/O 引脚的多路复用最多支持 8 个功能通道。这些功能通道可能用作输入或输出。当 I/O 处于输入模式下，输出缓冲器禁用并处于高阻态。输入缓冲器在 I/O 用作模拟电压输入时被禁用以避免潜在的漏电，其余情况下均被使能。输入信号可以通过一个滤波器来实现消除可能的毛刺。

GPIOx.MUXSEL 位段用来进行通道选择，从而决定该引脚的输入输出特性以及上下拉功能。需要注意的是，当 GPIOx.MUXSEL 配置为模拟输入时，GPIOx 的输入输出特性以及上下拉功能配置无效。当 GPIOx.MUXSEL 配置为数字功能时，上下拉控制是互斥的。例如，若 GPIO10.MUXSEL = 2，则选择通道 2 为输出功能，并且 GPIO10 引脚的上拉和下拉控制分别由 GPIO10.PU2 和 GPIO10.PD2 决定。

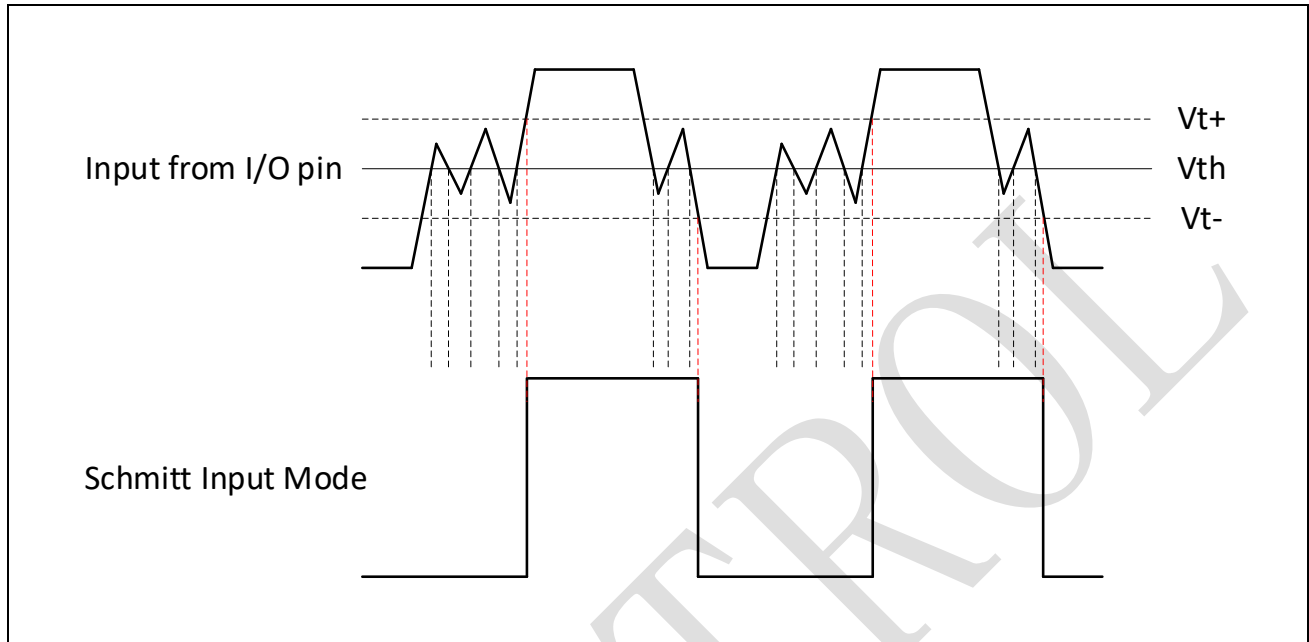
图 4-2: PINMUX 结构框图



4.2.1 施密特输入模式

如图4-3所示，施密特输入模式始终使能，该模式能够防止输入信号上小的波动（噪声）引起的误触发。

图 4-3：施密特输入模式

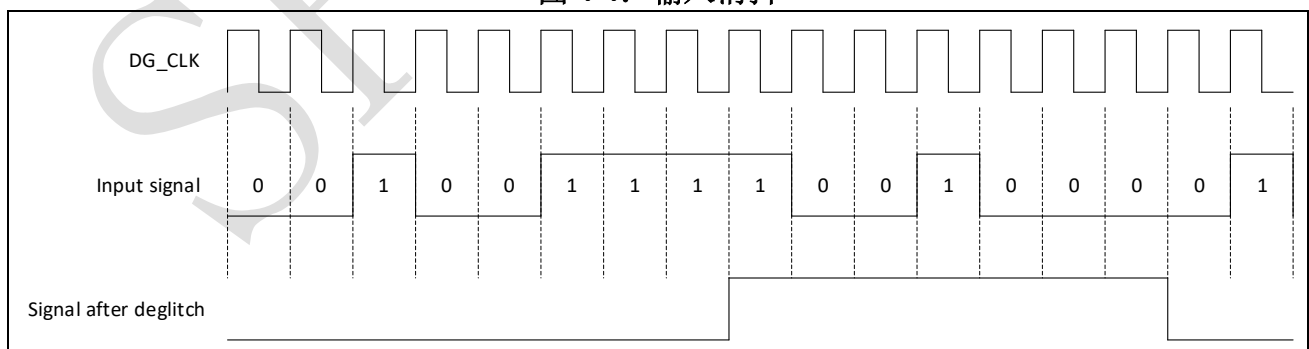


1. V_{t+} – 施密特输入模式下，输入上翻转阈值； V_{t-} – 施密特输入模式下，输入下翻转阈值。
2. V_{th} 的典型值为 1.5V， V_{t+} 的典型值为 1.67V， V_{t-} 的典型值为 1.38V。

4.2.2 消抖控制

输入消抖功能可以通过配置寄存器 GPIOx.DEGLITCH 来使能。输入消抖的原理如图4-4所示。当消抖模块的输入信号翻转后保持连续3个CLK_DG时钟周期不变，则输出会对应翻转，否则输出保持为原先的值不变。

图 4-4：输入消抖



4.2.3 I/O 可选功能

I/O 引脚各个通道对应的功能如表 4-1 所示。

表 4-1: I/O 引脚功能通道定义

引脚	通道（绿色表示默认功能）							
	0	1	2	3	4	5	6	7
GPIO0	GPIO0	ANA_IN0	—	—	—	—	COMP0L	ECAP_APWM
GPIO1	GPIO1	ANA_IN1	—	—	—	—	COMP0H	ECAP_APWM
GPIO2	GPIO2	ANA_IN2	—	—	—	—	COMP1H	ECAP_APWM
GPIO3	GPIO3	ANA_IN3	—	—	—	—	COMP0L	ECAP_APWM
GPIO4	GPIO4	ANA_IN4	—	—	—	—	COMP0H	ECAP_APWM
GPIO10	GPIO10	ANA_IN10	SPI_MOSI	SPI_MISO	—	—	COMP1H	ECAP_APWM
GPIO11	GPIO11	ANA_IN11	SPI_MISO	SPI_MOSI	DCLK	EPWRTZO	COMP0L	ECAP_APWM
GPIO16	GPIO16	PWM_SYNCO	UART_TXD	UART_RXD	PWM2A	PWM5A	SPI_SFRM	ECAP_APWM
GPIO17	GPIO17	XIN	UART_RXD	UART_TXD	PWM2B	PWM5B	SPI_SCLK	ECAP_APWM
GPIO34	GPIO34	UART_TXD	UART_RXD	I2C_SDA	SPI_MOSI	SPI_MISO	I2C_SCL	ECAP_APWM
GPIO35	GPIO35	UART_RXD	UART_TXD	I2C_SCL	SPI_MISO	SPI_MOSI	I2C_SDA	ECAP_APWM
GPIO38	GPIO38	SWD	I2C_SDA	SPI_MOSI	SPI_MISO	PWM2A	SPI_SCLK	ECAP_APWM
GPIO39	GPIO39	SWCK	I2C_SCL	SPI_MISO	SPI_MOSI	PWM2B	SPI_SFRM	ECAP_APWM
GPIO40	GPIO40	SPI_SCLK	UART_TXD	DCLK	EPWRTZO	EPWRTZIO	SWV	ECAP_APWM

4.2.4 调试模式

当 TRSTn 引脚为低时，GPIO38 ~ GPIO40 用作由用户软件配置的功能。当 TRSTn 引脚为高时，GPIO38 用作 SWD，GPIO39 用作 SWCK，GPIO40 可以用作 SWV。

当 printf 底层通过 ITM（Instrumentation Trace Macrocell）实现，同时 GPIO40 指定为 SWV（Serial Wire Viewer）模式时，每次调用 printf，均可以在 GPIO40 观察到对应信号

为了使用 SWV 功能，需要将 PINMUX->GPIO40 的 MUXSEL 位段配置为 6，并将 ARM® Cortex-M4F 位于地址 0xE000EDFC 的寄存器（CoreDebug->DEMCR）的第 24 位配置为 1。

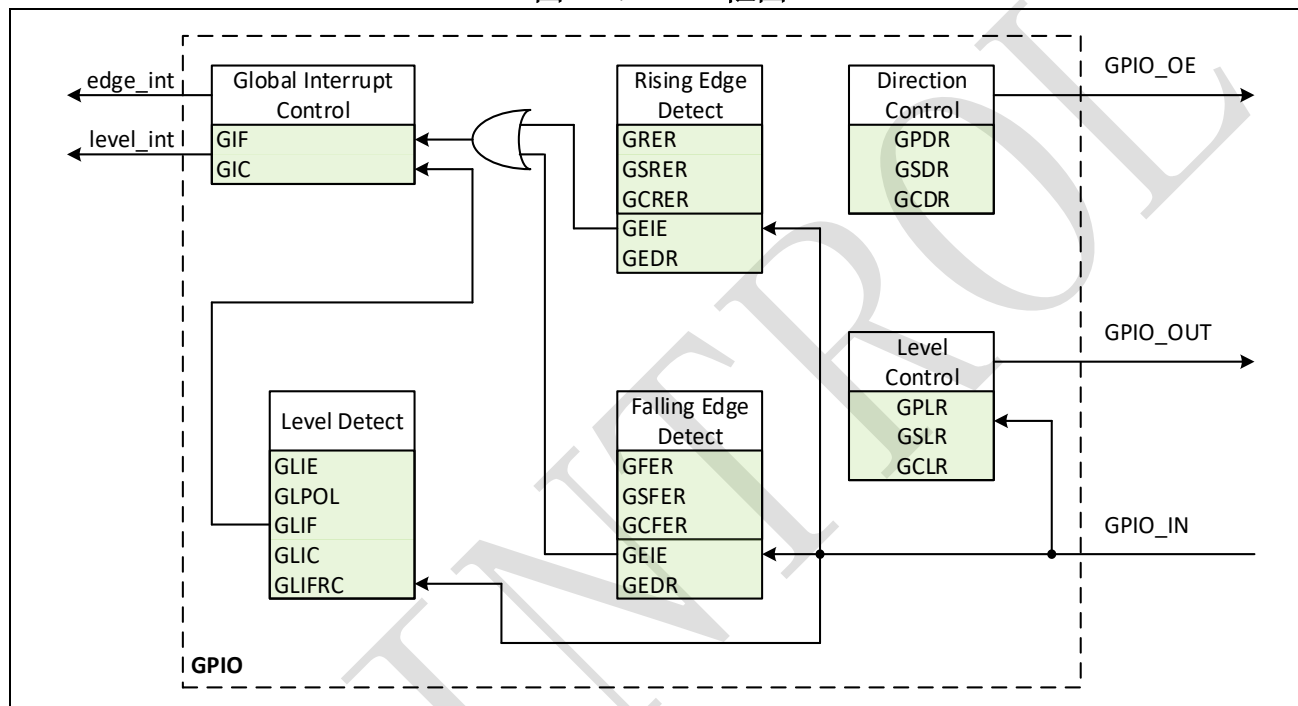
4.3 GPIO 控制模块

如图 4-5 所示，GPIO 控制模块包括电平控制、方向控制、边沿检测和电平检测。

14 个 I/O 引脚分为 2 组：GPIO0~GPIO17 和 GPIO34~GPIO40，分别对应于序号为 0 和 1 的寄存器，且寄存器的每一位对应于一个 GPIO 引脚。

为了控制的方便，提供了按位使能和按位禁用的控制寄存器。用户可以通过对“GS”前缀的寄存器写 1 来使能相应的控制，或者通过对“GC”前缀的寄存器写 1 来禁用相应的控制。对这些按位使能或禁用的寄存器写 0 无效。

图 4-5：GPIO 框图



4.3.1 方向和电平控制

所有 GPIO 引脚默认用于输入。引脚方向可以通过 GPDRx (x = 0, 1) 寄存器配置。用户也可以通过往 GSDRx 的对应位段写 1 来设置引脚为输出，或者通过往 GCDRx 的对应位段写 1 来设置引脚为输入，GPDRx 会对应更新。

当配置为输出引脚时，输出电平可以通过 GPLRx 寄存器或者 GSLRx/GCLR 寄存器配置。无论往这些寄存器里写入何值，从 GPLRx 寄存器回读的值总是反映 GPIO 端口状态，即 I/O 引脚的电平。

需要注意的是，为了消除逻辑上的亚稳态，从引脚输入的电平会先经过同步采样，因此 GPLRx 的回读值会比引脚上实际电平有两个 AHB 时钟（即 CPU 时钟）的延时。鉴于这个原因，如果需要软件在输出引脚上产生特定波形，推荐通过写 GSLRx 寄存器让相应引脚输出高电平，以及通过写 GCLR 寄存器让相应引脚输出低电平。若采用风格为对 GPLRx 寄存器做“读取-更改-写回”操作的代码无法实现预期的波形，除非在写回后再添加延时（例如：NOP 指令）以等待 GPLR 回读值更新。

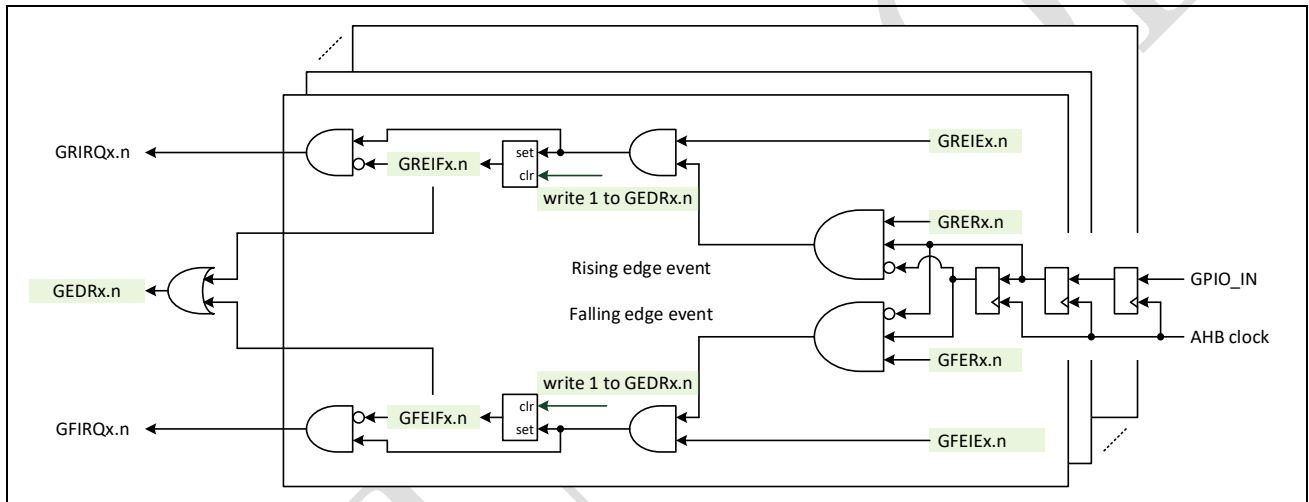
4.3.2 边沿触发的中断

如图 4-6 所示，在用 AHB 时钟对于输入重新采样后，可以进一步检测信号的上升沿和下降沿。该特性通过 GRERx 和 GFERx 寄存器使能。GSRERx/GSFERx 和 GCRERx/GCFERx 是按位控制的寄存器，可以用于对 GRERx/GFERx 置位或者清零。

GPIO 边沿检测寄存器 GEDRx 指示边沿检测状态。相应位写 1 清零该位，写 0 无效。

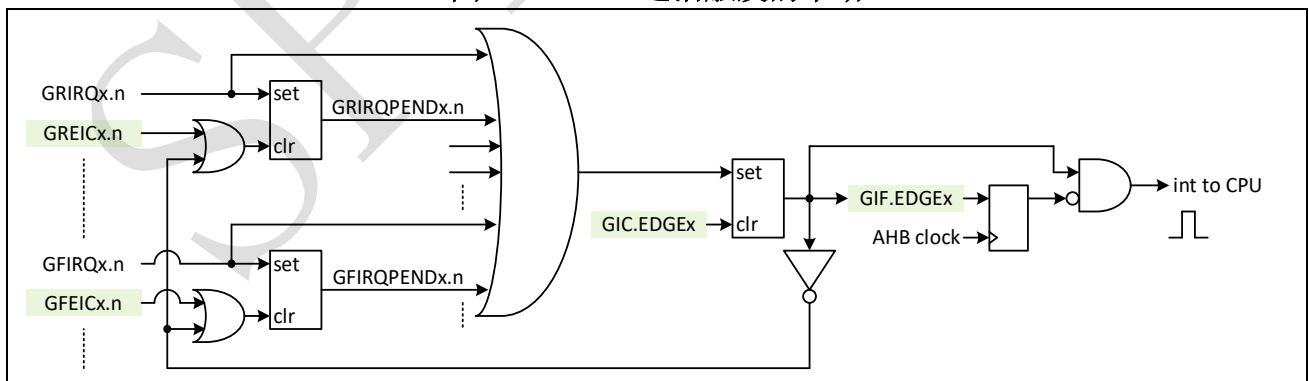
通过使能 GREIEx 或 GFEIEx 的对应位段，每当在采样后的输入信号上检测到上升沿或下降沿时，中断标志 GREIFx 或 GFEIFx 以及边沿检测状态寄存器 GEDRx 的对应位段会被置位，同时全局中断标志寄存器 GIF 的 EDGE 位段置位，触发 CPU 中断。如图 4-6 所示，向 GEDRx 的 n 位段写 1，会清除中断标志 GREIFx 和 GFEIFx 以及边沿检测状态寄存器 GEDRx 的 n 位段。如图 4-7 所示，向全局中断标志清除寄存器 GIC 的 EDGE 位段写 1 会清除全局中断标志寄存器 GIF 的 EDGE 位段。

图 4-6: GPIO 边沿检测



如图 4-6 所示，当上升沿或者下降沿事件发生时，对应的 GREIFx.n 或 GFEIFx.n 为 0，则视为有效的中断请求 GRIRQx.n 或 GFIRQx.n。

图 4-7: GPIO 边沿触发的中断



如图 4-7 所示，每个端口所有有效的中断请求通过逻辑或，最终产生边沿触发的中断信号送去 CPU，且锁存到全局中断标志寄存器 GIF 的 EDGE 位段。该位段可以通过往 GIC 寄存器的 EDGE 位段写 1 清除。SPD1121 提供 1 个边沿触发的中断。

当引脚 n 上的边沿触发一个中断请求时，所属的 GIF.EDGE 中断标志可能已经由于之前同一个端口上的引脚 m 的边沿事件置位，由此阻碍了新的中断脉冲的产生。为了避免中断的丢

失，引入了中断等待机制。在 $GIF.EDGE = 1$ 期间产生的新的中断请求会被锁存起来等待响应。当 $GIF.EDGE$ 标志位被清零时，等待中的请求会自我清除，并立刻触发新的中断并重新将 $GIF.EDGE$ 标志置位。

另一种可能是引脚 m 和 n 上的边沿事件发生的时间非常靠近。因此在执行由引脚 m 边沿事件触发的中断服务程序（ISR）过程中，后到的引脚 n 边沿事件在软件读取 $GREIFx$ 和 $GFEIFx$ 寄存器前已经被锁存。换言之，尽管 ISR 的执行是由引脚 m 的边沿事件触发，但软件在读取 $GREIFx$ 和 $GFEIFx$ 寄存器以判断中断触发源的时候会同时看到引脚 m 和 n 上的请求。在执行完对这些请求的服务代码后，用户应当通过往 $GEDRx$ 位段写 1 清除 $GREIFx$ 和 $GFEIFx$ 锁存的标志位。该操作会同时清除等待中的请求，由此避免对同一个请求触发多次中断并多次服务。

综上所述，推荐的边沿触发的中断服务程序代码结构如下，从而既不会丢失中断，也不会造成同一个中断请求多次响应和服务：

- 读取 $GREIFx$ 和 $GFEIFx$ 寄存器，检查需要服务哪些中断请求
 - 其中包含了触发本次 ISR 执行的中断请求（例如引脚 m ），和自此之后到读取寄存器之前这段时间新的请求（例如引脚 n ）。
 - 从本次 ISR 执行触发之后所有新的事件（例如引脚 n 和引脚 u ）都会被锁存为等待中的请求。
- 依据读取到的值，服务相应的请求
 - 例如引脚 m 和 n 的请求得到响应。
 - 在上述读取寄存器之后新的请求（例如引脚 u ）在本次 ISR 执行过程由于没有被看到所以无法服务。
- 清除 $GREIFx$ 和 $GFEIFx$ 寄存器对应的标志位
 - 会同时清除对应的等待中的请求（例如清除引脚 m 和 n 的标志位时，会同时清除后到的 n 事件锁存的等待）。
 - 未被看到的请求（例如引脚 u ）的标志位继续保持在 $GREIFx$ 或 $GFEIFx$ 寄存器中，且由于没有清除标志位的操作，这些请求保持等待状态。
- 清除 $GIF.EDGE$ 标志位
 - 否则将会阻碍之后端口 x 上引脚边沿事件产生中断脉冲。
- 若存在等待中的请求（例如引脚 u ），则立即产生新的中断脉冲， $GIF.EDGE$ 标志重新置位，且清除这些请求的等待状态
 - 新一轮的 ISR 执行中，引脚 u 的请求会被看到并得到服务，而之前已经服务的引脚 n 的请求已经被清除因此不会再次服务。

4.3.3 电平触发的中断

如图 4-8 所示，通过使能 $GLIEx$ 的对应位段，每当采样后的输入信号为 $GLPOLx$ 寄存器对应位段指定的电平时，中断标志寄存器 $GLIFx$ 的对应位段会被置位，同时全局中断标志寄存器中 GIF 的 $LEVEL$ 位段被置位，触发 CPU 中断。如图 4-8 所示，向中断清除寄存器 $GLICx$ 的 n 位段写 1，会清除中断标志寄存器 $GLIF$ 的 n 位段。如图 4-9 所示，向全局中断标志清除寄存器 GIC 的 $LEVEL$ 位段写 1 会清除全局中断寄存器 GIF 的 $LEVEL$ 位段。

为了方便调试，也可以通过往 $GLIFRCx$ 位段写 1 来强制产生电平事件。

图 4-8: GPIO 电平检测

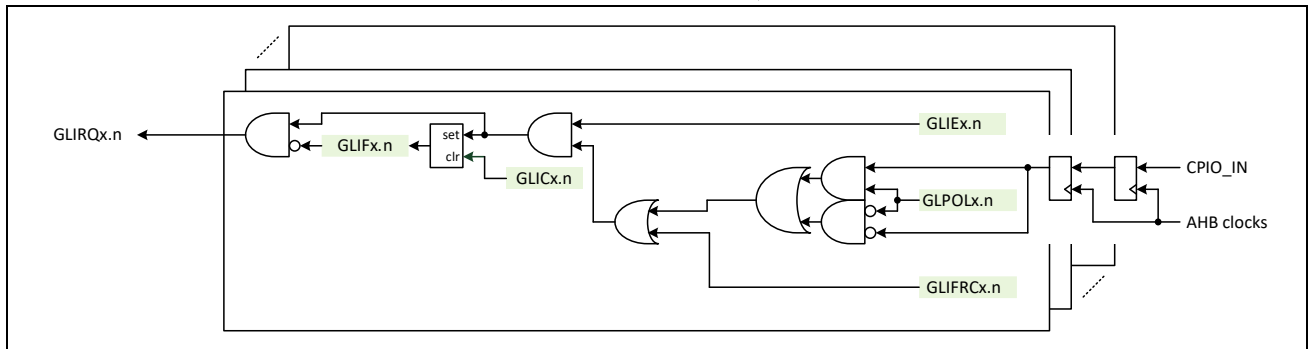
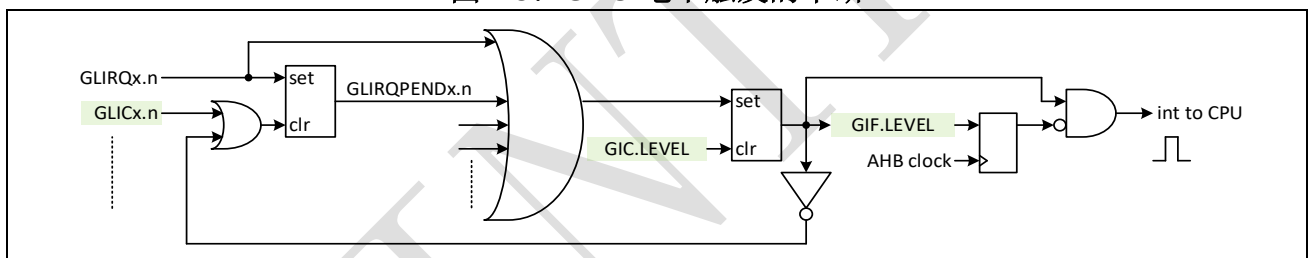


图 4-8 中名为 GLIRQx.n 的信号是有效的电平触发的中断请求。若电平事件发生时，该事件被使能用于触发中断，且 GLIFx 寄存器对应位段为 0，则视作有效的中断请求。

如图 4-9 所示，每个端口所有有效的中断请求通过逻辑或，最终产生电平触发的中断信号送去 CPU，且锁存到全局中断标志寄存器 GIF 的 LEVEL 位段。该位段可以通过往 GIC 寄存器的 LEVEL 位段写 1 清除。SPD1121 提供 1 个电平触发的中断。

与边沿触发的中断类似，此处也引入了中断等待机制。由此可以妥善处理在中断服务程序开始执行后新产生的电平中断请求，从而既不会丢失中断，也不会造成同一个中断请求多次响应和服务。

图 4-9: GPIO 电平触发的中断



推荐的电平触发的中断服务程序代码结构如下：

- 读取 GLIFx 寄存器，检查需要服务哪些中断请求
 - 其中包含了触发本次 ISR 执行的中断请求（例如引脚 m），和自此之后到读取寄存器之前这段时间新的请求（例如引脚 n）。
 - 从本次 ISR 执行触发之后所有新的事件（例如引脚 n 和引脚 u）都会被锁存为等待中的请求。
- 依据读取到的值，服务相应的请求
 - 例如引脚 m 和 n 的请求得到响应。
 - 在上述读取寄存器之后新的请求（例如引脚 u）在本次 ISR 执行过程由于没有被看到所以无法服务。
- 清除 GLIFx 寄存器对应的标志位
 - 会同时清除对应的等待中的请求（例如清除引脚 m 和 n 的标志位时，会同时清除后到的 n 事件锁存的等待）。
 - 未被看到的请求（例如引脚 u）的标志位继续保持在 GLIFx 寄存器中，且由于没有清除标志位的操作，这些请求保持等待状态。
- 清除 GIF.LEVEL 标志位
 - 否则将会阻碍之后端口 x 上引脚电平事件产生中断脉冲。

- 若存在等待中的请求（例如引脚 u），则立即产生新的中断脉冲，GIF.LEVEL 标志重新置位，且清除这些请求的等待状态
 - 新一轮的 ISR 执行中，引脚 u 的请求会被看到并得到服务，而之前已经服务的引脚 n 的请求已经被清除因此不会再次服务。

SPIN TROL

4.4 寄存器

4.4.1 PINMUX 寄存器列表

表 4-2: PINMUX 模块基地址

外设模块	基地址
PINMUX	0x4000 0300

表 4-3: PINMUX 寄存器列表

寄存器	偏移地址	描述	复位值
GPIO0*	0x0000	GPIO0 引脚控制寄存器	0x01000011
GPIO1*	0x0004	GPIO1 引脚控制寄存器	0x01000011
GPIO2*	0x0008	GPIO2 引脚控制寄存器	0x01000011
GPIO3*	0x000C	GPIO3 引脚控制寄存器	0x01000011
GPIO4*	0x0010	GPIO4 引脚控制寄存器	0x01000011
GPIO10*	0x0028	GPIO10 引脚控制寄存器	0x01000011
GPIO11*	0x002C	GPIO11 引脚控制寄存器	0x01000011
GPIO16*	0x0040	GPIO16 引脚控制寄存器	0x08000818
GPIO17*	0x0044	GPIO17 引脚控制寄存器	0x04000418
GPIO34*	0x0088	GPIO34 引脚控制寄存器	0x4D004C18
GPIO35*	0x008C	GPIO35 引脚控制寄存器	0x4B004A18
GPIO38*	0x0098	GPIO38 引脚控制寄存器	0x06000618
GPIO39*	0x009C	GPIO39 引脚控制寄存器	0x04020418
GPIO40*	0x00A0	GPIO40 引脚控制寄存器	0x01000018
PINMUXREGKEY	0x00A4	PINMUX 模块写使能寄存器	0x1ACCE551

注意： 由*标记的寄存器仅当 PINMUXREGKEY = 0x1ACCE551 才可以改写。

4.4.2 PINMUX 寄存器

表 4-4 到 4-33 列出了 GPIO PINMUX 相关寄存器的定义。

4.4.2.1 GPIO0 引脚控制寄存器（GPIO0）

表 4-4: GPIO0 引脚控制寄存器（GPIO0）位段定义

GPIO0 (GPIO0 Pin Control Register)							
Access: PINMUX->GPIO0 Offset: 0x0000 Default: 0x01000011							
31	30	29	28	27	26	25	24
RESERVED						PU1	PU0
23	22	21	20	19	18	17	16
RESERVED						PD1	PD0
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
STRENGTH		DEGLITCH	SMT	IE	MUXSEL		

表 4-5: GPIO0 引脚控制寄存器（GPIO0）位段描述

位段	位段名	属性	复位值	描述
31:26	RESERVED_31_26	RO	0x0	保留
25	PU1	RW	0x0	通道 1 的上拉使能 0: 关闭 1: 使能
24	PU0	RW	0x1	通道 0 的上拉使能 0: 关闭 1: 使能
23:18	RESERVED_23_18	RO	0x0	保留
17	PD1	RW	0x0	通道 1 的下拉使能 0: 关闭 1: 使能
16	PD0	RW	0x0	通道 0 的下拉使能 0: 关闭 1: 使能
15:8	RESERVED_15_8	RO	0x0	保留
7:6	STRENGTH	RW	0x0	输出驱动力 0: 4.5 mA 1: 9 mA 2: 13.5 mA 3: 18 mA
5	DEGLITCH	RW	0x0	输入消抖使能 0: 关闭 1: 使能

位段	位段名	属性	复位值	描述
4	SMT	RO	0x1	施密特输入使能
3	IE	RO	0x0	当 MUXSEL != 1 时，始终使能输入缓冲区
2:0	MUXSEL	RW	0x1	输出通道以及上拉和下拉控制选择 0: GPIO0 1: ANA_IN0 6: COMPOL 输出 7: ECAP APWM 模式输出

4.4.2.2 GPIO1 引脚控制寄存器（GPIO1）

表 4-6: GPIO1 引脚控制寄存器（GPIO1）位段定义

GPIO1 (GPIO1 Pin Control Register) Access: PINMUX->GPIO1 Offset: 0x0004 Default: 0x01000011							
31	30	29	28	27	26	25	24
RESERVED						PU1	PU0
23	22	21	20	19	18	17	16
RESERVED						PD1	PD0
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
STRENGTH		DEGLITCH	SMT	IE	MUXSEL		

表 4-7: GPIO1 引脚控制寄存器（GPIO1）位段描述

位段	位段名	属性	复位值	描述
31:26	RESERVED_31_26	RO	0x0	保留
25	PU1	RW	0x0	通道 1 的上拉使能 0: 关闭 1: 使能
24	PU0	RW	0x1	通道 0 的上拉使能 0: 关闭 1: 使能
23:18	RESERVED_23_18	RO	0x0	保留
17	PD1	RW	0x0	通道 1 的下拉使能 0: 关闭 1: 使能
16	PD0	RW	0x0	通道 0 的下拉使能 0: 关闭 1: 使能
15:8	RESERVED_15_8	RO	0x0	保留
7:6	STRENGTH	RW	0x0	输出驱动力 0: 4.5 mA 1: 9 mA 2: 13.5 mA 3: 18 mA
5	DEGLITCH	RW	0x0	输入消抖使能 0: 关闭 1: 使能
4	SMT	RO	0x1	施密特输入使能

位段	位段名	属性	复位值	描述
3	IE	RO	0x0	当 MUXSEL != 1 时，始终使能输入缓冲区
2:0	MUXSEL	RW	0x1	输出通道以及上拉和下拉控制选择 0: GPIO1 1: ANA_IN1 6: COMP0H 输出 7: ECAP APWM 模式输出

4.4.2.3 GPIO2 引脚控制寄存器（GPIO2）

表 4-8: GPIO2 引脚控制寄存器（GPIO2）位段定义

GPIO2 (GPIO2 Pin Control Register)							
Access: PINMUX->GPIO2 Offset: 0x0008 Default: 0x01000011							
31	30	29	28	27	26	25	24
RESERVED						PU1	PU0
23	22	21	20	19	18	17	16
RESERVED						PD1	PD0
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
STRENGTH		DEGLITCH	SMT	IE	MUXSEL		

表 4-9: GPIO2 引脚控制寄存器（GPIO2）位段描述

位段	位段名	属性	复位值	描述
31:26	RESERVED_31_26	RO	0x0	保留
25	PU1	RW	0x0	通道 1 的上拉使能 0: 关闭 1: 使能
24	PU0	RW	0x1	通道 0 的上拉使能 0: 关闭 1: 使能
23:18	RESERVED_23_18	RO	0x0	保留
17	PD1	RW	0x0	通道 1 的下拉使能 0: 关闭 1: 使能
16	PD0	RW	0x0	通道 0 的下拉使能 0: 关闭 1: 使能
15:8	RESERVED_15_8	RO	0x0	保留
7:6	STRENGTH	RW	0x0	输出驱动力 0: 4.5 mA 1: 9 mA 2: 13.5 mA 3: 18 mA
5	DEGLITCH	RW	0x0	输入消抖使能 0: 关闭 1: 使能
4	SMT	RO	0x1	施密特输入使能

位段	位段名	属性	复位值	描述
3	IE	RO	0x0	当 MUXSEL != 1 时，始终使能输入缓冲区
2:0	MUXSEL	RW	0x1	通道选择 0: GPIO2 1: ANA_IN2 6: COMP1H 输出 7: ECAP APWM 模式输出

4.4.2.4 GPIO3 引脚控制寄存器（GPIO3）

表 4-10: GPIO3 引脚控制寄存器（GPIO3）位段定义

GPIO3 (GPIO3 Pin Control Register) Access: PINMUX->GPIO3 Offset: 0x000C Default: 0x01000011							
31	30	29	28	27	26	25	24
RESERVED						PU1	PU0
23	22	21	20	19	18	17	16
RESERVED						PD1	PD0
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
STRENGTH		DEGLITCH	SMT	IE	MUXSEL		

表 4-11: GPIO3 引脚控制寄存器（GPIO3）位段描述

位段	位段名	属性	复位值	描述
31:26	RESERVED_31_26	RO	0x0	保留
25	PU1	RW	0x0	通道 1 的上拉使能 0: 关闭 1: 使能
24	PU0	RW	0x1	通道 0 的上拉使能 0: 关闭 1: 使能
23:18	RESERVED_23_18	RO	0x0	保留
17	PD1	RW	0x0	通道 1 的下拉使能 0: 关闭 1: 使能
16	PD0	RW	0x0	通道 0 的下拉使能 0: 关闭 1: 使能
15:8	RESERVED_15_8	RO	0x0	保留
7:6	STRENGTH	RW	0x0	输出驱动力 0: 4.5 mA 1: 9 mA 2: 13.5 mA 3: 18 mA
5	DEGLITCH	RW	0x0	输入消抖使能 0: 关闭 1: 使能
4	SMT	RO	0x1	施密特输入使能

位段	位段名	属性	复位值	描述
3	IE	RO	0x0	当 MUXSEL != 1 时，始终使能输入缓冲区
2:0	MUXSEL	RW	0x1	通道选择 0: GPIO3 1: ANA_IN3 6: COMP0L 输出 7: ECAP APWM 模式输出

4.4.2.5 GPIO4 引脚控制寄存器（GPIO4）

表 4-12: GPIO4 引脚控制寄存器（GPIO4）位段定义

GPIO4 (GPIO4 Pin Control Register) Access: PINMUX->GPIO4 Offset: 0x0010 Default: 0x01000011							
31	30	29	28	27	26	25	24
RESERVED						PU1	PU0
23	22	21	20	19	18	17	16
RESERVED						PD1	PD0
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
STRENGTH		DEGLITCH	SMT	IE	MUXSEL		

表 4-13: GPIO4 引脚控制寄存器（GPIO4）位段描述

位段	位段名	属性	复位值	描述
31:26	RESERVED_31_26	RO	0x0	保留
25	PU1	RW	0x0	通道 1 的上拉使能 0: 关闭 1: 使能
24	PU0	RW	0x1	通道 0 的上拉使能 0: 关闭 1: 使能
23:18	RESERVED_23_18	RO	0x0	保留
17	PD1	RW	0x0	通道 1 的下拉使能 0: 关闭 1: 使能
16	PD0	RW	0x0	通道 0 的下拉使能 0: 关闭 1: 使能
15:8	RESERVED_15_8	RO	0x0	保留
7:6	STRENGTH	RW	0x0	输出驱动力 0: 4.5 mA 1: 9 mA 2: 13.5 mA 3: 18 mA
5	DEGLITCH	RW	0x0	输入消抖使能 0: 关闭 1: 使能
4	SMT	RO	0x1	施密特输入使能

位段	位段名	属性	复位值	描述
3	IE	RO	0x0	当 MUXSEL != 1 时，始终使能输入缓冲区
2:0	MUXSEL	RW	0x1	通道选择 0: GPIO4 1: ANA_IN4 6: COMP0H 输出 7: ECAP APWM 模式输出

4.4.2.6 GPIO10 引脚控制寄存器（GPIO10）

表 4-14: GPIO10 引脚控制寄存器（GPIO10）位段定义

GPIO10 (GPIO10 Pin Control Register)							
Access: PINMUX->GPIO10 Offset: 0x0028 Default: 0x01000011							
31	30	29	28	27	26	25	24
RESERVED				PU3	PU2	PU1	PU0
23	22	21	20	19	18	17	16
RESERVED				PD3	PD2	PD1	PD0
15	14	13	12	11	10	9	8
RESERVED				CHDFLT3	CHDFLT2	RESERVED	
7	6	5	4	3	2	1	0
STRENGTH		DEGLITCH	SMT	IE	MUXSEL		

表 4-15: GPIO10 引脚控制寄存器（GPIO10）位段描述

位段	位段名	属性	复位值	描述
31:28	RESERVED_31_28	RO	0x0	保留
27	PU3	RW	0x0	通道 3 的上拉使能 0: 关闭 1: 使能
26	PU2	RW	0x0	通道 2 的上拉使能 0: 关闭 1: 使能
25	PU1	RW	0x0	通道 1 的上拉使能 0: 关闭 1: 使能
24	PU0	RW	0x1	通道 0 的上拉使能 0: 关闭 1: 使能
23:20	RESERVED_23_20	RO	0x0	保留
19	PD3	RW	0x0	通道 3 的下拉使能 0: 关闭 1: 使能
18	PD2	RW	0x0	通道 2 的下拉使能 0: 关闭 1: 使能
17	PD1	RW	0x0	通道 1 的下拉使能 0: 关闭 1: 使能
16	PD0	RW	0x0	通道 0 的下拉使能

位段	位段名	属性	复位值	描述
				0: 关闭 1: 使能
15:12	RESERVED_15_12	RO	0x0	保留
11	CHDFLT3	RW	0x0	当 MUXSEL != 3 时, 通道 3 的默认输入值
10	CHDFLT2	RW	0x0	当 MUXSEL != 2 时, 通道 2 的默认输入值
9:8	RESERVED_9_8	RO	0x0	保留
7:6	STRENGTH	RW	0x0	输出驱动力 0: 4.5 mA 1: 9 mA 2: 13.5 mA 3: 18 mA
5	DEGLITCH	RW	0x0	输入消抖使能 0: 关闭 1: 使能
4	SMT	RO	0x1	施密特输入使能
3	IE	RO	0x0	当 MUXSEL != 1 时, 始终使能输入缓冲区
2:0	MUXSEL	RW	0x1	通道选择 0: GPIO10 1: ANA_IN10 2: SPI_MOSI 3: SPI_MISO 6: COMP1H 输出 7: ECAP APWM 模式输出

4.4.2.7 GPIO11 引脚控制寄存器（GPIO11）

表 4-16: GPIO11 引脚控制寄存器（GPIO11）位段定义

GPIO11 (GPIO11 Pin Control Register)							
Access: PINMUX->GPIO11 Offset: 0x002C Default: 0x01000011							
31	30	29	28	27	26	25	24
RESERVED				PU3	PU2	PU1	PU0
23	22	21	20	19	18	17	16
RESERVED				PD3	PD2	PD1	PD0
15	14	13	12	11	10	9	8
RESERVED				CHDFLT3	CHDFLT2	RESERVED	
7	6	5	4	3	2	1	0
STRENGTH		DEGLITCH	SMT	IE	MUXSEL		

表 4-17: GPIO11 引脚控制寄存器（GPIO11）位段描述

位段	位段名	属性	复位值	描述
31:28	RESERVED_31_28	RO	0x0	保留
27	PU3	RW	0x0	通道 3 的上拉使能 0: 关闭 1: 使能
26	PU2	RW	0x0	通道 2 的上拉使能 0: 关闭 1: 使能
25	PU1	RW	0x0	通道 1 的上拉使能 0: 关闭 1: 使能
24	PU0	RW	0x1	通道 0 的上拉使能 0: 关闭 1: 使能
23:20	RESERVED_23_20	RO	0x0	保留
19	PD3	RW	0x0	通道 3 的下拉使能 0: 关闭 1: 使能
18	PD2	RW	0x0	通道 2 的下拉使能 0: 关闭 1: 使能
17	PD1	RW	0x0	通道 1 的下拉使能 0: 关闭 1: 使能
16	PD0	RW	0x0	通道 0 的下拉使能

位段	位段名	属性	复位值	描述
				0: 关闭 1: 使能
15:12	RESERVED_15_12	RO	0x0	保留
11	CHDFLT3	RW	0x0	当 MUXSEL != 3 时, 通道 3 的默认输入值
10	CHDFLT2	RW	0x0	当 MUXSEL != 2 时, 通道 2 的默认输入值
9:8	RESERVED_9_8	RO	0x0	保留
7:6	STRENGTH	RW	0x0	输出驱动力 0: 4.5 mA 1: 9 mA 2: 13.5 mA 3: 18 mA
5	DEGLITCH	RW	0x0	输入消抖使能 0: 关闭 1: 使能
4	SMT	RO	0x1	施密特输入使能
3	IE	RO	0x0	当 MUXSEL != 1 时, 始终使能输入缓冲区
2:0	MUXSEL	RW	0x1	通道选择 0: GPIO11 1: ANA_IN11 2: SPI_MISO 3: SPI_MOSI 4: CLKDET 模块 DCLK 时钟输出监测 5: EPWR TZ0 或 TZ1 输出 6: COMPOL 输出 7: ECAP APWM 模式输出

4.4.2.8 GPIO16 引脚控制寄存器（GPIO16）

表 4-18: GPIO16 引脚控制寄存器（GPIO16）位段定义

GPIO16 (GPIO16 Pin Control Register)						
Access: PINMUX->GPIO16 Offset: 0x0040 Default: 0x08000818						
31 RESERVED	30 PU6	29 PU5	28 PU4	27 PU3	26 RESERVED	25 RESERVED
23 RESERVED	22 PD6	21 PD5	20 PD4	19 RESERVED	18 RESERVED	17 RESERVED
15 RESERVED	14 CHDFLT6	13 RESERVED	12 RESERVED	11 RESERVED	10 RESERVED	9 RESERVED
7 STRENGTH	6 STRENGTH	5 DEGLITCH	4 SMT	3 IE	2 MUXSEL	1 MUXSEL
0 MUXSEL	0 MUXSEL	0 MUXSEL	0 MUXSEL	0 MUXSEL	0 MUXSEL	0 MUXSEL

表 4-19: GPIO16 引脚控制寄存器（GPIO16）位段描述

位段	位段名	属性	复位值	描述
31	RESERVED_31	RO	0x0	保留
30	PU6	RW	0x0	通道 6 的上拉使能 0: 关闭 1: 使能
29	PU5	RW	0x0	通道 5 的上拉使能 0: 关闭 1: 使能
28	PU4	RW	0x0	通道 4 的上拉使能 0: 关闭 1: 使能
27	PU3	RW	0x1	通道 3 的上拉使能 0: 关闭 1: 使能
26:25	RESERVED_26_25	RO	0x0	保留
24	PU0	RW	0x0	通道 0 的上拉使能 0: 关闭 1: 使能
23	RESERVED_23	RO	0x0	保留
22	PD6	RW	0x0	通道 6 的下拉使能 0: 关闭 1: 使能
21	PD5	RW	0x0	通道 5 的下拉使能 0: 关闭 1: 使能

位段	位段名	属性	复位值	描述
20	PD4	RW	0x0	通道 4 的下拉使能 0: 关闭 1: 使能
19:17	RESERVED_19_17	RO	0x0	保留
16	PD0	RW	0x0	通道 0 的下拉使能 0: 关闭 1: 使能
15	RESERVED_15	RO	0x0	保留
14	CHDFLT6	RW	0x0	当 MUXSEL != 6 时, 通道 6 的默认输入值
13:8	RESERVED_13_8	RO	0x8	保留
7:6	STRENGTH	RW	0x0	输出驱动力 0: 4.5 mA 1: 9 mA 2: 13.5 mA 3: 18 mA
5	DEGLITCH	RW	0x0	输入消抖使能 0: 关闭 1: 使能
4	SMT	RO	0x1	施密特输入使能
3	IE	RO	0x1	始终使能输入缓冲区
2:0	MUXSEL	RW	0x0	通道选择 0: GPIO16 1: PWM 同步信号输出 2: UART_TXD 3: UART_RXD 4: PWM2A 5: PWM5A 6: SPI_SFRM 7: ECAP APWM 模式输出

4.4.2.9 GPIO17 引脚控制寄存器（GPIO17）

表 4-20: GPIO17 引脚控制寄存器（GPIO17）位段定义

GPIO17 (GPIO17 Pin Control Register)							
Access: PINMUX->GPIO17 Offset: 0x0044 Default: 0x04000418							
31 RESERVED	30 PU6	29 PU5	28 PU4	27 RESERVED	26 PU2	25 PU1	24 PU0
23 RESERVED	22 PD6	21 PD5	20 PD4	19 RESERVED	18 RESERVED	17 PD1	16 PD0
15 RESERVED	14 CHDFLT6	13 RESERVED	12 RESERVED	11 RESERVED	10 RESERVED	9 RESERVED	8 RESERVED
7 RESERVED	6 STRENGTH	5 DEGLITCH	4 SMT	3 IE	2 RESERVED	1 MUXSEL	0 RESERVED

表 4-21: GPIO17 引脚控制寄存器（GPIO17）位段描述

位段	位段名	属性	复位值	描述
31	RESERVED_31	RO	0x0	保留
30	PU6	RW	0x0	通道 6 的上拉使能 0: 关闭 1: 使能
29	PU5	RW	0x0	通道 5 的上拉使能 0: 关闭 1: 使能
28	PU4	RW	0x0	通道 4 的上拉使能 0: 关闭 1: 使能
27	RESERVED_27	RO	0x0	保留
26	PU2	RW	0x1	通道 2 的上拉使能 0: 关闭 1: 使能
25	PU1	RW	0x0	通道 1 的上拉使能 0: 关闭 1: 使能
24	PU0	RW	0x0	通道 0 的上拉使能 0: 关闭 1: 使能
23	RESERVED_23	RO	0x0	保留
22	PD6	RW	0x0	通道 6 的下拉使能 0: 关闭 1: 使能

位段	位段名	属性	复位值	描述
21	PD5	RW	0x0	通道 5 的下拉使能 0: 关闭 1: 使能
20	PD4	RW	0x0	通道 4 的下拉使能 0: 关闭 1: 使能
19:18	RESERVED_19_18	RO	0x0	保留
17	PD1	RW	0x0	通道 1 的下拉使能 0: 关闭 1: 使能
16	PD0	RW	0x0	通道 0 的下拉使能 0: 关闭 1: 使能
15	RESERVED_15	RO	0x0	保留
14	CHDFLT6	RW	0x0	当 MUXSEL != 6 时, 通道 6 的默认输入值
13:8	RESERVED_13_8	RO	0x4	保留
7:6	STRENGTH	RW	0x0	输出驱动力 0: 4.5 mA 1: 9 mA 2: 13.5 mA 3: 18 mA
5	DEGLITCH	RW	0x0	输入消抖使能 0: 关闭 1: 使能
4	SMT	RO	0x1	施密特输入使能
3	IE	RO	0x1	始终使能输入缓冲区
2:0	MUXSEL	RW	0x0	通道选择 0: GPIO17 1: XIN 2: UART_RXD 3: UART_TXD 4: PWM2B 5: PWM5B 6: SPI_SCLK 7: ECAP APWM 模式输出

4.4.2.10 GPIO34 引脚控制寄存器（GPIO34）

表 4-22: GPIO34 引脚控制寄存器（GPIO34）位段定义

GPIO34 (GPIO34 Pin Control Register)							
Access: PINMUX->GPIO34 Offset: 0x0088 Default: 0x4D004C18							
31	30	29	28	27	26	25	24
RESERVED		PU5	PU4	RESERVED	PU2	RESERVED	PU0
23	22	21	20	19	18	17	16
RESERVED		PD5	PD4	RESERVED			PD0
15	14	13	12	11	10	9	8
RESERVED		CHDFLT5	CHDFLT4	RESERVED			
7	6	5	4	3	2	1	0
STRENGTH		DEGLITCH	SMT	IE	MUXSEL		

表 4-23: GPIO34 引脚控制寄存器（GPIO34）位段描述

位段	位段名	属性	复位值	描述
31:30	RESERVED_31_30	RO	0x1	保留
29	PU5	RW	0x0	通道 5 的上拉使能 0: 关闭 1: 使能
28	PU4	RW	0x0	通道 4 的上拉使能 0: 关闭 1: 使能
27	RESERVED_27	RO	0x1	保留
26	PU2	RW	0x1	通道 2 的上拉使能 0: 关闭 1: 使能
25	RESERVED_25	RO	0x0	保留
24	PU0	RW	0x1	通道 0 的上拉使能 0: 关闭 1: 使能
23:22	RESERVED_23_22	RO	0x0	保留
21	PD5	RW	0x0	通道 5 的下拉使能 0: 关闭 1: 使能
20	PD4	RW	0x0	通道 4 的下拉使能 0: 关闭 1: 使能
19:17	RESERVED_19_17	RO	0x0	保留
16	PD0	RW	0x0	通道 0 的下拉使能

位段	位段名	属性	复位值	描述
				0: 关闭 1: 使能
15:14	RESERVED_15_14	RO	0x1	保留
13	CHDFLT5	RW	0x0	当 MUXSEL != 5 时, 通道 5 的默认输入值
12	CHDFLT4	RW	0x0	当 MUXSEL != 4 时, 通道 4 的默认输入值
11:8	RESERVED_11_8	RO	0xC	保留
7:6	STRENGTH	RW	0x0	输出驱动力 0: 4.5 mA 1: 9 mA 2: 13.5 mA 3: 18 mA
5	DEGLITCH	RW	0x0	输入消抖使能 0: 关闭 1: 使能
4	SMT	RO	0x1	施密特输入使能
3	IE	RO	0x1	始终使能输入缓冲区
2:0	MUXSEL	RW	0x0	通道选择 0: GPIO34 1: UART_TXD 2: UART_RXD 3: I2C_SDA 4: SPI_MOSI 5: SPI_MISO 6: I2C_SCL 7: ECAP APWM 模式输出

4.4.2.11 GPIO35 引脚控制寄存器（GPIO35）

表 4-24: GPIO35 引脚控制寄存器（GPIO35）位段定义

GPIO35 (GPIO35 Pin Control Register)					
Access: PINMUX->GPIO35		Offset: 0x008C		Default: 0x4B004A18	
31	30	29	28	27	26
RESERVED		PU5	PU4	RESERVED	
25	24	23	22	21	20
PU1	PU0	PD5	PD4	RESERVED	
17	16	15	14	13	12
RESERVED		CHDFLT5	CHDFLT4	RESERVED	
11	10	9	8	7	6
IE	MUXSEL		DEGLITCH	SMT	STRENGTH

表 4-25: GPIO35 引脚控制寄存器（GPIO35）位段描述

位段	位段名	属性	复位值	描述
31:30	RESERVED_31_30	RO	0x1	保留
29	PU5	RW	0x0	通道 5 的上拉使能 0: 关闭 1: 使能
28	PU4	RW	0x0	通道 4 的上拉使能 0: 关闭 1: 使能
27:26	RESERVED_27_26	RO	0x2	保留
25	PU1	RW	0x1	通道 1 的上拉使能 0: 关闭 1: 使能
24	PU0	RW	0x1	通道 0 的上拉使能 0: 关闭 1: 使能
23:22	RESERVED_23_22	RO	0x0	保留
21	PD5	RW	0x0	通道 5 的下拉使能 0: 关闭 1: 使能
20	PD4	RW	0x0	通道 4 的下拉使能 0: 关闭 1: 使能
19:17	RESERVED_19_17	RO	0x0	保留
16	PD0	RW	0x0	通道 0 的下拉使能

位段	位段名	属性	复位值	描述
				0: 关闭 1: 使能
15:14	RESERVED_15_14	RO	0x1	保留
13	CHDFLT5	RW	0x0	当 MUXSEL != 5 时, 通道 5 的默认输入值
12	CHDFLT4	RW	0x0	当 MUXSEL != 4 时, 通道 4 的默认输入值
11:8	RESERVED_11_8	RO	0xA	保留
7:6	STRENGTH	RW	0x0	输出驱动力 0: 4.5 mA 1: 9 mA 2: 13.5 mA 3: 18 mA
5	DEGLITCH	RW	0x0	输入消抖使能 0: 关闭 1: 使能
4	SMT	RO	0x1	施密特输入使能
3	IE	RO	0x1	始终使能输入缓冲区
2:0	MUXSEL	RW	0x0	通道选择 0: GPIO35 1: UART_RXD 2: UART_TXD 3: I2C_SCL 4: SPI_MISO 5: SPI_MOSI 6: I2C_SDA 7: ECAP APWM 模式输出

4.4.2.12 GPIO38 引脚控制寄存器（GPIO38）

表 4-26: GPIO38 引脚控制寄存器（GPIO38）位段定义

GPIO38 (GPIO38 Pin Control Register)						
Access: PINMUX->GPIO38 Offset: 0x0098 Default: 0x06000618						
31 RESERVED	30 PU6	29 PU5	28 PU4	27 PU3	26 RESERVED	25 RESERVED
23 RESERVED	22 PD6	21 PD5	20 PD4	19 PD3	18 RESERVED	17 RESERVED
15 RESERVED	14 CHDFLT6	13 RESERVED	12 CHDFLT4	11 CHDFLT3	10 RESERVED	9 RESERVED
7 STRENGTH	6 STRENGTH	5 DEGLITCH	4 SMT	3 IE	2 MUXSEL	1 MUXSEL
						0

表 4-27: GPIO38 引脚控制寄存器（GPIO38）位段描述

位段	位段名	属性	复位值	描述
31	RESERVED_31	RO	0x0	保留
30	PU6	RW	0x0	通道 6 的上拉使能 0: 关闭 1: 使能
29	PU5	RW	0x0	通道 5 的上拉使能 0: 关闭 1: 使能
28	PU4	RW	0x0	通道 4 的上拉使能 0: 关闭 1: 使能
27	PU3	RW	0x0	通道 3 的上拉使能 0: 关闭 1: 使能
26:25	RESERVED_26_25	RO	0x3	保留
24	PU0	RW	0x0	通道 0 的上拉使能 0: 关闭 1: 使能
23	RESERVED_23	RO	0x0	保留
22	PD6	RW	0x0	通道 6 的下拉使能 0: 关闭 1: 使能
21	PD5	RW	0x0	通道 5 的下拉使能 0: 关闭 1: 使能

位段	位段名	属性	复位值	描述
20	PD4	RW	0x0	通道 4 的下拉使能 0: 关闭 1: 使能
19	PD3	RW	0x0	通道 3 的下拉使能 0: 关闭 1: 使能
18:17	RESERVED_18_17	RO	0x0	保留
16	PD0	RW	0x0	通道 0 的下拉使能 0: 关闭 1: 使能
15	RESERVED_15	RO	0x0	保留
14	CHDFLT6	RW	0x0	当 MUXSEL != 6 时, 通道 6 的默认输入值
13	RESERVED_13	RO	0x0	保留
12	CHDFLT4	RW	0x0	当 MUXSEL != 4 时, 通道 4 的默认输入值
11	CHDFLT3	RW	0x0	当 MUXSEL != 3 时, 通道 3 的默认输入值
10:8	RESERVED_10_8	RO	0x6	保留
7:6	STRENGTH	RW	0x0	输出驱动力 0: 4.5 mA 1: 9 mA 2: 13.5 mA 3: 18 mA
5	DEGLITCH	RW	0x0	输入消抖使能 0: 关闭 1: 使能
4	SMT	RO	0x1	施密特输入使能
3	IE	RO	0x1	始终使能输入缓冲区
2:0	MUXSEL	RW	0x0	通道选择 0: GPIO38 1: SWD 2: I2C_SDA 3: SPI_MOSI 4: SPI_MISO 5: PWM2A 6: SPI_SCLK 7: ECAP APWM 模式输出

4.4.2.13 GPIO39 引脚控制寄存器（GPIO39）

表 4-28: GPIO39 引脚控制寄存器（GPIO39）位段定义

GPIO39 (GPIO39 Pin Control Register)						
Access: PINMUX->GPIO39 Offset: 0x009C Default: 0x04020418						
31 RESERVED	30 PU6	29 PU5	28 PU4	27 PU3	26 RESERVED	25 RESERVED
23 RESERVED	22 PD6	21 PD5	20 PD4	19 PD3	18 RESERVED	17 RESERVED
15 RESERVED	14 CHDFLT6	13 RESERVED	12 CHDFLT4	11 CHDFLT3	10 RESERVED	9 RESERVED
7 STRENGTH	6 STRENGTH	5 DEGLITCH	4 SMT	3 IE	2 MUXSEL	1 MUXSEL
						0

表 4-29: GPIO39 引脚控制寄存器（GPIO39）位段描述

位段	位段名	属性	复位值	描述
31	RESERVED_31	RO	0x0	保留
30	PU6	RW	0x0	通道 6 的上拉使能 0: 关闭 1: 使能
29	PU5	RW	0x0	通道 5 的上拉使能 0: 关闭 1: 使能
28	PU4	RW	0x0	通道 4 的上拉使能 0: 关闭 1: 使能
27	PU3	RW	0x0	通道 3 的上拉使能 0: 关闭 1: 使能
26:25	RESERVED_26_25	RO	0x2	保留
24	PU0	RW	0x0	通道 0 的上拉使能 0: 关闭 1: 使能
23	RESERVED_23	RO	0x0	保留
22	PD6	RW	0x0	通道 6 的下拉使能 0: 关闭 1: 使能
21	PD5	RW	0x0	通道 5 的下拉使能 0: 关闭 1: 使能

位段	位段名	属性	复位值	描述
20	PD4	RW	0x0	通道 4 的下拉使能 0: 关闭 1: 使能
19	PD3	RW	0x0	通道 3 的下拉使能 0: 关闭 1: 使能
18:17	RESERVED_18_17	RO	0x1	保留
16	PD0	RW	0x0	通道 0 的下拉使能 0: 关闭 1: 使能
15	RESERVED_15	RO	0x0	保留
14	CHDFLT6	RW	0x0	当 MUXSEL != 6 时, 通道 6 的默认输入值
13	RESERVED_13	RO	0x0	保留
12	CHDFLT4	RW	0x0	当 MUXSEL != 4 时, 通道 4 的默认输入值
11	CHDFLT3	RW	0x0	当 MUXSEL != 3 时, 通道 3 的默认输入值
10:8	RESERVED_10_8	RO	0x4	保留
7:6	STRENGTH	RW	0x0	输出驱动力 0: 4.5 mA 1: 9 mA 2: 13.5 mA 3: 18 mA
5	DEGLITCH	RW	0x0	输入消抖使能 0: 关闭 1: 使能
4	SMT	RO	0x1	施密特输入使能
3	IE	RO	0x1	始终使能输入缓冲区
2:0	MUXSEL	RW	0x0	通道选择 0: GPIO39 1: SWCK 2: I2C_SCL 3: SPI_MISO 4: SPI_MOSI 5: PWM2B 6: SPI_SFRM 7: ECAP APWM 模式输出

4.4.2.14 GPIO40 引脚控制寄存器（GPIO40）

表 4-30: GPIO40 引脚控制寄存器（GPIO40）位段定义

GPIO40 (GPIO40 Pin Control Register) Access: PINMUX->GPIO40 Offset: 0x00A0 Default: 0x01000018							
31	30	29	28	27	26	25	24
RESERVED						PU1	PU0
23	22	21	20	19	18	17	16
RESERVED						PD1	PD0
15	14	13	12	11	10	9	8
RESERVED						CHDFLT1	RESERVED
7	6	5	4	3	2	1	0
STRENGTH		DEGLITCH	SMT	IE	MUXSEL		

表 4-31: GPIO40 引脚控制寄存器（GPIO40）位段描述

位段	位段名	属性	复位值	描述
31:26	RESERVED_31_26	RO	0x0	保留
25	PU1	RW	0x0	通道 1 的上拉使能 0: 关闭 1: 使能
24	PU0	RW	0x1	通道 0 的上拉使能 0: 关闭 1: 使能
23:18	RESERVED_23_18	RO	0x0	保留
17	PD1	RW	0x0	通道 1 的下拉使能 0: 关闭 1: 使能
16	PD0	RW	0x0	通道 0 的下拉使能 0: 关闭 1: 使能
15:10	RESERVED_15_10	RO	0x0	保留
9	CHDFLT1	RW	0x0	当 MUXSEL != 1 时，通道 1 的默认输入值
8	RESERVED_8	RO	0x0	保留
7:6	STRENGTH	RW	0x0	输出驱动力 0: 4.5 mA 1: 9 mA 2: 13.5 mA 3: 18 mA
5	DEGLITCH	RW	0x0	输入消抖使能 0: 关闭 1: 使能

位段	位段名	属性	复位值	描述
4	SMT	RO	0x1	施密特输入使能
3	IE	RO	0x1	始终使能输入缓冲区
2:0	MUXSEL	RW	0x0	通道选择 0: GPIO40 1: SPI_SCLK 2: UART_TXD 3: CLKDET 模块 DCLK 时钟输出监测 4: EPWR TZ0 输出监视 5: EPWR TZ1 输出监视 6: SWV 7: ECAP APWM 模式输出

4.4.2.15 PINMUX 模块写使能寄存器（PINMUXREGKEY）

表 4-32: PINMUX 模块写使能寄存器（PINMUXREGKEY）位段定义

PINMUXREGKEY (PINMUX Register Write-Allow Key Register) Access: PINMUX->PINMUXREGKEY Offset: 0x00A4 Default: 0x1ACCE551							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

表 4-33: PINMUX 模块写使能寄存器（PINMUXREGKEY）位段描述

位段	位段名	属性	复位值	描述
31:0	KEY	RW	0x1ACCE551	写入 0x1ACCE551 以解锁受保护的 PINMUX 寄存器

4.4.3 GPIO 寄存器列表

表 4-34: GPIO 模块基地址

外设模块	基地址
GPIO	0x4000 3000

表 4-35: GPIO 寄存器列表

寄存器	偏移地址	描述	复位值
GPLR0	0x0000	GPIO 引脚电平寄存器 0	0x00000000
GPLR1	0x0004	GPIO 引脚电平寄存器 1	0x000001CC
GPDR0*	0x000C	GPIO 引脚方向寄存器 0	0x00000000
GPDR1*	0x0010	GPIO 引脚方向寄存器 1	0x00000000
GSLR0	0x0018	GPIO 引脚输出置 1 寄存器 0	0x00000000
GSLR1	0x001C	GPIO 引脚输出置 1 寄存器 1	0x00000000
GCLR0	0x0024	GPIO 引脚输出清除寄存器 0	0x00000000
GCLR1	0x0028	GPIO 引脚输出清除寄存器 1	0x00000000
GRER0*	0x0030	GPIO 上升沿检测使能寄存器 0	0x00000000
GRER1*	0x0034	GPIO 上升沿检测使能寄存器 1	0x00000000
GFER0*	0x003C	GPIO 下降沿检测使能寄存器 0	0x00000000
GFER1*	0x0040	GPIO 下降沿检测使能寄存器 1	0x00000000
GEDR0	0x0048	GPIO 边沿检测状态寄存器 0	0x00000000
GEDR1	0x004C	GPIO 边沿检测状态寄存器 1	0x00000000
GSDR0*	0x0054	GPIO 引脚按位设置输出寄存器 0	0x00000000
GSDR1*	0x0058	GPIO 引脚按位设置输出寄存器 1	0x00000000
GCDR0*	0x0060	GPIO 引脚按位设置输入寄存器 0	0x00000000
GCDR1*	0x0064	GPIO 引脚按位设置输入寄存器 1	0x00000000
GSRER0*	0x006C	GPIO 上升沿检测按位使能寄存器 0	0x00000000
GSRER1*	0x0070	GPIO 上升沿检测按位使能寄存器 1	0x00000000
GCRER0*	0x0078	GPIO 上升沿检测按位关闭寄存器 0	0x00000000
GCRER1*	0x007C	GPIO 上升沿检测按位关闭寄存器 1	0x00000000
GSFER0*	0x0084	GPIO 下降沿检测按位使能寄存器 0	0x00000000
GSFER1*	0x0088	GPIO 下降沿检测按位使能寄存器 1	0x00000000
GCFER0*	0x0090	GPIO 下降沿检测按位关闭寄存器 0	0x00000000
GCFER1*	0x0094	GPIO 下降沿检测按位关闭寄存器 1	0x00000000
GEIER0*	0x009C	GPIO 边沿中断使能寄存器 0	0x00000000

寄存器	偏移地址	描述	复位值
<u>GEIE1*</u>	0x00A0	GPIO 边沿中断使能寄存器 1	0x00000000
<u>GLIF0*</u>	0x00A8	GPIO 电平中断标志寄存器 0	0x00000000
<u>GLIF1*</u>	0x00AC	GPIO 电平中断标志寄存器 1	0x00000000
<u>GLIE0*</u>	0x00B4	GPIO 电平中断使能寄存器 0	0x00000000
<u>GLIE1*</u>	0x00B8	GPIO 电平中断使能寄存器 1	0x00000000
<u>GLIC0</u>	0x00C0	GPIO 电平中断清除寄存器 0	0x00000000
<u>GLIC1</u>	0x00C4	GPIO 电平中断清除寄存器 1	0x00000000
<u>GLIFRC0*</u>	0x00CC	GPIO 电平中断强制寄存器 0	0x00000000
<u>GLIFRC1*</u>	0x00D0	GPIO 电平中断强制寄存器 1	0x00000000
<u>GLIPOLO*</u>	0x00D8	GPIO 电平中断极性寄存器 0	0x00000000
<u>GLIPOL1*</u>	0x00DC	GPIO 电平中断极性寄存器 1	0x00000000
<u>GIF</u>	0x00E4	GPIO 全局中断标志寄存器	0x00000000
<u>GIC</u>	0x00E8	GPIO 全局中断清除寄存器	0x00000000
<u>GPIOREGKEY</u>	0x00EC	GPIO 模块写使能寄存器	0x1ACCE551
<u>GREIF0</u>	0x00F0	GPIO 上升沿中断标志寄存器 0	0x00000000
<u>GREIF1</u>	0x00F4	GPIO 上升沿中断标志寄存器 1	0x00000000
<u>GFEIF0</u>	0x00F8	GPIO 下降沿中断标志寄存器 0	0x00000000
<u>GFEIF1</u>	0x00FC	GPIO 下降沿中断标志寄存器 1	0x00000000

注意： 由*标记的寄存器仅当 GPIOREGKEY=0x1ACCE551 才可以改写。

4.4.4 GPIO 寄存器

表 4-36 到表 4-125 列出了 GPIO 功能 IP 相关寄存器的定义。

4.4.4.1 GPIO 引脚电平寄存器 0 (GPLR0)

表 4-36: GPIO 引脚电平寄存器 0 (GPLR0) 位段定义

GPLR0 (GPIO Pin Level Register 0) Offset: 0x0 Default: 0x00030000							
Access: GPIO -> GPLR0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-37: GPIO 引脚电平寄存器 0 (GPLR0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	GPIO _n 引脚电平 (n = 0~31)。 0: 写 0 设置对应引脚输出低, 读 0 表示当前电平为低 1: 写 1 设置对应引脚输出高, 读 1 表示当前电平为高

4.4.4.2 GPIO 引脚电平寄存器 1 (GPLR1)

表 4-38: GPIO 引脚电平寄存器 1 (GPLR1) 位段定义

GPLR1 (GPIO Pin Level Register 1) Offset: 0x4 Default: 0x000001CC							
Access: GPIO -> GPLR1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-39: GPIO 引脚电平寄存器 1 (GPLR1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	RW	0x1CC	GPIO _n 引脚电平 (n = 32~40)。 0: 写 0 设置对应引脚输出低, 读 0 表示当前电平为低 1: 写 1 设置对应引脚输出高, 读 1 表示当前电平为高

4.4.4.3 GPIO 引脚方向寄存器 0 (GPDR0)

表 4-40: GPIO 引脚方向寄存器 0 (GPDR0) 位段定义

GPDR0 (GPIO Pin Direction Register 0) Offset: 0xC Default: 0x00000000							
Access: GPIO -> GPDR0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-41: GPIO 引脚方向寄存器 0 (GPDR0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	设置 GPIO _n 引脚方向 (n = 0~31) 0: 设置对应引脚为输入方向 1: 设置对应引脚为输出方向

4.4.4.4 GPIO 引脚方向寄存器 1 (GPDR1)

表 4-42: GPIO 引脚方向寄存器 1 (GPDR1) 位段定义

GPDR1 (GPIO Pin Direction Register 1) Offset: 0x10 Default: 0x00000000							
Access: GPIO -> GPDR1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-43: GPIO 引脚方向寄存器 1 (GPDR1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	RW	0x0	设置 GPIO _n 引脚方向 (n = 32~40) 0: 设置对应引脚为输入方向 1: 设置对应引脚为输出方向

4.4.4.5 GPIO 引脚输出置 1 寄存器 0 (GSLR0)

表 4-44: GPIO 引脚输出置 1 寄存器 0 (GSLR0) 位段定义

GSLR0 (GPIO Pin Output Set Register 0) Offset: 0x18 Default: 0x00000000							
Access: GPIO -> GSLR0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-45: GPIO 引脚输出置 1 寄存器 0 (GSLR0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	W1S	0x0	设置 GPIO _n 引脚输出高 (n = 0~31) 0: 写 0 无效, 总是读回 0 1: 写 1 设置对应引脚输出高, 位段自动清零

4.4.4.6 GPIO 引脚输出置 1 寄存器 1 (GSLR1)

表 4-46: GPIO 引脚输出置 1 寄存器 1 (GSLR1) 位段定义

GSLR1 (GPIO Pin Output Set Register 1) Offset: 0x1C Default: 0x00000000							
Access: GPIO -> GSLR1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-47: GPIO 引脚输出置 1 寄存器 1 (GSLR1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	W1S	0x0	设置 GPIO _n 引脚输出高 (n = 32~40) 0: 写 0 无效, 总是读回 0 1: 写 1 设置对应引脚输出高, 位段自动清零

4.4.4.7 GPIO 引脚输出清除寄存器 0 (GCLR0)

表 4-48: GPIO 引脚输出清除寄存器 0 (GCLR0) 位段定义

GCLR0 (GPIO Pin Output Clear Register 0) Offset: 0x24 Default: 0x00000000							
Access: GPIO -> GCLR0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-49: GPIO 引脚输出清除寄存器 0 (GCLR0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	W1S	0x0	设置 GPIO _n 引脚输出低 (n = 0~31) 0: 写 0 无效, 总是读回 0 1: 写 1 设置对应引脚输出低, 位段自动清零

4.4.4.8 GPIO 引脚输出清除寄存器 1 (GCLR1)

表 4-50: GPIO 引脚输出清除寄存器 1 (GCLR1) 位段定义

GCLR1 (GPIO Pin Output Clear Register 1) Offset: 0x28 Default: 0x00000000							
Access: GPIO -> GCLR1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-51: GPIO 引脚输出清除寄存器 1 (GCLR1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	W1S	0x0	设置 GPIO _n 引脚输出低 (n = 32~40) 0: 写 0 无效, 总是读回 0 1: 写 1 设置对应引脚输出低, 位段自动清零

4.4.4.9 GPIO 上升沿检测使能寄存器 0 (GRER0)

表 4-52: GPIO 上升沿检测使能寄存器 0 (GRER0) 位段定义

GRER0 (GPIO Rising Edge Detect Enable Register 0) Offset: 0x30 Default: 0x00000000							
Access: GPIO -> GRER0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-53: GPIO 上升沿检测使能寄存器 0 (GRER0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	使能 GPIO _n 引脚上升沿检测 (n = 0~31) 0: 关闭 1: 使能

4.4.4.10 GPIO 上升沿检测使能寄存器 1 (GRER1)

表 4-54: GPIO 上升沿检测使能寄存器 1 (GRER1) 位段定义

GRER1 (GPIO Rising Edge Detect Enable Register 1) Offset: 0x34 Default: 0x00000000							
Access: GPIO -> GRER1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-55: GPIO 上升沿检测使能寄存器 1 (GRER1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	RW	0x0	使能 GPIO _n 引脚上升沿检测 (n = 32~40) 0: 关闭 1: 使能

4.4.4.11 GPIO 下降沿检测使能寄存器 0 (GFER0)

表 4-56: GPIO 下降沿检测使能寄存器 0 (GFER0) 位段定义

GFER0 (GPIO Falling Edge Detect Enable Register 0) Offset: 0x3C Default: 0x00000000							
Access: GPIO -> GFER0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-57: GPIO 下降沿检测使能寄存器 0 (GFER0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	使能 GPIO _n 引脚下降沿检测 (n = 0~31) 0: 关闭 1: 使能

4.4.4.12 GPIO 下降沿检测使能寄存器 1 (GFER1)

表 4-58: GPIO 下降沿检测使能寄存器 1 (GFER1) 位段定义

GFER1 (GPIO Falling Edge Detect Enable Register 1) Offset: 0x40 Default: 0x00000000							
Access: GPIO -> GFER1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-59: GPIO 下降沿检测使能寄存器 1 (GFER1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	RW	0x0	使能 GPIO _n 引脚下降沿检测 (n = 32~40) 0: 关闭 1: 使能

4.4.4.13 GPIO 边沿检测状态寄存器 0 (GEDR0)

表 4-60: GPIO 边沿检测状态寄存器 0 (GEDR0) 位段定义

GEDR0 (GPIO Edge Detect Status Register 0) Offset: 0x48 Default: 0x00000000							
Access: GPIO -> GEDR0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-61: GPIO 边沿检测状态寄存器 0 (GEDR0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	W1C	0x0	GPIO_n 引脚边沿检测状态（上升沿或下降沿，n = 0~31） 0: 读 0，未检测到边沿；写 0 无效 1: 读 1，检测到边沿；写 1 清除检测状态。

4.4.4.14 GPIO 边沿检测状态寄存器 1 (GEDR1)

表 4-62: GPIO 边沿检测状态寄存器 1 (GEDR1) 位段定义

GEDR1 (GPIO Edge Detect Status Register 1) Offset: 0x4C Default: 0x00000000							
Access: GPIO -> GEDR1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-63: GPIO 边沿检测状态寄存器 1 (GEDR1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	W1C	0x0	GPIO_n 引脚边沿检测状态（上升沿或下降沿，n = 32~40） 0: 读 0，未检测到边沿；写 0 无效 1: 读 1，检测到边沿；写 1 清除检测状态。

4.4.4.15 GPIO 引脚按位设置输出寄存器 0 (GSDR0)

表 4-64: GPIO 引脚按位设置输出寄存器 0 (GSDR0) 位段定义

GSDR0 (GPIO Pin Bitwise Set Direction Register 0) Offset: 0x54 Default: 0x00000000							
Access: GPIO -> GSDR0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-65: GPIO 引脚按位设置输出寄存器 0 (GSDR0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	WO	0x0	设置 GPIO _n 引脚为输出 (n = 0~31) 0: 写 0 无效, 总是读回 0 1: 写 1 将 GPDR0 的相应位段置 1 并设置 GPIO _n 为输出, 位段自动清零。

4.4.4.16 GPIO 引脚按位设置输出寄存器 1 (GSDR1)

表 4-66: GPIO 引脚按位设置输出寄存器 1 (GSDR1) 位段定义

GSDR1 (GPIO Pin Bitwise Set Direction Register 1) Offset: 0x58 Default: 0x00000000							
Access: GPIO -> GSDR1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-67: GPIO 引脚按位设置输出寄存器 1 (GSDR1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	WO	0x0	设置 GPIO _n 引脚为输出 (n = 32~40) 0: 写 0 无效, 总是读回 0 1: 写 1 将 GPDR1 的相应位段置 1 并设置 GPIO _n 为输出, 位段自动清零。

4.4.4.17 GPIO 引脚按位设置输入寄存器 0 (GCDR0)

表 4-68: GPIO 引脚按位设置输入寄存器 0 (GCDR0) 位段定义

GCDR0 (GPIO Pin Bitwise Clear Direction Register 0) Offset: 0x60 Default: 0x00000000							
Access: GPIO -> GCDR0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-69: GPIO 引脚按位设置输入寄存器 0 (GCDR0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	WO	0x0	设置 GPIO _n 引脚为输入 (n = 0~31) 0: 写 0 无效, 总是读回 0 1: 写 1 清除 GPDR0 的相应位段并设置 GPIO _n 为输入, 位段自动清零。

4.4.4.18 GPIO 引脚按位设置输入寄存器 1 (GCDR1)

表 4-70: GPIO 引脚按位设置输入寄存器 1 (GCDR1) 位段定义

GCDR1 (GPIO Pin Bitwise Clear Direction Register 1) Offset: 0x64 Default: 0x00000000							
Access: GPIO -> GCDR1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-71: GPIO 引脚按位设置输入寄存器 1 (GCDR1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	WO	0x0	设置 GPIO _n 引脚为输入 (n = 32~40) 0: 写 0 无效, 总是读回 0 1: 写 1 清除 GPDR1 的相应位段并设置 GPIO _n 为输入, 位段自动清零。

4.4.4.19 GPIO 上升沿检测按位使能寄存器 0 (GSRER0)

表 4-72: GPIO 上升沿检测按位使能寄存器 0 (GSRER0) 位段定义

GSRER0 (GPIO Bitwise Set Rising Edge Detect Enable Register 0) Offset: 0x6C Default: 0x00000000 Access: GPIO -> GSRER0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-73: GPIO 上升沿检测按位使能寄存器 0 (GSRER0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	WO	0x0	使能 GPIO _n 的上升沿检测 (n = 0~31) 0: 写 0 无效, 总是读回 0 1: 写 1 将 GRER0 的相应位段置 1 并使能 GPIO _n 的上升沿检测, 位段自动清零。

4.4.4.20 GPIO 上升沿检测按位使能寄存器 1 (GSRER1)

表 4-74: GPIO 上升沿检测按位使能寄存器 1 (GSRER1) 位段定义

GSRER1 (GPIO Bitwise Set Rising Edge Detect Enable Register 1) Offset: 0x70 Default: 0x00000000 Access: GPIO -> GSRER1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-75: GPIO 上升沿检测按位使能寄存器 1 (GSRER1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	WO	0x0	使能 GPIO _n 的上升沿检测 (n = 32~40) 0: 写 0 无效, 总是读回 0 1: 写 1 将 GRER1 的相应位段置 1 并使能 GPIO _n 的上升沿检测, 位段自动清零。

4.4.4.21 GPIO 上升沿检测按位关闭寄存器 0 (GCRER0)**表 4-76: GPIO 上升沿检测按位关闭寄存器 0 (GCRER0) 位段定义**

GCRER0 (GPIO Bitwise Clear Rising Edge Detect Enable Register 0) Offset: 0x78 Default: 0x00000000 Access: GPIO -> GCRER0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-77: GPIO 上升沿检测按位关闭寄存器 0 (GCRER0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	WO	0x0	关闭 GPIO _n 的上升沿检测 (n = 0~31) 0: 写 0 无效, 总是读回 0 1: 写 1 清除 GRER0 的相应位段并关闭 GPIO _n 的上升沿检测, 位段自动清零。

4.4.4.22 GPIO 上升沿检测按位关闭寄存器 1 (GCRER1)**表 4-78: GPIO 上升沿检测按位关闭寄存器 1 (GCRER1) 位段定义**

GCRER1 (GPIO Bitwise Clear Rising Edge Detect Enable Register 1) Offset: 0x7C Default: 0x00000000 Access: GPIO -> GCRER1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-79: GPIO 上升沿检测按位关闭寄存器 1 (GCRER1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	WO	0x0	关闭 GPIO _n 的上升沿检测 (n = 32~40) 0: 写 0 无效, 总是读回 0 1: 写 1 清除 GRER1 的相应位段并关闭 GPIO _n 的上升沿检测, 位段自动清零。

4.4.4.23 GPIO 下降沿检测按位使能寄存器 0 (GSFER0)

表 4-80: GPIO 下降沿检测按位使能寄存器 0 (GSFER0) 位段定义

GSFER0 (GPIO Bitwise Set Falling Edge Detect Enable Register 0) Offset: 0x84 Default: 0x00000000							
Access: GPIO -> GSFER0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-81: GPIO 下降沿检测按位使能寄存器 0 (GSFER0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	W1S	0x0	使能 GPIO _n 的下降沿检测 (n = 0~31) 0: 写 0 无效, 总是读回 0 1: 写 1 将 GFER0 的相应位段置 1 并使能 GPIO _n 的下降沿检测, 位段自动清零。

4.4.4.24 GPIO 下降沿检测按位使能寄存器 1 (GSFER1)

表 4-82: GPIO 下降沿检测按位使能寄存器 1 (GSFER1) 位段定义

GSFER1 (GPIO Bitwise Set Falling Edge Detect Enable Register 1) Offset: 0x88 Default: 0x00000000							
Access: GPIO -> GSFER1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-83: GPIO 下降沿检测按位使能寄存器 1 (GSFER1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	W1S	0x0	使能 GPIO _n 的下降沿检测 (n = 32~40) 0: 写 0 无效, 总是读回 0 1: 写 1 将 GFER1 的相应位段置 1 并使能 GPIO _n 的下降沿检测, 位段自动清零。

4.4.4.25 GPIO 下降沿检测按位关闭寄存器 0 (GCFER0)

表 4-84: GPIO 下降沿检测按位关闭寄存器 0 (GCFER0) 位段定义

GCFER0 (GPIO Bitwise Clear Falling Edge Detect Enable Register 0) Offset: 0x90 Default: 0x00000000 Access: GPIO -> GCFER0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-85: GPIO 下降沿检测按位关闭寄存器 0 (GCFER0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	WO	0x0	关闭 GPIO _n 的下降沿检测 (n = 0~31) 0: 写 0 无效, 总是读回 0 1: 写 1 清除 GFER0 的相应位段并关闭 GPIO _n 的下降沿检测, 位段自动清零。

4.4.4.26 GPIO 下降沿检测按位关闭寄存器 1 (GCFER1)

表 4-86: GPIO 下降沿检测按位关闭寄存器 1 (GCFER1) 位段定义

GCFER1 (GPIO Bitwise Clear Falling Edge Detect Enable Register 1) Offset: 0x94 Default: 0x00000000 Access: GPIO -> GCFER1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-87: GPIO 下降沿检测按位关闭寄存器 1 (GCFER1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	WO	0x0	关闭 GPIO _n 的下降沿检测 (n = 32~40) 0: 写 0 无效, 总是读回 0 1: 写 1 清除 GFER1 的相应位段并关闭 GPIO _n 的下降沿检测, 位段自动清零。

4.4.4.27 GPIO 边沿中断使能寄存器 0 (GEIE0)

表 4-88: GPIO 边沿中断使能寄存器 0 (GEIE0) 位段定义

GEIE0 (GPIO Edge-Triggered Interrupt Enable Register 0) Offset: 0x9C Default: 0x00000000							
Access: GPIO -> GEIE0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-89: GPIO 边沿中断使能寄存器 0 (GEIE0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	GPIO _n 边沿中断使能 (n = 0~31) 0: 关闭 1: 使能

4.4.4.28 GPIO 边沿中断使能寄存器 1 (GEIE1)

表 4-90: GPIO 边沿中断使能寄存器 1 (GEIE1) 位段定义

GEIE1 (GPIO Edge-Triggered Interrupt Enable Register 1) Offset: 0xA0 Default: 0x00000000							
Access: GPIO -> GEIE1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-91: GPIO 边沿中断使能寄存器 1 (GEIE1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	RW	0x0	GPIO _n 边沿中断使能 (n = 32~40) 0: 关闭 1: 使能

4.4.4.29 GPIO 电平中断标志寄存器 0 (GLIF0)

表 4-92: GPIO 电平中断标志寄存器 0 (GLIF0) 位段定义

GLIF0 (GPIO Level-Triggered Interrupt Flag Register 0) Offset: 0xA8 Default: 0x00000000							
Access: GPIO -> GLIF0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-93: GPIO 电平中断标志寄存器 0 (GLIF0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0x0	GPIO _n 电平中断标志 (n = 0~31) 0: 中断未发生 1: 中断已发生

4.4.4.30 GPIO 电平中断标志寄存器 1 (GLIF1)

表 4-94: GPIO 电平中断标志寄存器 1 (GLIF1) 位段定义

GLIF1 (GPIO Level-Triggered Interrupt Flag Register 1) Offset: 0xAC Default: 0x00000000							
Access: GPIO -> GLIF1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-95: GPIO 电平中断标志寄存器 1 (GLIF1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	RO	0x0	GPIO _n 电平中断标志 (n = 32~40) 0: 中断未发生 1: 中断已发生

4.4.4.31 GPIO 电平中断使能寄存器 0 (GLIE0)

表 4-96: GPIO 电平中断使能寄存器 0 (GLIE0) 位段定义

GLIE0 (GPIO Level-Triggered Interrupt Enable Register 0) Offset: 0xB4 Default: 0x00000000							
Access: GPIO -> GLIE0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-97: GPIO 电平中断使能寄存器 0 (GLIE0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	GPIO _n 引脚电平触发中断使能 (n = 0~31) 0: 关闭 1: 使能

4.4.4.32 GPIO 电平中断使能寄存器 1 (GLIE1)

表 4-98: GPIO 电平中断使能寄存器 1 (GLIE1) 位段定义

GLIE1 (GPIO Level-Triggered Interrupt Enable Register 1) Offset: 0xB8 Default: 0x00000000							
Access: GPIO -> GLIE1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-99: GPIO 电平中断使能寄存器 1 (GLIE1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	RW	0x0	GPIO _n 引脚电平触发中断使能 (n = 32~40) 0: 关闭 1: 使能

4.4.4.33 GPIO 电平中断清除寄存器 0 (GLIC0)

表 4-100: GPIO 电平中断清除寄存器 0 (GLIC0) 位段定义

GLIC0 (GPIO Level-Triggered Interrupt Clear Register 0) Offset: 0xC0 Default: 0x00000000							
Access: GPIO -> GLIC0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-101: GPIO 电平中断清除寄存器 0 (GLIC0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	W1C	0x0	GPIO _n 引脚电平中断标志清除 (n = 0~31) 0: 写 0 无效, 总是读回 0 1: 写 1 清除 GLIF0 相应标志位, 位段自动清零

4.4.4.34 GPIO 电平中断清除寄存器 1 (GLIC1)

表 4-102: GPIO 电平中断清除寄存器 1 (GLIC1) 位段定义

GLIC1 (GPIO Level-Triggered Interrupt Clear Register 1) Offset: 0xC4 Default: 0x00000000							
Access: GPIO -> GLIC1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-103: GPIO 电平中断清除寄存器 1 (GLIC1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	W1C	0x0	GPIO _n 引脚电平中断标志清除 (n = 0~31) 0: 写 0 无效, 总是读回 0 1: 写 1 清除 GLIF1 相应标志位, 位段自动清零

4.4.4.35 GPIO 电平中断强制寄存器 0 (GLIFRC0)

表 4-104: GPIO 电平中断强制寄存器 0 (GLIFRC0) 位段定义

GLIFRC0 (GPIO Level-Triggered Interrupt Force Register 0) Offset: 0xCC Default: 0x00000000							
Access: GPIO -> GLIFRC0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-105: GPIO 电平中断强制寄存器 0 (GLIFRC0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	W1S	0x0	软件强制产生 GPIO _n 电平中断 (n = 0~31) 0: 写 0 无效, 总是读回 0 1: 写 1 强制产生 GPIO _n 引脚的电平中断, 位段自动清零

4.4.4.36 GPIO 电平中断强制寄存器 1 (GLIFRC1)

表 4-106: GPIO 电平中断强制寄存器 1 (GLIFRC1) 位段定义

GLIFRC1 (GPIO Level-Triggered Interrupt Force Register 1) Offset: 0xD0 Default: 0x00000000							
Access: GPIO -> GLIFRC1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-107: GPIO 电平中断强制寄存器 1 (GLIFRC1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	W1S	0x0	软件强制产生 GPIO _n 电平中断 (n = 32~40) 0: 写 0 无效, 总是读回 0 1: 写 1 强制产生 GPIO _n 引脚的电平中断, 位段自动清零

4.4.4.37 GPIO 电平中断极性寄存器 0 (GLIPOL0)**表 4-108: GPIO 电平中断极性寄存器 0 (GLIPOL0) 位段定义**

GLIPOL0 (GPIO Level-Triggered Interrupt Polarity Register 0) Offset: 0xD8 Default: 0x00000000 Access: GPIO -> GLIPOL0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-109: GPIO 电平中断极性寄存器 0 (GLIPOL0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	GPIO _n 引脚触发电平中断的极性 (n = 0~31) 0: GPIO _n 低电平触发中断 1: GPIO _n 高电平触发中断

4.4.4.38 GPIO 电平中断极性寄存器 1 (GLIPOL1)**表 4-110: GPIO 电平中断极性寄存器 1 (GLIPOL1) 位段定义**

GLIPOL1 (GPIO Level-Triggered Interrupt Polarity Register 1) Offset: 0xDC Default: 0x00000000 Access: GPIO -> GLIPOL1.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							VAL
7	6	5	4	3	2	1	0
VAL							

表 4-111: GPIO 电平中断极性寄存器 1 (GLIPOL1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	RW	0x0	GPIO _n 引脚触发电平中断的极性 (n = 32~40) 0: GPIO _n 低电平触发中断 1: GPIO _n 高电平触发中断

4.4.4.39 GPIO 全局中断标志寄存器（GIF）

表 4-112: GPIO 全局中断标志寄存器（GIF）位段定义

GIF (GPIO Interrupt Flag Register) Offset: 0xE4 Default: 0x00000000							
Access: GPIO -> GIF.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED						LEVEL	EDGE

表 4-113: GPIO 全局中断标志寄存器（GIF）位段描述

位段	位段名	属性	复位值	描述
31:2	RESERVED_31_2	RO	0x0	保留
1	LEVEL	RO	0x0	GPIO 电平中断标志，本位段被清除前，GPIO 上新的电平事件将无法触发中断 0: 中断未发生 1: 中断已发生
0	EDGE	RO	0x0	GPIO 边沿中断标志，本位段被清除前，GPIO 上新的电平事件将无法触发中断 0: 中断未发生 1: 中断已发生

4.4.4.40 GPIO 全局中断清除寄存器（GIC）

表 4-114: GPIO 全局中断清除寄存器（GIC）位段定义

GIC (GPIO Interrupt Flag Clear Register) Offset: 0xE8 Default: 0x00000000							
Access: GPIO -> GIC.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED						LEVEL	EDGE

表 4-115: GPIO 全局中断清除寄存器（GIC）位段描述

位段	位段名	属性	复位值	描述
31:2	RESERVED_31_2	RO	0x0	保留
1	LEVEL	W1C	0x0	GPIO 电平中断标志清除 0: 写 0 无效，总是读回 0 1: 写 1 清除电平中断标志，位段自动清零。
0	EDGE	W1C	0x0	GPIO 边沿中断标志清除 0: 写 0 无效，总是读回 0 1: 写 1 清除边沿中断标志，位段自动清零。

4.4.4.41 GPIO 模块写使能寄存器（GPIOREGKEY）

表 4-116: GPIO 模块写使能寄存器（GPIOREGKEY）位段定义

GPIOREGKEY (GPIO Register Write-Allow Key Register) Offset: 0xEC Default: 0x1ACCE551							
Access: GPIO -> GPIOREGKEY.all							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

表 4-117: GPIO 模块写使能寄存器（GPIOREGKEY）位段描述

位段	位段名	属性	复位值	描述
31:0	KEY	RW	0x1ACCE551	写入 0x1ACCE551 以解锁受保护的 GPIO 寄存器

4.4.4.42 GPIO 上升沿中断标志寄存器 0 (GREIF0)

表 4-118: GPIO 上升沿中断标志寄存器 0 (GREIF0) 位段定义

GREIF (GPIO Rising Edge Interrupt Flag Register 0) Access: GPIO->GREIF[n] Offset: 0x00F0 Default: 0x00000000							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-119: GPIO 上升沿中断标志寄存器 0 (GREIF0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0x0	锁存的 GPIO _n 上升沿中断标志 (n = 0~31) 0: GPIO _n 引脚不曾触发上升沿中断 1: GPIO _n 引脚曾触发上升沿中断

4.4.4.43 GPIO 上升沿中断标志寄存器 1 (GREIF1)

表 4-120: GPIO 上升沿中断标志寄存器 1 (GREIF1) 位段定义

GREIF (GPIO Rising Edge Interrupt Flag Register 1) Access: GPIO->GREIF[n] Offset: 0x00F4 Default: 0x00000000							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-121: GPIO 上升沿中断标志寄存器 1 (GREIF1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	RO	0x0	锁存的 GPIO _n 上升沿中断标志 (n = 32~40) 0: GPIO _n 引脚不曾触发上升沿中断 1: GPIO _n 引脚曾触发上升沿中断

4.4.4.44 GPIO 下降沿中断标志寄存器 0 (GFEIF0)

表 4-122: GPIO 下降沿中断标志寄存器 0 (GFEIF0) 位段定义

GFEIF (GPIO Falling Edge Interrupt Flag Register 0) Access: GPIO->GFEIF[n] Offset: 0x00F8 Default: 0x00000000							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-123: GPIO 下降沿中断标志寄存器 0 (GFEIF0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0x0	锁存的 GPIO _n 下降沿中断标志 (n = 0~31) 0: GPIO _n 引脚不曾触发下降沿中断 1: GPIO _n 引脚曾触发下降沿中断

4.4.4.45 GPIO 下降沿中断标志寄存器 1 (GFEIF1)

表 4-124: GPIO 下降沿中断标志寄存器 1 (GFEIF1) 位段定义

GREIF (GPIO Falling Edge Interrupt Flag Register 1) Access: GPIO->GREIF[n] Offset: 0x00FC Default: 0x00000000							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 4-125: GPIO 下降沿中断标志寄存器 1 (GFEIF1) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:0	VAL	RO	0x0	锁存的 GPIO _n 下降沿中断标志 (n = 0~31) 0: GPIO _n 引脚不曾触发下降沿中断 1: GPIO _n 引脚曾触发下降沿中断

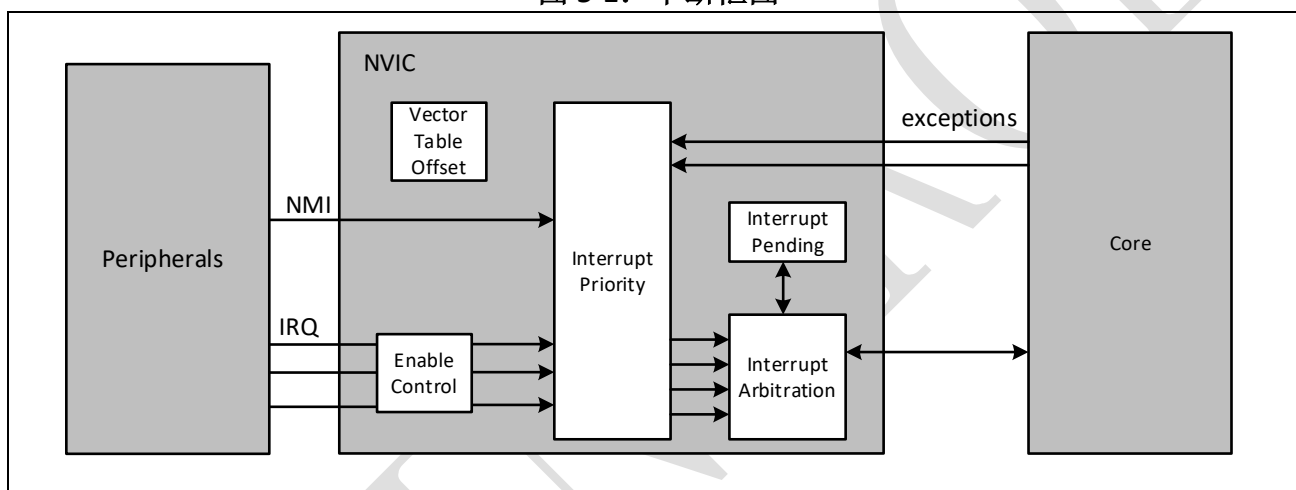
5 NVIC

5.1 概述

嵌套向量中断控制器（Nested Vectored Interrupt Controller, NVIC）是 ARM® Cortex-M4 内核的组成部分，与内核紧密配合，可实现低延迟中断处理和高效的晚到中断。SPD1121 内置 NVIC 具备如下特性：

- 支持 29 个中断向量（不包括 Cortex-M4 的 16 个系统异常）；
- 支持 16 个可编程优先级（4 位优先级配置），数值越大优先级越低（0 为优先级最高）；
- 支持咬尾中断和晚到处理；
- 支持不可屏蔽中断（NMI），用于处理关键硬件错误。

图 5-1：中断框图



5.2 中断向量表

中断向量表是 Cortex-M4 处理器的中断处理核心结构，其内容第一个 word 数据为主堆栈（MSP）指针，第二个 word 数据为复位处理程序入口地址，后续按中断号顺序排列。处理器通过中断向量表的基地址（SCB->VTOR）和中断号获取中断服务函数入口地址。

SPD1121 的中断向量表如表 5-1 所示。

表 5-1: SPD1121 的中断向量表

异常号	CMSIS 中断号	缩略语	优先级	描述	偏移地址
—	—	MSP	—	初始化主堆栈指针	0x0000 0000
1	—	Reset	-3, 最高级	复位	0x0000 0004
2	-14	NMI	-2	不可屏蔽中断， 看门狗定时器 0 超时触发	0x0000 0008
3	-13	Hard Fault	-1	所有类型的故障	0x0000 000C
4	-12	MemManage	可配置	存储器管理故障	0x0000 0010
5	-11	Bus Fault	可配置	总线系统故障	0x0000 0014
6	-10	Usage Fault	可配置	用法故障	0x0000 0018
7	—	—	—	保留	0x0000 001C
8	—	—	—	保留	0x0000 0020
9	—	—	—	保留	0x0000 0024
10	—	—	—	保留	0x0000 0028
11	-5	SVCall	可配置	通过 SVC 指令调用的系统 服务	0x0000 002C
12	-4	Debug Monitor	可配置	调试监视器	0x0000 0030
13	—	—	—	保留	0x0000 0034
14	-2	PendSV	可配置	可挂起的系统服务	0x0000 0038
15	-1	SysTick	可配置	系统节拍定时器中断	0x0000 003C
16	0	MEM	可配置	存储器数据异常中断	0x0000 0040
17	1	POWER	可配置	电源过压欠压异常中断	0x0000 0044
18	2	CLOCK	可配置	时钟异常中断	0x0000 0048
19	3	WDT1	可配置	看门狗定时器 1 中断	0x0000 004C
20	4	GPIO_LEVEL	可配置	GPIO 引脚电平触发的中断	0x0000 0050
21	5	GPIO_EDGE	可配置	GPIO 引脚边沿触发的中断	0x0000 0054
22	6	UART	可配置	UART 中断	0x0000 0058
23	7	SPI	可配置	SPI 中断	0x0000 005C
24	8	I2C	可配置	I2C 中断	0x0000 0060
25	9	ADC0	可配置	ADC 通道 0 转换结束中断	0x0000 0064
26	10	ADC1	可配置	ADC 通道 1 转换结束中断	0x0000 0068
27	11	ADC2	可配置	ADC 通道 2 转换结束中断	0x0000 006C
28	12	ADC3	可配置	ADC 通道 3 转换结束中断	0x0000 0070
29	13	ADC4	可配置	ADC 通道 4 转换结束中断	0x0000 0074
30	14	ADC5	可配置	ADC 通道 5 转换结束中断	0x0000 0078
31	15	ADC6	可配置	ADC 通道 6 转换结束中断	0x0000 007C
32	16	ADC7	可配置	ADC 通道 7 转换结束中断	0x0000 0080

异常号	CMSIS 中断号	缩略语	优先级	描述	偏移地址
33	17	PWM2	可配置	PWM2 中断	0x0000 0084
34	18	PWM3	可配置	PWM3 中断	0x0000 0088
35	19	PWM4	可配置	PWM4 中断	0x0000 008C
36	20	PWM5	可配置	PWM5 中断	0x0000 0090
37	21	PWM2TZ	可配置	PWM2 封锁中断	0x0000 0094
38	22	PWM3TZ	可配置	PWM3 封锁中断	0x0000 0098
39	23	PWM4TZ	可配置	PWM4 封锁中断	0x0000 009C
40	24	PWM5TZ	可配置	PWM5 封锁中断	0x0000 00A0
41	25	ECAP	可配置	ECAP 中断	0x0000 00A4
42	26	TIMER0	可配置	TIMER0 中断	0x0000 00A8
43	27	TIMER1	可配置	TIMER1 中断	0x0000 00AC
44	28	CRC	可配置	CRC 中断	0x0000 00B0

5.3 中断使能

如图 5-1 所示，外设的中断信号，首先要传递给 NVIC 模块，随后才能被 Core 处理。也即，在实际使用过程中除了要使能外设中断外，也需要调用 CMSIS 库的 NVIC_EnableIRQ (MODULE_IRQn)函数使能 NVIC 对此中断的处理。

5.4 中断优先级

优先级对于中断来说非常重要，决定中断服务程序执行的优先级。在 ARMv7 架构中提供 8 比特用于设置优先级，但 CM4 仅使用高 4 个比特。

表 5-2：中断优先级寄存器（IP/SHP）

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
用于设置优先级				未实现			

如表 5-2 所示，该高 4 个比特可以被分组为两个位段：抢占优先级和子优先级。

- 抢占优先级：抢占优先级决定了抢占行为，当系统正在响应中断 L 时，来了抢占优先级更高的中断 H，则 H 抢占 CPU 执行权，L 丢失 CPU 执行权。
- 子优先级：当抢占优先级相同的中断有多个挂起时，则优先响应子优先级最高的中断。

如表 5-3 所示，用户也可通过调用 CMSIS 库的 NVIC_SetPriorityGrouping(PriorityGroup)函数进行重新分组，其中 PriorityGroup 取值范围为 0~7。

表 5-3：优先级分组寄存器（AIRC.R. PRIGROUP）

PriorityGroup	表达抢占优先级的位段	表达子优先级的位段
0（默认）	[bit7:bit4]	-
1	[bit7:bit4]	-
2	[bit7:bit4]	-
3	[bit7:bit4]	-
4	[bit7:bit5]	[bit4:bit4]
5	[bit7:bit6]	[bit5:bit4]
6	[bit7:bit7]	[bit6:bit4]
7	-	[bit7:bit4]

中断优先级可以通过调用 CMSIS 库的 NVIC_SetPriority(MODULE_IRQn, priority)函数进行设置，其中 priority 取值范围为 0~15，值越小优先级越高。

如表 5-4 所示，将 2 比特设置为抢占优先级，2 比特设置为子优先级，并且抢占优先级为 2，子优先级为 1，则需要依次调用 NVIC_SetPriorityGrouping(5)，以及 NVIC_SetPriority(MODULE_IRQn, (1<<2)|(1<<0))。

表 5-4：配置中断优先级寄存器（IP/SHP）

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
抢占优先级		子优先级		未实现			
1	0	0	1	未实现			

5.5 中断仲裁和挂起

在 Cortex-M 处理器的中断管理体系中，中断仲裁和中断挂起是两个核心机制，共同确保系统能够高效、有序地处理多个并发中断请求。

中断仲裁机制：

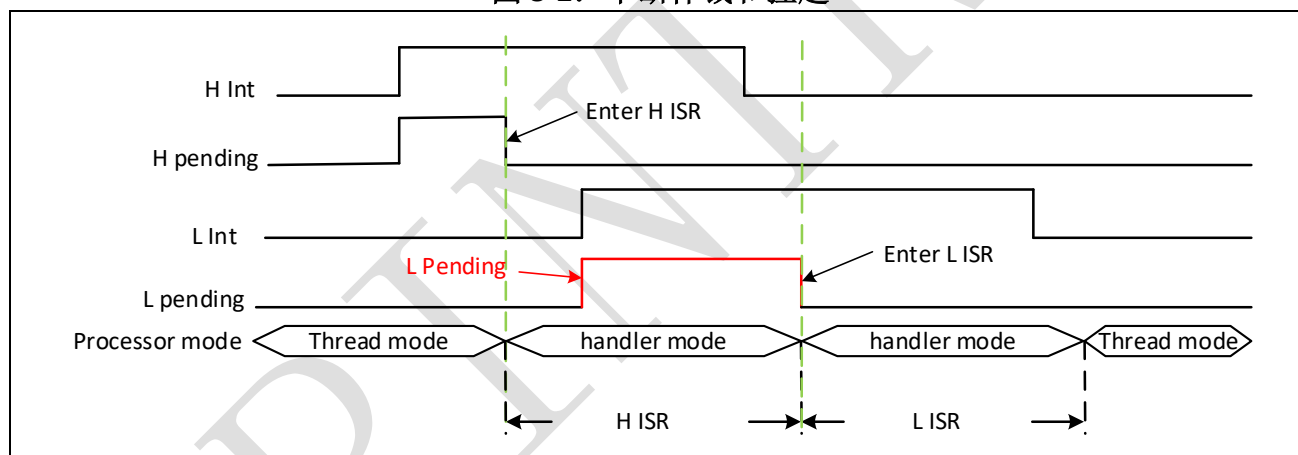
- 高抢占优先级中断可立即抢占低优先级中断；
- 同抢占优先级中断按子优先级排序，若子优先级相同则按中断号排序（中断号越小优先级越高）；
- 低优先级中断永远无法抢占高优先级中断；

中断挂起机制：

- 当中断被触发但无法立即响应时，NVIC 会将其挂起；

如图 5-4 所示，当低优先级中断（L Int）产生时，此时 CPU 正在处理高优先级中断服务函数（H ISR），则低优先级中断（L Int）会被挂起。当 CPU 处理完高优先级中断中断服务函数（H ISR），则被挂起的低优先级中断（L Int）将获得 CPU 执行权。

图 5-2：中断仲裁和挂起

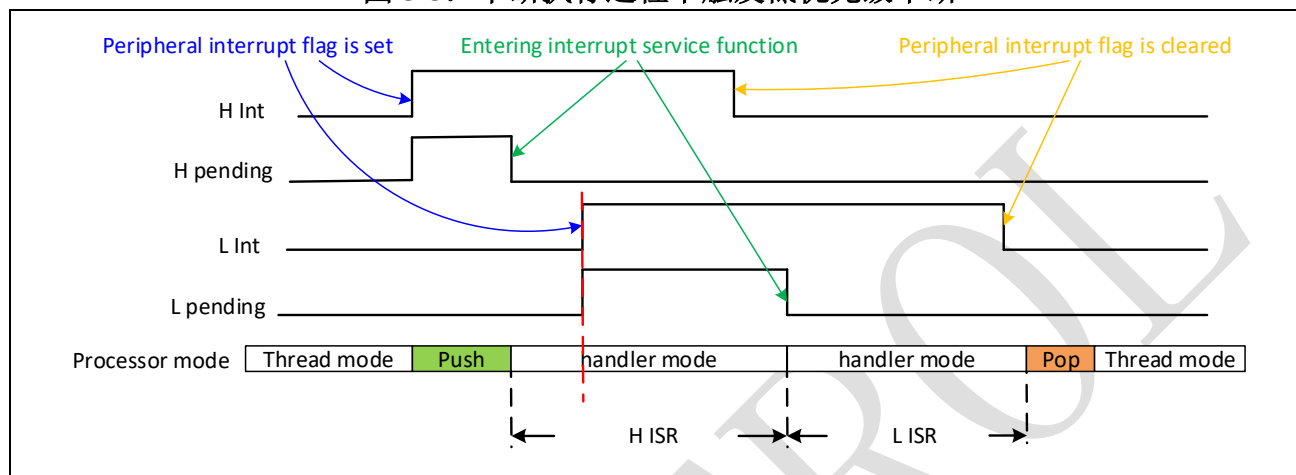


5.6 应用场景

5.6.1 同优先级和低优先级挂起

当高优先级中断正在执行时，可能存在以下三种场景触发相同优先级或低优先级的中断。

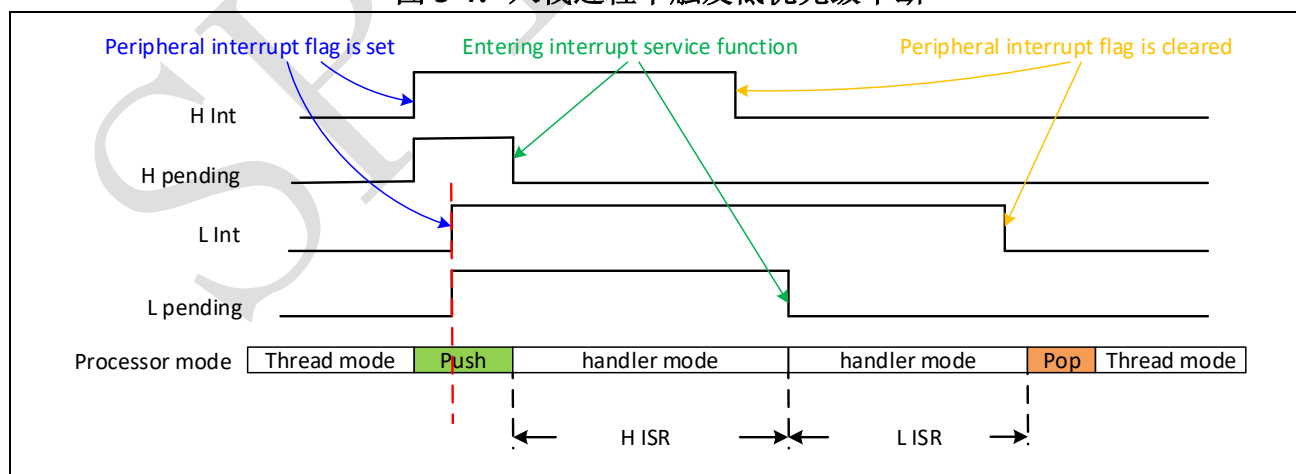
图 5-3：中断执行过程中触发低优先级中断



- 如图 5-3 所示，在高优先级中断服务函数（H ISR）执行过程中触发相同优先级或低优先级的中断（L Int），NVIC 会按以下流程处理：

 1. 将相同优先级或更低优先级的中断（L Int）挂起。
 2. 高优先级中断的服务函数（H ISR）继续执行直至完成。
 3. 执行完高优先级服务函数（H ISR）后，直接跳转执行相同优先级或更低优先级的中断服务程序（L ISR）。
 4. 最终完成同优先级或更低优先级中断（L Int）的出栈操作，返回主程序。

图 5-4：入栈过程中触发低优先级中断

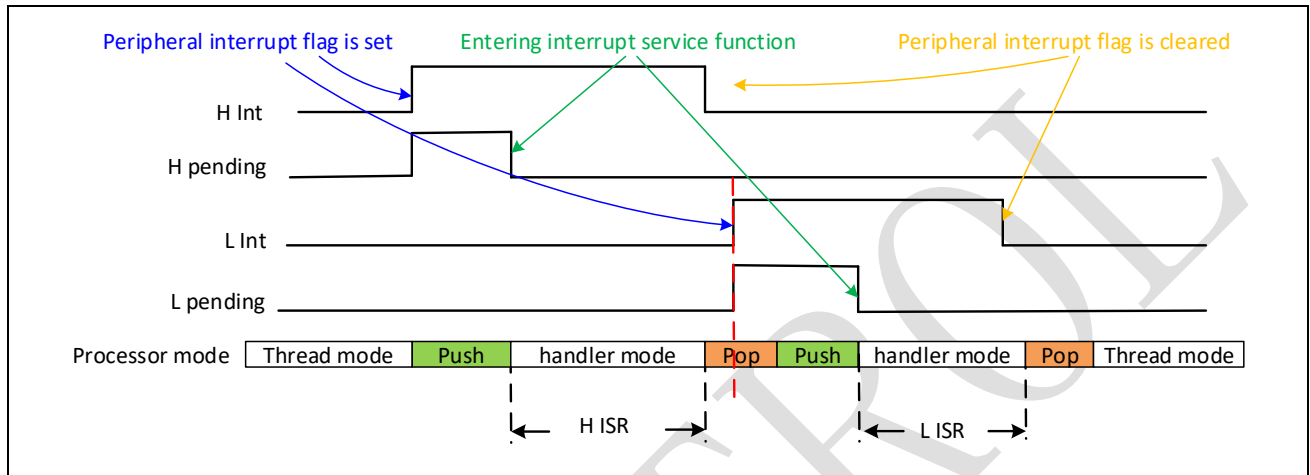


- 如图 5-4 所示，在高优先级中断（H Int）入栈过程中触发相同优先级或低优先级的中断（L Int），NVIC 将按以下流程处理：

 1. 立即将相同优先级或更低优先级的中断（L Int）挂起。

2. 高优先级中断（H Int）的入栈操作会继续执行，不会被打断。
3. 执行高优先级中断服务函数（H ISR）。
4. 执行完高优先级服务函数（H ISR）后，直接跳转执行相同优先级或更低优先级的中断服务函数（L ISR）。
5. 最终完成同优先级或更低优先级中断（L Int）的出栈操作，返回主程序。

图 5-5：出栈过程中触发低优先级中断

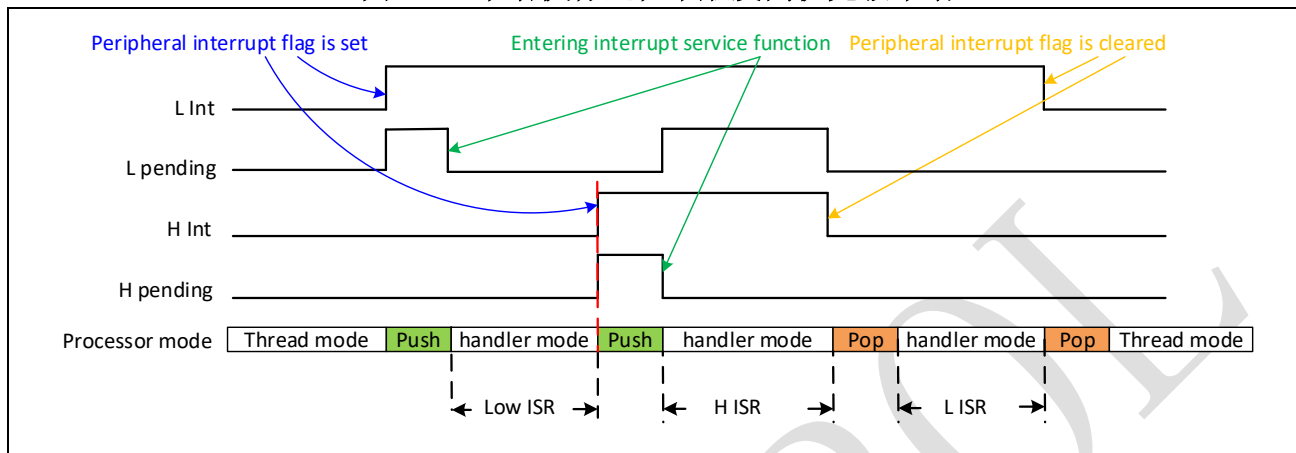


- 如图 5-5 所示，在高优先级中断（H Int）出栈过程中触发相同优先级或低优先级的中断（L Int），NVIC 会按以下流程处理：
 1. 将同优先级或更低优先级的中断（L Int）挂起。
 2. 高优先级中断（H Int）的出栈操作会继续执行，不会被打断。
 3. 同优先级或更低优先级的中断（L Int）执行入栈操作。
 4. 开始执行同优先级或更低优先级中断的服务函数（L ISR）。
 5. 最终完成同优先级或更低优先级中断（L Int）的出栈操作，返回主程序。

5.6.2 高优先级抢占

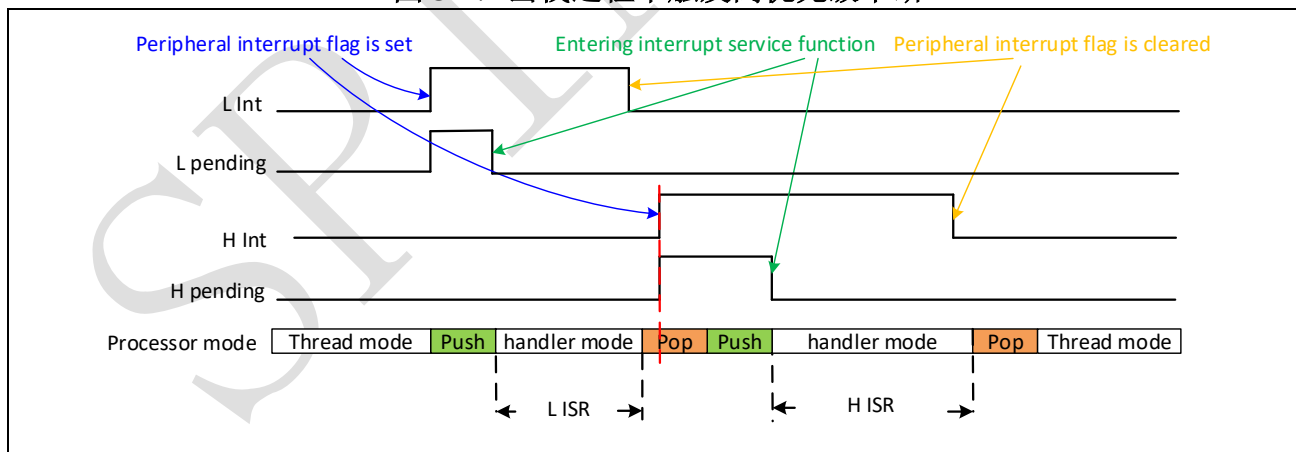
当低优先级中断（L Int）正在执行时，可能存在以下三种场景触发高优先级的中断（H Int）。

图 5-6：中断执行过程中触发高优先级中断



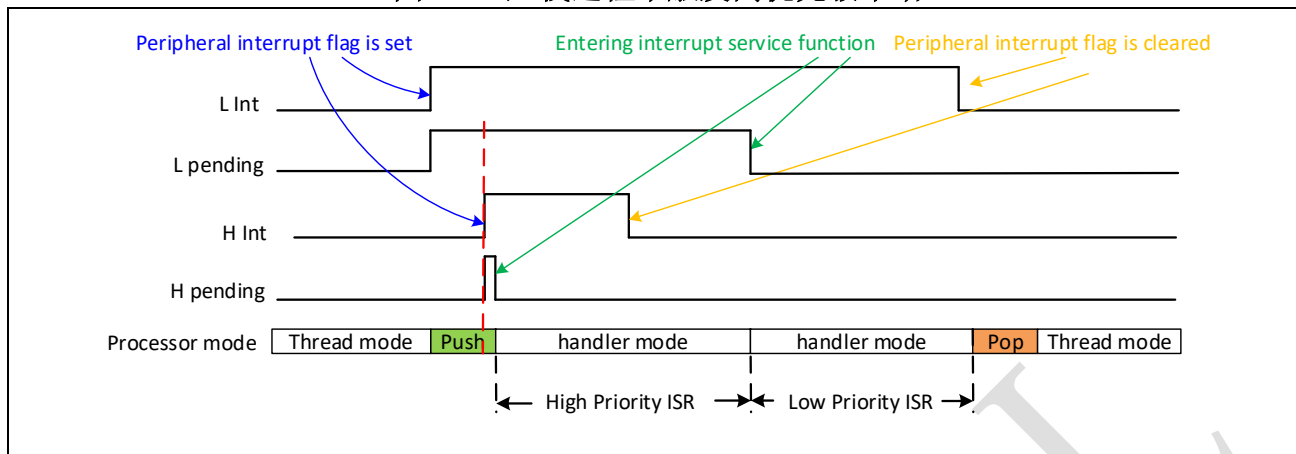
- 如图 5-6 所示，在低优先级中断服务函数（L ISR）执行过程中触发高优先级的中断（H Int），NVIC 会按以下流程处理：
 1. 高优先级中断（H Int）执行入栈操作。
 2. 低优先级中断（L Int）挂起。
 3. 执行高优先级中断服务函数（H ISR）。
 4. 高优先级中断（H Int）执行出栈操作。
 5. 恢复执行被挂起的低优先级中断服务函数（L ISR）。
 6. 最终完成低优先级中断（L Int）的出栈操作，返回主程序。

图 5-7：出栈过程中触发高优先级中断



- 如图 5-7 所示，在低优先级中断（L Int）出栈过程中触发高优先级的中断（H Int），NVIC 会按以下流程处理：
 1. 低优先级中断（L Int）的出栈操作会继续执行，不会被打断。
 2. 高优先级中断（H Int）执行入栈操作。
 3. 执行高优先级中断的服务函数（H ISR）。
 4. 最终完成高优先级中断（H Int）的出栈操作，返回主程序。

图 5-8：入栈过程中触发高优先级中断



- 如图 5-8 所示，在低优先级中断（L Int）入栈过程中触发高优先级的中断（H Int），NVIC 会按以下流程处理：
 1. 低优先级中断（L Int）入栈操作会继续执行，不会被打断。
 2. 直接执行高优先级中断服务函数（H ISR）。
 3. 高优先级中断服务函数（H ISR）执行完毕后，直接继续执行低优先级服务函数（L ISR）。
 4. 最后完成低优先级中断（L Int）的出栈操作，返回主程序。

6 循环冗余校验模块（CRC）

6.1 概述

循环冗余校验（Cyclic Redundancy Check）或者叫做多项式码校验是一种哈希函数，被设计用于检查原始计算数据的意外变化，通常用于数据传输或者存储过程中的错误检测。

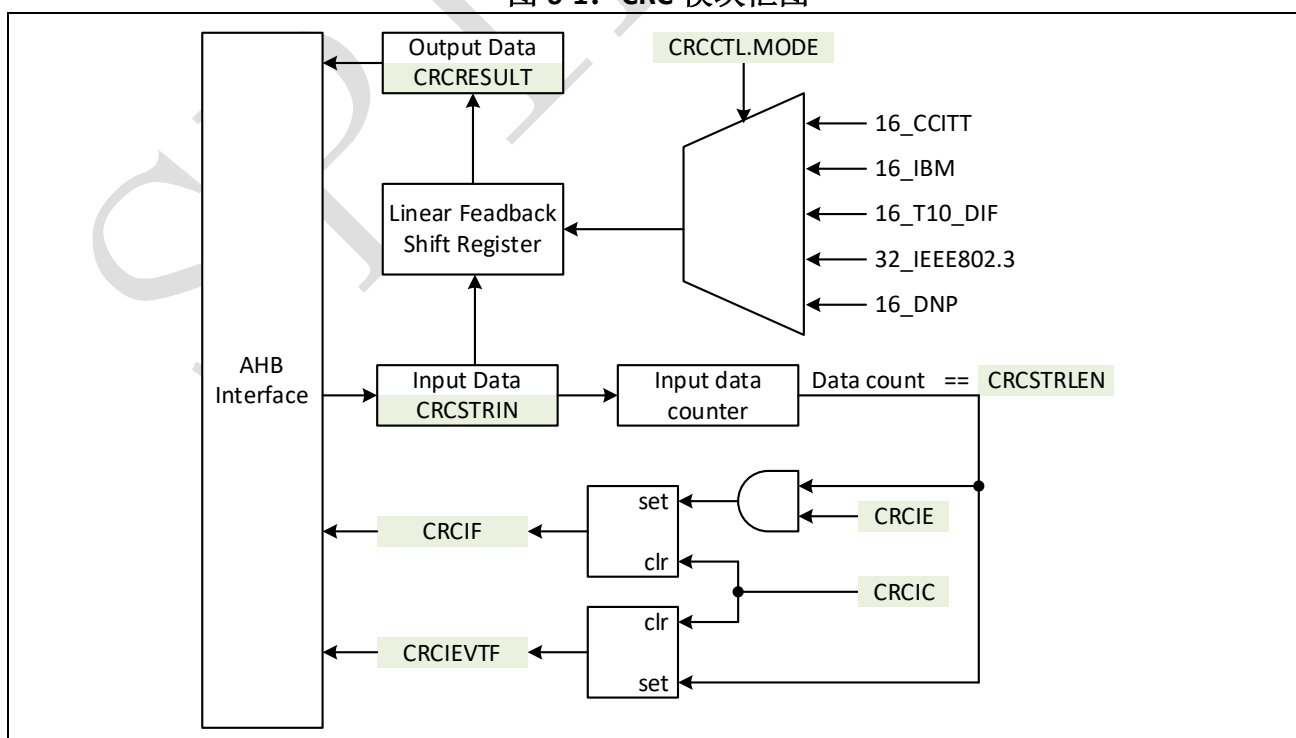
SPD1121 提供 1 个 CRC 模块，其特性如下：

- 支持 32 位并行数据流输入。
- 支持多达 32 位 CRC 值输出。
- 支持多达 2^{32} （4294967296）字节长度的数据的 CRC 计算。
- 支持如下 5 个 CRC 标准以及对应的多项式
 - CRC-16-CCITT: $x^{16} + x^{12} + x^5 + 1$
 - CRC-16-IBM: $x^{16} + x^{15} + x^2 + 1$
 - CRC-16-T10-DIF: $x^{16} + x^{15} + x^{11} + x^9 + x^8 + x^7 + x^5 + x^4 + x^2 + 1$
 - CRC-32-IEEE 802.3: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$
 - CRC-16-DNP: $x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^2 + 1$

如图 6-1 所示，CRC 模块包含如下子模块：

- 输入输出数据流控制。
- 数据字节长度控制。
- 线性反馈移位寄存器：依据不同多项式进行 CRC 运算。
- AHB 接口以及中断信号管理。

图 6-1：CRC 模块框图



6.2 CRC 工作流程

用户可以根据以下步骤操作 CRC 模块：

- 步骤 1：关闭 CRC（设置 CRCCTL.EN 位为 0）
- 步骤 2：关闭中断（设置 CRCIE.DONE 位为 0）
- 步骤 3：清除中断标志（设置 CRCIC.DONE 为 1）
- 步骤 4：配置数据流长度（设置 CRCSTRLEN 寄存器）
- 步骤 5：配置 CRC 工作模式（设置 CRCCTL.MODE 位段）
- 步骤 6：使能中断（设置 CRCIE.DONE 位为 1）
- 步骤 7：使能 CRC（设置 CRCCTL.EN 位为 1）
- 步骤 8：写入数据流，等待中断发生（如果中断发生，转到步骤 9）
- 步骤 9：获取 CRC 计算结果（读 CRCRESULT 寄存器）
- 步骤 10：CRC 操作完成

注意：CRC 数据流输入寄存器一次接受一个字（32 位）数据。如果输入数据不是 4 字节对齐的，则会在数据流的开始填充 0。例如，如果数据流由 5 个字节构成，从低地址开始依次为：0xA1, 0xA2, 0xA3, 0xA4, 0xA5，那么下面两个字数据应该被写入数据流输入寄存器：

- 0xA1000000
- 0xA5A4A3A2

CRC 结果的位序为：

- 16 位 CRC: $x_0 \sim x_{15}$ [MSB->LSB]
 - 32 位 CRC: $x_0 \sim x_{31}$ [MSB->LSB]
-

6.3 寄存器

6.3.1 CRC 寄存器列表

表 6-1: CRC 模块基地址

外设模块	基地址
CRC	0x4000 8000

表 6-2: CRC 寄存器列表

寄存器	偏移地址	描述	复位值
CRCIF	0x00	CRC 中断标志寄存器	0x00000000
CRCIEVTF	0x04	CRC 中断事件标志寄存器	0x00000000
CRCIC	0x08	CRC 中断清除寄存器	0x00000000
CRCIE	0x0C	CRC 中断使能寄存器	0x00000000
CRCCTL	0x10	CRC 控制寄存器	0x00000000
CRCSTRLEN	0x14	CRC 数据流长度寄存器	0x00000000
CRCSTRIN	0x18	CRC 数据流输入寄存器	0x00000000
CRCRESULT	0x1C	CRC 结果寄存器	0x00000000

6.3.2 CRC 寄存器

表 6-3 到表 6-18 提供了 CRC 模块相关寄存器的详细信息。

6.3.2.1 CRC 中断标志寄存器 (CRCIF)

表 6-3: CRC 中断标志寄存器 (CRCIF) 位段定义

CRCIF (CRC Interrupt Flag Register)							
Access: CRC->CRCIF		Offset: 0x0000		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							DONE

表 6-4: CRC 中断标志寄存器 (CRCIF) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED_31_1	RO	0x0	保留
0	DONE	RO	0x0	CRC 计算完成中断标志 0: 中断未发生 1: 中断已发生, 在清除该比特位之前, 不会产生进一步的中断。

6.3.2.2 CRC 中断事件标志寄存器（CRCIEVTF）

表 6-5: CRC 中断事件标志寄存器（CRCIEVTF）位段定义

CRCIEVTF (CRC Interrupt Event Flag Register)							
Access: CRC->CRCIEVTF Offset: 0x0004 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							DONE

表 6-6: CRC 中断事件标志寄存器（CRCIEVTF）位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED_31_1	RO	0x0	保留
0	DONE	RO	0x0	CRC 计算完成中断事件标志 0: 中断未发生 1: 中断已发生

6.3.2.3 CRC 中断清除寄存器（CRCIC）

表 6-7: CRC 中断清除寄存器（CRCIC）位段定义

CRCIC (CRC Interrupt Clear Register)							
Access: CRC->CRCIC Offset: 0x0008 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							DONE

表 6-8: CRC 中断清除寄存器（CRCIC）位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED_31_1	RO	0x0	保留
0	DONE	W1C	0x0	清除 CRCIF 和 CRCIEVTF 寄存器中的标志位 0: 写 0 无效，总是读回 0 1: 写 1 清除 CRCIF 和 CRCIEVTF 寄存器中的标志位，本位段自动清零

6.3.2.4 CRC 中断使能寄存器（CRCIE）

表 6-9: CRC 中断使能寄存器（CRCIE）位段定义

CRCIE (CRC Interrupt Enable Register)							
Access: CRC->CRCIE		Offset: 0x000C		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							DONE

表 6-10: CRC 中断使能寄存器（CRCIE）位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED_31_1	RO	0x0	保留
0	DONE	RW	0x0	使能中断（由 CRCIEVTF 标志位触发） 0: 关闭中断请求以及 CRCIF 寄存器中相关标志位的产生 1: 使能中断请求以及 CRCIF 寄存器中相关标志位的产生

6.3.2.5 CRC 控制寄存器（CRCCTL）

表 6-11: CRC 控制寄存器（CRCCTL）位段定义

CRCCTL (CRC Control Register) Access: CRC->CRCCTL Offset: 0x0010 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED			CONTINUOUS	MODE			EN

表 6-12: CRC 控制寄存器（CRCCTL）位段描述

位段	位段名	属性	复位值	描述
31:5	RESERVED_31_5	RO	0x0	保留
4	CONTINUOUS	RW	0x0	连续框架计算 0: CRCCTL 时清除内部寄存器。EN=0 所以下一帧计算是一个新的开始 1: 内部寄存器保存 CRCCTL 时的值。EN=0，因此下一帧计算连续到当前帧
3:1	MODE	RW	0x0	CRC 模式选择 0: $x^{16}+x^{12}+x^5+1$ (CRC-16-CCITT, CRC-CCITT) 1: $x^{16}+x^{15}+x^2+1$ (CRC-16, CRC-16-IBM, CRC-16-ANSI) 2: $x^{16}+x^{15}+x^{11}+x^9+x^8+x^7+x^5+x^4+x^2+x+1$ (CRC-16-T10-DIF) 3: $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$ (CRC-32-IEEE802.3) 4: $x^{16}+x^{13}+x^{12}+x^{11}+x^{10}+x^8+x^6+x^5+x^2+1$ (CRC-16-DNP)
0	EN	RW	0x0	使能 CRC 计算 0: 关闭 CRC 计算 1: 使能 CRC 计算，CRC 计算完成后自动清零

6.3.2.6 CRC 数据流长度寄存器 (CRCSTRLEN)

表 6-13: CRC 数据流长度寄存器 (CRCSTRLEN) 位段定义

CRCSTRLEN (CRC Stream Length Register)							
Access: CRC->CRCSTRLEN		Offset: 0x0014		Default: 0x00000000			
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 6-14: CRC 数据流长度寄存器 (CRCSTRLEN) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	设置输入流长度为(CRCSTRLEN+1)字节

6.3.2.7 CRC 数据流输入寄存器 (CRCSTRIN)

表 6-15: CRC 数据流输入寄存器 (CRCSTRIN) 位段定义

CRCSTRIN (CRC Stream Input Register)							
Access: CRC->CRCSTRIN		Offset: 0x0018		Default: 0x00000000			
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 6-16: CRC 数据流输入寄存器 (CRCSTRIN) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	输入数据寄存器

6.3.2.8 CRC 结果寄存器（CRCRESULT）

表 6-17: CRC 结果寄存器（CRCRESULT）位段定义

CRCRESULT (CRC Result Register)							
Access: CRC->CRCRESULT		Offset: 0x001C		Default: 0x00000000			
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 6-18: CRC 结果寄存器（CRCRESULT）位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0x0	CRC 计算结果

7 通用定时器

7.1 概述

SPD1121 内置 2 个 32 位通用定时器（General Purpose Timers, GPT），其特性如下：

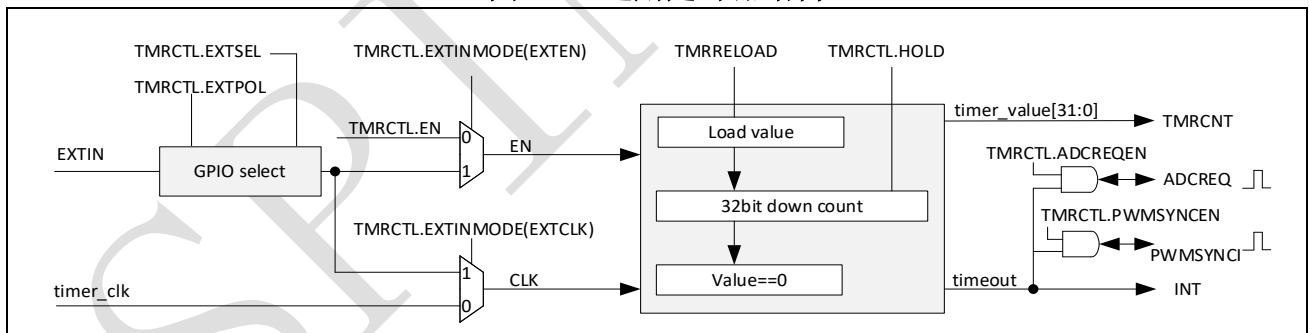
- 专属 TIMER 时钟
- 32 位向下计数器
- 当计数到 0 时产生中断
- 当计数到 0 时产生 ADCREQ 信号
- 当计数到 0 时产生 PWMSYNCI 事件
- 捕获外部输入作为定时器的使能信号
- 捕获外部输入作为定时器的时钟源

7.2 功能描述

每一个通用定时器（GPT）都工作在周期性计时模式。如图 7-1 所示，当使能定时器之后，计数器加载 TMRRELOAD 寄存器的值，并开始向下计数直到 0。然后，计数器从 TMRRELOAD 寄存器的值重新开始，并向向下计数直到 0。当计数器向下计数到 0 时，有如下动作发生：

- 如果使能了定时器中断，将会有中断产生，这个中断标志将会一直存在直到被手动清除
- 如果 TMRCTL.ADCREQEN 使能，将会产生一个 ADCREQ 事件
- 如果 TMRCTL.PWMSYNCEN 使能，将会产生一个 PWMSYNCI 事件

图 7-1：通用定时器结构



TMRCTL.EXTSEL 可以选择不同的 GPIO 作为外部输入信号脚（EXTIN），外部信号输入脚可用作 TIMER 的使能信号或时钟信号，极性可配。详细信息，请参见表 7-3 和表 7-4。

在系统复位之后，定时器默认为关闭状态。

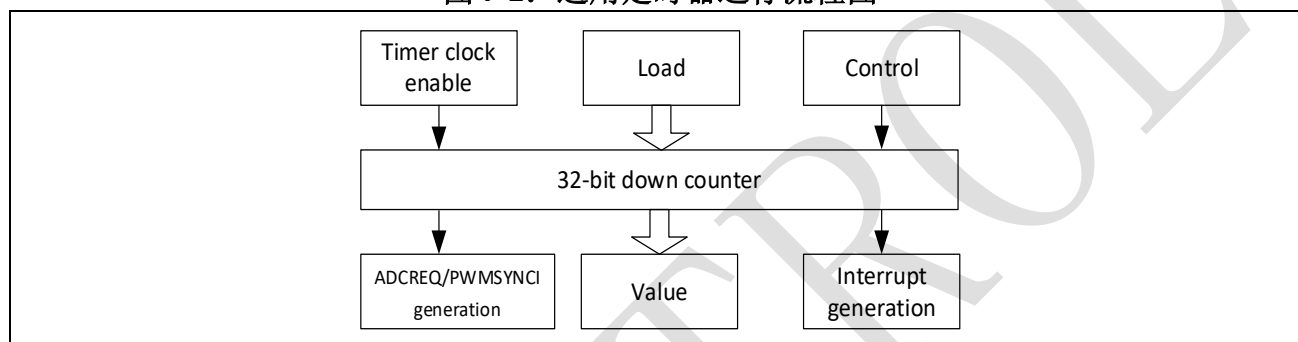
7.3 初始化流程

应当遵照下面的流程来启动定时器：

- 关闭定时器，并清除定时器中断
- 配置定时器重加载寄存器（TMRRELOAD）
- 配置定时器控制寄存器（TMRCTL）
- 使能定时器开始计数

定时器运行运行流程框图如图 7-2 所示。

图 7-2：通用定时器运行流程图



7.4 寄存器

7.4.1 TIMER 寄存器列表

表 7-1: TIMER 模块基地址

外设模块	基地址
TIMER0	0x4000 7000
TIMER1	0x4000 7020

表 7-2: TIMER 寄存器列表

寄存器	偏移地址	描述	复位值
TMRCTL	0x0000	定时器控制寄存器	0x00000000
TMRCNT	0x0004	定时器计数器值寄存器	0x00000000
TMRRELOAD	0x0008	定时器超时重载寄存器	0x00000000
TMRIF	0x000C	定时器中断标志寄存器	0x00000000
TMRIEVTF	0x0010	定时器中断事件标志寄存器	0x00000000
TMRIE	0x0014	定时器中断使能寄存器	0x00000000
TMRIFRC	0x0018	定时器中断强制寄存器	0x00000000
TMRIC	0x001C	定时器中断清除寄存器	0x00000000

7.4.2 TIMER 寄存器

7.4.2.1 定时器控制寄存器（TMRCTL）

表 7-3：定时器控制寄存器（TMRCTL）位段定义

TMRCTL (Timer Control Register)							
Access: TIMER->TMRCTL Offset: 0x0000 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED		SHDWLD	EXTSEL				
7	6	5	4	3	2	1	0
EXTSEL	EXTPOL	EXTINMODE		PWMSYNCEN	ADCREQEN	HOLD	EN

表 7-4：定时器控制寄存器（TMRCTL）位段描述

位段	位段名	属性	复位值	描述
31:14	RESERVED_31_14	RO	0x0	保留
13	SHDWLD	RW	0x0	重载影子使能 0: TMRRELOAD 上的更新将立即加载到计数器中 1: TMRRELOAD 上的更新不影响当前计数器的值
12:7	EXTSEL	RW	0x0	选择作为外部输入源的 GPIO 管脚
6	EXTPOL	RW	0x0	外部输入极性 0: 低有效 1: 高有效
5:4	EXTINMODE	RW	0x0	外部输入模式。 0: 关闭外部输入 1: 无效 2: 外部输入作为定时器时钟信号 3: 外部输入作为定时器使能信号
3	PWMSYNCEN	RW	0x0	使能 PWMSYNCCI 信号产生 0: 不产生 PWMSYNCCI 信号 1: 当 TMRCNT 向下计数到 0 时，产生 PWMSYNCCI
2	ADCREQEN	RW	0x0	使能 ADCREQ 信号产生 0: 不产生 ADCREQ 信号 1: 当 TMRCNT 向下计数到 0 时，产生 ADCREQ
1	HOLD	RW	0x0	在定时器被关闭时持有计数器值 0: 定时器被关闭时将计数器重置为 0 1: 定时器被关闭时保持计数器值

位段	位段名	属性	复位值	描述
0	EN	RW	0x0	使能定时器 0: 关闭 1: 使能

7.4.2.2 定时器计数器值寄存器 (TMRCNT)

表 7-5: 定时器计数器值寄存器 (TMRCNT) 位段定义

TMRCNT (Timer Counter Value Register) Access: TIMER->TMRCNT Offset: 0x0004 Default: 0x00000000							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 7-6: 定时器计数器值寄存器 (TMRCNT) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	定时器计数器的当前计数值

7.4.2.3 定时器超时重载寄存器 (TMRRELOAD)

表 7-7: 定时器超时重载寄存器 (TMRRELOAD) 位段定义

TMRRELOAD (Timer Reload Value Register) Access: TIMER->TMRRELOAD Offset: 0x0008 Default: 0x00000000							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 7-8: 定时器超时重载寄存器 (TMRRELOAD) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	计数器超时重载值

7.4.2.4 定时器中断标志寄存器 (TMRIF)

表 7-9: 定时器中断标志寄存器 (TMRIF) 位段定义

TMRIF (Timer Interrupt Flag Register)							
Access: TIMER->TMRIF Offset: 0x000C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							INT

表 7-10: 定时器中断标志寄存器 (TMRIF) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED_31_1	RO	0x0	保留
0	INT	RO	0x0	定时器中断标志 0: 中断未产生。 1: 中断已产生, 并已经通知 CPU, 在这个标志位被清除前不会产生新的中断。

7.4.2.5 定时器中断事件标志寄存器 (TMRIEVTF)

表 7-11: 定时器中断事件标志寄存器 (TMRIEVTF) 位段定义

TMRIEVTF (Timer Interrupt Event Flag Register)							
Access: TIMER->TMRIEVTF Offset: 0x0010 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							INT

表 7-12: 定时器中断事件标志寄存器 (TMRIEVTF) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED_31_1	RO	0x0	保留
0	INT	RO	0x0	定时器原始中断状态 当 TMRcnt 向下计数到 0 时, 本位段被置“1”, 且只能通过向 TMRIC 寄存器写 1 清除。 0: 中断未产生 1: 中断已产生

7.4.2.6 定时器中断使能寄存器 (TMRIE)

表 7-13: 定时器中断使能寄存器 (TMRIE) 位段定义

TMRIE (Timer Interrupt Enable Register) Access: TIMER->TMRIE Offset: 0x0014 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							INT

表 7-14: 定时器中断使能寄存器 (TMRIE) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED_31_1	RO	0x0	保留
0	INT	RW	0x0	使能定时器中断 这个位段不影响 TMRIEVTf 以及 PWMSYNCl/ADCREQ 信号的产生。 0: 不发送中断信号给 CPU 1: 当 TMRIEVTf=1 时向 CPU 发送中断信号

7.4.2.7 定时器中断强制寄存器 (TMRIFRC)

表 7-15: 定时器中断强制寄存器 (TMRIFRC) 位段定义

TMRIFRC (Timer Interrupt Force Register) Access: TIMER->TMRIFRC Offset: 0x0018 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							INT

表 7-16: 定时器中断强制寄存器 (TMRIFRC) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED_31_1	RO	0x0	保留
0	INT	W1S	0x0	定时器软件强制中断控制位 0: 写 0 无效, 读总是读回 0 1: 写 1 强制置位 TMRIEVTf 标志位, 本位段自动清零。

7.4.2.8 定时器中断清除寄存器（TMRIC）

表 7-17：定时器中断清除寄存器（TMRIC）位段定义

TMRIC (Timer Interrupt Clear Register)							
Access: TIMER->TMRIC Offset: 0x001C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							INT

表 7-18：定时器中断清除寄存器（TMRIC）位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED_31_1	RO	0x0	保留
0	INT	W1C	0x0	定时器中断清除 0: 写 0 无效，读总是读回 0 1: 写 1 清除 TMRIEVTF 和 TMRIF 标志位 本位段会自动清零。

8 看门狗定时器（WDT）

8.1 概述

看门狗定时器（WDT）可以在系统运行失败（由于软件错误）的情况下，重新恢复系统的控制，以增加应用的可靠性。WDT 在计时到一个给定的超时数值时，可以产生系统复位或者一个中断。SPD1121 有 2 个 32 位看门狗定时器。

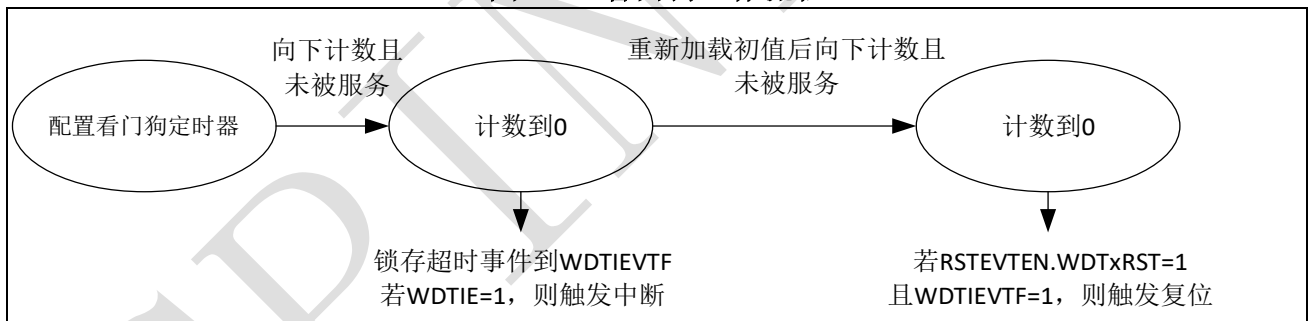
每一个看门狗都有如下特性：

- AHB 总线寄存器接口。
- 专属时钟，通过相应的 WDTxCLKCTL 寄存器配置。
- 32 位向下计数器，在可配置的计数窗口内允许刷新计数值。
- 当发生超时事件时，可配置产生复位或者中断。

8.2 功能描述

如图 8-1 所示，通过置位 WDTCTL.EN 位段来使能 WDT 后，计数器从 WDTLOAD 寄存器预设的超时值开始递减计数。每当计数到 0 时，将重新装载 WDTLOAD 值并再次向下计数。与此同时，超时事件会被锁存到 WDTIEVTF 寄存器中。如果 WDTCTL.INTEN 为 1，还会产生一个中断，且锁存中断标志到 WDTIF 寄存器。如果使能了复位功能且再次计数到 0 时 WDTIEVTF 标志未被清除，则会产生一个复位请求。

图 8-1：看门狗工作流程



当向 WDTLOAD 写入新的数值时，计数器将会立即从新的数值重新开始计数。当向 WDTIC 写入 1 时，WDTIEVTF 中锁存的超时事件标志和 WDTIF 中锁存的中断标志都会被清除。

通过向 WDTIC 写 1 来服务（刷新）看门狗定时器，此时计数值将被复位为 WDTLOAD。

可以通过 WDTCTL 寄存器，配置看门狗定时器在 CPU 处于 halted 或 lockup 时是否关闭。

出于安全方面的考虑，为了防止软件意外操作致使看门狗关闭，关键的控制寄存器均有写保护。用户在配置看门狗定时器前，需要先向 WDTREGKEY 寄存器写入 0x1ACCE551，才可以对其他寄存器进行写访问。配置完后，需要向 WDTREGKEY 寄存器写入任何其它数值以重新开启写保护。

注意： WDT0 的超时事件被用作 NMI 中断。

8.3 初始化流程

需要按照下面的操作流程来启动看门狗定时器：

- 配置 WDTREGKEY 寄存器，以使能其他寄存器的写访问；
- 配置 WDTLOAD 寄存器，设置超时值；
- 通过 WDTCTL.INTEN 配置是否触发中断；
- 通过 WDTCTL.RSTEN 配置是否触发复位；
- 通过向 WDTCTL.EN 位段写 1 启动定时器计数；
- 向 WDTREGKEY 写入 0x0（任何不为 0x1ACCE551 的值），禁止对其他寄存器的写访问。

8.4 寄存器

8.4.1 WDT 寄存器列表

表 8-1: WDT 模块基地址

外设模块	基地址
WDT0	0x4000 1000
WDT1	0x4000 2000

表 8-2: WDT 寄存器列表

寄存器	偏移地址	描述	复位值
WDTLOAD*	0x00	看门狗定时器超时加载寄存器	0x03D09000
WDTCNT	0x04	看门狗定时器计数器当前值寄存器	0x03D09000
WDTCTL*	0x08	看门狗定时器控制寄存器	0x0000001F
WDTIC	0x0C	看门狗定时器中断清除寄存器	0x00000000
WDTIEVTF	0x10	看门狗定时器中断事件标志寄存器	0x00000000
WDTIF	0x14	看门狗定时器中断标志寄存器	0x00000000
WDTREGKEY	0x18	看门狗定时器写使能寄存器	0x1ACCE551

- 注意:
1. 被*标记的寄存器只有在 WDTREGKEY=0x1ACCE551 时, 才可以改写。
 2. 如果 WDTREGKEY 的值不等于 0x1ACCE551, 那么所有寄存器的读取值的 LSB 位都为 1。

8.4.2 WDT 寄存器

8.4.2.1 看门狗定时器超时加载寄存器（WDTLOAD）

表 8-3：看门狗定时器超时加载寄存器（WDTLOAD）位段定义

WDTLOAD (Watchdog Timer Load Register)							
Access: WDT->WDTLOAD		Offset: 0x0000		Default: 0x03D09000			
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 8-4：看门狗定时器超时加载寄存器（WDTLOAD）位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x03D09000	超时加载值（最小有效值为 1） 计数器从该加载值递减。当本寄存器被写入新的数值时，计数器将会立即从新的数值重新开始向下计数。

8.4.2.2 看门狗定时器计数器当前值寄存器（WDTCNT）

表 8-5：看门狗定时器计数器当前值寄存器（WDTCNT）位段定义

WDTCNT (Watchdog Timer Current Value Register)							
Access: WDT->WDTCNT		Offset: 0x0004		Default: 0x03D09000			
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 8-6：看门狗定时器计数器当前值寄存器（WDTCNT）位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0x3D09000	递减计数器当前计数值

8.4.2.3 看门狗定时器控制寄存器 (WDTCTL)

表 8-7: 看门狗定时器控制寄存器 (WDTCTL) 位段定义

WDTCTL (Watchdog Timer Control Register) Access: WDT->WDTCTL Offset: 0x0008 Default: 0x0000001F							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED			EN	LOCKUPRUN	HALTEDRUN	RSTEN	INTEN

表 8-8: 看门狗定时器控制寄存器 (WDTCTL) 位段描述

位段	位段名	属性	复位值	描述
31:3	RESERVED_31_3	RO	0x0	保留
4	EN	RW	0x1	开启看门狗定时器 0: 关闭定时器并将其重置为 WDTLOAD 1: 使能定时器
3	LOCKUPRUN	RW	0x1	CPU 处于 lockup 状态时看门狗定时器是否运行 0: 关闭看门狗定时器并复位计数值为 WDTLOAD 1: 根据 EN 位段决定是否使能看门狗定时器
2	HALTEDRUN	RW	0x1	CPU 处于 halted 状态时看门狗定时器是否运行 0: 关闭看门狗定时器并复位计数值为 WDTLOAD 1: 根据 EN 位段决定是否使能看门狗定时器
1	RSTEN	RW	0x1	使能看门狗系统复位请求 0: 关闭系统复位请求 1: 使能系统服务请求
0	INTEN	RW	0x1	中断使能 本位段不影响 WDTIEVTF 的锁存 0: 不产生中断 1: 计数到 0 时产生中断

8.4.2.4 看门狗定时器中断清除寄存器（WDTIC）

表 8-9：看门狗定时器中断清除寄存器（WDTIC）位段定义

WDTIC (Watchdog Timer Interrupt Clear Register)							
Access: WDT->WDTIC Offset: 0x000C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							INT

表 8-10：看门狗定时器中断清除寄存器（WDTIC）位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED_31_1	RO	0x0	保留
0	INT	W1C	0x0	中断清除 0: 写入 0 无效，读总是 0 1: 写 1 清除 WDTIEVTF 和 WDTIF，本位段自动清零

8.4.2.5 看门狗定时器中断事件标志寄存器（WDTIEVTF）

表 8-11：看门狗定时器中断事件标志寄存器（WDTIEVTF）位段定义

WDTIEVTF (Watchdog Timer Interrupt Event Flag Register)							
Access: WDT->WDTIEVTF Offset: 0x0010 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							INT

表 8-12：看门狗定时器中断事件标志寄存器（WDTIEVTF）位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED_31_1	RO	0x0	保留
0	INT	RO	0x0	超时（计数到 0）事件标志 0: 超时事件未发生 1: 超时事件已发生

8.4.2.6 看门狗定时器中断标志寄存器 (WDTIF)

表 8-13: 看门狗定时器中断标志寄存器 (WDTIF) 位段定义

WDTIF (Watchdog Timer Interrupt Flag Register) Access: WDT->WDTIF Offset: 0x0014 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							INT

表 8-14: 看门狗定时器中断标志寄存器 (WDTIF) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED_31_1	RO	0x0	保留
0	INT	RO	0x0	中断标志 0: 中断未发生 1: 中断已发生

8.4.2.7 看门狗定时器写使能寄存器 (WDTREGKEY)

表 8-15: 看门狗定时器写使能寄存器 (WDTREGKEY) 位段定义

WDTREGKEY (Watchdog Register Write-Allow Key Register) Access: WDT->WDTREGKEY Offset: 0x0018 Default: 0x00000000							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

表 8-16: 看门狗定时器写使能寄存器 (WDTREGKEY) 位段描述

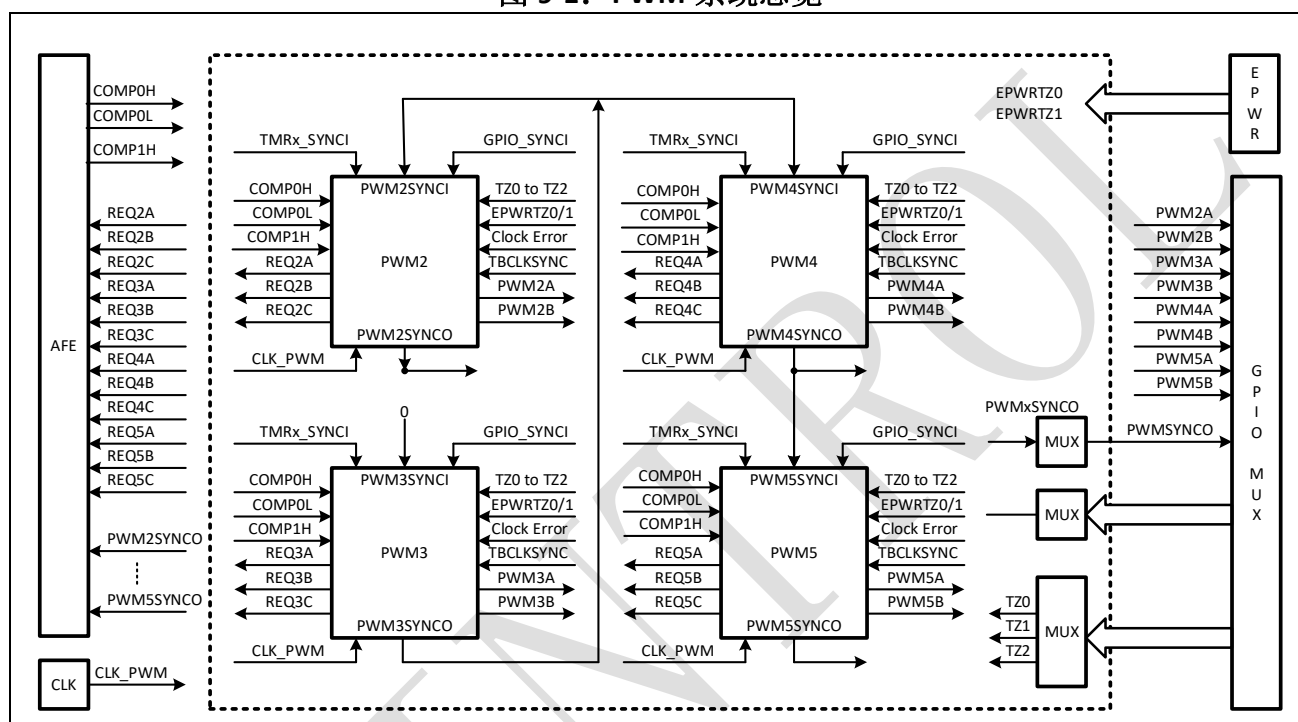
位段	位段名	属性	复位值	描述
31:0	KEY	RW	0x0	写入 0x1ACCE551 使能受保护的看门狗寄存器

9 PWM

9.1 PWM 概述

在诸如数控电机、开关电源、不间断电源等电力系统中，PWM 扮演了重要角色。SPD1121 提供了 4 个 PWM 模块，每个 PWM 模块有 PWMxA 和 PWMxB 两个输出。它们按照如图 9-1 所示方式串接在一起。

图 9-1: PWM 系统总览



每个 PWM 模块均支持如下特性：

- 周期可配的 16 位计时器；
- 软件直接强制改变 PWM 输出；
- 独立可配的相位控制；
- 周期性的相位同步；
- 上升沿延迟和下降沿延迟独立可配置的死区控制；
- 异常事件下可配的周期性或者一次性封锁；
- 封锁时可以配置 PWM 输出为高、低或高阻；
- 数字比较或者封锁信号输入可产生封锁事件；
- 所有事件可以用于产生 CPU 中断或者 ADC REQ（开始采样）信号。

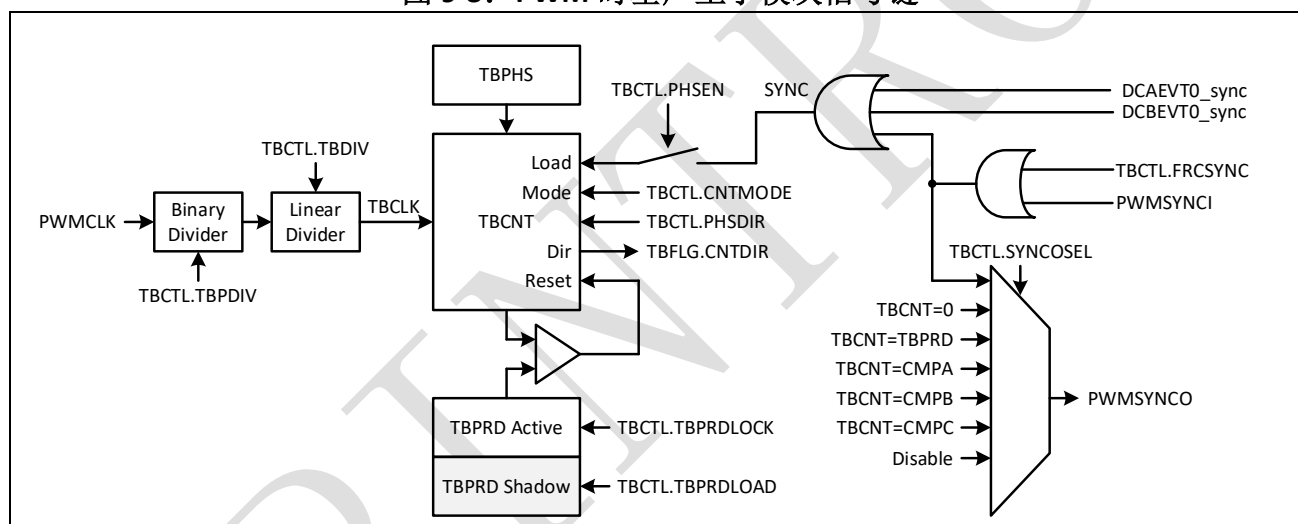
- 比较器结果可以产生数字比较事件
- ADC REQ 信号（PWMxREQA, PWMxREQB 和 PWMxREQC）
每个 PWM 有 3 个 ADC REQ 信号。触发 ADC REQ 信号的事件在事件触发（Event Trigger）子模块中配置。
- PWM 输出（PWMxA 和 PWMxB）
PWM 输出配置为与 GPIO 管脚复用，如 [IO 引脚功能通道定义](#) 所示。

9.2 时基产生 (TB) 子模块

时基产生（time-based）子模块负责 PWM 的时序控制，其关键信号如图 9-3 所示。该子模块有如下配置项：

- 设置时基计数时钟相对于 PWM 输入时钟的分频比；
- 设置时基计数器（TBCNT）模式：
 - 向上计数
 - 向下计数
 - 上下计数
- 设置时基计数器（TBCNT）周期；
- 设置相对于其他 PWM 的相位；
- 产生 TBCNT=TBPRD 和 TBCNT=0 事件；
- 设置同步触发事件以及事件发生时计数器采取的操作。

图 9-3: PWM 时基产生子模块信号链



PWM 计数波形周期由计数模式 (TBCTL.CNTMODE) 和周期 (TBPRD) 寄存器共同决定。

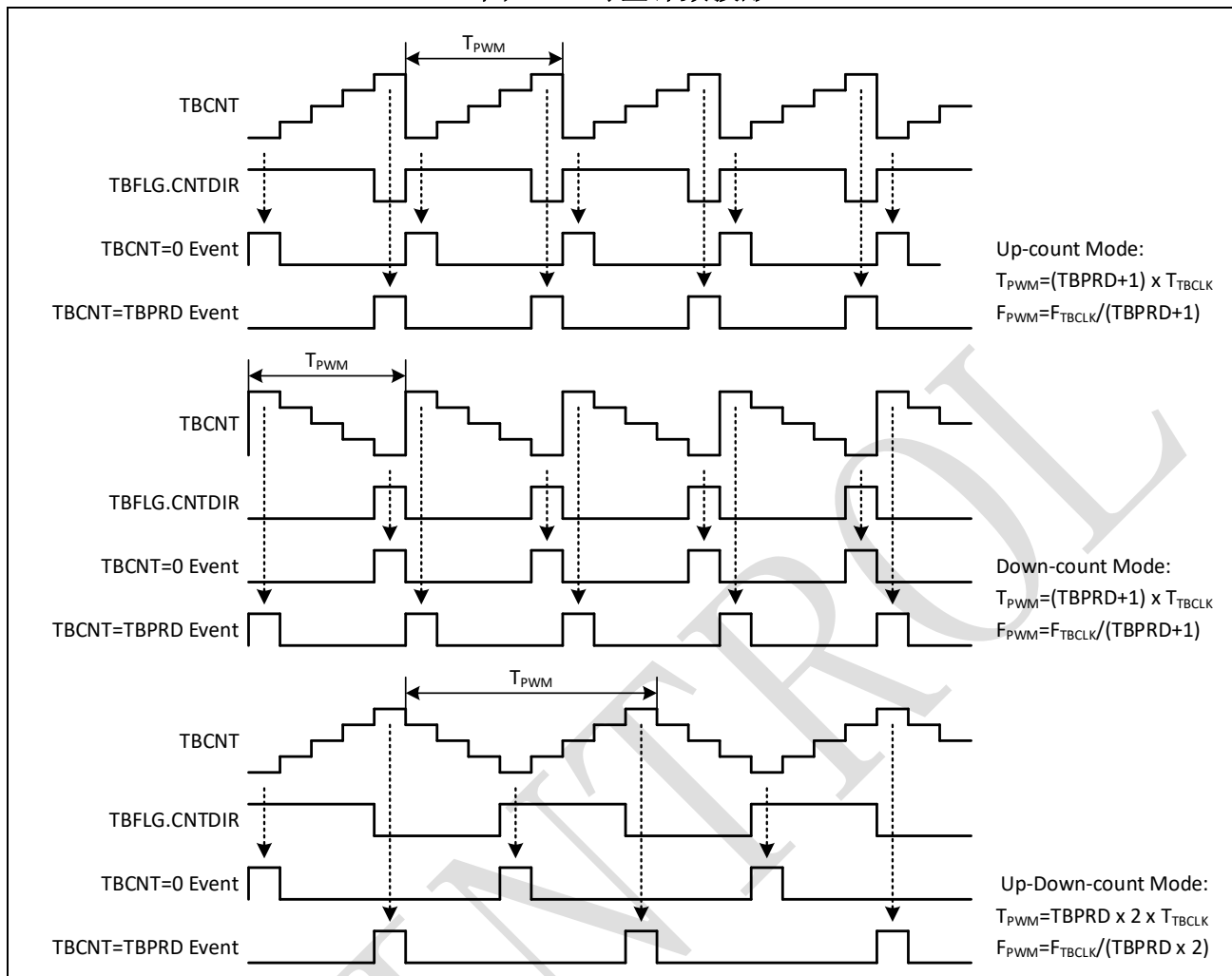
图 9-4 展示了向上计数、向下计数和上下计数三种情况下波形周期和 TBPRD 寄存器的关系。

向上计数模式时，时基计数器从 0 开始一直递增计数到 TBPRD，然后复位到 0 重新开始递增计数。从 TBFLG 寄存器 CNTDIR 位段读回的值在 TBCNT=TBPRD 时为 0，其余总是为 1。

向下计数模式时，时基计数器从 TBPRD 开始一直递减计数到 0，然后复位到 TBPRD 重新开始递减计数。从 TBFLG 寄存器 CNTDIR 位段读回的值在 TBCNT=0 时为 1，其余总是为 0。

上下计数模式时，时基计数器从 0 开始递增计数到 TBPRD，然后递减计数回 0，如此周而复始。从 TBFLG 寄存器 CNTDIR 位段读回的值，在从 0 到 (TBPRD-1) 递增计数时为 1，在从 TBPRD 到 1 递减计数时为 0。

图 9-4: 时基计数波形



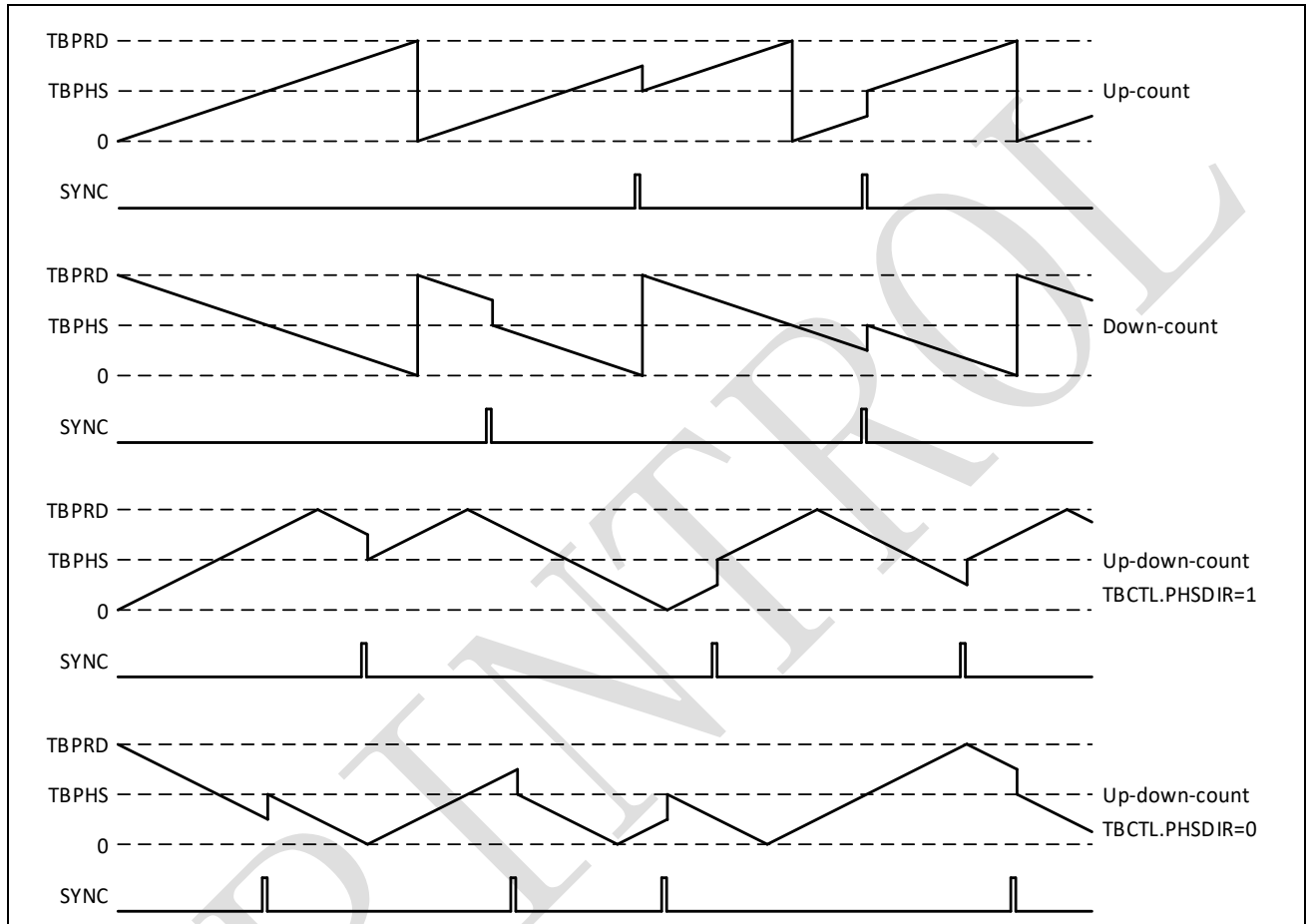
SPD1121 为 TBPRD 寄存器提供了影子寄存器。在该模式下，对 TBPRDS 写操作只会先更新影子寄存器的值而不会立即更新实际值，然后根据 TBCTL.TBPRDLOAD 位段的设置，当 TBCNT=0 或者同步事件发生时才从影子寄存器中更新。

相位同步功能使得 PWM 之间可以自动实现时基同步，可以通过设置 TBCTL.PHSEN 位段为 1 来开启这个功能。此时，一旦发生同步事件，TBCNT 会被更新为时基相位寄存器（TBPHS）的值。如图 9-5 所示，对于向上计数和向下计数模式，同步前后计数方向保持不变；对于上下计数方式，按照时基控制寄存器（TBCTL）的 PHSDIR 位段来决定同步后的计数方向。此外，如图 9-3 所示，同步事件可以是外部 PWMSYNCI 输入信号、本级 PWM 产生的数字比较同步信号 A 和 B、本级 PWM 软件强制同步。

当 TBCTL.PHSEN=0 时，PWM 忽略输入的同步事件信号，但这些信号仍然会被送往 PWMSYNCO 输出给后级 PWM 模块用于其同步。

- 注意：
1. 从同步信号输入到同步操作真正完成需要一个 PWM 时钟，因此当 TBCTL 寄存器的 TBPDIV 和 TBDIV 位段同时为 0（即不分频）时，会有一拍时基时钟的延时。用户在设置 TBPHS 寄存器时需要考虑这个因素。
 2. PWMCFG 模块中的 FRCSYNC 等全局同步控制提供了更加灵活便捷的跨 PWM 同步方式。

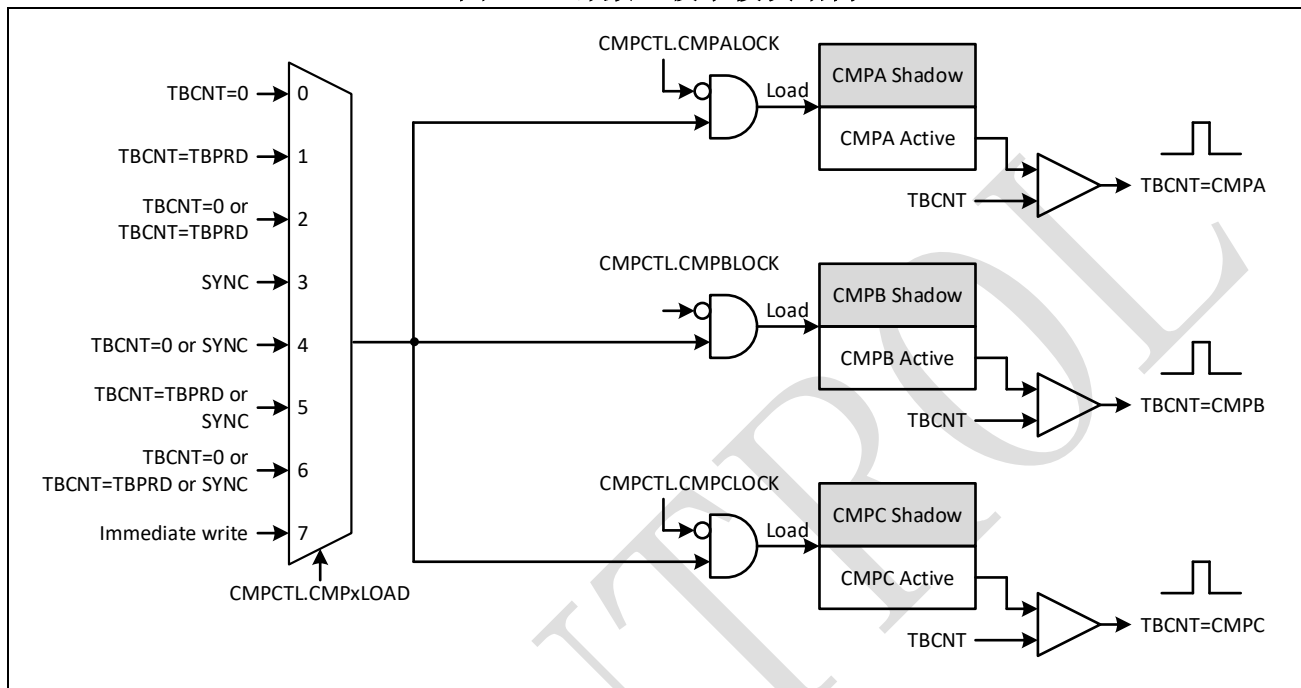
图 9-5：时基相位同步波形



9.3 计数比较（CC）子模块

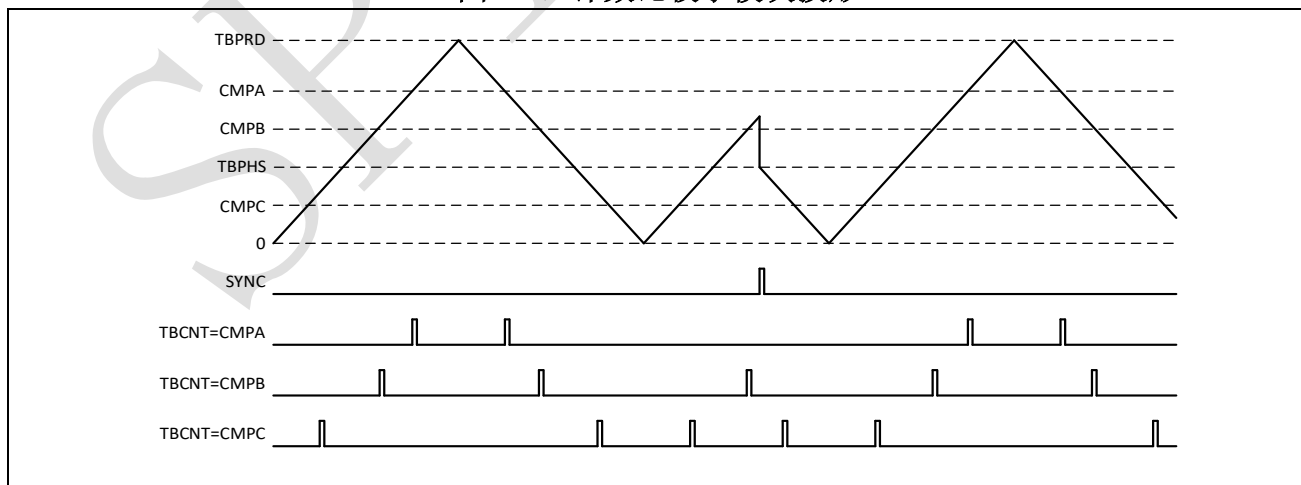
计数比较（counter-compare）子模块持续将 TBCNT 与计数比较寄存器 A（CMPA）、计数比较寄存器 B（CMPB）、计数比较寄存器 C（CMPC）的值进行比对。当 TBCNT 的值与这些寄存器的值相等时，产生相应的事件。

图 9-6：计数比较子模块结构



计数比较子模块的基本结构如图 9-6 所示。计数比较寄存器同样具备影子寄存功能，此时对计数比较寄存器的写操作会先更新影子寄存器的值，实际比较值会根据计数比较控制寄存器（CMPCTL）的 CMPxLOAD 位段定义的时间点更新。图 9-7 给出了计数比较子模块波形示例。

图 9-7：计数比较子模块波形



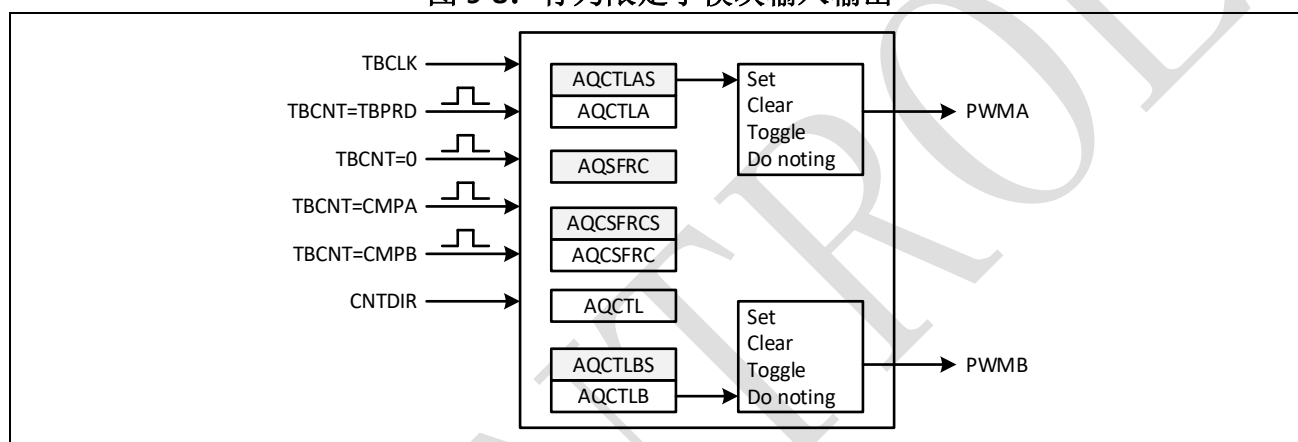
9.4 行为限定（AQ）子模块

行为限定（action-qualifier）子模块决定了哪些事件会转换为哪种行为并产生相应的 PWMA 和 PWMB 开关波形。

如图 9-8 所示，这些事件包括：

- TBCNT=TBPRD
- TBCNT=0
- TBCNT=CMPA
- TBCNT=CMPB
- 软件强制事件

图 9-8：行为限定子模块输入输出



行为限定模块管理这些事件同时发生时的响应优先级，并基于 AQCTLA 和 AQCTLB 寄存器的配置在对应事件发生时做出如下操作：

- 输出置高
- 输出置低
- 输出翻转
- 不操作

可以通过 AQCTL.AQCTLALOAD、AQCTL.AQCTLBLOAD 和 AQSFRC.CSFLOAD 寄存器位段来选择影子寄存模式或是直接更新模式。

表 9-1 到表 9-3 列举了不同计数模式下的行为限定事件优先级。

表 9-1：向上计数时的行为限定事件优先级

优先级	事件
1（最高）	软件强制控制
2	TBCNT=TBPRD（PRD）
3	TBCNT=CMPB（CBU）
4	TBCNT=CMPA（CAU）
5（最低）	TBCNT=0（ZRO）

表 9-2：向下计数时的行为限定事件优先级

优先级	事件
1（最高）	软件强制控制
2	TBCNT=0（ZRO）
3	TBCNT=CMPB（CBD）
4	TBCNT=CMPA（CAD）
5（最低）	TBCNT=TBPRD（PRD）

表 9-3：上下计数时的行为限定事件优先级

优先级	事件	
	从 0 向上计数到 TBPRD-1 时	从 TBPRD 向下计数到 1 时
1（最高）	软件强制控制	软件强制控制
2	TBCNT=CMPB（CBU）	TBCNT=CMPB（CBD）
3	TBCNT=CMPA（CAU）	TBCNT=CMPA（CAD）
4（最低）	TBCNT=0（ZRO）	TBCNT=TBPRD（PRD）

图 9-9 和图 9-10 示例了行为限定模块产生的对称和非对称 PWM 波形。

图 9-9：上下计数时的 PWM 双边沿对称波形

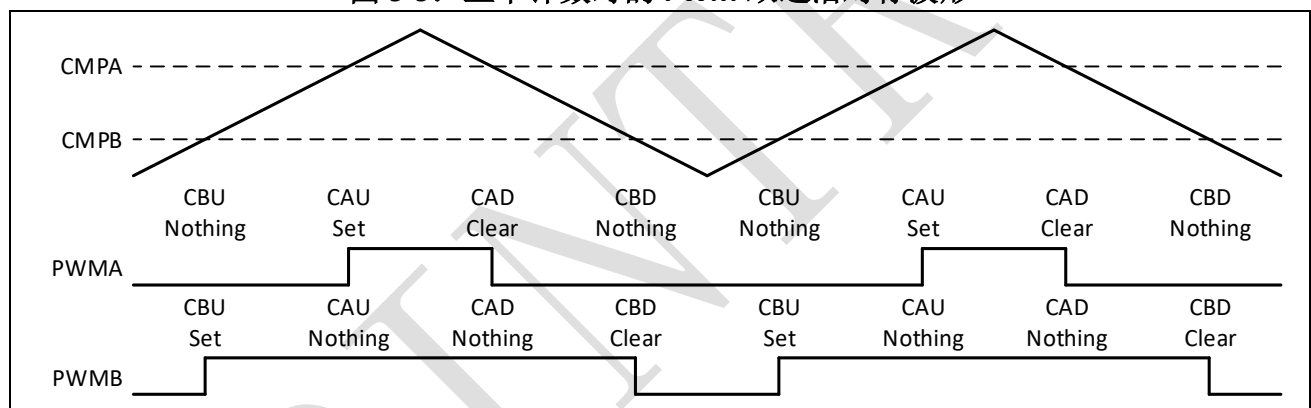
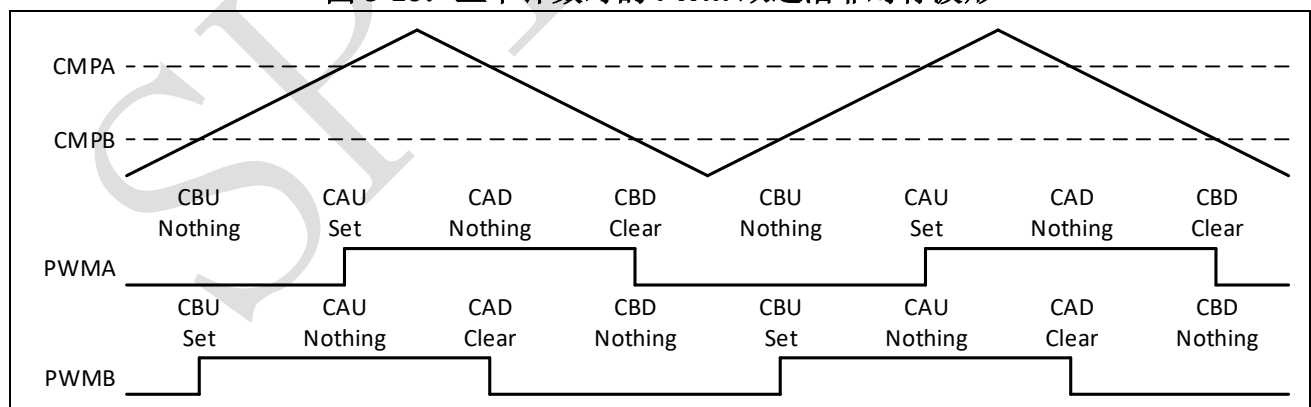


图 9-10：上下计数时的 PWM 双边沿非对称波形



9.5 死区控制（DB）子模块

死区控制子模块将行为限定模块输出的波形做进一步整形，包括对原始波形的上升沿和下降沿做独立的延时以及极性控制。该模块的关键功能包括：

- 基于来自行为限定模块的 PWMA/PWMB 输入产生有相对死区的 PWMA 和 PWMB 信号对；
- 对上升沿加可配置的延迟；
- 对下降沿加可配置的延迟；
- 支持半时钟周期延迟；
- 调整输出信号极性；
- 绕开死区控制信号整形，将行为限定模块输出的波形直接原样输出。

图 9-11：死区控制子模块结构

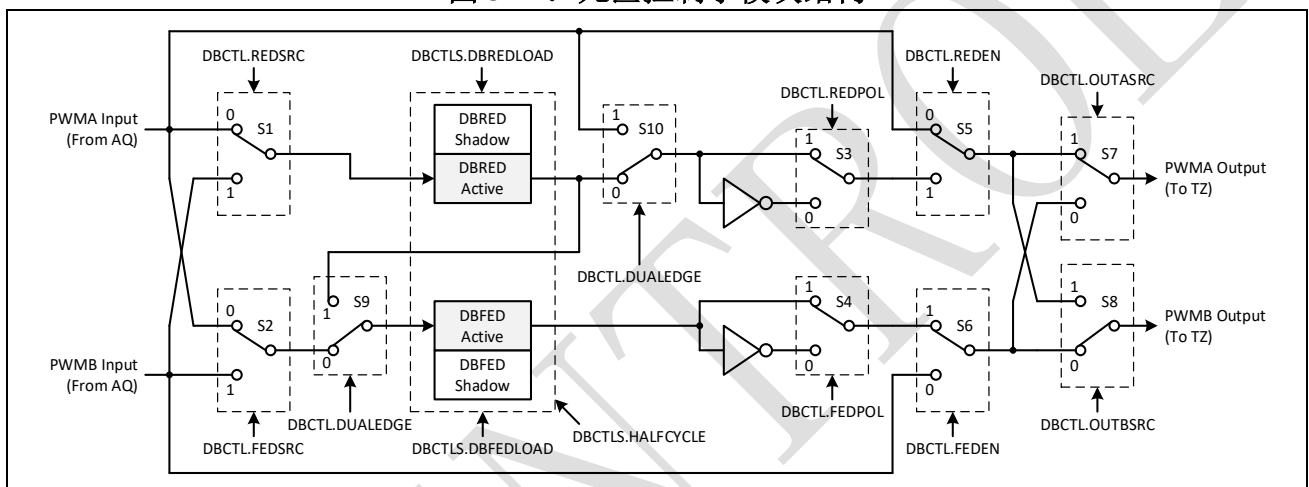


图 9-11 给出了死区控制模块的基本结构。死区控制寄存器（DBCTL）的 REDSRC 和 FEDSRC 位段设定了上升沿延迟通路和下降沿延迟通路的输入信号。随后 S3 到 S10 的各种开关组合可以产生灵活多变的波形。以 DBCTL.REDSRC=0 且 DBCTL.FEDSRC=0 为例，PWMA 被配置为两个延迟通路的输入，典型的死区控制工作模式如表 9-4 所示。

作为一种特殊情形，可以通过设置 DBCTL 寄存器的 DUALEDGE 位段实现双边沿延迟，即对同一路信号完成上升沿和下降沿的同时延迟。这个功能可以便于产生周期和占空比完全相同但存在指定相位差的两路波形。

表 9-4：典型的死区控制工作模式

模式	描述	REDPOL S3	FEDPOL S4	REDEN S5	FEDEN S6
1	同时将 PWMA 和 PWMB 原样输出	X	X	0	0
2	高有效的互补波形	1	0	1	1
3	低有效的互补波形	0	1	1	1
4	高有效的波形	0	0	1	1
5	低有效的波形	1	1	1	1
6	PWMA 输出 = PWMA 原始输入（无延迟） PWMB 输出 = PWMA 原始输入加下降沿延迟	0 或 1	0 或 1	0	1
7	PWMA 输出 = PWMA 原始输入加上上升沿延迟 PWMB 输出 = PWMA 原始输入（无延迟）	0 或 1	0 或 1	1	0

图 9-12 示例了死区控制的典型波形。

图 9-12: 死区控制典型波形

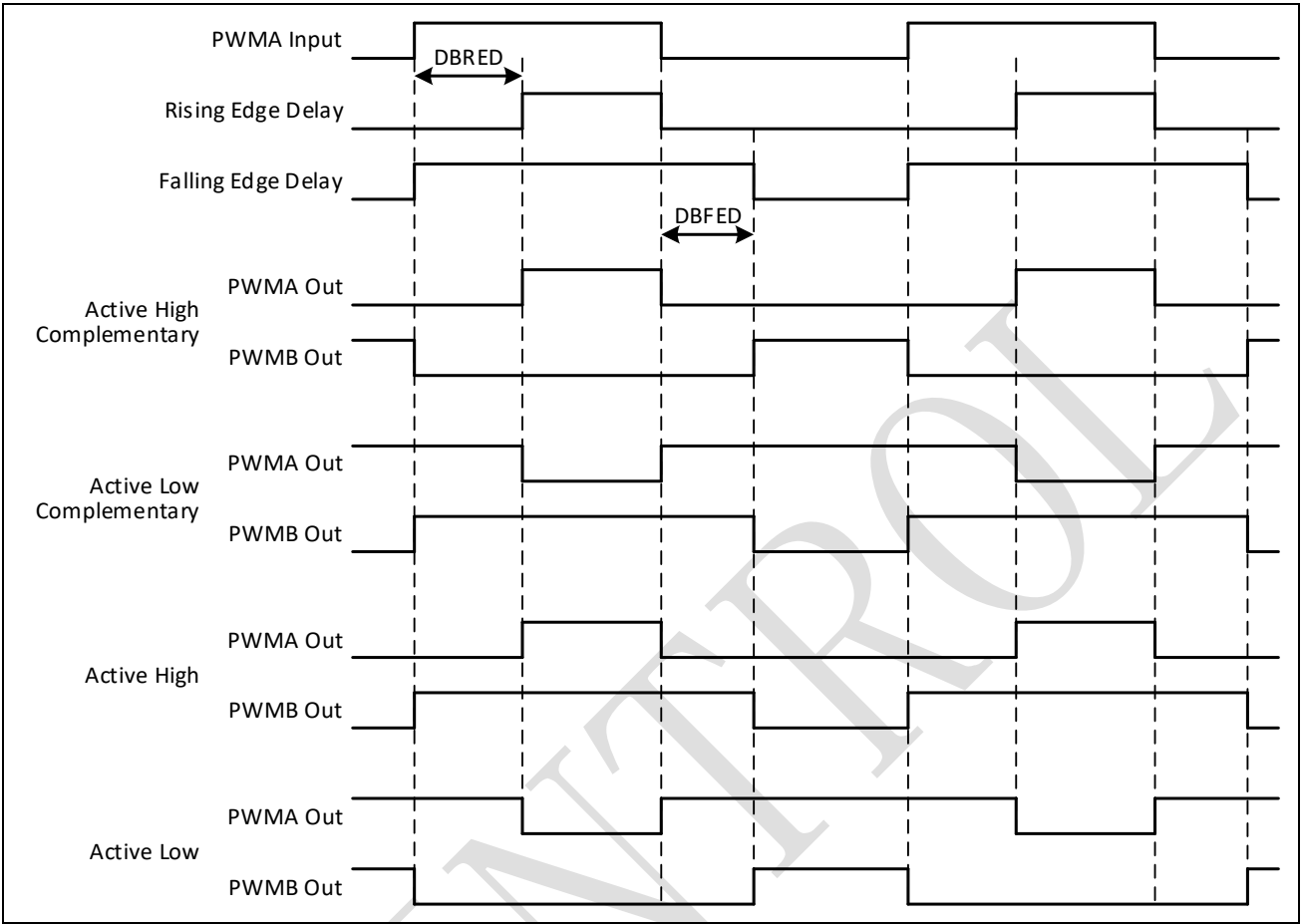
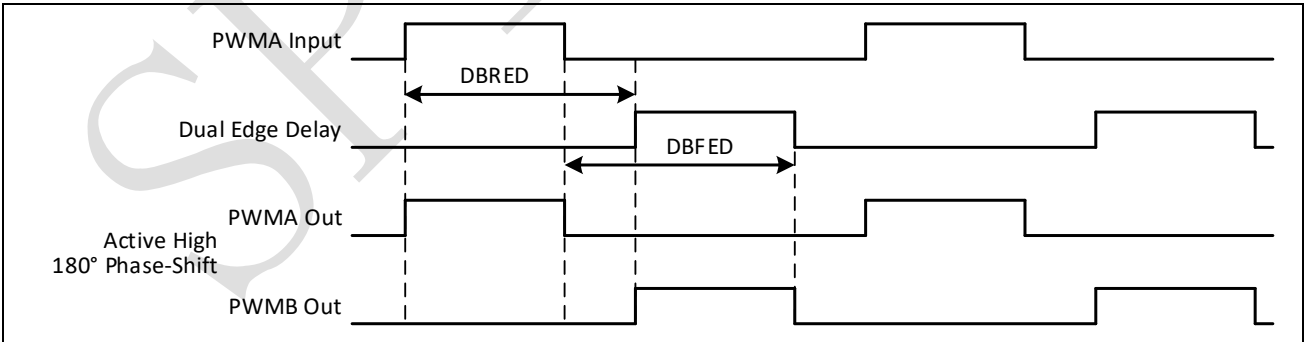


图 9-13 示例了通过使能双边沿延迟并设置 DBRED 和 DBFED 为波形周期的一半来产生 180° 相位差的两路波形。

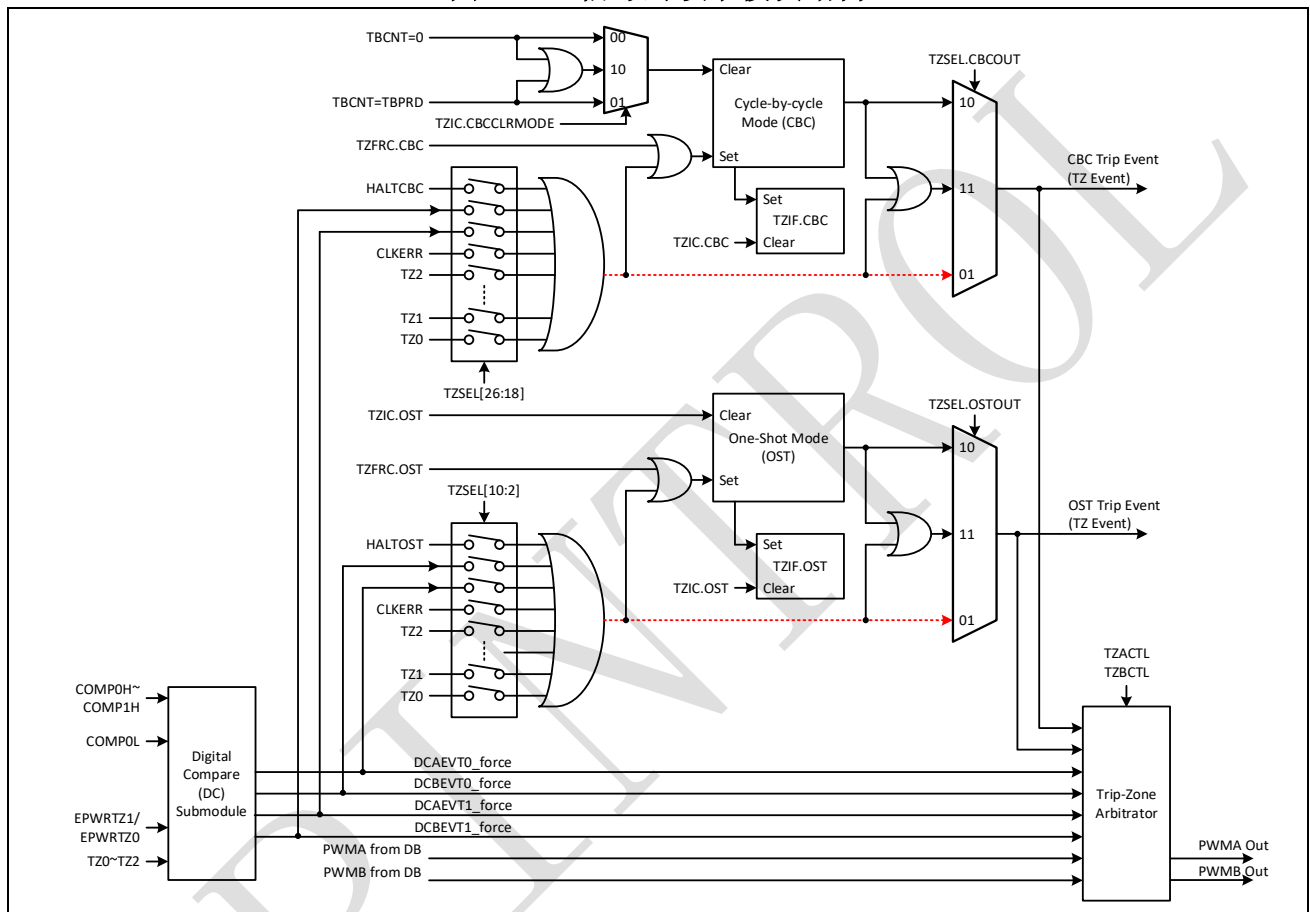
图 9-13: 双边沿延迟死区控制波形



9.6 信号封锁（TZ）子模块

信号封锁（trip-zone）子模块处理当外部错误事件或封锁条件发生时的 PWM 输出。如图 9-14 所示，输入事件可以是来自 GPIO 的 TZ 信号、来自 PLL 失锁或者 CLKDET 时钟检测模块的时钟错误信号、CPU 调试信号、以及数字比较（DC）子模块的输出。可以根据 PWMCFG 模块中的 TZ#SRCCTL 寄存器选择 3 个 TZ 信号（TZ0~TZ2）具体对应的 GPIO 以及信号的极性。此外支持软件触发的封锁，并且可以配置各种封锁事件触发 CPU 中断。

图 9-14：信号封锁子模块结构



封锁事件发生时，PWMA 和 PMWB 输出可以被强制为如下几种情形：

- 置高
- 置低
- 高阻
- 维持原样不变

PWMA 输出的事件动作由寄存器 TZACTL 控制，这些事件动作的优先级为 TZU > TZD > DCAEVT0U > DCAEVT0D > DCAEVT1U > DCAEVT1D。这意味着 TZACTL.DCAEVT0U 的配置只有在 TZACTL.TZAU 设置为“维持原样不变”时才有效。相同的原理适用于 PWMB 的封锁事件动作控制寄存器 TZBCTL。

SPD1121 的 PWM 同时支持一次性封锁（OST）和周期性（CBC）封锁。一次性封锁通常用于诸如严重过流或者电路短路等重大异常，周期性封锁常用于限流。

当一次性封锁发生时，PWMA 和 PWMB 的输出会立刻按照 TZACTL 和 TZBCTL 寄存器相应位段的设置改变。同时，TZIF.OST 标志位置 1。如果用户通过封锁中断使能寄存器（TZIE）开启了中断，则同时会产生 PWM_TZINT 中断。TZIF.OST 标志位只能通过往 TZIC.OST 位段写 1 手动清除。

当周期性封锁发生时，PWMA 和 PWMB 的输出会立刻按照 TZACTL 和 TZBCTL 寄存器相应位段的设置改变。同时，TZIF.CBC 标志位置 1。如果用户通过封锁中断使能寄存器（TZIE）开启了中断，则同时会产生 PWM_TZINT 中断。根据 TZIC.CBCCLRPUL 位段的设置，PWM 会在 TBCNT 计数到 0 或者 TBPRD 的时候检查封锁条件是否已不存在，若封锁条件已不存在则自动撤销对输出的强制。但是，TZIF.CBC 标志位依然需要通过往 TZIC.CBC 位段写 1 来手动清除。如果清除 TZIF.CBC 标志位的时候封锁条件依然存在，则该标志位会立即又被置位。

注意：图 9-14 中虚线所示的异步逻辑通路主要用于观测原始封锁事件信号（相应的 TZSEL.OSTOUT 或 TZSEL.CBCOUT 设定为 1）。实际应用中，必须选中锁存的通路来使能上述一次性和周期性封锁功能，即 TZSEL.OSTOUT 和 TZSEL.CBCOUT 必须设定为 2 或 3。

根据数字比较封锁事件选择寄存器（TZDCSEL）中 DCAH/DCAL 和 DCBH/DCBL 位段的设置，会产生数字比较事件 DCAEVT0/1 和 DCBEVT0/1。当数字比较事件发生时，PWMA 和 PWMB 的输出会立刻按照 TZACTL 和 TZBCTL 寄存器相应位段的设置改变。同时，TZIF.DCAEVT0/1 或 TZIF.DCBEVT0/1 标志位置 1。如果用户通过封锁中断使能寄存器（TZIE）开启了中断，则同时会产生 PWM_TZINT 中断。若数字比较事件消除，则自动撤销对输出的强制。但是，TZIF 寄存器中的相应标志位依然需要通过往 TZIC 相应位段写 1 来手动清除。如果清除标志位时数字比较事件依然存在，则标志位会立即又被置位。

图 9-15 示例了信号封锁中断产生逻辑。

图 9-15: 信号封锁中断产生

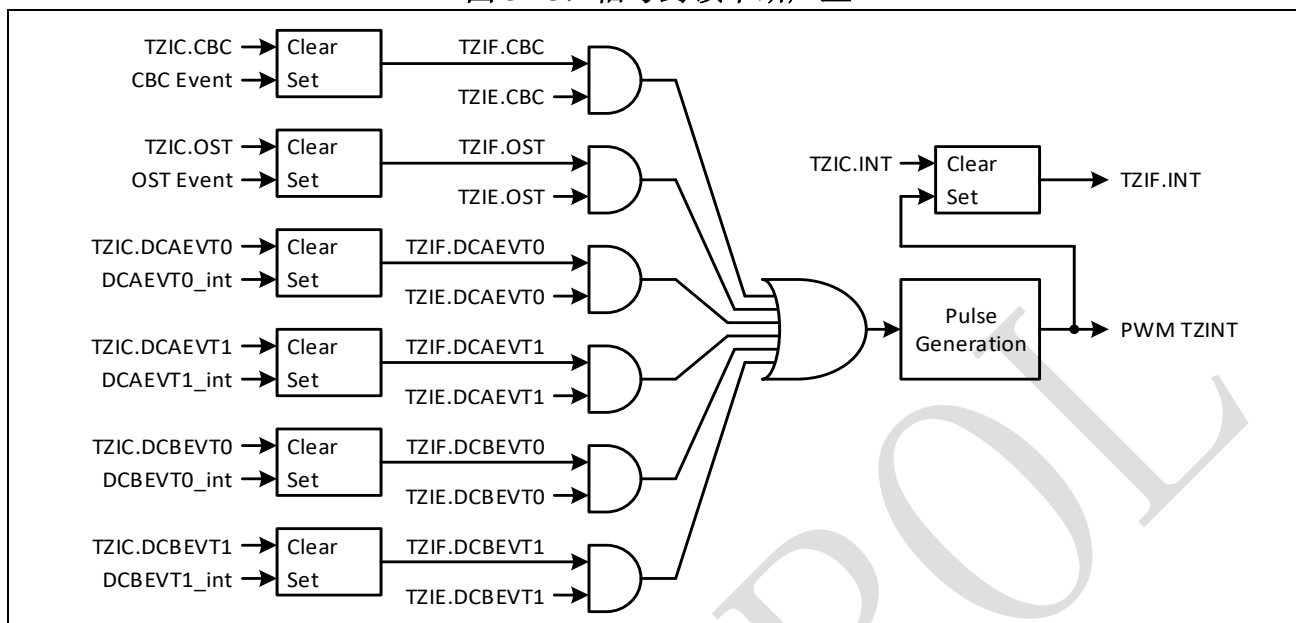


表 9-5 列举了影响 PWM 最终输出波形的各个事件的优先级。

表 9-5: 影响 PWM 最终输出波形的事件优先级

优先级	事件
1（最高）	封锁（Trip-zone）事件
2	通过 AQCSFRC 寄存器的软件持续强制
3	死区（Dead-Band）控制逻辑
4	通过 AQSFRC 寄存器的软件单次强制
5（最低）	行为限定（Action-Qualifier）控制逻辑

9.7 数字比较（DC）子模块

如图 9-16 所示，数字比较（digital compare）模块将来自 GPIO 的 TZ0~TZ2 事件以及比较器（COMP）的 COMPxL 和 COMPxH 事件作为输入，根据用户配置产生原始的 DCAL、DCAH、DCBL、DCBH 信号，再从这四个信号的各种组合中产生 DCAEVT0、DCAEVT1、DCBEVT0、DCBEVT1 事件。

图 9-16: 数字比较子模块结构

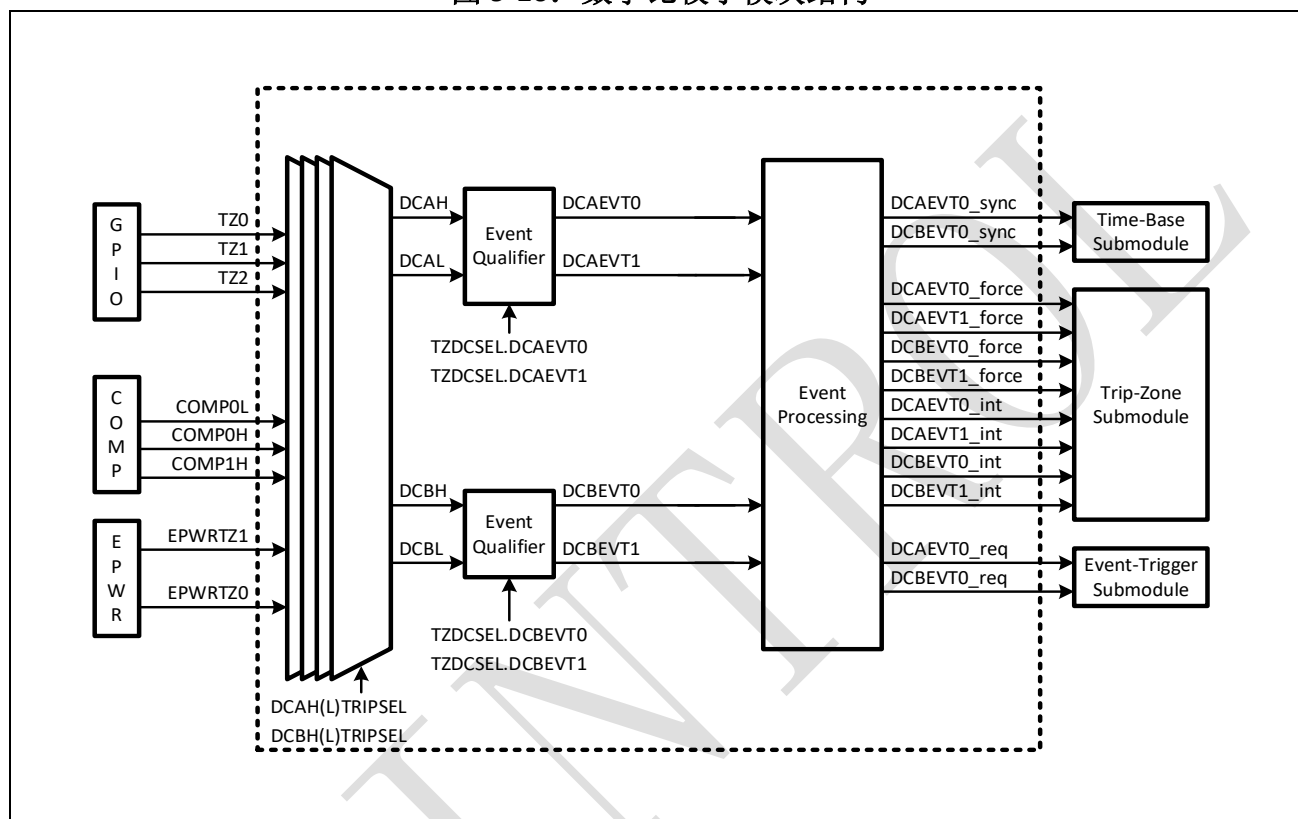
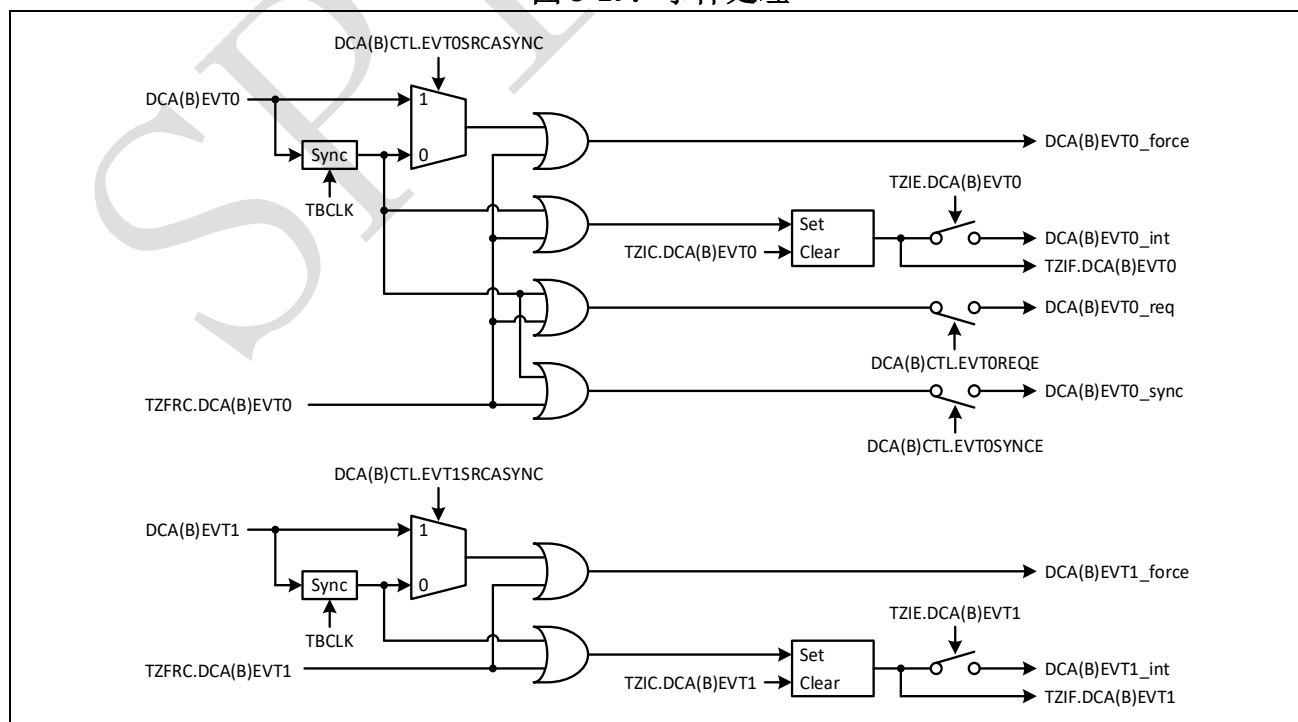


图 9-17: 事件处理



如图 9-17 所示，DCAEVT0、DCAEVT1、DCBEVT0、DCBEVT1 信号送至事件处理模块，根据 DCACTL 和 DCBCTL 寄存器的配置，可以进一步产生如下信号：

- 送往信号封锁子模块的强制信号
DCAEVT0_force、DCAEVT1_force、DCBEVT0_force、DCBEVT1_force
- 送往时基产生子模块的同步信号
DCAEVT0_sync、DCBEVT0_sync
- ADC 采样请求信号
DCAEVT0_req、DCBEVT0_req
- 中断
DCAEVT0_int、DCAEVT1_int、DCBEVT0_int、DCBEVT1_int

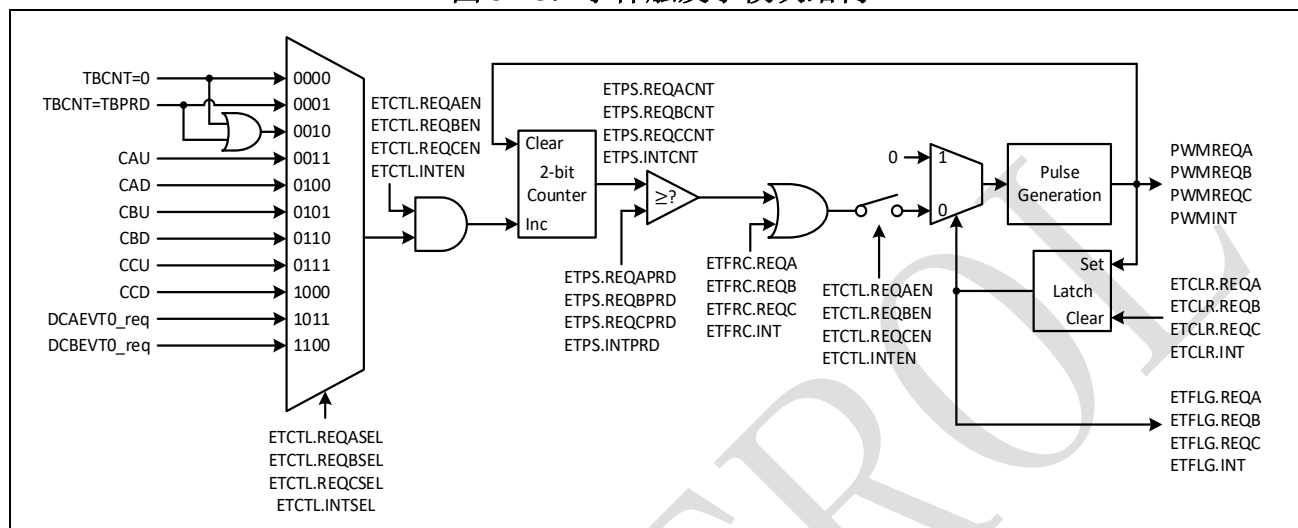
下面给出一个用 COMP0~COMP1 做相电流监测并产生数字比较封锁信号来控制 PWMxA 和 PWMxB 输出的例子。

- 通过 DCAHTRIPSEL 寄存器设定 DCAH 事件为 COMP0H、COMP0L、COMP1H 信号的逻辑或；
- 设定 TZDCSEL.DCAEVT0=4 来选择 DCAH=1 触发 DCAEVT0 事件；
- 通过 TZACTL 寄存器的 DCAEVT0U 和 DCAEVT0D 位段设定 DCAEVT0 发生时 PWMxA 的输出；
- 通过 TZBCTL 寄存器的 DCBEVT0U 和 DCBEVT0D 位段设定 DCBEVT0 发生时 PWMxB 的输出。

9.8 事件触发（ET）子模块

事件触发（event-trigger）子模块管理时基产生子模块、计数比较子模块和数字比较子模块产生的各种事件，并根据配置在对应事件发生时产生送至 CPU 的中断请求脉冲或送至 ADC 的采样起始脉冲。

图 9-18：事件触发子模块结构



事件触发子模块的具体结构如图 9-18 所示。为了便于软件调试，除前述硬件信号外，SPD1121 也支持由软件强制触发的 REQA、REQB、REQC 和 INT 脉冲。基于 2 位的计数器，脉冲产生可以有如下场景：

- 从不产生；
- 每次事件到来均产生脉冲；
- 每两次事件到来产生一个脉冲；
- 每三次事件到来产生一个脉冲。

9.9 寄存器

9.9.1 PWM 寄存器列表

表 9-6: PWM 模块基地址

外设模块	基地址
PWM2	0x4000 9200
PWM3	0x4000 9300
PWM4	0x4000 9400
PWM5	0x4000 9500

表 9-7: PWM 寄存器列表

寄存器	偏移地址	描述	复位值
SHADOWSTS	0x0000	影子状态寄存器	0x00000000
TBCTL*	0x0014	时基控制寄存器	0x00000006
TBPRDS	0x0018	时基周期影子寄存器	0x00000000
TBPRD	0x001C	时基周期活动寄存器	0x00000000
TBPHS	0x0020	时基相位寄存器	0x00000000
TBCNT	0x0024	时基计数器寄存器	0x00000000
TBFLG	0x0028	时基标志寄存器	0x00000000
TBCLR	0x002C	清除时基标志寄存器	0x00000000
CMPCTL*	0x0030	计数比较控制寄存器	0x00000000
CMPAS	0x0034	计数比较阈值 A 寄存器	0x00000000
CMPA	0x0038	计数比较阈值 A 有效值寄存器	0x00000000
CMPBS	0x003C	计数比较阈值 B 寄存器	0x00000000
CMPB	0x0040	计数比较阈值 B 有效值寄存器	0x00000000
CMPCS	0x0044	计数比较阈值 C 寄存器	0x00000000
CMPC	0x0048	计数比较阈值 C 有效值寄存器	0x00000000
AQCTL*	0x0054	行为限定控制寄存器	0x00000000
AQCTLAS	0x0058	行为限定 A 路输出控制寄存器	0x00000000
AQCTLA	0x005C	行为限定 A 路输出控制有效值寄存器	0x00000000
AQCTLBS	0x0060	行为限定 B 路输出控制寄存器	0x00000000
AQCTLB	0x0064	行为限定 B 路输出控制有效值寄存器	0x00000000
AQSFRC	0x0068	行为限定软件强制寄存器	0x00000000
AQCSFRCS	0x006C	行为限定软件持续强制寄存器	0x00000000
AQCSFRC	0x0070	行为限定软件持续强制有效值寄存器	0x00000000
DBCTL5	0x0074	死区产生控制寄存器	0x00000070
DBCTL	0x0078	死区产生控制有效值寄存器	0x00000070
DBREDS	0x007C	死区产生上升沿延迟量寄存器	0x00000000
DBRED	0x0080	死区产生上升沿延迟量有效值寄存器	0x00000000
DBFEDS	0x0084	死区产生下降沿延迟量寄存器	0x00000000
DBFED	0x0088	死区产生下降沿延迟量有效值寄存器	0x00000000
TZSEL*	0x008C	封锁事件选择寄存器	0x00030003
TZFLG	0x0090	封锁状态寄存器	0x00000000
TZCLR	0x0094	封锁状态清除寄存器	0x00000000

寄存器	偏移地址	描述	复位值
TZDCSEL*	0x0098	数字比较封锁事件寄存器	0x00000000
TZACTL	0x009C	封锁 A 路输出控制寄存器	0x00000000
TZBCTL	0x00A0	封锁 B 路输出控制寄存器	0x00000000
TZIF	0x00A4	封锁中断标志寄存器	0x00000000
TZIC	0x00A8	封锁中断清除寄存器	0x00000000
TZIE*	0x00AC	封锁中断使能寄存器	0x00000000
TZFRC	0x00B0	封锁软件强制寄存器	0x00000000
DCALTRIPSEL*	0x00B4	数字比较 DCAL 事件选择寄存器	0x00000000
DCAHTRIPSEL*	0x00B8	数字比较 DCAH 事件选择寄存器	0x00000000
DCBLTRIPSEL*	0x00BC	数字比较 DCBL 事件选择寄存器	0x00000000
DCBHTRIPSEL*	0x00C0	数字比较 DCBH 事件选择寄存器	0x00000000
DCACTL*	0x00C4	数字比较 A 控制寄存器	0x00000000
DCBCTL*	0x00C8	数字比较 B 控制寄存器	0x00000000
ETCTL*	0x00E8	事件触发控制寄存器	0x0007B16F
ETPS	0x00EC	事件触发预分频寄存器	0x00000000
ETFLG	0x00F0	事件触发标志寄存器	0x00000000
ETCLR	0x00F4	事件触发清除寄存器	0x00000000
ETFRC	0x00F8	事件触发软件强制寄存器	0x00000000
PWMREGKEY	0x00FC	PWM 模块写使能寄存器	0x1ACCE551

注意： 由*标记的寄存器仅当 PWMREGKEY=0x1ACCE551 才可以改写。

9.9.2 PWM 寄存器

9.9.2.1 影子状态寄存器（SHADOWSTS）

表 9-8：影子状态寄存器（SHADOWSTS）位段定义

SHADOWSTS (Shadow Status Register)							
Access: PWM->SHADOWSTS Offset: 0x0000 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED					DBFEDS	DBREDS	DBCTL5
7	6	5	4	3	2	1	0
AQCSFRCS	AQCTLBS	AQCTLAS	RESERVED	CMPCS	CMPBS	CMPAS	TBPRDS

表 9-9：影子状态寄存器（SHADOWSTS）位段描述

位段	位段名	属性	复位值	描述
31:11	RESERVED_31_11	RO	0x0	保留
10	DBFEDS	RO	0x0	DBFEDS 寄存器影子状态 0: DBFED=DBFEDS, 即已经用影子值重置有效值 1: DBFEDS 已经写入新的影子值, 但尚未重置到有效值中
9	DBREDS	RO	0x0	DBREDS 寄存器影子状态 0: DBRED=DBREDS, 即已经用影子值重置有效值 1: DBREDS 已经写入新的影子值, 但尚未重置到有效值中
8	DBCTL5	RO	0x0	DBCTL5 寄存器影子状态 0: DBCTL=DBCTL5, 即已经用影子值重置有效值 1: DBCTL5 已经写入新的影子值, 但尚未重置到有效值中
7	AQCSFRCS	RO	0x0	AQCSFRCS 寄存器影子状态 0: AQCSFRC=AQCSFRCS, 即已经用影子值重置有效值 1: AQCSFRCS 已经写入新的影子值, 但尚未重置到有效值中
6	AQCTLBS	RO	0x0	AQCTLBS 寄存器影子状态 0: AQCTLB=AQCTLBS, 即已经用影子值重置有效值 1: AQCTLBS 已经写入新的影子值, 但尚未重置到有效值中

位段	位段名	属性	复位值	描述
5	AQCTLAS	RO	0x0	AQCTLAS 寄存器影子状态 0: AQCTLA=AQCTLAS, 即已经用影子值重置有效值 1: AQCTLAS 已经写入新的影子值, 但尚未重置到有效值中
4	RESERVED_4	RO	0x0	保留
3	CMPCS	RO	0x0	CMPCS 寄存器影子状态 0: CMPC=CMPCS, 即已经用影子值重置有效值 1: CMPCS 已经写入新的影子值, 但尚未重置到有效值中
2	CMPBS	RO	0x0	CMPBS 寄存器影子状态 0: CMPB=CMPBS, 即已经用影子值重置有效值 1: CMPBS 已经写入新的影子值, 但尚未重置到有效值中
1	CMPAS	RO	0x0	CMPAS 寄存器影子状态 0: CMPA=CMPAS, 即已经用影子值重置有效值 1: CMPAS 已经写入新的影子值, 但尚未重置到有效值中
0	TBPRDS	RO	0x0	TBPRDS 寄存器影子状态 0: TBPRD=TBPRDS, 即已经用影子值重置有效值 1: TBPRDS 已经写入新的影子值, 但尚未重置到有效值中

9.9.2.2 时基控制寄存器 (TBCTL)

表 9-10: 时基控制寄存器 (TBCTL) 位段定义

TBCTL (Time-Base Control Register) Access: PWM->TBCTL Offset: 0x0014 Default: 0x00000006							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED				HALTRUN		TBPDIV	
15	14	13	12	11	10	9	8
TBPDIV	TBDIV			FRCSYNC	SYNCOSEL		
7	6	5	4	3	2	1	0
TBPRDLOCK	TBPRDLOAD		PHSDIR	PHSEN	CNTMODE		CNTRUN

表 9-11: 时基控制寄存器 (TBCTL) 位段描述

位段	位段名	属性	复位值	描述
31:20	RESERVED_31_20	RO	0x0	保留
19:18	HALTRUN	RW	0x0	CPU 停止时 PWM 时基计数器的行为 调试或者 CPU 错误异常都有可能导致 CPU 停止 0: 完成下一拍计数后停止 1: 完成整个计数周期后停止 (向上计数模式为 TBCNT=TBPRD, 其余模式计数至 TBCNT=0) 2: 保持正常计数
17:15	TBDIV	RW	0x0	时基计数时钟 TBCLK 相对于 PWM 时钟的二进制预分频, 总的分频比为 $2^{TBDIVBIN} \times (TBDIVLIN+1)$ 0: 1 分频 (复位默认值) 1: 2 分频 2: 4 分频 3: 8 分频 4: 16 分频 5: 32 分频 6: 64 分频 7: 128 分频
14:12	TBDIV	RW	0x0	时基计数时钟 TBCLK 预分频后进一步分频比 总的分频比为 $2^{TBDIVBIN} \times (TBDIVLIN+1)$ 0: 1 分频 (复位默认值) 1: 2 分频 2: 3 分频 3: 4 分频 4: 5 分频 5: 6 分频 6: 7 分频 7: 8 分频

位段	位段名	属性	复位值	描述
11	FRCSYNC	W1S	0x0	软件强制同步 0: 写 0 无效, 读总是 0 1: 写 1 产生强制同步, 本位段自动清零, 且不受 PWMREGKEY 保护
10:8	SYNCOSEL	RW	0x0	同步输出选择 0: SYNCI 和软件强制的同步事件 1: TBCNT=0 事件 2: TBCNT=TBPRD 事件 3: TBCNT=CMPA 事件 4: TBCNT=CMPC 事件 5: TBCNT=CMPC 事件 7: 关闭 SYNCO 输出
7	TBPRDLOCK	RW	0x0	TBPRD 有效值锁定 0: TBPRDA 存储的有效值根据 TBCTL.TBPRDLOAD 位段指定的时机重置 1: TBPRDA 存储的有效值维持不变
6:5	TBPRDLOAD	RW	0x0	TBPRD 有效值重置方式 TBCTL.TBPRDLOCK=1 时本位段无效 0: TBCNT=0 时用 TBPRDS 重置 TBPRD 1: SYNC 事件时用 TBPRDS 重置 TBPRD 2: SYNC 事件或 TBCNT=0 时用 TBPRDS 重置 TBPRD 3: 写 TBPRDS 寄存器时同时更新 TBPRD
4	PHSDIR	RW	0x0	相位同步后的计数方向 本位段仅当 TBCNT 处于上下计数模式时有效 (CNTMODE=2) 0: 同步事件发生后向下计数 1: 同步事件发生后向上计数
3	PHSEN	RW	0x0	相位同步使能 0: 不用 TBPHS 寄存器更新 TBCNT 计数值 1: 同步事件 (SYNCI 输入、软件强制同步或数字比较同步事件) 发生时, 将 TBCNT 计数值更新为 TBPHS 寄存器的值
2:1	CNTMODE	RW	0x3	计数模式 0: 向下计数 1: 向上计数 2: 上下计数 3: 停止计数并维持原计数值 (复位默认项)

位段	位段名	属性	复位值	描述
0	CNTRUN	RW	0x0	计数开关控制 0: 计数关闭 1: 计数开始

9.9.2.3 时基周期寄存器 (TBPRDS)

表 9-12: 时基周期寄存器 (TBPRDS) 位段定义

TBPRDS (Time-Base Period Shadow Register) Access: PWM->TBPRDS Offset: 0x0018 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 9-13: 时基周期寄存器 (TBPRDS) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	VAL	RW	0x0	时基计数周期

9.9.2.4 时基周期有效值寄存器（TBPRD）

表 9-14：时基周期有效值寄存器（TBPRD）位段定义

TBPRD (Time-Base Period Active Register)							
Access: PWM->TBPRD Offset: 0x001C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 9-15：时基周期有效值寄存器（TBPRD）位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	VAL	RO	0x0	时基计数周期有效值 当 TBCTL.PRDLOAD=3，写 TBPRDS 寄存器直接同时改变本寄存器值； 否则，根据 TBCTL.PRDLOAD 指定的方式从 TBPRDS 寄存器重置本寄存器值。

9.9.2.5 时基相位寄存器 (TBPHS)

表 9-16: 时基相位寄存器 (TBPHS) 位段定义

TBPHS (Time-Base Phase Register)							
Access: PWM->TBPHS Offset: 0x0020 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 9-17: 时基相位寄存器 (TBPHS) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	VAL	RW	0x0	同步事件发生时时基计数器待更新的相位 TBCTL.PHSEN=0: 忽略同步事件, TBCNT 计数 值维持不变。 TBCTL.PHSEN=1: 同步事件发生时更新 TBCNT 为本寄存器值。

9.9.2.6 时基计数值寄存器 (TBCNT)

表 9-18: 时基计数值寄存器 (TBCNT) 位段定义

TBCNT (Time-Base Counter Register)							
Access: PWM->TBCNT Offset: 0x0024 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 9-19: 时基计数值寄存器 (TBCNT) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	VAL	RW	0x0	时基计数器当前值

9.9.2.7 时基状态寄存器 (TBFLG)

表 9-20: 时基状态寄存器 (TBFLG) 位段定义

TBFLG (Time-Base Flag Register)							
Access: PWM->TBFLG Offset: 0x0028 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							CNTDIR
7	6	5	4	3	2	1	0
RESERVED	CNTCMPC	CNTCMPB	CNTCMPA	CNTPRD	CNTZRO	CNTMAX	SYNCI

表 9-21: 时基状态寄存器 (TBFLG) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8	CNTDIR	RO	0x0	时基计数器计数方向 0: 时基计数器正递减计数 1: 时基计数器正递增计数
7	RESERVED_7	RO	0x0	保留
6	CNTCMPC	RO	0x0	时基计数至 CMPC 标志 0: TBCNT 从未计数至 CMPC 1: TBCNT 曾计数至 CMPC
5	CNTCMPB	RO	0x0	时基计数至 CMPB 标志 0: TBCNT 从未计数至 CMPB 1: TBCNT 曾计数至 CMPB
4	CNTCMPA	RO	0x0	时基计数至 CMPA 标志 0: TBCNT 从未计数至 CMPA 1: TBCNT 曾计数至 CMPA
3	CNTPRD	RO	0x0	时基计数至 TBPRD 标志 0: TBCNT 从未计数至 TBPRD 1: TBCNT 曾计数至 TBPRD
2	CNTZRO	RO	0x0	时基计数至 0 标志 0: TBCNT 从未计数至 0 1: TBCNT 曾计数至 0
1	CNTMAX	RO	0x0	时基计数至 0xFFFF 标志 0: TBCNT 从未计数至 0xFFFF 1: TBCNT 曾计数至 0xFFFF
0	SYNCI	RO	0x0	输入同步事件发生标志 0: 输入同步事件从未发生 1: 输入同步事件曾经发生

9.9.2.8 时基状态清除寄存器 (TBCLR)

表 9-22: 时基状态清除寄存器 (TBCLR) 位段定义

TBCLR (Time-Base Flag Clear Register) Access: PWM->TBCLR Offset: 0x002C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	CNTCMPC	CNTCMPB	CNTCMPA	CNTPRD	CNTZRO	CNTMAX	SYNCI

表 9-23: 时基状态清除寄存器 (TBCLR) 位段描述

位段	位段名	属性	复位值	描述
31:7	RESERVED_31_7	RO	0x0	保留
6	CNTCMPC	W1C	0x0	时基计数至 CMPC 状态清除 0: 写 0 无效, 读总是 0 1: 写 1 清除对应的标志, 本位段自动清零
5	CNTCMPB	W1C	0x0	时基计数至 CMPB 状态清除 0: 写 0 无效, 读总是 0 1: 写 1 清除对应的标志, 本位段自动清零
4	CNTCMPA	W1C	0x0	时基计数至 CMPA 状态清除 0: 写 0 无效, 读总是 0 1: 写 1 清除对应的标志, 本位段自动清零
3	CNTPRD	W1C	0x0	时基计数至 TBPRD 状态清除 0: 写 0 无效, 读总是 0 1: 写 1 清除对应的标志, 本位段自动清零
2	CNTZRO	W1C	0x0	时基计数至 0 状态清除 0: 写 0 无效, 读总是 0 1: 写 1 清除对应的标志, 本位段自动清零
1	CNTMAX	W1C	0x0	时基计数至 0xFFFF 状态清除 0: 写 0 无效, 读总是 0 1: 写 1 清除对应的标志, 本位段自动清零
0	SYNCI	W1C	0x0	同步输入事件标志清除 0: 写 0 无效, 读总是 0 1: 写 1 清除对应的标志, 本位段自动清零

9.9.2.9 计数比较控制寄存器 (CMPCTL)

表 9-24: 计数比较控制寄存器 (CMPCTL) 位段定义

CMPCTL (Counter-Compare Control Register) Access: PWM->CMPCTL Offset: 0x0030 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED				CMPCLK	CMPCLD		
7	6	5	4	3	2	1	0
CMPBLK	CMPBLD			CMPALK	CMPALD		

表 9-25: 计数比较控制寄存器 (CMPCTL) 位段描述

位段	位段名	属性	复位值	描述
31:12	RESERVED_31_12	RO	0x0	保留
11	CMPCLK	RW	0x0	CMPC 有效值锁定 0: CMPC 存储的有效值在 CMPCTL.CMPCLD 指定的时机用 CMPCS 的值重置 1: CMPC 存储的有效值维持不变
10:8	CMPCLD	RW	0x0	CMPC 有效值重置方式 CMPCTL.CMPCLK=1 时本位段无效 0: TBCNT=0 时用 CMPCS 重置 CMPC 1: TBCNT=TBPRD 时用 CMPCS 重置 CMPC 2: TBCNT=0 或 TBCNT=TBPRD 时用 CMPCS 重置 CMPC 3: SYNC 事件时用 CMPCS 重置 CMPC 4: SYNC 事件或 TBCNT=0 时用 CMPCS 重置 CMPC 5: SYNC 事件或 TBCNT=TBPRD 时用 CMPCS 重置 CMPC 6: SYNC 事件、TBCNT=0 或 TBCNT=TBPRD 时用 CMPCS 重置 CMPC 7: 写 CMPCS 寄存器时同时更新 CMPC
7	CMPBLK	RW	0x0	CMPB 有效值锁定 0: CMPB 存储的有效值在 CMPCTL.CMPBLD 指定的时机用 CMPBS 的值重置 1: CMPB 存储的有效值维持不变
6:4	CMPBLD	RW	0x0	CMPB 有效值重置方式 CMPCTL.CMPBLK=1 时本位段无效

位段	位段名	属性	复位值	描述
				0: TBCNT=0 时用 CMPBS 重置 CMPB 1: TBCNT=TBPRD 时用 CMPBS 重置 CMPB 2: TBCNT=0 或 TBCNT=TBPRD 时用 CMPBS 重置 CMPB 3: SYNC 事件时用 CMPBS 重置 CMPB 4: SYNC 事件或 TBCNT=0 时用 CMPBS 重置 CMPB 5: SYNC 事件或 TBCNT=TBPRD 时用 CMPBS 重置 CMPB 6: SYNC 事件、TBCNT=0 或 TBCNT=TBPRD 时用 CMPBS 重置 CMPB 7: 写 CMPBS 寄存器时同时更新 CMPB
3	CMPALOCK	RW	0x0	CMPA 有效值锁定 0: CMPA 存储的有效值在 CMPATL.CMPALOAD 指定的时机用 CMPAS 的值重置 1: CMPA 存储的有效值维持不变
2:0	CMPALOAD	RW	0x0	CMPA 有效值重置方式 CMPATL.CMPALOCK=1 时本位段无效 0: TBCNT=0 时用 CMPAS 重置 CMPA 1: TBCNT=TBPRD 时用 CMPAS 重置 CMPA 2: TBCNT=0 或 TBCNT=TBPRD 时用 CMPAS 重置 CMPA 3: SYNC 事件时用 CMPAS 重置 CMPA 4: SYNC 事件或 TBCNT=0 时用 CMPAS 重置 CMPA 5: SYNC 事件或 TBCNT=TBPRD 时用 CMPAS 重置 CMPA 6: SYNC 事件、TBCNT=0 或 TBCNT=TBPRD 时用 CMPAS 重置 CMPA 7: 写 CMPAS 寄存器时同时更新 CMPA

9.9.2.10 计数比较阈值 A 寄存器 (CMPAS)

表 9-26: 计数比较阈值 A 寄存器 (CMPAS) 位段定义

CMPAS (Counter-Compare A Threshold Shadow Register) Access: PWM->CMPAS Offset: 0x0034 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 9-27: 计数比较阈值 A 寄存器 (CMPAS) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	VAL	RW	0x0	计数比较阈值 A

9.9.2.11 计数比较阈值 A 有效值寄存器 (CMPA)

表 9-28: 计数比较阈值 A 有效值寄存器 (CMPA) 位段定义

CMPA (Counter-Compare A Threshold Active Register) Access: PWM->CMPA Offset: 0x0038 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 9-29: 计数比较阈值 A 有效值寄存器 (CMPA) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	VAL	RO	0x0	计数比较阈值 A 有效值 TBCNT 等于该值时, 产生 TBCNT=CMPP 事件

9.9.2.12 计数比较阈值 B 寄存器 (CMPBS)

表 9-30: 计数比较阈值 B 寄存器 (CMPBS) 位段定义

CMPBS (Counter-Compare B Threshold Shadow Register) Access: PWM->CMPBS Offset: 0x003C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 9-31: 计数比较阈值 B 寄存器 (CMPBS) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	VAL	RW	0x0	计数比较阈值 B

9.9.2.13 计数比较阈值 B 寄存器 (CMPB)

表 9-32: 计数比较阈值 B 寄存器 (CMPB) 位段定义

CMPB (Counter-Compare B Threshold Active Register) Access: PWM->CMPB Offset: 0x0040 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 9-33: 计数比较阈值 B 寄存器 (CMPB) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	VAL	RO	0x0	计数比较阈值 B 有效值 TBCNT 等于该值时, 产生 TBCNT=CMPB 事件

9.9.2.14 计数比较阈值 C 寄存器 (CMPCS)

表 9-34: 计数比较阈值 C 寄存器 (CMPCS) 位段定义

CMPCS (Counter-Compare C Threshold Shadow Register)							
Access: PWM->CMPCS Offset: 0x0044 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 9-35: 计数比较阈值 C 寄存器 (CMPCS) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	VAL	RW	0x0	计数比较阈值 C

9.9.2.15 计数比较阈值 C 有效值寄存器 (CMPC)

表 9-36: 计数比较阈值 C 有效值寄存器 (CMPC) 位段定义

CMPC (Counter-Compare C Threshold Active Register)							
Access: PWM->CMPC Offset: 0x0048 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 9-37: 计数比较阈值 C 有效值寄存器 (CMPC) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	VAL	RO	0x0	计数比较阈值 C 有效值 TBCNT 等于该值时, 产生 TBCNT=CMPC 事件

9.9.2.16 行为限定控制寄存器 (AQCTL)

表 9-38: 行为限定控制寄存器 (AQCTL) 位段定义

AQCTL (Action-Qualifier Control Register) Access: PWM->AQCTL Offset: 0x0054 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
AQCTLBLOCK	AQCTLBLOAD			AQCTLALOCK	AQCTLALOAD		

表 9-39: 行为限定控制寄存器 (AQCTL) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	AQCTLBLOCK	RW	0x0	AQCTLB 有效值锁定 0: AQCTLB 维持的有效值由 AQCTL.AQCTLBLOAD 指定的时机用 AQCTLBS 的值重置 1: AQCTLB 存储的有效值维持不变
6:4	AQCTLBLOAD	RW	0x0	AQCTLB 有效值重置方式 AQCTL.AQCTLBLOCK=1 本位段无效 0: TBCNT=0 时用 AQCTLBS 重置 AQCTLB 1: TBCNT=TBPRD 时用 AQCTLBS 重置 AQCTLB 2: TBCNT=0 或 TBCNT=TBPRD 时用 AQCTLBS 重置 AQCTLB 3: SYNC 事件时用 AQCTLBS 重置 AQCTLB 4: SYNC 事件或 BCNT=0 时用 AQCTLBS 重置 AQCTLB 5: SYNC 事件或 BCNT=TBPRD 时用 AQCTLBS 重置 AQCTLB 6: SYNC 事件、TBCNT=0 或 TBCNT=TBPRD 时用 AQCTLBS 重置 AQCTLB 7: 写 AQCTLBS 寄存器时同时更新 AQCTLB
3	AQCTLALOCK	RW	0x0	AQCTLA 有效值锁定 0: AQCTLA 维持的有效值由 AQCTL.AQCTLALOAD 指定的时机用 AQCTLAS 的值重置 1: AQCTLA 存储的有效值维持不变
2:0	AQCTLALOAD	RW	0x0	AQCTLA 有效值重置方式 AQCTL.AQCTLALOCK=1 本位段无效

位段	位段名	属性	复位值	描述
				0: TBCNT=0 时用 AQCTLAS 重置 AQCTLA 1: TBCNT=TBPRD 时用 AQCTLAS 重置 AQCTLA 2: TBCNT=0 或 TBCNT=TBPRD 时用 AQCTLAS 重置 AQCTLA 3: SYNC 事件时用 AQCTLAS 重置 AQCTLA 4: SYNC 事件或 BCNT=0 时用 AQCTLAS 重置 AQCTLA 5: SYNC 事件或 BCNT=TBPRD 时用 AQCTLAS 重置 AQCTLA 6: SYNC 事件、TBCNT=0 或 TBCNT=TBPRD 时用 AQCTLAS 重置 AQCTLA 7: 写 AQCTLAS 寄存器时同时更新 AQCTLA

9.9.2.17 行为限定 A 路输出控制寄存器 (AQCTLAS)

表 9-40: 行为限定 A 路输出控制寄存器 (AQCTLAS) 位段定义

AQCTLAS (Action-Qualifier Output A Control Shadow Register)							
Access: PWM->AQCTLAS Offset: 0x0058 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED				CBD		CBU	
7	6	5	4	3	2	1	0
CAD		CAU		PRD		ZRO	

表 9-41: 行为限定 A 路输出控制寄存器 (AQCTLAS) 位段描述

位段	位段名	属性	复位值	描述
31:12	RESERVED_31_12	RO	0x0	保留
11:10	CBD	RW	0x0	TBCNT=CMPB 事件发生且 TBCNT 递减计数时的 A 路输出 0: 不变 (禁止操作) 1: 清零 (PWMxA 输出低) 2: 置高 (PWMxA 输出高) 3: 翻转 (PWMxA 当前输出翻转)
9:8	CBU	RW	0x0	TBCNT=CMPB 事件发生且 TBCNT 递增计数时的 A 路输出 0: 不变 (禁止操作) 1: 清零 (PWMxA 输出低) 2: 置高 (PWMxA 输出高) 3: 翻转 (PWMxA 当前输出翻转)
7:6	CAD	RW	0x0	TBCNT=CPMA 事件发生且 TBCNT 递减计数时的 A 路输出 0: 不变 (禁止操作) 1: 清零 (PWMxA 输出低) 2: 置高 (PWMxA 输出高) 3: 翻转 (PWMxA 当前输出翻转)
5:4	CAU	RW	0x0	TBCNT=CPMA 事件发生且 TBCNT 递增计数时的 A 路输出 0: 不变 (禁止操作) 1: 清零 (PWMxA 输出低) 2: 置高 (PWMxA 输出高) 3: 翻转 (PWMxA 当前输出翻转)
3:2	PRD	RW	0x0	TBCNT=TBPRD 事件时的 A 路输出

位段	位段名	属性	复位值	描述
				0: 不变（禁止操作） 1: 清零（PWMxA 输出低） 2: 置高（PWMxA 输出高） 3: 翻转（PWMxA 当前输出翻转）
1:0	ZRO	RW	0x0	TBCNT=0 事件时的 A 路输出 0: 不变（禁止操作） 1: 清零（PWMxA 输出低） 2: 置高（PWMxA 输出高） 3: 翻转（PWMxA 当前输出翻转）

9.9.2.18 行为限定 A 路输出控制有效值寄存器 (AQCTLA)

表 9-42: 行为限定 A 路输出控制有效值寄存器 (AQCTLA) 位段定义

AQCTLA (Action-Qualifier Output A Control Active Register) Access: PWM->AQCTLA Offset: 0x005C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED				CBD		CBU	
7	6	5	4	3	2	1	0
CAD		CAU		PRD		ZRO	

表 9-43: 行为限定 A 路输出控制有效值寄存器 (AQCTLA) 位段描述

位段	位段名	属性	复位值	描述
31:12	RESERVED_31_12	RO	0x0	保留
11:10	CBD	RO	0x0	TBCNT=CMPB 事件发生且 TBCNT 递减计数时的 A 路输出 0: 不变 (禁止操作) 1: 清零 (PWMxA 输出低) 2: 置高 (PWMxA 输出高) 3: 翻转 (PWMxA 当前输出翻转)
9:8	CBU	RO	0x0	TBCNT=CMPB 事件发生且 TBCNT 递增计数时的 A 路输出 0: 不变 (禁止操作) 1: 清零 (PWMxA 输出低) 2: 置高 (PWMxA 输出高) 3: 翻转 (PWMxA 当前输出翻转)
7:6	CAD	RO	0x0	TBCNT=CPMA 事件发生且 TBCNT 递减计数时的 A 路输出 0: 不变 (禁止操作) 1: 清零 (PWMxA 输出低) 2: 置高 (PWMxA 输出高) 3: 翻转 (PWMxA 当前输出翻转)
5:4	CAU	RO	0x0	TBCNT=CPMA 事件发生且 TBCNT 递增计数时的 A 路输出 0: 不变 (禁止操作) 1: 清零 (PWMxA 输出低) 2: 置高 (PWMxA 输出高) 3: 翻转 (PWMxA 当前输出翻转)
3:2	PRD	RO	0x0	TBCNT=TBPRD 事件时的 A 路输出

位段	位段名	属性	复位值	描述
				0: 不变（禁止操作） 1: 清零（PWMxA 输出低） 2: 置高（PWMxA 输出高） 3: 翻转（PWMxA 当前输出翻转）
1:0	ZRO	RO	0x0	TBCNT=0 事件时的 A 路输出 0: 不变（禁止操作） 1: 清零（PWMxA 输出低） 2: 置高（PWMxA 输出高） 3: 翻转（PWMxA 当前输出翻转）

9.9.2.19 行为限定 B 路输出控制寄存器 (AQCTLBS)

表 9-44: 行为限定 B 路输出控制寄存器 (AQCTLBS) 位段定义

AQCTLBS (Action-Qualifier Output B Control Shadow Register)							
Access: PWM->AQCTLBS Offset: 0x0060 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED				CBD		CBU	
7	6	5	4	3	2	1	0
CAD		CAU		PRD		ZRO	

表 9-45: 行为限定 B 路输出控制寄存器 (AQCTLBS) 位段描述

位段	位段名	属性	复位值	描述
31:12	RESERVED_31_12	RO	0x0	保留
11:10	CBD	RW	0x0	TBCNT=CMPB 事件发生且 TBCNT 递减计数时的 B 路输出 0: 不变 (禁止操作) 1: 清零 (PWMxB 输出低) 2: 置高 (PWMxB 输出高) 3: 翻转 (PWMxB 当前输出翻转)
9:8	CBU	RW	0x0	TBCNT=CMPB 事件发生且 TBCNT 递增计数时的 B 路输出 0: 不变 (禁止操作) 1: 清零 (PWMxB 输出低) 2: 置高 (PWMxB 输出高) 3: 翻转 (PWMxB 当前输出翻转)
7:6	CAD	RW	0x0	TBCNT=CMPA 事件发生且 TBCNT 递减计数时的 B 路输出 0: 不变 (禁止操作) 1: 清零 (PWMxB 输出低) 2: 置高 (PWMxB 输出高) 3: 翻转 (PWMxB 当前输出翻转)
5:4	CAU	RW	0x0	TBCNT=CMPA 事件发生且 TBCNT 递增计数时的 B 路输出 0: 不变 (禁止操作) 1: 清零 (PWMxB 输出低) 2: 置高 (PWMxB 输出高) 3: 翻转 (PWMxB 当前输出翻转)
3:2	PRD	RW	0x0	TBCNT=TBPRD 事件时的 B 路输出

位段	位段名	属性	复位值	描述
				0: 不变（禁止操作） 1: 清零（PWMxB 输出低） 2: 置高（PWMxB 输出高） 3: 翻转（PWMxB 当前输出翻转）
1:0	ZRO	RW	0x0	TBCNT=0 事件时的 B 路输出 0: 不变（禁止操作） 1: 清零（PWMxB 输出低） 2: 置高（PWMxB 输出高） 3: 翻转（PWMxB 当前输出翻转）

9.9.2.20 行为限定 B 路输出控制有效值寄存器 (AQCTLB)

表 9-46: 行为限定 B 路输出控制有效值寄存器 (AQCTLB) 位段定义

AQCTLB (Action-Qualifier Output B Control Active Register) Access: PWM->AQCTLB Offset: 0x0064 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED				CBD		CBU	
7	6	5	4	3	2	1	0
CAD		CAU		PRD		ZRO	

表 9-47: 行为限定 B 路输出控制有效值寄存器 (AQCTLB) 位段描述

位段	位段名	属性	复位值	描述
31:12	RESERVED_31_12	RO	0x0	保留
11:10	CBD	RO	0x0	TBCNT=CMPB 事件发生且 TBCNT 递减计数时的 B 路输出 0: 不变 (禁止操作) 1: 清零 (PWMxB 输出低) 2: 置高 (PWMxB 输出高) 3: 翻转 (PWMxB 当前输出翻转)
9:8	CBU	RO	0x0	TBCNT=CMPB 事件发生且 TBCNT 递增计数时的 B 路输出 0: 不变 (禁止操作) 1: 清零 (PWMxB 输出低) 2: 置高 (PWMxB 输出高) 3: 翻转 (PWMxB 当前输出翻转)
7:6	CAD	RO	0x0	TBCNT=CMPA 事件发生且 TBCNT 递减计数时的 B 路输出 0: 不变 (禁止操作) 1: 清零 (PWMxB 输出低) 2: 置高 (PWMxB 输出高) 3: 翻转 (PWMxB 当前输出翻转)
5:4	CAU	RO	0x0	TBCNT=CMPA 事件发生且 TBCNT 递增计数时的 B 路输出 0: 不变 (禁止操作) 1: 清零 (PWMxB 输出低) 2: 置高 (PWMxB 输出高) 3: 翻转 (PWMxB 当前输出翻转)
3:2	PRD	RO	0x0	TBCNT=TBPRD 事件时的 B 路输出

位段	位段名	属性	复位值	描述
				0: 不变（禁止操作） 1: 清零（PWMxB 输出低） 2: 置高（PWMxB 输出高） 3: 翻转（PWMxB 当前输出翻转）
1:0	ZRO	RO	0x0	TBCNT=0 事件时的 B 路输出 0: 不变（禁止操作） 1: 清零（PWMxB 输出低） 2: 置高（PWMxB 输出高） 3: 翻转（PWMxB 当前输出翻转）

9.9.2.21 行为限定软件强制寄存器（AQSFRC）

表 9-48：行为限定软件强制寄存器（AQSFRC）位段定义

AQSFRC (Action-Qualifier Software Force Register) Access: PWM->AQSFRC Offset: 0x0068 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
CSFLOAD		OTSFB	ACTSFB		OTSFA	ACTSFA	

表 9-49：行为限定软件强制寄存器（AQSFRC）位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7:6	CSFLOAD	RW	0x0	AQCSFRC 寄存器有效值重置方式 0: TBCNT=0 时用 AQCSFRCS 更新 AQCSFRC 1: TBCNT=TBPRD 时用 AQCSFRCS 更新 AQCSFRC 2: TBCNT=0 或 TBCNT=TBPRD 时用 AQCSFRCS 更新 AQCSFRC 3: 写 AQCSFRCS 寄存器时同时更新 AQCSFRC
5	OTSFB	W1S	0x0	B 路输出行为限定单次软件强制 0: 写 0 无效，总是读回 0 1: 写 1 产生单次行为限定软件强制事件，本位段自动清零
4:3	ACTSFB	RW	0x0	B 路行为限定单次软件强制时的输出 0: 不变（禁止操作） 1: 清零（PWMxB 输出低）

位段	位段名	属性	复位值	描述
				2: 置高 (PWMxB 输出高) 3: 翻转 (PWMxB 当前输出翻转)
2	OTSFA	W1S	0x0	A 路输出单次软件强制 0: 写 0 无效, 总是读回 0 1: 写 1 产生单次行为限定软件强制事件, 本位段自动清零
1:0	ACTSFA	RW	0x0	A 路行为限定单次软件强制时的输出 0: 不变 (禁止操作) 1: 清零 (PWMxA 输出低) 2: 置高 (PWMxA 输出高) 3: 翻转 (PWMxA 当前输出翻转)

9.9.2.22 行为限定软件持续强制寄存器 (AQCSFRCS)

表 9-50: 行为限定软件持续强制寄存器 (AQCSFRCS) 位段定义

AQCSFRCS (Action-Qualifier Continuous Software Force Shadow Register) Access: PWM->AQCSFRCS Offset: 0x006C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED				CSFB		CSFA	

表 9-51: 行为限定软件持续强制寄存器 (AQCSFRCS) 位段描述

位段	位段名	属性	复位值	描述
31:4	RESERVED_31_4	RO	0x0	保留
3:2	CSFB	RW	0x0	B 路软件持续强制 0: 不强制 1: 强制 B 路输出持续为低 2: 强制 B 路输出持续为高 3: 不强制
1:0	CSFA	RW	0x0	A 路软件持续强制 0: 不强制 1: 强制 B 路输出持续为低 2: 强制 B 路输出持续为高 3: 不强制

9.9.2.23 行为限定软件持续强制有效值寄存器（AQCSFRC）

表 9-52：行为限定软件持续强制有效值寄存器（AQCSFRC）位段定义

AQCSFRC (Action-Qualifier Continuous Software Force Active Register) Access: PWM->AQCSFRC Offset: 0x0070 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED				CSFB		CSFA	

表 9-53：行为限定软件持续强制有效值寄存器（AQCSFRC）位段描述

位段	位段名	属性	复位值	描述
31:4	RESERVED_31_4	RO	0x0	保留
3:2	CSFB	RO	0x0	B 路软件持续强制 0: 不强制 1: 强制 B 路输出持续为低 2: 强制 B 路输出持续为高 3: 不强制
1:0	CSFA	RO	0x0	A 路软件持续强制 0: 不强制 1: 强制 B 路输出持续为低 2: 强制 B 路输出持续为高 3: 不强制

9.9.2.24 死区产生控制寄存器 (DBCTL5)

表 9-54: 死区产生控制寄存器 (DBCTL5) 位段定义

DBCTL5 (Dead-Band Generator Control Shadow Register) Access: PWM->DBCTL5 Offset: 0x0074 Default: 0x00000070							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED					DBFEDLOCK	DBFEDLOAD	
15	14	13	12	11	10	9	8
DBREDLOCK	DBREDLOAD		DBCTLLOCK	DBCTLLOAD		HALFCYCLE	DUALEDGE
7	6	5	4	3	2	1	0
OUTBSRC	OUTASRC	FEDPOL	REDPOL	FEDSRC	REDSRC	FEDEN	REDEN

表 9-55: 死区产生控制寄存器 (DBCTL5) 位段描述

位段	位段名	属性	复位值	描述
31:19	RESERVED_31_19	RO	0x0	保留
18	DBFEDLOCK	RW	0x0	DBFED 有效值锁定 该字段由 PWMREGKEY 保护 0: DBFED 维持的有效值由 DBCTL.DBFEDLOAD 指定的时机用 DBFEDS 的值重置 1: DBFED 存储的有效值维持不变
17:16	DBFEDLOAD	RW	0x0	DBFED 有效值重置方式 DBCTL.DBFEDLOCK=1 时本位段无效 该字段由 PWMREGKEY 保护 0: TBCNT=0 时用 DBFEDS 重置 DBFED 1: TBCNT=TBPRD 时用 DBFEDS 重置 DBFED 2: TBCNT=0 或 TBCNT=TBPRD 时用 DBFEDS 重置 DBFED 3: 写 DBFEDS 寄存器时同时更新 DBFED
15	DBREDLOCK	RW	0x0	DBRED 有效值锁定 该字段由 PWMREGKEY 保护 0: DBRED 存储的有效值由 DBCTL.DBREDLOAD 指定的时机用 DBREDS 的值重置 1: DBRED 存储的有效值维持不变
14:13	DBREDLOAD	RW	0x0	DBRED 有效值重置方式 DBCTL.DBREDLOCK=1 时本位段无效 该字段由 PWMREGKEY 保护

位段	位段名	属性	复位值	描述
				0: TBCNT=0 时用 DBREDS 重置 DBRED 1: TBCNT=TBPRD 时用 DBREDS 重置 DBRED 2: TBCNT=0 或 TBCNT=TBPRD 时用 DBREDS 重置 DBRED 3: 写 DBREDS 寄存器时同时更新 DBRED
12	DBCTLLOCK	RW	0x0	DBCTL[9:0]有效值锁定 该字段由 PWMREGKEY 保护 0: DBCTL 存储的有效值由 DBCTL.DBCTLLOAD 指定的时机用 DBCTL[9:0]的值重置 1: DBCTLA 存储的有效值维持不变
11:10	DBCTLLOAD	RW	0x0	DBCTL 有效值重置方式 DBCTL.DBCTLLOCK=1 时本位段无效 该字段由 PWMREGKEY 保护 0: TBCNT=0 时用 DBCTL[9:0]重置 DBCTL 1: TBCNT=TBPRD 时用 DBCTL[9:0]重置 DBCTL 2: TBCNT=0 或 TBCNT=TBPRD 时用 DBCTL[9:0]重置 DBCTL 3: 写 DBCTL[9:0]时同时更新 DBCTL
9	HALFCYCLE	RW	0x0	半时钟周期模式使能 0: 全时钟周期模式，即死区控制计数时钟为时基计数时钟 TBCLK 1: 半时钟周期模式，即死区控制计数时钟为 2 倍时基计数时钟 TBCLK
8	DUALEDGE	RW	0x0	双边沿延迟模式 0: 上升沿延迟和下降沿延迟为两个独立通路 1: 对同一个信号输入先做上升沿延迟，再做下降沿延迟
7	OUTBSRC	RW	0x0	B 路输出源 0: 下降沿延迟通路作为 B 路输出 1: 上升沿延迟通路作为 B 路输出
6	OUTASRC	RW	0x1	A 路输出源 0: 下降沿延迟通路作为 A 路输出 1: 上升沿延迟通路作为 A 路输出
5	FEDPOL	RW	0x1	下降沿延迟通路输出极性 0: 下降沿延迟通路输出取反变为低电平有效 1: 下降沿延迟通路输出保持原始极性
4	REDPOL	RW	0x1	上升沿延迟通路输出极性

位段	位段名	属性	复位值	描述
				0: 上升沿延迟通路输出取反变为低电平有效 1: 上升沿延迟通路输出保持原始极性
3	FEDSRC	RW	0x0	下降沿延迟通路输入源 0: A 路输入作为下降沿延迟通路源 1: B 路输入作为下降沿延迟通路源
2	REDSRC	RW	0x0	上升沿延迟通路输入源 0: A 路输入作为上升沿延迟通路源 1: B 路输入作为上升沿延迟通路源
1	FEDEN	RW	0x0	下降沿延时通路使能 0: 绕过下降沿延时通路，将行为限定子模块输出的信号直接送至下一级 1: 启用下降沿延时通路
0	REDEN	RW	0x0	上升沿延时通路使能 0: 绕过上升沿延时通路，将行为限定子模块输出的信号直接送至下一级 1: 启用上升沿延时通路

9.9.2.25 死区产生控制有效值寄存器 (DBCTL)

表 9-56: 死区产生控制有效值寄存器 (DBCTL) 位段定义

DBCTL (Dead-Band Generator Control Active Register)							
Access: PWM->DBCTL Offset: 0x0078 Default: 0x00000070							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED						HALFCYCLE	DUALEDGE
7	6	5	4	3	2	1	0
OUTBSRC	OUTASRC	FEDPOL	REDPOL	FEDSRC	REDSRC	FEDEN	REDEN

表 9-57: 死区产生控制有效值寄存器 (DBCTL) 位段描述

位段	位段名	属性	复位值	描述
31:10	RESERVED_31_10	RO	0x0	保留
9	HALFCYCLE	RO	0x0	半时钟周期模式使能 0: 全时钟周期模式, 即死区控制计数时钟为时基计数时钟 TBCLK 1: 半时钟周期模式, 即死区控制计数时钟为 2 倍时基计数时钟 TBCLK
8	DUALEDGE	RO	0x0	双边沿延迟模式 0: 上升沿延迟和下降沿延迟为两个独立通路 1: 对同一个信号输入先做上升沿延迟, 再做下降沿延迟
7	OUTBSRC	RO	0x0	B 路输出源 0: 下降沿延迟通路作为 B 路输出 1: 上升沿延迟通路作为 B 路输出
6	OUTASRC	RO	0x1	A 路输出源 0: 下降沿延迟通路作为 A 路输出 1: 上升沿延迟通路作为 A 路输出
5	FEDPOL	RO	0x1	下降沿延迟通路输出极性 0: 下降沿延迟通路输出取反以得到低电平有效的信号 1: 下降沿延迟通路输出保持原始高电平有效
4	REDPOL	RO	0x1	上升沿延迟通路输出极性 0: 上升沿延迟通路输出取反以得到低电平有效的信号 1: 上升沿延迟通路输出保持原始高电平有效
3	FEDSRC	RO	0x0	下降沿延迟通路输入源

位段	位段名	属性	复位值	描述
				0: A 路输入作为下降沿延迟通路源 1: B 路输入作为下降沿延迟通路源
2	REDSRC	RO	0x0	上升沿延迟通路输入源 0: A 路输入作为上升沿延迟通路源 1: B 路输入作为上升沿延迟通路源
1	FEDEN	RO	0x0	下降沿延时通路使能 0: 绕开下降沿延时通路, 将行为限定子模块输出的信号直接送至下一级 1: 启用下降沿延时通路
0	REDEN	RO	0x0	上升沿延时通路使能 0: 绕开上升沿延时通路, 将行为限定子模块输出的信号直接送至下一级 1: 启用上升沿延时通路

9.9.2.26 死区产生上升沿延迟量寄存器（DBREDS）

表 9-58：死区产生上升沿延迟量寄存器（DBREDS）位段定义

DBREDS (Dead-Band Generator Rising Edge Delay Shadow Register)							
Access: PWM->DBREDS Offset: 0x007C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 9-59：死区产生上升沿延迟量寄存器（DBREDS）位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	VAL	RW	0x0	上升沿延迟量

9.9.2.27 死区产生上升沿延迟量有效值寄存器（DBRED）

表 9-60：死区产生上升沿延迟量有效值寄存器（DBRED）位段定义

DBRED (Dead-Band Generator Rising Edge Delay Active Register)							
Access: PWM->DBRED Offset: 0x0080 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 9-61：死区产生上升沿延迟量有效值寄存器（DBRED）位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	VAL	RO	0x0	上升沿延迟量有效值

9.9.2.28 死区产生下降沿延迟量寄存器 (DBFEDS)

表 9-62: 死区产生下降沿延迟量寄存器 (DBFEDS) 位段定义

DBFEDS (Dead-Band Generator Falling Edge Delay Shadow Register)							
Access: PWM->DBFEDS Offset: 0x0084 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 9-63: 死区产生下降沿延迟量寄存器 (DBFEDS) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	VAL	RW	0x0	下降沿延迟量

9.9.2.29 死区产生下降沿延迟量有效值寄存器 (DBFED)

表 9-64: 死区产生下降沿延迟量有效值寄存器 (DBFED) 位段定义

DBFED (Dead-Band Generator Falling Edge Delay Active Register)							
Access: PWM->DBFED Offset: 0x0088 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 9-65: 死区产生下降沿延迟量有效值寄存器 (DBFED) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	VAL	RO	0x0	下降沿延迟量有效值

9.9.2.30 封锁事件选择寄存器 (TZSEL)

表 9-66: 封锁事件选择寄存器 (TZSEL) 位段定义

TZSEL (Trip-Zone Event Select Register)							
Access: PWM->TZSEL Offset: 0x008C Default: 0x00030003							
31	30	29	28	27	26	25	24
RESERVED					HALTCBC	DCBEVT1	DCAEVT1
23	22	21	20	19	18	17	16
CLKERRCBC	RESERVED		TZ2CBC	TZ1CBC	TZ0CBC	CBCOUT	
15	14	13	12	11	10	9	8
RESERVED					HALTOST	DCBEVT0	DCAEVT0
7	6	5	4	3	2	1	0
CLKERR0ST	RESERVED		TZ2OST	TZ1OST	TZ0OST	OSTOUT	

表 9-67: 封锁事件选择寄存器 (TZSEL) 位段描述

位段	位段名	属性	复位值	描述
31:27	RESERVED_31_27	RO	0x0	保留
26	HALTCBC	RW	0x0	选中（逻辑或）调试或异常等原因引起的 CPU 停止（lockup 或者 halted）作为周期性封锁事件 0: 不选中 1: 选中
25	DCBEVT1	RW	0x0	选中（逻辑或）数字比较事件 DCBEVT1 作为周期性封锁事件 0: 不选中 1: 选中
24	DCAEVT1	RW	0x0	选中（逻辑或）数字比较事件 DCAEVT1 作为周期性封锁事件 0: 不选中 1: 选中
23	CLKERRCBC	RW	0x0	选中（逻辑或）时钟错误作为周期性封锁事件 0: 不选中 1: 选中
22:21	RESERVED_22_21	RO	0x0	保留
20	TZ2CBC	RW	0x0	选中（逻辑或）TZ2 作为周期性封锁事件 0: 不选中 1: 选中
19	TZ1CBC	RW	0x0	选中（逻辑或）TZ1 作为周期性封锁事件 0: 不选中 1: 选中
18	TZ0CBC	RW	0x0	选中（逻辑或）TZ0 作为周期性封锁事件

位段	位段名	属性	复位值	描述
				0: 不选中 1: 选中
17:16	CBCOUT	RW	0x3	周期性封锁信号输出选择 0: 不输出周期性封锁事件 1: 仅选择异步通路，即各选中事件原始逻辑或 2: 仅选择锁存通路 3: 同时选择异步通路和锁存通路（逻辑或）
15:11	RESERVED_15_11	RO	0x0	保留
10	HALTOST	RW	0x0	选中（逻辑或）调试或异常等原因引起的 CPU 停止（lockup 或者 halted）作为一次性封锁事件 0: 不选中 1: 选中
9	DCBEVT0	RW	0x0	选中（逻辑或）DCBEVT0 作为一次性封锁事件 0: 不选中 1: 选中
8	DCAEVT0	RW	0x0	选中（逻辑或）DCAEVT0 作为一次性封锁事件 0: 不选中 1: 选中
7	CLKERROST	RW	0x0	选中（逻辑或）时钟错误作为一次性封锁事件 0: 不选中 1: 选中
6:5	RESERVED_6_5	RO	0x0	保留
4	TZ2OST	RW	0x0	选中（逻辑或）TZ2 作为一次性封锁事件 0: 不选中 1: 选中
3	TZ1OST	RW	0x0	选中（逻辑或）TZ1 作为一次性封锁事件 0: 不选中 1: 选中
2	TZ0OST	RW	0x0	选中（逻辑或）TZ0 作为一次性封锁事件 0: 不选中 1: 选中
1:0	OSTOUT	RW	0x3	一次性封锁信号输出选择 0: 不输出一一次性封锁事件 1: 仅选择异步通路，即各选中事件原始逻辑或 2: 仅选择锁存通路 3: 同时选择异步通路和锁存通路（逻辑或）

9.9.2.31 封锁状态寄存器 (TZFLG)

表 9-68: 封锁状态寄存器 (TZFLG) 位段定义

TZFLG (Trip-Zone Flag Register)							
Access: PWM->TZFLG Offset: 0x0090 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED						HALTCBC	DCBEVT1
15	14	13	12	11	10	9	8
DCAEVT1	CLKERRCBC	RESERVED		TZ2CBC	TZ1CBC	TZ0CBC	HALTOST
7	6	5	4	3	2	1	0
DCBEVT0	DCAEVT0	CLKERROST	RESERVED		TZ2OST	TZ1OST	TZ0OST

表 9-69: 封锁状态寄存器 (TZFLG) 位段描述

位段	位段名	属性	复位值	描述
31:18	RESERVED_31_18	RO	0x0	保留
17	HALTCBC	RO	0x0	锁存的 CPU 停止（由调试或异常错误等原因引起）触发的周期性封锁标志 0: 该封锁事件未发生 1: 该封锁事件发生
16	DCBEVT1	RO	0x0	锁存的数字比较 DCBEVT1 触发的周期封锁标志 0: 该封锁事件未发生 1: 该封锁事件发生
15	DCAEVT1	RO	0x0	锁存的数字比较 DCAEVT1 触发的周期封锁标志 0: 该封锁事件未发生 1: 该封锁事件发生
14	CLKERRCBC	RO	0x0	锁存的时钟错误触发的周期封锁标志 0: 该封锁事件未发生 1: 该封锁事件发生
13:12	RESERVED_13_12	RO	0x0	保留
11	TZ2CBC	RO	0x0	锁存的 TZ2 触发的周期封锁标志 0: 该封锁事件未发生 1: 该封锁事件发生
10	TZ1CBC	RO	0x0	锁存的 TZ1 触发的周期封锁标志 0: 该封锁事件未发生 1: 该封锁事件发生
9	TZ0CBC	RO	0x0	锁存的 TZ0 触发的周期封锁标志 0: 该封锁事件未发生 1: 该封锁事件发生

位段	位段名	属性	复位值	描述
8	DBGOST	RO	0x0	锁存的 CPU 停止（由调试或异常错误等原因引起）触发的一次性封锁标志 0: 该封锁事件未发生 1: 该封锁事件发生
7	DCBEVT0	RO	0x0	锁存的数字比较 DCBEVT0 触发的一次封锁标志 0: 该封锁事件未发生 1: 该封锁事件发生
6	DCAEVT0	RO	0x0	锁存的数字比较 DCAEVT0 触发的一次封锁标志 0: 该封锁事件未发生 1: 该封锁事件发生
5	CLKERROST	RO	0x0	锁存的时钟错误触发的一次封锁标志 0: 该封锁事件未发生 1: 该封锁事件发生
4:3	RESERVED_4_3	RO	0x0	保留
2	TZ2OST	RO	0x0	锁存的 TZ2 触发的一次封锁标志 0: 该封锁事件未发生 1: 该封锁事件发生
1	TZ1OST	RO	0x0	锁存的 TZ1 触发的一次封锁标志 0: 该封锁事件未发生 1: 该封锁事件发生
0	TZ0OST	RO	0x0	锁存的 TZ0 触发的一次封锁标志 0: 该封锁事件未发生 1: 该封锁事件发生

9.9.2.32 封锁状态清除寄存器 (TZCLR)

表 9-70: 封锁状态清除寄存器 (TZCLR) 位段定义

TZCLR (Trip-Zone Flag Clear Register)							
Access: PWM->TZCLR Offset: 0x0094 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED						HALTCBC	DCBEVT1
15	14	13	12	11	10	9	8
DCAEVT1	CLKERRCBC	RESERVED		TZ2CBC	TZ1CBC	TZ0CBC	HALTOST
7	6	5	4	3	2	1	0
DCBEVT0	DCAEVT0	CLKERROST	RESERVED		TZ2OST	TZ1OST	TZ0OST

表 9-71: 封锁状态清除寄存器 (TZCLR) 位段描述

位段	位段名	属性	复位值	描述
31:18	RESERVED_31_18	RO	0x0	保留
17	DBGCBC	W1C	0x0	清除由 CPU 停止（因为调试或错误异常等原因引起）触发的周期封锁标志 0: 写 0 无效，读总是 0 1: 写 1 清除锁存的标志，本位段自动清零
16	DCBEVT1	W1C	0x0	清除由 DCBEVT1 触发的周期封锁标志 0: 写 0 无效，读总是 0 1: 写 1 清除锁存的标志，本位段自动清零
15	DCAEVT1	W1C	0x0	清除由 DCAEVT1 触发的周期封锁标志 0: 写 0 无效，读总是 0 1: 写 1 清除锁存的标志，本位段自动清零
14	CLKERRCBC	W1C	0x0	清除由时钟错误触发的周期封锁标志 0: 写 0 无效，读总是 0 1: 写 1 清除锁存的标志，本位段自动清零
13:12	RESERVED_13_12	RO	0x0	保留
11	TZ2CBC	W1C	0x0	清除由 TZ2 触发的周期封锁标志 0: 写 0 无效，读总是 0 1: 写 1 清除锁存的标志，本位段自动清零
10	TZ1CBC	W1C	0x0	清除由 TZ1 触发的周期封锁标志 0: 写 0 无效，读总是 0 1: 写 1 清除锁存的标志，本位段自动清零
9	TZ0CBC	W1C	0x0	清除由 TZ0 触发的周期封锁标志 0: 写 0 无效，读总是 0 1: 写 1 清除锁存的标志，本位段自动清零

位段	位段名	属性	复位值	描述
8	DBGOST	W1C	0x0	清除由 CPU 停止（因为调试或错误异常等原因引起）触发的一次封锁标志 0: 写 0 无效，读总是 0 1: 写 1 清除锁存的标志，本位段自动清零
7	DCBEVT0	W1C	0x0	清除由 DCBEVT0 触发的一次封锁标志 0: 写 0 无效，读总是 0 1: 写 1 清除锁存的标志，本位段自动清零
6	DCAEVT0	W1C	0x0	清除由 DCAEVT0 触发的一次封锁标志 0: 写 0 无效，读总是 0 1: 写 1 清除锁存的标志，本位段自动清零
5	CLKERROST	W1C	0x0	清除由时钟错误触发的一次封锁标志 0: 写 0 无效，读总是 0 1: 写 1 清除锁存的标志，本位段自动清零
4:3	RESERVED_4_3	RO	0x0	保留
2	TZ2OST	W1C	0x0	清除由 TZ2 触发的一次封锁标志 0: 写 0 无效，读总是 0 1: 写 1 清除锁存的标志，本位段自动清零
1	TZ1OST	W1C	0x0	清除由 TZ1 触发的一次封锁标志 0: 写 0 无效，读总是 0 1: 写 1 清除锁存的标志，本位段自动清零
0	TZ0OST	W1C	0x0	清除由 TZ0 触发的一次封锁标志 0: 写 0 无效，读总是 0 1: 写 1 清除锁存的标志，本位段自动清零

9.9.2.33 数字比较封锁事件选择寄存器 (TZDCSEL)

表 9-72: 数字比较封锁事件选择寄存器 (TZDCSEL) 位段定义

TZDCSEL (Trip-Zone Digital Compare Event Select Register)							
Access: PWM->TZDCSEL Offset: 0x0098 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED				DCBEVT1			DCBEVT0
7	6	5	4	3	2	1	0
DCBEVT0		DCAEVT1			DCAEVT0		

表 9-73: 数字比较封锁事件选择寄存器 (TZDCSEL) 位段描述

位段	位段名	属性	复位值	描述
31:12	RESERVED_31_12	RO	0x0	保留
11:9	DCBEVT1	RW	0x0	数字比较封锁事件 DCBEVT1 选择 0: 禁用事件 1: DCBL 为低 2: DCBL 为高 3: DCBH 为低 4: DCBH 为高 5: DCBL 为低且 DCBH 为高 6: DCBL 为高且 DCBH 为低 7: DCBL 为高且 DCBH 为高
8:6	DCBEVT0	RW	0x0	数字比较封锁事件 DCBEVT0 选择 0: 禁用事件 1: DCBL 为低 2: DCBL 为高 3: DCBH 为低 4: DCBH 为高 5: DCBL 为低且 DCBH 为高 6: DCBL 为高且 DCBH 为低 7: DCBL 为高且 DCBH 为高
5:3	DCAEVT1	RW	0x0	数字比较封锁事件 DCAEVT1 选择 0: 禁用事件 1: DCAL 为低 2: DCAL 为高 3: DCAH 为低 4: DCAH 为高 5: DCAL 为低且 DCAH 为高 6: DCAL 为高且 DCAH 为低 7: DCAL 为高且 DCAH 为高

位段	位段名	属性	复位值	描述
2:0	DCAEVT0	RW	0x0	数字比较封锁事件 DCAEVT0 选择 0: 禁用事件 1: DCAL 为低 2: DCAL 为高 3: DCAH 为低 4: DCAH 为高 5: DCAL 为低且 DCAH 为高 6: DCAL 为高且 DCAH 为低 7: DCAL 为高且 DCAH 为高

9.9.2.34 封锁 A 路输出控制寄存器 (TZACTL)

表 9-74: 封锁 A 路输出控制寄存器 (TZACTL) 位段定义

TZACTL (Trip-Zone Output A Control Register) Access: PWM->TZACTL Offset: 0x009C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED						DCAEVT1D	
15	14	13	12	11	10	9	8
DCAEVT1D	DCAEVT1U			DCAEVT0D			DCAEVT0U
7	6	5	4	3	2	1	0
DCAEVT0U		TZAD			TZAU		

表 9-75: 封锁 A 路输出控制寄存器 (TZACTL) 位段描述

位段	位段名	属性	复位值	描述
31:18	RESERVED_31_18	RO	0x0	保留
17:15	DCAEVT1D	RW	0x0	DCAEVT1 发生且 TBCNT 递减计数时的封锁输出 0: A 路输出高阻 1: A 路输出置低 2: A 路输出置高 3: A 路输出翻转 7: A 路输出维持原样不变
14:12	DCAEVT1U	RW	0x0	DCAEVT1 发生且 TBCNT 递增计数时的封锁输出 0: A 路输出高阻 1: A 路输出置低 2: A 路输出置高 3: A 路输出翻转 7: A 路输出维持原样不变
11:9	DCAEVT0D	RW	0x0	DCAEVT0 发生且 TBCNT 递减计数时的封锁输出

位段	位段名	属性	复位值	描述
				0: A 路输出高阻 1: A 路输出置低 2: A 路输出置高 3: A 路输出翻转 7: A 路输出维持原样不变
8:6	DCAEVT0U	RW	0x0	DCAEVT0 发生且 TBCNT 递增计数时的封锁输出 0: A 路输出高阻 1: A 路输出置低 2: A 路输出置高 3: A 路输出翻转 7: A 路输出维持原样不变
5:3	TZAD	RW	0x0	TZ 事件发生且 TBCNT 递减计数时的封锁输出 0: A 路输出高阻 1: A 路输出置低 2: A 路输出置高 3: A 路输出翻转 7: A 路输出维持原样不变
2:0	TZAU	RW	0x0	TZ 事件发生且 TBCNT 递增计数时的封锁输出 0: A 路输出高阻 1: A 路输出置低 2: A 路输出置高 3: A 路输出翻转 7: A 路输出维持原样不变

9.9.2.35 封锁 B 路输出控制寄存器 (TZBCTL)

表 9-76: 封锁 B 路输出控制寄存器 (TZBCTL) 位段定义

TZBCTL (Trip-Zone Ouput B Control Register)							
Access: PWM->TZBCTL Offset: 0x00A0 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED						DCBEVT1D	
15	14	13	12	11	10	9	8
DCBEVT1D	DCBEVT1U			DCBEVT0D			DCBEVT0U
7	6	5	4	3	2	1	0
DCBEVT0U		TZBD			TZBU		

表 9-77: 封锁 B 路输出控制寄存器 (TZBCTL) 位段描述

位段	位段名	属性	复位值	描述
31:18	RESERVED_31_18	RO	0x0	保留
17:15	DCBEVT1D	RW	0x0	DCBEVT1 发生且 TBCNT 递减计数时的封锁输出 0: B 路输出高阻 1: B 路输出置低 2: B 路输出置高 3: B 路输出翻转 7: B 路输出维持原样不变
14:12	DCBEVT1U	RW	0x0	DCBEVT1 发生且 TBCNT 递增计数时的封锁输出 0: B 路输出高阻 1: B 路输出置低 2: B 路输出置高 3: B 路输出翻转 7: B 路输出维持原样不变
11:9	DCBEVT0D	RW	0x0	DCBEVT0 发生且 TBCNT 递减计数时的封锁输出 0: B 路输出高阻 1: B 路输出置低 2: B 路输出置高 3: B 路输出翻转 7: B 路输出维持原样不变
8:6	DCBEVT0U	RW	0x0	DCBEVT0 发生且 TBCNT 递增计数时的封锁输出 0: B 路输出高阻 1: B 路输出置低 2: B 路输出置高 3: B 路输出翻转 7: B 路输出维持原样不变
5:3	TZBD	RW	0x0	TZ 事件发生且 TBCNT 递减计数时的封锁输出

位段	位段名	属性	复位值	描述
				0: B 路输出高阻 1: B 路输出置低 2: B 路输出置高 3: B 路输出翻转 7: B 路输出维持原样不变
2:0	TZBU	RW	0x0	TZ 事件发生且 TBCNT 递增计数时的封锁输出 0: B 路输出高阻 1: B 路输出置低 2: B 路输出置高 3: B 路输出翻转 7: B 路输出维持原样不变

9.9.2.36 封锁中断标志寄存器 (TZIF)

表 9-78: 封锁中断标志寄存器 (TZIF) 位段定义

TZIF (Trip-Zone Flag Register) Access: PWM->TZIF Offset: 0x00A4 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED		RAWDCBEVT1	RAWDCBEVT0	RAWDCAEVT1	RAWDCAEVT0	RAWCBC	RAWOST
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	INT	DCBEVT1	DCBEVT0	DCAEVT1	DCAEVT0	CBC	OST

表 9-79: 封锁中断标志寄存器 (TZIF) 位段描述

位段	位段名	属性	复位值	描述
31:22	RESERVED_31_22	RO	0x0	保留
21	RAWDCBEVT1	RO	0x0	数字比较 B 事件 1 原始标志 0: 事件未发生 1: 事件发生
20	RAWDCBEVT0	RO	0x0	数字比较 B 事件 0 原始标志 0: 事件未发生 1: 事件发生
19	RAWDCAEVT1	RO	0x0	数字比较事件 1 的原始标志 0: 事件未发生 1: 事件发生
18	RAWDCAEVT0	RO	0x0	数字比较事件 0 原始标志 0: 事件未发生 1: 事件发生

位段	位段名	属性	复位值	描述
17	RAWCBC	RO	0x0	周期性封锁事件原始标志 0: 事件未发生 1: 事件发生
16	RAWOST	RO	0x0	一次封锁事件标志 0: 一次封锁事件未发生 1: 一次封锁事件发生。该位段清零前新的一次封锁事件不会触发新的中断。
15:7	RESERVED_15_7	RO	0x0	保留
6	INT	RO	0x0	全局封锁中断标志 0: 封锁中断未发生 1: 封锁中断发生，该位段清零前封锁事件不会产生新的中断
5	DCBEVT1	RO	0x0	DCBEVT1 封锁事件标志 0: DCBEVT1 事件未发生 1: DCBEVT1 事件发生，该位段清零前新的 DCBEVT1 事件不会触发新的中断
4	DCBEVT0	RO	0x0	DCBEVT0 封锁事件标志 0: DCBEVT0 事件未发生 1: DCBEVT0 事件发生，该位段清零前新的 DCBEVT0 事件不会触发新的中断
3	DCAEVT1	RO	0x0	DCAEVT1 封锁事件标志 0: DCAEVT1 事件未发生 1: DCAEVT1 事件发生，该位段清零前新的 DCAEVT1 事件不会触发新的中断
2	DCAEVT0	RO	0x0	DCAEVT0 封锁事件标志 0: DCAEVT0 事件未发生 1: DCAEVT0 事件发生，该位段清零前新的 DCAEVT0 事件不会触发新的中断
1	CBC	RO	0x0	周期封锁事件标志 0: 周期封锁事件未发生 1: 周期封锁事件发生，该位段清零前新的周期封锁事件不会触发新的中断
0	OST	RO	0x0	一次封锁事件标志 0: 一次封锁事件未发生 1: 一次封锁事件发生，该位段清零前新的一次封锁事件不会触发新的中断

9.9.2.37 封锁中断清除寄存器 (TZIC)

表 9-80: 封锁中断清除寄存器 (TZIC) 位段定义

TZIC (Trip-Zone Clear Register)							
Access: PWM->TZIC Offset: 0x00A8 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							CBCCLRMODE
7	6	5	4	3	2	1	0
CBCCLRMODE	INT	DCBEVT1	DCBEVT0	DCAEVT1	DCAEVT0	CBC	OST

表 9-81: 封锁中断清除寄存器 (TZIC) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8:7	CBCCLRMODE	RW	0x0	周期封锁标志自动清除方式 0: TBCNT=0 时自动清除 1: TBCNT=TBPRD 时自动清除 2: TBCNT=0 或 TBCNT=TBPRD 时自动清除 3: 不自动清除
6	INT	W1C	0x0	全局封锁中断标志清除 0: 写 0 无效, 读总是 0 1: 写 1 清除中断标志, 本位段自动清零
5	DCBEVT1	W1C	0x0	DCBEVT1 事件标志清除 0: 写 0 无效, 读总是 0 1: 写 1 清除事件和中断标志, 本位段自动清零
4	DCBEVT0	W1C	0x0	DCBEVT0 事件标志清除 0: 写 0 无效, 读总是 0 1: 写 1 清除事件和中断标志, 本位段自动清零
3	DCAEVT1	W1C	0x0	DCAEVT1 事件标志清除 0: 写 0 无效, 读总是 0 1: 写 1 清除事件和中断标志, 本位段自动清零
2	DCAEVT0	W1C	0x0	DCAEVT0 事件标志清除 0: 写 0 无效, 读总是 0 1: 写 1 清除事件和中断标志, 本位段自动清零
1	CBC	W1C	0x0	周期封锁事件标志手动清除 0: 写 0 无效, 读总是 0 1: 写 1 清除事件和中断标志, 本位段自动清零

位段	位段名	属性	复位值	描述
0	OST	W1C	0x0	一次封锁事件标志清除 0: 写 0 无效, 读总是 0 1: 写 1 清除事件和中断标志, 本位段自动清零

9.9.2.38 封锁中断使能寄存器 (TZIE)

表 9-82: 封锁中断使能寄存器 (TZIE) 位段定义

TZIE (Trip-Zone Interrupt Enable Register) Access: PWM->TZIE Offset: 0x00AC Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED		DCBEVT1	DCBEVT0	DCAEVT1	DCAEVT0	CBC	OST

表 9-83: 封锁中断使能寄存器 (TZIE) 位段描述

位段	位段名	属性	复位值	描述
31:6	RESERVED_31_6	RO	0x0	保留
5	DCBEVT1	RW	0x0	数字比较 B 事件 1 中断使能 0: 关闭中断 1: 使能中断
4	DCBEVT0	RW	0x0	数字比较 B 事件 0 中断使能 0: 关闭中断 1: 使能中断
3	DCAEVT1	RW	0x0	数字比较 A 事件 1 中断使能 0: 关闭中断 1: 使能中断
2	DCAEVT0	RW	0x0	数字比较 A 事件 0 中断使能 0: 关闭中断 1: 使能中断
1	CBC	RW	0x0	周期封锁事件中断使能 0: 关闭中断 1: 使能中断
0	OST	RW	0x0	一次封锁事件中断使能 0: 关闭中断 1: 使能中断

9.9.2.39 封锁中断强制寄存器 (TZFRC)

表 9-84: 封锁中断强制寄存器 (TZFRC) 位段定义

TZFRC (Trip-Zone Force Register)							
Access: PWM->TZFRC Offset: 0x00B0 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED		DCBEVT1	DCBEVT0	DCAEVT1	DCAEVT0	CBC	OST

表 9-85: 封锁中断强制寄存器 (TZFRC) 位段描述

位段	位段名	属性	复位值	描述
31:6	RESERVED_31_6	RO	0x0	保留
5	DCBEVT1	W1S	0x0	软件强制 DCBEVT1 封锁事件 0: 写 0 无效, 读总是 0 1: 写 1 产生 DCBEVT1 封锁并置位 TZIF.DCBEVT1 本位段自动清零
4	DCBEVT0	W1S	0x0	软件强制 DCBEVT0 封锁事件 0: 写 0 无效, 读总是 0 1: 写 1 产生 DCBEVT0 封锁并置位 TZIF.DCBEVT0 本位段自动清零
3	DCAEVT1	W1S	0x0	软件强制 DCAEVT1 封锁事件 0: 写 0 无效, 读总是 0 1: 写 1 产生 DCAEVT1 封锁并置位 TZIF.DCAEVT1 本位段自动清零
2	DCAEVT0	W1S	0x0	软件强制 DCAEVT0 封锁事件 0: 写 0 无效, 读总是 0 1: 写 1 产生 DCAEVT0 封锁并置位 TZIF.DCAEVT0 本位段自动清零
1	CBC	W1S	0x0	软件强制周期封锁事件 0: 写 0 无效, 读总是 0 1: 写 1 产生周期封锁并置位 TZIF.CBC 本位段自动清零
0	OST	W1S	0x0	软件强制一次封锁事件 0: 写 0 无效, 读总是 0 1: 写 1 产生一次封锁并置位 TZIF.OST 本位段自动清零

9.9.2.40 数字比较 DCAL 事件选择寄存器 (DCALTRIPSEL)

表 9-86: 数字比较 DCAL 事件选择寄存器 (DCALTRIPSEL) 位段定义

DCALTRIPSEL (Digital Compare AL Trip Select Register)							
Access: PWM->DCALTRIPSEL Offset: 0x00B4 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED				COMP1H	RESERVED	COMP0H	COMP0L
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	EPWRTZ1	EPWRTZ0	RESERVED		TZ2	TZ1	TZ0

表 9-87: 数字比较 DCAL 事件选择寄存器 (DCALTRIPSEL) 位段描述

位段	位段名	属性	复位值	描述
31:20	RESERVED_31_20	RO	0x0	保留
19	COMP1H	RW	0x0	选中 (逻辑或) COMP1H 为 DCAL 事件 0: 不选中 1: 选中
18	RESERVED_18	RO	0x0	保留
17	COMP0H	RW	0x0	选中 (逻辑或) COMP0H 为 DCAL 事件 0: 不选中 1: 选中
16	COMP0L	RW	0x0	选中 (逻辑或) COMP0L 为 DCAL 事件 0: 不选中 1: 选中
15:7	RESERVED_15_7	RO	0x0	保留
6	EPWRTZ1	RW	0x0	选中 (逻辑或) EPWRTZ1 为 DCAL 事件 0: 不选中 1: 选中
5	EPWRTZ0	RW	0x0	选中 (逻辑或) EPWRTZ0 为 DCAL 事件 0: 不选中 1: 选中
4:3	RESERVED_4_3	RO	0x0	保留
2	TZ2	RW	0x0	选中 (逻辑或) TZ2 为 DCAL 事件 0: 不选中 1: 选中
1	TZ1	RW	0x0	选中 (逻辑或) TZ1 为 DCAL 事件

位段	位段名	属性	复位值	描述
				0: 不选中 1: 选中
0	TZ0	RW	0x0	选中（逻辑或）TZ0 为 DCAL 事件 0: 不选中 1: 选中

9.9.2.41 数字比较 DCAH 事件选择寄存器 (DCAHTRIPSEL)

表 9-88: 数字比较 DCAH 事件选择寄存器 (DCAHTRIPSEL) 位段定义

DCAHTRIPSEL (Digital Compare AH Trip Select Register)							
Access: PWM->DCAHTRIPSEL Offset: 0x00B8 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED				COMP1H	RESERVED	COMP0H	COMP0L
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	EPWRTZ1	EPWRTZ0	RESERVED		TZ2	TZ1	TZ0

表 9-89: 数字比较 DCAH 事件选择寄存器 (DCAHTRIPSEL) 位段描述

位段	位段名	属性	复位值	描述
31:20	RESERVED_31_20	RO	0x0	保留
19	COMP1H	RW	0x0	选中（逻辑或）COMP1H 为 DCAH 事件 0: 不选中 1: 选中
18	RESERVED_18	RO	0x0	保留
17	COMP0H	RW	0x0	选中（逻辑或）COMP0H 为 DCAH 事件 0: 不选中 1: 选中
16	COMP0L	RW	0x0	选中（逻辑或）COMP0L 为 DCAH 事件 0: 不选中 1: 选中
15:7	RESERVED_15_7	RO	0x0	保留
6	EPWRTZ1	RW	0x0	选中（逻辑或）EPWRTZ1 为 DCAH 事件 0: 不选中 1: 选中
5	EPWRTZ0	RW	0x0	选中（逻辑或）EPWRTZ0 为 DCAH 事件 0: 不选中 1: 选中
4:3	RESERVED_4_3	RO	0x0	保留
2	TZ2	RW	0x0	选中（逻辑或）TZ2 为 DCAH 事件 0: 不选中 1: 选中
1	TZ1	RW	0x0	选中（逻辑或）TZ1 为 DCAH 事件

位段	位段名	属性	复位值	描述
				0: 不选中 1: 选中
0	TZ0	RW	0x0	选中（逻辑或）TZ0 为 DCAH 事件 0: 不选中 1: 选中

9.9.2.42 数字比较 DCBL 事件选择寄存器 (DCBLTRIPSEL)

表 9-90: 数字比较 DCBL 事件选择寄存器 (DCBLTRIPSEL) 位段定义

DCBLTRIPSEL (Digital Compare BL Trip Select Register)							
Access: PWM->DCBLTRIPSEL Offset: 0x00BC Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED				COMP1H	RESERVED	COMP0H	COMP0L
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	EPWRTZ1	EPWRTZ0	RESERVED		TZ2	TZ1	TZ0

表 9-91: 数字比较 DCBL 事件选择寄存器 (DCBLTRIPSEL) 位段描述

位段	位段名	属性	复位值	描述
31:26	RESERVED_31_26	RO	0x0	保留
19	COMP1H	RW	0x0	选中（逻辑或）COMP1H 为 DCBL 事件 0: 不选中 1: 选中
18	RESERVED_18	RO	0x0	保留
17	COMP0H	RW	0x0	选中（逻辑或）COMP0H 为 DCBL 事件 0: 不选中 1: 选中
16	COMP0L	RW	0x0	选中（逻辑或）COMP0L 为 DCBL 事件 0: 不选中 1: 选中
15:17	RESERVED_15_17	RO	0x0	保留
6	EPWRTZ1	RW	0x0	选中（逻辑或）EPWRTZ1 为 DCBL 事件 0: 不选中 1: 选中
5	EPWRTZ0	RW	0x0	选中（逻辑或）EPWRTZ0 为 DCBL 事件 0: 不选中 1: 选中
4:3	RESERVED_4_3	RO	0x0	保留
2	TZ2	RW	0x0	选中（逻辑或）TZ2 为 DCBL 事件 0: 不选中 1: 选中
1	TZ1	RW	0x0	选中（逻辑或）TZ1 为 DCBL 事件 0: 不选中 1: 选中

位段	位段名	属性	复位值	描述
0	TZ0	RW	0x0	选中（逻辑或）TZ0 为 DCBL 事件 0: 不选中 1: 选中

9.9.2.43 数字比较 DCBH 事件选择寄存器 (DCBHTRIPSEL)

表 9-92: 数字比较 DCBH 事件选择寄存器 (DCBHTRIPSEL) 位段定义

DCBHTRIPSEL (Digital Compare BH Trip Select Register)							
Access: PWM->DCBHTRIPSEL Offset: 0x00C0 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED				COMP1H	RESERVED	COMP0H	COMP0L
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	EPWRTZ1	EPWRTZ0	RESERVED		TZ2	TZ1	TZ0

表 9-93: 数字比较 DCBH 事件选择寄存器 (DCBHTRIPSEL) 位段描述

位段	位段名	属性	复位值	描述
31:20	RESERVED_31_20	RO	0x0	保留
19	COMP1H	RW	0x0	选中（逻辑或）COMP1H 为 DCBH 事件 0: 不选中 1: 选中
18	RESERVED_18	RO	0x0	保留
17	COMP0H	RW	0x0	选中（逻辑或）COMP0H 为 DCBH 事件 0: 不选中 1: 选中
16	COMP0L	RW	0x0	选中（逻辑或）COMP0L 为 DCBH 事件 0: 不选中 1: 选中
15:7	RESERVED_15_7	RO	0x0	保留
6	EPWRTZ1	RW	0x0	选中（逻辑或）EPWRTZ1 为 DCBH 事件 0: 不选中 1: 选中
5	EPWRTZ0	RW	0x0	选中（逻辑或）EPWRTZ0 为 DCBH 事件 0: 不选中 1: 选中
4:3	RESERVED_4_3	RO	0x0	保留
2	TZ2	RW	0x0	选中（逻辑或）TZ2 为 DCBH 事件 0: 不选中 1: 选中
1	TZ1	RW	0x0	选中（逻辑或）TZ1 为 DCBH 事件

位段	位段名	属性	复位值	描述
				0: 不选中 1: 选中
0	TZ0	RW	0x0	选中（逻辑或）TZ0 为 DCBH 事件 0: 不选中 1: 选中

9.9.2.44 数字比较 A 控制寄存器（DCACTL）

表 9-94：数字比较 A 控制寄存器（DCACTL）位段定义

DCACTL (Digital Compare A Control Register) Access: PWM->DCACTL Offset: 0x00C4 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED		EVT1SRCASYNCE	RESERVED	EVT0SYNCE	EVT0REQE	EVT0SRCASYNCE	RESERVED

表 9-95：数字比较 A 控制寄存器（DCACTL）位段描述

位段	位段名	属性	复位值	描述
31:6	RESERVED_31_6	RO	0x0	保留
5	EVT1SRCASYNCE	RW	0x0	DCAEVT1 源信号通路选择 0: 同步通路 1: 异步通路
4	RESERVED_4	RO	0x0	保留
3	EVT0SYNCE	RW	0x0	DCAEVT0 SYNC 信号产生使能 0: 不产生 DCAEVT0 SYNC 信号 1: 产生 DCAEVT0 SYNC 信号
2	EVT0REQE	RW	0x0	DCAEVT0 REQ 信号产生使能 0: 不产生 DCAEVT0 REQ 信号 1: 产生 DCAEVT0 REQ 信号
1	EVT0SRCASYNCE	RW	0x0	DCAEVT0 源信号通路选择 0: 同步通路 1: 异步通路
0	RESERVED_0	RO	0x0	保留

9.9.2.45 数字比较 B 控制寄存器 (DCBCTL)

表 9-96: 数字比较 B 控制寄存器 (DCBCTL) 位段定义

DCBCTL (Digital Compare B Control Register)							
Access: PWM->DCBCTL Offset: 0x00C8 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED		EVT1SRCASYNC	RESERVED	EVT0SYNCE	EVT0REQE	EVT0SRCASYNC	RESERVED

表 9-97: 数字比较 B 控制寄存器 (DCBCTL) 位段描述

位段	位段名	属性	复位值	描述
31:6	RESERVED_31_6	RO	0x0	保留
5	EVT1SRCASYNC	RW	0x0	DCBEVT1 源信号通路选择 0: 同步通路 1: 异步通路
4	RESERVED_4	RO	0x0	保留
3	EVT0SYNCE	RW	0x0	DCBEVT0 SYNC 信号产生使能 0: 不产生 DCBEVT0 SYNC 信号 1: 产生 DCBEVT0 SYNC 信号
2	EVT0REQE	RW	0x0	DCBEVT0 REQ 信号产生使能 0: 不产生 DCBEVT0 REQ 信号 1: 产生 DCBEVT0 REQ 信号
1	EVT0SRCASYNC	RW	0x0	DCBEVT0 源信号通路选择 0: 同步通路 1: 异步通路
0	RESERVED_0	RO	0x0	保留

9.9.2.46 事件触发控制寄存器（ETCTL）

表 9-98：事件触发控制寄存器（ETCTL）位段定义

ETCTL (Event-Trigger Control Register)							
Access: PWM->ETCTL Offset: 0x00E8 Default: 0x0007B16F							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED				REQCEN	REQCSEL		
15	14	13	12	11	10	9	8
REQCSEL	REQBEN	REQBSEL				REQAEN	REQASEL
7	6	5	4	3	2	1	0
REQASEL			INTEN	INTSEL			

表 9-99：事件触发控制寄存器（ETCTL）位段描述

位段	位段名	属性	复位值	描述
31:20	RESERVED_31_20	RO	0x0	保留
19	REQCEN	RW	0x0	ADC 采样开始脉冲 C（REQC）使能 0: 关闭 1: 使能
18:15	REQCSEL	RW	0xF	REQC 事件选择 0: TBCNT=0 时触发事件 1: TBCNT=TBPRD 时触发事件 2: TBCNT=0 或 TBCNT=TBPRD 时触发事件 3: TBCNT=CMPA 且 TBCNT 递增计数时触发 4: TBCNT=CMPA 且 TBCNT 递减计数时触发 5: TBCNT=CMPB 且 TBCNT 递增计数时触发 6: TBCNT=CMPB 且 TBCNT 递减计数时触发 7: TBCNT=CMPC 且 TBCNT 递增计数时触发 8: TBCNT=CMPC 且 TBCNT 递减计数时触发 11: DCAEVT0.REQ 事件触发 12: DCBEVT0.REQ 事件触发 其他: 不产生（禁止）REQC 事件
14	REQBEN	RW	0x0	ADC 采样开始脉冲 B（REQB）使能 0: 关闭 1: 使能
13:10	REQBSEL	RW	0xC	REQB 事件选择 0: TBCNT=0 时触发事件 1: TBCNT=TBPRD 时触发事件 2: TBCNT=0 或 TBCNT=TBPRD 时触发事件 3: TBCNT=CMPA 且 TBCNT 递增计数时触发 4: TBCNT=CMPA 且 TBCNT 递减计数时触发 5: TBCNT=CMPB 且 TBCNT 递增计数时触发 6: TBCNT=CMPB 且 TBCNT 递减计数时触发

位段	位段名	属性	复位值	描述
				7: TBCNT=CMPC 且 TBCNT 递增计数时触发 8: TBCNT=CMPC 且 TBCNT 递减计数时触发 11: DCAEVT0.REQ 事件触发 12: DCBEVT0.REQ 事件触发 其他: 不产生（禁止）REQB 事件
9	REQAEN	RW	0x0	ADC 采样开始脉冲 A（REQA）使能 0: 关闭 1: 使能
8:5	REQASEL	RW	0xB	REQA 事件选择 0: TBCNT=0 时触发事件 1: TBCNT=TBPRD 时触发事件 2: TBCNT=0 或 TBCNT=TBPRD 时触发事件 3: TBCNT=CMPA 且 TBCNT 递增计数时触发 4: TBCNT=CMPA 且 TBCNT 递减计数时触发 5: TBCNT=CMPB 且 TBCNT 递增计数时触发 6: TBCNT=CMPB 且 TBCNT 递减计数时触发 7: TBCNT=CMPC 且 TBCNT 递增计数时触发 8: TBCNT=CMPC 且 TBCNT 递减计数时触发 11: DCAEVT0.REQ 事件触发 12: DCBEVT0.REQ 事件触发 其他: 不产生（禁止）REQA 事件
4	INTEN	RW	0x0	中断脉冲使能 0: 关闭 1: 使能
3:0	INTSEL	RW	0xF	中断事件选择 0: TBCNT=0 时触发事件 1: TBCNT=TBPRD 时触发事件 2: TBCNT=0 或 TBCNT=TBPRD 时触发事件 3: TBCNT=CMPA 且 TBCNT 递增计数时触发 4: TBCNT=CMPA 且 TBCNT 递减计数时触发 5: TBCNT=CMPB 且 TBCNT 递增计数时触发 6: TBCNT=CMPB 且 TBCNT 递减计数时触发 7: TBCNT=CMPC 且 TBCNT 递增计数时触发 8: TBCNT=CMPC 且 TBCNT 递减计数时触发 11: DCAEVT0.REQ 事件触发 12: DCBEVT0.REQ 事件触发 其他: 不产生（禁止）中断事件

9.9.2.47 事件触发预分频寄存器 (ETPS)

表 9-100: 事件触发预分频寄存器 (ETPS) 位段定义

ETPS (Event-Trigger Prescale Register)							
Access: PWM->ETPS		Offset: 0x00EC		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED		REQCNT		RESERVED		REQPRD	
23	22	21	20	19	18	17	16
RESERVED		REQBCNT		RESERVED		REQBPRD	
15	14	13	12	11	10	9	8
RESERVED		REQACNT		RESERVED		REQAPRD	
7	6	5	4	3	2	1	0
RESERVED		INTCNT		RESERVED		INTPRD	

表 9-101: 事件触发预分频寄存器 (ETPS) 位段描述

位段	位段名	属性	复位值	描述
31:30	RESERVED_31_30	RO	0x0	保留
29:28	REQCNT	RO	0x0	REQC 事件计数 该位段记录了 ETCTL.REQCSEL 指定的 REQC 事件发生的次数。当 REQC 脉冲产生时，该位段自动清零。 0: 事件未发生 1: 事件发生了 1 次 2: 事件发生了 2 次 3: 事件发生了 3 次
27:26	RESERVED_27_26	RO	0x0	保留
25:24	REQPRD	RW	0x0	REQC 事件周期 当 ETCTL.REQCEN=1 且 REQCNT =REQCPRD 时产生 REQC 脉冲。 0: 禁用 REQC 计数器且不产生 REQC 事件 1: 当 ETPS.REQCNT=1（每次）时产生 REQC 2: 当 ETPS.REQCNT=2（每 2 次）时产生 REQC 3: 当 ETPS.REQCNT=3（每 3 次）时产生 REQC
23:22	RESERVED_23_22	RO	0x0	保留
21:20	REQBCNT	RO	0x0	REQB 事件计数 该位段记录了 ETCTL.REQBSEL 指定的 REQB 事件发生的次数。当 REQB 脉冲产生时，该位段自动清零。 0: 事件未发生 1: 事件发生了 1 次 2: 事件发生了 2 次 3: 事件发生了 3 次
19:18	RESERVED_19_18	RO	0x0	保留

位段	位段名	属性	复位值	描述
17:16	REQBPRD	RW	0x0	REQB 事件周期 当 ETCTL.REQBEN=1 且 REQBCNT =REQBPRD 时产生 REQB 脉冲。 0: 禁用 REQB 计数器且不产生 REQB 事件 1: 当 ETPS.REQBCNT=1（每次）时产生 REQB 2: 当 ETPS.REQBCNT=2（每 2 次）时产生 REQB 3: 当 ETPS.REQBCNT=3（每 3 次）时产生 REQB
15:14	RESERVED_15_14	RO	0x0	保留
13:12	REQACNT	RO	0x0	REQA 事件计数 该位段记录了 ETCTL.REQASEL 指定的 REQA 事件发生的次数。当 REQA 脉冲产生时，该位段自动清零。 0: 事件未发生 1: 事件发生了 1 次 2: 事件发生了 2 次 3: 事件发生了 3 次
11:10	RESERVED_11_10	RO	0x0	保留
9:8	REQAPRD	RW	0x0	REQA 事件周期 当 ETCTL.REQAEN=1 且 REQACNT =REQAPRD 时产生 REQA 脉冲。 0: 禁用 REQA 计数器且不产生 REQA 事件 1: 当 ETPS.REQACNT=1（每次）时产生 REQA 2: 当 ETPS.REQACNT=2（每 2 次）时产生 REQA 3: 当 ETPS.REQACNT=3（每 3 次）时产生 REQA
7:6	RESERVED_7_6	RO	0x0	保留
5:4	INTCNT	RO	0x0	中断事件计数 该位段记录了 ETCTL.INTSEL 指定的中断事件发生的次数。当中断脉冲产生时，该位段自动清零。 0: 事件未发生 1: 事件发生了 1 次 2: 事件发生了 2 次 3: 事件发生了 3 次
3:2	RESERVED_3_2	RO	0x0	保留
1:0	INTPRD	RW	0x0	中断事件周期 当 ETCTL.INTEN=1 且 INTCNT =INTPRD 时产生中断脉冲。 0: 禁用中断计数器且不产生中断事件 1: 当 ETPS.INTCNT=1（每次）时产生中断脉冲

位段	位段名	属性	复位值	描述
				2: 当 ETPS.INTCNT=2 (每 2 次) 时产生中断脉冲 3: 当 ETPS.INTCNT=3 (每 3 次) 时产生中断脉冲

9.9.2.48 事件触发标志寄存器 (ETFLG)

表 9-102: 事件触发标志寄存器 (ETFLG) 位段定义

ETFLG (Event-Trigger Flag Register) Access: PWM->ETFLG Offset: 0x00F0 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED				REQC	REQB	REQA	INT

表 9-103: 事件触发标志寄存器 (ETFLG) 位段描述

位段	位段名	属性	复位值	描述
31:4	RESERVED_31_4	RO	0x0	保留
3	REQC	RO	0x0	送至 ADC 的采样开始脉冲 REQC 发生的标志 0: REQC 脉冲未产生 1: REQC 脉冲产生 本位段置 1 时不会阻止后续 REQC 脉冲产生。
2	REQB	RO	0x0	送至 ADB 的采样开始脉冲 REQB 发生的标志 0: REQB 脉冲未产生 1: REQB 脉冲产生 本位段置 1 时不会阻止后续 REQB 脉冲产生。
1	REQA	RO	0x0	送至 ADC 的采样开始脉冲 REQA 发生的标志 0: REQA 脉冲未产生 1: REQA 脉冲产生 本位段置 1 时不会阻止后续 REQA 脉冲产生。
0	INT	RO	0x0	送至 CPU 的中断发生的标志 0: 中断脉冲未产生 1: 中断脉冲产生 本位段置 1 时将阻止后续中断脉冲发生。

9.9.2.49 事件触发标志清除寄存器 (ETCLR)

表 9-104: 事件触发标志清除寄存器 (ETCLR) 位段定义

ETCLR (Event-Trigger Clear Register)							
Access: PWM->ETCLR		Offset: 0x00F4		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED				REQC	REQB	REQA	INT

表 9-105: 事件触发标志清除寄存器 (ETCLR) 位段描述

位段	位段名	属性	复位值	描述
31:4	RESERVED_31_4	RO	0x0	保留
3	REQC	W1C	0x0	REQC 标志清除 0: 写 0 无效, 读总是 0 1: 写清除 ETFLG.REQC 标志, 本位段自动清零
2	REQB	W1C	0x0	REQB 标志清除 0: 写 0 无效, 读总是 0 1: 写清除 ETFLG.REQB 标志, 本位段自动清零
1	REQA	W1C	0x0	REQA 标志清除 0: 写 0 无效, 读总是 0 1: 写清除 ETFLG.REQA 标志, 本位段自动清零
0	INT	W1C	0x0	中断标志清除 0: 写 0 无效, 读总是 0 1: 写清除 ETFLG.INT 标志, 本位段自动清零

9.9.2.50 事件触发软件强制寄存器（ETFRC）

表 9-106：事件触发软件强制寄存器（ETFRC）位段定义

ETFRC (Event-Trigger Force Register)							
Access: PWM->ETFRC		Offset: 0x00F8		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED				REQC	REQB	REQA	INT

表 9-107：事件触发软件强制寄存器（ETFRC）位段描述

位段	位段名	属性	复位值	描述
31:4	RESERVED_31_4	RO	0x0	保留
3	REQC	W1S	0x0	软件强制 REQC 事件 ETCTL.REQCEN=0 或 ETPS.REQCPRD=0 时位段无效 0: 写 0 无效，读总是 0 1: 写 1 强制产生一次 REQC 事件，位段自动清零
2	REQB	W1S	0x0	软件强制 REQB 事件 ETCTL.REQBEN=0 或 ETPS.REQBPRD=0 时位段无效 0: 写 0 无效，读总是 0 1: 写 1 强制产生一次 REQB 事件，位段自动清零
1	REQA	W1S	0x0	软件强制 REQA 事件 ETCTL.REQAEN=0 或 ETPS.REQAPRD=0 时位段无效 0: 写 0 无效，读总是 0 1: 写 1 强制产生一次 REQA 事件，位段自动清零
0	INT	W1S	0x0	软件强制中断事件 ETCTL.INTEN=0 或 ETPS.INTPRD=0 时位段无效 0: 写 0 无效，读总是 0 1: 写 1 强制产生一次中断事件，位段自动清零

9.9.2.51 PWM 模块写使能寄存器 (PWMREGKEY)

表 9-108: PWM 模块写使能寄存器 (PWMREGKEY) 位段定义

PWMREGKEY (PWM Register Write-Allow Key Register)							
Access: PWM->PWMREGKEY Offset: 0x00FC Default: 0x1ACCE551							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

表 9-109: PWM 模块写使能寄存器 (PWMREGKEY) 位段描述

位段	位段名	属性	复位值	描述
31:0	KEY	RW	0x1ACCE551	写入 0x1ACCE551 以解锁受保护的 PWM 寄存器

9.9.3 PWMCFG 寄存器列表

表 9-110: PWMCFG 模块基地址

外设模块	基地址
PWMCFG	0x4000 9F00

表 9-111: PWMCFG 寄存器列表

寄存器	偏移地址	描述	复位值
TZ0SRCCTL*	0x00	TZ0 信号源控制寄存器	0x0000007F
TZ1SRCCTL*	0x04	TZ1 信号源控制寄存器	0x0000007F
TZ2SRCCTL*	0x08	TZ2 信号源控制寄存器	0x0000007F
FRCSYNC	0x14	全局 PWM 软件强制同步寄存器	0x00000000
GPIOSYNCCTL*	0x18	GPIO 触发 PWMSYNCI 信号源控制寄存器	0x0000007F
GPIOSYNCEN*	0x1C	GPIO 触发 PWMSYNCI 使能寄存器	0x00000008
TMR0SYNCEN*	0x20	Timer 0 触发 PWMSYNCI 使能寄存器	0x00000000
TMR1SYNCEN*	0x24	Timer 1 触发 PWMSYNCI 使能寄存器	0x00000000
ECAPSYNCEN	0x28	ECAP 强制 PWMSYNCI 使能寄存器	0x00000000
SYNCOCTL*	0x2C	PWMSYNCO 输出控制寄存器	0x00000007
REQAOCTL*	0x30	PWMREQA 输出控制寄存器	0x00000007
REQBOCTL*	0x34	PWMREQB 输出控制寄存器	0x00000007
REQCOCTL*	0x38	PWMREQC 输出控制寄存器	0x00000007
PWMCFGREGKEY	0x3C	PWMCFG 模块写使能寄存器	0x1ACCE551

注意： 由*标记的寄存器仅当 PWMCFGKEY=0x1ACCE551 时才可以改写。

9.9.4 PWMCFG 寄存器

9.9.4.1 TZ0 信号源控制寄存器 (TZ0SRCCTL)

表 9-112: TZ0 信号源控制寄存器 (TZ0SRCCTL) 位段定义

TZ0SRCCTL (TZ0 Source Control Register)							
Access: PWMCFG->TZ0SRCCTL Offset: 0x0000 Default: 0x0000007F							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	POL	IOSEL					

表 9-113: TZ0 信号源控制寄存器 (TZ0SRCCTL) 位段描述

位段	位段名	属性	复位值	描述
31:7	RESERVED_31_7	RO	0x0	保留
6	POL	RW	0x1	TZ0 极性 0: GPIO 输入低时触发 TZ0 1: GPIO 输入高时触发 TZ0
5:0	IOSEL	RW	0x3F	触发 TZ0 的 GPIO 编号

9.9.4.2 TZ1 信号源控制寄存器 (TZ1SRCCTL)

表 9-114: TZ1 信号源控制寄存器 (TZ1SRCCTL) 位段定义

TZ1SRCCTL (TZ1 Source Control Register)							
Access: PWMCFG->TZ1SRCCTL Offset: 0x0004 Default: 0x0000007F							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	POL	IOSEL					

表 9-115: TZ1 信号源控制寄存器 (TZ1SRCCTL) 位段描述

位段	位段名	属性	复位值	描述
31:7	RESERVED_31_7	RO	0x0	保留
6	POL	RW	0x1	TZ1 极性 0: GPIO 输入低时触发 TZ1 1: GPIO 输入高时触发 TZ1
5:0	IOSEL	RW	0x3F	触发 TZ1 的 GPIO 编号

9.9.4.3 TZ2 信号源控制寄存器 (TZ2SRCCTL)

表 9-116: TZ2 信号源控制寄存器 (TZ2SRCCTL) 位段定义

TZ2SRCCTL (TZ2 Source Control Register)							
Access: PWMCFG->TZ2SRCCTL Offset: 0x0008 Default: 0x0000007F							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	POL	IOSEL					

表 9-117: TZ2 信号源控制寄存器 (TZ2SRCCTL) 位段描述

位段	位段名	属性	复位值	描述
31:7	RESERVED_31_7	RO	0x0	保留
6	POL	RW	0x1	TZ2 极性 0: GPIO 输入低时触发 TZ2 1: GPIO 输入高时触发 TZ2
5:0	IOSEL	RW	0x3F	触发 TZ2 的 GPIO 编号

9.9.4.4 全局 PWM 软件强制同步寄存器（FRCSYNC）

表 9-118: 全局 PWM 软件强制同步寄存器（FRCSYNC）位段定义

FRCSYNC (Global PWM Force Synchronization Register)							
Access: PWMCFG->FRCSYNC Offset: 0x0014 Default: 0x00000000							
31	30	29	28	27	26	25	24
PWMCLK	RESERVED						
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED		PWM5SYNC	PWM4SYNC	PWM3SYNC	PWM2SYNC	RESERVED	

表 9-119: 全局 PWM 软件强制同步寄存器（FRCSYNC）位段描述

位段	位段名	属性	复位值	描述
31	PWMCLK	W1S	0x0	软件强制同步所有 PWM 的时钟 0: 写 0 无效，读总是 0 1: 写 1 同步所有 PWM 的时钟，本位段自动清零
30:6	RESERVED_30_6	RO	0x0	保留
5	PWM5SYNC	W1S	0x0	软件强制 PWM5 同步 0: 写 0 无效，读总是 0 1: 写 1 产生 PWM5 同步脉冲，本位段自动清零
4	PWM4SYNC	W1S	0x0	软件强制 PWM4 同步 0: 写 0 无效，读总是 0 1: 写 1 产生 PWM4 同步脉冲，本位段自动清零
3	PWM3SYNC	W1S	0x0	软件强制 PWM3 同步 0: 写 0 无效，读总是 0 1: 写 1 产生 PWM3 同步脉冲，本位段自动清零
2	PWM2SYNC	W1S	0x0	软件强制 PWM2 同步 0: 写 0 无效，读总是 0 1: 写 1 产生 PWM2 同步脉冲，本位段自动清零
1:0	RESERVED_1_0	RO	0x0	保留

9.9.4.5 GPIO 触发 PWMSYNCl 信号源控制寄存器 (GPIOSYNCCTL)

表 9-120: GPIO 触发 PWMSYNCl 信号源控制寄存器 (GPIOSYNCCTL) 位段定义

GPIOSYNCCTL (GPIO PWMSYNCl Control Register)							
Access: PWMCFG->GPIOSYNCCTL		Offset: 0x0018		Default: 0x0000007F			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	POL	IOSEL					

表 9-121: GPIO 触发 PWMSYNCl 信号源控制寄存器 (GPIOSYNCCTL) 位段描述

位段	位段名	属性	复位值	描述
31:7	RESERVED_31_7	RO	0x0	保留
6	POL	RW	0x1	PWMSYNCl 极性 0: GPIO 输入低时触发 PWMSYNCl 1: GPIO 输入高时触发 PWMSYNCl
5:0	IOSEL	RW	0x3F	触发 PWMSYNCl 的 GPIO 输入引脚编号

9.9.4.6 GPIO 触发 PWMSYNCl 使能寄存器 (GPIOSYNCEN)

表 9-122: GPIO 触发 PWMSYNCl 使能寄存器 (GPIOSYNCEN) 位段定义

GPIOSYNCEN (GPIO Force PWMSYNCl Enable Register)							
Access: PWMCFG->GPIOSYNCEN Offset: 0x001C Default: 0x00000008							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED		PWM5SYNC	PWM4SYNC	PWM3SYNC	PWM2SYNC	RESERVED	

表 9-123: GPIO 触发 PWMSYNCl 使能寄存器 (GPIOSYNCEN) 位段描述

位段	位段名	属性	复位值	描述
31:6	RESERVED_31_6	RO	0x0	保留
5	PWM5SYNC	RW	0x0	使能 GPIO 触发 PWM5 同步 0: 关闭 1: 使能
4	PWM4SYNC	RW	0x0	使能 GPIO 触发 PWM4 同步 0: 关闭 1: 使能
3	PWM3SYNC	RW	0x1	使能 GPIO 触发 PWM3 同步 0: 关闭 1: 使能
2	PWM2SYNC	RW	0x0	使能 GPIO 触发 PWM2 同步 0: 关闭 1: 使能
1:0	RESERVED_1_0	RO	0x0	保留

9.9.4.7 Timer 0 触发 PWMSYNCl 使能寄存器 (TMR0SYNCEN)

表 9-124: Timer 0 触发 PWMSYNCl 使能寄存器 (TMR0SYNCEN) 位段定义

TMR0SYNCEN (Timer 0 Force PWMSYNCl Enable Register)							
Access: PWMCFG->TMR0SYNCEN Offset: 0x0020 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED		PWM5SYNC	PWM4SYNC	PWM3SYNC	PWM2SYNC	RESERVED	

表 9-125: Timer 0 触发 PWMSYNCl 使能寄存器 (TMR0SYNCEN) 位段描述

位段	位段名	属性	复位值	描述
31:6	RESERVED_31_6	RO	0x0	保留
5	PWM5SYNC	RW	0x0	使能 TIMERO 触发 PWM5 同步 0: 关闭 1: 使能
4	PWM4SYNC	RW	0x0	使能 TIMERO 触发 PWM4 同步 0: 关闭 1: 使能
3	PWM3SYNC	RW	0x0	使能 TIMERO 触发 PWM3 同步 0: 关闭 1: 使能
2	PWM2SYNC	RW	0x0	使能 TIMERO 触发 PWM2 同步 0: 关闭 1: 使能
1:0	RESERVED_1_0	RO	0x0	保留

9.9.4.8 Timer 1 触发 PWMSYNCl 使能寄存器 (TMR1SYNCEN)

表 9-126: Timer 1 触发 PWMSYNCl 使能寄存器 (TMR1SYNCEN) 位段定义

TMR1SYNCEN (Timer 1 Force PWMSYNCl Enable Register)							
Access: PWMCFG->TMR1SYNCEN Offset: 0x0024 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED		PWM5SYNC	PWM4SYNC	PWM3SYNC	PWM2SYNC	RESERVED	

表 9-127: Timer 1 触发 PWMSYNCl 使能寄存器 (TMR1SYNCEN) 位段描述

位段	位段名	属性	复位值	描述
31:6	RESERVED_31_6	RO	0x0	保留
5	PWM5SYNC	RW	0x0	使能 TIMER1 触发 PWM5 同步 0: 关闭 1: 使能
4	PWM4SYNC	RW	0x0	使能 TIMER1 触发 PWM4 同步 0: 关闭 1: 使能
3	PWM3SYNC	RW	0x0	使能 TIMER1 触发 PWM3 同步 0: 关闭 1: 使能
2	PWM2SYNC	RW	0x0	使能 TIMER1 触发 PWM2 同步 0: 关闭 1: 使能
1:0	RESERVED_1_0	RO	0x0	保留

9.9.4.9 ECAP 强制 PWMSYNCl 使能寄存器 (ECAPSYNCEN)

表 9-128: ECAP 强制 PWMSYNCl 使能寄存器 (ECAPSYNCEN) 位段定义

ECAPSYNCEN (ECAP Force PWMSYNCl Enable Register)							
Access: PWMCFG->ECAPSYNCEN Offset: 0x0028 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED		PWM5SYNC	PWM4SYNC	PWM3SYNC	PWM2SYNC	RESERVED	

表 9-129: ECAP 强制 PWMSYNCl 使能寄存器 (ECAPSYNCEN) 位段描述

位段	位段名	属性	复位值	描述
31:6	RESERVED_31_6	RO	0x0	保留
5	PWM5SYNC	RW	0x0	使能 TIMER2 触发 PWM5 同步 0: 关闭 1: 使能
4	PWM4SYNC	RW	0x0	使能 TIMER2 触发 PWM4 同步 0: 关闭 1: 使能
3	PWM3SYNC	RW	0x0	使能 TIMER2 触发 PWM3 同步 0: 关闭 1: 使能
2	PWM2SYNC	RW	0x0	使能 TIMER2 触发 PWM2 同步 0: 关闭 1: 使能
1:0	RESERVED_1_0	RO	0x0	保留

9.9.4.10 PWMSYNCO 输出控制寄存器 (SYNCOCTL)

表 9-130: PWMSYNCO 输出控制寄存器 (SYNCOCTL) 位段定义

SYNCOCTL (PWMSYNCO Control Register)							
Access: PWMCFG->SYNCOCTL Offset: 0x002C Default: 0x00000007							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED		SRCSEL			POL	DURATION	

表 9-131: PWMSYNCO 输出控制寄存器 (SYNCOCTL) 位段描述

位段	位段名	属性	复位值	描述
31:6	RESERVED_31_6	RO	0x0	保留
5:3	SRCSEL	RW	0x0	PWMSYNCO 信号源选择 2: 选择 PWM2 的 PWMSYNCO 输出 3: 选择 PWM3 的 PWMSYNCO 输出 4: 选择 PWM4 的 PWMSYNCO 输出 5: 选择 PWM5 的 PWMSYNCO 输出
2	POL	RW	0x1	PWMSYNCO 输出到 IO 管脚的极性 0: PWMSYNCO 有效时将管脚置低 1: PWMSYNCO 有效时将管脚置高
1:0	DURATION	RW	0x3	PWMSYNCO 输出到 IO 管脚的脉冲宽度 0: 脉冲宽度为 4 个 PWM 时钟周期 1: 脉冲宽度为 8 个 PWM 时钟周期 2: 脉冲宽度为 16 个 PWM 时钟周期 3: 脉冲宽度为 32 个 PWM 时钟周期

9.9.4.11 PWMREQA 输出控制寄存器 (REQAOCTL)

表 9-132: PWMREQA 输出控制寄存器 (REQAOCTL) 位段定义

REQAOCTL (PWMREQA Output Control Register)							
Access: PWMCFG->REQAOCTL Offset: 0x0030 Default: 0x00000007							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							REQA5EN
7	6	5	4	3	2	1	0
REQA4EN	REQA3EN	REQA2EN	RESERVED		POL	DURATION	

表 9-133: PWMREQA 输出控制寄存器 (REQAOCTL) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8	REQA5EN	RW	0x0	使能 PWM5REQA 输出到引脚（逻辑或） 0: 关闭 1: 使能
7	REQA4EN	RW	0x0	使能 PWM4REQA 输出到引脚（逻辑或） 0: 关闭 1: 使能
6	REQA3EN	RW	0x0	使能 PWM3REQA 输出到引脚（逻辑或） 0: 关闭 1: 使能
5	REQA2EN	RW	0x0	使能 PWM2REQA 输出到引脚（逻辑或） 0: 关闭 1: 使能
4:3	RESERVED_4_3	RO	0x0	保留
2	POL	RW	0x1	PWMREQA 输出到 IO 管脚的极性 0: PWMREQA 有效时将管脚置低 1: PWMREQA 有效时将管脚置高
1:0	DURATION	RW	0x3	PWMREQA 输出到 IO 管脚的脉冲宽度 0: 脉冲宽度为 4 个 PWM 时钟周期 1: 脉冲宽度为 8 个 PWM 时钟周期 2: 脉冲宽度为 16 个 PWM 时钟周期 3: 脉冲宽度为 32 个 PWM 时钟周期

9.9.4.12 PWMREQB 输出控制寄存器 (REQBOCTL)

表 9-134: PWMREQB 输出控制寄存器 (REQBOCTL) 位段定义

REQBOCTL (PWMREQB Output Control Register)							
Access: PWMCFG->REQBOCTL Offset: 0x0034 Default: 0x00000007							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							REQB5EN
7	6	5	4	3	2	1	0
REQB4EN	REQB3EN	REQB2EN	RESERVED		POL	DURATION	

表 9-135: PWMREQB 输出控制寄存器 (REQBOCTL) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8	REQB5EN	RW	0x0	使能 PWM5REQB 输出到引脚（逻辑或） 0: 关闭 1: 使能
7	REQB4EN	RW	0x0	使能 PWM4REQB 输出到引脚（逻辑或） 0: 关闭 1: 使能
6	REQB3EN	RW	0x0	使能 PWM3REQB 输出到引脚（逻辑或） 0: 关闭 1: 使能
5	REQB2EN	RW	0x0	使能 PWM2REQB 输出到引脚（逻辑或） 0: 关闭 1: 使能
4:3	RESERVED_4_3	RO	0x0	保留
2	POL	RW	0x1	PWMREQB 输出到 IO 管脚的极性 0: PWMREQB 有效时将管脚置低 1: PWMREQB 有效时将管脚置高
1:0	DURATION	RW	0x3	PWMREQB 输出到 IO 管脚的脉冲宽度 0: 脉冲宽度为 4 个 PWM 时钟周期 1: 脉冲宽度为 8 个 PWM 时钟周期 2: 脉冲宽度为 16 个 PWM 时钟周期 3: 脉冲宽度为 32 个 PWM 时钟周期

9.9.4.13 PWMREQC 输出控制寄存器 (REQCOCTL)

表 9-136: PWMREQC 输出控制寄存器 (REQCOCTL) 位段定义

REQCOCTL (PWMREQC Output Control Register)							
Access: PWMCFG->REQCOCTL Offset: 0x0038 Default: 0x00000007							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							REQC5EN
7	6	5	4	3	2	1	0
REQC4EN	REQC3EN	REQC2EN	RESERVED		POL	DURATION	

表 9-137: PWMREQC 输出控制寄存器 (REQCOCTL) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8	REQC5EN	RW	0x0	使能 PWM5REQC 输出到引脚（逻辑或） 0: 关闭 1: 使能
7	REQC4EN	RW	0x0	使能 PWM4REQC 输出到引脚（逻辑或） 0: 关闭 1: 使能
6	REQC3EN	RW	0x0	使能 PWM3REQC 输出到引脚（逻辑或） 0: 关闭 1: 使能
5	REQC2EN	RW	0x0	使能 PWM2REQC 输出到引脚（逻辑或） 0: 关闭 1: 使能
4:3	RESERVED_4_3	RO	0x0	保留
2	POL	RW	0x1	PWMREQC 输出到 IO 管脚的极性 0: PWMREQC 有效时将管脚置低 1: PWMREQC 有效时将管脚置高
1:0	DURATION	RW	0x3	PWMREQC 输出到 IO 管脚的脉冲宽度 0: 脉冲宽度为 4 个 PWM 时钟周期 1: 脉冲宽度为 8 个 PWM 时钟周期 2: 脉冲宽度为 16 个 PWM 时钟周期 3: 脉冲宽度为 32 个 PWM 时钟周期

9.9.4.14 PWMCFG 模块写使能寄存器（PWMCFGREGKEY）

表 9-138: PWMCFG 模块写使能寄存器（PWMCFGREGKEY）位段定义

PWMCFGREGKEY (PWMCFG Register Write-Allow Key Register)							
Access: PWMCFG->PWMCFGREGKEY			Offset: 0x003C		Default: 0x1ACCE551		
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

表 9-139: PWMCFG 模块写使能寄存器（PWMCFGREGKEY）位段描述

位段	位段名	属性	复位值	描述
31:0	KEY	RW	0x1ACCE551	写入 0x1ACCE551 解锁受保护的 PWMCFG 寄存器

10 增强的捕获模块（ECAP）

10.1 概述

增强的捕获模块（ECAP）用于对外部输入事件的时序关系进行计时，应用场景包括：

- 机电设备的速度测量（例如使用 Hall 传感器的电动车）
- 测量脉冲信号的周期和占空
- 对于占空比编码系统的解码

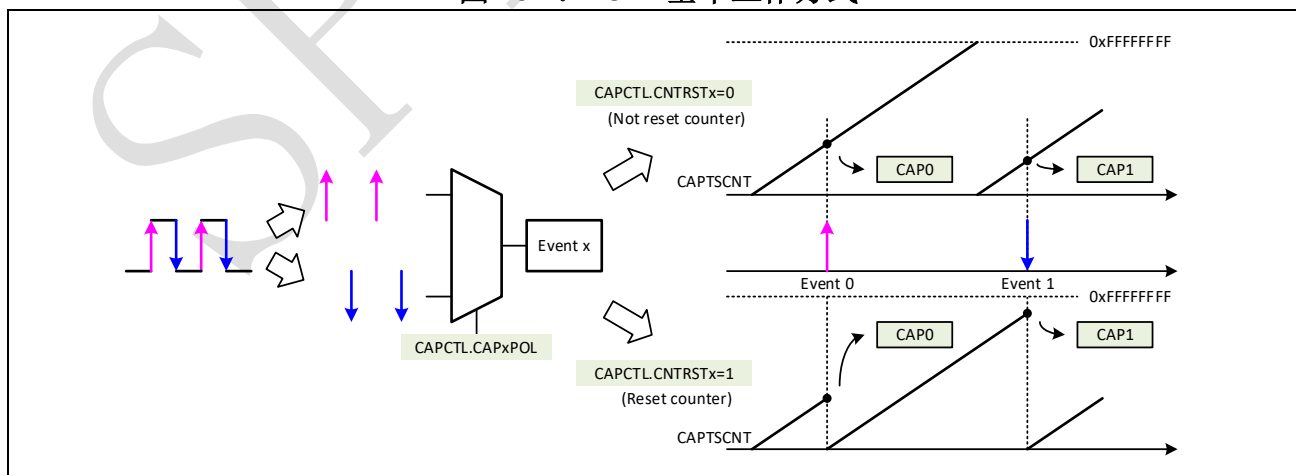
SPD1121 中的 ECAP 具备如下特性：

- 灵活的输入捕获引脚：任意 GPIO 均可被配置为捕获引脚
- 支持双引脚捕获
- 基于 32 位定时器的计数器
- 4 个 32 位时间标签捕获寄存器
- 与外部事件同步的 4 级序列器
- 4 个事件均可独立选择边沿极性（上升/下降沿）
- 4 个事件均可支持中断
- 支持 APWM 模式，能产生简单的 PWM 波形

10.2 功能描述

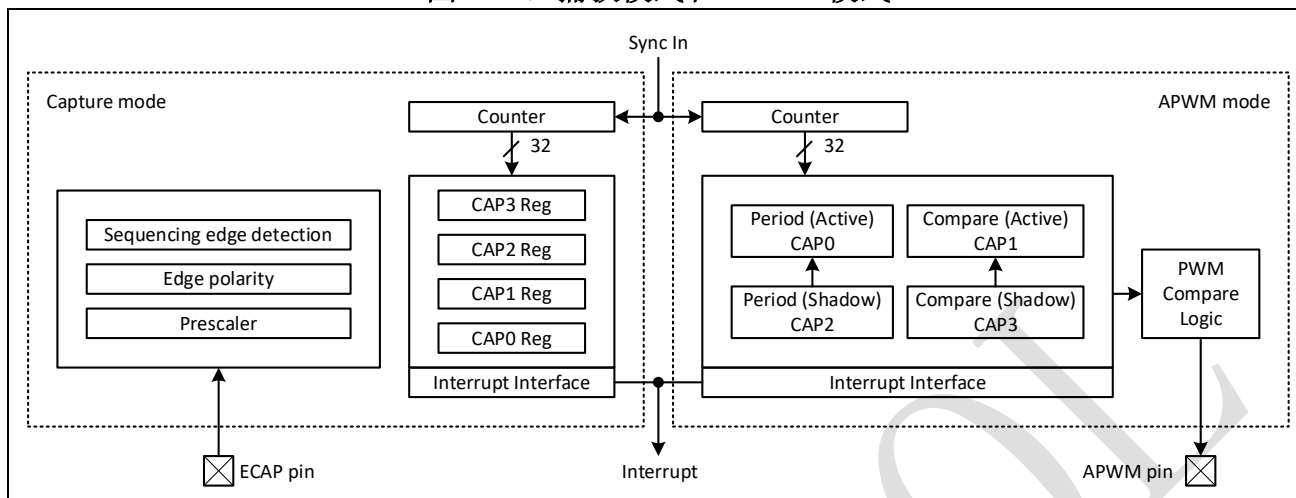
ECAP 的两种基本工作模式如图 10-1 所示。根据 CAPCTL.CAPxPOL 位段的配置，输入信号的上升沿或者下降沿被逐一编号成最多 4 个具有时序先后的捕获事件（CEVT0、CEVT1、CEVT2、CEVT3）。每个捕获事件发生时，时间戳计数器根据 CAPCTL.CNTRSTx 位段的配置决定是否复位。通过两者的配置，用户可以实现诸如信号周期和占空计算等复杂的捕获和测量功能

图 10-1: ECAP 基本工作方式



10.3 捕获模式和辅助 APWM 模式

图 10-2: 捕获模式和 APWM 模式

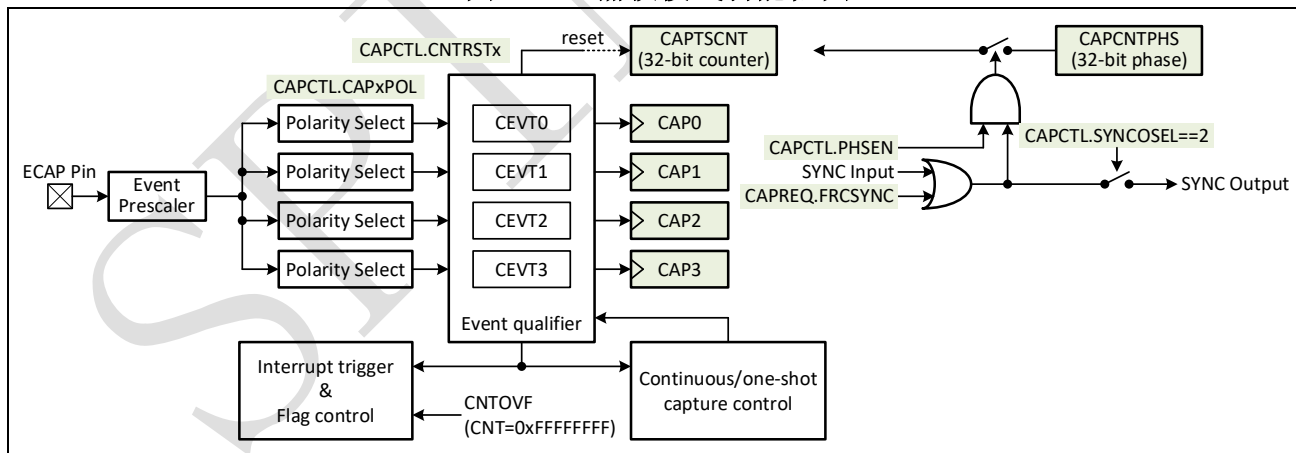


如图 10-2 所示，当 ECAP 模块不用作输入捕获时，用户可以将其用作有 32 位计数能力的单通道 PWM 模块（APWM 模式）。此时计数器处于向上计数模式，并提供非对称 PWM 波形。CAP0 和 CAP1 分别用作周期 PRD 和计数比较值 CMP 的有效寄存器，CAP2 和 CAP3 分别用作周期 PRD 和计数比较值 CMP 的影子寄存器。

10.4 捕获模式详述

捕获模式的功能框图如图 10-3 所示。其中 ECAP 输入来自于由 CAPSRCCTL 寄存器指定的 GPIO 引脚，该引脚的通道复用控制需要使能 GPIO 通道的输入。

图 10-3: 捕获模式功能框图



10.4.1 事件预分频

输入的待捕获事件（持续脉冲序列）可以进行预分频，比例为 $N = \text{CAPCTL.EVTDIV} + 1$ ，这在高频输入作为输入事件时非常有用。图 10-4 和图 10-5 分别给出了预分频的控制逻辑和波形示例。

图 10-4：事件预分频控制

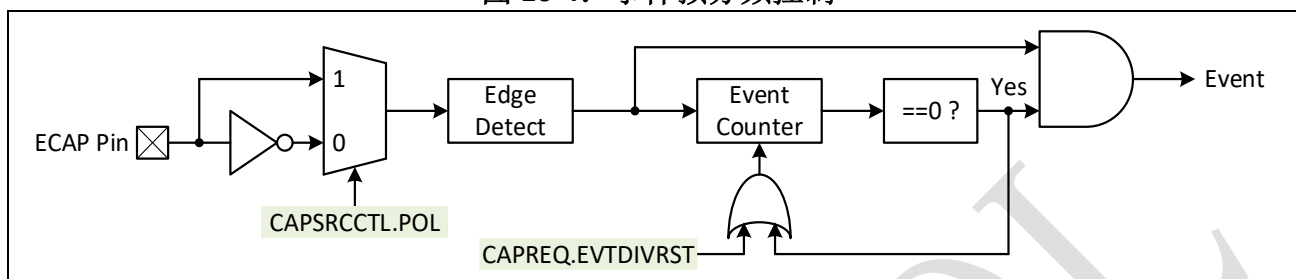
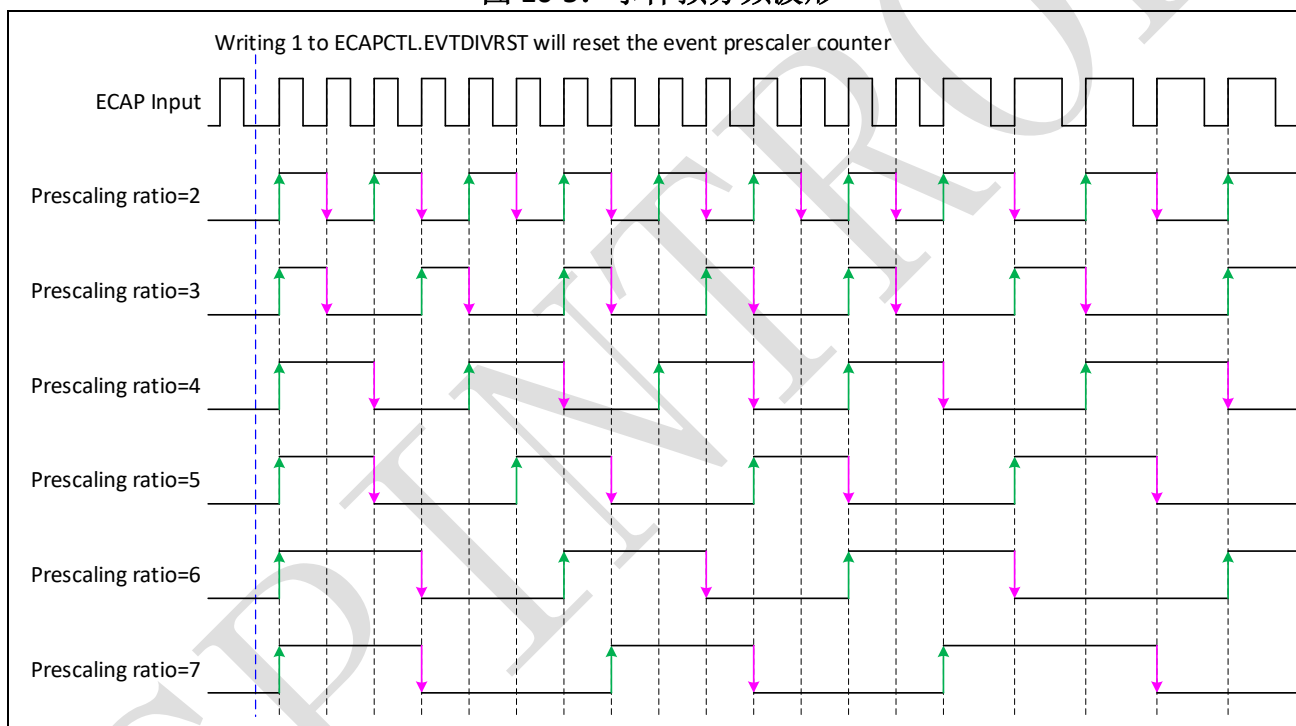


图 10-5：事件预分频波形



注意： ECAP 的事件分频在硬件上采用了类似时钟分频的电路结构，因此上电初始化需要额外的四个事件周期来初始化分频操作。在一次性捕获模式下，用户非常关心每个事件实际导致的行为，此时初始化需求就会造成问题。因此用户可以在软件上通过主动翻转 GPIO 来生成用于初始化分频的 4 个事件来解决这个问题，详见 SDK 中的示例代码。

10.4.2 边沿选择和排序编号

每个事件（最多 4 个）对应的输入信号边沿（上升沿或下降沿）都通过独立的选择器配置：

- 每个边沿事件都通过模 4 计数器按顺序编号为 CEVT0~CEVT3
- 每个事件触发的捕获，其结果由模 4 计数器控制存入对应的 CAPx 寄存器

10.4.3 单引脚捕获和双引脚捕获

CAPCTL.DUALPIN 可以使能双引脚捕获模式。

- CAPCTL.DUALPIN=0 时为单引脚捕获模式：CAP0 ~ CAP3 用于捕获单引脚。可在 CAPSRCCTL.IOSEL 设置捕获引脚的 GPIO 端口号
- CAPCTL.DUALPIN=1 时为双引脚捕获模式：使用 CAP0/CAP2 捕获引脚 0，使用 CAP1/CAP3 捕获引脚 1。可在 CAPSRCCTL.IOSEL1 设置第二个捕获引脚的 GPIO 端口号。

10.4.4 持续捕获和一次性捕获控制

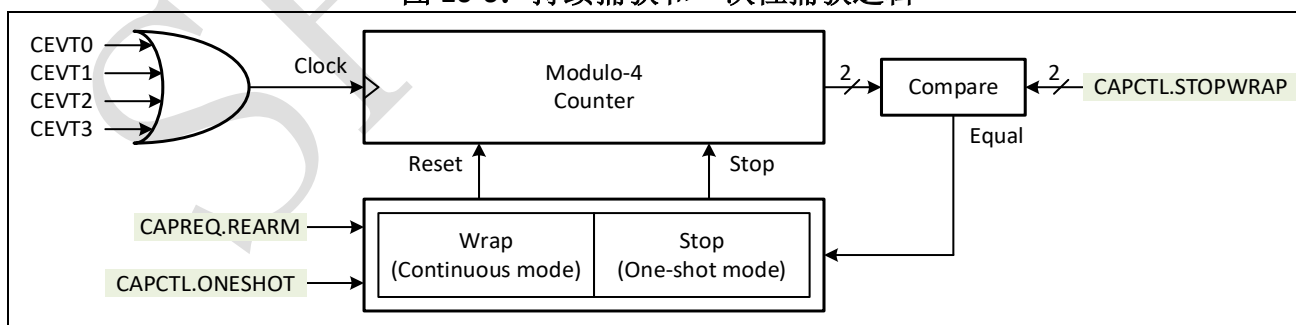
模 4（2 比特）计数器在 CEVT0~CEVT3 事件发生时逐一递增计数，它记录了当前事件的编号，并与 CAPCTL.STOPWARP 位段按照如下方式配合控制捕获的时序：

- 持续捕获模式下，模 4 事件计数器从 0 开始递增计数，一旦计数到 CAPCTL.STOPWARP 则自动归零，并继续周而复始地计数。
- 在一次性捕获模式下，模 4 事件计数器从 0 开始递增计数，一旦计数到 CAPCTL.STOPWARP 则停止计数，并阻止后续的捕获和对 CAP0~CAP3 寄存器的更新。

一次性计数模式下，可以通过软件方式（向 CAPREQ.REARM 写 1）将模 4 事件计数状态复位。模 4 事件计数复位后，ECAP 模块等待 1 到 4 个（由 CAPCTL.STOPWARP 指定）捕获事件，然后锁定模 4 事件计数和 CAP0~CAP3 的内容。新的模 4 事件计数软件复位可以让 ECAP 模块开始新一轮的捕获序列，并解锁 CAP0~CAP3 寄存器以存放新的捕获值。

持续捕获模式下，CAP0~CAP3 根据模 4 事件计数控制轮流不断存入最新的捕获值。

图 10-6：持续捕获和一次性捕获逻辑

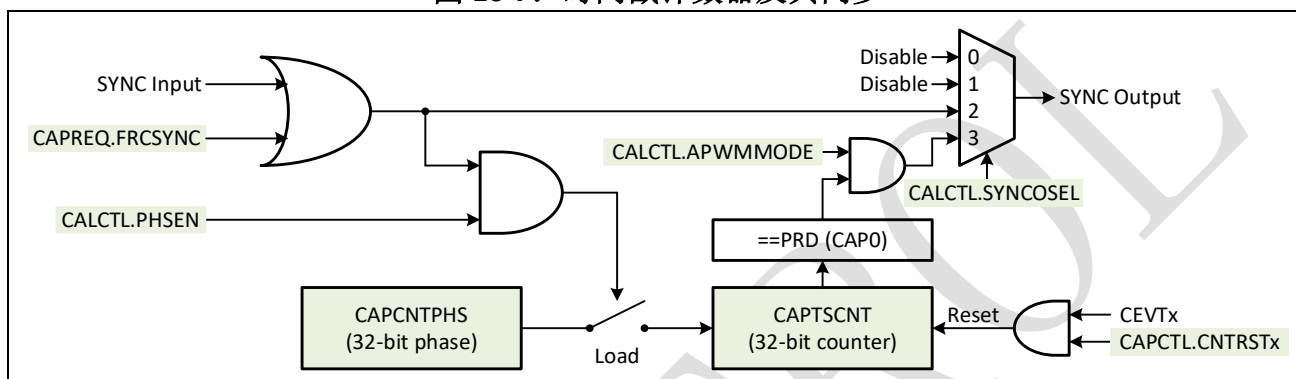


10.4.5 32 比特计数器和相位控制

32 位计数器提供了事件捕获的时间戳，并直接由 CLK_ECAP 作为计数时钟。此外，相位寄存器 CAPCNTPHS 指定了硬件或者软件同步发生时相对于其他计数器的相位偏移，这在需要指定跨模块相位偏移的 PWM 模式下非常有用。

在任意一个捕获事件发生时，均可以配置计数器复位，便于实现对时间差的捕获和计算。例如将上升沿时捕获并复位计数器，在下一个下降沿时再次捕获，则可以得到输入信号的占空。

图 10-7: 时间戳计数器及其同步



10.4.6 CAP0-CAP3 寄存器

这 4 个 32 位计数器在对应的捕获事件发生时存入当时的时间戳计数器的计数值。在一次性捕获模式下，完成 CAPCTL.STOPWARP 指定次数的捕获后，CAP0~CAP3 自动停止捕获和更新。

在 APWM 模式下，CAP0 和 CAP1 分别用作周期 PRD 和计数比较值 CMP 的有效寄存器，CAP2 和 CAP3 分别用作周期 PRD 和计数比较值 CMP 的影子寄存器。

10.4.7 中断控制

捕获事件（CEVT0-CEVT3）、APWM 事件（CNT=PRD 和 CNT=CMF）均可以触发中断。此外，计数溢出（0xFFFFFFFF→0x00000000）也可以作为独立的事件（CNTOVF）用来触发 CPU 中断。

这 7 个中断事件（CEVT0、CEVT1、CEVT2、CEVT3、CNTOVF、CNT=PRD 和 CNT=CMP）可以通过中断使能寄存器 CAPIE 独立配置是否触发 CPU 中断。

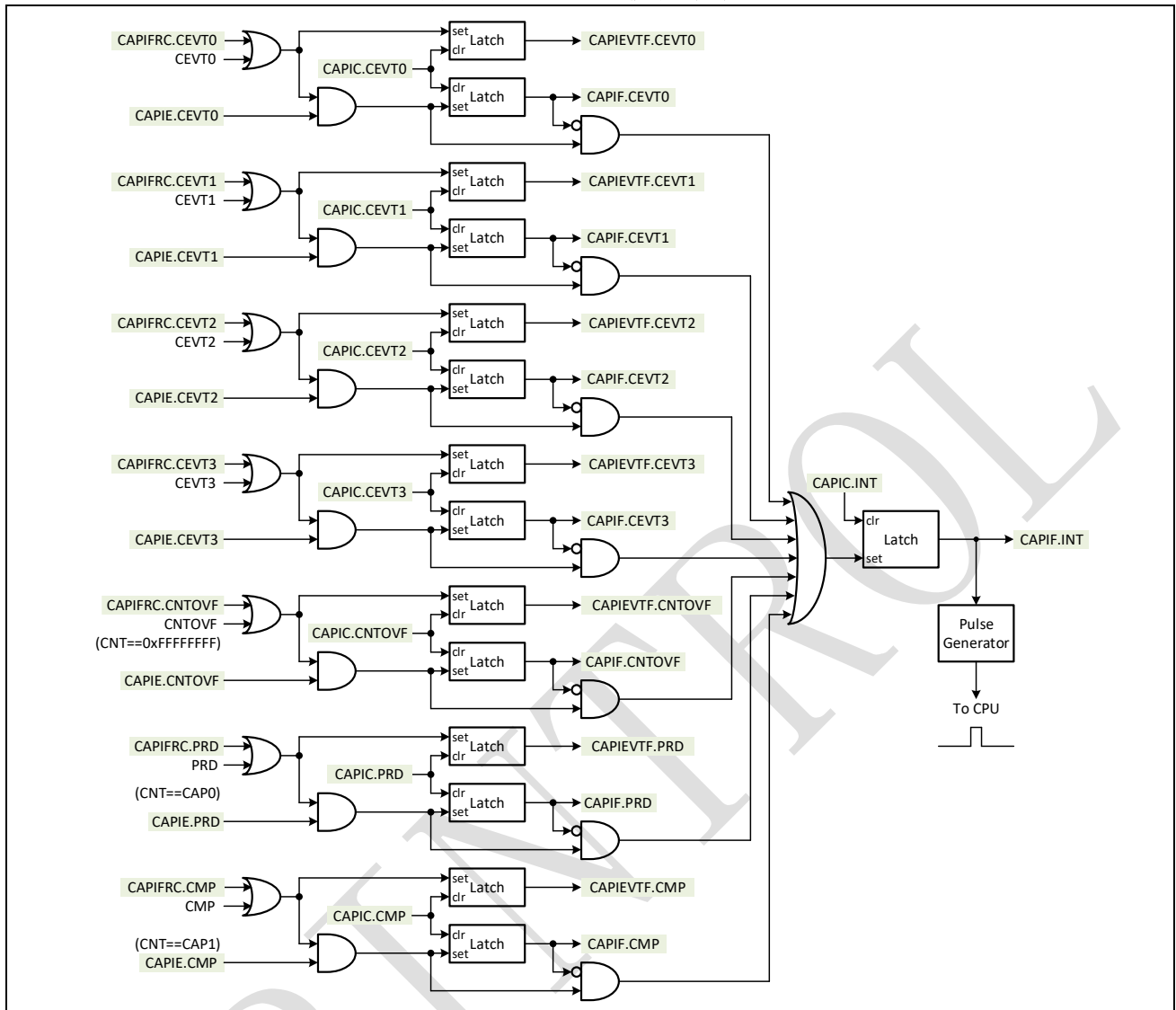
捕获模式下有效的中断事件包括：CEVT0、CEVT1、CEVT2、CEVT3、CNTOVE。

APWM 模式下有效的中断事件包括: CNT=PRD、CNT=COMP、CNTOVF。

如图 10-8 所示，每当中断事件发生时，会记录到 **CAPIEVTF** 寄存器对应位。但仅当通过 **CAPIE** 对应位使能后，该事件才会触发中断，并记录到 **CAPIF** 寄存器对应位。中断服务代码里必须通过中断清除寄存器 **CAPIC** 同时清除事件标志和全局中断标志才能保证后续事件能触发新的中断。若保持 **CAPIF** 中的事件中断标志为 1，则阻止后续同样的事件触发中断，但不影响其他事件触发中断；若保持 **CAPIF** 中的全局中断标志为 1，则阻止后续所有事件触发中断。往 **CAPIC** 的位段写 1 也会同时清除 **CAPIEVTF** 的对应位。

用户可以通过 **CAPIERC** 寄存器软件强制一个中断事件，用于代码调试。

图 10-8: ECAP 模块中断



10.4.8 影子寄存及其控制

在捕获模式下，不允许 CAP0 和 CAP1 从对应的影子寄存器 CAP2 和 CAP3 中读入值并更新；

在 APWM 模式下，影子模式被激活，用户可以选用如下任意一种写入方式：

- 直接写入：通过 CAP0 写入计数周期 PRD 和通过 CAP1 写入计数比较值 CMP，此时对应的 CAP2 和 CAP3 也立即做相应的更新。
- 影子写入：通过 CAP2 写入计数周期 PRD 的影子值和通过 CAP3 写入计数比较值 CMP 的影子值。此时对应的 CAP0 和 CAP1 保持不变。当计数到当前 CAP0（即当前有效 PRD）时，分别用 CAP2 和 CAP3 的值来更新 CAP0 和 CAP1。

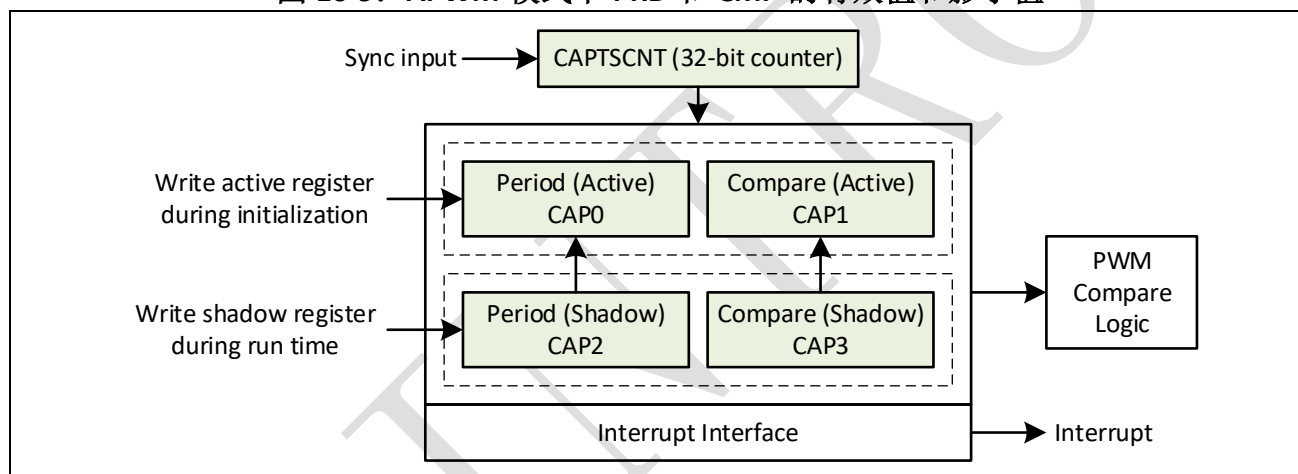
10.5 APWM 模式详述

在 APWM 模式下，32 位计数器工作在向上计数模式，此时捕获寄存器的行为如下：

- CAP0 用作周期有效寄存器（APRD）。
- CAP1 用作比较值有效寄存器（ACMP）。
- CAP2 用作周期影子寄存器（SPRD）。
- CAP3 用作比较值影子寄存器（SCMP）。
- 写 CAP0 和 CAP1 会立即同步更新 CAP2 和 CAP3。
- 写 CAP2 和 CAP3 会启动影子模式
 - 当 TSCNT = CAP0 时，将 CAP2 加载到 CAP0，以及将 CAP3 加载到 CAP1。

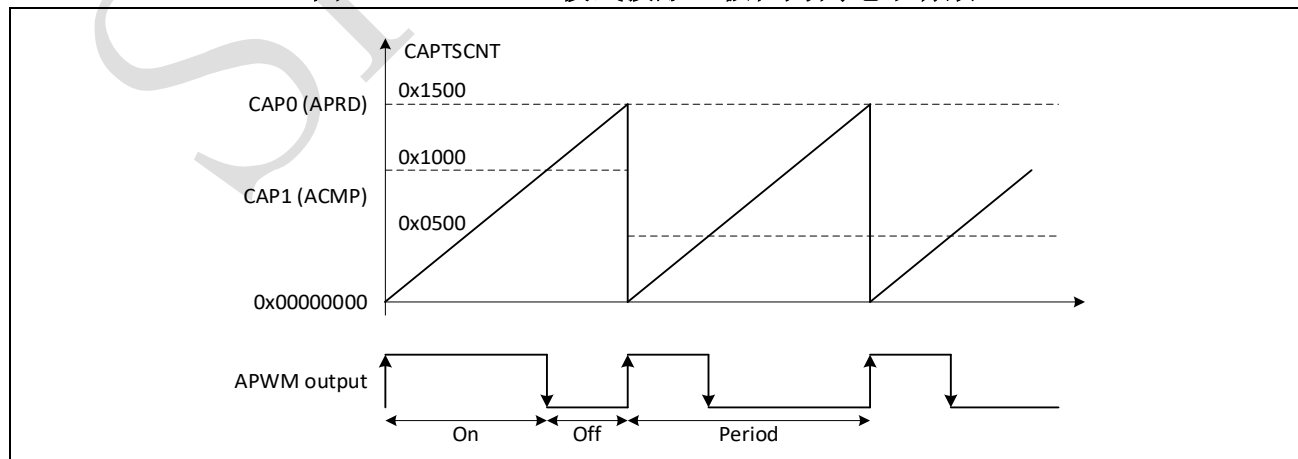
用户可以在初始化时通过 CAP0 和 CAP1 设置有效的周期和比较值，对于后续运行时的更新，用户只需要使用影子寄存器 CAP2 和 CAP3。

图 10-9: APWM 模式下 PRD 和 CMP 的有效值和影子值



如图 10-10 所示，CAPCTL.APWMPOL=1 时，APWM 极性为高电平有效。CAPTSCNT 不断地从 0 开始递增计数到 CAP0 又回到 0 继续计数。当计数到 0 时 APWM 输出变高，当计数到 CAP1 时输出变低。

图 10-10: APWM 模式波形（极性为高电平有效）



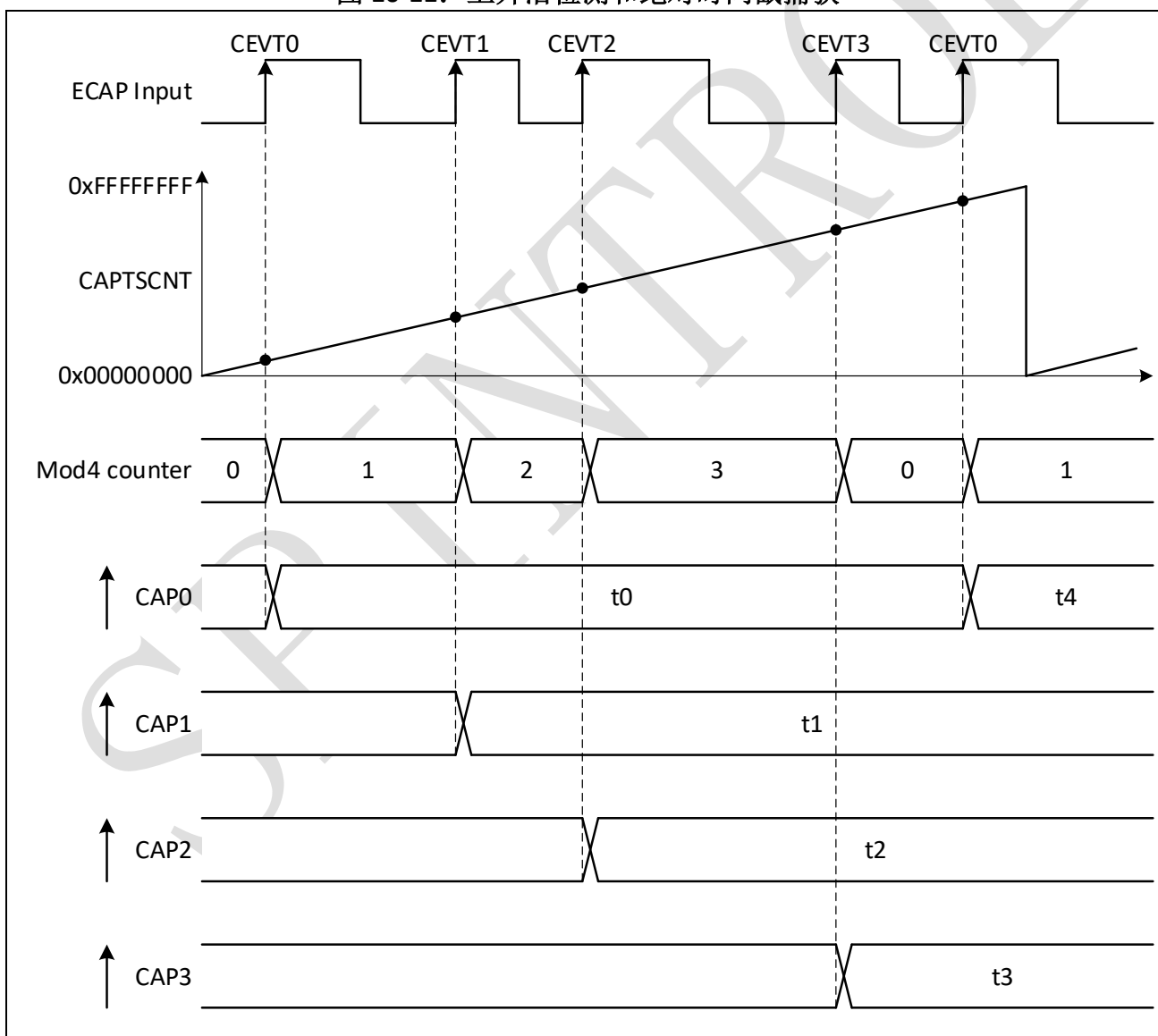
10.6 ECAP 的应用

10.6.1 上升沿触发的绝对时间戳捕获

如图 10-11 所示，持续捕获模式（模 4 事件计数自动归零持续计数）下，配置 CAPTSCNT 持续递增计数而不复位，输入上升沿视作有效事件，由此可以获取周期（及频率）信息。

每个有效事件下，CAPTSCNT 当前值被作为时间戳捕获到对应的 CAPx 寄存器，然后模 4 事件计数加 1。当计数到 0xFFFFFFF 时自动回到 0x00000000 继续计数（未在图 10-11 中示例给出）。同时，计数溢出标志 CNTOVF 被置位，如若使能，还将触发 CPU 中断。图中标记了各个有效事件的时间点，从中可以看出每到 CEVT3 时 4 个捕获已经依次完成，因此可以选择 CEVT3 作为中断触发源，便于读取所有 CAPx 寄存器的值。输入信号的各个周期可以依次计算为 CAP1 – CAP0、CAP2 – CAP1、CAP3 – CAP2。

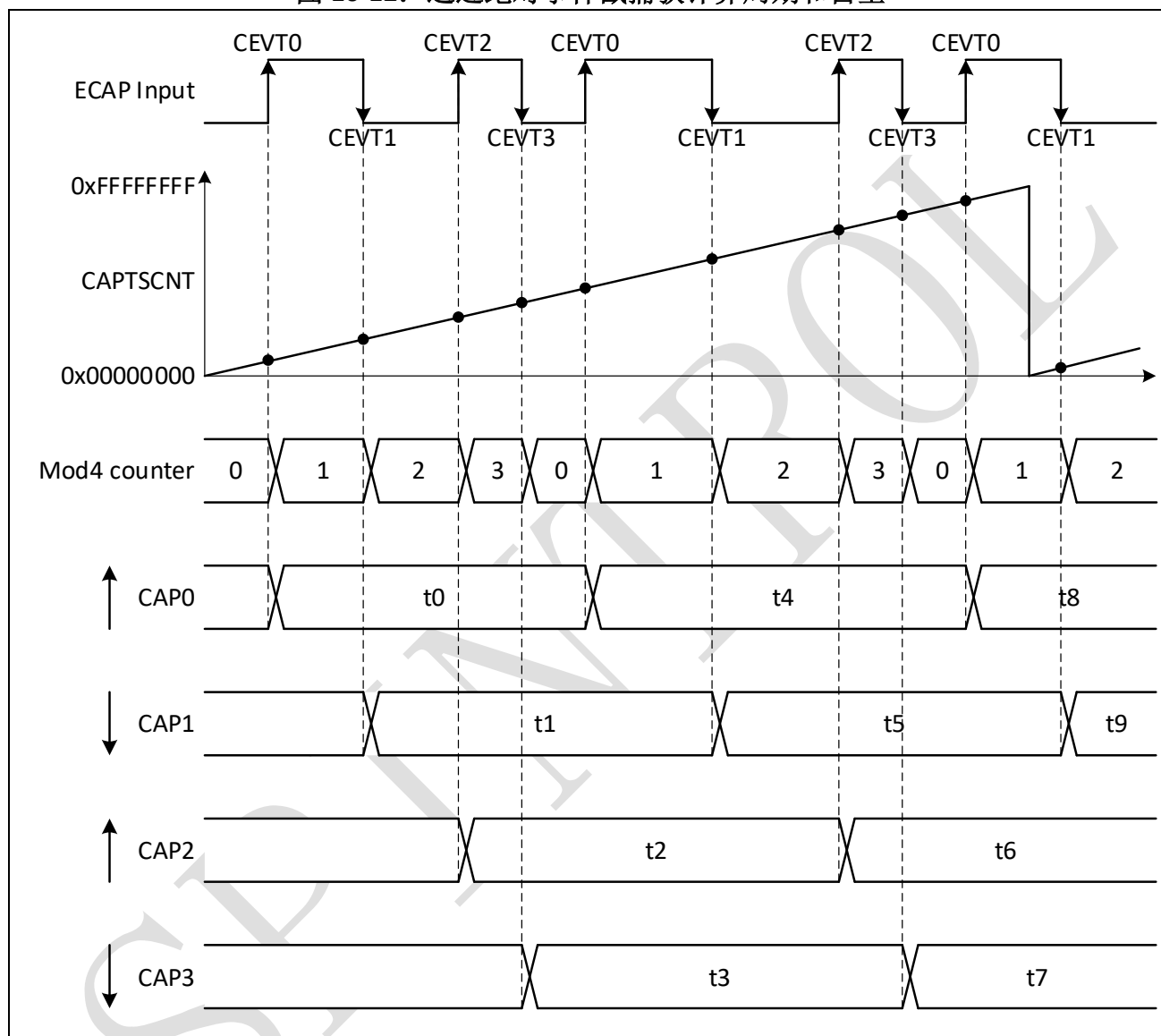
图 10-11：上升沿检测和绝对时间戳捕获



10.6.2 由绝对时间戳捕获周期和占空

图 10-12 所示情形与上例几乎相同，只是捕获事件改为上升沿和下降沿交替，由此给出了周期和占空的信息。例如：第一个周期为 $T1 = t2 - t0$ ，其中高电平占空为 $H1 = t1 - t0$ ，低电平占空为 $L1 = t2 - t1$ 。

图 10-12：通过绝对事件戳捕获计算周期和占空

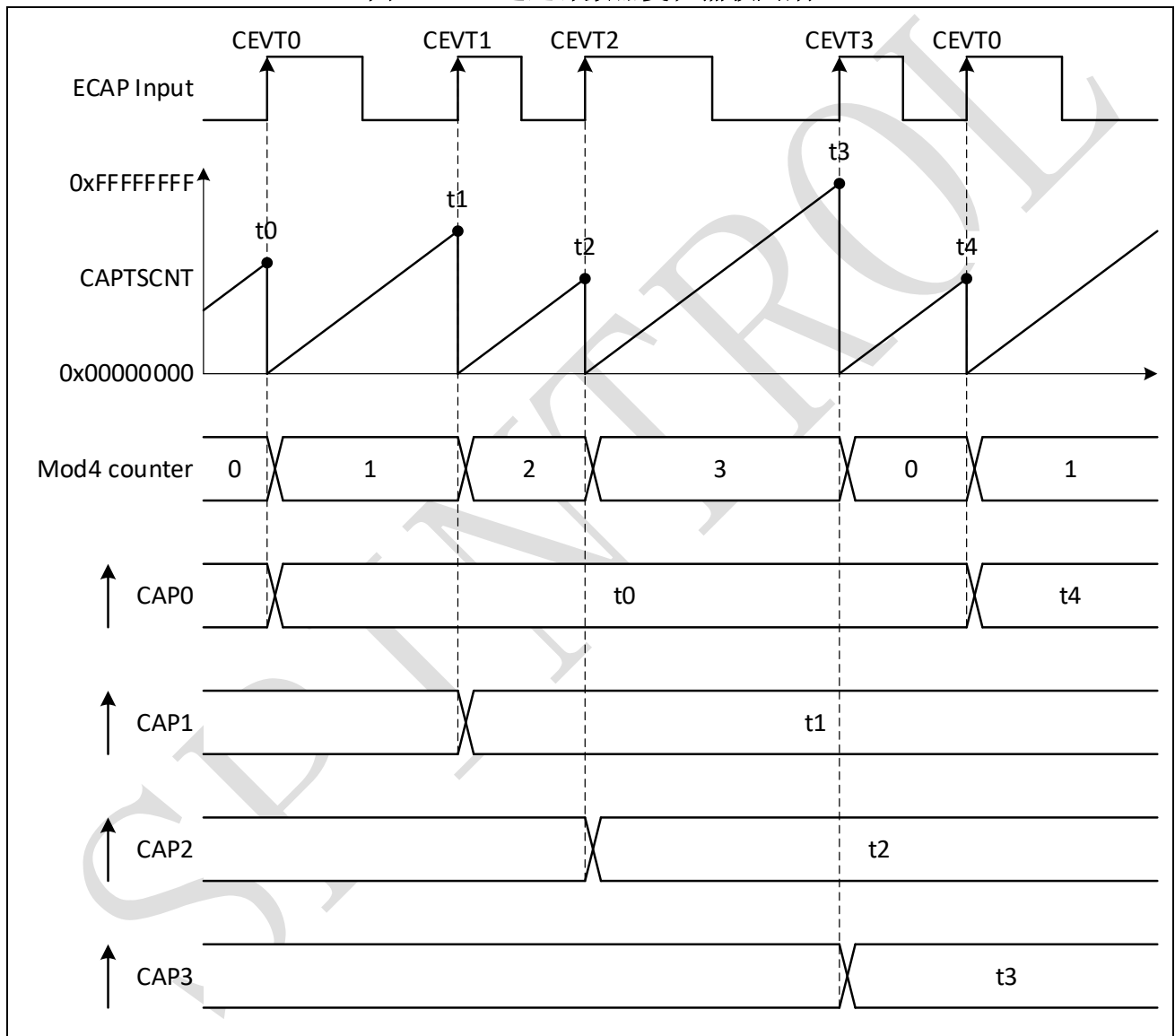


10.6.3 通过计数器复位捕获相对时间差

图 10-13 示例了 ECAP 如何从输入脉冲序列里获取时间差的信息。此处同样采用了持续捕获的模式并设置事件为上升沿有效。所不同的是，每次事件发生时，在捕获计数器当前值后都会复位计数器。

可以看出，这个方式的好处是 CAPx 寄存器的内容已经是所关心的周期信息而不需要 CPU 参与计算，即周期 $T1 = t1$ ， $T2 = t2$ 等。从图中还可以看出，CEVT0 适于作为中断触发源，因为此时 T1、T2、T3、T4 的信息都已经成功获得。

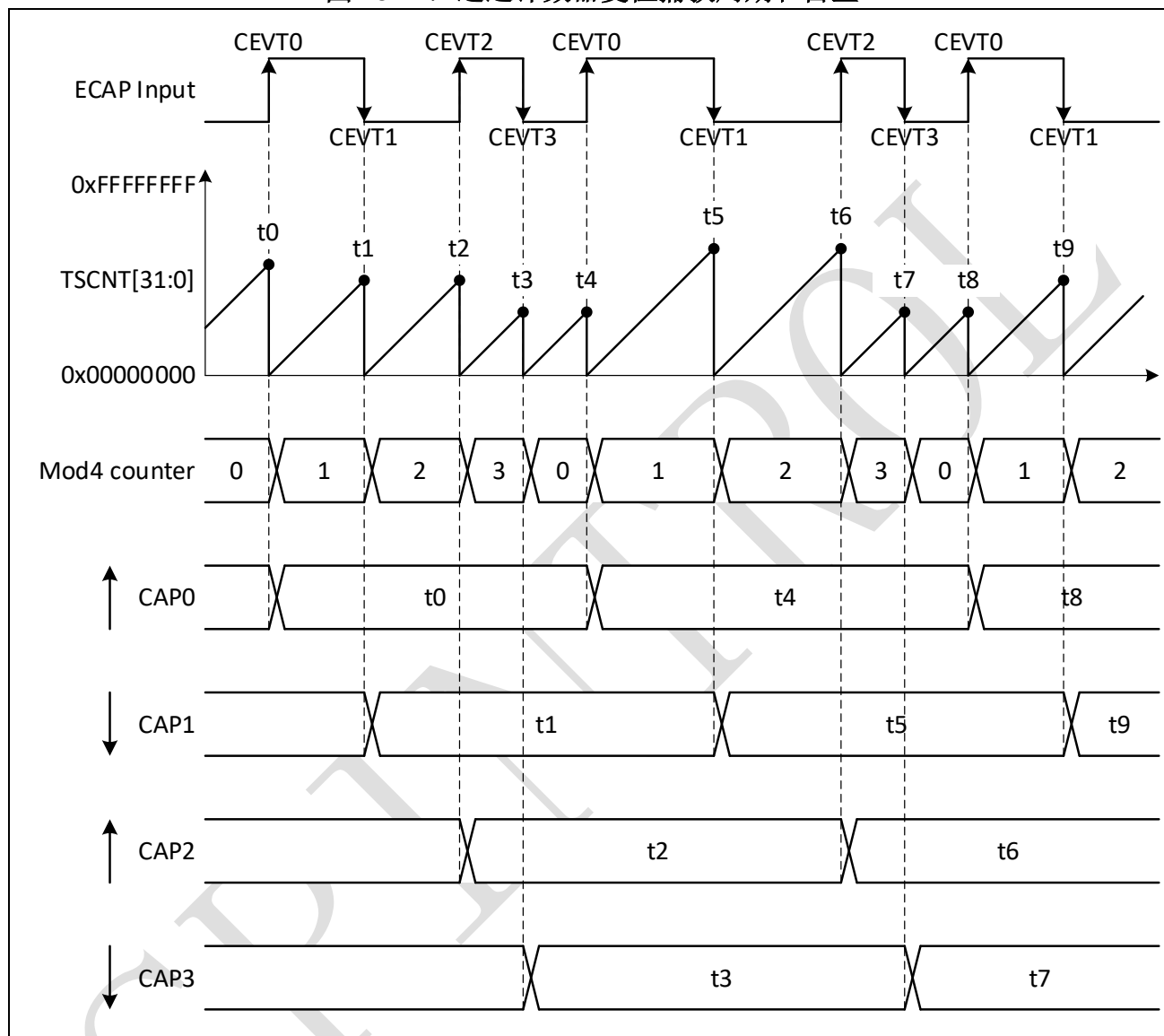
图 10-13：通过计数器复位捕获周期



10.6.4 相对事件模式捕获周期和占空

图 10-14 所示工作模式与上例几乎相同，只是捕获事件改为上升沿和下降沿交替，由此给出周期和占空信息。例如：第一个 $T_1 = t_1 + t_2$ ，其中高电平占空 t_1 ，低电平占空 t_2 。

图 10-14：通过计数器复位捕获周期和占空

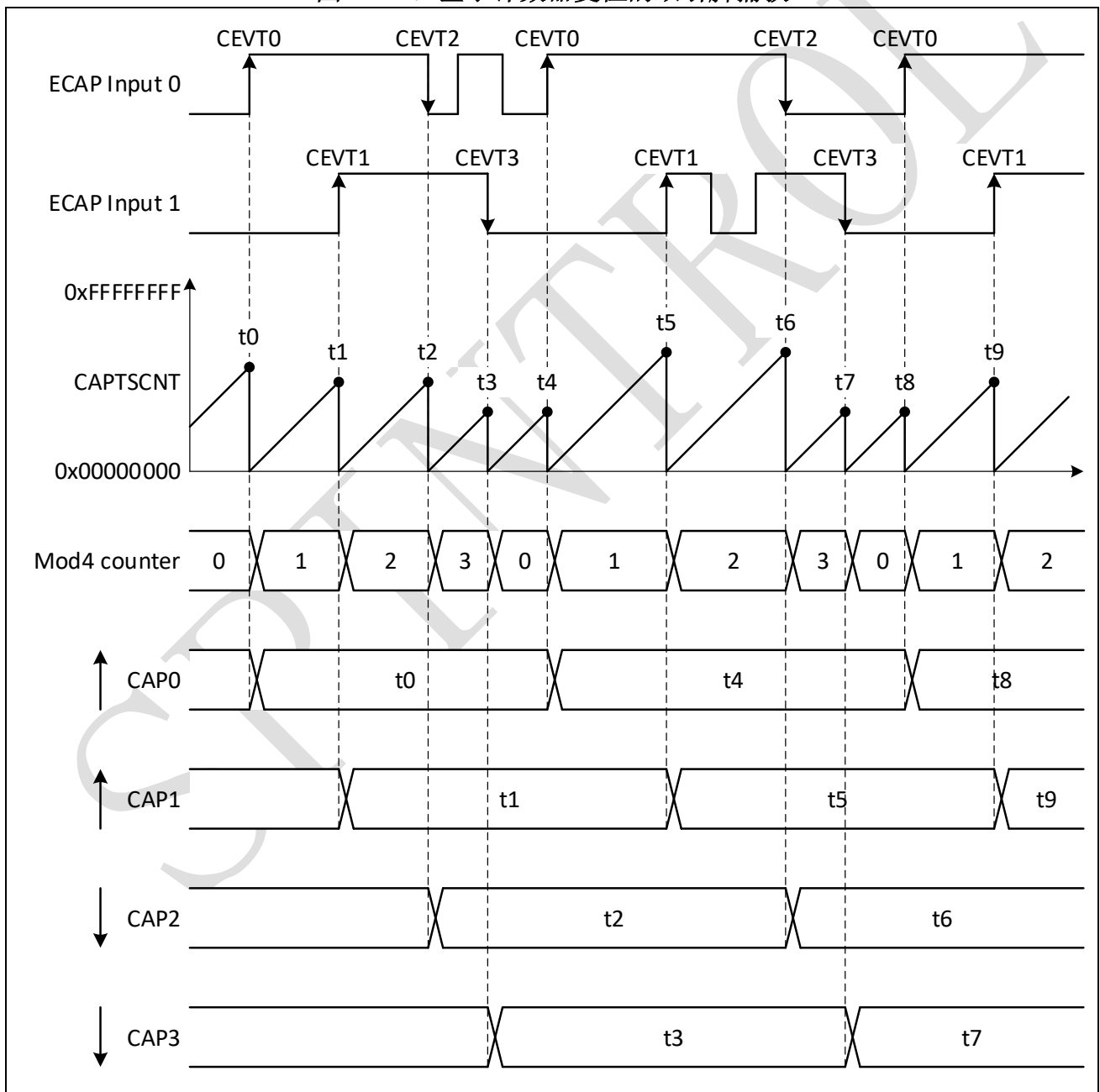


10.6.5 基于计数器复位的双引脚捕获

图 10-15 示例了双引脚捕获的波形。其中 CAP0 和 CAP2 分别用于捕获引脚 0 的上升和下降沿，CAP1 和 CAP3 分别用于捕获引脚 1 的上升和下降沿，并通过计数器复位捕获时间差。由此：

- CAP1 捕获了引脚 0 上升沿到相邻引脚 1 上升沿的时间差；
- CAP2 捕获了引脚 1 上升沿到相邻引脚 0 下降沿的时间差；
- CAP3 捕获了引脚 0 下降沿到相邻引脚 1 下降沿的时间差；
- CAP0 捕获了引脚 1 下降沿到相邻引脚 0 上升沿的时间差。

图 10-15：基于计数器复位的双引脚捕获



10.7 寄存器

10.7.1 ECAP 寄存器列表

表 10-1: ECAP 模块基地址

外设模块	基地址
ECAP	0x4000 A000

表 10-2: ECAP 寄存器列表

寄存器	偏移地址	描述	复位值
<u>CAPSRCCTL*</u>	0x00	捕获信号输入控制寄存器	0x007F007F
<u>CAPSYNCICTL*</u>	0x04	同步信号输入控制寄存器	0x0000007F
<u>CAPTSCNT*</u>	0x08	时间戳计数寄存器	0x00000000
<u>CAPCNTPHS*</u>	0x0C	时间戳计数相位寄存器	0x00000000
<u>CAP0*</u>	0x10	捕获值 0 寄存器	0x00000000
<u>CAP1*</u>	0x14	捕获值 1 寄存器	0x00000000
<u>CAP2*</u>	0x18	捕获值 2 寄存器	0x00000000
<u>CAP3*</u>	0x1C	捕获值 3 寄存器	0x00000000
<u>CAPCTL*</u>	0x20	捕获控制寄存器	0x000C0000
<u>CAPIF</u>	0x24	捕获中断标志寄存器	0x00000000
<u>CAPIE*</u>	0x28	捕获中断使能寄存器	0x00000000
<u>CAPIC</u>	0x2C	捕获中断清除寄存器	0x00000000
<u>CAPIFRC*</u>	0x30	捕获中断强制寄存器	0x00000000
<u>CAPREGKEY</u>	0x34	ECAP 模块写使能寄存器	0x1ACCE551
<u>CAPIEVTF</u>	0x38	捕获中断事件标志寄存器	0x00000000

注意： 由*标识的寄存器仅当 CAPREGKEY=0x1ACCE551 时才可以改写。

10.7.2 ECAP 寄存器

表 10-3 到表 10-30 给出了 ECAP 模块寄存器的定义。

10.7.2.1 捕获信号输入控制寄存器（CAPSRCCTL）

表 10-3：捕获信号输入控制寄存器（CAPSRCCTL）位段定义

CAPSRCCTL (Capture Source Input Control Register)							
Access: ECAP->CAPSRCCTL Offset: 0x0000 Default: 0x007F007F							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED	POL1	IOSEL1					
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	POL	IOSEL					

表 10-4：捕获信号输入控制寄存器（CAPSRCCTL）位段描述

位段	位段名	属性	复位值	描述
31:23	RESERVED_31_23	RO	0x0	保留
22	POL1	RW	0x1	在双引脚模式的第二个引脚输入信号极性输入 0: 输入低时有效 1: 输入高时有效
21:16	IOSEL1	RW	0x3F	在双引脚模式下，第二个捕获引脚的 GPIO 端口号（只能为偶数）
15:7	RESERVED_15_7	RO	0x0	保留
6	POL	RW	0x1	输入信号极性 0: 输入低时有效 1: 输入高时有效
5:0	IOSEL	RW	0x3F	作为捕获信号输入的 GPIO 端口号

10.7.2.2 同步信号输入控制寄存器（CAPSYNCICCTL）

表 10-5：同步信号输入控制寄存器（CAPSYNCICCTL）位段定义

CAPSYNCICCTL (Capture Synchronization Source Input Control Register)							
Access: ECAP->CAPSYNCICCTL Offset: 0x0004 Default: 0x0000007F							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	POL	IOSEL					

表 10-6：同步信号输入控制寄存器（CAPSYNCICCTL）位段描述

位段	位段名	属性	复位值	描述
31:7	RESERVED_31_7	RO	0x0	保留
6	POL	RW	0x1	同步信号极性 0: 输入低时触发同步 1: 输入高时触发同步
5:0	IOSEL	RW	0x3F	作为同步信号输入的 GPIO 端口号

10.7.2.3 时间戳计数寄存器（CAPTSCNT）

表 10-7：时间戳计数寄存器（CAPTSCNT）位段定义

CAPTSCNT (Time-Stamp Counter Register)							
Access: ECAP->CAPTSCNT Offset: 0x0008 Default: 0x00000000							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 10-8：时间戳计数寄存器（CAPTSCNT）位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	32 位计数器，用作捕获时的时间戳

10.7.2.4 时间戳相位寄存器 (CAPCNTPHS)

表 10-9: 时间戳相位寄存器 (CAPCNTPHS) 位段定义

CAPCNTPHS (Counter Phase Offset Register)							
Access: ECAP->CAPCNTPHS				Offset: 0x000C		Default: 0x00000000	
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 10-10: 时间戳相位寄存器 (CAPCNTPHS) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	时间戳计数器在同步时的相位值，可以编程为相位滞后/超前

10.7.2.5 捕获值 0 寄存器 (CAP0)

表 10-11: 捕获值 0 寄存器 (CAP0) 位段定义

CAP0 (Capture Register 0)							
Access: ECAP->CAP0				Offset: 0x0010		Default: 0x00000000	
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 10-12: 捕获值 0 寄存器 (CAP0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	捕获模式 - 该寄存器在捕获事件 0 发生时，更新为时间戳计数器 (CAPTSCNT) 的值。 APWM 模式 - 写入该寄存器将设置 PRD 的有效值，并使用相同的值更新 CAP2。这模拟了快速模式。 - 在 CAPTSCNT=PRD (CAP0) 时，该寄存器更新为 CAP2 的值。这模拟了影子模式。

10.7.2.6 捕获值 1 寄存器 (CAP1)

表 10-13: 捕获值 1 寄存器 (CAP1) 位段定义

CAP1 (Capture Register 1)							
Access: ECAP->CAP1		Offset: 0x0014		Default: 0x00000000			
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 10-14: 捕获值 1 寄存器 (CAP1) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	捕获模式 - 该寄存器在捕获事件 1 发生时，更新为时间戳计数器 (CAPTSCNT) 的值。 APWM 模式 - 写入该寄存器将设置 PRD 的有效值，并使用相同的值更新 CAP3。这模拟了快速模式。 - 在 CAPTSCNT=PRD (CAP1) 时，该寄存器更新为 CAP3 的值。这模拟了影子模式。

10.7.2.7 捕获值 2 寄存器 (CAP2)

表 10-15: 捕获值 2 寄存器 (CAP2) 位段定义

CAP2 (Capture Register 2)							
Access: ECAP->CAP2		Offset: 0x0018		Default: 0x00000000			
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 10-16: 捕获值 2 寄存器 (CAP2) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	捕获模式 - 该寄存器在捕获事件 2 发生时，更新为时间戳计数器 (CAPTSCNT) 的值。 APWM 模式 - 写入该寄存器将设置 PRD 的影子值。当 CAPTSCNT=PRD (CAPO) 时，CAPO 将被更新为该寄存器的值。这模拟了影子模式。

10.7.2.8 捕获值 3 寄存器 (CAP3)

表 10-17: 捕获值 3 寄存器 (CAP3) 位段定义

CAP3 (Capture Register 3)							
Access: ECAP->CAP3		Offset: 0x001C		Default: 0x00000000			
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 10-18: 捕获值 3 寄存器 (CAP3) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	捕获模式 - 该寄存器在捕获事件 3 发生时，更新为时间戳计数器 (CAPTSCNT) 的值。 APWM 模式 - 写入该寄存器将设置 CMP 的影子值。当 CAPTSCNT=PRD (CAPO) 时，CAP1 将被更新为该寄存器的值。这模拟了影子模式。

10.7.2.9 捕获控制寄存器 (CAPCTL)

表 10-19: 捕获控制寄存器 (CAPCTL) 位段定义

CAPCTL (Capture Control Register)							
Access: ECAP->CAPCTL		Offset: 0x0020		Default: 0x000C0000			
31 DUALPIN	30 EVTDIVRST	29 DBGRUN	28	27 APWMPOL	26 APWMMODE	25 FRCSYNC	24 SYNCOSEL
23 SYNCOSEL	22 PHSEN	21 TSCNTRUN	20 REARM	19 STOPWRAP	18	17 ONESHOT	16 EVTDIV
15	14	13	12	11	10	9	8 CAPLDEN
EVTDIV							
7 CNTRST3	6 CAP3POL	5 CNTRST2	4 CAP2POL	3 CNTRST1	2 CAP1POL	1 CNTRST0	0 CAPOPOL

表 10-20: 捕获控制寄存器 (CAPCTL) 位段描述

位段	位段名	属性	复位值	描述
31	DUALPIN	RW	0x0	双引脚捕获模式 0: CAP0 ~ CAP3 用于捕获单引脚 1: 使用 CAP0/CAP2 捕获引脚 0, 使用 CAP1/CAP3 捕获引脚 1
30	EVTDIVRST	W1C	0x0	事件预分器重置请求 0: 写 0 无效, 总是回读 0 1: 写 1 重置事件分频器, 本位段自动清零
29:28	DBGRUN	RW	0x0	由于调试或异常错误导致的 CPU 停止事件发生时的时间戳计数器的运行方式 0: 立即停止 1: 计数到 0 时停止 2: 保持正常计数 3: 保持正常计数
27	APWMPOL	RW	0x0	APWM 输出信号极性。本位段仅在 APWM 模式有效 0: TSCNT 小于 CMP 时, APWM 输出低电平; TSCNT 大于等于 CMP 且小于 PRD 时, 输出高电平 1: TSCNT 小于 CMP 时, APWM 输出高电平; TSCNT 大于等于 CMP 且小于 PRD 时, 输出低电平
26	APWMMODE	RW	0x0	捕获和 APWM 模式选择 0: 捕获模式 在该模式下: - 关闭 CAPTSCNT 计数到 PRD 时的自动复位 - 关闭 CAP2/CAP3 对 CAP0/CAP1 的影子功能 - 允许用户使能 CAP0~CAP3 捕获时间戳计数 - CAPx 输入端口生效

位段	位段名	属性	复位值	描述
				1: APWM 模式 在该模式下： - CAPTSCNT 计数到 PRD 时复位为 0 继续计数 - 开启 CAP2/CAP3 对 CAP0/CAP1 的影子功能 - 禁止捕获时间戳计数到 CAP0~CAP3 寄存器 - APWMx 输出端口生效
25	FRCSYNC	W1S	0x0	软件强制时间戳计数同步 0: 写 0 无效，总是读回 0 1: 写 1 将 CAPTSCNT 重置为 CAPCNTPHS，本位段自动清零
24:23	SYNCOSEL	RW	0x0	同步输出选择 0: 禁用同步输出 1: 禁用同步输出 2: 选择同步输入信号作为输出（直通） 3: 选择 CNT = PRD 时间作为同步输出
22	PHSEN	RW	0x0	相位同步使能 0: 关闭相位同步 1: 使能相位同步，在输入同步事件或者软件强制的同步事件发生时将 CAPTSCNT 重置为 CAPCNTPHS
21	TSCNTRUN	RW	0x0	时间戳计数器运行控制 0: 停止计数 1: 保持计数
20	REARM	W1S	0x0	重置模 4 事件计数器 0: 写 0 无效，总是读回 0 1: 写 1 重置模 4 事件计数器为 0，本位段自动清零
19:18	STOPWRAP	RW	0x3	一次性捕获模式下的停止值或持续捕获模式下的环回值 0: 一次性捕获模式下事件 0 发生后即停止 持续捕获模式下事件 0 发生后重置事件计数 1: 一次性捕获模式下事件 1 发生后即停止 持续捕获模式下事件 1 发生后重置事件计数 2: 一次性捕获模式下事件 2 发生后即停止 持续捕获模式下事件 2 发生后重置事件计数 3: 一次性捕获模式下事件 3 发生后即停止 持续捕获模式下事件 3 发生后重置事件计数
17	ONESHOT	RW	0x0	一次性捕获模式使能 本位段仅在捕获模式下有效 0: 持续捕获模式 1: 一次性捕获模式使能

位段	位段名	属性	复位值	描述
16:9	EVTDIV	RW	0x0	事件滤波预分频 每 EVTDIV+1 次事件触发一次捕获
8	CAPLDEN	RW	0x0	使能 CAP0~CAP3 的捕获功能 0: 关闭 1: 使能
7	CNTRST3	RW	0x0	捕获事件 3 发生时复位时间戳计数 (CAPTSCNT) 0: 捕获事件 3 发生时不复位时间戳计数 (绝对时间模式) 1: 捕获事件 3 发生时复位时间戳计数 (相对时间模式)
6	CAP3POL	RW	0x0	捕获事件 3 极性选择 0: 输入下降沿时触发捕获事件 3 1: 输入上升沿时触发捕获事件 3
5	CNTRST2	RW	0x0	捕获事件 2 发生时复位时间戳计数 (CAPTSCNT) 0: 捕获事件 2 发生时不复位时间戳计数 (绝对时间模式) 1: 捕获事件 2 发生时复位时间戳计数 (相对时间模式)
4	CAP2POL	RW	0x0	捕获事件 2 极性选择 0: 输入下降沿时触发捕获事件 2 1: 输入上升沿时触发捕获事件 2
3	CNTRST1	RW	0x0	捕获事件 1 发生时复位时间戳计数 (CAPTSCNT) 0: 捕获事件 1 发生时不复位时间戳计数 (绝对时间模式) 1: 捕获事件 1 发生时复位时间戳计数 (相对时间模式)
2	CAP1POL	RW	0x0	捕获事件 1 极性选择 0: 输入下降沿时触发捕获事件 1 1: 输入上升沿时触发捕获事件 1
1	CNTRST0	RW	0x0	捕获事件 0 发生时复位时间戳计数 (CAPTSCNT) 0: 捕获事件 0 发生时不复位时间戳计数 (绝对时间模式) 1: 捕获事件 0 发生时复位时间戳计数 (相对时间模式)
0	CAP0POL	RW	0x0	捕获事件 0 极性选择 0: 输入下降沿时触发捕获事件 0 1: 输入上升沿时触发捕获事件 0

10.7.2.10 捕获中断标志寄存器（CAPIF）

表 10-21: 捕获中断标志寄存器（CAPIF）位段定义

CAPIF (Capture Interrupt Flag Register)							
Access: ECAP->CAPIF		Offset: 0x0024		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
INT	CMP	PRD	CNTOVF	CEVT3	CEVT2	CEVT1	CEVT0

表 10-22: 捕获中断标志寄存器（CAPIF）位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	INT	RO	0x0	全局中断标志 0: 中断未触发 1: 中断触发。本位段清零前，将不会产生新的中断。
6	CMP	RO	0x0	计数比较相等事件标志 本位段仅在 APWM 模式下有效。 0: 事件未发生 1: 事件发生。本位段清零前，新的计数比较相等事件将不会触发新的中断
5	PRD	RO	0x0	计数到 PRD 事件标志 本位段仅在 APWM 模式下有效。 0: 事件未发生 1: 事件发生。本位段清零前，新的计数到 PRD 事件将不会触发新的中断。
4	CNTOVF	RO	0x0	计数溢出事件（计数到 0xFFFFFFFF）标志 本位段在 APWM 模式和 CAP 模式下均有效。 0: 事件未发生 1: 事件发生。本位段清零前，新的计数溢出事件将不会触发新的中断。
3	CEVT3	RO	0x0	捕获事件 3 标志 本位段仅在捕获模式下有效。 0: 事件未发生 1: 事件发生。本位段清零前，新的捕获事件 3 将不会触发新的中断。

位段	位段名	属性	复位值	描述
2	CEVT2	RO	0x0	捕获事件 2 标志 本位段仅在捕获模式下有效。 0: 事件未发生 1: 事件发生。本位段清零前，新的捕获事件 2 将不会触发新的中断。
1	CEVT1	RO	0x0	捕获事件 1 标志 本位段仅在捕获模式下有效。 0: 事件未发生 1: 事件发生。本位段清零前，新的捕获事件 1 将不会触发新的中断。
0	CEVT0	RO	0x0	捕获事件 0 标志 本位段仅在捕获模式下有效。 0: 事件未发生 1: 事件发生。本位段清零前，新的捕获事件 0 将不会触发新的中断。

10.7.2.11 捕获中断使能寄存器（CAPIE）

表 10-23: 捕获中断使能寄存器（CAPIE）位段定义

CAPIE (Capture Interrupt Enable Register)							
Access: ECAP->CAPIE Offset: 0x0028 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	CMP	PRD	CNTOVF	CEVT3	CEVT2	CEVT1	CEVT0

表 10-24: 捕获中断使能寄存器（CAPIE）位段描述

位段	位段名	属性	复位值	描述
31:7	RESERVED_31_7	RO	0x0	保留
6	CMP	RW	0x0	计数比较相等事件中断使能 0: 关闭 1: 使能
5	PRD	RW	0x0	计数到 PRD 事件中断使能 0: 关闭 1: 使能
4	CNTOVF	RW	0x0	计数溢出（计数到 0xFFFFFFFF）事件中断使能 0: 关闭 1: 使能
3	CEVT3	RW	0x0	捕获事件 3 中断使能 0: 关闭 1: 使能
2	CEVT2	RW	0x0	捕获事件 2 中断使能 0: 关闭 1: 使能
1	CEVT1	RW	0x0	捕获事件 1 中断使能 0: 关闭 1: 使能
0	CEVT0	RW	0x0	捕获事件 0 中断使能 0: 关闭 1: 使能

10.7.2.12 捕获中断清除寄存器 (CAPIC)

表 10-25: 捕获中断清除寄存器 (CAPIC) 位段定义

CAPIC (Capture Interrupt Clear Register)							
Access: ECAP->CAPIC Offset: 0x002C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
INT	CMP	PRD	CNTOVF	CEVT3	CEVT2	CEVT1	CEVT0

表 10-26: 捕获中断清除寄存器 (CAPIC) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	INT	W1C	0x0	全局中断清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除中断和标志位。本位段自动清零。
6	CMP	W1C	0x0	计数比较相等标志清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除标志位。本位段自动清零。
5	PRD	W1C	0x0	计数至 PRD 标志清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除标志位。本位段自动清零。
4	CNTOVF	W1C	0x0	计数溢出标志清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除标志位。本位段自动清零。
3	CEVT3	W1C	0x0	捕获事件 3 标志清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除标志位。本位段自动清零。
2	CEVT2	W1C	0x0	捕获事件 2 标志清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除标志位。本位段自动清零。
1	CEVT1	W1C	0x0	捕获事件 1 标志清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除标志位。本位段自动清零。
0	CEVT0	W1C	0x0	捕获事件 0 标志清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除标志位。本位段自动清零。

10.7.2.13 捕获中断强制寄存器（CAPIFRC）

表 10-27: 捕获中断强制寄存器（CAPIFRC）位段定义

CAPIFRC (Capture Interrupt Force Register)							
Access: ECAP->CAPIFRC Offset: 0x0030 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	CMP	PRD	CNTOVF	CEVT3	CEVT2	CEVT1	CEVT0

表 10-28: 捕获中断强制寄存器（CAPIFRC）位段描述

位段	位段名	属性	复位值	描述
31:7	RESERVED_31_7	RO	0x0	保留
6	CMP	W1S	0x0	计数比较相等事件软件强制 0: 写 0 无效，总是读回 0 1: 写 1 强制产生一个事件。本位段自动清零。
5	PRD	W1S	0x0	计数到 PRD 事件软件强制 0: 写 0 无效，总是读回 0 1: 写 1 强制产生一个事件。本位段自动清零。
4	CNTOVF	W1S	0x0	计数溢出事件软件强制 0: 写 0 无效，总是读回 0 1: 写 1 强制产生一个事件。本位段自动清零。
3	CEVT3	W1S	0x0	捕获事件 3 软件强制 0: 写 0 无效，总是读回 0 1: 写 1 强制产生一个事件。本位段自动清零。
2	CEVT2	W1S	0x0	捕获事件 2 软件强制 0: 写 0 无效，总是读回 0 1: 写 1 强制产生一个事件。本位段自动清零。
1	CEVT1	W1S	0x0	捕获事件 1 软件强制 0: 写 0 无效，总是读回 0 1: 写 1 强制产生一个事件。本位段自动清零。
0	CEVT0	W1S	0x0	捕获事件 0 软件强制 0: 写 0 无效，总是读回 0 1: 写 1 强制产生一个事件。本位段自动清零。

10.7.2.14 ECAP 模块写保护寄存器（CAPREGKEY）

表 10-29: ECAP 模块写保护寄存器（CAPREGKEY）位段定义

CAPREGKEY (Capture Register Write-Allow Key Register)							
Access: ECAP->CAPREGKEY Offset: 0x0034 Default: 0x1ACCE551							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

表 10-30: ECAP 模块写保护寄存器（CAPREGKEY）位段描述

位段	位段名	属性	复位值	描述
31:0	KEY	RW	0x1ACCE551	写入 0x1ACCE551 解锁受保护的 ECAP 寄存器

10.7.2.15 捕获事件标志寄存器 (CAPIEVTF)

表 10-31: 捕获事件标志寄存器 (CAPIEVTF) 位段定义

CAPIEVTF (Capture Interrupt Event Flag Register)							
Access: ECAP->CAPIEVTF		Offset: 0x0038		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	CMP	PRD	CNTOVF	CEVT3	CEVT2	CEVT1	CEVT0

表 10-32: 捕获事件标志寄存器 (CAPIEVTF) 位段描述

位段	位段名	属性	复位值	描述
31:7	RESERVED_31_7	RO	0x0	保留
6	CMP	RO	0x0	CAPTSCNT=CAP1 (CMP) 事件标志 该标志仅在 APWM 模式下是有效的。 0: 事件未发生 1: 事件已发生
5	PRD	RO	0x0	CAPTSCNT=CAPO (PRD) 事件标志 该标志仅在 APWM 模式下是有效的。 0: 事件未发生 1: 事件已发生
4	CNTOVF	RO	0x0	计数器溢出 (CAPTSCNT=0xFFFFFFFF) 事件标志 该标志在捕获和 APWM 模式下是有效的。 0: 事件未发生 1: 事件已发生
3	CEVT3	RO	0x0	捕获事件 3 原始标志。 该标志仅在捕获模式下是有效的。 0: 事件未发生 1: 事件已发生
2	CEVT2	RO	0x0	捕获事件 2 原始标志。 该标志仅在捕获模式下是有效的。 0: 事件未发生 1: 事件已发生
1	CEVT1	RO	0x0	捕获事件 1 原始标志。 该标志仅在捕获模式下是有效的。 0: 事件未发生 1: 事件已发生
0	CEVT0	RO	0x0	捕获事件 0 原始标志。 该标志仅在捕获模式下是有效的。 0: 事件未发生 1: 事件已发生

11 模数转换器（ADC）

11.1 ADC 概述

SPD1121 提供一个 13 位高速模数转换器，其中，ADC 前端包含 2 个采样保持电路 SH0 和 SH1，它们可以依次运行，也可以同步运行，有 7 个外部模拟输入源可以将信号送入 ADC 采样。

ADC 模块主要包含以下功能：

- 13 位分辨率；
- 最小 250 ns 转换时间；
- 2 个差分采样保持电路；
- 模拟信号输入范围：0~3.339V；
- 内置 1.2V 参考电压；
- 输入可来自于外部 IO 输入、温度传感器、内部电源、可编程增益放大器输出、D2SBUF 的输出；
- 支持外部输入开路和短路检测；
- 数字逻辑的时钟使能控制独立可配，最高频率同 CPU 频率；
- 8 个转换控制通道 CH，触发源、输入信号、采样和转换时间、自动平均样本数均可独立配置，转换完成后产生独立的 EOC 事件；
- 支持转换优先级控制；
- 支持如下事件触发采样转换请求
 - 软件触发；
 - EOC 事件触发；
 - PWM REQ 信号；
 - TIMER REQ 信号；
 - 外部引脚请求。

11.2 ADC 架构

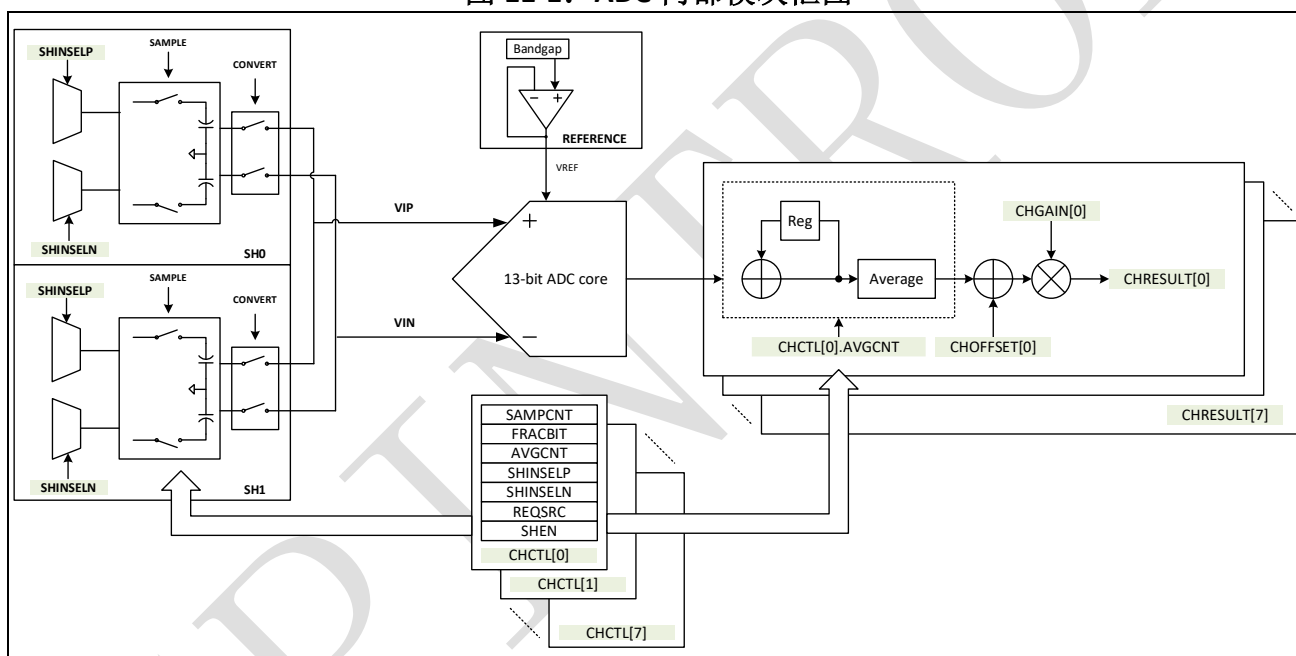
如图 11-1 所示，ADC 模拟电路包括：

- 2 个采样保持器（SH0，SH1）
- 1 个 ADC 内核
- ADC 基准电路

数字电路包括：

- 8 个控制通道配置 CH0~CH7；
- 8 个控制通道各自多次采样转换求平均、偏移和增益校准后的最终转换结果 CHRESULT0~CHRESULT7。
- 与芯片其它模块的接口（图中未展示）

图 11-1: ADC 内部模块框图



11.3 通道 CH 工作原理

ADC 基于控制通道 CH 工作，每个通道可以通过 CHCTL[x]和 ADCCTL 寄存器独立配置：

- 通过 CHCTL[x].REQSRC 配置触发通道 CH 转换请求信号；
- 通过 CHCTL[x].SHINSELP 和 CHCTL[x].SHINSELN 配置输入的差分电压信号源；
- 通过 CHCTL[x].AVGCNT 配置对输入多次采样转换并求平均的次数；
- 通过 CHCTL[x].SAMPCNT 配置采样时间；
- 通过 ADCCTL.CONVCNT 配置转换时间。

若有必要，多个控制通道可以有相同的配置，从而可以实现如下场景：

- 在不同的触发事件下对于不同的输入独立采样转换；
- 用单一触发源对同一个输入做过采样；
- 用单一触发源创建一系列不同输入的转换。

11.3.1 请求信号

每个 CH 通道都需要配置请求信号，请求信号有多种选择，可以配成相同的请求信号，也可以配置成不同的请求信号。下面是 CH 可以选择的请求信号类型：

- 软件请求信号
- 通用定时器 Timer0/1 请求信号
- 外部 GPIO 请求信号
- PWMxREQA、PWMxREQB 和 PWMxREQC (x= 2 ~ 5) 请求信号

请求信号的配置选项如表 11-1 所示，具体信息请参考 CHCTL[x]寄存器的位段定义。

表 11-1 列举了可用的请求信号。如果 REQSRC 等于 3，代表选择了 EXTIN（外部 REQ 事件）作为请求信号。如何利用寄存器 EXTREQCTL 设置 GPIO 电平作为一个外部 REQ 请求信号请参见表 11-29 和表 11-30。

注意，CHREQFLG 寄存器的对应标志位在触发请求信号时置位，并在信号得到响应（即开始执行）时清零。若通道 x 触发请求信号时 CHREQFLG.CHx 已经置位，则判定为溢出，并锁存到 CHREQOVF.CHx 标志位。溢出标志可以通过往 CHREQOVFC.CHx 写 1 清除。不仅如此，ADC 中断信号 EOC0~7 也可以被用作下一次转换的请求信号，可以通过寄存器 CHLINK 来设置。这种模式在需要一系列顺序转换时非常有用。章节 11.4 介绍了 ADC 中断信号的相关信息。

表 11-1：请求信号选择

REQSRC	请求信号
其他	无效选择
15	PWM5REQC
14	PWM5REQB
13	PWM5REQA
12	PWM4REQC
11	PWM4REQB
10	PWM4REQA
9	PWM3REQC

REQSRC	请求信号
8	PWM3REQB
7	PWM3REQA
6	PWM2REQC
5	PWM2REQB
4	PWM2REQA
3	EXTIN
2	Timer 1
1	Timer 0
0	Software

11.3.2 ADC 采样模式和通道选择

CHCTL[x].SHEN 位段控制 ADC 的采样模式，分为顺序采样和同步采样两种。

如表 11-2 所示，当 SHEN 等于 1 或 2 时，所有通道的采样模式都为顺序采样模式，通过采样保持器 SH0 或 SH1 单独采样；当 SHEN 等于 3 时，通道分别通过采样保持器 SH0 和 SH1 同步采样。

同步采样模式下，ADC 所有通道被分为 8 组，其中 CH0 和 CH1 为一组，CH2 和 CH3 为一组，以此类推。请求信号、平均次数、采样时间等配置只能通过 CH0、CH2 这些偶数编号的通道设置，即配置每组第一个通道 CH[x]后，第二个通道 CH[x+1]将保持同样配置，无需额外选择。每组第一个通道 CH[x]对应采样保持器 SH0，第二个通道 CH[x+1]对应采样保持器 SH1。

表 11-2: 采样模式描述

CHCTL[x].SHEN[3:0]	ADC 采样模式
3	SH0 和 SH1 同步采样
2	SH1 单独采样
1	SH0 单独采样
0	关闭

- 同步采样模式下，在采样结束后，转换器会先转换采样保持器 0 的信号，然后转换采样保持器 1。
- 每组 CH[x]和 CH[x+1]的转换结果被分别存储结果寄存器 CHRESULT[x]和 CHRESULT[x+1]中。

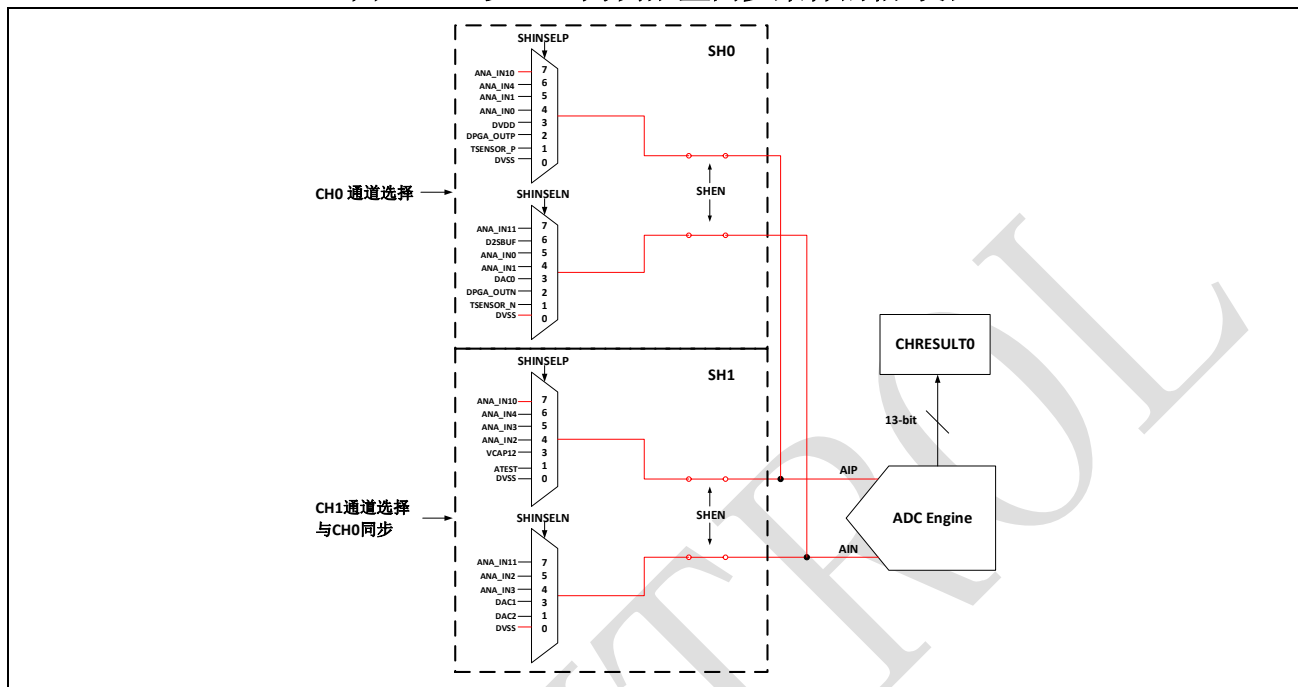
CHCTL[x].SHINSELN 为所选采样保持器的负端输入，CHCTL[x].SHINSELP 为所选采样保持器的正端输入，采样保持器 SH0 和 SH1 的通道选择如表 11-3 所示。

表 11-3: ADC 通道选择

Option	SHINSELOP	SHINSEL0N	SHINSEL1P	SHINSEL1N
7	ANA_IN10	ANA_IN11	ANA_IN10	ANA_IN11
6	ANA_IN4	D2SBUF	ANA_IN4	-
5	ANA_IN1	ANA_IN0	ANA_IN3	ANA_IN2
4	ANA_IN0	ANA_IN1	ANA_IN2	ANA_IN3
3	DVDD	DAC0	VCAP12	DAC1
2	DPGA_OUTP	DPGA_OUTN	-	-

Option	SHINSEL0P	SHINSEL0N	SHINSEL1P	SHINSEL1N
1	TSENSOR_P	TSENSOR_N	ATEST	DAC2
0	DVSS	DVSS	DVSS	DVSS

图 11-2: 以 CH0 为例配置同步采样的信号流



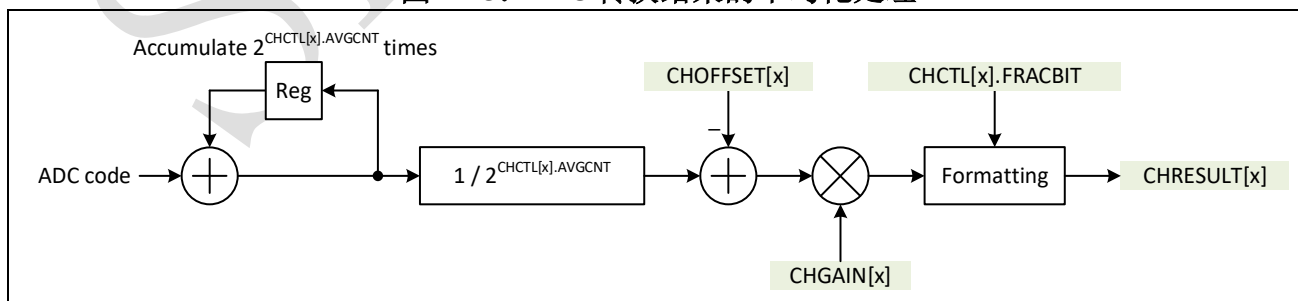
11.3.3 转换优先级

当多个通道 CH 同一时间接收到 REQ 请求时，按照 CH 编号越小优先级越高的固定优先级顺序。

11.3.4 平均化处理

众所周知，采样系统中存在热噪声。这些热噪声来自输入信号、电容采样噪声、比较器噪声、基准电压噪声和电源噪声，等等。如果我们使用软件进行数据后处理，将会等待较长时间，且消耗 CPU 资源。CH 中的平均化处理在硬件上完成了这一目的。这样，ADC 就可以连续采样多次，并在硬件上做平均处理，将平均后的结果存入 CHRESULT[x] 寄存器。

图 11-3: ADC 转换结果的平均化处理



每个通道的控制寄存器 CHCTL[x] 中的位段 AVGCNT 可以控制平均化次数。

表 11-4: 平均化次数选择

AVGCNT[7:0]	平均化次数
7	128
6	64
5	32
4	16
3	8
2	4
1	2
0	1

11.3.5 ADC 采样和转换窗口

采样器电压建立到预期的输入电压所需的时间取决于信号源的驱动力，因此有必要对采样和转换时间窗口大小提供可编程性。这可以通过 CHCTL[x]中的 SAMP CNT 和 ADCCTL 中的 CONVCNT 位段实现：

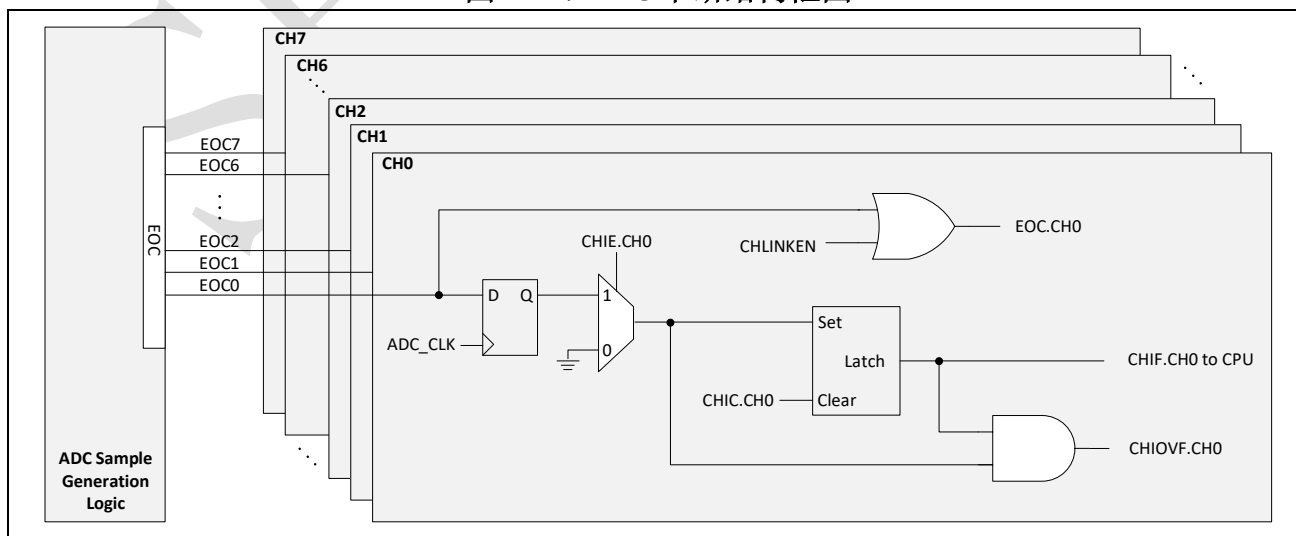
- 通道 x 的采样时间为(CHCTL[x].SAMP CNT + 1)个 ADC 时钟周期;
- 通道 x 的转换时间为(ADCCTL.CONV CNT + 1)个 ADC 时钟周期;
- 单次处理模拟输入的总时间为采样时间和转换时间的和。

11.4 转换结束和产生中断原理

因为 ADC 有 8 个通道，所以也有 8 个转换结束标志（EOC，End-of-Conversion），在每次转换结束时发生。顺序采样模式下，EOC[x]与每个通道对应；同步采样模式下，EOC[x]与每组的第一个通道对应。每个 EOCx 信号都可以被配置成其它 CH 的 REQ 信号，并且产生 ADC 中断送入 CPU。

图 11-4 展示了 ADC 中断结构的框图。

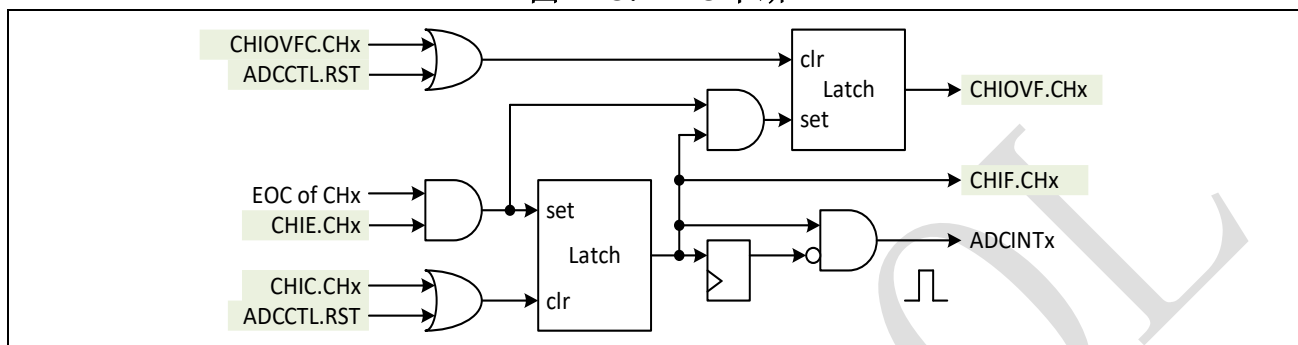
图 11-4: ADC 中断结构框图



如图 11-5 所示，通道 x 的 EOC（转换结束）事件可以通过将 CHIE.CHx 设置为 1 来使能，从而产生中断请求 ADCINTx 给 CPU。中断标志锁存在相应的 CHIF.CHx 中，可以通过向 CHIC.CHx 或 ADCCTL.RST 写 1 来清除。如果在 CHIF.CHx 已置位时出现中断请求，则会发生中断溢出事件，该事件将被锁存到 CHIOVF.CHx，可以通过向 CHIOVFC.CHx 或 ADCCTL.RST 写 1 来清除。

请注意，CHIE.CHx 的配置不影响使用 CHx 的 EOC 事件来触发请求信号。

图 11-5: ADC 中断



11.5 ADC 时序

图 11-6：顺序转换模式时序

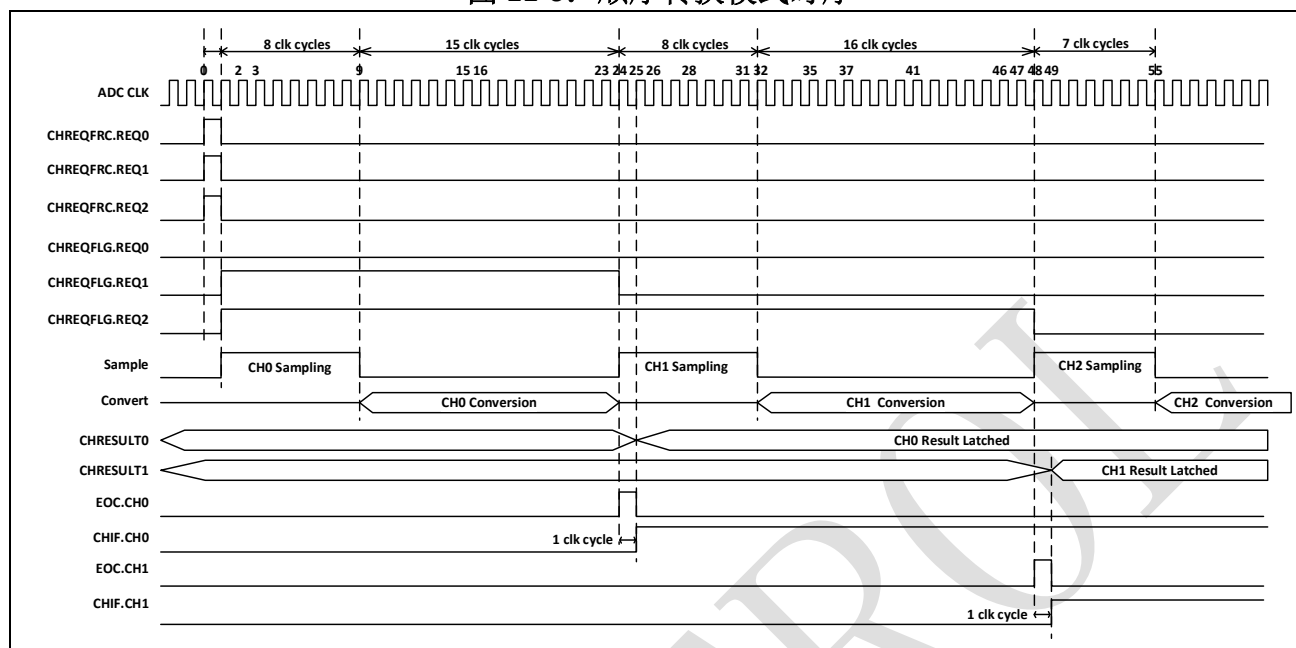
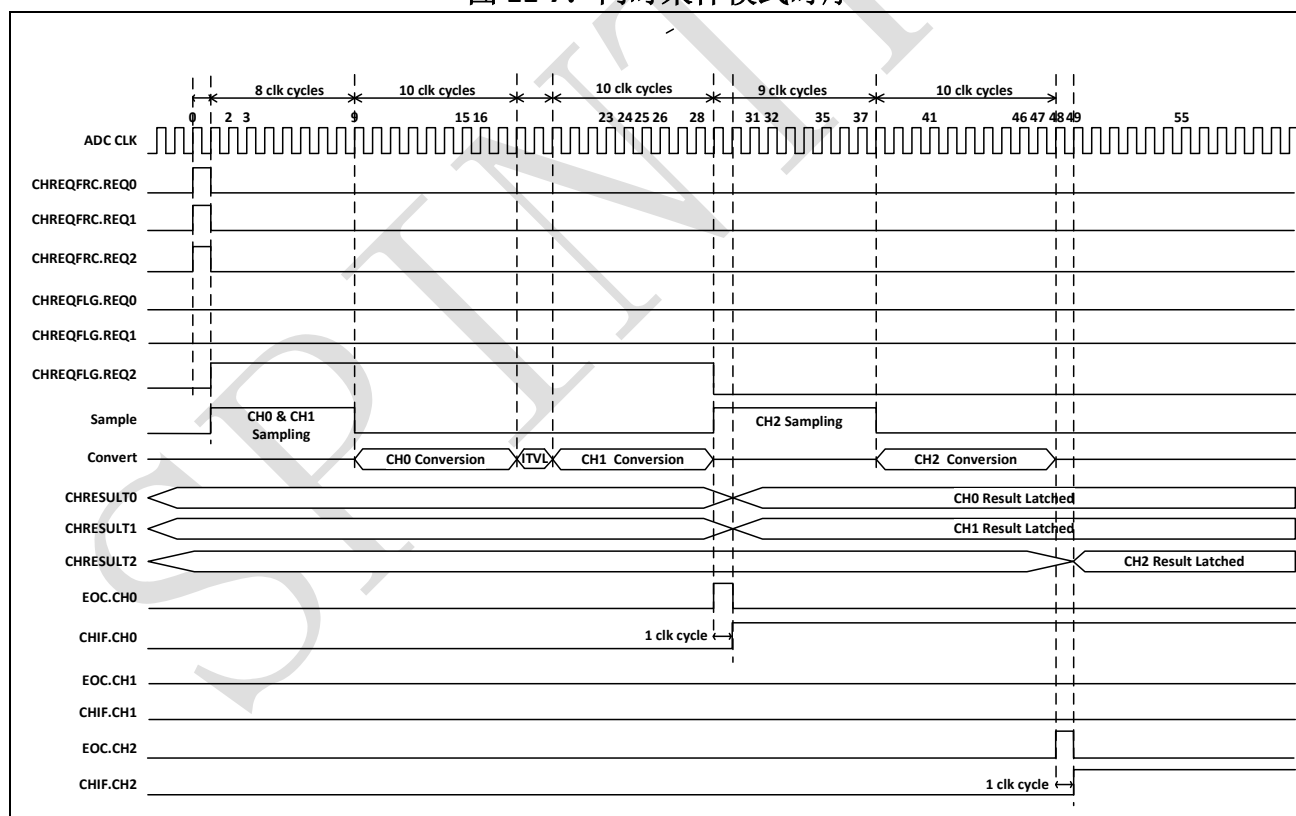


图 11-7：同时采样模式时序



11.6 ADC 校准

转换器存在固有的失调和增益误差。出厂时，每个 ADC 的失调和增益误差系数都会逐一测试并存储在 Flash 闪存里，并在上电时自动加载到 SHOFFSET[n] (n=0, 1) 和 SHGAIN[n] (n=0, 1) 寄存器中。用户可以将它们复制到通道 CHx 对应的 CHOFFSET[x] 和 CHGAIN[x] (x=0~7)，也可以通过更改寄存器 CHOFFSET[x] 和 CHGAIN[x] (x=0~7) 的值进行实时校准。

11.7 ADC 结果

11.7.1 满幅度测量范围

ADC 的满幅度测量范围从 -FS 到 +FS:

$$FS = 3.339V$$

但是，对于每个通道，其电压不能超过 DVDD。如果 DVDD 为 3.3V，则表示各通道输入电压不能超过 3.3V。

11.7.2 ADC 输出码值计算

ADC 所需的参考电压由带隙基准电路产生。量化的满幅值是 3.339V，从而差分输入范围是 -3.339V ~ 3.339V。

当 CHCTL[x].FRACBIT = 0 时，CHRESULT[x] 中的小数部分会被截断。因此，输出码值的范围是 -4096 到 4095，如图 11-8 所示。对于 ANA_INP 和 ANA_INN 引脚输入的差分电压，预期的码值是：

$$CHRESULT[x] = (V_{ANA\_INP} - V_{ANA\_INN}) / 3.339 * 4096$$

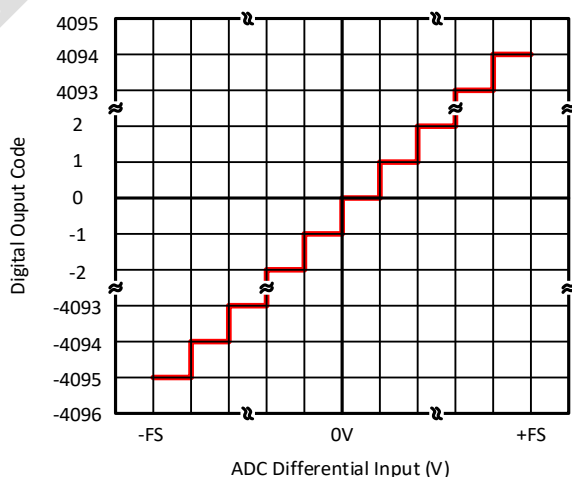
当 CHCTL[x].FRACBIT = 1 时，CHRESULT[x] 的低 7 位是小数部分。因此，输出码值的范围是 -524288 到 524287。对于 ANA_INP 和 ANA_INN 引脚输入的差分电压，预期的码值是：

$$CHRESULT[x] = (V_{ANA\_INP} - V_{ANA\_INN}) / 3.339 * 524288$$

寄存器 SHRESULT 中的数值是未经过失调和增益误差校准的原始码值，始终包含低 7 位的小数部分。

通过对符号位的扩展，CHRESULT[x] 和 SHRESULT 均可以按照 32 位有符号数来读取。

图 11-8: ADC 转换曲线



11.7.3 通道码值计算

如图 11-1 所示，每个通道有独立的失调（OFFSET0~7）和增益误差（GAIN0~7）。为此，在硬件上分配了独立的失调寄存器 CHOFFSETx（x=0~7）和增益寄存器 CHGAINx（x=0~7）。这允许用户根据每个通道的实际情况设定不同的校准参数。

通道码值计算公式为：

$$\text{CHRESULT}[i] = (\text{SHRESULT}[n] - \text{CHOFFSET}[i]) \times \text{CHGAIN}[i]$$

在上面的公式中，参数 i 表示通道编号，其范围从 0 到 7，参数 n 表示采样保持器编号，其范围为 0 到 1。

11.8 ADC 安全特性

11.8.1 输入浮空检测

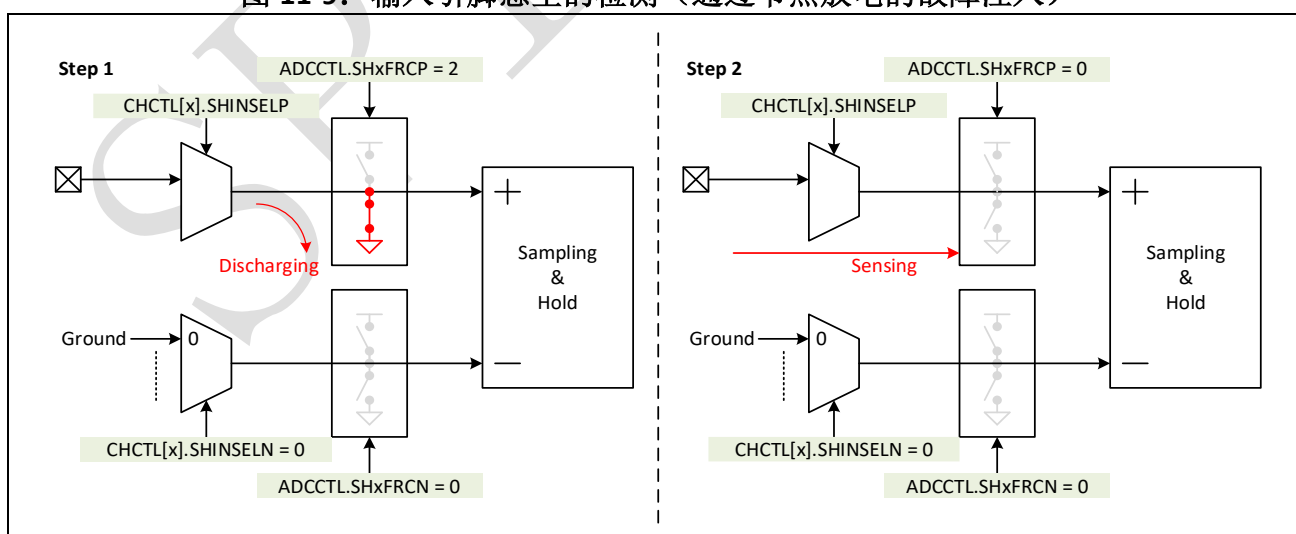
如果用于模拟输入的引脚未能可靠地连接到外部传感器，则无法正确测量电压，从而在某些应用中导致潜在的严重损坏。ADC 模块中提供了输入浮空检测电路来找出此类错误，从而激活用户自定义的故障安全程序。

有两种可选方案来检测输入浮空，它们都是基于故障注入，但分别是将节点充电至电源和将节点放电至地。

如图 11-9 所示，通过节点放电的故障注入，可以分两步检测连接到采样器正输入的引脚是否浮空。

- 对输入节点放电
 - 通过配置 CHCTL[x].SHINSELP 将采样器的正输入端连接到外部引脚；
 - 设置 CHCTL[x].SHINSELN = 0，将采样器负输入端连到地；
 - 设置 ADCCTL.SHxFRCP（x=0/1）=2，从而将采样器的正输入端下拉到地；
 - 等待大约 10us 使得输入节点完全放电到电压为 0。

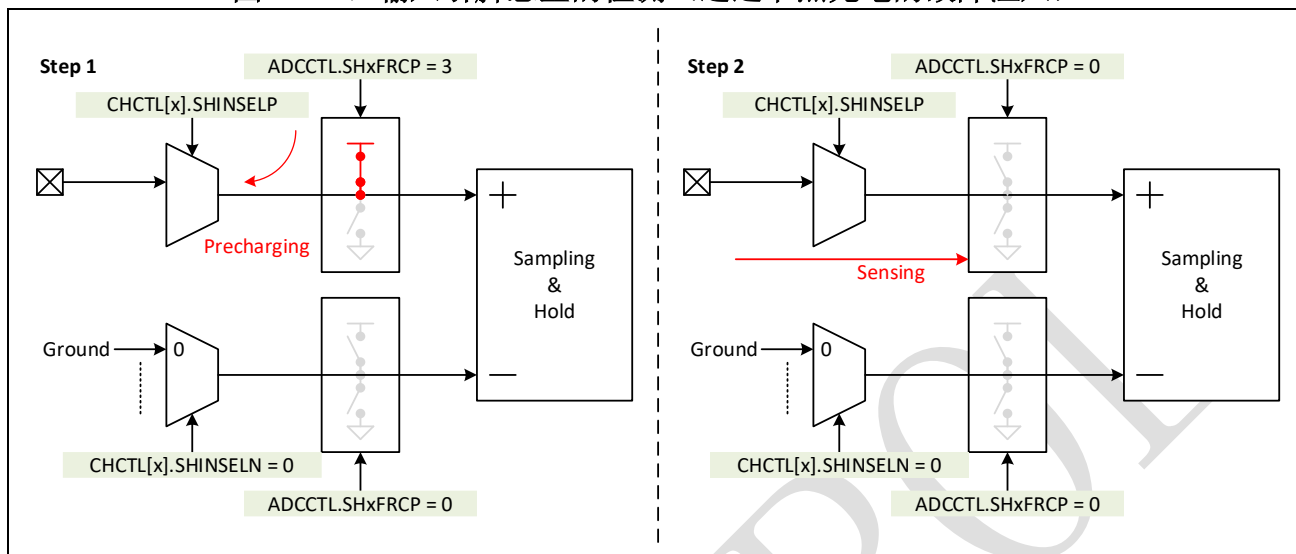
图 11-9：输入引脚悬空的检测（通过节点放电的故障注入）



- 恢复正常连接，检测节点连接
 - 设置 ADCCTL.SHxFRCP = 0，关闭下拉通路；
 - 触发通道 x 的请求信号并得到转换结果 CHRESULT[x]；

- 如果回读值总是 0，表明采样器正输入端不能跟踪外部输入的变化，即正输入端浮空。用类似的方法可以检测采样器负输入端是否浮空。

图 11-10：输入引脚悬空的检测（通过节点充电的故障注入）



如图 11-10 所示，通过节点充电的故障注入，可以分两步检测连接到采样器正输入的引脚是否悬空。

- 对输入节点充电
 - 通过配置 $CHCTL[x].SHINSELP$ 将采样器的正输入端连接到外部引脚；
 - 设置 $CHCTL[x].SHINSELN = 0$ ，将采样器负输入端连到地；
 - 设置 $ADCCTL.SHxFRCP = 3$ ，从而将采样器的正输入端上拉到电源；
 - 等待大约 10us 使得输入节点完全充电到电源电压。
- 恢复正常连接，检测节点连接
 - 设置 $ADCCTL.SHxFRCP = 0$ ，关闭上拉通路；
 - 触发通道 x 的请求信号并得到转换结果 $CHRESULT[x]$ ；
 - 如果回读值总是 4048 ($3.3/3.339 \times 4096$) 附近，表明采样器正输入端不能跟踪外部输入的变化，即正输入端浮空。

用类似的方法可以检测采样器负输入端是否浮空。

11.8.2 输入信号短路检测

模拟输入也可能出现短路情况。可以按照如下步骤使用充电或放电电路注入故障来检测：

- 获取正常连接时的码值
 - 设置 ADCCTL.SHxFRCF 和 ADCCTL.SHxFRCN (x=0/1) 为 0，关闭对采样器输入的上下拉；
 - 通过配置 CHCTL[x].SHINSELN 和 CHCTL[x].SHINSELN 配置采样器的输入引脚；
 - 触发通道 x 的请求信号并得到第一个 CHRESULT[x]。
- 获取通过充电注入故障时的码值
 - 根据采样器具体是正端或者负端连接到外部模拟输入引脚，配置 ADCCTL.SHxFRCF 或 ADCCTL.SHxFRCN 为 3，使能充电故障注入；
 - 等待大约 10us 从而输入节点充电完成；
 - 触发通道 x 的请求信号并得到第二个 CHRESULT[x]。
- 获取通过放电注入故障时的码值
 - 根据采样器具体是正端或者负端连接到外部模拟输入引脚，配置 ADCCTL.SHxFRCF 或 ADCCTL.SHxFRCN 为 2，使能放电故障注入；
 - 等待大约 10us 从而输入节点放电完成；
 - 触发通道 x 的请求信号并得到第三个 CHRESULT[x]。
- 若三个 CHRESULT[x] 非常接近，则判定为输入短路。

注意： 所能检测的短路工况仅限于输入短路到一个低阻节点，其驱动力超过上下拉诊断电路提供的 8uA 的充放电能力。

11.9 寄存器

11.9.1 ADC 寄存器列表

表 11-5: ADC 模块基地址

外设模块	基地址
ADC	0x4000 8C00

表 11-6: ADC 寄存器列表

寄存器	偏移地址	描述	复位值
CHIF	0x0000	ADC 中断标志寄存器	0x00000000
CHIC	0x0004	ADC 中断清除寄存器	0x00000000
CHIOVF	0x0008	ADC 中断溢出标志寄存器	0x00000000
CHIOVFC	0x000C	ADC 中断溢出清除寄存器	0x00000000
CHIE*	0x0010	ADC 中断使能寄存器	0x00000000
CHREQFLG	0x0018	ADC REQ 标志寄存器	0x00000000
CHREQFRC	0x001C	ADC REQ 软件强制寄存器	0x00000000
CHREQOVF	0x0020	ADC REQ 溢出寄存器	0x00000000
CHREQOVFC	0x0024	ADC REQ 溢出清除寄存器	0x00000000
CHLINKEN*	0x0028	ADC 通道连接使能寄存器	0x00000000
CHLINK*	0x002C	ADC 通道连接寄存器	0x00000000
EXTREQCTL*	0x0034	ADC 外部 REQ 控制寄存器	0x0000007F
CHCTL*	0x0038	ADC REQ 控制寄存器	0x00000000
CHOFFSET*	0x0078	ADC 偏移校准寄存器	0x00000000
CHGAIN*	0x00B8	ADC 增益校准寄存器	0x00008000
SHOFFSET*	0x00F8	ADC SHx 偏移校准寄存器	0x00000000
SHGAIN*	0x0104	ADC SHx 增益校准寄存器	0x00008000
SHFLG	0x0110	ADC 状态寄存器	0x00000000
SHFLGCLR	0x0114	ADC 状态清除寄存器	0x00000000
ADCCTL*	0x0118	ADC 控制寄存器	0x00000000
SHRESULT	0x0124	ADC SHx 原始结果寄存器	0x00000000
CHRESULT	0x0130	ADC 转换结果寄存器	0x00000000
TSENSCTL*	0x02B0	温度传感器控制寄存器	0x00000000
ADCREGKEY	0x02B4	ADC 模块写使能寄存器	0x1ACCE551

注意： 由*标识的寄存器仅当 ADCREGKEY=0x1ACCE551 时才可以改写。

11.9.2 ADC 寄存器

11.9.2.1 ADC 中断标志寄存器（CHIF）

表 11-7：ADC 中断标志寄存器（CHIF）位段定义

CHIF (ADC Interrupt Flag Register)							
Access: ADC->CHIF		Offset: 0x0000		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

表 11-8：ADC 中断标志寄存器（CHIF）位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	CH7	RO	0x0	ADC 中断 7 标志 0: ADC 中断 7 未发生 1: ADC 中断 7 已发生
6	CH6	RO	0x0	ADC 中断 6 标志 0: ADC 中断 6 未发生 1: ADC 中断 6 已发生
5	CH5	RO	0x0	ADC 中断 5 标志 0: ADC 中断 5 未发生 1: ADC 中断 5 已发生
4	CH4	RO	0x0	ADC 中断 4 标志 0: ADC 中断 4 未发生 1: ADC 中断 4 已发生
3	CH3	RO	0x0	ADC 中断 3 标志 0: ADC 中断 3 未发生 1: ADC 中断 3 已发生
2	CH2	RO	0x0	ADC 中断 2 标志 0: ADC 中断 2 未发生 1: ADC 中断 2 已发生
1	CH1	RO	0x0	ADC 中断 1 标志 0: ADC 中断 1 未发生 1: ADC 中断 1 已发生
0	CH0	RO	0x0	ADC 中断 0 标志 0: ADC 中断 0 未发生 1: ADC 中断 0 已发生

11.9.2.2 ADC 中断清除寄存器 (CHIC)

表 11-9: ADC 中断清除寄存器 (CHIC) 位段定义

CHIC (ADC Interrupt Flag Clear Register)							
Access: ADC->CHIC		Offset: 0x0004		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

表 11-10: ADC 中断清除寄存器 (CHIC) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	CH7	W1C	0x0	清除 ADC 中断 7 标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除 CHIF.INT7。本位段自动清零。
6	CH6	W1C	0x0	清除 ADC 中断 6 标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除 CHIF.INT6。本位段自动清零。
5	CH5	W1C	0x0	清除 ADC 中断 5 标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除 CHIF.INT5。本位段自动清零。
4	CH4	W1C	0x0	清除 ADC 中断 4 标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除 CHIF.INT4。本位段自动清零。
3	CH3	W1C	0x0	清除 ADC 中断 3 标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除 CHIF.INT3。本位段自动清零。
2	CH2	W1C	0x0	清除 ADC 中断 2 标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除 CHIF.INT2。本位段自动清零。
1	CH1	W1C	0x0	清除 ADC 中断 1 标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除 CHIF.INT1。本位段自动清零。
0	CH0	W1C	0x0	清除 ADC 中断 0 标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除 CHIF.INT0。本位段自动清零。

11.9.2.3 ADC 中断溢出标志寄存器 (CHIOVF)

表 11-11: ADC 中断溢出标志寄存器 (CHIOVF) 位段定义

CHIOVF (ADC Interrupt Overflow Flag Register)							
Access: ADC->CHIOVF Offset: 0x0008 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

表 11-12: ADC 中断溢出标志寄存器 (CHIOVF) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	CH7	RO	0x0	ADC 中断 7 溢出标志 0: 未溢出 1: ADC 中断 7 溢出
6	CH6	RO	0x0	ADC 中断 6 溢出标志 0: 未溢出 1: ADC 中断 6 溢出
5	CH5	RO	0x0	ADC 中断 5 溢出标志 0: 未溢出 1: ADC 中断 5 溢出
4	CH4	RO	0x0	ADC 中断 4 溢出标志 0: 未溢出 1: ADC 中断 4 溢出
3	CH3	RO	0x0	ADC 中断 3 溢出标志 0: 未溢出 1: ADC 中断 3 溢出
2	CH2	RO	0x0	ADC 中断 2 溢出标志 0: 未溢出 1: ADC 中断 2 溢出
1	CH1	RO	0x0	ADC 中断 1 溢出标志 0: 未溢出 1: ADC 中断 1 溢出
0	CH0	RO	0x0	ADC 中断 0 溢出标志 0: 未溢出 1: ADC 中断 0 溢出

11.9.2.4 ADC 中断溢出清除寄存器 (CHIOVFC)

表 11-13: ADC 中断溢出清除寄存器 (CHIOVFC) 位段定义

CHIOVFC (ADC Interrupt Overflow Flag Clear Register)							
Access: ADC->CHIOVFC Offset: 0x000C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

表 11-14: ADC 中断溢出清除寄存器 (CHIOVFC) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	CH7	W1C	0x0	清除 ADC 中断 7 溢出标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除溢出标志。本位段自动清除。
6	CH6	W1C	0x0	清除 ADC 中断 6 溢出标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除溢出标志。本位段自动清除。
5	CH5	W1C	0x0	清除 ADC 中断 5 溢出标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除溢出标志。本位段自动清除。
4	CH4	W1C	0x0	清除 ADC 中断 4 溢出标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除溢出标志。本位段自动清除。
3	CH3	W1C	0x0	清除 ADC 中断 3 溢出标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除溢出标志。本位段自动清除。
2	CH2	W1C	0x0	清除 ADC 中断 2 溢出标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除溢出标志。本位段自动清除。
1	CH1	W1C	0x0	清除 ADC 中断 1 溢出标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除溢出标志。本位段自动清除。
0	CH0	W1C	0x0	清除 ADC 中断 0 溢出标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除溢出标志。本位段自动清除。

11.9.2.5 ADC 中断使能寄存器 (CHIE)

表 11-15: ADC 中断使能寄存器 (CHIE) 位段定义

CHIE (ADC Interrupt Enable Register)							
Access: ADC->CHIE		Offset: 0x0010		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

表 11-16: ADC 中断使能寄存器 (CHIE) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	CH7	RW	0x0	ADC 中断 7 使能 0: 关闭 ADC 中断 7 1: 使能 ADC 中断 7
6	CH6	RW	0x0	ADC 中断 6 使能 0: 关闭 ADC 中断 6 1: 使能 ADC 中断 6
5	CH5	RW	0x0	ADC 中断 5 使能 0: 关闭 ADC 中断 5 1: 使能 ADC 中断 5
4	CH4	RW	0x0	ADC 中断 4 使能 0: 关闭 ADC 中断 4 1: 使能 ADC 中断 4
3	CH3	RW	0x0	ADC 中断 3 使能 0: 关闭 1: 使能 ADC 中断 3
2	CH2	RW	0x0	ADC 中断 2 使能 0: 关闭 ADC 中断 2 1: 使能 ADC 中断 2
1	CH1	RW	0x0	ADC 中断 1 使能 0: 关闭 ADC 中断 1 1: 使能 ADC 中断 1
0	CH0	RW	0x0	ADC 中断 0 使能 0: 关闭 ADC 中断 0 1: 使能 ADC 中断 0

11.9.2.6 ADC 通道请求标志寄存器 (CHREQFLG)

表 11-17: ADC 通道请求标志寄存器 (CHREQFLG) 位段定义

CHREQFLG (ADC Start Of Conversion Request Flag Register)							
Access: ADC->CHREQFLG Offset: 0x0018 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

表 11-18: ADC 通道请求标志寄存器 (CHREQFLG) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	CH7	RO	0x0	CH7 (开始采样) 标志 0: CH7 没有采样请求 1: 接收到请求信号, CH7 采样请求挂起 当采样开始时, 本位段自动清零。
6	CH6	RO	0x0	CH6 (开始采样) 标志 0: CH6 没有采样请求 1: 接收到请求信号, CH6 采样请求挂起 当采样开始时, 本位段自动清零。
5	CH5	RO	0x0	CH5 (开始采样) 标志 0: CH5 没有采样请求 1: 接收到请求信号, CH5 采样请求挂起 当采样开始时, 本位段自动清零。
4	CH4	RO	0x0	CH4 (开始采样) 标志 0: CH4 没有采样请求 1: 接收到请求信号, CH4 采样请求挂起 当采样开始时, 本位段自动清零。
3	CH3	RO	0x0	CH3 (开始采样) 标志 0: CH3 没有采样请求 1: 接收到请求信号, CH3 采样请求挂起 当采样开始时, 本位段自动清零。
2	CH2	RO	0x0	CH2 (开始采样) 标志 0: CH2 没有采样请求 1: 接收到请求信号, CH2 采样请求挂起 当采样开始时, 本位段自动清零。
1	CH1	RO	0x0	CH1 (开始采样) 标志

位段	位段名	属性	复位值	描述
				0: CH1 没有采样请求 1: 接收到请求信号，CH1 采样请求挂起 当采样开始时，本位段自动清零。
0	CH0	RO	0x0	CH0（开始采样）标志 0: CH0 没有采样请求 1: 接收到请求信号，CH0 采样请求挂起 当采样开始时，本位段自动清零。

11.9.2.7 ADC 通道请求强制寄存器 (CHREQFRC)

表 11-19: ADC 通道请求强制寄存器 (CHREQFRC) 位段定义

CHREQFRC (ADC Channel Request Force Register)							
Access: ADC->CHREQFRC Offset: 0x001C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

表 11-20: ADC 通道请求强制寄存器 (CHREQFRC) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	CH7	W1S	0x0	请求强制 CH7 (开始采样) 0: 写 0 无效, 总是读回 0 1: 写 1 强制 CH7, 本位段自动清零。
6	CH6	W1S	0x0	请求强制 CH6 (开始采样) 0: 写 0 无效, 总是读回 0 1: 写 1 强制 CH6, 段自动清零。
5	CH5	W1S	0x0	请求强制 CH5 (开始采样) 0: 写 0 无效, 总是读回 0 1: 写 1 强制 CH5, 位段自动清零。
4	CH4	W1S	0x0	请求强制 CH4 (开始采样) 0: 写 0 无效, 总是读回 0 1: 写 1 强制 CH4, 位段自动清零。
3	CH3	W1S	0x0	请求强制 CH3 (开始采样) 0: 写 0 无效, 总是读回 0 1: 写 1 强制 CH3, 位段自动清零。
2	CH2	W1S	0x0	请求强制 CH2 (开始采样) 0: 写 0 无效, 总是读回 0 1: 写 1 强制 CH2, 位段自动清零。
1	CH1	W1S	0x0	请求强制 CH1 (开始采样) 0: 写 0 无效, 总是读回 0 1: 写 1 强制 CH1, 位段自动清零。
0	CH0	W1S	0x0	请求强制 CH0 (开始采样) 0: 写 0 无效, 总是读回 0 1: 写 1 强制 CH0, 位段自动清零。

11.9.2.8 ADC REQ 溢出寄存器 (CHREQOVF)

表 11-21: ADC REQ 溢出寄存器 (CHREQOVF) 位段定义

CHREQOVF (ADC Channel Request Overflow Register)							
Access: ADC->CHREQOVF		Offset: 0x0020		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

表 11-22: ADC REQ 溢出寄存器 (CHREQOVF) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	CH7	RO	0x0	CH7 请求溢出标志 0: 中断未溢出 1: 中断溢出
6	CH6	RO	0x0	CH6 请求溢出标志 0: 中断未溢出 1: 中断溢出
5	CH5	RO	0x0	CH5 请求溢出标志 0: 中断未溢出 1: 中断溢出
4	CH4	RO	0x0	CH4 请求溢出标志 0: 中断未溢出 1: 中断溢出
3	CH3	RO	0x0	CH3 请求溢出标志 0: 中断未溢出 1: 中断溢出
2	CH2	RO	0x0	CH2 请求溢出标志 0: 中断未溢出 1: 中断溢出
1	CH1	RO	0x0	CH1 请求溢出标志 0: 中断未溢出 1: 中断溢出
0	CH0	RO	0x0	CH0 请求溢出标志 0: 中断未溢出 1: 中断溢出

11.9.2.9 ADC REQ 溢出清除寄存器 (CHREQOVFC)

表 11-23: ADC REQ 溢出清除寄存器 (CHREQOVFC) 位段定义

CHREQOVFC (ADC Channel Request Overflow Clear Register)							
Access: ADC->CHREQOVFC Offset: 0x0024 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

表 11-24: ADC REQ 溢出清除寄存器 (CHREQOVFC) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	CH7	W1C	0x0	清除 CH7 溢出标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除溢出标志, 本位段自动清零。
6	CH6	W1C	0x0	清除 CH6 溢出标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除溢出标志, 本位段自动清零。
5	CH5	W1C	0x0	清除 CH5 溢出标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除溢出标志, 本位段自动清零。
4	CH4	W1C	0x0	清除 CH4 溢出标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除溢出标志, 本位段自动清零。
3	CH3	W1C	0x0	清除 CH3 溢出标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除溢出标志, 本位段自动清零。
2	CH2	W1C	0x0	清除 CH2 溢出标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除溢出标志, 本位段自动清零。
1	CH1	W1C	0x0	清除 CH1 溢出标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除溢出标志, 本位段自动清零。
0	CH0	W1C	0x0	清除 CH0 溢出标志 0: 写 0 无效, 总是读回 0 1: 写 1 清除溢出标志, 本位段自动清零。

11.9.2.10 ADC 通道连接使能寄存器（CHLINKEN）

表 11-25: ADC 通道连接使能寄存器（CHLINKEN）位段定义

CHLINKEN (ADC Channel Link Enable Register)							
Access: ADC->CHLINKEN		Offset: 0x0028		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

表 11-26: ADC 通道连接使能寄存器（CHLINKEN）位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	CH7	RW	0x0	通道 7 连接使能 0: 关闭 1: 使能
6	CH6	RW	0x0	通道 6 连接使能 0: 关闭 1: 启用
5	CH5	RW	0x0	通道 5 连接使能 0: 关闭 1: 启用
4	CH4	RW	0x0	通道 4 连接使能 0: 关闭 1: 启用
3	CH3	RW	0x0	通道 3 连接使能 0: 关闭 1: 启用
2	CH2	RW	0x0	通道 2 连接使能 0: 关闭 1: 启用
1	CH1	RW	0x0	通道 1 连接使能 0: 关闭 1: 启用
0	CH0	RW	0x0	通道 0 连接使能 0: 关闭 1: 启用

11.9.2.11 ADC 通道连接寄存器 (CHLINK)

表 11-27: ADC 通道连接寄存器 (CHLINK) 位段定义

CHLINK (ADC Channel Link Register)							
Access: ADC->CHLINK Offset: 0x002C Default: 0x00000000							
31 RESERVED	30	29 CH7	28	27 RESERVED	26	25 CH6	24
23 RESERVED	22	21 CH5	20	19 RESERVED	18	17 CH4	16
15 RESERVED	14	13 CH3	12	11 RESERVED	10	9 CH2	8
7 RESERVED	6	5 CH1	4	3 RESERVED	2	1 CH0	0

表 11-28: ADC 通道连接寄存器 (CHLINK) 位段描述

位段	位段名	属性	复位值	描述
31	RESERVED_31	RO	0x0	保留
30:28	CH7	RW	0x0	中断事件 7 REQ
27	RESERVED_27	RO	0x0	保留
26:24	CH6	RW	0x0	中断事件 6 REQ
23	RESERVED_23	RO	0x0	保留
22:20	CH5	RW	0x0	中断事件 5 REQ
19	RESERVED_19	RO	0x0	保留
18:16	CH4	RW	0x0	中断事件 4 REQ
15	RESERVED_15	RO	0x0	保留
14:12	CH3	RW	0x0	中断事件 3 REQ
11	RESERVED_11	RO	0x0	保留
10:8	CH2	RW	0x0	中断事件 2 REQ
7	RESERVED_7	RO	0x0	保留
6:4	CH1	RW	0x0	中断事件 1 REQ
3	RESERVED_3	RO	0x0	保留
2:0	CH0	RW	0x0	中断事件 0 REQ

11.9.2.12 ADC 外部 REQ 控制寄存器（EXTREQCTL）

表 11-29: ADC 外部 REQ 控制寄存器（EXTREQCTL）位段定义

EXTREQCTL (ADC External Request Input Control Register)							
Access: ADC->EXTREQCTL		Offset: 0x0034		Default: 0x0000007F			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	POL	IOSEL					

表 11-30: ADC 外部 REQ 控制寄存器（EXTREQCTL）位段描述

位段	位段名	属性	复位值	描述
31:7	RESERVED_31_7	RO	0x0	保留
6	POL	RW	0x1	外部请求信号极性 0: 电平触发 1: 高电平触发
5:0	IOSEL	RW	0x3F	选择外部请求信号（GPIO 管脚号）

11.9.2.13 ADC 通道 x (x=[0:7]) 控制寄存器 (CHCTL)

表 11-31: ADC 通道 x (x=[0:7]) 控制寄存器 (CHCTL) 位段定义

CHCTL (ADC Channel Control Register)							
Access: ADC->CHCTL[n] Offset: 0x0038 Default: 0x00000000							
31 RESERVED	30 SHEN	29	28 RESERVED	27	26 REQSRC	25	24
23	22 SHINSELN	21	20	19 SHINSELP	18	17 AVGCNT	16
15 AVGCNT	14 FRACBIT	13	12	11	10	9	8
7	6	5	4	3	2	1	0
SAMP CNT							

表 11-32: ADC 通道 x (x=[0:7]) 控制寄存器 (CHCTL) 位段描述

位段	位段名	属性	复位值	描述
31	RESERVED_31	RO	0x0	保留
30:29	SHEN	RW	0x0	SH 使能 0: 关闭所有 SH 1: 使能 SH0 2: 使能 SH1 3: 同时使能 SH0 和 SH1 该选项仅对 CHCTL[0/2/4/6]有效
28	RESERVED_28	RO	0x0	保留
27:24	REQSRC	RW	0x0	CH 请求信号，将设置 CHREQFLG 中的位，可以被 CHLINK 改写 0: 仅软件 1: TIMER0 触发 REQ 事件 2: TIMER1 触发 REQ 事件 3: 由 EXTREQCTL 定义的外部输入 4: PWM2REQA 5: PWM2REQB 6: PWM2REQC 7: PWM3REQA 8: PWM3REQB 9: PWM3REQC 10: PWM4REQA 11: PWM4REQB 12: PWM4REQC 13: PWM5REQA 14: PWM5REQB 15: PWM5REQC
23:21	SHINSELN	RW	0x0	选择所选 SH 的负端输入 详情见表 11-3

位段	位段名	属性	复位值	描述
20:18	SHINSELP	RW	0x0	选择所选 SH 的正端输入 详情见表 11-3
17:15	AVGCNT	RW	0x0	选择平均计数值作为 ADC 结果 0: 无平均值 1: 每个 REQ 将触发 2 个连续的 ADC 操作，然后求平均值 2: 每个 REQ 将触发 4 次连续 ADC 操作，然后求平均值 3: 每个 REQ 将触发 8 次连续 ADC 操作，然后求平均值 4: 每个 REQ 将触发 16 次连续 ADC 操作，然后求平均值 5: 每个 REQ 将触发 32 次连续 ADC 操作，然后求平均值 6: 每个 REQ 将触发 64 个连续 ADC 操作，然后求平均值 7: 每个 REQ 将触发 128 次连续 ADC 操作，然后求平均值
14	FRACBIT	RW	0x0	由于平均值，ADC 结果保留 7 位最低有效位作为小数部分 0: ADC 结果为 13 位整数 1: ADC 结果为 20 位，7 位最低有效位为小数部分
13:8	RESERVED_13_8	RO	0x0	保留
7:0	SAMPCNT	RW	0x0	将采样窗口大小设置为 (SAMPCNT+1) ADC 时钟周期。

11.9.2.14 ADC 通道 x (x=[0:7]) 偏移校准寄存器 (CHOFFSET)

表 11-33: ADC 通道 x (x=[0:7]) 偏移校准寄存器 (CHOFFSET) 位段定义

CHOFFSET (ADC Channel Offset Trim Register)							
Access: ADC->CHOFFSET[n] Offset: 0x0078 Default: 0x00000000							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 11-34: ADC 通道 x (x=[0:7]) 偏移校准寄存器 (CHOFFSET) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	CHRESULTx 的偏移微调 (区分正负) 这是一个 13 位带扩展符号位的有符号数, 7 位最低有效位为小数部分。

11.9.2.15 ADC 通道 x (x=[0:7]) 增益校准寄存器 (CHGAIN)

表 11-35: ADC 通道 x (x=[0:7]) 增益校准寄存器 (CHGAIN) 位段定义

CHGAIN (ADC Channel Gain Trim Register)							
Access: ADC->CHGAIN[n] Offset: 0x00B8 Default: 0x00008000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 11-36: ADC 通道 x (x=[0:7]) 增益校准寄存器 (CHGAIN) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	VAL	RW	0x8000	CHRESULTx 的增益校准= (VAL/32768)

11.9.2.16 ADC SHx (x=[0:1]) 偏移校准寄存器 (SHOFFSET)

表 11-37: ADC SHx (x=[0:1]) 偏移校准寄存器 (SHOFFSET) 位段定义

SHOFFSET (ADC SHx Offset Trim Register)							
Access: ADC->SHOFFSET[n] Offset: 0x00F8 Default: 0x00000000							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 11-38: ADC SHx (x=[0:1]) 偏移校准寄存器 (SHOFFSET) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	SHx 的偏移校准 (区分正负) 这是一个 13 位带扩展符号位的有符号数, 7 位最低有效位为小数部分

11.9.2.17 ADC SHx (x=[0:1]) 增益校准寄存器 (SHGAIN)

表 11-39: ADC SHx (x=[0:1]) 增益校准寄存器 (SHGAIN) 位段定义

SHGAIN (ADC SHx Gain Trim Register)							
Access: ADC->SHGAIN[n] Offset: 0x0104 Default: 0x00008000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 11-40: ADC SHx (x=[0:1]) 增益校准寄存器 (SHGAIN) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	VAL	RW	0x8000	SHx 的增益校准= (VAL/32768)

11.9.2.18 ADC 状态寄存器 (SHFLG)

表 11-41: ADC 状态寄存器 (SHFLG) 位段定义

SHFLG (ADC Flag Register)							
Access: ADC->SHFLG		Offset: 0x0110		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED						CODE1RDY	CODEORDY

表 11-42: ADC 状态寄存器 (SHFLG) 位段描述

位段	位段名	属性	复位值	描述
31:2	RESERVED_31_2	RO	0x0	保留
1	CODE1RDY	RO	0x0	SHRESULT1 就绪标志 0: 未就绪 1: 就绪
0	CODEORDY	RO	0x0	SHRESULT0 就绪标志 0: 未就绪 1: 就绪

11.9.2.19 ADC 状态清除寄存器（SHFLGCLR）

表 11-43: ADC 状态清除寄存器（SHFLGCLR）位段定义

SHFLGCLR (ADC Flag Clear Register)							
Access: ADC->SHFLGCLR		Offset: 0x0114		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED						CODE1RDY	CODEORDY

表 11-44: ADC 状态清除寄存器（SHFLGCLR）位段描述

位段	位段名	属性	复位值	描述
31:2	RESERVED_31_2	RO	0x0	保留
1	CODE1RDY	W1C	0x0	清除 SHRESULT1 就绪标志 0: 写 0 无效，总是读回 0 1: 写 1 清除标志。本位段自动清零。
0	CODEORDY	W1C	0x0	清除 SHRESULT0 就绪标志 0: 写 0 无效，总是读回 0 1: 写 1 清除标志。本位段自动清零。

11.9.2.20 ADC 控制寄存器 (ADCCTL)

表 11-45: ADC 控制寄存器 (ADCCTL) 位段定义

ADCCTL (ADC Control Register)											
Access: ADC->ADCCTL		Offset: 0x0118		Default: 0x00000000							
31	30	29	28	27	26	25	24				
RESERVED		ITVLCNT									
23	22	21	20	19	18	17	16				
RESERVED		CONVCNT									
15	14	13	12	11	10	9	8				
RESERVED				SH1FRCN		SH1FRCP					
7	6	5	4	3	2	1	0				
SH0FRCN		SH0FRCP		RESERVED	RDY	RST	EN				

表 11-46: ADC 控制寄存器 (ADCCTL) 位段描述

位段	位段名	属性	复位值	描述
31:30	RESERVED_31_30	RO	0x0	保留
29:24	ITVLCNT	RW	0x0	将用于同步采样的两个相邻转换之间的时间间隔设置为 (ITVLCNT+1) ADC 时钟周期
23:22	RESERVED_23_22	RO	0x0	保留
21:16	CONVCNT	RW	0x0	将转换窗口大小设置为 (CONVCNT+1) ADC 时钟周期
15:12	RESERVED_15_12	RO	0x0	保留
11:10	SH1FRCN	RW	0x0	强制 SH1 负输入端连接 0: 关闭 2: 强制 SH1 负输入端接低电平 3: 强制 SH1 负输入端接高电平
9:8	SH1FRCP	RW	0x0	强制 SH1 正输入端连接 0: 关闭 2: 强制 SH1 正输入端接低电平 3: 强制 SH1 正输入端接高电平
7:6	SH0FRCN	RW	0x0	强制 SH0 负输入端连接 0: 关闭 2: 强制 SH0 负输入端接低电平 3: 强制 SH0 负输入端接高电平
5:4	SH0FRCP	RW	0x0	强制 SH0 正输入端连接 0: 关闭 2: 强制 SH0 正输入端接低电平 3: 强制 SH0 正输入端接高电平
3	RESERVED_3	RO	0x0	保留
2	RDY	RO	0x0	ADC 就绪指示

位段	位段名	属性	复位值	描述
				设置 EN=1 后，用户需要等待 RDY=1 后才能进一步进行 ADC 操作 0: 未就绪 1: 就绪
1	RST	W1C	0x0	ADC 数字逻辑复位 0: 写 0 无效，总是回读 0 1: 写 1 复位 ADC 数字逻辑，该位自动清零。
0	EN	RW	0x0	ADC 使能 0: 关闭 1: 使能

11.9.2.21 ADC SHx (x=[0:1]) 原始码值寄存器 (SHRESULT)

表 11-47: ADC SHx (x=[0:1]) 原始码值寄存器 (SHRESULT) 位段定义

SHRESULT (ADC SHx Raw Code Register)							
Access: ADC->SHRESULT[n]				Offset: 0x0124		Default: 0x00000000	
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 11-48: ADC SHx (x=[0:1]) 原始码值寄存器 (SHRESULT) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0x0	SHx 的原始码值 该值区分正负, 7 位最低有效位是小数部分

11.9.2.22 ADC 通道 x (x=[0:7]) 转换结果寄存器 (CHRESULT)

表 11-49: ADC 通道 x (x=[0:7]) 转换结果寄存器 (CHRESULT) 位段定义

CHRESULT (ADC CHx Result Register)							
Access: ADC->CHRESULT[n]				Offset: 0x0130		Default: 0x00000000	
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 11-50: ADC 通道 x (x=[0:7]) 转换结果寄存器 (CHRESULT) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0x0	ADC 通道 x 的转换结果 该值区分正负, 7 位最低有效位可以是由 CHCTL[X].FRACBIT 设定的小数部分

11.9.2.23 温度传感器控制寄存器（TSENSCTL）

表 11-51: 温度传感器控制寄存器（TSENSCTL）位段定义

TSENSCTL (Temperature Sensor Control Register) Offset: 0x2B0 Default: 0x00000000							
Access: ADC -> TSENSCTL.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	DEMSEL				SWAPBJT	RESERVED	EN

表 11-52: 温度传感器控制寄存器（TSENSCTL）位段描述

位段	位段名	属性	复位值	描述
31:7	RESERVED_31_7	RO	0x0	保留
6:3	DEMSEL	RW	0x0	选择电流分支，用于动态组件匹配
2	SWAPBJT	RW	0x0	交换 BJT 0: 不交换 BJT 1: 交换 BJT
1	RESERVED_1	RW	0x0	保留
0	EN	RW	0x0	温度传感器使能 0: 关闭 1: 使能

11.9.2.24 ADC 模块写使能寄存器（ADCREGKEY）

表 11-53: ADC 模块写使能寄存器（ADCREGKEY）位段定义

ADCREGKEY (ADC Register Write-Allow Key Register) Offset: 0x2B4 Default: 0x1ACCE551 Access: ADC -> ADCREGKEY.all							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

表 11-54: ADC 模块写使能寄存器（ADCREGKEY）位段描述

位段	位段名	属性	复位值	描述
31:0	KEY	RW	0x1ACCE551	写入 0x1ACCE551 以解锁受保护的 ADC 寄存器

12 温度传感器（T-sensor）

12.1 概述

SPD1121 提供了 1 个内部温度传感器用于测量芯片的结温。如图 11-1 所示，温度传感器被接入 ADC 采样保持器 A 的输入选择通道，用户可以通过 ADC 对温度进行测量，具体步骤如下：

1. 用校准后的 ADC 量出温度传感器的输出码值（Code）；
2. 用下式计算出对应的温度（℃）。

$$T = 1.734 \times \text{Code} - 278.45$$

注意： 1 个 ADC 码值对应 1.734℃。为满足温度传感器稳定时间的要求，ADC 采样时间至少应为 0.759us。SDK 中提供的函数 `ADC_GetTemperature()` 可以直接得到温度值。

12.2 精度提升

温度传感器的测量精度受内部器件固有失配（如电阻、晶体管参数偏差）的影响，通常存在一定的初始误差。由于温度传感器在多数应用中（如过温保护）对精度的要求较低，出厂时通常不会对其进行校准。

SPD1121 采用动态元件匹配（Dynamic Element Matching, DEM）技术，通过动态轮换内部元件的工作状态，平均化器件失配带来的误差，从而显著提升温度传感器的测量精度。该优化方法已集成至 SDK 的 `ADC_CalculateTemperature()` 函数中，用户可直接调用此函数获取修正后的温度值，无需额外配置。

12.3 寄存器

温度传感器相关寄存器的定义请参见表 11-51 和表 11-52。

13 差分可编程增益放大器（DPGA）

13.1 DPGA 概述

SPD1121 提供 1 个差分可编程增益放大器（DPGA），用于通过 13 位 ADC 测量接近地的差分电压，它的增益通过内部控制寄存器进行数字编程，其特点如下：

- 可编程增益（ G_{DPGA} ）：2、4、8、16、24、32、48 和 64
- 模拟输入引脚：DPGAP、DPGAN
- 差分输入电压： $\pm 2.7/G_{DPGA}$
- 宽广的共模输入范围（ $G_{DPGA} = 2$ ）：-1.5V ~ 2V
- 建立时间：250ns 至 1.4 μ s
- 增益误差在 $\pm 1.5\%$ 之内

13.2 输入及输入电压范围

DPGA 的输出范围为 0.3V 至 3V，因此 DPGA 的差分输入电压为 $\pm 2.7/G_{DPGA}$ 。DPGA 的增益设置和差分输入范围如下表所示。

表 13-1: 增益及差分输入/输出范围

DPGACTL.GAIN	G_{DPGA}	输入范围 (V)	输出范围 (V)
0	2	-1.35~1.35	0.3~3
1	4	-0.675~+0.675	0.3~3
2	8	-0.337~+0.337	0.3~3
3	16	-0.168~+0.168	0.3~3
4	24	-0.112~+0.112	0.3~3
5	32	-0.084~+0.084	0.3~3
6	48	-0.056~+0.056	0.3~3
7	64	-0.042~+0.042	0.3~3

13.3 应用

图 13-1 显示了如何将 DPGA 用作低侧电流感测放大器。电机电流通过感测电阻 R_{sh} 转换为电压，LP 滤波器用于抑制电磁干扰，建议使用的 R_{filt} 值为 12 欧姆， C_{filt} 为 1 纳法。DPGA 能够通过电阻 R_{sh} 测量正向和负向电流，输入信号经过 DPGA 放大，并送到 13 位 ADC。 R_{stray} 是从感测电阻到地的杂散电阻。

对于 DPGAP 引脚和 DPGAN 引脚的输入，输出如下：

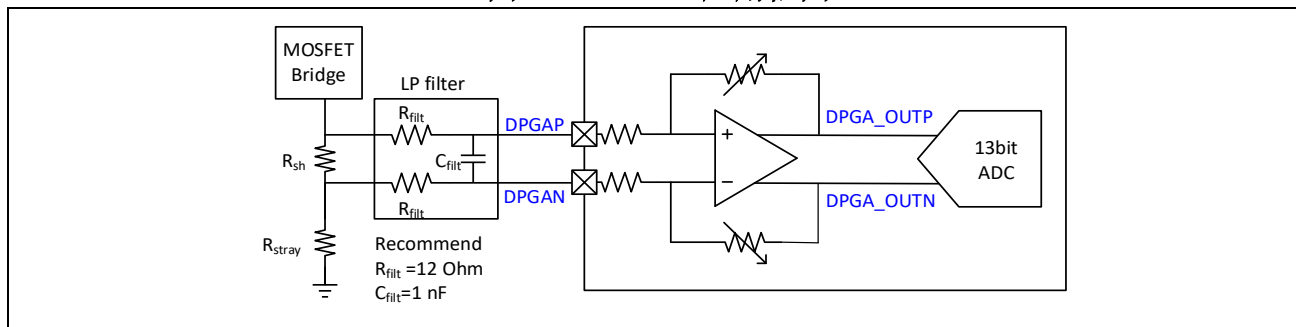
$$V_{DPGA\_OUTP} = V_{CMOUT} + (V_{DPGAP} - V_{DPGAN}) \times G_{DPGA} / 2$$

$$V_{DPGA\_OUTN} = V_{CMOUT} - (V_{DPGAP} - V_{DPGAN}) \times G_{DPGA} / 2$$

其中 V_{CMOUT} 是输出的共模电压，计算公式为：

$$V_{CMOUT} = V_{DVDD33} / 2$$

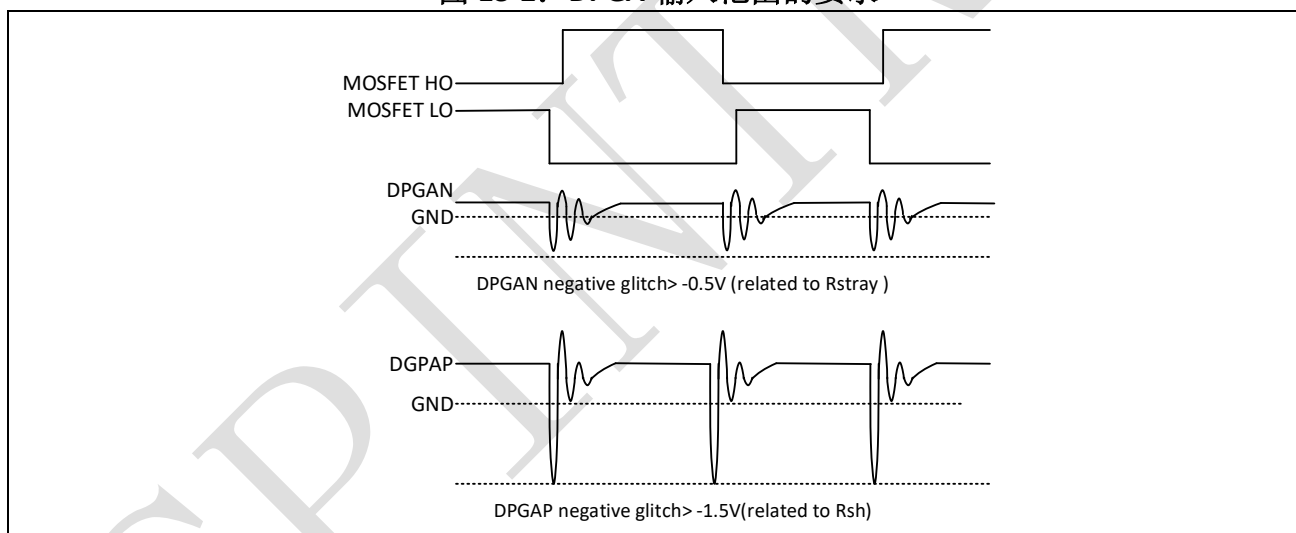
图 13-1: DPGA 应用框图



当电机桥驱动器的高侧 MOSFET 关闭并且低侧 MOSFET 打开时，由于寄生电感，DPGAP 和 DPGAN 处会产生电压尖峰，如图 13-2 所示。

- DPGAN 负向尖峰应在 -0.5V 以上，因为 DPGA 的输入共模范围为 -0.5V 至 2V。
 - PCB 上的 R_{stray} 应足够小；
- DPGAP 负向尖峰应在 -1.5V 以上，因为 DPGA 输入引脚可以承受 -1.5V。
 - PCB 上的 R_{sh} 应选择合理的值；

图 13-2: DPGA 输入范围的要求

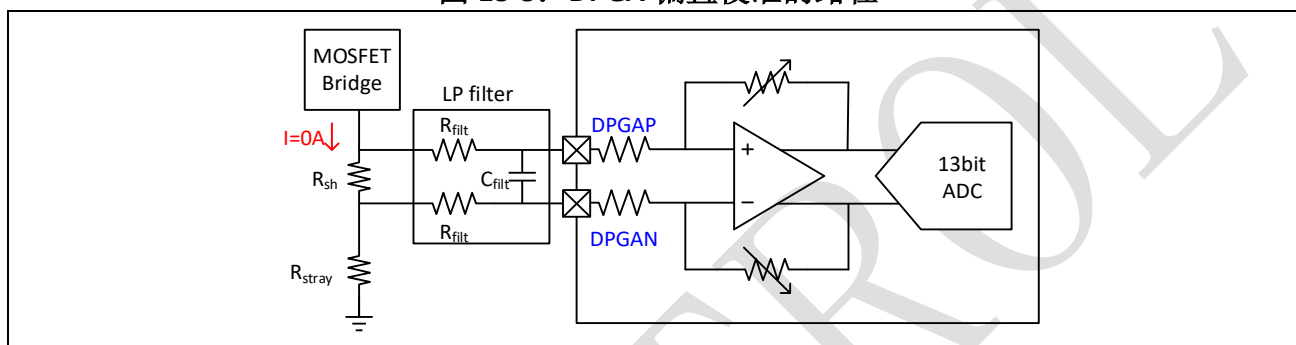


13.4 偏置校准

为了进行正确的偏置校准，应该测量从感应电阻到 DPGA 再到 13 位 ADC 的完整路径。为此，应确保感应电阻中没有电流（0A），如图 13-3 所示。以下是推荐的偏移校准步骤：

- 将 DPGA 输出连接到 ADC 输入
- 设置所需的 DPGA 增益
- 确保感应电阻中没有电流流过（0A）
- 进行 ADC 测量
- ADC 的数字输出即为完整路径的偏移值

图 13-3: DPGA 偏置校准的路径



13.5 寄存器

13.5.1 DPGA 寄存器列表

表 13-2: DPGA 模块基地址

外设模块	基地址
DPGA	0x4000 8C00

表 13-3: DPGA 寄存器列表

寄存器	偏移地址	描述	复位值
DPGACTL*	0x2C0	DPGA 控制寄存器	0x00000000
PGAREGKEY	0x2E0	PGA 模块写使能寄存器	0x1ACCE551

注意： 由*标识的寄存器仅当 PGAREGKEY =0x1ACCE551 时才可以改写。

13.5.2 DPGA 寄存器

13.5.2.1 DPGA 控制寄存器（DPGACTL）

表 13-4：DPGA 控制寄存器（DPGACTL）位段定义

DPGACTL (DPGA Control Register)							
Access: DPGA->DPGACTL Offset: 0x02C0 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED				D2SRO		D2SOE	D2SEN
7	6	5	4	3	2	1	0
PS	NIPU	PIPU	GAIN			BYP	EN

表 13-5：DPGA 控制寄存器（DPGACTL）位段描述

位段	位段名	属性	复位值	描述
31:12	RESERVED_31_12	RO	0x0	保留
11:10	D2SRO	RW	0x0	配置 DPGA D2S 缓冲到 GPIO3 的输出电阻 0: 0.5 kΩ 1: 1 kΩ 2: 5 kΩ 3: 10 kΩ
9	D2SOE	RW	0x0	使能 DPGA D2S 缓冲输出到 GPIO3 0: 关闭 1: 使能
8	D2SEN	RW	0x0	DPGA D2S 缓冲使能 0: 关闭 1: 使能
7	PS	RW	0x0	DPGA 输入上拉强度 0: 2 mA 1: 10 mA
6	NIPU	RW	0x0	DPGA 负端输入上拉使能 0: 关闭 1: 使能
5	PIPU	RW	0x0	DPGA 正端输入上拉使能 0: 关闭 1: 使能
4:2	GAIN	RW	0x0	DPGA 增益 0: 2x 1: 4x

位段	位段名	属性	复位值	描述
				2: 8x 3: 16x 4: 24x 5: 32x 3: 48x 7: 64x
1	BYP	RW	0x0	DPGA 旁路 0: 关闭 1: 使能
0	EN	RW	0x0	DPGA 使能 0: 关闭 1: 使能

13.5.2.2 PGA 模块写使能寄存器（PGAREGKEY）

表 13-6: PGA 模块写使能寄存器（PGAREGKEY）位段定义

PGAREGKEY (PGA Register Write-Allow Key Register)							
Access: PGA->PGAREGKEY		Offset: 0x02E0		Default: 0x1ACCE551			
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

表 13-7: PGA 模块写使能寄存器（PGAREGKEY）位段描述

位段	位段名	属性	复位值	描述
31:0	KEY	RW	0x1ACCE551	写入 0x1ACCE551 解锁受保护的 DPGA 寄存器

14 差分转单端缓冲器（D2S）

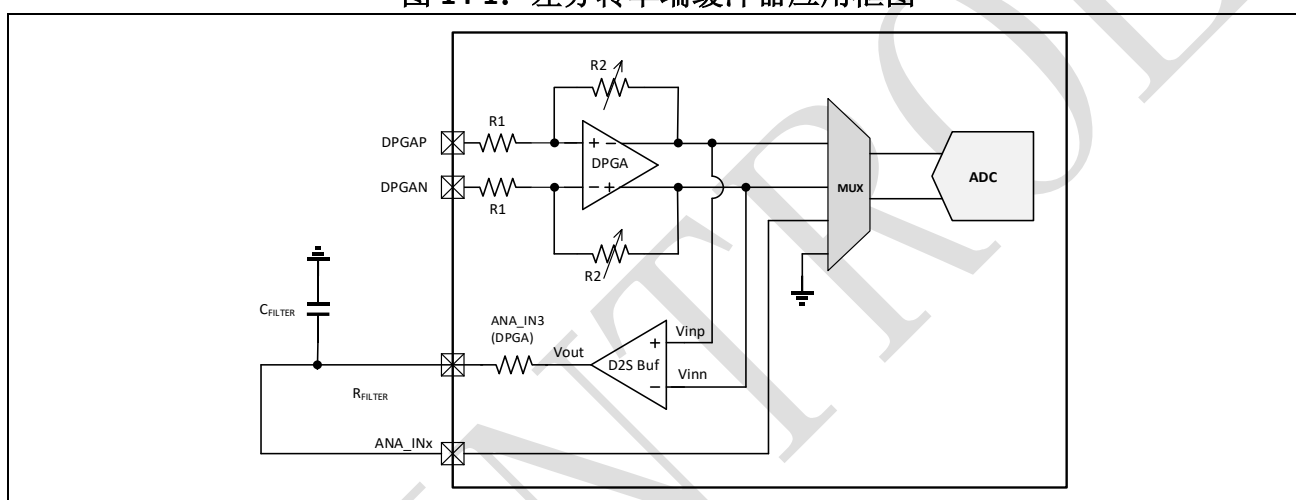
14.1 概述

SPD1121 提供 1 个 D2S（差分到单端）缓冲器，它可以将 DPGA 的差分输出转换为单端输出，并将其送到 ANA_IN3 引脚。通过外部的 RC 滤波器（通常， $R=1\text{k}\Omega$ ， $C=1\mu\text{F}$ ），可以将单端输出送回到其他 ANA_INx 引脚，并通过 13 位 ADC 进行测量。此时，13 位 ADC 可以获得感应电阻的直流平均电压。

14.2 功能描述

D2S 实际应用方式如图 14-1 所示。

图 14-1：差分转单端缓冲器应用框图



[1] 通常， $R_{\text{FILTER}} = 1\text{k}\Omega$ ， $C_{\text{FILTER}} = 1\mu\text{F}$ 。

D2S 缓冲器的增益固定为 0.5，D2S 缓冲器的输入和输出之间的方程式为：

$$V_{\text{out}} = (V_{\text{inp}} - V_{\text{inn}}) \times 0.5 + V_{\text{DD3}}/2$$

通常情况下， $V_{\text{DD3}}=3.3\text{V}$ ，因此输出电压为 $(V_{\text{inp}} - V_{\text{inn}}) \times 0.5 + 1.65$ 。DS2 缓冲器的输入范围为 0.3V 至 $V_{\text{DD3}} - 0.3\text{V}$ 。

注意：用户可以在 SDK 中调用 D2S_Enable()/D2S_Disable() 函数来使能/禁用 D2S 缓冲器。

15.3 模拟比较器输入选择

3 个模拟比较器（COMP0H/COMP0L/COMP1H）的输入选择如下表所示。

表 15-1：比较器输入通道选择

INSEL	COMP0L		COMP0H		COMP1H	
	输入信号	参考信号	输入信号	参考信号	输入信号	参考信号
0	D2SBUF	DAC1	D2SBUF	DAC0	D2SBUF	DAC2
1	—	DAC1	—	DAC0	—	DAC2
2	ANA_IN0	DAC1	ANA_IN0	DAC0	ANA_IN0	DAC2
3	ANA_IN1	DAC1	ANA_IN1	DAC0	ANA_IN1	DAC2
4	ANA_IN2	DAC1	ANA_IN2	DAC0	ANA_IN2	DAC2
5	ANA_IN3	DAC1	ANA_IN3	DAC0	ANA_IN3	DAC2
6	ANA_IN10	DAC1	ANA_IN10	DAC0	ANA_IN10	DAC2
7	ANA_IN11	DAC1	ANA_IN11	DAC0	ANA_IN11	DAC2

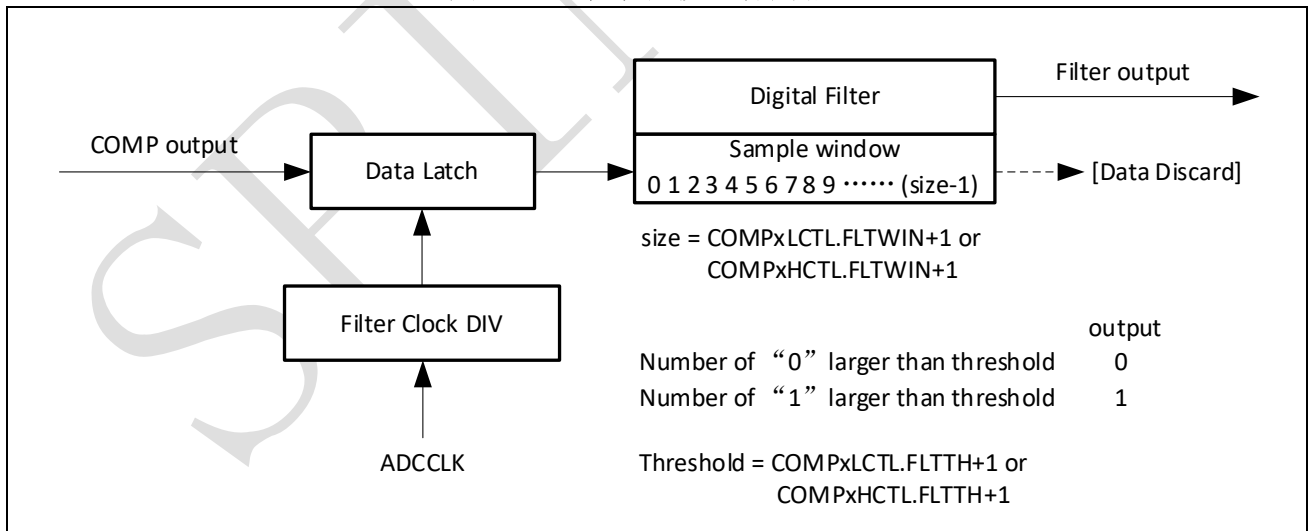
15.4 数字滤波器

比较器的输出被送到数字滤波器的采样窗口，采样时钟来自 ADCCLK，可由 COMPxLCTL.FLT DIV/COMPxHCTL.FLT DIV 来设置分频大小，窗口大小由 COMPxLCTL.FLTWIN/COMPxHCTL.FLTWIN 设置，最大为 32。通过 COMPxLCTL.FLTTH/COMPxHCTL.FLTTH 设置阈值的大小，数字滤波器可以计算采样窗口内 0 和 1 发生的次数，如果 0 发生的次数大于之前设置的的阈值大小，则数字滤波器的输出会变为 0，同理，如果 1 发生的次数大于阈值，数字滤波器的输出会变为 1。

为了使滤波器工作正常，需要保证设置的阈值 FLTTH 要大于 FLTWIN/2。

数字滤波器的行为描述如图 15-2 所示。

图 15-2：数字滤波器行为描述



15.5 数模转换器（DAC）

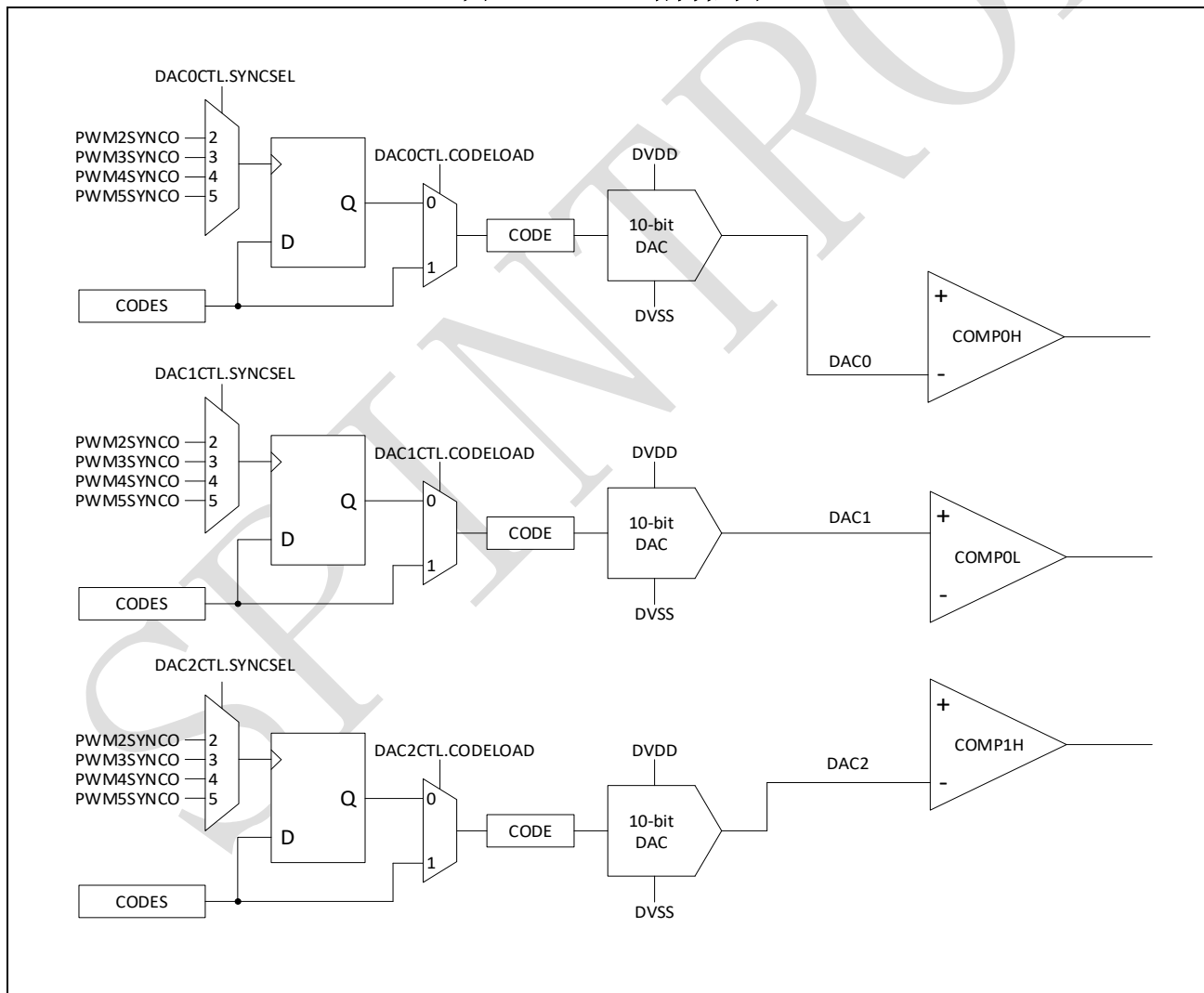
SPD1121 包含 3 个 10 比特 DAC，输出可由下面表达式计算：

$$V_{OUT} = \frac{DVDD}{2^{10}} * CODE \quad (DVDD \text{ 典型值为 } 3.3V)$$

DAC 结构框图如图 15-3 所示，DAC 有两组寄存器：CODES 和 CODE，其中 CODE 是只读寄存器，可以直接控制 10 比特 DAC 的输出。CODES 是用户可写的影子（shadow）寄存器，其值可以立即或者由下一个 PWMSYNC 事件同步，再送到 CODE 中，这取决于 DACxCTL.CODELOAD 的设置。如果 DACxCTL.CODELOAD=0（影子模式），用户改变 CODES 值，要等到下一个 PWMSYNC 信号来临后，才能传送到 CODE 中。如果 DACxCTL.CODELOAD=1（直接模式），用户改变 CODES 值，会直接送到 CODE 中，DAC 的输出也会立即变化。

比较器电压过高阈值和电压过低阈值分别由 3 个 DAC 产生，如何选择 DAC 输入请参考表 15-1。

图 15-3: DAC 结构框图



15.6 寄存器

15.6.1 COMP 寄存器列表

表 15-2: COMP 模块基地址

外设模块	基地址
COMP	0x4000 8C00

表 15-3: COMP 寄存器列表

寄存器	偏移地址	描述	复位值
COMPFLT	0x2F0	比较器滤波输出寄存器	0x00000000
COMPFLG	0x2F4	比较器状态寄存器	0x00000000
COMPCLR	0x2F8	比较器状态清除寄存器	0x00000000
COMPOCTL*	0x2FC	比较器 0 控制寄存器	0x00000000
COMPOLCTL*	0x300	COMPOL 控制寄存器	0x00000088
COMP0HCTL*	0x304	COMP0H 控制寄存器	0x00000088
COMP1CTL*	0x308	比较器 1 控制寄存器	0x00000000
COMP1HCTL*	0x310	COMP1H 控制寄存器	0x00000088
DAC0CTL*	0x35C	DAC0 控制寄存器	0x00000002
DAC0CODES	0x360	DAC0 码值寄存器	0x00000000
DAC0CODE	0x364	DAC0 有效码值寄存器	0x00000000
DAC1CTL*	0x384	DAC1 控制寄存器	0x00000002
DAC1CODES	0x388	DAC1 码值寄存器	0x00000000
DAC1CODE	0x38C	DAC1 有效码值寄存器	0x00000000
DAC2CTL*	0x3AC	DAC2 控制寄存器	0x00000002
DAC2CODES	0x3B0	DAC2 码值寄存器	0x00000000
DAC2CODE	0x3B4	DAC2 有效码值寄存器	0x00000000
COMPREGKEY	0x3FC	COMP 模块写使能寄存器	0x1ACCE551

注意： 由*标识的寄存器仅当 COMPREGKEY=0x1ACCE551 时才可以改写。

15.6.2 COMP 寄存器

15.6.2.1 比较器滤波输出寄存器（COMPFLT）

表 15-4：比较器滤波输出寄存器（COMPFLT）位段定义

COMPFLT (Comparator Filter Output Register)							
Access: COMP->COMPFLT Offset: 0x02F0 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED				COMP1H	RESERVED	COMP0H	COMP0L

表 15-5：比较器滤波输出寄存器（COMPFLT）位段描述

位段	位段名	属性	复位值	描述
31:4	RESERVED_31_4	RO	0x0	保留
3	COMP1H	RO	0x0	COMP1H 实时滤波状态 0: 滤波后的 COMP1H 输出为低电平 1: 滤波后的 COMP1H 输出为高电平
2	RESERVED_2	RO	0x0	保留
1	COMP0H	RO	0x0	COMP0H 实时滤波状态 0: 滤波后的 COMP0H 输出为低电平 1: 滤波后的 COMP0H 输出为高电平
0	COMP0L	RO	0x0	COMP0L 实时滤波状态 0: 滤波后的 COMP0L 输出为低电平 1: 滤波后的 COMP0L 输出为高电平

15.6.2.2 比较器状态寄存器 (COMPFLG)

表 15-6: 比较器状态寄存器 (COMPFLG) 位段定义

COMPFLG (Comparator Flag Register)							
Access: COMP->COMPFLG				Offset: 0x02F4		Default: 0x00000000	
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED				COMP1H	RESERVED	COMP0H	COMP0L

表 15-7: 比较器状态寄存器 (COMPFLG) 位段描述

位段	位段名	属性	复位值	描述
31:4	RESERVED_31_4	RO	0x0	保留
3	COMP1H	RO	0x0	锁存的 COMP1H 滤波结果标志 0: COMP1H 滤波结果一直为低 1: COMP1H 滤波结果曾经为高
2	RESERVED_2	RO	0x0	保留
1	COMP0H	RO	0x0	锁存的 COMP0H 滤波结果标志 0: COMP0H 滤波结果一直为低 1: COMP0H 滤波结果曾经为高
0	COMP0L	RO	0x0	锁存的 COMP0L 滤波结果标志 0: COMP0L 滤波结果一直为低 1: COMP0L 滤波结果曾经为高

15.6.2.3 比较器状态清除寄存器 (COMPCLR)

表 15-8: 比较器状态清除寄存器 (COMPCLR) 位段定义

COMPCLR (Comparator Flag Clear Register)							
Access: COMP->COMPCLR Offset: 0x02F8 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED				COMP1H	RESERVED	COMP0H	COMP0L

表 15-9: 比较器状态清除寄存器 (COMPCLR) 位段描述

位段	位段名	属性	复位值	描述
31:4	RESERVED_31_4	RO	0x0	保留
3	COMP1H	W1C	0x0	COMP1H 标志清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除标志, 本位段自动清零
2	RESERVED_2	RO	0x0	保留
1	COMP0H	W1C	0x0	COMP0H 标志清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除标志, 本位段自动清零
0	COMP0L	W1C	0x0	COMP0L 标志清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除标志, 本位段自动清零

15.6.2.4 比较器 0 控制寄存器 (COMP0CTL)

表 15-10: 比较器 0 控制寄存器 (COMP0CTL) 位段定义

COMP0CTL (Comparator 0 Control Register)							
Access: COMP->COMP0CTL Offset: 0x02FC Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED				SYNCSEL			RESERVED
7	6	5	4	3	2	1	0
INSEL			HYSTSEL		RESERVED	ENH	ENL

表 15-11: 比较器 0 控制寄存器 (COMP0CTL) 位段描述

位段	位段名	属性	复位值	描述
31:12	RESERVED_31_12	RO	0x0	保留
11:9	SYNCSEL	RW	0x0	选择 PWM 同步输出 2: PWM2 同步输出 3: PWM3 同步输出 4: PWM4 同步输出 5: PWM5 同步输出 其他: 没有选择同步输出
8	RESERVED_8	RO	0x0	保留
7:5	INSEL	RW	0x0	输入选择 0: 差分到单端缓冲区输出 2: ADC0 输入 (来自 GPIO0) 3: ADC1 输入 (来自 GPIO1) 4: ADC2 输入 (来自 GPIO2) 5: ADC3 输入 (来自 GPIO3) 6: ADC10 输入 (来自 GPIO10) 7: ADC11 输入 (来自 GPIO11)
4:3	HYSTSEL	RW	0x0	迟滞选择 0: 0 mV 1: 12 mV 2: 24 mV 3: 36 mV
2	RESERVED_2	RO	0x0	保留
1	ENH	RW	0x0	COMP0H 使能 0: 关闭 1: 使能

位段	位段名	属性	复位值	描述
0	ENL	RW	0x0	COMPOL 使能 0: 关闭 1: 使能

SPIN TROL

15.6.2.5 COMP0L 控制寄存器 (COMP0LCTL)

表 15-12: COMP0L 控制寄存器 (COMP0LCTL) 位段定义

COMP0LCTL (COMP0L Control Register)							
Access: COMP->COMP0LCTL				Offset: 0x0300		Default: 0x00000088	
31	30	29	28	27	26	25	24
RESERVED		FLTRST	FLTTH				
23	22	21	20	19	18	17	16
FLTWIN					FLTDIV		
15	14	13	12	11	10	9	8
FLTDIV							SYNCCLREN
7	6	5	4	3	2	1	0
POL4GPIO	SEL4GPIO			POL4PWM	SEL4PWM		

表 15-13: COMP0L 控制寄存器 (COMP0LCTL) 位段描述

位段	位段名	属性	复位值	描述
31:30	RESERVED_31_30	RO	0x0	保留
29	FLTRST	W1S	0x0	复位并初始化滤波器 0: 写 0 无效, 总是读回 0 1: 写 1 用当前输入值填充滤波器的寄存器链, 本位段自动清零
28:24	FLTTH	RW	0x0	当样本里有(FLTTH+1)个 0 或 1 时输出对应翻转
23:19	FLTWIN	RW	0x0	设置滤波样本数目为(FLTWIN+1)
18:9	FLTDIV	RW	0x0	滤波采样时钟为 ADC 时钟经过(FLTDIV+1)分频
8	SYNCCLREN	RW	0x0	使能 PWMSYNCO 清除 COMPFLG.COMP0L 标志 0: 关闭 1: 使能
7	POL4GPIO	RW	0x1	输出到 GPIO 的 COMP0L 输出极性 0: 与 COMP0L 输出极性相反 1: 与 COMP0L 输出极性相同
6:4	SEL4GPIO	RW	0x0	输出到 GPIO 的 COMP0L 输出选择 0: 原始信号 1: 经过时钟同步后的信号 2: 滤波输出的实时结果 3: 滤波输出锁存的标志 4: 原始输出或锁存的标志 5: 同步后的信号或锁存的标志 6: 滤波输出的结果或锁存的标志 7: 原始输出、同步后的信号或锁存的标志

位段	位段名	属性	复位值	描述
3	POL4PWM	RW	0x1	COMP0L 输出到 PWM 的封锁信号的极性 0: 与 COMP0L 输出极性相反 1: 与 COMP0L 输出极性相同
2:0	SEL4PWM	RW	0x0	COMP0L 输出到 PWM 的封锁信号的选择 0: 原始信号 1: 经过时钟同步后的信号 2: 滤波输出的实时结果 3: 滤波输出锁存的标志 4: 原始输出或锁存的标志 5: 同步后的信号或锁存的标志 6: 滤波输出的结果或锁存的标志 7: 原始输出、同步后的信号或锁存的标志

15.6.2.6 COMP0H 控制寄存器 (COMP0HCTL)

表 15-14: COMP0H 控制寄存器 (COMP0HCTL) 位段定义

COMP0HCTL (COMP0H Control Register)							
Access: COMP->COMP0HCTL Offset: 0x0304 Default: 0x00000088							
31	30	29	28	27	26	25	24
RESERVED		FLTRST	FLTTH				
23	22	21	20	19	18	17	16
FLTWIN					FLTDIV		
15	14	13	12	11	10	9	8
FLTDIV							SYNCCLREN
7	6	5	4	3	2	1	0
POL4GPIO	SEL4GPIO			POL4PWM	SEL4PWM		

表 15-15: COMP0H 控制寄存器 (COMP0HCTL) 位段描述

位段	位段名	属性	复位值	描述
31:30	RESERVED_31_30	RO	0x0	保留
29	FLTRST	W1C	0x0	复位并初始化滤波器 0: 写 0 无效, 总是读回 0 1: 写 1 用当前输入值填充滤波器的寄存器链, 本位段自动清零
28:24	FLTTH	RW	0x0	当样本里有(FLTTH+1)个 0 或 1 时输出对应翻转
23:19	FLTWIN	RW	0x0	设置滤波样本数目为(FLTWIN+1)
18:9	FLTDIV	RW	0x0	滤波采样时钟为 ADC 时钟经过(FLTDIV+1)分频
8	SYNCCLREN	RW	0x0	使能 PWMSYNCO 清除 COMPFLG.COMP0H 标志 0: 关闭 1: 使能
7	POL4GPIO	RW	0x1	输出到 GPIO 的 COMP0H 输出极性 0: 与 COMP0H 输出极性相反 1: 与 COMP0H 输出极性相同
6:4	SEL4GPIO	RW	0x0	输出到 GPIO 的 COMP0H 输出选择 0: 原始信号 1: 经过时钟同步后的信号 2: 滤波输出的实时结果 3: 滤波输出锁存的标志 4: 原始输出或锁存的标志 5: 同步后的信号或锁存的标志 6: 滤波输出的结果或锁存的标志 7: 原始输出、同步后的信号或锁存的标志
3	POL4PWM	RW	0x1	COMP0H 输出到 PWM 的封锁信号的极性 0: 与 COMP0H 输出极性相反 1: 与 COMP0H 输出极性相同

位段	位段名	属性	复位值	描述
2:0	SEL4PWM	RW	0x0	COMP0H 输出到 PWM 的封锁信号的选择 0: 原始信号 1: 经过时钟同步后的信号 2: 滤波输出的实时结果 3: 滤波输出锁存的标志 4: 原始输出或锁存的标志 5: 同步后的信号或锁存的标志 6: 滤波输出的结果或锁存的标志 7: 原始输出、同步后的信号或锁存的标志

15.6.2.7 比较器 1 控制寄存器 (COMP1CTL)

表 15-16: 比较器 1 控制寄存器 (COMP1CTL) 位段定义

COMP1CTL (Comparator 1 Control Register)							
Access: COMP->COMP1CTL Offset: 0x0308 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED				SYNCSEL			RESERVED
7	6	5	4	3	2	1	0
INSEL			HYSTSEL		RESERVED	ENH	RESERVED

表 15-17: 比较器 1 控制寄存器 (COMP1CTL) 位段描述

位段	位段名	属性	复位值	描述
31:12	RESERVED_31_12	RO	0x0	保留
11:9	SYNCSEL	RW	0x0	选择 PWM 同步输出 2: PWM2 同步输出 3: PWM3 同步输出 4: PWM4 同步输出 5: PWM5 同步输出
8	RESERVED_8	RO	0x0	保留
7:5	INSEL	RW	0x0	输入选择 0: 差分到单端缓冲区输出 2: ADC0 输入 (来自 GPIO0) 3: ADC1 输入 (来自 GPIO1) 4: ADC2 输入 (来自 GPIO2) 5: ADC3 输入 (来自 GPIO3) 6: ADC10 输入 (来自 GPIO10) 7: ADC11 输入 (来自 GPIO11)
4:3	HYSTSEL	RW	0x0	迟滞选择 0: 0 mV 1: 12 mV 2: 24 mV 3: 36 mV
2	RESERVED_2	RO	0x0	保留
1	ENH	RW	0x0	COMP1H 使能 0: 关闭 1: 使能
0	RESERVED_0	RO	0x0	保留

15.6.2.8 COMP1H 控制寄存器 (COMP1HCTL)

表 15-18: COMP1H 控制寄存器 (COMP1HCTL) 位段定义

COMP1HCTL (COMP1H Control Register)							
Access: COMP->COMP1HCTL Offset: 0x0310 Default: 0x00000088							
31	30	29	28	27	26	25	24
RESERVED		FLTRST	FLTTH				
23	22	21	20	19	18	17	16
FLTWIN					FLTDIV		
15	14	13	12	11	10	9	8
FLTDIV							SYNCCLREN
7	6	5	4	3	2	1	0
POL4GPIO	SEL4GPIO			POL4PWM	SEL4PWM		

表 15-19: COMP1H 控制寄存器 (COMP1HCTL) 位段描述

位段	位段名	属性	复位值	描述
31:30	RESERVED_31_30	RO	0x0	保留
29	FLTRST	W1C	0x0	复位并初始化滤波器 0: 写 0 无效, 总是读回 0 1: 写 1 用当前输入值填充滤波器的寄存器链, 本位段自动清零
28:24	FLTTH	RW	0x0	当样本里有(FLTTH+1)个 0 或 1 时输出对应翻转
23:19	FLTWIN	RW	0x0	设置滤波样本数目为(FLTWIN+1)
18:9	FLTDIV	RW	0x0	滤波采样时钟为 ADC 时钟经过(FLTDIV+1)分频
8	SYNCCLREN	RW	0x0	使能 PWMSYNCO 清除 COMPFLG.COMP1H 标志 0: 关闭 1: 使能
7	POL4GPIO	RW	0x1	输出到 GPIO 的 COMP1H 输出极性 0: 与 COMP1H 输出极性相反 1: 与 COMP1H 输出极性相同
6:4	SEL4GPIO	RW	0x0	输出到 GPIO 的 COMP1H 输出选择 0: 原始信号 1: 经过时钟同步后的信号 2: 滤波输出的实时结果 3: 滤波输出锁存的标志 4: 原始输出或锁存的标志 5: 同步后的信号或锁存的标志 6: 滤波输出的结果或锁存的标志 7: 原始输出、同步后的信号或锁存的标志

位段	位段名	属性	复位值	描述
3	POL4PWM	RW	0x1	COMP0L 输出到 PWM 的封锁信号的极性 0: 与 COMP1H 输出极性相反 1: 与 COMP1H 输出极性相同
2:0	SEL4PWM	RW	0x0	COMP1H 输出到 PWM 的封锁信号的选择 0: 原始信号 1: 经过时钟同步后的信号 2: 滤波输出的实时结果 3: 滤波输出锁存的标志 4: 原始输出或锁存的标志 5: 同步后的信号或锁存的标志 6: 滤波输出的结果或锁存的标志 7: 原始输出、同步后的信号或锁存的标志

15.6.2.9 DAC0 控制寄存器 (DAC0CTL)

表 15-20: DAC0 控制寄存器 (DAC0CTL) 位段定义

DAC0CTL (DAC0 Control Register)							
Access: COMP->DAC0CTL Offset: 0x035C Default: 0x00000002							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED			SYNCSEL			CODELOAD	EN

表 15-21: DAC0 控制寄存器 (DAC0CTL) 位段描述

位段	位段名	属性	复位值	描述
31:5	RESERVED_31_5	RO	0x0	保留
4:2	SYNCSEL	RW	0x0	PWM 同步输出信号选择 2: PWM2 同步输出 3: PWM3 同步输出 4: PWM4 同步输出 5: PWM5 同步输出
1	CODELOAD	RW	0x1	DAC 码值加载模式 0: DAC 码值在 PWMSYNC 事件发生时更新 1: DAC 码值在写 DAC0CODES 时立即更新
0	EN	RW	0x0	DAC 使能 0: 关闭 1: 使能

15.6.2.10 DAC0 码值影子寄存器 (DAC0CODES)

表 15-22: DAC0 码值影子寄存器 (DAC0CODES) 位段定义

DAC0CODES (DAC0 Code Shadow Register)							
Access: COMP->DAC0CODES Offset: 0x0360 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED						VAL	
7	6	5	4	3	2	1	0
VAL							

表 15-23: DAC0 码值影子寄存器 (DAC0CODES) 位段描述

位段	位段名	属性	复位值	描述
31:10	RESERVED_31_10	RO	0x0	保留
9:0	VAL	RW	0x0	DAC 码值影子值 当 DAC0CTL.CODELOAD=0 时, 写本位段, 只会影响影子码值 (shadow code)。 当 DAC0CTL.CODELOAD=1 时, 写本位段, 会同时影响影子码值 (shadow code) 和有效码值 (active code)。

15.6.2.11 DAC0 有效码值寄存器 (DAC0CODE)

表 15-24: DAC0 有效码值寄存器 (DAC0CODE) 位段定义

DAC0CODE (DAC0 Code Active Register)							
Access: COMP->DAC0CODE Offset: 0x0364 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED						VAL	
7	6	5	4	3	2	1	0
VAL							

表 15-25: DAC0 有效码值寄存器 (DAC0CODE) 位段描述

位段	位段名	属性	复位值	描述
31:10	RESERVED_31_10	RO	0x0	保留
9:0	VAL	RO	0x0	有效的 DAC0 码值

15.6.2.12 DAC1 控制寄存器（DAC1CTL）

表 15-26: DAC1 控制寄存器（DAC1CTL）位段定义

DAC1CTL (DAC1 Control Register)							
Access: COMP->DAC1CTL Offset: 0x0384 Default: 0x00000002							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED			SYNCSEL			CODELOAD	EN

表 15-27: DAC1 控制寄存器（DAC1CTL）位段描述

位段	位段名	属性	复位值	描述
31:5	RESERVED_31_5	RO	0x0	保留
4:2	SYNCSEL	RW	0x0	PWM 同步输出信号选择 2: PWM2 同步输出 3: PWM3 同步输出 4: PWM4 同步输出 5: PWM5 同步输出
1	CODELOAD	RW	0x1	DAC1 码值加载模式 0: DAC1 码值在 PWMSYNC 事件发生时更新 1: DAC1 码值在写 DAC1CODES 时立即更新
0	EN	RW	0x0	DAC1 使能 0: 关闭 1: 使能

15.6.2.13 DAC1 码值影子寄存器（DAC1CODES）

表 15-28: DAC1 码值影子寄存器（DAC1CODES）位段定义

DAC1CODE (DAC1 Code Register) Offset: 0x388 Default: 0x00000000							
Access: COMP -> DAC1CODE.all							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED						VAL	
7	6	5	4	3	2	1	0
VAL							

表 15-29: DAC1 码值影子寄存器（DAC1CODES）位段描述

位段	位段名	属性	复位值	描述
31:10	RESERVED_31_10	RO	0x0	保留
9:0	VAL	RW	0x0	DAC1 影子码值 当 DAC1CTL.CODELOAD=0 时，写本位段，只会影响影子码值（shadow code）。 当 DAC1CTL.CODELOAD=1 时，写本位段，会同时影响影子码值（shadow code）和有效码值（active code）。

15.6.2.14 DAC1 有效码值寄存器（DAC1CODE）

表 15-30: DAC1 有效码值寄存器（DAC1CODE）位段定义

DAC1CODE (DAC1 Code Active Register)							
Access: COMP->DAC1CODE Offset: 0x038C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED						VAL	
7	6	5	4	3	2	1	0
VAL							

表 15-31: DAC1 有效码值寄存器（DAC1CODE）位段描述

位段	位段名	属性	复位值	描述
31:10	RESERVED_31_10	RO	0x0	保留
9:0	VAL	RO	0x0	有效的 DAC1 码值

15.6.2.15 DAC2 控制寄存器（DAC2CTL）

表 15-32: DAC2 控制寄存器（DAC2CTL）位段定义

DAC2CTL (DAC2 Control Register)							
Access: COMP->DAC2CTL Offset: 0x03AC Default: 0x00000002							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED			SYNCSEL			CODELOAD	EN

表 15-33: DAC2 控制寄存器（DAC2CTL）位段描述

位段	位段名	属性	复位值	描述
31:5	RESERVED_31_5	RO	0x0	保留
4:2	SYNCSEL	RW	0x0	PWM 同步输出信号选择 2: PWM2 同步输出 3: PWM3 同步输出 4: PWM4 同步输出 5: PWM5 同步输出
1	CODELOAD	RW	0x1	DAC2 码值加载模式 0: DAC2 码值在 PWMSYNC 事件发生时更新 1: DAC2 码值在写 DAC2CODES 时立即更新
0	EN	RW	0x0	DAC2 使能 0: 关闭 1: 使能

15.6.2.16 DAC2 码值影子寄存器 (DAC2CODES)

表 15-34: DAC2 码值影子寄存器 (DAC2CODES) 位段定义

DAC2CODES (DAC2 Code Shadow Register)							
Access: COMP->DAC2CODES Offset: 0x03B0 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED						VAL	
7	6	5	4	3	2	1	0
VAL							

表 15-35: DAC2 码值影子寄存器 (DAC2CODES) 位段描述

位段	位段名	属性	复位值	描述
31:10	RESERVED_31_10	RO	0x0	保持
9:0	VAL	RW	0x0	DAC2 影子码值 当 DAC2CTL.CODELOAD=0 时, 写本位段, 只会影响影子码值 (shadow code)。 当 DAC2CTL.CODELOAD=1 时, 写本位段, 会同时影响影子码值 (shadow code) 和有效码值 (active code)。

15.6.2.17 DAC2 有效码值寄存器 (DAC2CODE)

表 15-36: DAC2 有效码值寄存器 (DAC2CODE) 位段定义

DAC2CODE (DAC2 Code Active Register)							
Access: COMP->DAC2CODE Offset: 0x03B4 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED						VAL	
7	6	5	4	3	2	1	0
VAL							

表 15-37: DAC2 有效码值寄存器 (DAC2CODE) 位段描述

位段	位段名	属性	复位值	描述
31:10	RESERVED_31_10	RO	0x0	保留
9:0	VAL	RO	0x0	有效的 DAC2 码值

15.6.2.18 COMP 模块写使能寄存器（COMPREGKEY）

表 15-38: COMP 模块写使能寄存器（COMPREGKEY）位段定义

COMPREGKEY (COMP Register Write-Allow Key Register)							
Access: COMP->COMPREGKEY				Offset: 0x03FC		Default: 0x1ACCE551	
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

表 15-39: COMP 模块写使能寄存器（COMPREGKEY）位段描述

位段	位段名	属性	复位值	描述
31:0	KEY	RW	0x1ACCE551	写入 0x1ACCE551 解锁受保护的 COMP 寄存器

16 通用异步收发器 (UART)

16.1 UART 概述

SPD1121 实现了一个通用异步收发器 (UART) 端口。用来执行对从外围设备接收到的数据字符进行串到并的转换, 对从 CPU 接收到的数据字符执行并到串的转换。

CPU 能够从线路状态寄存器 UARTLSR 中读到一个完整的 UART 状态。状态信息包括传输的类型和状态、错误状态 - 如奇偶校验, 溢出, 帧中断或者间断 (break) 中断。

UART 端口可以工作在 FIFO 模式下, 也可以工作在非 FIFO 模式下。在 FIFO 模式下, 一个 64 字节的 TXFIFO (Transmit FIFO) 保存来自 CPU 的数据直到它被发送到串行链路上; 一个 64 字节的 RXFIFO (Receive FIFO) 缓存来自串行链路上的数据直到它被 CPU 读取。在非 FIFO 模式下, TXFIFO 和 RXFIFO 是被绕过的, 使用 THR 寄存器 (Transmit Holding Register) 和 RBR 寄存器 (Receive Buffer Register) 作为替代。

UART 包括一个可编程的波特率产生器, 能够对输入时钟进行 $1 \sim (2^{16}-1)$ 分频。它能够产生一个 16 倍的时钟信号来能够驱动内部发送和接收逻辑。软件可以通过编写中断来满足它的需求, 这样可以最小化处理通信链路所需的计算量。UART 的工作既可以通过软件轮询来控制也可以通过中断来驱动。

UART 端口支持波特率有: 9600, 19.2K, 38.4K, 57.6K, 115.2K, 230K, 460K, 921K, 1.8M 以及 3.6M。

16.2 UART 主要特性

UART 模块具有以下特性:

- 在串行数据上添加或者删除标准异步通信位 (开始位, 停止位, 校验位);
- 独立控制的发送、接收、线路状态和数据集 (data-set) 中断;
- 可编程串行接口;
 - 5 - 8 个数据位;
 - 偶检验、奇校验或者无校验检测;
 - 支持 1 个、1.5 个以及 2 个停止位生成;
 - 自动波特率检测;
- 64 字节 TXFIFO;
- 64 字节 RXFIFO;
- 完备的状态报告功能;
- 产生和检测线路间断 (break) 的功能;
- 内部诊断功能, 包括:
 - 回环 (loopback) 控制: 用于通信链路故障隔离;
 - 间断、检验位以及帧错误仿真;
- 完全优先处理的中断系统控制;

16.3 UART 信号描述

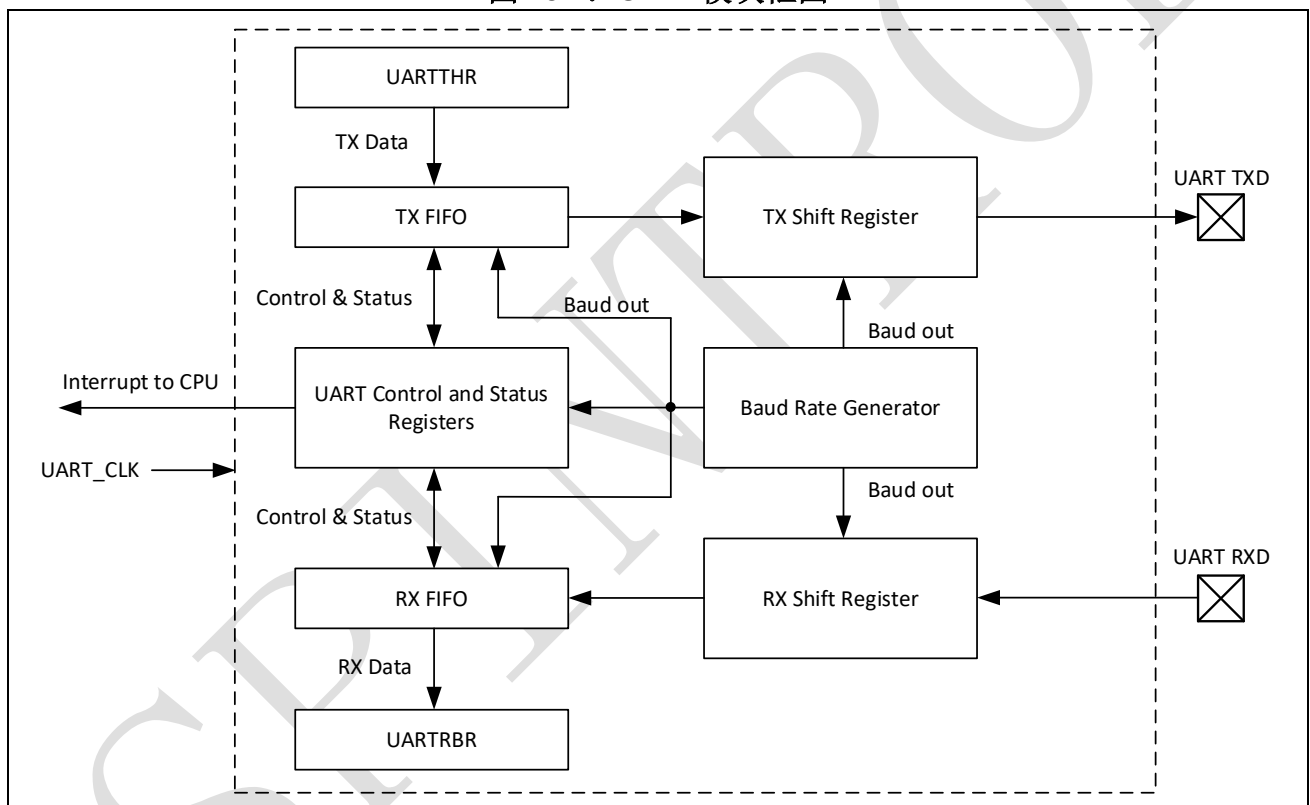
表 16-1 列出并描述了连接到 UART 模块的外部信号。传输数字 CMOS 电平信号的引脚通过 GPIO 连接。更多详细信息，请参考 GPIO 引脚配置寄存器。

表 16-1: UART 信号描述

名称	类型	描述
RXD	输入	串行输入 接收移位寄存器（Receive Shift register）的串行数据输入。
TXD	输出	串行输出 串行数据输出，复位操作后，TXD 信号被设置为逻辑“1”。

UART 模块的框图如图 16-1 所示。

图 16-1: UART 模块框图



16.4 UART 操作

图 16-2 展示了一个 UART 数据帧的格式。

图 16-2: UART 数据帧格式示例

Start Bit	Data <0>	Data <1>	Data <2>	Data <3>	Data <4>	Data <5>	Data <6>	Data <7>	Parity Bit	Stop Bit 1	Stop Bit 2
TXD or RXD pin											
	LSB							MSB			

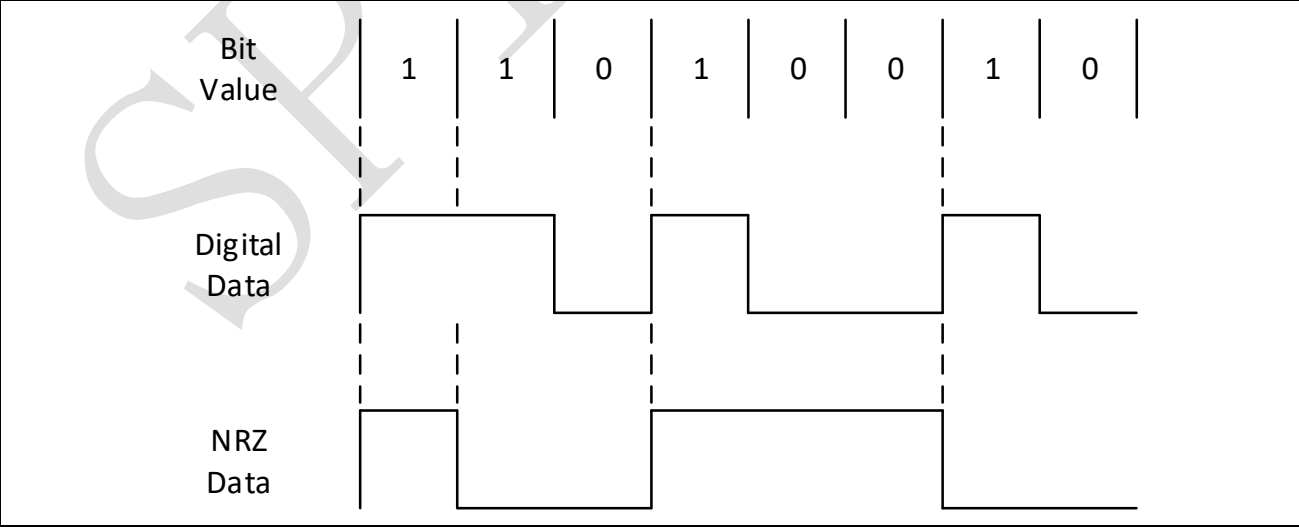
接收数据采样计数的频率是波特率数值的 16 倍。16 倍频的时钟是由波特率生成器产生的。每个比特位在中间采样 3 次。图 16-2 中带深色背景标记的比特位是可选的，可以通过软件编程来控制。

每个数据帧的长度在 9~11 位，取决于可编程的数据位大小以及是否使能奇偶校验。一个数据帧开始传输是通过传输一个开始位（指高电平到低电平的转换）。开始位后紧随着 8 位数据（LSB 先传输）。数据位后面是可选的校验位。校验位被置“1”：偶校验被使能且数据字节包含奇数个 1，或者奇校验被使能且数据字节包含偶数个 1。数据帧的结束是一个停止位。停止位是指逻辑 1 持续一个比特位的时间。

UART 模块有 2 个 FIFO：一个用于发送，一个用于接收。TXFIFO 是 64 字节深度，8 位宽度；RXFIFO 是 64 字节深度，11 位宽度，其中 3 位用于追踪错误。

UART 可以使用 NRZ 编码来表示单个比特位的值。将 UARTIER.NRZME 置“1”可以使能 NRZ 编码。比特 1 代表线路电平转换，比特 0 代表没有线路电平转换。图 16-3 展示了数据字节 0b0100_1011 的 NRZ 编码。这个字节的 LSB 位首先传输。

图 16-3: NRZ 编码示例 – 0b0100_1011



16.4.1 Reset 复位

UART 在复位后是关闭的。要想使能 UART，软件必须编程 GPIO 引脚配置寄存器，然后将 UARTIER.UUE 位段置“1”。当 UART 使能后，接收器等待一个帧的开始位；如果在 THR 寄存器中的数据是有效的，发送器进行数据发送。要被发送的数据可以在 UART 单元使能之前被写入到 THR 寄存器。在 FIFO 模式下，数据先被从 FIFO 发送到 THR 寄存器，然后再发送到管脚上。

当 UART 单元被关闭时，发送器或者接收器完成当前的字节传输后，会停止发送或者接收更多的数据。在 FIFO 中的数据不会被清除，当 UART 使能时，这些数据的传输会重新开始。

16.4.2 FIFO 操作

UART 模块有一个 TXFIFO 和一个 RXFIFO，每个 FIFO 能够保存 64 字节的数据。编程 I/O 是一种将数据移入或移出 FIFO 的方法。在编程 I/O 模式下，使用查询模式。

16.4.2.1 FIFO 中断模式操作

UARTMCR.OUT2 位段是 UART 中断的全局使能位，只有将该位段置“1”，才能使能 UART 中断。

16.4.2.2 接收中断

若要让一个接收中断可以发生，RXFIFO 和接收中断必须使能。当 FIFO 到达触发阈值时，UARTIIR.IID 位段的值会改变，表明接收数据是有效的。当 FIFO 掉到触发阈值以下时，UARTIIR.IID 位段的值会改变，表明在等待下一个中断。软件读取 UARTIIR.IID 位段的值，可以确定中断的产生原因。

接收线路状态（receive-line-status）中断有最高的优先级；接收数据有效（received-data-available）中断的优先级要低一些。只有当字符数据在被放入 FIFO 前出错，线路状态中断（line-status interrupt）才会产生。

当一个字符从移位寄存器传输到 RXFIFO 时，UARTLSR.DR 位段会被置“1”。当 FIFO 为空时，UARTLSR.DR 位段被清零。

注意： 如果使能了接收中断但是没有使能 RXFIFO，那么只要 UARTRBR 寄存器中接收到数据就会产生一个接收中断。

16.4.2.3 接收字符超时中断

当 RXFIFO 和接收超时中断被使能时，并且以下条件存在时，接收超时中断会发生：

- 最近一次收到的字符是在接收 4 个连续字符所需时间之前接收到的
- 最近一次 FIFO 读取是接收 4 个连续字符所需时间之前执行的

在从 RXFIFO 中读取一个字符之后或者接收到一个新的开始位，超时中断被清除并且超时计时器复位。如果超时中断没有发生，当接收到一个新的字符或者读取 RXFIFO 时，超时计时器复位。

16.4.2.4 发送中断

当 FIFO 使能时，且发送中断使能，才会产生发送中断；当 FIFO 关闭时，写一次 THR 寄存器就会产生一个新的发送中断。当 TXFIFO 至少是半空时，会发生发送数据请求中断（transmit data-request interrupt）。当 THR 寄存器被写或者 UARTIIR 寄存器被读时，中断被清除。

16.4.2.5 清除尾部字节（trailing bytes）

CPU 必须消除尾部字节。接收超时中断发出信号通知尾部字节的存在。当处理接收超时中断服务时，CPU 使用下面的过程：

1. 读取 UARTLSR 寄存器，检查错误。
2. 通过 UARTIER.RTOIE 位段关闭接收器超时中断。
3. 从 UART FIFO 中读取数据。
4. 读取 UARTLSR 寄存器，检查错误，然后转到步骤 3。如果 UARTLSR.DR 位段的值为“1”，转到步骤 5。
5. 当 FIFO 中不在有数据，通过 UARTIER.RTOIE 位段重新使能接收超时中断。
6. 完成。

16.4.2.6 FIFO 查询操作模式

当 FIFO 使能后，清除 UARTIER 寄存器的位[4:0]，会让 UART 端口工作于 FIFO 查询模式。接收器和发送器被分别控制，可以单独一个或者两个都处于查询模式。在查询模式下，软件通过 UARTLSR 寄存器检查接收器和发送器的状态。为了接收和传输数据的服，CPU 通过查询以下位段：

- 接收数据服务—CPU 检查 UARTLSR.DR 位段，当一或更多的字节保留在 RXFIFO 或者 RBR 寄存器中时，UARTLSR.DR 位段被置“1”。
- 发送数据服务—CPU 检查 UARTLSR.TDRQ 位段，当发送器需要数据时，UARTLSR.TDRQ 位段被置“1”。

CPU 也可以检查 UARTLSR.TXDONE 位段，当 TXFIFO 为空时，UARTLSR.TXDONE 位段被置“1”。

16.4.3 自动波特率检测

UART 模块支持自动波特率检测。当使能自动波特率检测后，UART 对开始位脉冲持续的时钟周期进行计数，然后将这个计数值写入到 UARTACR 寄存器中，用于计算波特率。当 UARTACR 寄存器被写后，可以产生自动波特率锁定（auto-baud-lock）中断（如果该中断被使能）。UART 可以根据波特率的值，自动编程分频锁存寄存器（Divisor Latch Registers）。如果需要的话，CPU 可以读取 UARTACR 寄存器的值，并使用这个信息计算波特率以及编程 UARTDLL 和 UARTDLH 寄存器。在波特率被编程之后，CPU 通过预定的字符（通常是 AT 或 at）被正确的接收进行验证。

如果 UART 要去编程分频锁存寄存器，软件可以基于公式来计算波特率。在大多数商业电子产品中见到的波特率值被称为“普通的”波特率，如表 16-2 所示。表 16-2 所示的波特率都可以通过 UART 来编程。基于公式来计算波特率的方法适用于波特率较高的情形。但是，如果远程发送器的实际波特率误差大于 1%时，在波特率低于 28.8kbps 的情形下，通信可能会失败。

当波特率被检测到后，自动波特率电路可以通过清楚 UARTACR.CNTVAL 位段来关闭自身。如果要重新使能自动波特率检测，将 UARTABR.ABE 位段再次置“1”即可。

- 注意：
- 1. 在发送或者接收数据时，波特率不可变。
 - 2. UART 配置完成且使能之后，才允许配置自动波特率功能。

16.4.4 可编程的波特率产生器

UART 包含了一个可编程波特率产生器。它将一个固定的输入时钟除进行分频，用于产生想要的波特率。波特率采用 UART 外设时钟来计算。

波特率产生器的输出频率是波特率的 16 倍。2 个 8 位的分频锁存寄存器（UARTDLL 和 UARTDLH）以二进制格式存储 16 位的分频值。在初始化期间，需要加载这些分频值，以确保波特率产生器正确地工作。如果任意一个分频锁存寄存器被写入 0x0，16 倍频的时钟会停止。被移入或移出 UART 的数据的波特率由下面的式给出：

$$\text{BaudRate} = \frac{\text{UART\_CLK}}{16 \times \text{Divisor}}$$

例如，如果分频值是 24，且 UART 时钟为 24 MHz，波特率为 62500bps。

注意： 在发送或者接收数据时，不允许改变波特率。

参考表 16-2，表中是一系列推荐的波特率值，基于分频值（UARTDLH:UARTDLL）。

复位后，分频值为 0x0002。在发送或者接收数据时，不允许改变波特率（写寄存器 UARTDLH 和 UARTDLL）。

表 16-2：推荐的波特率

推荐的波特率（bps）
9600
19200
38400
57600
115200
230400
460800
921600
1842000
3686400

16.5 寄存器

16.5.1 UART 寄存器列表

表 16-3: UART 模块基地址

外设模块	基地址
UART	0x4000 4000

表 16-4: UART 寄存器列表

寄存器	偏移地址	描述	复位值
UARTBR	0x00	UART 接收缓存寄存器	0x00000000
UARTTHR	0x00	UART 发送保持寄存器	0x00000000
UARTDLL	0x00	UART 分频系数低字节寄存器	0x00000002
UARTDLH	0x04	UART 分频系数高字节寄存器	0x00000000
UARTIER	0x04	UART 中断使能寄存器	0x00000000
UARTIIR	0x08	UART 中断识别寄存器	0x00000001
UARTFCR	0x08	UART FIFO 控制寄存器	0x00000000
UARTLCR	0x0C	UART 线路控制寄存器	0x00000000
UARTMCR	0x10	UART 调制解调器控制寄存器	0x00000000
UARTLSR	0x14	UART 线路状态寄存器	0x00000060
UARTFOR	0x24	UART 接收 FIFO 占用寄存器	0x00000000
UARTABR	0x28	UART 自动波特率控制寄存器	0x00000000
UARTACR	0x2C	UART 自动波特率计数寄存器	0x00000000

16.5.2 UART 寄存器

表 16-5 到表 16-30 提供了 UART 模块相关寄存器的详细信息。

16.5.2.1 UART 接收缓存寄存器（UARTBR）

表 16-5: UART 接收缓存寄存器（UARTBR）位段定义

UARTBR (UART Receive Buffer Register)							
Access: UART->UARTBR		Offset: 0x0000		Default: 0x00000000			
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 16-6: UART 接收缓存寄存器（UARTBR）位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0x0	读此寄存器获取接收的数据。

16.5.2.2 UART 发送保持寄存器（UARTTHR）

表 16-7: UART 发送保持寄存器（UARTTHR）位段定义

UARTTHR (UART Transmit Holding Register)							
Access: UART->UARTTHR		Offset: 0x0000		Default: 0x00000000			
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 16-8: UART 发送保持寄存器（UARTTHR）位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	WO	0x0	写此寄存器设置要发送的数据

16.5.2.3 UART 分频系数低字节寄存器 (UARTDLL)

表 16-9: UART 分频系数低字节寄存器 (UARTDLL) 位段定义

UARTDLL (UART Divisor Latch Low Byte Register)							
Access: UART->UARTDLL Offset: 0x0000 Default: 0x00000002							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
VAL							

表 16-10: UART 分频系数低字节寄存器 (UARTDLL) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7:0	VAL	RW	0x2	波特率分频系数低字节

16.5.2.4 UART 分频系数高字节寄存器 (UARTDLH)

表 16-11: UART 分频系数高字节寄存器 (UARTDLH) 位段定义

UARTDLH (UART Divisor Latch High Byte Register)							
Access: UART->UARTDLH Offset: 0x0004 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
VAL							

表 16-12: UART 分频系数高字节寄存器 (UARTDLH) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7:0	VAL	RW	0x0	波特率分频系数高字节

16.5.2.5 UART 中断使能寄存器 (UARTIER)

表 16-13: UART 中断使能寄存器 (UARTIER) 位段定义

UARTIER (UART Interrupt Enable Register)							
Access: UART->UARTIER		Offset: 0x0004		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	UUE	NRZME	RTOIE	RESERVED	RLSE	TXIE	RXIE

表 16-14: UART 中断使能寄存器 (UARTIER) 位段描述

位段	位段名	属性	复位值	描述
31:7	RESERVED_31_7	RO	0x0	保留
6	UUE	RW	0x0	使能 UART 单元 0: 关闭 UART 单元 1: 使能 UART 单元
5	NRZME	RW	0x0	使能 NRZM 编码 0: 关闭 NRZM 编码 采用通用的电平编码，低电平为 0，高电平为 1 1: 使能 NRZM 编码 低电平时信号保持不变，高电平时信号翻转
4	RTOIE	RW	0x0	使能接收器超时中断（触发源 UARTIIR.TOD） 0: 关闭接收器数据超时中断 1: 使能接收器数据超时中断
3	RESERVED_3	RW	0x0	保留
2	RLSE	RW	0x0	使能接收器线路状态中断（触发源 UARTIIR.IID） 0: 关闭接收器线路状态中断 1: 使能接收器线路状态中断
1	TIE	RW	0x0	使能发送数据请求中断（触发源 UARTIIR.IID） 0: 关闭 TXFIFO 数据请求中断 1: 使能 TXFIFO 数据请求中断
0	RAVIE	RW	0x0	使能接收器数据有效中断（触发源 UARTIIR.IID） 0: 关闭接收器数据有效中断 （接收字节数达到阈值） 1: 使能接收器数据有效中断 （接收字节数达到阈值）

16.5.2.6 UART 中断识别寄存器 (UARTIIR)

表 16-15: UART 中断识别寄存器 (UARTIIR) 位段定义

UARTIIR (UART Interrupt Identification Register)							
Access: UART->UARTIIR Offset: 0x0008 Default: 0x00000001							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
FIFOSTS		RESERVED	ABL	TOD	IID		NIP

表 16-16: UART 中断识别寄存器 (UARTIIR) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7:6	FIFOSTS	RO	0x0	FIFO 模式使能状态 0: 非 FIFO 模式 3: FIFO 模式
5	RESERVED_5	RO	0x0	保留
4	ABL	RC	0x0	自动波特率锁定 0: 自动波特率电路未编程分频系数锁存寄存器 1: 自动波特率电路已编程分频系数锁存寄存器
3	TOD	RC	0x0	检测到超时中断 0: 没有挂起的超时中断 1: 超时中断挂起 (仅 FIFO 模式有效)
2:1	IID	RC	0x0	中断触发源 0: 无效选项 1: 请求要发送数据 2: 接收到的数据有效 3: 接收错误 (溢出/校验/帧/间断/FIFO 错误)
0	NIP	RC	0x1	无中断挂起 0: 有中断挂起 (低有效) 1: 无中断挂起

16.5.2.7 UART FIFO 控制寄存器 (UARTFCR)

表 16-17: UART FIFO 控制寄存器 (UARTFCR) 位段定义

UARTFCR (UART FIFO Control Register)							
Access: UART->UARTFCR		Offset: 0x0008		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RXTH		BUS32	RESERVED	TXTH	CLRTF	CLRRF	TRFIFOE

表 16-18: UART FIFO 控制寄存器 (UARTFCR) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7:6	RXTH	WO	0x0	接收数据读取请求中断阈值 当 FIFO 的字节数等于或大于设置的阈值时触发中断。 0: FIFO 中有 1 个或更多个字节时产生中断 1: FIFO 中有 8 个或更多个字节时产生中断 2: FIFO 中有 16 个或更多个字节时产生中断 3: FIFO 中有 32 个或更多个字节时产生中断
5	BUS32	WO	0x0	总线宽度访问时发送和接收的数据 0: 8 位 忽略 UARTRBR 和 UARTTHR 的 MSB。 1: 32 位
4	RESERVED_4	WO	0x0	保留
3	TXTH	WO	0x0	发送中断触发阈值 0: 当 FIFO 半空时, 产生中断请求 1: 当 FIFO 为空时, 产生中断请求
2	CLRTF	W1C	0x0	清空 TXFIFO 它不影响传输移位寄存器中的数据。 发送数据请求事件在发送 FIFO 清除时产生。 0: 写 0 无效, 读总是读回 0 1: 写 1 清空 TXFIFO 本位段自动清零。
1	CLRRF	W1C	0x0	清空 RXFIFO 它不影响接收移位寄存器中的数据。 在接收 FIFO 清除时, UARTLSR 寄存器中的

位段	位段名	属性	复位值	描述
				FIFOERR 和 DR 位以及接收数据读取请求将被清除。 0: 写 0 无效，读总是读回 0 1: 写 1 清空 TXFIFO 本位段自动清零。
0	TRFIFOE	WO	0x0	使能 TXFIFO 和 RXFIFO 当两个 FIFO 被禁用时，它们中的数据将被清除。 0: FIFO 被关闭 1: FIFO 被使能

16.5.2.8 UART 线路控制寄存器 (UARTLCR)

表 16-19: UART 线路控制寄存器 (UARTLCR) 位段定义

UARTLCR (UART Line Control Register)							
Access: UART->UARTLCR		Offset: 0x000C		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
DLAB	SB	STKYP	EPS	PEN	STB	WLS	

表 16-20: UART 线路控制寄存器 (UARTLCR) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	DLAB	RW	0x0	分频寄存器访问锁定位 在读或者写操作期间，必须置“1”，以便可以访问波特率产生器的分频系数锁存寄存器（Divisor Latch registers）。当要访问 RBR 寄存器，THR 寄存器或者 UARTIER 寄存器时，必须清零。 0: 访问 UARTRBR，UARTTHR 和 UARTIER 1: 访问分频系数锁存寄存器（UARTDLL 和 UARTDLH）
6	SB	RW	0x0	设置间断（Break） 发送一个 break 信号给接收的 UART。只在 TXD 引脚上起作用，对发送逻辑没有影响。在 FIFO 模式下，要等到发送器空闲时（UARTLSR.TXDONE=1）才可以设置和清除 SB 位。 0: TXD 输出没有影响 1: 强制 TXD 输出为 0
5	STKYP	RW	0x0	粘滞校验（Sticky Parity） 本位段在 PEN=0 时，会被忽略。 0: 校验位没有影响 1: 强制校验位的值与 EPS 位的值相反
4	EPS	RW	0x0	偶校验选择 本位段在 PEN=0 时，会被忽略。 0: 发送或者检查奇校验 1: 发送或者检查偶校验

位段	位段名	属性	复位值	描述
3	PEN	RW	0x0	使能校验位 0: 无校验位 1: 使能检验位
2	STB	RW	0x0	停止位 0: 1 位停止位 1: 1.5 位停止位如果数据位为 5 位 2 位停止位如果数据位为 6/7/8 位
1:0	WLS	RW	0x0	每次传输的数据位长度 0: 5 位字符 1: 6 位字符 2: 7 位字符 3: 8 位字符

16.5.2.9 UART 调制解调器控制寄存器 (UARTMCR)

表 16-21: UART 调制解调器控制寄存器 (UARTMCR) 位段定义

UARTMCR (UART Modem Control Register) Access: UART->UARTMCR Offset: 0x0010 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED			RESERVED	GIE	RESERVED		

表 16-22: UART 调制解调器控制寄存器 (UARTMCR) 位段描述

位段	位段名	属性	复位值	描述
31:5	RESERVED_31_5	RO	0x0	保留
4	RESERVED_4	RW	0x0	保留
3	GIE	RW	0x0	使能全局中断 0: 关闭 UART 中断到 CPU 1: 使能 UART 中断到 CPU
2:0	RESERVED_2_0	RO	0x0	保留

16.5.2.10 UART 线路状态寄存器（UARTLSR）

表 16-23: UART 线路状态寄存器（UARTLSR）位段定义

UARTLSR (UART Line Status Register)							
Access: UART->UARTLSR		Offset: 0x0014		Default: 0x00000060			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
FIFOERR	TXDONE	TDRQ	BI	FE	PE	OE	DR

表 16-24: UART 线路状态寄存器（UARTLSR）位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	FIFOERR	RO	0x0	FIFO 错误状态 在非 FIFO 模式下，本位段总是 0。在 FIFO 模式下，当 FIFO 中的任一数据有至少一个校验错误，帧错误，或者 break 标志，FIFOERR 位段置“1”。CPU 读 UARTLSR 寄存器不会复位本位段。当所有的出错的数据都被从 FIFO 中读出后，FIFOERR 位段复位。 0: 非 FIFO 模式或者 RXFIFO 中没有错误 1: RXFIFO 中至少有一个数据有错误
6	TXDONE	RO	0x1	发送完成 当 THR（ransmit Holding Register）和 TSR（Transmit Shift Register）都为空时，本位段置“1”。当 UARTTHR 或者 TSR 包含一个数据时，本位段清零。在 FIFO 模式下，TXFIFO 和 TSR 都为空时，本位段置“1”。 0: TSR、THR 或者 FIFO 中有数据 1: 发送器中所有数据都被移出了（Shift out）
5	TDRQ	RO	0x1	发送数据请求 表示 UART 准备好接受一个新的数据用于发送。另外，当 UARTIER.TIE=1 时，本位段可以导致 UART 发给 CPU 一个中断。当一个字符数据从 THR 传输到 TSR 时，TDRQ 位段置“1”。当加载数据到 THR 时，本位段清零。在 FIFO 模式下，当 FIFO 中字符数据的一半已经加载到 TSR 中或者 UARTFCR.CLRTF 位段置“1”时，TDRQ 位段置“1”。当 FIFO 已有一半

位段	位段名	属性	复位值	描述
				以上的数据，本位段清零。当加载到 FIFO 中的数据超过 64 个字符时，超过的字符会丢失掉。 0: THR 或者 FIFO 有数据等待被移位出去 1: TXFIFO 中只有一半或者小于一半的数据
4	BI	RO	0x0	接收中断（break）中断 当接收数据的输入低电平保持时间比全“字”的传输时间（起始位+数据位+校验位+停止位的总时间）还长时，BI 位段置“1”。当 CPU 读 UARTLSR 寄存器时，BI 位段被清零。在 FIFO 模式下，只需一个等于 0x00 的字符被加载到 FIFO 中，不管 break 信号的长度。BI 展示了在 FIFO 前部字符的 break 状态，而不是最近接收到的字符。 0: 未接收到 break 信号 1: 已接收到 break 信号
3	FE	RO	0x0	接收帧错误 帧错误 FE 是指接收到的字符没有有效的停止位。当检测到最后一个数据位或者校验位之后的比特位为 0 时，FE 置“1”。当 CPU 读 UARTLSR 寄存器，FE 被清零。在一个帧错误之后，UART 会重新同步。为了重新同步，UART 会假设帧错误是由于下一个开始位，因此，UART 会采样这个开始位两次，然后再读到数据里。在 FIFO 模式下，FE 表示的是 FIFO 前部字符的帧错误，而不是最近接收到的字符。 0: 没有帧错误 1: 检测到无效的停止位
2	PE	RO	0x0	接收校验错误 表示接收到的数据字符没有正确的偶校验或奇校验位（由 UARTLCR.EPS 位来选择）。一旦检测到一个检验错误，PE 立即被置“1”。当 CPU 读 UARTLSR 寄存器时，本位段被清零。在 FIFO 模式下，PE 表示的是 FIFO 前部字符的校验错误，而不是最近接收到的字符。 0: 没有检验错误 1: 发生了检验错误
1	OE	RO	0x0	接收溢出错误 在非 FIFO 模式下，表示在下一个字符被接收之前，RBR（Receive Buffer Register）中的数据没有被 CPU 读取。新的字符会丢失。在 FIFO 模式下，OE 表示 FIFO 的 64 字节空间已满，最近接收到的

位段	位段名	属性	复位值	描述
				字节已被丢弃。一旦检测到一个溢出信号，OE 立即被置“1”。当 CPU 读 UARTLSR 寄存器时，本位段被清零。 0: 没有数据溢出 1: 接收数据已溢出
0	DR	RO	0x0	接收数据就绪（例如，非空） 当一个完整的字符被接收到并被传输到 RBR 或者 FIFO 时，本位段置“1”。在非 FIFO 模式下，当 RBR 被读取时，DR 位段被清零。在 FIFO 模式下，如果 FIFO 为空（最后一个字符被从 UATRBR 中读取）或者使用 UARTFCR.CLRRF 将 FIFO 复位，DR 位段被清零。 0: 没有接收到数据 1: UARTBR 或者 FIFO 中的数据是有效的

16.5.2.11 UART 接收 FIFO 占用寄存器（UARTFOR）

表 16-25: UART 接收 FIFO 占用寄存器（UARTFOR）位段定义

UARTFOR (UART Receive FIFO Occupancy Register)							
Access: UART->UARTFOR Offset: 0x0024 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	BYTECNT						

表 16-26: UART 接收 FIFO 占用寄存器（UARTFOR）位段描述

位段	位段名	属性	复位值	描述
31:7	RESERVED_31_7	RO	0x0	保留
6:0	BYTECNT	RO	0x0	RXFIFO 中保留的字节数（0~64）

16.5.2.12 UART 自动波特率控制寄存器 (UARTABR)

表 16-27: UART 自动波特率控制寄存器 (UARTABR) 位段定义

UARTABR (UART Auto-Baud Control Register)							
Access: UART->UARTABR		Offset: 0x0028		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED					ABUP	ABLIE	ABE

表 16-28: UART 自动波特率控制寄存器 (UARTABR) 位段描述

位段	位段名	属性	复位值	描述
31:3	RESERVED_31_3	RO	0x0	保留
2	ABUP	RW	0x0	编程 DLR (Divisor Latch Registers) 的方法 0: 软件编程 DLR 1: UART 编程 DLR
1	ABLIE	RW	0x0	使能自动波特率锁定中断 (触发源 UARTIIR.ABL) 0: 关闭自动波特率锁定中断 1: 使能自动波特率锁定中断
0	ABE	RW	0x0	使能自动波特率检测 0: 关闭自动波特率检测 1: 使能自动波特率检测

16.5.2.13 UART 自动波特率计数寄存器 (UARTACR)

表 16-29: UART 自动波特率计数寄存器 (UARTACR) 位段定义

UARTACR (UART Auto-Baud Count Register)							
Access: UART->UARTACR Offset: 0x002C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
CNTVAL							
7	6	5	4	3	2	1	0
CNTVAL							

表 16-30: UART 自动波特率计数寄存器 (UARTACR) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15:0	CNTVAL	RO	0x0	UART 起始位的时钟周期数

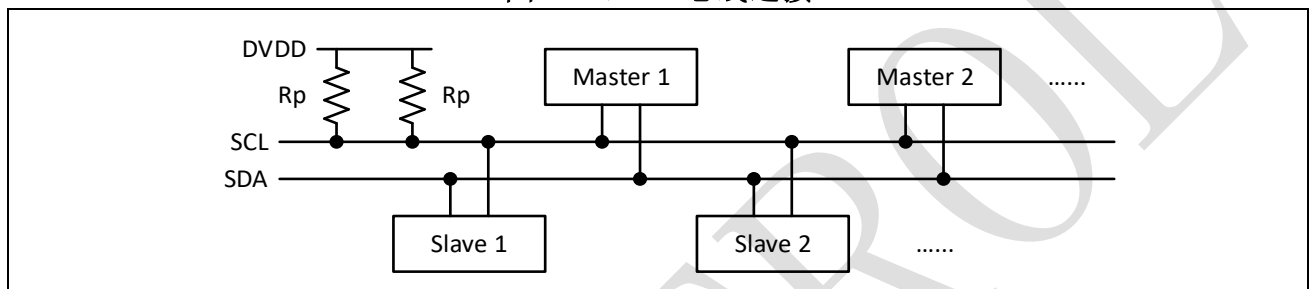
17 内部集成电路总线（I2C）

17.1 概述

I2C 模块为 MCU 和任何通过 I2C 串行总线相连的 I2C 总线兼容设备之间提供了一个接口。连接在 I2C 总线上的外部设备，通过两线 I2C 接口能够串行地发送数据或者接收数据。

I2C 总线连接如图 17-1 所示，支持多主机从机模式。每个 I2C 从机设备通过其唯一的地址来区分。I2C 设备既可以作为发送器，也可以作为接收器。需要注意的是，主机是在总线上发起数据传输并生成时钟信号以允许数据传输的设备。

图 17-1：I2C 总线连接



SPD1121 提供了 1 个 I2C 模块，其特性如下：

- 兼容《I2C-Bus Specification and User Manual Rev6》；
- 支持多种速度模式
 - 标准模式（最高 100Kbps）；
 - 快速及快速增强模式（最高 1Mbps）；
 - 高速模式（最高 3.4Mbps）；
- 支持主机和从机；
- 支持 7 位或 10 位寻址模式；
- 支持 7 位或 10 位组合格式传输；
- 支持发送异常中止的侦测；
- 收发各有 16 字节的 FIFO。

17.2 I2C 端口信号

表 17-1 罗列并描述了外部端口信号 SCL 和 SDA 如何连接到 I2C 模块。它们采用通道复用最终连到 GPIO 引脚。参考章节 4 配置引脚通道功能。

表 17-1：I2C 端口信号

信号名	类型	描述
SCL	输入/输出	I2C 时钟线 开漏输出（内部有约 41 kΩ 上拉）
SDA	输入/输出	I2C 数据线 开漏输出（内部有约 41 kΩ 上拉）

17.3 I2C 协议

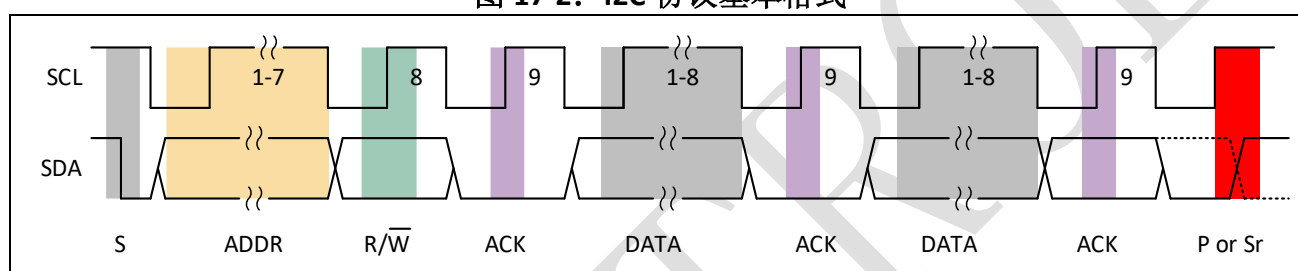
17.3.1 字节格式

I2C 的字节格式有以下特点：

- 以字节为传输单位，且从最高位开始传输；
- 每次传输的字节数不受限制；
- 每个字节后都要有一个应答位，且应答位由接收机发出。

I2C 传输基本格式如图 17-2，其中开始（S）、重复开始（Sr）、从机地址（ADDR）、读写控制位（R/W）和停止（P）都是由主机发出。I2C 总是以开始作为数据传输的起点，以停止作为传输的终点。

图 17-2：I2C 协议基本格式



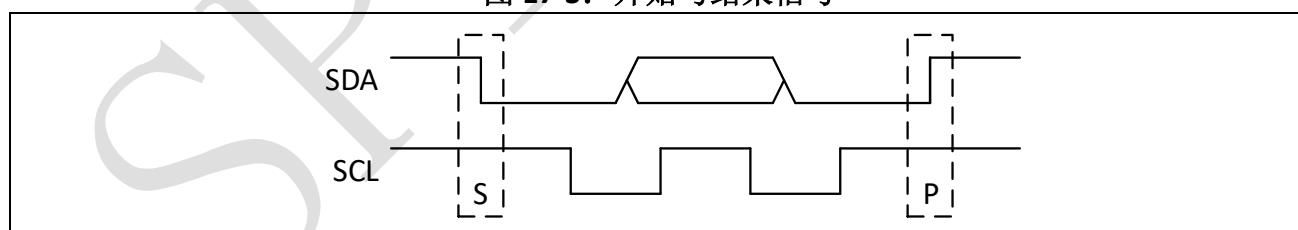
17.3.2 开始信号与停止信号

SDA 在 SCL 为高电平期间出现下降沿为开始信号，表示数据传输开始。I2C 模块配置为主机模式时，当发送 FIFO 不为空，则发出开始信号。

SDA 在 SCL 为高电平期间出现上升沿为停止信号，表示数据传输结束。I2C 模块配置为主机模块时，当发送 FIFO 为空，则发出停止信号。

数据传输时，SDA 仅在 SCL 为低电平期间发生改变。

图 17-3：开始与结束信号



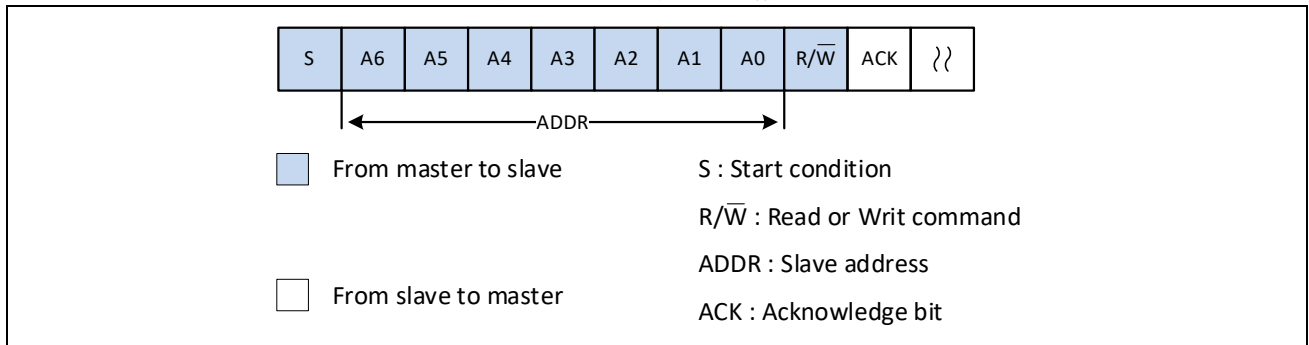
17.3.3 地址模式

I2C 模块主机模式寻址由 I2CMasterADDR 配置；从机模式下地址由 I2CSLVAADDR、I2CCTL.SLVAADDR10B 配置。

主机模式下发送 FIFO 非空则进行寻址并发送 FIFO 中的数据。

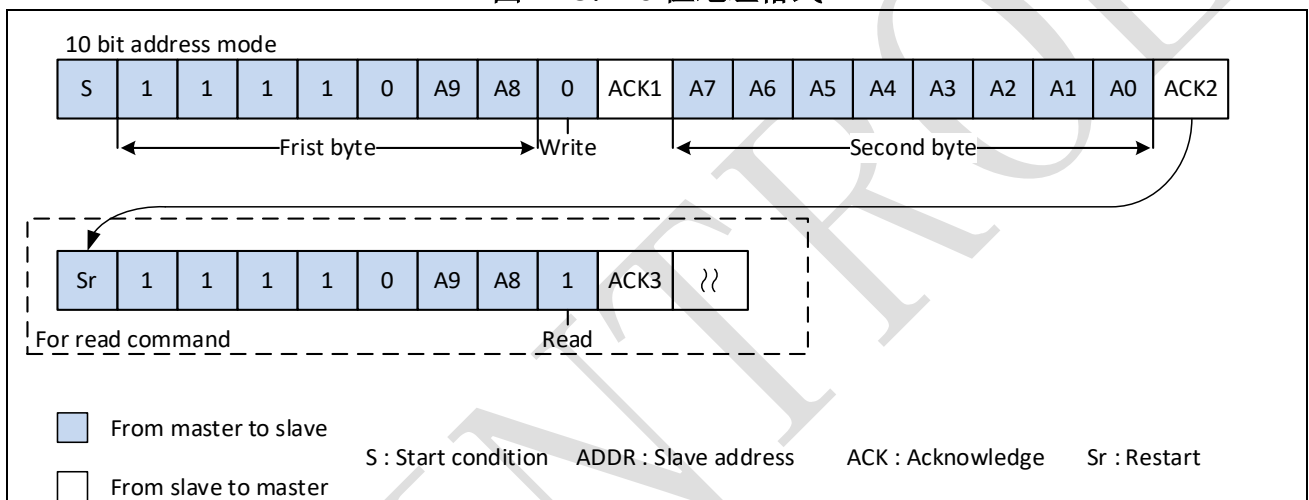
7 位地址寻址格式如图 17-4 所示，7 位地址之后是一个读写控制位，表示主机对从机的操作：0 表示写，1 表示读。当地址匹配，从机发出地址应答。

图 17-4: 7 位地址格式



10 位地址寻址格式图 17-5 如所示，10 位地址需要两个字节表示。若主机读从机数据，则需要发送虚框中的第三个字节。

图 17-5: 10 位地址格式



17.3.4 应答与非应答

在传输完一个字节之后为应答位，接收机的 SDA 发出低电平表示应答，高电平表示非应答。非应答会在下面情况产生：

- 从机地址不匹配；
- 主机接收完毕；
- 从机作为接收机，接收 FIFO 满；
- 高速主机码之后的应答位。

17.4 I2C 时序参数

17.4.1 电平时间配置

各速度模式下的 SCL 电平时间由表 17-2 确定，由此决定 SCL 的周期 $T_{SCL} = t_{low}(SCL) + t_{high}(SCL)$ 。表中的 T_{CLK_I2C} 为 I2C 模块的时钟周期。

表 17-2: I2C 各速度模式 SCL 电平时间

速度模式	高电平时间 $t_{high}(SCL)$	低电平时间 $t_{low}(SCL)$
标准速度	$I2CSSHCNT \times T_{CLK\_I2C}$	$I2CSSLCNT \times T_{CLK\_I2C}$
快速/快速增强	$I2CFSHCNT \times T_{CLK\_I2C}$	$I2CFSLCNT \times T_{CLK\_I2C}$
高速	$I2CHSHCNT \times T_{CLK\_I2C}$	$I2CHSLCNT \times T_{CLK\_I2C}$

[1] 实际速率还取决于上拉电阻阻值、过滤设置等条件。

17.4.2 滤波配置

I2C 模块支持消抖滤波，以避免 I2C 总线的尖峰脉冲干扰对通信的影响。各速度模式下的滤波时间由表 17-3 确定。表中的 T_{CLK_I2C} 为 I2C 模块的时钟周期。

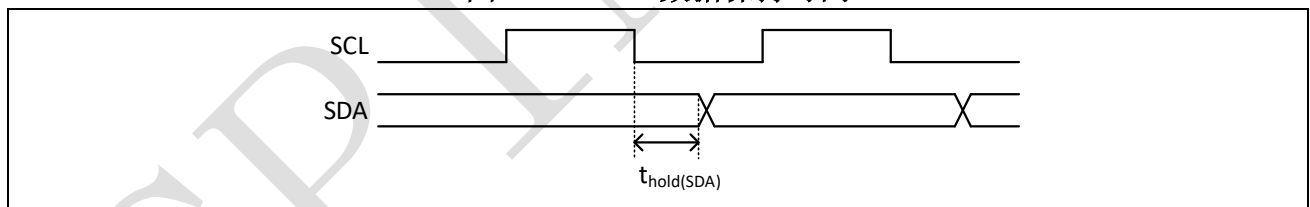
表 17-3: I2C 各速度模式滤波时间

速度模式	能滤除的最大尖峰脉冲宽度 t_{spike}
标准速度	$I2CFSSPKLEN \times T_{CLK\_I2C}$
快速/快速增强	$I2CFSSPKLEN \times T_{CLK\_I2C}$
高速	$I2CHSSPKLEN \times T_{CLK\_I2C}$

17.4.3 数据保持时间配置

数据保持时间如图 17-6 所示， $t_{hold}(SDA) = I2CSDAHOLD \times T_{CLK\_I2C}$ ，主机从机均要配置。

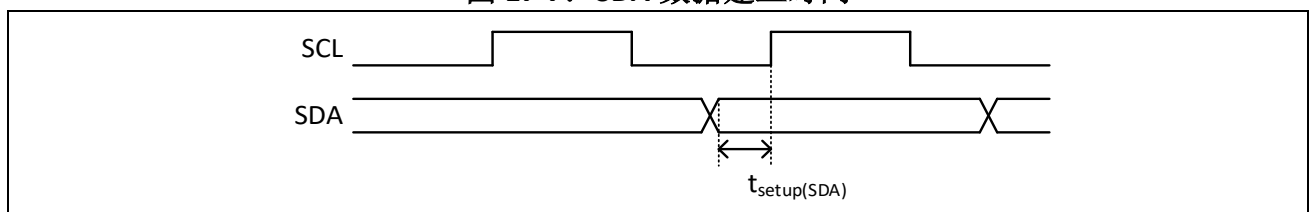
图 17-6: SDA 数据保持时间



17.4.4 数据建立时间配置

如图 17-7 所示，SDA 的建立时间要求可以配置为： $t_{setup}(SDA) = (I2CSDASETUP-1) \times T_{CLK\_I2C}$ 。仅适用于从机发送器控制读取请求响应时，在 SDA 数据变化到 SCL 上升沿采样之间，拉低 SCL。

图 17-7: SDA 数据建立时间



17.5 I2C 主机模式

主机模式下使用流程如下：

- 配置 I2CENABLE.EN=0 来关闭 I2C 模块；
- 配置 I2CCTL.MASTER=1、I2CCTL.SLVDIS=0 来使能主机模式；
- 配置 I2CCTL.RESTARTEN 来选择是否生成重复的开始信号；
- 配置 I2CCTL.SPEED 选择速度模式；
- 通过 I2CSSLCNT、I2CSSHCNT、I2CFSLCNT、I2CFSHCNT、I2CHSLCNT、I2CHSHCNT 配置不同速率下的时钟电平时间；
- 通过 I2CFSSPKLEN、I2CHSSPKLEN 配置尖峰滤波；
- 通过 I2CSDAHOLD 配置数据保持时间；
- 通过 I2CENABLE.EN=1 使能 I2C 模块；
- 通过 I2CMasterADDR、I2CHSMADDR 配置主机寻址信息；
- 通过 I2CDATACMD 写入读写命令和收发数据。

17.5.1 主机发送数据

发送 FIFO 的位宽为 9，最高位表示读写命令（0 为写，1 为读），低 8 位表示数据。往 I2CDATACMD 寄存器写 0x0xx 可向发送 FIFO 中写入数据和发送命令。

当发送 FIFO 非空就会发送开始信号并寻址、发送数据；当从机无应答、1 个字节发送结束且发送 FIFO 为空就会发送停止信号以结束传输。

在传输过程中从机无应答则会停止传输，并自动清空发送 FIFO。

注意： 主机模式下使能了广播请求（General Call）时，不应该往发送 FIFO 写读指令。

17.5.2 主机接收数据

往 I2CDATACMD 写 0x100 可向发送 FIFO 中写入读命令。

当发送 FIFO 非空就会发送开始信号并寻址、接收从机数据；当从机无应答、1 个字节接收完成并发送 FIFO 为空就会发送非应答和停止信号以结束传输。

总线上在主机发送完 Read 电平后，收到非应答会停止传输，并自动清空发送 FIFO。

主机接收到的数据存放在接收 FIFO 中，通过 I2CDATACMD 可以读取这些数据。

17.5.3 主机组合传输

如果连续写入发送 FIFO 中的命令发生改变，那么数据传输方向也会发生相应的改变。

- 若 I2CCTL.RESTARTEN=0，主机会发送停止信号和开始信号来改变传输方向；
- 若 I2CCTL.RESTARTEN=1，主机会发送重复开始信号来改变传输方向。

主机接收到的数据存放在接收 FIFO 中，通过 I2CDATACMD 可以读取这些数据。

17.6 I2C 从机模式

从机模式下使用流程如下：

- 配置 I2CENABLE.EN=0 来关闭 I2C 模块；
- 配置 I2CCTL.MASTER=0、I2CCTL.SLVDIS=1 来使能主机模式；
- 配置 I2CCTL.SPEED 选择速度模式；
- 通过 I2CSSLCNT、I2CSSHCNT、I2CFSLCNT、I2CFSHCNT、I2CHSLCNT、I2CHSHCNT 配置不同速率下的时钟电平时间；
- 通过 I2CFSSPKLEN、I2CHSSPKLEN 配置尖峰滤波；
- 通过 I2CSDASETUP、I2CSDAHOLD 配置数据建立、保持时间；
- 通过 I2CSLVADDR 配置从机地址寄存器；
- 通过 I2CCTL.EN=1 使能 I2C 模块。

17.6.1 地址匹配

当主机寻址地址与 I2CSLVADDR 地址相同时表示地址匹配，从机回 ACK 并会处理主机的读写请求。

若 I2CACKGC.ACKGC=1，从机会应答 0000_000 广播保留地址，同时接收主机广播数据。

17.6.2 从机发送数据

当主机有读请求时触发 RDREQ 中断事件，从机向发送 FIFO 写入数据和发送命令。主机读请求结束时触发 RXDONE 中断事件。

为保证数据完整性，从机必须等到主机读请求后才能写发送 FIFO，否则触发 TXABRT 中断事件并记录中止源到 I2CTXABRTSRC 寄存器，同时清空发送 FIFO。

同理，从机未把发送 FIFO 所有数据发送完毕时，未检测到主机应答，就触发 TXABRT 中断事件并记录发送中止源到 I2CTXABRTSRC 寄存器，同时清空发送 FIFO。

注意： 从机模式下不应该往发送 FIFO 写读指令。

17.6.3 从机接收数据

从机会将接收数据存放在接收 FIFO 中，通过 I2CDATACMD 可以读取这些数据。从机接收到数据之后发送 ACK 表示数据接收成功，若接收数据失败则发送 NACK。

当收到广播地址时触发 GENCALL 中断事件，广播数据将存入接收 FIFO。

17.7 I2C 发送中止

I2C 模块在发送数据时出现异常会中止传输，并且有以下行为：

- 清空发送 FIFO；
- 触发 TXABRT 中断事件，I2CIEVTF.TXABRT 置 1；
- 记录中止源在 I2CTXABRTSRC 寄存器中。

17.8 寄存器

17.8.1 I2C 寄存器列表

表 17-1: I2C 模块基地址

外设模块	基地址
I2C	0x4000 6000

表 17-2: I2C 寄存器列表

寄存器	偏移地址	描述	复位值
I2CCTL	0x00	I2C 控制寄存器	0x0000007F
I2CMASTERADDR	0x04	I2C 主机地址寄存器	0x00001055
I2CSLVADDR	0x08	I2C 从机地址寄存器	0x00000055
I2CHSMADDR	0x0C	I2C 高速主机模式地址寄存器	0x00000001
I2CDATACMD	0x10	I2C 数据和命令寄存器	0x00000000
I2CSSHCNT	0x14	I2C 标准速度模式时钟高电平计数寄存器	0x000001F4
I2CSSLCNT	0x18	I2C 标准速度模式时钟低电平计数寄存器	0x0000024C
I2CFSHCNT	0x1C	I2C 快速模式时钟高电平计数寄存器	0x0000004B
I2CFSLCNT	0x20	I2C 快速模式时钟低电平计数寄存器	0x000000A3
I2CHSHCNT	0x24	I2C 高速模式时钟高电平计数寄存器	0x00000008
I2CHSLCNT	0x28	I2C 高速模式时钟低电平计数寄存器	0x00000014
I2CIF	0x2C	I2C 中断标志寄存器	0x00000000
I2CIE	0x30	I2C 中断使能寄存器	0x000008FF
I2CIEVTF	0x34	I2C 中断原始标志寄存器	0x00000000
I2CRXTH	0x38	I2C 接收 FIFO 长度阈值寄存器	0x00000000
I2CTXTH	0x3C	I2C 发送 FIFO 长度阈值寄存器	0x00000000
I2CINTCLR	0x40	I2C 中断清除寄存器	0x00000000
I2CRXUDFCLR	0x44	I2C RXUDF 中断清除寄存器	0x00000000
I2CRXOVFCLR	0x48	I2C RXOVF 中断清除寄存器	0x00000000
I2CTXOVFCLR	0x4C	I2C TXOVF 中断清除寄存器	0x00000000
I2CRDREQCLR	0x50	I2C RDREQ 清除寄存器	0x00000000
I2CTXABRTCLR	0x54	I2C TXABRT 中断清除寄存器	0x00000000
I2CRXDONECLR	0x58	I2C RXDONE 中断清除寄存器	0x00000000
I2CACTCLR	0x5C	I2C ACT 中断清除寄存器	0x00000000
I2CSTOPDETCLR	0x60	I2C STOPDET 中断清除寄存器	0x00000000
I2CSTARTDETCLR	0x64	I2C STARTDET 中断清除寄存器	0x00000000
I2CGENCALLCLR	0x68	I2C GENCALL 中断清除寄存器	0x00000000
I2CENABLE	0x6C	I2C 使能寄存器	0x00000000
I2CSTS	0x70	I2C 状态寄存器	0x00000006
I2CTFLVL	0x74	I2C 发送 FIFO 当前长度寄存器	0x00000000
I2CRFLVL	0x78	I2C 接收 FIFO 当前长度寄存器	0x00000000
I2CSDAHOLD	0x7C	I2C SDA 保持时间寄存器	0x00000001
I2CTXABRTSRC	0x80	I2C 发送中止源寄存器	0x00000000
I2CSDASETUP	0x94	I2C SDA 建立时间寄存器	0x00000064
I2CACKGC	0x98	I2C 应答广播寻址寄存器	0x00000001

寄存器	偏移地址	描述	复位值
I2CENSTS	0x9C	I2C 使能状态寄存器	0x00000000
I2CFSSPKLEN	0xA0	I2C 快速模式尖峰抑制寄存器	0x00000006
I2CHSSPKLEN	0xA4	I2C 高速模式尖峰抑制寄存器	0x00000002

17.8.2 I2C 寄存器

17.8.2.1 I2C 控制寄存器 (I2CCTL)

表 17-3: I2C 控制寄存器 (I2CCTL) 位段定义

I2CCTL (I2C Control Register)							
Access: I2C->I2CCTL Offset: 0x0000 Default: 0x0000007F							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	SLVDIS	RESTARTEN	ADDRSTS	SLVADDR10B	SPEED		MASTER

表 17-4: I2C 控制寄存器 (I2CCTL) 位段描述

位段	位段名	属性	复位值	描述
31:7	RESERVED	RO	0x0	保留。
6	SLVDIS	RW	0x1	关闭 I2C 从机模式。 0: 使能 1: 关闭 注意: 本位段值需要与 MASTER 保持一致
5	RESTARTEN	RW	0x1	主机模式下决定 RESTART 信号是否需要发送。 旧标准 I2C 从机不支持处理 RESTART 信号。 0: 关闭 1: 使能 当 RESTART 信号被禁止时, 主机被禁止执行以下操作: - 在一个传输内改变方向 - 发送开始字节 (START BYTE) - 工作在高速模式 - 7 位寻址模式下的混合格式传输 - 10 位寻址模式时的读操作 - 在一次传输中发送多字节时, 用 STOP 信号和随后的 START 信号代替 RESTART 信号, 即, 将操作分解为多个 I2C 传输 如果执行上述操作, 会导致 I2CIEVTF.TXABRT 寄存器位被置“1”。
4	ADDRSTS	RO	0x1	I2CMasterADDR.MASTERADDR10B 寄存器位的只读备份。 0: 7 位寻址模式;

位段	位段名	属性	复位值	描述
				1: 10 位寻址模式;
3	SLVADDR10B	RW	0x1	从机模式下的寻址模式。 0: 7 位寻址, 忽略 10 位寻址的传输; I2CSLVADDR 低 7 位用于比较寻址地址。 1: 10 位寻址, I2CSLVADDR 低 10 位用于比较寻址地址。
2:1	SPEED	RW	0x3	I2C 主机模式下的速度模式。 0: 不可用, 若写 0, 硬件会自动设置为 3; 1: 标准速度模式 (100 kbit/s); 2: 快速模式 (400 kbit/s); 3: 高速模式 (3.4 Mbit/s)。
0	MASTER	RW	0x1	使能 I2C 主机模式。 0: 关闭 I2C 主机模式; 1: 使能 I2C 主机模式。 注意: 本位段值需要与 SLVDIS 保持一致。

17.8.2.2 I2C 主机地址寄存器 (I2CMASTERADDR)

表 17-5: I2C 主机地址寄存器 (I2CMASTERADDR) 位段定义

I2CMASTERADDR (I2C Master Address Register)							
Access: I2C->I2CMASTERADDR				Offset: 0x0004		Default: 0x00001055	
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED			MASTERADDR10B	SPECIAL	GCORSTART	TARADDR	
7	6	5	4	3	2	1	0
TARADDR							

表 17-6: I2C 主机地址寄存器 (I2CMASTERADDR) 位段描述

位段	位段名	属性	复位值	描述
31:13	RESERVED	RO	0x0	保留。
12	MASTERADDR10B	RW	0x1	主机模式寻址模式。 0: 7 位寻址模式; 1: 10 位寻址模式。
11	SPECIAL	RW	0x0	使能特殊命令。 0: 关闭特殊命令, 忽略 GCORSTART 位段; 1: 使能 GCORSTART 指定的特殊命令。
10	GCORSTART	RW	0x0	特殊命令选择。 0: 广播请求, 此时只有写操作可以执行。 尝试发起读操作会触发 TXABRT 置“1”。 I2C 会停留在广播请求模式直到 SPECIAL 位被清除。 1: 开始字节 (Start Byte)。
9:0	TARADDR	RW	0x55	主机传输的目标地址。 对于特殊命令: - 发起广播请求时忽略本位段。 - 向本位段写一次, 产生开始字节。 本位段的值不可以与 I2CSLVADDR 相同。

17.8.2.3 I2C 从机地址寄存器 (I2CSLVADDR)

表 17-7: I2C 从机地址寄存器 (I2CSLVADDR) 位段定义

I2CSLVADDR (I2C Slave Address Register)							
Access: I2C->I2CSLVADDR		Offset: 0x0008		Default: 0x00000055			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED						VAL	
7	6	5	4	3	2	1	0
VAL							

表 17-8: I2C 从机地址寄存器 (I2CSLVADDR) 位段描述

位段	位段名	属性	复位值	描述
31:10	RESERVED	RO	0x0	保留。
9:0	VAL	RW	0x55	I2C 从机模式时的地址。 注意: - 本寄存器只有在 I2CENABLE 为 0 时才可以写入 - 不可以为保留的地址: 0x00 ~ 0x07、0x78 ~ 0x7f

17.8.2.4 I2C 高速主机模式地址寄存器 (I2CHSMADDR)

表 17-9: I2C 高速主机模式地址寄存器 (I2CHSMADDR) 位段定义

I2CHSMADDR (I2C High Speed Master Mode Code Address Register)							
Access: I2C->I2CHSMADDR Offset: 0x000C Default: 0x00000001							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED					VAL		

表 17-10: I2C 高速主机模式地址寄存器 (I2CHSMADDR) 位段描述

位段	位段名	属性	复位值	描述
31:3	RESERVED	RO	0x0	保留。
2:0	VAL	RW	0x1	高速模式下，8 位主机代码（格式为 0b00001xxx）的低 3 位。 注意：本寄存器只有在 I2CENABLE 为 0 时才可以写入。

17.8.2.5 I2C 数据和命令寄存器 (I2CDATACMD)

表 17-11: I2C 数据和命令寄存器 (I2CDATACMD) 位段定义

I2CDATACMD (I2C Data Buffer and Command Register)									
Access: I2C->I2CDATACMD			Offset: 0x0010			Default: 0x00000000			
31	30	29	28	27	26	25	24		
RESERVED									
23	22	21	20	19	18	17	16		
RESERVED									
15	14	13	12	11	10	9	8		
RESERVED							CMD		
7	6	5	4	3	2	1	0		
DATA									

表 17-12: I2C 数据和命令寄存器 (I2CDATACMD) 位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED	RO	0x0	保留。
8	CMD	RW	0x0	主机模式。 0: 写请求; 1: 读请求。 从机模式。 0: 接收模式下总是读回 0; 发送模式下需要将本位段设置为 0 1: 接收模式下写 1 无效; 发送模式下写 1 会触发 TXABRT 中断。
7:0	DATA	RW	0x0	I2C 总线上发送或者接受的数据 - 向本寄存器写数据, 会使数据被发送。 - 读本寄存器, 会获取接收到的数据。

17.8.2.6 I2C 标准速度模式时钟高电平计数寄存器 (I2CSSHCNT)

表 17-13: I2C 标准速度模式时钟高电平计数寄存器 (I2CSSHCNT) 位段定义

I2CSSHCNT (Standard Speed I2C Clock SCL High Count Register)							
Access: I2C->I2CSSHCNT		Offset: 0x0014		Default: 0x000001F4			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 17-14: I2C 标准速度模式时钟高电平计数寄存器 (I2CSSHCNT) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED	RO	0x0	保留。
15:0	VAL	RW	0x1F4	标准速度模式下, SCL 高电平持续时间。 最小的有效值为 6, 如果写入的值小于 6, 硬件会自动重新设为 6。 注意: 本寄存器只有在 I2CENABLE 为 0 时才可以写入。

17.8.2.7 I2C 标准速度模式时钟低电平计数寄存器 (I2CSSLCNT)

表 17-15: I2C 标准速度模式时钟低电平计数寄存器 (I2CSSLCNT) 位段定义

I2CSSLCNT (Standard Speed I2C Clock SCL Low Count Register)							
Access: I2C->I2CSSLCNT		Offset: 0x0018		Default: 0x0000024C			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 17-16: I2C 标准速度模式时钟低电平计数寄存器 (I2CSSLCNT) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED	RO	0x0	保留。
15:0	VAL	RW	0x24C	标准速度模式下，SCL 低电平持续时间。 最小的有效值为 8，如果写入的值小于 8，硬件会自动重新设为 8。 注意：本寄存器只有在 I2CENABLE 为 0 时才可以写入。

17.8.2.8 I2C 快速模式时钟高电平计数寄存器 (I2CFSHCNT)

表 17-17: I2C 快速模式时钟高电平计数寄存器 (I2CFSHCNT) 位段定义

I2CFSHCNT (Fast Speed I2C Clock SCL High Count Register)							
Access: I2C->I2CFSHCNT		Offset: 0x001C		Default: 0x0000004B			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 17-18: I2C 快速模式时钟高电平计数寄存器 (I2CFSHCNT) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED	RO	0x0	保留
15:0	VAL	RW	0x4B	快速模式下, SCL 高电平持续时间。 最小的有效值为 6, 如果写入的值小于 6, 硬件会自动重新设为 6。 注意: 本寄存器只有在 I2CENABLE 为 0 时才可以写入。

17.8.2.9 I2C 快速模式时钟低电平计数寄存器 (I2CFSLCNT)

表 17-19: I2C 快速模式时钟低电平计数寄存器 (I2CFSLCNT) 位段定义

I2CFSLCNT (Fast Speed I2C Clock SCL Low Count Register)							
Access: I2C->I2CFSLCNT		Offset: 0x0020		Default: 0x000000A3			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 17-20: I2C 快速模式时钟低电平计数寄存器 (I2CFSLCNT) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED	RO	0x0	保留。
15:0	VAL	RW	0xA3	快速模式下，SCL 低电平持续时间。 最小的有效值为 8，如果写入的值小于 8，硬件会自动重新设为 8。 注意：本寄存器只有在 I2CENABLE 为 0 时才可以写入。

17.8.2.10 I2C 高速模式时钟高电平计数寄存器（I2CHSHCNT）

表 17-21：I2C 高速模式时钟高电平计数寄存器（I2CHSHCNT）位段定义

I2CHSHCNT (High Speed I2C Clock SCL High Count Register)							
Access: I2C->I2CHSHCNT		Offset: 0x0024		Default: 0x00000008			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 17-22：I2C 高速模式时钟高电平计数寄存器（I2CHSHCNT）位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED	RO	0x0	保留。
15:0	VAL	RW	0x8	高速模式下，SCL 高电平持续时间。 最小的有效值为 6，如果写入的值小于 6，硬件会自动重新设为 6。 注意：本寄存器只有在 I2CENABLE 为 0 时才可以写入。

17.8.2.11 I2C 高速模式时钟低电平计数寄存器 (I2CHSLCNT)

表 17-23: I2C 高速模式时钟低电平计数寄存器 (I2CHSLCNT) 位段定义

I2CHSLCNT (High Speed I2C Clock SCL Low Count Register)							
Access: I2C->I2CHSLCNT Offset: 0x0028 Default: 0x00000014							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 17-24: I2C 高速模式时钟低电平计数寄存器 (I2CHSLCNT) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED	RO	0x0	保留。
15:0	VAL	RW	0x14	高速模式下，SCL 低电平持续时间。 最小的有效值为 8，如果写入的值小于 8，硬件会自动重新设为 8。 注意：本寄存器只有在 I2CENABLE 为 0 时才可以写入。

17.8.2.12 I2C 中断标志寄存器 (I2CIF)

表 17-25: I2C 中断标志寄存器 (I2CIF) 位段定义

I2CIF (I2C Interrupt Flag Register)							
Access: I2C->I2CIF		Offset: 0x002C		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED				GENCALL	STARTDET	STOPDET	ACT
7	6	5	4	3	2	1	0
RXDONE	TXABRT	RDREQ	TXDREQ	TXOVF	RXDAV	RXOVF	RXUDF

表 17-26: I2C 中断标志寄存器 (I2CIF) 位段描述

位段	位段名	属性	复位值	描述
31:12	RESERVED	RO	0x0	保留。
11	GENCALL	RO	0x0	广播寻址接收并应答标志。 0: 未接收到广播寻址; 1: 接收到广播寻址。
10	STARTDET	RO	0x0	START 或者 RESTART 信号标志。 0: 未检测到 START、RESTART 信号; 1: 检测到 START、RESTART 信号。
9	STOPDET	RO	0x0	STOP 信号标志。 0: 未检测到 STOP 信号; 1: 检测到 STOP 信号。
8	ACT	RO	0x0	I2C 总线活动状态标志。 0: 未检测到 I2C 总线活动; 1: 检测到 I2C 总线活动。
7	RXDONE	RO	0x0	作为从机发送器时, 外部主机接收数据结束标志。 0: 发送数据被主机应答, 主机继续收数据; 1: 发送数据违背应答, 主机结束数据接收。
6	TXABRT	RO	0x0	发送中止标志。 0: 未发生发送中止; 1: 发送已中止, 发送 FIFO 复位, 读取 I2CTXABRTCLR 清除该标志。
5	RDREQ	RO	0x0	作为从机时, 检测到外部主机读请求。 0: 未接收到主机的读请求;

位段	位段名	属性	复位值	描述
				1: 已接收到主机的读请求。 处理器必须响应该中断，并将请求的数据写入到 I2CDATACMD 寄存器。
4	TXDREQ	RO	0x0	发送请求标志。 0: 发送 FIFO 中数据量大于阈值； 1: 发送 FIFO 中数据量小于或等于阈值。
3	TXOVF	RO	0x0	发送 FIFO 上溢标志。 0: 发送 FIFO 未上溢； 1: 发送 FIFO 上溢。
2	RXDAV	RO	0x0	接收数据有效标志。 0: 接收 FIFO 中数据量低于阈值； 1: 接收 FIFO 中数据量等于或者高于阈值。
1	RXOVF	RO	0x0	接收 FIFO 上溢标志。接收 FIFO 满之后，任何接收到的数据都会丢失。 0: 接收 FIFO 未上溢； 1: 接收 FIFO 上溢。
0	RXUDF	RO	0x0	接收 FIFO 下溢标志。 0: 接收 FIFO 未下溢； 1: 接收 FIFO 下溢。

17.8.2.13 I2C 中断使能寄存器 (I2CIE)

表 17-27: I2C 中断使能寄存器 (I2CIE) 位段定义

I2CIE (I2C Interrupt Enable Register)							
Access: I2C->I2CIE		Offset: 0x0030		Default: 0x000008FF			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED				GENCALL	STARTDET	STOPDET	ACT
7	6	5	4	3	2	1	0
RXDONE	TXABRT	RDREQ	TXDREQ	TXOVF	RXDAV	RXOVF	RXUDF

表 17-28: I2C 中断使能寄存器 (I2CIE) 位段描述

位段	位段名	属性	复位值	描述
31:12	RESERVED	RO	0x0	保留。
11	GENCALL	RW	0x1	使能广播寻址接收并应答 (GENCALL) 中断。 0: 关闭 1: 使能
10	STARTDET	RW	0x0	使能 START 或者 RESTART 信号 (STARTDET) 中断。 0: 关闭 1: 使能
9	STOPDET	RW	0x0	使能 STOP 信号 (STOPDET) 中断。 0: 关闭 1: 使能
8	ACT	RW	0x0	使能 I2C 总线活动 (ACT) 中断。 0: 关闭 1: 使能
7	RXDONE	RW	0x1	使能接收完成 (RXDONE) 中断。 0: 关闭 1: 使能
6	TXABRT	RW	0x1	使能发送中止 (TXABRT) 中断。 0: 关闭 1: 使能
5	RDREQ	RW	0x1	使能读请求 (RDREQ) 中断。 0: 关闭 1: 使能

位段	位段名	属性	复位值	描述
4	TXDREQ	RW	0x1	使能发送请求（TXDREQ）中断。 0: 关闭 1: 使能
3	TXOVF	RW	0x1	使能发送 FIFO 上溢（TXOVF）中断。 0: 关闭 1: 使能
2	RXDAV	RW	0x1	使能接收数据有效（RXDAV）中断。 0: 关闭 1: 使能
1	RXOVF	RW	0x1	使能接收 FIFO 上溢（RXOVF）中断。 0: 关闭 1: 使能
0	RXUDF	RW	0x1	使能接收 FIFO 下溢（RXUDF）中断。 0: 关闭 1: 使能

17.8.2.14 I2C 中断原始标志寄存器 (I2CIEVTF)

表 17-29: I2C 中断原始标志寄存器 (I2CIEVTF) 位段定义

I2CIEVTF (I2C Interrupt Event Flag Register)							
Access: I2C->I2CIEVTF Offset: 0x0034 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED				GENCALL	STARTDET	STOPDET	ACT
7	6	5	4	3	2	1	0
RXDONE	TXABRT	RDREQ	TXDREQ	TXOVF	RXDAV	RXOVF	RXUDF

表 17-30: I2C 中断原始标志寄存器 (I2CIEVTF) 位段描述

位段	位段名	属性	复位值	描述
31:12	RESERVED	RO	0x0	保留。
11	GENCALL	RO	0x0	广播寻址接收并应答标志。 0: 未接收到广播寻址; 1: 接收到广播寻址。
10	STARTDET	RO	0x0	START 或者 RESTART 信号标志。 0: 未检测到 START、RESTART 信号; 1: 检测到 START、RESTART 信号。
9	STOPDET	RO	0x0	STOP 信号标志。 0: 未检测到 STOP 信号; 1: 检测到 STOP 信号。
8	ACT	RO	0x0	I2C 总线活动状态标志。 0: 未检测到 I2C 总线活动; 1: 检测到 I2C 总线活动。
7	RXDONE	RO	0x0	作为从机发送器时, 外部主机接收数据结束标志。 0: 发送数据被主机应答, 主机继续收数据; 1: 发送数据违背应答, 主机结束数据接收。
6	TXABRT	RO	0x0	发送中止标志。 0: 未发生发送中止; 1: 发送已中止, 发送 FIFO 复位, 读取 I2CTXABRTCLR 清除该标志。
5	RDREQ	RO	0x0	作为从机时, 检测到外部主机读请求。 0: 未接收到主机的读请求;

位段	位段名	属性	复位值	描述
				1: 已接收到主机的读请求。 处理器必须响应该中断，并将请求的数据写入到 I2CDATACMD 寄存器。
4	TXDREQ	RO	0x0	发送请求标志。 0: 发送 FIFO 中数据量大于阈值； 1: 发送 FIFO 中数据量小于或等于阈值。
3	TXOVF	RO	0x0	发送 FIFO 上溢标志。 0: 发送 FIFO 未上溢； 1: 发送 FIFO 上溢。
2	RXDAV	RO	0x0	接收数据有效标志。 0: 接收 FIFO 中数据量低于阈值； 1: 接收 FIFO 中数据量等于或者高于阈值。
1	RXOVF	RO	0x0	接收 FIFO 上溢标志。接收 FIFO 满之后，任何接收到的数据都会丢失。 0: 接收 FIFO 未上溢； 1: 接收 FIFO 上溢。
0	RXUDF	RO	0x0	接收 FIFO 下溢标志。 0: 接收 FIFO 未下溢； 1: 接收 FIFO 下溢。

17.8.2.15 I2C 接收 FIFO 长度阈值寄存器 (I2CRXTH)

表 17-31: I2C 接收 FIFO 长度阈值寄存器 (I2CRXTH) 位段定义

I2CRXTH (I2C Receive FIFO Threshold Register)							
Access: I2C->I2CRXTH Offset: 0x0038 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
KEY				VAL			

表 17-32: I2C 接收 FIFO 长度阈值寄存器 (I2CRXTH) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED	RO	0x0	保留。
7:4	KEY	WO	0x0	写 I2CRXTH.VAL 时，必须向本位段写 0，否则 I2CRXTH.VAL 的值总是 0xF。 本位段总是读回 0。
3:0	VAL	RW	0x0	接收 FIFO 长度阈值。 当 I2CRFLVL.VAL > I2CRXTH.VAL，触发 RXDAV 中断。

17.8.2.16 I2C 发送 FIFO 长度阈值寄存器 (I2CTXTH)

表 17-33: I2C 发送 FIFO 长度阈值寄存器 (I2CTXTH) 位段定义

I2CTXTH (I2C Transmit FIFO Threshold Register)							
Access: I2C->I2CTXTH Offset: 0x003C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
KEY				VAL			

表 17-34: I2C 发送 FIFO 长度阈值寄存器 (I2CTXTH) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED	RO	0x0	保留。
7:4	KEY	WO	0x0	写 I2CTXTH.VAL 时，必须向本位段写 0，否则 I2CTXTH.VAL 的值总是 0xF。 本位段总是读回 0。
3:0	VAL	RW	0x0	发送 FIFO 长度阈值。 当 I2CTFLVL.VAL ≤ I2CTXTH.VAL，触发 TXDREQ 中断。

17.8.2.17 I2C 中断清除寄存器 (I2CINTCLR)

表 17-35: I2C 中断清除寄存器 (I2CINTCLR) 位段定义

I2CINTCLR (Clear Combined and Individual Interrupt Register)						
Access: I2C->I2CINTCLR Offset: 0x0040 Default: 0x00000000						
31	30	29	28	27	26	25
RESERVED						
23	22	21	20	19	18	17
RESERVED						
15	14	13	12	11	10	9
RESERVED						
7	6	5	4	3	2	1
RESERVED						0
						CLR

表 17-36: I2C 中断清除寄存器 (I2CINTCLR) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED	RO	0x0	保留。
0	CLR	RO	0x0	读取该寄存器会清除所有中断标志位以及 I2CTXABRTSRC 寄存器。 注意: - RDREQ、TXDREQ 只能通过读取接收 FIFO、写入发送 FIFO 来清除。 - 正确配置发送 Start Byte 功能后, 才能清除 I2CTXABRTSRC.STARTNORESTART (即 I2CCTL.RESTARTEN = 1, I2CMasterADDR.GCORSTART = 1)。

17.8.2.18 I2C RXUDF 中断清除寄存器 (I2CRXUDFCLR)

表 17-37: I2C RXUDF 中断清除寄存器 (I2CRXUDFCLR) 位段定义

I2CRXUDFCLR (Clear RXUDF Interrupt Register)							
Access: I2C->I2CRXUDFCLR				Offset: 0x0044		Default: 0x00000000	
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							CLR

表 17-38: I2C RXUDF 中断清除寄存器 (I2CRXUDFCLR) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED	RO	0x0	保留。
0	CLR	RO	0x0	读取该寄存器清除 I2CIEVTF、I2CIF 中的 RXUDF 标志。

17.8.2.19 I2C RXOVF 中断清除寄存器 (I2CRXOVFCLR)

表 17-39: I2C RXOVF 中断清除寄存器 (I2CRXOVFCLR) 位段定义

I2CRXOVFCLR (Clear RXOVF Interrupt Register)							
Access: I2C->I2CRXOVFCLR				Offset: 0x0048		Default: 0x00000000	
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							CLR

表 17-40: I2C RXOVF 中断清除寄存器 (I2CRXOVFCLR) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED	RO	0x0	保留。
0	CLR	RO	0x0	读取该寄存器清除 I2CIEVTF、I2CIF 中的 RXOVF 标志。

17.8.2.20 I2C TXOVF 中断清除寄存器 (I2CTXOVFCLR)

表 17-41: I2C TXOVF 中断清除寄存器 (I2CTXOVFCLR) 位段定义

I2CTXOVFCLR (Clear TXOVF Interrupt Register)							
Access: I2C->I2CTXOVFCLR				Offset: 0x004C		Default: 0x00000000	
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							CLR

表 17-42: I2C TXOVF 中断清除寄存器 (I2CTXOVFCLR) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED	RO	0x0	保留。
0	CLR	RO	0x0	读取该寄存器清除 I2CIEVTF、I2CIF 中的 TXOVF 标志。

17.8.2.21 I2C RDREQ 中断清除寄存器 (I2CRDREQCLR)

表 17-43: I2C RDREQ 中断清除寄存器 (I2CRDREQCLR) 位段定义

I2CRDREQCLR (Clear RDREQ Interrupt Register)							
Access: I2C->I2CRDREQCLR				Offset: 0x0050		Default: 0x00000000	
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							CLR

表 17-44: I2C RDREQ 中断清除寄存器 (I2CRDREQCLR) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED	RO	0x0	保留。
0	CLR	RO	0x0	读取该寄存器清除 I2CIEVTF、I2CIF 中的 RDREQ 标志。

17.8.2.22 I2C TXABRT 中断清除寄存器 (I2CTXABRTCLR)

表 17-45: I2C TXABRT 中断清除寄存器 (I2CTXABRTCLR) 位段定义

I2CTXABRTCLR (Clear TXABRT Interrupt Register)							
Access: I2C->I2CTXABRTCLR Offset: 0x0054 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							CLR

表 17-46: I2C TXABRT 中断清除寄存器 (I2CTXABRTCLR) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED	RO	0x0	保留。
0	CLR	RO	0x0	读取该寄存器清除 I2CIEVTF、I2CIF 中的 TXABRT 标志，以及 I2CTXABRTSRC 寄存器；也会将发送 FIFO 从复位状态中释放，允许 CPU 继续写发送 FIFO。 注意：正确配置发送 Start Byte 功能后，才能清除 I2CTXABRTSRC.STARTNORESTART（即 I2CCTL.RESTARTEN = 1，I2CMasterADDR.GCORSTART = 1）。

17.8.2.23 I2C RXDONE 中断清除寄存器 (I2CRXDONECLR)

表 17-47: I2C RXDONE 中断清除寄存器 (I2CRXDONECLR) 位段定义

I2CRXDONECLR (Clear RXDONE Interrupt Register)							
Access: I2C->I2CRXDONECLR Offset: 0x0058 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							CLR

表 17-48: I2C RXDONE 中断清除寄存器 (I2CRXDONECLR) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED	RO	0x0	保留。
0	CLR	RO	0x0	读取该寄存器清除 I2CIEVTF、I2CIF 中的 RXDONE 标志。

17.8.2.24 I2C ACT 中断清除寄存器 (I2CACTCLR)

表 17-49: I2C ACT 中断清除寄存器 (I2CACTCLR) 位段定义

I2CACTCLR (Clear ACT Interrupt Register)							
Access: I2C->I2CACTCLR Offset: 0x005C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							CLR

表 17-50: I2C ACT 中断清除寄存器 (I2CACTCLR) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED	RO	0x0	保留
0	CLR	RO	0x0	如果 I2C 不再活动时，读取该寄存器清除 I2CIEVTF、I2CIF 中的 ACT 标志。 如果 I2C 总线仍然保持活动，ACT 标志会被继续置位。 如果 I2C 模块被关闭且总线上不再有活动，ACT 标志会被硬件自动清除。

17.8.2.25 I2C STOPDET 中断清除寄存器 (I2CSTOPDETCCLR)

表 17-51: I2C STOPDET 中断清除寄存器 (I2CSTOPDETCCLR) 位段定义

I2CSTOPDETCCLR (Clear STOPDET Interrupt Register)							
Access: I2C->I2CSTOPDETCCLR				Offset: 0x0060		Default: 0x00000000	
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							CLR

表 17-52: I2C STOPDET 中断清除寄存器 (I2CSTOPDETCCLR) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED	RO	0x0	保留。
0	CLR	RO	0x0	读取该寄存器清除 I2CIEVTF、I2CIF 中的 STOPDET 标志。

17.8.2.26 I2C STARTDET 中断清除寄存器 (I2CSTARTDETCCLR)

表 17-53: I2C STARTDET 中断清除寄存器 (I2CSTARTDETCCLR) 位段定义

I2CSTARTDETCCLR (Clear STARTDET Interrupt Register)							
Access: I2C->I2CSTARTDETCCLR				Offset: 0x0064		Default: 0x00000000	
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							CLR

表 17-54: I2C STARTDET 中断清除寄存器 (I2CSTARTDETCCLR) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED	RO	0x0	保留。
0	CLR	RO	0x0	读取该寄存器清除 I2CIEVTF、I2CIF 中的 STARTDET 标志。

17.8.2.27 I2C GENCALL 中断清除寄存器 (I2CGENCALLCLR)

表 17-55: I2C GENCALL 中断清除寄存器 (I2CGENCALLCLR) 位段定义

I2CGENCALLCLR (Clear GENCALL Interrupt Register)							
Access: I2C->I2CGENCALLCLR Offset: 0x0068 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							CLR

表 17-56: I2C GENCALL 中断清除寄存器 (I2CGENCALLCLR) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED	RO	0x0	保留。
0	CLR	RO	0x0	读取该寄存器清除 I2CIEVTF、I2CIF 中的 GENCALL 标志。

17.8.2.28 I2C 使能寄存器 (I2CENABLE)

表 17-57: I2C 使能寄存器 (I2CENABLE) 位段定义

I2CENABLE (I2C Enable Register)							
Access: I2C->I2CENABLE Offset: 0x006C Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							EN

表 17-58: I2C 使能寄存器 (I2CENABLE) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED	RO	0x0	保留。
0	EN	RW	0x0	I2C 模块使能 0: 关闭 1: 使能 当 I2C 被关闭时，会发生以下情形： - 发送 FIFO 和接收 FIFO 被复位清空。 - I2CIF 寄存器中的状态位仍然有效直到 I2C 进入 IDLE 状态。 - 如果 I2C 正处于发送状态，那么 I2C 会在当前传输完成后停止并清空发送 FIFO 的内容。 - 如果 I2C 正处于接收状态，那么 I2C 会在当前字节后停止当前传输并发送 NACK 信号。

17.8.2.29 I2C 状态寄存器 (I2CSTS)

表 17-59: I2C 状态寄存器 (I2CSTS) 位段定义

I2CSTS (I2C Status Register)							
Access: I2C->I2CSTS		Offset: 0x0070		Default: 0x00000006			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	SACT	MACT	RFF	RFNE	TFE	TFNF	ACT

表 17-60: I2C 状态寄存器 (I2CSTS) 位段描述

位段	位段名	属性	复位值	描述
31:7	RESERVED	RO	0x0	保留。
6	SACT	RO	0x0	从机状态机活动状态。 0: 从机状态机处于空闲状态; 1: 从机状态机处于忙碌状态。
5	MACT	RO	0x0	主机状态机活动状态。 0: 主机状态机处于空闲状态; 1: 主机状态机未处于忙碌状态。
4	RFF	RO	0x0	接收 FIFO 已经满了。 0: 接收 FIFO 未滿; 1: 接收 FIFO 已经满了。
3	RFNE	RO	0x0	接收 FIFO 非空。 0: 接收 FIFO 为空; 1: 接收 FIFO 非空。
2	TFE	RO	0x1	发送 FIFO 完全空。 0: 发送 FIFO 非空; 1: 发送 FIFO 完全空。
1	TFNF	RO	0x1	发送 FIFO 未滿。 0: 发送 FIFO 已经满了; 1: 发送 FIFO 未滿。
0	ACT	RO	0x0	I2C 活动状态 (SACT 和 MACT 的逻辑或)。 0: 主机和从机状态机都处于空闲状态; 1: 主机或从机状态机处于活动状态。

17.8.2.30 I2C 发送 FIFO 当前长度寄存器 (I2CTFLVL)

表 17-61: I2C 发送 FIFO 当前长度寄存器 (I2CTFLVL) 位段定义

I2CTFLVL (I2C Transmit FIFO Level Register)							
Access: I2C->I2CTFLVL Offset: 0x0074 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED			VAL				

表 17-62: I2C 发送 FIFO 当前长度寄存器 (I2CTFLVL) 位段描述

位段	位段名	属性	复位值	描述
31:5	RESERVED	RO	0x0	保留。
4:0	VAL	RO	0x0	发送 FIFO 中有效数据个数。

17.8.2.31 I2C 接收 FIFO 当前长度寄存器 (I2CRFLVL)

表 17-63: I2C 接收 FIFO 当前长度寄存器 (I2CRFLVL) 位段定义

I2CRFLVL (I2C Receive FIFO Level Register)							
Access: I2C->I2CRFLVL Offset: 0x0078 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED			VAL				

表 17-64: I2C 接收 FIFO 当前长度寄存器 (I2CRFLVL) 位段描述

位段	位段名	属性	复位值	描述
31:5	RESERVED	RO	0x0	保留。
4:0	VAL	RO	0x0	接收 FIFO 中有效数据个数。

17.8.2.32 I2C SDA 保持时间寄存器 (I2CSDAHOLD)

表 17-65: I2C SDA 保持时间寄存器 (I2CSDAHOLD) 位段定义

I2CSDAHOLD (I2C SDA Hold-Time Register)							
Access: I2C->I2CSDAHOLD				Offset: 0x007C		Default: 0x00000001	
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 17-66: I2C SDA 保持时间寄存器 (I2CSDAHOLD) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED	RO	0x0	保留。
15:0	VAL	RW	0x1	配置 SDA 信号保持时间，即 SCL 下降沿后 SDA 保持不变的时间。 - 主机模式下，最小的有效值为 1； - 从机模式下，最小的有效值为 7； - 任何模式下，小于 SCL 低电平周期 - 1。 实际时间为 I2CSDAHOLD x T_{CLK_I2C}。 注意：本寄存器只有在 I2CENABLE 为 0 时才可以写入。

17.8.2.33 I2C 发送中止源寄存器 (I2CTXABRTSRC)

表 17-67: I2C 发送中止源寄存器 (I2CTXABRTSRC) 位段定义

I2CTXABRTSRC (I2C Transmit Abort Source Register)							
Access: I2C->I2CTXABRTSRC Offset: 0x0080 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
SLVRDINTX	SARBLOST	SLVFLUSHTF	MARBLOST	MASTERDIS	RD10BNORESTART	STARTNORESTART	HSNORESTART
7	6	5	4	3	2	1	0
STARTACKDET	HSACKDET	GCREAD	GCNACK	TXDATANACK	ADDR10B2NACK	ADDR10B1NACK	ADDR7BNACK

表 17-68: I2C 发送中止源寄存器 (I2CTXABRTSRC) 位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED	RO	0x0	保留。
15	SLVRDINTX	RO	0x0	I2C 角色: 从机发送器。 从机响应外部主机读请求, 却错误的发送了读命令 (I2CDATAACMD.CMD = 1), 本位段置 1。
14	SARBLOST	RO	0x0	I2C 角色: 从机发送器。 从机向外部主机发送数据时丢失仲裁, 本位段置 1; I2CTXABRTSRC.MARBLOST 也会置 1。
13	SLVFLUSHTF	RO	0x0	I2C 角色: 从机发送器。 从机接收到一个读命令时, 发送 FIFO 中存在一些数据, 则本位段会置 1, 并产生一个 TXABRT 中断去清空发送 FIFO。
12	MARBLOST	RO	0x0	I2C 角色: 主机发送器/从机发送器。 本位段在下列情况下会被置“1”: - 主机失去仲裁; - 当 I2CTXABRTSRC.SARBLOST 被置 1, 从机发送器失去仲裁。
11	MASTERDIS	RO	0x0	I2C 角色: 主机发送器/主机接收器。 用户试图去启动一次主机操作而主机模式被关闭时, 本位段会被置 1。
10	RD10BNORESTART	RO	0x0	I2C 角色: 主机接收器。 发送重复的开始信号功能被关闭 (I2CCTL.RESTARTEN = 0) 时, 而主机发送一个 10 位寻址的读命令时, 本位段会被置 1。
9	STARTNORESTART	RO	0x0	I2C 角色: 主机发送器/主机接收器。

位段	位段名	属性	复位值	描述
				发送重复的开始信号功能被关闭（I2CCTL.RESTARTEN = 0）时，用户试图去发送一个 START Byte 时，本位段会被置 1。 按照下面的配置可以避免这个错误： - I2CCTL.RESTARTEN=1； - I2CMasterADDR.SPECIAL=0 或者 I2CMasterADDR.GCORSTART=0； 否则的话，通过读 I2CTXABRTCLR 清除本位段后，下一个时钟周期本位段还会被置 1。
8	HSNORESTART	RO	0x0	I2C 角色：主机发送器/主机接收器。 发送重复的开始信号功能被关闭（I2CCTL.RESTARTEN = 0）时，用户试图在高速模式下传输数据，本位段会被置 1。
7	STARTACKDET	RO	0x0	I2C 角色：主机发送器/主机接收器。 主机发送一个 START Byte，START Byte 字节被错误的应答时，本位段被置 1。
6	HSACKDET	RO	0x0	I2C 角色：主机发送器/主机接收器。 主机工作在高速模式下，高速主机码被错误的应答时，本位段被置 1。
5	GCREAD	RO	0x0	I2C 角色：主机发送器。 主机发送一个广播寻址，却错误的发送了读命令（I2CDATAACMD.CMD = 1）时，本位段被置 1。
4	GCNACK	RO	0x0	I2C 角色：主机发送器。 主机发送一个广播寻址，总线上没有从机应答时，本位段被置 1。
3	TXDATANACK	RO	0x0	I2C 角色：主机发送器。 主机接收到地址应答，在之后发送的数据字节没有收到外部从机的应答，本位段被置 1。
2	ADDR10B2NACK	RO	0x0	I2C 角色：主机发送器/主机接收器。 主机工作在 10 位寻址模式，地址的第二个字节没有被外部从机应答，本位段被置 1。
1	ADDR10B1NACK	RO	0x0	I2C 角色：主机发送器/主机接收器。 主机工作在 10 位寻址模式，地址的第一个字节没有被外部从机应答，本位段被置 1。
0	ADDR7BNACK	RO	0x0	I2C 角色：主机发送器/主机接收器。 主机工作在 7 位寻址模式，地址字节没有被外部从机应答，本位段被置 1。

17.8.2.34 I2C SDA 建立时间寄存器 (I2CSDASETUP)

表 17-69: I2C SDA 建立时间寄存器 (I2CSDASETUP) 位段定义

I2CSDASETUP (I2C SDA Setup Register)							
Access: I2C->I2CSDASETUP				Offset: 0x0094		Default: 0x00000064	
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
VAL							

表 17-70: I2C SDA 建立时间寄存器 (I2CSDASETUP) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED	RO	0x0	保留。
7:0	VAL	RW	0x64	配置 SDA 信号建立时间，即作为从机发送器控制读取请求响应时，SDA 数据变化到 SCL 上升沿采样的时间（在此期间，SCL 被从机拉低）。 最小有效值为 2。 实际时间为$(I2CSDASETUP-1) \times T_{CLK\_I2C}$。 例如，如果需要的建立时间为 1000ns，而 CLK_I2C 的频率为 10MHz，本寄存器的值建议设置为 11。

17.8.2.35 I2C 应答广播寻址 (I2CACKGC)

表 17-71: I2C 应答广播寻址 (I2CACKGC) 位段定义

I2CACKGC (I2C ACK General Call Register)									
Access: I2C->I2CACKGC Offset: 0x0098 Default: 0x00000001									
31	30	29	28	27	26	25	RESERVED		
23	22	21	20	19	18	17	RESERVED		
15	14	13	12	11	10	9	RESERVED		
7	6	5	4	3	2	1	RESERVED		
							0		
							ACKGC		

表 17-72: I2C 应答广播寻址 (I2CACKGC) 位段描述

位段	位段名	属性	复位值	描述
31:1	RESERVED	RO	0x0	保留。
0	ACKGC	RW	0x1	对广播寻址 (General Call) 的响应。 0: 用 NACK 响应广播寻址; 1: 用 ACK 响应广播寻址。

17.8.2.36 I2C 使能状态寄存器 (I2CENSTS)

表 17-73: I2C 使能状态寄存器 (I2CENSTS) 位段定义

I2CENSTS (I2C Enable Status Register)															
Access: I2C->I2CENSTS					Offset: 0x009C					Default: 0x00000000					
31	30	29	28	27	26					25					24
RESERVED															
23	22	21	20	19	18					17					16
RESERVED															
15	14	13	12	11	10					9					8
RESERVED															
7	6	5	4	3	2					1					0
RESERVED					SLVRDLOST					SLVDISONBUSY					EN

表 17-74: I2C 使能状态寄存器 (I2CENSTS) 位段描述

位段	位段名	属性	复位值	描述
31:3	RESERVED	RO	0x0	保留。
2	SLVRDLOST	RO	0x0	由于 I2C 关闭导致的从机已接收的数据丢失。 0: 在 I2CENABLE.EN 从 1 变为 0 之前, 从机接收数据已完成; 1: 在 I2CENABLE.EN 从 1 变为 0 之前, 从机接收数据被中止 (已从 I2C 传输中接收了至少一个字节数据), 则该位段值 1。
1	SLVDISONBUSY	RO	0x0	当从机处于忙状态时被关闭。 0: 从机工作期间, I2CENABLE.EN 保持稳定; 1: 从机接收地址、数据时, I2CENABLE.EN 从 1 变为 0, 则该位段值 1。
0	EN	RO	0x0	I2C 使能状态。 0: I2C 完全处于关闭状态, 此时 SLVRDLOST、SLVDISONBUSY 值有效。 1: I2C 处于使能状态。

17.8.2.37 I2C 快速模式尖峰抑制寄存器 (I2CFSSPKLEN)

表 17-75: I2C 快速模式尖峰抑制寄存器 (I2CFSSPKLEN) 位段定义

I2CFSSPKLEN (Fast Speed I2C Spike Suppresion Limit Register)							
Access: I2C->I2CFSSPKLEN Offset: 0x00A0 Default: 0x00000006							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
VAL							

表 17-76: I2C 快速模式尖峰抑制寄存器 (I2CFSSPKLEN) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED	RO	0x0	保留。
7:0	VAL	RW	0x6	快速模式下，SCL 或者 SDA 信号线上能被尖峰抑制逻辑过滤掉的最长尖峰持续时间。 – 最小值为 1，如果写入 0，会被硬件自动设为 1； – 必须小于等于 I2CSSHCNT-5、I2CFSHCNT-5； – 必须小于等于 I2CSSLCNT-7、I2CFSHCNT-7。 实际时间为 I2CFSSPKLEN x T_{CLK_I2C}。 注意：本寄存器只有在 I2CENABLE 为 0 时才可以写入。

17.8.2.38 I2C 高速模式尖峰抑制寄存器（I2CHSSPKLEN）

表 17-77：I2C 高速模式尖峰抑制寄存器（I2CHSSPKLEN）位段定义

I2CHSSPKLEN (High Speed I2C Spike Suppresion Limit Register)							
Access: I2C->I2CHSSPKLEN Offset: 0x00A4 Default: 0x00000002							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
VAL							

表 17-78：I2C 高速模式尖峰抑制寄存器（I2CHSSPKLEN）位段描述

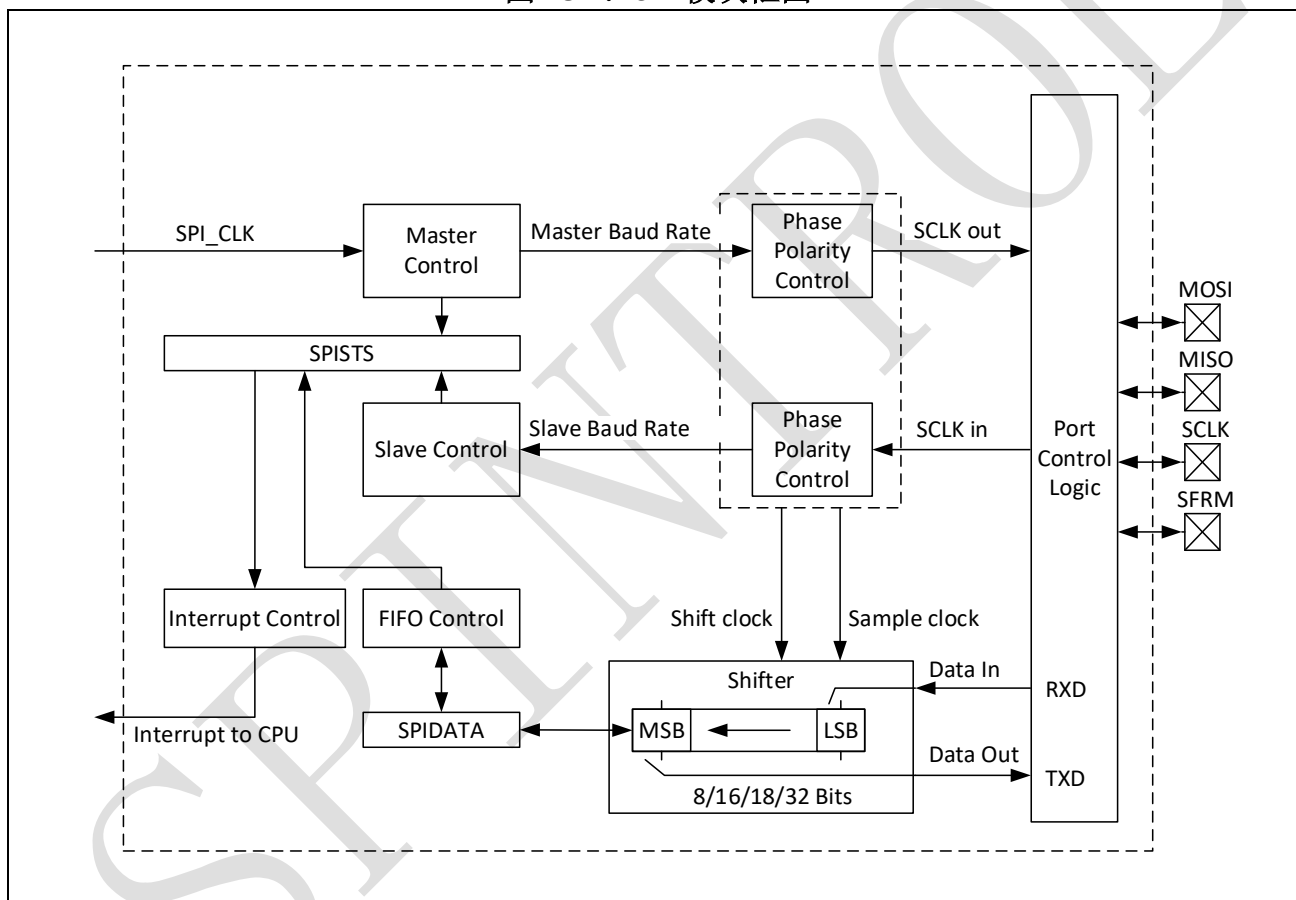
位段	位段名	属性	复位值	描述
31:8	RESERVED	RO	0x0	保留。
7:0	VAL	RW	0x2	高速模式下，SCL 或者 SDA 信号线上能被尖峰抑制逻辑过滤掉的最长尖峰持续时间。 – 最小值为 1，如果写入 0，会被硬件自动设为 1； – 必须小于等于 I2CHSHCNT-5； – 必须小于等于 I2CHSLCNT-7。 实际时间为 I2CHSSPKLEN x T_{CLK_I2C}。 注意：本寄存器只有在 I2CENABLE=0 时才可以写入。

18 串行外围设备接口 SPI

18.1 SPI 概述

SPD1121 的 SPI 接口是一个同步全双工串行数据传输端口。SPI 接口可以用来连接多种外设设备，如 FLASH，ADC，CODEC 等。SPI 接口可以配置工作在主机模式（相连的外设设备作为从机）或者从机模式（相连的外设设备作为主机）。通过可配置的时钟，SPI 端口支持不同的通信速率。串行数据帧的长度可以被配置为 8，16，18 或者 32 位。发送数据和接收数据分别有一个独立的 FIFO。两个 FIFO 都是 16 x 32 位的。两个 FIFO 可以被 CPU 装载或者清空。SPI 模块可以通过查询模式或者中断请求模式来控制。SPI 模块主要由控制逻辑、状态和数据寄存器、移位逻辑、FIFO 控制逻辑、主从控制逻辑以及端口控制逻辑组成，如图 18-1 所示。

图 18-1: SPI 模块框图



18.2 SPI 主要特性

SPD1121 SPI 模块具有以下特性：

- 数据传输速率高达 30 Mbps。
- 支持主从模式。
- 支持全双工。
- 支持只收模式。
- 可编程的数据帧长度：1~32 位。
- 可编程的片选信号（SFRM）极性选择。
- 数据位发送顺序为 MSB 位优先。
- 可编程的时钟极性和相位。
- 两个独立的发送 FIFO（TXFIFO）和接收 FIFO（RXFIFO），每个 FIFO 的大小为 16 x 32 位，都支持压缩模式（Packed mode）。

18.3 SPI 信号描述

表 18-1 描述了 SPD1121 SPI 接口的总线信号。

表 18-1：SPI 信号描述

信号名	类型	描述
MOSI	输入/输出	SPI 模块数据传输信号 主机模式：发送数据 从机模式：接收数据
MISO	输入/输出	SPI 模块数据传输信号 主机模式：接收数据 从机模式：发送数据
SCLK	输入/输出	SPI 模块时钟信号 主机模式：发送数据时，输出时钟信号 从机模式：接收时钟信号
SFRM	输入/输出	SPI 模块片选信号 主机模式：发送数据时，输出片选信号 从机模式：接收片选信号

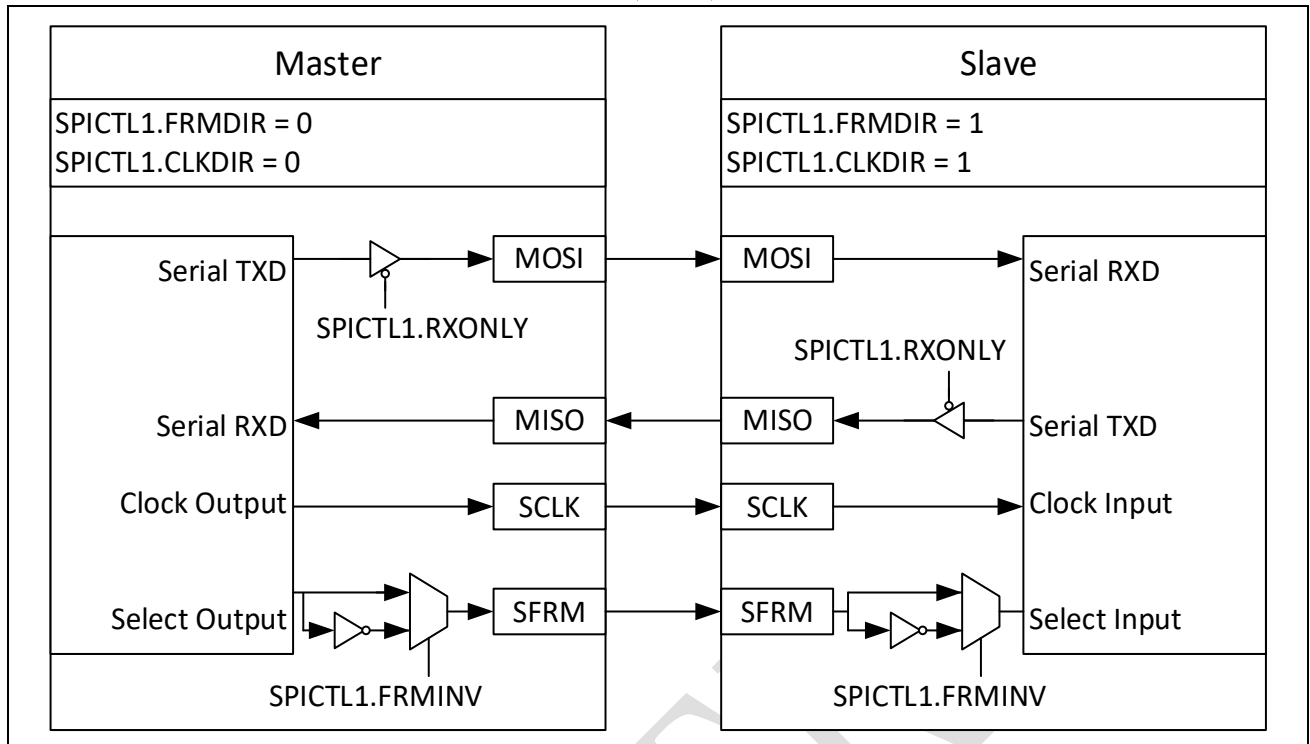
18.4 SPI 功能描述

本章节介绍 SPI 的各种功能操作，包括连接、波特率产生、FIFO 操作、数据帧格式和中断。

18.4.1 功能操作介绍

SPI 可以被配置为主机或者从机。主机发送信号 SCLK 和 SFRM 给从机，从机基于信号 SCLK 和 SFRM 收发数据。主机数据口（MOSI/MISO）与从机名称相同的数据口相连。主机通过 MOSI 发送数据，从 MISO 接收数据；从机从 MOSI 接收数据，通过 MISO 发送数据。SPI 主机和从机之间通信的典型连接如图 18-2 所示。

图 18-2: SPI 主机-从机连接示意图



数据传输存在三种可能的方式：

- 主机发送数据；从机也发送数据
- 主机发送数据；从机发送无效数据（Dummy data）
- 主机发送无效数据；从机发送数据

当发送无效数据时，如果不想从 TXD 信号接口向外部 MOSI 口发送数据，可以把 SPICTL1.RXONLY 寄存器位置“1”。

18.4.2 波特率产生

SPI 的波特率时钟由 SPI 时钟（SPICLK）产生，SPICLK 由 SYSCLK1 分频得到，SPI 时钟分频器为 16 位。

18.4.3 FIFO 操作

SPD1121 SPI 拥有两个独立的 FIFO，分别用于发送和接收。CPU 可以装载和清空这些 FIFO。CPU 每次访问可以传输一个 FIFO 条目（FIFO Entry），而且这些访问必须是 32 位宽度的。CPU 写数据到 TXFIFO 是 32 位宽度的，会忽略超过 FIFO 数据大小（SPICTL0.ESIZESEL 和 SPICTL0.SIZESEL）的高位数据。CPU 从 RXFIFO 读数据也是 32 位宽度的，FIFO 数据大小不够 32 位的，会在高位补 0。

对于数据发送，CPU 向 SPIDATA 写入数据，数据被存入 TXFIFO，SPI 接口从 TXFIFO 中获取数据，将数据串行化并通过 TXD 信号接口将数据发送到外部设备。对于数据接收，SPI 接口将 RXD 信号接口上的串行数据转换为并行数据并写入到 RXFIFO 中，CPU 通过 SPIDATA 读出 RXFIFO 中的数据。

当 FIFO 触发阈值 SPIRXTH 或者 SPITXTH 被超过时，会产生一个中断服务请求，用来通知 CPU 清空 RXFIFO 或重新装满 TXFIFO。

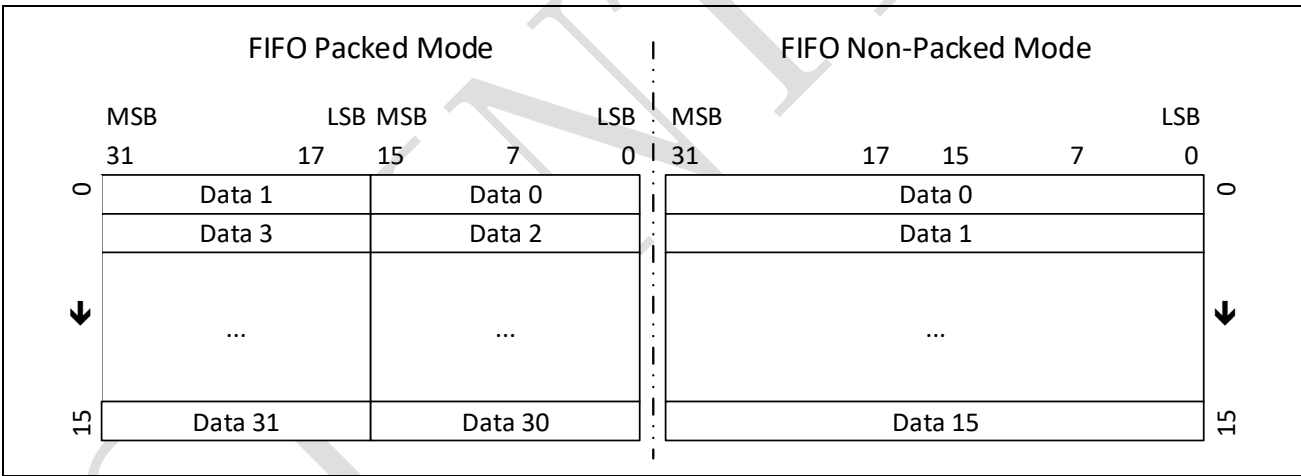
注意： 如果 SPI 被配置为从机且工作在只收模式下（SPICTL1.RXONLY=1），则在接收数据时无需写 TXFIFO，但写 TXFIFO 仍然会发送数据。

18.4.3.1 FIFO 存储的并行数据格式

数据大小可以是 8，16，18 或者 32 位。FIFO 中的数据是以 32 位形式存储或者以 16 位形式存储（压缩模式）。对于 32 位形式存储或者 16 位形式存储，存储的数据都是右对齐的，LSB 位在位 0。对于读操作，未使用的位会被填充 0；对于写操作，未使用的位不用去关心。SPI 模块中的逻辑会自动格式化在 TXFIFO 中的数据，以便数据可以在 TXD 信号上按选定的帧格式被恰当地发送。

当 FIFO 工作在压缩模式（packed mode）时，如图 18-3 所示，每一个 FIFO 是 32 x 16 位的，总共有 32 个数据。每一个数据在长度上可以是 8 位或者 16 位。当数据被串行化并被发送时，位 15~0 先被发送，紧接着发送位 31~16。对压缩模式来说，最佳做法是把 FIFO 的一个条目（entry）看做两个数据。因此，当 CPU 一次写或者读 32 位数据时，每次写或者读需要传输 2 个数据。整个 FIFO（32 位）都必须以这种模式来读或者写。此外，FIFO 的触发阈值也需要以 32 位读写来计算，不能基于 16 位。

图 18-3: FIFO 压缩和非压缩模式



读取 SPISTS.RFODDSTS 和 SPISTS.TFODDSTS 可以判断 FIFO 中的数据是偶数个还是奇数个。这些数据位有助于在 FIFO 压缩模式下正确地读取数据。

18.4.3.2 RXFIFO 中尾部字节（trailing bytes）

当 RXFIFO 中的数据个数小于 RXFIFO 的触发阈值且没有接收到其他数据，那么这些剩余字节数据就被称作 RXFIFO 尾部字节。RXFIFO 中的尾部字节是通过 CPU 查询模式来处理的。根据 SPISTS.RNE 寄存器位段的指示，读取所有的尾部字节。

18.4.3.3 FIFO 阈值

每个 FIFO 都有一个可配置的触发阈值，用来触发中断。当 RXFIFO 中的条目个数超过了 RXFIFO 的触发阈值（SPICTL1.RFTH 的值），会产生一个中断（如果该中断被使能）去通知 CPU 清空 RXFIFO。当 TXFIFO 中的条目个数小于或者等于 TXFIFO 的触发阈值（SPICTL1.TFTH 的值）加 1，会产生一个中断（如果该中断被使能）去通知 CPU 将 TXFIFO 填满。

CPU 可以查询 SPI 状态寄存器来判断 FIFO 中有多少数据或者 FIFO 是满还是空。软件负责选择合适的 RXFIFO/TXFIFO 触发阈值，以防止 RXFIFO 上溢或者 TXFIFO 下溢。

18.4.4 数据帧格式

18.4.4.1 时钟相位和极性控制

SPICTL1.CLKPOL 位段控制时钟的极性，用来指定时钟是高有效还是低有效，对传输格式没有显著影响。SPICTL1.CLKPHS 位段控制时钟相位，用来从两个基础的传输格式中选择一个。

对主机和与之通信的从机来说，时钟相位和极性必须是相同的。在一些情况下，时钟相位和极性在多个传输中间会被改变，以允许一个主机设备和不同需求的主机之间进行通信。

时钟信号 SCLK 只在传输数据时才翻转，而非一直在运行。

18.4.4.2 SFRM 极性控制

SPICTL1.FRMINV 位段控制片选信号的极性选择。通常情况下，空闲时，SFRM 为高电平；传输数据时，SFRM 为低电平。

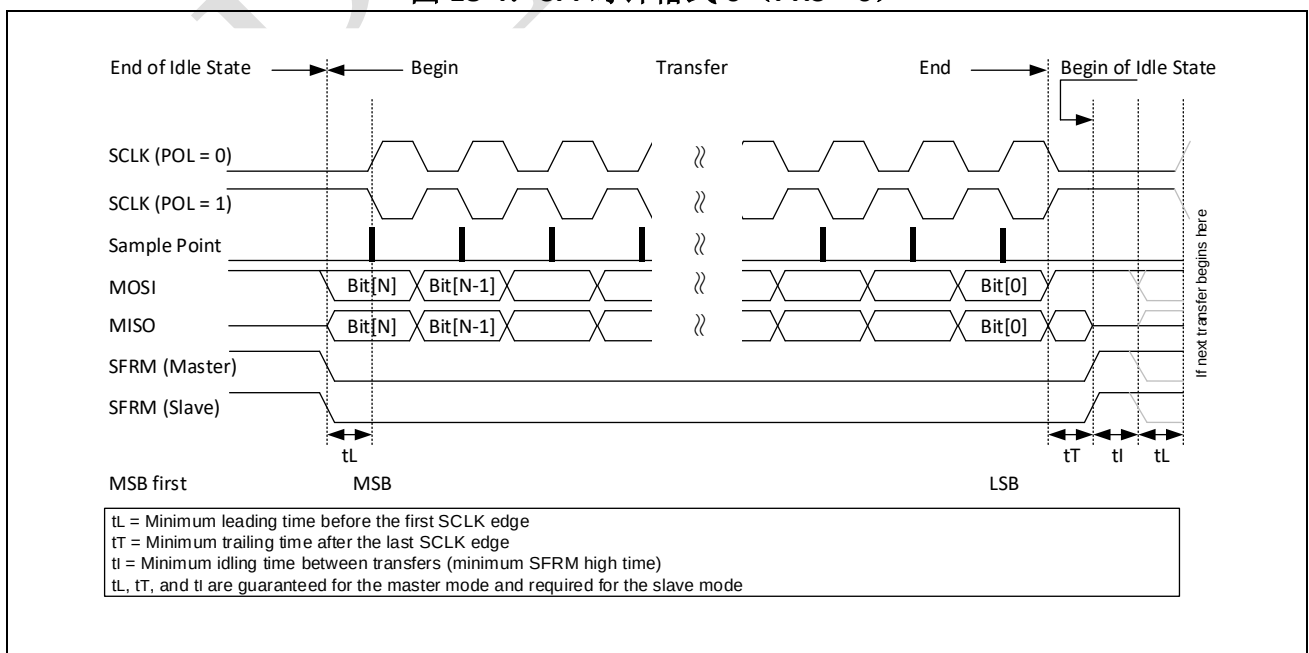
18.4.4.3 PHS = 0 时的帧格式

当 SPI 端口被关闭或者处于空闲模式，SFRM 信号为高电平，MOSI 和 MISO 为高阻态，SCL 为低电平（SPICTL1.CLKPOL = 0）或者高电平（SPICTL1.CLKPOL = 1），如图 18-4 所示。

当发送数据准备就绪且 SPICTL1.CLKPHS 为低电平时，主机和从机的第一个 SCLK 边沿用来采样第一个数据位，SFRM 变低（比 SCLK 第一个边沿早半个时钟周期）并在传输期间保持为低电平。SFRM 在最后一个 SCLK 边沿之后半个时钟周期变高。

发送数据的 MSB 位会在第一个 SCLK 边沿之前半个时钟周期被驱动到 MOSI/MISO 上。发送数据剩下的位会在偶数时钟边沿被驱动到 MOSI/MISO 上。在 SCLK 的奇数边沿会采样 MOSI/MISO 的数据位并移位到 LSB，为了下个时钟边沿的 MOSI/MISO 数据位，MSB 会被移位出去并被锁住。每个数据帧可以传输 1~32 位数据。

图 18-4: SPI 时钟格式 0 (PHS = 0)



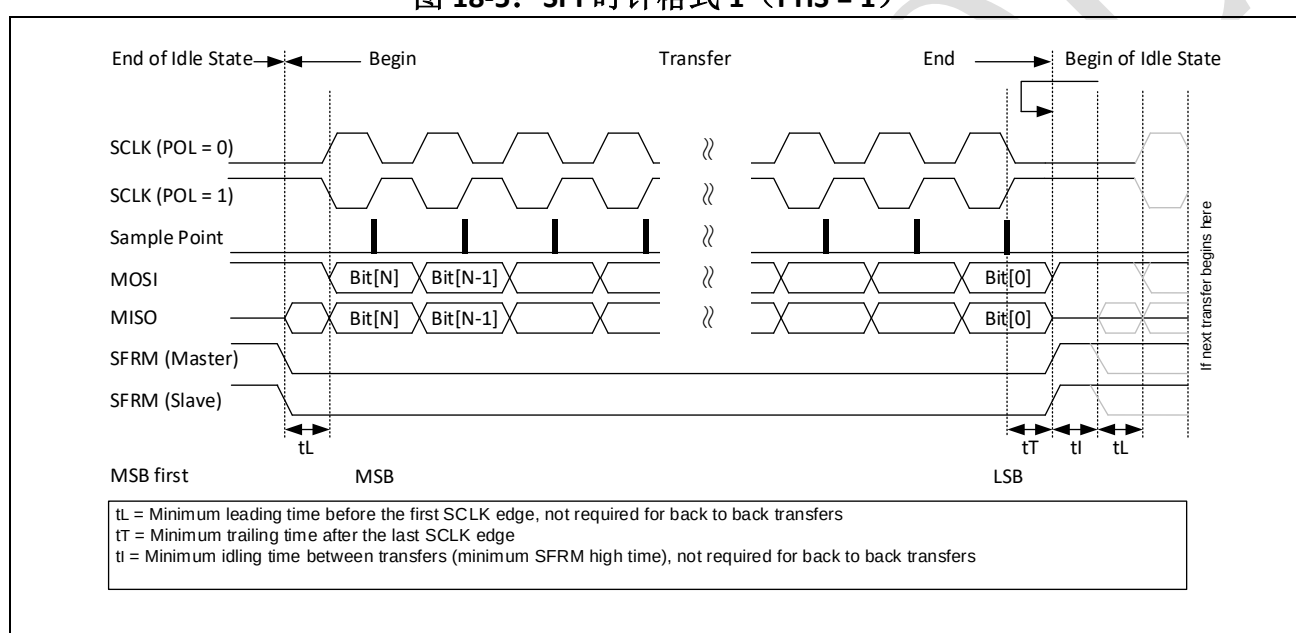
18.4.4.4 PHS = 1 时的帧格式

当 SPI 端口被关闭或者处于空闲模式，SFRM 信号为高电平，MOSI 和 MISO 为高阻态，SCL 为低电平（SPICTL1.CLKPOL = 0）或者高电平（SPICTL1.CLKPOL = 1），如图 18-5 所示。

当发送数据准备就绪且 SPICTL1.CLKPHS 为高电平时，主机和从机的第二个 SCLK 边沿用来采样第一个数据位，SFRM 变低（比 SCLK 第一个边沿早半个时钟周期）并在传输期间保持为低电平。SFRM 在最后一个 SCLK 边沿之后半个时钟周期变高。

发送数据的 MSB 位会在第一个 SCLK 边沿之前半个时钟周期被驱动到 MOSI/MISO 上。发送数据剩下的位会在奇数时钟边沿被驱动到 MOSI/MISO 上。在 SCLK 的偶数边沿会采样 MOSI/MISO 的数据位并移位到 LSB，为了下个时钟边沿的 MOSI/MISO 数据位，MSB 会被移位出去并被锁住。每个数据帧可以传输 1~32 位数据。

图 18-5: SPI 时钟格式 1（PHS = 1）



18.4.4.5 Back to back 传输模式

对于 back-to-back 传输，SFRM 信号在每个数据字传输之间保持有效。发送器和接收器要配置为相同的数据字大小，而且内部会跟踪数据帧的开始和结束。传输过程中一个数据帧的 LSB 位传输完后，紧接着传输下一帧的 MSB 位。

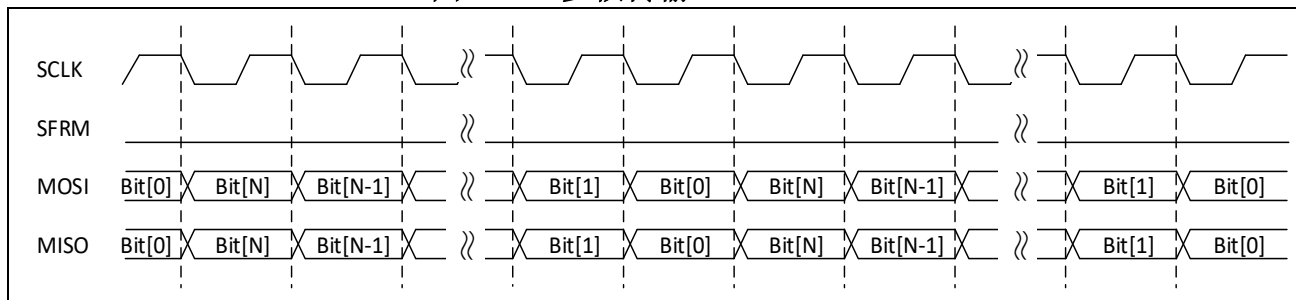
为了使能 back to back 模式，需要确保在传输数据时，TXFIFO 中至少有一个数据，并且在当前数据传输完成之前，TXFIFO 应当为空。

FIFO 压缩模式下，传输模式一定是 back to back 模式。

SPI 可以作为主机或者从机，当时时钟和片选信号的方向必须是相同的。因此，SPICTL1.CLKDIR 和 SPICTL1.FRMDIR 位段必须都是“1”或者“0”。

注意： 当片选信号无效时，SPI 的输入时钟必须无效。因此，当 SPI 为从机时，SPICTL1.SLVCLKSEL 位必须置“1”。

图 18-6: 多帧传输 (Motorola)



18.4.5 SPI 中断

SPI 状态寄存器 (SPISTS) 包含溢出错误标志位以及 TXFIFO 和 RXFIFO 服务请求标志位。每一个硬件检测的事件都会产生一个中断请求去通知中断控制器。SPISTS 寄存器也包含一些标志位用来指示 SPI 接口是否忙于发送数据, TXFIFO 是否非满以及 RXFIFO 是否非空。会有相应的中断信号被送到中断控制器。这些事件可以产生下面的中断: 接收器超时、RXFIFO 上溢、RXFIFO 服务请求以及 TXFIFO 服务请求。

产生中断的位段会保持为“1”, 直到这些位被写“1”清零。一旦一个状态位被清除, 相应的中断也被清除。可读/写的位段被称作状态位 (状态位被称为粘滞位, 一旦被硬件置“1”, 只能通过向该位写“1”清零), 只读的位段被称作标志位。向粘滞位写“1”会清零该位, 写“0”无效。只读位被置“1”后, 可以被硬件自动清零, 对只读位写无效的。一些产生中断请求的位段在控制寄存器 (SPICTL0/ SPICTL1) 中有相应的掩码位。

SPISTS 中的所有位都是只读的, 除了 RFOVF, TFUDF 和 RXTO 这些位是可读/写的。

SPISTS 可读/写的位段的复位值是 0。当 SPICTL0.EN 位段被清零时, 所有 SPISTS 位段的值都会返回到复位值。

18.5 寄存器

18.5.1 SPI 寄存器列表

表 18-2: SPI 模块基地址

外设模块	基地址
SPI	0x4000 5000

表 18-3: SPI 寄存器列表

寄存器	偏移地址	描述	复位值
SPICTL0	0x00	SPI 控制寄存器 0	0x00000000
SPICTL1	0x04	SPI 控制寄存器 1	0x00000000
SPISTS	0x08	SPI 状态寄存器	0x0040F004
SPIFRC	0x0C	SPI 中断强制寄存器	0x00000000
SPIDATA	0x10	SPI 数据寄存器	0x00000000
SPITO	0x28	SPI 超时时间寄存器	0x00000000

18.5.2 SPI 寄存器

18.5.2.1 SPI 控制寄存器 0 (SPICTL0)

表 18-4: SPI 控制寄存器 0 (SPICTL0) 位段定义

SPICTL0 (SPI Control Register 0)						
Access: SPI->SPICTL0 Offset: 0x0000 Default: 0x00000000						
31 RESERVED	30 RESERVED	29 FPACKEN	28 RESERVED	27 RESERVED	26 RESERVED	25 RESERVED
23 TFINTMSK	22 RFINTMSK	21 RESERVED	20 ESIZESEL	19 RESERVED	18 RESERVED	17 RESERVED
15 RESERVED	14 RESERVED	13 RESERVED	12 RESERVED	11 RESERVED	10 RESERVED	9 RESERVED
7 EN	6 RESERVED	5 FRMSEL	4 FRMSEL	3 RESERVED	2 SIZESEL	1 SIZESEL
0 RESERVED	0 RESERVED	0 RESERVED	0 RESERVED	0 RESERVED	0 RESERVED	0 RESERVED

表 18-5: SPI 控制寄存器 0 (SPICTL0) 位段描述

位段	位段名	属性	复位值	描述
31	RESERVED_31	RW	0x0	保留
30	RESERVED_30	RO	0x0	保留
29	FPACKEN	RW	0x0	使能 FIFO 压缩模式 0: 关闭 1: 使能
28:27	RESERVED_28_27	RO	0x0	保留
26:24	RESERVED_26_24	RW	0x0	保留
23	TFINTMSK	RW	0x0	TXFIFO 下溢中断掩码 0: TFUDF 事件产生中断请求 1: TFUDF 事件不产生中断请求
22	RFINTMSK	RW	0x0	RXFIFO 上溢中断掩码 0: RFOVF 事件产生中断请求 1: RFOVF 事件不产生中断请求
21	RESERVED_21	RO	0x0	保留
20	ESIZESEL	RW	0x0	扩展数据大小选择 0: 在 SIZESEL 位段值前补 0, 设置 SIZESEL 的范围为 1~16 位 1: 在 SIZESEL 位段值前补 1, 设置 SIZESEL 的范围为 17~32 位
19:8	RESERVED_19_8	RO	0x0	保留
7	EN	RW	0x0	使能 SPI 模块 0: 关闭 1: 使能

位段	位段名	属性	复位值	描述
6	RESERVED_6	RO	0x0	保留
5:4	FRMSEL	RW	0x0	帧格式选择 0: Motorola SPI 其他值: 无效
3:0	SIZESEL	RW	0x0	数据帧大小 0: 1 位 (ESIZESEL=0) 或 17 位 (ESIZESEL=1) 1: 2 位 (ESIZESEL=0) 或 18 位 (ESIZESEL=1) 2: 3 位 (ESIZESEL=0) 或 19 位 (ESIZESEL=1) 3: 4 位 (ESIZESEL=0) 或 20 位 (ESIZESEL=1) 4: 5 位 (ESIZESEL=0) 或 21 位 (ESIZESEL=1) 5: 6 位 (ESIZESEL=0) 或 22 位 (ESIZESEL=1) 6: 7 位 (ESIZESEL=0) 或 23 位 (ESIZESEL=1) 7: 8 位 (ESIZESEL=0) 或 24 位 (ESIZESEL=1) 8: 9 位 (ESIZESEL=0) 或 25 位 (ESIZESEL=1) 9: 10 位 (ESIZESEL=0) 或 26 位 (ESIZESEL=1) 10: 11 位 (ESIZESEL=0) 或 27 位 (ESIZESEL=1) 11: 12 位 (ESIZESEL=0) 或 28 位 (ESIZESEL=1) 12: 13 位 (ESIZESEL=0) 或 29 位 (ESIZESEL=1) 13: 14 位 (ESIZESEL=0) 或 30 位 (ESIZESEL=1) 14: 15 位 (ESIZESEL=0) 或 31 位 (ESIZESEL=1) 15: 16 位 (ESIZESEL=0) 或 32 位 (ESIZESEL=1)

18.5.2.2 SPI 控制寄存器 1 (SPICTL1)

表 18-6: SPI 控制寄存器 1 (SPICTL1) 位段定义

SPICTL1 (SPI Control Register 1)							
Access: SPI->SPICTL1 Offset: 0x0004 Default: 0x00000000							
31 TXTRITIME	30 TXTRIEN	29 RESERVED	28 SLVCLKSEL	27 RESERVED	26 RESERVED	25 CLKDIR	24 FRMDIR
23 RXONLY	22 TRAIL	21 RESERVED	20 RESERVED	19 RESERVED	18 RESERVED	17 RESERVED	16 FRMINV
15 RESERVED	14 RESERVED	13 RESERVED	12 RESERVED	11 RESERVED	10 RESERVED	9 RESERVED	8 RESERVED
7 RESERVED	6 RESERVED	5 RESERVED	4 CLKPHS	3 CLKPOL	2 RESERVED	1 TFIE	0 RFIE

表 18-7: SPI 控制寄存器 1 (SPICTL1) 位段描述

位段	位段名	属性	复位值	描述
31	TXTRITIME	RW	0x0	TXD 三态输出时刻选择 0: SPI_TXD 输出在 LSB 位开始后的 1/2 时钟周期变为三态 1: SPI_TXD 输出在 LSB 位结束时的时钟边沿变为三态
30	TXTRIEN	RW	0x0	使能 TXD 三态输出 0: SPI_TXD 输出信号不是三态 1: 当不发送数据时, SPI_TXD 输出为三态
29	RESERVED_29	RW	0x0	保留
28	SLVCLKSEL	RW	0x0	从机时钟控制 0: 当 SPICTL1.CLKDIR=1 时, 时钟输入 SPI_SCLK 无效 1: 当 SPICTL1.CLKDIR=1 时, 时钟输入 SPI_SCLK 有效
27:26	RESERVED_27_26	RO	0x0	保留
25	CLKDIR	RW	0x0	SPI 时钟信号 SPI_SCLK 方向 0: 主机模式, SPI 端口驱动时钟信号 SPI_SCLK 1: 从机模式, SPI 端口接收时钟信号 SPI_SCLK
24	FRMDIR	RW	0x0	SPI 片选信号 SPI_SFRM 方向 0: 主机模式, SPI 端口驱动片选信号 SPI_SFRM 1: 从机模式, SPI 端口接收片选信号 SPI_SFRM
23	RXONLY	RW	0x0	只收模式 0: 收发模式 1: 只收模式, SPI_TXD 输出高阻
22	RESERVED_22	RW	0x0	保留

位段	位段名	属性	复位值	描述
21	RESERVED_21	RW	0x0	保留
20	RESERVED_20	RW	0x0	保留
19	RESERVED_19	RW	0x0	保留
18:17	RESERVED_18_17	RO	0x0	保留
16	FRMINV	RW	0x0	反转片选信号极性 0: SPI_SFRM 空闲时为高电平，低电平有效 1: SPI_SFRM 空闲时为低电平，高电平有效
15	RESERVED_15	RW	0x0	保留
14	RESERVED_14	RW	0x0	保留
13:10	RFTH	RW	0x0	RXFIFO 中断触发阈值 本位段的值应该为预设的阈值减 1
9:6	TFTH	RW	0x0	TXFIFO 中断触发阈值 本位段的值应该为预设的阈值减 1
5	RESERVED_5	RO	0x0	保留
4	CLKPHS	RW	0x0	Motorola SPI SPI_SCLK 相位配置 0: 在帧传输开始后的第一个 SPI_SCLK 边沿开始采样数据 1: 在帧传输开始后的第二个 SPI_SCLK 边沿开始采样数据
3	CLKPOL	RW	0x0	Motorola SPI SPI_SCLK 极性配置 0: SPI_SCLK 空闲时为低电平 1: SPI_SCLK 空闲时为高电平
2	RESERVED_2	RW	0x0	保留
1	TFIE	RW	0x0	使能 TXFIFO 中断 0: 关闭 1: 使能
0	RFIE	RW	0x0	使能 RXFIFO 中断 0: 关闭 1: 使能

18.5.2.3 SPI 状态寄存器 (SPISTS)

表 18-8: SPI 状态寄存器 (SPISTS) 位段定义

SPISTS (SPI Status Register)							
Access: SPI->SPISTS Offset: 0x0008 Default: 0x0040F004							
31 RFODDSTS	30 TFODDSTS	29	28	27	26	25	24
RESERVED							
23 RESERVED	22 SLVCLKSTS	21 TFUDF	20 RESERVED	19 RXTO	18	17	16
RESERVED				RESERVED			
15	14	13	12	11	10	9	8
RFLVL				TFLVL			
7 RFOVF	6 RFS	5 TFS	4 BUSY	3 RNE	2 TNF	1	0
RESERVED							

表 18-9: SPI 状态寄存器 (SPISTS) 位段描述

位段	位段名	属性	复位值	描述
31	RFODDSTS	RO	0x0	RXFIFO 奇数数据状态 本位段仅在 FIFO 压缩模式使能时使用 0: RXFIFO 条目有 2 个数据 1: RXFIFO 条目有 1 个数据
30	TFODDSTS	RO	0x0	TXFIFO 奇数数据状态 总的数据个数为 TFLVL*2+TFODDSTS 本位段仅在 FIFO 压缩模式使能时使用 0: TXFIFO 条目有偶数个数据 1: TXFIFO 条目有奇数个数据
29:24	RESERVED_29_24	RO	0x0	保留
23	RESERVED_23	RW	0x0	保留
22	SLVCLKSTS	RO	0x0	从机时钟状态 0: 从机 SPI 时钟准备就绪 1: 从机 SPI 忙于同步信号
21	TFUDF	RW	0x0	TXFIFO 下溢标志 0: 读到 0 表明没有 TXFIFO 下溢, 写 0 无效 1: 读到 1 表明发生 TXFIFO 下溢, 写 1 清除该位
20	RESERVED_20	RO	0x0	保留
19	RXTO	RW	0x0	接收器超时标志 0: 读到 0 表明没有接收超时, 写 0 无效 1: 读到 1 表明发生接收超时, 写 1 清除该位
18:16	RESERVED_18_16	RO	0x0	保留
15:12	RFLVL	RO	0xF	RXFIFO 水平, 即数据 (32 位) 个数 本位段的值为 RXFIFO 中的条目个数减 1

位段	位段名	属性	复位值	描述
11:8	TFLVL	RO	0x0	TXFIFO 水平，即数据（32 位）个数 本位段的值为 TXFIFO 中的条目个数
7	RFOVF	RW	0x0	RXFIFO 上溢标志 0: 读到 0 表明没有 RXFIFO 上溢，写 0 无效 1: 读到 1 表明发生 RXFIFO 上溢，写 1 清除该位
6	RFS	RO	0x0	RXFIFO 服务请求标志 0: RXFIFO 水平等于或者小于 RXFIFO 阈值（RFTH）或者 SPI 已被关闭 1: RXFIFO 水平超过 RXFIFO 阈值（RFTH）
5	TFS	RO	0x0	TXFIFO 服务请求标志 0: TXFIFO 水平超过阈值（TFTH+1）或者 SPI 已被关闭 1: TXFIFO 水平等于或者小于阈值（TFTH+1）
4	BUSY	RO	0x0	SPI 忙状态 0: SPI 处于空闲状态或者被关闭 1: SPI 正在发送或者接收数据
3	RNE	RO	0x0	RXFIFO 非空 0: RXFIFO 空 1: RXFIFO 非空
2	TNF	RO	0x1	TXFIFO 非满 0: TXFIFO 满 1: TXFIFO 非满
1:0	RESERVED_1_0	RO	0x0	保留

18.5.2.4 SPI 中断强制寄存器 (SPIFRC)

表 18-10: SPI 中断强制寄存器 (SPIFRC) 位段定义

SPIFRC (SPI Interrupt Force Register)							
Access: SPI->SPIFRC		Offset: 0x000C		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RFOVF	RFREQ	TFREQ	RESERVED				

表 18-11: SPI 中断强制寄存器 (SPIFRC) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	RFOVF	RW	0x0	强制 RXFIFO 上溢中断 (不可屏蔽) 0: 写 0 释放中断强制 1: 写 1 强制 RXFIFO 上溢中断 本位段需要手动清零。
6	RFREQ	RW	0x0	强制 RXFIFO 服务请求中断 (不可屏蔽) 0: 写 0 释放中断强制 1: 写 1 强制 RXFIFO 服务请求中断 本位段需要手动清零。
5	TFREQ	RW	0x0	强制 TXFIFO 服务请求中断 (不可屏蔽) 0: 写 0 释放中断强制 1: 写 1 强制 TXFIFO 服务请求中断 本位段需要手动清零。
4:0	RESERVED_4_0	RO	0x0	保留

18.5.2.5 SPI 数据寄存器（SPIDATA）

表 18-12: SPI 数据寄存器（SPIDATA）位段定义

SPIDATA (SPI Data Register)							
Access: SPI->SPIDATA		Offset: 0x0010		Default: 0x00000000			
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 18-13: SPI 数据寄存器（SPIDATA）位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	要写入 TXFIFO 的数据或者从 RXFIFO 读取的数据

18.5.2.6 SPI 超时时间寄存器（SPITO）

表 18-14: SPI 超时时间寄存器（SPITO）位段定义

SPITO (SPI Time Out Register)							
Access: SPI->SPITO		Offset: 0x0028		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 18-15: SPI 超时时间寄存器（SPITO）位段描述

位段	位段名	属性	复位值	描述
31:24	RESERVED_31_24	RO	0x0	保留
23:0	VAL	RW	0x0	超时时间：定义为 CLK_SPI*(VAL+1)个时钟周期

19 嵌入式 Flash 存储器（FLASH）

19.1 Flash 主要特性

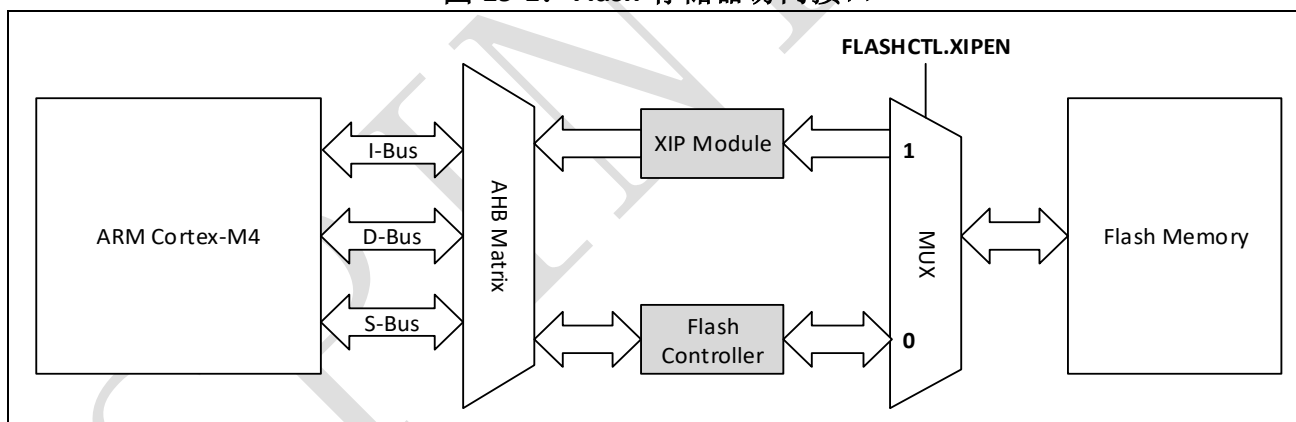
SPD1121 Flash 存储器模块的主要特性如下：

- 存储器最高容量可达 128KB 或 64KB
- 存储器构成：
 - 主存储区块：包含 256（或 128）个扇区，每个扇区包含 128 x 32 比特位
 - NVR 存储区块：包含 2 个扇区，每个扇区包含 128 x 32 比特位
- 擦写寿命：最少 100000 次擦写周期 @ $T_J=85^{\circ}\text{C}$
- 数据保存时间：超过 10 年 @ $T_J=85^{\circ}\text{C}$
- 支持误码检查和校正（Error Checking and Correction, ECC）

Flash 存储器接口特性：

- XIP（Execute-In-Place）读接口
- Flash 控制器接口：支持读/编程/擦除操作
- 写入保护

图 19-1: Flash 存储器访问接口



19.2 Flash 架构

Flash 存储器由一个主存储区和一个 NVR 存储区构成。主存储区由 256（或 128）个扇区构成，每个扇区包含 128 个基本存储器单元（32-bit）；NVR 存储区由 2 个扇区构成，每个扇区包含 128 个基本存储器单元（32-bit）。具体信息如图 19-2 和表 19-1 所示。

Flash 存储器的基本存储单元为 32 比特位宽（字），用于存储代码和数据常量。

NVR 存储区由两部分构成：

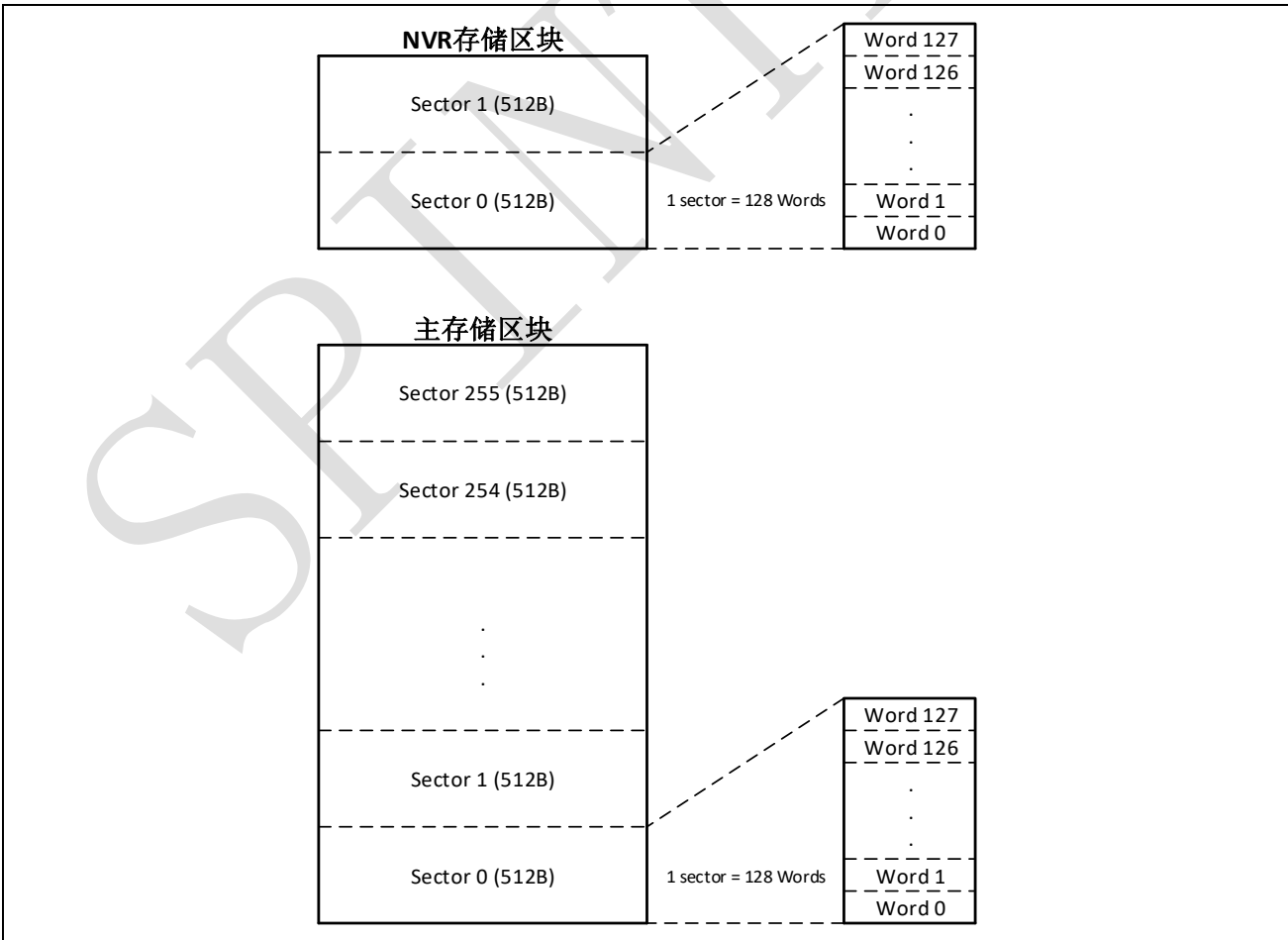
- OTP 存储区，该存储区实现 OTP 的方法是通过向该区域第一个存储器单元中写入数据 0x4C4F434B 实现的；如果第一个存储器单元的数据不等于 0x4C4F434B，那么该存储区就只能被当做普通的数据存储区使用。
- 配置字（Configuration Words）存储区，该区域的配置字用于实现 WDT 的开机使能以及 Debug 口锁定的功能。

对 Flash 主存储区和 NVR 存储区的写操作是通过 Flash 控制器接口实现的，如图 19-1 所示。编程和擦除操作所需要的高电压是由 Flash 模块内部产生的。

Flash 有扇区保护模式以防止非法的访问（写/擦除）。

在使用 Flash 控制器接口对存储器进行操作之前，必须将 XIP 接口关闭（FLASHCTL.XIPEN = 0）。也就是说，在同一时刻，Flash 控制器接口和 XIP 接口只能有一个接口处于工作状态。

图 19-2: Flash 模块逻辑结构



[1] 如果是 64KB 的片上 Flash，只包含 128 个扇区

表 19-1: Flash 模块构成

区块	名称	起始地址	大小
主存储区	扇区 0	0x1000 0000	512 字节
	扇区 1	0x1000 0200	512 字节
	扇区 2	0x1000 0400	512 字节
	扇区 3	0x1000 0600	512 字节
	...	...	...
	扇区 255	0x1001 FE00	512 字节
NVR 存储区	OTP 存储区	0x11000400	128 字 (Words)
	配置字存储区	0x11000600	128 字 (Words)
Flash 存储器接口寄存器	FLASHCTL	0x4000 8800	4 字节
	FLASHADDR	0x4000 8804	4 字节
	FLASHDIN	0x4000 8808	4 字节
	FLASHDOUT	0x4000 880C	4 字节
	FLASHTIMING0	0x4000 8814	4 字节
	FLASHTIMING1	0x4000 8818	4 字节
	FLASHTIMING2	0x4000 881C	4 字节
	FLASHTIMING3	0x4000 8820	4 字节
	FLASHTIMING4	0x4000 8824	4 字节
	FLASHTIMING5	0x4000 8828	4 字节
	FLASHWPO	0x4000 882C	4 字节
	FLASHWP1	0x4000 8830	4 字节
	FLASHWP2	0x4000 8834	4 字节
	FLASHWP3	0x4000 8838	4 字节
	FLASHREGKEY	0x4000 883C	4 字节

19.3 读操作

CPU 内核可以通过 XIP 接口或者 Flash 控制器接口从 Flash 模块中读取数据。

19.3.1 XIP 接口读操作

通过 XIP 接口, Flash 存储器可以被看做一个普通的存储空间, 能够以字节、半字或者字的方式直接通过地址访问。XIP 接口的主要任务就是产生读 Flash 存储器所需要的控制信号。Cortex-M4 内核使用 XIP 接口在 I 总线上取指令, 在 D 总线上取数据。

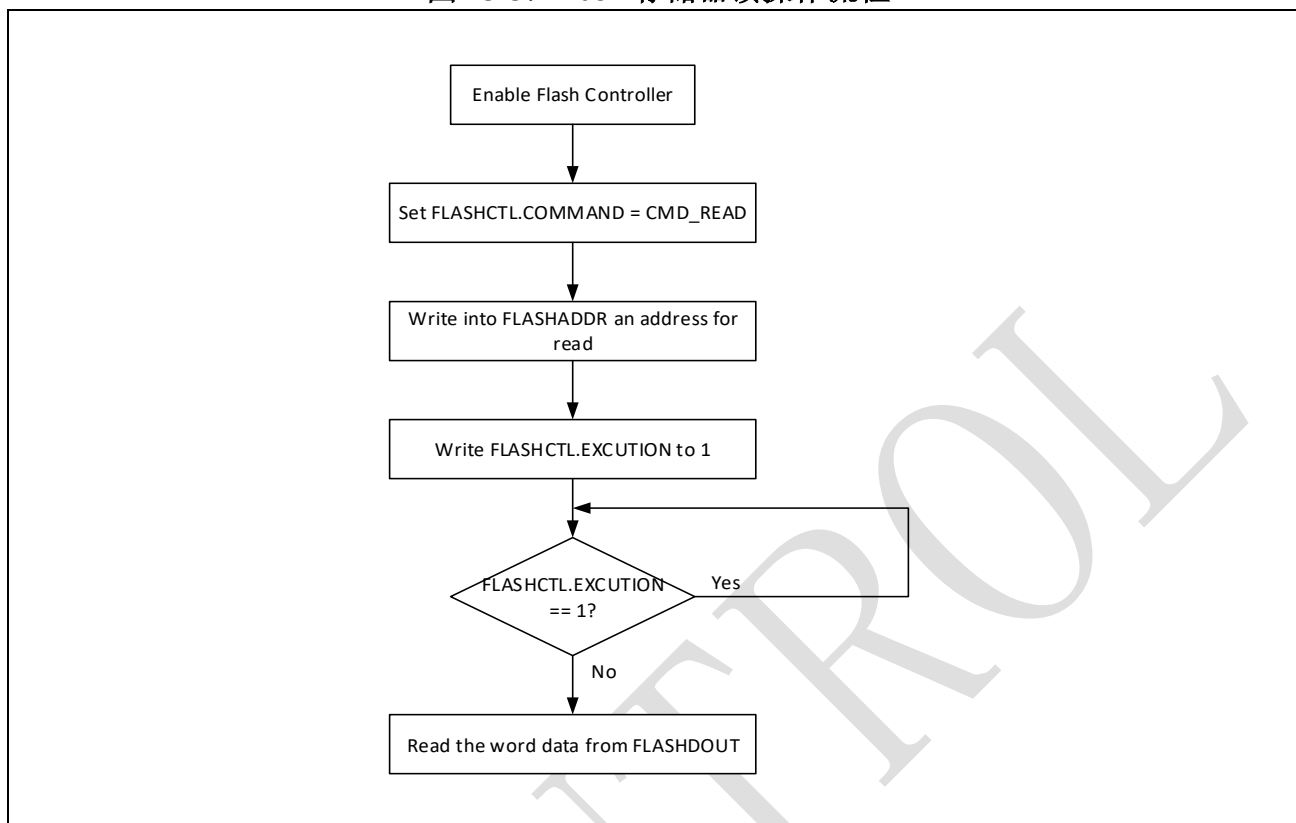
19.3.2 Flash 控制器接口读操作

CPU 内核也可以通过 Flash 控制器接口从 Flash 存储器中读取数据, 但是只能以字 (32 比特) 的方式读取。该读操作的过程如下:

- 设置 FLASHCTL 寄存器 COMMAND 位的值为 CMD_READ
- 设置 FLASHADDR 寄存器的值为要读取的存储器单元的地址
- 将 FLASHCTL 寄存器的 EXECUTION 位的值置 1
- 等待 EXECUTION 位的值复位

- 从 FLASHDOUT 寄存器中读取字（32 比特）数据

图 19-3: Flash 存储器读操作流程



19.4 编程和擦除操作

Flash 存储器的编程和擦除操作是通过 Flash 控制器接口实现的。Flash 控制器接口包含 15 个 32 位的寄存器：

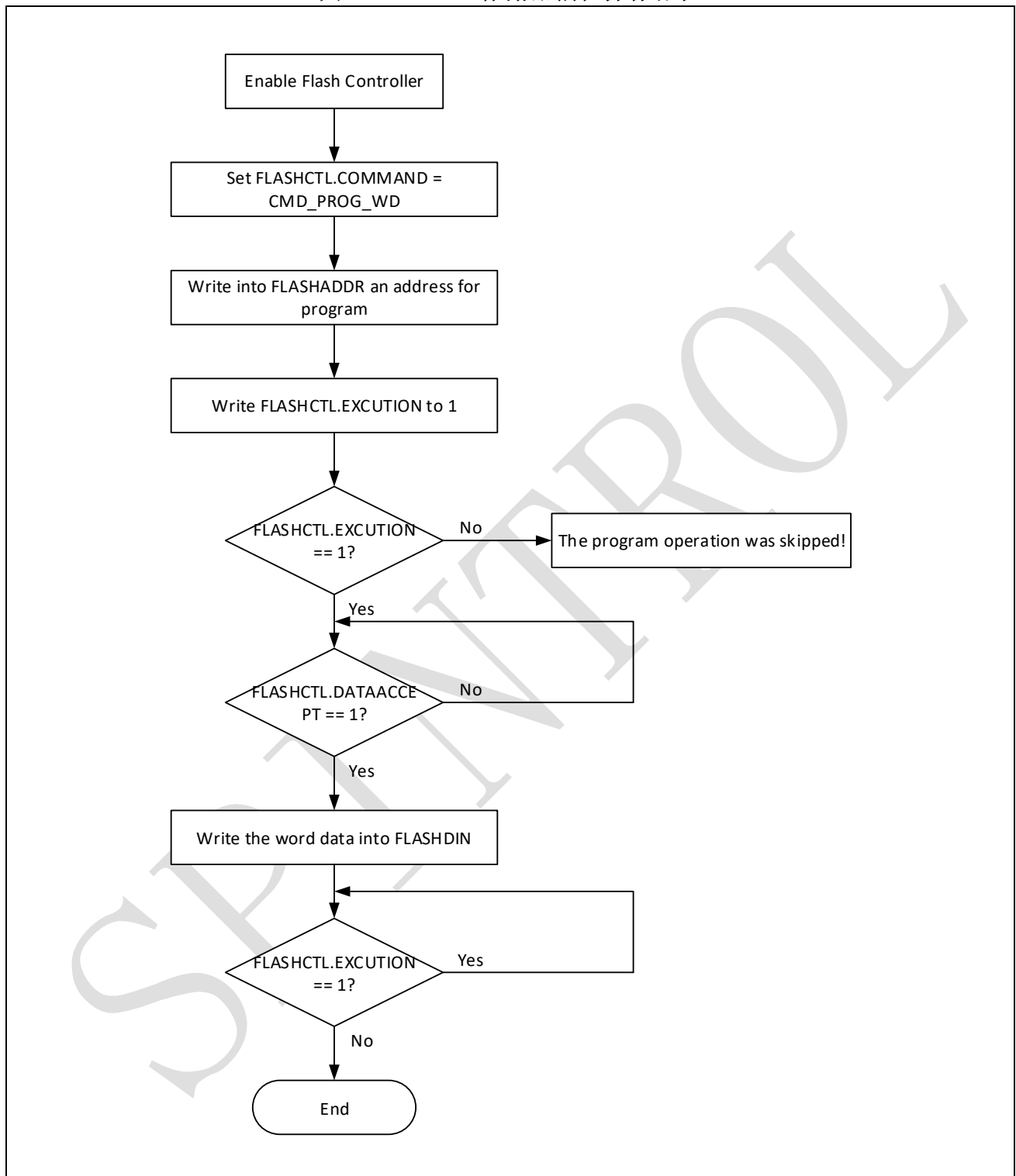
- Flash 控制寄存器（FLASHCTL）
- Flash 地址寄存器（FLASHADDR）
- Flash 数据输入寄存器（FLASHDIN）
- Flash 数据输出寄存器（FLASHDOUT）
- Flash 定时寄存器（FLASHTIMING0 ~ FLASHTIMING5）
- Flash 写保护寄存器（FLASHWPO ~ FLASHWPP3）
- Flash 控制器写使能寄存器（FLASHREGKEY）

19.4.1 使能 Flash 控制器

在芯片复位后，Flash 控制器处于关闭状态，可以通过将 FLASHCTL.XIPEN 设置为 0 来使能 Flash 控制器。然而，芯片上电后，FLASHCTL 寄存器处于写保护状态。因此，需要先向 FLASHREGKEY 寄存器中写入解锁值（0x1ACCE551）以便使能对这些被写保护的寄存器的写访问。

19.4.2 Flash 存储器编程操作

图 19-4: Flash 存储器编程操作流程



Flash 存储器的编程操作粒度可以到一个字（32 比特）。编程操作的过程如下：

- 设置 FLASHCTL 寄存器的 COMMAND 位的值为 CMD_PROG_WD
- 设置 FLASHADDR 寄存器的值为要编程的存储器单元的地址
- 将 FLASHCTL 寄存器的 EXCUTION 位的值置 1，然后再检查 EXCUTION 位的值是否被置 1；

如果未被置 1，则说明该存储器单元不可以被编程

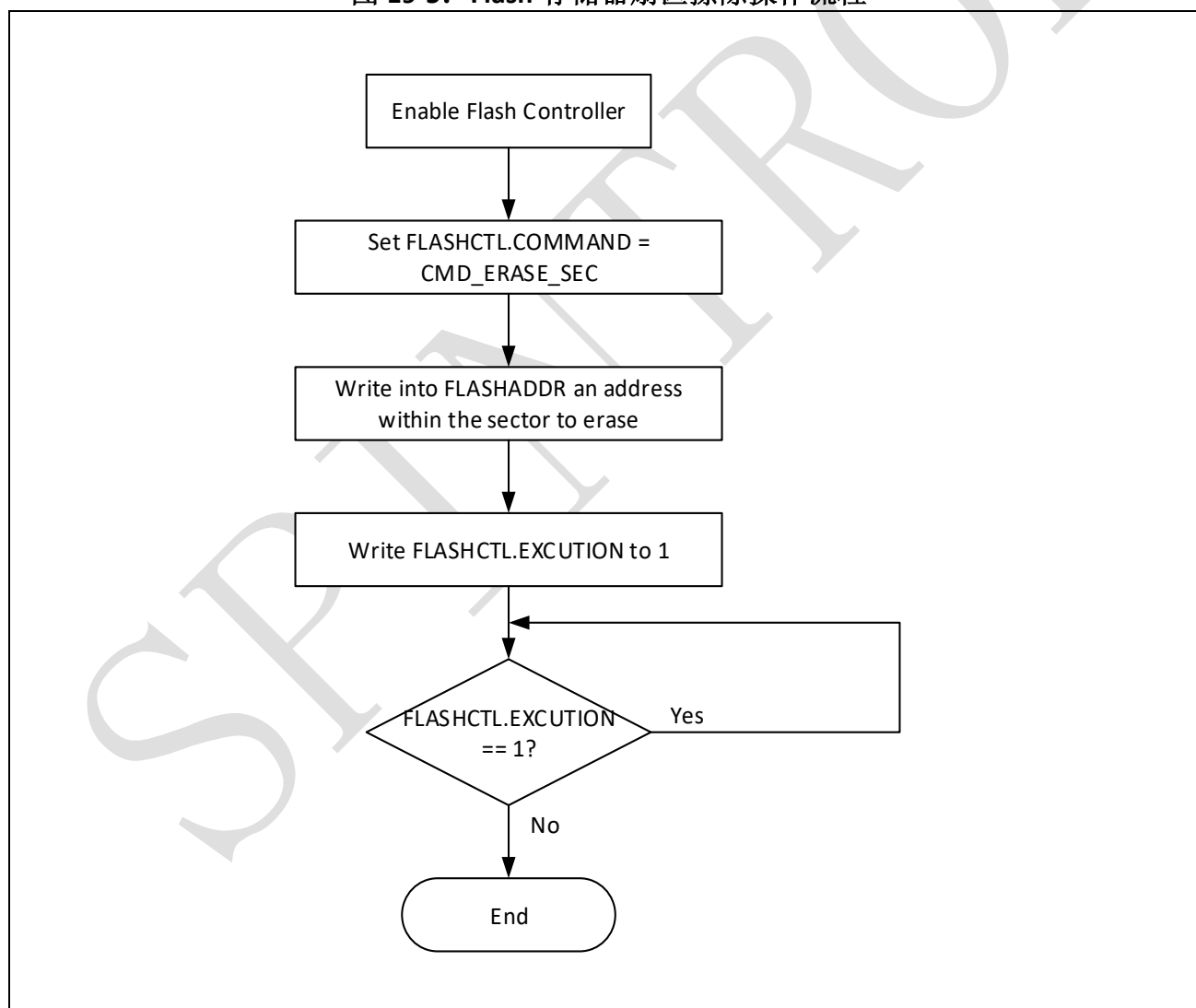
- 等待 FLASHCTL 寄存器的 DATAACCEPT 位的值被置 1
- 向 FLASHDIN 寄存器中写入一个 32 位数据
- 等待 FLASHCTL 寄存器的 EXECUTION 位的值复位

如果要编程的主存储器单元被设置了写保护（通过设置 FLASHWPPx 寄存器），那么 FLASHCTL 寄存器的 EXECUTION 位的值就不能够被置“1”，Flash 控制器会忽略对该存储器单元的编程操作。

19.4.3 Flash 存储器擦除操作

Flash 存储器的主存储区块可以按扇区擦除、块擦除或者整片擦除，而 NVR 存储区块只能按扇区擦除。

图 19-5: Flash 存储器扇区擦除操作流程



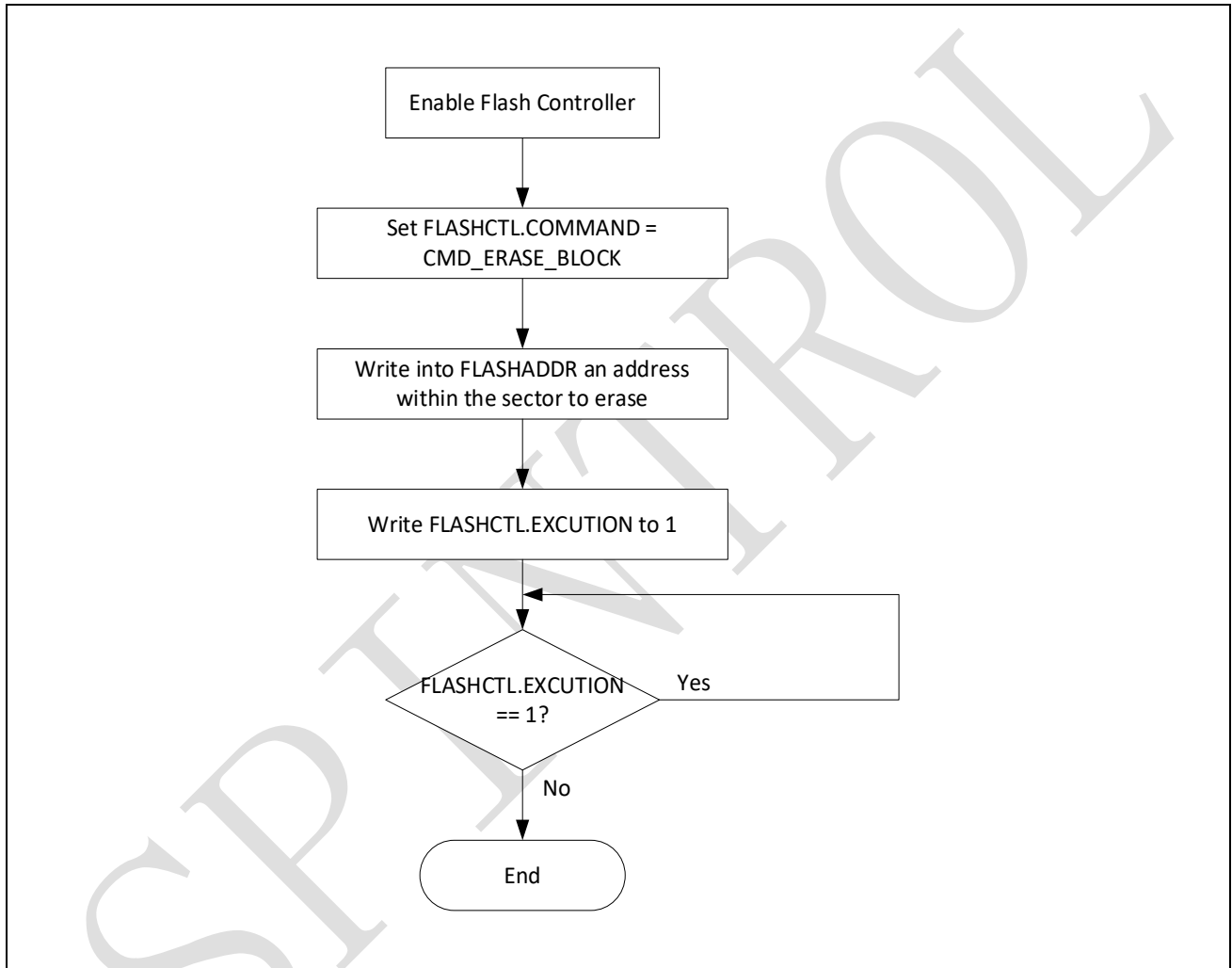
扇区擦除

Flash 存储器的扇区擦除可以通过 Flash 控制器的扇区擦除特性来实现。擦除一个扇区的过程如下：

- 设置 FLASHCTL 寄存器的 COMMAND 位的值为 CMD_ERASE_SEC
- 设置 FLASHADDR 寄存器的值为要擦除的扇区地址
- 将 FLASHCTL 寄存器的 EXECUTION 位的值置 1
- 等待 EXECUTION 位的值复位

如果要擦除的扇区被设置了写保护（通过设置 FLASHWPPx 寄存器），那么 FLASHCTL 寄存器的 EXECUTION 位的值就不能够被置“1”，Flash 控制器会忽略对该扇区的擦除操作。

图 19-6: Flash 存储器块擦除操作流程



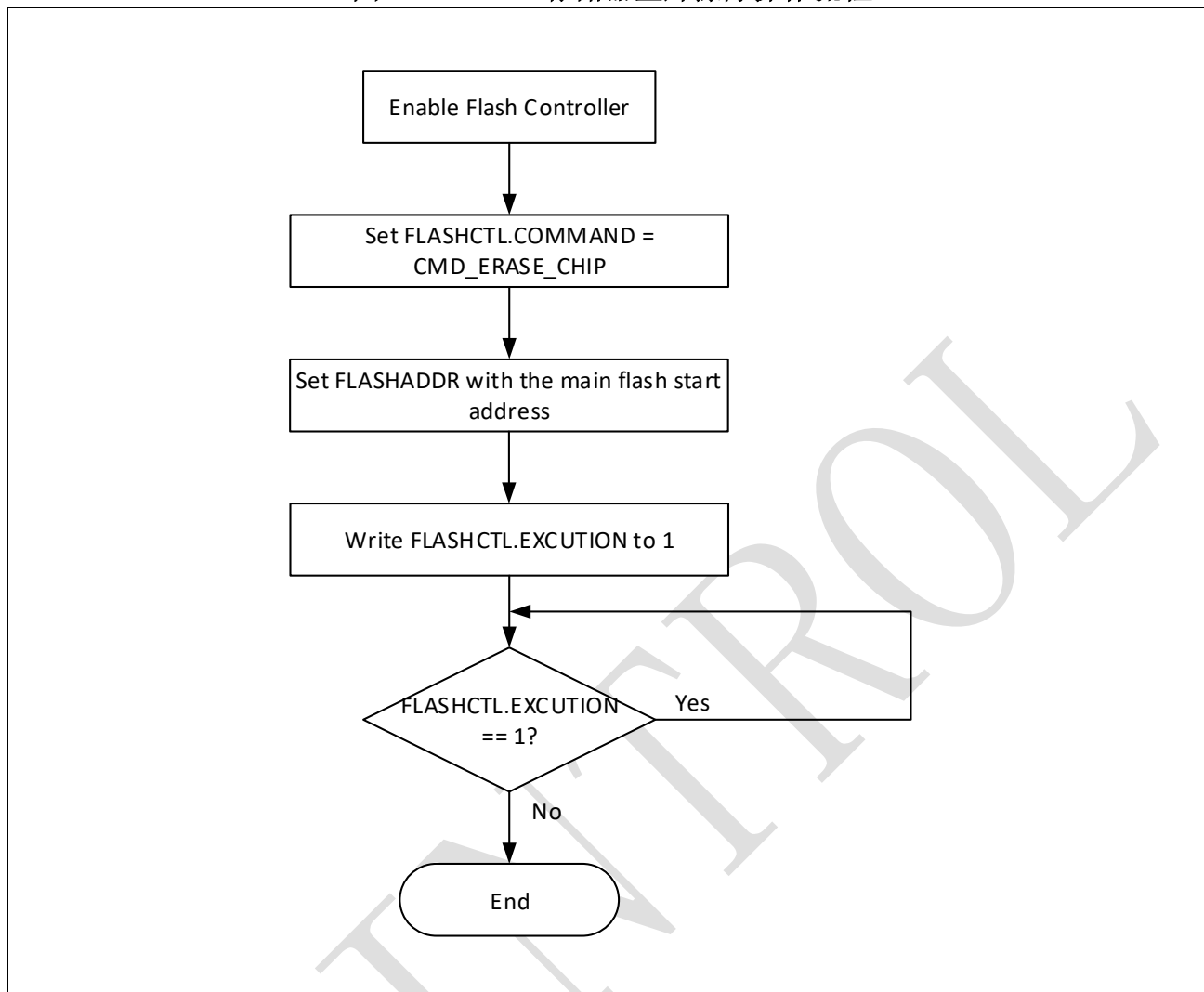
块擦除

Flash 存储器的块擦除可以通过 Flash 控制器的块擦除特性来实现。擦除一个块（从要擦除的扇区开始，共 8 个扇区为一个块）的过程如下：

- 设置 FLASHCTL 寄存器的 COMMAND 位的值为 CMD_ERASE_BLOCK
- 设置 FLASHADDR 寄存器的值为要擦除的块的起始地址
- 将 FLASHCTL 寄存器的 EXECUTION 位的值置 1
- 等待 EXECUTION 位的值复位

如果要擦除的扇区被设置了写保护（通过设置 FLASHWPPx 寄存器），那么 FLASHCTL 寄存器的 EXECUTION 位的值就不能够被置“1”，Flash 控制器会忽略对该扇区的擦除操作。

图 19-7: Flash 存储器整片擦除操作流程



整片擦除

整片擦除指令用于擦除 Flash 主存储区的所有扇区。NVR 存储区不会受到整片擦除指令的影响。推荐的整片擦除过程如下：

- 设置 FLASHCTL 寄存器的 COMMAND 位的值为 CMD_ERASE_CHIP
- 设置 FLASHADDR 寄存器的值为 Flash 主存储区的起始地址
- 将 FLASHCTL 寄存器的 EXCUTION 位的值置 1
- 等待 EXCUTION 位的值复位

19.5 保护机制

19.5.1 写保护

Flash 主存储区的扇区可以被保护起来，以免遭程序跑飞而导致的非法写入操作。Flash 主存储区的写保护的可以实现单个扇区的保护。如果对一个使能了写保护的扇区进行编程或者擦除操作，该操作会被忽略掉。

扇区的写保护功能可以通过将 FLASHWPy 寄存器相应位的值置 1 来激活；而将 FLASHWPy 寄存器相应位的值清零会关闭相应扇区的写保护功能。

19.6 Flash 时序设置

为了防止 Flash 存储器出现异常，对 Flash 的时序是有要求的。Flash 的时序是一系列事件之间时序关系的数值。在 XIP 接口和 Flash 控制器中设计有一个有限状态机（FSM），该有限状态机通过对时钟周期的计数判断是否超时，从而保证时序的要求。为了方便用户设置 Flash 时序，在 SDK 中提供了相应的函数 FLASH_SetTiming。该函数以 CPU 内核时钟（HCLK）频率作为输入参数，可以自行设置 Flash 相关的时序参数。

无论在何时，只要想提高 HCLK 的频率，就必须在改变 HCLK 频率前调用 FLASH_SetTiming 函数，这是增加 Flash 时序参数的数值所要求的。

无论在何时，只要想降低 HCLK 的频率，就必须在改变 HCLK 频率后调用 FLASH_SetTiming 函数，这是降低 Flash 时序参数的数值所要求的。

19.7 配置字（Configuration Words）

配置字的构成及其功能描述如表 19-2 所示。配置字的值可以通过 XIP 接口或者 Flash 控制器接口按照表 19-2 所示的地址进行读取。新写入的配置字需要在芯片复位并重新加载后才能生效。

表 19-2：配置字的构成及其描述

地址	配置字名称	配置字描述
0x1100060C	CHIP_SECURITY	芯片调试接口锁定字
0x11000614		均为 0xFFFFFFFF：芯片调试接口将不被锁定 其他数值：芯片调试接口将被锁定
0x11000700	WDT_ENABLE	看门狗使能字 0xFFFFFFFF：芯片启动时禁用看门狗 其他数值：芯片启动时启用看门狗

19.8 寄存器

19.8.1 Flash 寄存器列表

表 19-3: Flash 控制器模块基地址

外设模块	基地址
FLASH	0x4000 8800

表 19-4: Flash 控制器寄存器列表

寄存器	偏移地址	描述	复位值
FLASHCTL*	0x0000	Flash 控制寄存器	0x00000000
FLASHADDR	0x0004	Flash 地址寄存器	0x00000000
FLASHDIN	0x0008	Flash 数据输入寄存器	0x00000000
FLASHDOUT	0x000C	Flash 数据输出寄存器	0x00000000
FLASHTIMING0*	0x0014	Flash 定时寄存器 0	0x3F2C0081
FLASHTIMING1*	0x0018	Flash 定时寄存器 1	0x2C80C818
FLASHTIMING2*	0x001C	Flash 定时寄存器 2	0x30015DBE
FLASHTIMING3*	0x0020	Flash 定时寄存器 3	0x00017046
FLASHTIMING4*	0x0024	Flash 定时寄存器 4	0x440000C0
FLASHTIMING5*	0x0028	Flash 定时寄存器 5	0x0001B000
FLASHWP0*	0x002C	Flash 写保护寄存器 0	0x00000000
FLASHWP1*	0x0030	Flash 写保护寄存器 1	0x00000000
FLASHWP2*	0x0034	Flash 写保护寄存器 2	0x00000000
FLASHWP3*	0x0038	Flash 写保护寄存器 3	0x00000000
FLASHREGKEY	0x003C	Flash 控制器写使能寄存器	0x1ACCE551

注意： 被 ‘*’ 标记的寄存器只有当 FLASHREGKEY=0x1ACCE551 时才允许写操作。

19.8.2 Flash 寄存器

19.8.2.1 Flash 控制寄存器（FLASHCTL）

表 19-5: Flash 控制寄存器（FLASHCTL）位段定义

FLASHCTL (Flash Control Register)							
Access: FLASH->FLASHCTL Offset: 0x0000 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED	XIPEN	EXECUTION	DATAACCEPT		COMMAND		

表 19-6: Flash 控制寄存器（FLASHCTL）位段描述

位段	位段名称	属性	复位值	描述
31:7	RESERVED_31_7	RO	0x0	Reserved
6	XIPEN	RW	0x0	XIP 接口使能 0: 关闭 1: 使能
5	EXECUTION	RW	0x0	命令处于执行状态 0: 命令执行完毕 1: 命令正在执行
4	DATAACCEPT	RO	0x0	该位为 0 时，FLASHDIN 的值不可变
3:0	COMMAND	RW	0x0	Flash 命令 0: 上电 1: 读 2: Verify read 0 3: Verify read 1 4: 预编程字 5: 程序字 6: 预程序页 7: 程序页 8: 擦除扇区 09: 擦除块 10: 整片擦除 13: Recall read 14: 设置配置 15: 断电

19.8.2.2 Flash 地址寄存器（FLASHADDR）

表 19-7: Flash 地址寄存器（FLASHADDR）位段定义

FLASHADDR (Flash Address Register)							
Access: FLASH->FLASHADDR				Offset: 0x0004		Default: 0x00000000	
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 19-8: Flash 地址寄存器（FLASHADDR）位段描述

位段	位段名称	属性	复位值	描述
31:0	VAL	RW	0x0	Flash 地址

19.8.2.3 Flash 数据输入寄存器（FLASHDIN）

表 19-9: Flash 数据输入寄存器（FLASHDIN）位段定义

FLASHDIN (Flash Data Input Register)							
Access: FLASH->FLASHDIN				Offset: 0x0008		Default: 0x00000000	
31	30	29	28	27	26	25	24
FLASHDIN							
23	22	21	20	19	18	17	16
FLASHDIN							
15	14	13	12	11	10	9	8
FLASHDIN							
7	6	5	4	3	2	1	0
FLASHDIN							

表 19-10: Flash 数据输入寄存器（FLASHDIN）位段描述

位段	位段名称	属性	复位值	描述
31:0	FLASHDIN	RW	0x0	Flash 输入数据

19.8.2.4 Flash 数据输出寄存器（FLASHDOUT）

表 19-11: Flash 数据输出寄存器（FLASHDOUT）位段定义

FLASHDOUT (Flash Data Output Register)							
Access: FLASH->FLASHDOUT Offset: 0x000C Default: 0x00000000							
31	30	29	28	27	26	25	24
FLASHDOUT							
23	22	21	20	19	18	17	16
FLASHDOUT							
15	14	13	12	11	10	9	8
FLASHDOUT							
7	6	5	4	3	2	1	0
FLASHDOUT							

表 19-12: Flash 数据输出寄存器（FLASHDOUT）位段描述

位段	位段名称	属性	复位值	描述
31:0	FLASHDOUT	RO	0x0	Flash 输出数据

19.8.2.5 Flash 定时寄存器 0（FLASHTIMING0）

表 19-13: Flash 定时寄存器 0（FLASHTIMING0）位段定义

FLASHTIMING0 (Flash Timing Register 0)							
Access: FLASH->FLASHTIMING0 Offset: 0x0014 Default: 0x3F2C0081							
31	30	29	28	27	26	25	24
RESERVED		TNVSCHIP					
23	22	21	20	19	18	17	16
TNVSSEC						RESERVED	
15	14	13	12	11	10	9	8
RESERVED				TAA			
7	6	5	4	3	2	1	0
TAA		TAEPH					

表 19-14: Flash 定时寄存器 0（FLASHTIMING0）位段描述

位段	位段名称	属性	复位值	描述
31:30	RESERVED_31_30	RO	0x0	保留
29:24	TNVSCHIP	RW	0x3F	—
23:18	TNVSSEC	RW	0xB	—
17:12	RESERVED_17_12	RO	0x0	保留
11:6	TAA	RW	0x2	—
5:0	TAEPH	RW	0x1	—

19.8.2.6 Flash 定时寄存器 1 (FLASHTIMING1)

表 19-15: Flash 定时寄存器 1 (FLASHTIMING1) 位段定义

FLASHTIMING1 (Flash Timing Register 1)							
Access: FLASH->FLASHTIMING1 Offset: 0x0018 Default: 0x2C80C818							
31	30	29	28	27	26	25	24
RESERVED		TMS					
23	22	21	20	19	18	17	16
TMS			RESERVED		TUS		
15	14	13	12	11	10	9	8
TUS				TSBE			
7	6	5	4	3	2	1	0
TSBE		TSSE					

表 19-16: Flash 定时寄存器 1 (FLASHTIMING1) 位段描述

位段	位段名称	属性	复位值	描述
31:30	RESERVED_31_30	RO	0x0	保留
29:21	TMS	RW	0x164	—
20:19	RESERVED_20_19	RO	0x0	保留
18:12	TUS	RW	0xC	—
11:6	TSBE	RW	0x20	—
5:0	TSSE	RW	0x18	—

19.8.2.7 Flash 定时寄存器 2 (FLASHTIMING2)

表 19-17: Flash 定时寄存器 2 (FLASHTIMING2) 位段定义

FLASHTIMING2 (Flash Timing Register 2)							
Access: FLASH->FLASHTIMING2 Offset: 0x001C Default: 0x30015DBE							
31	30	29	28	27	26	25	24
RESERVED		TRCVPROG					
23	22	21	20	19	18	17	16
RESERVED						TPROG	
15	14	13	12	11	10	9	8
TPROG				TPGS			
7	6	5	4	3	2	1	0
TPGS		TSCE					

表 19-18: Flash 定时寄存器 2 (FLASHTIMING2) 位段描述

位段	位段名称	属性	复位值	描述
31:30	RESERVED_31_30	RO	0x0	保留
29:24	TRCVPROG	RW	0x30	—
23:18	RESERVED_23_18	RO	0x0	保留
17:12	TPROG	RW	0x15	—
11:6	TPGS	RW	0x36	—
5:0	TSCE	RW	0x3E	—

19.8.2.8 Flash 定时寄存器 3 (FLASHTIMING3)

表 19-19: Flash 定时寄存器 3 (FLASHTIMING3) 位段定义

FLASHTIMING3 (Flash Timing Register 3)							
Access: FLASH->FLASHTIMING3 Offset: 0x0020 Default: 0x00017046							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED						TAAR	
15	14	13	12	11	10	9	8
TAAR				TAPS			
7	6	5	4	3	2	1	0
TAPS		TPREPROG					

表 19-20: Flash 定时寄存器 3 (FLASHTIMING3) 位段描述

位段	位段名称	属性	复位值	描述
31:18	RESERVED_31_18	RO	0x0	保留
17:12	TAAR	RW	0x17	—
11:6	TAPS	RW	0x1	—
5:0	TPREPROG	RW	0x6	—

19.8.2.9 Flash 定时寄存器 4 (FLASHTIMING4)

表 19-21: Flash 定时寄存器 4 (FLASHTIMING4) 位段定义

FLASHTIMING4 (Flash Timing Register 4)							
Access: FLASH->FLASHTIMING4 Offset: 0x0024 Default: 0x440000C0							
31	30	29	28	27	26	25	24
RESERVED	TRCVERASE						
23	22	21	20	19	18	17	16
RESERVED						RESERVED	
15	14	13	12	11	10	9	8
RESERVED				TRW			
7	6	5	4	3	2	1	0
TRW		RESERVED					

表 19-22: Flash 定时寄存器 4 (FLASHTIMING4) 位段描述

位段	位段名称	属性	复位值	描述
31	RESERVED_31	RO	0x0	保留
30:24	TRCVERASE	RW	0x44	—
23:18	RESERVED_23_18	RO	0x0	保留
17:12	RESERVED_17_12	RO	0x0	保留
11:6	TRW	RW	0x3	—
5:0	RESERVED_5_0	RO	0x0	保留

19.8.2.10 Flash 定时寄存器 5（FLASHTIMING5）

表 19-23: Flash 定时寄存器 5（FLASHTIMING5）位段定义

FLASHTIMING5 (Flash Timing Register 5)							
Access: FLASH->FLASHTIMING5 Offset: 0x0028 Default: 0x0001B000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED						TRSVREAD	
15	14	13	12	11	10	9	8
TRSVREAD				RESERVED			
7	6	5	4	3	2	1	0
RESERVED							

表 19-24: Flash 定时寄存器 5（FLASHTIMING5）位段描述

位段	位段名称	属性	复位值	描述
31:18	RESERVED_31_18	RO	0x0	保留
17:12	TRSVREAD	RW	0x1B	–
11:0	RESERVED_11_0	RO	0x0	保留

19.8.2.11 Flash 写保护寄存器 0 (FLASHWP0)

表 19-25: Flash 写保护寄存器 0 (FLASHWP0) 位段定义

FLASHWP0 (Flash Write Protect Register 0)							
Access: FLASH->FLASHWP0		Offset: 0x002C		Default: 0x00000000			
31 UNIT31	30 UNIT30	29 UNIT29	28 UNIT28	27 UNIT27	26 UNIT26	25 UNIT25	24 UNIT24
23 UNIT23	22 UNIT22	21 UNIT21	20 UNIT20	19 UNIT19	18 UNIT18	17 UNIT17	16 UNIT16
15 UNIT15	14 UNIT14	13 UNIT13	12 UNIT12	11 UNIT11	10 UNIT10	9 UNIT9	8 UNIT8
7 UNIT7	6 UNIT6	5 UNIT5	4 UNIT4	3 UNIT3	2 UNIT2	1 UNIT1	0 UNIT0

表 19-26: Flash 写保护寄存器 0 (FLASHWP0) 位段描述

位段	位段名称	属性	复位值	描述
31	UNIT31	RW	0x0	Flash 单元 31 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
30	UNIT30	RW	0x0	Flash 单元 30 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
29	UNIT29	RW	0x0	Flash 单元 29 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
28	UNIT28	RW	0x0	Flash 单元 28 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
27	UNIT27	RW	0x0	Flash 单元 27 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
26	UNIT26	RW	0x0	Flash 单元 26 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
25	UNIT25	RW	0x0	Flash 单元 25 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
24	UNIT24	RW	0x0	Flash 单元 24 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
23	UNIT23	RW	0x0	Flash 单元 23 写保护

位段	位段名称	属性	复位值	描述
				0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
22	UNIT22	RW	0x0	Flash 单元 22 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
21	UNIT21	RW	0x0	Flash 单元 21 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
20	UNIT20	RW	0x0	Flash 单元 20 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
19	UNIT19	RW	0x0	Flash 单元 19 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
18	UNIT18	RW	0x0	Flash 单元 18 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
17	UNIT17	RW	0x0	Flash 单元 17 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
16	UNIT16	RW	0x0	Flash 单元 16 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
15	UNIT15	RW	0x0	Flash 单元 15 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
14	UNIT14	RW	0x0	Flash 单元 14 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
13	UNIT13	RW	0x0	Flash 单元 13 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
12	UNIT12	RW	0x0	Flash 单元 12 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
11	UNIT11	RW	0x0	Flash 单元 11 写保护

位段	位段名称	属性	复位值	描述
				0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
10	UNIT10	RW	0x0	Flash 单元 10 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
9	UNIT9	RW	0x0	Flash 单元 9 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
8	UNIT8	RW	0x0	Flash 单元 8 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
7	UNIT7	RW	0x0	Flash 单元 7 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
6	UNIT6	RW	0x0	Flash 单元 6 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
5	UNIT5	RW	0x0	Flash 单元 5 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
4	UNIT4	RW	0x0	Flash 单元 4 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
3	UNIT3	RW	0x0	Flash 单元 3 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
2	UNIT2	RW	0x0	Flash 单元 2 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
1	UNIT1	RW	0x0	Flash 单元 1 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
0	UNIT0	RW	0x0	Flash 单元 0 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除

19.8.2.12 Flash 写保护寄存器 1 (FLASHWP1)

表 19-27: Flash 写保护寄存器 1 (FLASHWP1) 位段定义

FLASHWP1 (Flash Write Protect Register 1)							
Access: FLASH->FLASHWP1		Offset: 0x0030		Default: 0x00000000			
31 UNIT63	30 UNIT62	29 UNIT61	28 UNIT60	27 UNIT59	26 UNIT58	25 UNIT57	24 UNIT56
23 UNIT55	22 UNIT54	21 UNIT53	20 UNIT52	19 UNIT51	18 UNIT50	17 UNIT49	16 UNIT48
15 UNIT47	14 UNIT46	13 UNIT45	12 UNIT44	11 UNIT43	10 UNIT42	9 UNIT41	8 UNIT40
7 UNIT39	6 UNIT38	5 UNIT37	4 UNIT36	3 UNIT35	2 UNIT34	1 UNIT33	0 UNIT32

表 19-28: Flash 写保护寄存器 1 (FLASHWP1) 位段描述

位段	位段名称	属性	复位值	描述
31	UNIT63	RW	0x0	Flash 单元 63 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
30	UNIT62	RW	0x0	Flash 单元 62 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
29	UNIT61	RW	0x0	Flash 单元 61 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
28	UNIT60	RW	0x0	Flash 单元 60 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
27	UNIT59	RW	0x0	Flash 单元 59 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
26	UNIT58	RW	0x0	Flash 单元 58 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
25	UNIT57	RW	0x0	Flash 单元 57 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
24	UNIT56	RW	0x0	Flash 单元 56 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
23	UNIT55	RW	0x0	Flash 单元 55 写保护

位段	位段名称	属性	复位值	描述
				0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
22	UNIT54	RW	0x0	Flash 单元 54 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
21	UNIT53	RW	0x0	Flash 单元 53 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
20	UNIT52	RW	0x0	Flash 单元 52 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
19	UNIT51	RW	0x0	Flash 单元 51 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
18	UNIT50	RW	0x0	Flash 单元 50 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
17	UNIT49	RW	0x0	Flash 单元 49 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
16	UNIT48	RW	0x0	Flash 单元 48 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
15	UNIT47	RW	0x0	Flash 单元 47 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
14	UNIT46	RW	0x0	Flash 单元 46 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
13	UNIT45	RW	0x0	Flash 单元 45 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
12	UNIT44	RW	0x0	Flash 单元 44 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
11	UNIT43	RW	0x0	Flash 单元 43 写保护

位段	位段名称	属性	复位值	描述
				0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
10	UNIT42	RW	0x0	Flash 单元 42 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
9	UNIT41	RW	0x0	Flash 单元 41 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
8	UNIT40	RW	0x0	Flash 单元 40 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
7	UNIT39	RW	0x0	Flash 单元 39 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
6	UNIT38	RW	0x0	Flash 单元 38 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
5	UNIT37	RW	0x0	Flash 单元 37 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
4	UNIT36	RW	0x0	Flash 单元 36 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
3	UNIT35	RW	0x0	Flash 单元 35 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
2	UNIT34	RW	0x0	Flash 单元 34 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
1	UNIT33	RW	0x0	Flash 单元 33 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
0	UNIT32	RW	0x0	Flash 单元 32 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除

19.8.2.13 Flash 写保护寄存器 2 (FLASHWP2)

表 19-29: Flash 写保护寄存器 2 (FLASHWP2) 位段定义

FLASHWP2 (Flash Write Protect Register 2)							
Access: FLASH->FLASHWP2				Offset: 0x0034		Default: 0x00000000	
31 UNIT95	30 UNIT94	29 UNIT93	28 UNIT92	27 UNIT91	26 UNIT90	25 UNIT89	24 UNIT88
23 UNIT87	22 UNIT86	21 UNIT85	20 UNIT84	19 UNIT83	18 UNIT82	17 UNIT81	16 UNIT80
15 UNIT79	14 UNIT78	13 UNIT77	12 UNIT76	11 UNIT75	10 UNIT74	9 UNIT73	8 UNIT72
7 UNIT71	6 UNIT70	5 UNIT69	4 UNIT68	3 UNIT67	2 UNIT66	1 UNIT65	0 UNIT64

表 19-30: Flash 写保护寄存器 2 (FLASHWP2) 位段描述

位段	位段名称	属性	复位值	描述
31	UNIT95	RW	0x0	Flash 单元 95 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
30	UNIT94	RW	0x0	Flash 单元 94 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
29	UNIT93	RW	0x0	Flash 单元 93 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
28	UNIT92	RW	0x0	Flash 单元 92 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
27	UNIT91	RW	0x0	Flash 单元 91 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
26	UNIT90	RW	0x0	Flash 单元 90 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
25	UNIT89	RW	0x0	Flash 单元 89 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
24	UNIT88	RW	0x0	Flash 单元 88 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
23	UNIT87	RW	0x0	Flash 单元 87 写保护

位段	位段名称	属性	复位值	描述
				0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
22	UNIT86	RW	0x0	Flash 单元 86 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
21	UNIT85	RW	0x0	Flash 单元 85 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
20	UNIT84	RW	0x0	Flash 单元 84 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
19	UNIT83	RW	0x0	Flash 单元 83 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
18	UNIT82	RW	0x0	Flash 单元 82 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
17	UNIT81	RW	0x0	Flash 单元 81 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
16	UNIT80	RW	0x0	Flash 单元 80 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
15	UNIT79	RW	0x0	Flash 单元 79 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
14	UNIT78	RW	0x0	Flash 单元 78 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
13	UNIT77	RW	0x0	Flash 单元 77 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
12	UNIT76	RW	0x0	Flash 单元 76 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
11	UNIT75	RW	0x0	Flash 单元 75 写保护

位段	位段名称	属性	复位值	描述
				0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
10	UNIT74	RW	0x0	Flash 单元 74 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
9	UNIT73	RW	0x0	Flash 单元 73 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
8	UNIT72	RW	0x0	Flash 单元 72 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
7	UNIT71	RW	0x0	Flash 单元 71 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
6	UNIT70	RW	0x0	Flash 单元 70 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
5	UNIT69	RW	0x0	Flash 单元 69 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
4	UNIT68	RW	0x0	Flash 单元 68 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
3	UNIT67	RW	0x0	Flash 单元 67 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
2	UNIT66	RW	0x0	Flash 单元 66 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
1	UNIT65	RW	0x0	Flash 单元 65 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
0	UNIT64	RW	0x0	Flash 单元 64 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除

19.8.2.14 Flash 写保护寄存器 3 (FLASHWP3)

表 19-31: Flash 写保护寄存器 3 (FLASHWP3) 位段定义

FLASHWP3 (Flash Write Protect Register 3)							
Access: FLASH->FLASHWP3				Offset: 0x0038		Default: 0x00000000	
31 UNIT127	30 UNIT126	29 UNIT125	28 UNIT124	27 UNIT123	26 UNIT122	25 UNIT121	24 UNIT120
23 UNIT119	22 UNIT118	21 UNIT117	20 UNIT116	19 UNIT115	18 UNIT114	17 UNIT113	16 UNIT112
15 UNIT111	14 UNIT110	13 UNIT109	12 UNIT108	11 UNIT107	10 UNIT106	9 UNIT105	8 UNIT104
7 UNIT103	6 UNIT102	5 UNIT101	4 UNIT100	3 UNIT99	2 UNIT98	1 UNIT97	0 UNIT96

表 19-32: Flash 写保护寄存器 3 (FLASHWP3) 位段描述

位段	位段名称	属性	复位值	描述
31	UNIT127	RW	0x0	Flash 单元 127 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
30	UNIT126	RW	0x0	Flash 单元 126 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
29	UNIT125	RW	0x0	Flash 单元 125 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
28	UNIT124	RW	0x0	Flash 单元 124 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
27	UNIT123	RW	0x0	Flash 单元 123 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
26	UNIT122	RW	0x0	Flash 单元 122 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
25	UNIT121	RW	0x0	Flash 单元 121 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
24	UNIT120	RW	0x0	Flash 单元 120 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
23	UNIT119	RW	0x0	Flash 单元 119 写保护

位段	位段名称	属性	复位值	描述
				0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
22	UNIT118	RW	0x0	Flash 单元 118 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
21	UNIT117	RW	0x0	Flash 单元 117 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
20	UNIT116	RW	0x0	Flash 单元 116 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
19	UNIT115	RW	0x0	Flash 单元 115 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
18	UNIT114	RW	0x0	Flash 单元 114 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
17	UNIT113	RW	0x0	Flash 单元 113 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
16	UNIT112	RW	0x0	Flash 单元 112 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
15	UNIT111	RW	0x0	Flash 单元 111 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
14	UNIT110	RW	0x0	Flash 单元 110 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
13	UNIT109	RW	0x0	Flash 单元 109 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
12	UNIT108	RW	0x0	Flash 单元 108 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
11	UNIT107	RW	0x0	Flash 单元 107 写保护

位段	位段名称	属性	复位值	描述
				0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
10	UNIT106	RW	0x0	Flash 单元 106 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
9	UNIT105	RW	0x0	Flash 单元 105 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
8	UNIT104	RW	0x0	Flash 单元 104 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
7	UNIT103	RW	0x0	Flash 单元 103 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
6	UNIT102	RW	0x0	Flash 单元 102 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
5	UNIT101	RW	0x0	Flash 单元 101 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
4	UNIT100	RW	0x0	Flash 单元 100 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
3	UNIT99	RW	0x0	Flash 单元 99 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
2	UNIT98	RW	0x0	Flash 单元 98 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
1	UNIT97	RW	0x0	Flash 单元 97 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除
0	UNIT96	RW	0x0	Flash 单元 96 写保护 0: 可以被编程或者擦除 1: 不可以被编程或者扇区擦除

19.8.2.15 Flash 控制器写使能寄存器（FLASHREGKEY）

表 19-33: Flash 控制器写使能寄存器（FLASHREGKEY）位段定义

FLASHREGKEY (Flash Register Write-Allow Key Register)							
Access: FLASH->FLASHREGKEY Offset: 0x003C Default: 0x1ACCE551							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

表 19-34: Flash 控制器写使能寄存器（FLASHREGKEY）位段描述

位段	位段名称	属性	复位值	描述
31:0	KEY	RW	0x1ACCE551	写入 0x1ACCE551 解锁受保护的 Flash 寄存器

20 系统控制模块

20.1 概述

系统控制模块提供：

- 64 位产品识别号
- 64 位唯一设备识别号
- 存储器错误中断控制寄存器
- 存储器 ECC 使能寄存器
- 存储器错误地址
- 存储器锁定状态寄存器
- 复位事件控制寄存器
- 系统引导信息

20.2 64 位产品识别号

64 位产品识别号的含义和取值如表 20-1 所示。

表 20-1：64 位产品识别号定义

寄存器位段	含义	值
CID1[31:24]	产品类别	0x44: “D” 的 ASCII 码，表示为 SPD 含预驱的控制器产品
CID1[23:20]	产品名称	1
CID1[19:16]		1
CID1[15:12]		2
CID1[11:8]		1
CID1[7:0]	温度等级	0x41: “A” 的 ASCII 码，表示结温范围是 -40 ~ +125 °C
CID0[31:24]	封装材料	0x50: “P” 的 ASCII 码，表示封装材料为塑料
CID0[23:16]	封装类型	0x49: “I” 的 ASCII 码，表示封装类型为 QFN
CID0[15:8]	引脚数目	0x26: 表示引脚数为 38
CID0[7:0]	后缀	—

20.3 64 位唯一设备识别号

设备唯一识别号非常适合用于：

- 用作序列号
- 用作秘钥以提高 Flash 存储器中的安全性
 - 例如在对 Flash 存储器编程之前，用户将其与软件加密原语和和协议相结合来使用。
- 激活安全启动过程等

64 位唯一设备识别号基于出厂时间等信息为每颗 SPD1121 分配独有标识，其格式如表 20-2 所示。

表 20-2: 64 位设备唯一识别号定义

位段	UID1		UID0	
	含义	示例	含义	示例
[31:28]	出厂时间：年份	2	出厂时间：时	1
[27:24]		0		0
[23:20]		2	出厂时间：分	1
[19:16]		5		0
[15:12]	出厂时间：月份	0	出厂时间：秒	3
[11: 8]		4		5
[7: 4]	出厂时间：日	1	检验编号	0
[3: 0]		5		1

20.4 存储器错误中断控制寄存器

MEMIE、MEMIF 和 MEMIC 寄存器用来使能、标识以及清除存储器错误中断。

20.5 存储器 ECC 使能寄存器

MEMECCEN 寄存器使能 ROM 和 Flash 存储器的 ECC 特性。

20.6 存储器错误地址

ROMERRADDR、FLASHERRADDR 分别记录了 ROM、Flash 闪存发生 ECC 错误时对应的访问地址。RAM0ERRADDR、RAM1ERRADDR 分别记录了 RAM0、RAM1 发生偶校验错误时对应的访问地址。

20.7 存储器锁定状态寄存器

MEMLOCKSTS 寄存器记录了 Flash USEROTP 和 USERINFO 扇区的锁定状态。

20.8 复位事件控制寄存器

RSTEV TEN、RSTEV TFLG、RSTEV TCLR 寄存器分别用来使能、标识和清除复位事件。

20.9 系统引导信息

BOOTINFO 寄存器提供了自上电后的引导次数、引导配置引脚（GPIO40）当前电平等信息。

20.10 寄存器

20.10.1 系统寄存器列表

表 20-3: SYSTEM 模块基地址

外设模块	基地址
SYSTEM	0x4000 0000

表 20-4: SYSTEM 寄存器列表

寄存器	偏移地址	描述	复位值
CID0	0x0000	产品识别号寄存器 0	0x00000000
CID1	0x0004	产品识别号寄存器 1	0x00000000
UID0	0x0008	唯一设备识别号寄存器 0	0x00000000
UID1	0x000C	唯一设备识别号寄存器 1	0x00000000
RND0	0x0010	随机数寄存器 0	0x00000000
RND1	0x0014	随机数寄存器 1	0x00000000
REV0	0x0018	版本信息寄存器 0	0x5059A0A0
REV1	0x001C	版本信息寄存器 1	0x202403A0
MEMIF	0x0020	存储器错误中断标志寄存器	0x00000000
MEMIC	0x0024	存储器错误中断清除寄存器	0x00000000
MEMIE*	0x0028	存储器错误中断使能寄存器	0x00000000
MEMECCEN*	0x002C	存储器 ECC 使能寄存器	0x00000003
MEMLOCKSTS	0x0030	存储器锁定状态寄存器	0x00000000
RSTEVTF LG	0x0034	复位事件状态寄存器	0x00000000
RSTEVTF CLR	0x0038	复位事件清除寄存器	0x00000000
RSTEVTF EN*	0x003C	复位事件使能寄存器	0x00000001
SYSREGKEY	0x0044	SYSTEM 模块写使能寄存器	0x1ACCE551
ROMERRADDR	0x0048	ROM 读取 ECC 错误地址寄存器	0xFFFFFFFF
FLASHERRADDR	0x004C	Flash 读取 ECC 错误地址寄存器	0xFFFFFFFF
RAM0ERRADDR	0x0050	RAM0 读取奇偶校验错误地址寄存器	0xFFFFFFFF
RAM1ERRADDR	0x0054	RAM1 读取奇偶校验错误地址寄存器	0xFFFFFFFF
NVREG0	0x0058	非易失寄存器 0	0x00000000
NVREG1	0x005C	非易失寄存器 1	0x00000000
BOOTINFO	0x0060	启动信息寄存器	0x80000000

注意： 由*标识的寄存器仅当 SYSREGKEY=0x1ACCE551 时才可以改写。

20.10.2 系统寄存器

20.10.2.1 产品识别号寄存器 0 (CID0)

表 20-5: 产品识别号寄存器 0 (CID0) 位段定义

CID0 (Chip ID Register 0) Access: SYSTEM->CID0 Offset: 0x0000 Default: 0x00000000							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 20-6: 产品识别号寄存器 0 (CID0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0x0	64 位产品识别号的低 32 位, 详见表 20-1

20.10.2.2 产品识别号寄存器 1 (CID1)

表 20-7: 产品识别号寄存器 1 (CID1) 位段定义

CID1 (Chip ID Register 1) Access: SYSTEM->CID1 Offset: 0x0004 Default: 0x00000000							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 20-8: 产品识别号寄存器 1 (CID1) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0x0	64 位产品识别号的高 32 位, 详见表 20-1

20.10.2.3 唯一设备识别号寄存器 0 (UID0)

表 20-9: 唯一设备识别号寄存器 0 (UID0) 位段定义

UID0 (Unique ID Register 0)							
Access: SYSTEM->UID0		Offset: 0x0008		Default: 0x00000000			
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 20-10: 唯一设备识别号寄存器 0 (UID0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0x0	64 位唯一设备识别号的低 32 位, 详见表 20-2

20.10.2.4 唯一设备识别号寄存器 1 (UID1)

表 20-11: 唯一设备识别号寄存器 1 (UID1) 位段定义

UID1 (Unique ID Register 1)							
Access: SYSTEM->UID1		Offset: 0x000C		Default: 0x00000000			
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 20-12: 唯一设备识别号寄存器 1 (UID1) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0x0	64 位唯一设备识别号的高 32 位, 详见表 20-2

20.10.2.5 随机数寄存器 0 (RND0)

表 20-13: 随机数寄存器 0 (RND0) 位段定义

RND0 (Random Number Register 0)							
Access: SYSTEM->RND0		Offset: 0x0010		Default: 0x00000000			
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 20-14: 随机数寄存器 0 (RND0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0x0	随机数 0

20.10.2.6 随机数寄存器 1 (RND1)

表 20-15: 随机数寄存器 1 (RND1) 位段定义

RND1 (Random Number Register 1)							
Access: SYSTEM->RND1		Offset: 0x0014		Default: 0x00000000			
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 20-16: 随机数寄存器 1 (RND1) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0x0	随机数 1

20.10.2.7 版本信息寄存器 0 (REV0)

表 20-17: 版本信息寄存器 0 (REV0) 位段定义

REV0 (Revision Information Register 0)							
Access: SYSTEM->REV0		Offset: 0x0018		Default: 0x5059A0A0			
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 20-18: 版本信息寄存器 0 (REV0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0x5059A0A0	64 位版本号的低 32 位

20.10.2.8 版本信息寄存器 1 (REV1)

表 20-19: 版本信息寄存器 1 (REV1) 位段定义

REV1 (Revision Information Register 1)							
Access: SYSTEM->REV1		Offset: 0x001C		Default: 0x202403A0			
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 20-20: 版本信息寄存器 1 (REV1) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0x202403A0	64 位版本号的高 32 位

20.10.2.9 存储器错误中断标志寄存器 (MEMIF)

表 20-21: 存储器错误中断标志寄存器 (MEMIF) 位段定义

MEMIF (Memory Error Interrupt Flag Register)							
Access: SYSTEM->MEMIF		Offset: 0x0020		Default: 0x00000000			
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
MEMINT	RESERVED	RAM1PE	RAMOPE	FLASHMBE	FLASHSBE	ROMMBE	ROMSBE

表 20-22: 存储器错误中断标志寄存器 (MEMIF) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	MEMINT	RO	0x0	存储器错误全局中断标志 0: 中断未发生 1: 中断已发生, 本位段被清零前不会产生新的中断
6	RESERVED_6	RO	0x0	保留
5	RAM1PE	RO	0x0	RAM1 读取时奇偶校验错误中断标志 0: 中断未发生 1: 中断已发生, 本位段被清零前不会产生新的中断
4	RAMOPE	RO	0x0	RAM0 读取时奇偶校验错误中断标志 0: 中断未发生 1: 中断已发生, 本位段被清零前不会产生新的中断
3	FLASHMBE	RO	0x0	FLASH 读取时 ECC 多比特错误中断标志 0: 中断未发生 1: 中断已发生, 本位段被清零前不会产生新的中断
2	FLASHSBE	RO	0x0	FLASH 读取时 ECC 单比特错误中断标志 0: 中断未发生 1: 中断已发生, 本位段被清零前不会产生新的中断
1	ROMMBE	RO	0x0	ROM 读取时 ECC 多比特错误中断标志 0: 中断未发生

位段	位段名	属性	复位值	描述
				1: 中断已发生，本位段被清零前不会产生新的中断
0	ROMSBE	RO	0x0	ROM 读取时 ECC 单比特错误中断标志 0: 中断未发生 1: 中断已发生，本位段被清零前不会产生新的中断

20.10.2.10 存储器错误中断清除寄存器 (MEMIC)

表 20-23: 存储器错误中断清除寄存器 (MEMIC) 位段定义

MEMIC (Memory Error Interrupt Clear Register)							
Access: SYSTEM->MEMIC Offset: 0x0024 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
MEMINT	RESERVED	RAM1PE	RAMOPE	FLASHMBE	FLASHSBE	ROMMBE	ROMSBE

表 20-24: 存储器错误中断清除寄存器 (MEMIC) 位段描述

位段	位段名	属性	复位值	描述
31:8	RESERVED_31_8	RO	0x0	保留
7	MEMINT	W1C	0x0	全局存储器错误中断清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除中断标志, 本位段自动清零
6	RESERVED_6	RO	0x0	保留
5	RAM1PE	W1C	0x0	RAM1 读取时奇偶校验错误中断清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除中断标志, 本位段自动清零
4	RAMOPE	W1C	0x0	RAM0 读取时奇偶校验错误中断清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除中断标志, 本位段自动清零
3	FLASHMBE	W1C	0x0	FLASH 读取时 ECC 多比特错误中断清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除中断标志位, 本位段自动清零
2	FLASHSBE	W1C	0x0	FLASH 读取时 ECC 单比特错误中断清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除中断标志位, 本位段自动清零
1	ROMMBE	W1C	0x0	ROM 读取时 ECC 多比特错误中断清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除中断标志位, 本位段自动清零
0	ROMSBE	W1C	0x0	ROM 读取时 ECC 单比特错误中断清除 0: 写 0 无效, 总是读回 0 1: 写 1 清除中断标志位, 本位段自动清零

20.10.2.11 存储器错误中断使能寄存器 (MEMIE)

表 20-25: 存储器错误中断使能寄存器 (MEMIE) 位段定义

MEMIE (Memory Error Interrupt Enable Register)							
Access: SYSTEM->MEMIE Offset: 0x0028 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED		RAM1PE	RAMOPE	FLASHMBE	FLASHSBE	ROMMBE	ROMSBE

表 20-26: 存储器错误中断使能寄存器 (MEMIE) 位段描述

位段	位段名	属性	复位值	描述
31:6	RESERVED_31_6	RO	0x0	保留
5	RAM1PE	RW	0x0	RAM1 读取时奇偶校验错误中断使能 0: 关闭 1: 使能
4	RAMOPE	RW	0x0	RAM0 读取时奇偶校验错误中断使能 0: 关闭 1: 使能
3	FLASHMBE	RW	0x0	Flash 读取时 ECC 多比特错误中断使能 0: 关闭 1: 使能
2	FLASHSBE	RW	0x0	Flash 读取时 ECC 单比特错误中断使能 0: 关闭 1: 使能
1	ROMMBE	RW	0x0	ROM 读取时 ECC 多比特错误中断使能 0: 关闭 1: 使能
0	ROMSBE	RW	0x0	ROM 读取时 ECC 单比特错误中断使能 0: 关闭 1: 使能

20.10.2.12 存储器 ECC 使能寄存器 (MEMECCEN)**表 20-27: 存储器 ECC 使能寄存器 (MEMECCEN) 位段定义**

MEMECCEN (Memory ECC Enable Register)							
Access: SYSTEM->MEMECCEN Offset: 0x002C Default: 0x00000003							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED						FLASHECC	ROMECC

表 20-28: 存储器 ECC 使能寄存器 (MEMECCEN) 位段描述

位段	位段名	属性	复位值	描述
31:2	RESERVED_31_2	RO	0x0	保留
1	FLASHECC	RW	0x1	FLASH ECC 使能 0: 关闭 1: 使能
0	ROMECC	RW	0x1	ROM ECC 使能 0: 关闭 1: 使能

20.10.2.13 存储器锁定状态寄存器（MEMLOCKSTS）

表 20-29：存储器锁定状态寄存器（MEMLOCKSTS）位段定义

MEMLOCKSTS (Memory Lock Status Register)							
Access: SYSTEM->MEMLOCKSTS				Offset: 0x0030		Default: 0x00000000	
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							USEROTP
7	6	5	4	3	2	1	0
RESERVED			USERINFO	RESERVED			

表 20-30：存储器锁定状态寄存器（MEMLOCKSTS）位段描述

位段	位段名	属性	复位值	描述
31:9	RESERVED_31_9	RO	0x0	保留
8	USEROTP	RO	0x0	锁定 Flash USEROTP 扇区 0: 可读、写、擦除 USEROTP 扇区 1: USEROTP 扇区只读
7:5	RESERVED_7_5	RO	0x0	保留
4	USERINFO	RO	0x0	锁定 Flash USERINFO 扇区 当用户擦除 USERINFO 扇区时，整个 Flash 主存储区也会被擦除以保护代码。 0: 可读、写、擦除 USERINFO 扇区 1: USERINFO 扇区只读
3:0	RESERVED_3_0	RO	0x0	保留

20.10.2.14 复位事件状态寄存器（RSTEVTFLG）

表 20-31: 复位事件状态寄存器（RSTEVTFLG）位段定义

RSTEVTFLG (Reset Event Flag Register)							
Access: SYSTEM->RSTEVTFLG Offset: 0x0034 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
PINRST	SYSRESET	WDT1RST	WDTORST	CLKDETERR	PLLUNLOCK	VDD33OV	VDD33UV
7	6	5	4	3	2	1	0
VDD12OV	VDD12UV1	VDD12UV0	RESERVED	RAM1PE	RAMOPE	FLASHMBE	ROMMBE

表 20-32: 复位事件状态寄存器（RSTEVTFLG）位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15	PINRST	RO	0x0	XRSTn 引脚触发复位标志 0: 事件未发生 1: 事件发生
14	SYSRESET	RO	0x0	SYSRESET 触发复位标志 0: 事件未发生 1: 事件发生
13	WDT1RST	RO	0x0	WDT1 复位事件触发的复位状态 0: 事件未发生 1: 事件发生
12	WDTORST	RO	0x0	WDT0 复位事件触发的复位状态 0: 事件未发生 1: 事件发生
11	CLKDETERR	RO	0x0	时钟交叉检测错误复位事件标志 0: 事件未发生 1: 事件发生
10	PLLUNLOCK	RO	0x0	PLL 失锁错误事件触发的复位状态 0: 事件未发生 1: 事件发生
9	VDD33OV	RO	0x0	3.3V 过压事件触发的复位状态 0: 事件未发生 1: 事件发生
8	VDD33UV	RO	0x0	3.3V 欠压事件触发的复位状态 0: 事件未发生 1: 事件发生

位段	位段名	属性	复位值	描述
7	VDD12OV	RO	0x0	1.2V 过压事件触发的复位状态 0: 事件未发生 1: 事件发生
6	VDD12UV1	RO	0x0	1.2V 欠压事件 1 触发的复位状态 0: 事件未发生 1: 事件发生
5	VDD12UV0	RO	0x0	1.2V 欠压事件 0 触发的复位状态 0: 事件未发生 1: 事件发生
4	RESERVED_4	RO	0x0	保留
3	RAM1PE	RO	0x0	RAM1 读取时奇偶校验错误触发复位标志 0: 事件未发生 1: 事件发生
2	RAM0PE	RO	0x0	RAM0 读取时奇偶校验错误触发复位标志 0: 事件未发生 1: 事件发生
1	FLASHMBE	RO	0x0	FLASH 读取时 ECC 多比特错误触发的复位状态 0: 事件未发生 1: 事件发生
0	ROMMBE	RO	0x0	ROM 读取时 ECC 多比特错误触发的复位状态 0: 事件未发生 1: 事件发生

20.10.2.15 复位事件状态清除寄存器（RSTEVTCCLR）

表 20-33: 复位事件状态清除寄存器（RSTEVTCCLR）位段定义

RSTEVTCCLR (Reset Event Status Clear Register)							
Access: SYSTEM->RSTEVTCCLR Offset: 0x0038 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
PINRST	SYSRESET	WDT1RST	WDTORST	CLKDETERR	PLLUNLOCK	VDD33OV	VDD33UV
7	6	5	4	3	2	1	0
VDD12OV	VDD12UV1	VDD12UV0	RESERVED	RAM1PE	RAMOPE	FLASHMBE	ROMMBE

表 20-34: 复位事件状态清除寄存器（RSTEVTCCLR）位段描述

位段	位段名	属性	复位值	描述
31:16	RESERVED_31_16	RO	0x0	保留
15	PINRST	W1C	0x0	XRSTn 引脚复位事件清除 0: 写 0 无效, 读总是 0 1: 写 1 清除复位事件标志, 本位段自动清零
14	SYSRESET	W1C	0x0	SYSRESET 复位事件清除 0: 写 0 无效, 读总是 0 1: 写 1 清除复位事件标志, 本位段自动清零
13	WDT1RST	W1C	0x0	WDT1 复位事件清除 0: 写 0 无效, 读总是 0 1: 写 1 清除复位事件标志, 本位段自动清零
12	WDTORST	W1C	0x0	WDT0 复位事件清除 0: 写 0 无效, 读总是 0 1: 写 1 清除复位事件标志, 本位段自动清零
11	CLKDETERR	W1C	0x0	时钟交叉检测错误复位事件清除 0: 写 0 无效, 读总是 0 1: 写 1 清除复位事件标志, 本位段自动清零
10	PLLUNLOCK	W1C	0x0	PLL 失锁复位事件清除 0: 写 0 无效, 读总是 0 1: 写 1 清除复位事件标志, 本位段自动清零
9	VDD33OV	W1C	0x0	3.3V 过压复位事件清除 0: 写 0 无效, 读总是 0 1: 写 1 清除复位事件标志, 本位段自动清零
8	VDD33UV	W1C	0x0	3.3V 欠压复位事件清除 0: 写 0 无效, 读总是 0 1: 写 1 清除复位事件标志, 本位段自动清零

位段	位段名	属性	复位值	描述
7	VDD12OV	W1C	0x0	1.2V 过压复位事件清除 0: 写 0 无效, 读总是 0 1: 写 1 清除复位事件标志, 本位段自动清零
6	VDD12UV1	W1C	0x0	1.2V 欠压事件 1 触发的复位状态清除 0: 写 0 无效, 读总是 0 1: 写 1 清除复位事件标志, 本位段自动清零
5	VDD12UV0	W1C	0x0	1.2V 欠压事件 0 触发的复位状态清除 0: 写 0 无效, 读总是 0 1: 写 1 清除复位事件标志, 本位段自动清零
4	RESERVED_4	RO	0x0	保留
3	RAM1PE	W1C	0x0	RAM1 读取时奇偶校验错误复位事件清除 0: 写 0 无效, 读总是 0 1: 写 1 清除复位事件标志, 本位段自动清零
2	RAM0PE	W1C	0x0	RAM0 读取时奇偶校验错误复位事件清除 0: 写 0 无效, 读总是 0 1: 写 1 清除复位事件标志, 本位段自动清零
1	FLASHMBE	W1C	0x0	FLASH 读取时 ECC 多比特错误复位事件清除 0: 写 0 无效, 读总是 0 1: 写 1 清除复位事件标志, 本位段自动清零
0	ROMMBE	W1C	0x0	ROM 读取时 ECC 多比特错误复位事件清除 0: 写 0 无效, 读总是 0 1: 写 1 清除复位事件标志, 本位段自动清零

20.10.2.16 复位事件使能寄存器 (RSTEV TEN)

表 20-35: 复位事件使能寄存器 (RSTEV TEN) 位段定义

RSTEV TEN (Reset Event Enable Register)							
Access: SYSTEM->RSTEV TEN				Offset: 0x003C		Default: 0x00000001	
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED				CLKDETERR	PLLUNLOCK	VDD33OV	VDD33UV
7	6	5	4	3	2	1	0
VDD12OV	VDD12UV1	VDD12UV0	RESERVED	RAM1PE	RAMOPE	FLASHMBE	ROMMBE

表 20-36: 复位事件使能寄存器 (RSTEV TEN) 位段描述

位段	位段名	属性	复位值	描述
31:12	RESERVED_31_12	RO	0x0	保留
11	CLKDETERR	RW	0x0	时钟交叉检测错误复位事件使能 0: 关闭 1: 使能
10	PLLUNLOCK	RW	0x0	PLL 失锁复位事件使能 0: 关闭 1: 使能
9	VDD33OV	RW	0x0	3.3V 电源过压复位事件使能 0: 关闭 1: 使能
8	VDD33UV	RW	0x0	3.3V 电源欠压复位事件使能 0: 关闭 1: 使能
7	VDD12OV	RW	0x0	1.2V 电源过压复位事件使能 0: 关闭 1: 使能
6	VDD12UV1	RW	0x0	1.2V 欠压事件 1 触发复位使能 0: 关闭 1: 使能
5	VDD12UV0	RW	0x0	1.2V 欠压事件 0 触发复位使能 0: 关闭 1: 使能
4	RESERVED_4	RO	0x0	保留

位段	位段名	属性	复位值	描述
3	RAM1PE	RW	0x0	RAM1 读取时 ECC 多比特错误复位事件使能 0: 关闭 1: 使能
2	RAM0PE	RW	0x0	RAM0 读取时 ECC 多比特错误复位事件使能 0: 关闭 1: 使能
1	FLASHMBE	RW	0x0	Flash 读取时 ECC 多比特错误复位事件使能 0: 关闭 1: 使能
0	ROMMBE	RW	0x1	ROM 读取时 ECC 多比特错误复位事件使能 0: 关闭 1: 使能

20.10.2.17 SYSTEM 模块写使能寄存器（SYSREGKEY）

表 20-37: SYSTEM 模块写使能寄存器（SYSREGKEY）位段定义

SYSREGKEY (System Register Write-Allow Key Register)							
Access: SYSTEM->SYSREGKEY Offset: 0x0044 Default: 0x1ACCE551							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

表 20-38: SYSTEM 模块写使能寄存器（SYSREGKEY）位段描述

位段	位段名	属性	复位值	描述
31:0	KEY	RW	0x1ACCE551	写入 0x1ACCE551 解锁受保护的 SYSTEM 寄存器

20.10.2.18 ROM 读取 ECC 错误地址寄存器 (ROMERRADDR)

表 20-39: ROM 读取 ECC 错误地址寄存器 (ROMERRADDR) 位段定义

ROMERRADDR (ROM Read ECC Error Address Register)							
Access: SYSTEM->ROMERRADDR Offset: 0x0048 Default: 0xFFFFFFFF							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 20-40: ROM 读取 ECC 错误地址寄存器 (ROMERRADDR) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0xFFFFFFFF	ROM 读取时发生 ECC 错误的地址

20.10.2.19 Flash 读取 ECC 错误地址寄存器 (FLASHERRADDR)

表 20-41: Flash 读取 ECC 错误地址寄存器 (FLASHERRADDR) 位段定义

FLASHERRADDR (Flash Read ECC Error Address Register)							
Access: SYSTEM->FLASHERRADDR Offset: 0x004C Default: 0xFFFFFFFF							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 20-42: Flash 读取 ECC 错误地址寄存器 (FLASHERRADDR) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0xFFFFFFFF	Flash 读取时发生 ECC 错误的地址

20.10.2.20 RAM0 读取奇偶校验错误地址寄存器 (RAM0ERRADDR)

表 20-43: RAM0 读取奇偶校验错误地址寄存器 (RAM0ERRADDR) 位段定义

RAM0ERRADDR (RAM0 Read Parity Error Address Register)							
Access: SYSTEM->RAM0ERRADDR Offset: 0x0050 Default: 0xFFFFFFFF							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 20-44: RAM0 读取奇偶校验错误地址寄存器 (RAM0ERRADDR) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0xFFFFFFFF	RAM0 读取时发生奇偶校验错误的地址

20.10.2.21 读取奇偶校验错误地址寄存器 (RAM1ERRADDR)

表 20-45: 读取奇偶校验错误地址寄存器 (RAM1ERRADDR) 位段定义

RAM1ERRADDR (RAM1 Read Parity Error Address Register)							
Access: SYSTEM->RAM1ERRADDR Offset: 0x0054 Default: 0xFFFFFFFF							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 20-46: 读取奇偶校验错误地址寄存器 (RAM1ERRADDR) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RO	0xFFFFFFFF	RAM1 读取时发生奇偶校验错误的地址

20.10.2.22 非易失寄存器 0 (NVREG0)

表 20-47: 非易失寄存器 0 (NVREG0) 位段定义

NVREG0 (Non-Volatile Register 0)							
Access: SYSTEM->NVREG0 Offset: 0x0058 Default: 0x00000000							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 20-48: 非易失寄存器 0 (NVREG0) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	该寄存器中的值只能通过断电重启重置

20.10.2.23 非易失寄存器 1 (NVREG1)

表 20-49: 非易失寄存器 1 (NVREG1) 位段定义

NVREG1 (Non-Volatile Register 1)							
Access: SYSTEM->NVREG1 Offset: 0x005C Default: 0x00000000							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

表 20-50: 非易失寄存器 1 (NVREG1) 位段描述

位段	位段名	属性	复位值	描述
31:0	VAL	RW	0x0	该寄存器中的值只能通过断电重启重置

20.10.2.24 启动信息寄存器（BOOTINFO）

表 20-51：启动信息寄存器（BOOTINFO）位段定义

BOOTINFO （ Boot Information Register ）							
Access: SYSTEM->BOOTINFO Offset: 0x0060 Default: 0x80000000							
31	30	29	28	27	26	25	24
BOOTPINLVL	RESERVED						
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
BOOTCNT							
7	6	5	4	3	2	1	0
BOOTCNT							

表 20-52：启动信息寄存器（BOOTINFO）位段描述

位段	位段名	属性	复位值	描述
31	BOOTPINLVL	RO	0x1	启动引脚 GPIO40 输入电平
30:16	RESERVED_30_16	RO	0x0	保留
15:0	BOOTCNT	RW	0x0	上电后引导的次数 每当重新上电时本位段复位成 0，之后每运行一次引导程序（例如：由复位引起），会导致该计数值加 1。 由于重新上电后也会运行引导，所以用户能看到的本位段的最小值是 1。

21 调试行为（Debug Behavior）

21.1 看门狗定时器

当 CPU 进入 HALT 模式时，看门狗定时器（WDT）可以暂停运行或者保持正常运行。可以通过 WDTCTL 寄存器的 HALTEDRUN 位段来控制 WDT 的调试功能，如表 21-1 所示。

表 21-1: CPU HALT 模式下 WDT 调试行为

WDTCTL.HALTEDRUN	描述
0	当 CPU 进入 HALT 模式，WDT 计数器立即停止并被复位到 WDTLOAD。
1	当 CPU 进入 HALT 模式，WDT 计数器保持正常计数

当 CPU 进入 LOCKUP 模式时，看门狗定时器（WDT）可以暂停运行或者保持正常运行。可以通过 WDTCTL 寄存器的 LOCKUPRUN 位段来控制 WDT 的调试功能，如表 21-2 所示。

表 21-2: CPU LOCKUP 模式下 WDT 调试行为

WDTCTL.LOCKUPRUN	描述
0	当 CPU 进入 LOCKUP 模式，WDT 计数器立即停止并被复位到 WDTLOAD。
1	当 CPU 进入 LOCKUP 模式，WDT 计数器保持正常计数

21.2 通用定时器

通用定时器没有特殊的调试特性。当 CPU 进入 HALT 模式时，通用定时器的计数器保持正常计数。

21.3 PWM

当 CPU 进入 HALT 模式时，PWM 可以暂停运行或者继续保持运行。可以通过 TBCTL 寄存器的 HALTRUN 位段来控制 PWM 的调试功能，如表 21-3 所示。

表 21-3: PWM 调试行为

TBCTL.HALTRUN	描述
0	当 CPU 进入 HALT 模式，PWM 时基计数器在完成下一拍计数后停止
1	当 CPU 进入 HALT 模式，PWM 时基计数器在完成整个计数周期后停止（向上计数模式为 TBCNT=TBPRD，其余模式计数至 TBCNT=0）
2	当 CPU 进入 HALT 模式，PWM 时基计数器保持正常计数

21.4 ECAP

当 CPU 进入 HALT 模式时，ECAP 可以暂停运行或者继续保持运行。可以通过 CAPCTL 寄存器的 DBGRUN 位段来控制 ECAP 的调试功能，如表 21-4 所示。

表 21-4: ECAP 调试行为

TBCTL.DBGRUN	描述
0	当 CPU 进入 HALT 模式，ECAP 时间戳计数器立即停止
1	当 CPU 进入 HALT 模式，ECAP 时间戳计数器在计数到 0 时停止
2	当 CPU 进入 HALT 模式，ECAP 时间戳计数器保持正常计数

21.5 UART

外部调试器可以通过调试接口读 UART 模块的寄存器。需要注意的是，Debug 读操作可以产生和 CPU 读操作相同的效果。表 21-5 列出了一些寄存器的调试行为。

表 21-5: UART 调试行为

寄存器	描述
UARTBR	在非 FIFO 模式下，Debug 读该寄存器会清空 UARTBR 寄存器。在 FIFO 模式下，Debug 读该寄存器会获得 RXFIFO 最前面的数据字节，与此同时，UARTFOR 寄存器的值也会减 1。
UARTIR	Debug 读 UARTIR 寄存器会清除、或改变其部分位段。
UARTLSR	Debug 读 UARTIR 寄存器会清除其部分位段。

21.6 SPI

外部调试器可以通过调试接口读 SPI 模块的寄存器。需要注意的是，Debug 读操作可以产生和 CPU 读操作相同的效果。表 21-6 列出了一些寄存器的调试行为。

表 21-6: SPI 调试行为

寄存器	描述
SPIDATA	如果 RXFIFO 非空，Debug 读该寄存器会获得 RXFIFO 最前面的数据，与此同时，SPISTS.RFLVL 寄存器位段的值也会减 1。

21.7 I2C

外部调试器可以通过调试接口读 I2C 模块的寄存器。需要注意的是，Debug 读操作可以产生和 CPU 读操作相同的效果。表 21-6 列出了一些寄存器的调试行为。

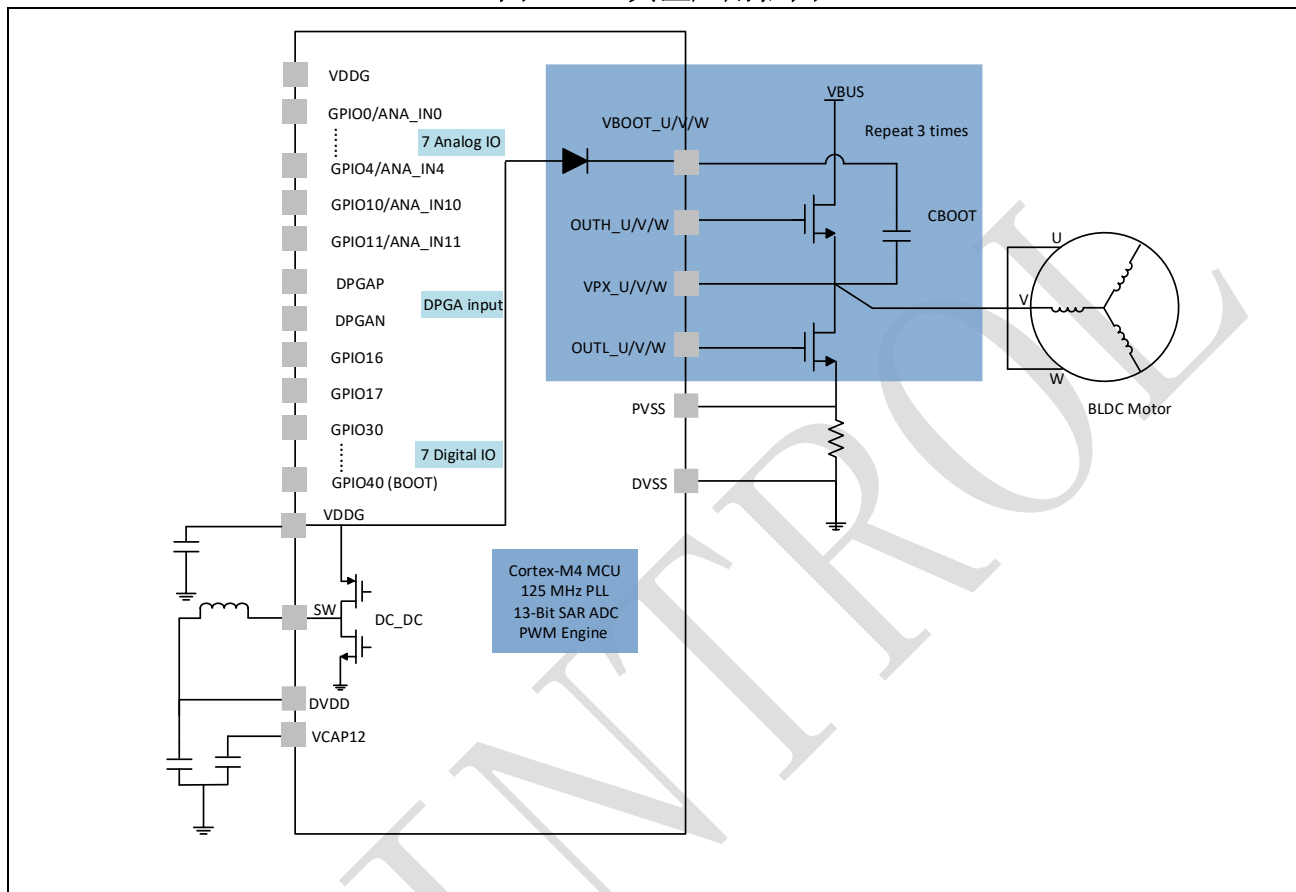
表 21-7: I2C 调试行为

寄存器	描述
I2CDATACMD	如果 RXFIFO 非空且 I2CDATACMD.CMD = 1，Debug 读该寄存器会获得 RXFIFO 最前面的数据，与此同时，I2CRFLVL 寄存器的值也会减 1。
I2CINTCLR	Debug 读该寄存器会清除所有中断标志位以及 I2CTXABRTSRC 寄存器。 注意：I2CTXABRTSRC.STARTNORESTART 只有在正确配置发送 Start Byte 功能后，才能被清除。
I2CRXUDFCLR	Debug 读该寄存器会清除 RXUDF 标志
I2CRXOVFCLR	Debug 读该寄存器会清除 RXOVF 标志
I2CTXOVFCLR	Debug 读该寄存器会清除 TXOVF 标志
I2CRDREQCLR	Debug 读该寄存器会清除 RDREQ 标志
I2CTXABRTCLR	Debug 读该寄存器会清除 TXABRT 标志和 I2CTXABRTSRC 寄存器 注意：I2CTXABRTSRC.STARTNORESTART 只有在正确配置发送 Start Byte 功能后，才能被清除。
I2CRXDONECLR	Debug 读该寄存器会清除 RXDONE 标志
I2CACTCLR	如果 I2C 不在活动时，Debug 读该寄存器会清除 ACT 标志
I2CSTOPDETCLR	Debug 读该寄存器会清除 STOPDET 标志
I2CSTARTDETCLR	Debug 读该寄存器会清除 STARTDET 标志
I2CGENCALLCLR	Debug 读该寄存器会清除 GENCALL 标志

22 降压开关电源（DC-DC）

22.1 概述

图 22-1：典型应用框图



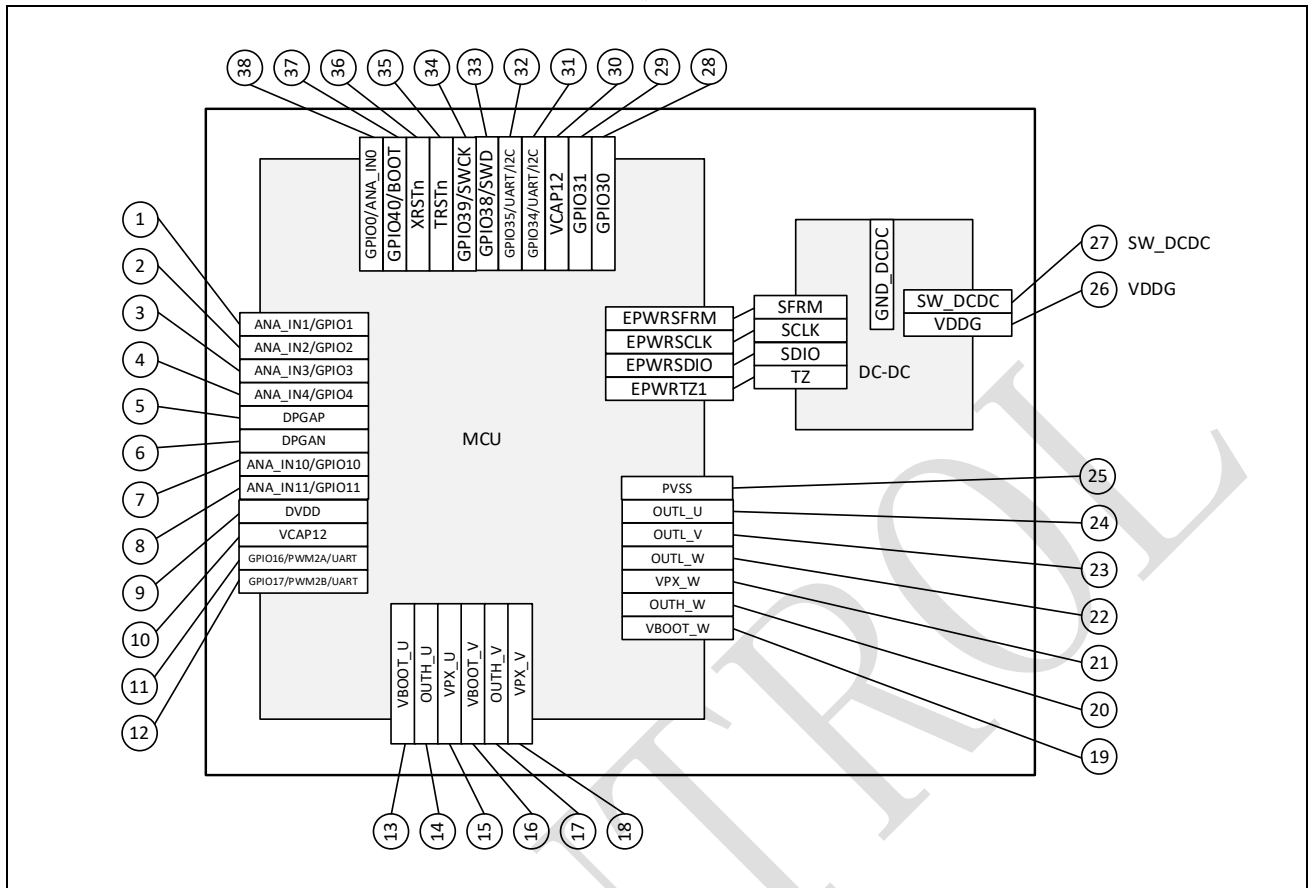
SPD1121 典型应用框图如图所示，具体无源器件参数及布局要求请参考数据手册。

DC-DC 采用 11V 至 20V 的输入电压，不仅能为 MCU 提供 3.3V 电源，而且还可以为电路板上的其他器件提供电源，只需要总电流低于 400 mA 即可。由于采用完全集成的设计，客户只需放置电感器和电容器即可使 DC-DC 工作，不需要外部功率 FET、二极管或者补偿网络。DC-DC 启动的 VDDG 电压定义为欠电压级别，触发电压为 5.4V，释放电压为 5.9V。

在 MCU 和 DC-DC 模块之间使用 Spintrol 标准的 ePower 控制接口。当 EPWRCTL.ENSPI 设置为 1 时，MCU 的 SPI 外设端口将自动连接到 EPWRSDIO（SPI 数据通道）、EPWRSFRM（SPI 帧通道）和 EPWRSCLK（SPI 时钟）引脚，MCU 就可以通过 SPI 接口与 DC-DC 模块进行通信。

QFN38 封装的 SPD1121 内部 MCU 和 DC-DC 模块部分的连接细节如图 22-2 所示。该图还提供了器件内部信号连接到引脚的信息。

图 22-2: MCU 和 DC-DC 模块连接示意图 (QFN38)



ePower 接口的映射细节如表 22-1 所示。MCU GPIO 通道的详细定义请参见表 4-1，以便正确配置 MCU 引出的 GPIO 管脚。

表 22-1: MCU 通道与 DC-DC 模块的连接关系

MCU 通道描述	DC-DC 模块信号描述
EPWRSFRM	SPI 帧（片选）信号
EPWRSCLK	SPI 时钟信号
EPWRS DIO	SPI 数据信号
EPWRTZ1	封锁（Trip-zone）信号输出

22.2 DC-DC 模块控制

MCU 通过控制 ePower 外设寄存器使能与开关稳压器的 SPI 通信接口。之后，MCU 可以通过 SPI 接口写 DC-DC 模块内部寄存器的方式来控制 DC-DC 模块的行为。

22.2.1 DC-DC 模块初始化

DC-DC 模块初始化流程如下：

- 编程 MCU ePower 寄存器以启用 DC-DC 模块控制
 - 对寄存器 EPWRREGKEY 写入 0x1ACCE551 以禁用写保护并使能 MCU 对 ePower 控制寄存器的写操作。
 - 对 ePower 寄存器 EPWRTZ1CTL 和 EPWRCTL 进行编程。尤其是要设置 EPWRCTL.ENSPI，以使能 MCU 与 DC-DC 模块之间的 SPI 通信接口。

- 对 ePower 寄存器 EPWRREGKEY 写入不等于 0x1ACCE551 的任何值，以使能写保护并禁用 MCU 对 ePower 控制寄存器的写操作。本步骤是可选操作。
- 编程 MCU SPI 外设寄存器
- 选择 SPI 作为正常的主模式，帧大小为 32 位宽，SPI_SCLK 极性和相位均为零，帧信号的极性设为低电平。在执行这些操作之前，请记住使能 SPI 时钟。
- 查询 DC-DC 模块 ID
- MCU 向 DC-DC 模块发送请求码（0x514E4944），DC-DC 模块将返回带有 DC-DC 模块 ID 的代码。在成功获取 ID 后，DC-DC 模块就准备好接受 MCU 的控制——读/写 DC-DC 寄存器。
- 寄存器初始化
- DC-DC 模块中的内部寄存器可以通过电源循环复位为默认值，但不能在发生其他复位事件时复位。为了在 MCU 的复位事件后复位 DC-DC 模块寄存器，MCU 需要发送复位请求码（0x5253544E）；寄存器复位成功后，DC-DC 将发送应答码（0x446F6E65）。

22.2.2 MCU 和 DC-DC 模块的通信格式

MCU 与 DC-DC 之间的通信格式是基于 SPI 格式的。只有一个数据引脚 EPWRSDIO 用于传输数据。

- 置位 EPWRCTL.SPIDIR: MCU 将数据传输到 DC-DC
- 清除 EPWRCTL.SPIDIR: MCU 从 DC-DC 接收数据

SPI 的通信格式如图 22-3 和图 22-4 所示。

图 22-3: MCU 向 DC-DC 发送数据 (PHA=0)

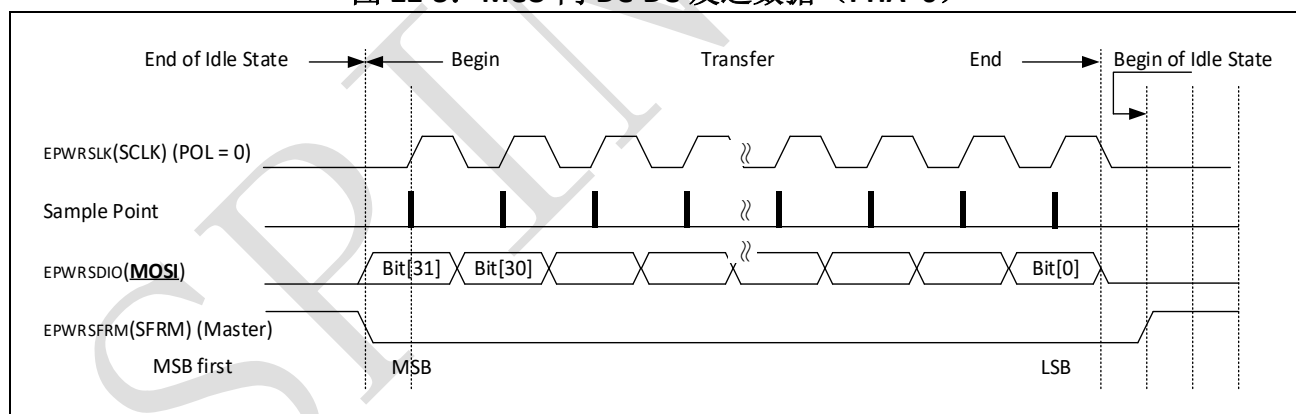
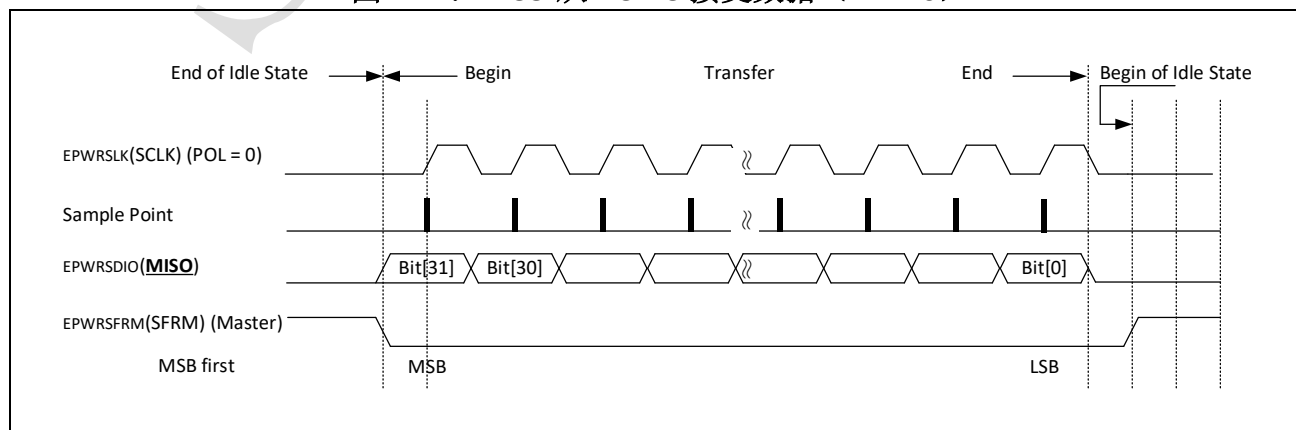


图 22-4: MCU 从 DC-DC 接受数据 (PHA=0)



22.2.3 通信协议和命令

MCU 和 DC-DC 之间的 SPI 通信有五种命令类型。对于每种类型，MCU 总是首先将数据传输到 DC-DC，然后 DC-DC 通过将数据发送回 MCU 来应答。

- 查询 DC-DC ID
此命令请求 DC-DC 将其 ID 发送到 MCU。
MCU 发送：0x514E4944。
DC-DC 应答：(0x4944<<16)+(0x56<<8)+(ID)。
DC-DC ID 是一个 8 位宽的数字。
- 寄存器写入请求
此命令仅将 TXDATA 写入 DC-DC 影子寄存器。
注：MCU 成功向 DC-DC 发送写寄存器执行请求后，TXDATA 将被写入实际的 DC-DC 寄存器。
MCU 发送：(0x57<<24)+(0x2<<21)+(ADDR<<16)+(TXDATA<<8)+0x5A。
DC-DC 应答：(0x57<<24)+(0x5<<21)+(ADDR<<16)+(TXDATA<<8)+0x5A。
ADDR 是 DC-DC 寄存器地址，5 位宽；TXDATA 是要写入 DC-DC 寄存器的数据，8 位宽。
- 寄存器执行请求
此命令请求 DC-DC 将影子寄存器中的值写入实际寄存器。
MCU 发送：0x45786563
DC-DC 应答：0x446F6E65
- 寄存器读取请求
此命令读取 DC-DC 实际寄存器值。成功接收 MCU 请求后，DC-DC 将寄存器值发送给 MCU。
MCU 发送：(0x52<<24) + (0x2<<21) + (ADDR<<16) + 0x445A
DC-DC 应答：(0x52<<24) + (0x5<<21) + (ADDR<<16) + (RXDATA<<8) + RXDATA
ADDR 是 DC-DC 寄存器地址，5 位宽；RXDATA 是寄存器值，8 位宽。
- 寄存器重置请求
此命令请求 DC-DC 将所有内部寄存器复位为默认值。
MCU 发送：0x5253544E
DC-DC 应答：0x446F6E65

22.3 DC-DC 模块特性

22.3.1 可编程的开关频率

开关频率的配置，请参见表 22-26。默认值为 1.0 MHz。

22.3.2 PWM 和 PFM 模式切换

对于 PFM 模式，电感电流的纹波可以估计为（下标为引脚名）：

$$V_{Ripple\_PFM} = \frac{T_{sw}^2 \cdot V_{DVDD}}{2 \cdot L \cdot C_{DVDD} \cdot V_{VDDG}}$$

对于 PWM 模式，电感电流的纹波可以估计为：

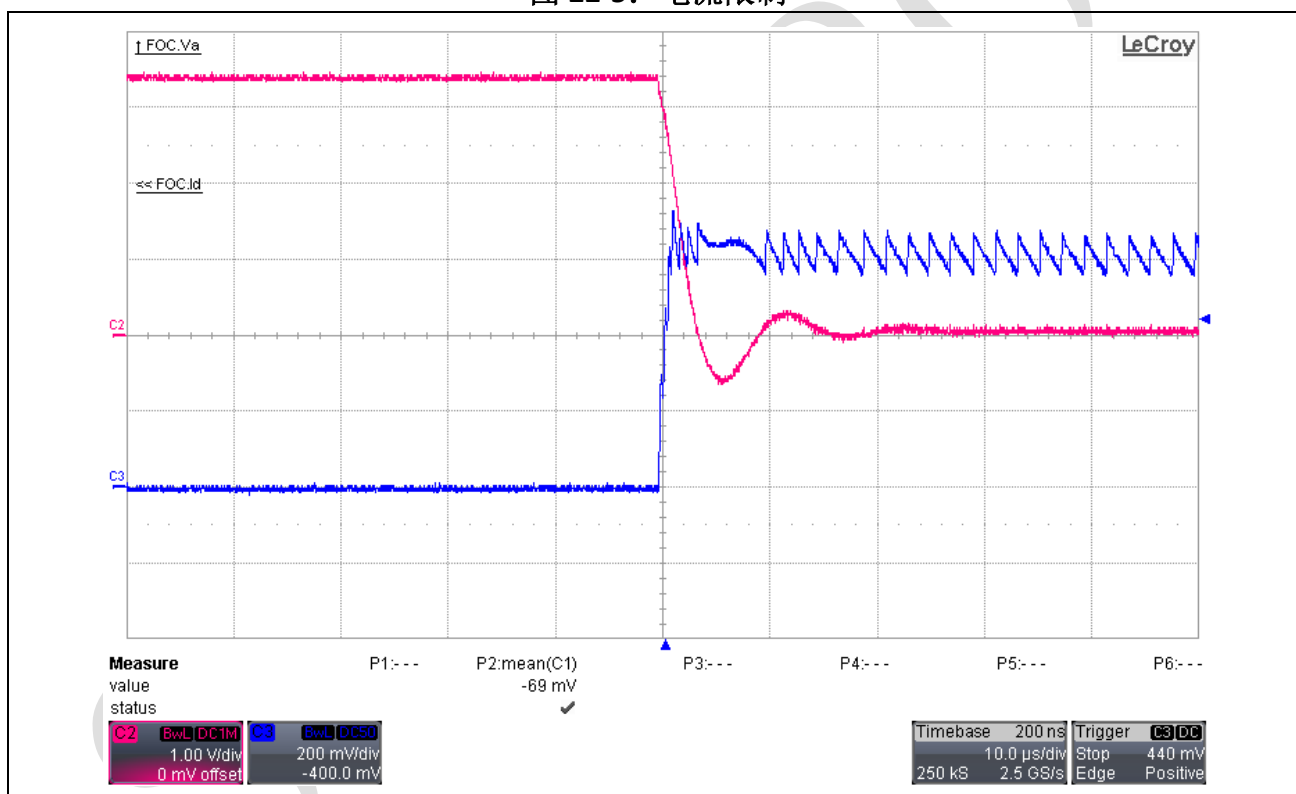
$$V_{Ripple\_PWM} = \frac{T_{sw}^2 \cdot V_{DVDD} \cdot (V_{VDDG} - V_{DVDD})}{8 \cdot L \cdot C_{DVDD} \cdot V_{VDDG}}$$

在轻负载下，DC-DC 将在 PFM 模式下运行，以节省电源，但代价是 3.3V 输出的纹波更高。对于纹波敏感和调节值敏感的应用，建议将 DC-DC 配置为强制 PWM 模式（BUCKTLO.FRCPWM=1）。在轻负载的这种模式下，稳压器将允许电流从 SW_DCDC 流向地，这会降低效率，但提供更小的输出纹波。

22.3.3 输出短路保护

DC-DC 的输出电流限制为小于 400 mA。增加此功能是为了安全，以防止输出接地短路时产生大电流，这将导致芯片和电路板的可靠性问题。如果将大于约 450 mA 的负载电流施加到 DCDC 输出，DC-DC 将降低其 3.3V 输出，并作为约 450 mA 电流源运行。图 22-5 显示了 3.3V 输出突然短路到地时的电感电流（蓝色曲线）和 VDDC 电压（红色曲线）行为。

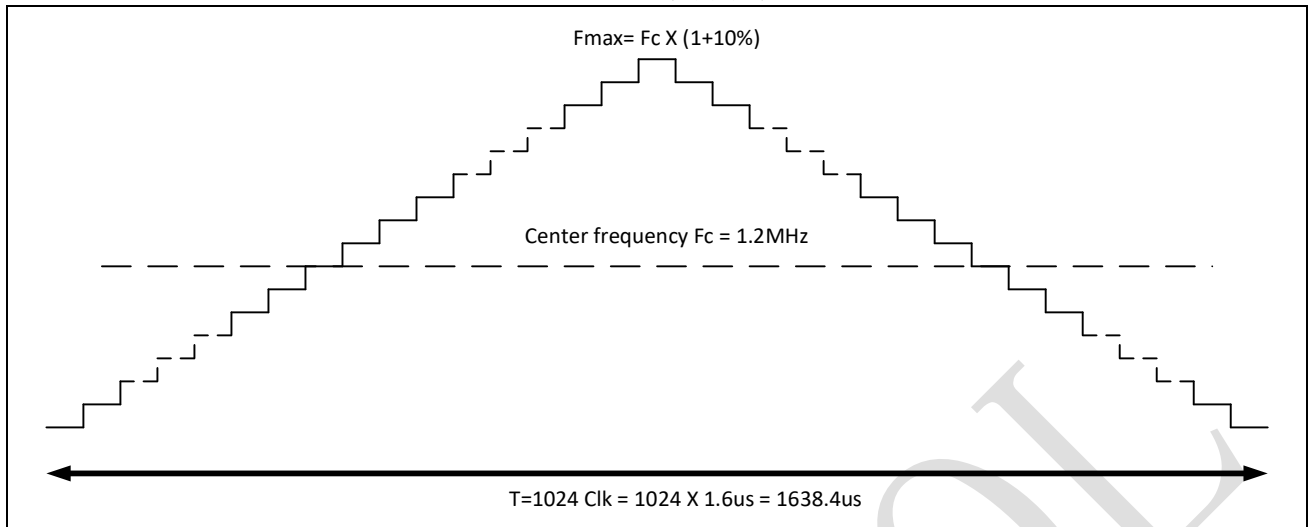
图 22-5：电流限制



22.3.4 频率抖动选项

为了减少 EMI，在 Buck 中增加了开关频率抖动选项。在 1024 个时钟周期内，开关频率可以围绕中心频率线性变化±10%，如图 22-6 所示。相关的详细选项，请参见表 22-26。

图 22-6：频率抖动行为



22.4 寄存器

22.4.1 EPWR 寄存器列表

表 22-2: EPWR 模块基地址

外设模块	基地址
EPWR	0x4000 0400

表 22-3: EPWR 寄存器列表

寄存器	偏移地址	描述	复位值
EPWRTZFLT	0x00	ePower 滤波后的封锁事件寄存器	0x00000000
EPWRTZFLG	0x04	ePower 封锁状态标志寄存器	0x00000000
EPWRTZCLR	0x08	ePower 封锁状态清除寄存器	0x00000000
EPWRTZ0CTL*	0x0C	ePower 封锁事件 0 控制寄存器	0x00000288
EPWRTZ1CTL*	0x10	ePower 封锁事件 1 控制寄存器	0x00000288
EPWRCTL*	0x14	ePower 控制寄存器	0x00000000
EPWRREGKEY	0x18	ePower 模块写使能寄存器	0x1ACCE551

注意： 由*标识的寄存器仅当 EPWRREGKEY=0x1ACCE551 才可以改写。

22.4.2 EPWR 寄存器

22.4.2.1 ePower 滤波后的封锁事件寄存器（EPWRTZFLT）

表 22-4: ePower 滤波后的封锁事件寄存器（EPWRTZFLT）位段定义

EPWRTZFLT (ePower Module Filtered Trip-Zone Event Register)							
Access: EPWR->EPWRTZFLT Offset: 0x0000 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED						EPWRTZ1	EPWRTZ0

表 22-5: ePower 滤波后的封锁事件寄存器（EPWRTZFLT）位段描述

位段	位段名	属性	复位值	描述
31:2	RESERVED_31_2	RO	0x0	保留
1	EPWRTZ1	RO	0x0	滤波后的 ePower 封锁事件 1 状态（不带锁存） 0: ePower 封锁事件 1 未发生 1: ePower 封锁事件 1 已发生
0	EPWRTZ0	RO	0x0	滤波后的 ePower 封锁事件 0 状态（不带锁存） 0: ePower 封锁事件 0 未发生 1: ePower 封锁事件 0 已发生

22.4.2.2 ePower 封锁状态寄存器 (EPWRTZFLG)

表 22-6: ePower 封锁状态寄存器 (EPWRTZFLG) 位段定义

EPWRTZFLG (ePower Module Latched Trip-Zone Flag Register)							
Access: EPWR->EPWRTZFLG Offset: 0x0004 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED						EPWRTZ1	EPWRTZ0

表 22-7: ePower 封锁状态寄存器 (EPWRTZFLG) 位段描述

位段	位段名	属性	复位值	描述
31:2	RESERVED_31_2	RO	0x0	保留
1	EPWRTZ1	RO	0x0	滤波后的 ePower 封锁事件 1 状态 (带锁存) 0: ePower 封锁事件 1 未发生 1: ePower 封锁事件 1 已发生
0	EPWRTZ0	RO	0x0	滤波后的 ePower 封锁事件 0 状态 (带锁存) 0: ePower 封锁事件 0 未发生 1: ePower 封锁事件 0 已发生

22.4.2.3 ePower 封锁状态清除寄存器 (EPWRTZCLR)

表 22-8: ePower 封锁状态清除寄存器 (EPWRTZCLR) 位段定义

EPWRTZCLR (ePower Module Trip-Zone Clear Register)							
Access: EPWR->EPWRTZCLR Offset: 0x0008 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED						EPWRTZ1	EPWRTZ0

表 22-9: ePower 封锁状态清除寄存器 (EPWRTZCLR) 位段描述

位段	位段名	属性	复位值	描述
31:2	RESERVED_31_2	RO	0x0	保留
1	EPWRTZ1	W1C	0x0	清除锁存的 ePower 封锁事件 1 状态 0: 写 0 无效, 读总是读回 0 1: 写 1 清除 EPWRTZSTS.EPWRTZ1 本位段自动清零。
0	EPWRTZ0	W1C	0x0	清除锁存的 ePower 封锁事件 1 状态 0: 写 0 无效, 读总是读回 0 1: 写 1 清除 EPWRTZSTS.EPWRTZ0 本位段自动清零。

22.4.2.4 ePower 封锁事件 0 控制寄存器 (EPWRTZ0CTL)

表 22-10: ePower 封锁事件 0 控制寄存器 (EPWRTZ0CTL) 位段定义

EPWRTZ0CTL (ePower Module Trip-Zone Event 0 Control Register)									
Access: EPWR->EPWRTZ0CTL			Offset: 0x000C			Default: 0x00000288			
31 FLTRST	30 29		28 27		26	25 24		FLTWIN	
23 22 21 FLTWIN			20 19		18 17		16 FLTDIV		
15 14		13 12		11		10 SYNCCLREN	9 INPOL	8 IE	
7 POL4PIN		6 5 4 SEL4PIN		3 POL4PWM		2 1 0 SEL4PWM			

表 22-11: ePower 封锁事件 0 控制寄存器 (EPWRTZ0CTL) 位段描述

位段	位段名	属性	复位值	描述
31	FLTRST	W1C	0x0	用滤波器输入复位其内部寄存器 0: 写 0 无效, 读总是读回 0 1: 写 1 会用滤波器输入填充其内部寄存器 本位段自动清零。
30:26	FLTTH	RW	0x0	滤波器阈值为 (FLTTH+1) 个 1 或者 0
25:21	FLTWIN	RW	0x0	滤波器窗口大小为 (FLTWIN+1) 个采样数据
20:11	FLTDIV	RW	0x0	滤波器时钟分频系数为 (FLTDIV+1)
10	SYNCCLREN	RW	0x0	使能 PWMSYNC 信号清除锁存的 EPWRTZO 状态 0: PWMSYNC 不影响 EPWRTZO 状态 1: PWMSYNC 清除锁存的 EPWRTZO 状态
9	INPOL	RW	0x1	封锁信号输入极性 0: 当输入为低电平时产生封锁事件 1: 当输入为高电平时产生封锁事件
8	IE	RW	0x0	封锁事件输入使能 0: 禁能封锁信号输入 1: 使能封锁信号输入
7	POL4PIN	RW	0x1	输出到 GPIO 的封锁信号输出极性 0: 当 EPWRTZO 事件发生时, 输出低电平 1: 当 EPWRTZO 事件发生时, 输出高电平
6:4	SEL4PIN	RW	0x0	输出到 GPIO 的封锁信号输出选择 0: 原始输出 1: PWM 时钟的同步输出 2: 数字滤波器输出 3: 锁存的数字滤波器输出 4: 原始输出或者锁存的数字滤波器输出

位段	位段名	属性	复位值	描述
				5: PWM 时钟的同步输出或者锁存的数字滤波器输出 6: 数字滤波器输出或者锁存的数字滤波器输出 7: 原始输出或 PWM 时钟同步输出或锁存的数字滤波器输出
3	POL4PWM	RW	0x1	输出到 PWM 模块的封锁信号输出极性 0: 当 EPWRTZO 事件发生时, 输出低电平 1: 当 EPWRTZO 事件发生时, 输出高电平
2:0	SEL4PWM	RW	0x0	输出到 PWM 模块的封锁信号输出选择 0: 原始输出 1: PWM 时钟的同步输出 2: 数字滤波器输出 3: 锁存的数字滤波器输出 4: 原始输出或者锁存的数字滤波器输出 5: PWM 时钟的同步输出或者锁存的数字滤波器输出 6: 数字滤波器输出或者锁存的数字滤波器输出 7: 原始输出或 PWM 时钟同步输出或锁存的数字滤波器输出

22.4.2.5 ePower 封锁事件 1 控制寄存器 (EPWRTZ1CTL)

表 22-12: ePower 封锁事件 1 控制寄存器 (EPWRTZ1CTL) 位段定义

EPWRTZ1CTL (ePower Module Trip-Zone Event 1 Control Register)									
Access: EPWR->EPWRTZ1CTL			Offset: 0x0010		Default: 0x00000288				
31 FLTRST	30 29		28 FLTTH		27 26		25 24 FLTWIN		
23 22 FLTWIN			21 20		19 18 FLTDIV			17 16	
15 14		13 FLTDIV		12 11		10 SYNCCLREN	9 INPOL	8 IE	
7 POL4PIN	6 5 SEL4PIN		4		3 POL4PWM	2 1 SEL4PWM		0	

表 22-13: ePower 封锁事件 1 控制寄存器 (EPWRTZ1CTL) 位段描述

位段	位段名	属性	复位值	描述
31	FLTRST	W1C	0x0	用滤波器输入复位其内部寄存器 0: 写 0 无效, 读总是读回 0 1: 写 1 会用滤波器输入填充其内部寄存器 本位段自动清零。
30:26	FLTTH	RW	0x0	滤波器阈值为 (FLTTH+1) 个 1 或者 0
25:21	FLTWIN	RW	0x0	滤波器窗口大小为 (FLTWIN+1) 个采样数据
20:11	FLTDIV	RW	0x0	滤波器时钟分频系数为 (FLTDIV+1)
10	SYNCCLREN	RW	0x0	使能 PWMSYNC 信号清除锁存的 EPWRTZ1 状态 0: PWMSYNC 不影响 EPWRTZ1 状态 1: PWMSYNC 清除锁存的 EPWRTZ1 状态
9	INPOL	RW	0x1	封锁信号输入极性 0: 当输入为低电平时产生封锁事件 1: 当输入为高电平时产生封锁事件
8	IE	RW	0x0	封锁事件输入使能 0: 禁能封锁信号输入 1: 使能封锁信号输入
7	POL4PIN	RW	0x1	输出到 GPIO 的封锁信号输出极性 0: 当 EPWRTZ1 事件发生时, 输出低电平 1: 当 EPWRTZ1 事件发生时, 输出高电平
6:4	SEL4PIN	RW	0x0	输出到 PWM 模块的封锁信号输出选择 0: 原始输出 1: PWM 时钟的同步输出 2: 数字滤波器输出 3: 锁存的数字滤波器输出 4: 原始输出或者锁存的数字滤波器输出

位段	位段名	属性	复位值	描述
				5: PWM 时钟的同步输出或者锁存的数字滤波器输出 6: 数字滤波器输出或者锁存的数字滤波器输出 7: 原始输出或 PWM 时钟同步输出或锁存的数字滤波器输出
3	POL4PWM	RW	0x1	输出到 PWM 模块的封锁信号输出极性 0: 当 EPWRTZ1 事件发生时, 输出低电平 1: 当 EPWRTZ1 事件发生时, 输出高电平
2:0	SEL4PWM	RW	0x0	输出到 PWM 模块的封锁信号输出选择 0: 原始输出 1: PWM 时钟的同步输出 2: 数字滤波器输出 3: 锁存的数字滤波器输出 4: 原始输出或者锁存的数字滤波器输出 5: PWM 时钟的同步输出或者锁存的数字滤波器输出 6: 数字滤波器输出或者锁存的数字滤波器输出 7: 原始输出或 PWM 时钟同步输出或锁存的数字滤波器输出

22.4.2.6 ePower 控制寄存器 (EPWRCTL)

表 22-14: ePower 控制寄存器 (EPWRCTL) 位段定义

EPWRCTL (ePower Module Control Register)							
Access: EPWR->EPWRCTL Offset: 0x0014 Default: 0x00000000							
31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED						TZSEL4GPIO11	SYNCSEL1
7	6	5	4	3	2	1	0
SYNCSEL1		SYNCSELO			SPIDIR	ENSPI	EN

表 22-15: ePower 控制寄存器 (EPWRCTL) 位段描述

位段	位段名	属性	复位值	描述
31:10	RESERVED_31_9	RO	0x0	保留
9	TZSEL4GPIO11	RW	0x0	选择输出到 GPIO11 的封锁信号 0: EPWRTZ0 1: EPWRTZ1
8:6	SYNCSEL1	RW	0x0	从 PWM[SYNCSEL1]中选择 SYNC，以清除锁存的 EPWRTZ1 标志
5:3	SYNCSELO	RW	0x0	从 PWM[SYNCSELO]选择 SYNC，以清除锁存的 EPWRTZ0 标志
2	SPIDIR	RW	0x0	SPI 数据传输方向 0: 接收来自预驱模块的数据 1: 向预驱模块发送数据
1	ENSPI	RW	0x0	使能 ePower SPI 接口 0: 关闭 1: 使能
0	EN	RW	0x0	使能 ePower 模块 0: 关闭 1: 使能

22.4.2.7 ePower 模块写使能寄存器（EPWRREGKEY）

表 22-16: ePower 模块写使能寄存器（EPWRREGKEY）位段定义

EPWRREGKEY (ePower Module Register Write-Allow Key Register)							
Access: EPWR->EPWRREGKEY		Offset: 0x0018		Default: 0x1ACCE551			
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

表 22-17: ePower 模块写使能寄存器（EPWRREGKEY）位段描述

位段	位段名	属性	复位值	描述
31:0	KEY	RW	0x1ACCE551	写入 0x1ACCE551 解锁受保护的 ePower 寄存器

22.4.3 Buck 模块寄存器列表

表 22-18: Buck 模块寄存器列表

寄存器	偏移地址	描述	复位值
GLBSTS	0x0	全局状态寄存器	0x00
GLBSTSCLR	0x1	全局状态清除寄存器	0x00
TZCTL	0x2	封锁事件控制寄存器	0x08
BUCKCTL0	0x3	Buck 控制寄存器 0	0x80
BUCKCTL1	0x4	Buck 控制寄存器 1	0xC3

22.4.4 Buck 模块寄存器

22.4.4.1 全局状态寄存器（GLBSTS）

表 22-19：全局状态寄存器（GLBSTS）位段定义

GLBSTS (Global Status Register) Offset: 0x00 Default: 0x00							
7	6	5	4	3	2	1	0
RESERVED			EAUNITY	PFMMODE	OC	OTWARN	OV

表 22-20：全局状态寄存器（GLBSTS）位段描述

位段	位段名	属性	复位值	描述
7:5	RESERVED_7_5	RO	0x0	保留
4	EAUNITY	RO	0x0	误差放大器单位增益工作模式指示标志 0: 误差放大器未工作于单位增益模式 1: 误差放大器工作于单位增益模式
3	PFMMODE	RO	0x0	降压 PFM 操作模式指示标志 0: Buck 在 PWM 模式下运行 1: Buck 在 PFM 模式下运行
2	OC	RO	0x0	锁存的过电流状态 0: 过电流事件未发生 1: 过电流事件已发生
1	OTWARN	RO	0x1	锁存的过温警告状态 0: 过温警告事件未发生 1: 过温警告事件已发生
0	OV	RO	0x0	锁存的过电压状态 0: 过电压事件未发生 1: 过电压事件已发生

22.4.4.2 全局状态清除寄存器（GLBSTSCLR）

表 22-21：全局状态清除寄存器（GLBSTSCLR）位段定义

GLBSTSCLR (Global Status Clear Register) Offset: 0x01 Default: 0x00							
7	6	5	4	3	2	1	0
RESERVED					OC	OTWARN	OV

表 22-22：全局状态清除寄存器（GLBSTSCLR）位段描述

位段	位段名	属性	复位值	描述
7:3	RESERVED_7_3	RO	0x0	保留
2	OC	W1C	0x0	清除锁存的过电流状态 0: 写 0 无效，读总是读回 0 1: 写 1 清除 GLBSTS.OC 本位段自动清零。
1	OTWARN	W1C	0x0	清除锁存的过温度警告状态 0: 写 0 无效，读总是读回 0 1: 写 1 清除 GLBSTS.OTWARN 本位段自动清零。
0	OV	W1C	0x0	清除锁存的过电压状态 0: 写 0 无效，读总是读回 0 1: 写 1 清除 GLBSTS.OV 本位段自动清零。

22.4.4.3 封锁事件控制寄存器 (TZCTL)

表 22-23: 封锁事件控制寄存器 (TZCTL) 位段定义

TZCTL (Trip-Zone Control Register) Offset: 0x02 Default: 0x08							
7	6	5	4	3	2	1	0
RESERVED				POL	OCEN	OTWARNEN	OVEN

表 22-24: 封锁事件控制寄存器 (TZCTL) 位段描述

位段	位段名	属性	复位值	描述
7:4	RESERVED_7_4	RO	0x0	保留
3	POL	RW	0x1	封锁信号输出极性 0: 当封锁事件发生时, 输出低电平 1: 当封锁事件发生时, 输出高电平
2	OCEN	RW	0x0	使能过电流触发封锁事件 0: 过电流不触发封锁事件 1: 过电流触发封锁事件
1	OTWARNEN	RW	0x0	使能过温度警告触发封锁事件 0: 过温度警告不触发封锁事件 1: 过温度警告触发封锁事件
0	OVEN	RW	0x0	使能过电压触发封锁事件 0: 过电压不触发封锁事件 1: 过电压触发封锁事件

22.4.4.4 Buck 控制寄存器 0 (BUCKCTL0)

表 22-25: Buck 控制寄存器 0 (BUCKCTL0) 位段定义

BUCKCTL0 (Buck Control Register 0) Offset: 0x03 Default: 0x80							
7	6	5	4	3	2	1	0
FREQSEL		DBMCMP4PFM	DBZCMP4PFM	NOOC4PFM	FRCPWM	V33AON	PWDN

表 22-26: Buck 控制寄存器 0 (BUCKCTL0) 位段描述

位段	位段名	属性	复位值	描述
7:6	FREQSEL	RW	0x2	Buck 开关频率选择 0: 350kHz 1: 550kHz 2: 1000kHz 3: 2000kHz
5	DBMCMP4PFM	RW	0x0	PFM 模式下的去偏置主比较器 0: 偏置电流正常 1: 偏置电流减少一半
4	DBZCMP4PFM	RW	0x0	PFM 模式下的去偏置 ZCD 比较器 0: 偏置电流正常 1: 偏置电流减少一半
3	NOOC4PFM	RW	0x0	在 PFM 模式下禁用过电流检测 0: 过电流检测正常工作 1: 禁用过电流检测
2	FRCPWM	RW	0x0	强制 PWM 操作模式 0: 降压操作模式由电路控制 1: 强制 Buck 在 PWM 模式下运行
1	V33AON	W1S	0x0	3.3V 始终开启 0: 如果调整值为过电压，则无 3.3V 输出 1: 即使在过电压时，3.3V 也打开
0	PWDN	W1S	0x0	Buck DCDC 开关电源掉电 0: Buck DCDC 开关电源工作中 1: Buck DCDC 开关电源掉电

22.4.4.5 Buck 控制寄存器 1 (BUCKCTL1)

表 22-27: Buck 控制寄存器 1 (BUCKCTL1) 位段定义

BUCKCTL1 (Buck Control Register 1) Offset: 0x04 Default: 0xC3					
7	6	5	4	3	2
SSAMP		SSEN	SELOTWARN		1
					0
					SELOV

表 22-28: Buck 控制寄存器 1 (BUCKCTL1) 位段描述

位段	位段名	属性	复位值	描述
7:6	SSAMP	RW	0x3	频率展宽幅度 0: 2.5% 1: 3.3% 2: 5.0% 3: 10%
5	SSEN	RW	0x0	使能频率展宽 0: 频率展宽关闭 1: 频率展宽打开
4:2	SELOTWARN	RW	0x0	过温警告阈值选择 0: 146 °C 1: 138 °C 2: 127 °C 3: 122 °C 4: 112 °C 5: 102 °C 6: 162 °C 7: 无效
1:0	SELOV	RW	0x2	过压阈值选择 (迟滞为 0.16V) 0: 当 $V_{DDG} > 21.98V$ 时报告过压 当 $V_{DDG} < 19.00V$ 时撤销过压 1: 当 $V_{DDG} > 25.37V$ 时报告过压 当 $V_{DDG} < 23.00$ 时撤销过压 2: 当 $V_{DDG} > 28.28V$ 时报告过压 当 $V_{DDG} < 26.00V$ 时撤销过压 3: 当 $V_{DDG} > 31.62V$ 时报告过压 当 $V_{DDG} < 29.00V$ 时撤销过压

23 预驱模块（高压模块）

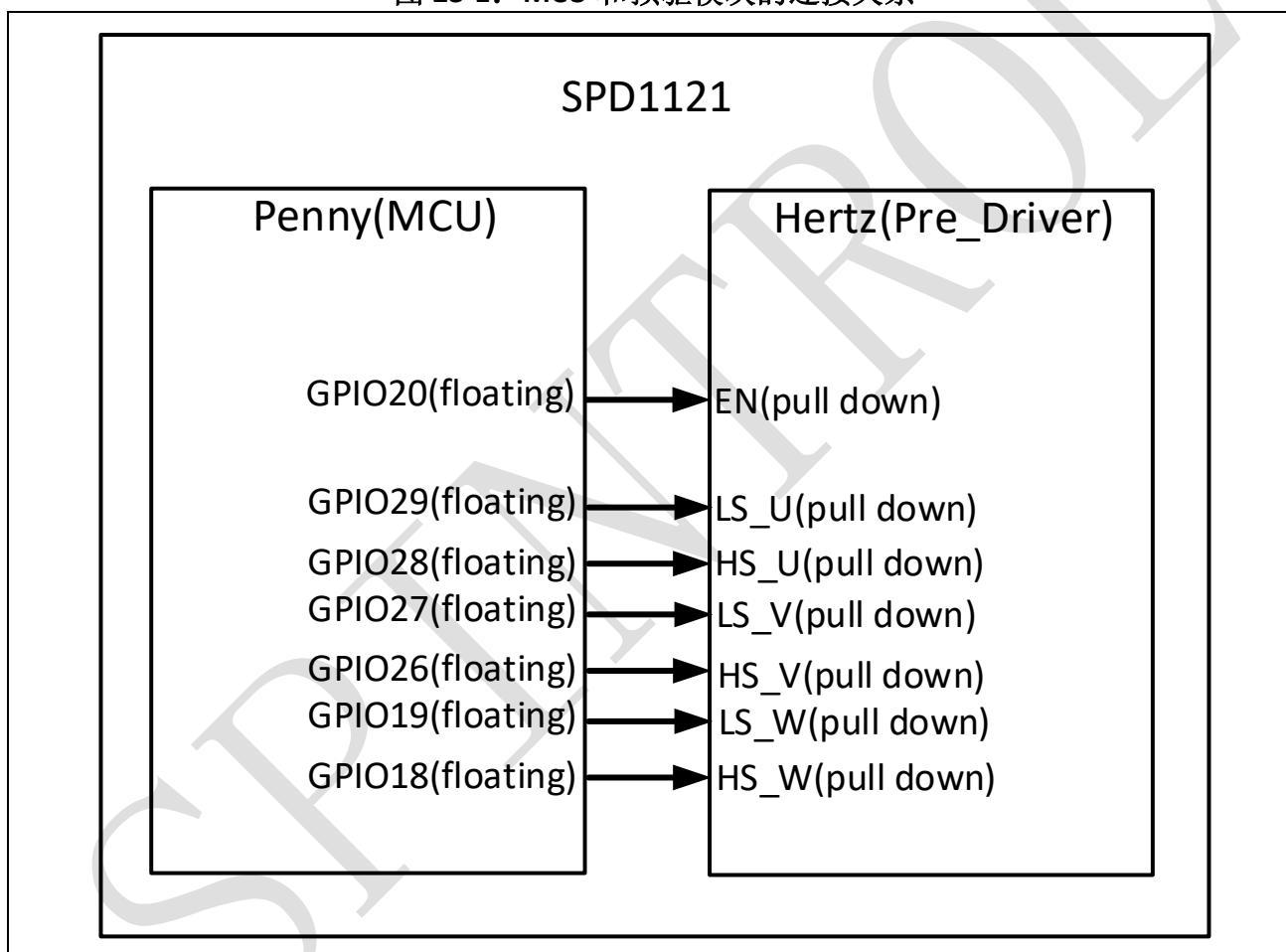
23.1 概述

SPD1121 是一种 SiP（系统级封装）芯片，内部集成了 MCU、开关电源和高压预驱模块，可以实现单个芯片驱动电机的应用。预驱模块提供了一个高压三相预驱，用于控制外部 MOSFET。典型应用框图如图 22-1 所示。

23.2 低压模块和高压模块的通信

低压 MCU 和高压部分在芯片内部的连接关系如图 23-1 所示。

图 23-1：MCU 和预驱模块的连接关系



低压 MCU 和高压部分的接口及其信号描述如表 23-1 所示。

表 23-1：MCU 和预驱模块的连接接口

MCU 引脚	引脚配置	预驱信号描述
GPIO20	GPIO 输出	预驱的使能控制
GPIO18	PWM3A	预驱 W 相上端 PWM 波形输入
GPIO19	PWM3B	预驱 W 相下端 PWM 波形输入
GPIO26	PWM4A	预驱 V 相上端 PWM 波形输入
GPIO27	PWM4B	预驱 V 相下端 PWM 波形输入
GPIO28	PWM5A	预驱 U 相上端 PWM 波形输入
GPIO29	PWM5B	预驱 U 相下端 PWM 波形输入

预驱部分集成了两个保护机制：

- VDDG 欠压，阈值为 8.9V，当故障发生时，关闭所有的 PWM 输出信号。
- 预驱部分还集成了噪声滤波器，防止开关毛刺影响正确的 PWM 信号传输。使能噪声滤波器后，从 GPIO 到 OUTH_U/V/W、OUTL_U/V/W 将增加大约 250ns 的额外延迟，但是 GPIO 上任何短于 250ns 的逻辑脉冲都将被视为噪声并被滤除。总传播延迟为 400ns。