



# HC32F472 系列

## 32 位 ARM® Cortex®-M4 微控制器

---

# 参考手册

Rev1.0 2023 年 12 月

## 声 明

- ★ 小华半导体有限公司（以下简称：“XHSC”）保留随时更改、更正、增强、修改小华半导体产品和/或本文档的权利，恕不另行通知。用户可在下单前获取最新相关信息。XHSC 产品依据购销基本合同中载明的销售条款和条件进行销售。
- ★ 客户应针对您的应用选择合适的 XHSC 产品，并设计、验证和测试您的应用，以确保您的应用满足相应标准以及任何安全、安保或其它要求。客户应对此独自承担全部责任。
- ★ XHSC 在此确认未以明示或暗示方式授予任何知识产权许可。
- ★ XHSC 产品的转售，若其条款与此处规定不同，XHSC 对此类产品的任何保修承诺无效。
- ★ 任何带有“®”或“™”标识的图形或字样是 XHSC 的商标。所有其他在 XHSC 产品上显示的产品或服务名称均为其各自所有者的财产。
- ★ 本通知中的信息取代并替换先前版本中的信息。

©2023 小华半导体有限公司 保留所有权利

## 目 录

声 明.....	2
目 录.....	3
表索引.....	36
图索引.....	41
文档约定.....	51
寄存器相关缩略语.....	51
词汇表.....	51
关于本手册.....	51
简介 (Overview) .....	52
<b>1 存储器映射 (Memory Mapping) .....</b>	<b>53</b>
1.1 存储器映射.....	53
1.2 外部空间映射.....	59
1.3 位段空间.....	59
1.4 地址重映射.....	59
1.5 重映射寄存器.....	61
1.5.1 访问保护寄存器 (MMF_REMPRT) .....	61
1.5.2 重映射控制寄存器 (MMF_REMCRx) (x=0, 1).....	62
<b>2 总线架构 (BUS) .....</b>	<b>63</b>
2.1 概述.....	63
2.2 总线架构.....	64
2.3 总线功能.....	65
<b>3 复位控制 (RMU) .....</b>	<b>66</b>
3.1 简介.....	66
3.2 复位方式和复位标志位.....	67
3.3 复位时序.....	69
3.3.1 上电复位 .....	69
3.3.2 NRST 引脚复位 .....	69
3.3.3 欠压复位 .....	70
3.3.4 可编程电压检测 1 复位、可编程电压检测 2 复位 .....	71
3.3.5 看门狗复位、专用看门狗复位.....	72
3.3.6 掉电唤醒复位 .....	73
3.3.7 软件复位 .....	73
3.3.8 MPU 错误复位.....	74

3.3.9	RAM 奇偶校验复位.....	74
3.3.10	RAMECC 复位 .....	74
3.3.11	时钟频率异常复位 .....	75
3.3.12	外部高速振荡器异常停振复位.....	75
3.3.13	Cortex-M4 Lockup 复位 .....	76
3.3.14	复位方式的判断.....	76
3.4	功能复位控制 .....	76
3.5	寄存器说明.....	77
3.5.1	功能复位控制 0(RMU_FRST0) .....	78
3.5.2	功能复位控制 1(RMU_FRST1) .....	79
3.5.3	功能复位控制 2(RMU_FRST2) .....	80
3.5.4	功能复位控制 3(RMU_FRST3) .....	81
3.5.5	复位控制寄存器 (RMU_PRSTCR0) .....	83
3.5.6	复位标志寄存器 0 (RMU_RSTF0) .....	84
<b>4</b>	<b>时钟控制器 (CMU) .....</b>	<b>86</b>
4.1	简介 .....	86
4.2	系统框图 .....	87
4.2.1	系统框图 .....	87
4.2.2	时钟频率测量框图 .....	88
4.3	时钟源规格.....	89
4.4	工作时钟规格 .....	90
4.5	晶振电路 .....	92
4.5.1	外部高速振荡器.....	92
4.5.2	外部高速振荡器故障检测.....	94
4.5.3	外部高速振荡器小数分频.....	96
4.5.4	外部低速振荡器.....	97
4.6	内部 RC 时钟 .....	98
4.6.1	HRC 时钟 .....	98
4.6.2	MRC 时钟 .....	98
4.6.3	LRC 时钟 .....	99
4.6.4	SWDTLRC 时钟 .....	99
4.7	PLL 时钟 .....	100
4.8	时钟切换步骤 .....	100
4.8.1	时钟源切换 .....	101
4.8.2	时钟分频切换 .....	102

4.9	时钟输出功能 .....	103
4.10	时钟频率测量 .....	104
4.10.1	时钟频率测量 .....	104
4.10.2	数字滤波功能 .....	105
4.10.3	中断/复位功能 .....	105
4.11	寄存器说明 .....	106
4.11.1	CMU XTAL 配置寄存器(CMU_XTALCFGR).....	108
4.11.2	CMU XTAL 配置寄存器(CMU_XTALSTBCR).....	108
4.11.3	CMU XTAL 小数分频配置寄存器(CMU_XTALDIVR).....	109
4.11.4	CMU XTAL 小数分频控制存器(CMU_XTALDIVCR).....	109
4.11.5	CMU XTAL 控制寄存器(CMU_XTALCR).....	110
4.11.6	CMU XTAL 振荡故障控制寄存器(CMU_XTALSTDSCR) .....	111
4.11.7	CMU XTAL 振荡故障状态寄存器(CMU_XTALSTDSSR) .....	112
4.11.8	CMU XTAL32 配置寄存器(CMU_XTAL32CFGR).....	112
4.11.9	CMU XTAL32 滤波寄存器(CMU_XTAL32NFR).....	113
4.11.10	CMU XTAL32 控制寄存器(CMU_XTAL32CR).....	113
4.11.11	CMU HRC 校准寄存器(CMU_HRCTRM).....	114
4.11.12	CMU HRC 控制寄存器(CMU_HRCCR) .....	114
4.11.13	CMU MRC 校准寄存器(CMU_MRCTRM).....	115
4.11.14	CMU MRC 控制寄存器(CMU_MRCCR) .....	115
4.11.15	CMU LRC 校准寄存器(CMU_LRCTRM).....	116
4.11.16	CMU LRC 控制寄存器(CMU_LRCCR) .....	116
4.11.17	CMU PLLH 配置寄存器(CMU_PLLHCFGR) .....	117
4.11.18	CMU PLLH 控制存器(CMU_PLLHCR) .....	119
4.11.19	CMU 时钟源稳定状态器(CMU_OSCSTBSR) .....	119
4.11.20	CMU 系统时钟源切换寄存器(CMU_CKSWR) .....	120
4.11.21	CMU 时钟分频配置寄存器(CMU_SCFGR) .....	121
4.11.22	CMU USB 时钟配置存器(CMU_USBCKCFGR) .....	123
4.11.23	CMU CAN 时钟配置存器(CMU_CANCKCFGR) .....	124
4.11.24	CMU AD/TRNG/DA 时钟配置存器(CMU_PERICKSEL) .....	126
4.11.25	CMU 调试时钟配置存器(CMU_TPIUCKCFGR) .....	126
4.11.26	CMU MCO1 配置存器(CMU_MCO1CFGR) .....	127
4.11.27	CMU MCO2 配置存器(CMU_MCO2CFGR) .....	128
4.11.28	FCM 下限比较值寄存器(FCM_LVR) .....	128
4.11.29	FCM 上限比较值寄存器(FCM_UVR) .....	129

4.11.30 FCM 计数器值寄存器(FCM_CNTR).....	129
4.11.31 FCM 开始停止寄存器(FCM_STR) .....	129
4.11.32 FCM 测量对象控制寄存器(FCM_MCCR) .....	130
4.11.33 FCM 测量基准控制寄存器(FCM_RCCR) .....	131
4.11.34 FCM 中断复位控制寄存器(FCM_RIER) .....	132
4.11.35 FCM 标志寄存器(FCM_SR) .....	132
4.11.36 FCM 标志位清除寄存器(FCM_CLR) .....	133
<b>5 电源控制 (PWC) .....</b>	<b>134</b>
5.1 简介 .....	134
5.2 电源分布 .....	135
5.3 电源电压检测单元(PVD)说明 .....	136
5.3.1 上电复位/掉电复位动作说明 .....	136
5.3.2 欠压复位(BOR)说明 .....	137
5.3.3 可编程电压检测 1(PVD1)、可编程电压检测 2(PVD2) .....	138
5.3.4 PVD1、 PVD2 中断/复位框图 .....	139
5.3.5 输入/输出引脚 .....	139
5.3.6 PVD1 中断和复位.....	140
5.3.7 PVD2 中断和复位.....	142
5.3.8 内部电压采样和检测功能.....	143
5.4 唤醒定时器.....	144
5.5 动作模式与低功耗模式 .....	144
5.5.1 睡眠模式 .....	147
5.5.2 停止模式 .....	147
5.5.3 掉电模式 .....	148
5.5.4 降低系统时钟速度 .....	151
5.5.5 关闭不使用的时钟源.....	151
5.5.6 功能时钟停止 .....	152
5.5.7 关闭不使用的 RAM.....	152
5.6 寄存器保护功能.....	153
5.7 寄存器说明.....	154
5.7.1 电源模式控制寄存器 0 (PWC_PWRC0) .....	156
5.7.2 电源模式控制寄存器 1 (PWC_PWRC1) .....	157
5.7.3 电源模式控制寄存器 4 (PWC_PWRC4) .....	157
5.7.4 掉电唤醒使能寄存器 0(PWC_PDWKE0) .....	158
5.7.5 掉电唤醒使能寄存器 1(PWC_PDWKE1) .....	159

5.7.6	掉电唤醒使能寄存器 2(PWC_PDWKE2) .....	160
5.7.7	掉电唤醒事件边沿选择寄存器(PWC_PDWKES) .....	160
5.7.8	掉电唤醒标志寄存器 0(PWC_PDWKF0) .....	161
5.7.9	掉电唤醒标志寄存器 1(PWC_PDWKF1) .....	162
5.7.10	电源模式控制寄存器 5 (PWC_PWRC5).....	163
5.7.11	电源模式控制寄存器 6 (PWC_PWRC6) .....	164
5.7.12	功能时钟控制 0(PWC_FCG0).....	165
5.7.13	功能时钟控制 1(PWC_FCG1).....	167
5.7.14	功能时钟控制 2(PWC_FCG2).....	169
5.7.15	功能时钟控制 3(PWC_FCG3).....	171
5.7.16	PWC_FCG0 保护控制(PWC_FCG0PC) .....	173
5.7.17	功能保护控制寄存器(PWC_FPRC).....	173
5.7.18	STOP 模式控制寄存器(PWC_STPMCR).....	174
5.7.19	RAM 功耗控制寄存器 0(PWC_RAMPC0) .....	174
5.7.20	外设 RAM 低功耗控制寄存器 (PWC_PRAMLPC) .....	175
5.7.21	PVD 控制寄存器 0(PWC_PVDCR0).....	175
5.7.22	PVD 控制寄存器 1(PWC_PVDCR1).....	176
5.7.23	PVD 滤波控制寄存器(PWC_PVDFCR).....	177
5.7.24	PVD 电平控制寄存器(PWC_PVDLCR).....	178
5.7.25	PVD 中断控制寄存器(PWC_PVDICR).....	179
5.7.26	PVD 检测状态寄存器(PWC_PVDDSR) .....	180
5.7.27	唤醒计时器控制寄存器(PWC_WKTCR) .....	180
<b>6</b>	<b>初始化配置 (ICG) .....</b>	<b>181</b>
6.1	简介 .....	181
6.2	寄存器说明.....	182
6.2.1	初始化配置寄存器 0(ICG0) .....	182
6.2.2	初始化配置寄存器 1(ICG1) .....	184
6.2.3	初始化配置寄存器 3(ICG3) .....	185
6.2.4	初始化配置寄存器 4(ICG4) .....	185
<b>7</b>	<b>嵌入式 Flash (EFM) .....</b>	<b>186</b>
7.1	简介 .....	186
7.2	主要特性 .....	186
7.3	嵌入式 Flash .....	187
7.4	读接口.....	189
7.4.1	CPU 时钟和 Flash 读取时间之间的关系 .....	189

7.5	Flash 读加速缓存 .....	189
7.6	Flash 编程和擦除操作 .....	190
7.6.1	解锁 EFM_KEY1 寄存器 .....	190
7.6.2	写保护功能 .....	191
7.6.3	单次编程无回读功能 .....	191
7.6.4	单编程回读功能 .....	191
7.6.5	连续编程功能 .....	192
7.6.6	擦除功能 .....	192
7.6.7	全擦除功能 .....	193
7.6.8	数据安全保护 .....	193
7.6.9	D-BUS 读保护功能 .....	194
7.6.10	BGO 功能 .....	195
7.6.11	中断 .....	196
7.7	一次性可编程字节 (OTP) .....	197
7.8	引导交换 .....	199
7.9	寄存器说明 .....	201
7.9.1	Flash 访问写保护寄存器(EFM_FAPRT) .....	202
7.9.2	Flash 密钥 1 寄存器(EFM_KEY1) .....	202
7.9.3	Flash 密钥 2 寄存器(EFM_KEY2) .....	203
7.9.4	Flash 停止寄存器(EFM_FSTP) .....	203
7.9.5	Flash 读模式寄存器(EFM_FRMC) .....	204
7.9.6	Flash 擦写模式寄存器(EFM_FWMC) .....	205
7.9.7	Flash 状态寄存器(EFM_FSR) .....	206
7.9.8	Flash 状态清除寄存器(EFM_FSCLR) .....	208
7.9.9	Flash 中断许可寄存器(EFM_FITE) .....	210
7.9.10	Flash 引导交换状态寄存器(EFM_FSWP) .....	210
7.9.11	芯片专属标志寄存器(EFM_CHIPID) .....	211
7.9.12	UNIQUE ID 寄存器 0(EFM_UQID0) .....	211
7.9.13	UNIQUE ID 寄存器 1(EFM_UQID1) .....	211
7.9.14	UNIQUE ID 寄存器 2(EFM_UQID2) .....	212
7.9.15	Flash 写保护锁定寄存器(EFM_WLOCK) .....	212
7.9.16	Flash0 写保护寄存器(EFM_F0NWPRT) .....	213
7.9.17	Flash1 写保护寄存器(EFM_F1NWPRT) .....	214
7.10	注意事项 .....	215
8	内置 SRAM (SRAM) .....	216

8.1 简介 .....	216
8.2 寄存器说明.....	218
8.2.1 SRAM 校验控制寄存器 (SRAM_CKCR) .....	219
8.2.2 SRAM 校验控制保护寄存器 (SRAM_CKPR) .....	221
8.2.3 SRAM 校验状态寄存器 (SRAM_CKSR) .....	222
8.2.4 SRAM0 错误注入使能寄存器 (SRAM0_EIEN) .....	222
8.2.5 SRAM0 错误注入位使能寄存器 0 (SRAM0_EIBIT0) .....	223
8.2.6 SRAM0 错误注入位使能寄存器 1 (SRAM0_EIBIT1) .....	223
8.2.7 SRAM0 校验错误地址寄存器 (SRAM0_ECCERRADDR) .....	223
8.2.8 Ret_SRAM 错误注入使能寄存器(SRAMB_EIEN) .....	224
8.2.9 Ret_SRAM 错误注入位使能寄存器 0(SRAMB_EIBIT0) .....	224
8.2.10 Ret_SRAM 错误注入位使能寄存器 1(SRAMB_EIBIT1) .....	225
8.2.11 Ret_SRAM 校验错误地址寄存器(SRAMB_ECCERRADDR) .....	225
<b>9 通用 IO (GPIO) .....</b>	<b>226</b>
9.1 简介 .....	226
9.2 端口功能概要 .....	227
9.3 动作说明 .....	228
9.3.1 通用输入输出 GPIO 功能 .....	228
9.3.2 周边功能 .....	228
9.3.3 双周边功能 .....	229
9.3.4 Event Port 输入输出功能 .....	229
9.3.5 外部中断 EIRQ 输入功能 .....	229
9.3.6 模拟功能 .....	230
9.3.7 通用控制 .....	230
9.4 寄存器说明.....	231
9.4.1 通用输入数据寄存器 (PIDRx) .....	232
9.4.2 通用输出数据寄存器 (PODRx) .....	232
9.4.3 通用输出许可寄存器 (POERx) .....	232
9.4.4 通用输出置位寄存器 (POSRx) .....	233
9.4.5 通用输出复位寄存器 (PORRx) .....	233
9.4.6 通用输出翻转寄存器 (POTRx) .....	233
9.4.7 特殊控制寄存器 (PSPCR) .....	234
9.4.8 公共控制寄存器 (PCCR) .....	235
9.4.9 写保护寄存器 (PWPR) .....	235
9.4.10 通用控制寄存器 (PCRxy) .....	236

9.4.11	功能选择寄存器 (PFSR <sub>xy</sub> ) .....	238
9.4.12	Event Port 方向选择寄存器 (PEVNTDIRm) .....	239
9.4.13	Event Port 输入数据寄存器 (PEVNTIDRm) .....	239
9.4.14	Event Port 输出数据寄存器 (PEVNTODRm) .....	240
9.4.15	Event Port 输出数据复位寄存器 (PEVNTORRm) .....	240
9.4.16	Event Port 输出数据置位寄存器 (PEVNTOSRm) .....	241
9.4.17	Event Port 上升沿输入许可寄存器 (PEVNTRISRm) .....	241
9.4.18	Event Port 下降沿输入许可寄存器 (PEVNTFALRm) .....	242
9.4.19	Event Port 输入滤波控制寄存器 (PEVNTNFCR) .....	243
9.4.20	32bit 访问 .....	244
9.5	注意事项 .....	245
<b>10</b>	<b>中断控制器 (INTC) .....</b>	<b>246</b>
10.1	简介 .....	246
10.2	系统框图 .....	247
10.3	向量表 .....	248
10.3.1	中断向量表 .....	248
10.3.2	中断事件表 .....	254
10.4	功能说明 .....	269
10.4.1	不可屏蔽中断 .....	269
10.4.2	外部中断 .....	269
10.4.3	中断事件多路器 .....	270
10.4.4	软件中断 .....	270
10.4.5	事件输入 .....	270
10.4.6	WFE 事件管理 .....	271
10.4.7	噪声滤波器 .....	271
10.4.8	低功耗模式返回 .....	272
10.5	寄存器说明 .....	273
10.5.1	不可屏蔽中断使能寄存器 (INTC_NMIER) .....	275
10.5.2	不可屏蔽中断标志寄存器 (INTC_NMIFR) .....	276
10.5.3	不可屏蔽中断标志清除寄存器 (INTC_NMIFCR) .....	277
10.5.4	外部中断控制寄存器(INTC_EIRQCR <sub>x</sub> )(x=0~15) .....	278
10.5.5	外部中断标志寄存器(INTC{EIFR}) .....	279
10.5.6	外部中断标志清除寄存器(INTC{EIFCR}) .....	279
10.5.7	中断事件选择寄存器 (INTC_INTSEL0~15) .....	280
10.5.8	中断事件屏蔽寄存器 (INTC_INTEN0~15) .....	280

10.5.9	停止模式唤醒事件使能寄存器 (INTC_WKEN) .....	281
10.5.10	软件中断寄存器 (INTC_SWIER) .....	282
10.5.11	事件使能寄存器 (INTC_EVTER) .....	282
10.5.12	中断使能寄存器 (INTC_IER) .....	282
10.6	使用注意事项 .....	283
<b>11</b>	<b>自动运行系统 (AOS) .....</b>	<b>284</b>
11.1	简介 .....	284
11.1.1	功能概览 .....	284
11.1.2	模块框图 .....	285
11.2	功能描述 .....	286
11.2.1	AOS 源事件列表 .....	286
11.2.2	AOS 目标列表 .....	286
11.2.3	专用触发源 .....	288
11.2.4	公共触发源 .....	288
11.3	寄存器说明 .....	289
11.3.1	寄存器一览 .....	289
11.3.2	外设触发事件寄存器 (INTSFTTRG) .....	290
11.3.3	DCU 触发源选择寄存器 (DCU_TRGSELx, x=1~3) .....	290
11.3.4	DMA1 传输启动触发源选择寄存器 (DMA1_TRGSELx, x=0~7) .....	291
11.3.5	DMA2 传输启动触发源选择寄存器 (DMA2_TRGSELx, x=0~7) .....	292
11.3.6	DMA 通道重置触发源选择寄存器 (DMA_RC_TRGSEL) .....	293
11.3.7	Timer6 硬件触发事件选择寄存器 (TMR6_TRGSELx, x=0~3) .....	294
11.3.8	Timer4 硬件触发事件选择寄存器 (TMR4_TRGSEL) .....	295
11.3.9	Event Port 触发源选择寄存器 (PEVNT_TRGSEL12, PEVNT_TRGSEL34) .....	296
11.3.10	Timer0 硬件触发事件选择寄存器 (TMR0_TRGSEL) .....	297
11.3.11	Timer2 硬件触发事件选择寄存器 (TMR2_TRGSEL) .....	298
11.3.12	HASH 硬件触发事件选择寄存器 A (HASH_TRGSEL0) .....	299
11.3.13	HASH 硬件触发事件选择寄存器 B (HASH_TRGSEL1) .....	299
11.3.14	TimerA 硬件触发事件选择寄存器 (TMRA_TRGSELx, x=0~3) .....	300
11.3.15	OTS 触发源选择寄存器 (OTS_TRGSEL) .....	300
11.3.16	ADC1 片内触发源选择寄存器 (ADC1_TRGSELx, x=0,1) .....	301
11.3.17	ADC2 片内触发源选择寄存器 (ADC2_TRGSELx, x=0,1) .....	302
11.3.18	ADC3 片内触发源选择寄存器 (ADC3_TRGSELx, x=0,1) .....	303
11.3.19	公共触发源选择寄存器 (AOS_COMTRGSEL1, 2) .....	303
<b>12</b>	<b>存储保护单元 (MPU) .....</b>	<b>304</b>

12.1 简介 .....	304
12.2 功能描述 .....	305
12.2.1 区域范围设置 .....	305
12.2.2 权限设置 .....	305
12.2.3 MPU 动作选择 .....	305
12.2.4 启动 MPU .....	305
12.3 应用举例 .....	306
12.3.1 只允许部分空间访问 .....	306
12.3.2 只禁止部分空间访问 .....	306
12.4 寄存器说明 .....	307
12.4.1 区域范围描述寄存器 MPU_RGDN (n=0~15) .....	308
12.4.2 状态标志寄存器 MPU_SR .....	308
12.4.3 标志清除寄存器 MPU_ECLR .....	309
12.4.4 写保护寄存器 MPU_WP .....	309
12.4.5 IP 访问保护寄存器 MPU_IPPR .....	310
12.4.6 MSP 保护开始地址寄存器 MPU_MSPPBA .....	312
12.4.7 MSP 保护控制寄存器 MPU_MSPPCTL .....	312
12.4.8 PSP 保护开始地址寄存器 MPU_PSPPBA .....	313
12.4.9 PSP 保护控制寄存器 MPU_PSPPCTL .....	313
12.4.10 SMPU1 区域使能寄存器 MPU_S1RGE .....	314
12.4.11 SMPU1 区域写权限寄存器 MPU_S1RGWP .....	315
12.4.12 SMPU1 区域读权限寄存器 MPU_S1RGRP .....	316
12.4.13 SMPU1 控制寄存器 MPU_S1CR .....	317
12.4.14 SMPU2 区域使能寄存器 MPU_S2RGE .....	318
12.4.15 SMPU2 区域写权限寄存器 MPU_S2RGWP .....	319
12.4.16 SMPU2 区域读权限寄存器 MPU_S2RGRP .....	320
12.4.17 SMPU2 控制寄存器 MPU_S2CR .....	321
12.4.18 FMPU 区域使能寄存器 MPU_FRGE .....	322
12.4.19 FMPU 区域写权限寄存器 MPU_FRGWP .....	323
12.4.20 FMPU 区域读权限寄存器 MPU_FRGRP .....	324
12.4.21 FMPU 控制寄存器 MPU_FCR .....	325
<b>13 键盘扫描控制模块 (KEYSCAN) .....</b>	<b>326</b>
13.1 简介 .....	326
13.2 KEYS defense 系统框图 .....	327
13.3 管脚说明 .....	327

13.4 功能说明 .....	328
13.4.1 按键识别功能 .....	328
13.4.2 键盘扫描功能 .....	328
13.4.3 使用上的注意事项 .....	328
13.5 寄存器说明.....	329
13.5.1 KEYS defense 扫描控制寄存器 (KEYSCAN_SCR) .....	330
13.5.2 KEYS defense 扫描使能寄存器 (KEYCAN_SER) .....	332
13.5.3 KEYS defense 扫描状态寄存器 (KEYSCAN_SSR) .....	332
<b>14 内部时钟校准器 (CTC) .....</b>	<b>333</b>
14.1 简介 .....	333
14.2 结构框图 .....	333
14.3 功能说明 .....	334
14.3.1 参考时钟 .....	334
14.3.2 频率校准 .....	339
14.3.3 错误中断 .....	340
14.3.4 编程指南 .....	340
14.4 寄存器说明.....	341
14.4.1 时钟校准控制寄存器 1 (CTC_CR1) .....	342
14.4.2 时钟校准控制寄存器 2 (CTC_CR2) .....	344
14.4.3 时钟校准状态寄存器 (CTC_STR) .....	345
14.4.4 时钟校准计数器 (CTC_CNT) .....	345
<b>15 DMA 控制器 (DMA) .....</b>	<b>346</b>
15.1 简介 .....	346
15.2 模块示意图.....	347
15.3 功能描述 .....	348
15.3.1 使能 DMA 控制器 .....	348
15.3.2 通道选择和通道优先级 .....	348
15.3.3 启动 DMA .....	348
15.3.4 数据块 .....	348
15.3.5 传输地址控制 .....	348
15.3.6 传输次数 .....	349
15.3.7 中断和事件信号输出 .....	349
15.3.8 连锁传输 .....	350
15.3.9 不连续地址传输 .....	352
15.3.10 通道重置 .....	353

15.3.11	传输提前终止 .....	354
15.4	应用举例 .....	355
15.4.1	存储器到存储器的传输 .....	355
15.4.2	存储器到外围电路的传输 .....	357
15.4.3	存储器到存储器的链锁传输 .....	358
15.5	寄存器说明 .....	360
15.5.1	寄存器一览 .....	360
15.5.2	DMA 使能寄存器(DMA_EN) .....	361
15.5.3	中断状态寄存器 0(DMA_INTSTAT0) .....	361
15.5.4	中断状态寄存器 1(DMA_INTSTAT1) .....	362
15.5.5	中断屏蔽寄存器 0(DMA_INTMASK0) .....	362
15.5.6	中断屏蔽寄存器 1(DMA_INTMASK1) .....	363
15.5.7	中断复位寄存器 0(DMA_INTCLR0) .....	363
15.5.8	中断复位寄存器 1(DMA_INTCLR1) .....	364
15.5.9	通道使能寄存器(DMA_CHEN) .....	364
15.5.10	通道使能复位寄存器(DMA_CHENCLR) .....	365
15.5.11	通道重置控制寄存器(DMA_RCFGCTL) .....	366
15.5.12	传输请求状态寄存器(DMA_REQSTAT) .....	368
15.5.13	通道状态观测寄存器(DMA_CHSTAT) .....	368
15.5.14	传输源地址寄存器(DMA_SARx) ( $x=0\sim 7$ ) .....	369
15.5.15	传输目标地址寄存器(DMA_DARx) ( $x=0\sim 7$ ) .....	369
15.5.16	数据控制寄存器(DMA_DTCTLx) ( $x=0\sim 7$ ) .....	370
15.5.17	重复区域大小寄存器(DMA_RPTx) ( $x=0\sim 7$ ) .....	371
15.5.18	重复区域大小寄存器 B(DMA_RPTBx) ( $x=0\sim 7$ ) .....	372
15.5.19	源设备不连续地址传输控制寄存器(DMA_SNSEQCTLx) ( $x=0\sim 7$ ) .....	373
15.5.20	源设备不连续地址传输控制寄存器 B(DMA_SNSEQCTLBx) ( $x=0\sim 7$ ) .....	374
15.5.21	目标设备不连续地址传输控制寄存器(DMA_DNSEQCTLx) ( $x=0\sim 7$ ) .....	375
15.5.22	目标设备不连续地址传输控制寄存器 B(DMA_DNSEQCTLBx) ( $x=0\sim 7$ ) .....	376
15.5.23	链指针寄存器(DMA_LLPx) ( $x=0\sim 7$ ) .....	377
15.5.24	通道控制寄存器(DMA_CHCTLx) ( $x=0\sim 7$ ) .....	378
15.5.25	通道监视寄存器(DMA_MONSARx, DMA_MONDARx, DMA_MONDTCTLx, DMA_MONRPTx, DMA_MONSSEQCTLx, DMA_MONDNSEQCTLx) ( $x=0\sim 7$ ) .....	379
15.6	使用注意事项 .....	379
<b>16</b>	<b>电压比较器 (CMP)</b> .....	<b>380</b>
16.1	简介 .....	380

16.2 功能框图 .....	381
16.3 功能说明 .....	384
16.3.1 输入选择 .....	384
16.3.2 电压比较 .....	385
16.3.3 窗口比较 .....	386
16.3.4 空白窗口 .....	388
16.3.5 迟滞功能 .....	391
16.3.6 噪声滤波 .....	391
16.3.7 中断事件 .....	391
16.3.8 外部输出 .....	391
16.4 注意事项 .....	392
16.4.1 模块停止功能 .....	392
16.4.2 模块停止时的功耗 .....	392
16.4.3 停止低功耗模式时的功耗 .....	392
16.5 寄存器说明 .....	393
16.5.1 比较器工作模式寄存器 (CMPx_MDR, x=1~4) .....	394
16.5.2 比较器滤波中断寄存器 (CMPx_FIR, x=1~4) .....	395
16.5.3 比较器输出控制寄存器 (CMPx_OCR, x=1~4) .....	396
16.5.4 比较器结果监视寄存器 (CMPx_OMR, x=1~4) .....	396
16.5.5 比较器输入选择寄存器 (CMPx_PMSR, x=1~4) .....	397
16.5.6 比较器输出空白窗口设定寄存器 1 (CMPx_BWSR1, x=1~4) .....	398
16.5.7 比较器输出空白窗口设定寄存器 2 (CMPx_BWSR2, x=1~4) .....	400
<b>17 模数转换模块 (ADC) .....</b>	<b>401</b>
17.1 简介 .....	401
17.2 ADC 系统框图 .....	403
17.3 功能说明 .....	407
17.3.1 ADC 时钟 .....	407
17.3.2 通道选择 .....	407
17.3.3 触发源选择 .....	408
17.3.4 序列 A 单次扫描模式 .....	408
17.3.5 序列 A 连续扫描模式 .....	409
17.3.6 双序列扫描模式 .....	410
17.3.7 模拟看门狗功能 .....	413
17.3.8 模拟输入的采样时间和转换时间 .....	413
17.3.9 ADC 数据寄存器自动清除功能 .....	415

17.3.10 转换数据平均计算功能 .....	415
17.3.11 可编程增益放大器 PGA .....	416
17.3.12 多 ADC 协同工作模式 .....	417
17.3.13 中断和事件信号输出 .....	422
<b>17.4 寄存器说明.....</b>	<b>423</b>
17.4.1    寄存器一览 .....	423
17.4.2    ADC 启动寄存器 ADC_STR .....	424
17.4.3    ADC 控制寄存器 0 ADC_CR0.....	425
17.4.4    ADC 控制寄存器 1 ADC_CR1.....	426
17.4.5    ADC 转换开始触发寄存器 ADC_TRGSR .....	427
17.4.6    ADC 通道选择寄存器 A ADC_CHSELRA .....	428
17.4.7    ADC 通道选择寄存器 B ADC_CHSELRB .....	429
17.4.8    ADC 平均通道选择寄存器 ADC_AVCHSELR.....	430
17.4.9    ADC 采样周期寄存器 ADC_SSTRx, x=0~15/ADC_SSTRL .....	431
17.4.10   ADC 通道映射控制寄存器 ADC_CHMUXR .....	432
17.4.11   ADC 中断状态寄存器 ADC_ISR .....	433
17.4.12   ADC 中断许可寄存器 ADC_ICR .....	433
17.4.13   ADC 中断状态复位寄存器 ADC_ISCLRR .....	434
17.4.14   ADC 协同模式控制寄存器 ADC_SYNCCR .....	435
17.4.15   ADC 数据寄存器 ADC_DRy, ADC1 y=0~23, ADC2 y=0~23, ADC3 y=0~24.....	436
17.4.16   模拟看门狗控制寄存器 ADC_AWDSCR .....	437
17.4.17   模拟看门狗状态寄存器 ADC_AWDSR.....	438
17.4.18   模拟看门狗状态复位寄存器 ADC_AWDSCLLR .....	438
17.4.19   模拟看门狗阀值寄存器 ADC_AWD0DR0、ADC_AWD0DR1、ADC_AWD1DR0、 ADC_AWD1DR1 .....	439
17.4.20   模拟看门狗比较通道选择寄存器 ADC_AWD0CHSR、ADC_AWD1CHSR .....	439
17.4.21   可编程增益放大器控制寄存器 ADC_PGACR .....	440
17.4.22   可编程增益放大器参考地选择寄存器 ADC_PGA_VSSENR .....	441
<b>17.5 使用注意事项 .....</b>	<b>442</b>
17.5.1    数据寄存器读取时注意事项 .....	442
17.5.2    扫描完成中断处理注意事项 .....	442
17.5.3    模块停止与低功耗设定的注意事项 .....	442
17.5.4    ADC 转换模拟通道输入的引脚设定 .....	442
17.5.5    噪声控制 .....	442
<b>18 数模转换器 (DAC) .....</b>	<b>443</b>

18.1 简介 .....	443
18.2 功能框图 .....	444
18.3 功能说明 .....	445
18.3.1 DAC 转换 .....	445
18.3.2 同步转换 .....	445
18.3.3 输出电压 .....	446
18.3.4 ADC 转换优先模式 .....	446
18.4 注意事项 .....	448
18.4.1 模块停止功能 .....	448
18.4.2 停止低功耗模式 .....	448
18.4.3 掉电低功耗模式 .....	448
18.4.4 使用输出缓冲器 .....	448
18.5 寄存器说明 .....	449
18.5.1 DAC 数据寄存器 (DAC <sub>x</sub> _DAD <sub>y</sub> , y=1~2) .....	450
18.5.2 DAC 控制寄存器 (DAC <sub>x</sub> _DSCR) .....	451
18.5.3 DAC 模拟输出控制寄存器 (DAC <sub>x</sub> _DAOCR) .....	452
18.5.4 DAC ADC 转换优先控制寄存器 (DAC <sub>x</sub> _DAADPCR) .....	453
<b>19 温度传感器 (OTS) .....</b>	<b>454</b>
19.1 简介 .....	454
19.2 使用说明 .....	455
19.3 寄存器说明 .....	457
19.3.1 OTS 控制寄存器 (OTS_CTL) .....	458
19.3.2 OTS 数据寄存器 1 (OTS_DR1) .....	458
19.3.3 OTS 数据寄存器 2 (OTS_DR2) .....	459
19.3.4 OTS 误差补偿寄存器 (OTS_ECR) .....	459
19.3.5 OTS 预置温度数据寄存器 (OTS_PDR1,2,3) .....	459
<b>20 高级控制定时器 (Timer6) .....</b>	<b>460</b>
20.1 简介 .....	460
20.2 基本框图 .....	460
20.3 功能说明 .....	462
20.3.1 波形模式 .....	462
20.3.2 时钟源选择 .....	463
20.3.3 计数方向 .....	463
20.3.4 比较输出 .....	464
20.3.5 捕获输入 .....	465

20.3.6	计数器刷新	466
20.3.7	软件同步	467
20.3.8	硬件同步	468
20.3.9	脉宽测量	471
20.3.10	周期测量	472
20.3.11	缓存功能	473
20.3.12	数字滤波	478
20.3.13	通用 PWM 输出	479
20.3.14	周期间隔响应	486
20.3.15	正交编码计数	487
20.3.16	EMB 控制	492
20.3.17	典型应用例	493
20.3.18	功能汇总表	499
20.4	中断及事件说明	500
20.4.1	中断输出	500
20.4.2	事件输出	500
20.5	寄存器说明	502
20.5.1	通用计数值寄存器 (TMR6_CNTER)	504
20.5.2	通用刷新值寄存器 (TMR6_UPDAR)	504
20.5.3	通用周期基准值寄存器 (TMR6_PERmR) (m=A~C)	504
20.5.4	通用比较基准值寄存器 (TMR6_GCMmR) (m=A~F)	505
20.5.5	专用比较基准值寄存器 (TMR6_SCmR) (m=A~F)	505
20.5.6	死区时间基准值寄存器 (TMR6_DTmnR) (m=D、U&&n=A、B)	505
20.5.7	通用控制寄存器 (TMR6_GCONR)	506
20.5.8	中断控制寄存器 (TMR6_ICONR)	507
20.5.9	缓存控制寄存器 (TMR6_BCONR)	508
20.5.10	死区控制寄存器 (TMR6_DCONR)	510
20.5.11	端口控制寄存器 (TMR6_PCNAR)	511
20.5.12	端口控制寄存器 (TMR6_PCNBR)	513
20.5.13	滤波控制寄存器 (TMR6_FCNGR)	515
20.5.14	滤波控制寄存器 (TMR6_FCNTR)	516
20.5.15	有效周期寄存器 (TMR6_VPERR)	517
20.5.16	状态标志寄存器 (TMR6_STFLR)	518
20.5.17	硬件启动事件选择寄存器 (TMR6_HSTAR)	520
20.5.18	硬件停止事件选择寄存器 (TMR6_HSTPR)	522

20.5.19 硬件清零事件选择寄存器 (TMR6_HCLRR) .....	524
20.5.20 硬件刷新事件选择寄存器 (TMR6_HUPDR) .....	526
20.5.21 硬件捕获事件选择寄存器 A (TMR6_HCPAR) .....	528
20.5.22 硬件捕获事件选择寄存器 B (TMR6_HCPBR) .....	530
20.5.23 硬件递加事件选择寄存器 (TMR6_HCUPR) .....	532
20.5.24 硬件递减事件选择寄存器 (TMR6_HCDOR) .....	534
20.5.25 软件同步启动控制寄存器 (TMR6_SSTAR) .....	536
20.5.26 软件同步停止控制寄存器 (TMR6_SSTPR) .....	537
20.5.27 软件同步清零控制寄存器 (TMR6_SCLRR) .....	538
20.5.28 软件同步刷新控制寄存器 (TMR6_SUPDR) .....	539
20.6 使用注意事项 .....	540
<b>21 通用控制定时器 (Timer4) .....</b>	<b>542</b>
21.1 简介 .....	542
21.2 基本框图 .....	542
21.3 功能说明 .....	544
21.3.1 基本动作 .....	544
21.3.2 缓存功能 .....	547
21.3.3 通用 PWM 输出 .....	552
21.3.4 周期间隔响应 .....	557
21.3.5 EMB 控制 .....	559
21.3.6 监测输出 .....	560
21.3.7 硬件启动 .....	561
21.4 中断及事件说明 .....	562
21.4.1 通用计数比较匹配中断及事件 .....	562
21.4.2 计数周期匹配中断及事件 .....	562
21.4.3 重载计数匹配中断及事件 .....	562
21.4.4 专用比较匹配事件 .....	562
21.5 寄存器说明 .....	564
21.5.1 计数值寄存器 (TMR4_CNTR) .....	566
21.5.2 周期基准寄存器 (TMR4_CPSR) .....	566
21.5.3 控制状态寄存器 (TMR4_CCSR) .....	567
21.5.4 有效周期寄存器 (TMR4_CVPR) .....	569
21.5.5 通用比较基准寄存器 (TMR4_OCCRm) .....	569
21.5.6 通用控制状态寄存器 (TMR4_OCSRn) .....	570
21.5.7 通用扩展控制寄存器 (TMR4_OCERn) .....	571

21.5.8 通用模式控制寄存器 (TMR4_OCMRm) .....	573
21.5.9 专用比较基准寄存器 (TMR4_SCCRm) .....	578
21.5.10 专用控制状态寄存器 (TMR4_SCSRm) .....	579
21.5.11 专用扩展控制寄存器 (TMR4_SCER) .....	581
21.5.12 专用模式控制寄存器 (TMR4_SCMRm) .....	581
21.5.13 PWM 基本控制寄存器 (TMR4_POCRn) .....	582
21.5.14 PWM 状态控制寄存器 (TMR4_PSCR) .....	583
21.5.15 PWM 滤波控制寄存器 (TMR4_PFSRn) .....	586
21.5.16 PWM 死区控制寄存器 (TMR4_PDA<B>Rn) .....	586
21.5.17 重载控制状态寄存器 (TMR4_RCSR) .....	587
21.5.18 专用中断控制寄存器 (TMR4_SCIR) .....	588
21.5.19 专用标志控制寄存器 (TMR4_SCFR) .....	588
<b>22 紧急刹车模块 (EMB) .....</b>	<b>589</b>
22.1 简介 .....	589
22.2 功能描述 .....	591
22.2.1 概述 .....	591
22.2.2 外部端口输入电平变化时控制 PWM 信号输出 .....	593
22.2.3 PWM 输出端口电平发生同相（同高或同低）时停止 PWM 信号输出 .....	594
22.2.4 根据电压比较器比较结果停止 PWM 信号输出 .....	594
22.2.5 系统错误发生时停止 PWM 信号输出 .....	595
22.2.6 写寄存器软件控制 PWM 信号输出 .....	595
22.3 寄存器说明 .....	596
22.3.1 EMB 控制寄存器 1_0~5(EMB_CTL1_0~5) .....	597
22.3.2 EMB 控制寄存器 1_6(EMB_CTL1_6) .....	599
22.3.3 EMB 控制寄存器 2_0~5(EMB_CTL2_0~5) .....	601
22.3.4 EMB 控制寄存器 2_6(EMB_CTL2_6) .....	603
22.3.5 EMB 软件输出使能控制寄存器(EMB_SOEx)(x=0~6) .....	605
22.3.6 EMB 状态寄存器(EMB_STATx) (x=0~6) .....	606
22.3.7 EMB 状态复位寄存器(EMB_STATCLRx) (x=0~6) .....	608
22.3.8 EMB 中断许可寄存器(EMB_INTENx)(x=0~6) .....	609
22.3.9 EMB 控制 PWM 输出释放方式选择寄存器(EMB_RLSSELx)(x=0~6) .....	610
<b>23 通用定时器 (TimerA) .....</b>	<b>611</b>
23.1 简介 .....	611
23.2 基本框图 .....	611
23.3 功能说明 .....	613

23.3.1	波形模式 .....	613
23.3.2	时钟源选择 .....	613
23.3.3	比较输出 .....	614
23.3.4	捕获输入 .....	615
23.3.5	同步启动 .....	616
23.3.6	数字滤波 .....	617
23.3.7	缓存功能 .....	617
23.3.8	级联计数 .....	618
23.3.9	PWM 输出 .....	619
23.3.10	正交编码计数 .....	620
23.3.11	红外功能 (IFRP) .....	624
23.4	中断及事件说明 .....	625
23.4.1	比较匹配中断及事件 .....	625
23.4.2	周期匹配中断及事件 .....	625
23.5	寄存器说明 .....	626
23.5.1	通用计数值寄存器 (TMRA_CNTER) .....	628
23.5.2	周期基准值寄存器 (TMRA_PERAR) .....	628
23.5.3	比较基准值寄存器 (TMRA_CMPARM) (m=1~4) .....	629
23.5.4	控制状态寄存器 L (TMRA_BCSTRL) .....	630
23.5.5	控制状态寄存器 H (TMRA_BCSTRH) .....	631
23.5.6	中断控制寄存器 (TMRA_ICONR) .....	631
23.5.7	事件控制寄存器 (TMRA_ECONR) .....	632
23.5.8	滤波控制寄存器 (TMRA_FCONR) .....	633
23.5.9	状态标志寄存器 (TMRA_STFLR) .....	634
23.5.10	缓存控制寄存器 (TMRA_BCONRM) (m=1~2) .....	635
23.5.11	捕获控制寄存器 (TMRA_CCONRM) (m=1~4) .....	636
23.5.12	端口控制寄存器 (TMRA_PCONRM) (m=1~4) .....	637
23.5.13	硬件触发事件选择寄存器 (TMRA_HCONR) .....	639
23.5.14	硬件递加事件选择寄存器 (TMRA_HCUPR) .....	641
23.5.15	硬件递减事件选择寄存器 (TMRA_HCDOR) .....	643
24	通用定时器 (Timer2) .....	645
24.1	简介 .....	645
24.2	基本框图 .....	645
24.3	功能说明 .....	647
24.3.1	时钟源选择 .....	647

24.3.2 比较输出 .....	648
24.3.3 硬件触发 .....	649
24.3.4 数字滤波 .....	651
24.4 中断及事件说明.....	652
24.4.1 中断输出 .....	652
24.4.2 事件输出 .....	652
24.5 寄存器说明.....	653
24.5.1 计数值寄存器 (TMR2_CNTmR) (m=A、B) .....	654
24.5.2 基准值寄存器 (TMR2_CMPmR) (m=A、B) .....	654
24.5.3 基本控制寄存器 (TMR2_BCONR) .....	655
24.5.4 中断控制寄存器 (TMR2_ICONR) .....	657
24.5.5 端口控制寄存器 (TMR2_PCONR) .....	658
24.5.6 硬件控制寄存器 (TMR2_HCONR) .....	660
24.5.7 状态标志寄存器 (TMR2_STFLR) .....	662
24.6 使用注意事项 .....	663
<b>25 通用定时器 (Timer0) .....</b>	<b>664</b>
25.1 简介 .....	664
25.2 基本框图 .....	664
25.3 功能说明 .....	665
25.3.1 时钟源选择 .....	665
25.3.2 基本计数 .....	666
25.3.3 硬件触发 .....	666
25.4 中断及事件说明.....	667
25.4.1 中断输出 .....	667
25.4.2 事件输出 .....	667
25.5 寄存器说明.....	668
25.5.1 计数值寄存器 (TMR0_CNTmR) (m=A~B) .....	669
25.5.2 基准值寄存器 (TMR0_CMPmR) (m=A~B) .....	669
25.5.3 基本控制寄存器 (TMR0_BCONR) .....	670
25.5.4 状态标志寄存器 (TMR0_STFLR) .....	673
25.6 使用注意事项 .....	674
<b>26 实时时钟 (RTC) .....</b>	<b>675</b>
26.1 简介 .....	675
26.2 基本框图 .....	676
26.3 功能说明 .....	677

26.3.1	上电设定 .....	677
26.3.2	RTC 计数开始设定.....	677
26.3.3	系统低功耗模式切换.....	677
26.3.4	读出计数寄存器.....	677
26.3.5	写入计数寄存器.....	678
26.3.6	闹钟设定 .....	678
26.3.7	时钟误差补偿 .....	678
26.3.8	1Hz 输出 .....	678
26.4	中断说明 .....	680
26.4.1	闹钟中断 .....	680
26.4.2	定周期中断.....	680
26.5	寄存器说明.....	681
26.5.1	控制寄存器 0 (RTC_CRO) .....	682
26.5.2	控制寄存器 1 (RTC_CR1) .....	682
26.5.3	控制寄存器 2 (RTC_CR2) .....	683
26.5.4	控制寄存器 3 (RTC_CR3) .....	684
26.5.5	秒计数寄存器 (RTC_SEC) .....	684
26.5.6	分计数寄存器 (RTC_MIN) .....	685
26.5.7	时计数寄存器 (RTC_HOUR) .....	685
26.5.8	日计数寄存器 (RTC_DAY) .....	687
26.5.9	周计数寄存器 (RTC_WEEK) .....	687
26.5.10	月计数寄存器 (RTC_MON) .....	688
26.5.11	年计数寄存器 (RTC_YEAR) .....	688
26.5.12	分闹钟寄存器 (RTC_ALMMIN) .....	688
26.5.13	时闹钟寄存器 (RTC_ALMHOUR) .....	689
26.5.14	周闹钟寄存器 (RTC_ALM WEEK) .....	689
26.5.15	时钟误差补偿寄存器 (RTC_ERRCRH、RTC_ERRCRL) .....	690
27	看门狗计数器 (WDT/ SWDT) .....	692
27.1	简介 .....	692
27.2	功能说明 .....	693
27.2.1	启动看门狗 .....	693
27.2.2	硬件启动方式 .....	693
27.2.3	软件启动方式 .....	694
27.2.4	刷新动作 .....	695
27.2.5	标志位 .....	695

27.2.6 中断复位 .....	695
27.2.7 计数下溢 .....	696
27.2.8 刷新错误 .....	697
27.3 寄存器说明.....	698
27.3.1 控制寄存器 (SWDT_CR、WDT_CR) .....	699
27.3.2 状态寄存器 (SWDT_SR、WDT_SR) .....	701
27.3.3 刷新寄存器 (SWDT_RR、WDT_RR) .....	701
27.4 使用注意事项 .....	702
<b>28 通用同步异步收发器 (USART) .....</b>	<b>703</b>
28.1 简介 .....	703
28.2 USART 系统框图.....	705
28.3 管脚说明 .....	705
28.4 功能说明 .....	706
28.4.1 UART .....	706
28.4.2 多处理器通信 .....	716
28.4.3 UART-LIN .....	720
28.4.4 智能卡 .....	723
28.4.5 时钟同步模式 .....	727
28.4.6 数字滤波功能 .....	732
28.4.7 中断 .....	733
28.5 寄存器说明.....	734
28.5.1 状态寄存器 (USART_SR) .....	735
28.5.2 发送数据寄存器 (USART_TDR) .....	738
28.5.3 接收数据寄存器 (USART_RDR) .....	738
28.5.4 波特率寄存器 (USART_BRR) .....	739
28.5.5 控制寄存器 1 (USART_CR1) .....	740
28.5.6 控制寄存器 2 (USART_CR2) .....	743
28.5.7 控制寄存器 3 (USART_CR3) .....	745
28.5.8 预分频寄存器 (USART_PR) .....	747
28.5.9 LIN 波特率测量计数寄存器 (USART_LBMC) .....	748
28.5.10 USART1 滤波控制寄存器 (USART1_NFC) .....	749
28.6 使用注意事项 .....	750
28.6.1 UART 注意事项 .....	750
28.6.2 时钟同步模式注意事项 .....	750
28.6.3 其他注意事项 .....	750

<b>29 集成电路总线 (I2C) .....</b>	<b>751</b>
29.1 简介 .....	751
29.2 I2C 系统框图 .....	752
29.2.1 系统框图 .....	752
29.2.2 结构图 .....	753
29.3 动作说明 .....	754
29.3.1 I2C 协议 .....	754
29.3.2 地址匹配 .....	763
29.3.3 SMBus 动作 .....	768
29.3.4 复位 .....	769
29.3.5 中断和事件信号输出 .....	770
29.3.6 可编程数字滤波 .....	772
29.4 应用软件设定 I2C 初始化流程 .....	772
29.5 寄存器说明 .....	773
29.5.1 I2C 控制寄存器 1(I2C_CR1) .....	774
29.5.2 I2C 控制寄存器 2(I2C_CR2) .....	776
29.5.3 I2C 控制寄存器 3(I2C_CR3) .....	778
29.5.4 I2C 控制寄存器 4(I2C_CR4) .....	779
29.5.5 I2C 从机地址寄存器 0(I2C_SLR0) .....	780
29.5.6 I2C 从机地址寄存器 1(I2C_SLR1) .....	781
29.5.7 I2C SCL 电平超时控制寄存器(I2C_SLTR) .....	782
29.5.8 I2C 状态寄存器(I2C_SR) .....	783
29.5.9 I2C 状态清零寄存器(I2C_CLR) .....	788
29.5.10 I2C 数据发送寄存器(I2C_DTR) .....	789
29.5.11 I2C 数据接收寄存器(I2C_DRR) .....	790
29.5.12 I2C 数据移位寄存器(I2C_DSR) .....	790
29.5.13 I2C 时钟控制寄存器(I2C_CCR) .....	791
29.5.14 I2C 滤波控制寄存器(I2C_FLTR) .....	793
29.5.15 I2CFIFO 控制寄存器 (I2C_FSTR) .....	794
29.5.16 I2C 从机地址接收寄存器 (I2C_SLVADRR) .....	794
<b>30 串行外设接口 (SPI) .....</b>	<b>795</b>
30.1 简介 .....	795
30.2 SPI 系统框图 .....	796
30.3 管脚说明 .....	796
30.4 SPI 动作系统说明 .....	797

30.4.1 主机模式的管脚状态.....	797
30.4.2 从机模式的管脚状态.....	797
30.4.3 SPI 系统连接实例.....	798
<b>30.5 数据通信说明 .....</b>	<b>799</b>
30.5.1 波特率.....	799
30.5.2 数据格式 .....	800
30.5.3 传送格式 .....	803
30.5.4 通信方式 .....	805
30.5.5 通信间隔时间 .....	807
<b>30.6 运行说明 .....</b>	<b>809</b>
30.6.1 运行模式概要 .....	809
30.6.2 SPI 运行模式时的主机动作.....	810
30.6.3 SPI 运行模式时的从机动作.....	811
30.6.4 时钟同步运行模式中的主机动作 .....	812
30.6.5 时钟同步运行模式中的从机动作 .....	813
30.6.6 几种 SPI 动作的处理流程 .....	814
<b>30.7 奇偶校验位自诊断 .....</b>	<b>815</b>
<b>30.8 错误检测 .....</b>	<b>816</b>
30.8.1 欠载错误 .....	816
30.8.2 模式错误 .....	817
30.8.3 过载错误 .....	817
30.8.4 奇偶校验错误 .....	819
<b>30.9 SPI 的初始化 .....</b>	<b>820</b>
30.9.1 清除 SPE 位进行初始化.....	820
30.9.2 系统复位初始化.....	820
<b>30.10 中断 .....</b>	<b>821</b>
<b>30.11 可供使用的事件触发源 .....</b>	<b>821</b>
<b>30.12 寄存器说明.....</b>	<b>822</b>
30.12.1 SPI 数据寄存器 (SPI_DR) .....	822
30.12.2 SPI 控制寄存器 (SPI_CR) .....	823
30.12.3 SPI 通信配置寄存器 1 (SPI_CFG1) .....	824
30.12.4 SPI 状态寄存器 (SPI_SR) .....	826
30.12.5 SPI 通信配置寄存器 2 (SPI_CFG2) .....	827
<b>31 四线式串行外设接口 (QSPI) .....</b>	<b>829</b>
31.1 简介 .....	829

31.2 内存映射 .....	831
31.2.1 内部总线空间 .....	831
31.2.2 ROM 空间及总线的地址宽度 .....	832
31.3 QSPI 总线 .....	833
31.3.1 SPI 协议 .....	833
31.3.2 SPI 模式 .....	835
31.4 QSPI 总线的时序调整 .....	836
31.4.1 QSPI 总线基准时钟 .....	836
31.4.2 SPI 总线基准时钟 .....	837
31.4.3 QSSN 信号最小高电平宽度 .....	837
31.4.4 QSSN 的建立时间 .....	838
31.4.5 QSSN 的保持时间 .....	838
31.4.6 串行数据接收延迟 .....	839
31.5 用于 ROM 访问的 SPI 指令介绍 .....	840
31.5.1 现有的 QSPI-ROM 指令参考 .....	840
31.5.2 标准读指令 .....	840
31.5.3 快速读指令 .....	841
31.5.4 二线式输出快速读指令 .....	842
31.5.5 二线式输入输出快速读指令 .....	844
31.5.6 四线式输出快速读指令 .....	845
31.5.7 四线式输入输出快速读指令 .....	846
31.5.8 进入 4-Byte 模式指令 .....	847
31.5.9 退出 4-Byte 模式指令 .....	847
31.5.10 写许可指令 .....	848
31.6 QSPI 总线周期的安排 .....	848
31.6.1 独立转换的单一闪存读取 .....	848
31.6.2 使用预读取功能进行闪存读取 .....	848
31.6.3 预读取的终止 .....	849
31.6.4 预读取状态监测 .....	849
31.6.5 使用 QSPI 总线周期延长功能进行闪存读取 .....	850
31.7 XIP 控制 .....	850
31.7.1 XIP 模式的设置 .....	851
31.7.2 XIP 模式的退出 .....	851
31.8 QSIO2 和 QSIO3 的管脚状态 .....	852
31.9 直接通信模式 .....	852

31.9.1	关于直接通信模式 .....	852
31.9.2	直接通信模式的设置 .....	852
31.9.3	直接通信模式下 QSPI 总线周期的生成 .....	853
31.10	中断 .....	853
31.11	使用上的注意事项 .....	853
31.11.1	QSPI 寄存器的设置顺序 .....	853
31.11.2	模块停止信号的设置 .....	853
31.12	寄存器说明 .....	854
31.12.1	QSPI 控制寄存器 (QSCR) .....	855
31.12.2	QSPI 片选控制寄存器 (QSCSCR) .....	858
31.12.3	QSPI 格式控制寄存器 (QSFCR) .....	859
31.12.4	QSPI 状态寄存器 (QSSR) .....	861
31.12.5	QSPI 指令代码寄存器 (QSCCMD) .....	863
31.12.6	QSPI 直接通信指令寄存器 (QSDCOM) .....	863
31.12.7	QSPI XIP 模式代码寄存器 (QSXCMD) .....	864
31.12.8	QSPI 标志清除寄存器 (QSSR2) .....	864
31.12.9	QSPI 外部扩展地址寄存器 (QSEXAR) .....	865
<b>32</b>	<b>可编程逻辑阵列 (PLA)</b> .....	<b>866</b>
32.1	简介 .....	866
32.2	PLA 模块框图 .....	867
32.2.1	模块框图 .....	867
32.2.2	动作说明 .....	868
32.2.3	PLA 输入多路选择器 .....	869
32.2.4	查找表 (LUT) 控制 .....	871
32.2.5	PLA 输出 .....	872
32.3	寄存器说明 .....	873
32.3.1	全局控制寄存器 (PLA_GCTL) .....	874
32.3.2	状态寄存器 (PLA_STAT) .....	875
32.3.3	多路选择器选择寄存器 (PLAx_MUXS) ( $x=0\ldots15$ ) .....	876
32.3.4	查找表控制寄存器 (PLAx_LUTCTL) ( $x=0\ldots15$ ) .....	877
32.3.5	控制寄存器 (PLAx_CTL) ( $x=0\ldots15$ ) .....	877
32.4	注意事项 .....	878
<b>33</b>	<b>高精度参考电压 (VREF)</b> .....	<b>879</b>
33.1	简介 .....	879
33.2	功能描述 .....	879

33.3 寄存器说明.....	882
33.3.1 BGR 控制寄存器 (VREF_BGRCR) .....	882
33.3.2 电流源控制寄存器 (VREF_CSCR) .....	883
33.3.3 VREF 控制寄存器 (VREF_CR) .....	884
<b>34 USB2.0 全速模块 (USBFS) .....</b>	<b>885</b>
34.1 USBFS 简介 .....	885
34.2 USBFS 主要特性 .....	885
34.2.1 通用特性 .....	885
34.2.2 主机模式特性 .....	886
34.2.3 设备模式特性 .....	886
34.3 USBFS 系统框图 .....	887
34.4 USBFS 管脚说明 .....	887
34.5 USBFS 功能说明 .....	888
34.5.1 USBFS 时钟以及工作模式 .....	888
34.5.2 USBFS 模式决定 .....	888
34.5.3 USBFS 主机功能 .....	888
34.5.4 USBFS 设备功能 .....	893
34.5.5 USBFS SOF 脉冲管脚输出功能 .....	896
34.5.6 USBFS 功耗控制 .....	897
34.5.7 USBFS 动态更新 USBFS_HFIR 寄存器 .....	897
34.5.8 USBFS 数据 FIFO .....	898
34.5.9 USBFS 主机 FIFO 架构 .....	898
34.5.10 USBFS 设备 FIFO 架构 .....	900
34.5.11 USBFS FIFO RAM 分配 .....	901
34.5.12 USBFS 系统性能 .....	902
34.5.13 USBFS 中断和事件 .....	903
34.6 USBFS 编程模型 .....	904
34.6.1 USBFS 模块初始化 .....	904
34.6.2 USBFS 主机初始化 .....	904
34.6.3 USBFS 设备初始化 .....	905
34.6.4 USBFS DMA 模式 .....	905
34.6.5 USBFS 主机编程模型 .....	906
34.6.6 USBFS 设备编程模型 .....	907
34.6.7 USBFS 操作模型 .....	909
34.7 寄存器说明 .....	924

34.7.1	USBFS 系统控制寄存器.....	928
34.7.2	USBFS 全局寄存器.....	929
34.7.3	USBFS 主机模式寄存器.....	956
34.7.4	USBFS 设备模式寄存器.....	971
34.7.5	USBFS 时钟门控控制寄存器.....	998
<b>35</b>	<b>CAN FD 控制器 (CAN FD) .....</b>	<b>999</b>
35.1	简介 .....	999
35.2	CAN FD 系统框图 .....	1000
35.3	管脚说明 .....	1000
35.4	功能说明 .....	1001
35.4.1	动作模式 .....	1001
35.4.2	波特率设定 .....	1001
35.4.3	发送缓冲器 .....	1004
35.4.4	接收缓冲器 .....	1004
35.4.5	接收筛选寄存器组 .....	1005
35.4.6	数据发送 .....	1006
35.4.7	单次数据发送 .....	1006
35.4.8	取消数据发送 .....	1006
35.4.9	数据接收 .....	1007
35.4.10	错误处理 .....	1007
35.4.11	节点关闭 .....	1008
35.4.12	仲裁失败位置捕捉 .....	1008
35.4.13	回环模式 .....	1008
35.4.14	静默模式 .....	1009
35.4.15	软件复位功能 .....	1010
35.4.16	向上兼容 CAN-FD 功能 .....	1011
35.4.17	时间触发 TTCAN .....	1011
35.4.18	TDC 和 RDC .....	1014
35.4.19	中断 .....	1014
35.5	寄存器说明 .....	1015
35.5.1	CAN 全局控制寄存器 (CAN_SYCTLREG) .....	1016
35.5.2	CAN 接收 BUF 寄存器 (CAN_RBUF) .....	1017
35.5.3	CAN 发送 BUF 寄存器 (CAN_TBUF) .....	1020
35.5.4	CAN 配置和状态寄存器 (CAN_CFG_STAT) .....	1022
35.5.5	CAN 命令寄存器 (CAN_TCMD) .....	1023

35.5.6	CAN 发送控制寄存器 (CAN_TCTRL) .....	1025
35.5.7	CAN 接收控制寄存器 (CAN_RCTRL) .....	1027
35.5.8	CAN 接收和发送中断使能寄存器 (CAN_RTIE) .....	1028
35.5.9	CAN 接收和发送中断标志寄存器 (CAN_RTIF) .....	1029
35.5.10	CAN 错误中断使能和标志寄存器 (CAN_ERRINT) .....	1031
35.5.11	CAN Slow 位时序寄存器 (CAN_SBT) .....	1032
35.5.12	CAN Fast 位时序寄存器 (CAN_FBT) .....	1033
35.5.13	CAN 发送器延迟补偿寄存器 (CAN_TDC) .....	1034
35.5.14	CAN 错误和仲裁失败捕捉寄存器 (CAN_EALCAP) .....	1034
35.5.15	CAN 警告限定寄存器 (CAN_LIMIT) .....	1035
35.5.16	CAN 接收错误计数器寄存器 (CAN_RECNT) .....	1035
35.5.17	CAN 发送错误计数器寄存器 (CAN_TECNT) .....	1036
35.5.18	CAN 筛选器组控制寄存器 (CAN_ACFCTRL) .....	1037
35.5.19	CAN 筛选器组使能寄存器 (CAN_ACFEN) .....	1038
35.5.20	CAN 筛选器组 code 和 mask 寄存器 (CAN_ACF) .....	1039
35.5.21	TTCAN TB slot 指针寄存器 (CAN_TBSLOT) .....	1040
35.5.22	TTCAN 时间触发配置寄存器 (CAN_TTCFG) .....	1041
35.5.23	TTCAN 参考消息寄存器 (CAN_REF_MSG) .....	1042
35.5.24	TTCAN 触发配置寄存器 (CAN_TRG_CFG) .....	1043
35.5.25	TTCAN 触发时间寄存器 (CAN_TT_TRIG) .....	1044
35.5.26	TTCAN 触发看门时间寄存器 (CAN_TT_WTRIG) .....	1044
35.6	使用注意事项 .....	1045
35.6.1	CAN 总线抗干扰措施 .....	1045
35.6.2	CAN 控制器噪声制约 .....	1045
<b>36</b>	<b>MDIO 串行总线接口 (MDIO) .....</b>	<b>1046</b>
36.1	简介 .....	1046
36.2	模块示意图 .....	1046
36.3	功能描述 .....	1046
36.3.1	MDIO 帧结构 .....	1046
36.4	应用举例 .....	1048
36.4.1	应用框图 .....	1048
36.5	寄存器说明 .....	1051
36.5.1	MDIO 模块控制寄存器(MDCON) .....	1052
36.5.2	MDIO 模块控制寄存器 1(MDCON1) .....	1053
36.5.3	MDIO 接收帧控制信息寄存器(MDFRM) .....	1054

36.5.4	MDIO 接收数据寄存器(MDRXD) .....	1055
36.5.5	MDIO 接收地址寄存器(MDADR) .....	1056
36.5.6	MDIO 传输数据寄存器(MDTXD) .....	1057
36.5.7	MDIO 物理地址寄存器(MDPHY).....	1058
36.5.8	MDIO 进程信号寄存器(MDSTA) .....	1059
36.5.9	MDIO 中断许可寄存器(MDIEN).....	1061
36.5.10	MDIO 读取物理地址引脚寄存器(MDPIN).....	1062
36.5.11	超时寄存器 (MDTO) .....	1063
36.6	使用注意事项 .....	1064
<b>37</b>	<b>外部存储器控制器 (EXMC)</b> .....	<b>1065</b>
37.1	概要 .....	1065
37.2	基本特性 .....	1065
37.2.1	功能列表 .....	1065
37.2.2	控制器架构 .....	1066
37.2.3	基本访问规范 .....	1067
37.2.4	地址映射 .....	1068
37.2.5	协议接口 .....	1070
37.3	功能说明 .....	1071
37.3.1	SMC-SRAM/PSRAM/NOR Flash 控制器 .....	1071
37.4	寄存器说明 .....	1090
37.4.1	SMC-SRAM/PSRAM/NOR Flash 寄存器 .....	1091
37.5	使用注意事项 .....	1099
<b>38</b>	<b>加密协处理模块 (CPM)</b> .....	<b>1100</b>
38.1	简介 .....	1100
38.2	加解密算法处理器 (AES) .....	1100
38.2.1	算法简介 .....	1100
38.2.2	AES 模块功能描述 .....	1102
38.2.3	加密操作流程 .....	1102
38.2.4	解密操作流程 .....	1102
38.2.5	数据示例 .....	1103
38.2.6	运行时间说明 .....	1106
38.2.7	操作注意事项 .....	1106
38.2.8	寄存器说明 .....	1107
38.3	安全散列算法 (HASH) .....	1111
38.3.1	算法简介 .....	1111

38.3.2 操作流程 .....	1112
38.3.3 消息填充 .....	1113
38.3.4 HMAC 运算 .....	1114
38.3.5 中断说明 .....	1117
38.3.6 硬件触发事件选择 .....	1117
38.3.7 寄存器说明 .....	1118
38.4 真随机数发生器 (TRNG) .....	1123
38.4.1 模块框图 .....	1123
38.4.2 操作流程 .....	1123
38.4.3 中断及事件输出 .....	1123
38.4.4 寄存器说明 .....	1124
<b>39 CRC 运算 (CRC) .....</b>	<b>1126</b>
39.1 简介 .....	1126
39.2 功能框图 .....	1126
39.3 功能说明 .....	1126
39.3.1 CRC16 编码模式 .....	1127
39.3.2 CRC16 校验模式 .....	1127
39.3.3 CRC32 编码模式 .....	1127
39.3.4 CRC32 校验模式 .....	1127
39.4 寄存器说明 .....	1128
39.4.1 控制寄存器 (CRC_CR) .....	1128
39.4.2 结果寄存器 (CRC_RESLT) .....	1128
39.4.3 数据寄存器 (CRC_DAT) .....	1129
<b>40 滤波数学加速器 (FMAC) .....</b>	<b>1130</b>
40.1 简介 .....	1130
40.2 基本框图 .....	1130
40.3 操作流程 .....	1131
40.4 模块使能 .....	1131
40.5 系数归一化 .....	1131
40.6 中断及事件说明 .....	1132
40.6.1 中断输出 .....	1132
40.6.2 事件输出 .....	1132
40.7 寄存器说明 .....	1133
40.7.1 模块使能寄存器 (FMAC_ENR) .....	1133
40.7.2 基本控制寄存器 (FMAC_CTR) .....	1134

40.7.3	中断控制寄存器 (FMAC_IER) .....	1134
40.7.4	数据输入寄存器 (FMAC_DTR) .....	1135
40.7.5	滤波系数寄存器 (FMAC_COR0~16) .....	1135
40.7.6	结果输出寄存器 0 (FMAC_RTR0) .....	1135
40.7.7	结果输出寄存器 1 (FMAC_RTR1) .....	1136
40.7.8	运算状态寄存器 (FMAC_STR) .....	1136
<b>41</b>	<b>数据计算单元 (DCU) .....</b>	<b>1137</b>
41.1	概要 .....	1137
41.2	功能描述 .....	1138
41.2.1	加法模式 .....	1138
41.2.2	减法模式 .....	1138
41.2.3	硬件触发启动模式 .....	1138
41.2.4	比较模式 .....	1139
41.2.5	中断和事件信号输出 .....	1140
41.3	寄存器说明 .....	1141
41.3.1	DCU 控制寄存器(DCUx_CTL)(x=1~3) .....	1142
41.3.2	DCU 标志寄存器(DCUx_FLAG)(x=1~3) .....	1143
41.3.3	DCU 数据寄存器(DCUx_DATAy)(x=1~3,y=0,1,2) .....	1144
41.3.4	DCU 标志复位寄存器(DCUx_FLAGCLR) (x=1~3) .....	1145
41.3.5	DCU 中断和事件寄存器(DCUx_INTEVTSEL) (x=1~3) .....	1146
<b>42</b>	<b>数学运算单元 (MAU) .....</b>	<b>1148</b>
42.1	简介 .....	1148
42.2	功能说明 .....	1148
42.2.1	开方运算 .....	1148
42.2.2	正弦运算 .....	1149
42.3	中断及事件说明 .....	1149
42.3.1	中断输出 .....	1149
42.3.2	事件输出 .....	1149
42.4	寄存器说明 .....	1150
42.4.1	控制状态寄存器 (MAU_CSR) .....	1150
42.4.2	数据输入寄存器 0 (MAU_DTR0) .....	1151
42.4.3	结果输出寄存器 0 (MAU_RTR0) .....	1151
42.4.4	数据输入寄存器 1 (MAU_DTR1) .....	1151
42.4.5	结果输出寄存器 1 (MAU_RTR1) .....	1152
<b>43</b>	<b>调试控制器 (DBG) .....</b>	<b>1153</b>

43.1 简介 .....	1153
43.2 DBGC 系统框图 .....	1153
43.3 SWJ-DP 调试端口 (SWD 和 JTAG) .....	1154
43.3.1 JTAG-DP 或 SW-DP 的切换机制 .....	1155
43.4 引脚排列和调试端口引脚 .....	1155
43.4.1 SWJ 调试端口引脚 .....	1155
43.4.2 灵活的 SWJ-DP 引脚分配 .....	1155
43.4.3 JTAG 引脚上的内部上拉 .....	1156
43.4.4 使用串行接口以及释放未使用的调试引脚以作 GPIO .....	1156
43.5 寄存器说明 .....	1157
43.5.1 芯片专属标志寄存器 (CHIPID) .....	1157
43.5.2 DBG 状态寄存器 (MCUDBGSTAT) .....	1158
43.5.3 外设调试暂停寄存器 (MCUSTPCTL) .....	1159
43.5.4 调试组件配置寄存器 (MCUTRACECTL) .....	1161
43.5.5 外设调试暂停寄存器 2 (MCUSTPCTL2) .....	1162
43.6 SW 调试端口 .....	1164
43.6.1 SW 协议简介 .....	1164
43.7 TPIU (跟踪端口接口单元) .....	1164
43.7.1 简介 .....	1164
43.7.2 TRACE 引脚分配 .....	1165
43.7.3 MCU 内部 TRACECLKIN 连接 .....	1166
43.7.4 TPIU 寄存器 .....	1166
43.7.5 TPIU 配置示例 .....	1166
<b>版本修订记录 .....</b>	<b>1167</b>

## 表索引

表 1-1 存储器映射 .....	53
表 1-2 QSPI 地址空间分配 .....	59
表 1-3 目标地址配置示例 .....	60
表 1-4 寄存器列表 .....	61
表 3-1 复位方式及产生条件 .....	67
表 3-2 复位方式及复位标志 .....	68
表 3-3 RMU 寄存器一览 .....	77
表 4-1 时钟源主要特性 .....	89
表 4-2 各个内部时钟的规格 .....	90
表 4-3 寄存器一览 .....	106
表 5-1 BOR 配置 .....	137
表 5-2 PVD1/ PVD2 特性 .....	138
表 5-3 运行模式 .....	144
表 5-4 低功耗模式 .....	144
表 5-5 低功耗模式的运行条件及各模块在低功耗模式下的状态 .....	145
表 5-6 掉电模式子模式 .....	149
表 5-7 RAM 模块与 RAM 掉电控制位 .....	152
表 5-8 寄存器保护列表 .....	153
表 5-9 寄存器一览表 .....	154
表 6-1 寄存器一览表 .....	181
表 7-1 CPU 时钟频率和 Flash 读等待周期对照表 .....	189
表 7-2 Flash 实际读周期数 .....	190
表 7-3 OTP 地址分布表 .....	197
表 7-4 寄存器一览表 .....	201
表 8-1 SRAM 空间分配 .....	216
表 8-2 SRAM 寄存器一览 .....	218
表 9-1 PORT 寄存器一览 1 .....	231
表 9-2 PORT 寄存器一览 2 .....	231
表 9-3 32bit 访问时 PORT 寄存器一览 .....	244
表 10-1 外部中断输入管脚 .....	246
表 10-2 中断向量表 .....	248
表 10-3 中断事件表 .....	254
表 10-4 INTC 寄存器列表 .....	273

表 11-1 AOS 目标列表 .....	286
表 11-2 寄存器列表 .....	289
表 12-1 寄存器列表 .....	307
表 13-1 KEYS CAN 管脚说明 .....	327
表 13-2 KEYS CAN 寄存器一览表 .....	329
表 14-1 HRC 目标频率为 20MHz 时的测量误差 .....	334
表 14-2 HRC 目标频率为 16MHz 时的测量误差 .....	337
表 14-3 CTC 寄存器一览表 .....	341
表 15-1 通道重置说明 .....	353
表 15-2 寄存器一览表 .....	360
表 16-1 CMP 输入输出引脚 .....	382
表 16-2 CMP 模拟输入对照表 (HC32F472KCHI/HC32F472KEHI) .....	384
表 16-3 CMP 模拟输入对照表 (HC32F472JCTI/HC32F472JETI/HC32F472PCTI/HC32F472PETI) .....	385
表 16-4 CMP 寄存器一览表 .....	393
表 16-5 定时器窗口 PWM 一览表 .....	399
表 17-1 各 ADC 单元规格 (HC32F472KCHI/HC32F472KEHI) .....	404
表 17-2 各 ADC 单元规格 (HC32F472JCTI/HC32F472JETI/HC32F472PCTI/HC32F472PETI) .....	405
表 17-3 序列 A 和 B 的各种竞争 .....	411
表 17-4 AD 转换时间 .....	414
表 17-5 ADC 寄存器一览 .....	423
表 18-1 DAC 输入输出引脚 .....	444
表 18-2 DACx 寄存器一览表 .....	449
表 18-3 DAC 转换和模拟输出控制 .....	452
表 19-1 OTS 预置温度数据 .....	456
表 19-2 Ehrc 使用及设定方法 .....	456
表 19-3 OTS 寄存器一览表 .....	457
表 20-1 Timer6 的基本功能及特性 .....	460
表 20-2 Timer6 端口列表 .....	461
表 20-3 不同模式下的功能对比表 .....	499
表 20-4 Timer6 寄存器列表 .....	502
表 20-5 计数器 (CNTER) 控制优先级 .....	540
表 20-6 PWMA 端口输出控制优先级 .....	540
表 20-7 PWMB 端口输出控制优先级 .....	541
表 21-1 Timer4 的基本功能及特性 .....	542
表 21-2 Timer4 端口列表 .....	543

表 21-3 寄存器列表 .....	564
表 21-4 PWM 端口输出状态与寄存器设定值.....	585
表 22-1 端口分配表 .....	593
表 22-2 EMB group 对照表 .....	594
表 22-3 寄存器一览 .....	596
表 23-1 TimerA 的基本功能及特性.....	611
表 23-2 TimerA 端口列表.....	612
表 23-3 TimerA 寄存器列表 .....	626
表 23-4 内部触发事件 HTSSR 选择关系对应表 .....	627
表 24-1 Timer2 的基本功能及特性.....	645
表 24-2 Timer2 端口列表.....	647
表 24-3 Timer2 寄存器列表 .....	653
表 25-1 Timer0 寄存器列表 .....	668
表 26-1 RTC 的基本规格 .....	675
表 26-2 寄存器列表 .....	681
表 27-1 看门狗计数器的基本特性 .....	692
表 27-2 寄存器列表 .....	698
表 28-1 USART 管脚说明 .....	705
表 28-2 DIV_Fraction 为 0 时 UART 接收器的容差 .....	711
表 28-3 DIV_Fraction 不为 0 时 UART 接收器的容差 .....	711
表 28-4 UART 中断/事件表 .....	716
表 28-5 多处理器模式中断/事件表 .....	720
表 28-6 LIN 中断/事件表 .....	722
表 28-7 智能卡模式中断/事件表 .....	727
表 28-8 时钟同步模式中断/事件表 .....	732
表 28-9 USART 整体中断一览表 .....	733
表 28-10 USART 寄存器一览表 .....	734
表 28-11 波特率计算公式（小数波特率无效 FBME=0） .....	739
表 28-12 波特率计算公式（小数波特率有效 FBME=1） .....	739
表 29-1 输入/ 输出引脚 .....	753
表 29-2 中断一览表 .....	770
表 29-3 事件信号输出一览表 .....	771
表 29-4 寄存器一览表 .....	773
表 30-1 SPI 特性要点 .....	795
表 30-2 管脚说明.....	796

表 30-3 主机模式时 SPI 管脚状态说明.....	797
表 30-4 从机模式时 SPI 管脚状态说明.....	797
表 30-5 部分设定值的位速.....	799
表 30-6 SPI 模式和寄存器设定关系 .....	809
表 30-7 错误检测对应表 .....	816
表 30-8 SPI 中断源说明 .....	821
表 30-9 寄存器一览 .....	822
表 31-1 QSPI 主要规格 .....	829
表 31-2 QSPI 管脚 .....	830
表 31-3 QSPI 总线基准时钟选择一览表 .....	836
表 31-4 参考指令一览表 .....	840
表 31-5 QIO2 和 QIO3 的管脚状态 .....	852
表 31-6 QSPI 寄存器一览表 .....	854
表 32-1 PLAxMUX0 输入选择.....	869
表 32-2 PLAxMUX1 输入选择.....	870
表 32-3 LUT 控制 .....	871
表 32-4 PLA 输出端口对应表 .....	872
表 32-5 寄存器一览 .....	873
表 33-1 AVCCMON.....	880
表 33-2 VREF 电路模式.....	881
表 33-3 寄存器一览表 .....	882
表 34-1 USBFS 管脚说明.....	887
表 34-2 USBFS_GLB 中断事件表 .....	903
表 34-3 USBFS 系统控制寄存器一览表.....	925
表 34-4 USBFS 时钟门控控制寄存器一览表.....	926
表 35-1 CAN 管脚说明 .....	1000
表 35-2 CAN 位时间设定规则.....	1002
表 35-3 20MHz 通信时钟时波特率设定建议 .....	1003
表 35-4 40MHz 通信时钟时波特率设定建议 .....	1003
表 35-5 80MHz 通信时钟时波特率设定建议 .....	1003
表 35-6 软件复位范围表 .....	1010
表 35-7 CAN 中断表.....	1014
表 35-8 CAN 寄存器一览表 .....	1015
表 35-9 标准格式 CAN 接收邮箱格式 .....	1017
表 35-10 扩展格式 CAN 接收邮箱格式 .....	1018

表 35-11 DLC 控制位 .....	1019
表 35-12 标准格式 CAN 发送邮箱格式 .....	1020
表 35-13 扩展格式 CAN 发送邮箱格式 .....	1021
表 36-1 MDIO 不同类型帧 .....	1047
表 36-2 MDIO 操作码 .....	1048
表 36-3 MDIO 寄存器一览表 .....	1051
表 37-1 EXMC 的基本功能 .....	1065
表 37-2 AHB 访问宽度和存储器位宽对应访问方式表 .....	1067
表 37-3 SMC 协议接口 .....	1070
表 37-4 EXMC 端口功能分配 .....	1070
表 37-5 单次读动作基本设定例 .....	1074
表 37-6 地址数据线复用单次读动作基本设定例 .....	1079
表 37-7 单次写动作基本设定例 .....	1080
表 37-8 地址数据线复用单次写动作基本设定例 .....	1085
表 37-9 地址数据线复用同步方式单次写动作基本设定例 .....	1087
表 37-10 EXMC 寄存器列表 .....	1090
表 38-1 128 位操作寄存器示例 .....	1103
表 38-2 192 位操作寄存器示例 .....	1104
表 38-3 256 位操作寄存器示例 .....	1105
表 38-4 AES 加解密运行时间 .....	1106
表 38-5 寄存器列表 .....	1107
表 38-6 寄存器一览 .....	1118
表 38-7 寄存器一览 .....	1124
表 39-1 CRC 寄存器列表 .....	1128
表 40-1 FMAC 寄存器列表 .....	1133
表 41-1 寄存器一览 .....	1141
表 42-1 MAU 寄存器列表 .....	1150
表 43-1 SWJ 调试端口引脚 .....	1155
表 43-2 灵活的 SWJ-DP 引脚分配 .....	1156
表 43-3 寄存器列表 .....	1157
表 43-4 异步模式引脚 .....	1165
表 43-5 同步模式引脚 .....	1165
表 43-6 引脚释放 .....	1166

## 图索引

图 2-1 总线架构图 .....	64
图 3-1 上电复位.....	69
图 3-2 NRST 复位时序 .....	69
图 3-3 欠压复位.....	70
图 3-4 可编程电压检测 1 复位 .....	71
图 3-5 可编程电压检测 2 复位 .....	71
图 3-6 看门狗和专用看门狗复位 .....	72
图 3-7 掉电唤醒复位.....	73
图 3-8 软件复位.....	73
图 3-9 MPU 错误复位.....	74
图 3-10 RAM 奇偶校验复位.....	74
图 3-11 RAMECC 复位 .....	74
图 3-12 时钟频率异常复位.....	75
图 3-13 外部高速振荡异常复位.....	75
图 3-14 Cortex-M4 Lockup 复位 .....	76
图 4-1 时钟系统框图.....	87
图 4-2 时钟频率测量框图.....	88
图 4-3 外部高速振荡器连接事例 .....	92
图 4-4 外部时钟输入的连接事例图.....	93
图 4-5 外部高速振荡器故障检测例 .....	94
图 4-6 系统时钟选择 XTAL，检测到 XTAL 振荡故障例 .....	95
图 4-7 外部低速振荡器连接事例 .....	97
图 4-8 时钟源切换 .....	101
图 4-9 时钟分频切换.....	102
图 4-10 时钟频率测量时序图 .....	104
图 5-1 电源构成图 .....	135
图 5-2 上电复位、掉电复位波形 .....	136
图 5-3 欠压复位波形 .....	137
图 5-4 PVD1 中断/复位框图 .....	139
图 5-5 PVD2 中断/复位框图 .....	139
图 5-6 电源监视 1 中断时序图 .....	140
图 5-7 电源监视 1 复位时序图 .....	141
图 5-8 电源监视 2 中断运行时序图 .....	142

图 5-9 电源监视 2 复位运行时序图 .....	143
图 5-10 内部电压采样示意图 .....	143
图 5-11 PTWKn 结构框图 .....	151
图 7-1 512KBytes 产品 Flash 地址分布 .....	187
图 7-2 256KBytes 产品单 Flash 地址分布 .....	188
图 7-3 256KBytes 产品双 Flash 地址分布 .....	188
图 7-4 512KBytes 产品 OTP 功能未使能时, Flash 地址分布 .....	199
图 7-5 512KBytes 产品 OTP 功能使能时, Flash 地址分布 .....	199
图 7-6 256KBytes 产品 OTP 功能未使能, Flash 地址分配 .....	200
图 7-7 启动引导交换 1 .....	200
图 7-8 启动引导交换 2 .....	200
图 8-1 错误注入功能示意图 .....	217
图 9-1 端口基本结构示意图 .....	227
图 10-1 中断系统框图 .....	247
图 10-2 中断请求/事件输入选择 .....	270
图 10-3 数字滤波器工作示意图 .....	271
图 11-1 AOS 系统框图 .....	285
图 13-1 KEYS defense 系统框图 .....	327
图 13-2 键盘扫描功能示意图 .....	328
图 14-1 CTC 的基本框图 .....	333
图 14-2 CTC 校准示意图 .....	339
图 15-1 DMA 结构图 .....	347
图 15-2 连锁传输示意图 .....	351
图 15-3 不连续地址传输示意图 (以源地址为例) .....	352
图 15-4 不连续式重置示意图 .....	354
图 15-5 应用例 1: 存储器到存储器传输 .....	356
图 15-6 应用例 2: 存储器到外围电路的传输 .....	358
图 16-1 CMP 功能框图 .....	381
图 16-2 电压比较示意图 .....	386
图 16-3 窗口比较示意图 .....	387
图 16-4 窗口比较原理图 .....	387
图 16-5 窗口输出示意图一 .....	388
图 16-6 窗口输出示意图二 .....	389
图 16-7 窗口输出示意图三 .....	390
图 17-1 ADC 框图 .....	403

图 17-2 通道映射示意图 .....	407
图 17-3 序列 A 单次扫描模式 .....	409
图 17-4 连续扫描 .....	409
图 17-5 双序列扫描模式（序列 A 从被中断通道重新启动） .....	411
图 17-6 双序列扫描模式（序列 A 从第一个通道重新启动） .....	412
图 17-7 模拟看门狗保护区域（比较模式） .....	413
图 17-8 ADC 转换时间 .....	414
图 17-9 平均功能有效时的转换动作 .....	415
图 17-10 AD 与 PGA 通道示意图 .....	416
图 17-11 单次并行触发模式（三 ADC） .....	417
图 17-12 单次延迟触发模式（三 ADC） .....	419
图 17-13 循环并行触发模式（三 ADC） .....	420
图 17-14 循环延迟触发模式（两 ADC） .....	421
图 17-15 循环延迟触发模式（三 ADC） .....	421
图 17-16 ADC 中断和事件输出时序 .....	422
图 18-1 DAC 转换通道框图 .....	444
图 18-2 DAC 转换示意图 .....	445
图 18-3 ADC 转换优先模式动作示意图 .....	447
图 19-1 OTS 功能框图 .....	454
图 20-1 Timer6 基本框图 .....	461
图 20-2 锯齿波波形（递加计数） .....	462
图 20-3 三角波波形 .....	462
图 20-4 比较输出动作 .....	464
图 20-5 捕获输入动作 .....	465
图 20-6 硬件刷新动作 .....	466
图 20-7 软件同步动作 .....	468
图 20-8 硬件同步动作 .....	470
图 20-9 脉宽测量 .....	471
图 20-10 周期测量 .....	472
图 20-11 单缓存方式比较输出时序 .....	474
图 20-12 双缓存方式捕获输入时序 .....	475
图 20-13 锯齿波模式时计数缓存动作 .....	475
图 20-14 三角波模式时计数缓存动作 1 .....	476
图 20-15 三角波模式时计数缓存动作 2 .....	477
图 20-16 捕获输入端口的滤波功能 .....	478

图 20-17 单边对齐独立 PWM .....	479
图 20-18 双边对称独立 PWM .....	480
图 20-19 软件设定 GCMBR 互补 PWM 波输出 .....	481
图 20-20 硬件设定 GCMBR 互补 PWM 波输出 .....	482
图 20-21 双边非对称 PWM 输出.....	483
图 20-22 6 相单边对齐独立 PWM.....	484
图 20-23 带死区时间 3 相双边对称互补 PWM .....	485
图 20-24 周期间隔有效请求信号动作.....	486
图 20-25 位置模式-基本计数.....	488
图 20-26 位置计数模式-相位差计数（1 倍计数） .....	488
图 20-27 位置计数模式-相位差计数（2 倍计数） .....	489
图 20-28 位置计数模式-相位差计数（4 倍计数） .....	489
图 20-29 位置计数模式-方向计数.....	489
图 20-30 公转计数模式-Z 相计数 .....	490
图 20-31 公转计数模式-位置溢出计数.....	490
图 20-32 公转计数模式-混合计数.....	491
图 20-33 公转计数模式-混合计数 Z 相屏蔽动作例 1.....	491
图 20-34 公转计数模式-混合计数 Z 相屏蔽动作例 2.....	492
图 20-35 锯齿波模式时中断&事件输出例.....	501
图 21-1 Timer4 基本框图.....	543
图 21-2 Timer4 锯齿波波形 .....	544
图 21-3 Timer4 三角波波形 .....	544
图 21-4 Timer4 锯齿波模式计数动作 .....	544
图 21-5 Timer4 三角波模式计数动作 .....	545
图 21-6 锯齿波模式波形输出例.....	546
图 21-7 三角波模式波形输出例.....	546
图 21-8 缓存无效时修改锯齿波计数周期.....	547
图 21-9 缓存使能时修改锯齿波计数周期.....	548
图 21-10 缓存使能时修改三角波计数周期.....	548
图 21-11 OCCR 缓冲数据传输（周期间隔响应链接禁止时） .....	549
图 21-12 OCCR 缓冲数据传输（周期间隔响应链接使能） .....	550
图 21-13 输出比较缓冲数据传输（OCMR 缓冲使能） .....	550
图 21-14 SCCR 缓冲传输操作（周期间隔响应链接传输禁止时） .....	551
图 21-15 SCCR 缓冲传输操作（周期间隔响应链接传输使能时） .....	552
图 21-16 锯齿波独立 PWM 输出例 .....	553

图 21-17 三角波独立 PWM 输出例 .....	553
图 21-18 三角波扩展 PWM 输出 .....	554
图 21-19 软件实现互补 PWM 输出 .....	554
图 21-20 死区定时器模式下的互补 PWM 输出 .....	555
图 21-21 脉宽异常时的死区定时器模式下的波形输出 .....	556
图 21-22 死区定时器滤波模式下的互补 PWM 输出 .....	557
图 21-23 周期间隔响应时序图 .....	558
图 21-24 专用事件输出信号周期间隔响应输出 .....	559
图 21-25 PWM 周期输出监测示例 .....	561
图 21-26 比较启动模式下专用事件输出 TIM4_<t>_ADSM 时序 .....	561
图 21-27 延时启动模式下专用事件输出信号的输出时序 .....	563
图 22-1 EMB 结构框图 .....	590
图 22-2 EMB 通道功能框图 .....	592
图 23-1 TimerA 基本框图 .....	612
图 23-2 锯齿波波形（递加计数） .....	613
图 23-3 三角波波形 .....	613
图 23-4 比较输出动作 .....	614
图 23-5 捕获输入动作 .....	615
图 23-6 软件同步动作 .....	616
图 23-7 时钟输入端口的滤波功能 .....	617
图 23-8 锯齿波模式时缓存动作 .....	618
图 23-9 32 位级联计数动作 .....	618
图 23-10 单边对齐 PWM 输出例 .....	619
图 23-11 双边对称 PWM 输出例 .....	620
图 23-12 位置模式-基本计数 .....	621
图 23-13 位置计数模式-相位差计数（1 倍计数） .....	621
图 23-14 位置计数模式-相位差计数（2 倍计数） .....	622
图 23-15 位置计数模式-相位差计数（4 倍计数） .....	622
图 23-16 位置计数模式-方向计数 .....	622
图 23-17 公转计数模式-Z 相计数 .....	623
图 23-18 公转计数模式-位置溢出计数 .....	623
图 23-19 公转计数模式-混合计数 .....	624
图 23-20 红外输出信号特性示意图 .....	624
图 24-1 Timer2 基本框图 .....	646
图 24-2 比较输出动作 .....	648

图 24-3 硬件启动、清零动作 .....	649
图 24-4 捕获输入动作 .....	650
图 24-5 脉宽测量 .....	650
图 24-6 周期测量 .....	651
图 24-7 TRIG 输入端口的数字滤波 .....	651
图 25-1 Timer0 基本框图 .....	664
图 25-2 Timer0 计数时序图 .....	666
图 26-1 RTC 的基本框图 .....	676
图 27-1 硬件启动例 .....	693
图 27-2 软件启动例 .....	694
图 27-3 各种刷新动作时序示例（动作确认，刷新要求信号的下降沿等） .....	695
图 27-4 计数器下溢动作例 .....	696
图 27-5 计数器刷新动作例 .....	697
图 28-1 USART 系统框图 .....	705
图 28-2 UART 数据格式 .....	707
图 28-3 USART 发送数据图例 1 .....	709
图 28-4 USART 发送数据图例 2 .....	709
图 28-5 USART 内部同步和采样时序 .....	710
图 28-6 USART 接收数据图例 1 .....	712
图 28-7 USART 接收数据图例 2 .....	713
图 28-8 多处理器通信图例 .....	716
图 28-9 多处理器模式数据格式 .....	717
图 28-10 多处理器模式发送数据图例 .....	718
图 28-11 多处理器模式接收数据图例 1 .....	719
图 28-12 多处理器模式接收数据图例 2 .....	719
图 28-13 LIN 总线数据行为 .....	720
图 28-14 智能卡连接示意图 .....	723
图 28-15 智能卡模式同步时序和采样时序图 .....	724
图 28-16 智能卡模式数据格式 .....	724
图 28-17 智能卡模式发送数据图例 .....	726
图 28-18 智能卡模式接收数据图例 .....	726
图 28-19 时钟同步模式数据格式 .....	728
图 28-20 时钟同步模式发送数据图例 1 .....	729
图 28-21 时钟同步模式发送数据图例 2 .....	730
图 28-22 时钟同步模式接收数据图例 1 .....	731

图 28-23 时钟同步模式接收数据图例 2.....	731
图 29-1 I2C 系统框图.....	752
图 29-2 I2C 总线的结构例.....	753
图 29-3 I <sup>2</sup> C 总线的时序图 .....	754
图 29-4 I2C 总线的数据格式 .....	755
图 29-5 7 位地址格式的主机发送数据时序图（例） .....	756
图 29-6 7 位地址格式的主机接收数据的时序图（例） .....	757
图 29-7 7 位地址格式的从机发送模式时序图（例） .....	758
图 29-8 7 位地址格式从机接收模式时序图（例） .....	759
图 29-9 SCL 同步时序.....	760
图 29-10 从机发送时序图（1） .....	761
图 29-11 从机发送时序图（2） .....	761
图 29-12 从机接收时序图 .....	762
图 29-13 快速 ACK/NACK 时序图 .....	762
图 29-14 选择 7 位地址格式时的时序.....	763
图 29-15 选择 10 位地址格式时的时序.....	764
图 29-16 广播地址匹配时序图 .....	765
图 29-17 SMBus 主机地址匹配时序图.....	766
图 29-18 SMBus 报警响应地址匹配时序图.....	767
图 29-19 SMBus 默认地址时序图.....	768
图 29-20 数字滤波电路框图.....	772
图 30-1 系统框图.....	796
图 30-2 主机模式结构.....	798
图 30-3 三线式时钟同步运行 .....	799
图 30-4 数据格式.....	800
图 30-5 MSB 先传，奇偶校验无效 .....	801
图 30-6 LSB 先传，奇偶校验无效 .....	801
图 30-7 MSB 先传，奇偶校验有效 .....	802
图 30-8 LSB 先传，奇偶校验有效 .....	802
图 30-9 数据传送格式图（CPHA=0） .....	803
图 30-10 数据传送格式（CPHA=1） .....	804
图 30-11 全双工同步串行通信 .....	805
图 30-12 只进行发送通信 .....	806
图 30-13 常规通信 .....	807
图 30-14 连续通信.....	808

图 30-15 奇偶校验流程.....	815
图 30-16 过载错误处理.....	817
图 30-17 启用时钟自动停止功能时的动作示意图 (CPHA=1) .....	818
图 30-18 启用时钟自动停止功能时的动作示意图 (CPHA=0) .....	818
图 30-19 奇偶校验错误.....	819
图 31-1 QSPI 的模块构成图 .....	830
图 31-2 默认区域设定及 AHB 总线空间内存映射关系图 .....	831
图 31-3 QSPI-ROM 空间内存映像图 .....	832
图 31-4 扩展式 SPI 协议动作示意图 1 (快速读模式) .....	833
图 31-5 扩展式 SPI 协议动作示意图 2 (四线式输入输出快速读模式) .....	833
图 31-6 二线式 SPI 协议动作示意图 (快速读模式) .....	834
图 31-7 四线式 SPI 协议动作示意图 (快速读模式) .....	834
图 31-8 串行接口的基本时序图.....	835
图 31-9 基准时钟选择 HCLK 三分频时输出时钟占空比补正示意图.....	837
图 31-10 QSSL 建立时间配置示意图 .....	838
图 31-11 QSSN 保持时间配置示意图 .....	838
图 31-12 数据接收延迟示意图 .....	839
图 31-13 标准读总线周期示意图.....	841
图 31-14 快速读总线周期示意图.....	841
图 31-15 选择 XIP 模式的快速读总线周期示意图 .....	842
图 31-16 二线式输出快速读总线周期示意图 .....	842
图 31-17 选择 XIP 模式的二线式输出快速读总线周期示意图 .....	843
图 31-18 二线式输入输出快速读总线周期示意图 .....	844
图 31-19 选择 XIP 模式的二线式输入输出快速读总线周期示意图 .....	844
图 31-20 四线式输出快速读总线周期示意图 .....	845
图 31-21 选择 XIP 模式的四线式输出快速读总线周期示意图 .....	845
图 31-22 四线式输入输出快速读总线周期示意图 .....	846
图 31-23 选择 XIP 模式的四线式输入输出快速读总线周期示意图 .....	846
图 31-24 进入 4-Byte 模式指令总线周期示意图 .....	847
图 31-25 退出 4-Byte 模式指令总线周期示意图 .....	847
图 31-26 写许可指令总线周期示意图.....	848
图 31-27 独立转换的单一闪存数据读取操作示意图 .....	848
图 31-28 预读取功能有效时数据读取操作示意图 .....	849
图 31-29 使用 QSPI 总线周期延长功能的数据读取操作示意图 .....	850
图 31-30 XIP 模式控制示意图 .....	851

图 32-1 PLA 模块框图.....	867
图 32-2 PLA 单元框图.....	868
图 33-1 AVCC 分压电路 .....	879
图 33-2 内部 VREF 电路与 VREFINT 电路原理图.....	880
图 34-1 USBFS 系统框图.....	887
图 34-2 USBFS 主机模式系统构建图 .....	889
图 34-3 USBFS 设备模式系统构建图 .....	893
图 34-4 USBFS 动态更新 USBFS_HFIR 寄存器示意图.....	898
图 34-5 USBFS 主机模式下 FIFO 架构示意图.....	898
图 34-6 USBFS 设备模式下 FIFO 架构示意图 .....	900
图 34-7 USBFS 控制状态寄存器存储器映射.....	925
图 35-1 CANFD 系统框图.....	1000
图 35-2 CAN 位时间定义图 .....	1001
图 35-3 CANFD TBUF 寄存器写发送缓冲器和示意图 .....	1004
图 35-4 CAN RBUF 寄存器读接收缓冲器示意图.....	1005
图 35-5 CAN ACF 寄存器访问筛选器组示意图 .....	1005
图 35-6 CANFD LBMI 和 LBME 示意图 .....	1009
图 35-7 TDC 功能示意图.....	1014
图 36-1 MDIO 模块结构图.....	1046
图 36-2 MDIO 帧结构.....	1047
图 36-3 MDIO 应用框图 .....	1048
图 36-4 MDIO 从机通讯流程图.....	1049
图 36-5 MDIO 从机接收通信时序图.....	1050
图 36-6 MDIO 从机发送通信时序图.....	1050
图 36-7 MDIO 从机连续通信时序图.....	1050
图 37-1 EXMC 架构图 .....	1066
图 37-2 外部空间地址分配.....	1068
图 37-3 SMC 的地址空间划分 .....	1069
图 37-4 SMC 状态图 .....	1072
图 37-5 SMC 初始设定流程 .....	1073
图 37-6 单次读动作基本时序（异步方式（RSYN=0）&16 位位宽（MW=01）） .....	1075
图 37-7 单次读动作基本时序（异步方式（RSYN=0）&32 位位宽（MW=10）） .....	1076
图 37-8 单次读动作基本时序（同步方式（RSYN=1）&16 位位宽（MW=01）） .....	1077
图 37-9 单次读动作基本时序（同步方式（RSYN=1）&32 位位宽（MW=10）） .....	1078
图 37-10 地址数据线复用单次读动作基本时序（异步方式（RSYN=0）&32 位位宽（MW=10）） ...	1079

图 37-11 地址数据线复用单次读动作基本时序（异步方式（RSYN=0）& 32 位位宽（MW=10））	1080
图 37-12 单次写动作基本时序（异步方式（WSYN=0）&16 位位宽（MW=01）& BLSS=1）	1081
图 37-13 单次写动作基本时序（异步方式（WSYN=0）&32 位位宽（MW=10）& BLSS=1）	1082
图 37-14 单次写动作基本时序（同步方式（WSYN=1）&16 位位宽（MW=01）& BLSS=0）	1083
图 37-15 单次写动作基本时序（同步方式（WSYN=1）&32 位位宽（MW=10）& BLSS=0）	1084
图 37-16 地址数据线复用单次写动作基本时序（异步方式（WSYN=0）&16 位位宽（MW=01））	1086
图 37-17 地址数据线复用单次写动作基本时序（同步方式（WSYN=1）& 16 位位宽（MW=01））	1088
图 37-18 同步模式下 SMC 访问例	1099
图 38-1 AES 加解密流程示意图	1100
图 38-2 AES 加密流程图	1101
图 38-3 HASH 算法流程图	1111
图 38-4 TRNG 系统框图	1123
图 39-1 CRC 应用示意图	1126
图 40-1 FIR 基本框图	1130
图 41-1 DCU 硬件触发启动	1139
图 42-1 正弦运算角度值示意图	1149
图 43-1 调试控制系统	1153
图 43-2 调试控制系统	1154
图 43-3 JTAG-DP 到 SW-DP 切换时序	1155
图 43-4 TPIU 框图	1164

## 文档约定

### 寄存器相关缩略语

本手册寄存器描述中的缩略语。

缩写	中文释义
RW	读写，软件可以读写该位。
R	只读，软件只能读取该位。
W	只写，软件只能写入该位。读值不确定。
RW0	软件可以读写该位，写0有效，写1无效。
RW1	软件可以读写该位，写1有效，写0无效。
RC	只读，读操作后该位自动清零。
RWA	可读，解锁或安全模式下可写。
RW1A	可读，解锁或安全模式下写1，写0无限制。
RWC	可读，写任意值清零该位。
RW1C	可读，写1清零该位。
RS	只读，读操作后该位自动置位。
Res	保留位，必须保持复位值。

### 词汇表

本手册中专业术语与缩写词的定义。

专业术语/缩写词	英文全称	中文释义
Byte	-	字节，8位数据长度。
Half word	-	半字，16位数据长度。
Word	-	字，32位数据长度。
Double word	-	双字，64位数据长度。
AHB	Advanced High-performance Bus	高级高性能总线。
APB	Advanced Peripheral Bus	高速外设总线。
DMA	Direct Memory Access	直接存储器访问。
.....	.....	.....

### 关于本手册

本手册主要介绍芯片的功能、操作事项和使用方法。关于芯片各型号产品的外设可用性与数量信息，以及外设接口、电气特性、管脚封装等规格参数，请参阅对应的“数据手册”。

## 简介 (Overview)

HC32F472 系列是基于 ARM® Cortex®-M4 32-bit RISC CPU，最高工作频率 120MHz 的高性能 MCU。Cortex®-M4 内核集成了浮点运算单元 (FPU) 和 DSP，实现单精度浮点算术运算，支持所有 ARM 单精度数据处理指令和数据类型，支持完整 DSP 指令集。内核集成了 MPU 单元，同时叠加 DMAC 专用 MPU 单元，保障系统运行的安全性。

HC32F472 系列集成了高速片上存储器，包括最大 512KB 的 Flash，最大 64KB 的 SRAM、4 KB 的掉电保持 SRAM。集成了 Flash 访问加速单元，实现 CPU 在 Flash 上的单周期程序执行。轮询式总线矩阵支持多个总线主机同时访问存储器和外设，提高运行性能。总线主机包括 CPU，DMA，USB 专用 DMA。除总线矩阵外，支持外设间数据传递，基本算术运算和事件相互触发，可以显著降低 CPU 的事务处理负荷。

HC32F472 系列集成了丰富的外设功能。包括：内置高精度 VREF；3 个独立的 12bit 2.5MSPS ADC；2 个增益可调 PGA；8 个 12 位 DAC。；4 个高速电压比较器 (CMP)；10 个多功能 PWM Timer (Timer6) 支持 20 路互补 PWM 输出；1 个电机 PWM Timer (Timer4) 支持 3 路互补 PWM 输出；4 个 16bit 通用 Timer (TimerA) 及 2 个 32bit 通用 Timer(TimerA)支持正交编码输入及 24 路 Duty 独立可设 PWM 输出；13 个串行通信接口 (I2C/UART/SPI)；1 个 QSPI 接口；3 路 CAN；内置 USBFS PHY 的 USBFS Controller；1 个外部扩张总线控制器，包括 SMC 控制器；1 个数学运算单元 (MAU) 和 4 个滤波数学加速器 (FMAC)。

HC32F472 系列支持宽电压范围 (1.8-3.6V)，宽温度范围 (-40-105°C) 和各种低功耗模式。

## 典型应用

HC32F472 系列提供 100pin、48pin 的 LQFP 封装，64pin 的 BGA 封装，主要用于中、高速光模块、高精度模拟、多路 CAN 等应用场景。

# 1 存储器映射 (Memory Mapping)

## 1.1 存储器映射

该 MCU 支持 4GB 的线性地址空间，地址从 0x0000 0000 到 0xFFFF FFFF，其中包含程序和数据。详细存储器映射请参阅下表。

表 1-1 存储器映射

存储器分类		开始地址	结束地址	空间大小	模块*3	保护*4	说明
系统	私有外设 外部总线	0xE0100000	0xFFFFFFFF	511MB	Reserved		自定义空间
		0xE00FF000	0xE00FFFFF	4KB	ROMTABLE		调试控制寄存器 区域
		0xE0042400	0xE00FEFFF	755KB			
		0xE0042000	0xE00423FF	1KB	DBG C		
		0xE0041000	0xE0041FFF	4KB	ETM		
	私有外设 内部总线	0xE0040000	0xE0040FFF	4KB	TPIU		
		0xE000F000	0xE003FFFF	196KB			系统控制空间 NVIC/MPU 等
		0xE000E000	0xE000EFFF	4KB	SCS		
		0xE0003000	0xE000DFFF	44KB			
		0xE0002000	0xE0002FFF	4KB	FPB		
		0xE0001000	0xE0001FFF	4KB	DWT		
		0xE0000000	0xE0000FFF	4KB	ITM		
外部设备	-	0xA0000000	0xFFFFFFFF	1024MB	Reserved		
外部 存储	AHB5 时钟：HCLK	0x98000000	0x9FFFFFFF	128MB	QSPI		
		0x88200000	0x97FFFFFF	254MB	BLANK		
	AHB5 时钟：BCLK	0x88100000	0x881FFFFF	1MB	BLANK		
		0x88000800	0x880FFFFFF	1022KB	BLANK		
		0x88000400	0x880007FF	1KB	BLANK		
		0x88000000	0x880003FF	1KB	SMCR		
		0x80000000	0x87FFFFFF	128MB	BLANK		
		0x60000000	0x7FFFFFFF	512MB	SMC		
外设	-	0x44000000	0x5FFFFFFF	448MB	Reserved		
		0x42000000	0x43FFFFFF	32MB	PeriBitBand		
		0x40100000	0x41FFFFFF	31MB	Reserved		
		0x400C0000	0x400FFFFFF	256KB	BLANK		
	AHB2	0x40080000	0x400BFFFF	256KB	USBFS		

存储器分类		开始地址	结束地址	空间大小	模块*3	保护*4	说明
AHB1 时钟: HCLK	时钟: PCLK1	0x40078800	0x4007FFFF	30KB	BLANK		
		0x40078400	0x400787FF	1KB	BLANK		
		0x40078000	0x400703FF	1KB	CAN_1		
	AHB3 时钟: PCLK1	0x40071800	0x40077FFF	26KB	BLANK		
		0x40071400	0x400717FF	1KB	CAN_2		
		0x40071000	0x400713FF	1KB	CAN_3		
		0x40060000	0x40070FFF	68KB	BLANK		
	APB4 时钟: PCLK3	0x40059000	0x4005FFFF	28KB	BLANK		
		0x40058C00	0x40058FFF	1KB	FMAC_4		
		0x40058800	0x40058BFF	1KB	FMAC_3		
		0x40058400	0x400587FF	1KB	FMAC_2		
		0x40058000	0x400583FF	1KB	FMAC_1		
		0x40057C00	0x40057FFF	1KB	BLANK		
		0x40057800	0x40057BFF	1KB	BLANK		
		0x40057400	0x400577FF	1KB	BLANK		
		0x40057000	0x400573FF	1KB	BLANK		
		0x40056C00	0x40056FFF	1KB	BLANK		
		0x40056800	0x40056BFF	1KB	DCU_3		
		0x40056400	0x400567FF	1KB	DCU_2		
		0x40056000	0x400563FF	1KB	DCU_1		
		0x40055800	0x40055BFF	1KB	BLANK		
		0x40055800	0x40055BFF	1KB	BLANK		
		0x40055400	0x400557FF	1KB	PERIC		周边模块控制寄存器
		0x40055000	0x400553FF	1KB	MAU		禁止访问
		0x40054400	0x40054FFF	3KB	BLANK		
		0x40054000	0x400543FF	1KB	CMU	带保护	
		0x40053800	0x40053FFF	2KB	GPIO		
		0x40053400	0x400537FF	1KB	DMA_2		
		0x40053000	0x400533FF	1KB	DMA_1		
		0x40052000	0x40052FFF	4KB	BLANK		
		0x40051000	0x40051FFF	4KB	INTC	带保护	
		0x40050C00	0x40050FFF	1KB	KEYSCAN		
		0x40050800	0x40050BFF	1KB	RAMIF	带保护	
		0x40050400	0x400507FF	1KB	BLANK		
		0x40050000	0x400503FF	1KB	DMPU	带保护	

存储器分类		开始地址	结束地址	空间大小	模块*3	保护*4	说明
		0x4004F000	0x4004F3FF	1KB	BLANK		
		0x4004EC00	0x4004EFFF	1KB	BLANK		
		0x4004E800	0x4004EBFF	1KB	BLANK		
		0x4004E400	0x4004E7FF	1KB	BLANK		
		0x4004E000	0x4004E3FF	1KB	BLANK		
		0x4004D000	0x4004DFFF	1KB	BLANK		
		0x4004CC00	0x4004CFFF	1KB	PWC CMU	带保护	
		0x4004C800	0x4004CBFF	1KB	BLANK		
		0x4004C400	0x4004C7FF	1KB	WKTMR		
		0x4004C000	0x4004C3FF	1KB	RTC	带保护	
		0x4004AC00	0x4004BFFF	5KB	BLANK		
		0x4004A800	0x4004ABFF	1KB	OTS		
		0x4004A400	0x4004A7FF	1KB	BLANK		
		0x4004A000	0x4004A3FF	1KB	BLANK		
		0x40049C00	0x40049FFF	1KB	CTC		
		0x40049800	0x40049BFF	1KB	BLANK		
		0x40049400	0x400497FF	1KB	SWDT	带保护	
		0x40049000	0x400493FF	1KB	WDT	带保护	
		0x40048800	0x40048FFF	2KB	BLANK		
		0x40048400	0x400487FF	1KB	FCM		
		0x40048000	0x400483FF	1KB	FCG	带保护	
外设	APB3 时钟:PCLK4	0x40042400	0x40047FFF	23KB	BLANK		
		0x40042000	0x400423FF	1KB	TRNG	带保护	
		0x40041C00	0x40041FFF	1KB	DAC_4		
		0x40041800	0x40041BFF	1KB	DAC_3		
		0x40041400	0x400417FF	1KB	DAC_2		
		0x40041000	0x400413FF	1KB	DAC_1		
		0x40040C00	0x40040FFF	1KB	BLANK		
		0x40040800	0x40040BFF	1KB	ADC_3		
		0x40040400	0x400407FF	1KB	ADC_2		
		0x40040000	0x400403FF	1KB	ADC_1		
APB5 时钟:PCLK0	APB5 时钟:PCLK0	0x4003C000	0x4003FFFF	16KB	TMR6		
		0x4003BC00	0x4003BFFF	1KB	I2C_3		
		0x4003B800	0x4003BBFF	1KB	I2C_2		
		0x4003B400	0x4003B7FF	1KB	I2C_1		
		0x4003B000	0x4003B3FF	1KB	CMP_2		
		0x4003AC00	0x4003AFFF	1KB	TimerA_4		

存储器分类		开始地址	结束地址	空间大小	模块*3	保护*4	说明
外设	APB2 时钟:PCLK1	0x4003A800	0x4003ABFF	1KB	TimerA_3		
		0x4003A400	0x4003A7FF	1KB	TimerA_2		
		0x4003A000	0x4003A3FF	1KB	TimerA_1		
		0x40038800	0x40039FFF	6KB	BLANK		
		0x40038400	0x400387FF	1KB	CMP_1		
		0x40038000	0x400383FF	1KB	Timer4_1		
		0x40028800	0x40037FFF	62KB	BLANK		
		0x40028400	0x400287FF	1KB	MDIO		
		0x40028000	0x400283FF	1KB	BLANK		
		0x40027C00	0x40027FFF	1KB	BLANK		
		0x40027800	0x40027BFF	1KB	BLANK		
		0x40027400	0x400277FF	1KB	BLANK		
		0x40027000	0x400273FF	1KB	BLANK		
		0x40026C00	0x40026FFF	1KB	BLANK		
		0x40026800	0x40026BFF	1KB	BLANK		
		0x40026400	0x400267FF	1KB	TimerA_6		
		0x40026000	0x400263FF	1KB	TimerA_5		
		0x40025800	0x40025FFF	2KB	BLANK		
		0x40025400	0x400257FF	1KB	Timer2_4		
		0x40025000	0x400253FF	1KB	Timer2_3		
		0x40024C00	0x40024FFF	1KB	Timer2_2		
		0x40024800	0x40024BFF	1KB	Timer2_1		
		0x40024400	0x400247FF	1KB	Timer0_2		
		0x40024000	0x400243FF	1KB	Timer0_1		
		0x40022800	0x40023FFF	6KB	BLANK		
		0x40022400	0x400227FF	1KB	BLANK		
		0x40022000	0x400223FF	1KB	BLANK		
		0x40021C00	0x40021FFF	1KB	BLANK		
		0x40021800	0x40021BFF	1KB	BLANK		
		0x40021400	0x400217FF	1KB	USART_6		
		0x40021000	0x400213FF	1KB	USART_5		
		0x40020C00	0x40020FFF	1KB	USART_4		
		0x40020800	0x40020BFF	1KB	BLANK		
		0x40020400	0x400207FF	1KB	SPI_4		
		0x40020000	0x400203FF	1KB	SPI_3		
APB1 时钟:PCLK1		0x4001E800	0x4001FFFF	6KB	BLANK		
		0x4001E400	0x4001E7FF	1KB	BLANK		
		0x4001E000	0x4001E3FF	1KB	BLANK		
		0x4001DC00	0x4001DFFF	1KB	BLANK		

存储器分类		开始地址	结束地址	空间大小	模块*3	保护*4	说明
		0x4001D800	0x4001DBFF	1KB	BLANK		
		0x4001D400	0x4001D7FF	1KB	USART_3		
		0x4001D000	0x4001D3FF	1KB	USART_2		
		0x4001CC00	0x4001CFFF	1KB	USART_1		
		0x4001C800	0x4001CBFF	1KB	BLANK		
		0x4001C400	0x4001C7FF	1KB	SPI_2		
		0x4001C000	0x4001C3FF	1KB	SPI_1		
		0x40018000	0x4001BFFF	16KB	BLANK		
		0x40017C00	0x40017FFF	1KB	EMB		
		0x40010C00	0x40017BFF	28KB	BLANK		
		0x40010800	0x40010BFF	1KB	AOS		内部触发事件寄存器区域
		0x40010400	0x400107FF	1KB	EFM	带保护	
		0x40010000	0x400103FF	1KB	BLANK		
AHB4 时钟: PCLK1		0x40009400	0x4000FFFF	27KB	BLANK		
		0x40009000	0x400093FF	1KB	BLANK		
		0x40008C00	0x40008FFF	1KB	CRC	带保护	
		0x40008800	0x40008BFF	1KB	BLANK		
		0x40008400	0x400087FF	1KB	HASH	带保护	
		0x40008000	0x400083FF	1KB	AES	带保护	
		0x40000000	0x40007FFF	32KB	Reserved		
SRAM 时钟: HCLK		0x24000000	0x3FFFFFFF	448MB	Reserved		
		0x22000000	0x23FFFFFF	32MB	SRAMBitBand		
		0x20100000	0x21FFFFFF	31MB	Reserved		
		0x200D9000	0x200FFFFFF	156KB	Reserved		
		0x200F0000	0x200F0FFF	4KB	Ret_SRAM	带保护	
		0x20060000	0x200EFFFF	576KB	Reserved		
		0x20040000	0x2005FFFF	128KB	Reserved		
		0x20008000	0x2003FFFF	224KB	Reserved		
		0x20000000	0x20007FFF	32KB	SRAM0		ECC RAM
CODE	SRAM 时钟: HCLK	0x1FFF8000	0x1FFFFFFF	32KB	SRAMH		
		0x03006000	0x1FFF7FFF	463.94MB	BLANK		
	OTP,Flash 时钟: HCLK	0x0300400C	0x03005FFF	8180B	BLANK		
		0x03004000	0x0300400B	12 B	数据安全保护		用于配置数据安全保护
		0x03002004	0x03003FFF	8187B	BLANK		
		0x03002000	0x03002003	4 B	引导交换		
		0x03001ADC	0x03001FFF	1316B	BLANK		

存储器分类		开始地址	结束地址	空间大小	模块*3	保护*4	说明
REMAPPING 时钟：HCLK		0x03000000	0x03001AD B	6876B	OTP		
	-	0x02100000	0x02FFFFFF	15M	BLANK		
	REMAPPING 时钟：HCLK	0x02080000	0x020FFFFF	512KB	REMAPP1		地址重映射区域 1
		0x02000000	0x0207FFFF	512KB	REMAPP0		地址重映射区域 0
	-	0x00080000	0x01FFFFFF	31.5M	BLANK		
	Flash 时钟：HCLK	0x00040000	0x0007FFFF	256KB	Embedded Flash 1		
		0x00000000	0x0003FFFF	256KB	Embedded Flash 0		

\*1 请参考 ARM Cortex®-M4 说明手册“存储器系统”。

\*2 总线说明请参考【总线架构（BUS）】总线章节。

\*3 Reserved：访问总线会引起总线错误；BLANK：写访问无效，读访问时读到0。

\*4 带保护功能的模块，在保护功能有效时只支持CPU特权模式访问。具体寄存器及说明参考【存储保护单元（MPU）】章节。

## 1.2 外部空间映射

QSPI 空间被区分为 2 段空间，包括 QSPI I/O 寄存器空间 64MB 和外部 QSPI 设备空间 64MB。分配关系请参考下图。

表 1-2 QSPI 地址空间分配

QSPI	0x98000000	0x9FFFFFFF	128MB	QSPI I/O 寄存器	0x9C000000	0x9FFFFFFF	64MB
				外部 QSPI 设备	0x98000000	0x9BFFFFFF	64MB

## 1.3 位段空间

Cortex-M4 存储器映射包括两个位段区域。这些区域将存储器别名区域中的每个字映射到存储器位段区域中的相应位。在别名区域写入字时，相当于对位段区域的目标位执行读-修改-写操作。

在该 MCU 中，外设寄存器和 SRAM 均映射到一个位段区域，这样可实现单个位段的读写操作。这些操作仅适用于 Cortex-M4 访问，对于其它总线主接口（如 DMA）无效。

## 1.4 地址重映射

本 MCU 提供 2 个重映射地址，可配置存储器地址重映射功能，源地址可设为主闪存 FLASH 地址和高速 SRAM 地址。

重映射地址 0：

0x0200\_0000~0x0208\_0000(视重映射空间而定 MMF\_REMCR0/1.RMSIZE[4:0])

重映射地址 1：

0x0208\_0000~0x0210\_0000(视重映射空间而定 MMF\_REMCR0/1.RMSIZE[4:0])

重映射功能有效时，地址对应表如下：

表 1-3 目标地址配置示例

寄存器设定	重映射地址 (CPU地址—CPUADDR[31:0])	源地址		
		高 3 位 地址	中地址	低地址
RMSIZE[4:0]=01110情形 (重映射空间：16K)	0x02000000~0x02003FFF	全0	RMTADDR[16:2]	CPUADDR[13:0]
RMSIZE[4:0]=01111情形 (重映射空间：32K)	0x02000000~0x02007FFF	全0	RMTADDR[16:3]	CPUADDR[14:0]
RMSIZE[4:0]=10000情形 (重映射空间：64K)	0x02080000~0x0208FFFF	全0	RMTADDR[16:4]	CPUADDR[15:0]
RMSIZE[4:0]=10001情形 (重映射空间：128K)	0x02080000~0x0209FFFF	全0	RMTADDR[16:5]	CPUADDR[16:0]

举例，使用重映射地址 0 功能，设定源地址为主闪存 FLASH 地址 0x00008000，重映射空间 32K，寄存器 MMF\_REMCR0 需设定为 0x8000800F。

使用重映射地址 1 功能，设定源地址为高速 SRAM 地址 0x1FFF8000，重映射空间为 16K，寄存器 MMF\_REMCR1 需设定为 0x9FFF800E。

**注意：**源地址的起始地址应设为重映射空间的整数倍。

## 1.5 重映射寄存器

重映射模块共有三个寄存器。地址空间如下：

寄存器地址：0x40010500

表 1-4 寄存器列表

寄存器名	符号	偏移地址	位宽	复位值
访问保护寄存器	MMF_REMPRT	0x0000	32	0x00000000
重映射控制寄存器0	MMF_REMCR0	0x0004	32	0x00000000
重映射控制寄存器1	MMF_REMCR1	0x0008	32	0x00000000

### 1.5.1 访问保护寄存器 (MMF\_REMPRT)

复位值：0x0000\_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
															-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MMF_REMPRT[15:0]															
位	标记	位名	功能	读写											
b31~16	Reserved	-	读出时为“0”,写入时写“0”	R											
b15~0	MMF_REMPRT[15:0]	保护寄存器	寄存器MMF_REMCR0和MMF_REMCR1写保护： 先对MMF_REMPRT[15:0]写入0x0123接着写入0x3210解除保护； 寄存器MMF_REMCR0和MMF_REMCR1写保护状态时，读寄存器为0 寄存器MMF_REMCR0和MMF_REMCR1解除写保护状态时，读寄存器 为1	R/W											

## 1.5.2 重映射控制寄存器 (MMF\_REMCRx) (x=0, 1)

复位值: 0x0000\_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EN	-	RMTADDR[16:4]													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RMTADDR[3:0]				-						RMSIZE[4:0]					
<hr/>															
位	标记	位名	功能	读写											
b31	EN	重映射有效位	0: 重映射无效 1: 重映射有效	R/W											
b30~29	Reserved	-	读出时为“0”,写入时写“0”	R											
b28~12	RMTADDR[16:0]	源地址	有效位数和RMSIZE[4:0]设定有关。 设定可参考表1-3。	R/W											
b11~b5	Reserved	-	读出时为“0”,写入时写“0”	R											
b4~b0	RMSIZE[4:0]	重映射空间	设定重映射空间 00000~01011: 保留, 设定禁止 01100: 4KB 01101: 8KB 01110: 16KB 01111: 32KB 10000: 64KB 10001: 128KB 10010: 256KB 10011: 512KB (256KB 产品设定禁止) 10100~11111: 保留, 设定禁止	R/W											

## 2 总线架构 (BUS)

### 2.1 概述

主系统由 32 位多层 AHB 总线矩阵构成，可实现以下主机总线和从机总线的互连：

- 主机总线
  - Cortex-M4 内核 CPU-I 总线, CPU-D 总线, CPU-S 总线
  - 系统 DMA\_1 总线, 系统 DMA\_2 总线
  - USBFS\_DMA 总线
- 从机总线
  - Flash ICODE 总线
  - Flash DCODE 总线
  - Flash MCODE 总线(CPU 以外其他主机访问 Flash 的总线)
  - 高速 SRAM(SRAMH 32KB)总线
  - 系统 SRAM(SRAM0 32KB)总线
  - 系统 SRAM(Ret\_SRAM 4KB)总线
  - APB1 外设总线(EMB/SPI/USART/AOS/EFM)
  - APB2 外设总线(Timers/SPI/USART/MDIO)
  - APB3 外设总线(ADC/DAC/TRNG)
  - APB4 外设总线(FCM/WDT/SWDT/PWC/CTC/OTS/RTC/WKTM/PLA)
  - APB5 外设总线(Timers/CMP/I2C)
  - AHB1 外设总线(DMPU/KEYSCAN/INTC/DCU(GPIO/DMA/CMU/MAU/FMAC)
  - AHB2 外设总线(CAN/USBFS)
  - AHB3 外设总线(CAN)
  - AHB4 外设总线(AES/HASH/CRC)
  - AHB5 外设总线(SMC/SMCR/ QSPI)

借助总线矩阵，可以实现主机总线到从机总线高效率的并发访问。

## 2.2 总线架构

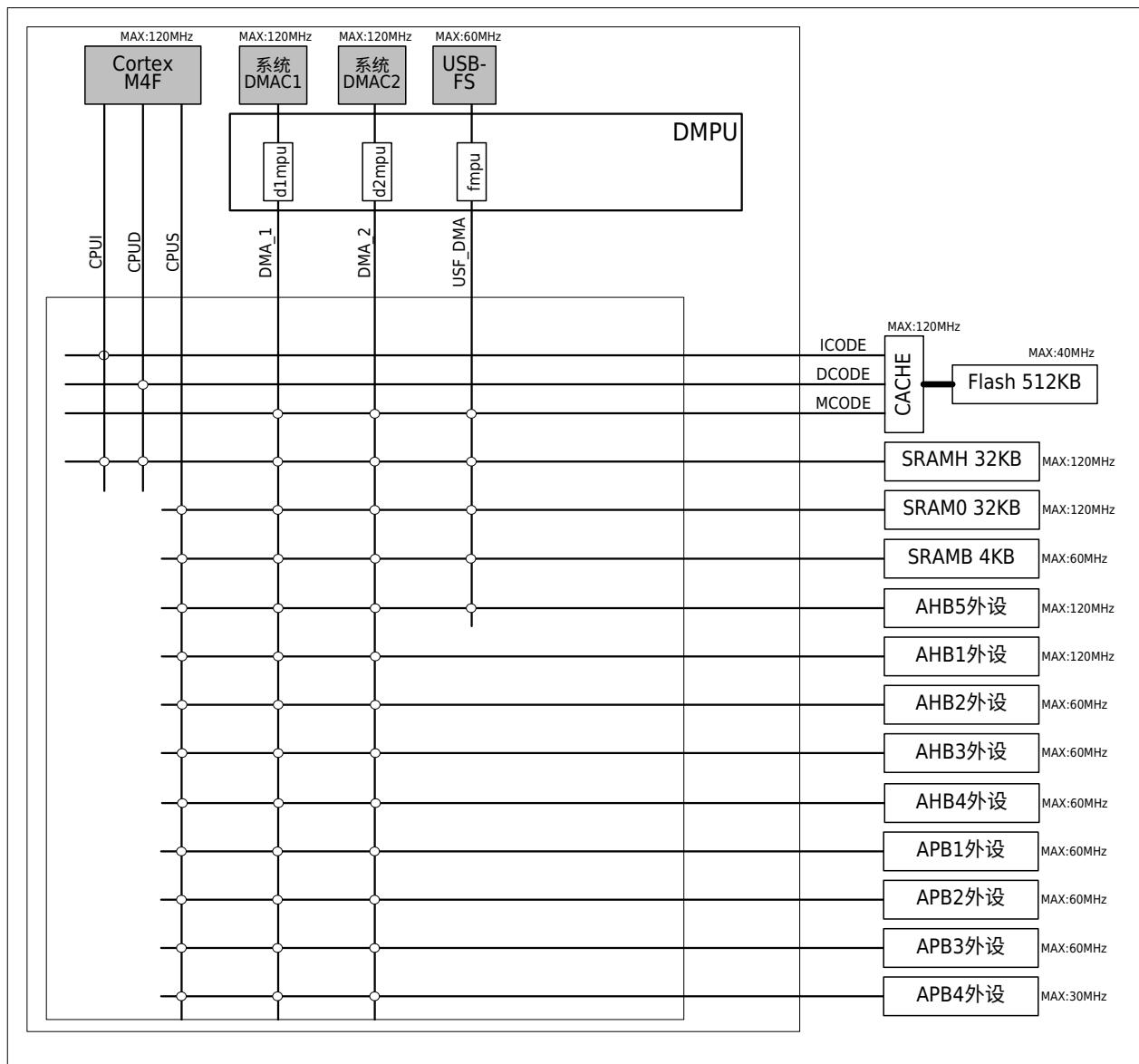


图 2-1 总线架构图

总线矩阵用于各主机总线之间的访问仲裁管理。仲裁采用循环调度算法。

- CPU-I 总线

Cortex-M4 内核的指令总线, CPU 通过此总线获取指令。访问对象是包含代码的 Flash 和 SRAMH。

- CPU-D 总线

Cortex-M4 内核的数据总线, CPU 通过此总线进行立即数加载和调试访问。访问对象是包含代码或数据的 Flash 和 SRAMH。

- CPU-S 总线

Cortex-M4 内核的系统总线, CPU 通过此总线访问外设或者系统 SRAM, 也可以通过此总线获取指令和立即数 (效率低于通过 CPU-I 总线和 CPU-D 总线)。访问对象为 SRAMH/SRAM0/Ret\_SRAM 所有外设以及外部扩展空间。

- DMA\_1 总线, DMA\_2 总线

系统 DMA\_1/系统 DMA\_2 专用总线, DMA\_1/DMA\_2 通过此总线访问数据存储器和外设, 访问对象为 Flash/SRAMH/SRAM0/Ret\_SRAM 所有外设以及外部扩展空间。

- USBFS-DMA 总线

USBFS 的 DMA 专用总线, USBFS 通过此总线访问所有存储器空间。访问对象为 Flash/SRAMH/SRAM0/Ret\_SRAM 以及外部扩展空间。

## 2.3 总线功能

总线负责实现主机对从机的读写访问。主机模块工作频率比从机模块高 (如 CPU-S 访问 RTC) 时, 总线自动进行降频同步处理。主机模块工作频率比从机模块低 (如 USBFS\_DMA 访问 SRAMH) 时, 总线自动进行升频同步处理。

通过总线矩阵, 不同主机总线的访问目标不冲突时, 各访问能够同时进行。例如 CPU-I 访问 Flash, CPU-D 访问 SRAMH, CPUS 访问 APB 外设, DMA\_1 访问 SRAM0, DMA\_2 访问 Ret\_SRAM, USBFS-DMA 访问 AHB5 的外部扩展空间, 这些访问可以同时进行。

### 3 复位控制 (RMU)

#### 3.1 简介

芯片配置了 15 种复位方式。

- 上电复位 (POR)
- NRST 引脚复位 (NRST)
- 欠压复位 (BOR)
- 可编程电压检测 1 复位 (PVD1R)
- 可编程电压检测 2 复位 (PVD2R)
- 看门狗复位 (WDTR)
- 专用看门狗复位 (SWDTR)
- 掉电唤醒复位 (PDRST)
- 软件复位 (SRST)
- MPU 错误复位 (MPUR)
- RAM 奇偶校验复位 (RAMPR)
- RAMECC 复位 (RAMECCR)
- 时钟异常复位 (CKFER)
- 外部高速振荡器异常停振复位 (XTALER)
- Cortex-M4 LockUP 复位 (LKUPR)

## 3.2 复位方式和复位标志位

复位方式及产生条件如表 3-1 所示。

表 3-1 复位方式及产生条件

复位方式	产生条件
上电复位	VCC 上电
NRST 引脚复位	NRST 引脚输入低电平
欠压复位	VCC 电压降至低于 VBOR 电压
可编程电压检测 1 复位	VCC 电压降至低于 PVD1 电压
可编程电压检测 2 复位	VCC 电压降到低于 PVD2 电压
看门狗复位	看门狗定时器产生刷新错误或者溢出错误
专用看门狗复位	专用看门狗发生刷新错误或者溢出错误
掉电唤醒复位	通过设置掉电模式产生的复位，内核在掉电唤醒事件发生后从复位状态唤醒
软件复位	设置复位寄存器位（ARM 寄存器 AIRCR.SYSRESETREQ 位）
MPU 错误复位	MPU 访问错误产生的复位
RAM 奇偶校验复位	RAM 发生奇偶校验错误时产生的复位
RAM ECC 错误复位	RAM 发生 ECC 错误时产生的复位
时钟频率异常复位	时钟频率监视功能（FCM）检测到时钟周期错误时
外部高速振荡器异常停振复位	外部高速振荡器异常停振时产生的复位
Cortex-M4 LockUP 复位(LKUPR)	当 Cortex-M4 遇到严重的异常时，它会将自己的 PC 指针停在当前地址处，并锁死自己，在几个时钟周期延时之后复位整个芯片

发生复位时，芯片根据复位方式将对应的复位标志位进行置位，复位标志位如表 3-2 所示。比如发生引脚复位，引脚复位标志位 PINRF 被置 1，在 PINRF 被置位后通过写 CLRF 可以将 PINRF 清零。

表 3-2 复位方式及复位标志

复位标志	复位方式													Cortex-M4 Lockup 复位	
	上电复位	引脚复位	欠压复位	电压检测 1 复位	电压检测 2 复位	看门狗复位	专用看门狗复位	掉电唤醒复位	软件复位	MPU 错误复位	RAM 奇偶校验错误复位	RAM ECC 错误复位	时钟频率异常复位	外部高速振荡器异常停振	
上电复位标志 (RMU_RSTF0.PORF)	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—
引脚复位标志 (RMU_RSTF0.PINRF)	✗	✓	—	—	—	—	—	—	—	—	—	—	—	—	—
欠压复位标志 (RMU_RSTF0.BORF)	✗	✗	✓	—	—	—	—	—	—	—	—	—	—	—	—
可编程电压检测 1 复位标志 (RMU_RSTF0.PVD1RF)	✗	✗	✗	✓	—	—	—	—	—	—	—	—	—	—	—
可编程电压检测 2 复位标志 (RMU_RSTF0.PVD2RF)	✗	✗	✗	—	✓	—	—	—	—	—	—	—	—	—	—
看门狗复位标志 (RMU_RSTF0.WDRF)	✗	✗	✗	—	—	✓	—	✗	—	—	—	—	—	—	—
专用看门狗复位标志 (RMU_RSTF0.SWDRF)	✗	✗	✗	—	—	—	✓	✗	—	—	—	—	—	—	—
掉电唤醒复位标志 (RMU_RSTF0.PDRF)	✗	—	—	—	—	—	—	✓	—	—	—	—	—	—	—
软件复位标志 (RMU_RSTF0.SWRF)	✗	✗	✗	—	—	—	—	✗	✓	—	—	—	—	—	—
MPU 错误复位 (RMU_RSTF0.MPUERF)	✗	✗	✗	—	—	—	—	✗	—	✓	—	—	—	—	—
RAM 奇偶错误复位 (RMU_RSTF0.RAPERF)	✗	✗	✗	—	—	—	—	✗	—	—	✓	—	—	—	—
RAM ECC 复位 (RMU_RSTF0.RAECRF)	✗	✗	✗	—	—	—	—	✗	—	—	—	✓	—	—	—
时钟频率异常复位 (RMU_RSTF0.CKFERF)	✗	✗	✗	—	—	—	—	✗	—	—	—	—	✓	—	—
外部高速振荡器异常停振复位 (RMU_RSTF0.XTALERF)	✗	✗	✗	—	—	—	—	✗	—	—	—	—	—	✓	—
Cortex-M4 Lockup 复位 (RMU_RSTF0.LKUPRF)	✗	✗	✗	—	—	—	—	✗	—	—	—	—	—	—	✓

✓：置位 X:清零 —：不变

### 3.3 复位时序

#### 3.3.1 上电复位

上电复位是上电复位电路引起的内部复位，时序如图 3-1。在将 NRST 引脚置为高电平的状态下接通电源，就产生上电复位。VCC 电压高于上电复位电压  $V_{POR}$  后经过一定的时间 ( $T_{RSTPOR}$ ) 之后芯片内部复位解除，CPU 开始执行代码。产生上电复位时，上电复位标志 RMU\_RSTF0.PORF 被置位。上电复位的详细说明请参照【上电复位/掉电复位动作说明】。

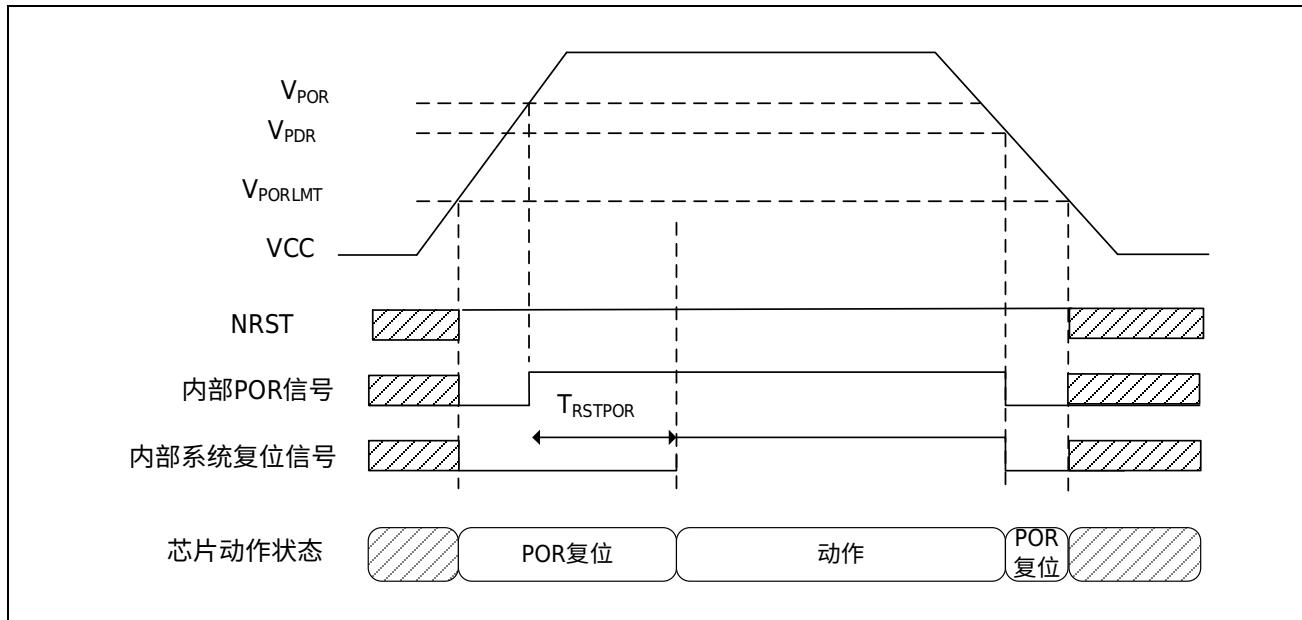


图 3-1 上电复位

#### 3.3.2 NRST 引脚复位

引脚复位是 NRST 引脚被驱动为低电平引起的复位，复位时序如图 3-2。NRST 管脚维持  $T_{NRST}$  宽度以上的低电平后，经过一定的内部复位时间 ( $T_{INRST}$ )，解除内部复位。

产生 NRST 引脚复位时，引脚复位标志 RMU\_RSTF0.PINRF 被置位。

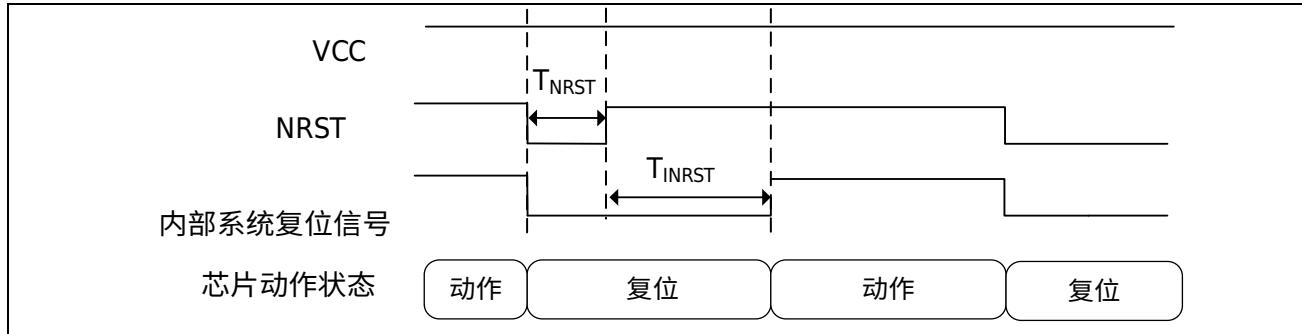


图 3-2 NRST 复位时序

### 3.3.3 欠压复位

欠压复位是电压监测电路引起的内部复位，时序如图 3-3。欠压通过 ICG 寄存器设置为复位使能后，如果 VCC 电压低于监测电压  $V_{BOR}$ ，RMU\_RSTF0.VBORF 被置位。当 VCC 电压高于监测电压  $V_{BOR}$  经过  $V_{BOR}$  的复位时间( $T_{RSTBOR}$ )后解除复位。

欠压的复位设定请参考 【欠压复位(BOR)说明】。

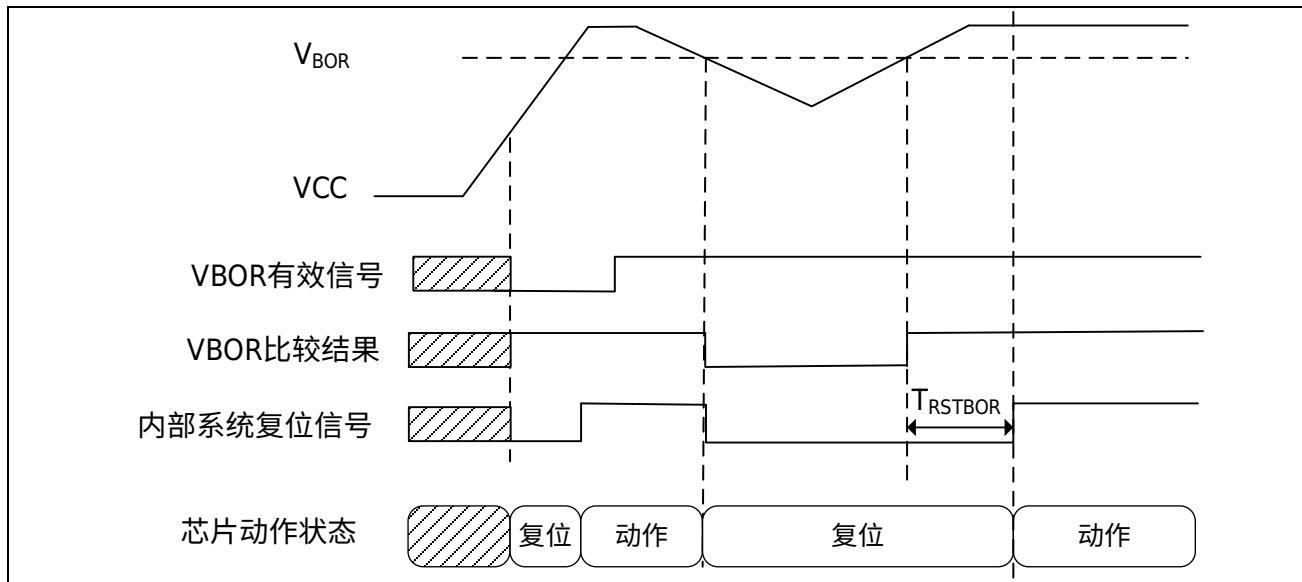


图 3-3 欠压复位

### 3.3.4 可编程电压检测 1 复位、可编程电压检测 2 复位

可编程电压检测 1 和可编程电压检测 2 复位时电压监测电路引起的复位。

在可编程电压检测 1 有效并设置为复位使能后，如果 VCC 低于可编程电压检测 1 的监测电压，产生可编程电压检测 1 复位，RMU\_RSTF0.PVD1F 被置位。当 VCC 电压高于可编程电压检测 1 的监测电压后，经过 PVD1 的复位时间( $T_{IPVD1}$ )后解除复位。

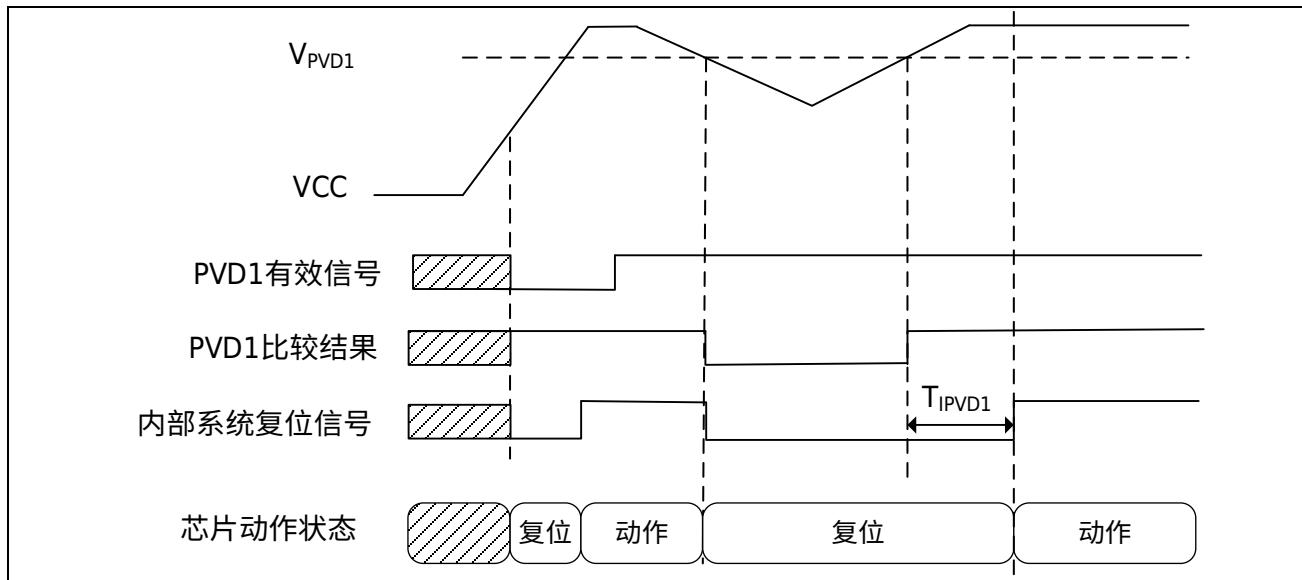


图 3-4 可编程电压检测 1 复位

在可编程电压检测 2 有效并设置为复位使能后，如果 VCC 低于可编程电压检测 2 的监测电压，产生可编程电压检测 2 复位，RMU\_RSTF0.PVD2F 被置位。当 VCC 电压高于可编程电压检测 2 的监测电压后，经过 PVD2 的复位时间( $T_{IPVD2}$ )后解除复位。

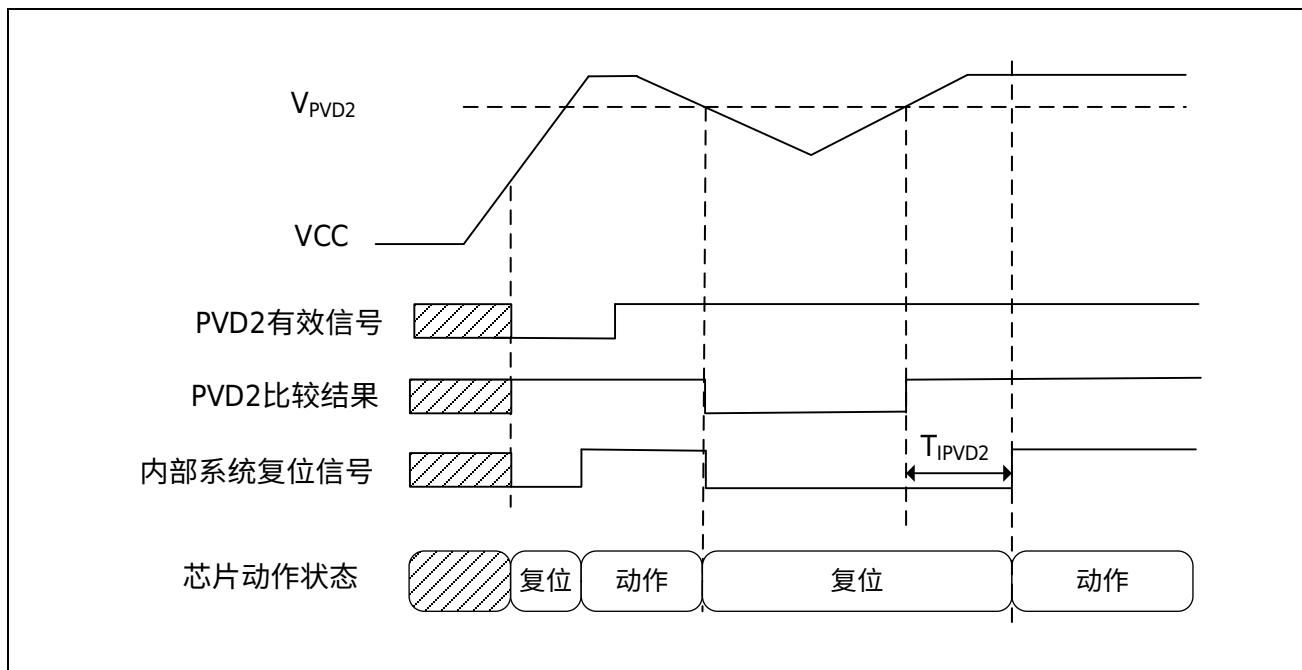


图 3-5 可编程电压检测 2 复位

可编程电压检测 1、可编程电压检测 2 的复位设定请参考【电源电压检测单元(PVD)说明】。

### 3.3.5 看门狗复位、专用看门狗复位

看门狗复位是看门狗定时器引起的内部复位，专用看门狗复位时专用看门狗定时器引起的内部复位，复位时序如图 3-6。

设定看门狗复位有效后，在看门狗定时器产生下溢或者不在刷新允许期间进行写操作时产生看门狗复位。看门狗复位将 RMU\_RSTF0.WDRF 置位。产生看门狗复位后，经过内部复位时间  $T_{RIPT}$  之后，芯片解除复位。

设定专用看门狗复位有效后，在专用看门狗定时器产生下溢或者不在刷新允许期间进行写操作时产生看门狗复位。专用看门狗复位将 RMU\_RSTF0.SWDRF 置位。产生专用看门狗复位后，经过内部复位时间  $T_{RIPT}$  之后，芯片解除复位。

PWC\_PWRC6.WDRTNE=1 时产生看门狗复位时，芯片的端口状态保持看门狗复位前的状态。看门狗复位产生后，可以通过置位 PWC\_PWRC6.IORTNCLR 解除端口的保持状态。PWC\_PWRC6.WDRTNE=0 时产生看门狗复位时，芯片的端口状态将初始化成高阻状态。

PWC\_PWRC6.WDRTNE=1 时产生专用看门狗复位时，芯片的端口状态保持专用看门狗复位前的状态。专用看门狗复位产生后，可以通过置位 PWC\_PWRC6.IORTNCLR 解除端口的保持状态。PWC\_PWRC6.WDRTNE=0 时产生专用看门狗复位时，芯片的端口状态将初始化成高阻状态。

PWC\_PWRC6.WDRDAC=1 时产生看门狗复位时，芯片里的 DAC 模块保持看门狗复位前的状态。看门狗复位产生后，可以通过置位 RMU\_FRST3.DAC1/2/3/4 将 DAC 模块复位。PWC\_PWRC6.WDRDAC=0 时产生看门狗复位时，芯片里的 DAC 模块被复位。

PWC\_PWRC6.WDRDAC=1 时产生专用看门狗复位时，芯片里的 DAC 模块保持专用看门狗复位前的状态。专用看门狗复位产生后，可以通过清零 RMU\_FRST3.DAC1/2/3/4 将 DAC 模块复位。PWC\_PWRC6.WDRDAC=0 时产生专用看门狗复位时，芯片里的 DAC 模块被复位。

有关看门狗复位和专用看门狗复位的详细内容，请参考【看门狗计数器 (WDT/ SWDT)】。

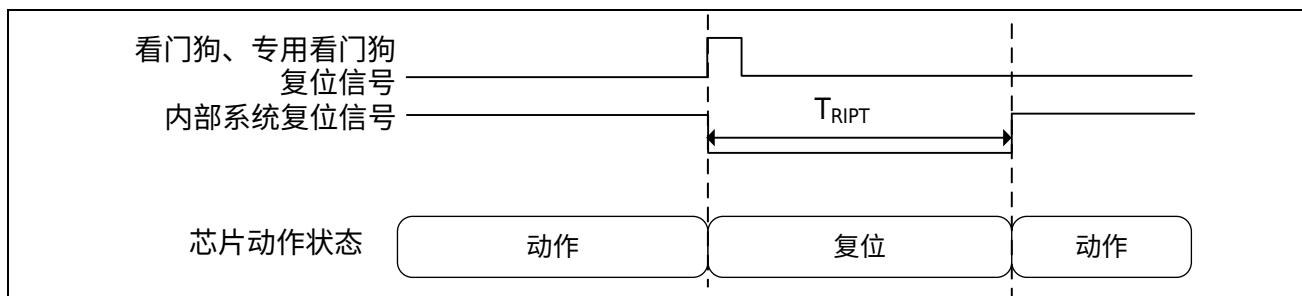


图 3-6 看门狗和专用看门狗复位

### 3.3.6 掉电唤醒复位

掉电唤醒复位是芯片在设置 PWC\_PWRC0.PWDN 为 1 时执行 WFI 命令，进入掉电模式后，通过掉电模式唤醒事件解除掉电模式时产生的内部复位，时序如图 3-7。在解除掉电模式并经过返回时间( $T_{IPDX}$ ,  $x=1,2,3,4$ )后，解除掉电唤醒复位。返回时间根据设定的具体掉电模式有所不同，在掉电模式 1 时最小，掉电模式 3 时最大。

有关掉电唤醒复位的详细内容，请参照【掉电模式】。

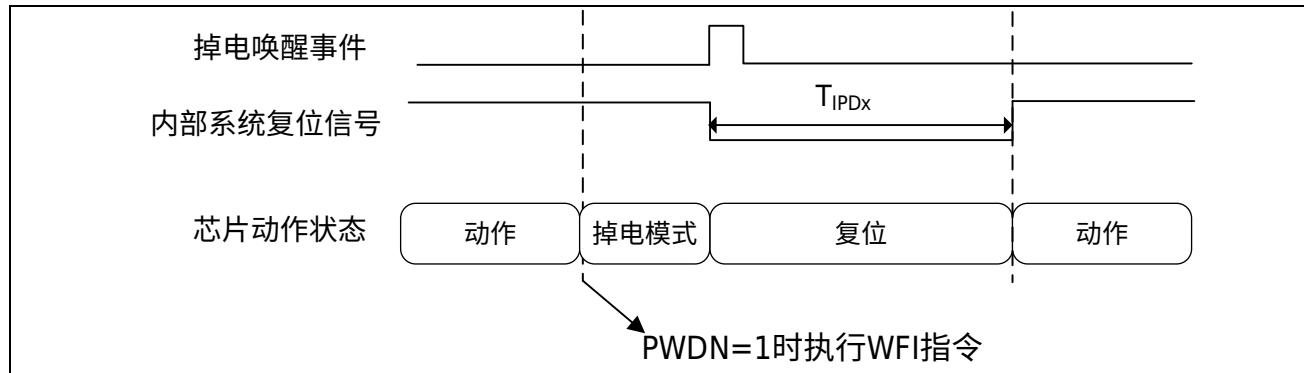


图 3-7 掉电唤醒复位

### 3.3.7 软件复位

通过写 ARM 寄存器 AIRCR 的 SYSRESETREQ 位产生软件复位。产生软件复位时，RMU\_RSTF0.SWRF 位被置位。经过内部复位时间  $T_{RIPT}$  之后，芯片解除复位。

PWC\_PWRC6.SWRTNE=1 时产生软件复位时，芯片的端口状态保持软件复位前的状态。软件复位产生后，可以通过置位 PWC\_PWRC6.IORTNCLR 解除端口的保持状态。PWC\_PWRC6.SWRTNE=0 时产生软件复位时，芯片的端口状态将初始化成高阻状态。

PWC\_PWRC6.SWRDAC=1 时产生软件复位时，芯片里的 DAC 模块保持软件复位前的状态。软件复位产生后，可以通过置位 RMU\_FRST3.DAC1/2/3/4 将 DAC 模块复位。PWC\_PWRC6.SWRDAC=0 时产生软件复位时，芯片里的 DAC 模块被复位。

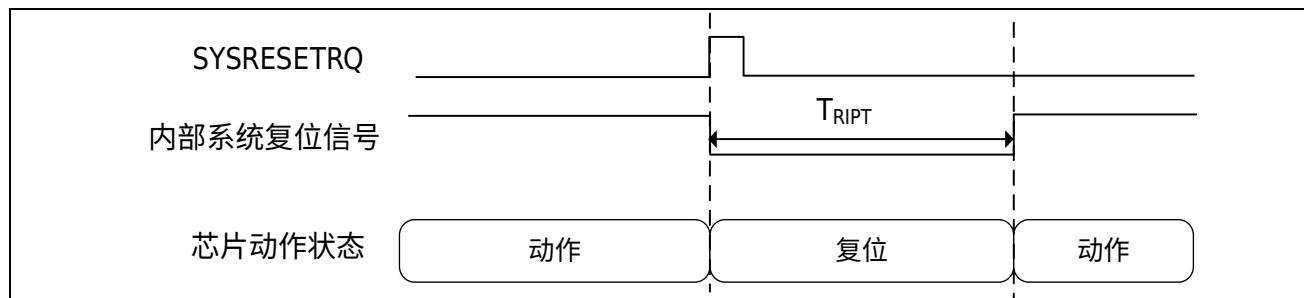


图 3-8 软件复位

### 3.3.8 MPU 错误复位

MPU 错误复位将 RMU\_RSTF0.MPUERF 置位，时序如图 3-9。经过内部复位时间  $T_{RIPT}$  之后，芯片解除复位。

有关 MPU 错误复位的设定，请参照【存储保护单元（MPU）】。

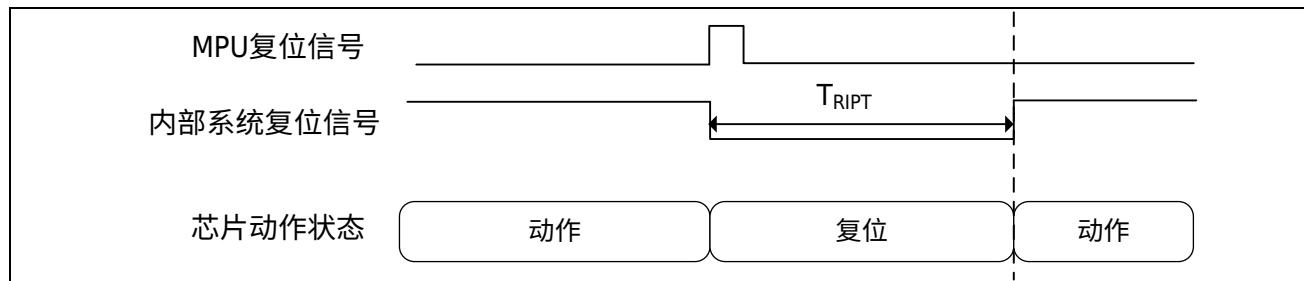


图 3-9 MPU 错误复位

### 3.3.9 RAM 奇偶校验复位

RAM 奇偶校验发生错误时，产生 RAM 奇偶校验复位，时序如图 3-10。RAM 奇偶校验错误将 RMU\_RSTF0.RAPERF 置位。经过内部复位时间  $T_{RIPT}$  之后，芯片解除复位。

RAM 奇偶校验错误复位的设定，请参照【内置 SRAM (SRAM)】。

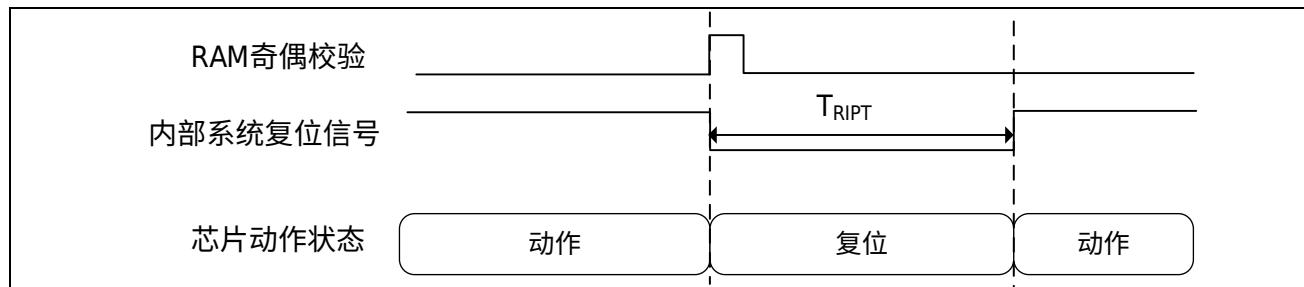


图 3-10 RAM 奇偶校验复位

### 3.3.10 RAMECC 复位

RAMECC 校验发生错误时，产生 RAMECC 复位，时序如图 3-11。RAMECC 复位将 RMU\_RSTF0.RAECRF 置位。经过内部复位时间  $T_{RIPT}$  之后，芯片解除复位。

RAMECC 复位的设定，请参照【内置 SRAM (SRAM)】。

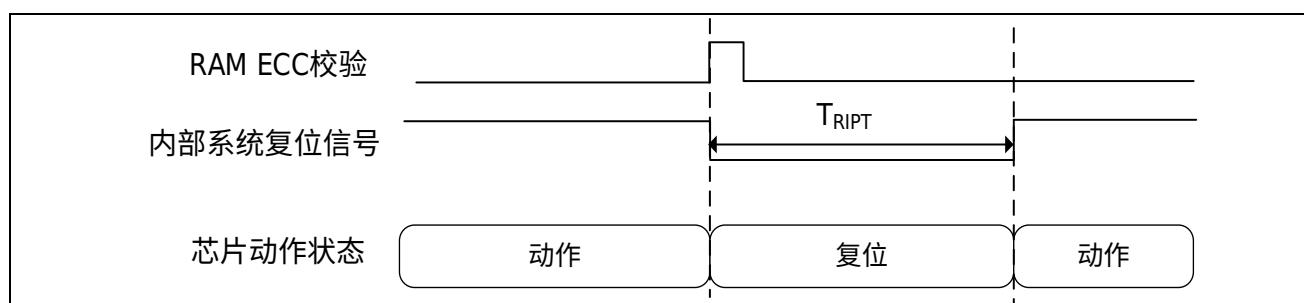


图 3-11 RAMECC 复位

### 3.3.11 时钟频率异常复位

芯片内置的 FCM 模块在监测到时钟频率发生异常时,如果设置为复位有效就会产生时钟频率异常复位,时序如图 3-12。时钟频率异常复位将 RMU\_RSTF0.CKFERF 置位。经过内部复位时间  $T_{RIPT}$  之后,芯片解除复位。

时钟频率异常复位的设定,请参照【时钟频率测量】。

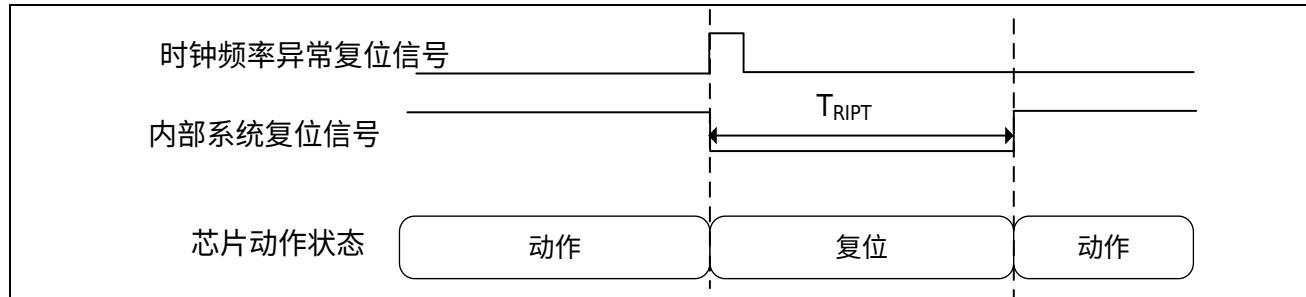


图 3-12 时钟频率异常复位

### 3.3.12 外部高速振荡器异常停振复位

当芯片的震荡停止检测模块有效并且复位使能后,如果发生外部高速振荡器异常停振时,产生外部高速振荡器异常停振复位, RMU\_RSTF0.XTALERF 被置位。经过内部复位时间  $T_{RIPT}$  之后,芯片解除复位。

外部高速振荡器异常停振复位的设定,请参照【外部高速振荡器故障检测】。

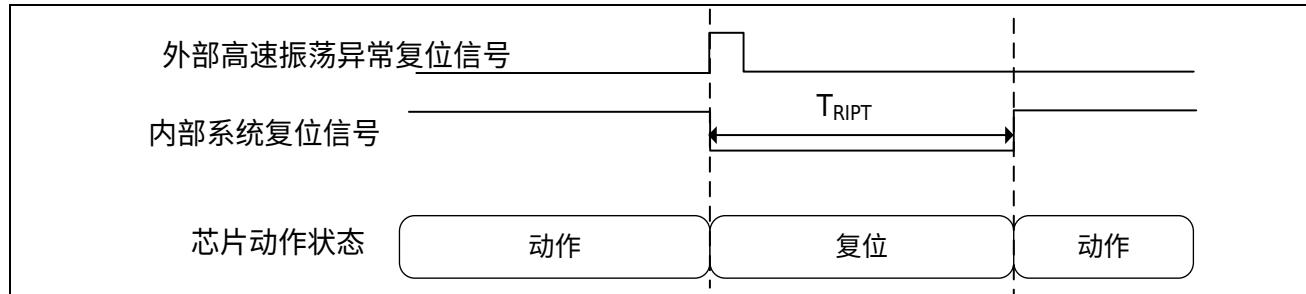


图 3-13 外部高速振荡异常复位

### 3.3.13 Cortex-M4 Lockup 复位

当 Cortex-M4 遇到严重的异常时，它会将自己的 PC 指针停在当前地址处，并锁死自己，并在几个时钟周期延时之后复位整个芯片，时序如图 3-14。Cortex-M4 Lockup 复位将 RSTF0.LKUPRF 置位。经过内部复位时间  $T_{RIPT}$  之后，芯片解除复位。

Cortex-M4 Lockup 复位必须在 RMU\_PRSTCR0.LKUPREN 置位后才能使用。

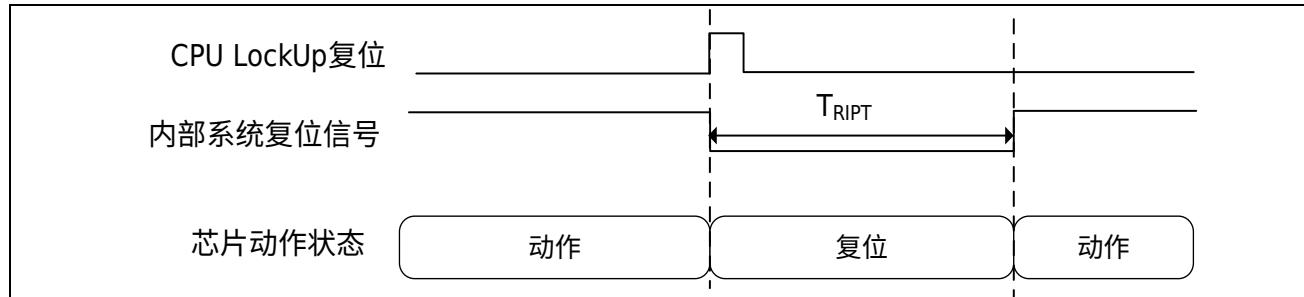


图 3-14 Cortex-M4 Lockup 复位

### 3.3.14 复位方式的判断

根据 RMU\_RSTF0 的复位标志可以判断复位方式。当两个或者两个以上复位同时产生时，可能会产生多个复位标志。RMU\_RSTF0 中的 MULTIRF 位为 1 时表示发生多个复位。在读 RMU\_RSTF0 之后，通过置位 CLRF 位能够将所有的复位标志清 0。在置位将 RMU\_RSTF0 清零后，至少需要等待 6 个 CPU 时钟周期后，才能再次读取 RMU\_RSTF0 寄存器。

## 3.4 功能复位控制

芯片为大部分功能预置了功能复位控制寄存器 RMU\_FRSTx(x=1,2,3,4)。通过往相应功能的复位位写 0，可以将该功能复位到初始状态。RMU\_FRSTx 的详细描述参照本章寄存器部分。

### 3.5 寄存器说明

寄存器一览如表 3-3 所示。

BASE ADDR: 0x4004CCE0

表 3-3 RMU 寄存器一览

寄存器名	符号	偏移地址	位宽	复位值
功能复位控制寄存器0	RMU_FRST0	0xE0	32	0xFFFFFFFF
功能复位控制寄存器1	RMU_FRST1	0xE4	32	0xFFFFFFFF
功能复位控制寄存器2	RMU_FRST2	0xE8	32	0xFFFFFFFF
功能复位控制寄存器3	RMU_FRST3	0xEC	32	0xFFFFFFFF
复位控制寄存器	RMU_PRSTCR0	0x10	32	0x40
复位标志寄存器0	RMU_RSTF0	0x14	32	根据不同的复位方式复位值不同

### 3.5.1 功能复位控制 0(RMU\_FRST0)

复位值:0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	DCU3	DCU2	DCU1	CRC	TRNG	HASH	AES	MAU	CTC	AOS	FCM
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DMA2	DMA1	KEY	-	PLA	-	-	-	-	-	-	-	-	-	-	-

位	标记	位名	功能	读写
b31-b27	Reserved	-	读出时为“11111”，写入时写“11111”	R/W
b26	DCU3	DCU3复位控制	0: 数字计算单元DCU3功能复位 1: 数字计算单元DCU3功能无效	R/W
b25	DCU2	DCU2复位控制	0: 数字计算单元DCU2功能复位 1: 数字计算单元DCU2功能无效	R/W
b24	DCU1	DCU1复位控制	0: 数字计算单元DCU1功能复位 1: 数字计算单元DCU1功能无效	R/W
b23	CRC	CRC复位控制	0: CRC功能复位 1: CRC功能无效	R/W
b22	TRNG	TRNG复位控制	0: 加密协处理模块CPM中的真随机发生器TRNG功能复位 1: 加密协处理模块CPM中的真随机发生器TRNG功能无效	R/W
b21	HASH	HASH复位控制	0: 加密协处理模块CPM的安全散列算法模块HASH功能复位 1: 加密协处理模块CPM的安全散列算法模块HASH功能无效	R/W
b20	AES	AES复位控制	0: 加密协处理模块CPM中的加解密算法处理器AES功能复位 1: 加密协处理模块CPM中的加解密算法处理器AES功能无效	R/W
b19	MAU	MAU复位控制	0: 数学运算单元MAU功能复位 1: 数学运算单元MAU功能无效	R/W
b18	CTC	CTC复位控制	0: 内部时钟校准器CTC功能复位 1: 内部时钟校准器CTC功能无效	R/W
b17	AOS	AOS复位控制	0: 自动运行系统AOS功能复位 1: 自动运行系统AOS功能无效	R/W
b16	FCM	FCM复位控制	0: 时钟控制器CMU中的时钟频率测量模块FCM功能复位 1: 时钟控制器CMU中的时钟频率测量模块FCM功能无效	R/W
b15	DMA2	DMA2复位控制	0: DMA控制器DMA2功能复位 1: DMA控制器DMA2功能无效	R/W
b14	DMA1	DMA1复位控制	0: DMA控制器DMA1功能复位 1: DMA控制器DMA1功能无效	R/W
b13	KEY	KEYSCAN复位控制	0: 键盘扫描控制模块KEYSCAN功能复位 1: 键盘扫描控制模块KEYSCAN功能无效	R/W
b12	Reserved	-	读出时为“1”，写入时写“1”	R/W
b11	PLA	PLA复位控制	0: PLA功能复位 1: PLA功能无效	R/W
b10-b0	Reserved	-	读出时为“111111111111”，写入时写“111111111111”	R/W

### 3.5.2 功能复位控制 1(RMU\_FRST1)

复位值：0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	FMAC4	FMAC3	FMAC2	FMAC1
b23	b22	b21	b20	b19	b18	b17	b16
-	USBFS	-	-	SPI4	SPI3	SPI2	SPI1
b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	-	-	-	MDIO
b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	QSPI	-	-	-

位	标记	位名	功能	读写
b31-b28	Reserved	-	读出时为“1”，写入时写“1”	R/W
b27	FMAC4	FMAC4复位控制	0: 滤波数学加速器FMAC单元4功能复位 1: 滤波数学加速器FMAC单元4功能无效	R/W
b26	FMAC3	FMAC3复位控制	0: 滤波数学加速器FMAC单元3功能复位 1: 滤波数学加速器FMAC单元3功能无效	R/W
b25	FMAC2	FMAC2复位控制	0: 滤波数学加速器FMAC单元2功能复位 1: 滤波数学加速器FMAC单元2功能无效	R/W
b24	FMAC1	FMAC1复位控制	0: 滤波数学加速器FMAC单元1功能复位 1: 滤波数学加速器FMAC单元1功能无效	R/W
b23	Reserved	-	读出时为“1”，写入时写“1”	R/W
b22	USBFS	USBFS复位控制	0: USB2.0全速模块USBFS功能复位 1: USB2.0全速模块USBFS功能无效	R/W
b21~b20	Reserved	-	读出时为“1”，写入时写“1”	R/W
b19	SPI4	SPI4复位控制	0: 串行外设接口SPI单元4功能复位 1: 串行外设接口SPI单元4功能无效	R/W
b18	SPI3	SPI3复位控制	0: 串行外设接口SPI单元3功能复位 1: 串行外设接口SPI单元3功能无效	R/W
b17	SPI2	SPI2复位控制	0: 串行外设接口SPI单元2功能复位 1: 串行外设接口SPI单元2功能无效	R/W
b16	SPI1	SPI1复位控制	0: 串行外设接口SPI单元1功能复位 1: 串行外设接口SPI单元1功能无效	R/W
b15~b9	Reserved	-	读出时为“1”，写入时写“1”	R/W
b8	MDIO	MDIO复位控制	0: MDIO功能复位 1: MDIO功能无效	R/W
b7-b4	Reserved	-	读出时为“1”，写入时写“1”	R/W
b3	QSPI	QSPI复位控制	0: 四线式串行外设接口QSPI功能复位 1: 四线式串行外设接口QSPI功能使能	
b2-b0	Reserved	-	读出时为“111”，写入时写“111”	R/W

### 3.5.3 功能复位控制 2(RMU\_FRST2)

复位值：0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	TMRA	-	-	-	TMR2
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EMB	-	-	TMR0	-	TMR4	-	-	-	-	-	-	-	-	-	TMR6

位	标记	位名	功能	读写
b31~b21	Reserved	-	读出时为“1111111111”，写入时写“1111111111”	R/W
b20	TMRA	TIMERA复位控制	0: TimerA单元1-6功能复位 1: TimerA单元1-6功能复位	R/W
b19-b17	Reserved	-	读出时为“111”，写入时写“111”	R/W
b16	TMR2	TIMER2复位控制	0: TIMER2单元1-4功能复位 1: TIMER2单元1-4功能无效	R/W
b15	EMB	EMB复位控制	0: 紧急刹车模块EMB功能复位 1: 紧急刹车模块EMB功能无效	R/W
b14-b13	Reserved	-	读出时为“11”，写入时写“11”	R/W
b12	TMR0	TIMER0复位控制	0: TIMER0单元1,2功能复位 1: TIMER0单元1,2功能无效	R/W
b11	Reserved	-	读出时为“1”，写入时写“1”	R/W
b10	TMR4	TIMER4复位控制	0: TIMER4功能复位 1: TIMER4功能无效	R/W
b9-b1	Reserved	-	读出时为“11111111”，写入时写“11111111”	R/W
b0	TMR6	TIMER6复位控制	0: TIMER6单元1到单元10功能复位 1: TIMER6单元1到单元10功能无效	R/W

### 3.5.4 功能复位控制 3(RMU\_FRST3)

复位值：0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24
-	-					USART6	USART5
b23	b22	b21	b20	b19	b18	b17	b16
USART4	USART3	USART2	USART1	-			SMC
b15	b14	b13	b12	b11	b10	b9	b8
	-	VREF	OTS	-	-	CMP34	CMP12
b7	b6	b5	b4	b3	b2	b1	b0
DAC4	DAC3	DAC2	DAC1	-	ADC3	ADC2	ADC1

位	标记	位名	功能	读写
b31-b26	Reserved	-	读出时为“111111”，写入时写“111111”	R/W
b25	USART6	USART6复位控制	0: 通用同步异步收发器USART单元6功能复位 1: 通用同步异步收发器USART单元6功能无效	R/W
b24	USART5	USART5复位控制	0: 通用同步异步收发器USART单元5功能复位 1: 通用同步异步收发器USART单元5功能无效	R/W
b23	USART4	USART4复位控制	0: 通用同步异步收发器USART单元4功能复位 1: 通用同步异步收发器USART单元4功能无效	R/W
b22	USART3	USART3复位控制	0: 通用同步异步收发器USART单元3功能复位 1: 通用同步异步收发器USART单元3功能无效	R/W
b21	USART2	USART2复位控制	0: 通用同步异步收发器USART单元2功能复位 1: 通用同步异步收发器USART单元2功能无效	R/W
b20	USART1	USART1复位控制	0: 通用同步异步收发器USART单元1功能复位 1: 通用同步异步收发器USART单元1功能无效	R/W
b19-b17	Reserved	-	读出时为“111”，写入时写“111”	R/W
b16	SMC	SMC复位控制	0: 外部存储控制器(EXMC)的SMC控制器功能有效 1: 外部存储控制器(EXMC)的SMC控制器功能无效	R/W
b15-b14	Reserved	-	读出时为“11”，写入时写“11”	R/W
b13	VREF	VREF复位控制	0: VREF功能有效 1: VREF功能无效	R/W
b12	OTS	OTS复位控制	0: 温度传感器OTS功能有效 1: 温度传感器OTS功能无效	R/W
b11-b10	Reserved	-	读出时为“11”，写入时写“11”	R/W
b9	CMP34	CMP34复位控制	0: 电压比较器CMP单元3和单元4功能复位有效 1: 电压比较器CMP单元3和单元4功能复位解除	R/W
b8	CMP12	CMP12复位控制	0: 电压比较器CMP单元1和单元2功能复位有效 1: 电压比较器CMP单元1和单元2功能复位解除	R/W
b7-b6	Reserved	-	读出时为“11”，写入时写“11”	R/W
b7	DAC4	DAC4复位控制	0: 数模转换器DAC单元4功能复位 1: 数模转换器DAC单元4功能无效	R/W
b6	DAC3	DAC3复位控制	0: 数模转换器DAC单元3功能复位 1: 数模转换器DAC单元3功能无效	R/W

b5	DAC2	DAC2复位控制	0: 数模转换器DAC单元2功能复位 1: 数模转换器DAC单元2功能无效	R/W
b4	DAC1	DAC1复位控制	0: 数模转换器DAC单元1功能复位 1: 数模转换器DAC单元1功能无效	R/W
b3	Reserved	-	读出时为“1”，写入时写“1”	R/W
b2	ADC3	ADC3复位控制	0: 模数转换模块ADC单元3功能复位 1: 模数转换模块ADC单元3功能无效	R/W
b1	ADC2	ADC2复位控制	0: 模数转换模块ADC单元2功能复位 1: 模数转换模块ADC单元2功能无效	R/W
b0	ADC1	ADC1复位控制	0: 模数转换模块ADC单元1功能复位 1: 模数转换模块ADC单元1功能无效	R/W

### 3.5.5 复位控制寄存器 (RMU\_PRSTCR0)

复位值：0x40

b7	b6	b5	b4	b3	b2	b1	b0
-	-	LKUPREN	-	-	-	-	-

位	标记	位名	功能	读写
b7	Reserved	-	读出时为“0”，写入时写“0”	R/W
b6	Reserved	-	读出为1，写入1。	R/W
b5	LKUPREN	LOCKUP复位使能	0: LOCKUP复位无效 1: LOCKUP复位使能	R/W
b4-b0	Reserved	-	读出时为“0”，写入时写“0”	R/W

### 3.5.6 复位标志寄存器 0 (RMU\_RSTF0)

复位值：0xFFFFh (根据复位方式不同，复位值不同)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CLRF	MULTIRF	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	LKUPRF	XTALERF	CKFERF	RAECRF	RAPERF	MPUERF	SWRF	PDRF	SWDRF	WDRF	PVD2RF	PVD1RF	BORF	PINRF	PORF

位	标记	位名	功能	读写
b31	CLRF	清除复位 标志	软件置1，用于清除复位标志位。 读出时为0。置位动作必须在读取RMU_RSTF0之后进行。 0：无操作 1：清零复位标志	R/W
b30	MULTIRF	2个以上复位 发生标志位	发生两个及两个以上复位时，由硬件置位。 通过置位CLRF清零。 0：未发生两个及两个以上复位 1：发生两个及两个以上复位时	R/W
b29-b15	Reserved	-	读出时为“0”，写入时写“0”	R/W
b14	LKUPRF	Cortex-M4 Lockup复位	发生Cortex-M4 Lockup复位时，由硬件置位。 通过置位CLRF清零。 0：未发生Cortex-M4 Lockup复位 1：发生Cortex-M4 Lockup复位	R/W
b13	XTALERF	外部高速振荡器异常 停振复位标志	发生外部高速振荡器异常停振复位时，由硬件置位。 通过置位CLRF清零。 0：未发生外部高速振荡器异常停振复位 1：发生外部高速振荡器异常停振复位	R/W
b12	CKFERF	时钟频率异常 复位标志	发生时钟频率异常复位时，由硬件置位。 通过置位CLRF清零。 0：未发生时钟频率异常复位 1：发生时钟频率异常复位	R/W
b11	RAECRF	RAMECC复位 标志	发生RAMECC复位时，由硬件置位。 通过置位CLRF清零。 0：未发生RAMECC复位 1：发生RAMECC复位	R/W
b10	RAPERF	RAM奇偶错误 复位标志	发生RAM奇偶校验错误复位时，由硬件置位。通过置位CLRF清零。 0：未发生RAM奇偶校验错误复位 1：发生RAM奇偶校验错误复位	R/W
b9	MPUERF	MPU错误复位标志	发生MPU错误复位时，由硬件置位。 通过置位CLRF清零。 0：未发生MPU错误复位 1：发生MPU错误复位	R/W
b8	SWRF	软件复位标志	发生软件复位复位时，由硬件置位。 通过置位CLRF清零。 0：未发生软件复位复位 1：发生软件复位复位	R/W
b7	PDRF	掉电模式复位	发生掉电模式复位时，由硬件置位。	R/W

			通过置位清零。	
			0：未发生掉电模式复位	
			1：发生掉电模式复位	
			发生专用看门狗复位时，由硬件置位。	
b6	SWDRF	专用看门狗复位标志	通过置位清零。 0：未发生专用看门狗复位 1：发生专用看门狗复位	R/W
b5	WDRF	看门狗复位标志	发生看门狗复位时，由硬件置位。 通过置位清零。 0：未发生看门狗复位 1：发生看门狗复位	R/W
b4	PVD2RF	可编程电压检测2复位标志	发生可编程电压检测2复位时，由硬件置位。 通过置位清零。 0：未发生可编程电压检测2复位 1：发生可编程电压检测2复位	R/W
b3	PVD1RF	可编程电压检测1复位标志	发生可编程电压检测1复位时，由硬件置位。 通过置位清零。 0：未发生可编程电压检测1复位 1：发生可编程电压检测1复位	R/W
b2	BORF	欠压复位标志	发生欠压复位时，由硬件置位。 通过置位清零。 0：未发生欠压复位 1：发生欠压复位	R/W
b1	PINRF	NRST引脚复位标志	发生引脚复位时，由硬件置位。 通过置位清零。 0：未发生NRST复位 1：发生NRST复位	R/W
b0	PORF	上电复位标志	发生上电复位时，由硬件置位。 通过置位清零。 0：未发生上电复位 1：发生上电复位	R/W

## 4 时钟控制器 (CMU)

### 4.1 简介

时钟控制单元提供了一系列频率的时钟功能，包括：一个外部高速振荡器、一个外部低速振荡器、一个 PLL 时钟、一个内部高速振荡器、一个内部中速振荡器、一个内部低速振荡器、一个 SWDT 专用内部低速振荡器、时钟预分频器、时钟多路复用和时钟门控电路。

时钟控制单元还提供时钟频率测量功能。时钟频率测量电路 (FCM) 使用测定基准时钟对测定对象时钟进行监视测定。在超出设定范围时发生中断或者复位。

AHB、APB 和 Cortex-M4 时钟都源自系统时钟，系统时钟的源可选择 6 个时钟源：

1. 外部高速振荡器 (XTAL)
2. 外部低速振荡器 (XTAL32)
3. PLLH 时钟 (PLLH)
4. 内部高速振荡器 (HRC)
5. 内部中速振荡器 (MRC)
6. 内部低速振荡器 (LRC)

系统时钟的最大运行时钟频率可以达到 120MHz。SWDT 有独立的时钟源：SWDT 专用内部低速振荡器 (SWDTLRC)。实时时钟 (RTC) 使用外部低速振荡器或者内部低速振荡器作为时钟源。USB-FS 的 48MHz 时钟可以选择系统时钟、PLLH 作为时钟源。

对于每一个时钟源，在未使用时都可以单独打开和关闭，以降低功耗。

## 4.2 系统框图

### 4.2.1 系统框图

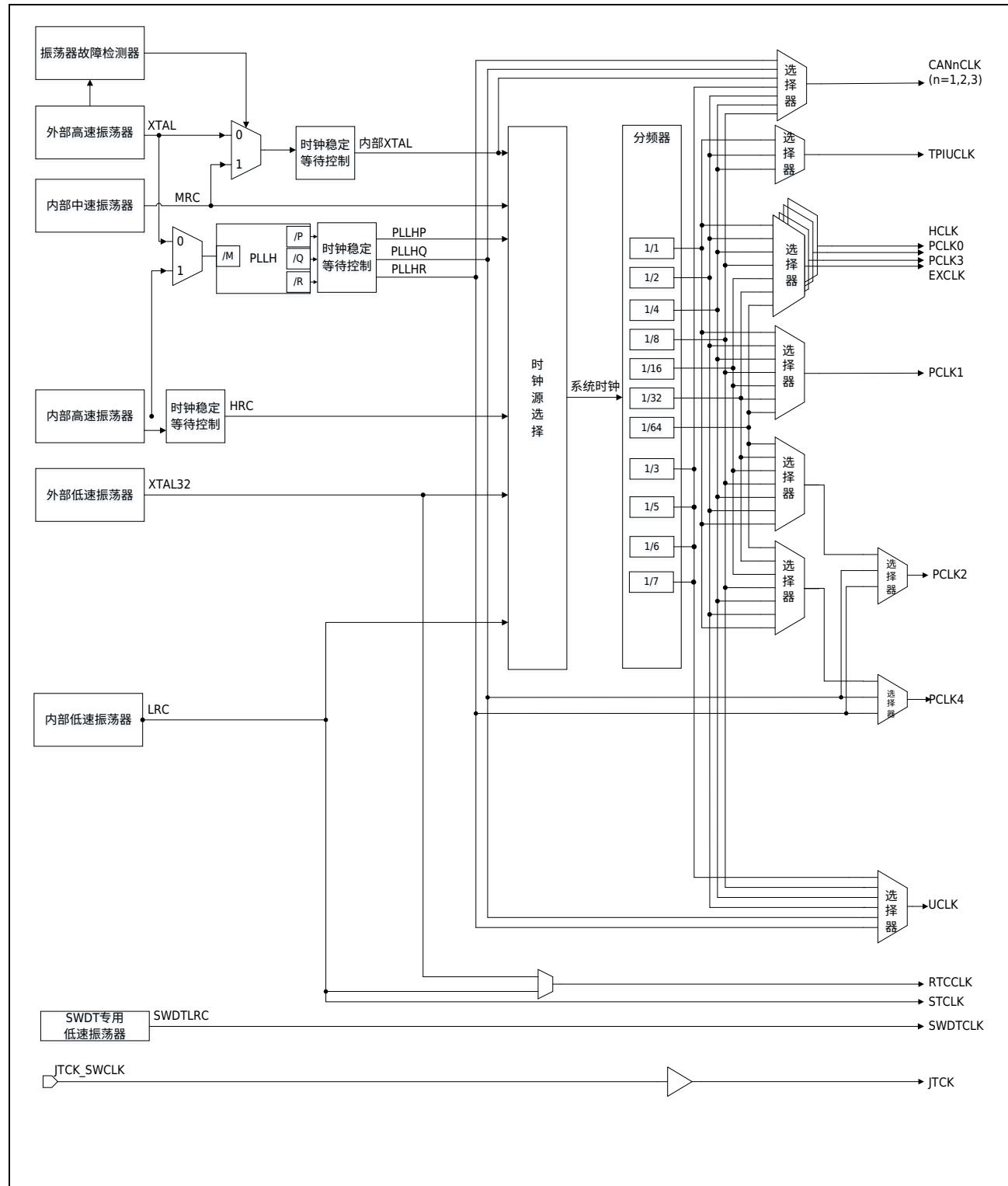


图 4-1 时钟系统框图

#### 4.2.2 时钟频率测量框图

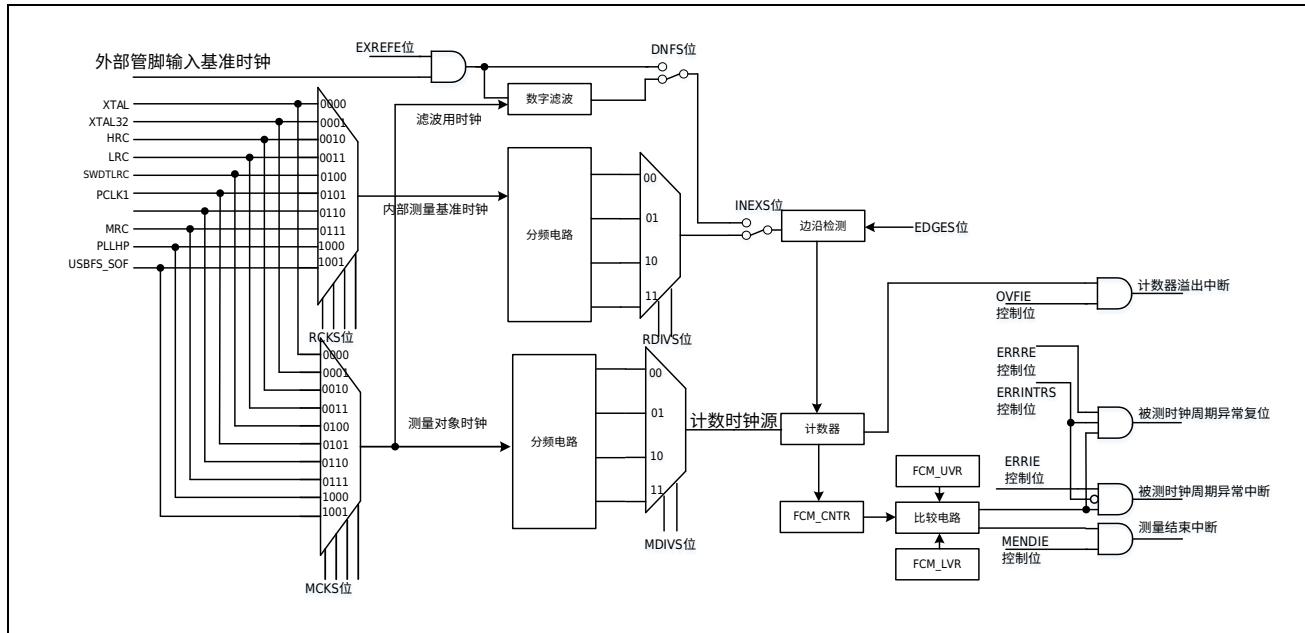


图 4-2 时钟频率测量框图

## 4.3 时钟源规格

各时钟源的主要特性如下表所示。

表 4-1 时钟源主要特性

时钟源	规格
外部高速振荡器 (XTAL)	晶振的频率范围：4~25MHz 外部时钟输入：最高 25MHz 振荡器故障检测功能
外部低速振荡器 (XTAL32)	晶振的频率：32.768kHz
PLLH 时钟 (PLLH)	输入时钟：外部高速振荡器或者内部高速振荡器 PLLH 输入时钟分频：1~25 任意分频可选 PFD 输入时钟频率=输入时钟/PLLH 输入时钟分频，频率范围 1MHz~25MHz PLLH 倍频系数：20~480 倍 VCO 振荡频率：240MHz~480MHz PLLHQ 输出分频比：2~16 任意分频 PLLHP 输出分频比：2~16 任意分频 PLLHR 输出分频比：2~16 任意分频 PLLHP 输出频率=（输入时钟/PLLH 输入时钟分频）*PLLH 倍频系数/PLLHP 输出分频比 PLLHQ 输出频率=（输入时钟/PLLH 输入时钟分频）*PLLH 倍频系数/PLLHQ 输出分频比 PLLHR 输出频率=（输入时钟/PLLH 输入时钟分频）*PLLH 倍频系数/PLLHR 输出分频比
内部高速振荡器 (HRC)	频率：16MHz 或者 20MHz 用户可写寄存器对频率微调
内部中速振荡器 (MRC)	频率：8MHz 用户可写寄存器对频率微调
内部低速振荡器 (LRC)	频率：32.768kHz 用户可写寄存器对频率微调
SWDT 专用内部低速振荡器 (SWDTLRC)	频率：10kHz

## 4.4 工作时钟规格

表 4-2 各个内部时钟的规格

时钟	作用范围	规格
HCLK	CPU、DMA(n=1、2)、EFM、SRAM0、SRAMH、 MPU、GPIO、DCUn(n=1~3)、INTC、QSPI、 FIRn(n=1~4)、MAU、KEYSCAN	最高频率 120MHz。 由 CMU_SCFG 寄存器 HCLKS 位配置。 可选系统时钟的分频：1, 2, 4, 8, 16, 32, 64
PCLK0	Timer6、Timer4、TimerAn(n=1~4)、 I2Cn(n=1~3)、CMPl(n=1~2)、CANn(n=1~3)控 制逻辑	最高频率 120MHz 由 CMU_SCFG 寄存器 PCLK0S 位配置。 可选系统时钟的分频：1, 2, 4, 8, 16, 32, 64
PCLK1	USBFS 控制逻辑、USARTn (n=1~6)、 SPl(n=1~4)、Timer0n(n=1、2)、 Timer2n(n=1~4)、TimerAn(n=5~6)、EMB、 CRC、HASH、AES、	最高频率 60MHz 由 CMU_SCFG 寄存器 PCLK1S 位配置。 可选系统时钟的分频：1, 2, 4, 8, 16, 32, 64
PCLK2	ADC 转换时钟	最高频率 60MHz 由 CMU_SCFG 寄存器的 PCLK2S 位和 CMU_PERICKSEL 寄存器 PERICKSEL 位配置。 可选系统时钟的分频：1, 2, 4, 8, 16, 32, 64 可选独立时钟源： PLLHQ,PLLHR
PCLK3	RTC (控制逻辑)、WDT、SWDT (控制逻辑)、 WKTM、OTS、FCM、CTC、PLA、VREF 控制逻辑	最高频率 60MHz 由 CMU_SCFG 寄存器 PCLK3S 位配置。 可选系统时钟的分频：1, 2, 4, 8, 16, 32, 64
PCLK4	ADCn(n=1~3) (控制逻辑)、DACn(n=1~4) (控 制逻辑)	最高频率 60MHz 由 CMU_SCFG 寄存器的 PCLK4S 位和 CMU_PERICKSEL 寄存器 PERICKSEL 位配置。 可选系统时钟的分频：1, 2, 4, 8, 16, 32, 64 可选独立时钟源： PLLHQ,PLLHR
EXCLK	SMC	最高频率 60MHz 由 CMU_SCFG 寄存器的 EXCKS 位配置。 可选系统时钟的分频：1, 2, 4, 8, 16, 32, 64
UCLK	USBFS FullSpeed 通信用时钟	频率 48MHz 由 CMU_USBCKCFGR 寄存器的 USBCKS 位配置。 可选系统时钟分频 2, 3, 4, 5, 6, 7, 8。 可选独立时钟源： PLLHQ,PLLHR
CANnCLK	CAN1/CAN2/CAN3 通信时钟	最高频率 80MHz 由 CMU_CANCKCFGR 寄存器的 CAN3CKS/CAN2CKS/ CAN1CKS 位配置。 可选系统时钟分频 2, 3, 4, 5, 6, 7, 8。

时钟	作用范围	规格
		可选独立时钟源: PLLHQ,PLLHR,XTAL
STCLK	SYSTICK 定时器外部参考时钟	内部低速振荡器时钟。 CPU 的 SYSTICK 控制和状态寄存器的 CLKSOURCE 位配置 SYSTICK 定时器时钟。
SWDTCLK	SWDT 计数器用时钟	频率 10kHz
TCK	JTAG 用时钟	最高频率 25MHz
TPIUCLK	Cortex-M4 调试跟踪器用时钟	最高频率 30MHz 由 CMU_TPIUCKCFGR 寄存器的 TPIUCKS 位配置。 可选系统时钟的分频：1, 2, 4

各时钟之间需遵守下列规则：

- HCLK 频率 $\geq$ PCLK0、PCLK1、PCLK2、PCLK3、PCLK4 频率
- PCLK0 频率 $\geq$ PCLK1 频率, PCLK0 频率 $\geq$ PCLK3 频率
- PCLK2 频率: PCK4 频率=1:8,1:4,1:2,1:1,2:1,4:1,8:1
- SWDT 使用时: PCLK3 频率 > SWDTCLK 频率

## 4.5 晶振电路

### 4.5.1 外部高速振荡器

#### 4.5.1.1 振荡器模式

外部高速振荡器可为系统时钟提供更为精确时钟源。

XTAL 通过 CMU\_XTALCR 的 XTALSTP 位打开和关闭。

CMU\_OSCSTBSR 的 XTALSTBF 标志位指示外部高速振荡器是否稳定，稳定时间通过寄存器 CMU\_XTALSTBCR 配置。CMU\_XTALSTBCR 设定的稳定时间一定要大于等于晶振厂商要求的稳定时间。

晶振的电路常数因晶振和安装电路的寄生电容而不同，因此必须和晶振厂商仔细商谈后决定。振荡器的各种特性与用户的电路板设计密接相关，晶振和负载电容必须尽可能地靠近振荡器的引脚，以尽量减小输出失真和起振稳定时间。负载电容值必须根据所选振荡器的不同做适当调整。在振荡电路附近不能通过信号线，否则就可能因电感而不能正常振荡。

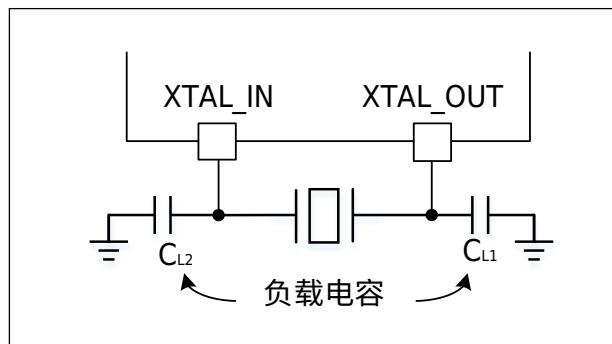


图 4-3 外部高速振荡器连接事例

#### 4.5.1.2 时钟输入模式

时钟输入模式下，必须提供外部时钟源。此模式通过 CMU\_XTALCFGR 的 XTALMS 位置“1”和 CMU\_XTALCR 的 XTALSTP 位置“0”进行选择。必须使用占空比约为 50%的外部时钟信号来驱动 XTAL\_IN 引脚。此时 XTAL\_OUT 引脚可根据寄存器设定配置成 GPIO。

外部时钟输入的连接例如下图所示。

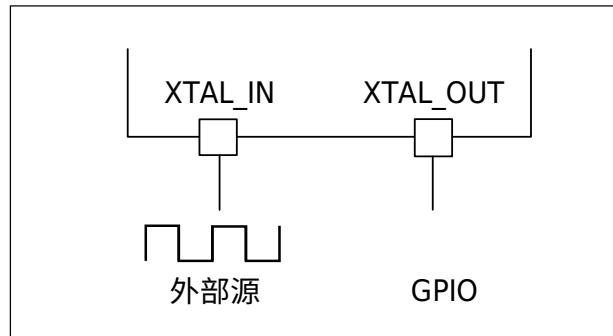


图 4-4 外部时钟输入的连接事例图

#### 4.5.2 外部高速振荡器故障检测

振荡器故障检测是检测外部高速振荡器（XTAL）振荡是否正常。

通过寄存器 CMU\_XTALSTDCR 的 XTALSTDE 位打开或关闭。

复位解除后，外部高速振荡器停止振荡，外部高速振荡器故障检测功能无效。要将外部高速振荡器故障检测功能置为有效时，必须使外部高速振荡器振荡，并且在等到外部高速振荡器稳定即 CMU\_OSCSTBSR.XTALSTBF 为 1，通过寄存器 CMU\_XTALSTDCR 的 XTALSTDE 位打开。

PLLH 选择 XTAL 时钟作为输入源时，只能选择 XTAL 振荡故障产生复位功能。

因为振荡器故障检测是检测外部因素导致的振荡器异常振荡，所以要通过软件使外部高速振荡器停止振荡或者转移到停止模式和掉电模式时，将振荡器振荡故障检测功能无效。

如果外部高速振荡器发生故障，动作波形如下图所示。操作流程参照【检测到 XTAL 故障检测动作】。

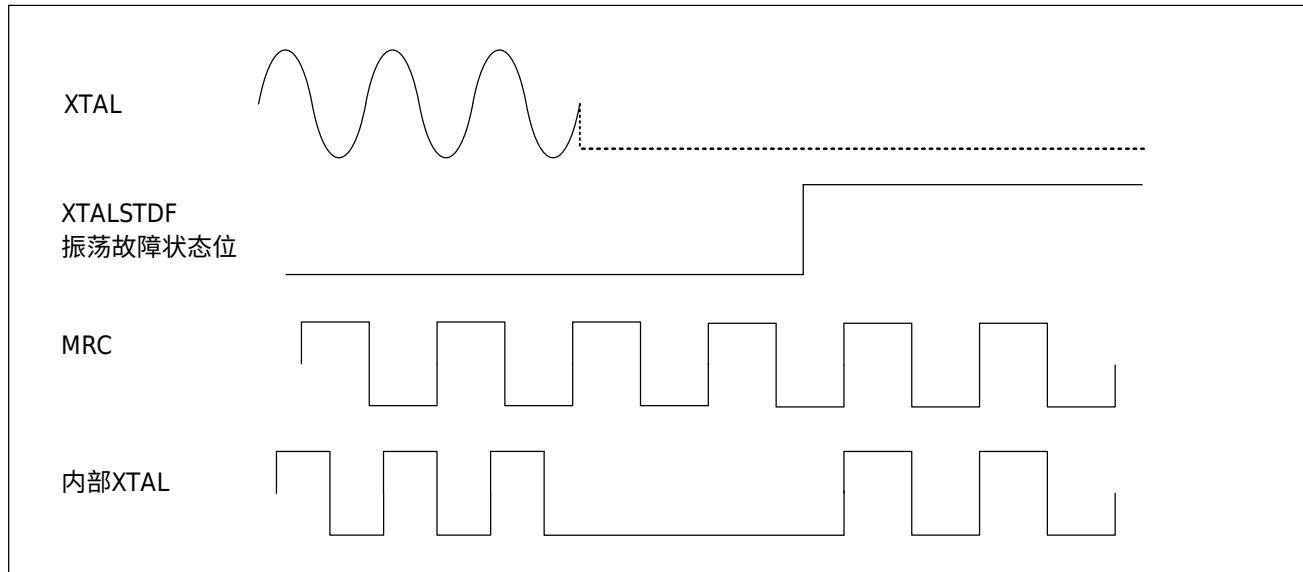


图 4-5 外部高速振荡器故障检测例

#### 4.5.2.1 检测到 XTAL 故障检测动作

当检测到外部高速振荡器振荡故障时，如果系统时钟选择外部高速振荡器作为系统时钟，系统时钟会自动切换到 MRC。

当检测到外部高速振荡器振荡故障时，可触发 EMB，将 Timer6/Timer4 的 PWM 输出置成 Hz 输出。参考【紧急刹车模块（EMB）】章。

系统时钟选择成 XTAL，检测到 XTAL 故障时，动作例如下图所示。

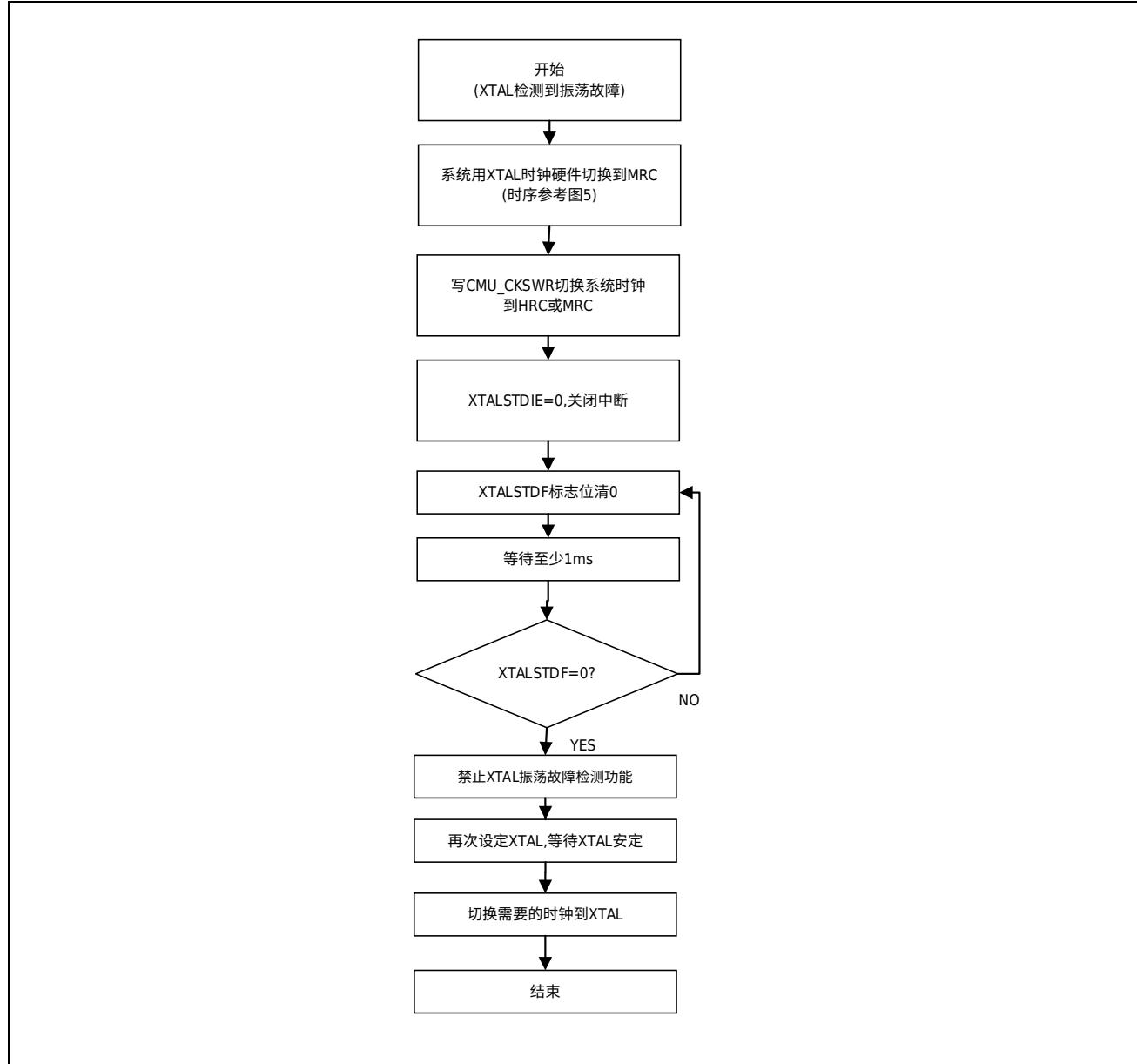


图 4-6 系统时钟选择 XTAL，检测到 XTAL 振荡故障例

#### 4.5.2.2 检测到 XTAL 振荡故障产生中断复位

XTAL 振荡故障中断可配置成可屏蔽中断或非可屏蔽中断，参考【中断控制器（INTC）】章。

XTAL 振荡故障配置成复位时，检测到 XTAL 振荡故障，芯片产生复位，复位动作参考【复位控制（RMU）】章。

#### 4.5.3 外部高速振荡器小数分频

通过设定 CMU\_XTALDIVR 寄存器，外部高速振荡器时钟可以分频到 32.768KHz 供 RTC 使用。

外部高速振荡器 10MHz 分频到 32.768KHz 设定例：

$$\text{分频数} = \frac{10000000\text{Hz}}{32768\text{Hz}} = \frac{78125 \text{ (10 进制)}}{256 \text{ (10 进制)}} = \frac{1312D \text{ (16 进制)}}{100 \text{ (16 进制)}}$$

numerator[16:0]寄存器设定为 0x1312D。denominator[10:0]寄存器设定为 0x100。即寄存器 CMU\_XTALDIVR 设定为 0x1312D100。再通过 CMU\_XTALDIVCR 设定开始。

外部高速振荡器 8MHz 分频到 32.768KHz 设定例：

$$\text{分频数} = \frac{8000000\text{Hz}}{32768\text{Hz}} = \frac{31250 \text{ (10 进制)}}{128 \text{ (10 进制)}} = \frac{7A12 \text{ (16 进制)}}{80 \text{ (16 进制)}}$$

numerator[16:0]寄存器设定为 0x7A12。denominator[10:0]寄存器设定为 0x80。即寄存器 CMU\_XTALDIVR 设定为 0x07A12080。再通过 CMU\_XTALDIVCR 设定开始。

#### 4.5.4 外部低速振荡器

32.768kHz 的外部低速振荡器可为系统时钟、实时时钟电路（RTC）提供更为精确时钟源。具有功耗低且精度高的优点。

XTAL32 通过 CMU\_XTAL32CR 的 XTAL32STP 位打开和关闭。

晶振的电路常数因晶振和安装电路的寄生电容而不同，因此必须和晶振厂商仔细商谈后决定。振荡器的各种特性与用户的电路板设计密接相关，晶振和负载电容必须尽可能地靠近振荡器的引脚，以尽量减小输出失真和起振稳定时间。负载电容值必须根据所选驱动能力不同做适当调整。在振荡电路附近不能通过信号线，否则就可能因电感而不能正常振荡。

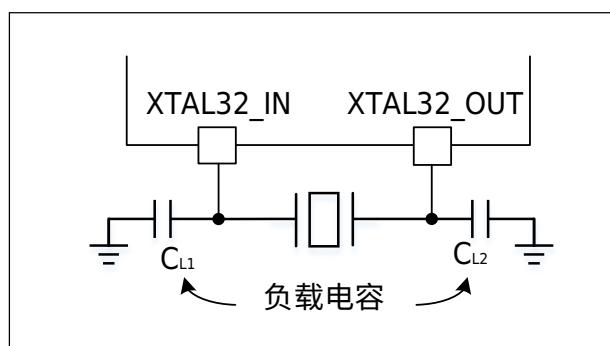


图 4-7 外部低速振荡器连接事例

XTAL32 初次上电的初始化流程如下所示：

1. CMU\_XTAL32CR.XTAL32STP 位写 1，停止 XTAL32
2. 通过 CMU\_XTAL32FGR 设定匹配的 XTAL32 驱动能力
3. 通过 CMU\_XTAL32NFR 设定滤波功能
4. CMU\_XTAL32CR.XTAL32STP 位写 0，XTAL32 振荡
5. 软件等待 XTAL32 稳定，稳定时间参考电器特性章节

如果不使用外部低速振荡器，将 CMU\_XTAL32CR 的 XTAL32STP 位设 1，关闭外部低速振荡器。

## 4.6 内部 RC 时钟

### 4.6.1 HRC 时钟

HRC 时钟信号由内部高速振荡器生成，可直接用作系统时钟，或者用作 PLLH 输入。HRC 的频率可由 ICG1. HRCFREQSEL 配置成 16MHz 或者 20MHz。

HRC 振荡器的优点是成本较低（无需使用外部组件）。此外，其启动速度也要比 XTAL 晶振块，但即使校准后，其精度也不及外部晶振。

#### 频率校准

因为生产工艺不同，不同芯片的 RC 振荡器频率也不同，因此会对每个器件进行出厂校准，保证精度参考数据手册电气特性中内部高速（HRC）振荡器章节。

如果应用受到温度变化影响，则这可能也会影响到 RC 振荡器的速度。用户可通过寄存器对 HRC 频率进行微调。

CMU\_OSCSTBSR 中的 HRCSTBF 标志指示 HRC 是否稳定。在启动时，硬件将此位置 1 后，HRC 才可以使用。

HRC 可通过 CMU\_HRCCR 控制寄存器中的 HRCSTP 位打开或关闭。

### 4.6.2 MRC 时钟

MRC 时钟信号由内部 8MHz 中速振荡器生成，可直接用作系统时钟。

MRC 振荡器的优点是启动速度快。

#### 频率校准

因为生产工艺不同，不同芯片的 RC 振荡器频率也不同，因此会对每个器件进行出厂校准，保证精度参考数据手册电气特性中内部中速（MRC）振荡器章节。

如果应用受到温度变化影响，则这可能也会影响到 RC 振荡器的速度。用户可通过寄存器对 MRC 频率进行微调。

MRC 可通过 CMU\_MRCCR 控制寄存器中的 MRCSTP 位打开或关闭。

MRC 时钟还可作为备份时钟源使用，以防 XTAL 晶振发生故障。请参见【检测到 XTAL 故障检测动作】。

### 4.6.3 LRC 时钟

LRC 时钟信号由内部 32.768kHz 低速振荡器生成，可直接用作系统时钟。LRC 可作为低功耗时钟源在掉电模式和停止模式下保持运行，供 Timer0/KEYSCAN 使用。

LRC 振荡器的启动速度快。

#### 频率校准

因为生产工艺不同，不同芯片的 RC 振荡器频率也不同，因此会对每个器件进行出厂校准，保证精度参考数据手册电气特性中内部低速（LRC）振荡器章节。

如果应用受到电压或温度变化影响，则这可能也会影响到 RC 振荡器的速度。用户可通过寄存器对 LRC 频率进行微调。

LRC 可通过 CMU\_LRCCR 控制寄存器中的 LRCSTP 位打开或关闭。

### 4.6.4 SWDTLRC 时钟

SWDTLRC 时钟信号由内部 10kHz 低速振荡器生成，SWDT 专属时钟。SWDT 已通过 ICG 设置的方式启动，则 SWDT 专用内部低速振荡器将强制打开且不可禁止。

因为生产工艺不同，不同芯片的 RC 振荡器频率也不同，因此会对每个器件进行出厂校准，保证精度参考数据手册电气特性中 SWDT 专用内部低速（SWDTLRC）振荡器章节。

## 4.7 PLL 时钟

HC32F472 器件具有一个 PLL：

- PLLH 由 XTAL 或 HRC 振荡器提供时钟信号，并具有三个不同的输出时钟：
  - P 分频器输出用于生成系统时钟（最高达 120 MHz）
  - 三个输出都可用于生成 USB、ADC、CAN 时钟。

PLLH 输入时钟源，可选择 HRC 或 XTAL 振荡器作为时钟源，由 CMU\_PLLHCFGR.PLLSRC 位配置。在 HRC 或 XTAL 振荡器稳定后，再对 PLL 进行配置。

PLLH 的分频系数 M、N、P、Q、R 可独立配置。由于在 PLL 使能后 PLL 配置参数便不可更改，所以建议先对 PLL 进行配置，然后再使能。

当进入掉电和停止模式后，PLL 将由硬件禁止。

## 4.8 时钟切换步骤

在系统复位后，默认系统时钟为 MRC。通过设定寄存器 CMU\_CKSW 切换时钟源，切换步骤参照时钟源切换。只有在目标时钟源已稳定的状态下，才可以从一个时钟源切换到另一个时钟源。

时钟切换时需要正确配置 Flash 的等待周期，防止系统时钟频率大于 Flash 的最大动作频率。参照【CPU 时钟和 Flash 读取时间之间的关系】章节进行配置。

#### 4.8.1 时钟源切换

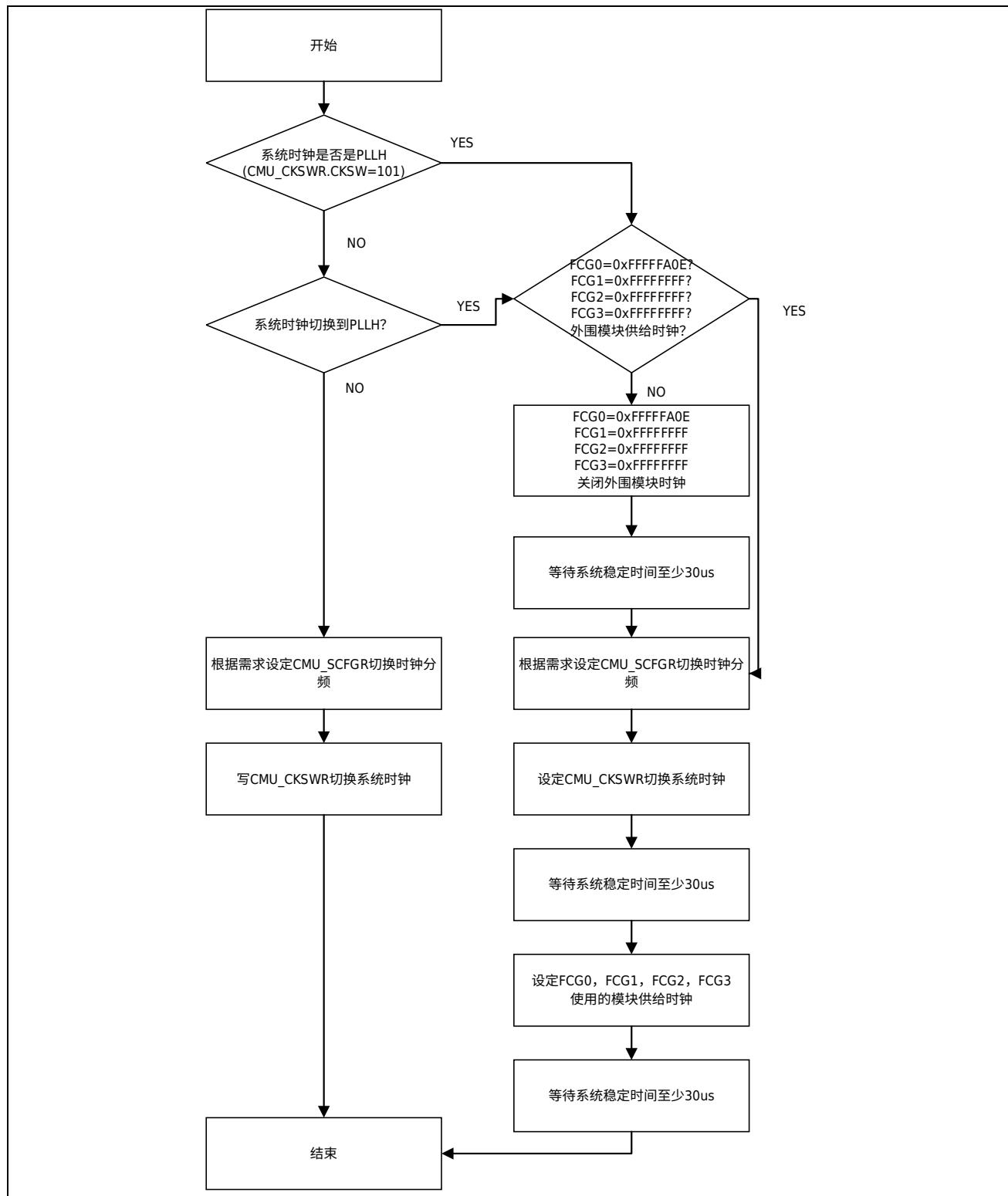


图 4-8 时钟源切换

#### 4.8.2 时钟分频切换

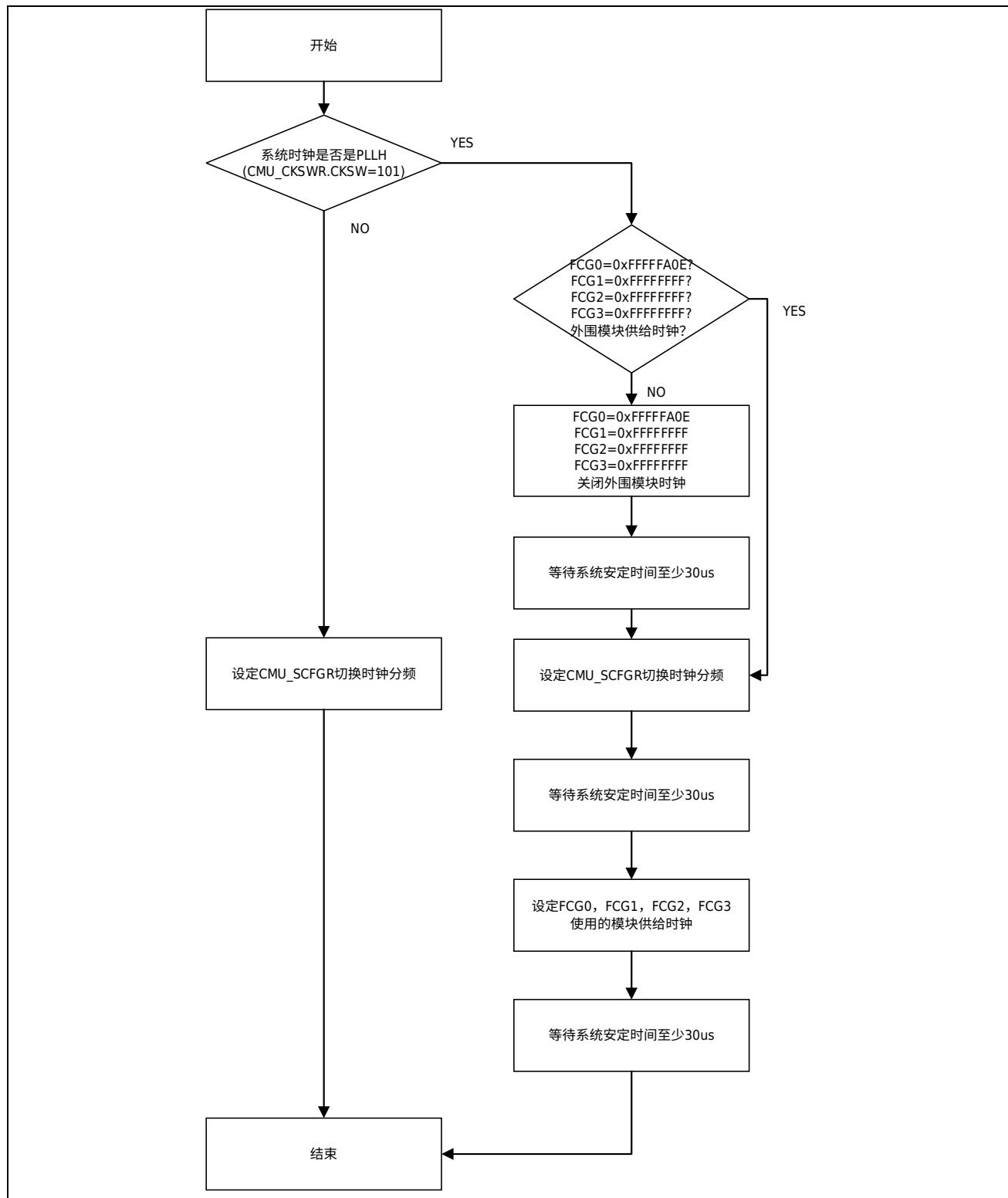


图 4-9 时钟分频切换

## 4.9 时钟输出功能

共有两路时钟输出：

■ MCO\_1

用户可通过可配置的预分配器（从 1 到 128）向 MCO\_1 引脚输出不同的时钟源：

- HRC 时钟
- MRC 时钟
- LRC 时钟
- XTAL 时钟
- XTAL32 时钟
- PLLHP/PLLHQ 时钟
- 系统时钟

所需的时钟源通过 CMU\_MCO1CFG.R.MCO1SEL 位选择。

■ MCO\_2

用户可通过可配置的预分配器（从 1 到 128）向 MCO\_2 引脚输出不同的时钟源：

- HRC 时钟
- MRC 时钟
- LRC 时钟
- XTAL 时钟
- XTAL32 时钟
- PLLHP/PLLHQ 时钟
- 系统时钟

所需的时钟源通过 CMU\_MCO2CFG.R.MCO2SEL 位选择。

MCO\_1/MCO\_2 输出时钟不得超过 100 MHz（最大 I/O 速度）。

## 4.10 时钟频率测量

### 4.10.1 时钟频率测量

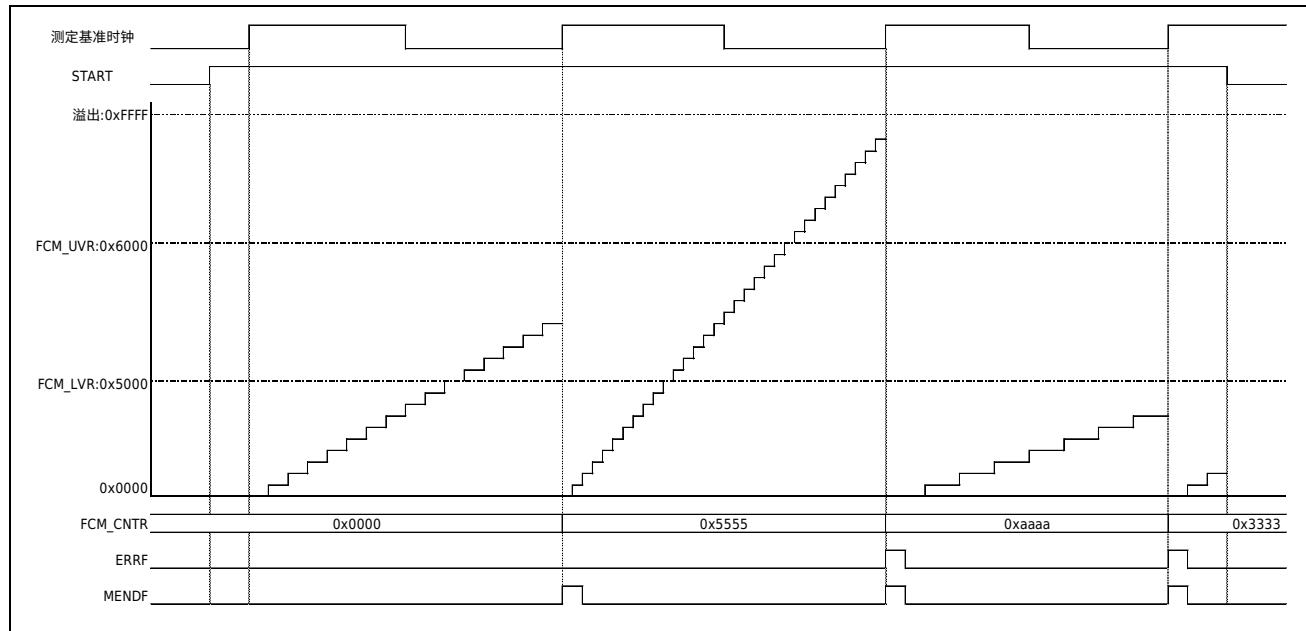


图 4-10 时钟频率测量时序图

1. 用 FCM\_MCCR/FCM\_RCCR 选择基准时钟被测量时钟，时钟的分频以及选择基准时钟的有效边沿。
2. FCM\_STR 的 START 位写 1 后，检测到 EDGES 位选择的有效边沿，计数器就开始递增计数。
3. 在检测到基准时钟下一个 EDGES 位选择的有效边沿时，将计数器的值保存到 FCM\_CNTR 寄存器，并且与 FCM\_LVR/FCM\_UVR 的设定值进行比较。当  $FCM\_LVR \leq FCM\_CNTR \leq FCM\_UVR$  时，被测时钟频率测量正常。当  $FCM\_LVR > FCM\_CNTR$  或者  $FCM\_CNTR > FCM\_UVR$  时，被测时钟频率异常，根据 ERRINTRS/ERRRE/ERRIE 设定可以发生中断或者复位。
4. FCM\_STR 的 START 位写 0 后，计数器计数停止并清零。

#### 4.10.2 数字滤波功能

外部管脚输入参考时钟 FCMREF 具有数字滤波功能。数字滤波器功能根据 DNFS 位选择的采样时钟进行 3 次采样，3 次采样的电平相同时，将此电平送到内部。

数字滤波功能可以设定数字滤波功能有效无效以及采样时钟。

#### 4.10.3 中断/复位功能

时钟频率测量电路有三种中断请求。分别是：

- 1) 频率异常中断
- 2) 频率测量结束中断
- 3) 计数器溢出中断

时钟频率测量电路有一种复位请求：

- 1) 频率异常复位

## 4.11 寄存器说明

表 4-3 寄存器一览

基准地址 1: 0x40048400

寄存器名	符号	偏移地址	位宽	复位值
FCM下限比较值寄存器	FCM_LVR	0x00	32	0x00000000
FCM上限比较值寄存器	FCM_UVR	0x04	32	0x00000000
FCM计数器值寄存器	FCM_CNTR	0x08	32	0x00000000
FCM开始停止寄存器	FCM_STR	0x0C	32	0x00000000
FCM测量对象控制寄存器	FCM_MCCR	0x10	32	0x00000000
FCM测量基准控制寄存器	FCM_RCCR	0x14	32	0x00000000
FCM中断复位控制寄存器	FCM_RIER	0x18	32	0x00000000
FCM标志寄存器	FCM_SR	0x1C	32	0x00000000
FCM标志位清除寄存器	FCM_CLR	0x20	32	0x00000000

基准地址 2: 0x4004CC00

CMU_XTAL配置寄存器	CMU_XTALCFGR	0x50	8	0x80
CMU_XTAL32控制寄存器	CMU_XTAL32CR	0x54	8	0x00
CMU_XTAL32配置寄存器	CMU_XTAL32CFGR	0x58	8	0x00
CMU_XTAL32滤波寄存器	CMU_XTAL32NFR	0x68	8	0x00
CMU_LRC控制寄存器	CMU_LRCCR	0x6C	8	0x00
CMU_LRC校准寄存器	CMU_LRCTRM	0x74	8	0x00

基准地址 3: 0x40048000

CMU_XTAL小数分频配置寄存器	CMU_XTALDIVR	0x18	32	0x03D09040
CMU_XTAL小数分频控制寄存器	CMU_XTALDIVCR	0x1C	32	0x00000000

基准地址 4: 0x40054000

寄存器名	符号	偏移地址	位宽	复位值
CMU_XTAL稳定配置寄存器	CMU_XTALSTBCR	0x0A2	8	0x05
CMU_XTAL控制寄存器	CMU_XTALCR	0x032	8	0x01
CMU_XTAL振荡故障控制寄存器	CMU_XTALSTDCCR	0x040	8	0x00
CMU_XTAL振荡故障状态寄存器	CMU_XTALSTDTSR	0x041	8	0x00
CMU_HRC控制寄存器	CMU_HRCCR	0x036	8	由ICG1.HRCSTP 值决定
CMU_HRC校准寄存器	CMU_HRCTRM	0x062	8	0x00
CMU_MRC控制寄存器	CMU_MRCCR	0x038	8	0x80
CMU_MRC校准寄存器	CMU_MRCTRM	0x061	8	0x00
CMU_PLLH配置寄存器	CMU_PLLHCFGR	0x100	32	0x11101300
CMU_PLLH控制寄存器	CMU_PLLHCR	0x02A	8	0x01
CMU_时钟源稳定状态寄存器	CMU_OSCSTBSR	0x03C	8	0x00
CMU_系统时钟源切换寄存器	CMU_CKSWR	0x026	8	0x01
CMU_时钟分频配置寄存器	CMU_SCFGR	0x020	32	0x00000000
CMU_USB时钟配置寄存器	CMU_USBCKCFGR	0x024	8	0x40
CMU_CAN时钟配置寄存器	CMU_CANCKCFGR	0x018	16	0x0ddd
CMU_AD/TRNG/DA时钟配置寄存器	CMU_PERICKSEL	0x010	16	0x0000
CMU_调试时钟配置寄存器	CMU_TPIUCKCFGR	0x03F	8	0x00
CMU_MCO1配置寄存器	CMU_MCO1CFGR	0x03D	8	0x00
CMU_MCO2配置寄存器	CMU_MCO2CFGR	0x03E	8	0x00

#### 4.11.1 CMU XTAL 配置寄存器(CMU\_XTALCFGR)

复位值：0x80

b7	b6	b5	b4	b3	b2	b1	b0
-	-	XTALDRV[1:0]		XTALMS	-	-	-

位	标记	位名	功能	读写
b7	Reserved	-	写入时写“1”	W
b6	Reserved	-	写入时写“0”	R/W
b5~b4	XTALDRV[1:0]	XTAL驱动能力选择	00: 高驱动能力(建议20~25MHz晶振) 01: 中驱动能力(建议16~20MHz晶振) 10: 小驱动能力(建议8~16MHz晶振) 11: 超小驱动能力(建议4~8MHz晶振)	R/W
b3	XTALMS	XTAL模式选择位	0: 振荡器模式 1: 外部时钟输入模式	R/W
b2~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

#### 4.11.2 CMU XTAL 配置寄存器(CMU\_XTALSTBCR)

复位值：0x05

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	XTALSTB[3:0]		

位	标记	位名	功能	读写
b7~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3~b0	XTALSTB[3:0]	XTAL稳定时间选择	0001: 稳定计数器35周期 0010: 稳定计数器67周期 0011: 稳定计数器131周期 0100: 稳定计数器259周期 0101: 稳定计数器547周期 0110: 稳定计数器1059周期 0111: 稳定计数器2147周期 1000: 稳定计数器4291周期 1001: 稳定计数器8163周期 稳定计数器的一个计数周期=LRC周期/8 CMU_XTALCR.XTALSTP位1且CMU_OSCSTBSR.XTALSTBF位为0的状态下配置此寄存器。	R/W

### 4.11.3 CMU XTAL 小数分频配置寄存器(CMU\_XTALDIVR)

复位值：0x03d09040

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16			
-	-	-		numerator[16:4]														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0			
numerator[3:0]				-	denominator[10:0]													

位	标记	位名	功能	读写
b31~b29	Reserved	-	读出时为“0”,写入时写“0”	R/W
b28~b12	numerator[16:0]	XTAL小数分频分子设定值	XTAL小数分频分子设定值, 使用方法参考【外部高速振荡器小数分频】章节	R/W
b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10~b0	denominator[10:0]	XTAL小数分频分母设定值	XTAL 小数分频分母设定值, 使用方法参考【外部高速振荡器小数分频】章节。请将此值设定为 $2^N$ 。例如 0x100 即分母为 256。0x80 即分母为 128.	R/W

### 4.11.4 CMU XTAL 小数分频控制存器(CMU\_XTALDIVCR)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
															FRADIVEN

位	标记	位名	功能	读写
b31~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	FRADIVEN	XTAL小数分频开始	0: XTAL小数分频停止 1: XTAL小数分频开始	W

#### 4.11.5 CMU XTAL 控制寄存器(CMU\_XTALCR)

复位值：0x01

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	XTALSTP

位	标记	位名	功能	读写
b7~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	XTALSTP	XTAL振荡器开启停止位	0: XTAL振荡器振荡 1: XTAL振荡器停止	R/W

##### 注意：

- XTAL 选作系统时钟或者 PLLH 时钟源时,禁止 XTALSTP 写“1”停止 XTAL 振荡器。
- 软件设定 XTAL 振荡器振荡,通过 XTALSTBF 位确认 XTAL 振荡器稳定后,才可以进入停止模式、掉电模式或者软件设定 XTAL 振荡器停止。
- 软件设定 XTAL 振荡器停止,通过 XTALSTBF 位确认 XTAL 振荡器停止后,才可以进入停止模式、掉电模式或者再次启动 XTAL 振荡器。

#### 4.11.6 CMU XTAL 振荡故障控制寄存器(CMU\_XTALSTDSCR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
XTALSTDE - - - - - XTALSTDRI XTALSTDRE XTALSTDIE							
位 标记 位名 功能 读写							
b7	XTALSTDE	XTAL振荡故障检测功能允许	0: 禁止XTAL振荡故障检测 1: 允许XTAL振荡故障检测  注： 振荡器故障检测是检测外部因素导致的振荡器异常振荡，在进入停止模式或者掉电模式前，请将振荡器振荡故障检测功能无效。	R/W			
b6~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W			
b2	XTALSTDRI	XTAL振荡故障复位中断选择	0: XTAL振荡故障产生中断 1: XTAL振荡故障产生复位  注： PLLH选择XTAL时钟作为输入源时，只能选择XTAL振荡故障产生复位功能。	R/W			
b1	XTALSTDRE	XTAL振荡故障复位允许	0: 禁止XTAL振荡故障复位 1: 允许XTAL振荡故障复位	R/W			
b0	XTALSTDIE	XTAL振荡故障中断允许	0: 禁止XTAL振荡故障中断 1: 允许XTAL振荡故障中断  通过EMB将Timer6/Timer4的PWM输出置成Hiz输出， XTALSTDIE位需要设置成1。	R/W			

##### 注意：

- XTAL 选作系统时钟或者 PLLH 时钟源时，禁止 XTALSTP 写“1”停止 XTAL 振荡器。

#### 4.11.7 CMU XTAL 振荡故障状态寄存器(CMU\_XTALSTDSR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	XTALSTDF

位	标记	位名	功能	读写
b7~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	XTALSTDF	XTAL振荡故障状态位	0: 未检测到XTAL振荡故障 1: 检测到XTAL振荡故障 置位条件： XTALSTDE=1的条件下, XTAL振荡故障 清零条件：系统时钟选择XTAL以外时钟时, 读1写0。	R/W

#### 4.11.8 CMU XTAL32 配置寄存器(CMU\_XTAL32CFGR)

复位值：0xXX

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	XTAL32DRV[2:0]

位	标记	位名	功能	读写
b7~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~b0	XTAL32DRV[2:0]	XTAL32驱动能力选择	000: 中驱动能力 001: 大驱动能力 其他: 禁止设定 注: 使用方法参考电气特性章节【晶振/陶瓷谐振器产生的低速外部时钟】	R/W

**4.11.9 CMU XTAL32 滤波寄存器(CMU\_XTAL32NFR)**

复位值：0xXX

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	XTAL32NF[1:0]

位	标记	位名	功能	读写
b7~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1~b0	XTAL32NF[1:0]	XTAL32振荡器滤波选择	00: RUN模式/停止模式/掉电模式, XTAL32的3us滤波有效 01: RUN模式XTAL32的3us滤波有效, 停止模式或掉电模式 XTAL32的3us滤波无效 10: 设定禁止 11: RUN模式/停止模式/掉电模式, XTAL32的3us滤波无效	R/W

**4.11.10 CMU XTAL32 控制寄存器(CMU\_XTAL32CR)**

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	XTAL32STP

位	标记	位名	功能	读写
b7~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	XTAL32STP	XTAL32振荡器开启停止位	0: XTAL32振荡器振荡 1: XTAL32振荡器停止	R/W

**注意：**

- XTAL32 选作系统时钟源时, 禁止 XTAL32STP 写“1”停止 XTAL32 振荡器。
- 软件设定 XTAL32 动作开始, 等待 5 个 XTAL32 周期后, 才可以再次停止 XTAL32。
- 软件设定 XTAL32 停止, 等待 5 个 XTAL32 周期后, 才可以再次启动 XTAL32。

#### 4.11.11 CMU HRC 校准寄存器(CMU\_HRCTRM)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0	
HRCTRM[7:0]								
位	标记	位名	功能					读写
			频率校准需在 HRC 频率保证范围内。					
b7~b0	HRCTRM[7:0]	HRC频率校准位	10000000: -128	10000001: -127	.....	11111111: -1	00000000: 中心Code	R/W
			00000001: +1	.....	01111110: +126	01111111: +127		

**注意：**

- 频率校准需在 HRC 频率保证范围内。

#### 4.11.12 CMU HRC 控制寄存器(CMU\_HRCCR)

复位值：由 ICG1.HRCSTP 值决定

b7	b6	b5	b4	b3	b2	b1	b0	
HRCSTP								
位	标记	位名	功能					读写
b31~b1	Reserved	-	读出时为“0”,写入时写“0”					R/W
b0	HRCSTP	HRC振荡器开启停止位	0: HRC振荡器振荡	1: HRC振荡器停止	根据ICG1.HRCSTOP配置，复位后HRC开始停止。			R/W

**注意：**

- HRC 选作系统时钟源或 PLLH 时钟源时，禁止 CMU\_HRCCR.HRCSTP 写“1”停止 HRC 时钟。
- 软件设定 HRC 振荡，通过 HRCSTBF 位确认 HRC 稳定后，才可以进入停止模式、掉电模式或者停止 HRC。
- 软件设定 HRC 停止，通过 HRCSTBF 位确认 PLLH 停止后，才可以进入停止模式、掉电模式或者再次启动 HRC。

**4.11.13 CMU MRC 校准寄存器(CMU\_MRCTRIM)**

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
MRCTRIM[7:0]							
位	标记	位名	功能				读写
			10000000: -128				
			10000001: -127				
			.....				
			11111111: -1				
b7~b0	MRCTRIM[7:0]	MRC频率校准位	00000000: 中心Code				R/W
			00000001: +1				
			.....				
			01111110: +126				
			01111111: +127				

**注意：**

- 频率校准需在 MRC 频率保证范围内。

**4.11.14 CMU MRC 控制寄存器(CMU\_MRCCR)**

复位值：0x80

b7	b6	b5	b4	b3	b2	b1	b0
MRCSTP							
位	标记	位名	功能				读写
b7	Reserved	-	读出时为“1”,写入时写“1”				R/W
b6~b1	Reserved	-	读出时为“0”,写入时写“0”				R/W
			0: MRC振荡器振荡				
			1: MRC振荡器停止				
b0	MRCSTP	MRC振荡器开启停止位	注： 1) XTAL振荡故障功能有效时，本位同时清零，MRC振荡。 2) PWC_STPMCR.CKSMRC位为1时的停止模式唤醒动作，在MRC振荡器处于振荡状态下设定。				R/W

**注意：**

- MRC 选作系统时钟源时，禁止 MRCSTP 写“1”停止 MRC 时钟。
- 软件设定 MRC 振荡，等待 5 个 MRC 周期后，才可以进入停止模式、掉电模式或者停止 MRC。
- 软件设定 MRC 停止，等待 5 个 MRC 周期后，才可以进入停止模式、掉电模式或者再次启动 MRC。
- MRC 用作 RTC 校准时钟。RTC 未初始化时，MRC 有振荡的可能性。RTC 校准功能有效时，MRCSTP 位设定无效，MRC 可能振荡。

**4.11.15 CMU LRC 校准寄存器(CMU\_LRCTRM)**

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
LRCTRM[7:0]							
位	标记	位名	功能				读写
b7~b0	LRCTRM[7:0]	LCR频率校准位	10000000: -128				R/W
			10000001: -127				
			.....				
			11111111: -1				
			00000000: 中心Code				
			00000001: +1				
			.....				
			01111110: +126				
			01111111: +127				

**注意：**

- 频率校准需在 LRC 频率保证范围内。

**4.11.16 CMU LRC 控制寄存器(CMU\_LRCCR)**

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	LRCSTP
位	标记	位名	功能				读写
b31~b1	Reserved	-	读出时为“0”，写入时写“0”				R/W
b0	LRCSTP	LRC振荡器开启停止位	0: LRC振荡器振荡				R/W
			1: LRC振荡器停止				

**注意：**

- LRC 选作系统时钟源时，禁止 LRCSTP 写“1”停止 LRC 时钟。
- 软件设定 LRC 动作开始，等待 5 个 LRC 周期后，才可以进入停止模式、掉电模式或者停止 LRC。
- 软件设定 LRC 停止，等待 5 个 LRC 周期后，才可以进入停止模式、掉电模式或者再次启动 LRC。
- 等待 XTAL 振荡器、HRC、PLLH 时钟稳定时，LRCSTP 位设定无视，LRC 强制振荡。

#### 4.11.17 CMU PLLH 配置寄存器(CMU\_PLLHCFGR)

复位值：0x11101300

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PLLHP[3:0]		PLLHQ[3:0]		PLLHR[3:0]		-		-		-		PLLHN[8]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PLLHN[7:0]				PLLS RC		-		-		PLLHM[1:0]					

位	标记	位名	功能	读写
b31-b28	PLLHP[3:0]	系统时钟用PLLH分频系数	用于PLLHP时钟的频率，在PLLH停止条件下写PLLHP。 PLLH输出时钟频率=PLLH的VCO频率/PLLHP 0000: 禁止设定 0001: 2分频 0010: 3分频 0011: 4分频 ..... 1101: 14分频 1110: 15分频 1111: 16分频	R/W
b27-b24	PLLHQ[3:0]	系统时钟用PLLH分频系数	用于PLLHQ时钟的频率，在PLLH停止条件下写PLLHQ。 PLLH输出时钟频率=PLLH的VCO频率/PLLHQ 0000: 禁止设定 0001: 2分频 0010: 3分频 0011: 4分频 ..... 1101: 14分频 1110: 15分频 1111: 16分频	R/W
b23-b20	PLLHR[3:0]	系统时钟用PLLH分频系数	用于PLLHR时钟的频率，在PLLH停止条件下写PLLHR。 PLLH输出时钟频率=PLLH的VCO频率/PLLHR 0000: 禁止设定 0001: 2分频 0010: 3分频 0011: 4分频 ..... 1101: 14分频 1110: 15分频 1111: 16分频	R/W
b19-b17	-	-	读出时为“0”,写入时写“0”	R/W
b16-b8	PLLHN[8:0]	PLLH倍频系数	用于控制PLLH的VCO的倍频系数，在PLLH停止条件下写PLLHN。 确保PLLH的VCO频率介于240MHz到480MHz之间。 PLLH的VCO频率=PLLH的VCO输入频率 * PLLHN 000010011: 20 000010100: 21 000010101: 22 000010110: 23 .....	b16-b8

			111011101: 478	
			111011110: 479	
			111011111: 480	
b7	PLL SRC	PLLH输入时钟源选择	0: 选择外部高速振荡器作为PLLH的输入时钟 1: 选择内部高速振荡器作为PLLH的输入时钟	R/W
b6-b5	-	-	读出时为“0”,写入时写“0”	R/W
			用于在PLLH的VCO之前对PLLH输入时钟进行分频。在PLLH停止条件下写PLLHM。确保PLLH的PFD输入时钟频率介于1MHz到25MHz之间。	
b4-b0	PLLHM[4:0]	PLLH输入时钟分频系数	00000: 1分频 00001: 2分频 00010: 3分频 ..... 10111: 24分频 其他禁止	R/W

**4.11.18 CMU PLLH 控制存器(CMU\_PLLHCR)**

复位值：0x01

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	PLLHOFF

位	标记	位名	功能	读写
b7~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	PLLHOFF	PLLH使能	用于开始停止PLLH。如果PLLH时钟用作系统时钟时,请不要将此位设成1。 0: PLLH动作开始 1: PLLH停止	R/W

**注意：**

- PLLH 选作系统时钟源时，禁止 PLLHOFF 写“1”停止 PLLH 时钟。
- 软件设定 PLLH 动作开始，通过 PLLHSTBF 位确认 PLLH 稳定后，才可以进入停止模式，掉电模式或者软件设定停止 PLLH。
- 软件设定 PLLH 停止，通过 PLLHSTBF 位确认 PLLH 停止后，才可以进入停止模式，掉电模式或者再次启动 PLLH。
- PLLH 选择 XTAL 振荡器作时钟源时，通过 XTALSTBF 位确认 XTAL 振荡器稳定后，才可以设定 PLLH 动作开始。PLLH 选择 HRC 作时钟源时，通过 HRCSTBF 位确认 HRC 稳定后，才可以设定 PLLH 动作开始。

**4.11.19 CMU 时钟源稳定状态器(CMU\_OSCSTBSR)**

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	PLLHSTBF	-	XTALSTBF	-	-	HRCSTBF

位	标记	位名	功能	读写
b7-b6	Reserved	-	读出时为“0”,写入时写“0”	R
b5	PLLHSTBF	PLLH稳定标志位	0: PLLH停止或者未稳定 1: PLLH稳定	R
b4	Reserved	-	读出时为“0”,写入时写“0”	R
b3	XTALSTBF	XTAL稳定标志位	0: XTAL停止或者未稳定 1: XTAL稳定	R
b2~b1	Reserved	-	读出时为“0”,写入时写“0”	R
b0	HRCSTBF	HRC稳定标志位	0: HRC停止或者未稳定 1: HRC稳定	R

#### 4.11.20 CMU 系统时钟源切换寄存器(CMU\_CKSWR)

复位值：0x01

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	CKSW[2:0]

位	标记	位名	功能	读写
b31~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
			0 0 0: 选择HRC时钟作为系统时钟	
			0 0 1: 选择MRC时钟作为系统时钟	
			0 1 0: 选择LRC时钟作为系统时钟	
			0 1 1: 选择XTAL时钟作为系统时钟	
			1 0 0: 选择XTAL32时钟作为系统时钟	
			1 0 1: 选择PLLH作为系统时钟	
b2-b0	CKSW[2:0]	系统时钟源切换	1 1 0: 禁止设定	R/W
			1 1 1: 禁止设定	
			注：	
			1, 切换的目标时钟源，需保证在时钟稳定状态。	
			2, 流程参照【时钟源切换】章节	
			3, PWC_STPMCR.CKSMRC位为1时，停止模式唤醒后，此寄存器初始化，系统时钟源选择MRC时钟。	

**4.11.21 CMU 时钟分频配置寄存器(CMU\_SCFGR)**

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	HCLKS[2:0]			-	EXCKS[2:0]			-	PCLK4S[2:0]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	PCLK3S[2:0]			-	PCLK2S[2:0]			-	PCLK1S[2:0]			-	PCLK0S[2:0]		

位	标记	位名	功能	读写
b31~b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26~24	HCLKS[2:0]	HCLK时钟分频选择位	000: 系统时钟的1分频 001: 系统时钟的2分频 010: 系统时钟的4分频 011: 系统时钟的8分频 100: 系统时钟的16分频 101: 系统时钟的32分频 110: 系统时钟的64分频 111: 禁止设定  注: PWC_STPMCR.CKSMRC位为1时, 停止模式唤醒后, 此寄存器初始化, HCLK为系统时钟的1分频。	R/W
b23	Reserved	-	读出时为“0”,写入时写“0”	R/W
b22~20	EXCKS[2:0]	ExMC时钟分频选择位	000: 系统时钟的1分频 001: 系统时钟的2分频 010: 系统时钟的4分频 011: 系统时钟的8分频 100: 系统时钟的16分频 101: 系统时钟的32分频 110: 系统时钟的64分频 111: 禁止设定  注: PWC_STPMCR.CKSMRC位为1时, 停止模式唤醒后, 此寄存器初始化, EXCLK为系统时钟的1分频。	R/W
b19	Reserved	-	读出时为“0”,写入时写“0”	R/W
b18~16	PCLK4S[2:0]	PCLK4时钟分频选择位	000: 系统时钟的1分频 001: 系统时钟的2分频 010: 系统时钟的4分频 011: 系统时钟的8分频 100: 系统时钟的16分频 101: 系统时钟的32分频 110: 系统时钟的64分频 111: 禁止设定  注: PWC_STPMCR.CKSMRC位为1时, 停止模式唤醒后, 此寄存器初始化, PCLK4为系统时钟的1分频。	R/W
b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14~12	PCLK3S[2:0]	PCLK3时钟分频选择位	000: 系统时钟的1分频 001: 系统时钟的2分频 010: 系统时钟的4分频	R/W

011: 系统时钟的8分频

100: 系统时钟的16分频

101: 系统时钟的32分频

110: 系统时钟的64分频

111: 禁止设定

注: PWC\_STPMCR.CKSMRC位为1时, 停止模式唤醒后, 此

寄存器初始化, PCLK3为系统时钟的1分频。

b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10~8	PCLK2S[2:0]	PCLK2时钟分频选择位	000: 系统时钟的1分频 001: 系统时钟的2分频 010: 系统时钟的4分频 011: 系统时钟的8分频 100: 系统时钟的16分频 101: 系统时钟的32分频 110: 系统时钟的64分频 111: 禁止设定 注: PWC_STPMCR.CKSMRC位为1时, 停止模式唤醒后, 此 寄存器初始化, PCLK2为系统时钟的1分频。	R/W
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6~4	PCLK1S[2:0]	PCLK1时钟分频选择位	000: 系统时钟的1分频 001: 系统时钟的2分频 010: 系统时钟的4分频 011: 系统时钟的8分频 100: 系统时钟的16分频 101: 系统时钟的32分频 110: 系统时钟的64分频 111: 禁止设定 注: PWC_STPMCR.CKSMRC位为1时, 停止模式唤醒后, 此 寄存器初始化, PCLK1为系统时钟的1分频。	R/W
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~0	PCLK0S[2:0]	PCLK0时钟分频选择位	000: 系统时钟的1分频 001: 系统时钟的2分频 010: 系统时钟的4分频 011: 系统时钟的8分频 100: 系统时钟的16分频 101: 系统时钟的32分频 110: 系统时钟的64分频 111: 禁止设定 注: PWC_STPMCR.CKSMRC位为1时, 停止模式唤醒后, 此 寄存器初始化, PCLK0为系统时钟的1分频。	R/W

#### 4.11.22 CMU USB 时钟配置寄存器(CMU\_USBCKCFGR)

复位值：0x40

b7	b6	b5	b4	b3	b2	b1	b0
			USBCKS[3:0]	-	-	-	-

位	标记	位名	功能	读写
b7~b4	USBCKS[3:0]	USB-FS的48MHz时钟源选择	0001: 系统时钟2分频 0010: 系统时钟3分频 0011: 系统时钟4分频 0100: 系统时钟5分频 0101: 系统时钟6分频 0110: 系统时钟7分频 0111: 系统时钟8分频 1000: PLLH/Q 1001: PLLH/R 其他禁止设定 注： 1. 切换的目标时钟源为PLLH时，需保证在PLLH时钟处于稳定状态。 2. PWC_STPMCR.CKSMRC位为1时，停止模式唤醒后，此寄存器初始化，USBCLK为系统时钟的5分频。	R/W
b3~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

**4.11.23 CMU CAN 时钟配置存器(CMU\_CANCKCFGR)**

复位值：0x0ddd

b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	CAN3CKS[3:0]			
b7	b6	b5	b4	b3	b2	b1	b0
CAN2CKS[3:0]				CAN1CKS[3:0]			

位	标记	位名	功能	读写
b15~b12	-	-		R/W
b11~b8	CAN3CKS[3:0]	CAN3通讯时钟选择	0001: 系统时钟2分频 0010: 系统时钟3分频 0011: 系统时钟4分频 0100: 系统时钟5分频 0101: 系统时钟6分频 0110: 系统时钟7分频 0111: 系统时钟8分频 1000: PLLH/Q 1001: PLLH/R 1101: XTAL 其他禁止设定	R/W
			注： 1. 切换的目标时钟源为PLLH时，需保证在PLLH时钟处于稳定状态。 2. 系统时钟选择PLLH时，需要将USB, CAN, QSPI, SPI, 通用定时器, FCM, ADC, DAC设定成模块停止状态，再写CMU_SCFGR寄存器切换时钟分频。写完CMU_CANCKCFGR寄存器后，软件等待系统稳定时间至少30μs。 3. PWC_STPMCR.CKSMRC位为1时，停止模式唤醒后，此寄存器初始化，CANCLK为XTAL时钟源。	
b7~b4	CAN2CKS[3:0]	CAN2通讯时钟选择	0001: 系统时钟2分频 0010: 系统时钟3分频 0011: 系统时钟4分频 0100: 系统时钟5分频 0101: 系统时钟6分频 0110: 系统时钟7分频 0111: 系统时钟8分频 1000: PLLH/Q 1001: PLLH/R 1101: XTAL 其他禁止设定	R/W
			注： 1. 切换的目标时钟源为PLLH时，需保证在PLLH时钟处于稳定状态。 2. 系统时钟选择PLLH时，需要将USB, CAN, QSPI, SPI, 通用定时器, FCM, ADC, DAC设定成模块停止状态，再写	

CMU\_SCFG寄存器切换时钟分频。写完CMU\_CANCKCFGR寄存器后，软件等待系统稳定时间至少30μs。

3. PWC\_STPMCR.CKSMRC位为1时，停止模式唤醒后，此寄存器初始化，CANCLK为XTAL时钟源。

0001: 系统时钟2分频

0010: 系统时钟3分频

0011: 系统时钟4分频

0100: 系统时钟5分频

0101: 系统时钟6分频

0110: 系统时钟7分频

0111: 系统时钟8分频

1000: PLLH/Q

1001: PLLH/R

1101: XTAL

其他禁止设定

R/W

b3~b0      CAN1CKS[3:0]    CAN通讯时钟选择

注：

1. 切换的目标时钟源为PLLH时，需保证在PLLH时钟处于稳定状态。
2. 系统时钟选择PLLH时，需要将USB, CAN, QSPI, SPI, 通用定时器, FCM, ADC, DAC设定成模块停止状态，再写CMU\_SCFG寄存器切换时钟分频。写完CMU\_CANCKCFGR寄存器后，软件等待系统稳定时间30μs。
3. PWC\_STPMCR.CKSMRC位为1时，停止模式唤醒后，此寄存器初始化，CANCLK为XTAL时钟源。

**4.11.24 CMU AD/TRNG/DA 时钟配置存器(CMU\_PERICKSEL)**

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	PERICKSEL[3:0]
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	

位	标记	位名	功能	读写
b15~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3~b0	PERICKSEL[3:0]	AD/TRNG时钟源选择	0000: CMU_SCFGGR设定的PCLK2/PCLK4 1000: PCLK2/PCLK4配置为PLLHQ 1001: PCLK2/PCLK4配置为PLLHR 除此以外禁止设定。	R/W

**注意：**

- 切换的目标时钟源为 PLLH 时，需保证在 PLLH 时钟稳定状态。

**4.11.25 CMU 调试时钟配置存器(CMU\_TPIUCKCFGR)**

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0	TPIUCKS[1:0]
TPIUCKOE	-	-	-	-	-	-	-	-

位	标记	位名	功能	读写
b7	TPIUCKOE	TPIU时钟供给允许位	0: 禁止 1: 允许	R/W
b6~b2	-	-	读出时为“0”,写入时写“0”	R/W
b1~0	TPIUCKS[1:0]	TPIU时钟分频选择位	00: 1分频 01: 2分频 10: 4分频 其他禁止设定	R/W

#### 4.11.26 CMU MCO1 配置存储器(CMU\_MCO1CFGR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
MCO1EN	MCO1DIV[2:0]			MCO1SEL[3:0]			
<hr/>							
位	标记	位名	功能		读写		
b7	MCO1EN	MCO_1输出许可	0: 禁止MCO_1输出 1: 允许MCO_1输出		R/W		
<hr/>							
b6~b4	MCO1DIV[2:0]	MCO_1分频选择	000: 1分频 001: 2分频 010: 4分频 011: 8分频 100: 16分频 101: 32分频 110: 64分频 111: 128分频		R/W		
b3~b0	MCO1SEL[3:0]	MCO_1时钟源选择	0000: HRC时钟 0001: MRC时钟 0010: LRC时钟 0011: XTAL时钟 0100: XTAL32时钟 0110: PLLHP 1000: PLLHQ 1011: 系统时钟 其他禁止设定。	R/W			

#### 4.11.27 CMU MCO2 配置存储器(CMU\_MCO2CFGR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0							
MCO2EN	MCO2DIV[2:0]			MCO2SEL[3:0]										
<hr/>														
位	标记	位名	功能			读写								
b7	MCO2EN	MCO_2输出许可			0: 禁止MCO_2输出 1: 允许MCO_2输出	R/W								
<hr/>														
b6~b4	MCO2DIV[2:0]		MCO_2分频选择			R/W								
<hr/>														
000: 1分频 001: 2分频 010: 4分频 011: 8分频 100: 16分频 101: 32分频 110: 64分频 111: 128分频														
<hr/>														
b3~b0	MCO2SEL[3:0]		MCO_2时钟源选择			R/W								
<hr/>														
0000: HRC时钟 0001: MRC时钟 0010: LRC时钟 0011: XTAL时钟 0100: XTAL32时钟 0110: PLLHP 1000: PLLHQ 1011: 系统时钟 其他禁止设定。														
<hr/>														

#### 4.11.28 FCM 下限比较值寄存器(FCM\_LVR)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LVR[15:0]															

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b0	LVR[15:0]	下限比较值	START位为0时配置此寄存器。	R/W

#### 4.11.29 FCM 上限比较值寄存器(FCM\_UVR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UVR[15:0]															

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b0	UVR[15:0]	上限比较值	START位为0时配置此寄存器。	R/W

#### 4.11.30 FCM 计数器值寄存器(FCM\_CNTR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNTR[15:0]															

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b0	CNTR[15:0]	计数器值	检测到基准时钟的EDGES位选择的有效边沿时, 将计数器值保存到此寄存器 (START=1后第一个有效边沿除外)	R

#### 4.11.31 FCM 开始停止寄存器(FCM\_STR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STAR T

位	标记	位名	功能	读写
b31~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	START	频率测量开始位	0: 频率测量停止 1: 频率测量开始	R/W

#### 4.11.32 FCM 测量对象控制寄存器(FCM\_MCCR)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	MCKS[3:0]			-	-	MDIVS[1:0]		

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
			0000: XTAL 0001: XTAL32 0010: HRC 0011: LRC	
b7~b4	MCKS[3:0]	测量对象时钟选择位	0100: SWDTLRC 0101: PCLK1 0111: MRC 1000: PLLHP 其他: 设定禁止	R/W
b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
			00: 不分频 01: 4分频 10: 8分频 11: 32分频	
b1~b0	MDIVS[1:0]	测量对象分频选择		

#### 4.11.33 FCM 测量基准控制寄存器(FCM\_RCCR)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EXREFE	-	EDGES[1:0]	-	-	DNFS[1:0]	INEXS	RCKS[3:0]	-	RDIVS[1:0]						

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15	EXREFE	外部管脚输入参考时钟FCMREF允许位	0: 禁止外部管脚输入参考时钟FCMREF 1: 允许外部管脚输入参考时钟FCMREF	R/W
b14	Reserved	-	读出时为“0”,写入时写“0”	R/W
b13~b12	EDGES[1:0]	测量基准边沿选择位	00: 上升沿 01: 下降沿 10: 上升和下降沿 11: 禁止设定	R/W
b11~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9~b8	DNFS[1:0]	数字滤波器功能选择位	00: 无滤波功能 01: MCKS位选择的时钟作为滤波时钟 10: MCKS位选择的时钟的4分频作为滤波时钟 11: MCKS位选择的时钟的16分频作为滤波时钟	R/W
b7	INEXS	测量基准, 内部时钟和端子选择位	0: 外部管脚输入参考时钟FCMREF 1: RCKS选择位选择的时钟	R/W
b6~b3	RCKS[3:0]	测量基准时钟选择位	0000: XTAL 0001: XTAL32 0010: HRC 0011: LRC 0100: SWDTLRC 0101: PCLK1 0111: MRC 1000: PLLHP 其他: 设定禁止	R/W
b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1~b0	RDIVS[1:0]	测量基准分频选择	00: 32分频 01: 128分频 10: 1024分频 11: 8192分频	

#### 4.11.34 FCM 中断复位控制寄存器(FCM\_RIER)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	ERRE	-	-	ERRINTRS	-	OVFIE	MENDIE	ERRIE

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	ERRE	频率异常复位允许位	0: 禁止 1: 允许	
b6~b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4	ERRINTRS	频率异常中断复位选择位	0: 频率异常发生中断 1: 频率异常发生复位	
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	OVFIE	计数器溢出中断允许位	0: 禁止计数器溢出中断 1: 允许计数器溢出中断	R/W
b1	MENDIE	测量结束中断允许位	0: 禁止测量结束发生中断 1: 允许测量结束发生中断	R/W
b0	ERRIE	频率异常中断允许位	0: 禁止频率异常发生中断 1: 允许频率异常发生中断	R/W

#### 4.11.35 FCM 标志寄存器(FCM\_SR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	OVF	MENDF	ERRF

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	OVF	计数器溢出标志位	0: 计数器未溢出 1: 计数器溢出	R
b1	MENDF	测量结束标志位	0: 测量中 1: 测量结束	R
b0	ERRF	频率异常标志位	0: 无频率异常发生 1: 发生频率异常	R

#### 4.11.36 FCM 标志位清除寄存器(FCM\_CLR)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

					-	-	-	-	-	-	-	-	OVF CLR	MEN DFCLR	ERRF CLR
--	--	--	--	--	---	---	---	---	---	---	---	---	---------	-----------	----------

位	标记	位名	功能	读写
b31~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	OVFCLR	计数器溢出标志清零位	写“1”计数器溢出标志位清零	W
b1	MENDFCLR	测量结束标志清零位	写“1”测量结束标志位清零	W
b0	ERRFCLR	频率异常标志清零位	写“1”测量结束标志位清零	W

## 5 电源控制 (PWC)

### 5.1 简介

电源控制器用来控制芯片的多个电源域在多个运行模式和低功耗模式下的电源供给、切换、检测。电源控制器由功耗控制逻辑(PWCL)、电源电压检测单元(PVD)、构成。

芯片的工作电压(VCC)为 1.8V 到 3.6V。电压调节器(LDO)为 VDD 域和 VDDR 域供电，VDDR 电压调压器(RLDO)在掉电模式为 VDDR 域供电。芯片通过功耗控制逻辑(PWC)提供了高速、超低速等两种运行模式，睡眠、停止和掉电等三种低功耗模式。

电源电压检测单元(PVD)提供了上电复位(POR)、掉电复位(PDR)、欠压复位(BOR)、可编程电压检测 1(PVD1)、可编程电压检测 2(PVD2)、基准电压测量通路等功能，其中 POR、PDR、BOR 通过检测 VCC 电压，控制芯片复位动作。 PVD1 通过检测 VCC 电压，根据寄存器设定使芯片产生复位或者中断。 PVD2 通过检测 VCC 电压或者外部输入检测电压，根据寄存器选择产生复位或者中断。基准电压测量通路，是使用 ADC 测量基准电压的功能。AVCC 电压测量功能，是指使用 ADC 测量 AVCC 的 1/2 分压，从而获得 AVCC 电压值的功能。

VDDR 区域在芯片进入掉电模式通过 RLDO 维持电源，保证实时时钟模块(RTC)、唤醒定时器(WKTM)能够继续动作、保持 4KB 的低功耗保持 SRAM(Ret\_SRAM)的数据。模拟模块配备了专用供电引脚，提高了模拟性能。

## 5.2 电源分布

图 5-1 是芯片的电源分布图。芯片由 VCC 域、VDD 电源域、VDDR 电源域、AVCC 电源域、VCCIO 域构成。

VCC 域通过 VCC/VSS 引脚供电，由低功耗控制逻辑(PWCL)、电源电压检测单元(PVD)、IO 电平保持电路、电压调节器(LDO)、VDDR 域调压器(RLDO)、外部高速振荡器(XTAL)、内部低速振荡器(LRC)、外部低速振荡器 (XTAL32) 等电路构成。

VCCIO 域通过 VCCIO 引脚供电，为 GPIO\_LP 电路供电。

VDD 域由 CPU、数字外设等数字逻辑、RAM、FLASH 等构成，通过 LDO 产生的 VDD 供电。在 VDD 域中的 RAM 被分为 2 个独立模块构成，可以通过寄存器独立控制每个模块的断电。

VDDR 域由 4KB 的 Ret\_SRAM、实时时钟(RTC)、唤醒定时器(WKTM)构成。在掉电模式下通过 RLDO 供电，掉电模式以外的模式中通过 LDO 供电。在掉电模式下，Ret\_SRAM 能够保持数据、实时时钟 RTC 和唤醒定时器 WKTM 能够继续动作。不需要使用 VDDR 域的功能时，可以通过置位 PWR\_PWRC0.VVDRSD，将 VDDR 域在掉电模式中切断电源，进一步降低功耗。

模拟电源域主要由模数转换器(ADC)、数模转换器 (DAC)、比较器(CMP)、可编程增益放大器(PGA)、模拟系的输入输出管脚构成，由 AVCC/AVSS 引脚供电。为了提供高精度的模拟性能，模拟区域配有独立电源。为了确保 ADC、DAC 具有更高的精度，ADC、DAC 的参考电压使用专用管脚 (VREFH, VREFL)。

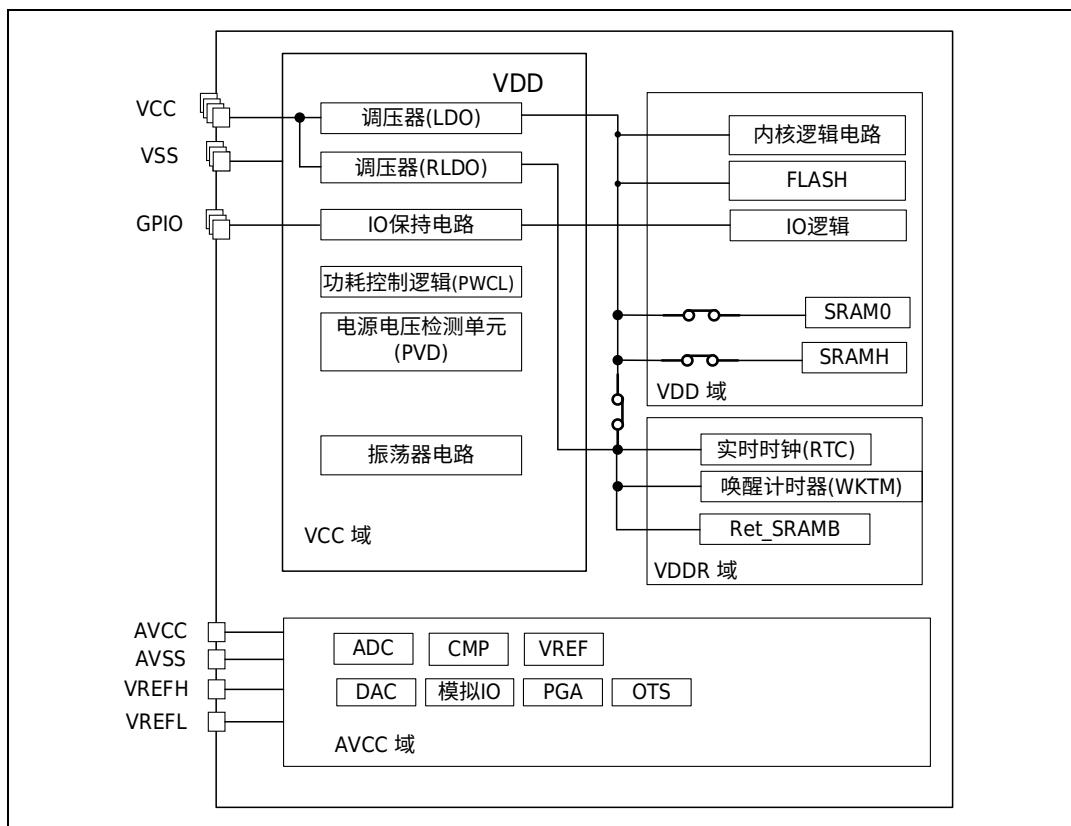


图 5-1 电源构成图

## 5.3 电源电压检测单元(PVD)说明

电源电压检测单元(PVD)包括上电复位(POR)、掉电复位(PDR)、欠压复位(BOR)、可编程电压检测1(PVD1)、可编程电压检测2(PVD2)，基准电压测量通路、AVCC电压测量等功能。

### 5.3.1 上电复位/掉电复位动作说明

芯片集成有上电复位、掉电复位电路。上电复位、掉电复位波形如图 5-2 所示，当 VCC 高于指定阈值  $V_{POR}$  之后，经过  $T_{RSTPOR}$  时间之后，芯片解除上电复位状态，CPU 开始执行代码。当 VCC 低于  $V_{PDR}$  时，芯片保持复位状态。使用上电复位时，复位管脚 NRST 必须为 1。如果复位管脚被下拉，芯片将通过管脚复位的方式复位启动。

$V_{POR}$ 、 $V_{PDR}$ 、 $T_{RSTPOR}$  等参数的详细信息，请参考**数据手册中电气特性**。

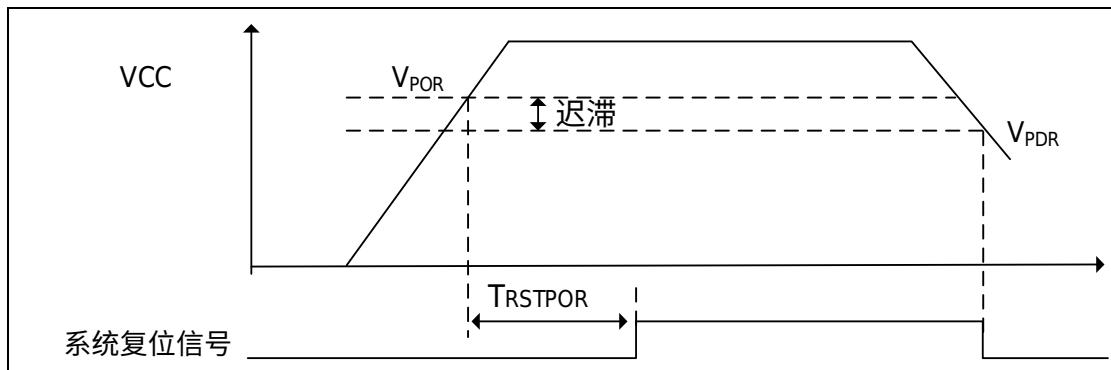


图 5-2 上电复位、掉电复位波形

### 5.3.2 欠压复位(BOR)说明

在上电期间，直到 VCC 高于 VBOR 为止，欠压复位(BOR)将使芯片处于复位状态。

VBOR 阈值通过初始化配置位(ICG)的 BOR\_lev、BORDIS 进行配置。BORDIS=0 时，BOR 检测电压可以从 4 个阈值中选择。BORDIS 被配置成 1 时，芯片通过上电复位、掉电复位进行复位控制。

表 5-1 BOR 配置

BORDIS	BOR_lev	说明
1	XX	BOR无效
0	00	BOR有效、选择BOR阈值0(VBOR0)
0	01	BOR有效、选择BOR阈值1(VBOR1)
0	10	BOR有效、选择BOR阈值2(VBOR2)
0	11	BOR有效、选择BOR阈值3(VBOR3)

BOR 阈值的电气特性，请参考电气特性章节。

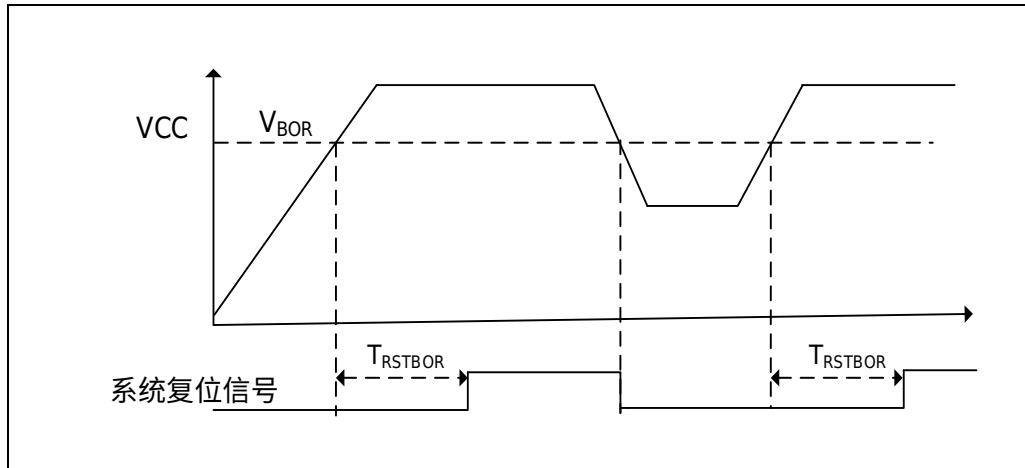


图 5-3 欠压复位波形

### 5.3.3 可编程电压检测 1(PVD1)、可编程电压检测 2(PVD2)

可编程电压检测 1 和可编程电压检测 2 通过检测 VCC 供电电压是否经过检测阈值，触发相应的复位或者中断动作。各检测电路分别可编程配置。

电源电压经过各个检测电路的阈值电压点时，可将该事件编程配置成复位/中断(可屏蔽/不可屏蔽)/AOS 触发功能。

可编程电压检测主要特性如表 5-2 所示。

表 5-2 PVD1/ PVD2 特性

项目	PVD1	PVD2
检测对象	VCC下降/上升过程中是否经过阈值电压点(VPVD1)	VCC下降/上升过程中是否经过阈值电压点(VPVD2)、外部输入电压的上升下降是否经过阈值电压点(VPVD2,PWR_PVDLCR.PVD2LVL[2:0]=111)
检测电压点	由PVD1LVL[2:0]配置	由PVD2LVL[2:0]配置
复位	复位：VCC<VPVD1； 复位解除：VCC> VPVD1经过一定复位处理时间。	复位：VCC<VPVD2； 复位解除：VCC> VPVD2经过一定复位处理时间。
中断	配置成电压检测1中断或者非可屏蔽中断 VCC下降经过阈值电压点(VPVD1) 或者 VCC上升经过阈值电压点(VPVD1) 或者VCC上升/下降经过阈值电压点(VPVD1)	配置成电压检测2中断或者非可屏蔽中断 VCC下降经过阈值电压点(VPVD2) 或者 VCC上升经过阈值电压点(VPVD2) 或者VCC上升/下降经过阈值电压点(VPVD2)
滤波功能	数字滤波	数字滤波
AOS触发功能	VCC下降经过阈值电压点(VPVD1)	VCC下降经过阈值电压点(VPVD2)

### 5.3.4 PVD1、PWD2 中断/复位框图

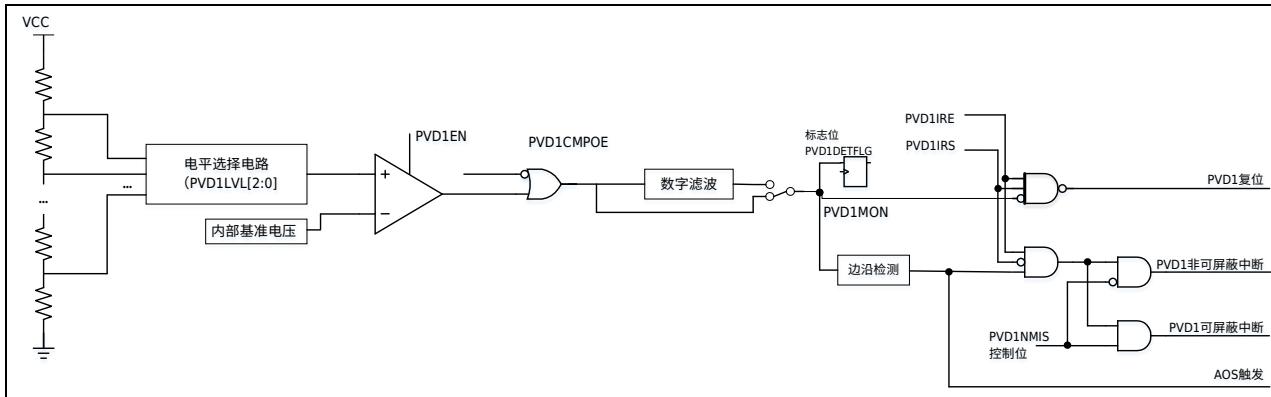


图 5-4 PVD1 中断/复位框图

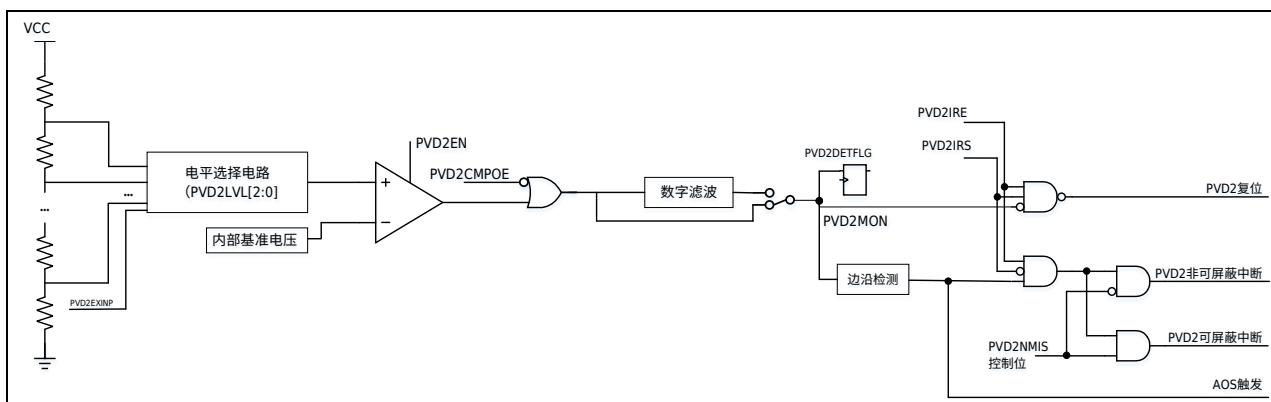


图 5-5 PVD2 中断/复位框图

### 5.3.5 输入/输出引脚

引脚名	输入/输出	功能
PVD2EXINP	输入	外部输入 PVD2 比较电压

### 5.3.6 PVD1 中断和复位

在停止模式或者掉电模式中使用 PVD1 电路时, 请遵守如下注意事项。

- 停止模式
  - 必须将数字滤波器无效。
- 掉电模式
  - 必须将数字滤波器无效。
  - PVD1IRS 置 0, 选择 PVD1 产生中断; 选择复位功能时, 不能进入掉电模式。

下图是电压检测 1 中断的运行时序图, PVD1DETFLG 需要清零后才能再次发生中断。

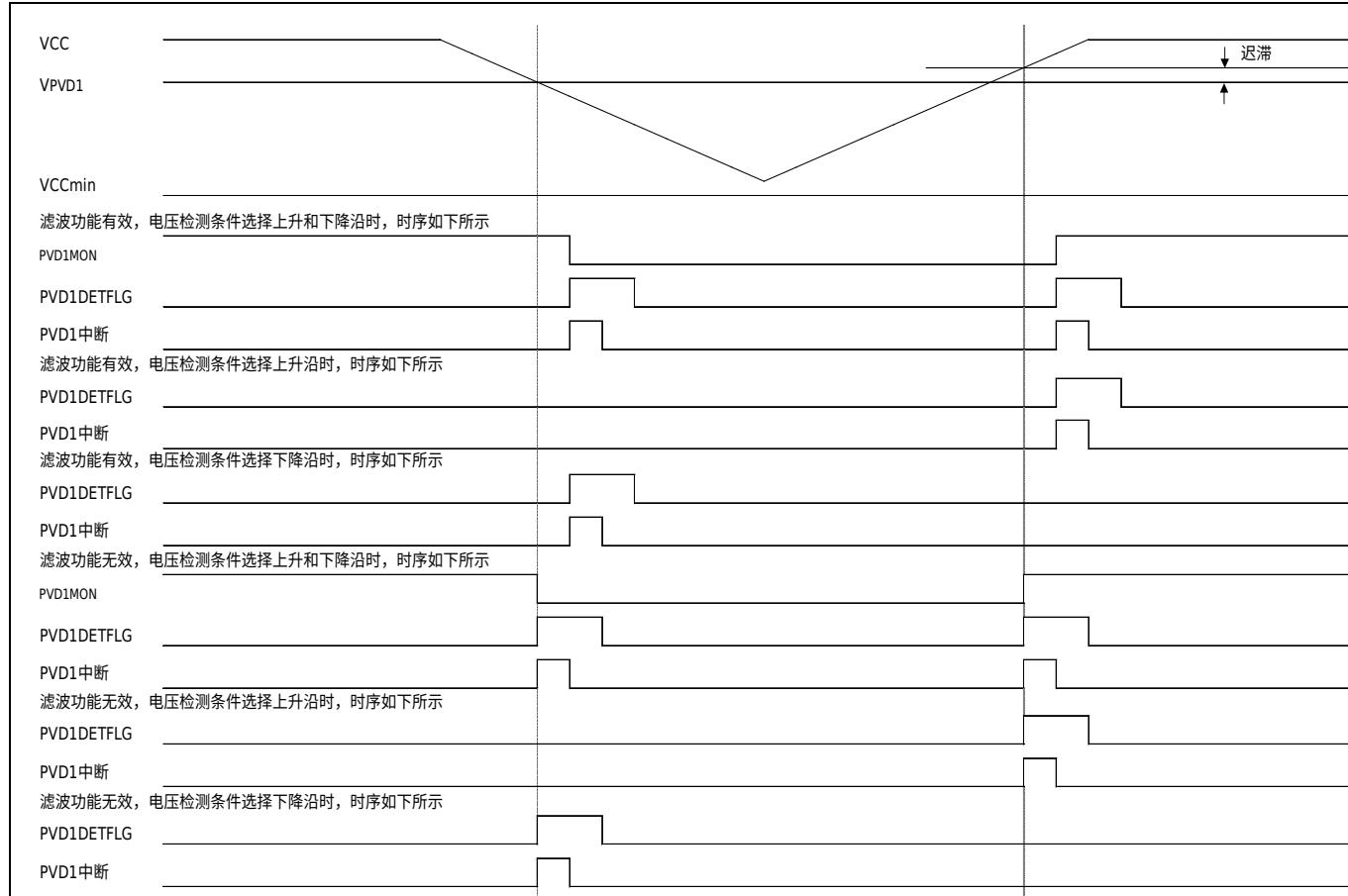


图 5-6 电源监视 1 中断时序图

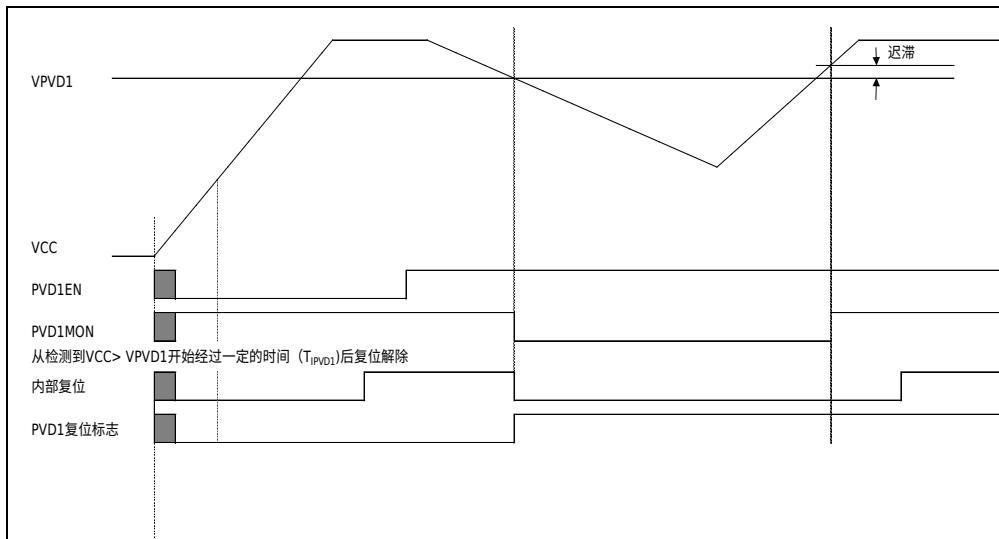


图 5-7 电源监视 1 复位时序图

### 5.3.7 PVD2 中断和复位

在停止模式或者掉电模式中使用 PVD2 电路时, 请遵守如下注意事项:

- 停止模式
  - 必须将数字滤波器无效。
- 掉电模式
  - 必须将数字滤波器无效。
  - PVD2INTRS 置 0, 选择 PVD2 产生中断; 选择复位功能时, 不能进入掉电模式。

下图是电压检测 2 中断的运行时序图, PVD2DETFLG 需要清零后才能再次发生中断。

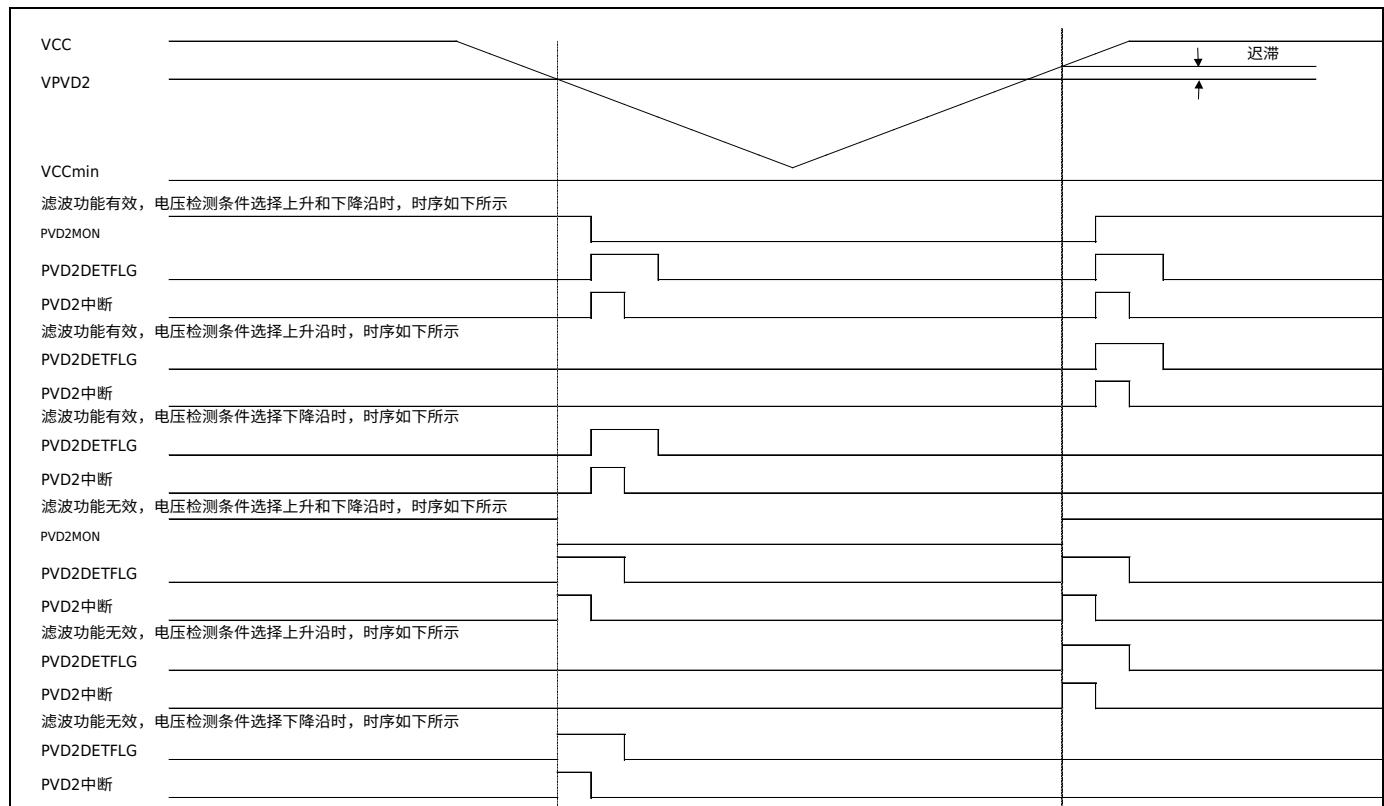


图 5-8 电源监视 2 中断运行时序图

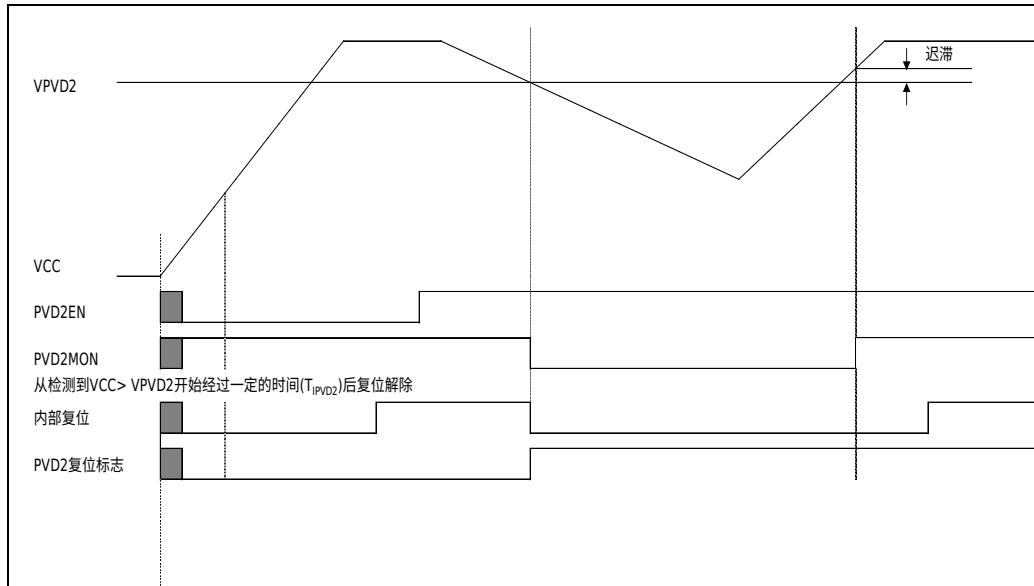


图 5-9 电源监视 2 复位运行时序图

### 5.3.8 内部电压采样和检测功能

芯片内部电压采样和检测功能是指基准电压测量通路，是使用 ADC 测量基准电压的功能。内部基准电压 ADVREF 约为 1.10V。

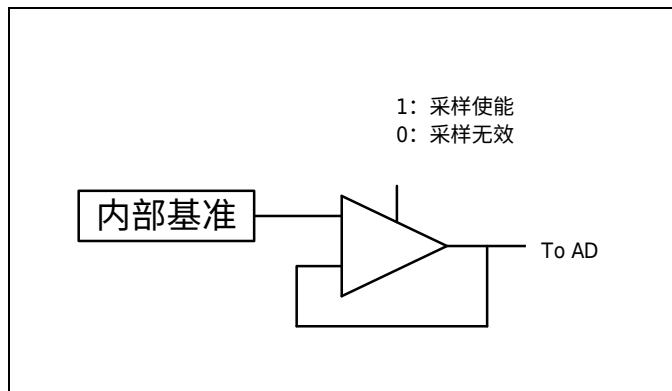


图 5-10 内部电压采样示意图

#### ■ 基准电压 ADVREF 测量通路

使用基准电压测量通路时，需要按照下述步骤选中基准电压测量通路。

1. PWC\_PWRC4.ADBUFE=1，使能内部电压测量功能
2. 设置 ADC 通道选择寄存器，选择基准电压 ADVREF 作为转换通道
3. 等待 50uS 后使用 ADC 测量内部基准电压

## 5.4 唤醒定时器

芯片内置了唤醒定时器 WKT，该计数器可以选择内部低速振荡器(LRC)、外部低速振荡器作为时钟源，在 RTC 使用外部低速振荡器作为时钟动作时还可以选择 64Hz 的内部时钟信号作为时钟源。该计数器是加计数的计数器，在 PWC\_WKTCR.WKTCE 置位后，计数器开始计数，当计数值等于 PWC\_WKTCR.WKTCMP[11:0]设置值后，计数器计数清零后重新计数的同时，产生匹配一致事件，该事件可以用作普通中断，可以用于唤醒停止模式，用于唤醒掉电模式。

WKT 寄存器值上电复位后不定，上电复位后，设置 PWC\_WKTCR。在启动 WKT 前，要确保计数时钟处于开启状态。

## 5.5 动作模式与低功耗模式

系统复位或者上电复位之后，芯片的所有电源域都处于供电状态，芯片进入运行模式。在运行模式下，CPU 通过 HCLK 提供时钟，并执行程序代码。

为了节省 CPU 不需要运行时的功耗，系统提供了睡眠模式、停止模式、掉电模式等三种低功耗模式。芯片可以配置的低功耗模式如表 5-4 所示。在睡眠模式中，芯片的 Cortex-M4 内核停止动作，外设保持运行状态；在停止模式中，芯片的外设和 CPU 都停止动作；在掉电模式中，VDD 域的电源被关闭，VDD 域的外设都停止动作。位于 VDDR 域中的实时时钟、唤醒定时器在低功耗模式中可以动作，Ret\_SRAM 可以保持数据；在不需要使用 VDDR 域中的实时时钟、唤醒定时器和 Ret\_SRAM 时，可以设置关闭 VDDR 域的调压器 RLDO，在进入掉电模式后可以进一步降低功耗。

用户可以根据应用选择运行模式和低功耗模式，以在低功耗、短启动时间、可唤醒源、和系统执行效率之间寻求最佳平衡。

低功耗模式的运行条件及各模块在低功耗模式下的状态如表 5-5 所示。

表 5-3 运行模式

运行模式	说明
运行模式 (Run Mode)	主频120MHz以下

表 5-4 低功耗模式

模式	说明
睡眠模式(Sleep Mode)	CPU时钟停止，外设保持运行
停止模式(Stop Mode)	芯片外设和CPU的时钟都停止
掉电模式 (Power Down Mode)	掉电模式1(PDMD1)
	掉电模式2(PDMD2)
	掉电模式3(PDMD3)

模式	说明
	PWC_PDWKF0/PWC_PDWKF1/RMU_RSTF0外芯片彻底复位。
掉电模式4(PDMD4)	VDD域掉电之外，VDDR域掉电，上电复位电路进入低功耗模式，电压检测单元(PVD)无效

表 5-5 低功耗模式的运行条件及各模块在低功耗模式下的状态

项目	睡眠模式	停止模式	掉电模式
进入	PWC_STPMCR.STOP=0 PWC_PWRC0.PWDN=0, WFI	PWC_STPMCR.STOP=1 PWC_PWRC0.PWDN=0, WFI	PWC_STPMCR.STOP=1 PWC_PWRC0.PWDN=1, WFI
解除	任意中断或者复位	停止模式下可以使用的中断或复位	掉电模式下可以使用的唤醒事件或者复位
外部高速振荡器	工作可设	停止	停止
外部低速振荡器	工作可设	工作可设	工作可设
内部高速振荡器	工作可设	停止	掉电
内部中速振荡器	工作可设	停止	掉电
内部低速振荡器	工作可设	工作可设	工作可设
WDT专用时钟振荡器	工作可设	工作可设	掉电
PLLA	工作可设	停止	掉电
PLLH	工作可设	停止	掉电
CPU	停止(保持)	停止(保持)	掉电
RAM	工作可设 可以设置工作，掉电	停止(保持) 根据进入待机前的设定，可以维持掉电或者睡眠	掉电
Flash	工作可设	停止(保持)	掉电，内容保持
DMA	工作可设	停止(保持)	掉电
调压器	工作 驱动可以调整	工作 驱动可以调整	停止
上电复位电路	工作	工作	工作 掉电模式1、掉电模式2复位电路精度可以保证、掉电模式3和掉电模式4下上电复位电压不保证
欠压复位BOR	工作可设	工作可设	掉电模式1工作可设 掉电模式2/3/4中停止
电压检测模块PVD	工作可设	工作可设	掉电模式1工作可设 掉电模式2/3/4中停止
WDT	工作可设	停止(保持)	掉电
SWDT	工作可设	工作可设	掉电
RTC	工作可设	工作可设	掉电模式1/2工作可设

项目	睡眠模式	停止模式	掉电模式
			掉电模式3/4下掉电
USB-FS	工作可设	停止(保持)	掉电
Timer0	工作可设	工作可设	掉电
Timer2	工作可设	工作可设	掉电
Ret_SRAM	工作可设 可以设置工作，掉电，睡眠	停止 (数据保持) 可以设置掉电，睡眠	掉电模式1/2停止(数据保持), 可以设置掉电，睡眠 掉电模式3/4下掉电
WKTM	工作可设	工作可设	掉电模式1/2工作可设 掉电模式3/4下掉电
其他周边外设模块	工作可设	停止(保持)	掉电
AD	工作可设	停止	掉电
DA	工作可设	工作可设	掉电
PGA	工作可设	工作可设	掉电
CMP	工作可设	工作可设	掉电
GPIO	工作可设	保持	保持或者高阻
PC14-PC15	工作可设	作为外部低速振荡器的管脚使用时，保持振荡器动作；设为GPIO或者其他外设功能时，请设定保持两个管脚为同样的电平	设为GPIO或者其他外设功能时，PC14和PC15的状态可以设置为保持或者高阻，请设定保持两个管脚为同样的电平
PF0-PF1	工作可设	作为外部高速振荡器使用时，振荡器停止震荡，管脚状态保持进入STOP模式前的状态；设为GPIO或者其他外设功能时，保持STOP之前的状态	作为外部高速振荡器使用时，振荡器停止震荡，管脚状态保持进入掉电模式前的状态；设为GPIO或者其他外设功能时，保持掉电模式之前的状态
NRST复位管脚	芯片外面通过电路上拉到VCC	芯片外面通过电阻上拉到VCC	芯片外面通过电阻上拉到VCC
PA11-PA12	工作可设	保持； 由于该管脚的电平被拉高时会产生电流，不使用USB的STOP模式唤醒功能时，进入STOP模式时禁止上拉。,	保持或者高阻； 由于该管脚的电平被拉高时会产生冗余电流，进入掉电模式时禁止上拉
MD	工作可设	保持；	保持； 芯片外面通过电阻接到地
JTAG	工作可设； 作为JTAG功能时，内置上拉电路有效	保持； 作为JTAG功能时，内置上拉电路有效	保持； 作为JTAG功能时，内置上拉电路有效

### 5.5.1 睡眠模式

在睡眠模式中，CPU 停止运行，它的内部寄存器保持进入睡眠模式之前的状态。看门狗和专用看门狗以外的外设和其他系统模块的动作状态不会改变。

通过 ICG 设置为自动启动时，如果 ICG 的 WDTSLPOFF 位为 1，看门狗在睡眠模式下停止计数；如果 WDTSLPOFF 位为 0，看门狗在睡眠模式下继续计数。如果 ICG 未设置为自动启动，通过软件启动的方式启动看门狗，则如果 WDT\_CR.SLPOFF 位为 1，看门狗在睡眠模式下停止计数；如果 WDT\_CR.SLPOFF 位为 0，看门狗在睡眠模式下不停止计数。

通过 ICG 设置为自动启动时，如果 ICG 的 SWDTSLOFF 位为 1，专用看门狗在睡眠模式下停止计数；如果 SWDTSLOFF 位为 0，专用看门狗在睡眠模式下继续计数。

- 进入睡眠模式

在 PWC\_STPMCR.STOP=0 时执行 WFI 指令即可进入睡眠模式。

- 退出睡眠模式

任意中断、复位都可以将芯片从睡眠模式唤醒。通过中断唤醒时，芯片进入中断处理程序；通过复位退出睡眠模式时，芯片进入复位状态。

### 5.5.2 停止模式

在停止模式中，CPU、大部分外设和时钟源都停止动作。芯片保持 CPU 内部寄存器和 SRAM 数据，外设状态和管脚状态。在停止模式下，由于大部分时钟源停止工作，调压器也降低了驱动能力，因此芯片功耗会显著降低。

通过 ICG 设置为自动启动时，如果 ICG 的 SWDTSLOFF 位为 1，专用看门狗在停止模式下停止计数；如果 SWDTSLOFF 位为 0，专用看门狗在停止模式下继续计数。

执行 WFI 指令进入停止模式之前，需要确保 FLASH 不处于编程或者擦除状态，并且振荡停止检测功能无效，否则芯片会进入睡眠模式而非停止模式。

执行 WFI 指令进入停止模式之前，必须确保 PWC\_PWRC0.PDMDS[1:0]配置为 0b00。

在停止模式下，ADC 和 DAC 也会发生功耗，除非在进入停止模式前将其禁止。要禁止 DAC，需要将 DACK.DAE,DAOE0,DAOE1 清“0”。要禁止 ADC，需要 ADC\_STR.START 位清“0”，置位 PWC\_FCG3.ADC3、PWC\_FCG3.ADC2、PWC\_FCG3.ADC1 后，执行 WFI 指令进入停止模式。

STOP 模式唤醒时，通过 PWC\_STPMCR 寄存器的位 CKSMRC 和 FLNWT 来选择唤醒后的时钟及是否需要等待 Flash 稳定。CKSMRC 用于控制唤醒后的时钟源，

CKSMRC =1 时，唤醒后的系统时钟源选择为 MRC；CKSMRC=0 时，唤醒后的系统时钟维持进入 STOP 之前的时钟源不变。FLNWT 用于控制唤醒后是否等待 Flash 稳定，FLNWT=0 时，唤醒时需要等待 Flash 稳定；FLNWT=1 时，唤醒是不需要等待 Flash 稳定。FLNWT 必须在程序运行在 RAM 上时才能置位，

否则芯片从 STOP 唤醒后的动作不可保证。在程序运行在 RAM 上进入 STOP 模式时，选择 CKSMRC =1，FLNWT=1 将可以在最短的时间内唤醒系统。

执行 WFI 指令进入停止模式之前，需要确保 DMA 处于停止状态，否则芯片可能会出现不可保证的动作。

STOP 模式下的漏电流在不同的电压温度是不同的，设置的驱动能力必须满足芯片的漏电需要。

执行 WFI 指令进入停止模式之前，需要将 EIRQ 的数字滤波设置为无效，否则该中断不能用于 STOP 唤醒。

芯片在超低速模式下需要设置 PWC\_PWRC1.STPDAS=11 后再进入 STOP 模式；如果 PWC\_PWRC1.STPDAS=00 时进入 STOP 模式，在 STOP 模式中芯片会消耗更多的电流。

通过非可屏蔽中断解除停止模式时，需要设置 INT\_NMIEER 的相应位使能该中断；通过可屏蔽中断解除停止模式时，需要设置 INT\_WUPENR 寄存器的相应位使能该中断的唤醒许可。在执行 WFI 或者 WFE 命令之前需要确保所有不用于停止模式唤醒用的中断已经被关闭。

- 进入停止模式
- 在 PWC\_STPMCR.STOP=1,PWC\_PWRC0.PWDN=0 时执行 WFI 指令即可进入停止模式。

表 5-5 给出了芯片的外设和时钟源在停止模式下的状态。

- 解除停止模式

停止模式可以通过复位和中断解除。可以用于解除停止模式的复位方式有管脚复位，上电复位，欠压复位(BOR)，可编程电压检测 1/2 复位，专用看门狗复位。可以用于解除停止模式的中断事件如下：

管脚中断EIRQ0-15、电压检测1中断、  
电压检测2中断、专用看门狗下溢中断、实时时钟的周期中断、闹钟中断、  
唤醒定时器中断、比较器中断、USART1 RX中断、Timer0比较匹配中断、  
Timer2比较匹配中断

芯片通过中断解除停止模式时，首先启动进入停止模式之前的使用到的时钟源。在所有的时钟源稳定之后，芯片解除停止模式。

### 5.5.3 掉电模式

在掉电模式中，VDD 域的所有模块的电源被切断，功耗可以达到最低。

通过 ICG 设置为自动启动时，如果 ICG 的 SWDTSLOFF 位为 1，专用看门狗将同 VDD 域的其他模块一样，电源被切断，不再计数。如果 SWDTSLOFF 位为 0，芯片将进入停止模式而非掉电模式，如果在 ICG 中设置为自动启动时，专用看门狗的振荡器和专用看门狗将继续运行。

当电压检测 1 和电压检测 2 的复位使能时，芯片将进入停止模式而非掉电模式。

执行 WFI 指令进入掉电模式之前，需要确保 FLASH 不处于编程或者擦除状态，并且振荡停止检测功能无效，否则芯片会进入睡眠模式而非掉电模式。

通过设置 PWC\_PWRC0.PDMD[1:0]，可以进一步降低掉电模式的功耗。掉电模式的子模式如表 5-6 所示。掉电模式 1 中电压检测电路可以使用，上电复位检测电路处于动作状态，由于唤醒时不需要等待 VCC 域参考电压、电压检测电路和上电复位检测电路的稳定，因此在实现低功耗的同时，唤醒时间最短。掉电模式 2 中，VCC 域参考电压电路，电压检测电路停止工作，上电复位检测电路处于动作状态，唤醒时需要等待 VCC 域参考电压电路、电压检测电路的稳定时间。掉电模式 3 中，VCC 域参考电压电路、电压检测电路、上电复位检测电路都停止工作，唤醒时需要等待这些电路的稳定，因此在实现最低功耗的同时，唤醒时间比掉电模式 2 和掉电模式 1 长。掉电模式 4 与掉电模式 3 中停止工作的电路相同，因此掉电模式 4 与掉电模式 3 具有相同的功耗。具体功耗数值和唤醒时间，请参考电气特性。

VDDR 域在掉电模式模式 1 和 掉电模式 2 下可以工作，所以实时时钟模块、唤醒计时器可以继续运行，并可以用于唤醒掉电模式。Ret\_SRAM 在掉电模式下依然可以保持数据。如果在掉电模式下实时时钟、唤醒计时器、 Ret\_SRAM 都不需要使用，可以置位

PWR\_PWRC1.VVDRSD 关闭低功耗调压器进一步降低功耗。

表 5-6 掉电模式子模式

掉电模式	PDMD[1:0]	功耗	唤醒时间	说明
掉电模式1	00	I <sub>PD1</sub>	T <sub>PD1</sub>	VCC域电源电压检测单元有效
掉电模式2	01	I <sub>PD2</sub>	T <sub>PD2</sub>	VCC域POR、PDR检测电路有效，BOR、PWD1、PWD2无效
掉电模式3	10	I <sub>PD3</sub>	T <sub>PD3</sub>	VCC域POR、PDR，BOR、PWD1、PWD2无效
掉电模式4	11	I <sub>PD4</sub>	T <sub>PD4</sub>	VCC域POR、PDR，BOR、PWD1、PWD2无效

功耗和唤醒时间的关系： I<sub>pd1</sub>>I<sub>pd2</sub>>I<sub>pd3</sub>=I<sub>pd4</sub>, T<sub>pd1</sub><T<sub>pd2</sub><T<sub>pd4</sub><T<sub>pd3</sub>

#### ■ 进入掉电模式

在 PWC\_STPMCR.STOP=1,PWC\_PWRC0.PWDN=1 时执行 WFI 指令即可进入掉电模式。

#### ■ 解除掉电模式

掉电模式可以通过掉电模式唤醒事件或者复位解除。能够用于唤醒掉电模式的复位有管脚复位、上电复位和欠压检测复位。能够用于掉电模式唤醒的事件包括：

WKUPn\_0/1/2/3(n=0/1/2/3)唤醒事件、实时时钟的闹钟和定时事件、电压检测1唤醒事件、电压检测2唤醒事件、唤醒定时器唤醒事件

从掉电模式 1、掉电模式 2 唤醒后，芯片复位后重新执行程序。唤醒事件可以通过掉电唤醒标志位查询，复位标志位可以通过 RSTF0.PDRF 查询。

在掉电模式 3 中 POR、PDR，BOR、PWD1、PWD2 电路都处于无效状态，从掉电模式 3 唤醒后，除 PWC\_PDWKF0/PWC\_PDWKF1/RSTF0 外的所有寄存器都被复位，芯片按照类似上电复位的方式工作；复位标志位可以通过 RMU\_RSTF0.PDRF 查询。

在掉电模式 4 中 POR、PDR、BOR、PWD1、PWD2 电路都处于无效状态，芯片复位后重新执行程序；复位标志位可以通过 RMU\_RSTF0.PDRF 查询。

掉电唤醒事件通过掉电唤醒使能寄存器(PWC\_PDWKE0-PDWKE3)、掉电唤醒事件边沿选择寄存器(PWC\_PDWKES)进行控制。当发生掉电唤醒事件时，该事件相对应的掉电唤醒标志(PWC\_PDWKF0-PWC\_PDWKF1)被置位。掉电唤醒后，如果不清除掉电唤醒标志，芯片不能再次进入掉电模式。掉电唤醒事件的边沿可以通过 PWC\_PDWKES 选择。

掉电模式唤醒时，VDD 域将重新供电，系统执行掉电唤醒复位，工作时钟内部中速振荡器。掉电模式唤醒时不被复位的寄存器如下表。

掉电模式	不被复位的寄存器
掉电模式1	PWC_PWRC0 PWC_PWRC1 PWC_PWRC3 PWC_PDWKE0 PWC_PDWKE1 PWC_PDWKE2
掉电模式2	PWC_PDWKES PWC_PDWKF0 PWC_PDWKF1 PWC_PVDCR0
掉电模式4	PWC_PVDCR1 PWC_PVDFCR PWC_PVDLCR PWC_PVIDCR PWC_PVDDSR
掉电模式3	PWC_PDWKF0 PWC_PDWKF1 RMU_RSTF0

#### ■ 解除掉电模式后的管脚状态

在掉电模式中，根据寄存器设定值，芯片管脚将保持进入掉电模式之前的状态或者高阻状态。如果 PWC\_PWRC0.IORTN[1:0]=10 或者 11,管脚状态在掉电模式中为高阻状态，掉电模式解除后管脚初始成高阻状态。如果 PWC\_PWRC0.IORTN[1:0]=00,管脚状态在掉电模式中保持掉电模式前的状态，唤醒后管脚初始成高阻状态。如果 PWC\_PWRC0.IORTN[1:0]=01,芯片管脚将保持进入掉电模式之前的状态，唤醒后即使对外设或者管脚的寄存器进行设定，芯片管脚的状态也不会改变。通过软件将 PWC\_PWRC0.IORTN 清零后，管脚状态才受外设或者管脚的寄存器设定所控制。

#### ■ WKTM 掉电模式唤醒事件

芯片内置了用于掉电唤醒的计数器 WKTM，该计数器可以选择内部低速振荡器、外部低速振荡器，在 RTC 使用外部低速振荡器作为时钟动作时还可以选择 64Hz 的内部时钟信号作为时钟源。该计数器是加计数的计数器，在 WKTC0.WKTCE 置位后，计数器开始计数，当计数值等于 WKTCMP[11:0] 设置值后，计数停止并产生唤醒事件，使芯片从掉电模式唤醒。再次使用 WKTM 时需要将 WKTC0.WKTCE 复位后再次置位。

#### ■ PTWK 掉电模式唤醒事件

芯片具有 4 个用于掉电模式唤醒的 PTWK 事件，PTWK0、PTWK1、PTWK2、PTWK3。通过软件设定 PWC\_PDWKE0/PWC\_PDWKE1 可以使能 WKUPn\_0、WKUPn\_1、WKUPn\_2、WKUPn\_3 等 4 个管脚中的一个用于触发 PTWK 事件。每个 PTWK 事件可以选择触发管脚的上升或者下降触发边沿，并具有独立的标志位。

PTWK 掉电唤醒事件配置流程如下：

1. 配置 PWC\_PDWKES 的相应位选择边沿
2. 配置 PWC\_PDWKE0 和 PWC\_PDWKE1 使能相应的管脚

### 3. 清除 PWC\_PDWKF0 中的相应标志

配置 PTWK<sub>n</sub> 的结构框图如图 5-11 所示。

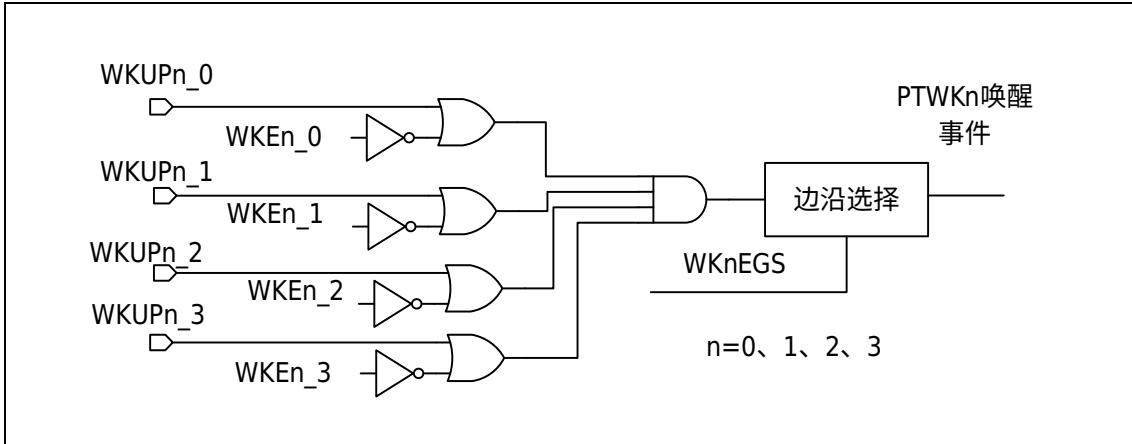


图 5-11 PTWK<sub>n</sub> 结构框图

#### ■ 降低功耗的方法

可以通过下列方法优化运行模式下的功耗。

1. 降低系统时钟速度
2. 关闭不使用的时钟源
3. 设定功能时钟控制寄存器 PWC\_FCG<sub>n</sub> ( $n=0/1/2/3$ ) 关闭不需要使用的功能
4. 关闭 RAM 的电源

## 5.5.4 降低系统时钟速度

在运行模式下，可通过对预分频寄存器编程来降低系统时钟(HCLK)、外部总线时钟(EXCLK)、周边外设时钟 PCLK0/PCLK1/PCLK2/PCLK3/PCLK4 的速度。进入睡眠模式前，也可以使用这些预分频器降低外设速度。有关详细信息，请参考【时钟控制器 (CMU)】。

## 5.5.5 关闭不使用的时钟源

芯片的系统时钟有 6 个时钟源：

- 外部高速振荡器(XTAL)
- 外部低速振荡器(XTAL32)
- PLLH 时钟(PLLH)
- 内部高速振荡器(HRC)
- 内部中速振荡器(MRC)
- 内部低速振荡器(LRC)

SWDT 具有独立的专用内部低速振荡器(SWDTLRC)；RTC 可以选择外部低速振荡器或者内部低速振荡器作为时钟源。HRC 和 PLL 都配有独立的电源电路，在 HRC 关闭后，可以通过置位 PWC\_PWRC1.VHRCSD

位关闭 HRC 的电源，进一步降低功耗；在 PLLH 都被关闭后，可以通过设置 PWC\_PWRC1.VPLLSD=11 关闭 PLLH 的电源。

有关详细信息，请参考【时钟控制器（CMU）】。

### 5.5.6 功能时钟停止

芯片的外设模块设有功能时钟停止功能，通过将寄存器对应的位置位，可以将不需要使用的模块停止运行，相应模块的时钟也停止供给，降低功耗。在模块停止状态下，模块内部的寄存器将维持停止之前的状态。

### 5.5.7 关闭不使用的 RAM

芯片里的每个 RAM 模块都配置了功能时钟停止位，掉电控制位，通过置位模块停止位停止给不需要使用的 RAM 提供时钟，从而降低功耗。通过置位模块的掉电控制位，可以让相应的 RAM 模块掉电，从而降低功耗。表 5-7 是 RAM 模块与掉电控制位的对应关系，通过设置 PWC\_RAMPC0 寄存器的相应 RAMPDCn(n=0-10)可以使对应的 RAM 掉电。

表 5-7 RAM 模块与 RAM 掉电控制位

RAM模块	说明	掉电控制位
SRAM0	0x20000000~0x200007FFF 地址空间用RAM	PWC_RAMPC0.RAMPDC0
SRAMH	0x1FFF8000~0x1FFFFFFF 地址空间用RAM	PWC_RAMPC0.RAMPDC10
CAN_1 RAM	CAN单元1用RAM	PWC_PRAMLPC.PRAMPDC0
CAN_2 RAM	CAN单元2用RAM	PWC_PRAMLPC.PRAMPDC1
CAN_3RAM	CAN单元3用RAM	PWC_PRAMLPC.PRAMPDC1
CACHERAM	Cache用RAM	PWC_PRAMLPC.PRAMPDC2
USBFS RAM	USBFS用RAM	PWC_PRAMLPC.PRAMPDC3
Ret_SRAM	0x200F0000~0x200F0FFF 地址空间用RAM	PWR_PWRC0.RETRAMSD

## 5.6 寄存器保护功能

寄存器保护功能用于将寄存器的写操作无效，以保护寄存器被意外改写。表 5-8 是寄存器保护位和被保护寄存器的列表。

表 5-8 寄存器保护列表

保护寄存器位	被保护寄存器
PWC_FPRC.FPRCB0	CMU_XTALCFG, CMU_XTALSTBCR, CMU_XTALCR, CMU_XTALSTDRCR, CMU_XTALSTDSCR, CMU_HRCTRM, CMU_HRCCR, CMU_MRCTR, CMU_MRCCR, CMU_PLLHCGR, CMU_PLLHCR, CMU_PLLACFGR, CMU_PLLACR, CMU_OSCSTBSR, CMU_CKSWR, CMU_SCFGR, CMU_USBCKCFGR, CMU_CANCKCFGR, CMU_TPIUCKCFGR, CMU_MCO1CFGR, CMU_MCO2CFGR, CMU_XTAL32CR, CMU_XTALC32CFGR, CMU_XTAL32NFR, CMU_LRCCR, CMU_LRCTRM
PWC_FPRC.FPRCB1	PWC_PWRC0, PWC_PWRC1, PWC_PWRC4, PWC_PWRC5, PWC_PWRC6, PWC_PDWKE0, PWC_PDWKE1, PWC_PDWKE2, PWC_PDWKES, PWC_PDWKF0, PWC_PDWKF1, CMU_PERICKSEL, CMU_I2SCKSEL, PWC_STPMCR, PWC_RAMPC0, PWC_PRAMLPC, RMU_RSTF0, PWC_WKTCR, RMU_FRST0, RMU_FRST1, RMU_FRST2, RMU_FRST3, RMU_PRSTCR0
PWC_FPRC.FPRCB3	PWC_PVDCR0, PWC_PVDCR1, PWC_PVDFCR, PWC_PVDLCR, PWC_PVDICR, PWC_PVDDSR

保护寄存器位	被保护寄存器
PWC_FCG0PC.B0	PWC_FCG0

## 5.7 寄存器说明

寄存器一览表如表 5-9 所示。

表 5-9 寄存器一览表

BASE ADDR: 0x4004CC00

寄存器名	符号	偏移地址	位宽	复位值
电源模式控制寄存器0	PWC_PWRC0	0x00	8	0x00
电源模式控制寄存器1	PWC_PWRC1	0x04	8	0x00
电源模式控制寄存器4	PWC_PWRC4	0x10	8	0x00
PVD控制寄存器0	PWC_PVDCR0	0x14	8	0x00
PVD控制寄存器1	PWC_PVDCR1	0x18	8	0x00
PVD滤波控制寄存器	PWC_PVDFCR	0x1C	8	0x11
PVD电平控制寄存器	PWC_PVDLCR	0x20	8	0x00
掉电唤醒使能寄存器0	PWC_PDWKE0	0x28	8	0x00
掉电唤醒使能寄存器1	PWC_PDWKE1	0x2C	8	0x00
掉电唤醒使能寄存器2	PWC_PDWKE2	0x30	8	0x00
掉电唤醒事件边沿选择寄存器	PWC_PDWKES	0x34	8	0x00
掉电唤醒标志寄存器0	PWC_PDWKF0	0x38	8	0x00
掉电唤醒标志寄存器1	PWC_PDWKF1	0x3C	8	0x00
电源模式控制寄存器5	PWC_PWRC5	0x40	8	0x00
电源模式控制寄存器6	PWC_PWRC6	0x44	8	0x00
PVD中断控制寄存器	PWC_PVDICR	0xC0	8	0x00
PVD检测状态寄存器	PWC_PVDDSR	0xC4	8	0x11
RAM功耗控制寄存器0	PWC_RAMPC0	0xC8	32	0x00000000
外设RAM低功耗控制寄存器	PWC_PRAMLPC	0xD0	32	0x00000000

BASE ADDR: 0x40054000

寄存器名	符号	偏移地址	位宽	复位值
STOP模式控制寄存器	PWC_STPMCR	0x0C	16	0x0000
功能保护控制寄存器	PWC_FPRC	0x3FE	8	0x00

BASE ADDR: 0x4004C400

寄存器名	符号	偏移地址	位宽	复位值
唤醒计时器控制寄存器	PWC_WKTCR	0x00	16	0x0000

BASE ADDR:0x40048000

寄存器名	符号	偏移地址	位宽	复位值
功能时钟控制0	PWC_FCG0	0x00	32	0xFFFFFA0E
功能时钟控制1	PWC_FCG1	0x04	32	0xFFFFFFFF
功能时钟控制2	PWC_FCG2	0x08	32	0xFFFFFFFF
功能时钟控制3	PWC_FCG3	0x0C	32	0xFFFFFFFF
PWC_FCG0保护控制	PWC_FCG0PC	0x10	32	0x00000000

### 5.7.1 电源模式控制寄存器 0 (PWC\_PWRCO)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
PWDN	-	IORTN[1:0]	-	-	-	PDMDS[1:0]	
<hr/>							
位	标记	位名	功能			读写	
b7	PWDN	掉电模式控制位	0: 掉电模式无效 1: 掉电模式使能			R/W	
b6	Reserved	-	读出时为“0”，写入时写“0”			R/W	
b5-b4	IORTN[1:0]	掉电模式下IO保持控制	00: 掉电模式中IO保持状态，掉电唤醒后硬件解除IO保持状态 01: 掉电模式中IO保持状态，掉电唤醒后将IORTN[1:0]设为00b，解除IO保持状态 1x: 掉电模式中及掉电唤醒后IO为高阻			R/W	
b3-b2	Reserved	-	读出时为“0”，写入时写“0”			R/W	
b1-b0	PDMDS[1:0]	掉电模式选择控制	PDMD[1:0] 00: 掉电模式1 01: 掉电模式2 10: 掉电模式3 11: 掉电模式4			R/W	

### 5.7.2 电源模式控制寄存器 1 (PWC\_PWRC1)

复位值: 0xC0

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	VHRCSD	VPLLSD[1:0]	
位	标记	位名	功能			读写	
b7-b6	Reserved	-	读出时为“11”，写入时写“11”			R/W	
b5-b4	Reserved	-	读出时为“0”，写入时写“0”			R/W	
b2	VHRCSD	HRC电源关闭	在HRC不使用时，置位VHRCSD后关闭HRC用电源，进一步降低功耗。VHRCSD清零后，需要等待25uS后再启动HRC模块			R/W	
b1-b0	VPLLSD[1:0]	PLL电源关闭	00: PLL电源使能 11: PLL电源关闭 01: 设定禁止 10: 设定禁止 在PLL关闭并等待50us后，设置VPLLSD=11后关闭PLL用电源，进一步降低功耗。设置VPLLSD=00后，需要等待25uS后再启动PLL模块。			R/W	

### 5.7.3 电源模式控制寄存器 4 (PWC\_PWRC4)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
ADBUFE	-	-	-	-	-	-	-
位	标记	位名	功能			读写	
b7	ADBUFE	内部电压采样使能	使用AD采样芯片内部电压时，需要设置本位为1 0: 无效 1: 有效			R/W	
b6-b0	Reserved	-	读出时为“0”，写入时写“0”			R/W	

### 5.7.4 掉电唤醒使能寄存器 0(PWC\_PDWKE0)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
WKE13	WKE12	WKE11	WKE10	WKE03	WKE02	WKE01	WKE00

位	标记	位名	功能	读写
b7	WKE13	WKUP1_3唤醒事件使能	0: WKUP1_3唤醒事件无效 1: WKUP1_3唤醒事件使能	R/W
b6	WKE12	WKUP1_2唤醒事件使能	0: WKUP1_2唤醒事件无效 1: WKUP1_2唤醒事件使能	R/W
b5	WKE11	WKUP1_1唤醒事件使能	0: WKUP1_1唤醒事件无效 1: WKUP1_1唤醒事件使能	R/W
b4	WKE10	WKUP1_0唤醒事件使能	0: WKUP1_0唤醒事件无效 1: WKUP1_0唤醒事件使能	R/W
b3	WKE03	WKUP0_3唤醒事件使能	0: WKUP0_3唤醒事件无效 1: WKUP0_3唤醒事件使能	R/W
b2	WKE02	WKUP0_2唤醒事件使能	0: WKUP0_2唤醒事件无效 1: WKUP0_2唤醒事件使能	R/W
b1	WKE01	WKUP0_1唤醒事件使能	0: WKUP0_1唤醒事件无效 1: WKUP0_1唤醒事件使能	R/W
b0	WKE00	WKUP0_0唤醒事件使能	0: WKUP0_0唤醒事件无效 1: WKUP0_0唤醒事件使能	R/W

### 5.7.5 掉电唤醒使能寄存器 1(PWC\_PDWKE1)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
WKE33	WKE32	WKE31	WKE30	WKE23	WKE22	WKE21	WKE20

位	标记	位名	功能	读写
b7	WKE33	WKUP3_3唤醒事件使能	0: WKUP3_3唤醒事件无效 1: WKUP3_3唤醒事件使能	R/W
b6	WKE32	WKUP3_2唤醒事件使能	0: WKUP3_2唤醒事件无效 1: WKUP3_2唤醒事件使能	R/W
b5	WKE31	WKUP3_1唤醒事件使能	0: WKUP3_1唤醒事件无效 1: WKUP3_1唤醒事件使能	R/W
b4	WKE30	WKUP3_0唤醒事件使能	0: WKUP3_0唤醒事件无效 1: WKUP3_0唤醒事件使能	R/W
b3	WKE23	WKUP2_3唤醒事件使能	0: WKUP2_3唤醒事件无效 1: WKUP2_3唤醒事件使能	R/W
b2	WKE22	WKUP2_2唤醒事件使能	0: WKUP2_2唤醒事件无效 1: WKUP2_2唤醒事件使能	R/W
b1	WKE21	WKUP2_1唤醒事件使能	0: WKUP2_1唤醒事件无效 1: WKUP2_1唤醒事件使能	R/W
b0	WKE20	WKUP2_0唤醒事件使能	0: WKUP2_0唤醒事件无效 1: WKUP2_0唤醒事件使能	R/W

### 5.7.6 掉电唤醒使能寄存器 2(PWC\_PDWKE2)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
WKTMWKE	-	RTCALMWKE	RTCPRDWKE	-	-	VD2WKE	VD1WKE

位	标记	位名	功能	读写
b7	WKTMWKE	WKTM唤醒事件使能	0: WKTM唤醒事件无效 1: WKTM唤醒事件使能	R/W
b6	Reserved	-	读出时为“0”，写入时写“0”	R/W
b5	RTCALMWKE	RTC闹钟唤醒事件使能	0: RTC闹钟唤醒事件无效 1: RTC闹钟唤醒事件使能	R/W
b4	RTCPRDWKE	RTC周期唤醒事件使能	0: RTC周期唤醒事件无效 1: RTC周期唤醒事件使能	R/W
b3-b2	Reserved	-	读出时为“0”，写入时写“0”	R/W
b1	VD2WKE	PVD2唤醒事件使能	0: PVD2唤醒事件无效 1: PVD2唤醒事件使能	R/W
b0	VD1WKE	PVD1唤醒事件使能	0: PVD1唤醒事件无效 1: PVD1唤醒事件使能	R/W

### 5.7.7 掉电唤醒事件边沿选择寄存器(PWC\_PDWKES)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	VD2EGS	VD1EGS	WK3EGS	WK2EGS	WK1EGS	WK0EGS

位	标记	位名	功能	读写
b7-b6	Reserved	-	读出时为“0”，写入时写“0”	R/W
b5	VD2EGS	VD2边沿选择	0: VCC < VPVD2 1: VCC > VPVD2	R/W
b4	VD1EGS	VD1边沿选择	0: VCC < VPVD1 1: VCC > VPVD1	R/W
b3	WK3EGS	PTWK3边沿选择	0: 下降沿 1: 上升沿	R/W
b2	WK2EGS	PTWK2边沿选择	0: 下降沿 1: 上升沿	R/W
b1	WK1EGS	PTWK1边沿选择	0: 下降沿 1: 上升沿	R/W
b0	WK0EGS	PTWK0边沿选择	0: 下降沿 1: 上升沿	R/W

### 5.7.8 掉电唤醒标志寄存器 0(PWC\_PDWKF0)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	VD2WKF	VD1WKF	PTWK3F	PTWK2F	PTWK1F	PTWK0F
<hr/>							
位	标记	位名	功能			读写	
b7-b6	Reserved	-	读出时为“0”，写入时写“0”			R/W	
b5	VD2WKF	PVD2唤醒标志位	0: 未发生PVD2唤醒事件 1: 发生PVD2唤醒事件 掉电唤醒后，需要写零清除本位。			R/W	
b4	VD1WKF	PVD1唤醒标志位	0: 未发生PVD1唤醒事件 1: 发生PVD1唤醒事件 掉电唤醒后，需要写零清除本位。			R/W	
b3	PTWK3F	PTWK3唤醒标志位	0: 未发生PTWK3唤醒事件 1: 发生PTWK3唤醒事件 掉电唤醒后，需要写零清除本位。			R/W	
b2	PTWK2F	PTWK2唤醒标志位	0: 未发生PTWK2唤醒事件 1: 发生PTWK2唤醒事件 掉电唤醒后，需要写零清除本位。			R/W	
b1	PTWK1F	PTWK1唤醒标志位	0: 未发生PTWK1唤醒事件 1: 发生PTWK1唤醒事件 掉电唤醒后，需要写零清除本位。			R/W	
b0	PTWK0F	PTWK0唤醒标志位	0: 未发生PTWK0唤醒事件 1: 发生PTWK0唤醒事件 掉电唤醒后，需要写零清除本位。			R/W	

### 5.7.9 掉电唤醒标志寄存器 1(PWC\_PDWKF1)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
WKTMWKF	-	RTCALMWKF	RTCPRDWKF	-	-	-	

位	标记	位名	功能	读写
b7	WKTMWKF	WKTM唤醒标志位	0: 未发生WKTM唤醒事件 1: 发生WKTM唤醒事件 掉电唤醒后，需要写零清除本位。	R/W
b6	Reserved	-	读出时为“0”，写入时写“0”	R/W
b5	RTCALMWKF	RTC闹钟唤醒标志位	0: 未发生RTC闹钟唤醒事件 1: 发生RTC闹钟唤醒事件 掉电唤醒后，需要写零清除本位。	R/W
b4	RTCPRDWKF	RTC周期唤醒标志位	0: 未发生RTC周期唤醒事件 1: 发生RTC周期唤醒事件 掉电唤醒后，需要写零清除本位。	R/W
b3~0	Reserved	-	读出时为“0”，写入时写“0”	R/W

### 5.7.10 电源模式控制寄存器 5 (PWC\_PWRC5)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
CSDIS	-	-	-	-	-	RETRAMSD	VVDRSD

位	标记	位名	功能	读写
b7	CSDIS	电流源无效控制	0: 电流源有效 1: 电流源无效 在XTAL32/RTC/WTKM/Ret_SRAM等都不需要使用时，可以置位CSDIS，以降低功耗。	R/W
b6	Reserved	-	读出时为“0”，写入时写“0”	R/W
b5	Reserved	-	读出时为“0”，写入时写“0”	R/W
b4	Reserved	-	读出时为“0”，写入时写“0”	R/W
b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2	Reserved	-	读出时为“0”，写入时写“0”	R/W
b1	RETRAMSD	保持RAM掉电控制	0: Ret_SRAM不掉电 1: Ret_SRAM 掉电	R/W
b0	VVDRSD	RLDO关闭控制位	0: RLDO有效 1: RLDO无效 置位VVDRSD且芯片进入掉电模式后RT/WKTM/Ret_SRAM的电源被关闭，降低功耗。	R/W

### 5.7.11 电源模式控制寄存器 6 (PWC\_PWRC6)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1-b0
WDRTNE	SWRTNE	WDRDAC	SWRDAC	-	IORTNCLR	RTCCKSEL[1:0]
<hr/>						
位	标记	位名	功能		读写	
b7	WDRTNE		0: WDT复位/SWDT复位可以使IO复位 1: WDT复位/SWDT复位不会让IO复位，通过IORTNCLR位写“1”清除本位才能使IO复位		R/W	
b6	SWRTNE		0: 软件复位可以使IO复位 1: 软件复位不会让IO复位，通过IORTNCLR位写“1”清除本位才能使IO复位		R/W	
b5	WDRDAC		0: WDT复位/SWDT复位可以使DAC复位 1: WDT复位/SWDT复位不会让DAC复位，可以通过写RMU_FRST3.DAC1/2/3/4软件复位位将DAC复位		R/W	
b4	SWRDAC		0: 软件复位可以使DAC复位 1: 软件复位不会让DAC复位，可以通过写RMU_FRST3.DAC1/2/3/4位将DAC复位		R/W	
b3	Reserved	-	读出时为“0”，写入时写“0”		R/W	
b2	IORTNCLR		在SWRTNE=1时发生软件复位后，IO保持复位前的状态；在WDRTNE=1时发生WDT复位/SWDT复位后，IO保持复位前的状态。通过往本位写1，清除由SWRTNE=1时产生的软件复位引起的IO保持状态，由WDRTNE=1时产生的WDT复位/SWDT复位因此的IO保持状态。本位读出值为0。		R/W	
b1-0	RTCCKSEL[1:0]		00: LRC 01: XTAL小数分频 10: 端子输入 11: 设定禁止。		R/W	

### 5.7.12 功能时钟控制 0(PWC\_FCG0)

复位值:0xFFFFFA0E

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	DCU3	DCU2	DCU1	CRC	TRNG	HASH	AES	MAU	CTC	AOS	FCM
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DMA2	DMA1	KEY	-	PLA	SRAMRET	-	-	-	-	-	SRAM0	-	-	-	SRAMH

位	标记	位名	功能	读写
b31-b27	Reserved	-	读出时为“1”，写入时写“1”	R/W
b26	DCU3	DCU3功能控制	0: 数字计算单元DCU3功能使能 1: 数字计算单元DCU3功能无效	R/W
b25	DCU2	DCU2功能控制	0: 数字计算单元DCU2功能使能 1: 数字计算单元DCU2功能无效	R/W
b24	DCU1	DCU1功能控制	0: 数字计算单元DCU1功能使能 1: 数字计算单元DCU1功能无效	R/W
b23	CRC	CRC功能控制	0: CRC功能使能 1: CRC功能无效	R/W
b22	TRNG	TRNG功能控制	0: 加密协处理模块CPM中的真随机发生器TRNG功能使能 1: 加密协处理模块CPM中的真随机发生器TRNG功能无效	R/W
b21	HASH	HASH功能控制	0: 加密协处理模块CPM中的安全散列算法模块HASH功能使能 1: 加密协处理模块CPM中的安全散列算法模块HASH功能无效	R/W
b20	AES	AES功能控制	0: 加密协处理模块CPM中的加解密算法处理器AES功能使能 1: 加密协处理模块CPM中的加解密算法处理器AES功能无效	R/W
b19	MAU	MAU功能控制	0: 数学运算单元MAU功能使能 1: 数学运算单元MAU功能无效	R/W
b18	CTC	CTC功能控制	0: 内部时钟校准器CTC功能使能 1: 内部时钟校准器CTC功能无效	R/W
b17	AOS	AOS功能控制	0: 自动运行系统AOS功能使能 1: 自动运行系统AOS功能无效	R/W
b16	FCM	FCM功能控制	0: 时钟控制器CMU中的时钟频率测量模块FCM功能使能 1: 时钟控制器CMU中的时钟频率测量模块FCM功能无效	R/W
b15	DMA2	DMA2功能控制	0: DMA控制器DMA2功能使能 1: DMA控制器DMA2功能无效	R/W
b14	DMA1	DMA1功能控制	0: DMA控制器DMA1功能使能 1: DMA控制器DMA1功能无效	R/W
b13	KEY	KEYSCAN功能控制	0: 键盘扫描控制模块KEYSCAN功能使能 1: 键盘扫描控制模块KEYSCAN功能无效	R/W
b12	Reserved	-	读出时为“1”，写入时写“1”	R/W
b11	PLA	PLA功能控制	0: PLA功能使能 1: PLA功能无效	R/W
b10	SRAMRET	Ret_SRAM功能控制	0: Ret_SRAM功能使能 1: Ret_SRAM功能无效	R/W
b9-b5	Reserved	-	读出时为“5'b10000”，写入时写“5'b10000”	R/W
b4	SRAM0	SRAM0功能控制	0: SRAM0 (0x20000000~0x20007FFF) 功能使能	R/W

1: SRAM0 (0x20000000~0x20007FFF) 功能无效				
b3-b1	Reserved	-	读出时为“1”，写入时写“1”	R/W
b0	SRAMH	SRAMH功能控制	0: SRAMH (0x1FFF8000~0x1FFFFFFF) 功能使能 1: SRAMH (0x1FFF8000~0x1FFFFFFF) 功能使能	R/W

### 5.7.13 功能时钟控制 1(PWC\_FCG1)

复位值：0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	FMAC4	FMAC3	FMAC2	FMAC1
b23	b22	b21	b20	b19	b18	b17	b16
-	USBFS	-	-	SPI4	SPI3	SPI2	SPI1
b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	-	-	-	MDIO
b7	b6	b5	b4	b3	b2	b1	b0
-	I2C3	I2C2	I2C1	QSPI	CAN3	CAN2	CAN1

位	标记	位名	功能	读写
b31-b28	Reserved	-	读出时为“1”，写入时写“1”	R/W
b27	FMAC4	FMAC4功能控制	0: 滤波数学加速器FMAC单元4功能使能 1: 滤波数学加速器FMAC单元4功能无效	R/W
b26	FMAC3	FMAC3功能控制	0: 滤波数学加速器FMAC单元3功能使能 1: 滤波数学加速器FMAC单元3功能无效	R/W
b25	FMAC2	FMAC2功能控制	0: 滤波数学加速器FMAC单元2功能使能 1: 滤波数学加速器FMAC单元2功能无效	R/W
b24	FMAC1	FMAC1功能控制	0: 滤波数学加速器FMAC单元1功能使能 1: 滤波数学加速器FMAC单元1功能无效	R/W
b23	Reserved	-	读出时为“1”，写入时写“1”	R/W
b22	USBFS	USBFS功能控制	0: USB2.0全速模块USBFS功能使能 1: USB2.0全速模块USBFS功能无效	R/W
b21~b20	Reserved	-	读出时为“1”，写入时写“1”	R/W
b19	SPI4	SPI4功能控制	0: 串行外设接口SPI单元4功能使能 1: 串行外设接口SPI单元4功能无效	R/W
b18	SPI3	SPI3功能控制	0: 串行外设接口SPI单元3功能使能 1: 串行外设接口SPI单元3功能无效	R/W
b17	SPI2	SPI2功能控制	0: 串行外设接口SPI单元2功能使能 1: 串行外设接口SPI单元2功能无效	R/W
b16	SPI1	SPI1功能控制	0: 串行外设接口SPI单元1功能使能 1: 串行外设接口SPI单元1功能无效	R/W
b15~b9	Reserved	-	读出时为“1”，写入时写“1”	R/W
b8	MDIO	MDIO功能控制	0: MDIO功能使能 1: MDIO功能无效	R/W
b7	Reserved	-	读出时为“1”，写入时写“1”	R/W
b6	I2C3	I2C3功能控制	0: 集成电路总线I2C单元3功能使能 1: 集成电路总线I2C单元3功能无效	R/W
b5	I2C2	I2C2功能控制	0: 集成电路总线I2C单元2功能使能 1: 集成电路总线I2C单元2功能无效	R/W
b4	I2C1	I2C1功能控制	0: 集成电路总线I2C单元1功能使能 1: 集成电路总线I2C单元1功能无效	R/W

b3	QSPI	QSPI功能控制	0: 四线式串行外设接口QSPI功能使能 1: 四线式串行外设接口QSPI功能无效	R/W
b2	CAN3	CAN3功能控制	0: CAN单元3功能使能 1: CAN单元3功能无效	R/W
b1	CAN2	CAN2功能控制	0: CAN单元2功能使能 1: CAN单元2功能无效	R/W
b0	CAN1	CAN1功能控制	0: CAN单元1功能使能 1: CAN单元1功能无效	R/W

### 5.7.14 功能时钟控制 2(PWC\_FCG2)

复位值：0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	TMRA_6	TMRA_5	TMRA_4	TMRA_3	TMRA_2	TMRA_1	TMR2_4	TMR2_3	TMR2_2	TMR2_1
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EMB	-	TMRO_2	TMRO_1	-	TMR4	TMR6_10	TMR6_9	TMR6_8	TMR6_7	TMR6_6	TMR6_5	TMR6_4	TMR6_3	TMR6_2	TMR6_1

位	标记	位名	功能	读写
b31~b26	Reserved	-	读出时为“1”，写入时写“1”	R/W
b25	TMRA_6	TMRA_6功能控制	0: TimerA单元6功能使能 1: TimerA单元6功能无效	R/W
b24	TMRA_5	TMRA_5功能控制	0: TimerA单元5功能使能 1: TimerA单元5功能无效	R/W
b23	TMRA_4	TMRA_4功能控制	0: TimerA单元4功能使能 1: TimerA单元4功能无效	R/W
b22	TMRA_3	TMRA_3功能控制	0: TimerA单元3功能使能 1: TimerA单元3功能无效	R/W
b21	TMRA_2	TMRA_2功能控制	0: TimerA单元2功能使能 1: TimerA单元2功能无效	R/W
b20	TMRA_1	TMRA_1功能控制	0: TimerA单元1功能使能 1: TimerA单元1功能无效	R/W
b19	TMR2_4	TMR2_4功能控制	0: Timer2单元4功能使能 1: Timer2单元4功能无效	R/W
b18	TMR2_3	TMR2_3功能控制	0: Timer2单元3功能使能 1: Timer2单元3功能无效	R/W
b17	TMR2_2	TMR2_2功能控制	0: Timer2单元2功能使能 1: Timer2单元2功能无效	R/W
b16	TMR2_1	TMR2_1功能控制	0: Timer2单元1功能使能 1: Timer2单元1功能无效	R/W
b15	EMB	EMB功能控制	0: 紧急刹车模块EMB功能使能 1: 紧急刹车模块EMB功能无效	R/W
b14	Reserved	-	读出时为“1”，写入时写“1”	R/W
b13	TMRO_2	TMRO_2功能控制	0: Timer0_2功能使能 1: Timer0_2功能无效	R/W
b12	TMRO_1	TMRO_1功能控制	0: Timer0_1功能使能 1: Timer0_1功能无效	R/W
b11	Reserved	-	读出时为“1”，写入时写“1”	R/W
b10	TMR4	TMR4功能控制	0: Timer4功能使能 1: Timer4功能无效	R/W
b9	TMR6_10	TMR6_10功能控制	0: Timer6单元10功能使能 1: Timer6单元10功能无效	R/W
b8	TMR6_9	TMR6_9功能控制	0: Timer6单元9功能使能 1: Timer6单元9功能无效	R/W
b7	TMR6_8	TMR6_8功能控制	0: Timer6单元8功能使能	R/W

			1: Timer6单元8功能无效	
b6	TMR6_7	TMR6_7功能控制	0: Timer6单元7功能使能 1: Timer6单元7功能无效	R/W
b5	TMR6_6	TMR6_6功能控制	0: Timer6单元6功能使能 1: Timer6单元6功能无效	R/W
b4	TMR6_5	TMR6_5功能控制	0: Timer6单元5功能使能 1: Timer6单元5功能无效	R/W
b3	TMR6_4	TMR6_4功能控制	0: Timer6单元4功能使能 1: Timer6单元4功能无效	R/W
b2	TMR6_3	TMR6_3功能控制	0: Timer6单元3功能使能 1: Timer6单元3功能无效	R/W
b1	TMR6_2	TMR6_2功能控制	0: Timer6单元2功能使能 1: Timer6单元2功能无效	R/W
b0	TMR6_1	TMR6_1功能控制	0: Timer6单元1功能使能 1: Timer6单元1功能无效	R/W

### 5.7.15 功能时钟控制 3(PWC\_FCG3)

复位值：0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	-	-	USART6	USART5
b23	b22	b21	b20	b19	b18	b17	b16
USART4	USART3	USART2	USART1	-	-	-	SMC
b15	b14	b13	b12	b11	b10	b9	b8
	-	VREF	OTS	-	-	CMP34	CMP12
b7	b6	b5	b4	b3	b2	b1	b0
DAC4	DAC3	DAC2	DAC1	CMBIAS	ADC3	ADC2	ADC1

位	标记	位名	功能	读写
b31-b30	Reserved	-	读出时为“1”，写入时写“1”	R/W
b25	USART6	USART6功能控制	0: 通用同步异步收发器USART单元6功能使能 1: 通用同步异步收发器USART单元6功能无效	R/W
b24	USART5	USART5功能控制	0: 通用同步异步收发器USART单元5功能使能 1: 通用同步异步收发器USART单元5功能无效	R/W
b23	USART4	USART4功能控制	0: 通用同步异步收发器USART单元4功能使能 1: 通用同步异步收发器USART单元4功能无效	R/W
b22	USART3	USART3功能控制	0: 通用同步异步收发器USART单元3功能使能 1: 通用同步异步收发器USART单元3功能无效	R/W
b21	USART2	USART2功能控制	0: 通用同步异步收发器USART单元2功能使能 1: 通用同步异步收发器USART单元2功能无效	R/W
b20	USART1	USART1功能控制	0: 通用同步异步收发器USART单元1功能使能 1: 通用同步异步收发器USART单元1功能无效	R/W
b19	Reserved	-	读出时为“1”，写入时写“1”	R/W
b16	SMC	SMC功能控制	0: 外部存储控制器(EXMC)的SMC控制器功能有效 1: 外部存储控制器(EXMC)的SMC控制器功能无效	R/W
b15-b14	Reserved	-	读出时为“1”，写入时写“1”	R/W
b13	VREF	VREF功能控制	0: VREF功能有效 1: VREF功能无效	R/W
b12	OTS	OTS功能控制	0: 温度传感器OTS功能有效 1: 温度传感器OTS功能无效	R/W
b11-b10	Reserved	-	读出时为“1”，写入时写“1”	R/W
b9	CMP34	CMP34功能控制	0: 电压比较器CMP通道3和4功能使能 1: 电压比较器CMP通道3和4功能无效	R/W
b8	CMP12	CMP12功能控制	0: 电压比较器CMP通道1和2功能使能 1: 电压比较器CMP通道1和2功能无效	R/W
b7-b6	Reserved	-	读出时为“1”，写入时写“1”	R/W
b7	DAC4	DAC4功能控制	0: 数模转换器DAC单元4功能使能 1: 数模转换器DAC单元4功能无效	R/W
b6	DAC3	DAC3功能控制	0: 数模转换器DAC单元3功能使能 1: 数模转换器DAC单元3功能无效	R/W

b5	DAC2	DAC2功能控制	0: 数模转换器DAC单元2功能使能 1: 数模转换器DAC单元2功能无效	R/W
b4	DAC1	DAC1功能控制	0: 数模转换器DAC单元1功能使能 1: 数模转换器DAC单元1功能无效	R/W
b3	CMBIAS	CMP/PGA基准电流源控制	0: CMP/PGA基准电流源有效 1: CMP/PGA基准电流源无效	R/W
b2	ADC3	ADC3功能控制	0: 模数转换模块ADC单元3功能使能 1: 模数转换模块ADC单元3功能无效	R/W
b1	ADC2	ADC2功能控制	0: 模数转换模块ADC单元2功能使能 1: 模数转换模块ADC单元2功能无效	R/W
b0	ADC1	ADC1功能控制	0: 模数转换模块ADC单元1功能使能 1: 模数转换模块ADC单元1功能无效	R/W

### 5.7.16 PWC\_FCG0 保护控制(PWC\_FCG0PC)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
FCG0PCWE[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PRT0
<hr/>															
位	标记	位名	功能	读写											
b31~b16	FCG0PCWE[15:0]	PWC_FCG0PC写使能	写入0xA5A5的同时改变PRT0位的值	R/W											
b15-b1	Reserved	-	读出时为“0”，写入时写“0”	R/W											
b0	PRT0	保护位	0: PWC_FCG0写无效 1: PWC_FCG0写使能	R/W											

### 5.7.17 功能保护控制寄存器(PWC\_FPRC)

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PWC_FPRCWE[7:0]								FPRCB7	FPRCB6	FPRCB5	FPRCB4	FPRCB3	FPRCB2	FPRCB1	FPRCB0
<hr/>															
位	标记	位名	功能	读写											
b15~b8	PWC_FPRCWE[7:0]	PWC_FPRC寄存器写使能	写入0xA5的同时能够更新PWC_FPRC值，否则对低8位写入值无效。读出时为0x00。	R/W											
b7	FPRCB7	FPRC位7	预留，读出时为“0”，写入时写“0”	R/W											
b6	FPRCB6	FPRC位6	预留，读出时为“0”，写入时写“0”	R/W											
b5	FPRCB5	FPRC位5	预留，读出时为“0”，写入时写“0”	R/W											
b4	FPRCB4	FPRC位4	预留，读出时为“0”，写入时写“0”	R/W											
b3	FPRCB3	FPRC位3	保护寄存器位，保护对象参考表7-9 0: 写保护 1: 写使能	R/W											
b2	FPRCB2	FPRC位2	预留，读出时为“0”，写入时写“0”	R/W											
b1	FPRCB1	FPRC位1	保护寄存器位，保护对象参考表7-9 0: 写保护 1: 写使能	R/W											
b0	FPRCB0	FPRC位0	保护寄存器位，保护对象参考表7-9 0: 写保护 1: 写使能	R/W											

### 5.7.18 STOP 模式控制寄存器(PWC\_STPMCR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
STOP	EXBUSOE	-	-	-	-	-	-	-	-	-	-	-	-	CKSMRC	FLNWT

位	标记	位名	功能	读写
b15	STOP	STOP模式选择位	0: STOP模式无效 1: STOP模式有效	R/W
b14	EXBUSOE	EXMC地址总线和输出控制信号的输出使能	掉电模式和停止模式下外部总线的地址总线信号和输出控制 0: 高阻 1: 保持掉电模式和停止模式前的状态	R/W
b13-b2	Reserved	-	读出时为“0”，写入时写“0”	R/W
b1	CKSMRC	时钟切换到MRC选项	0: 维持进入STOP模式之前的系统时钟及分频 1: STOP模式唤醒时系统时钟切换到MRC、SCKCFGR寄存器被初始化	R/W
b0	FLNWT	FLASH稳定等待控制	0: STOP模式唤醒时等待FLASH稳定 1: STOP模式唤醒时不等待FLASH稳定	R/W

### 5.7.19 RAM 功耗控制寄存器 0(PWC\_RAMPC0)

复位值: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	RAMPDC10	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	RAMPDC0

位	标记	位名	功能	读写
b31-b11	Reserved	-	读出时为“0”，写入时写“0”	R/W
b10	RAMPDC10	RAM掉电控制位10	0: 0x1FFF8000~0x1FFFFFF空间RAM动作 1: 0x1FFF8000~0x1FFFFFF空间RAM掉电	R/W
b9~b1	Reserved	-	读出时为“0”，写入时写“0”	R/W
b0	RAMPDC0	RAM掉电控制位0	0: 0x20000000~0x20007FFF空间RAM动作 1: 0x20000000~0x20007FFF空间RAM掉电	R/W

### 5.7.20 外设 RAM 低功耗控制寄存器 (PWC\_PRAMLPC)

复位值: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	PRAMPDC3	PRAMPDC2	PRAMPDC1	PRAMPDC0

位	标记	位名	功能	读写
b31~b4	Reserved	-	读出时为“0”，写入时写“0”	R/W
b3	PRAMPDC3	外设RAM掉电控制位3	USBFS RAM掉电控制 0: 不掉电 1: 掉电	R/W
b2	PRAMPDC2	外设RAM掉电控制位2	CACHE_RAM掉电控制 0: 不掉电 1: 掉电	R/W
b1	PRAMPDC1	外设RAM掉电控制位1	CAN_2 RAM/CAN_3 RAM掉电控制 0: 不掉电 1: 掉电	R/W
b0	PRAMPDC0	外设RAM掉电控制位0	CAN_1 RAM掉电控制 0: 不掉电 1: 掉电	R/W

### 5.7.21 PVD 控制寄存器 0(PWC\_PVDCR0)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	PVD2EN	PVD1EN	-	-	-	-	EXVCCINEN
位							
标记							
位名							
功能							
读写							
b7	Reserved	-			读出时为“0”，写入时写“0”		
b6	PVD2EN	电压检测2允许			0: 电压检测2电路无效 1: 电压检测2电路有效		
b5	PVD1EN	电压检测1允许			0: 电压检测1电路无效 1: 电压检测1电路有效		
b4-b1	Reserved	-			读出时为“0”，写入时写“0”		
b0	EXVCCINEN	外部VCC电压输入使能			0: 外部VCC电压输入无效 1: 外部VCC电压输入有效		

## 5.7.22 PVD 控制寄存器 1(PWC\_PVDCR1)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	PVD2CMPOE	PVD2IRS	PVD2IRE	-	PVD1CMPOE	PVD1IRS	PVD1IRE
<hr/>							
位	标记	位名	功能			读写	
b7	Reserved	-	读出时为“0”，写入时写“0”			R/W	
b6	PVD2CMPOE	PVD2比较结果输出使能	0: 禁止输出比较器2的比较结果 1: 允许输出比较器2的比较结果			R/W	
b5	PVD2IRS	PVD2中断复位选择	0: 在VCC变化满足检测条件时产生PVD2中断 1: 在下降过程中经过VPVD2产生PVD2复位  注：当PVD1IRS位为“1”或者PVD2IRS位为“1”时，不能进入掉电模式，要进入PD模式时，必须将PVD1IRS位置“0”并且将PVD2IRS位置“0”			R/W	
b4	PVD2IRE	PVD2中断复位使能	0: 禁止 1: 允许  注：请在PVD2EN位为“1”且PVD2CMPOE位为“1”状态下将PVD2IRE位写“1”			R/W	
b3	Reserved	-	读出时为“0”，写入时写“0”			R/W	
b2	PVD1CMPOE	PVD1比较结果输出使能	0: 禁止输出比较器1的比较结果 1: 允许输出比较器1的比较结果			R/W	
b1	PVD1IRS	PVD1中断复位选择	0: 在VCC变化满足检测条件时产生PVD1中断 1: 在下降过程中经过VPVD1产生PVD1复位  注1：当PVD1IRS位为“1”或者PVD2IRS位为“1”时，不能进入掉电模式，要进入PD模式时，必须将PVD1IRS位置“0”并且将PVD2IRS位置“0”			R/W	
b0	PVD1IRE	PVD1中断复位使能	0: 禁止 1: 允许  注：请在PVD1EN位为“1”且PVD1CMPOE位为“1”状态下将PVD1IRE位写“1”			R/W	

### 5.7.23 PVD 滤波控制寄存器(PWC\_PVDFCR)

复位值：0x11

b7	b6	b5	b4	b3	b2	b1	b0
-	PVD2NFCKS[1:0]		PVD2NFDIS	-	PVD1NFCKS[1:0]	PVD1NFDIS	
<hr/>							
位	标记	位名	功能		读写		
b7	Reserved	-	读出时为“0”，写入时写“0”		R/W		
b6~b5	PVD2NFCKS	PVD2数字滤波采样时钟选择	00: 2个滤波时钟周期 01: 4个滤波时钟周期 10: 8个滤波时钟周期 11: 16个滤波时钟周期 滤波时钟周期=LRC周期/8		R/W		
			注：只能在PVD2DNFE位为“1”时改写该位				
b4	PVD2NFDIS	PVD2数字滤波器屏蔽	0: 数字滤波器有效 1: 数字滤波器无效		R/W		
b3	Reserved	-	读出时为“0”，写入时写“0”		R/W		
b2~b1	PVD1NFCKS	PVD1数字滤波采样时钟选择	00: 2个滤波时钟周期 01: 4个滤波时钟周期 10: 8个滤波时钟周期 11: 16个滤波时钟周期 滤波时钟周期=LRC周期/8		R/W		
			注：只能在PVD1DNFE位为“1”时改写该位				
b0	PVD1NFDIS	PVD1数字滤波器屏蔽	0: 数字滤波器有效 1: 数字滤波器无效		R/W		

## 5.7.24 PVD 电平控制寄存器(PWC\_PVDLCR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	PVD2LVL[2:0]			-	PVD1LVL[2:0]		
<hr/>							
位	标记	位名	功能			读写	
b7	Reserved	-	读出时为“0”，写入时写“0”			R/W	
			000: 2.1V 001: 2.3V 010: 2.5V 011: 2.6V				
b6~b4	PVD2LVL[2:0]	PVD2阈值电压选择	100: 2.7V 101: 2.8V 110: 2.9V 111: 1.1V(仅在PWC_PVDCR0.EXVCCINEN=1时有效，其它情况请不要设定该值)			R/W	
b3	Reserved	-	读出时为“0”，写入时写“0”			R/W	
			000: 2.0V 001: 2.1V 010: 2.3V 011: 2.5V 100: 2.6V 101: 2.7V 110: 2.8V 111: 2.9V			R/W	
b2~b0	PVD1LVL[2:0]	PVD1阈值电压选择					

## 5.7.25 PVD 中断控制寄存器(PWC\_PVDICR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	PVD2EDGS[1:0]	PVD2NMIS	-	PVD1EDGS[1:0]	PVD1NMIS		
<hr/>							
位	标记	位名	功能			读写	
b7	Reserved	-	读出时为“0”，写入时写“0”			R/W	
b6~b5		PVD2EDGS[1:0]	PVD2检测条件选择			00: 当检测到VCC<VPVD2(下降)时 01: 当检测到VCC≥VPVD2(上升)时 10: 当检测到VCC<VPVD2(下降)时或者当检测 到VCC≥VPVD2(上升)时 11: 请勿设定	
b4	PVD2NMIS	PVD2中断类型选择	0: PVD2中断作为非可屏蔽中断 1: PVD2中断作为可屏蔽中断			R/W	
b3	Reserved	-	读出时为“0”，写入时写“0”			R/W	
b2~b1		PVD1EDGS[1:0]	PVD1检测条件选择			00: 当检测到VCC<VPVD1(下降)时 01: 当检测到VCC≥VPVD1(上升)时 10: 当检测到VCC<VPVD1(下降)时或者当检测 到VCC≥VPVD1(上升)时 11: 请勿设定	
b0	PVD1NMIS	PVD1中断类型选择	0: PVD1中断作为非可屏蔽中断 1: PVD1中断作为可屏蔽中断			R/W	

### 5.7.26 PVD 检测状态寄存器(PWC\_PVDDSR)

复位值：0x11

b7	b6	b5	b4	b3	b2	b1	b0
-	-	PVD2DETFLG	PVD2MON	-	-	PVD1DETFLG	PVD1MON
<hr/>							
位	标记	位名	功能			读写	
b7~b6	Reserved	-	读出时为“0”，写入时写“0”			R/W	
b5	PVD2DETFLG	PVD2检测标志位	0: PVD2未检测到VCC经过VPVD2 1: PVD2检测到VCC经过VPVD2 读出后写0能够清除本位。注：当PVD2EN位为“1”且 PVD2CMPOE位为“1”时，此标志位有效			R/W	
b4	PVD2MON	PVD2监视位	0: VCC < =VPVD2或者外部输入比较电压<=PVD2 内部基准电压 1: PVD2无效时或者VCC>VPVD2 或者外部输入比较 电压>PVD2内部基准电压			R	
b3~b2	Reserved	-	读出时为“0”，写入时写“0”			R/W	
b1	PVD1DETFLG	PVD1检测标志位	0: PVD1未检测到VCC经过VPVD1 1: PVD1检测到VCC经过VPVD1 读出后写0能够清除本位。 注：当PVD1EN位为“1”且PVD1CMPOE位为“1”时， 此标志位有效			R/W	
b0	PVD1MON	PVD1监视位	0: VCC < VPVD1 1: PVD1无效时或者VCC>VPVD1			R	

### 5.7.27 唤醒计时器控制寄存器(PWC\_WKTCR)

复位值：0x0000

b15	b14-b13	b12	b11-b0
WKTCE	WKCKS[1:0]	WKOVF	WKTMCMP[11:0]
<hr/>			
位	标记	位名	功能
b15	WKTCE	WKTMCMP[11:0]	0: WKTM停止 1: WKTM计数 注：WKTCE位会将计数器清零，计数器停止动作。
b14-b13	WKCKS[1:0]	WKTMCMP[11:0]	00: 64Hz时钟 01: XTAL32 10: LRC 11: 预留
b12	WKOVF	WKTMCMP[11:0]	0: 计数器与WKTMCMP值不一致 1: 计数器与WKTMCMP值一致 写0清除本标志位。
b11-b0	WKTMCMP[11:0]	WKTMCMP[11:0]	WKTMCMP[11:0]

## 6 初始化配置 (ICG)

### 6.1 简介

芯片复位解除后，硬件电路会读取 Flash 地址 0x00000400~0x0000045F 把数据加载到初始化配置寄存器。地址 0x0000\_0408~0x0000\_040B、0x0000\_0414~0x0000\_041F、0x00000438~0x0000045F 为预约功能，请写入全 1 保证芯片正常动作。Flash 引导交换有效，且 OTP 不使能时，该区域位于 Flash 块 1 扇区 0；否则该区域存在 Flash 块 0 扇区 0。用户可通过编程或擦除扇区 0 来修改初始化配置(Initial Config)寄存器。地址 0x0000\_0420~0x0000\_0437 为数据安全保护使能区，规格详见【数据安全保护】。寄存器复位值由 Flash 数据确定。

初始化配置寄存器地址一览表如下：

表 6-1 寄存器一览表

ICG\_BASE\_ADDR:0x00000400

寄存器名	符号	偏移地址	位宽	复位值
初始化配置寄存器0	ICG0	0x000	32	不定
初始化配置寄存器1	ICG1	0x004	32	不定
初始化配置寄存器3	ICG3	0x00C	32	不定
初始化配置寄存器4	ICG4	0x010	32	不定

## 6.2 寄存器说明

### 6.2.1 初始化配置寄存器 0(ICG0)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	WDT SLPO FF	WDTWDPT[3:0]				WDTCKS[3:0]				WDTPERI[1: 0]	WDTIT S	WDTA UTS	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	SWD TSLP OFF	SWDTWDPT[3:0]				SWDTCKS[3:0]				SWDTPERI[1 :0]	SWDTI TS	SWDT AUTS	

位	标记	位名	功能	读写
b31~b29	Reserved	-	功能预留位	R
b28	WDTSLPOFF	WDT在sleep模式下计数停止	0: WDT在sleep模式下计数不停止 1: WDT在sleep模式下计数停止	R
b27~b24	WDTWDPT[3:0]	刷新允许区域计数值百分比	WDT计数值刷新允许区间 0000: 0%~100% 0001: 0%~25% 0010: 25%~50% 0011: 0%~50% 0100: 50%~75% 0101: 0%~25%, 50%~75% 0110: 25%~75% 0111: 0%~75% 1000: 75%~100% 1001: 0%~25%, 75%~100% 1010: 25%~50%, 75%~100% 1011: 0%~50%, 75%~100% 1100: 50%~100% 1101: 0%~25%, 50%~100% 1110: 25%~100% 1111: 0%~100%	R
b23~b20	WDTCKS[3:0]	WDT计数时钟	0010: PCLK3/4 0110: PCLK3/64 0111: PCLK3/128 1000: PCLK3/256 1001: PCLK3/512 1010: PCLK3/1024 1011: PCLK3/2048 1101: PCLK3/8192 其它值: 预留	R
b19~b18	WDTPERI[1:0]	WDT计数溢出周期	00: 256 周期 01: 4096 周期 10: 16384 周期 11: 65536 周期	R

b17	WDTITS	WDT中断选择	0: 中断请求 1: 复位请求	R
b16	WDTAUTS	WDT自动启动	0: 复位后, WDT自动启动 (硬件启动) 1: 复位后, WDT停止状态	R
b15~b13	Reserved	-	功能预留位	R
b12	SWDTSLPOFF	SWDT在Sleep,Stop模式下计数停止	0: SWDT在sleep,stop模式下计数不停止 1: SWDT在sleep,stop模式下计数停止	R
SWDT计数值刷新允许区间				
0000: 0%~100%				
0001: 0%~25%				
0010: 25%~50%				
0011: 0%~50%				
0100: 50%~75%				
0101: 0%~25%,50%~75%				
0110: 25%~75%				
0111: 0%~75%				
1000: 75%~100%				
1001: 0%~25%,75%~100%				
1010: 25%~50%,75%~100%				
1011: 0%~50%,75%~100%				
1100: 50%~100%				
1101: 0%~25%,50%~100%				
1110: 25%~100%				
1111: 0%~100%				
0000: SWDTCLK				
0100: SWDTCLK/16				
0101: SWDTCLK/32				
0110: SWDTCLK/64				
0111: SWDTCLK/128				
1000: SWDTCLK/256				
1011: SWDTCLK/2048				
其它值: 预留				
00: 256 周期				
01: 4096 周期				
10: 16384 周期				
11: 65536 周期				
b1	SWDTITS	SWDT中断选择	0: 中断请求 1: 复位请求	R
b0	WDTAUTS	WDT自动启动	0: 复位后, WDT自动启动 (硬件启动) 1: 复位后, WDT停止状态	R

## 6.2.2 初始化配置寄存器 1(ICG1)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	BOR_DIS	BOR_LEV[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	HRC_STOP	-	-	-	-	-	-	-	HRC_FREQ_SEL
<hr/>															
位	标记	位名	功能	读写											
b31~b19	Reserved	-	功能预留位	R											
b18	BORDIS	BOR动作选择	0: 复位后允许BOR动作 1: 复位后禁止BOR动作	R											
b17~b16	BOR_LEV[1:0]	BOR阈值电压选择	00: 1.9v 01: 2.0v 10: 2.1v 11: 2.3v	R											
b15~b9	Reserved	-	功能预留位	R											
b8	HRCSTOP	HRC振荡停止位	0: HRC振荡 1: HRC停止	R											
b7~b1	Reserved	-	读出时为“1”,写入时写“1”	R											
b0	HRCFREQSEL	HRC频率选择	0: 20MHz 1: 16MHz	R											

### 6.2.3 初始化配置寄存器 3(ICG3)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DBUSPRT[15: 0]															

位	标记	位名	功能	读写
b31~b16	Reserved	-	功能预留位	R
b15~b0	DBUSPRT[15: 0]	对128Kbytes区域D-BUS 读保护功能	当DBUSPRT [15:0]=0x4450时， 对0x0000_0000~0x0001_FFFFh区域的D-BUS读保护使 能；否则D-BUS读保护无效。	R

### 6.2.4 初始化配置寄存器 4(ICG4)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BTENB[31: 0]															

位	标记	位名	功能	读写
系统复位后引导源选择				
b31~b0	BTENB[31: 0]	MD管脚	BTENB[31:0]	引导源选择
			0	*
			1	0x4844_0682
			1	上述以外
				系统存储器

## 7 嵌入式 Flash (EFM)

### 7.1 简介

Flash 接口通过 Flash ICODE、DCODE 和 MCODE 总线对 Flash 进行访问。该接口可对 Flash 执行编程，擦除和全擦除操作；通过指令预取和缓存机制加速代码执行。

### 7.2 主要特性

- 两块独立 Flash 构成 dual bank，可实现 BGO (BackGround Operation) 功能
- 134KBytes 的 OTP 空间
- ICODE 总线 16Bytes 预取指
- 两个独立缓存区：ICODE 总线缓存空间 2KBytes；DCODE 总线缓存空间 256Bytes
- 支持引导交换功能
- 支持数据安全保护

### 7.3 嵌入式 Flash

Flash 具有以下主要特性：

- 容量高达 512KBytes，由两块 256KBytes 的 Flash 构成，共 64 个扇区，每个扇区为 8Kbytes。块 0 中扇区 0~扇区 15 为可配置为 OTP 区域。
- OTP(One Time Program) 区域共 134KBytes，其中 128Kbytes 配置在块 0 地址 0x0000\_0000~0x0001\_FFFF，6KBytes 配置在地址 0x0300\_0000~0x0300\_17FF。地址 0x0300\_1800~0x0300\_1AD7 为 OTP 数据锁存区。
- 128 位宽数据读取，读缓存 128 位宽缓冲，加速代码执行。
- 编程单位为 4Bytes，擦除单位为 8KBytes。

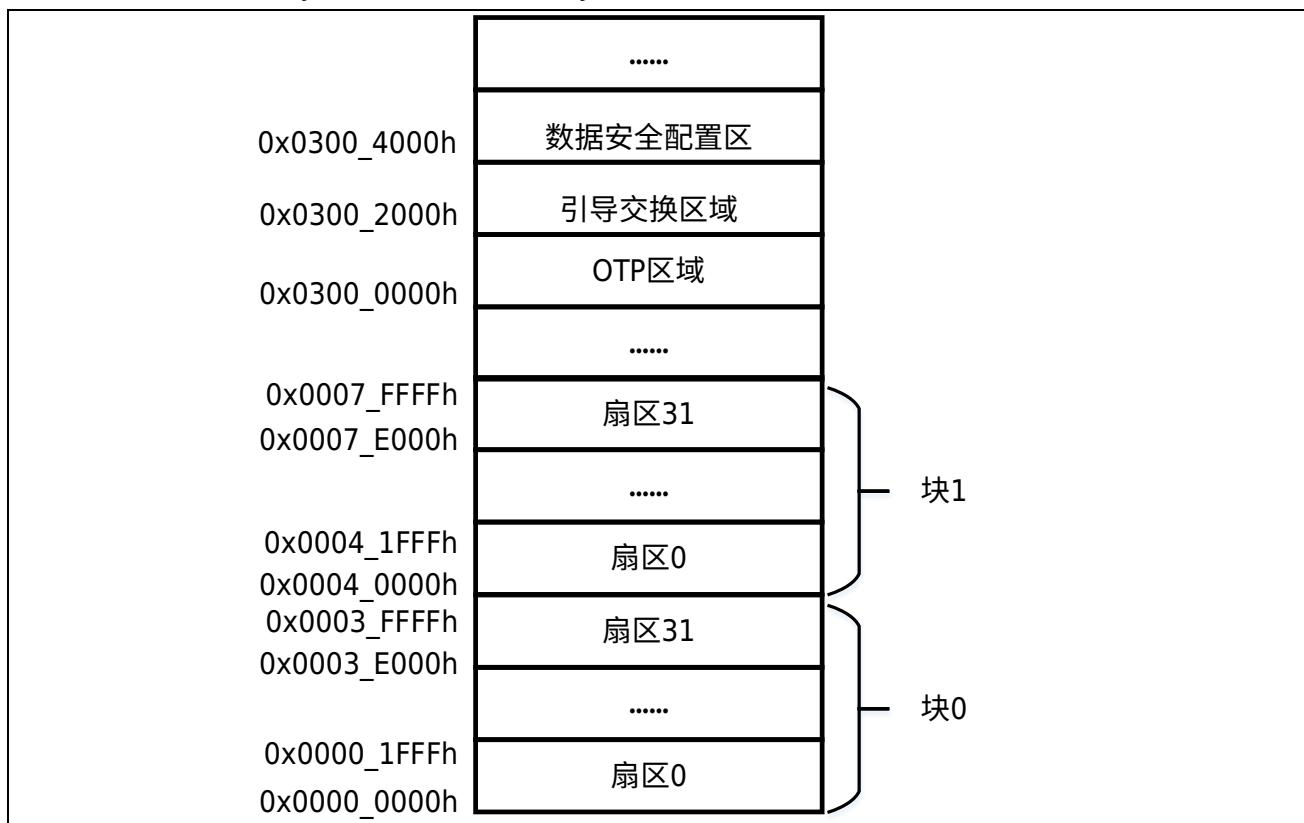


图 7-1 512KBytes 产品 Flash 地址分布

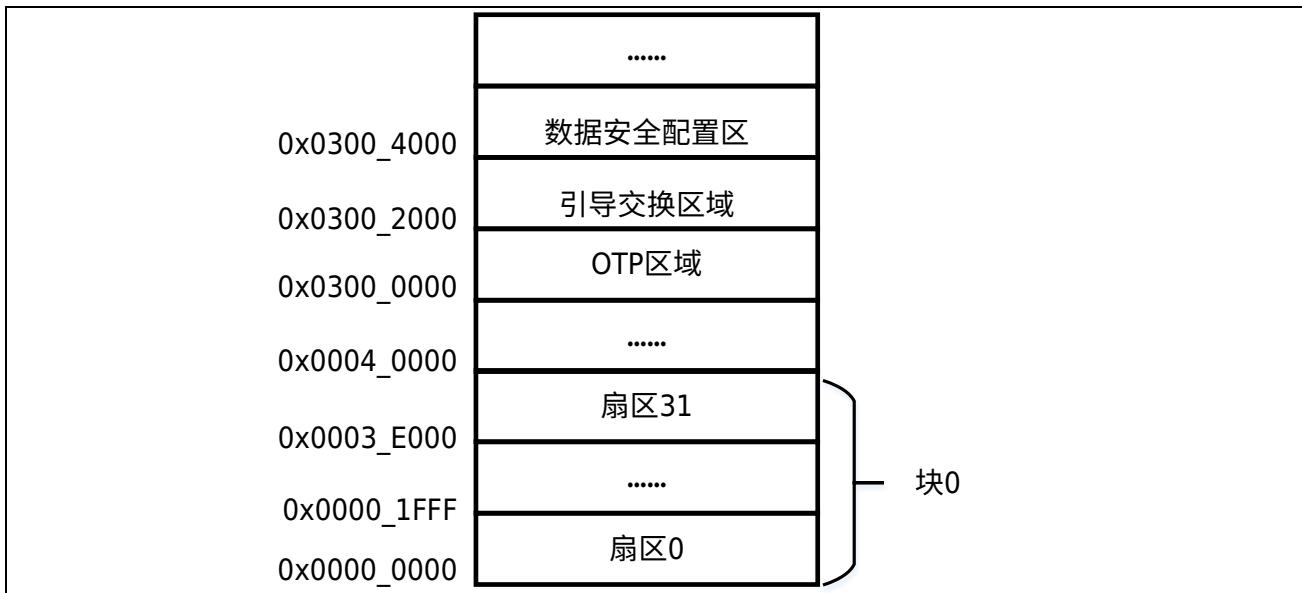


图 7-2 256KBytes 产品单 Flash 地址分布

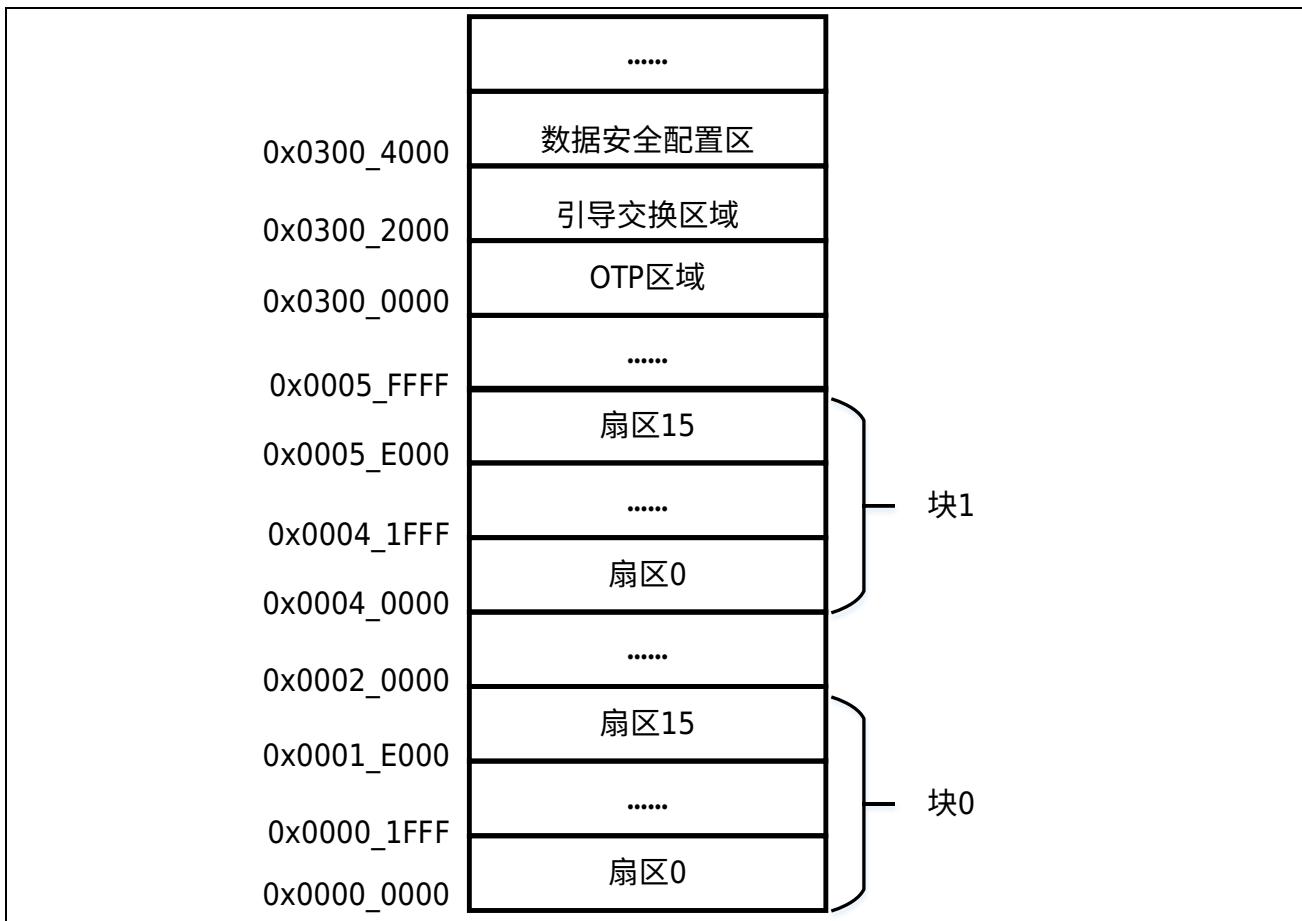


图 7-3 256KBytes 产品双 Flash 地址分布

OTP 区域 0x0300\_0000~0x0300\_1ADB，引导交换使能地址 0x0300\_2000~0x0300\_2003 和数据安全配置区 0x0300\_4000~0x0300\_400B 分别单独享有 1 个扇区，可进行编程，擦除，但全擦除禁止；

这三个扇区物理上从属于 FLASH 块 0，擦写标志位跟随 FLASH 块 0，但对 FLASH 块 0 全擦除这三个扇区数据不变。

## 7.4 读接口

### 7.4.1 CPU 时钟和 Flash 读取时间之间的关系

要正确读取 Flash 数据，用户需要根据 CPU 动作频率在 Flash 读模式寄存器(EFM\_FRMC)中正确设定等待周期数(FLWT[3:0])。

系统复位后，CPU 时钟源为 MRC(8MHz)，Flash 读等待周期为 0。建议用户按照以下步骤修改 CPU 主频和 Flash 读等待周期位。等待周期数请参照表 7-1。

**CPU 频率提高步骤：**

1. 将新的读等待周期设定值 (FLWT[3:0]) 写入寄存器 EFM\_FRMC。
2. 读取寄存器 EFM\_FRMC, 检查新的等待周期是否设定成功。
3. 通过设定系统时钟源切换寄存器 CMU\_CKSWR(CKSW[2:0]) 或者系统时钟配置寄存器 CMU\_SCFGR(HCLKS[2:0]) 来提升 CPU 时钟频率。
4. 读取寄存器 CMU\_CKSWR 或者 CMU\_SCFGR，检查新的设定是否成功。

**CPU 频率降低步骤：**

1. 通过设定系统时钟源切换寄存器 CMU\_CKSWR(CKSW[2:0]) 或者系统时钟配置寄存器 CMU\_SCFGR(HCLKS[2:0]) 来降低 CPU 时钟频率。
2. 读取寄存器 CMU\_CKSWR 或者 CMU\_SCFGR，检查新的设定是否成功。
3. 将新的读等待周期设定值 (FLWT[3:0]) 写入寄存器 EFM\_FRMC。
4. 读取寄存器 EFM\_FRMC，检查新的等待周期是否设定成功。

表 7-1 CPU 时钟频率和 Flash 读等待周期对照表

CPU时钟频率(hclk)	读等待周期设定
80MHz < Fhclk ≤ 120MHz	FLWT[3:0]=4'b0010
40MHz < Fhclk ≤ 80MHz	FLWT[3:0]=4'b0001
Fhclk ≤ 40MHz	FLWT[3:0]=4'b0000

## 7.5 Flash 读加速缓存

每次 Flash 读操作为 128 位读取，数据送给 CPU 同时也存入缓冲存储器，这 128 位数据可以是 4 行 32 位指令，也可以是 8 行 16 位指令，具体取决于烧写在 Flash 中的程序。

为了能快速读取 Flash 数据，Flash 控制器配置了读加速缓存，优化了读取等待周期。为了发挥处理器性能，该加速器将对 Flash 的 ICODE, DCODE 总线访问数据保存到缓存寄存器中，从而提高了程序执行速度。

系统提供 2.25Kbytes 空间做为缓存存储器，可以有效地减少因指令跳转而产生的时问损耗。通过 EFM\_FRMC 寄存器中的缓存使能 (ICACHE/DCACHE) 位置 1，来使缓存功能有效。每当出现指令或数据缺失 (即请求的指令或数据未存在于当前使用的指令行或缓存存储器中) 时，系统会将新读取的数据行(128 位)复制到缓存存储器中。如果 CPU 请求的指令或者数据已存在于缓存中，则无需任何延时即可立即获取。缓存存储器存满后，采用 LRU (最近最少使用) 策略确定缓存存储器中待替换的数据。

CPU 读取指令或数据时，Flash 地址在缓冲，缓存中命中时，读取 Flash 周期数会改变，具体请参考表 7-2。

表 7-2 Flash 实际读周期数

EFM_FRMC. FLWT[3:0] 设定	缓存不使能		缓存使能	
	缓冲命中	缓冲不命中	缓冲或缓存命中	缓冲，缓存都不命中
0	1	1	1	1
1	1	2	1	2
2	1	3	1	3
N(N>2)	1	N+1	1	N+1

## 7.6 Flash 编程和擦除操作

Flash 支持编程，扇区擦除，全擦除操作。

Flash 编程，扇区/全擦除地址末位必须以 4 对齐(末位地址为:0x0,0x4,0x8,0xC)，编程单位是 4Bytes，扇区擦除单位为 8Kbytes，全擦除根据寄存器设定可以是单个 Flash 块也可是两个 Flash 块。Flash 编程方式分为单次编程无回读模式，单次编程回读模式，连续编程模式三种。Flash 编程，擦除期间，设定 EFM\_FWMC.BUSHLDCTL=0，则总线被占有，直至擦写结束；EFM\_FWMC.BUSHLDCTL=1，则总线被释放，总线可以继续访问另一块 Flash 地址。Flash 编程，擦除前，请把缓存使能及预取指无效。以下分别介绍编程和擦除操作的设定步骤。

### 7.6.1 解锁 EFM\_KEY1 寄存器

复位解除后，Flash 编程，擦除模式寄存器 (EFM\_FWMC) 处于写禁止状态，需要先解除 Flash 访问保护寄存器 (EFM\_FAPRT)，然后再解除 EFM\_KEY1 的保护。以下步骤用于解除上述保护。

- 1) 解除 Flash 寄存器访问写保护(EFM\_FAPRT 先写 0x0123，再写 0x3210)
- 2) 解除 EFM\_KEY1 锁定(EFM\_KEY1 先写 0x01234567，再写 0xFEDCBA98)

如对 EFM\_KEY1 写入错误的序列，则会发生引发一次总线错误，并且自锁，EFM\_FWMC 寄存器将写禁止直至发生复位。

对 EFM\_FWMC 寄存器正确写入设定值后，如需回归到 EFM\_FWMC 写禁止状态，可对 EFM\_FWMC.KEY1LOCK 写入 1。

### 7.6.2 写保护功能

Flash 每个扇区有 1 个写保护位，存在于写保护寄存器 EFM\_F0/1NWPRT 中。每个写保护寄存器存在 1 个写保护锁定位(WLOCK1、WLOCK0)，写保护锁定位一旦设定为 1，写保护寄存器位只能从写许可设定为写禁止。当 Flash 扇区设定为写禁止时，对该扇区地址进行编程和擦除操作将不发生，并发生错误标志(EFM\_FSR.PRTWERR0/1=1)。

在全擦除模式，若该 Flash 块存在一个及以上扇区设定为写禁止，则全擦除不发生，并发生错误标志(EFM\_FSR.PRTWERR0/1=1)。

### 7.6.3 单次编程无回读功能

单编程无回读模式设定步骤如下：

- 1) 解除 Flash 的寄存器写保护。(EFM\_FAPRT 先写 0x0123，再写 0x3210)
- 2) 解除 EFM\_KEY1 锁定。(EFM\_KEY1 先写 0x01234567，再写 0xFEDCBA98)
- 3) 设定单次编程模式。(EFM\_FWMC.PEMODE[2:0]=001)
- 4) 解除写保护。(EFM\_F0/1NWPRT 对应位设定为 1)
- 5) 对编程地址写入 32 位数据
- 6) 等待 Flash 处于空闲状态。(EFM\_FSR.RDY0/1=1)
- 7) 读出编程地址值判断是否和写入值一致  
一致，表示编程成功，不一致，表示该 Flash 地址已遭破坏，永久废弃。
- 8) 清除编程结束标志位。(EFM\_FSR.OPTEND0/1)  
对已锁存的 OTP 地址发行单编程不回读写操作，编程不成功，标志位 EFM\_FSR.OTPWERRO 置位。

### 7.6.4 单编程回读功能

单编程回读模式是指编程结束后硬件自动读取编程地址数据并和写入数据对比，判断标志位 EFM\_FSR.MISMTCH0/1 验证写入数据正确。

单编程回读模式设定步骤如下：

- 1) 解除 Flash 的寄存器写保护。(EFM\_FAPRT 先写 0x0123，再写 0x3210)
- 2) 解除 EFM\_KEY1 锁定。(EFM\_KEY1 先写 0x01234567，再写 0xFEDCBA98)
- 3) 设定单次编程回读模式。(EFM\_FWMC.PEMODE[2:0]=010)
- 4) 解除写保护。(EFM\_F0/1NWPRT 对应位设定为 1)
- 5) 对编程地址写入 32 位数据
- 6) 等待 Flash 处于空闲状态。(EFM\_FSR.RDY0/1=1)
- 7) 判断编程自读取结果标志位。(EFM\_FSR.MISMTCH0/1)

如为 0，则表示编程成功；为 1 表示该 Flash 地址已遭破坏，永久废弃。

- 8) 清除编程结束标志位。(EFM\_FSR.OPTEND0/1)

对已锁存的 OTP 地址发行单编程回读写操作，编程不成功，标志位 EFM\_FSR.OTPWERRO 置位。

### 7.6.5 连续编程功能

当连续对 Flash 地址进行编程时，推荐使用连续编程模式。连续编程模式比单编程模式可以节省时间 50%以上。连续编程模式时，对 Flash 地址写命令间隔不能超过 16us。连续编程操作设定步骤如下：

- 1) 解除 Flash 的寄存器写保护。(EFM\_FAPRT 先写 0x0123，再写 0x3210)
- 2) 解除 EFM\_KEY1 锁定。(EFM\_KEY1 先写 0x01234567，再写 0xFEDCBA98)
- 3) 设定连续编程模式。(EFM\_FWMC.PEMOD[2:0]=011)
- 4) 解除写保护。(EFM\_F0/1NWPRT 对应位设定为 1)
- 5) 对编程地址写 32 位数据。(编程地址需要和执行程序分属不同 Flash 块或者程序在 Flash 以外区域运行)
- 6) 等待操作结束标志位(EFM\_FSR.OPTEND0/1)置位。
- 7) 清除操作结束标志位，直至读到 EFM\_FSR.OPTEND0/1 为 0。
- 8) 重复 5), 6), 7)直至所有数据写完。
- 9) 修改连续编程模式为非连续编程模式。(EFM\_FWMC.PEMOD[2:0]=000)
- 10) 等待 Flash 处于空闲状态。(EFM\_FSR.RDY0/1=1)
- 11) 读取编程地址并判断是否和写入值一致。
- 12) 一致，表示编程成功，不一致，表示该 Flash 地址已遭破坏，永久废弃。

对已锁存的 OTP 地址发送连续编程写操作，编程不成功，标志位 EFM\_FSR.OTPWERRO 置位。

### 7.6.6 擦除功能

对 Flash 进行扇区擦除操作后，该扇区内地址(8KBytes 空间)数据刷新为全 1。扇区擦除操作设定步骤如下：

- 1) 解除 Flash 的寄存器写保护(EFM\_FAPRT 先写 0x0123，再写 0x3210)。
- 2) 解除 EFM\_KEY1 锁定。(EFM\_KEY1 先写 0x01234567，再写 0xFEDCBA98)
- 3) 设定擦除模式(EFM\_FWMC.PEMOD[2:0]=100)。
- 4) 解除写保护。(EFM\_F0/1NWPRT 对应位设定为 1)
- 5) 对需要擦除扇区内的任意地址(地址需以 4 对齐)写入 32 位任意值。
- 6) 等待 Flash 处于空闲状态。(EFM\_FSR.RDY0/1=1)
- 7) 清除擦除结束标志位。(EFM\_FSR.OPTEND0/1)

对已锁存的 OTP 地址发行擦除操作，擦除不成功，OTP 区域数据保留，标志位

EFM\_FSR.OTPWERRO 置位。

### 7.6.7 全擦除功能

EFM 提供了单块 Flash 全擦除和两块 Flash 同时全擦除两种擦除方式。全擦除操作设定步骤如下：

- 1) 解除 Flash 的寄存器写保护(EM\_FAPRT 先写 0x0123，再写 0x3210)。
- 2) 解除 EFM\_KEY1 锁定。(EM\_KEY1 先写 0x01234567，再写 0xFEDCBA98)
- 3) 单块 Flash 全擦除时，设定 EM\_FWMC.PEMOD[2:0]=101。
- 4) 两块 Flash 同时全擦除时，设定 EM\_FWMC.PEMOD[2:0]=110。
- 5) 解除写保护。(EM\_F0/1NWPRT 所有位都设定为 1)
- 6) 单块 Flash 全擦除：对需要擦除 Flash 的任意地址(地址需以 4 对齐)写入 32 位任意值。
- 7) 两块 Flash 全擦除：对 Flash0 或 Flash1 地址(地址需以 4 对齐)写入 32 位任意值。全擦动作硬件会自动同步到另一块 Flash。
- 8) 等待 Flash 处于空闲状态。(EM\_FSR.RDY0/1=1)
- 9) 清除擦除结束标志位。(EM\_FSR.OPTEND0/1)

OTP 使能后，对 Flash 块 0 地址发行单块全擦除写操作，全擦除不成功，Flash0 数据保留，标志位 EM\_FSR.OTPWERRO 置位；对 Flash 块 0 地址发行两块全擦除写操作，Flash0 数据保留，Flash1 数据被全擦，标志位 EM\_FSR.OTPWERRO 置位。

### 7.6.8 数据安全保护

本产品对 FLASH 数据提供 4 个保护等级，以防不受信任的用户通过调试接口（JTAG 和 SWD 接口），ISP 接口(In System Program)和测试接口读取和篡改 FLASH。

保护级别 0：无保护

调试接口，ISP 接口和测试接口可以访问(读和改写)MCU 资源，包括 FLASH 数据。

保护级别 1：

FLASH 地址 0x0000\_0430~0x0000\_0433 编程写入数据 0xAF180402，同时地址 0x0300\_4000~0x0300\_400B 编程写入 96 位密码后，保护级别 1 使能。

保护级别 1 使能后

- 调试接口，ISP 接口和测试接口无法访问 MCU 资源。
- 用户程序无法对扇区 0 进行编程和扇区擦除。
- 地址 0x0300\_4000~0x0300\_400B 的数据无法读出。

激活保护级别 1 后，可通过密码认证方式和全擦除方式复归到保护级别 0。若 OTP 使能，则全擦除无效。密码认证方式和全擦除方式请咨询销售窗口。

保护级别 2:

FLASH 地址 0x0000\_0434-0x0000\_0437 编程写入数据 0xA85173AE，保护级别 2 使能。

保护级别 2 使能后

- 调试接口，ISP 接口和测试接口无法访问 MCU 资源。
- 用户程序无法对扇区 0 进行编程和扇区擦除。

激活保护级别 2 后，可通过全擦除方式复归到保护级别 0。若 OTP 使能，则全擦除无效。

保护级别 3:

FLASH 地址 0x0000\_0420-0x0000\_0423, 0x0000\_0424-0x0000\_0427, 0x0000\_0428-0x0000\_042B 编程写入数据都为 0x42545048 时，保护级别 3 使能。

保护级别 3 使能后

- 调试接口，ISP 接口和测试接口无法改写 FLASH 数据。
- 用户程序无法对扇区 0 进行编程和扇区擦除。

激活保护级别 3 后，可通过全擦除方式复归到保护级别 0。若 OTP 使能，则全擦除无效。

保护级别 1，保护级别 2 和保护级别 3 可单独使能，也可同时使能。同时使能时，保护措施叠加生效。

### 7.6.9 D-BUS 读保护功能

对地址 0x00000000~0x0001FFFF(128Kbytes)空间提供了 D-BUS 读保护。使能 D-BUS 保护，需设定 ICG3 寄存器 DBUSPRT [15:0] 为 0x4450，复位重启后，功能使能。CPU-PC 指针在此 128K 空间时（即 CPU 在此区域执行程序），CPU 和 DMAC 可以正常访问；当 CPU-PC 指针在此 128K 空间以外时（即 CPU 在 128K 以外区域执行程序），CPU 和 DMAC 对 128K 空间内进行 D-BUS 读则被禁止，并返回总线错误。D-BUS 是 CPU 的数据访问总线，包括操作数访问，堆栈数据访问，一般数据访问等。D-BUS 读保护使能时，用户需要注意以下事项。

- 1) 单个独立程序空间不能跨越 0x00020000 的边界。
- 2) CPU 在执行 128K 以外区域程序时，不能读取配置在 128K 空间上的中断向量表。
- 3) CPU 在执行 128K 以外区域程序时，只有通过跳转指令和子程序调用可以回到 128K 区域执行程序。

### 7.6.10 BGO 功能

Flash 编程，擦除目标地址和执行程序地址不在同一个 Flash 块，或者程序在 RAM 上执行，可设定 EFM\_FWMC.BUSHLDCTL 为 1（编程，擦除期间，总线释放），实现程序在执行同时 Flash 在进行编程或擦除。以下为 BGO 功能设定步骤。

- 1) 解除 Flash 的寄存器写保护。(EFM\_FAPRT 先写 0x0123，再写 0x3210)
- 2) 解除 EFM\_KEY1 锁定。(EFM\_KEY1 先写 0x01234567，再写 0xFEDCBA98)
- 3) 设定编程，擦除模式。(EFM\_FWMC.PEMODE[2:0]=001, 010, 011, 100, 101)
- 4) 解除写保护。(EFM\_F0/1NWPRT 对应位设定为 1)
- 5) 设定编程，擦除期间总线释放 (EFM\_FWMC.BUSHLDCTL=1)
- 6) 对编程地址写入 32 位数据。(编程地址和执行程序分属不同块 Flash)  
若执行程序在 RAM 上，可以对两块 Flash 同时编程，(全)擦除。
- 7) 等待擦写结束标志位置位。(EFM\_FSR.OPTEND0/1=1)

## 7.6.11 中断

EFM 模块共有 3 个中断，分别是 PE（编程/擦除）错误中断，总线冲突中断和操作结束中断，可通过读取 EFM\_FSR 寄存器查询状态标志位。

### 1. PE 错误中断 EFM\_PEERR:

**置位：**

- 对已锁存的 OTP 地址发出编程，擦除，全擦除操作(OTPWERRO=1)。
- OTP 功能启用后，对 0x0300\_0000~0x0300\_1ADB 空间进行擦除，全擦除操作 (OTPWERRO=1)。
- 对写保护扇区发出编程，擦除，全擦除操作(PRTWERRO/1=1)。
- 编程地址非以 4 对齐或者数据大小非 32 位(PGSZERRO/1=1)。
- 单编程回读模式时，编程地址硬件自动读取值与写入值不一致(MISMTCH0/1=1)。

**清零：**

寄存器 EFM\_FSCLR 对应标志清除位写 1，状态位清零。

### 2. Flash 读写冲突中断 EFM\_COLERR:

**置位：**

- Flash 忙期间(RDY0=0 或 RDY1=0)对该块 Flash 读写操作。(连续编程模式，对 Flash 连续写除外)

**清零：**

寄存器 EFM\_FSCLR 对应清除位置 1，状态位清零。

### 3. 操作结束中断 EFM\_OPTEND:

**置位：**

- 编程模式：单个地址编程结束(OPTEND0/1=1)。
- 擦除模式：扇区擦除，全擦除结束(OPTEND0/1=1)。

**清零：**

寄存器 EFM\_FSCLR 对应清除位置 1，状态位清零。

## 7.7 一次性可编程字节 (OTP)

本 MCU 提供最大 134KBytes 的 OTP 领域，分为 16 个 8KBytes，2 个 2KBytes，4 个 256Byte，32 个 16Bytes，128 个 4Bytes。地址分布如下。

表 7-3 OTP 地址分布表

sector	OTP块数据地址	容量	OTP块锁存地址
0	0x0000_0000~0x0000_1FFF	8KB	0x0300_1800~0x0300_1803
1	0x0000_2000~0x0000_3FFF	8KB	0x0300_1804~0x0300_1807
.	.	.	.
14	0x0001_C000~0x0001_DFFF	8KB	0x0300_1838~0x0300_183B
15	0x0001_E000~0x0001_FFFF	8KB	0x0300_183C~0x0300_183F
16	0x0300_0000~0x0300_07FF	2KB	0x0300_1840~0x0300_1843
17	0x0300_0800~0x0300_0FFF	2KB	0x0300_1844~0x0300_1847
18	0x0300_1000~0x0300_10FF	256B	0x0300_1848~0x0300_184B
19	0x0300_1100~0x0300_11FF	256B	0x0300_184C~0x0300_184F
20	0x0300_1200~0x0300_12FF	256B	0x0300_1850~0x0300_1853
21	0x0300_1300~0x0300_13FF	256B	0x0300_1854~0x0300_1857
22	0x0300_1400~0x0300_140F	16B	0x0300_1858~0x0300_185B
.	.	.	.
53	0x0300_15F0~0x0300_15FF	16B	0x0300_18D4~0x0300_18D7
54	0x0300_1600~0x0300_1603	4B	0x0300_18D8~0x0300_18DB
.	.	.	.
181	0x0300_17FC~0x0300_17FF	4B	0x0300_1AD4~0x0300_1AD7

启用 OTP 功能，需要预先对地址 0x0300\_1AD8~0x0300\_1ADB 编程写入 32 位包含 0 数据（推荐写入 32 位全 0）。当 OTP 块锁存地址数据包含 0 时，则锁存地址对应的块空间无法再次编程，擦除和全擦除。OTP 的 sector0~15 与 Flash 块 0 地址 0x0000\_0000~0x0001\_FFFF 共享 128KBytes 物理空间；其空间的锁存地址数据若为全 1，此区域为普通 Flash 空间，可多次编程和擦除；若对应的锁存地址数据包含 0，则对应的扇区为只读，Flash 块 0 将无法进行全擦除。对 OTP 锁存地址编程需要解锁 EFM\_KEY2 寄存器(EFM\_KEY2 先写 0x10325476，再写 0xEFCDAB89)。如对 EFM\_KEY2 寄存器写入错误的序列，则会发生引发一次总线错误，并且 EFM\_KEY2 寄存器会锁定直至发生复位。

对 OTP 区域操作请按以下步骤进行：

- 1) 解除 Flash 的寄存器写保护。(EFM\_FAPRT 先写 0x0123，再写 0x3210)。
- 2) 解除 EFM\_KEY1 锁定。(EFM\_KEY1 先写 0x01234567，再写 0xFEDCBA98)。
- 3) 设定编程模式。(EFM\_FWMC.PEMODE[2:0]=001, 010, 011)。

- 4) 根据需要对 OTP 地址写入数据。
- 5) 解除 EFM\_KEY2 锁定。(EFM\_KEY2 先写 0x10325476，再写 0xEFCDAB89)。
- 6) 对 OTP 使能地址 0x0300\_1AD8~0x0300\_1ADB 的 32bit 任意 bit 写 0。(推荐写入 32bit 全 0)
- 7) 对 OTP 锁存地址任意 bit 写入 0。(推荐写入 32bit 全 0)

芯片复位启动后，硬件会自动加载 OTP 锁存地址数据到电路中，因此，若 OTP 使能地址 0x0300\_1AD8~0x0300\_1ADB 已编程写入过 0，则需省略步骤 6)。

对已经锁存的 OTP 数据区域地址进行编程，擦除和全擦除，将会产生 OTPWERR0 错误标志。

OTP 锁存地址写完后，如需回归到 EFM\_KEY2 锁定状态，可对 EFM\_FWMC.KEY2LOCK 写入 1。

## 7.8 引导交换

对 Flash 块 0 地址 (0x0300\_2000) 进行编程或扇区擦除操作，经复位重启后，可实现引导交换。该地址数据为 0x5A5A5A 时，引导交换有效，为其余值时，引导交换无效。引导交换有效后，若 OTP 功能未启用(0x0300\_1AD8~0x0300\_1ADB 全 1)，Flash 块 0 和块 1 全体地址互换；且 OTP 使能地址无法进行编程。若 OTP 功能启用 (0x0300\_1AD8~0x0300\_1ADB 数据非全 1)，Flash 块 0 和块 1 地址部分互换。地址变换规则见下图 7-4，图 7-5。

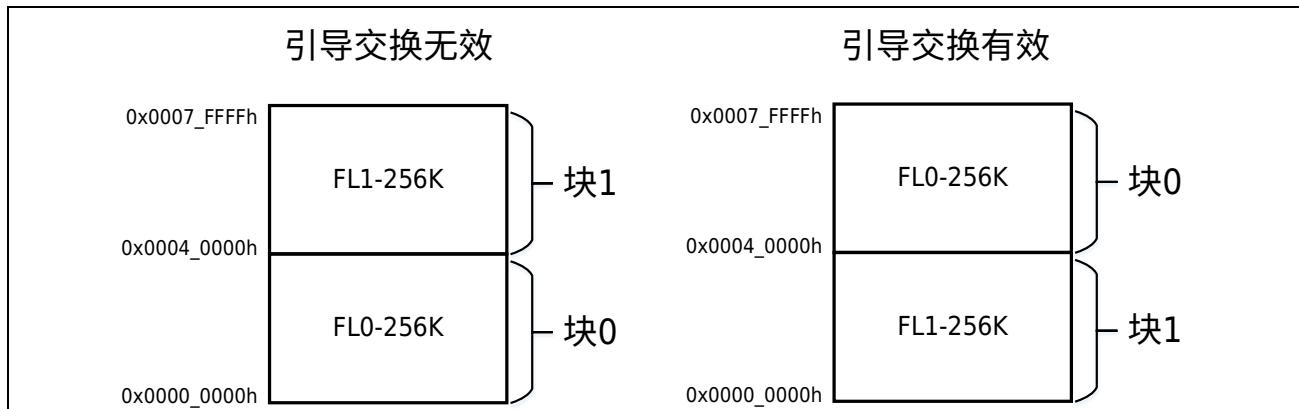


图 7-4 512KBytes 产品 OTP 功能未使能时，Flash 地址分布

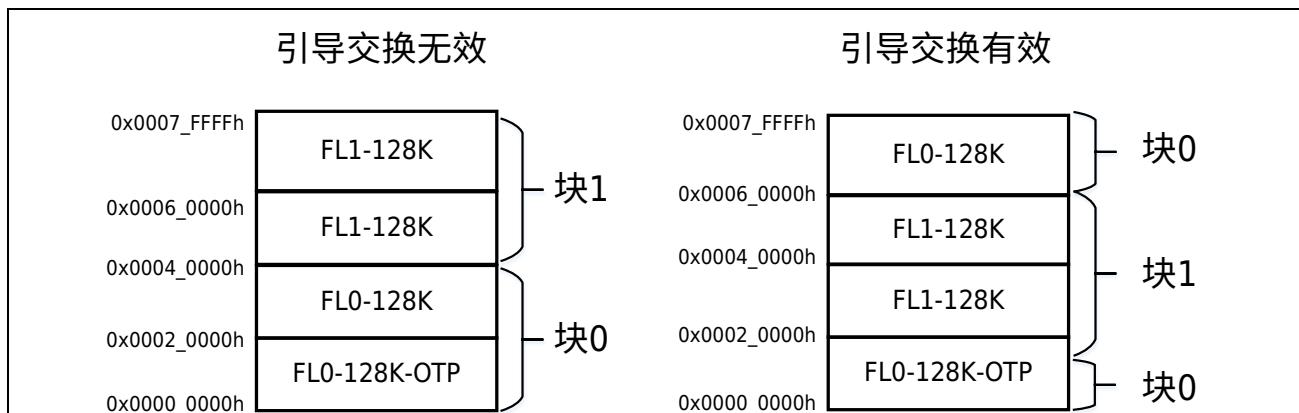


图 7-5 512KBytes 产品 OTP 功能使能时，Flash 地址分布

在 Flash 256KBytes 产品，单 FLASH 和双 FLASH 且 OTP 功能使能时支持引导交换，双 FLASH 且 OTP 功能不使能时，引导交换的地址变换规则见下图 7-6。

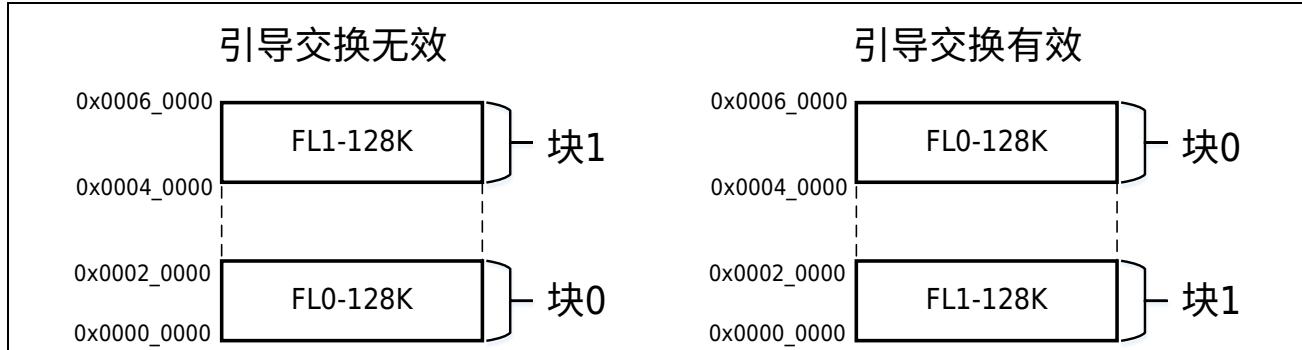


图 7-6 256KBytes 产品 OTP 功能未使能, Flash 地址分配

以下为 512KBytes 产品 OTP 未使能时，引导交换功能的使用示例。

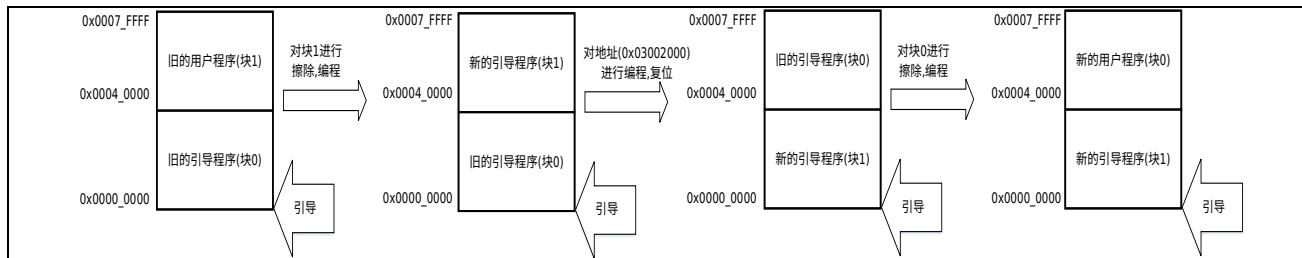


图 7-7 启动引导交换 1

当用户需要再次升级启动程序，由于地址 0x0300\_2000 已经被编程过了，再次升级则需要对地址 0x0300\_2000 进行扇区擦除。操作流程如图 7-8。

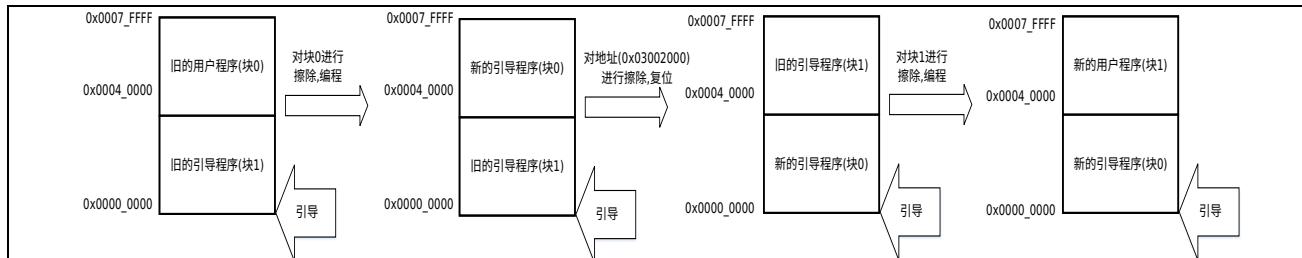


图 7-8 启动引导交换 2

## 7.9 寄存器说明

表 7-4 寄存器一览表

EFM\_BASE\_ADDR: 0x40010400

寄存器说明	寄存器名	偏移量	位宽	复位值
Flash访问写保护寄存器	EFM_FAPRT	0x0000	32	0x00000000
Flash密钥1寄存器	EFM_KEY1	0x0004	32	0x00000000
Flash密钥2寄存器	EFM_KEY2	0x0008	32	0x00000000
Flash停止寄存器	EFM_FSTP	0x0014	32	0x00000000
Flash读模式寄存器	EFM_FRMC	0x0018	32	0x00000000
Flash擦写模式寄存器	EFM_FWMC	0x001C	32	0x00030000
Flash状态寄存器	EFM_FSR	0x0020	32	0x01000100
Flash状态清除寄存器	EFM_FSCLR	0x0024	32	0x00000000
Flash中断许可寄存器	EFM_FITE	0x0028	32	0x00000000
Flash引导交换状态寄存器	EFM_FSWP	0x002C	32	不定
芯片专属标志寄存器	EFM_CHIPID	0x0040	32	0x48440682
UNIQUE ID寄存器0	FEM_UQID0	0x0050	32	不定
UNIQUE ID寄存器1	EFM_UQID1	0x0054	32	不定
UNIQUE ID寄存器2	EFM_UQID2	0x0058	32	不定
Flash写保护锁存寄存器	EFM_WLOCK	0x0180	32	0x00000000
Flash0写保护寄存器	EFM_F0NWPRT	0x0190	32	0x00000000
Flash1写保护寄存器	EFM_F1NWPRT	0x01A0	32	0x00000000

### 7.9.1 Flash 访问写保护寄存器(EFM\_FAPRT)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FAPRT[15:0]															

位	标记	位名	功能	读写
b31-b16	Reserved	-	读出时为“0”，写入时写“0”  EFM寄存器访问保护寄存器。  解除方法：对FAPRT先写“16位数据0x0123”再写“16位数据0x3210”。  在解除保护状态下，写入任意数据，EFM寄存器再次进入保护状态。	R/W
b15-b0	FAPRT[15:0]	EFM寄存器写保护	进入保护状态。  EFM寄存器访问保护有效时，该寄存器读出值为0x00000000。  EFM寄存器访问保护无效时，该寄存器读出值为0x00000001。	R/W

### 7.9.2 Flash 密钥 1 寄存器(EFM\_KEY1)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
KEY1[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
KEY1[15:0]															

位	标记	位名	功能	读写
b31-b0	KEY1[31:0]	密钥1寄存器	EFM_FWMC写保护寄存器。  对EFM_KEY1先写0x01234567，再写0xFEDCBA98，解除对EFM_FWMC的写保护。  该寄存器读出值为0x00000000。	R/W

### 7.9.3 Flash 密钥 2 寄存器(EMF\_KEY2)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
KEY2[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
KEY2[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31-b0	KEY2[31:0]	密钥2寄存器	对OTP锁存地址的写保护寄存器。 对EMF_KEY2先写0x10325476，再写0xEFCDAB89，解除对OTP锁存地址的写保护。 该寄存器读出值为0x00000000。	R/W											

### 7.9.4 Flash 停止寄存器(EMF\_FSTP)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	F1STP	F0STP
<hr/>															
位	标记	位名	功能	读写											
b31-b2	Reserved	-	读出时为“0”，写入时写“0”	R/W											
<hr/>															
b1	F1STP	Flash1停止模式控制	0: Flash1活动状态 1: Flash1处于停止模式 当寄存器位由1设为0后，请在确认FSR.RDY1位为1后，进行Flash1访问。	R/W											
<hr/>															
b0	F0STP	Flash0停止模式控制	0: Flash0活动状态 1: Flash0处于停止模式 当寄存器位由1设为0后，请在确认FSR.RDY0位为1后，进行Flash0访问。	R/W											

### 7.9.5 Flash 读模式寄存器(EMF\_FRMCR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	CRST	PREF ETE	DCAC HE	ICAC HE
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	LVM	-	-	-	-	-	-	-	FLWT[3:0]

位	标记	位名	功能	读写
b31~b20	Reserved	-	读出时为“0”，写入时写“0”	R/W
b19	CRST	缓存复位位	0: 缓存数据解除复位 1: 缓存数据复位（缓冲、缓存、预取指）	R/W
b18	PREFETE	ICODE预取指许可位	0: 关闭ICODE预取指功能 1: ICODE预取指功能使能	R/W
b17	DCACHE	DCODE缓存许可位	0: 关闭DCODE缓存功能 1: DCODE缓存功能使能	R/W
b16	ICACHE	ICODE缓存许可位	0: 关闭ICODE缓存功能 1: ICODE缓存功能使能	R/W
b15~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8	LVM	低电压读模式	0: 正常电压读模式 1: 低电压读模式 超低速运行模式时需要置LVM为“1”	R/W
b7-b4	Reserved	-	读出时为“0”，写入时写“0”	R/W
b3~b0	FLWT[3:0]	Flash读插入的等待周期	0000b: 不插入读等待周期 0001b: 插入1个读等待周期 0010b: 插入2个读等待周期 ..... 1110b: 插入14个读等待周期 1111b: 插入15个读等待周期	R/W

## 7.9.6 Flash 擦写模式寄存器(EFM\_FWMC)

复位值：0x00030000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	KEY2 LOCK	KEY1 LOC K
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	BUSH LDCT L	-	-	-	-	-	-	PEMOD[2:0]	

位	标记	位名	功能	读写
b31~b18	Reserved	-	读出时为“0”，写入时写“0”	R/W
b17	KEY2LOCK	Flash密钥2保护状态位	0: EFM_KEY2寄存器锁定解除。 1: EFM_KEY2寄存器锁定。  该位只能写1，写0无效。当EFM_KEY2写入错误序列后，该位保持为1直至复位。  该位读到0：表示EFM_KEY2序列解锁。 该位读到1：表示EFM_KEY2序列未解锁。	R/W
b16	KEY1LOCK	Flash密钥1保护状态位	0: EFM_KEY1寄存器锁定解除。 1: EFM_KEY1寄存器锁定。  该位只能写1，写0无效。当EFM_KEY1写入错误序列后，该位保持为1直至复位。  该位读到0：表示EFM_KEY1序列解锁。 该位读到1：表示EFM_KEY1序列未解锁。	R/W
b15~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8	BUSHLDCTL	Flash擦除，编程期间总线控制	0: Flash编程擦除期间，总线被占用。 1: Flash编程擦除期间，总线释放。	R/W
b7~b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2~b0	PEMOD[2:0]	Flash编程，擦除，全擦除模式	000: 只读模式 001: 单编程模式 010: 单编程回读模式 011: 连续编程模式 100: 扇区擦除模式 101: 单块Flash全擦除模式 110: 两块Flash全擦除模式 111: 只读模式	R/W

### 7.9.7 Flash 状态寄存器(EFM\_FSR)

复位值：0x01000100

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	RDY1	-	-	COLE RR1	OPTE ND1	MISM TCH1	PGSZ ERR1	PRTW ERR1	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	RDY0	-	-	COLE RR0	OPTE ND0	MISM TCH0	PGSZ ERR0	PRTW ERR0	OTP WER R0

位	标记	位名	功能	读写
b31~b25	Reserved	-	读出时为“0”，写入时写“0”	R/W
b24	RDY1	Flash块1忙/空闲状态	0: Flash块1忙状态 1: Flash块1空闲状态	R
b23~b22	Reserved	-	读出时为“0”，写入时写“0”	R/W
b21	COLERR1	Flash块1读写冲突错误标志位	置位条件： RDY1未置位情况下，对Flash1读写操作。 清零条件： EFM_FSCLR对应清零位写1。	R
b20	OPTEND1	Flash块1空间编程，擦除，全擦除操作结束标志位	置位条件： 编程/擦除/全擦除操作结束。 清零条件： EFM_FSCLR对应清零位写1。	R
b19	MISMTCH1	Flash块1空间单编程回读值不一致标志位	置位条件： 单编程回读模式，Flash1编程后读数据和写入值不一致。 清零条件： EFM_FSCLR对应清零位写1	R
b18	PGSZERR1	Flash块1空间编程地址和大小不对齐标志位	置位条件： 编程地址非以4对齐，或者发生非32位写操作。 清零条件： EFM_FSCLR对应清零位写1	R
b17	PRTWERR1	Flash块1写保护错误标志位	置位条件： 对写保护扇区进行编程，擦除，全擦除操作。 清零条件： EFM_FSCLR对应清零位写1	R
b16~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8	RDY0	Flash块0忙/空闲状态	0: Flash块0忙状态 1: Flash块0空闲状态	R
b7~b6	Reserved	-	读出时为“0”，写入时写“0”	R/W
b5	COLERR0	Flash块0读写冲突错误标志位	置位条件： RDY0未置位情况下，对Flash0读写操作。	R

			连续编程模式下，对Flash0写不置位。	
			清零条件：	
			EFM_FSCLR对应清零位写1。	
			置位条件：	
b4	OPTEND0	Flash块0空间编程，擦除，全擦除操作结束标志位	编程/擦除/全擦除操作结束。	R
			清零条件：	
			EFM_FSCLR对应清零位写1。	
			置位条件：	
b3	MISMTCH0	Flash块0空间单编程回读值不一致标志位	单编程回读模式，Flash0编程后读数据和写入值不一致。	R
			清零条件：	
			EFM_FSCLR对应清零位写1	
			置位条件：	
b2	PGSZERRO	Flash块0编程地址和大小不对齐标志位	编程地址非以4对齐，或者发生非32位写操作。	R
			清零条件：	
			EFM_FSCLR对应清零位写1	
			置位条件：	
b1	PRTWERRO	Flash块0写保护地址错误标志位	对写保护扇区进行编程，擦除，全擦除操作。	R
			清零条件：	
			EFM_FSCLR对应清零位写1	
			置位条件：	
b0	OTPWERRO	Flash块0的OTP擦写错误标志位	对OTP锁存区域进行编程，擦除，全擦除操作。	R
			清零条件：	
			EFM_FSCLR对应清零位写1	

### 7.9.8 Flash 状态清除寄存器(EM\_FSLR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	COLE RRCL R1	OPTE NDCL R1	MISM TCHC LR1	PGSZ ERRC LR1	PRTW ERRC LR1	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	COLE RRCL R0	OPTE NDCL R0	MISM TCHC LR0	PGSZ ERRC LR0	PRTW ERRC LR0	OTP WER RCLR 0

位	标记	位名	功能	读写
b31~b22	Reserved	-	读出时为“0”，写入时写“0”	R/W
b21	COLERRCLR1	清除读写冲突错误标志位	0: 不发生清除动作 1: 清除FSR.COLERR1位 该位读时，始终为0。	R/W
b20	OPTENDCLR1	清除操作结束标志	0: 不发生清除动作 1: 清除FSR.OPTEND1位 该位读时，始终为0。	R/W
b19	MISMTCHCLR1	清除编程回读不一致标志位	0: 不发生清除动作 1: 清除FSR.MISMTCH1位 该位读时，始终为0。	R/W
b18	PGSZERRCLR1	清除编程地址和大小不对齐标志位	0: 不发生清除动作 1: 清除FSR.PGSZERR1位 该位读时，始终为0。	R/W
b17	PEPRTERRCLR1	清除写保护错误标志位	0: 不发生清除动作 1: 清除FSR.PRTWERR1位 该位读时，始终为0。	R/W
b16~b6	Reserved	-	读出时为“0”，写入时写“0”	R/W
b5	COLERRCLR0	清除读写冲突错误标志位	0: 不发生清除动作 1: 清除FSR.COLERR0 该位读时，始终为0。	R/W
b4	OPTENDCLR0	清除操作结束标志	0: 不发生清除动作 1: 清除FSR.OPTEND0位 该位读时，始终为0。	R/W
b3	MISMTCHCLR0	清除编程回读不一致标志位	0: 不发生清除动作 1: 清除FSR.MISMTCH0位 该位读时，始终为0。	R/W
b2	PGSZERRCLR0	清除编程地址和大小不对齐标志位	0: 不发生清除动作 1: 清除FSR.PGSZERR0位 该位读时，始终为0。	R/W
b1	PEPRTERRCLR0	清除写保护错误标志位	0: 不发生清除动作 1: 清除FSR.PRTWERR0位 该位读时，始终为0。	R/W
b0	OTPWERRCLR0	清除OTP擦写错误标志位	0: 不发生清除动作 1: 清除FSR.OTPWERRO位	R/W

---

该位读时，始终为0。

---

### 7.9.9 Flash 中断许可寄存器(EFM\_FITE)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	COLE RRITE	OPTE NDIT E	PEER RITE

位	标记	位名	功能	读写
b31~b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2	COLERRITE	读写冲突错误中断许可	0: 读写冲突错误中断不许可 1: 读写冲突错误中断许可	R/W
b1	OPTENDITE	操作结束中断许可	0: 操作结束中断不许可 1: 操作结束中断许可	R/W
b0	PEERRITE	编程/擦除错误中断许可	0: 编程/擦除错误中断不许可 1: 编程/擦除错误中断许可	R/W

### 7.9.10 Flash 引导交换状态寄存器(EFM\_FSWP)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FSWP

位	标记	位名	功能	读写
b31-b1	Reserved	-	读出时为“0”，写入时写“0”	R/W
b0	FSWP	扇区0和扇区1地址交换位	0: 复位后，CPU从Flash块0启动。 1: 复位后，CPU从Flash块1启动。 寄存器初始值由Flash块0地址 0x03002000~0x03002003的值决定，其数据 为0x5A5A5A时，初值为1，其余值时为0。	R

### 7.9.11 芯片专属标志寄存器(EFM\_CHIPID)

复位值：0x48440682

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CHIPID[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CHIPID[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31-b0	CHIPID[31:0]	芯片专属标志	CHIPID[31:16]: HD的对应的ASCII码 CHIPID[15:0]: 产品型名 该寄存器只读。	R											

### 7.9.12 UNIQUE ID 寄存器 0(EFM\_UQID0)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
UQID0[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UQID0[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31-b0	UQID0[31:0]	唯一码	芯片唯一码	R											

### 7.9.13 UNIQUE ID 寄存器 1(EFM\_UQID1)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
UQID1[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UQID1[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31-b0	UQID1[31:0]	唯一码	芯片唯一码	R											

### 7.9.14 UNIQUE ID 寄存器 2(EFM\_UQID2)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
UQID2[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UQID2[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31-b0	UQID2[31:0]	唯一码	芯片唯一码	R											

### 7.9.15 Flash 写保护锁定寄存器(EFM\_WLOCK)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	WLOCK1	-	-	-	WLOCK0
<hr/>															
位	标记	位名	功能	读写											
b31-b5	Reserved	-	读出时为“0”，写入时写“0”	R/W											
b4	WLOCK1	F1NWPRT寄存器锁定位	0: F1NWPRT寄存器控制的扇区写禁止未lockup 1: F1NWPRT寄存器控制的扇区写禁止lockup 该位一旦设定为1，只能通过复位才能恢复成0。	R/W											
b3-b2	Reserved	-	读出时为“0”，写入时写“0”	R/W											
b0	WLOCK0	F0NWPRT寄存器锁定位	0: F0NWPRT寄存器控制的扇区写禁止未lockup 1: F0NWPRT寄存器控制的扇区写禁止lockup 该位一旦设定为1，只能通过复位才能恢复成0。	R/W											

### 7.9.16 Flash0 写保护寄存器(EM\_FONWPRT)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
FON WPRT 31	FON WPRT 30	FON WPRT 29	F0WP RT28	FON WPRT 27	FONW PRT2 6	FON WPRT 25	FON WPRT 24	FON WPRT 23	FON WPRT 22	FON WPRT 21	FON WPRT 20	FON WPRT 19	FON WPRT 18	FON WPRT 17	FON WPRT 16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FON WPRT 15	FON WPRT 14	FON WPRT 13	FON WPRT 12	FON WPRT 11	FONW PRT1 0	FON WPRT 9	FON WPRT 8	FON WPRT 7	FON WPRT 6	FON WPRT 5	FON WPRT 4	FON WPRT 3	FON WPRT 2	FON WPRT 1	FON WPRT 0

位	标记	位名	功能	读写
b31	FONWPRT31	Flash块0扇区31的写许可位	0: Flash块0扇区31的写禁止 1: Flash块0扇区31的写许可 WLOCK0=0时，该位写0或者写1许可。 WLOCK0=1时，该位只能0，写1无效。	R/W
b30	FONWPRT30	Flash块0扇区30的写保护位	0: Flash块0扇区30的写禁止 1: Flash块0扇区30的写许可 WLOCK0=0时，该位写0或者写1许可。 WLOCK0=1时，该位只能0，写1无效。	R/W
b29~b2	...	...	...	R/W
b1	FONWPRT1	Flash块0扇区1的写保护位	0: Flash块0扇区1的写禁止 1: Flash块0扇区1的写许可 WLOCK0=0时，该位写0或者写1许可。 WLOCK0=1时，该位只能0，写1无效。	R/W
b0	FONWPRT0	Flash块0扇区0的写保护位	0: Flash块0扇区0的写禁止 1: Flash块0扇区0的写许可 WLOCK0=0时，该位写0或者写1许可。 WLOCK0=1时，该位只能0，写1无效。	R/W

### 7.9.17 Flash1 写保护寄存器(EM\_F1NWPRT)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
F1N WPRT 31	F1N WPRT 30	F1N WPRT 29	F1N WPRT 28	F1N WPRT 27	F1NW PRT2 6	F1N WPRT 25	F1N WPRT 24	F1N WPRT 23	F1N WPRT 22	F1N WPRT 21	F1N WPRT 20	F1N WPRT 19	F1N WPRT 18	F1N WPRT 17	F1N WPRT 16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
F1N WPRT 15	F1N WPRT 14	F1N WPRT 13	F1N WPRT 12	F1N WPRT 11	F1NW PRT1 0	F1N WPRT 9	F1N WPRT 8	F1N WPRT 7	F1N WPRT 6	F1N WPRT 5	F1N WPRT 4	F1N WPRT 3	F1N WPRT 2	F1N WPRT 1	F1N WPRT 0

位	标记	位名	功能	读写
b31	F1NWPRT31	Flash块1扇区31的写保护位	0: Flash块1扇区31的写禁止 1: Flash块1扇区31的写许可 WLOCK1=0时，该位写0或者写1许可。 WLOCK1=1时，该位只能0，写1无效。	R/W
b30	F1NWPRT30	Flash块1扇区30的写保护位	0: Flash块1扇区30的写禁止。 1: Flash块1扇区30的写许可。 WLOCK1=0时，该位写0或者写1许可。 WLOCK1=1时，该位只能0，写1无效。	R/W
b29~b2	...	...	...	R/W
b1	F1NWPRT1	Flash块1扇区1的写保护位	0: Flash块1扇区1的写禁止 1: Flash块1扇区1的写许可 WLOCK1=0时，该位写0或者写1许可。 WLOCK1=1时，该位只能0，写1无效。	R/W
b0	F1NWPRT0	Flash块1扇区0的写保护位	0: Flash块1扇区0的写禁止 1: Flash块1扇区0的写许可 WLOCK1=0时，该位写0或者写1许可。 WLOCK1=1时，该位只能0，写1无效。	R/W

## 7.10 注意事项

1. 在 Flash 进行擦写时，发生复位、擦写操作会强制停止，Flash 数据将得不到保证。用户需在复位解除后对地址擦除后再次进行操作。
2. 擦写操作结束后硬件电路会自动复位缓存回路。
3. 对同一地址重复编程，不能确保数据的正确性。
4. 连续编程模式时，Flash 模拟电路会有高电压状态，长期高压状态会影响 Flash 特性，一旦连续编程结束请立即退出连续编程模式。禁止在连续编程模式下，MCU 进入低功耗模式（睡眠模式、停止模式、掉电模式）。
5. 使用读加速缓存时，请确保缓存 RAM 不掉电。（PWC\_PRAMLPC.PRAMPDC2=0）

## 8 内置 SRAM (SRAM)

### 8.1 简介

本产品带有 4KB 掉电模式保持 SRAM (Ret\_SRAM) 和 64KB 系统 SRAM (SRAM0/SRMAH)。

各 SRAM 可按照字节、半字（16 位）或全字（32 位）访问。所有 SRAM (SRAMH/SRAM0/Ret SRAM) 读写操作最快可以在 CPU 最高速度 (120MHz) 下执行。

Ret\_SRAM 可在 Power down 模式下提供 4KB 的数据保持空间。

SRAM0 和 Ret\_SRAM 带有 ECC 校验 (Error Checking and Correcting)，ECC 校验为纠一检二码，即可以纠正一位错误，检查两位错误。SRAMH 带有奇偶校验 (Even-parity check)，每字节数据带有一位校验位。SRAM 详细定义见表 8-1。

表 8-1 SRAM 空间分配

名称	容量	地址范围	校验方式
SRAM0	32KB	0x20000000~0x20007FFF	ECC
SRAMH	32KB	0x1FFF8000~0xFFFFFFF	Even-parity check
Ret_SRAM	4KB	0x200F0000~0x200F0FFF	ECC

#### 注意：

- 在允许产生 RAM 奇偶校验错误产生 NMI 中断和复位的情况下，当存取数据时，必须对所用 RAM 空间以字为单位进行初始化；当从 RAMH 空间执行指令时，必须对所用 RAM 空间+3 字的区域以字为单位进行初始化。
- 在允许产生 RAM ECC 校验错误产生 NMI 中断和复位的情况下，当存取数据时，必须对所用 RAM 空间以字为单位进行初始化。

系统 SRAM 的 ECC 校验带有错误注入功能。根据 SRAM 错误注入使能寄存器和 SRAM 错误注入位使能寄存器的设定，硬件电路将 SRAM 读数据按位取反，以达到错误注入的目的。通过 SRAM 错误注入位使能寄存器的设定，可以制造 1bit 错误和多 bit 错误，由于 ECC 电路可以检出 2bit 错误，因此超过 2bit 错误有可能不会被检出。

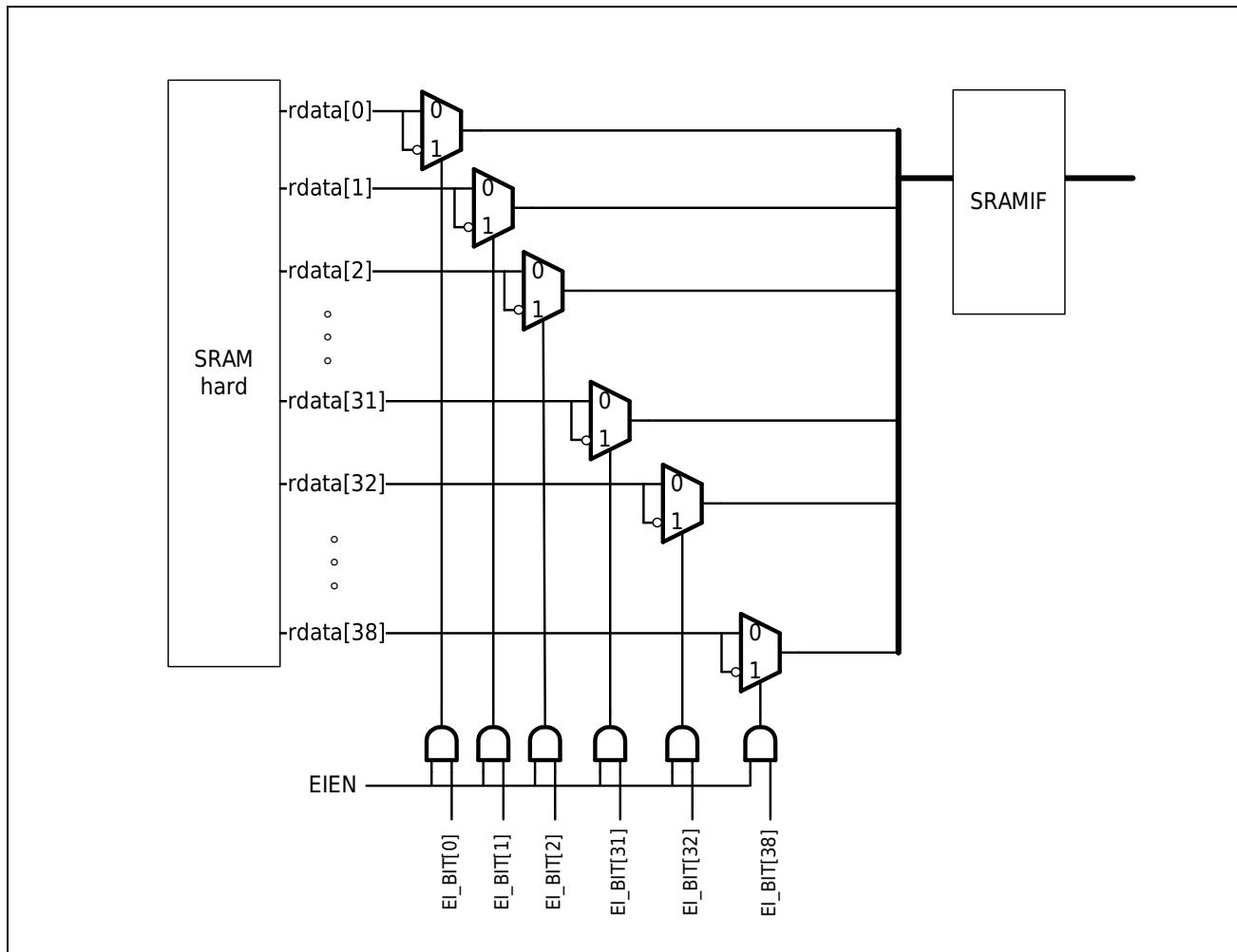


图 8-1 错误注入功能示意图

## 8.2 寄存器说明

表 8-2 SRAM 寄存器一览

寄存器名	起始地址	复位值
SRAM校验控制寄存器 (SRAM_CKCR)	0x40050808	0x00000000
SRAM校验控制保护寄存器 (SRAM_CKPR)	0x4005080C	0x00000000
SRAM校验状态寄存器 (SRAM_CKSR)	0x40050810	0x00000000
SRAM0错误注入使能寄存器 (SRAM0_EIEN)	0x40050814	0x00000000
SRAM0错误注入位使能寄存器0 (SRAM0_EIBIT0)	0x40050818	0x00000000
SRAM0错误注入位使能寄存器1 (SRAM0_EIBIT1)	0x4005081C	0x00000000
SRAM0校验错误地址寄存器 (SRAM0_ECCERRADDR)	0x40050820	0x00000000
Ret_SRAM错误注入使能寄存器 (SRAMB_EIEN) (SRAM_CKSR)	0x40050824	0x00000000
Ret_SRAM错误注入位使能寄存器0 (SRAMB_EIBIT0)	0x40050828	0x00000000
Ret_SRAM错误注入位使能寄存器1 (SRAMB_EIBIT1)	0x4005082C	0x00000000
Ret_SRAM校验错误地址寄存器 (SRAMB_ECCERRADDR)	0x40050830	0x00000000

### 8.2.1 SRAM 校验控制寄存器 (SRAM\_CKCR)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16		
Rev		SRAMB ECCMOD [1:0]	SRAM0ECCMOD [1:0]	Rev				SRAMBECC OAD	SRAM0ECC OAD								
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		
Rev														PYOAD			
<hr/>																	
位	标记	位名	功能										读写				
b31~b28	Reserved	-	读出时为“0”,写入时写“0”										R/W				
b27~b26	SRAMBECC MOD[1:0]	Ret_SRAM的ECC校验允许位	00: 禁止ECC校验功能 01: 若1位错误, ECC纠错, 不产生1位错误标志, 不产生中断/复位; 若2位错误, ECC检错, 产生2位错误标志, 产生中断/复位. 10: 若1位错误, ECC纠错, 产生1位错误标志, 不产生中断/复位; 若2位错误, ECC检错, 产生2位错误标志, 产生中断/复位. 11: 若1位错误, ECC纠错, 产生1位错误标志, 产生中断/复位; 若2位错误, ECC检错, 产生2位错误标志, 产生中断/复位.										R/W				
b25~b24	SRAM0ECC MOD[1:0]	SRAM0的ECC校验允许位	00: 禁止ECC校验功能 01: 若1位错误, ECC纠错, 不产生1位错误标志, 不产生中断/复位; 若2位错误, ECC检错, 产生2位错误标志, 产生中断/复位. 10: 若1位错误, ECC纠错, 产生1位错误标志, 不产生中断/复位; 若2位错误, ECC检错, 产生2位错误标志, 产生中断/复位. 11: 若1位错误, ECC纠错, 产生1位错误标志, 产生中断/复位; 若2位错误, ECC检错, 产生2位错误标志, 产生中断/复位.										R/W				
b23~b18	Reserved	-	读出时为“0”,写入时写“0”										R/W				
b17	SRAMB ECCOAD	Ret_SRAM ECC校验出错后操作	0: Non-maskable interrupt 1: Reset										R/W				
b16	SRAM0ECCOAD	SRAM0 ECC校验出错后操作	0: Non-maskable interrupt 1: Reset										R/W				
b15~b1	Reserved	-	读出时为“0”,写入时写“0”										R/W				
b0	PYOAD	奇偶校验 出错后操作	0: Non-maskable interrupt 1: Reset										R/W				

**注意：**

- 在允许产生 RAM 奇偶校验错误产生 NMI 中断和复位的情况下，当存取数据时，必须对所用 RAM 空间以字为单位进行初始化；当从 RAMH 空间执行指令时，必须对所用 RAM 空间+3 字的区域以字为单位进行初始化。
- 在允许产生 RAM ECC 校验错误产生 NMI 中断和复位的情况下，当存取数据时，必须对所用 RAM 空间以字为单位进行初始化。

## 8.2.2 SRAM 校验控制保护寄存器 (SRAM\_CKPR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Rev															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Rev								CKPRKW[6:0]						CKPRC	

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7~b1	CKPRKW[6:0]	写入关键码	对当前寄存器写时, 需在这些位写入“3b”使能当前寄存器的使能	R/W
b0	CKPRC	SRAM校验控制寄存器写使能	0: SRAM校验控制寄存器写入禁止 1: SRAM校验控制寄存器写入使能	R/W

CKPRC: 对 SRAMCKCR 寄存器的写进行控制。当 CKPRC 被置为 1, 对于 SRAMCKCR 的写入操作是允许的, 如果其被置为 0 时, 不能对 SRAMCKCR 进行写操作。当写入此位时, 必须同时对 CKPRKW[6:0]写入 3bh。

### 8.2.3 SRAM 校验状态寄存器 (SRAM\_CKSR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Rev															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Rev															
位	标记	位名	功能	读写											
b31~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b8	CACHERAM_PYERR	CACHERAM奇偶校验错误标志	0: 无奇偶校验错误发生 1: 有奇偶校验错误发生	R/W (注1)											
b7	SRAMB_2ERR	Ret_SRAM发生ECC 2-bit错误标志	0: 无2-bit ECC错误发生 1: 有2-bit ECC错误发生	R/W (注1)											
b6	SRAMB_1ERR	Ret_SRAM发生ECC 1-bit错误标志	0: 无1-bit ECC错误发生 1: 有1-bit ECC错误发生	R/W (注1)											
b5	SRAM0_2ERR	SRAM0发生ECC 2-bit错误标志	0: 无2-bit ECC错误发生 1: 有2-bit ECC错误发生	R/W (注1)											
b4	SRAM0_1ERR	SRAM0发生ECC 1-bit错误标志	0: 无1-bit ECC错误发生 1: 有1-bit ECC错误发生	R/W (注1)											
b3	SRAMH_PYERR	SRAMH奇偶 校验错误标志	0: 无奇偶校验错误发生 1: 有奇偶校验错误发生	R/W (注1)											
b2~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W											

注 1: 写 1 清 0。

### 8.2.4 SRAM0 错误注入使能寄存器 (SRAM0\_EIEN)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Rev															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Rev															
位	标记	位名	功能	读写											
b31~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b0	EIEN	错误注入使能	0: 错误注入禁止 1: 错误注入有效	R/W											

### 8.2.5 SRAM0 错误注入位使能寄存器 0 (SRAM0\_EIBIT0)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EIBIT[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIBIT[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b0	EIBIT[31:0]	错误注入位使能	控制SRAM0输出数据的0~31位是否取反 0: 不取反 1: 取反	R/W											

### 8.2.6 SRAM0 错误注入位使能寄存器 1 (SRAM0\_EIBIT1)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Rev															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Rev															
<hr/>															
位	标记	位名	功能	读写											
b31~b7	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b6~b0	EIBIT[38:32]	错误注入位使能	控制SRAM0输出数据的32~38位是否取反 0: 不取反 1: 取反	R/W											

### 8.2.7 SRAM0 校验错误地址寄存器 (SRAM0\_ECCERRADDR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Rev															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Rev															
<hr/>															
位	标记	位名	功能	读写											
b31~b15	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b14~b0	ECCERRADDR [14:0]	校验错误地址	记录SRAM0发生1bit或2bit ECC错误时的访问地址	R											

### 8.2.8 Ret\_SRAM 错误注入使能寄存器(SRAMB\_EIEN)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Rev															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Rev															
EIEN															
位	标记	位名	功能	读写											
b31~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b0	EIEN	错误注入使能	0: 错误注入禁止 1: 错误注入有效	R/W											

### 8.2.9 Ret\_SRAM 错误注入位使能寄存器 0(SRAMB\_EIBIT0)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EIBIT[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIBIT[15:0]															
控制Ret_SRAM输出数据的0~31位是否取反															
b31~b0	EIBIT[31:0]	错误注入位使能	0: 不取反 1: 取反	R/W											
位	标记	位名	功能	读写											

### 8.2.10 Ret\_SRAM 错误注入位使能寄存器 1(SRAMB\_EIBIT1)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Rev															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Rev										EIBIT[38:32]					
<hr/>															
位	标记	位名	功能	读写											
b31~b7	Reserved	-	读出时为“0”,写入时写“0” 控制Ret_SRAM输出数据的32~38位是否取反	R/W											
b6~b0	EIBIT[38:32]	错误注入位使能	0: 不取反 1: 取反	R/W											

### 8.2.11 Ret\_SRAM 校验错误地址寄存器(SRAMB\_ECCERRADDR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Rev															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Rev										ECCERRADDR [11:0]					
<hr/>															
位	标记	位名	功能	读写											
b31~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b11~b0	ECCERRADDR [11:0]	校验错误地址	记录Ret_SRAM发生1bit或2bit ECC错误时的访问地址	R											

## 9 通用 IO (GPIO)

本章中使用的一些略称：

- Px (x=A~F) 表示一组端口，如 PA 表示 PA0~PA15 这一组的 16 个 I/O 端口。
- Pxy (x= A~F, y=0~15, 以下同) 表示单个 I/O 端口，如 PB10 端口表示 PB 组中第 11 个 I/O。
- GPIO (General Purpose Input Output) 通用输入输出。
- NOD/POD (Nmos/Pmos Open Drain) NMOS/PMOS 开漏输出模式。

### 9.1 简介

主要特性：

- 每组 Port 配有 16 个 I/O Pin，根据实际配置可能不足 16 个
- 支持上拉与下拉输入
- 支持推挽，开漏输出模式
- 支持高，中，低型驱动模式
- 支持 CMOS/Schmitt 两种输入模式自由切换
- 支持外部中断的输入
- 支持 I/O pin 周边功能复用，一个 I/O pin 最多可具有 64 个可选择的复用功能
- 各个 I/O pin 可独立编程
- 各个 I/O pin 可以选择 2 个功能同时有效（不支持 2 个输出功能同时有效）

## 9.2 端口功能概要

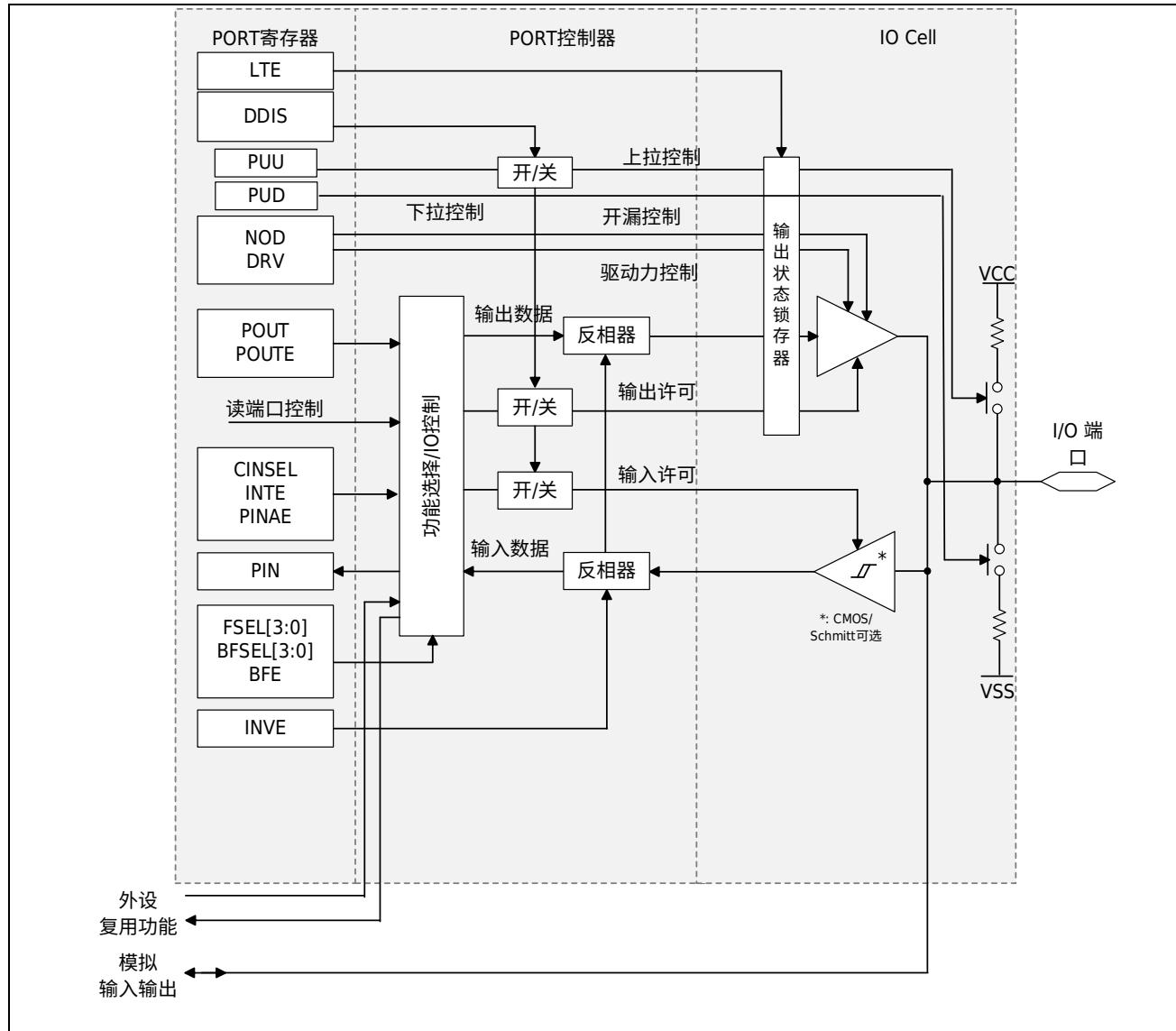


图 9-1 端口基本结构示意图

详细 GPIO 端口数目、5V 耐用、驱动能力配置请参考数据手册中引脚配置及功能章节。

## 9.3 动作说明

### 9.3.1 通用输入输出 GPIO 功能

#### 通用输入功能 GPI:

各 I/O 都具有通用输入 GPI 功能，且在数字功能禁止位 PCRxy.DDIS 为 0 时，GPI 功能始终有效，与功能选择寄存器中 PFSRxy 的 FSEL[5:0]设定值无关。通过访问端口输入数据寄存器 PIDRx 可以获取当前端口的状态。也可以通过端口控制寄存器 PCRxy 的 PIN 位查询相应的单 I/O 端口状态，PIDRx.PIN[y]寄存器位与 PCRxy.PIN 位等价。

默认情况下，为了降低功耗，I/O 的输入 MOS 是被关闭的。只有在读取 PIDRx，PCRxy 寄存器时，才会被打开。根据需要，也可以通过设置寄存器 PCRxy.PINAE 为 1，让 I/O 的输入 MOS 一直处于打开状态。

当系统运行在高速时钟下，且 PCRxy.PINAE 设置为 0 时，由于 I/O 输入存在延迟，单周期可能无法正确读取输入状态值。此时需要设置寄存器 PCCR.RDWWT[2:0]，插入若干等待周期。具体参考寄存器 PCCR 说明。

#### 通用输出功能 GPO:

除输入专用的端口，其他 I/O 端口都具有通用输出 GPO 功能。通过设置端口功能选择寄存器 PFSRxy.FSEL[5:0]为 0x0 可以有效 GPO 功能。

GPO 功能有效时，可以通过设置通用输出许可寄存器 POERx 来允许或者禁止 I/O 的输出，通用输出数据寄存器 PODRx 来控制的输出值。使用下面的 3 个寄存器也可以控制 I/O 的输出值：输出数据清零寄存器 PORRx，输出数据置位寄存器 POSRx，输出数据翻转寄存器 POTRx。对上述寄存器中相应位写 1 可使对应 I/O 输出 0、1、翻转。写 0 时 I/O 输出状态不改变。

上面的寄存器都是 16 个 PORT 一组一起操作的。为了方便对单个 I/O 的控制，也可通过设置 PCRxy.POUTE 来允许或者禁止 I/O 的输出，PCRxy.POUTE 寄存器位与 POERx.POUTE[y]等价。可通过设置 PCRxy.POOUT 来控制 I/O 的输出值，PCRxy.POOUT 寄存器位与 PODRx.POOUT[y]等价。PCRxy 适合用于控制单个 PORT，POERx/PODRx 适合于控制 16 位整组 PORT。

系统复位后，除了 JTAG 复用端口 PA13，PA14，PA15，PB3，PB4，副振荡器复用端口 PC14，PC15 以外，其他端口的初始功能均为 GPO(FSEL[5:0]=0x0)，且处于高阻态（输出禁止 POUTE=0）。

### 9.3.2 周边功能

通过功能选择寄存器 PFSRxy 的 FSEL[5:0]，每个端口可以配置最多 64 个功能。其中包括 FSEL[5:0]=0x0 对应的通用输出 GPO 功能。各端口具体配置的功能请参考数据手册中引脚功能表。

JTAG/SWD 调试功能，使用寄存器 PSPCR 选择。PSPCR.SPFE[z]，z=0~4 为 1 时，对应端口的 PFSRxy.FSEL[5:0]寄存器位无效，即 SPFE 优先级高于 FSEL。PSPCR 寄存器初始值为 0x1f，JTAG/SWD 功能有效。如果要将这些端口设置为 JTAG/SWD 以外的功能，需要先对对应 SPFE[z]位写 0。

### 9.3.3 双周边功能

有些应用情况，需要将一个端口同时设成两种功能。这种情况可以先由 PFSRxy.FSEL[5:0]选择好一个功能，再通过设置公共控制寄存器 PCCR.BFSEL[5:0]选择第二个功能，并设置 PFSRxy.BFE 为 1。例如：设置 PFSRxy.FSEL[5:0]=0x2，PCCR.BFSEL[5:0]=0x5，PFSRxy.BFE=0x1，则 Pxy 上的功能 2 与功能 5 就将同时有效。禁止在同一端口上同时有效 2 个输出功能。

### 9.3.4 Event Port 输入输出功能

支持 4 组 Event Port，每组 16 个端口。Event Port1 包含 EVNTP100~EVNTP115，Event Port2 包含 EVNTP200~EVNTP215，以此类推。EVNTPmn (m=1~4, n=0~15) 端口可以作为触发源，根据端口输入生成事件来触发其它周边设备（如 TIMER，ADC，DMA 等）开始特定的动作。也可以作为被触发对象，接受事件，自动输入或输出。

作为触发源时，设置 PEVNTRISRm,PEVNTFALRm,PEVNTNFCR 选择上升沿或下降沿检测，以及数字滤波功能，并设置功能选择寄存器 PFSRxy 选择 EVNTPmn 功能。当选择的边沿从端口输入时，则生成事件 EVENT\_PORTm，输出到其它周边设备以触发其开始动作。

作为被触发对象时，设置 PEVNTTRGSRm 选择触发事件源，设置 PEVNTDIRRm 选择输出或输入功能。输出功能时，选定事件发生时 EVNTPmn 根据 PEVNTODRm, PEVNTORRm, PEVNTOSRm 设定值输出指定电平或翻转。输入功能时，选定事件发生时，EVNTPmn 输入状态保存进寄存器 PEVNTIDRx。

使用 Event Port 功能时，需要先将功能时钟控制 0 寄存器(PWC\_FCG0)的自动运行系统 AOS 功能使能位设置为有效。

### 9.3.5 外部中断 EIRQ 输入功能

除 PF3/MD 引脚外，每个 I/O 端口都具有外部中断输入功能。当 PCRxy.INTE 位设为 1 时，此 I/O 将作为外部中断源 EIRQy 被允许输入（如：PA0 对应 EIRQ0，PA2 对应 EIRQ2）。每个 EIRQy 可配置的 I/O 不止一个，使用时每个 EIRQy 不要同时允许多个 I/O 输入。EIRQy 输入功能与 PFSRxy.FSEL 选择的周边功能（包括 GPIO）可同时有效。

I/O 端口作为外部中断 EIRQ 使用时，需要结合中断控制器 INTC，设置滤波，中断触发沿，中断号等。详细请参考【中断控制器（INTC）】。

### 9.3.6 模拟功能

部分 I/O 端口带有模拟输入输出功能(包括主副振荡器)。在用作模拟功能时,请将寄存器 PCRxy.DDIS 写 1, 禁止当前端口的数字功能。

### 9.3.7 通用控制

#### 1. 上/下拉电阻

各 I/O 端口带有内部的上拉电阻。可以设置寄存器 PCRxy.PUU 位允许此功能,在 I/O 端口无输入时,内部处于弱 1 状态。在 I/O 端口处于输出状态时,上拉功能将自动无效。

当 I/O 端口选择 I2Cx\_SCL/I2Cx\_SDA 功能时,将无视寄存器 PUU 的设定,内部上拉功能被强制无效。

PA11/USBFS\_DM, PA12/USBFS\_DP 引脚与 USB 端口复用, 2 个引脚均内藏约 400KΩ 的下拉电阻,且一直有效。

各 I/O 端口带有内部的下拉电阻。可以设置寄存器 PCRxy.PUD 位允许此功能,在 I/O 端口无输入时,内部处于弱 0 状态。在 I/O 端口处于输出状态时,下拉功能将自动无效。

当 I/O 端口选择 I2Cx\_SCL/I2Cx\_SDA 功能时,将无视寄存器 PUD 的设定,内部下拉功能被强制无效。

PCRxy.PUU 与 PCRxy.PUD 可以同时为 1 但当上拉许可有效时,下拉许可将被屏蔽,下拉无效。

寄存器可以写 2'b11,可读出 2'b11,但实际上上拉下拉不会同时有效。

#### 2. 驱动能力控制

各 I/O 端口都具有高、中、低 3 档驱动能力可调,可根据需要设置寄存器 PCRxy.DRV[1:0]。本功能只在端口处于输出状态时才有效。

#### 3. 开漏输出模式

设置 PCRxy.NOD 位,可以将 I/O 端口设置成 NMOS 开漏输出模式。当 NOD 有效时,对应端口可正常输出 0,而输出 1 时端口将处于高阻态。

当 I/O 端口选择 I2Cx\_SCL/I2Cx\_SDA 功能时,将无视寄存器 NOD 的设定,开漏输出模式强制有效。

#### 4. CMOS/Schmitt 输入模式

各 I/O 端口支持 CMOS 和 Schmitt 两种输入模式。可以设置寄存器 PCRxy.CINSEL 位选项,设置为 0 表示 Schmitt 输入,1 表示 CMOS 输入。默认为 Schmitt 输入。

在使用 I/O 输入功能时,请根据实际需求设置输入类型。

以上所述通用控制功能,如无特别说明,它们与端口具体选择的功能即 FSEL[5:0]的设置无关。

## 9.4 寄存器说明

表 9-1 PORT 寄存器一览 1

BASE\_ADDR: 0x40053800

寄存器名	符号	偏移地址	位宽	复位值
通用输入数据寄存器	PIDRx	0x00+0x10*n *1	16/32	0xFFFF
通用输出数据寄存器	PODRx	0x04+0x10*n	16/32	0x0000
通用输出许可寄存器	POERx	0x06+0x10*n	16/32	0x0000
通用输出置位寄存器	POSRx	0x08+0x10*n	16/32	0x0000
通用输出复位寄存器	PORRx	0x0A+0x10*n	16/32	0x0000
通用输出翻转寄存器	POTRx	0x0C+0x10*n	16/32	0x0000
特殊控制寄存器	PSPCR	0x3F4	16/32	0x001F
公共控制寄存器	PCCR	0x3F8	16/32	0x1000
写保护寄存器	PWPR	0x3FC	16/32	0x0000
通用控制寄存器	PCRxy	0x400+0x40*n+0x4*y	16/32	0x0X00 *2
功能选择寄存器	PFSRxy	0x402+0x40*n+0x4*y	16/32	0x0000

注 \*1: 地址计算公式中  $x=A\sim F$  对应  $n=0\sim 5$ 

\*2: 32K 副振荡器复用端口 PCRC14, PCRC15 的复位值为 0x8100。

表 9-2 PORT 寄存器一览 2

BASE\_ADDR: 0x40010800

寄存器名	符号	偏移地址	位宽	复位值
Event Port方向选择寄存器	PEVNTDIRm	0x100+0x1C*(m-1)	32	0x00000000
Event Port输入数据寄存器	PEVNTIDRm	0x104+0x1C*(m-1)	32	0x00000000
Event Port输出数据寄存器	PEVNTODRm	0x108+0x1C*(m-1)	32	0x00000000
Event Port输出数据复位寄存器	PEVNTORRm	0x10C+0x1C*(m-1)	32	0x00000000
Event Port输出数据置位寄存器	PEVNTOSRm	0x110+0x1C*(m-1)	32	0x00000000
Event Port上升沿输入许可寄存器	PEVNTTRISRm	0x114+0x1C*(m-1)	32	0x00000000
Event Port下降沿输入许可寄存器	PEVNTFALRm	0x118+0x1C*(m-1)	32	0x00000000
Event Port输入滤波控制寄存器	PEVNTNFCR	0x170	32	0x00000000

注:  $m=1\sim 4$

### 9.4.1 通用输入数据寄存器 (PIDRx)

复位值：0xXXXX

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN[15:0]															

位	标记	位名	功能	读写
b15~b0	PIN[15:0]	输入状态	0: I/O端口输入状态为低电平 1: I/O端口输入状态为高电平	R

本寄存器为只读寄存器，写无效。在数字功能未被禁止 DDIS=0 时，通过读取此寄存器可以获取端口的输入状态，与功能选择寄存器的 PFSRxy.FSEL[5:0]设定值无关。不存在端口对应位的读出值不定。在端口的数字功能禁止状态 DDIS=1 时，由于 I/O 输入 MOS 处于关闭状态，对应 PIN 位读出值为固定值 0x1。

### 9.4.2 通用输出数据寄存器 (PODRx)

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POUT[15:0]															

位	标记	位名	功能	读写
b15~b0	POUT[15:0]	输出数据	0: 输出低电平 1: 输出高电平	R/W

当 I/O 端口被设置成 GPO 功能时，改写此寄存器可以更改对应端口的输出状态。

### 9.4.3 通用输出许可寄存器 (POERx)

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POUTE[15:0]															

位	标记	位名	功能	读写
b15~b0	POUTE[15:0]	输出许可	0: 输出禁止 1: 输出许可	R/W

当 I/O 端口被设置成 GPO 功能时，且此寄存器设为 1 时，PODRx 设定值将输出到对应 I/O 端口。此寄存器设为 0 时，输出关闭，端口为高阻态。不存在端口对应位请不要写 1。

#### 9.4.4 通用输出置位寄存器 (POSRx)

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POS[15:0]															

位	标记	位名	功能	读写
b15~b0	POS[15:0]	输出高	0: 对应PODRx.POUT无变化 1: 对应PODRx.POUT置1	R/W

此寄存器的读出值始终为 0x0000。32bit 访问时，同一 I/O 的 POR[y]与 POS[y]同时写 1 时，POR[y]优先级更高，即对应 POUT[y]清零。

#### 9.4.5 通用输出复位寄存器 (PORRx)

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POR[15:0]															

位	标记	位名	功能	读写
b15~b0	POR[15:0]	输出低	0: 对应PODRx.POUT无变化 1: 对应PODRx.POUT清零	R/W

此寄存器的读出值始终为 0x0000。

#### 9.4.6 通用输出翻转寄存器 (POTRx)

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POT[15:0]															

位	标记	位名	功能	读写
b15~b0	POT[15:0]	输出翻转	0: 对应PODRx.POUT无变化 1: 对应PODRx.POUT取反	R/W

此寄存器的读出值始终为 0x0000。

### 9.4.7 特殊控制寄存器 (PSPCR)

复位值: 0x001f

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0											
-	-	-	-	-	-	-	-	-	-	-	-	SPFE[4:0]														

位	标记	位名	功能	读写
b15~b5	Reserved	-	读出时为0, 写时请写0	R/W
b4	SPFE[4]	特殊功能选择	0: NJTRST功能无效 1: NJTRST功能有效	R/W
b3	SPFE[3]	特殊功能选择	0: JTDI能无效 1: JTDI功能有效	R/W
b2	SPFE[2]	特殊功能选择	0: JTDO_TRACESWO功能无效 1: JTDO_TRACESWO功能有效	R/W
b1	SPFE[1]	特殊功能选择	0: JTMS_SWDIO功能无效 1: JTMS_SWDIO功能有效	R/W
b0	SPFE[0]	特殊功能选择	0: JTCK_SWCLK功能无效 1: JTCK_SWCLK功能有效	R/W

#### 注意:

- SPFE[4:0]功能选择位优先级高于 PFSRxy.FSEL[5:0]功能选择位。

### 9.4.8 公共控制寄存器 (PCCR)

复位值: 0x1000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-		RDWT[2:0]		-	-	-	-	-	-			BFSEL[5:0]			

位	标记	位名	功能	读写
b15	Reserved	-	读出时为0, 写时请写0	R/W
b14-b12	RDWT[2:0]	读端口等待	设置读寄存器PIDRx, PCRxy时插入的等待周期数 工作频率      最小等待周期      设定范围 ~50MHz      0周期      0~7 50~100MHz      1周期      1~7 100~120MHz      2周期      2~7	R/W
b11~b6	Reserved	-	读出时为0, 写时请写0	R/W
b5~b0	BFSEL[5:0]	副功能选择	各端口的功能配置请参考数据手册中引脚功能表	R/W

### 9.4.9 写保护寄存器 (PWPR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
					WP[7:0]			-	-	-	-	-	-	-	WE

位	标记	位名	功能	读写
b15~b8	WP[7:0]	写保护码	读出时为0x00 当b15~b8写入值为0xA5时, b0值写入WE 当写0xA5以外值时, WE自动清零	W
b7~b1	Reserved	-	读出时为0, 写时请写0	R/W
b0	WE	写许可	0: PSPCR, PCCR, PCRxy, PFSRxy寄存器写禁止 1: PSPCR, PCCR, PCRxy, PFSRxy寄存器写许可	R/W

### 9.4.10 通用控制寄存器 (PCRxy)

复位值: b0000\_000x\_0000\_0000 \*1

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DDIS	LTE	PINAE	INTE	-	CINSEL	INVE	PIN	PUD	PUU	DRV[1:0]	-	NOD	POUTE	POUT	

位	标记	位名	功能	读写
b15	DDIS	数字功能禁止	0: 数字功能有效 1: 数字功能禁止	R/W
b14	LTE	输出状态锁存	0: 输出锁存无效 1: 输出锁存有效	R/W
b13	PINAE	输入常开	0: 输入MOS常开无效 1: 输入MOS常开有效	R/W
b12	INTE	外部中断许可	0: 外部中断输入禁止 1: 外部中断输入许可	R/W
b11	Reserved	-	读出时为0, 写时请写0	R/W
b10	CINSEL	输入模式选择	0: Schmitt输入 1: CMOS输入	R/W
b9	INVE	反相许可	0: 输入输出数据不反相 1: 输入输出数据反相	R/W
b8	PIN	输入状态	0: I/O端口输入状态为低电平 1: I/O端口输入状态为高电平 与寄存器PIDRx中PIN[y]功能一致	R
b7	PUD	下拉许可	0: 内部下拉(pulldown)电阻无效 1: 内部下拉(pulldown)电阻有效	R/W
b6	PUU	上拉许可	0: 内部上拉(pullup)电阻无效 1: 内部上拉(pullup)电阻有效	R/W
b5~b4	DRV[1:0]	驱动模式选择	b00: 低驱动力模式 b01: 中驱动力模式 b10, b11: 高驱动力模式	R/W
b3	Reserved	-	读出时为0, 写时请写0	R/W
b2	NOD	NMOS开漏	0: 正常CMOS输出模式 1: NMOS开漏输出	R/W
b1	POUTE	输出许可	0: 输出禁止 1: 输出许可 与寄存器POERx中POUTE[y]功能一致	R/W
b0	POUT	输出数据	0: 输出低电平 1: 输出高电平 与寄存器PODRx中POUT[y]功能一致	R/W

DDIS 设为 1 时, 对应端口的所有数字功能全部强制无效, 包括通用输入输出, 周边的数字输入输出, 上拉功能, 以及外部中断输入功能。当端口作为模拟输入时, 请将 DDIS 位设为 1。

LTE 设为 1 输出锁存有效时, 端口当前的输出状态保持, 直至 LTE 写成 0。本功能主要在端口功能切换时使用。为避免功能切换时端口输出意想之外的毛刺而导致系统误动作, 在功能切换前, 先将 LTE 写 1

锁存住端口的输出状态，再改写寄存器选择寄存器切换功能，最后再将 LTE 写 0 解除锁存，端口状态更新为新功能。

INVE 设为 1 时，端口的输入输出数据都会进行反相，包含 GPIO 功能，和其它周边的输入输出功能。

CINSEL 用于设置 I/O 输入模式，作用于 GPI，EIRQ 及周边输入等所有数字输入功能。

\*1：以下端口通用控制寄存器 PCR 的复位值不是 b0000\_000x\_0000\_0000，请注意。XTAL32\_IN，XTAL32\_OUT 复用端口 PC15，PC14 的 PCRC15，PCRC14 寄存器的复位值为 0x8100。

### 9.4.11 功能选择寄存器 (PFSRxy)

复位值：0x0000 \*1

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	BFE	-	-			FSEL[5:0]			

位	标记	位名	功能	读写
b15~b9	Reserved	-	读出时为0, 写时请写0	R/W
b8	BFE	副功能许可	控制PCCR.BFSEL[5:0]选择的副功能是否有效 0: 副功能禁止 1: 副功能有效	R/W
b7~b6	Reserved	-	读出时为0, 写时请写0	R/W
b5~b0	FSEL[5:0]	功能选择	各端口的功能配置请参考 <a href="#">数据手册中引脚功能表</a>	R/W

各 I/O 端口可以通过 FSEL[5:0]选择配置在该端口上多个功能中的一个。参考[数据手册中引脚功能表](#)，FSEL[5:0]设为 b000000 表示选择 Func0，设为 b000001 表示选择 Func1，以此类推，设为 b001111 表示选择 Func15。其中 Func0 对应的通用输出功能 GPO。

#### 注意：

- PA13, PA14, PA15, PB3, PB4 端口复位后初始状态为 JTAG/SWD 功能有效，在配置 FSEL[5:0] 选择功能时需要先将寄存器 PSPCR 相应位写 0 无效 JTAG/SWD 功能。PC14, PC15 端口复位后初始状态为数字功能禁止状态，在选择数字功能时需要先将相应寄存器 PCRxy 的 DDIS 位写 0 有效数字功能。

#### 9.4.12 Event Port 方向选择寄存器 (PEVNTDIRm)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PDIR[15:0]															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为0, 写时请写0	R/W											
b15~b0	PDIR15:0]	方向选择	0: Event Port为输入功能 1: Event Port为输出功能	R/W											

#### 9.4.13 Event Port 输入数据寄存器 (PEVNTIDRm)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN[15:0]															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为0, 写时请写0	R/w											
b15~b0	PIN[15:0]	端口输入状态	0: 事件触发时Event Port输入状态为低电平 1: 事件触发时Event Port输入状态为高电平	R											

当 Event Port 的方向设置为输入状态时, 在设置的事件触发时, 相应 I/O 端口的输入状态被保存到本寄存器中。

#### 9.4.14 Event Port 输出数据寄存器 (PEVNTODRm)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
<hr/>															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
<hr/>															
POUT[15:0]															

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为0, 写时请写0	R/W
b15~b0	POUT[15:0]	端口输出值	0: Event Port输出低电平 1: Event Port输出高电平	R/W

当 Event Port 的方向设置为输出状态时, 写本寄存器, 在设置的事件触发前, Event Port 的初始输出值。当选定的事件触发后, 根据 PEVNTORRm, PEVNTOSRm 设定值, PEVNTODRm.POUT 相应位清 0, 置 1, 或者翻转, 同时输出到 EVNTPmn 端口上。

#### 9.4.15 Event Port 输出数据复位寄存器 (PEVNTORRm)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
<hr/>															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
<hr/>															
POR[15:0]															

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为0, 写时请写0	R/W
b15~b0	POR[15:0]	输出值复位	0: 事件触发时对应PEVNTODRm.POUT无变化 1: 事件触发时对应PEVNTODRm.POUT复位	R/W

当 PEVNTORRm.POR 与 PEVNTm.POS 都设为 1 时, 事件触发时对应 PEVNTODRm.POUT 翻转。

### 9.4.16 Event Port 输出数据置位寄存器 (PEVNTOSRm)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POS[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为0, 写时请写0	R/W											
b15~b0	POS[15:0]	输出值置位	0: 事件触发时对应PEVNTODRm.POUT无变化 1: 事件触发时对应PEVNTODRm.POUT置位	R/W											

当 PEVNTORRm.POR 与 PEVNTm.POS 都设为 1 时，事件触发时对应 PEVNTODRm.POUT 翻转。

### 9.4.17 Event Port 上升沿输入许可寄存器 (PEVNTRISRm)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RIS[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为0, 写时请写0	R/W											
b15~b0	RIS[15:0]	上升沿检测许可	0: EVNTPmn上升沿事件检出无效 1: EVNTPmn上升沿事件检出有效 PEVNTRISRm.RIS[n]对应EVNTPmn	R/W											

Event Port 作为事件源，当 RIS 位设为 1 时，对应 EVNTPmn 的输入上升沿时，输出事件，用以触发其他周边模块。EVNTPm0~15 的边沿事件是合并成一个事件 EVENT\_PORTm 输出的，其中任何一个端口检测出边沿后都会输出事件 EVENT\_PORTm。

#### 9.4.18 Event Port 下降沿输入许可寄存器 (PEVNTRISRm)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FAL[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为0, 写时请写0	R/W											
b15~b0	FAL[15:0]	下降沿检测许可	0: EVNTPmn下降沿事件检出无效 1: EVNTPmn下降沿事件检出有效 PEVNTRISRm.FAL[n]对应EVNTPmn	R/W											

Event Port 作为事件源, 当 FAL 位设为 1 时, 对应 EVNTP 的输入下降沿时, 输出事件, 用以触发其他周边模块。EVNTPm0~15 的边沿事件是合并成一个事件 EVENT\_PORTm 输出的, 其中任何一个端口检测出边沿后都会输出事件 EVENT\_PORTm。

### 9.4.19 Event Port 输入滤波控制寄存器 (PEVNTNFCR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-		DIVS4[1:0]		NFEN4		-		DIVS3[1:0]		NFEN3					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-		DIVS2[1:0]		NFEN2		-		DIVS1[1:0]		NFEN1					
<hr/>															
位	标记	位名	功能	读写											
b31~b27	Reserved	-	读出时为0, 写时请写0	R/W											
b26-b25	DIVS4[1:0]	数字滤波采样时钟选择	Event Port4数字滤波采样时钟选择 00: PCLK1 01: PCLK1/8 10: PCLK1/32 11: PCLK1/64	R/W											
b24	NFEN4	数字滤波许可	0: Event Port4 数字滤波无效 1: Event Port4 数字滤波有效	R/W											
b23~b19	Reserved	-	读出时为0, 写时请写0	R/W											
b18-b17	DIVS3[1:0]	数字滤波采样时钟选择	Event Port3数字滤波采样时钟选择 00: PCLK1 01: PCLK1/8 10: PCLK1/32 11: PCLK1/64	R/W											
b16	NFEN3	数字滤波许可	0: Event Port3 数字滤波无效 1: Event Port3 数字滤波有效	R/W											
b15~b11	Reserved	-	读出时为0, 写时请写0	R/W											
b10-b9	DIVS2[1:0]	数字滤波采样时钟选择	Event Port2数字滤波采样时钟选择 00: PCLK1 01: PCLK1/8 10: PCLK1/32 11: PCLK1/64	R/W											
b8	NFEN2	数字滤波许可	0: Event Port2 数字滤波无效 1: Event Port2 数字滤波有效	R/W											
b7~b3	Reserved	-	读出时为0, 写时请写0	R/W											
b2-b1	DIVS1[1:0]	数字滤波采样时钟选择	Event Port1数字滤波采样时钟选择 00: PCLK1 01: PCLK1/8 10: PCLK1/32 11: PCLK1/64	R/W											
b0	NFEN1	数字滤波许可	0: Event Port1 数字滤波无效 1: Event Port1 数字滤波有效	R/W											

### 9.4.20 32bit 访问

上面所述寄存器中，除 Event Port 相关寄存只支持 32bit 访问外，其他寄存器支持 16bit 和 32bit 访问，不支持 8bit 访问。32bit 访问时这些寄存器结合方式如下：

表 9-3 32bit 访问时 PORT 寄存器一览

地址	b31	~	b16	~	b15	~	b0
0x40053800+0x10*n *1	Reserved				PIDRx		
0x40053804+0x10*n	POERx				PODRx		
0x40053808+0x10*n	PORRx				POSRx		
0x4005380C+0x10*n	Reserved				POTRx		
0x40053BF4	Reserved				PSPCR		
0x40053BF8	Reserved				PCCR		
0x40053BFC	Reserved				PWPR		
0x40053C00+0x40*n+0x04*y	PFSRxy				PCRxy		

注 \*1：地址计算公式中 x=A~F 对应 n=0~5

## 9.5 注意事项

请不要将同一功能设置到多个端口上。

使用模拟功能时，请将相应端口的数字功能关闭(DDIS=1)。

请在输出锁存有效时 (LTE=1) 进行端口功能切换，以避免切换期间端口上输出期待之外毛刺。

## 10 中断控制器 (INTC)

### 10.1 简介

中断控制器 (INTC) 选择中断事件作为中断请求送到 NVIC 唤醒 WFI；选择中断事件作为事件输入 (RXEV) 唤醒 WFE；选择中断事件唤醒系统低功耗模式（休眠模式和停止模式）；控制外部中断和软件中断。

INTC 的主要规格如下：

1. NVIC 中断请求：INTC 配备了 358 个中断事件，处理后作为中断请求 (IRQ) 发送给 NVIC，支持 184 个 IRQ，每个 IRQ 对应一个或多个中断事件。
2. 可编程优先级：16 个可编程优先级（使用了 4 位中断优先级寄存器）。
3. 不可屏蔽中断：可以独立选择多种系统中断事件作为不可屏蔽中断，且各中断事件配备独立的使能选择、标志及标志清除寄存器。
4. 配备 16 个外部中断事件。
5. 配备多个中断事件，具体数目请参考【中断事件表】。
6. 配备 32 个软件中断事件。
7. 中断可唤醒系统休眠模式和停止模式。

表 10-1 外部中断输入管脚

管脚名	I/O	说明
EIRQ0	输入	外部中断事件0
EIRQ1	输入	外部中断事件1
EIRQ2	输入	外部中断事件2
EIRQ3	输入	外部中断事件3
EIRQ4	输入	外部中断事件4
EIRQ5	输入	外部中断事件5
EIRQ6	输入	外部中断事件6
EIRQ7	输入	外部中断事件7
EIRQ8	输入	外部中断事件8
EIRQ9	输入	外部中断事件9
EIRQ10	输入	外部中断事件10
EIRQ11	输入	外部中断事件11
EIRQ12	输入	外部中断事件12
EIRQ13	输入	外部中断事件13
EIRQ14	输入	外部中断事件14
EIRQ15	输入	外部中断事件15

## 10.2 系统框图

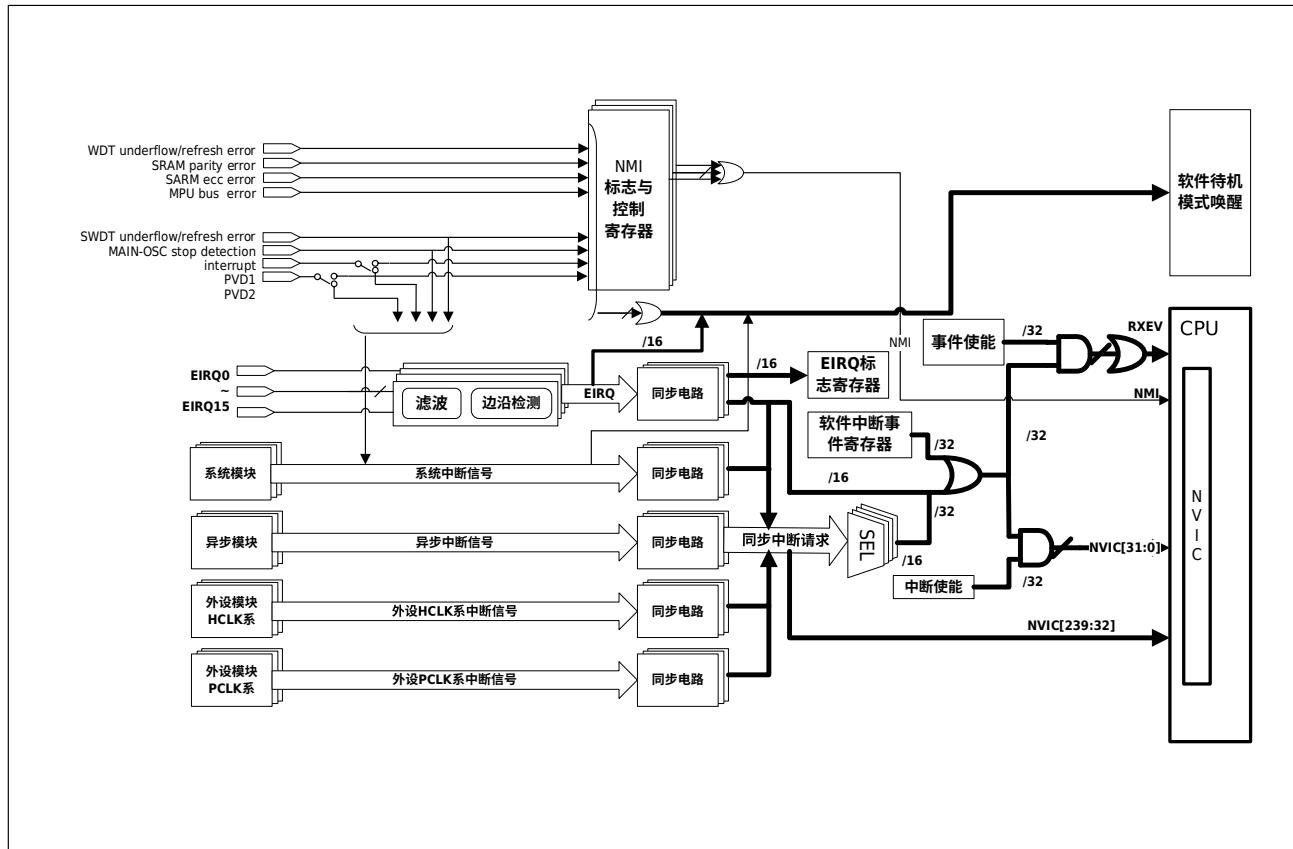


图 10-1 中断系统框图

## 10.3 向量表

### 10.3.1 中断向量表

表 10-2 中断向量表

向量地址	向量序号	IRQ序号	中断源	说明
ARM内核中断				
0x00000000	0	-	ARM core	Initial stack pointer
0x00000004	1	-	ARM core	Initial Program Counter
0x00000008	2	-	ARM core	Non-maskable Interrupt (NMI)
0x0000000C	3	-	ARM core	Hard Fault
0x00000010	4	-	ARM core	MemManage Fault
0x00000014	5	-	ARM core	Bus Fault
0x00000018	6	-	ARM core	Usage Fault
0x0000001C	7	-	ARM core	Reserved
0x00000020	8	-	ARM core	Reserved
0x00000024	9	-	ARM core	Reserved
0x00000028	10	-	ARM core	Reserved
0x0000002C	11	-	ARM core	Supervisor call (SVCall)
0x00000030	12	-	ARM core	Debug Monitor
0x00000034	13	-	ARM core	Reserved
0x00000038	14	-	ARM core	Pendable request for system service(PendableSrvReq)
0x0000003C	15	-	ARM core	System tick timer (SysTick)
非ARM内核中断				
0x00000040	16	0	INT_INTSEL0	寄存器INTC_INTSEL0选择的中断事件/软件中断0。
0x00000044	17	1	INT_INTSEL1	寄存器INTC_INTSEL1选择的中断事件/软件中断1。
0x00000048	18	2	INT_INTSEL2	寄存器INTC_INTSEL2选择的中断事件/软件中断2。
0x0000004C	19	3	INT_INTSEL3	寄存器INTC_INTSEL3选择的中断事件/软件中断3。
0x00000050	20	4	INT_INTSEL4	寄存器INTC_INTSEL4选择的中断事件/软件中断4。
0x00000054	21	5	INT_INTSEL5	寄存器INTC_INTSEL5选择的中断事件/软件中断5。
0x00000058	22	6	INT_INTSEL6	寄存器INTC_INTSEL6选择的中断事件/软件中断6。
0x0000005C	23	7	INT_INTSEL7	寄存器INTC_INTSEL7选择的中断事件/软件中断7。
0x00000060	24	8	INT_INTSEL8	寄存器INTC_INTSEL8选择的中断事件/软件中断8。
0x00000064	25	9	INT_INTSEL9	寄存器INTC_INTSEL9选择的中断事件/软件中断9。
0x00000068	26	10	INT_INTSEL10	寄存器INTC_INTSEL10选择的中断事件/软件中断10。
0x0000006C	27	11	INT_INTSEL11	寄存器INTC_INTSEL11选择的中断事件/软件中断11。
0x00000070	28	12	INT_INTSEL12	寄存器INTC_INTSEL12选择的中断事件/软件中断12。
0x00000074	29	13	INT_INTSEL13	寄存器INTC_INTSEL13选择的中断事件/软件中断13。

向量地址	向量序号	IRQ序号	中断源	说明
0x00000078	30	14	INTC_INTSEL14	寄存器INTC_INTSEL14选择的中断事件/软件中断14。
0x0000007C	31	15	INTC_INTSEL15	寄存器INTC_INTSEL15选择的中断事件/软件中断15。
0x00000080	32	16	PORT	外部中断0/软件中断16
0x00000084	33	17	PORT	外部中断1/软件中断17
0x00000088	34	18	PORT	外部中断2/软件中断18
0x0000008C	35	19	PORT	外部中断3/软件中断19
0x00000090	36	20	PORT	外部中断4/软件中断20
0x00000094	37	21	PORT	外部中断5/软件中断21
0x00000098	38	22	PORT	外部中断6/软件中断22
0x0000009C	39	23	PORT	外部中断7/软件中断23
0x000000A0	40	24	PORT	外部中断8/软件中断24
0x000000A4	41	25	PORT	外部中断9/软件中断25
0x000000A8	42	26	PORT	外部中断10/软件中断26
0x000000AC	43	27	PORT	外部中断11/软件中断27
0x000000B0	44	28	PORT	外部中断12/软件中断28
0x000000B4	45	29	PORT	外部中断13/软件中断29
0x000000B8	46	30	PORT	外部中断14/软件中断30
0x000000BC	47	31	PORT	外部中断15/软件中断31
0x000000C0	48	32	DMA_1	传输错误中断
0x000000C4	49	33	DMA_1	通道0传输完成中断/通道0数据块传输完成中断
0x000000C8	50	34	DMA_1	通道1传输完成中断/通道1数据块传输完成中断
0x000000CC	51	35	DMA_1	通道2传输完成中断/通道2数据块传输完成中断
0x000000D0	52	36	DMA_1	通道3传输完成中断/通道3数据块传输完成中断
0x000000D4	53	37	DMA_1	通道4传输完成中断/通道4数据块传输完成中断
0x000000D8	54	38	DMA_1	通道5传输完成中断/通道5数据块传输完成中断
0x000000DC	55	39	DMA_1	通道6传输完成中断/通道6数据块传输完成中断
0x000000E0	56	40	DMA_1	通道7传输完成中断/通道7数据块传输完成中断
0x000000E4	57	41	EFM	编程(擦除)错误中断/读写冲突错误中断
0x000000E8	58	42	EFM	操作结束中断
0x000000EC	59	43	QSPI	QSPI总线错误中断
0x000000F0	60	44	DCU_1	DCU_1中断
0x000000F4	61	45	DCU_2	DCU_2中断
0x000000F8	62	46	DCU_3	DCU_3中断
0x000000FC	63	47	DMA_2	传输错误中断
0x00000100	64	48	DMA_2	通道0传输完成中断/通道0数据块传输完成中断
0x00000104	65	49	DMA_2	通道1传输完成中断/通道1数据块传输完成中断
0x00000108	66	50	DMA_2	通道2传输完成中断/通道2数据块传输完成中断

向量地址	向量序号	IRQ序号	中断源	说明
0x0000010C	67	51	DMA_2	通道3传输完成中断/通道3数据块传输完成中断
0x00000110	68	52	DMA_2	通道4传输完成中断/通道4数据块传输完成中断
0x00000114	69	53	DMA_2	通道5传输完成中断/通道5数据块传输完成中断
0x00000118	70	54	DMA_2	通道6传输完成中断/通道6数据块传输完成中断
0x0000011C	71	55	DMA_2	通道7传输完成中断/通道7数据块传输完成中断
0x00000120	72	56	MAU	开方运算完成中断
0x00000124	73	57	FMAC_1	滤波运算完成中断
0x00000128	74	58	FMAC_2	滤波运算完成中断
0x0000012C	75	59	FMAC_3	滤波运算完成中断
0x00000130	76	60	FMAC_4	滤波运算完成中断
0x00000134	77	61	Timer0_1	通道A,B计数比较匹配中断/输入捕获中断
0x00000138	78	62	Timer0_2	通道A,B计数比较匹配中断/输入捕获中断
0x0000013C	79	63	Timer2_1	通道A,B计数比较匹配中断/输入捕获中断
0x00000140	80	64	Timer2_1	通道A,B计数溢出中断
0x00000144	81	65	Timer2_2	通道A,B计数比较匹配中断/输入捕获中断
0x00000148	82	66	Timer2_2	通道A,B计数溢出中断
0x0000014C	83	67	Timer2_3	通道A,B计数比较匹配中断/输入捕获中断
0x00000150	84	68	Timer2_3	通道A,B计数溢出中断
0x00000154	85	69	Timer2_4	通道A,B计数比较匹配中断/输入捕获中断
0x00000158	86	70	Timer2_4	通道A,B计数溢出中断
0x0000015C	87	71	RTC	闹钟中断/定周期中断
0x00000160	88	72	CMU	XTAL振荡故障中断
0x00000164	89	73	PWC	WKTM比较匹配中断
0x00000168	90	74	SWDT	计数下溢/刷新错误中断
0x0000016C	91	75	Timer6_1	通用计数比较匹配中断/输入捕获中断
0x00000170	92	76	Timer6_1	计数周期匹配中断
0x00000174	93	77	Timer6_1	死区时间错误中断
0x00000178	94	78	Timer6_1	专用计数比较匹配中断
0x0000017C	95	79	TimerA_1	计数周期匹配中断
0x00000180	96	80	TimerA_1	计数比较匹配中断
0x00000184	97	81	Timer6_2	通用计数比较匹配中断/输入捕获中断
0x00000188	98	82	Timer6_2	计数周期匹配中断
0x0000018C	99	83	Timer6_2	死区时间错误中断
0x00000190	100	84	Timer6_2	专用计数比较匹配中断
0x00000194	101	85	TimerA_2	计数周期匹配中断
0x00000198	102	86	TimerA_2	计数比较匹配中断
0x0000019C	103	87	Timer6_3	通用计数比较匹配中断/输入捕获中断

向量地址	向量序号	IRQ序号	中断源	说明
0x000001A0	104	88	Timer6_3	计数周期匹配中断
0x000001A4	105	89	Timer6_3	死区时间错误中断
0x000001A8	106	90	Timer6_3	专用计数比较匹配中断
0x000001AC	107	91	TimerA_3	计数周期匹配中断
0x000001B0	108	92	TimerA_3	计数比较匹配中断
0x000001B4	109	93	Timer6_4	通用计数比较匹配中断/输入捕获中断
0x000001B8	110	94	Timer6_4	计数周期匹配中断
0x000001BC	111	95	Timer6_4	死区时间错误中断
0x000001C0	112	96	Timer6_4	专用计数比较匹配中断
0x000001C4	113	97	TimerA_4	计数周期匹配中断
0x000001C8	114	98	TimerA_4	计数比较匹配中断
0x000001CC	115	99	Timer6_5	通用计数比较匹配中断/输入捕获中断
0x000001D0	116	100	Timer6_5	计数周期匹配中断
0x000001D4	117	101	Timer6_5	死区时间错误中断
0x000001D8	118	102	Timer6_5	专用计数比较匹配中断
0x000001DC	119	103	-	-
0x000001E0	120	104	Timer6_6	通用计数比较匹配中断/输入捕获中断
0x000001E4	121	105	Timer6_6	计数周期匹配中断
0x000001E8	122	106	Timer6_6	死区时间错误中断
0x000001EC	123	107	Timer6_6	专用计数比较匹配中断
0x000001F0	124	108	Timer6_7	通用计数比较匹配中断/输入捕获中断
0x000001F4	125	109	Timer6_7	计数周期匹配中断
0x000001F8	126	110	Timer6_7	死区时间错误中断
0x000001FC	127	111	Timer6_7	专用计数比较匹配中断
0x00000200	128	112	Timer6_8	通用计数比较匹配中断/输入捕获中断
0x00000204	129	113	Timer6_8	计数周期匹配中断
0x00000208	130	114	Timer6_8	死区时间错误中断
0x0000020C	131	115	Timer6_8	专用计数比较匹配中断
0x00000210	132	116	Timer6_9	通用计数比较匹配中断/输入捕获中断
0x00000214	133	117	Timer6_9	计数周期匹配中断
0x00000218	134	118	Timer6_9	死区时间错误中断
0x0000021C	135	119	Timer6_9	专用计数比较匹配中断
0x00000220	136	120	Timer6_10	通用计数比较匹配中断/输入捕获中断
0x00000224	137	121	Timer6_10	计数周期匹配中断
0x00000228	138	122	Timer6_10	死区时间错误中断
0x0000022C	139	123	Timer6_10	专用计数比较匹配中断
0x00000230	140	124	Timer4_1	通用计数比较匹配中断

向量地址	向量序号	IRQ序号	中断源	说明
0x00000234	141	125	Timer4_1	计数周期匹配中断
0x00000238	142	126	Timer4_1	重载计数匹配中断
0x0000023C	143	127	Timer4_1	专用计数比较匹配中断
0x00000240	144	128	CMP	CMP1中断
0x00000244	145	129	CMP	CMP2中断
0x00000248	146	130	CMP	CMP3中断
0x0000024C	147	131	CMP	CMP4中断
0x00000250	148	132	CAN_1	CAN_1 HOST中断
0x00000254	149	133	CAN_2	CAN_2 HOST中断
0x00000258	150	134	CAN_3	CAN_3 HOST中断
0x0000025C	151	135	USART_1	USART_1中断 (RX线唤醒停止模式中断除外)
0x00000260	152	136	USART_2	USART_2中断
0x00000264	153	137	SPI_1	SPI_1中断
0x00000268	154	138	SPI_2	SPI_2中断
0x0000026C	155	139	TimerA_5	计数周期匹配中断
0x00000270	156	140	TimerA_5	计数比较匹配中断
0x00000274	157	141	TimerA_6	计数周期匹配中断
0x00000278	158	142	TimerA_6	计数比较匹配中断
0x0000027C	159	143	USBFS	USBFS全局中断
0x00000280	160	144	PORT	Event Port1中断
0x00000284	161	145	PORT	Event Port2中断
0x00000288	162	146	PORT	Event Port3中断
0x0000028C	163	147	PORT	Event Port4中断
0x00000290	164	148	USART_3	USART_3中断
0x00000294	165	149	USART_4	USART_4中断
0x00000298	166	150	SPI_3	SPI_3中断
0x0000029C	167	151	SPI_4	SPI_4中断
0x000002A0	168	152	EMB	group0中断
0x000002A4	169	153	EMB	group1中断
0x000002A8	170	154	EMB	group2中断
0x000002AC	171	155	EMB	group3中断
0x000002B0	172	156	EMB	group4中断
0x000002B4	173	157	EMB	group5中断
0x000002B8	174	158	EMB	group6中断
0x000002BC	175	159	CPM	HASH中断
0x000002C0	176	160	USART_5	USART_5中断
0x000002C4	177	161	USART_6	USART_6中断

向量地址	向量序号	IRQ序号	中断源	说明
0x000002C8	178	162	MDIO	MDIO中断
0x000002CC	179	163	-	-
0x000002D0	180	164	I2C_1	I2C_1中断
0x000002D4	181	165	I2C_2	I2C_2中断
0x000002D8	182	166	I2C_3	I2C_3中断
0x000002DC	183	167	USART_1	RX线唤醒停止模式中断
0x000002E0	184	168	PWC	可编程电压检测1 (PVD1) 中断
0x000002E4	185	169	PWC	可编程电压检测1 (PVD2) 中断
0x000002E8	186	170	OTS	温度测量完成中断
0x000002EC	187	171	CMU	FCM时钟频率测量中断
0x000002F0	188	172	WDT	计数下溢/刷新错误中断
0x000002F4	189	173	CTC	HRC频率校准值溢出中断
0x000002F8	190	174	ADC_1	ADC_1中断
0x000002FC	191	175	ADC_2	ADC_2中断
0x00000300	192	176	ADC_3	ADC_3中断
0x00000304	193	177	CPM	TRNG中断
0x00000308	194	178	USART_1	USART_1_TCI中断
0x0000030C	195	179	USART_2	USART_2_TCI中断
0x00000310	196	180	USART_3	USART_3_TCI中断
0x00000314	197	181	USART_4	USART_4_TCI中断
0x00000318	198	182	USART_5	USART_5_TCI中断
0x0000031C	199	183	USART_6	USART_6_TCI中断

**注意：**

- INTC\_INTSEL 寄存器选择的中断事件编号请参考【中断事件表】。
- IRQ 序号 184~239 对应向量未配置任何中断事件，请勿使用。

### 10.3.2 中断事件表

中断事件由外设产生，作为 IRQ 输入时称为中断源；作为 RXEV 输入时称为事件源；作为外设触发事件时称为触发源；中断事件也可以作为从低功耗模式返回的条件。

表 10-3 中断事件表

序号	中 断 事 件 编 号	模块	中断名称	中 断 源	触 发 源	NVIC中断请求 (IRQ) 编号映射	
						类型A (INTC_INTSEL写入事件编号*1)	类型B (固定)
0	0x000	PORT	EIRQ0	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[16]
1	0x001		EIRQ1	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[17]
2	0x002		EIRQ2	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[18]
3	0x003		EIRQ3	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[19]
4	0x004		EIRQ4	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[20]
5	0x005		EIRQ5	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[21]
6	0x006		EIRQ6	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[22]
7	0x007		EIRQ7	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[23]
8	0x008		EIRQ8	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[24]
9	0x009		EIRQ9	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[25]
10	0x00A		EIRQ10	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[26]
11	0x00B		EIRQ11	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[27]
12	0x00C		EIRQ12	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[28]
13	0x00D		EIRQ13	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[29]
14	0x00E		EIRQ14	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[30]
15	0x00F		EIRQ15	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[31]
16	0x010	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
17	0x011	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
18	0x012	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
19	0x013	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
20	0x014	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
21	0x015	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
22	0x016	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
23	0x017	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
24	0x018	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
25	0x019	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
26	0x01A	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
27	0x01B	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
28	0x01C	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
29	0x01D	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
30	0x01E	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-

序号	中 断 事 件 编 号	模 块	中 断 名 称	中 断 源	触 发 源	NVIC中断请求（IRQ）编号映射	
						类型A (INTC_INTSEL写入事件编号*1)	类型B (固定)
31	0x01F	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
32	0x020	DMA_1	DMA_1_ERR	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[32]
33	0x021		DMA_1_TC0	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[33]
34	0x022		DMA_1_BTC0	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
35	0x023		DMA_1_TC1	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[34]
36	0x024		DMA_1_BTC1	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
37	0x025		DMA_1_TC2	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[35]
38	0x026		DMA_1_BTC2	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
39	0x027		DMA_1_TC3	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[36]
40	0x028		DMA_1_BTC3	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
41	0x029		DMA_1_TC4	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[37]
42	0x02A		DMA_1_BTC4	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
43	0x02B		DMA_1_TC5	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[38]
44	0x02C		DMA_1_BTC5	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
45	0x02D		DMA_1_TC6	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[39]
46	0x02E		DMA_1_BTC6	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
47	0x02F		DMA_1_TC7	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[40]
48	0x030		DMA_1_BTC7	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
49	0x031	EFM	EFM_PEERR	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[41]
50	0x032		EFM_RDCOL	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
51	0x033		EFM_OPTEND	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[42]
52	0x034	USBFS	USBFS_SOF	-	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	-
53	0x035	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
54	0x036	QSPI	QSPI_INTR	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[43]
55	0x037	DCU	DCU1	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[44]
56	0x038		DCU2	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[45]
57	0x039		DCU3	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[46]
58	0x03A	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
59	0x03B	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
60	0x03C	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
61	0x03D	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
62	0x03E	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
63	0x03F	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
64	0x040	DMA_2	DMA_2_ERR	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[47]
65	0x041		DMA_2_TC0	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[48]
66	0x042		DMA_2_BTC0	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑

序号	中断事件编号	模块	中断名称	中断源	触发源	NVIC中断请求(IRQ)编号映射	
						类型A (INTC_INTSEL写入事件编号*1)	类型B (固定)
67	0x043		DMA_2_TC1	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[49]
68	0x044		DMA_2_BTC1	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
69	0x045		DMA_2_TC2	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[50]
70	0x046		DMA_2_BTC2	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
71	0x047		DMA_2_TC3	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[51]
72	0x048		DMA_2_BTC3	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
73	0x049		DMA_2_TC4	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[52]
74	0x04A		DMA_2_BTC4	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
75	0x04B		DMA_2_TC5	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[53]
76	0x04C		DMA_2_BTC5	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
77	0x04D		DMA_2_TC6	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[54]
78	0x04E		DMA_2_BTC6	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
79	0x04F		DMA_2_TC7	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[55]
80	0x050		DMA_2_BTC7	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
81	0x051	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
82	0x052	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
83	0x053	MAU	MAU_SQRT	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[56]
84	0x054	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
85	0x055	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
86	0x056	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
87	0x057	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
88	0x058	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
89	0x059	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
90	0x05A	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
91	0x05B	FMAC	FMAC_1_FIR	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[57]
92	0x05C		FMAC_2_FIR	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[58]
93	0x05D		FMAC_3_FIR	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[59]
94	0x05E		FMAC_4_FIR	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[60]
95	0x05F	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
96	0x060	Timer0_1	TMR0_1_CMPA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[61]
97	0x061		TMR0_1_CMPB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
98	0x062	Timer0_2	TMR0_2_CMPA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[62]
99	0x063		TMR0_2_CMPB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
100	0x064	Timer2_1	TMR2_1_CMPA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[63]
101	0x065		TMR2_1_CMPB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
102	0x066		TMR2_1_OVFA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[64]

序号	中断事件编号	模块	中断名称	中断源	触发源	NVIC中断请求(IRQ)编号映射	
						类型A (INTC_INTSEL写入事件编号*1)	类型B (固定)
103	0x067		TMR2_1_OVFB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
104	0x068	Timer2_2	TMR2_2_CMPA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[65]
105	0x069		TMR2_2_CMPB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
106	0x06A		TMR2_2_OVFA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[66]
107	0x06B		TMR2_2_OVFB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
108	0x06C	Timer2_3	TMR2_3_CMPA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[67]
109	0x06D		TMR2_3_CMPB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
110	0x06E		TMR2_3_OVFA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[68]
111	0x06F		TMR2_3_OVFB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
112	0x070	Timer2_4	TMR2_4_CMPA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[69]
113	0x071		TMR2_4_CMPB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
114	0x072		TMR2_4_OVFA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[70]
115	0x073		TMR2_4_OVFB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
116	0x074	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
117	0x075	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
118	0x076	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
119	0x077	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
120	0x078	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
121	0x079	RTC	RTC_ALM	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[71]
122	0x07A		RTC_PRD	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
123	0x07B	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
124	0x07C	XTAL32	XTAL32_STOP	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
125	0x07D	XTAL	XTAL_STOP	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[72]
126	0x07E	WKT M	WKT M_PRD	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[73]
127	0x07F	SWDT	SWDT_REFUDF	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[74]
128	0x080	Timer6_1	TMR6_1_GCMA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[75]
129	0x081		TMR6_1_GCMB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
130	0x082		TMR6_1_GCMC	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
131	0x083		TMR6_1_GCMD	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
132	0x084		TMR6_1_GCME	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
133	0x085		TMR6_1_GCMF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
134	0x086		TMR6_1_GOVF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[76]
135	0x087		TMR6_1_GUDF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
136	0x088		TMR6_1_GDTE	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[77]
137	0x089		TMR6_1_SCMA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[78]
138	0x08A		TMR6_1_SCMB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑

序号	中断事件编号	模块	中断名称	中断源	触发源	NVIC中断请求(IRQ)编号映射	
						类型A (INTC_INTSEL写入事件编号*1)	类型B (固定)
139	0x08B	TimerA_1	TMRA_1_OVF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[79]
140	0x08C		TMRA_1_UDF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
141	0x08D		TMRA_1_CMP	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[80]
142	0x08E	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
143	0x08F	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
144	0x090	Timer6_2	TMR6_2_GCMA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[81]
145	0x091		TMR6_2_GCMB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
146	0x092		TMR6_2_GCMC	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
147	0x093		TMR6_2_GCMD	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
148	0x094		TMR6_2_GCME	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
149	0x095		TMR6_2_GCMF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
150	0x096		TMR6_2_GOVF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[82]
151	0x097		TMR6_2_GUDF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
152	0x098		TMR6_2_GDTE	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[83]
153	0x099		TMR6_2_SCMA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[84]
154	0x09A		TMR6_2_SCMB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
155	0x09B	TimerA_2	TMRA_2_OVF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[85]
156	0x09C		TMRA_2_UDF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
157	0x09D		TMRA_2_CMP	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[86]
158	0x09E	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
159	0x09F	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
160	0x0A0	Timer6_3	TMR6_3_GCMA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[87]
161	0x0A1		TMR6_3_GCMB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
162	0x0A2		TMR6_3_GCMC	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
163	0x0A3		TMR6_3_GCMD	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
164	0x0A4		TMR6_3_GCME	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
165	0x0A5		TMR6_3_GCMF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
166	0x0A6		TMR6_3_GOVF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[88]
167	0x0A7		TMR6_3_GUDF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
168	0x0A8		TMR6_3_GDTE	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[89]
169	0x0A9		TMR6_3_SCMA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[90]
170	0x0AA		TMR6_3_SCMB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
171	0x0AB	TimerA_3	TMRA_3_OVF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[91]
172	0x0AC		TMRA_3_UDF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
173	0x0AD		TMRA_3_CMP	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[92]
174	0x0AE	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-

序号	中断事件编号	模块	中断名称	中断源	触发源	NVIC中断请求(IRQ)编号映射	
						类型A (INTC_INTSEL写入事件编号*1)	类型B (固定)
175	0x0AF	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
176	0x0B0	Timer6_4	TMR6_4_GCMA	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[93]
177	0x0B1		TMR6_4_Gcmb	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
178	0x0B2		TMR6_4_Gcmc	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
179	0x0B3		TMR6_4_Gcmd	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
180	0x0B4		TMR6_4_Gcme	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
181	0x0B5		TMR6_4_Gcmf	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
182	0x0B6		TMR6_4_Govf	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[94]
183	0x0B7		TMR6_4_Gudf	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
184	0x0B8		TMR6_4_Gdte	/	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[95]
185	0x0B9		TMR6_4_Scma	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[96]
186	0x0BA		TMR6_4_Scmb	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
187	0x0BB	TimerA_4	TMRA_4_OVF	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[97]
188	0x0BC		TMRA_4_Udf	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
189	0x0BD		TMRA_4_Cmp	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[98]
190	0x0BE	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
191	0x0BF	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
192	0x0C0	Timer6_5	TMR6_5_GCMA	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[99]
193	0x0C1		TMR6_5_Gcmb	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
194	0x0C2		TMR6_5_Gcmc	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
195	0x0C3		TMR6_5_Gcmd	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
196	0x0C4		TMR6_5_Gcme	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
197	0x0C5		TMR6_5_Gcmf	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
198	0x0C6		TMR6_5_Govf	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[100]
199	0x0C7		TMR6_5_Gudf	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
200	0x0C8		TMR6_5_Gdte	/	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[101]
201	0x0C9		TMR6_5_Scma	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[102]
202	0x0CA		TMR6_5_Scmb	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
203	0x0CB	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[103]
204	0x0CC		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
205	0x0CD		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
206	0x0CE		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[103]
207	0x0CF	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
208	0x0D0	Timer6_6	TMR6_6_GCMA	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[104]
209	0x0D1		TMR6_6_Gcmb	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
210	0x0D2		TMR6_6_Gcmc	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑

序号	中 断 事 件 编 号	模 块	中 断 名 称	中 断 源	触 发 源	NVIC中断请求（IRQ）编号映射	
						类型A (INTC_INTSEL写入事件编号*1)	类型B (固定)
211	0x0D3	Timer6_6	TMR6_6_GCMD	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
212	0x0D4		TMR6_6_GCMF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
213	0x0D5		TMR6_6_GOVF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
214	0x0D6		TMR6_6_GUDF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[105]
215	0x0D7		TMR6_6_GDTE	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[106]
216	0x0D8		TMR6_6_SCMA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[107]
217	0x0D9		TMR6_6_SCMB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
218	0x0DA		-	-	-	-	-
219	0x0DB	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
220	0x0DC	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
221	0x0DD	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
222	0x0DE	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
223	0x0DF	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
224	0x0E0	Timer6_7	TMR6_7_GCMA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[108]
225	0x0E1		TMR6_7_GCMB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
226	0x0E2		TMR6_7_GCMC	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
227	0x0E3		TMR6_7_GCMD	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
228	0x0E4		TMR6_7_GCME	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
229	0x0E5		TMR6_7_GCMF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
230	0x0E6		TMR6_7_GOVF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[109]
231	0x0E7		TMR6_7_GUDF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
232	0x0E8		TMR6_7_GDTE	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[110]
233	0x0E9		TMR6_7_SCMA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[111]
234	0x0EA		TMR6_7_SCMB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
235	0x0EB	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
236	0x0EC	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
237	0x0ED	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
238	0x0EE	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
239	0x0EF	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
240	0x0F0	Timer6_8	TMR6_8_GCMA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[112]
241	0x0F1		TMR6_8_GCMB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
242	0x0F2		TMR6_8_GCMC	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
243	0x0F3		TMR6_8_GCMD	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
244	0x0F4		TMR6_8_GCME	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
245	0x0F5		TMR6_8_GCMF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
246	0x0F6		TMR6_8_GOVF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[113]

序号	中 断 事 件 编 号	模 块	中 断 名 称	中 断 源	触 发 源	NVIC中断请求（IRQ）编号映射	
						类型A (INTC_INTSEL写入事件编号*1)	类型B (固定)
247	0x0F7		TMR6_8_GUDF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
248	0x0F8		TMR6_8_GDTE	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[114]
249	0x0F9		TMR6_8_SCMA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[115]
250	0x0FA		TMR6_8_SCMB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
251	0x0FB	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
252	0x0FC	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
253	0x0FD	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
254	0x0FE	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
255	0x0FF	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
256	0x100	Timer6_9	TMR6_9_GCMA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[116]
257	0x101		TMR6_9_GCMB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
258	0x102		TMR6_9_GCMC	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
259	0x103		TMR6_9_GCMD	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
260	0x104		TMR6_9_GCME	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
261	0x105		TMR6_9_GCMF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
262	106		TMR6_9_GOVF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[117]
263	0x107		TMR6_9_GUDF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
264	0x108		TMR6_9_GDTE	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[118]
265	0x109		TMR6_9_SCMA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[119]
266	0x10A		TMR6_9_SCMB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
267	0x10B	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
268	0x10C	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
269	0x10D	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
270	0x10E	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
271	0x10F	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
272	0x110	Timer6_10	TMR6_10_GCMA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[120]
273	0x111		TMR6_10_GCMB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
274	0x112		TMR6_10_GCMC	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
275	0x113		TMR6_10_GCMD	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
276	0x114		TMR6_10_GCME	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
277	0x115		TMR6_10_GCMF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
278	0x116		TMR6_10_GOVF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[121]
279	0x117		TMR6_10_GUDF	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
280	0x118		TMR6_10_GDTE	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[122]
281	0x119		TMR6_10_SCMA	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[123]
282	0x11A		TMR6_10_SCMB	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑

序号	中断事件编号	模块	中断名称	中断源	触发源	NVIC中断请求(IRQ)编号映射	
						类型A (INTC_INTSEL写入事件编号*1)	类型B (固定)
283	0x11B	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
284	0x11C	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
285	0x11D	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
286	0x11E	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
287	0x11F	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
288	0x120	Timer4_1	TMR4_1_GCMUH	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[124]
289	0x121		TMR4_1_GCMUL	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
290	0x122		TMR4_1_GCMVH	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
291	0x123		TMR4_1_GCMVL	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
292	0x124		TMR4_1_GCMWH	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
293	0x125		TMR4_1_GCMWL	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
294	0x126		TMR4_1_GOVF	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[125]
295	0x127		TMR4_1_GUDF	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
296	0x128		TMR4_1_GRLU	/	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[126]
297	0x129		TMR4_1_GRLV	/	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
298	0x12A		TMR4_1_GRLW	/	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
299	0x12B		TMR4_1_SCM0	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[127]
300	0x12C		TMR4_1_SCM1	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
301	0x12D		TMR4_1_SCM2	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
302	0x12E		TMR4_1_SCM3	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
303	0x12F		TMR4_1_SCM4	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
304	0x130		TMR4_1_SCM5	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
305	0x131	CMP	CMP_1_IRQ	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[128]
306	0x132		CMP_2_IRQ	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[129]
307	0x133		CMP_3_IRQ	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[130]
308	0x134		CMP_4_IRQ	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[131]
309	0x135	CAN	CAN_1_INT	/	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[132]
310	0x136		CAN_2_INT	/	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[133]
311	0x137		CAN_3_INT	/	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[134]
312	0x138	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
313	0x139	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
314	0x13A	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
315	0x13B	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
316	0x13C	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
317	0x13D	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-

序号	中 断 事 件 编 号	模 块	中 断 名 称	中 断 源	触 发 源	NVIC中断请求（IRQ）编号映射	
						类型A (INTC_INTSEL写入事件编号*1)	类型B (固定)
318	0x13E	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
319	0x13F	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
320	0x140	USART_1	USART_1_EI	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[135]
321	0x141		USART_1_RI	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
322	0x142		USART_1_TI	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
323	0x143		USART_1_TCI	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[178]
324	0x144		USART_1_RTO	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[135]
325	0x145	USART_2	USART_2_EI	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[136]
326	0x146		USART_2_RI	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
327	0x147		USART_2_TI	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
328	0x148		USART_2_TCI	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[179]
329	0x149		USART_2_RTO	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[136]
330	0x14A	SPI_1	SPI_1_SPRI	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[137]
331	0x14B		SPI_1_SPTI	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
332	0x14C		SPI_1_SPII	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
333	0x14D		SPI_1_SPEI	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
334	0x14E		SPI_1_SPEND	-	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
335	0x14F	SPI_2	SPI_2_SPRI	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[138]
336	0x150		SPI_2_SPTI	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
337	0x151		SPI_2_SPII	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
338	0x152		SPI_2_SPEI	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
339	0x153		SPI_2_STEND	-	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
340	0x154	TimerA_5	TMRA_5_OVF	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[139]
341	0x155		TMRA_5_UDF	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
342	0x156		TMRA_5_CMP	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[140]
343	0x157	TimerA_6	TMRA_6_OVF	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[141]
344	0x158		TMRA_6_UDF	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
345	0x159		TMRA_6_CMP	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[142]
346	0x15A	USBFS	USBFS_GLB	/	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[143]
347	0x15B	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
348	0x15C	EVENT port	EVENT_PORT1	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[144]
349	0x15D		EVENT_PORT2	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[145]
350	0x15E		EVENT_PORT3	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[146]
351	0x15F		EVENT_PORT4	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[147]
352	0x160	USART_3	USART_3_BRKW_KPI	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[148]
353	0x161		USART_3_EI	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑

序号	中 断 事 件 编 号	模块	中断名称	中 断 源	触 发 源	NVIC中断请求（IRQ）编号映射	
						类型A (INTC_INTSEL写入事件编号*1)	类型B (固定)
354	0x162	USART_4	USART_3_RI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
355	0x163		USART_3_TI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
356	0x164		USART_3_TCI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[180]
357	0x165	USART_4	USART_4_EI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[149]
358	0x166		USART_4_RI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
359	0x167		USART_4_TI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
360	0x168		USART_4_TCI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[181]
361	0x169		USART_4_RTO	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[149]
362	0x16A	SPI_3	SPI_3_SPRI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[150]
363	0x16B		SPI_3_SPTI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
364	0x16C		SPI_3_SPII	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
365	0x16D		SPI_3_SPEI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
366	0x16E		SPI_3_SPEND	-	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
367	0x16F	SPI_4	SPI_4_SPRI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[151]
368	0x170		SPI_4_SPTI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
369	0x171		SPI_4_SPII	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
370	0x172		SPI_4_SPEI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
371	0x173		SPI_4_SPEND	-	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
372	0x174	EMB	EMB_GR0	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[152]
373	0x175		EMB_GR1	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[153]
374	0x176		EMB_GR2	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[154]
375	0x177		EMB_GR3	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[155]
376	0x178		EMB_GR4	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[156]
377	0x179		EMB_GR5	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[157]
378	0x17A		EMB_GR6	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[158]
379	0x17B	HASH	HASH_INT	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[159]
380	0x17C	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	
381	0x17D	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	
382	0x17E	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	
383	0x17F	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
384	0x180	USART_5	USART_5_EI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[160]
385	0x181		USART_5_RI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
386	0x182		USART_5_TI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
387	0x183		USART_5_TCI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[182]
388	0x184		USART_5_RTO	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[160]
389	0x185	USART_6	USART_6_BRKW KPI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[161]

序号	中断事件编号	模块	中断名称	中断源	触发源	NVIC中断请求(IRQ)编号映射	
						类型A (INTC_INTSEL写入事件编号*1)	类型B (固定)
390	0x186		USART_6_EI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
391	0x187		USART_6_RI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
392	0x188		USART_6_TI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
393	0x189		USART_6_TCI	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[183]
394	0x18A	USART_1	USART_1_TENDI	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[135]
395	0x18B	USART_2	USART_2_TENDI	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[136]
396	0x18C	USART_3	USART_3_TENDI	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[148]
397	0x18D	USART_4	USART_4_TENDI	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[149]
398	0x18E	USART_5	USART_5_TENDI	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[160]
399	0x18F	USART_6	USART_6_TENDI	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[161]
400	0x190	MDIO	MDIO_WRF	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[162]
401	0x191		MDIO_ADRF	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
402	0x192		MDIO_INCF	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
403	0x193		MDIO_RDF	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
404	0x194		MDIO_DEVM	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
405	0x195		MDIO_DEVN	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
406	0x196		MDIO_PHYM	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
407	0x197		MDIO_PHYN	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
408	0x198		MDIO_TANM	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
409	0x199		MDIO_TO	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
410	0x19A		MDIO_UDR	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
411	0x19B		MDIO_OVR	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
412	0x19C		MDIO_RBNE	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
413	0x19D		MDIO_AOS	-	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	-
414	0x19E	AOS	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
415	0x19F		AOS_STRG*2	-	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	-
416	0x1A0	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
417	0x1A1	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
418	0x1A2	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
419	0x1A3	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
420	0x1A4	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
421	0x1A5	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
422	0x1A6	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
423	0x1A7	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
424	0x1A8	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
425	0x1A9	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-

序号	中 断 事 件 编 号	模 块	中 断 名 称	中 断 源	触 发 源	NVIC中断请求(IRQ) 编号映射	
						类型A (INTC_INTSEL写入事件编号*1)	类型B (固定)
426	0x1AA	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
427	0x1AB	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
428	0x1AC	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
429	0x1AD	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
430	0x1AE	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
431	0x1AF	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[163]
432	0x1B0		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
433	0x1B1		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
434	0x1B2		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
435	0x1B3		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
436	0x1B4		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
437	0x1B5		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
438	0x1B6		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
439	0x1B7		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
440	0x1B8		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
441	0x1B9		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
442	0x1BA		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
443	0x1BB		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
444	0x1BC		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
445	0x1BD		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
446	0x1BE		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
447	0x1BF		-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
448	0x1C0	I2C_1 <sup>*4</sup>	I2C_1_RXI	✓	✓ <sup>*4</sup>	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[164]
449	0x1C1		I2C_1_TXI	✓	✓ <sup>*4</sup>	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
450	0x1C2		I2C_1_TEI	✓	✓ <sup>*4</sup>	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
451	0x1C3		I2C_1_EEI	✓	✓ <sup>*4</sup>	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
452	0x1C4	I2C_2 <sup>*4</sup>	I2C_2_RXI	✓	✓ <sup>*4</sup>	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[165]
453	0x1C5		I2C_2_TXI	✓	✓ <sup>*4</sup>	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
454	0x1C6		I2C_2_TEI	✓	✓ <sup>*4</sup>	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
455	0x1C7		I2C_2_EEI	✓	✓ <sup>*4</sup>	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
456	0x1C8	I2C_3 <sup>*4</sup>	I2C_3_RXI	✓	✓ <sup>*4</sup>	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[166]
457	0x1C9		I2C_3_TXI	✓	✓ <sup>*4</sup>	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
458	0x1CA		I2C_3_TEI	✓	✓ <sup>*4</sup>	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
459	0x1CB		I2C_3_EEI	✓	✓ <sup>*4</sup>	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
460	0x1CC	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
461	0x1CD	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-

序号	中 断 事 件 编 号	模 块	中 断 名 称	中 断 源	触 发 源	NVIC中断请求（IRQ）编号映射	
						类型A (INTC_INTSEL写入事件编号*1)	类型B (固定)
462	0x1CE	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
463	0x1CF	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
464	0x1D0	USART_1	USART_1_WUPI	/	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[167]
465	0x1D1	PVD	PVD_PVD1	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[168]
466	0x1D2		PVD_PVD2	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[169]
467	0x1D3	OTS	OTS	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[170]
468	0x1D4	FCM	FCMFERRI	/	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[171]
469	0x1D5		FCMMENDI	/	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
470	0x1D6		FCMCOVFI	/	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
471	0x1D7	WDT	WDT_REFUDF	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[172]
472	0x1D8	CTC	CTC_ERR	/	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[173]
473	0x1D9	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
474	0x1DA	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
475	0x1DB	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
476	0x1DC	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
477	0x1DD	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
478	0x1DE	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
479	0x1DF	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
480	0x1E0	ADC_1	ADC_1_EOCA	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[174]
481	0x1E1		ADC_1_EOCB	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
482	0x1E2		ADC_1_CMP0	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
483	0x1E3		ADC_1_CMP1	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
484	0x1E4	ADC_2	ADC_2_EOCA	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[175]
485	0x1E5		ADC_2_EOCB	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
486	0x1E6		ADC_2_CMP0	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
487	0x1E7		ADC_2_CMP1	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
488	0x1E8	ADC_3	ADC_3_EOCA	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[176]
489	0x1E9		ADC_3_EOCB	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
490	0x1EA		ADC_3_CMP0	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
491	0x1EB		ADC_3_CMP1	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
492	0x1EC	TRNG	TRNG_END	/	/	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[177]
493	0x1ED	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
494	0x1EE	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
495	0x1EF	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
496	0x1F0	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
497	0x1F1	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-

序号	中 断 事 件 编 号	模 块	中 断 名 称	中 断 源	触 发 源	NVIC中断请求（IRQ）编号映射	
						类型A (INTC_INTSEL写入事件编号*1)	类型B (固定)
498	0x1F2	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
499	0x1F3	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
500	0x1F4	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
501	0x1F5	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
502	0x1F6	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
503	0x1F7	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
504	0x1F8	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
505	0x1F9	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
506	0x1FA	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
507	0x1FB	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
508	0x1FC	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
509	0x1FD	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
510	0x1FE	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
511	0x1FF	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-

\*1: 中断选择寄存器选择的编号如果未配置中断事件，则该寄存器设定无效。

\*2: AOS\_STRG 由软件设定寄存器 INTSFTTRG 的 STRG 位产生。

\*3: “↑” 表示 IRQ 序号同上。

\*4: I2C\_1/ I2C\_2/ I2C\_3 的事件作为 AOS 源时仅可以触发 DMA 模块动作，不能触发其他 AOS 目标模块。

## 10.4 功能说明

### 10.4.1 不可屏蔽中断

不可屏蔽中断事件如下：

- 检测主发振器停止中断
- WDT 下溢/刷新中断
- SWDT 下溢/刷新中断
- 低电压检测 PVD1 中断
- 低电压检测 PVD2 中断
- SRAM 奇偶校验错误中断
- SRAM ECC 校验错误中断
- MPU 总线错误中断

不可屏蔽中断具有最高的优先级并且可以选择多个中断事件。各中断事件的状态可以通过中断标志寄存器（INTC\_NMIFR）查询。请在不可屏蔽中断处理退出前确认所有的标志位都为“0”。

需要使用不可屏蔽中断时，请按照如下流程设定：

1. 为不可屏蔽中断事件配置相应功能。
2. 对 INTC\_NMIFCR 各位写“1”，清除中断标志位，防止误动作。
3. 设定中断使能寄存器 INTC\_NMIER，使能相应的中断事件。

**注意：**

- 一旦 INTC\_NMIER 的相应位被设定为“1”，将不可更改，除非用 RESET 来复位。

### 10.4.2 外部中断

配备了 16 个外部中断事件，中断状态可以通过中断标志寄存器（INTC\_EIFR）查询。需要使用外部中断时，请按照如下流程设定：

1. 为相应的外部管脚设定外部中断功能。
2. 对 INTC{EIFCR} 各位写“1”，清除中断标志位，防止误动作。
3. 清除 INTC\_EIRQCRx.EFEN 位 ( $x=0\sim15$ )，禁止数字滤波器。
4. 设定 INTC\_EIRQCRx.EIRQTRG[1:0]位，选择触发边沿或电平；设定 EISMPCLK[1:0]位，选择数字滤波器采样时钟；设定 EFEN 位，使能数字滤波器。

#### 10.4.3 中断事件多路器

中断控制器为编号 0~15 的 NIVC 中断请求各配置了一个 512 选 1 的中断事件多路器。通过将中断事件编号写入相应的中断事件选择寄存器 INTC\_INTSEL0~15，可以在所有的中断事件中任意选择一个事件作为该中断请求的中断源。具体的中断事件编号请参考【中断事件表】章节。使用 NIVC 中断请求 0~15 时可将中断事件屏蔽寄存器 INTC\_INTEN 中对应的中断事件屏蔽位置 0，这样可以避免同一中断事件产生两次中断。

#### 10.4.4 软件中断

软件中断功能可以通过直接写软件中断寄存器 INTC\_SWIER 发生一次中断请求。一共配备了 32 个软件中断，对应的 NVIC 中断请求编号为 0~31，由中断使能寄存器 INTC\_IER 控制许可。

## 10.4.5 事件输入

编号为 0~31 的 NIVC 中断请求也可被用作事件输入 (RXEV) 来唤醒内核 (WFE)，

通过事件使能寄存器 INTC\_EVTER 控制许可。

中断请求和事件输入的功能框图如下：

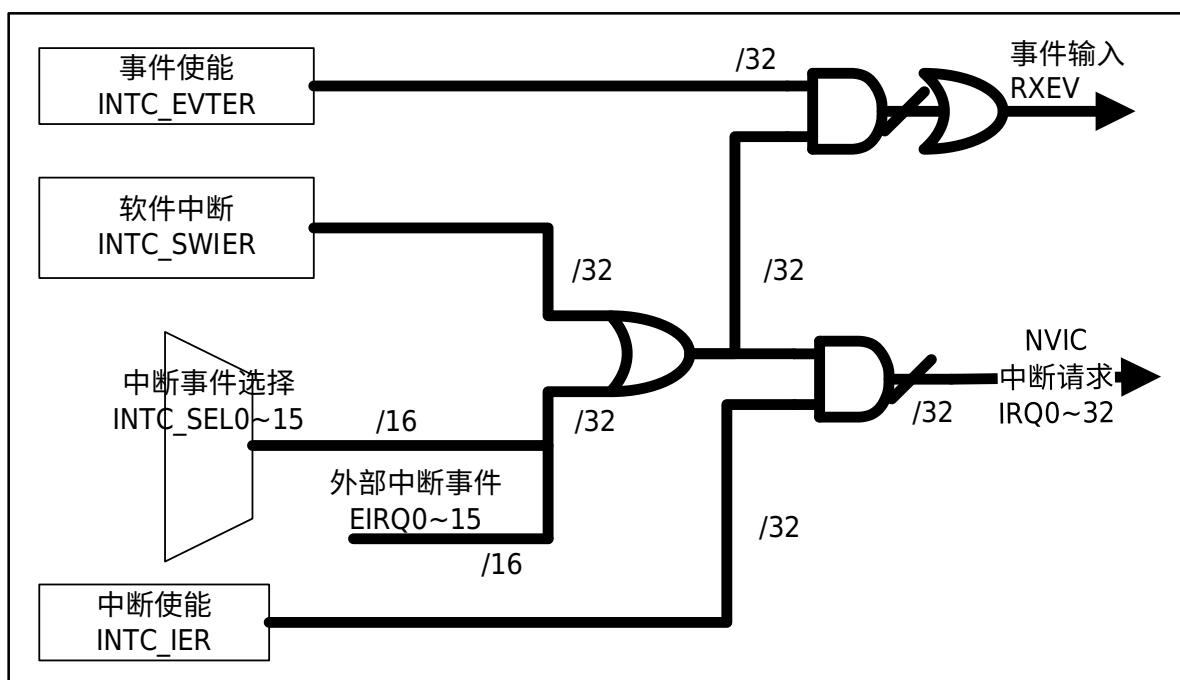


图 10-2 中断请求/事件输入选择

#### 10.4.6 WFE 事件管理

中断控制器可以通过中断请求、事件输入两种方式唤醒内核（WFE）。

- 中断请求唤醒内核

选择一个中断事件，根据选择的中断请求设定 INTC\_INTSEL 并使能 INT\_IER 寄存器（但不在 NVIC 中使能该中断请求）。INTC\_EVTER 寄存器设定为非使能，同时使能 Cortex-M4 内核控制寄存器 SCR 中的 SEVONPEND 位。执行 WFE 使内核进入休眠模式。当选择的中断事件发生时，内核由休眠模式被唤醒，但不进入中断处理程序。

- 事件输入唤醒内核

选择一个中断事件，根据选择的中断请求设定 INTC\_INTSEL 并使能 INTC\_EVTER 寄存器。INTC\_IER 寄存器设定为非使能。执行 WFE 使内核进入休眠模式。当选择的中断事件发生时，内核由休眠模式被唤醒。

#### 10.4.7 噪声滤波器

为了消除输入噪声，外部中断 EIRQx ( $x=0\sim15$ ) 配置了数字和模拟两种滤波器。

数字滤波器通过 INTC\_EIRQCRx 寄存器的 EFEN 位使能。使用时以 PCLK3 作为采样时钟对 EIRQ 输入信号进行采样，宽度小于 3 个采样周期的信号将被滤除。采样周期通过 INTC\_EIRQCRx 寄存器的 EISMPCLK[1:0]位设定。

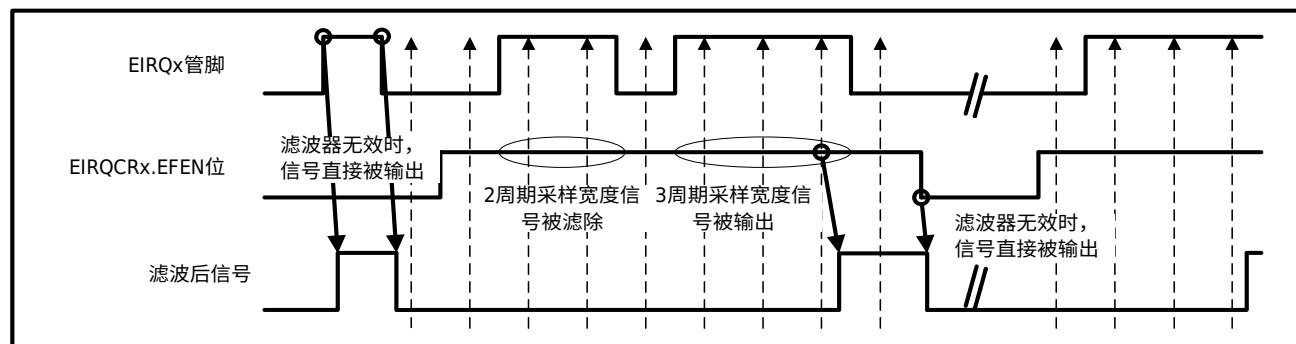


图 10-3 数字滤波器工作示意图

模拟滤波器通过 INTC\_EIRQCRx 寄存器的 NOCEN 位使能。使能后以 INTC\_EIRQCR 寄存器的 NOCSEL 位选择的滤波宽度对信号进行滤波。

两个滤波器彼此独立，可单独使能或者关闭。同时打开时，信号是先经过模拟滤波再经过数字滤波后进入到内部电路的。

在进入停止模式前，必须关闭数字滤波器。从停止模式返回后可再使能数字滤波器。模拟滤波器可以在停止模式时工作。

## 10.4.8 低功耗模式返回

### 10.4.8.1 休眠模式返回

通过中断请求唤醒睡眠模式时，需要进行如下设定：

- 选择一个中断事件作为中断请求  
如果使用 0~32 号中断请求需要设定 INTC\_IER 寄存器
- 在 NVIC 中使能该中断请求
- 使用不可屏蔽中断时需要设定 INTC\_NMIER 寄存器

### 10.4.8.2 停止模式返回

通过中断请求唤醒停止模式时，需要进行如下设定：

- 选择一个中断事件作为中断请求  
如果使用 0~32 号中断请求需要设定 INTC\_IER 寄存器。
- 通过 INTC\_WKEN 寄存器使能该中断事件  
如果使用不可屏蔽中断则通过 INTC\_NMIER 寄存器使能中断事件，可以唤醒停止模式的不可屏蔽中断有：检测主振荡器停止中断、SWDT 下溢/刷新中断、低电压检测 PVD1/PVD2 中断。
- 在 NVIC 中使能该中断请求

### 10.4.8.3 掉电模式返回

掉电模式的返回可由电源控制（PWC）章节所表示的条件、RES#管脚复位、上电复位以及欠压检测条件返回。返回后 CPU 进入复位中断处理。详细说明请参考【电源控制（PWC）】章节。

### 10.4.8.4 不可屏蔽中断和 WFI 指令

执行 WFI 指令前，请确认不可屏蔽中断标志寄存器 INTC\_NMIFR 的所有状态位为“0”。

## 10.5 寄存器说明

表 10-4 INTC 寄存器列表

INTC 基地址:0x40051000

寄存器名	符号	偏移地址	位宽	复位值
不可屏蔽中断使能寄存器	INTC_NMIER	0x0004	32	0x00000000
不可屏蔽中断标志寄存器	INTC_NMIFR	0x0008	32	0x00000000
不可屏蔽中断标志清除寄存器	INTC_NMIFCR	0x000C	32	0x00000000
外部中断控制寄存器0	INTC_EIRQCR0	0x0010	32	0x00000000
外部中断控制寄存器1	INTC_EIRQCR1	0x0014	32	0x00000000
外部中断控制寄存器2	INTC_EIRQCR2	0x0018	32	0x00000000
外部中断控制寄存器3	INTC_EIRQCR3	0x001C	32	0x00000000
外部中断控制寄存器4	INTC_EIRQCR4	0x0020	32	0x00000000
外部中断控制寄存器5	INTC_EIRQCR5	0x0024	32	0x00000000
外部中断控制寄存器6	INTC_EIRQCR6	0x0028	32	0x00000000
外部中断控制寄存器7	INTC_EIRQCR7	0x002C	32	0x00000000
外部中断控制寄存器8	INTC_EIRQCR8	0x0030	32	0x00000000
外部中断控制寄存器9	INTC_EIRQCR9	0x0034	32	0x00000000
外部中断控制寄存器10	INTC_EIRQCR10	0x0038	32	0x00000000
外部中断控制寄存器11	INTC_EIRQCR11	0x003C	32	0x00000000
外部中断控制寄存器12	INTC_EIRQCR12	0x0040	32	0x00000000
外部中断控制寄存器13	INTC_EIRQCR13	0x0044	32	0x00000000
外部中断控制寄存器14	INTC_EIRQCR14	0x0048	32	0x00000000
外部中断控制寄存器15	INTC_EIRQCR15	0x004C	32	0x00000000
停止模式唤醒事件使能寄存器	INTC_WKEN	0x0050	32	0x00000000
外部中断标志寄存器	INTC{EIFR}	0x0054	32	0x00000000
外部中断标志清除寄存器	INTC{EIFCR}	0x0058	32	0x00000000
中断事件选择寄存器0	INTC_INTSEL0	0x005C	32	0x000001FF
中断事件选择寄存器1	INTC_INTSEL1	0x0060	32	0x000001FF
中断事件选择寄存器2	INTC_INTSEL2	0x0064	32	0x000001FF
中断事件选择寄存器3	INTC_INTSEL3	0x0068	32	0x000001FF
中断事件选择寄存器4	INTC_INTSEL4	0x006C	32	0x000001FF
中断事件选择寄存器5	INTC_INTSEL5	0x0070	32	0x000001FF
中断事件选择寄存器6	INTC_INTSEL6	0x0074	32	0x000001FF
中断事件选择寄存器7	INTC_INTSEL7	0x0078	32	0x000001FF
中断事件选择寄存器8	INTC_INTSEL8	0x007C	32	0x000001FF
中断事件选择寄存器9	INTC_INTSEL9	0x0080	32	0x000001FF
中断事件选择寄存器10	INTC_INTSEL10	0x0084	32	0x000001FF

寄存器名	符号	偏移地址	位宽	复位值
中断事件选择寄存器11	INTC_INTSEL11	0x0088	32	0x000001FF
中断事件选择寄存器12	INTC_INTSEL12	0x008C	32	0x000001FF
中断事件选择寄存器13	INTC_INTSEL13	0x0090	32	0x000001FF
中断事件选择寄存器14	INTC_INTSEL14	0x0094	32	0x000001FF
中断事件选择寄存器15	INTC_INTSEL15	0x0098	32	0x000001FF
中断事件屏蔽寄存器0	INTC_INTEN0	0x025C	32	0xFFFFFFFF
中断事件屏蔽寄存器1	INTC_INTEN1	0x0260	32	0xFFFFFFFF
中断事件屏蔽寄存器2	INTC_INTEN2	0x0264	32	0xFFFFFFFF
中断事件屏蔽寄存器3	INTC_INTEN3	0x0268	32	0xFFFFFFFF
中断事件屏蔽寄存器4	INTC_INTEN4	0x026C	32	0xFFFFFFFF
中断事件屏蔽寄存器5	INTC_INTEN5	0x0270	32	0xFFFFFFFF
中断事件屏蔽寄存器6	INTC_INTEN6	0x0274	32	0xFFFFFFFF
中断事件屏蔽寄存器7	INTC_INTEN7	0x0278	32	0xFFFFFFFF
中断事件屏蔽寄存器8	INTC_INTEN8	0x027C	32	0xFFFFFFFF
中断事件屏蔽寄存器9	INTC_INTEN9	0x0280	32	0xFFFFFFFF
中断事件屏蔽寄存器10	INTC_INTEN10	0x0284	32	0xFFFFFFFF
中断事件屏蔽寄存器11	INTC_INTEN11	0x0288	32	0xFFFFFFFF
中断事件屏蔽寄存器12	INTC_INTEN12	0x028C	32	0xFFFFFFFF
中断事件屏蔽寄存器13	INTC_INTEN13	0x0290	32	0xFFFFFFFF
中断事件屏蔽寄存器14	INTC_INTEN14	0x0294	32	0xFFFFFFFF
中断事件屏蔽寄存器15	INTC_INTEN15	0x0298	32	0xFFFFFFFF
软件中断寄存器	INTC_SWIER	0x029C	32	0x00000000
事件使能寄存器	INTC_EVTER	0x02A0	32	0x00000000
中断使能寄存器	INTC_IER	0x02A4	32	0xFFFFFFFF

### 10.5.1 不可屏蔽中断使能寄存器 (INTC\_NMIER)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved		WDTEN	BUSERREN	RECERREN	RPARERREN	Reserved		XATLSTPEN	Reserve	PVD2EN	PVD1EN	SWDTEN	Reserve		

位	标记	位名	功能	读写
b31~b12	Reserved	-	读出时为“0”，写入时写“0”	R/W
b11	WDTEN	WDT下溢/刷新中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	R/W
b10	BUSERREN	MPU主总线错误中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	R/W
b9	RECCERREN	SRAM ECC校验错误中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	R/W
b8	RPARERREN	SRAM奇偶校验错误中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	R/W
b7~b6	Reserved	-	读出时为“0”，写入时写“0”	R/W
b5	XATLSTPEN	检测主发振器停止中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	R/W
b4	Reserved	-	读出时为“0”，写入时写“0”	R/W
b3	PVD2EN	低电压检测PVD2中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	R/W
b2	PVD1EN	低电压检测PVD1中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	R/W
b1	SWDTEN	SWDT下溢/刷新中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	R/W
b0	Reserved	-	读出时为“0”，写入时写“0”	R/W

### 10.5.2 不可屏蔽中断标志寄存器 (INTC\_NMIFR)

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved	WDT F	BUS ERRF	REC CERR F	RPAR ERRF	Reserved	XTAL STPF	Rese rved	PVD 2F	PVD 1F	SWD TF	Rese rved				
<hr/>															
位	标记	位名	功能										读写		
b31~b12	Reserved	-	读出时为“0”，写入时写“0”										R/W		
b11	WDTF	WDT下溢/刷新中断标志	0: 没有发生WDT下溢/刷新申请 1: 发生WDT下溢/刷新申请										R		
b10	BUSERRF	MPU主总线错误中断标志	0: 没有发生MPU主总线错误申请 1: 发生MPU主总线错误申请										R		
b9	RECCERRF	SRAM DED校验错误中断标志	0: 没有发生SRAM DED校验错误申请 1: 发生SRAM DED校验错误申请										R		
b8	RPARERRF	SRAM奇偶校验错误中断标志	0: 没有发生SRAM奇偶校验错误申请 1: 发生SRAM奇偶校验错误申请										R		
b7~b6	Reserved	-	读出时为“0”，写入时写“0”										R/W		
b5	XTALSTPF	检测主发振器停止中断标志	0: 没有发生检测主发振器停止申请 1: 发生检测主发振器停止申请										R		
b4	Reserved	-	读出时为“0”，写入时写“0”										R/W		
b3	PVD2F	低电压检测PVD2中断标志	0: 没有发生低电压检测PVD2申请 1: 发生低电压检测PVD2申请										R		
b2	PVD1F	低电压检测PVD1中断标志	0: 没有发生低电压检测PVD1申请 1: 发生低电压检测PVD1申请										R		
b1	SWDTF	SWDT下溢/刷新中断标志	0: 没有发生SWDT下溢/刷新申请 1: 发生SWDT下溢/刷新申请										R		
b0	Reserved	-	读出时为“0”，写入时写“0”										R/W		

### 10.5.3 不可屏蔽中断标志清除寄存器 (INTC\_NMIFCR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved	WDT FCLR	BUS ERRF CLR	REC CERR FCLR	RPAR ERRF CLR	Reserved	XTAL STPF CLR	Rese rved	PVD 2FCL R	PVD 1FCL R	SWD TFCL R	Rese rved				
<hr/>															
位	标记	位名	功能										读写		
b31~b12	Reserved	-	读出时为“0”，写入时写“0”										R/W[注1]		
b11	WDTFCLR	WDT下溢/刷新中断标志清除	0: 无效 1: 清除WDT下溢/刷新标志										R/W[注1]		
b10	BUSERRFCLR	MPU主总线错误中断标志清除	0: 无效 1: 清除MPU主总线错误标志										R/W[注1]		
b9	RECCERRFCLR	SRAM DED校验错误中断标志清除	0: 无效 1: 清除SRAM DED校验错误标志										R/W[注1]		
b8	RPARERRFCLR	SRAM奇偶校验错误中断标志清除	0: 无效 1: 清除SRAM奇偶校验错误标志										R/W[注1]		
b7~b6	Reserved	-	读出时为“0”，写入时写“0”										R/W[注1]		
b5	XTALSTPFCLR	检测主发振器停止中断标志清除	0: 无效 1: 清除检测主发振器停止标志										R/W[注1]		
b4	Reserved	-	读出时为“0”，写入时写“0”										R/W		
b3	PVD2FCLR	低电压检测PVD2中断标志清除	0: 无效 1: 清除低电压检测PVD2标志										R/W[注1]		
b2	PVD1FCLR	低电压检测PVD1中断标志清除	0: 无效 1: 清除低电压检测PVD1标志										R/W[注1]		
b1	SWDTFCLR	SWDT下溢/刷新中断标志清除	0: 无效 1: 清除SWDT下溢/刷新标志										R/W[注1]		
b0	Reserved	-	读出时为“0”，写入时写“0”										R/W[注1]		

[注 1] 只可写“1”，读出时为“0”。

### 10.5.4 外部中断控制寄存器(INTC\_EIRQCRx)(x=0~15)

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
NOC EN	Rese rved	NOCSEL[1:0]						EFEN	Rese rved	EISMPCLK [1:0]		Reserved		EIRQTRG[1:0]	
<hr/>															
位	标记	位名	功能											读写	
b31~b16	Reserved	-	读出时为“0”，写入时写“0”											R/W	
b15	NOCEN	EIRQ模拟滤波器使能	0: 禁止模拟滤波器功能 1: 允许模拟滤波器功能 滤波宽度由INTC_NOCCR.NOCSEL决定。											R/W	
b14	Reserved	-	读出时为“0”，写入时写“0”											R/W	
b13~b12	NOCSEL[1:0]	模拟滤波滤波宽度选择	0 0: 滤波宽度档位1 0 1: 滤波宽度档位2 1 0: 滤波宽度档位3 1 1: 滤波宽度档位4 各档位具体值请参考数据手册电气特性中EIRQ滤波特性章节。											R/W	
b11~b8	Reserved	-	读出时为“0”，写入时写“0”											R/W	
b7	EFEN	EIRQ数字滤波器使能	0: 禁止数字滤波器功能 1: 允许数字滤波器功能											R/W	
b6	Reserved	-	读出时为“0”，写入时写“0”											R/W	
b5~b4	EISMPCLK[1:0]	数字滤波采样周期选择	0 0: 1*PCLK3周期 0 1: 8*PCLK3周期 1 0: 32*PCLK3周期 1 1: 64*PCLK3周期											R/W	
b3~b2	Reserved	-	读出时为“0”，写入时写“0”											R/W	
b1~b0	EIRQTRG[1:0]	触发选择	0 0: 下降沿 0 1: 上升沿 1 0: 双边沿 1 1: 低电平											R/W	

### 10.5.5 外部中断标志寄存器(INTC\_EIFR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIF[15:0]															
位	标记	位名	功能										读写		
b31~b16	Reserved	-	读出时为“0”，写入时写“0”										R/W		
b15~b0	EIF	EIRQ标志位	0: EIRQ事件未发生，或写EIFCR位清除位 1: 选择的EIRQ事件发生										R		

### 10.5.6 外部中断标志清除寄存器(INTC{EIFCR})

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIFCLR[15:0]															
位	标记	位名	功能										读写		
b31~b16	Reserved	-	读出时为“0”，写入时写“0”										R/W		
b15~b0	EIFCLR	EIRQ标志清除位	0: 写“0”无效 1: 写“1”清除EIFR寄存器										R/W		

### 10.5.7 中断事件选择寄存器 (INTC\_INTSEL0~15)

复位值: 0x0000 01FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved									INTSEL[8:0]						
<hr/>															
位	标记	位名	功能	读写											
b31~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W											
b8~b0	INTSEL[8:0]	中断事件请求选择	9'h000~9'h1FE: 10.3.2中断事件表所对应的事件	R/W											

### 10.5.8 中断事件屏蔽寄存器 (INTC\_INTEN0~15)

复位值: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
INTEN[x*32+31:x*32+16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
INTEN[x*32+15:x*32]															
<hr/>															
位	标记	位名	功能	读写											
b31~b0	INTEN [x*32+31:x*32] x=0~15	中断使能	INTEN[511:0]与中断事件序号一一对应，设为“0”的中断事件将不能被 编号15以后的NVIC中断请求接收。 0: 中断事件无效 1: 中断事件有效 与未配置中断事件的序号对应的位为只读位，读出时为“1”。	R/W											

### 10.5.9 停止模式唤醒事件使能寄存器 (INTC\_WKEN)

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CMP4WKEN	CMP3WKEN	CMP2WKEN	USFWKEN	-	RXWKEN	TMR2OVFWKEN	TMR2CMPWKEN	TMR0OCMPWKEN	RTCPRDWKEN	RTCALMWKEN	WKTWMWKEN	CMP1WKEN	PVD2WKEN	PVD1WKEN	SWDTWKEN
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIRQWKEN[15:0]															

位	标记	位名	功能	读写
b31	CMP4WKEN	CMP4停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b30	CMP3WKEN	CMP3停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b29	CMP2WKEN	CMP2停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b28	USFWKEN	USBFS_WKUP停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b27	Reserved		读出时为“0”，写入时写“0”	R/W
b26	RXWKEN	USART_1_RX线停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b25	TMR2OVFWKEN	TMR2_1_OVFA停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b24	TMR2CMPWKEN	TMR2_1_CMPA停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b23	TMR0OCMPWKEN	TMR0_1_CMPA停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b22	RTCPRDWKEN	RTC_PRD停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b21	RTCALMWKEN	RTC_ALM停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b20	WKTWMWKEN	WKT_MWP停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b19	CMP1WKEN	CMP1停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b18	PVD2WKEN	PVD2停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b17	PVD1WKEN	PVD1停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b16	SWDTWKEN	SWDT_REFUDF停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b15~b0	EIRQWKEN[15:0]	EIRQ停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W

### 10.5.10 软件中断寄存器 (INTC\_SWIER)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SW IE[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SW IE[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b0	SWIE	软件中断	0: 无效 1: 软件中断发生	R/W											
注意: 由"0"写"1"后发生一次软件中断。再次发生必须先写"0"清除。															

### 10.5.11 事件使能寄存器 (INTC\_EVTER)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EVTE[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EVTE[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b0	EVTE	事件使能	0: 事件选择禁止 1: 事件选择许可	R/W											

### 10.5.12 中断使能寄存器 (INTC\_IER)

复位值: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IEN[31:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
IEN[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b0	IEN	中断使能	寄存器位0~31分别对应NVIC中断请求0~31。禁止时，各中断请求对应的中断事件将不能被NVIC接收。 0: 中断禁止 1: 中断许可	R/W											

## 10.6 使用注意事项

关于 ARM 核中断的描述, 请参考 ARM 手册 ARM Processor Cortex®-M4 Technical Reference Manual (ARM DDI 0439D)。

## 11 自动运行系统 (AOS)

### 11.1 简介

自动运行系统 (Automatic Operation System) 用于在不借助 CPU 的情况下实现外设硬件电路之间的联动。利用外设电路产生的事件作为 AOS 源 (AOS Source)，如定时器的比较匹配、定时溢出，RTC 的周期信号、通信模块的收发数据的各种状态（空闲，接收数据满，发送数据结束，发送数据空），ADC 的转换结束等，来触发其他外设电路动作。被触发的外设电路动作称为 AOS 目标 (AOS Target)。

#### 11.1.1 功能概览

- 共有 322 种 AOS 源，除特殊限制外，每个 AOS 目标可选择其中一个作为触发源。另外还可以通过公共触发源选择寄存器 1 和公共触发源选择寄存器 2 选择两个公共触发源，所有 AOS 目标共享这两个公共触发源。这样，对于一个 AOS 目标来说，3 个触发源中的任意一个发生触发事件时均可触发该 AOS 目标动作。
- 可以由外设电路硬件触发，也可通过写寄存器由软件触发。
- 能够作为 AOS 目标的外设及数量如下：
  - 3 个 DCU 触发目标
  - 17 个 DMA 触发目标
  - 4 个高级控制定时器(Timer6) 触发目标
  - 1 个通用控制定时器 4(Timer4) 触发目标
  - 1 个通用定时器 0(Timer0) 触发目标
  - 1 个通用定时器 2(Timer0) 触发目标
  - 4 个通用定时器 A(TimerA) 触发目标
  - 2 个 Event Port 触发目标
  - 2 个 HASH 触发目标
  - 1 个温度传感器(OTS) 触发目标
  - 3 组每组 2 个 ADC 触发目标，用于 ADC1~ADC3 序列触发

### 11.1.2 模块框图

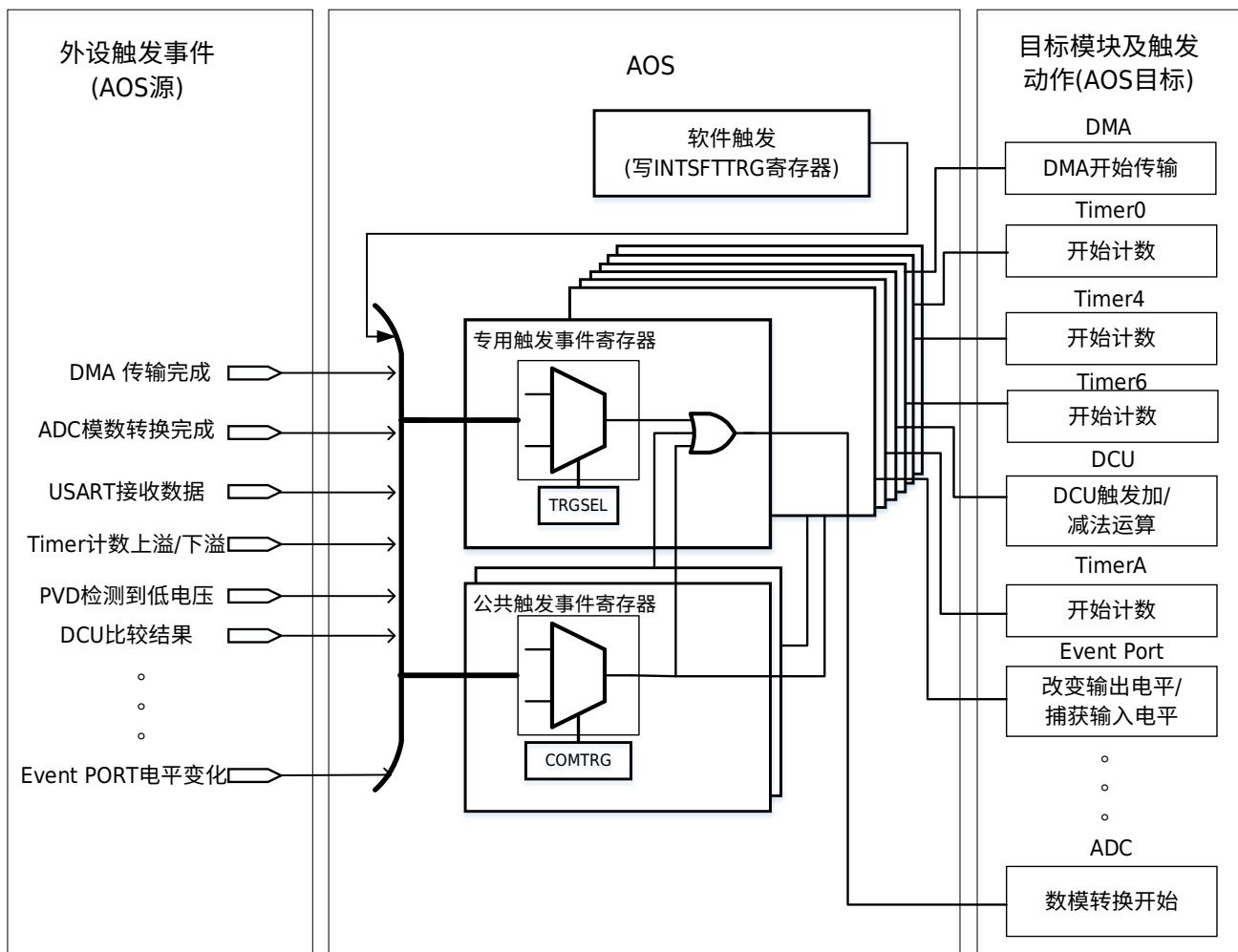


图 11-1 AOS 系统框图

## 11.2 功能描述

### 11.2.1 AOS 源事件列表

AOS 源事件编号见【中断控制器（INTC）】章的中断事件表。表中“触发源”列中打“√”的事件可以用作 AOS 源。

### 11.2.2 AOS 目标列表

表 11-1 AOS 目标列表

模块	序号	动作
DCU_1	0	触发加/减法运算，三角波/锯齿波模式下触发数据递增或递减
DCU_2	1	触发加/减法运算，三角波/锯齿波模式下触发数据递增或递减
DCU_3	2	触发加/减法运算，三角波/锯齿波模式下触发数据递增或递减
DMA_1	3	触发通道 0 开始传输
	4	触发通道 1 开始传输
	5	触发通道 2 开始传输
	6	触发通道 3 开始传输
	7	触发通道 4 开始传输
	8	触发通道 5 开始传输
	9	触发通道 6 开始传输
	10	触发通道 7 开始传输
DMA_2	11	触发通道 0 开始传输
	12	触发通道 1 开始传输
	13	触发通道 2 开始传输
	14	触发通道 3 开始传输
	15	触发通道 4 开始传输
	16	触发通道 5 开始传输
	17	触发通道 6 开始传输
	18	触发通道 7 开始传输
DMA_1&2	19	触发事件触发通道重置
Timer6	20	触发计数器启动/停止/清零/刷新/捕获/递增/递减
	21	触发计数器启动/停止/清零/刷新/捕获/递增/递减
	22	触发计数器启动/停止/清零/刷新/捕获/递增/递减
	23	触发计数器启动/停止/清零/刷新/捕获/递增/递减
Timer4	24	触发计数器启动/停止/清零/捕获
Event Port	25	触发 Event Port1&2 动作
	26	触发 Event Port3&4 动作
Timer0	27	触发计数器启动/停止/清零/捕获
Timer2	28	触发计数器启动/停止/清零/捕获

模块	序号	动作
HASH	29	DMA 完成一个数据块传输触发 HASH 运算
TimerA	30	DMA 完成所有数据块传输触发 HASH 运算
	31	触发计数器单元 1/5 启动/停止/清零/递增/递减
	32	触发计数器单元 2/6/输入捕获
	33	触发计数器单元 2/6/启动/停止/清零/递增/递减
	34	触发计数器单元 1/5/输入捕获
	35	触发计数器单元 3 启动/停止/清零/递增/递减
OTS	36	触发计数器单元 4 输入捕获
ADC_1	37	触发计数器单元 4 启动/停止/清零/递增/递减
ADC_2	38	触发序列 A 开始模数转换
ADC_2	39	触发序列 B 开始模数转换
ADC_3	40	触发序列 A 开始模数转换
ADC_3	41	触发序列 B 开始模数转换

### 11.2.3 专用触发源

具有 AOS 目标的外设电路模块为每个 AOS 目标配有一个专用的外设触发源选择寄存器，当这个寄存器写入 AOS 源对应的事件编号时，该 AOS 目标即选择这个 AOS 源作为触发源。当 AOS 源的事件发生时，这个事件将通过 AOS 传递给 AOS 目标，作为 AOS 目标的外设电路根据自身设定开始动作。

### 11.2.4 公共触发源

除了各个 AOS 目标的专用外设触发源选择寄存器外，AOS 还配置了两个公共触发源选择寄存器 (AOS\_COMTRGSEL1,AOS\_COMTRGSEL2)。用于实现多个 AOS 源触发同一 AOS 目标的功能。使用时首先在 AOS 目标专用外设触发源选择寄存器中将公共触发源使能位置为有效，再在公共触发源选择寄存器中写入 AOS 源对应的事件编号。当 AOS 源的事件发生时，这个事件将通过 AOS 的公共触发源传递给 AOS 目标，作为 AOS 目标的外设电路根据自身设定开始动作。当同时设置专用触发源和公共触发源后，可以最多同时有 3 个 AOS 源触发同一 AOS 目标，3 个 AOS 源中的任意一个发生触发事件时，都将触发该 AOS 目标。

所有 AOS 目标共享这两个公共触发源。因此使用时当其它 AOS 目标不使用公共触发源选择寄存器选定的事件时，需要在它的专用外设触发源选择寄存器中将公共触发源使能位置为无效，以防止错误的触发动作。

## 11.3 寄存器说明

### 11.3.1 寄存器一览

表 11-2 寄存器列表

寄存器地址：0x40010800

寄存器名	符号	偏移地址	位宽	初始值
外设触发事件寄存器	INTSFTTRG	0x00	32	0x00000000
DCU 触发源选择寄存器	DCU_TRGSELx (x=1~3)	0x04,0x08,0x0C	32	0x000001FF
DMA1 传输启动触发源选择寄存器	DMA1_TRGSELx (x=0~7)	0x10,0x14,0x18,0x1C, 0x20,0x24,0x28,0x2C	32	0x000001FF
DMA2 传输启动触发源选择寄存器	DMA2_TRGSELx (x=0~7)	0x30,0x34,0x38,0x3C, 0x40,0x44,0x48,0x4C	32	0x000001FF
DMA 通道重置触发源选择寄存器	DMA_RC_TRGSEL	0x50	32	0x000001FF
Timer6 硬件触发事件选择寄存器	TMR6_TRGSELx (x=0~3)	0x54,0x58,0x5C,0x60	32	0x000001FF
Time4 硬件触发事件选择寄存器	TMR4_TRGSEL	0x64	32	0x000001FF
Event Port1,2 触发源选择寄存器	PEVNT_TRGSEL12	0x68	32	0x000001FF
Event Port3,4 触发源选择寄存器	PEVNT_TRGSEL34	0x6C	32	0x000001FF
Timer0 硬件触发事件选择寄存器	TMR0_TRGSEL	0x70	32	0x000001FF
Timer2 硬件触发事件选择寄存器	TMR2_TRGSEL	0x74	32	0x000001FF
HASH 硬件触发事件选择寄存器 A	HASH_TRGSEL0	0x7C	32	0x000001FF
HASH 硬件触发事件选择寄存器 B	HASH_TRGSEL1	0x78	32	0x000001FF
TimerA 硬件触发事件选择寄存器	TMRA_TRGSELx (x=0~3)	0x80,0x84,0x88,0x8C	32	0x000001FF
OTS 触发源选择寄存器	OTS_TRGSEL	0x90	32	0x000001FF
ADC1 片内触发源选择寄存器	ADC1_TRGSELx (x=0,1)	0x94,0x98	32	0x000001FF
ADC2 片内触发源选择寄存器	ADC2_TRGSELx (x=0,1)	0x9C,0xA0	32	0x000001FF
ADC3 片内触发源选择寄存器	ADC3_TRGSELx (x=0,1)	0xA4,0xA8	32	0x000001FF
公共触发源选择寄存器 1	AOS_COMTRGSEL1	0xAC	32	0x000001FF
公共触发源选择寄存器 2	AOS_COMTRGSEL2	0xB0	32	0x000001FF

### 11.3.2 外设触发事件寄存器 (INTSFTTRG)

该寄存器每次写“1”将产生一次触发事件。

偏移地址：0x00

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
SFTG															

位	标记	位名	功能	读写
b31~b1	Reserved	-	读出时为“0”，写入时写“0”	R/W
b0	SFTG	软件触发	0：不产生软件触发事件 1：产生一次软件触发事件 将该位置 1 产生一次外设触发事件，软件写 0 无效	W

### 11.3.3 DCU 触发源选择寄存器 (DCU\_TRGSELx, x=1~3)

在 DCU 选择硬件触发启动模式后，请将触发事件的编号写入该寄存器中。当编号对应的外设事件发生后，DCU 将由该事件触发启动并进行运算。

每个 DCU 单元对应一个寄存器。

偏移地址：0x04, 0x08, 0x0C

复位值：0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
TRGSEL[8:0]															

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0：禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 1：允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	R/W
b30	COMEN[0]	公共触发使能	0：禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 1：允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

### 11.3.4 DMA1 传输启动触发源选择寄存器 (DMA1\_TRGSELx, x=0~7)

在 DMA1 使能后，将触发事件的编号写入该寄存器中，当编号对应的外设事件发生时，DMA1 将由该事件触发启动并进行传输。

每个数据传输通道对应一个寄存器。

偏移地址：0x10, 0x14, 0x18, 0x1C, x20, 0x24, 0x28, 0x2C

复位值: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TRGSEL[8:0]							
<hr/>															
位	标记	位名	功能										读写		
b31	COMEN[1]	公共触发使能	0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标										R/W		
b30	COMEN[0]	公共触发使能	0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标										R/W		
b29~b9	Reserved	-	读出时为“0”，写入时写“0”										R/W		
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号										R/W		

### 11.3.5 DMA2 传输启动触发源选择寄存器 (DMA2\_TRGSELx, x=0~7)

在 DMA2 使能后，将触发事件的编号写入该寄存器中，当编号对应的外设事件发生时，DMA2 将由该事件触发启动并进行传输。

每个数据传输通道对应一个寄存器。

偏移地址：0x30, 0x34, 0x38, 0x3C, x40, 0x44, 0x48, 0x4C

复位值: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TRGSEL[8:0]							

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	R/W
b30	COMEN[0]	公共触发使能	0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

### 11.3.6 DMA 通道重置触发源选择寄存器 (DMA\_RC\_TRGSEL)

在 DMA 使能后，将触发事件的编号写入该寄存器中，当编号对应的外设事件发生时，DMA 将由该事件触发进行通道重置。

DMA1 和 DMA2 共用这个寄存器。

偏移地址：0x50

复位值: 0x0000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TRGSEL[8:0]							

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	R/W
b30	COMEN[0]	公共触发使能	0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

### 11.3.7 Timer6 硬件触发事件选择寄存器 (TMR6\_TRGSELx, x=0~3)

在 Timer6 选择内部触发硬件同步后，将触发事件的编号写入该寄存器中，当编号对应的外设电路事件发生时，Timer6 将由该事件触发硬件同步（包括启动/停止/清零/刷新/递增/递减/输入捕获）。

所有 Timer6 单元共用这 4 个寄存器。

偏移地址：0x54, 0x58, 0x5C, 0x60

复位值: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TRGSEL[8:0]							

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	R/W
b30	COMEN[0]	公共触发使能	0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

### 11.3.8 Timer4 硬件触发事件选择寄存器 (TMR4\_TRGSEL)

在 Timer4 选择内部触发硬件同步后，将触发事件的编号写入该寄存器中，当编号对应的外设电路事件发生时，Timer4 将由该事件触发硬件同步启动。

偏移地址：0x64

复位值: 0x0000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TRGSEL[8:0]							

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	R/W
b30	COMEN[0]	公共触发使能	0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

### 11.3.9 Event Port 触发源选择寄存器 (PEVNT\_TRGSEL12, PEVNT\_TRGSEL34)

将触发事件的编号写入该寄存器中，当编号对应的外设事件发生时，Event Port 被触发输出指定电平或者锁存端口状态。Event Port1,2 共用 PEVNT\_TRGSEL12 寄存器，Event Port3,4 共用 PEVNT\_TRGSEL34 寄存器。

偏移地址：0x68, 0x6C

复位值: 0x0000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TRGSEL[8:0]							

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	R/W
b30	COMEN[0]	公共触发使能	0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

### 11.3.10 Timer0 硬件触发事件选择寄存器 (TMR0\_TRGSEL)

在 Timer0 选择硬件触发后, 将触发事件的编号写入该寄存器中, 当编号对应的外设事件发生时, Timer0 将由该事件触发动作 (包括启动/停止/清零/输入捕获)。

所有 Timer0 单元及其通道共用这个寄存器。

偏移地址: 0x70

复位值: 0x0000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TRGSEL[8:0]							

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	R/W
b30	COMEN[0]	公共触发使能	0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	R/W
b29~b9	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

### 11.3.11 Timer2 硬件触发事件选择寄存器 (TMR2\_TRGSEL)

在 Timer2 选择内部硬件触发后，将触发事件的编号写入该寄存器中，当编号对应的外设事件发生后，Timer2 将由该事件触发动作（包括启动/停止/清零/输入捕获）。

所有 Timer2 单元及其通道共用这个寄存器。

偏移地址：0x74

复位值: 0x0000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TRGSEL[8:0]							

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	R/W
b30	COMEN[0]	公共触发使能	0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

### 11.3.12 HASH 硬件触发事件选择寄存器 A (HASH\_TRGSEL0)

当数据写入 HASH\_DR 后，通过该寄存器选择硬件事件触发源，以启动 HASH 运算。注意，请勿选择 DMA\_BTCx(x=0~7)以外的其他触发信号。

偏移地址：0x7C

复位值: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TRGSEL[8:0]							

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	R/W
b30	COMEN[0]	公共触发使能	0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

### 11.3.13 HASH 硬件触发事件选择寄存器 B (HASH\_TRGSEL1)

当最后一组数据写入 HASH\_DR 后，通过该寄存器选择硬件事件触发源，以启动 HASH 进行最后一次运算。注意，请勿选择 DMA\_TCx(x=0~7)以外的其他触发信号。

偏移地址：0x78

复位值: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TRGSEL[8:0]							

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	R/W
b30	COMEN[0]	公共触发使能	0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

### 11.3.14 TimerA 硬件触发事件选择寄存器 (TMRA\_TRGSELx, x=0~3)

在 TimerA 选择内部硬件触发后，将触发事件的编号写入该寄存器中，当编号对应的外设事件发生时，TimerA 将由该事件触发动作（包括启动/停止/清零/输入捕获）。

所有 TimerA 单元共用这 4 个寄存器。

偏移地址：0x80, 0x84, 0x88, 0x8C

复位值: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TRGSEL[8:0]							

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	R/W
b30	COMEN[0]	公共触发使能	0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

### 11.3.15 OTS 触发源选择寄存器 (OTS\_TRGSEL)

在 OTS 选择硬件触发后，将触发事件的编号写入该寄存器中，当编号对应的外设事件发生时，OTS 将由该事件触发开始测温。

偏移地址：0x90

复位值: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TRGSEL[8:0]							

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	R/W
b30	COMEN[0]	公共触发使能	0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

### 11.3.16 ADC1 片内触发源选择寄存器 (ADC1\_TRGSELx, x=0,1)

在 ADC1 选择片内触发模式后，将触发事件的编号写入该寄存器中，当编号对应的外设事件发生时，ADC1 将由该事件触发模数转换。

ADC1\_TRGSEL0, 1 寄存器分别对应序列 A, B。

偏移地址：0x94, 0x98

复位值: 0x0000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TRGSEL[8:0]							

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	R/W
b30	COMEN[0]	公共触发使能	0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

### 11.3.17 ADC2 片内触发源选择寄存器 (ADC2\_TRGSELx, x=0,1)

在 ADC2 选择片内触发模式后，将触发事件的编号写入该寄存器中，当编号对应的外设事件发生时，ADC2 将由该事件触发模数转换。

ADC2\_TRGSEL0, 1 寄存器分别对应序列 A, B。

偏移地址：0x9C, 0xA0

复位值: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TRGSEL[8:0]							

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	R/W
b30	COMEN[0]	公共触发使能	0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

### 11.3.18 ADC3 片内触发源选择寄存器 (ADC3\_TRGSELx, x=0,1)

在 ADC3 选择片内触发模式后，将触发事件的编号写入该寄存器中，当编号对应的外设事件发生时，ADC3 将由该事件触发模数转换。

ADC3\_TRGSEL0, 1 寄存器分别对应序列 A, B。

偏移地址：0xA4, 0xA8

复位值: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TRGSEL[8:0]							

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	R/W
b30	COMEN[0]	公共触发使能	0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

### 11.3.19 公共触发源选择寄存器 (AOS\_COMTRGSEL1, 2)

在 AOS\_COMTRGSEL1,2 写入触发事件的编号，当编号对应的外设电路事件发生时，如果一个或多个 AOS 目标的专用触发源选择寄存器的 COMEN 位值为“1”，则编号对应的外设事件将触发这一个或多个 AOS 目标启动。

偏移地址：0xAC, 0xB0

复位值: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								COMTRG[8:0]							

位	标记	位名	功能	读写
b31~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	COMTRG[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

## 12 存储保护单元 (MPU)

### 12.1 简介

MPU 可以提供对存储器的保护，通过阻止非授权的访问，可以提高系统的安全性。

本芯片内置了 1 个针对 CPU 的 MPU 单元，1 个针对 CPU 主栈指针的 MPU 单元，1 个针对 CPU 线程栈指针的 MPU 单元，3 个针对 DMA 的 MPU 单元和 1 个针对 IP 的 MPU 单元。

模块	内容
ARM MPU	CPU 的存储保护单元 8 区域，详见 ARM MPU 说明
MSPMPU	CPU 主栈指针存储保护单元
PSPMPU	CPU 线程栈指针存储保护单元
系统 DMA_1 MPU: SMPU1	系统 DMA_1 的存储保护单元 16 区域，8 区域系统 DMA 专用，8 区域所有 DMA 共用
系统 DMA_2 MPU: SMPU2	系统 DMA_2 的存储保护单元 16 区域，8 区域系统 DMA 专用，8 区域所有 DMA 共用
USBFS-DMA MPU: FMPU	USBFS-DMA 的存储保护单元 8 区域，所有 DMA 共用
IPMPU	针对系统 IP 和安全相关 IP 的访问保护单元

其中 ARM MPU 提供 CPU 对全部 4G 地址空间的访问权限控制，介绍略。

MSPMPU/ PSPMPU 分别提供对 CPU 的主栈指针/线程栈指针的保护，指针超出设定范围时，可以设置 MPU 动作为不可屏蔽中断/复位。

SMPU1/ SMPU2/ FMPU 分别提供系统 DMA\_1/系统 DMA\_2/USBFS-DMA 对全部 4G 地址空间的读写访问权限控制。对禁止空间发生访问时，可以设置 MPU 动作为无视/总线错误/不可屏蔽中断/复位。

IPMPU 提供非特权模式时对系统 IP 和安全相关 IP 的访问权限控制。

## 12.2 功能描述

### 12.2.1 区域范围设置

MPU 以区域为单位对存储空间进行权限管理。每个区域可以独立设定基址地址和区域大小，可设的范围为 32Byte~4GByte，大小必须为  $2^{n}\text{Byte}$  ( $n=5\sim32$ )，对应的基址低 n 位为 0。

没有被任何区域覆盖的地址空间称为背景区域。

### 12.2.2 权限设置

每个区域包括背景区域可以针对各个 DMA 独立设置允许读/禁止读和允许写/禁止写。如果不同区域之间发生地址重叠，则设定的禁止优先。

### 12.2.3 MPU 动作选择

发生被禁止的访问时，该访问被无视（读访问读到 0，写访问忽略）的同时，可以设置对应的动作，可以设置为：

- 无视
- 总线错误
- 不可屏蔽中断
- 复位

### 12.2.4 启动 MPU

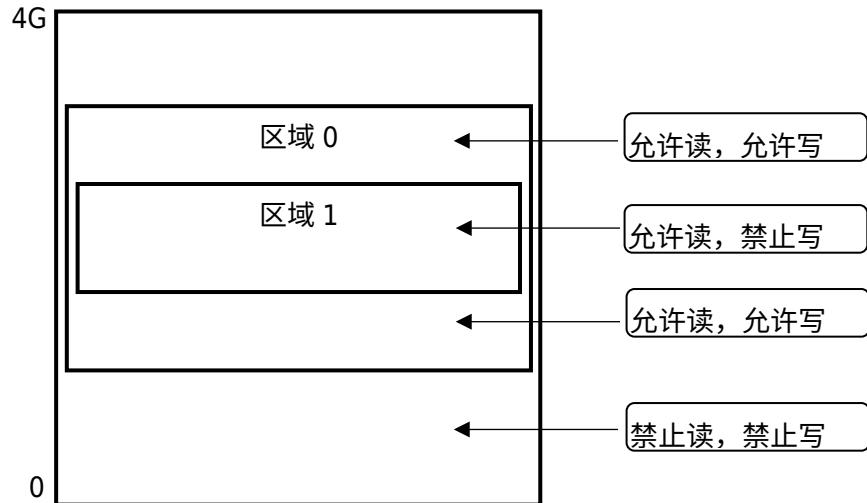
MSPMPU/ PSPMPU/ SMPU1/ SMPU2/ FMPU 可以独立使能。

建议在设置好区域范围/权限设置/动作选择后再使能 MPU。

## 12.3 应用举例

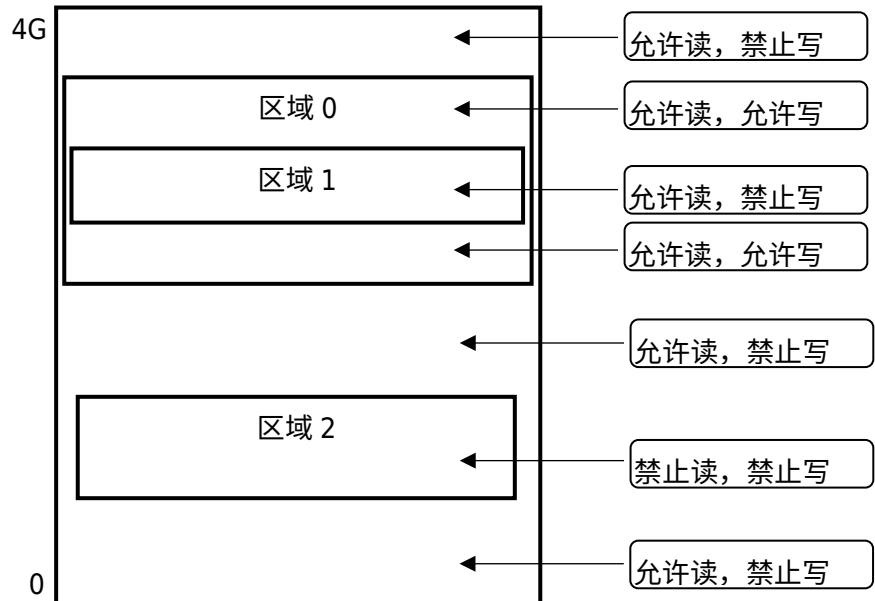
### 12.3.1 只允许部分空间访问

例：将背景区域权限设置为禁止读/禁止写，区域 0 设置为允许读/允许写，区域 1 设置为允许读禁止写，区域 0 范围覆盖区域 1。



### 12.3.2 只禁止部分空间访问

例：将背景区域权限设置为允许读/禁止写，区域 0 设置为允许读/允许写，区域 1 设置为允许读/禁止写，区域 0 覆盖区域 1，区域 2 设置为禁止读/禁止写。



## 12.4 寄存器说明

本模块的寄存器只能通过 CPU 来设置。

MPU 基准地址: 0x40050000

表 12-1 寄存器列表

偏移地址	寄存器名	初始值	名称	写保护
0x00~0x3C	MPU_RGD0~15	0x00000000	区域 0~15 范围描述寄存器	MPUWE
0x40	MPU_SR	0x00000000	状态标志寄存器	无
0x44	MPU_ECLR	0x00000000	标志清除寄存器	无
0x48	MPU_WP	0x00000000	写保护寄存器	WKEY
0x4C	MPU_IPPR	0x00000000	IP 访问保护寄存器	MPUWE
0x50	MPU_MSPPBA	0x00000000	MSP 保护开始地址寄存器	MPUWE
0x54	MPU_MSPPCTL	0x00000003	MSP 保护控制寄存器	MPUWE
0x58	MPU_PSPPBA	0x00000000	PSP 保护开始地址寄存器	MPUWE
0x5C	MPU_PSPPCTL	0x00000003	PSP 保护控制寄存器	MPUWE
0x60	MPU_S1RGE	0x00000000	SMPU1 区域使能寄存器	MPUWE
0x64	MPU_S1RGWP	0x00000000	SMPU1 区域写权限寄存器	MPUWE
0x68	MPU_S1RGRP	0x00000000	SMPU1 区域读权限寄存器	MPUWE
0x6C	MPU_S1CR	0x00000000	SMPU1 控制寄存器	MPUWE
0x70	MPU_S2RGE	0x00000000	SMPU2 区域使能寄存器	MPUWE
0x74	MPU_S2RGWP	0x00000000	SMPU2 区域写权限寄存器	MPUWE
0x78	MPU_S2RGRP	0x00000000	SMPU2 区域读权限寄存器	MPUWE
0x7C	MPU_S2CR	0x00000000	SMPU2 控制寄存器	MPUWE
0x80	MPU_FRGE	0x00000000	FMPU 区域使能寄存器	MPUWE
0x84	MPU_FRGWP	0x00000000	FMPU 区域写权限寄存器	MPUWE
0x88	MPU_FRGRP	0x00000000	FMPU 区域读权限寄存器	MPUWE
0x8C	MPU_FCR	0x00000000	FMPU 控制寄存器	MPUWE

### 12.4.1 区域范围描述寄存器 MPU\_RGDr (n=0~15)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
MPURGnADDR [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MPURGnADDR[15:5]										MPURGnSIZE[4:0]					
<hr/>															
位	标记	位名	功能	读写											
b31~b5	MPURGnADDR[31:5]	区域基址	设定区域 n 的基地址, 有效位数与区域大小有关, 低 (MPURGnSIZE+1)位固定为 0	R/W											
<hr/>															
b4~b0	MPURGnSIZE[4:0]	区域大小	设定区域 n 的大小 00000~00011: 保留, 设定禁止 00100: 32Byte 00101: 64Byte ... 11110: 2GByte 11111: 4GByte	R/W											
<hr/>															

### 12.4.2 状态标志寄存器 MPU\_SR

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved										MSPEF	PSPEF	FMPUEAF	SMPU2EAF	SMPU1EAF	
<hr/>															
位	标记	位名	功能	读写											
b31~b5	Reserved	-	保留位, 读出为 0, 写入时写 0	R/W											
b4	MSPEF	MSP 错误标志	0: MSP 指针未超出设定范围 1: MSP 指针超出了设定范围	R											
b3	PSPEF	PSP 错误标志	0: PSP 指针未超出设定范围 1: PSP 指针超出了设定范围	R											
b2	FMPUEAF	FMPU 错误标志	0: USBFS-DMA 未发生错误访问 1: USBFS-DMA 发生了错误访问	R											
b1	SMPU2EAF	SMPU2 错误标志	0: 系统 DMA_2 未发生错误访问 1: 系统 DMA_2 发生了错误访问	R											
b0	SMPU1EAF	SMPU1 错误标志	0: 系统 DMA_1 未发生错误访问 1: 系统 DMA_1 发生了错误访问	R											
<hr/>															

对本寄存器的写操作会被忽略, 清除错误标志请使用 MPUECLR。

### 12.4.3 标志清除寄存器 MPU\_ECLR

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Reserved																
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Reserved												MSP ECLR	PSP ECLR	FMPU ECLR	SMPU2 ECLR	SMPU1E CLR
位	标记	位名	功能	读写												
b31~b5	Reserved	-	保留位, 读出为 0, 写入时写 0	R/W												
b4	MSPECLR	MSP 错误标志清除	写入 1 可以将 MSPEF 清 0	R/W												
b3	PSPECLR	PSP 错误标志清除	写入 1 可以将 PSPEF 清 0	R/W												
b2	FMPUECLR	FMPU 错误标志清除	写入 1 可以将 FMPUEAF 清 0	R/W												
b1	SMPU2ECLR	SMPU2 错误标志清除	写入 1 可以将 SMPU2EAF 清 0	R/W												
b0	SMPU1ECLR	SMPU1 错误标志清除	写入 1 可以将 SMPU1EAF 清 0	R/W												

本寄存器读出值固定为 0x00000000。

### 12.4.4 写保护寄存器 MPU\_WP

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
WKEY[15:1]												MPUWE			
位	标记	位名	功能	读写											
b31~b16	Reserved	-	保留位, 读出为 0, 写入时写 0	R/W											
b15~b1	WKEY[15:1]	写入码	在写入 MPUWE 时, 必须同时向 WKEY 写入 15'b1001_0110_1010_010, 读出为 0	R/W											
b0	MPUWE	MPU 寄存器写允许	0: MPU 地址寄存器/控制寄存器不允许写入 1: MPU 地址寄存器/控制寄存器允许写入	RW											

向本寄存器写入 0x96A5 可以将 MPUWE 置 1, 写入 0x96A4 可以将 MPUWE 清 0, 写入其他值不能改变 MPUWE。

## 12.4.5 IP 访问保护寄存器 MPU\_IPPR

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
BUSERRE E	-	MSTPW RP	MSTPRD P	SYSCW RP	SYSCRD P	INTCWR P	INTCRD P	SRAMC WRP	SRAMC RDP	DMPUW RP	DMPUR DP	RTCWR P	RTCRDP	BKSRAM WRP	BKSRA MRDP
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SWDTW RP	SWDTR DP	WDTWR P	WDTRD P	-	-	EFMWR P	EFMRD P	CRCWR P	CRCRD P	TRNGW RP	TRNGR DP	HASHW RP	HASHR DP	AESWRP	AESRD P

位	标记	位名	功能	读写
b31	BUSERRE	总线错误允许	0: 忽略对保护 IP 的访问 1: 发生对保护 IP 的访问时, 返回总线错误	R/W
b30	Reserved	-	保留位, 读出为 0, 写入时写 0	R/W
b29	MSTPW <sub>RP</sub>	MSTP 写保护	0: 允许对 PWC_FCG0/1/2/3、PWC_FCG0PC 的写操作 1: 禁止对 PWC_FCG0/1/2/3、PWC_FCG0PC 的写操作	R/W
b28	MSTPRD <sub>P</sub>	MSTP 读保护	0: 允许对 PWC_FCG0/1/2/3、PWC_FCG0PC 的读操作 1: 禁止对 PWC_FCG0/1/2/3、PWC_FCG0PC 的读操作	R/W
b27	SYSCW <sub>RP</sub>	SYSC 写保护	0: 允许对 RMU/CMU/PWC 的写操作 1: 禁止对 RMU/CMU/PWC 的写操作 注意: BUSERRE 设置为 1 时, 对 RMU/PWC 的写操作不会发生总线错误	R/W
b26	SYSCRD <sub>P</sub>	SYSC 读保护	0: 允许对 RMU/CMU/PWC 的读操作 1: 禁止对 RMU/CMU/PWC 的读操作 注意: BUSERRE 设置为 1 时, 对 RMU/PWC 的读操作不会发生总线错误	R/W
b25	INTCWR <sub>P</sub>	INTC 写保护	0: 允许对 INTC 的写操作 1: 禁止对 INTC 的写操作	R/W
b24	INTCRD <sub>P</sub>	INTC 读保护	0: 允许对 INTC 的读操作 1: 禁止对 INTC 的读操作	R/W
b23	SRAMCWR <sub>P</sub>	SRAMC 写保护	0: 允许对 SRAMC 的写操作 1: 禁止对 SRAMC 的写操作	R/W
b22	SRAMCRD <sub>P</sub>	SRAMC 读保护	0: 允许对 SRAMC 的读操作 1: 禁止对 SRAMC 的读操作	R/W
b21	DMPUW <sub>RP</sub>	DMPU 写保护	0: 允许对 DMPU 的写操作 1: 禁止对 DMPU 的写操作	R/W
b20	DMPURD <sub>P</sub>	DMPU 读保护	0: 允许对 DMPU 的读操作 1: 禁止对 DMPU 的读操作	R/W
b19	RTCWR <sub>P</sub>	RTC 写保护	0: 允许对 RTC 的写操作 1: 禁止对 RTC 的写操作	R/W
b18	RTCRDP	RTC 读保护	0: 允许对 RTC 的读操作 1: 禁止对 RTC 的读操作	R/W
b17	BKSRAMWR <sub>P</sub>	BKSRAM 写保护	0: 允许对 Ret SRAM 的写操作 1: 禁止对 Ret SRAM 的写操作	R/W
b16	BKSRAMRD <sub>P</sub>	BKSRAM 读保护	0: 允许对 Ret SRAM 的读操作 1: 禁止对 Ret SRAM 的读操作	R/W
b15	SWDTWR <sub>P</sub>	SWDT 写保护	0: 允许对 SWDT 的写操作 1: 禁止对 SWDT 的写操作	R/W

b14	SWDTRDP	SWDT 读保护	0: 允许对 SWDT 的读操作 1: 禁止对 SWDT 的读操作	R/W
b13	WDTWRP	WDT 写保护	0: 允许对 WDT 的写操作 1: 禁止对 WDT 的写操作	R/W
b12	WDTRDP	WDT 读保护	0: 允许对 WDT 的读操作 1: 禁止对 WDT 的读操作	R/W
b11-b10	Reserved	-	保留位, 读出为 0, 写入时写 0	R/W
b9	EFMWRP	EFM 写保护	0: 允许对 EFM 的写操作 1: 禁止对 EFM 的写操作	R/W
b8	EFMRDP	EFM 读保护	0: 允许对 EFM 的读操作 1: 禁止对 EFM 的读操作	R/W
b7	CRCWRP	CRC 写保护	0: 允许对 CRC 的写操作 1: 禁止对 CRC 的写操作	R/W
b6	CRCRDP	CRC 读保护	0: 允许对 CRC 的读操作 1: 禁止对 CRC 的读操作	R/W
b5	TRNGWRP	TRNG 写保护	0: 允许对 TRNG 的写操作 1: 禁止对 TRNG 的写操作	R/W
b4	TRNGRDP	TRNG 读保护	0: 允许对 TRNG 的读操作 1: 禁止对 TRNG 的读操作	R/W
b3	HASHWRP	HASH 写保护	0: 允许对 HASH 的写操作 1: 禁止对 HASH 的写操作	R/W
b2	HASHRDP	HASH 读保护	0: 允许对 HASH 的读操作 1: 禁止对 HASH 的读操作	R/W
b1	AESWRP	AES 写保护	0: 允许对 AES 的写操作 1: 禁止对 AES 的写操作	R/W
b0	AESRDP	AES 读保护	0: 允许对 AES 的读操作 1: 禁止对 AES 的读操作	R/W

特权模式下不受本寄存器影响可以读写访问对象 IP。

## 12.4.6 MSP 保护开始地址寄存器 MPU\_MSPPBA

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
MSPPBA[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MSPPBA[15:2]															
Reserved															
位	标记	位名	功能	读写											
b31~b2	MSPPBA[31:2]	允许范围开始地址	设定 MSP 指针允许范围开始地址	R/W											
b1~b0	Reserved	-	保留位，读出为 0，写入时写 0	R/W											

## 12.4.7 MSP 保护控制寄存器 MPU\_MSPPCTL

复位值：0x00000003

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
MSPPE	MSPPACT	Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MSPPSIZE[15:2]															
Reserved															
位	标记	位名	功能	读写											
b31	MSPPE	MSP 保护使能	0: MSP 指针保护无效 1: MSP 指针保护有效	R/W											
b30	MSPPACT	MSP 保护动作选择	0: 不可屏蔽中断 1: 复位												
b29~b16	Reserved	-	保留位，读出为 0，写入时写 0	R/W											
设定 MSP 指针允许范围，设定值 4Byte~64Kbyte															
b15~b2	MSPPSIZE[15:2]	允许范围大小	当 MSPPSIZE=0x0 时，MSP 指针允许范围为 4Byte，当 MSPPSIZE=0xFFFF 时，MSP 指针允许范围为 64Kbyte	R/W											
b1~b0	Reserved	-	保留位，读出为 1，写入时写 1	R/W											

### 12.4.8 PSP 保护开始地址寄存器 MPU\_PSPBAA

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PSPBAA[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PSPBAA[15:2]															
Reserved															
位	标记	位名	功能	读写											
b31~b2	PSPBAA[31:2]	允许范围开始地址	设定 PSP 指针允许范围开始地址	R/W											
b1~b0	Reserved	-	保留位，读出为 0，写入时写 0	R/W											

### 12.4.9 PSP 保护控制寄存器 MPU\_PSPCTL

复位值：0x00000003

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PSPPE	PSPPACT	Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PSPPSIZE[15:2]															
Reserved															
位	标记	位名	功能	读写											
b31	PSPPE	PSP 保护使能	0: PSP 指针保护无效 1: PSP 指针保护有效	R/W											
b30	PSPPACT	PSP 保护动作选择	0: 不可屏蔽中断 1: 复位												
b29~b16	Reserved	-	保留位，读出为 0，写入时写 0	R/W											
设定 PSP 指针允许范围，设定值 4Byte~64Kbyte															
b15~b2	PSPPSIZE[15:2]	允许范围大小	当 PSPPSIZE=0x0 时，PSP 指针允许范围为 4Byte，当 PSPPSIZE=0xFFFF 时，PSP 指针允许范围为 64Kbyte	R/W											
b1~b0	Reserved	-	保留位，读出为 1，写入时写 1	R/W											

## 12.4.10 SMPU1 区域使能寄存器 MPU\_S1RGE

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S1RG15E	S1RG14E	S1RG13E	S1RG12E	S1RG11E	S1RG10E	S1RG9E	S1RG8E	S1RG7E	S1RG6E	S1RG5E	S1RG4E	S1RG3E	S1RG2E	S1RG1E	S1RG0E
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	保留位, 读出为 0, 写入时写 0	R/W											
b15	S1RG15E	SMPU1 区域 15 使能	0: SMPU1 的区域 15 无效 1: SMPU1 的区域 15 有效	R/W											
b14	S1RG14E	SMPU1 区域 14 使能	0: SMPU1 的区域 14 无效 1: SMPU1 的区域 14 有效	R/W											
b13	S1RG13E	SMPU1 区域 13 使能	0: SMPU1 的区域 13 无效 1: SMPU1 的区域 13 有效	R/W											
b12	S1RG12E	SMPU1 区域 12 使能	0: SMPU1 的区域 12 无效 1: SMPU1 的区域 12 有效	R/W											
b11	S1RG11E	SMPU1 区域 11 使能	0: SMPU1 的区域 11 无效 1: SMPU1 的区域 11 有效	R/W											
b10	S1RG10E	SMPU1 区域 10 使能	0: SMPU1 的区域 10 无效 1: SMPU1 的区域 10 有效	R/W											
b9	S1RG9E	SMPU1 区域 9 使能	0: SMPU1 的区域 9 无效 1: SMPU1 的区域 9 有效	R/W											
b8	S1RG8E	SMPU1 区域 8 使能	0: SMPU1 的区域 8 无效 1: SMPU1 的区域 8 有效	R/W											
b7	S1RG7E	SMPU1 区域 7 使能	0: SMPU1 的区域 7 无效 1: SMPU1 的区域 7 有效	R/W											
b6	S1RG6E	SMPU1 区域 6 使能	0: SMPU1 的区域 6 无效 1: SMPU1 的区域 6 有效	R/W											
b5	S1RG5E	SMPU1 区域 5 使能	0: SMPU1 的区域 5 无效 1: SMPU1 的区域 5 有效	R/W											
b4	S1RG4E	SMPU1 区域 4 使能	0: SMPU1 的区域 4 无效 1: SMPU1 的区域 4 有效	R/W											
b3	S1RG3E	SMPU1 区域 3 使能	0: SMPU1 的区域 3 无效 1: SMPU1 的区域 3 有效	R/W											
b2	S1RG2E	SMPU1 区域 2 使能	0: SMPU1 的区域 2 无效 1: SMPU1 的区域 2 有效	R/W											
b1	S1RG1E	SMPU1 区域 1 使能	0: SMPU1 的区域 1 无效 1: SMPU1 的区域 1 有效	R/W											
b0	S1RG0E	SMPU1 区域 0 使能	0: SMPU1 的区域 0 无效 1: SMPU1 的区域 0 有效	R/W											

## 12.4.11 SMPU1 区域写权限寄存器 MPU\_S1RGWP

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S1RG15 WP	S1RG14 WP	S1RG13 WP	S1RG12 WP	S1RG11 WP	S1RG10 WP	S1RG9 WP	S1RG8 WP	S1RG7 WP	S1RG6 WP	S1RG5 WP	S1RG4 WP	S1RG3 WP	S1RG2 WP	S1RG1 WP	S1RG0 WP
位	标记	位名	功能	读写											
b31~b16	Reserved	-	保留位, 读出为 0, 写入时写 0	R/W											
b15	S1RG15WP	SMPU1 区域 15 写权限	0: 区域 15 允许系统 DMA_1 写 1: 区域 15 禁止系统 DMA_1 写	R/W											
b14	S1RG14WP	SMPU1 区域 14 写权限	0: 区域 14 允许系统 DMA_1 写 1: 区域 14 禁止系统 DMA_1 写	R/W											
b13	S1RG13WP	SMPU1 区域 13 写权限	0: 区域 13 允许系统 DMA_1 写 1: 区域 13 禁止系统 DMA_1 写	R/W											
b12	S1RG12WP	SMPU1 区域 12 写权限	0: 区域 12 允许系统 DMA_1 写 1: 区域 12 禁止系统 DMA_1 写	R/W											
b11	S1RG11WP	SMPU1 区域 11 写权限	0: 区域 11 允许系统 DMA_1 写 1: 区域 11 禁止系统 DMA_1 写	R/W											
b10	S1RG10WP	SMPU1 区域 10 写权限	0: 区域 10 允许系统 DMA_1 写 1: 区域 10 禁止系统 DMA_1 写	R/W											
b9	S1RG9WP	SMPU1 区域 9 写权限	0: 区域 9 允许系统 DMA_1 写 1: 区域 9 禁止系统 DMA_1 写	R/W											
b8	S1RG8WP	SMPU1 区域 8 写权限	0: 区域 8 允许系统 DMA_1 写 1: 区域 8 禁止系统 DMA_1 写	R/W											
b7	S1RG7WP	SMPU1 区域 7 写权限	0: 区域 7 允许系统 DMA_1 写 1: 区域 7 禁止系统 DMA_1 写	R/W											
b6	S1RG6WP	SMPU1 区域 6 写权限	0: 区域 6 允许系统 DMA_1 写 1: 区域 6 禁止系统 DMA_1 写	R/W											
b5	S1RG5WP	SMPU1 区域 5 写权限	0: 区域 5 允许系统 DMA_1 写 1: 区域 5 禁止系统 DMA_1 写	R/W											
b4	S1RG4WP	SMPU1 区域 4 写权限	0: 区域 4 允许系统 DMA_1 写 1: 区域 4 禁止系统 DMA_1 写	R/W											
b3	S1RG3WP	SMPU1 区域 3 写权限	0: 区域 3 允许系统 DMA_1 写 1: 区域 3 禁止系统 DMA_1 写	R/W											
b2	S1RG2WP	SMPU1 区域 2 写权限	0: 区域 2 允许系统 DMA_1 写 1: 区域 2 禁止系统 DMA_1 写	R/W											
b1	S1RG1WP	SMPU1 区域 1 写权限	0: 区域 1 允许系统 DMA_1 写 1: 区域 1 禁止系统 DMA_1 写	R/W											
b0	S1RG0WP	SMPU1 区域 0 写权限	0: 区域 0 允许系统 DMA_1 写 1: 区域 0 禁止系统 DMA_1 写	R/W											

## 12.4.12 SMPU1 区域读权限寄存器 MPU\_S1RGRP

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S1RG15RP	S1RG14RP	S1RG13RP	S1RG12RP	S1RG11RP	S1RG10RP	S1RG9RP	S1RG8RP	S1RG7RP	S1RG6RP	S1RG5RP	S1RG4RP	S1RG3RP	S1RG2RP	S1RG1RP	S1RG0RP
位	标记	位名	功能												读写
b31~b16	Reserved	-	保留位, 读出为 0, 写入时写 0												R/W
b15	S1RG15RP	SMPU1 区域 15 读权限	0: 区域 15 允许系统 DMA_1 读 1: 区域 15 禁止系统 DMA_1 读												R/W
b14	S1RG14RP	SMPU1 区域 14 读权限	0: 区域 14 允许系统 DMA_1 读 1: 区域 14 禁止系统 DMA_1 读												R/W
b13	S1RG13RP	SMPU1 区域 13 读权限	0: 区域 13 允许系统 DMA_1 读 1: 区域 13 禁止系统 DMA_1 读												R/W
b12	S1RG12RP	SMPU1 区域 12 读权限	0: 区域 12 允许系统 DMA_1 读 1: 区域 12 禁止系统 DMA_1 读												R/W
b11	S1RG11RP	SMPU1 区域 11 读权限	0: 区域 11 允许系统 DMA_1 读 1: 区域 11 禁止系统 DMA_1 读												R/W
b10	S1RG10RP	SMPU1 区域 10 读权限	0: 区域 10 允许系统 DMA_1 读 1: 区域 10 禁止系统 DMA_1 读												R/W
b9	S1RG9RP	SMPU1 区域 9 读权限	0: 区域 9 允许系统 DMA_1 读 1: 区域 9 禁止系统 DMA_1 读												R/W
b8	S1RG8RP	SMPU1 区域 8 读权限	0: 区域 8 允许系统 DMA_1 读 1: 区域 8 禁止系统 DMA_1 读												R/W
b7	S1RG7RP	SMPU1 区域 7 读权限	0: 区域 7 允许系统 DMA_1 读 1: 区域 7 禁止系统 DMA_1 读												R/W
b6	S1RG6RP	SMPU1 区域 6 读权限	0: 区域 6 允许系统 DMA_1 读 1: 区域 6 禁止系统 DMA_1 读												R/W
b5	S1RG5RP	SMPU1 区域 5 读权限	0: 区域 5 允许系统 DMA_1 读 1: 区域 5 禁止系统 DMA_1 读												R/W
b4	S1RG4RP	SMPU1 区域 4 读权限	0: 区域 4 允许系统 DMA_1 读 1: 区域 4 禁止系统 DMA_1 读												R/W
b3	S1RG3RP	SMPU1 区域 3 读权限	0: 区域 3 允许系统 DMA_1 读 1: 区域 3 禁止系统 DMA_1 读												R/W
b2	S1RG2RP	SMPU1 区域 2 读权限	0: 区域 2 允许系统 DMA_1 读 1: 区域 2 禁止系统 DMA_1 读												R/W
b1	S1RG1RP	SMPU1 区域 1 读权限	0: 区域 1 允许系统 DMA_1 读 1: 区域 1 禁止系统 DMA_1 读												R/W
b0	S1RG0RP	SMPU1 区域 0 读权限	0: 区域 0 允许系统 DMA_1 读 1: 区域 0 禁止系统 DMA_1 读												R/W

**12.4.13 SMPU1 控制寄存器 MPU\_S1CR**

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								SMPU1E	Reserved		SMPU1ACT[1:0]	SMPU1BWP	SMPU1BRP		
<hr/>															
位	标记	位名	功能	读写											
b31~b8	Reserved	-	保留位, 读出为 0, 写入时写 0	R/W											
b7	SMPU1E	SMPU1 使能	0: SMPU1 无效 1: SMPU1 有效	R/W											
b6~b4	Reserved	-	保留位, 读出为 0, 写入时写 0	R/W											
设定系统 DMA_1 发生被禁止的访问时的动作															
b3~b2	SMPU1ACT[1:0]	SMPU1 动作选择	00: 无视 (读访问读到 0, 写访问忽略) 01: 无视+DMA 总线错误 10: 无视+不可屏蔽中断 11: 复位	R/W											
b1	SMPU1BWP	SMPU1 背景写权限设置	0: SMPU1 背景空间允许系统 DMA_1 写 1: SMPU1 背景空间禁止系统 DMA_1 写	R/W											
b0	SMPU1BRP	SMPU1 背景读权限设置	0: SMPU1 背景空间允许系统 DMA_1 读 1: SMPU1 背景空间禁止系统 DMA_1 读	R/W											

多个区域设定重叠时, 优先级为: 设定的禁止&gt;设定的允许。

## 12.4.14 SMPU2 区域使能寄存器 MPU\_S2RGE

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S2RG15 E	S2RG14 E	S2RG13 E	S2RG12 E	S2RG11 E	S2RG10 E	S2RG9E	S2RG8E	S2RG7E	S2RG6E	S2RG5E	S2RG4E	S2RG3E	S2RG2E	S2RG1E	S2RG0E
位	标记	位名	功能	读写											
b31~b16	Reserved	-	保留位, 读出为 0, 写入时写 0	R/W											
b15	S2RG15E	SMPU2 区域 15 使能	0: SMPU2 的区域 15 无效 1: SMPU2 的区域 15 有效	R/W											
b14	S2RG14E	SMPU2 区域 14 使能	0: SMPU2 的区域 14 无效 1: SMPU2 的区域 14 有效	R/W											
b13	S2RG13E	SMPU2 区域 13 使能	0: SMPU2 的区域 13 无效 1: SMPU2 的区域 13 有效	R/W											
b12	S2RG12E	SMPU2 区域 12 使能	0: SMPU2 的区域 12 无效 1: SMPU2 的区域 12 有效	R/W											
b11	S2RG11E	SMPU2 区域 11 使能	0: SMPU2 的区域 11 无效 1: SMPU2 的区域 11 有效	R/W											
b10	S2RG10E	SMPU2 区域 10 使能	0: SMPU2 的区域 10 无效 1: SMPU2 的区域 10 有效	R/W											
b9	S2RG9E	SMPU2 区域 9 使能	0: SMPU2 的区域 9 无效 1: SMPU2 的区域 9 有效	R/W											
b8	S2RG8E	SMPU2 区域 8 使能	0: SMPU2 的区域 8 无效 1: SMPU2 的区域 8 有效	R/W											
b7	S2RG7E	SMPU2 区域 7 使能	0: SMPU2 的区域 7 无效 1: SMPU2 的区域 7 有效	R/W											
b6	S2RG6E	SMPU2 区域 6 使能	0: SMPU2 的区域 6 无效 1: SMPU2 的区域 6 有效	R/W											
b5	S2RG5E	SMPU2 区域 5 使能	0: SMPU2 的区域 5 无效 1: SMPU2 的区域 5 有效	R/W											
b4	S2RG4E	SMPU2 区域 4 使能	0: SMPU2 的区域 4 无效 1: SMPU2 的区域 4 有效	R/W											
b3	S2RG3E	SMPU2 区域 3 使能	0: SMPU2 的区域 3 无效 1: SMPU2 的区域 3 有效	R/W											
b2	S2RG2E	SMPU2 区域 2 使能	0: SMPU2 的区域 2 无效 1: SMPU2 的区域 2 有效	R/W											
b1	S2RG1E	SMPU2 区域 1 使能	0: SMPU2 的区域 1 无效 1: SMPU2 的区域 1 有效	R/W											
b0	S2RG0E	SMPU2 区域 0 使能	0: SMPU2 的区域 0 无效 1: SMPU2 的区域 0 有效	R/W											

## 12.4.15 SMPU2 区域写权限寄存器 MPU\_S2RGWP

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S2RG15 WP	S2RG14 WP	S2RG13 WP	S2RG12 WP	S2RG11 WP	S2RG10 WP	S2RG9 WP	S2RG8 WP	S2RG7 WP	S2RG6 WP	S2RG5 WP	S2RG4 WP	S2RG3 WP	S2RG2 WP	S2RG1 WP	S2RG0 WP
位	标记	位名	功能	读写											
b31~b16	Reserved	-	保留位, 读出为 0, 写入时写 0	R/W											
b15	S2RG15WP	SMPU2 区域 15 写权限	0: 区域 15 允许系统 DMA_2 写 1: 区域 15 禁止系统 DMA_2 写	R/W											
b14	S2RG14WP	SMPU2 区域 14 写权限	0: 区域 14 允许系统 DMA_2 写 1: 区域 14 禁止系统 DMA_2 写	R/W											
b13	S2RG13WP	SMPU2 区域 13 写权限	0: 区域 13 允许系统 DMA_2 写 1: 区域 13 禁止系统 DMA_2 写	R/W											
b12	S2RG12WP	SMPU2 区域 12 写权限	0: 区域 12 允许系统 DMA_2 写 1: 区域 12 禁止系统 DMA_2 写	R/W											
b11	S2RG11WP	SMPU2 区域 11 写权限	0: 区域 11 允许系统 DMA_2 写 1: 区域 11 禁止系统 DMA_2 写	R/W											
b10	S2RG10WP	SMPU2 区域 10 写权限	0: 区域 10 允许系统 DMA_2 写 1: 区域 10 禁止系统 DMA_2 写	R/W											
b9	S2RG9WP	SMPU2 区域 9 写权限	0: 区域 9 允许系统 DMA_2 写 1: 区域 9 禁止系统 DMA_2 写	R/W											
b8	S2RG8WP	SMPU2 区域 8 写权限	0: 区域 8 允许系统 DMA_2 写 1: 区域 8 禁止系统 DMA_2 写	R/W											
b7	S2RG7WP	SMPU2 区域 7 写权限	0: 区域 7 允许系统 DMA_2 写 1: 区域 7 禁止系统 DMA_2 写	R/W											
b6	S2RG6WP	SMPU2 区域 6 写权限	0: 区域 6 允许系统 DMA_2 写 1: 区域 6 禁止系统 DMA_2 写	R/W											
b5	S2RG5WP	SMPU2 区域 5 写权限	0: 区域 5 允许系统 DMA_2 写 1: 区域 5 禁止系统 DMA_2 写	R/W											
b4	S2RG4WP	SMPU2 区域 4 写权限	0: 区域 4 允许系统 DMA_2 写 1: 区域 4 禁止系统 DMA_2 写	R/W											
b3	S2RG3WP	SMPU2 区域 3 写权限	0: 区域 3 允许系统 DMA_2 写 1: 区域 3 禁止系统 DMA_2 写	R/W											
b2	S2RG2WP	SMPU2 区域 2 写权限	0: 区域 2 允许系统 DMA_2 写 1: 区域 2 禁止系统 DMA_2 写	R/W											
b1	S2RG1WP	SMPU2 区域 1 写权限	0: 区域 1 允许系统 DMA_2 写 1: 区域 1 禁止系统 DMA_2 写	R/W											
b0	S2RG0WP	SMPU2 区域 0 写权限	0: 区域 0 允许系统 DMA_2 写 1: 区域 0 禁止系统 DMA_2 写	R/W											

**12.4.16 SMPU2 区域读权限寄存器 MPU\_S2RGRP**

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S2RG15RP	S2RG14RP	S2RG13RP	S2RG12RP	S2RG11RP	S2RG10RP	S2RG9RP	S2RG8RP	S2RG7RP	S2RG6RP	S2RG5RP	S2RG4RP	S2RG3RP	S2RG2RP	S2RG1RP	S2RG0RP
位	标记	位名	功能	读写											
b31~b16	Reserved	-	保留位，读出为 0，写入时写 0	R/W											
b15	S2RG15RP	SMPU2 区域 15 读权限	0: 区域 15 允许系统 DMA_2 读 1: 区域 15 禁止系统 DMA_2 读	R/W											
b14	S2RG14RP	SMPU2 区域 14 读权限	0: 区域 14 允许系统 DMA_2 读 1: 区域 14 禁止系统 DMA_2 读	R/W											
b13	S2RG13RP	SMPU2 区域 13 读权限	0: 区域 13 允许系统 DMA_2 读 1: 区域 13 禁止系统 DMA_2 读	R/W											
b12	S2RG12RP	SMPU2 区域 12 读权限	0: 区域 12 允许系统 DMA_2 读 1: 区域 12 禁止系统 DMA_2 读	R/W											
b11	S2RG11RP	SMPU2 区域 11 读权限	0: 区域 11 允许系统 DMA_2 读 1: 区域 11 禁止系统 DMA_2 读	R/W											
b10	S2RG10RP	SMPU2 区域 10 读权限	0: 区域 10 允许系统 DMA_2 读 1: 区域 10 禁止系统 DMA_2 读	R/W											
b9	S2RG9RP	SMPU2 区域 9 读权限	0: 区域 9 允许系统 DMA_2 读 1: 区域 9 禁止系统 DMA_2 读	R/W											
b8	S2RG8RP	SMPU2 区域 8 读权限	0: 区域 8 允许系统 DMA_2 读 1: 区域 8 禁止系统 DMA_2 读	R/W											
b7	S2RG7RP	SMPU2 区域 7 读权限	0: 区域 7 允许系统 DMA_2 读 1: 区域 7 禁止系统 DMA_2 读	R/W											
b6	S2RG6RP	SMPU2 区域 6 读权限	0: 区域 6 允许系统 DMA_2 读 1: 区域 6 禁止系统 DMA_2 读	R/W											
b5	S2RG5RP	SMPU2 区域 5 读权限	0: 区域 5 允许系统 DMA_2 读 1: 区域 5 禁止系统 DMA_2 读	R/W											
b4	S2RG4RP	SMPU2 区域 4 读权限	0: 区域 4 允许系统 DMA_2 读 1: 区域 4 禁止系统 DMA_2 读	R/W											
b3	S2RG3RP	SMPU2 区域 3 读权限	0: 区域 3 允许系统 DMA_2 读 1: 区域 3 禁止系统 DMA_2 读	R/W											
b2	S2RG2RP	SMPU2 区域 2 读权限	0: 区域 2 允许系统 DMA_2 读 1: 区域 2 禁止系统 DMA_2 读	R/W											
b1	S2RG1RP	SMPU2 区域 1 读权限	0: 区域 1 允许系统 DMA_2 读 1: 区域 1 禁止系统 DMA_2 读	R/W											
b0	S2RG0RP	SMPU2 区域 0 读权限	0: 区域 0 允许系统 DMA_2 读 1: 区域 0 禁止系统 DMA_2 读	R/W											

## 12.4.17 SMPU2 控制寄存器 MPU\_S2CR

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								SMPU2E	Reserved		SMPU2ACT[1:0]	SMPU2BWP	SMPU2BRP		
<hr/>															
位	标记	位名	功能	读写											
b31~b8	Reserved	-	保留位, 读出为 0, 写入时写 0	R/W											
b7	SMPU2E	SMPU2 使能	0: SMPU2 无效 1: SMPU2 有效	R/W											
b6~b4	Reserved	-	保留位, 读出为 0, 写入时写 0	R/W											
设定系统 DMA_2 发生被禁止的访问时的动作															
b3~b2	SMPU2ACT[1:0]	SMPU2 动作选择	00: 无视 (读访问读到 0, 写访问忽略) 01: 无视+DMA 总线错误 10: 无视+不可屏蔽中断 11: 复位	R/W											
b1	SMPU2BWP	SMPU2 背景写权限设置	0: SMPU2 背景空间允许系统 DMA_2 写 1: SMPU2 背景空间禁止系统 DMA_2 写	R/W											
b0	SMPU2BRP	SMPU2 背景读权限设置	0: SMPU2 背景空间允许系统 DMA_2 读 1: SMPU2 背景空间禁止系统 DMA_2 读	R/W											

多个区域设定重叠时, 优先级为: 设定的禁止>设定的允许

## 12.4.18 FMPU 区域使能寄存器 MPU\_FRGE

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16		
Reserved																	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		
Reserved										FRG7E	FRG6E	FRG5E	FRG4E	FRG3E	FRG2E	FRG1E	FRG0E

位	标记	位名	功能	读写
b31~b8	Reserved	-	保留位, 读出为 0, 写入时写 0	R/W
b7	FRG7E	FMPU 区域 7 使能	0: FMPU 的区域 7 无效 1: FMPU 的区域 7 有效	R/W
b6	FRG6E	FMPU 区域 6 使能	0: FMPU 的区域 6 无效 1: FMPU 的区域 6 有效	R/W
b5	FRG5E	FMPU 区域 5 使能	0: FMPU 的区域 5 无效 1: FMPU 的区域 5 有效	R/W
b4	FRG4E	FMPU 区域 4 使能	0: FMPU 的区域 4 无效 1: FMPU 的区域 4 有效	R/W
b3	FRG3E	FMPU 区域 3 使能	0: FMPU 的区域 3 无效 1: FMPU 的区域 3 有效	R/W
b2	FRG2E	FMPU 区域 2 使能	0: FMPU 的区域 2 无效 1: FMPU 的区域 2 有效	R/W
b1	FRG1E	FMPU 区域 1 使能	0: FMPU 的区域 1 无效 1: FMPU 的区域 1 有效	R/W
b0	FRG0E	FMPU 区域 0 使能	0: FMPU 的区域 0 无效 1: FMPU 的区域 0 有效	R/W

## 12.4.19 FMPU 区域写权限寄存器 MPU\_FRGWP

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
位	标记	位名	功能	读写											
b31~b8	Reserved	-	保留位，读出为 0，写入时写 0	R/W											
b7	FRG7WP	FMPU 区域 7 写权限	0: 区域 7 允许 USBFS-DMA 写 1: 区域 7 禁止 USBFS-DMA 写	R/W											
b6	FRG6WP	FMPU 区域 6 写权限	0: 区域 6 允许 USBFS-DMA 写 1: 区域 6 禁止 USBFS-DMA 写	R/W											
b5	FRG5WP	FMPU 区域 5 写权限	0: 区域 5 允许 USBFS-DMA 写 1: 区域 5 禁止 USBFS-DMA 写	R/W											
b4	FRG4WP	FMPU 区域 4 写权限	0: 区域 4 允许 USBFS-DMA 写 1: 区域 4 禁止 USBFS-DMA 写	R/W											
b3	FRG3WP	FMPU 区域 3 写权限	0: 区域 3 允许 USBFS-DMA 写 1: 区域 3 禁止 USBFS-DMA 写	R/W											
b2	FRG2WP	FMPU 区域 2 写权限	0: 区域 2 允许 USBFS-DMA 写 1: 区域 2 禁止 USBFS-DMA 写	R/W											
b1	FRG1WP	FMPU 区域 1 写权限	0: 区域 1 允许 USBFS-DMA 写 1: 区域 1 禁止 USBFS-DMA 写	R/W											
b0	FRG0WP	FMPU 区域 0 写权限	0: 区域 0 允许 USBFS-DMA 写 1: 区域 0 禁止 USBFS-DMA 写	R/W											

## 12.4.20 FMPU 区域读权限寄存器 MPU\_FRGRP

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
位	标记	位名	功能	读写											
b31~b8	Reserved	-	保留位，读出为 0，写入时写 0	R/W											
b7	FRG7RP	FMPU 区域 7 读权限	0: 区域 7 允许 USBFS-DMA 读 1: 区域 7 禁止 USBFS-DMA 读	R/W											
b6	FRG6RP	FMPU 区域 6 读权限	0: 区域 6 允许 USBFS-DMA 读 1: 区域 6 禁止 USBFS-DMA 读	R/W											
b5	FRG5RP	FMPU 区域 5 读权限	0: 区域 5 允许 USBFS-DMA 读 1: 区域 5 禁止 USBFS-DMA 读	R/W											
b4	FRG4RP	FMPU 区域 4 读权限	0: 区域 4 允许 USBFS-DMA 读 1: 区域 4 禁止 USBFS-DMA 读	R/W											
b3	FRG3RP	FMPU 区域 3 读权限	0: 区域 3 允许 USBFS-DMA 读 1: 区域 3 禁止 USBFS-DMA 读	R/W											
b2	FRG2RP	FMPU 区域 2 读权限	0: 区域 2 允许 USBFS-DMA 读 1: 区域 2 禁止 USBFS-DMA 读	R/W											
b1	FRG1RP	FMPU 区域 1 读权限	0: 区域 1 允许 USBFS-DMA 读 1: 区域 1 禁止 USBFS-DMA 读	R/W											
b0	FRG0RP	FMPU 区域 0 读权限	0: 区域 0 允许 USBFS-DMA 读 1: 区域 0 禁止 USBFS-DMA 读	R/W											

## 12.4.21 FMPU 控制寄存器 MPU\_FCR

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								FMPUE	Reserved		FMPUACT[1:0]		FMPUBWP	FMPUBRP	
位	标记	位名	功能	读写											
b31~b8	Reserved	-	保留位, 读出为 0, 写入时写 0	R/W											
b7	FMPUE	FMPU 使能	0: FMPU 无效 1: FMPU 有效	R/W											
b6~b4	Reserved	-	保留位, 读出为 0, 写入时写 0	R/W											
				设定 USBFS-DMA 发生被禁止的访问时的动作											
				00: 无视 (读访问读到 0, 写访问忽略)											
b3~b2	FMPUACT[1:0]	FMPU 动作选择	01: 无视+总线错误 10: 无视+不可屏蔽中断 11: 复位	R/W											
b1	FMPUBWP	FMPU 背景写权限设置	0: FMPU 背景空间允许 USBFS-DMA 写 1: FMPU 背景空间禁止 USBFS-DMA 写	R/W											
b0	FMPUBRP	FMPU 背景读权限设置	0: FMPU 背景空间允许 USBFS-DMA 读 1: FMPU 背景空间禁止 USBFS-DMA 读	R/W											

## 13 键盘扫描控制模块（KEYSCAN）

### 13.1 简介

本产品搭载键盘扫描控制模块（KEYSCAN）1个单元。KEYSCAN 模块支持键盘阵列（行和列）扫描，列是由独立的扫描输出 KEYOUT<sub>m</sub>(m=0~7)驱动，而行 KEYIN<sub>n</sub>(n=0~15)则作为 EIRQ<sub>n</sub>(n=0~15)输入被检测。本模块通过行扫描查询法实现按键识别功能。

#### KEYSCAN 主要特性：

- EIRQ0~EIRQ15 可以独立选择作为键盘阵列的行输入
- KEYOUT 可以通过寄存器选择
- 间隔一定时间依次输出低电平来扫描键盘阵列
- 扫描时间可以设置
- EIRQ 中断检出时停止扫描，根据 SSR.INDEX 值和 IRQ 中断标志（INT\_EIFR.INT\_EIFR）来定位被按下的键

## 13.2 KEYS defense system block diagram

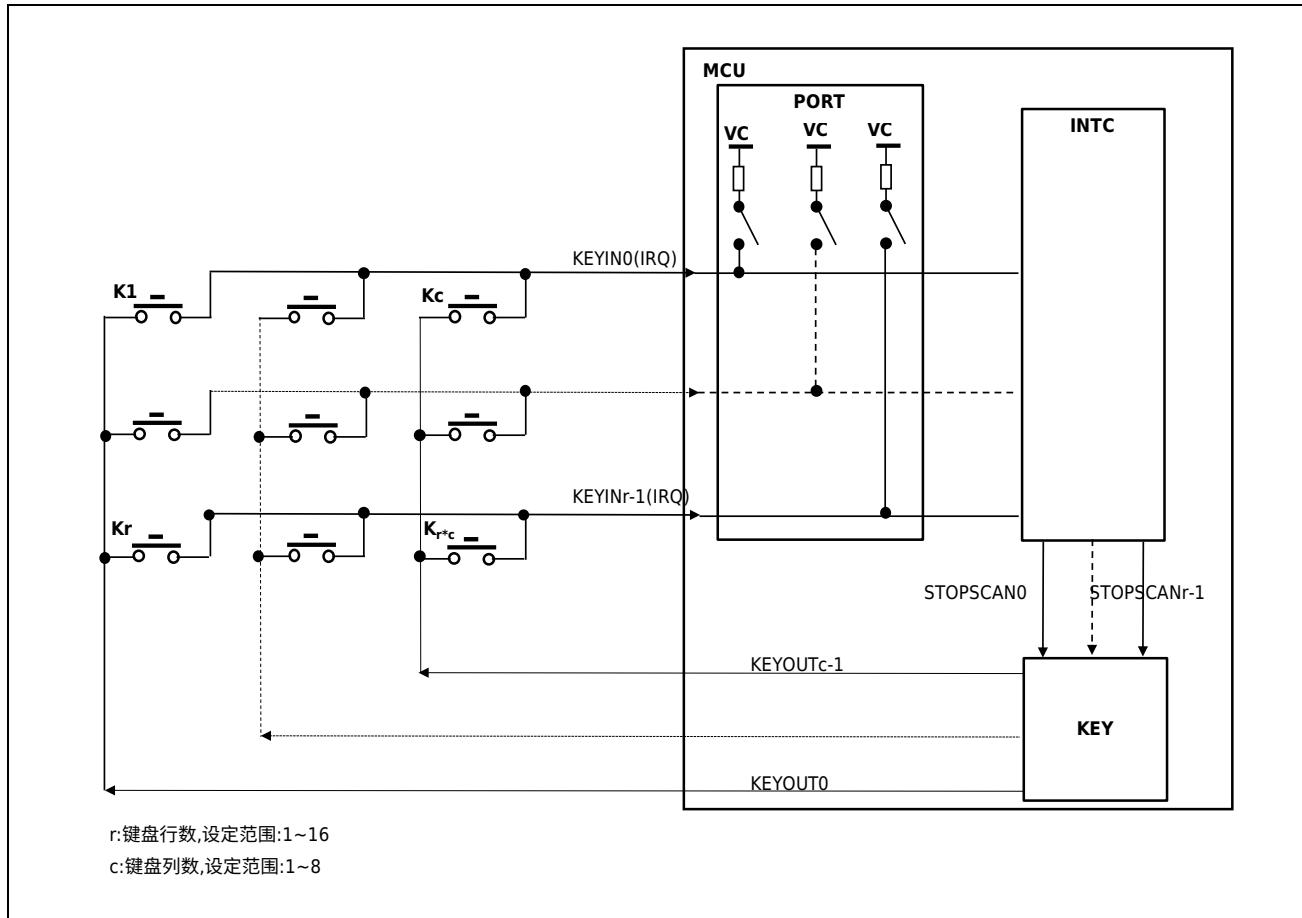


图 13-1 KEYS defense system block diagram

## 13.3 Pin description

表 13-1 KEYS defense pin description

管脚名	方向	功能描述
KEYINn	输入	键盘行输入信号
KEYOUTm	输出	键盘列输出信号

n:0~15 m:0~7

## 13.4 功能说明

本章将对键盘扫描功能和按键识别功能详细说明。

### 13.4.1 按键识别功能

当有键被按下时，键盘的行和列短接，行产生下降沿，从而产生 EIRQ 中断标志，通过比较中断标志位 (INT\_EIFR.INT\_EIFR) 和 SSR.INDEX[2:0] 的值来定位当前被按下的键。

通过寄存器 SCR.KEYINSEL[15:0]，KEYIN 可以从 EIRQ0~EIRQ15 中独立选择，通过寄存器 SCR.KEYOUTSEL[2:0] 可以选择使用的 KEYOUT 管脚，从而可以灵活的选择键盘的行列数目，最大可以支持 16 行\*8 列的键盘阵列。

### 13.4.2 键盘扫描功能

键盘扫描功能为：不断循环的给键盘阵列的列输出低电平，从而当有键被按下时，会产生对应的 EIRQ 中断标志。

当 SER.SEN 设置为 1 后，KEYOUT0 输出低电平，KEYOUT1~KEYOUTn(n 通过 SCR.KEYOUTSEL[2: 0] 设定) 为 HIZ，经过 SCR.T\_LLEVEL[4: 0] 设定的时间后，KEYOUT0~KEYOUTn 管脚全部为 HIZ，经过 SCR.T\_HIZ[2: 0] 设定的时间后，KEYOUT1 输出低电平，其余 KEYOUT 管脚为 HIZ，依次类推循环。有键被按下并产生 EIRQ 中断标志时，键盘扫描功能停止，对应的中断标志清零后，扫描自动重启。

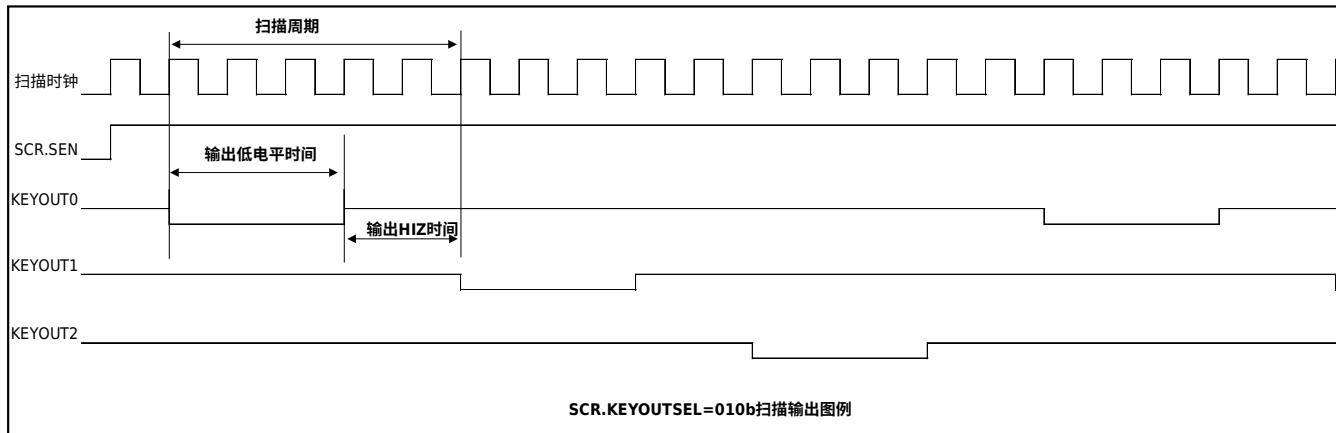


图 13-2 键盘扫描功能示意图

### 13.4.3 使用上的注意事项

本模块驱动键盘列，而键盘行检测则由中断控制模块 (INTC) 的外部 EIRQ 功能实现，EIRQ 需要选择下降沿检测，并开启数字滤波功能，设定合适的滤波时间。

STOP 模式下如果使用此功能，则需设定好扫描相关参数后，选择内部低速振荡器 LRC 或者外部低速振荡器 XTAL32 时钟作为扫描时钟。

如果使用芯片内部上拉电阻，则请参考 PORT 特性，选择合适的扫描时间以及滤波时间。

## 13.5 寄存器说明

表 13-2 KEYS defense 寄存器一览表

KEYSCAN\_BASE\_ADDR: 0x40050C00

寄存器名	符号	偏移地址	位宽	复位值
KEYSCAN扫描控制寄存器	KEYSCAN_SCR	0x00	32	0x0000 0000
KEYSCAN扫描使能寄存器	KEYSCAN_SER	0x04	32	0x0000 0000
KEYSCAN扫描状态寄存器	KEYSCAN_SSR	0x08	32	0x0000 0000

### 13.5.1 KEYS defense control register (KEYSCAN\_SCR)

KEYSCAN Scan Control Register

偏移地址：0x00

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16												
T_HIZ[2:0]		T_LLEVEL[4:0]					-	-	CKSEL[1:0]	-	KEYOUTSEL[2:0]																
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0												
KEYINSEL[15:0]																											
<hr/>																											
位	标记	位名	功能	读写																							
<hr/>																											
b31~b29      T-HIZ[2:0]      输出HIZ时间      KEYOUT输出低电平间HIZ时间 (扫描时钟数) 扫描周期=输出低电平时间+输出HIZ时间 设定值: HIZ周期数 000b: 4 001b: 8 010b: 16 011b: 32 100b: 64 101b: 256 110b: 512 111b: 1024 注意: SCR.T-HIZ[2:0]只能在SER.SEN=0时设定有效																											
<hr/>																											
b28~b24      T_LLEVEL[4:0]      输出低电平时间      KEYOUT输出低电平时间 (扫描时钟数) 扫描周期=输出低电平时间+输出HIZ时间 输出低电平时间=2的T_LLEVEL次幂个扫描时钟数 注意: SCR.T-LLEVEL[4:0]只能在SER.SEN=0时设定有效,且00000b和00001b 设定禁止, 最大可以设定值为11000b																											
<hr/>																											
b23~b22      Reserved      -      读出时为“0”,写入时写“0”																											
<hr/>																											
b21~b20      CKSEL[1:0]      扫描时钟源选择位      扫描时钟源选择位 00b: 系统时钟HCLK 01b: 内部低速振荡器LRC 10b: 外部低速振荡器XTAL32 11b: 设定禁止 注意: SCR.CKSEL[1:0]只能在SER.SEN=0时设定有效																											
<hr/>																											
b19      Reserved      -      读出时为“0”,写入时写“0”																											
<hr/>																											
b18~b16      KEYOUTSEL[2:0]      输出选择      KEYOUT输出选择位 设定值: 输出选择 000b: 禁止 001b: KEYOUT0~KEYOUT1 010b: KEYOUT0~KEYOUT2 011b: KEYOUT0~KEYOUT3 100b: KEYOUT0~KEYOUT4 101b: KEYOUT0~KEYOUT5 110b: KEYOUT0~KEYOUT6																											
<hr/>																											

---

111b: KEYOUT0~KEYOUT7

注意: SCR.KEYOUTSEL[2:0]只能在SER.SEN=0时设定有效

---

行输入选择位, 选择的行作为键盘阵列的行使用, 并作为

EIRQn (n: 0~15) 被检测

b15~b0      KEYINSEL[15:0]      行输入选择位

KEYINSEL[n]=0: KEYINSEL[n]不用作键盘阵列的行使用

R/W

KEYINSEL[n]=1: KEYINSEL[n]用作键盘阵列的行使用

n范围: 0~15

注意: SCR. KEYINSEL[15:0]只能在SER.SEN=0时设定有效

---

### 13.5.2 KEYS defense Scan Enable Register (KEYCAN\_SER)

KEYSCAN Scan Enable Register

偏移地址: 0x04

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

位	标记	位名	功能	读写
b31~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	SEN	扫描使能位	扫描使能位 0: 扫描禁止 1: 扫描使能	R/W

### 13.5.3 KEYS defense Scan Status Register (KEYSCAN\_SSR)

KEYSCAN Scan Status Register

偏移地址: 0x08

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

位	标记	位名	功能	读写
b31~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~b0	INDEX[2:0]	当前工作的 SCAN管脚索引位	当前工作的SCAN管脚索引位 000: 当前工作的SCAN管脚为KEYOUT0 001: 当前工作的SCAN管脚为KEYOUT1 010: 当前工作的SCAN管脚为KEYOUT2 011: 当前工作的SCAN管脚为KEYOUT3 100: 当前工作的SCAN管脚为KEYOUT4 101: 当前工作的SCAN管脚为KEYOUT5 110: 当前工作的SCAN管脚为KEYOUT6 111: 当前工作的SCAN管脚为KEYOUT7	R
			注意: SSR.INDEX[2:0]位为只读寄存器, 且只有在SER.SEN=1时读取的数据才有意义	

## 14 内部时钟校准器 (CTC)

### 14.1 简介

内部时钟校准器 (Clock Trimming Controller, 以下简称 CTC) 用于校准内部高速振荡器 (HRC) 的频率。虽然芯片在出厂时已经对 HRC 的频率做了校准，但由于不同的工作环境，HRC 的频率往往还是会产生偏差而影响系统精度。使用 CTC 可以基于外部的高精度参考时钟自动对 HRC 进一步校准，从而给系统提供一个精准的 HRC 时钟。

#### CTC 的主要特点：

- 硬件自动校准无需软件参与
- 三个外部高精度参考时钟源
- 连续校准和单次校准两种模式
- 可基于出厂校准值进行-32~+31 范围的调整
- 用于提示校准失败的错误中断

### 14.2 结构框图

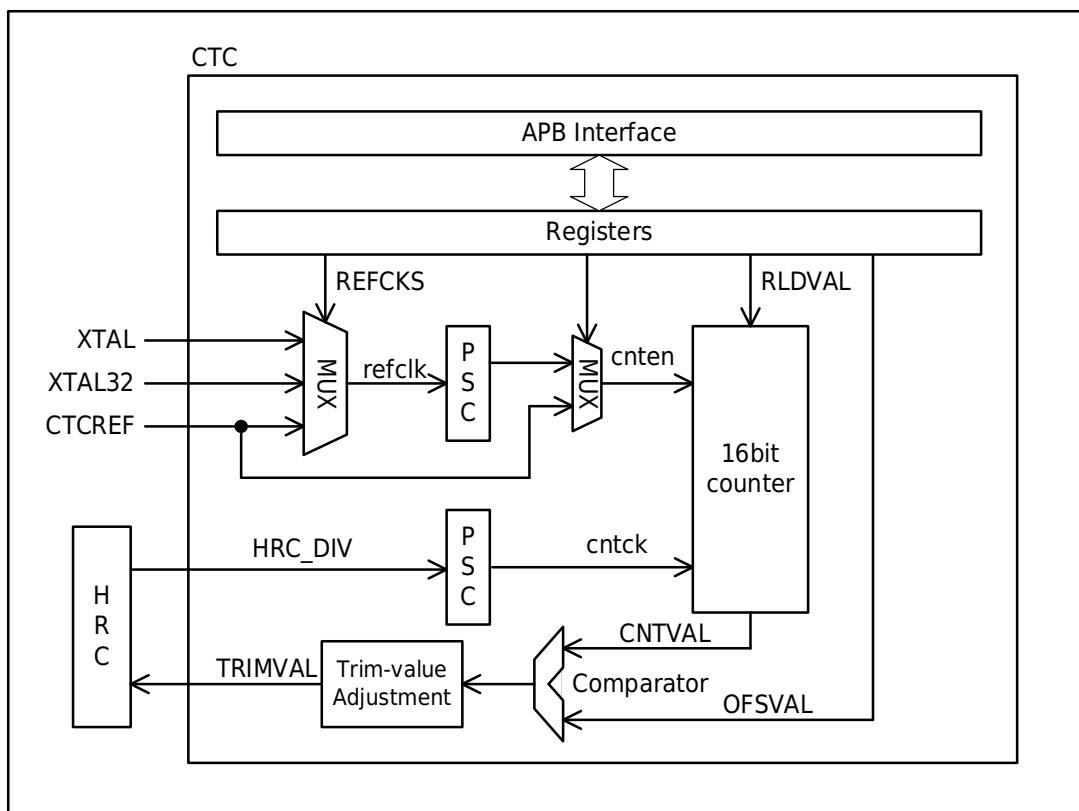


图 14-1 CTC 的基本框图

## 14.3 功能说明

### 14.3.1 参考时钟

CTC 连续校准时有以下三个参考时钟，通过 CTC\_CR1 寄存器的 REFCKS[1:0]位选择。

- 外部高速时钟 (XTAL)
- 外部低速时钟 (XTAL32)
- 外部基准时钟 (CTCREF)

HRC 的 Jitter 和校准过程中的计数误差都会影响校准精度，可以通过设置 CTC\_CR1 寄存器中的 REF\_PSC[2:0]和 HRC\_PSC[2:0]位分别为参考时钟和 HRC 选择合适的分频来减小误差。误差的计算公式如下所示。

$$TE = \frac{0.099\%}{\sqrt{(F_{hrc} \div F_{ref}) \times PSC_{ref} \div 10^5}} + \frac{4}{(F_{hrc} \div F_{ref}) \times PSC_{ref} \div PSC_{hrc}}$$

注：F<sub>hrc</sub> 表示 HRC 的目标频率，单位 MHz；

F<sub>ref</sub> 表示参考时钟的频率，单位 MHz；

PSC<sub>hrc</sub> 表示 HRC 分频数；PSC<sub>ref</sub> 表示参考时钟分频数；

表 14-1，表 14-2 分别列举了 HRC 目标频率为 16MHz 和 20MHz 时使用不同频率的参考时钟并在不同分频条件下校准时的误差值，供快速查阅。

表 14-1 HRC 目标频率为 20MHz 时的测量误差

Pref		1KHz							
PSCref		2	4	8	16	64	256	1024	4096
PSC <sub>hrc</sub>	1	0.167%	不可设						
	2	0.177%	0.121%	不可设	不可设	不可设	不可设	不可设	不可设
	4	0.197%	0.131%	0.088%	不可设	不可设	不可设	不可设	不可设
	8	0.237%	0.151%	0.098%	0.065%	不可设	不可设	不可设	不可设
	32	0.477%	0.271%	0.158%	0.095%	0.038%	不可设	不可设	不可设
	128	1.437%	0.751%	0.398%	0.215%	0.068%	0.024%	不可设	不可设
	512	不推荐	不推荐	1.358%	0.695%	0.188%	0.054%	0.017%	不可设
	2048	不推荐	不推荐	不推荐	不推荐	0.668%	0.174%	0.047%	0.013%
Pref		32.768KHz							
PSCref		2	4	8	16	64	256	1024	4096
PSC <sub>hrc</sub>	1	1.224%	0.797%	0.530%	0.358%	0.169%	不可设	不可设	不可设
	2	1.551%	0.961%	0.612%	0.399%	0.179%	不可设	不可设	不可设
	4	不推荐	1.289%	0.776%	0.481%	0.199%	0.089%	不可设	不可设
	8	不推荐	1.944%	1.103%	0.644%	0.240%	0.100%	不可设	不可设

	32	不推荐	不推荐	不推荐	1.628%	0.486%	0.161%	0.060%	不可设
	128	不推荐	不推荐	不推荐	不推荐	1.469%	0.407%	0.122%	0.040%
	512	不推荐	不推荐	不推荐	不推荐	不推荐	1.390%	0.367%	0.102%
	2048	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.350%	0.347%
Pref		100KHz							
PSCref		2	4	8	16	64	256	1024	4096
PSChrc	1	不推荐	1.607%	1.033%	0.678%	0.308%	0.146%	不可设	不可设
	2	不推荐	不推荐	1.283%	0.803%	0.339%	0.154%	不可设	不可设
	4	不推荐	不推荐	1.783%	1.053%	0.402%	0.170%	0.077%	不可设
	8	不推荐	不推荐	不推荐	1.553%	0.527%	0.201%	0.085%	不可设
	32	不推荐	不推荐	不推荐	不推荐	1.277%	0.388%	0.132%	0.050%
	128	不推荐	不推荐	不推荐	不推荐	不推荐	1.138%	0.319%	0.097%
	512	不推荐	不推荐	不推荐	不推荐	不推荐	1.069%	0.285%	
	2048	不推荐	1.035%						
Pref		1MHz							
PSCref		2	4	8	16	64	256	1024	4096
PSChrc	1	不推荐	不推荐	不推荐	不推荐	1.188%	0.516%	0.238%	不可设
	2	不推荐	不推荐	不推荐	不推荐	1.500%	0.594%	0.258%	0.119%
	4	不推荐	不推荐	不推荐	不推荐	不推荐	0.750%	0.297%	0.129%
	8	不推荐	不推荐	不推荐	不推荐	不推荐	1.063%	0.375%	0.148%
	32	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	0.844%	0.266%
	128	不推荐	0.734%						
	512	不推荐							
	2048	不推荐							
Pref		8MHz							
PSCref		2	4	8	16	64	256	1024	4096
PSChrc	1	不推荐	不推荐	不推荐	不推荐	不推荐	1.863%	0.775%	0.348%
	2	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	0.931%	0.388%
	4	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.244%	0.466%
	8	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.869%	0.622%
	32	不推荐	1.559%						
	128	不推荐							
	512	不推荐							
	2048	不推荐							
Pref		20MHz							
PSCref		2	4	8	16	64	256	1024	4096
PSChrc	1	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.369%	0.587%
	2	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.760%	0.684%
	4	不推荐	0.880%						

	8	不推荐	1.270%						
	32	不推荐							
	128	不推荐							
	512	不推荐							
	2048	不推荐							

注：“不可设”表示该设定超出了校准计数器的范围，无法完成校准。“不推荐”表示该设定误差超过 2%，故不推荐。

**表 14-2 HRC 目标频率为 16MHz 时的测量误差**

Fref		1KHz							
PSCref		2	4	8	16	64	256	1024	4096
PSChrc	1	0.188%	0.130%	不可设	不可设	不可设	不可设	不可设	不可设
	2	0.200%	0.136%	0.094%	不可设	不可设	不可设	不可设	不可设
	4	0.225%	0.149%	0.100%	0.068%	不可设	不可设	不可设	不可设
	8	0.275%	0.174%	0.113%	0.074%	不可设	不可设	不可设	不可设
	32	0.575%	0.324%	0.188%	0.112%	0.043%	不可设	不可设	不可设
	128	1.775%	0.924%	0.488%	0.262%	0.081%	0.028%	不可设	不可设
	512	不推荐	不推荐	1.688%	0.862%	0.231%	0.065%	0.020%	不可设
	2048	不推荐	不推荐	不推荐	不推荐	0.831%	0.215%	0.058%	0.016%
Fref		32.768KHz							
PSCref		2	4	8	16	64	256	1024	4096
PSChrc	1	1.411%	0.913%	0.603%	0.405%	0.190%	不可设	不可设	不可设
	2	1.821%	1.118%	0.706%	0.457%	0.203%	0.095%	不可设	不可设
	4	不推荐	1.528%	0.911%	0.559%	0.228%	0.101%	不可设	不可设
	8	不推荐	不推荐	1.320%	0.764%	0.279%	0.114%	0.051%	不可设
	32	不推荐	不推荐	不推荐	1.993%	0.587%	0.191%	0.070%	0.029%
	128	不推荐	不推荐	不推荐	不推荐	1.815%	0.498%	0.147%	0.048%
	512	不推荐	不推荐	不推荐	不推荐	不推荐	1.727%	0.454%	0.125%
	2048	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.683%	0.432%
Fref		100KHz							
PSCref		2	4	8	16	64	256	1024	4096
PSChrc	1	不推荐	1.863%	1.188%	0.775%	0.348%	0.164%	不可设	不可设
	2	不推荐	不推荐	1.500%	0.931%	0.388%	0.174%	不可设	不可设
	4	不推荐	不推荐	不推荐	1.244%	0.466%	0.194%	0.087%	不可设
	8	不推荐	不推荐	不推荐	1.869%	0.622%	0.233%	0.097%	不可设
	32	不推荐	不推荐	不推荐	不推荐	1.559%	0.467%	0.155%	0.058%
	128	不推荐	不推荐	不推荐	不推荐	不推荐	1.405%	0.390%	0.117%
	512	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.327%	0.351%
	2048	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.289%
Fref		1MHz							
PSCref		2	4	8	16	64	256	1024	4096
PSChrc	1	不推荐	不推荐	不推荐	不推荐	1.369%	0.587%	0.269%	不可设
	2	不推荐	不推荐	不推荐	不推荐	1.760%	0.684%	0.293%	0.134%
	4	不推荐	不推荐	不推荐	不推荐	不推荐	0.880%	0.342%	0.147%
	8	不推荐	不推荐	不推荐	不推荐	不推荐	1.270%	0.440%	0.171%
	32	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.026%	0.318%
	128	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	0.904%

	512	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐
	2048	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐
Fref		8MHz							
PSCref		2	4	8	16	64	256	1024	4096
PSChrc	1	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	0.887%	0.395%
	2	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.082%	0.444%
	4	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.473%	0.541%
	8	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	0.737%	
	32	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.908%	
	128	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	
	512	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	
	2048	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	
Fref		20MHz							
PSCref		2	4	8	16	64	256	1024	4096
PSChrc	1	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.582%	0.669%
	2	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	0.791%	
	4	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.035%
	8	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.523%
	32	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	
	128	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	
	512	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	
	2048	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	

注：“不可设”表示该设定超出了校准计数器的范围，无法完成校准。“不推荐”表示该设定误差超过 2%，故不推荐。

CTC 单次校准时只能选择从 CTCREF 输入基准信号，并且通过 CTC\_CR1 寄存器中的 REFEDG[1:0]位来选择基准信号的有效边沿。此时 REFPSC[2:0]的设定无效，每次校准只在基准信号输入的两个有效边沿间（如上升沿到上升沿，叫做一个校准周期）进行。单次校准的误差通过以下公式计算。

$$TE = \frac{0.099\%}{\sqrt{Fhrc \times Ttrim \div 10^5}} + \frac{4}{Fhrc \times Ttrim \div PSChrc}$$

注：Fhrc 表示 HRC 的目标频率，单位 MHz；

Ttrim 表示校准周期，单位 S；PSChrc 表示 HRC 分频数。

### 14.3.2 频率校准

在 CTC\_CR1 寄存器的 CTCEN 位写为 1 后，16 位校准计数器在参考时钟的触发下从 CTC\_CR2 寄存器中重载 RLDVAL 并开始向下计数，计数时钟由 HRC 提供。计数开始后，如果参考时钟触发计数停止，那么 16 位校准计数器将停止计数。否则 16 位校准计数器将持续计数并在计数值下溢为 0xFFFF 时停止。

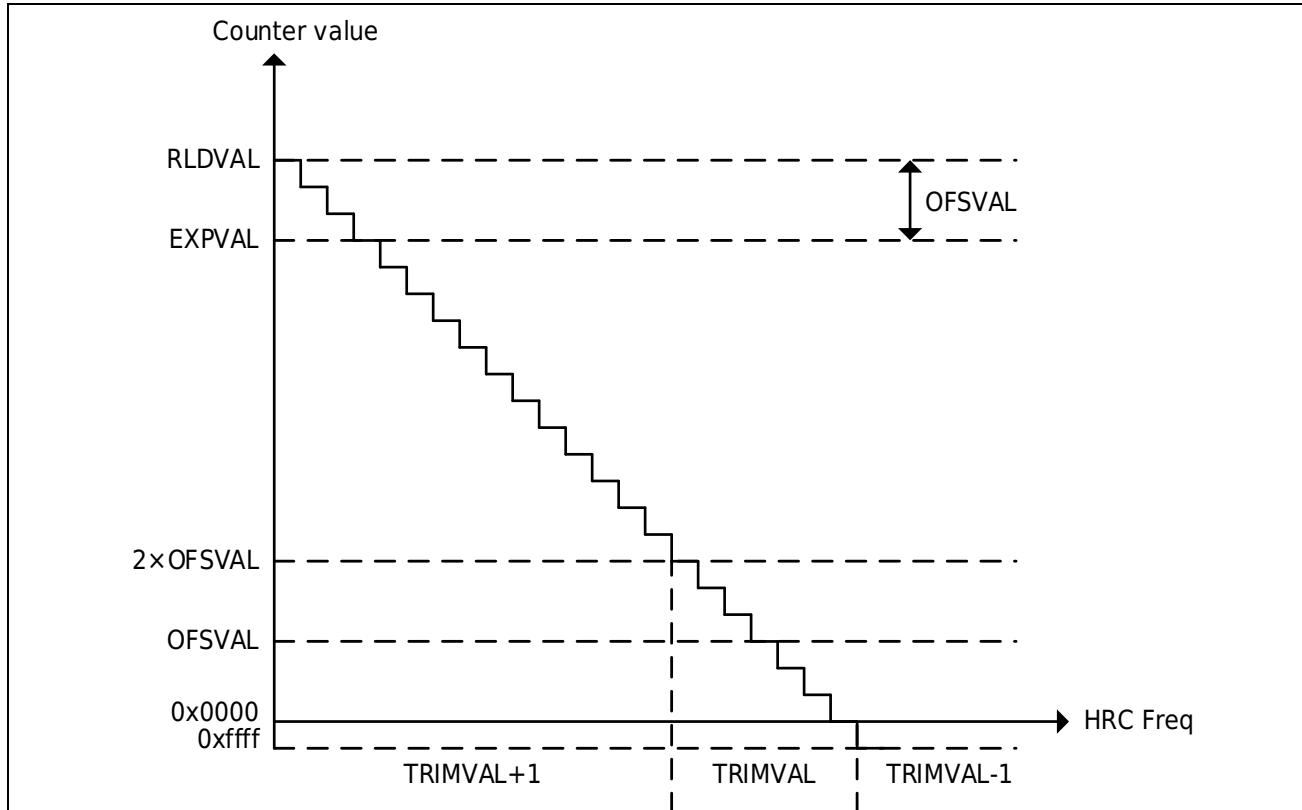


图 14-2 CTC 校准示意图

16 位校准计数器停止计数后，HRC 频率校准开始。由于在设定校准计数重载值 RLDVAL 时加上了校准目标偏差值 OFSVAL，因此如果 16 位校准计数器的值大于  $2 \times \text{OFSVAL}$  且没有发生下溢，说明当前频率比期望频率慢，需要增大 CTC\_CR1 寄存器中 TRMVAL[5:0] 的值来提高频率。反之，如果 16 位校准计数器的值发生下溢，则说明当前频率比期望频率快，需要减小 TRMVAL 的值以降低频率。而如果 16 位校准计数器的值小于  $2 \times \text{OFSVAL}$ ，说明当前频率处于期望频率范围内，就不需要调整 TRMVAL 的值。

- Counter  $> ( \text{OFSVAL} \times 2 )$  时停止计数，TRMVAL 的值自动加 1。
- $0 \leq \text{Counter} \leq (\text{OFSVAL} \times 2)$  时停止计数，TRMVAL 的值不变，CTC\_STR 寄存器的 TRIMOK 位自动变为 1。
- Counter = 0xFFFF，CTC\_CR1 寄存器的 TRMVAL 的值自动减 1。

连续校准在每次 TRMVAL 调整后会自动开始下一次校准，在软件将 CTCEN 清零或者发生校准错误时停止校准。单次校准在每次 TRMVAL 调整后会自动停止，需软件将 CTCEN 写 1 开始下一次校准。

### 14.3.3 错误中断

TRMVAL 的中间值是 0x00。如果自动加 1 的结果大于 0x1F，TRMVAL 的值将保持 0x1F 并发生校准上溢错误，这时 CTC\_STR 寄存器的 TRMOVF 位自动变为 1。反之，如果自动减 1 的结果小于 0x20，TRMVAL 的值将保持 0x20 并发生校准下溢错误，这时 CTC\_STR 寄存器的 TRMUDF 位自动变为 1。发生校准上溢错误或下溢错误时校准会自动停止（CTCEN 自动清零），如果 CTC\_CR1 寄存器中的 ERRIE 位为 1，就会发生一次校准错误中断请求。具体设定方法请参考本手册的《12. 中断控制器》章节。

### 14.3.4 编程指南

以使用 XTAL32 作为参考时钟将 HRC 校准到 16MHz 且校准目标偏差小于 0.5% 为例。这里 XTAL32 设为 16 分频，HRC 设为不分频，经查表 14-2 可知校准误差为 0.405%，小于 0.5% 的校准目标偏差，方案可行。

- 设定寄存器。

CTC\_CR2 = 0x1EAB0027，其中

$$\text{OFSVAL} = (16 \div 0.032768) \times 16 \times 0.5\% \approx 39 \ (0x27);$$

$$\text{RLDVAL} = (16 \div 0.032768) \times 16 + 39 \approx 7851 \ (0x1EAB);$$

- 设定寄存器。

CTC\_CR1 = 0x000000e3，其中

TRMVAL=0x00 (从中间值开始校准)

HRCPSC=000b (选择 HRC 不分频)

CTCEN=1 (启动 CTC)

ERRIE=1 (允许错误中断)

REFCKS=10b (选择 XTAL32 作为参考时钟进行连续校准)

REFPSC=011b (选择参考时钟的 8 分频)

- 一段时间后，确认 CTC\_STR.TRIMOK 位的状态。

如果 TRIMOK=1，CTCEN 写 0 停止 CTC；

如果 TRIMOK=0，回到 (3)；

- 确认 CTC\_STR.CTCBSY=0，后续操作。

途中如果发生错误中断，则表示由于超出可校准范围发生了校准上溢或者校准下溢，CTCEN 位会自动清零，校准停止。这时请修正 OFSVAL，RLDVAL 以及参考时钟设定后再次尝试校准。

需要注意：频率校准是通过 CTC\_CR1 寄存器中的 CTCEN 位来启动或停止的。因此，启动频率校准后，请勿改变任何其他寄存器的设定值。如需修改设置，请先将 CTCEN 位写 0，并确认 CTC\_STR 寄存器中的 CTCBSY 位为 0 后再进行修改。

## 14.4 寄存器说明

表 14-3 CTC 寄存器一览表

基准地址： 0x40049C00

寄存器名	符号	偏移地址	位宽	复位值
时钟校准控制寄存器1	CTC_CR1	0x00	32	0x00000000
时钟校准控制寄存器2	CTC_CR2	0x04	32	0x00000000
时钟校准状态寄存器	CTC_STR	0x08	32	0x00000000
时钟校准计数器	CTC_CNT	0x0c	16	0x0000

#### 14.4.1 时钟校准控制寄存器 1 (CTC\_CR1)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	TRMVAL[5:0]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	REFEDG[1:0]	-	HRCPSC[2:0]			CTCEN	ERRIE	REFCKS[1:0]	-	REFPSC[2:0]				

位	标记	位名	功能	读写
b31~b22	Reserved	-	读出时为“0”,写入时写“0”  TRMVAL为6位有符号校准值，中间值是0。 0x20: -32 0x21: -31 ..... 0x3F: -1 0x00: 0 (中间值) 0x01: +1	R
b21~b16	TRMVAL[5:0]	HRC校准值	..... 0x1E: +30 0x1F: +31  CTCEN位写1时TRMVAL为只读，由硬件自动修改。如需修改请将 CTCEN位写0并确认CTC_STR.CTCBSY位为0后再修改。  在任何情况下HRC最终的频率校准值均为出厂校准值加上TRMVAL 的结果。因此如果出厂校准值下的频率已满足要求无需校准，请将 TRMVAL设为0x00。	R/W
b15~b14	Reserved	-	读出时为“0”,写入时写“0”  用于REFCKS=01b时选择CTCREF的有效边沿。 00b: 选择CTCREF的上升沿到上升沿进行校准 01b: 选择CTCREF的上升沿到下降沿进行校准	R
b13~b12	REFEDG[1:0]	参考信号边沿选择	10b: 选择CTCREF的下降沿到上升沿进行校准 11b: 选择CTCREF的下降沿到下降沿进行校准  CTCEN为1时禁止修改REFEDG的值。如需修改请将CTCEN写0并确 认CTC_STR.CTCBSY为0后再修改。	R/W
b11	Reserved	-	读出时为“0”,写入时写“0”  000b:不分频 001b:2分频 010b:4分频 011b:8分频	R
b10~b8	HRCPSC[2:0]	HRC分频数选择	100b:32分频 101b:128分频 110b:512分频 111b:2048分频  CTCEN为1时禁止修改HRCPSC的值。如需修改请将CTCEN写0并确 认CTC_STR.CTCBSY为0后再修改。	R/W
b7	CTCEN	校准使能	用于启动频率校准。 0: 停止频率校准	R/W

			1: 启动频率校准 置位条件: 软件置位; 清零条件: (1)软件清零; (2)校准发生上溢或者下溢时自动清零; (3)单次校准结束时自动清零;	
b6	ERRIE	错误中断允许	用于选择发生校准上溢或者下溢时是否产生错误中断。 0: 禁止发生错误中断 1: 允许发生错误中断	R/W
b5~b4	REFCKS[1:0]	参考时钟选择	用于为频率校准选择参考时钟源。 00b: 选择CTCREF作为参考时钟 (连续校准) 01b: 选择CTCREF作为参考信号 (单次校准) 10b: 选择XTAL32作为参考时钟 (连续校准) 11b: 选择XTAL作为参考时钟 (连续校准) CTCEN为1时禁止修改REFCKS的值。如需修改请将CTCEN写0并确认CTC_STR.CTCBSY为0后再修改。	R/W
b3	Reserved	-	读出时为“0”,写入时写“0” REFCKS=00b/10b/11b时选择参考时钟的分频数。 000b: 2分频 001b: 4分频 010b: 8分频 011b: 16分频	R
b2~b0	REFPSC[2:0]	参考时钟分频数选择	100b: 64分频 101b: 256分频 110b: 1024分频 111b: 4096分频 CTCEN为1时禁止修改REFPSC的值。如需修改请将CTCEN写0并确认CTC_STR.CTCBSY为0后再修改。	R/W

#### 14.4.2 时钟校准控制寄存器 2 (CTC\_CR2)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RLDVAL[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OFSVAL[7:0]															
<hr/>															
位	标记	位名	功能	读写											
RLDVAL[15:0] 设定的是16位校准计数器的重载值。 连续校准时 RLDVAL = (Fhrc÷PSChrc)÷(Fref÷PSCref)+OFSVAL 单次校准时 RLDVAL = ((Fhrc÷PSChrc)×Ttrim )+OFSVAL Fhrc表示HRC目标频率； Fref表示参考时钟频率； Ttrim表示单次校准周期； OFSVAL表示校准计数偏差值； PSChrc表示HRC分频数； PSCref表示参考时钟分频数； CTCEN为1时禁止修改RLDVAL的值。如需修改请将CTCEN写0并确认CTC_STR.CTCBSY为0后再修改。															
b31~b16	RLDVAL[15:0]	校准计数重载值		R/W											
b15~b8	Reserved	-	读出时为“0”，写入时写“0”	R											
OFSVAL[7:0] 设定的是HRC在校准时允许的偏差值。 连续校准时 OFSVAL = (Fhrc÷PSChrc)÷(Fref÷PSCref)×TA 单次校准时 OFSVAL = ((Fhrc÷PSChrc)×Ttrim )×TA Fhrc表示HRC目标频率； Fref表示参考时钟频率； Ttrim表示单次校准周期； TA表示校准精度，例如0.5%； PSChrc表示HRC分频数； PSCref表示参考时钟分频数； CTCEN为1时禁止修改OFSVAL的值。如需修改请将CTCEN写0并确认CTC_STR.CTCBSY为0后再修改。															
b7~b0	OFSVAL[7:0]	校准目标偏差值		R/W											

#### 14.4.3 时钟校准状态寄存器 (CTC\_STR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

位	标记	位名	功能	读写
b31~b4	Reserved	-	读出时为“0”,写入时写“0”	R
b3	CTCBSY	CTC状态标志	0: CTC处于停止状态 1: CTC处于工作状态	R
b2	TRMUDF	校准下溢标志	校准下溢表示TRMVAL已减至0x20, 但仍未校准到目标范围。校准下溢时自动停止校准, ERRIE为1时将发生错误中断。 0: 未发生校准下溢 1: 发生了校准下溢 TRMUDF在CTCEN写1时清零。	R
b1	TRMOVF	校准上溢标志	校准上溢表示TRMVAL已增至0x1F, 但仍未校准到目标范围。校准上溢时自动停止校准, ERRIE为1时将发生错误中断。 0: 未发生校准上溢 1: 发生了校准上溢 TRMOVF在CTCEN写1时清零。	R
b0	TRIMOK	校准成功标志	0: HRC频率未校准到目标范围 1: HRC频率已校准到目标范围 TRIMOK在CTCEN写1时清零。	R

#### 14.4.4 时钟校准计数器 (CTC\_CNT)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CTCCNT[15:0]															
位	标记	位名	功能	读写											
b15~b0	CTCCNT[15:0]	校准计数器	校准计数器, 每次校准开始时重载RLDVAL的值并以此为起点向下计数。校准结束后计数器值将保持到下一次RLDVAL重载。 请将CTCEN写0并确认CTC_STR.CTCBSY为0后再读。	R											

## 15 DMA 控制器 (DMA)

### 15.1 简介

DMA 用于在存储器和外围功能模块之间传送数据，能够在 CPU 不参与的情况下实现存储器之间，存储器和外围功能模块之间以及外围功能模块之间的数据交换。

- DMA 总线独立于 CPU 总线，按照 AMBA AHB-Lite 总线协议传输
- 拥有 2 个 DMA 控制单元，共 16 个独立通道，可以独立操作不同的 DMA 传输功能
- 每个通道的启动源通过独立的触发源选择寄存器配置
- 每次请求传输一个数据块
- 数据块最小为 1 个数据，最多可以是 1024 个数据
- 每个数据的宽度可配置为 8bit, 16bit 或 32bit
- 可以配置最多 65535 次传输
- 源地址和目标地址可以独立配置为固定，自增，自减，循环或指定偏移量的跳转
- 可产生 3 种中断：块传输完成中断，传输完成中断，传输错误中断。每种中断都可以配置是否屏蔽。其中块传输完成，传输完成可作为事件输出，可作为其它外围模块的触发源
- 支持连锁传输功能，可实现一次请求传输多个数据块
- 支持外部事件触发通道重置
- 不使用时可设置进入模块停止状态以降低功耗
- DMA 访问时 AHB 总线中的 HPROT 值可通过寄存器设置

## 15.2 模块示意图

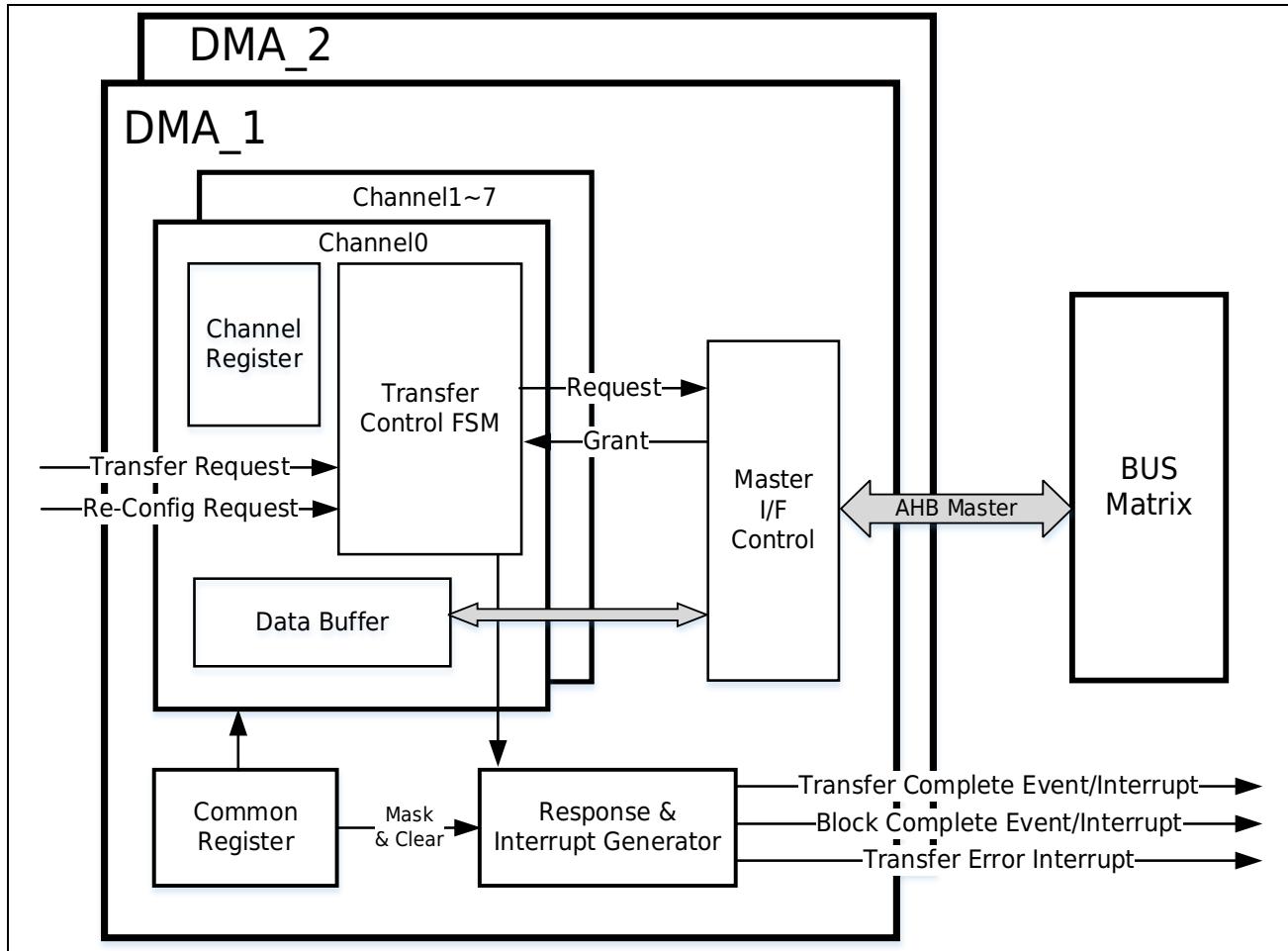


图 15-1 DMA 结构图

## 15.3 功能描述

### 15.3.1 使能 DMA 控制器

使用 DMA 时需要先将功能时钟控制 0 寄存器 FCG0.DMAX 位写 0 允许 DMA 的时钟供给，再将 DMA 使能寄存器 DMA\_EN.EN 位写 1。同时，DMA 需要外围电路来启动传输，所以 FCG0.AOS 外设电路触发功能位也需写 0。

当不使用 DMA，或者芯片需要进入 STOP 模式时，请将 DMA\_EN.EN 设定为 0，并写 FCG0 关闭 DMA 时钟。EN 写 0 前请先确认寄存器 DMA\_CHSTAT.DMAACT 为 0，确保 DMA 已完成所有传输。

### 15.3.2 通道选择和通道优先级

每个 DMA 控制单元中包含 8 个通道，每个通道可以独立配置传输功能。

8 个通道优先级顺序为：通道 0>通道 1>通道 2>...>通道 7。

当一个 DMA 单元有多个通道请求传输时将按照优先级顺序执行。但已处于传输中的通道不会被打断，高优先通道需等当前通道传输完成后才会启动。

### 15.3.3 启动 DMA

DMA 通过外围电路产生的请求启动，这些请求通过触发源选择寄存器 DMA\_TRGSELx(x=0~7)进行配置，各通道独立配置启动请求源。当外围电路产生启动请求或软件写寄存器产生启动请求时，并且 DMA 传输使能有效 DMA\_EN.EN=1，传输通道处于许可状态 DMA\_CHEN.CHEN[x]=1，则启动通道 x 传输。

使用前，需要先将功能时钟控制 0 寄存器(FCG0)的外设电路触发功能及 DMA 功能使能位置为有效。

### 15.3.4 数据块

DMA 每次启动传输的数据量用块(block)表示，块的大小由数据控制寄存器 DMA\_DTCTLx.BLKSIZE 设定，最大可以设置 1024 个数据。每个的数据宽度由 DMA\_CHCTLx.HSIZE 决定，可以是 8bit，16bit 或 32bit。

### 15.3.5 传输地址控制

传输的源地址和目标地址可以由寄存器设定为固定、递增、递减、重载或者不连续跳转。

**固定：**源地址、目标地址将在传输过程中固定不变。

**递增及递减：**源地址、目标地址将在每传输完成 1 个数据后根据 HSIZE 的值进行增减。例如当 HSIZE 为 8bit 时，地址将每次增加/减少 1，为 16bit 时每次增加/减少 2，为 32bit 时每次增加/减少 4。

**重载：**传输指定数量的数据后，源、目标地址将重新返回至最初的地址设定值。地址重载前需要传输的数据量，即重复区域的大小由寄存器 DMA\_RPT 设定。

**不连续地址传输：**传输指定数量的数据后，源、目标地址将跳过指定偏移量。地址跳转的偏移量，以及跳转前需要传输的数据量，即不连续区域的大小，由寄存器 DMA\_SNSEQCTL/DMA\_DNSEQCTL 设定。当地址重载与不连续跳转的条件同时满足时，执行地址重载。

### 15.3.6 传输次数

DMA 传输的总数据块的个数由数据控制寄存器 DMA\_DTCTLx 的 CNT 位设定。传输次数最多可设置 65535 次。每传输一个数据块寄存器值减 1，当寄存器值减为 0 时代表本通道全部数据传输完成，自动清除通道传输许可位 DMA\_CHEN.CHEN[x]，并产生传输完成中断。如果传输开始时 DMA\_DTCTLx.CNT 被设为 0，表示无限次传输，每次启动请求传输一个数据块，但不清除通道传输许可位，也不会产生传输完成中断。

**注意：**

- CHEN[x]自动清 0 后，需要再次启动此通道时，需要重新设置通道配置寄存器后再将 CHEN[x] 写 1 使能本通道。否则将按照上次传输结束时的状态开始后续的传输，即源/目标地址为上次传输对应的下个地址，CNT 为 0 的无限次传输。

### 15.3.7 中断和事件信号输出

DMA 控制器可以产生以下 3 种中断：

数据块完成中断 DMA\_BTCx：完成一个数据块传输后产生。

**传输完成中断 DMA\_TCx：**完成寄存器 DMA\_DTCTLx.CNT 设置的传输次数后产生。

**传输错误中断 DMA\_ERR：**当启动请求溢出（即通道的上一次请求还未响应时此通道再次触发启动请求）时，或者，传输过程中发生总线错误（如访问了非法地址或受保护的地址）时产生中断，其中，总线错误时会立即终止本次传输。

以上中断除了启动请求溢出错误外，其他中断都可以通过寄存器 DMA\_CHCTLx.IE 设置中断的有效或无效。另外所有中断还配有独立的 MASK 寄存器，对中断进行屏蔽。

上述的 DMA\_BTCx，DMA\_TCx 中断同时也可以作为事件信号输出，可用作其它外围电路的触发源，事件输出受 MASK 寄存器控制，但不受中断许可位 DMA\_CHCTLx.IE 控制。DMA\_BTCx，DMA\_TCx，DMA\_ERR 事件发生会将对应的状态位寄存器置位，不受 DMA\_CHCTLx.IE、或 MASK 寄存器影响。

### 15.3.8 连锁传输

DMA 控制器具有连锁传输功能。连锁传输需要配置以下 8 个寄存器共 8 个 word，称为一个描述符(descriptor)，包含了连锁传输的源地址、目标地址、数据控制信息、地址控制信息、链指针以及传输控制信息。

DMA\_SARx  
DMA\_DARx  
DMA\_DTCTLx  
DMA\_RPTx  
DMA\_SNSEQCTLx  
DMA\_DNSEQCTLx  
DMA\_LLPx  
DMA\_CHCTLx

其中 LLP 称为链指针(Linked-List Pointer)，其中的值代表下一个描述符在存储器中的首地址。使用连锁传输时首先写通道控制寄存器 DMA\_CHCTLx 的 LL PEN 将连锁传输使能，并将第一个传输的描述符信息写入相应寄存器中。再将后续传输的描述符按顺序初始化在存储器中。需要结束连锁传输时，将最后一个描述符中 DMA\_CHCTLx 的 LL PEN 置为无效，DMA 控制器将在传输完成后结束连锁传输。

当一个描述符的最后一次传输结束时：

若 LL PEN=1，LL PRUN=0，根据中断许可的配置，产生 BTC 和 TC 中断，通道许可 CHEN[x]不自动清 0。LLP 指定的下一个描述符从存储器中载入通道配置寄存器，等待下一次传输请求输入，开始新描述符的第一次传输。

若 LL PEN=1，LL PRUN=1，不产生 BTC 和 TC 中断，通道许可 CHEN[x]不自动清 0。LLP 指定的下一个描述符从存储器中载入通道配置寄存器后直接开始新描述符的第一次传输。

若 LL PEN=0，则链传输结束，根据中断许可的配置，产生 BTC 和 TC 中断，通道许可 CHEN[x]自动清 0。

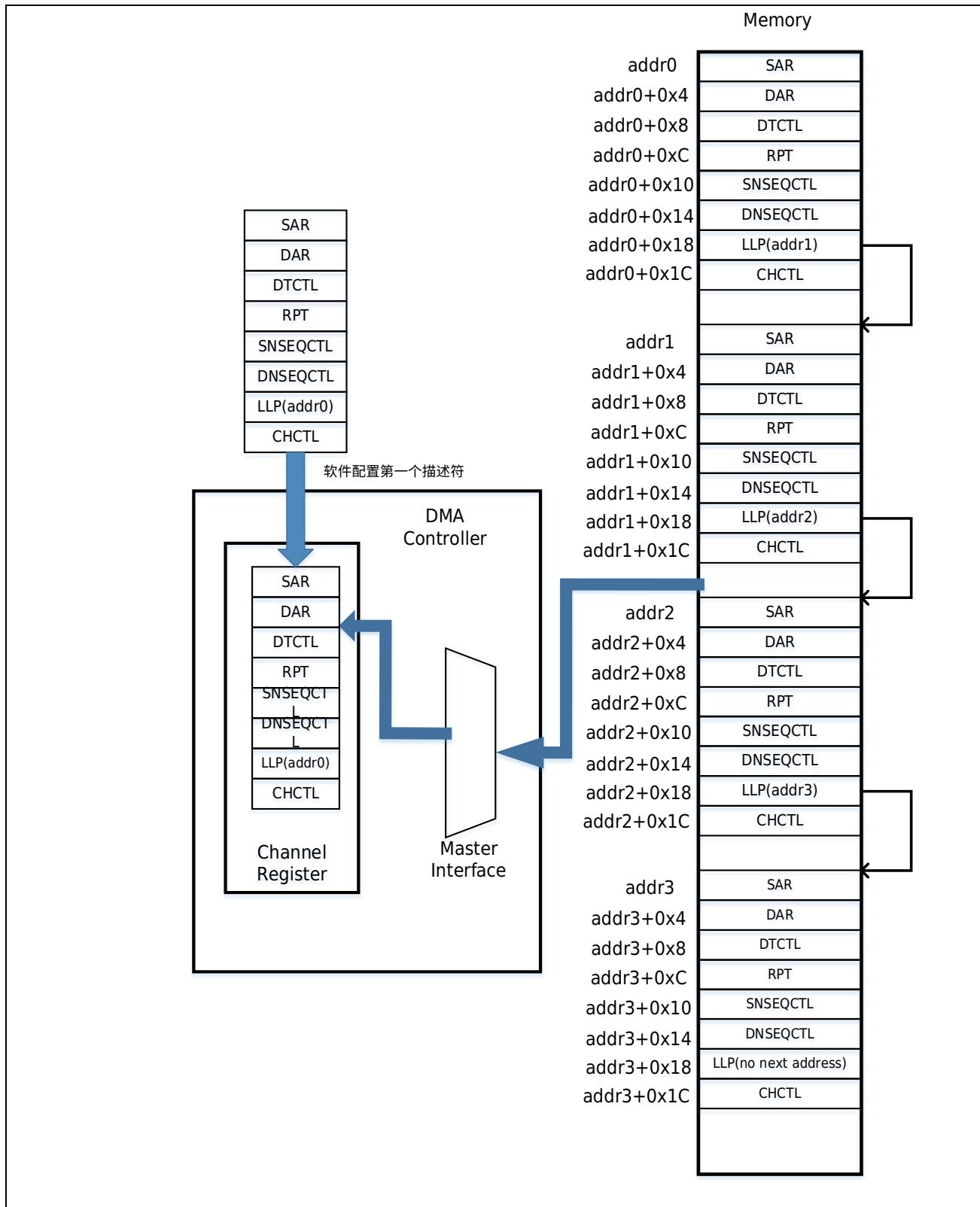


图 15-2 连锁传输示意图

### 15.3.9 不连续地址传输

使用不连续地址传输能够实现源地址和目标地址在传输完一定量的数据后按照一定的偏移量跳转。以源地址不连续传输为例：首先根据需要将通道控制寄存器 DMA\_CHCTLx.SNSEQEN 置 1 使不连续地址传输有效，跳转的方向按照 DMA\_CHCTLx.SINC 的设定向前或向后跳转，然后配置寄存器 DMA\_SNSEQCTLx 设置具体的跳转方式。目标地址的控制与源地址相似，对应控制寄存器分别为 DMA\_CHCTLx.DNSEQEN、DMA\_CHCTLx.DINC、DMA\_DNSEQCTLx。传输过程按照下图方式进行。

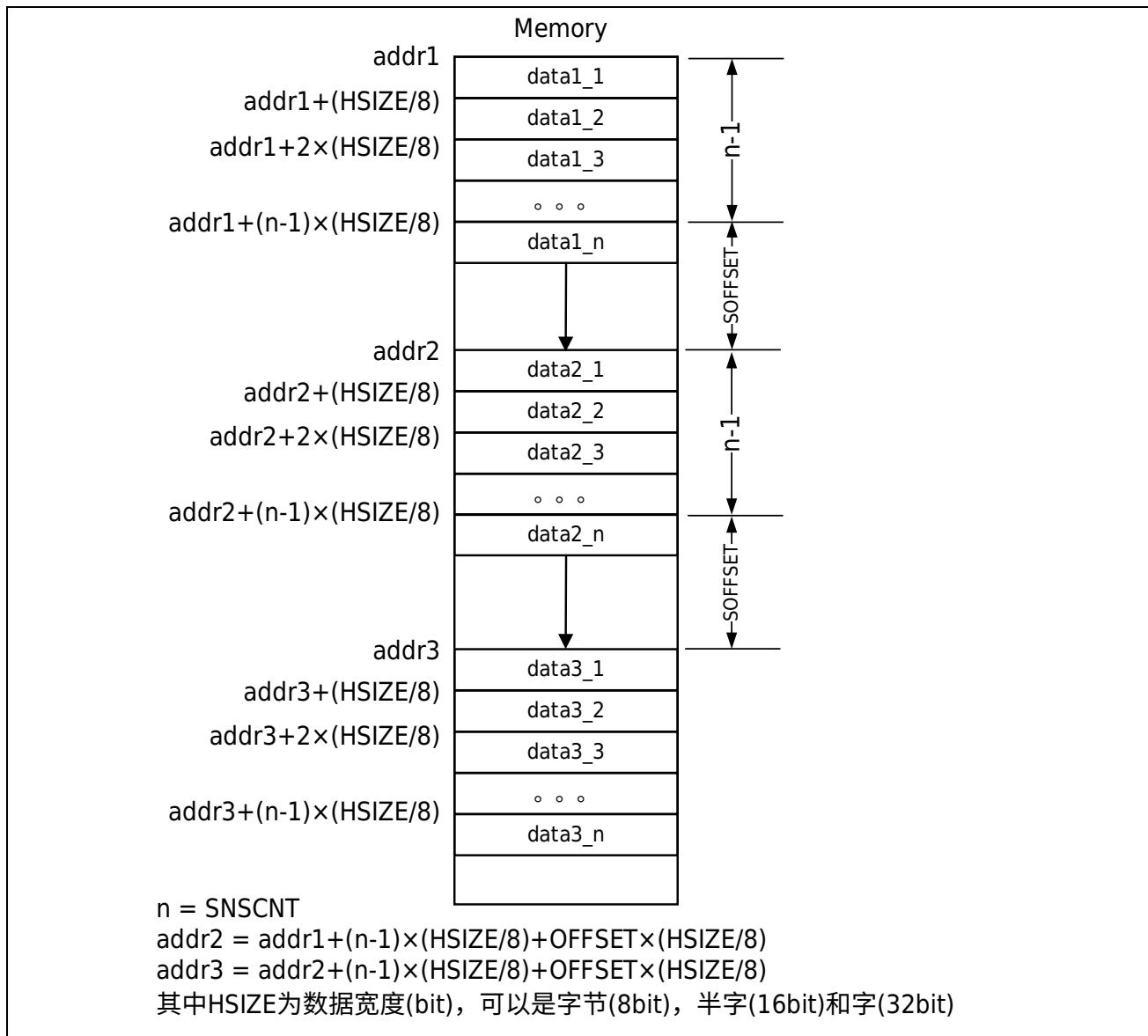


图 15-3 不连续地址传输示意图（以源地址为例）

### 15.3.10 通道重置

通道重置功能是指通过外围电路的事件请求来修改通道内部状态寄存器，重新配置下一次数据的传输方式。设置寄存器 DMA\_RCFGCTL.RCFGGEN 为 1，允许通道重置。通过触发源选择寄存器 DMA\_RC\_TRGSEL 选择重置请求源。当选择的重置请求源输入时，寄存器 DMA\_RCFGCTL.RCFGCHS 选择的通道会按指定的方式进行更新。重置请求只更新内部状态，不会启动实际的数据传输。

通道的重置方式有如下三种：链指针式、不连续式、重复式。

选择链指针式重置时，通道的描述符和内部状态全部更新为链指针 LLP 指向的新描述符。之后的传输请求，按新描述符进行传输。

选择不连续式、重载式重置时，通道内部状态按下表所述进行更新。

表 15-1 通道重置说明

通道内部状态	重置方式	
	不连续式	重载式
剩余传输次数计数器	更新为正常状态下，下一次地址不连续跳转发生后的数值	更新为正常状态下，下次重载发生后的数值
下次传输的源/目标地址	更新为下一个不连续传输区域的首地址	更新为寄存器DMA_SARx/DARx的初始设定值

注意：

- 当重置功能有效时，该通道使用寄存器 DMA\_RPTBx 和 DMA\_SNSEQCTLBx，DMA\_DNSEQCTLBx 控制传输地址的重载和不连续跳转。寄存器 DMA\_RPTx 和 DMA\_SNSEQCTLx, DMA\_DNSEQCTLx 无效。

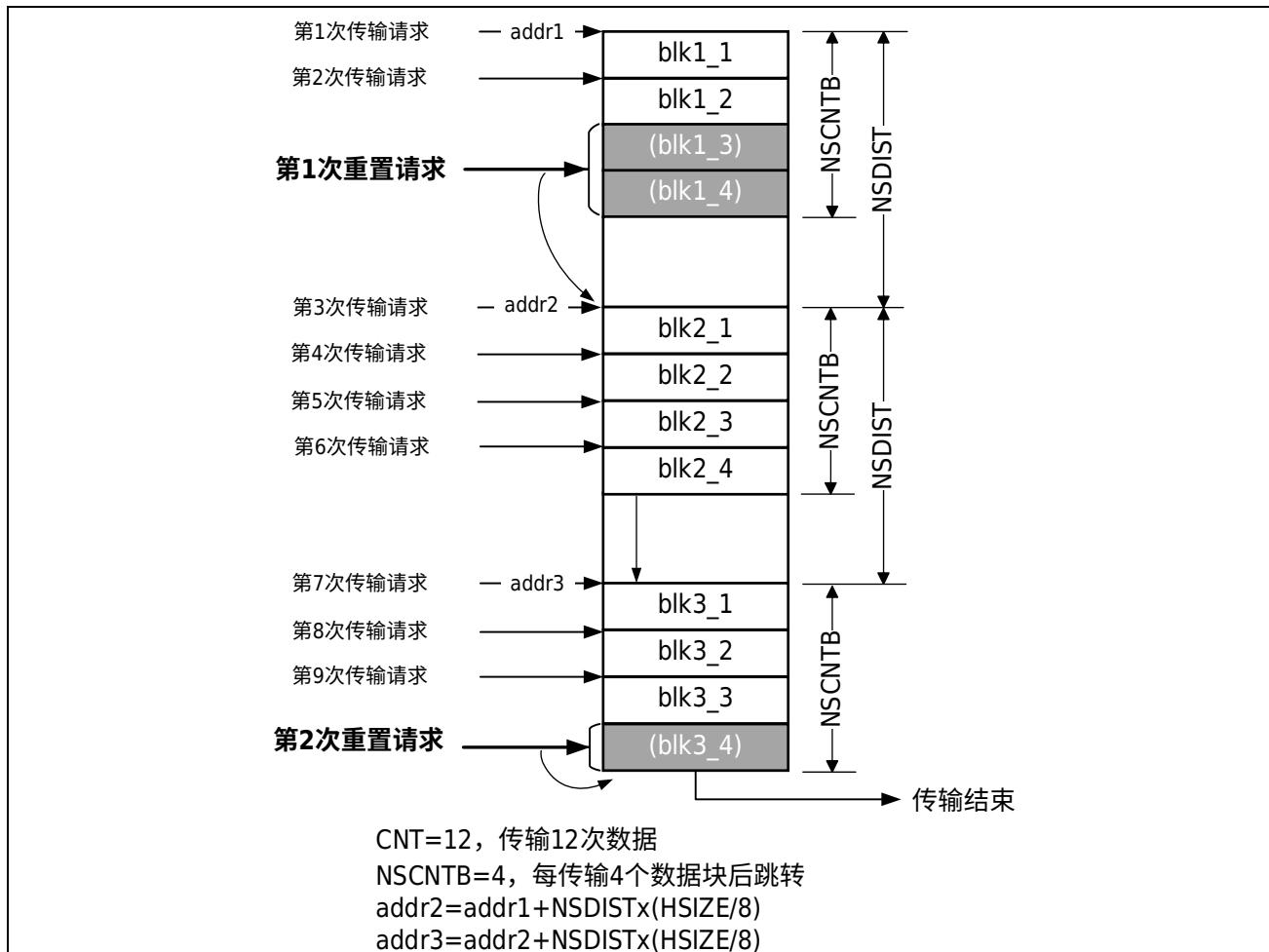


图 15-4 不连续式重置示意图

图 15-4 所示 DMA 动作中，每次传输请求启动一个数据块的传输。第 1 次重置请求发生后，控制器跳过了数据块 blk1\_3,blk1\_4，传输地址更新为下个不连续区域的首地址即 addr2。第 2 次重置请求发生后，剩余传输次数更新为 0，即所有数据传输完成，通道许可位自动清 0，产生传输完成中断和事件。

### 15.3.11 传输提前终止

传输过程中通道使能寄存器 DMA\_CHEN.CHENx 保持有效，非连锁传输时，数据控制寄存器 DMA\_DTCTLx 设定的传输次数完成后自动置为无效，连锁传输时，最后一次连锁传输的传输次数完成后自动置为无效。传输过程中如果软件对 DMA\_CHENCLR.CHENCLRx 写 1，则 DMA 将在完成当次数据读写后终止传输，且 DMA\_CHEN.CHENx 位清 0。

#### 注意：

- 软件对 CHENCLRx 位写 1 提前终止传输时，DMA 内部不会保存被终止时的传输状态。在没有重新设置通道配置寄存器（描述符）状态，将 CHENx 写 1 再次允许此通道，传输请求输入后，DMA 将重新传输被终止的数据块，而不是断点续传。

## 15.4 应用举例

### 15.4.1 存储器到存储器的传输

目标：将 12 个数据从 RAM 地址 0x20000000 传送到 0x20001000，数据宽度为 32bit。

1. 寄存器设定
  - DMA\_EN.EN 写 1 使能 DMA 控制器
  - 选择一个通道，例如通道 0，配置通道寄存器以实现：
    - 写 DMA\_SAR0 配置源地址为 SRAM 区域 0x20000000
    - 写 DMA\_DAR0 配置目标地址为 SRAM 区域 0x20001000
    - 写 DMA\_DCTRL0 配置数据块的大小为 4，传输次数为 3 次，每次传输完 1 个数据块后产生块传输完成中断，3 次传输完成后产生一个传输完成中断
    - 写 DMA\_RPT 寄存器配置源地址重复区域大小为 6，即传输完成 6 个地址后重新载入初始源地址
    - 配置通道控制寄存器 DMA\_CH0CTL 以实现：
      - \* 源地址和目标地址连锁传输无效
      - \* 源地址重载有效，和目标地址更新方式为自增
      - \* 数据宽度为 word(32bit)
      - \* 中断使能有效
    - 配置触发源控制器 DMA\_TRGSEL0，选择软件触发作为 DMA 通道 0 的启动请求
    - 通道使能位 DMA\_CHEN.CHENO 写 1，使能通道 0
  - 写外设事件软件触发寄存器 INTSFTTRG STRG 为 1，发送第一个软件启动请求，DMA 开始传输数据

## 2. 传输过程

由于传输的数据块大小为 4，当软件写 INTSFTTRG STRG 为 1 后开始第一次传输，当一个数据块传输完成后，传输次数 DMA\_DTCTL0.CNT 减 1，并产生一个块传输完成中断，软件可以在中断子程序中继续写 INTSFTTRG STRG 开始第二次传输。在第二次传输中，由于设置了源地址重复区域大小为 6，因此在传输完 2 个地址后源地址将重新载入初始地址 0x20000000 并继续传输剩余的 2 个地址。第二次传输完成后，传输次数 DMA\_DTCTL0.CNT 减 1，并产生一个块传输完成中断，软件可以在中断子程序中继续写 INTSFTTRG STRG 开始第三次传输。第三次传输完成后，传输次数 DMA\_DTCTL0.CNT 减为 0，即本次传输全部完成，DMA 产生一个块传输完成中断和一个传输完成中断，同时通道使能位 DMA\_CHEN.CHENO 将被自动清零。

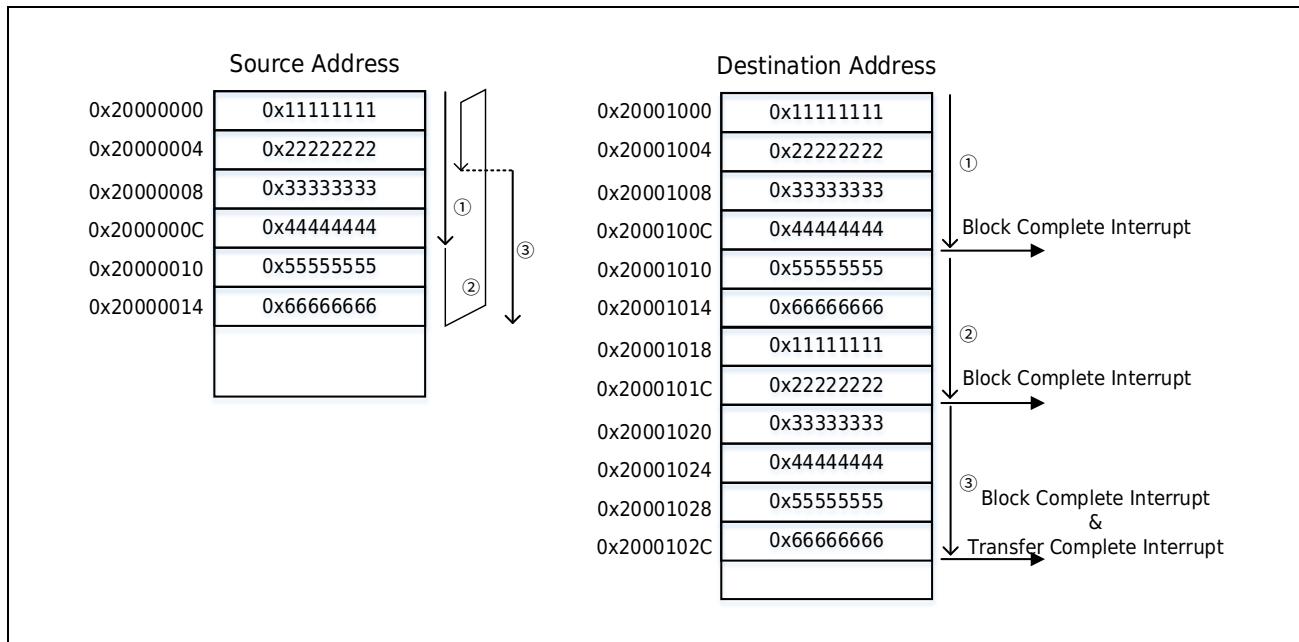


图 15-5 应用例 1：存储器到存储器传输

### 15.4.2 存储器到外围电路的传输

目标:将 10 个宽度为 half-word 的数据从 RAM 地址 0x20000000 传送到通信模块的发送缓冲寄存器中，通信模块每发送完一个数据产生一次传送请求。当最后一个数据发送完成后 DMA 产生一个传输完成中断。

#### 1. 寄存器设定

- DMA\_EN.EN 写 1 使能 DMA 控制器
- 配置 DMA\_INTMSK 寄存器，屏蔽块传输完成中断，允许传输完成中断
- 选择一个通道，配置通道寄存器，例如选择通道 0
  - 写 DMA\_SAR0 配置源地址为 SRAM 区域 0x20000000
  - 写 DMA\_DAR0 配置目标地址为外围电路的寄存器地址 0x40000000
  - 写 DMA\_DTCTL0 配置数据块的大小为 1，传输次数为 10 次，每个传输请求传送一次，每次 1 个数据。
  - 配置通道控制寄存器 DMA\_CH0CTL 以实现：
    - \* 源地址和目标地址连锁传输无效
    - \* 源地址更新方式为自增，目标地址为固定
    - \* 源地址和目标地址数据宽度为半字(16bit)
    - \* 中断使能有效
  - 配置触发源控制器 DMA\_TRGSEL0，选择通信模块的发送寄存器为空事件作为 DMA 通道 0 的启动请求
  - 通道使能位 DMA\_CHEN.CHENO 写 1，使能通道 0

## 2. 传输过程

通道使能后，DMA 等待来自通信模块的传输请求。当传输请求产生后，DMA 把 RAM 中的数据传送至通信模块的发送缓冲寄存器中，并等待第二次来自通信模块的传输请求，由于屏蔽了块传输完成中断，DMA 此时不产生中断请求。当 10 个数据全部传送完成后，DMA 产生一个传输完成中断，并且通道使能位 DMA\_CHEN.CHENO 将被自动清零。

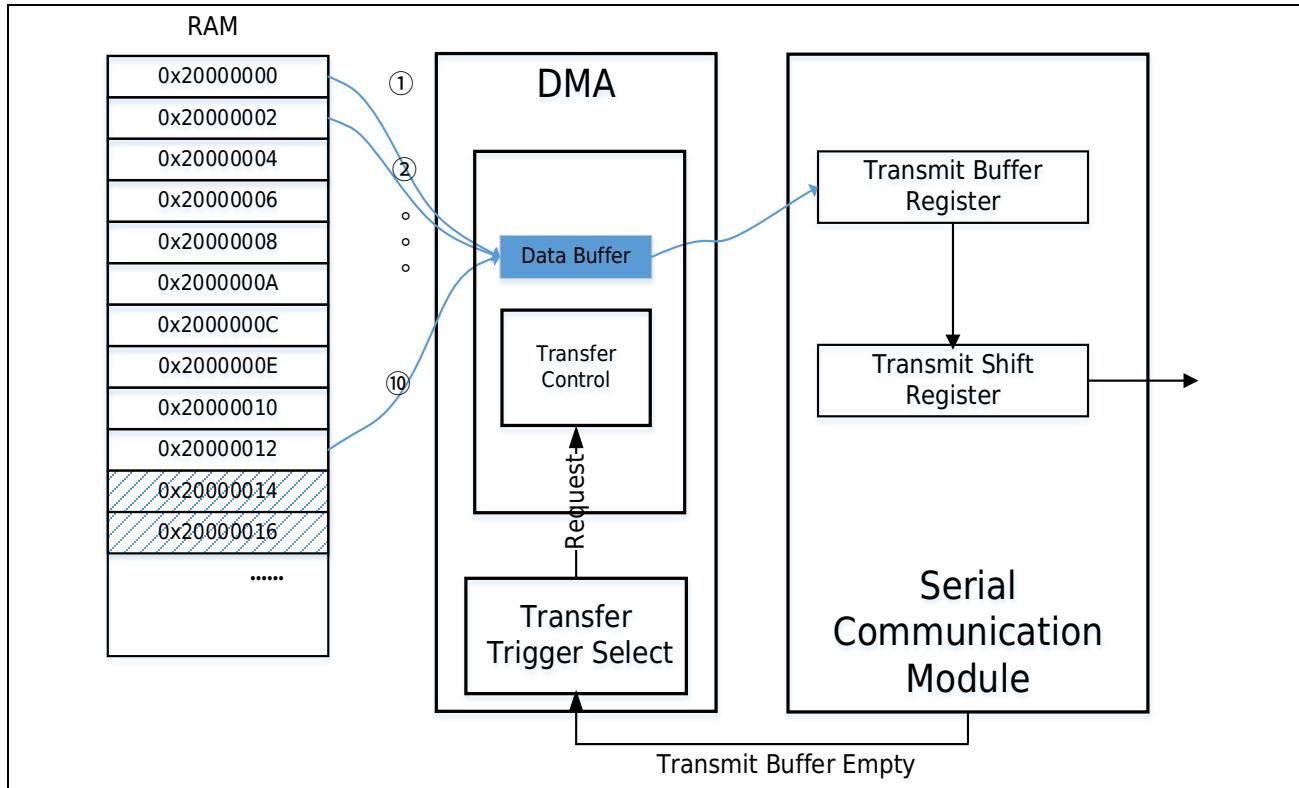


图 15-6 应用例 2：存储器到外围电路的传输

### 15.4.3 存储器到存储器的链锁传输

#### 1. 寄存器设定

- DMA\_EN.EN 写 1 使能 DMA 控制器
- 选择一个通道，配置通道寄存器，例如选择通道 0，配置第一次传输的描述符(descriptor0)
  - 写 DMA\_SAR0 配置源地址为 SRAM 区域 0x20000000
  - 写 DMA\_DAR0 配置目标地址为 SRAM 区域 0x20001000
  - 写 DMA\_DTCTL0 配置数据块的大小为 10
  - 在链指针寄存器 DMA\_LLPO 中写入第二个描述符(descriptor1)的地址 0x20002000
  - 配置通道控制寄存器 DMA\_CH0CTL，配置第一个数据块的传输参数以实现：
    - \* 连锁传输有效
    - \* 连锁传输模式为直接启动下一次传输
    - \* 源地址和目标地址更新方式为自增
    - \* 数据宽度为字(32bit)

- \* 中断使能无效
- 在 RAM 空间的 0x20002000 地址中配置第二次传输的描述符(descriptor1)，包括：
  - 0x20002000 中写入 32 位数据 0x20000100，此为第二次传输的源地址
  - 0x20002004 中写入 32 位数据 0x20001100，此为第二次传输的目标地址
  - 0x20002008 中配置数据块的大小为 20
  - 0x20002018 中写入 32 位数据 0x20002020，此为第三次传输的描述符(descriptor2)所在地址
  - 0x2000201C 中写入第二次传输的控制数据，该控制数据实现：
    - \* 连锁传输有效
    - \* 连锁传输模式为直接启动下一次传输
    - \* 源地址和目标地址更新方式为自增
    - \* 数据宽度为半字(16bit)
    - \* 中断使能无效
- 在 RAM 空间的 0x20002020 地址中配置第三次传输的描述符(descriptor2)，包括：
  - 0x20002020 中写入 32 位数据 0x20000200，此为第三次传输的源地址
  - 0x20002024 中写入 32 位数据 0x20001200，此为第三次传输的目标地址
  - 0x20002028 配置数据块的大小为 40
  - 0x20002038 中写入 32 位数据 0x0，代表该次传输为连鎖传输的最后一次传输
  - 0x2000203C 中写入第三次传输的控制数据实现：
    - \* 连鎖传输无效
    - \* 源地址和目标地址更新方式为自增
    - \* 数据宽度为字节(8bit)
    - \* 中断使能有效
- 配置传输启动触发源选择寄存器 DMA\_TRGSEL0，选择软件触发作为 DMA 通道 0 的启动请求
- 通道使能位 DMA\_CHEN.CHENO 写 1，使能通道 0
- 写软件触发寄存器 INTSFTTRG STRG 为 1，发送一个启动请求，DMA 开始传输数据

## 2. 传输过程

软件启动 DMA 开始传输。第一次传输完成后，由于设置了连鎖传输模式为直接启动下一次传输并且中断无效，DMA 将第二次传输的描述符(descriptor1)读入到通道寄存器中。根据描述符配置的参数直接开始第二次传输。第二次传输完成后，将第三次传输的描述符(descriptor2)读入到通道寄存器中。根据描述符配置的参数开始第三次传输。第三次传输完成后，根据配置信息该次为连鎖传输的最后一次，且由于中断使能有效，因此 DMA 将产生一个传输完成中断，并清除通道使能位 DMA\_CHEN.CHENO。

## 15.5 寄存器说明

### 15.5.1 寄存器一览

表 15-2 寄存器一览表

单元 1 BASE\_ADDR:0x40053000

单元 2 BASE\_ADDR:0x40053400

寄存器名	符号	偏移地址	位宽	复位值
DMA使能寄存器	DMA_EN	0x00	32	0x00000000
中断状态寄存器0	DMA_INTSTAT0	0x04	32	0x00000000
中断状态寄存器1	DMA_INTSTAT1	0x08	32	0x00000000
中断屏蔽寄存器0	DMA_INTMASK0	0x0C	32	0x00000000
中断屏蔽寄存器1	DMA_INTMASK1	0x10	32	0x00000000
中断复位寄存器0	DMA_INTCLR0	0x14	32	0x00000000
中断复位寄存器1	DMA_INTCLR1	0x18	32	0x00000000
通道使能寄存器	DMA_CHEN	0x1c	32	0x00000000
通道使能复位寄存器	DMA_CHENCLR	0x34	32	0x00000000
传输请求状态寄存器	DMA_REQSTAT	0x20	32	0x00000000
通道状态观测寄存器	DMA_CHSTAT	0x24	32	0x00000000
通道重置控制寄存器	DMA_RCFGCTL	0x2c	32	0x00000000
传输源地址寄存器	DMA_SARx *1	0x40+0x40*x	32	0x00000000
传输目标地址寄存器	DMA_DARx	0x44+0x40*x	32	0x00000000
数据控制寄存器	DMA_DTCTLx	0x48+0x40*x	32	0x00000001
重复区域大小寄存器	DMA_RPTx	0x4C+0x40*x	32	0x00000000
重复区域大小寄存器B	DMA_RPTBx			
源设备不连续地址传输控制寄存器	DMA_SNSEQCTLx	0x50+0x40*x	32	0x00000000
源设备不连续地址传输控制寄存器B	DMA_SNSEQCTLBx			
目标设备不连续地址传输控制寄存器	DMA_DNSEQCTLx	0x54+0x40*x	32	0x00000000
目标设备不连续地址传输控制寄存器B	DMA_DNSEQCTBx			
链指针寄存器	DMA_LLPx	0x58+0x40*x	32	0x00000000
通道控制寄存器	DMA_CHCTLx	0x5C+0x40*x	32	0x00001000
传输源地址监视寄存器	DMA_MONSARx	0x60+0x40*x	32	0x00000000
传输目标地址监视寄存器	DMA_MONDARx	0x64+0x40*x	32	0x00000000
数据控制监视寄存器	DMA_MONDTCTLx	0x68+0x40*x	32	0x00000001
重复区域计数器监视寄存器	DMA_MONRPTx	0x6C+0x40*x	32	0x00000000
源设备不连续传输计数器监视寄存器	DMA_MONSNSEQCTLx	0x70+0x40*x	32	0x00000000
目标设备不连续传输计数器监视寄存器	DMA_MONDNSEQCTLx	0x74+0x40*x	32	0x00000000

### 15.5.2 DMA 使能寄存器(DMA\_EN)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	EN
位	标记	位名	功能				读写								
b31-b1	Reserved	-	读出时为“0”,写入时写“0”				R/W								
b0	EN	DMA使能位	0: DMA无效 1: DMA使能				R/W								

### 15.5.3 中断状态寄存器 0(DMA\_INTSTAT0)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16							
-	-	-	-	-	-	-	-	-	REQERR[7:0]													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0							
-	-	-	-	-	-	-	-	-	TRNERR[7:0]													
位	标记	位名	功能				读写															
b31-b24	Reserved	-	读出时为“0”,写入时写“0”				R/W															
b23-b16	REQERR[7:0]	传输请求溢出错误中断位	0: 该通道没有发生传输请求溢出错误 1: 该通道发生了传输请求溢出错误, 即上次请求还处在等待状态 时又来了传输请求				R															
b15-b8	Reserved	-	读出时为“0”,写入时写“0”				R/W															
b7-b0	TRNERR[7:0]	传输错误中断位	0: 该通道没有发生传输错误 1: 该通道发生了传输错误				R															

### 15.5.4 中断状态寄存器 1(DMA\_INTSTAT1)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	BTC[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	TC[7:0]							
<hr/>															
位	标记	位名	功能	读写											
b31-b24	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b23-b16	BTC[7:0]	块传输完成中断位	该中断在完成一个数据块的传输后发生 0: 该通道没有块传输中断发生 1: 该通道发生块传输中断	R											
b15-b8	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b7-b0	TC[7:0]	传输完成中断位	该中断在完成传输次数寄存器DMA_CNTx设置的传输次数后发生 0: 该通道没有传输完成中断发生 1: 该通道发生传输完成中断	R											

### 15.5.5 中断屏蔽寄存器 0(DMA\_INTMASK0)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	MSKREQERR[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	MSKTRNERR[7:0]							
<hr/>															
位	标记	位名	功能	读写											
b31-b24	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b23-b16	MSKREQERR[7:0]	传输请求溢出中断屏蔽	0: 不屏蔽传输请求溢出中断 1: 屏蔽传输请求溢出中断	R/W											
b15-b8	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b7-b0	MSKTRNERR[7:0]	传输错误中断屏蔽	0: 不屏蔽传输错误中断 1: 屏蔽传输错误中断	R/W											

### 15.5.6 中断屏蔽寄存器 1(DMA\_INTMASK1)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	MSKBTC[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	MSKTC[7:0]							

位	标记	位名	功能	读写
b31-b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23-b16	MSKBTC[7:0]	块传输完成中断 屏蔽	0: 不屏蔽块传输完成中断 1: 屏蔽块传输完成中断	R/W
b15-b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7-b0	MSKTC[7:0]	传输完成中断屏蔽	0: 不屏蔽传输完成中断 1: 屏蔽传输完成中断	R/W

### 15.5.7 中断复位寄存器 0(DMA\_INTCLR0)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	CLRREQERR[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	CLRTRNERR[7:0]							

位	标记	位名	功能	读写
b31-b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23-b16	CLRREQERR[7:0]	传输请求溢出错误 误中断复位	写0无任何效果, 写1复位传输请求溢出错误中断状态位 读出永远为0	R/W
b15-b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7-b0	CLRTRNERR[7:0]	传输错误中断复位	写0无任何效果, 写1复位传输错误中断状态位 读出永远为0	R/W

### 15.5.8 中断复位寄存器 1(DMA\_INTCLR1)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16							
-	-	-	-	-	-	-	-	CLRBTC[7:0]														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0							
-	-	-	-	-	-	-	-	CLRTC[7:0]														

位	标记	位名	功能	读写
b31-b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23-b16	CLRBTC[7:0]	块传输完成中断 复位	写0无任何效果, 写1复位块传输完成中断状态位 读出永远为0	R/W
b15-b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7-b0	CLRTC[7:0]	传输完成中断复位	写0无任何效果, 写1复位传输完成中断状态位 读出永远为0	R/W

### 15.5.9 通道使能寄存器(DMA\_CHEN)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16							
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0							
-	-	-	-	-	-	-	-	CHEN[7:0]														

位	标记	位名	功能	读写
b31-b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7-b0	CHEN[7:0]	通道使能位	每位对应一通道。 0: 该通道无效 1: 该通道有效 对CHEN[x]写1, CHEN[x]置1。对CHEN[x]写0无效。 对DMA_CHENCLR.CHENCLR[x]位写1时, CHEN[x]清0, 并强制终止通道x正在进行的传输。传输过程中使能位保持为1, 当传输完传输次数寄存器DMA_DTCTLx.CNT设置的传输次数后将会自动清0。如果DMA_DTCTLx.CNT设置为0, 则传输完成后不会自动清零, 即无限次传输。 注意: 请在通道x的8个描述符寄存器全部设置结束后将CHEN[x]写1, 使能本通道。 CHEN[x]为1时, 对通道x的描述符寄存器的写操作无效。	R/W

### 15.5.10 通道使能复位寄存器(DMA\_CHENCLR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CHENCLR[7:0]															

位	标记	位名	功能	读写
b31-b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7-b0	CHENCLR[7:0]	通道使能位复位	写0无任何效果, 写1复位通道使能位CHEN 读出永远为0 写1时若本通道正在传输, 则DMA将在完成当次数据读写后强制终止传输。	R/W

### 15.5.11 通道重置控制寄存器(DMA\_RCFGCTL)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	CNTMD[1:0]	DARMD[1:0]	SARMD[1:0]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-		RCFGCHS[3:0]		-	-	-	-	-	-	-	RCFGLP	RCFGEN

位	标记	位名	功能	读写
b31-b22	Reserved	-	读出时为“0”,写入时写“0”	R/W
			00: 保持不变 01: 按源地址方式 10, 11: 按目标地址方式	
b21-b20	CNTMD[1:0]	剩余传输次数计数器重置方式	当选择源地址方式时, 且源地址选择不连续式重置, 则剩余传输次数计数器更新为DMA_SNSEQCTLBx.SNSCNTB指定传输次数后的状态; 源地址选择重复式时, 剩余传输次数计数器更新为DMA_RPTBx.SRPTB指定传输次数后的状态。源地址选择保持时, 剩余传输次数计数器也保持不变。 当选择目标地址方式时, 与源地址方式类似。	R/W
			00: 保持不变 01: 不连续式重置	
b19-b18	DARMD[1:0]	目标地址重置方式	下次传输的目标地址更新为 addr_base + (DNSDIST x HSIZE(bit)/8) 其中: addr_base表示当前不连续传输区域的首地址 10, 11: 重复式重置 下次传输的目标地址更新为DMA_DARx寄存器初始设置值。 注意: 本通道目标地址不连续传输有效 (DMA_CHCTLx.DNSEQEN=1) 时, DARMD[1:0]必须设为0b01。	R/W
			00: 保持不变 01: 不连续式重置	
b17-b16	SARMD[1:0]	源地址重置方式	下次传输的源地址更新为 addr_base + (SNSDIST x HSIZE(bit)/8) 其中: addr_base表示当前不连续传输区域的首地址 10, 11: 重复式重置 下次传输的源地址更新为DMA_SARx寄存器初始设置值。 注意: 本通道源地址不连续传输有效 (DMA_CHCTLx.SNSEQEN=1) 时, SARMD[1:0]必须设为0b01。	R/W
b15-b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
			0x0: 通道0 0x1: 通道1	
b11-b8	RCFGCHS[3:0]	重置通道选择	依此类推 0x7: 通道7 其它: Reserved, 设定禁止	R/W
b7-b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
			0: 链指针式重置无效 1: 链指针式重置有效	
b1	RCFGLLP	链指针式通道重置	注意: RCFGPLL设为1时, 通道将重新载入存储器中的新描述符, 故本寄存器的bit16-bit21全部无效。	R/W

---

b0	RCFGEN	通道重置许可	0: 禁止事件触发通道配置寄存器强制更新 1: 允许事件触发通道配置寄存器强制更新	R/W
----	--------	--------	----------------------------------------------	-----

---

**注意：**

- 请在 DMA\_EN.EN 为 0 时设置本寄存器，本寄存器必须在重置通道的第一次传输前设置。

### 15.5.12 传输请求状态寄存器(DMA\_REQSTAT)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RCFGREQ
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CHREQ[7:0]
<hr/>															
位	标记	位名	功能	读写											
b31-b17	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b16	RCFGREQ	通道重置请求标志	当外部重置请求输入后置1, 当通道重置启动后，或者通道重置禁止时清0。 0：无通道重置请求 1：有通道重置请求	R											
b15-b8	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b7~b0	CHREQ[7:0]	通道传输请求标志位	每位对应一个通道。 当外部传输请求输入后对应位置1, 当该通道传输启动后，发生传输出错，或传输许可位(DMAEN或 CHEN[x])写0时，该位清0。 当该位为1状态，该通道传输请求再次输入，则发生传输请求溢出 错误，第二次的请求被忽略。 0：该通道无传输请求 1：该通道有传输请求	R											

### 15.5.13 通道状态观测寄存器(DMA\_CHSTAT)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CHACT[7:0]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RCFGACT DMAACT
<hr/>															
位	标记	位名	功能	读写											
b31-b24	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b23-b16	CHACT[7:0]	传输动作中通道监视位	每位对应一个通道。 0：该通道空闲中 1：该通道处于动作中	R											
b15-b2	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b1	RCFGACT	DMA通道重置动作中监视位	0：DMA未处于通道重置动作中 1：DMA处于通道重置动作中	R											
b0	DMAACT	DMA动作中监视位	0：DMA未处于传输动作中 1：DMA处于传输动作中	R											

### 15.5.14 传输源地址寄存器(DMA\_SARx) (x=0~7)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SAR[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SAR[15:0]															

位	标记	位名	功能	读写
			设置传输源地址	
b31-b0	SAR[31:0]	传输源地址	注意: 传输数据宽度为16bit时，即DMA_CHCTLx.HSIZE=01，SAR[0]无效。 传输数据宽度为32bit时，即DMA_CHCTLx.HSIZE=1x，SAR[1:0]无效。	R/W

### 15.5.15 传输目标地址寄存器(DMA\_DARx) (x=0~7)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DAR[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DAR[15:0]															

位	标记	位名	功能	读写
			设置传输目标地址	
b31-b0	DAR[31:0]	传输目标地址	注意: 传输数据宽度为16bit时，即DMA_CHCTLx.HSIZE=01，DAR[0]无效。 传输数据宽度为32bit时，即DMA_CHCTLx.HSIZE=1x，DAR[1:0]无效。	R/W

### 15.5.16 数据控制寄存器(DMA\_DTCTLx) (x=0~7)

复位值: 0x00000001

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16							
CNT[15:0]																						
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0							
-	-	-	-	-	-	-	BLKSIZE[9:0]															

位	标记	位名	功能	读写
b31-b16	CNT[15:0]	传输次数	总的传输次数，每次请求启动一个数据块的传输，完成时传输次数计数器减1，当减到0时发生传输完成中断。如果设置为0，表示无限次传输，每次启动请求传输一个数据块，完成时传输次数计数器保持0不变，不会产生传输完成中断。	R/W
b15-b10	Reserved	-	读出时为“0”，写入时写“0”	R/W
b9-b0	BLKSIZE[9:0]	数据块的大小	设置数据块的大小，最大可以配置1024个数据。每个数据的宽度由DMA_CHCTLx寄存器的HSIZE位决定。寄存器值设为1则每次传输1个数据，设为0则每次传输1024个数据。	R/W

### 15.5.17 重复区域大小寄存器(DMA\_RPTx) (x=0~7)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	DRPT[9:0]									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	SRPT[9:0]									
位	标记	位名	功能										读写		
b31-b26	Reserved	-	读出时为“0”,写入时写“0”										R/W		
b25-b16	DRPT[9:0]	目标地址重复区域大小	设置目标地址重复区域大小 目标设备在每传输DRPT个数据后目标地址重载为DMA_DARx寄存器初始设定值。寄存器设为10则每传输10个数据后地址重载，设为0则每传1024个数据后地址重载。每个数据的宽度由DMA_CHCTLx寄存器的HSIZE位决定。										R/W		
b15-b10	Reserved	-	读出时为“0”,写入时写“0”										R/W		
b9-b0	SRPT[9:0]	源地址重复区域大小	设置源地址重复区域大小 源设备在每传输SRPT个数据后源地址重载为DMA_SARx寄存器初始设定值。寄存器设为10则每传输10个数据后地址重载，设为0则每传1024个数据后地址重载。每个数据的宽度由DMA_CHCTLx寄存器的HSIZE位决定。										R/W		

该寄存器配置源地址和目标地址重复区域的大小。使用重复地址传输需要配置 DMA\_CHCTLx 寄存器的 SRPTEN/DRPREN 位有效，并且配置 DMA\_CHCTLx 寄存器的 SINC/DINC 位使地址更新方式为自增或自减，如果为固定则地址重载功能无效。

DMA\_RPTx, DMA\_RPTBx 这两个寄存器共用同一个地址，都是用来定义重复区域大小的。要使用哪一个取决于该通道是否启用了重置功能。未启用重置时使用 DMA\_RPTx，启用重置时使用 DMA\_PRTBx。

### 15.5.18 重复区域大小寄存器 B(DMA\_RPTBx) (x=0~7)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16				
-	-	-	-	-	-	DRPT[9:0]													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0				
-	-	-	-	-	-	SRPT[9:0]													
<hr/>																			
位	标记	位名	功能	读写															
b31-b26	Reserved	-	读出时为“0”,写入时写“0”	R/W															
b25-b16	DRPTB[9:0]	目标地址重复区域大小	设置目标地址重复区域大小 目标设备在每传输DRPTB个数据块后目标地址重载为DMA_DARx寄存器初始设定值。数据块大小由DMA_DTCTLx.BLKSIZE与DMA_CHCTLx.HSIZE决定。 注意：本通道重置功能有效（DMA_RCFGCTL.RCFGEN=1）时，DRPTB[9:0]的设定值必须大于0。	R/W															
b15-b10	Reserved	-	读出时为“0”,写入时写“0”	R/W															
b9-b0	SRPTB[9:0]	源地址重复区域大小	设置源地址重复区域大小 源设备在每传输SRPTB个数据块后源地址重载为DMA_SARx寄存器初始设定值。数据块大小由DMA_DTCTLx.BLKSIZE与DMA_CHCTLx.HSIZE决定。 注意：本通道重置功能有效（DMA_RCFGCTL.RCFGEN=1）时，SRPTB[9:0] 的设定值必须大于0。	R/W															

该寄存器配置源地址和目标地址重复区域的大小。使用重复地址传输需要配置 DMA\_CHCTLx 寄存器的 SRPTEN/DRPREN 位有效，并且配置 DMA\_CHCTLx 寄存器的 SINC/DINC 位使地址更新方式为自增或自减，如果为固定则重复地址传输功能无效。

DMA\_RPTx, DMA\_RPTBx 这两个寄存器共用同一个地址，都是用来定义重复区域大小的。要使用哪一个取决于该通道是否启用了重置功能。未启用重置时使用 DMA\_RPTx，启用重置时使用 DMA\_PRTBx。

### 15.5.19 源设备不连续地址传输控制寄存器(DMA\_SNSEQCTLx) (x=0~7)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SNSCNT[11:0]												SOFFSET[19:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SOFFSET[15:0]															

位	标记	位名	功能	读写
b31-b20	SNSCNT[11:0]	源地址跳转的数据量	设置源地址跳转前传输的数据量大小。 源设备在每传输SNSCNT个数据后源地址按SOFFSET指定偏移量跳转。寄存器设为10则每传输10个数据后地址跳转，设为0则每传输4096个数据后地址跳转。	R/W
b19-b0	SOFFSET[19:0]	源地址跳转的地址偏移量	当不连续地址传输时，设置源地址跳转的偏移量。偏移量是相对于当前传输地址，即跳转前最后一次传输地址而言的。跳转的方向根据通道控制寄存器DMA_CHCTLx.SINC的值向前或向后跳转。参考图 15-3。当DMA_CHCTLx.SINC设为地址固定时，不连续地址传输无效。  跳转地址将根据数据的宽度(DMA_CHCTLx.HSIZE)设置的bit数和SOFFSET的值进行计算。  地址偏移量=SOFFSET×(HSIZE(bit)/8)  例如，当SOFFSET设为10，HSIZE为字(32bit)时，地址偏移量为 $10 \times 4 = 40$ ，若HSIZE为半字(16bit)时，偏移量为 $10 \times 2 = 20$ ，若HSIZE为字节(8bit)时，偏移量为 $10 \times 1 = 10$ 。  下次传输的源地址=当前传输的源地址±地址偏移量	R/W

使用源设备不连续传输需要配置 DMA\_CHCTLx 寄存器的 SNSEQEN 位有效，并且配置 DMA\_CHCTLx 寄存器的 SINC 位使地址更新方式为自增或自减。

DMA\_SNSEQCTLx, DMA\_SNSEQCTLBx 这两个寄存器共用同一个地址，都是用来定义不连续传输的。要使用哪一个取决于该通道是否启用了重置功能。未启用重置时使用 DMA\_SNSEQCTLx，启用重置时使用 DMA\_SNSEQCTLBx。

### 15.5.20 源设备不连续地址传输控制寄存器 B(DMA\_SNSEQCTLBx) (x=0~7)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SNSCNTB[11:0]												SNSDIST[19:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SNSDIST[15:0]															

位	标记	位名	功能	读写
b31-b20	SNSCNTB[11:0]	源地址跳转的数据量	设置源地址跳转前传输的数据量大小。 源设备在每传输SNSCNTB个数据块后源地址按SNSDIST指定地址间距跳转。数据块大小由DMA_DTCTLx.BLKSIZE与DMA_CHCTLx.HSIZE决定。 注意：本通道重置功能有效（DMA_RCFGCTL.RCFGEN=1）时，SNSCNTB[11:0]的设定值必须大于0。	R/W
b19-b0	SNSDIST[19:0]	源不连续区域地址间距	当不连续地址传输时，设置源设备两个不连续区域的间距。 跳转的方向根据通道控制寄存器DMA_CHCTLx.SINC的值向前或向后跳转。参考图 15-4。当DMA_CHCTLx.SINC设为地址固定时，不连续地址传输无效。 地址间距将根据数据的宽度(DMA_CHCTLx.HSIZE)设置的bit数和SNSDIST的值进行计算。 地址间距=SNSDIST×(HSIZE(bit)/8) 例如，当SNSDIST设为10，HSIZE为字(32bit)时，地址间距为 $10 \times 4 = 40$ ，若HSIZE为半字(16bit)时，间距为 $10 \times 2 = 20$ ，若HSIZE为字节(8bit)时，间距为 $10 \times 1 = 10$ 。 下次传输的源地址=当前源不连续区域首地址+地址间距	R/W

使用源设备不连续传输需要配置 DMA\_CHCTLx 寄存器的 SNSEQEN 位有效，并且配置 DMA\_CHCTLx 寄存器的 SINC 位使地址更新方式为自增或自减。

DMA\_SNSEQCTLx, DMA\_SNSEQCTLBx 这两个寄存器共用同一个地址，都是用来定义不连续传输的。要使用哪一个取决于该通道是否启用了重置功能。未启用重置时使用 DMA\_SNSEQCTLx，启用重置时使用 DMA\_SNSEQCTLBx。

### 15.5.21 目标设备不连续地址传输控制寄存器(DMA\_DNSEQCTLx) (x=0~7)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DNSCNT[11:0]												DOFFSET[19:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DOFFSET[15:0]															

位	标记	位名	功能	读写
b31-b20	DNSCNT[11:0]	目标地址跳转的数据量	设置目标地址跳转前传输的数据量大小。 目标设备在每传输DNSCNT个数据后目标地址按DOFFSET指定偏移量跳转。寄存器设为10则每传输10个数据后地址跳转，设为0则每传4096个数据后地址跳转。	R/W
b9-b0	DSOFFSET[19:0]	目标地址跳转的地址偏移量	当不连续地址传输时，设置目标地址跳转的偏移量。偏移量是相对于当前传输地址，即跳转前最后一次传输地址而言的。跳转的方向根据通道控制寄存器DMA_CHCTLx.DINC的值向前或向后跳转。参考图 15-3。当DMA_CHCTLx.DINC设为地址固定时，不连续地址传输将无效。  跳转地址将根据数据的宽度(DMA_CHCTLx.HSIZE)设置的bit数和DOFFSET的值进行计算。  地址偏移量=DOFFSET×(HSIZE(bit)/8)  例如，当DOFFSET设为10，HSIZE为字(32bit)时，地址偏移量为 $10 \times 4 = 40$ ，若HSIZE为半字(16bit)时，偏移量为 $10 \times 2 = 20$ ，若HSIZE为字节(8bit)时，偏移量为 $10 \times 1 = 10$ 。  下次传输的目标地址=当前传输的目标地址±地址偏移量	R/W

使用目标设备不连续传输需要配置 DMA\_CHCTLx 寄存器的 DNSEQEN 位有效，并且配置 DMA\_CHCTLx 寄存器的 DINC 位使地址更新方式为自增或自减。

DMA\_DNSEQCTLx, DMA\_DNSEQCTLBx 这两个寄存器共用同一个地址，都是用来定义不连续传输的。要使用哪一个取决于该通道是否启用了重置功能。未启用重置时使用 DMA\_DNSEQCTLx，启用重置时使用 DMA\_DNSEQCTLBx。

### 15.5.22 目标设备不连续地址传输控制寄存器 B(DMA\_DNSEQCTLBx) (x=0~7)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DNSCNTB[11:0]												DNSDIST[19:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DNSDIST[15:0]															

位	标记	位名	功能	读写
b31-b20	DNSCNTB[11:0]	目标地址跳转的数据量	设置目标地址跳转前传输的数据量大小。 目标设备在每传输DNSCNTB个数据块后目标地址按DNSDIST指定地址间距跳转。数据块大小由DMA_DTCTLx.BLKSIZE与DMA_CHCTLx.HSIZE决定。 注意：本通道重置功能有效（DMA_RCFGCTL.RCFGEN=1）时，DNSCNTB[11:0]的设定值必须大于0。	R/W
b19-b0	DNSDIST[19:0]	目标不连续区域地址间距	当不连续地址传输时，设置目标设备两个不连续区域的间距。 跳转的方向根据通道控制寄存器DMA_CHCTLx.DINC的值向前或向后跳转。参考图 15-4。当DMA_CHCTLx.DINC设为地址固定时，不连续地址传输将无效。 地址间距将根据数据的宽度(DMA_CHCTLx.HSIZE)设置的bit数和DNSDIST的值进行计算。 地址间距=DNSDIST×(HSIZE(bit)/8) 例如，当DNSDIST设为10，HSIZE为字(32bit)时，地址间距为 $10 \times 4 = 40$ ，若HSIZE为半字(16bit)时，间距为 $10 \times 2 = 20$ ，若HSIZE为字节(8bit)时，间距为 $10 \times 1 = 10$ 。 下次传输的目标地址=当前目标不连续区域首地址+地址间距	R/W

使用目标设备不连续传输需要配置 DMA\_CHCTLx 寄存器的 DNSEQEN 位有效，并且配置 DMA\_CHCTLx 寄存器的 DINC 位使地址更新方式为自增或自减。

DMA\_DNSEQCTLx, DMA\_DNSEQCTLBx 这两个寄存器共用同一个地址，都是用来定义不连续传输的。要使用哪一个取决于该通道是否启用了重置功能。未启用重置时使用 DMA\_DNSEQCTLx，启用重置时使用 DMA\_DNSEQCTLBx。

### 15.5.23 链指针寄存器(DMA\_LL\_Px) (x=0~7)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
LLP[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LLP[15:2]															
<hr/>															
位	标记	位名	功能	读写											
b31-b2	LLP[31:2]	链指针	连鎖传输有效时，设置下一次传输的描述符所在的地址，地址为字对齐，即LLP[1:0]固定为0	R/W											
b1-b0	Reserved	-	读出时为“0”，写入时写“0”	R/W											

### 15.5.24 通道控制寄存器(DMA\_CHCTLx) (x=0~7)

复位值: 0x00001000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
H PROT[3:2]	-	IE	LLPR UN	LLPE N	HSIZE[1:0]	DNS EQEN	SNSE QEN	DRP TEN	SRPT EN	DINC[1:0]	SINC[1:0]				

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b14	H PROT[3:2]	AHB总线中的 H PROT值	HPORT[3:2]位控制AHB总线的H PROT[3:2]。 当使用DMA访问SMC时,请将H PROT[3:2]设为2'b11; 其他情况请将H PROT[3:2]设为2'b00。	R/W
b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	IE	中断使能位	配置该通道是否产生中断。 0: 该通道不产生中断 1: 该通道产生中断	R/W
b11	LLPRUN	连锁传输模式选择	连锁传输有效时,设置当前传输完成时,载入链指针指向的新描述符后是否立即开始新描述符对应的传输 0: 不立即传输,等待下一个传输请求产生后开始传输 1: 新描述符载入后,立即开始传输	R/W
b10	LLPEN	连锁传输使能	0: 连锁传输无效 1: 连锁传输有效	R/W
b9~b8	HSIZE[1:0]	传输数据的宽度	00: 8bit 01: 16bit 10, 11: 32bit	R/W
b7	DNSEQEN	目标地址不连续传 输使能	0: 不允许不连续地址传输 1: 允许不连续地址传输	R/W
b6	SNSEQEN	源地址不连续传 输使能	0: 不允许不连续地址传输 1: 允许不连续地址传输	R/W
b5	DRPTEN	目标重复传输功 能使能位	设置是否允许目标地址重新载入初始值 0: 不重载 1: 重载	R/W
b4	SRPTEN	源重复传输功能使 能位	设置是否允许源地址重新载入初始值 0: 不重载 1: 重载	R/W
b3~b2	DINC[1:0]	目标地址的更新方 式	00: 固定 01: 递增 10, 11: 递减	R/W
b1~b0	SINC[1:0]	源地址的更新方式	00: 固定 01: 递增 10, 11: 递减	R/W

### 15.5.25 通道监视寄存器(DMA\_MONSAR<sub>x</sub>, DMA\_MONDAR<sub>x</sub>, DMA\_MONDTCTL<sub>x</sub>, DMA\_MONRPT<sub>x</sub>, DMA\_MONSNSEQCTL<sub>x</sub>, DMA\_MONDNSEQCTL<sub>x</sub>) (<sub>x</sub>=0~7)

这些监视寄存器与相应的通道配置寄存器对应，寄存器位配置一致，但全部为只读寄存器。

通道配置寄存器在 DMA 传输前后保持不变，而通道监视寄存器会在 DMA 每完成一个请求对应的传输后，即每完成一个数据块传输后更新。更新内容和方式如下：

- DMA\_MONSAR<sub>x</sub>.SAR[31:0], DMA\_MONDAR<sub>x</sub>.DAR[31:0]: 按通道配置寄存器设置的固定/递增/递减/重载/不连续跳转方式更新为下一次传输的地址。
- DMA\_MONDTCTL<sub>x</sub>.CNT[15:0]: 减 1，若已经是 0 则保持为 0。
- DMA\_MONRPT<sub>x</sub>.SRPT[9:0], DRPT[9:0]: 通道重置无效时，减去块大小，减到 0 时重载 DMA\_RPT<sub>x</sub> 设定值。重置有效时，减 1，减到 0 时重载原始设定值。
- DMA\_MONSNSEQCTL<sub>x</sub>.SNSCNT[11:0], DMA\_MONDNSEQCTL<sub>x</sub>.DNSCNT[11:0]: 通道重置无效时，减去块大小，减到 0 时重载 DMA\_SNSEQCTL<sub>x</sub>/DMA\_DNSEQCTL<sub>x</sub> 原始设定值。重置有效时，减 1，减到 0 时重载原始设定值。
- 以上之外的监视寄存器位保持与配置寄存器一致。

## 15.6 使用注意事项

- DMA 的寄存器只支持 32bit 读写，8/16bit 读写操作无效。
- DMA 传输时发生总线错误，且有其它通道正处于等待状态时，DMA 会进入锁死状态，无法响应后续的所有通道传输请求。进入此种状态后，无法通过配置 DMA 自身寄存器来解除锁死状态。需要执行软件复位，或者通知外围电路对系统进行复位。

检测 DMA 锁死的方法：查询错误标志位 DMA\_INTSTAT0.TRNERR[5:0]（可以放在 DMA 错误中断处理中），若标志位为 0，则表示未发生总线错误，DMA 未锁死。若标志位不为 0，继续查询通道状态位 DMA\_CHSTAT.CHAUT[5:0]，若 CHAUT[5:0]=0x0，表示未锁死。若其中某位，如通道 x 的 CHAUT[x]=1 且长时间保持 1（超过其正常传输所需时间），且即便将对应通道许可位 DMA\_CHEN.CHEN[x]软件清 0，CHAUT[x]仍保持 1，说明 DMA 已锁死。为避免 DMA 进入锁死状态，需要避免让 DMA 访问会发生总线错误的区域，譬如保留地址空间，受保护的地址空间等。若要访问可能有总线错误的区域时，可以禁用本 DMA 单元的其它通道，避免总线错误时有别的通道在等待。

## 16 电压比较器 (CMP)

### 16.1 简介

电压比较器 (Comparator, 简称 CMP) 是比较两个模拟电压大小的外设模块，提供了两组共 4 个比较通道，CMP1 和 CMP2、CMP3 和 CMP4。

#### CMP 具有以下主要特性：

- 所有比较通道均可独立进行电压比较
- 同组的两个比较通道组合可实现窗口比较
- 多路模拟输入 (IO, PGA, DAC, VREFINT)
- 迟滞电压可配置
- 比较结果输出配有噪声滤波器
- 比较结果输出配有空白窗口功能
- 比较结果变化沿可产生中断和外设触发事件
- 比较结果可唤醒系统低功耗模式
- 软件复位或者看门狗复位发生时输出保持
- 比较结果可输出到外部引脚 (CMP\_OUT)
- 比较结果可输出到紧急刹车模块
- 比较结果可输出到定时器模块

## 16.2 功能框图

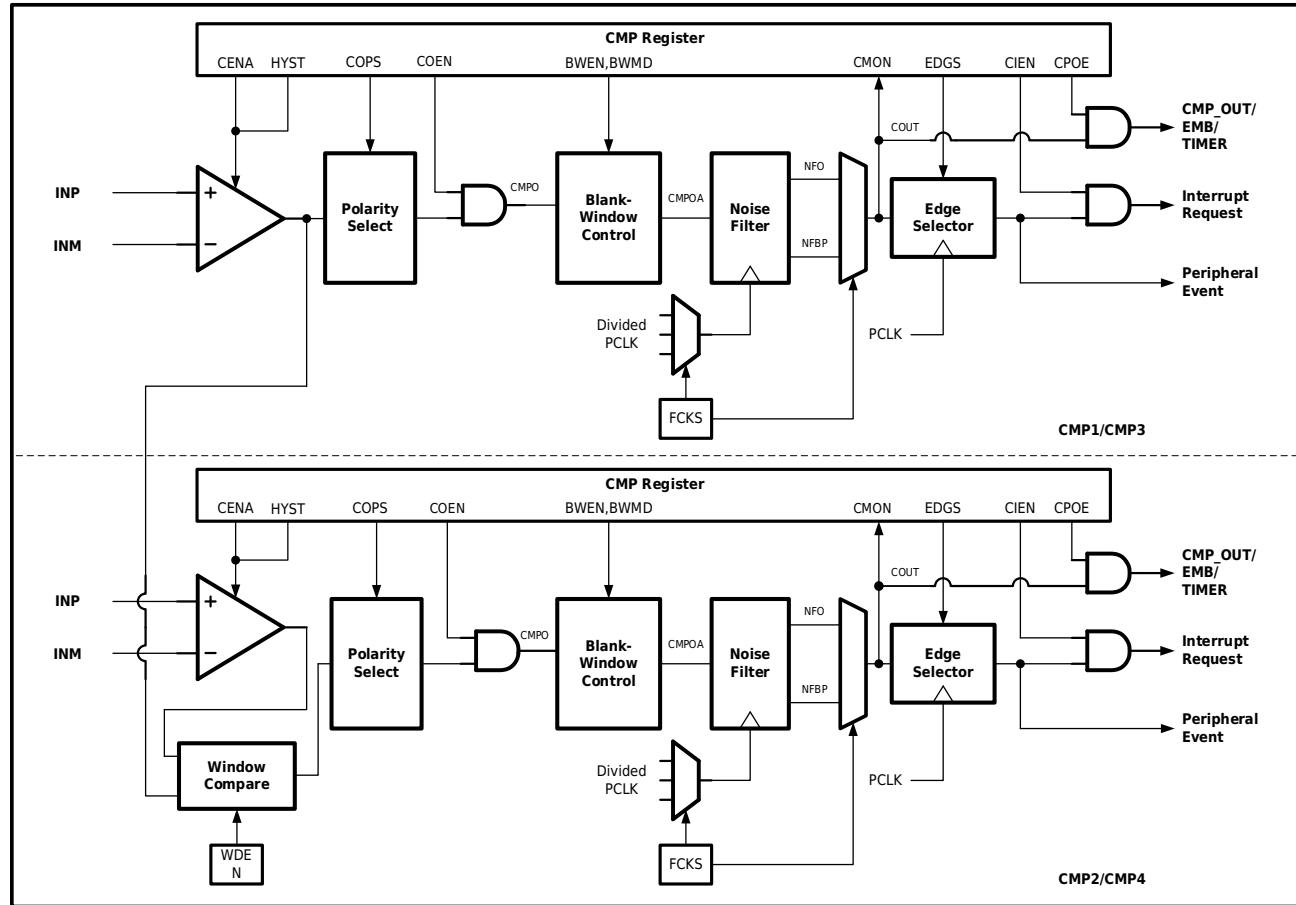


图 16-1 CMP 功能框图

表 16-1 CMP 输入输出引脚

引脚名称	输入/输出	功能
AVCC	I	模拟电源
AVSS	I	模拟地
PA0	I	CMP1_INM1
	O	CMP1_OUT
PA1	I	CMP1_INP1/CMP2_INP1
PA2	I	CMP1_INM2/CMP2_INM1
	O	CMP2_OUT
PA3	I	CMP1_INP2/CMP2_INP2
PA4	I	CMP1_INP3/CMP2_INP3/CMP1_INM3/CMP2_INM2/CMP3_INM2/CMP4_INM2
PA5	I	CMP1_INM4/CMP3_INM3/CMP4_INM3
PA6	I	CMP2_INM3/CMP4_INM4
	O	CMP1_OUT
PA7	I	CMP1_INP4/CMP2_INP4/CMP2_INM4/CMP3_INM4/CMP4_INM2
	O	CMP2_OUT
PA10	O	CMP3_OUT
PA11	O	CMP1_OUT
PA12	O	CMP2_OUT
PA13	O	CMP3_OUT
PA14	O	CMP3_OUT
PB0	I	CMP3_INP1/CMP4_INP1
PB1	O	CMP4_OUT
PB2	I	CMP4_INM1
PB6	O	CMP1_OUT
PB7	O	CMP2_OUT
PB8	O	CMP4_OUT
PB9	O	CMP2_OUT
PB10	O	CMP4_OUT
PB11	I	CMP3_INP2/CMP4_INP2
PB12	O	CMP4_OUT
PB15	I	CMP3_INM1
	O	CMP1_OUT
PC1	O	CMP3_OUT
PC6	O	CMP3_OUT
PD2	O	CMP3_OUT
PF0	O	CMP3_OUT
PF1	O	CMP1_OUT
PF2	O	CMP4_OUT

引脚名称	输入/输出	功能
PF3	O	CMP3_OUT
PF6	O	CMP1_OUT
PF7	O	CMP2_OUT

## 16.3 功能说明

### 16.3.1 输入选择

CMP1~4 的模拟输入按照下表分别进行配置。

表 16-2 CMP 模拟输入对照表 (HC32F472KCHI/HC32F472KEHI)

CMP1					CMP2				
序号	寄存器 PMSR.CVSL	正端输入 INP	寄存器 PMSR.RVSL	负端输入 INM	序号	寄存器 PMSR.CVSL	正端输入 INP	寄存器 PMSR.RVSL	负端输入 INM
1	0001b	PA1	0000000001b	VREFINT*1/4	1	0001b	PA1	0000000001b	VREFINT*1/4
2	0010b	PA3	0000000010b	VREFINT*1/2	2	0010b	PA3	0000000010b	VREFINT*1/2
3	0100b	PA4	0000000100b	VREFINT*3/4	3	0100b	PA4	0000000100b	VREFINT*3/4
4	1000b	PA7	0000001000b	VREFINT <sup>注1</sup>	4	1000b	PA7	0000001000b	VREFINT <sup>注1</sup>
5	-	-	0000010000b	DA3O2 <sup>注2</sup>	5	-	-	0000010000b	DA1O2 <sup>注2</sup>
6	-	-	0000100000b	DA3O1 <sup>注2</sup>	6	-	-	0000100000b	DA3O1 <sup>注2</sup>
7	-	-	0001000000b	PA0	7	-	-	0001000000b	PA2
8	-	-	0010000000b	PA2	8	-	-	0010000000b	PA4 <sup>注3</sup> (DAC1_OUT1)
9	-	-	0100000000b	PA4 <sup>注3</sup> (DAC1_OUT1)	9	-	-	0100000000b	PA6 <sup>注3</sup> (DAC2_OUT1)
10	-	-	1000000000b	PA5 <sup>注3</sup> (DAC1_OUT2)	10	-	-	1000000000b	PA7 <sup>注3</sup> (DAC2_OUT2)
CMP3					CMP4				
序号	寄存器 PMSR.CVSL	正端输入 INP	寄存器 PMSR.RVSL	负端输入 INM	序号	寄存器 PMSR.CVSL	正端输入 INP	寄存器 PMSR.RVSL	负端输入 INM
1	0001b	PB0	0000000001b	VREFINT*1/4	1	0001b	PB0	0000000001b	VREFINT*1/4
2	0010b	PB11	0000000010b	VREFINT*1/2	2	0010b	PB11	0000000010b	VREFINT*1/2
3	0100b	PGA1	0000000100b	VREFINT*3/4	3	0100b	PGA1	0000000100b	VREFINT*3/4
4	1000b	PGA2	0000001000b	VREFINT <sup>注1</sup>	4	1000b	PGA2	0000001000b	VREFINT <sup>注1</sup>
5	-	-	0000010000b	DA2O1 <sup>注2</sup>	5	-	-	0000010000b	DA2O2 <sup>注2</sup>
6	-	-	0000100000b	DA3O1 <sup>注2</sup>	6	-	-	0000100000b	DA3O1 <sup>注2</sup>
7	-	-	0001000000b	PB15	7	-	-	0001000000b	PB2
8	-	-	0010000000b	PA4 <sup>注3</sup> (DAC1_OUT1)	8	-	-	0010000000b	PA4 <sup>注3</sup> (DAC1_OUT1)
9			0100000000b	PA5 <sup>注3</sup> (DAC1_OUT2)	9			0100000000b	PA5 <sup>注3</sup> (DAC1_OUT2)
10			1000000000b	PA7 <sup>注3</sup> (DAC2_OUT2)	10			1000000000b	PA6 <sup>注3</sup> (DAC2_OUT1)

表 16-3 CMP 模拟输入对照表 (HC32F472JCTI/HC32F472JETI/HC32F472PCTI/HC32F472PETI)

CMP1					CMP2				
序号	寄存器 PMSR.CVSL	正端输入 INP	寄存器 PMSR.RVSL	负端输入 INM	序号	寄存器 PMSR.CVSL	正端输入 INP	寄存器 PMSR.RVSL	负端输入 INM
1	0001b	PA1	0000000001b	VREFINT*1/4	1	0001b	PA1	0000000001b	VREFINT*1/4
2	0010b	PA3	0000000010b	VREFINT*1/2	2	0010b	PA3	0000000010b	VREFINT*1/2
3	-	-	0000000100b	VREFINT*3/4	3	-	-	0000000100b	VREFINT*3/4
4	-	-	0000001000b	VREFINT <sup>注1</sup>	4	-	-	0000001000b	VREFINT <sup>注1</sup>
5	-	-	0000010000b	DA1O1 <sup>注2</sup>	5	-	-	0000010000b	DA1O1 <sup>注2</sup>
6	-	-	0000100000b	DA1O2 <sup>注2</sup>	6	-	-	0000100000b	DA1O2 <sup>注2</sup>
7	-	-	0001000000b	PA0	7	-	-	0001000000b	PA2
8	-	-	0010000000b	PA2	8	-	-	0010000000b	-
9	-	-	0100000000b	-	9	-	-	0100000000b	-
10	-	-	1000000000b	-	10	-	-	1000000000b	-

**注 1:** VREFINT 为内部参考电压，电压值为 1.1V。

**注 2:** 选择 DAC 内部通路时将 DAC 通道的模拟输出关闭可以释放对应的模拟 IO。

**注 3:** 选择 DAC 外部通路时必须打开 DAC 通道的模拟输出。DAC 模拟输出关闭时，对应模拟 IO 可以用于模拟输入。

### 16.3.2 电压比较

每个比较通道可以独立进行电压比较。以 CMP1 为例，设定步骤如下：

1. 关闭 CMP1 的模块停止功能 (PWR\_FCG3.bit8 写 0)。
2. 设定 CMP1\_PMSR 寄存器的 CVSL 位，选择正端电压；  
设定 CMP1\_PMSR 寄存器的 RVSL 位，选择负端电压。
3. 设定 CMP1\_MDR 寄存器的 HYST 位，选择迟滞电压。  
设定 CMP1\_MDR 寄存器的 CENA 位为 1，启动比较器。
4. 等待比较器的稳定时间 tCMP。(具体数值请参考产品数据手册)
5. 设定 CMP1\_FIR 寄存器的 FCKS 位，设置滤波采样周期数；  
设定 CMP1\_FIR 寄存器的 EDGS、CIEN 位，设置中断。
6. 设定 CMP1\_OCR 寄存器的 COPS、CPOE 位，设置外部输出。
7. 设定 CMP1\_OCR 寄存器的 COEN 位为 1，允许比较器输出。

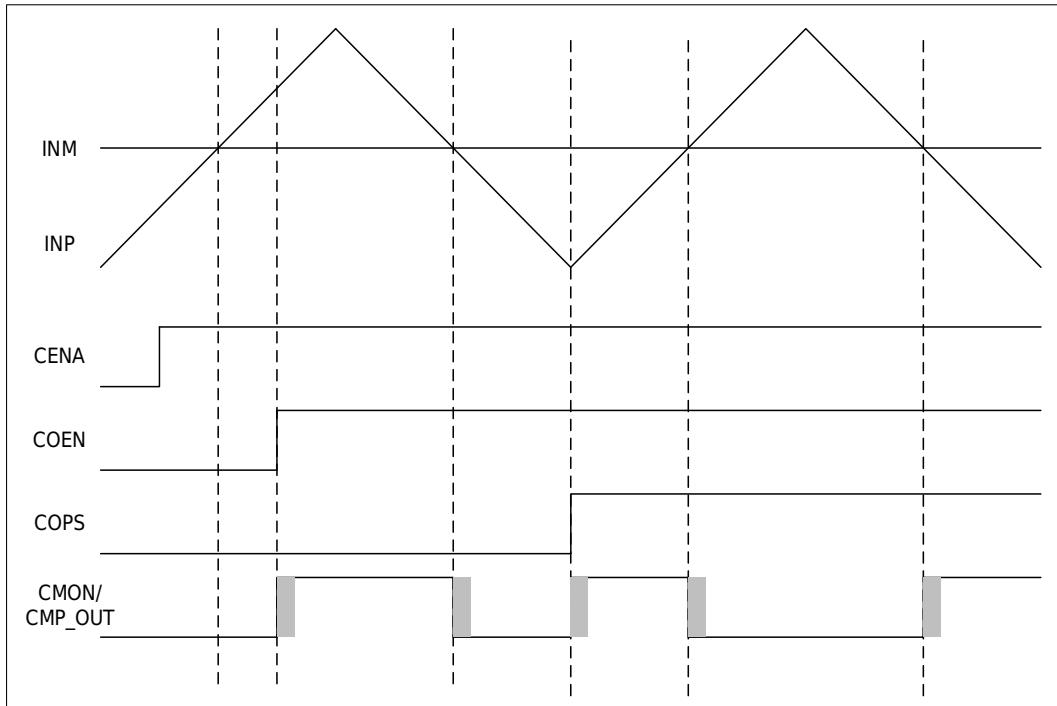


图 16-2 电压比较示意图

如图 16-2，在正相输出时，CMP 允许输出后如果正端电压高于负端电压，`CMPx_OMR` 寄存器的 CMON 位为 1；如果正端电压低于负端电压，则 CMON 位为 0。在反相输出时 CMON 的结果与正相输出时相反。如果使用数字滤波器，由于同步和滤波采样，CMON 的变化沿会至少晚 6 个 CMPCLK 周期（图中灰色部分所示）。

### 16.3.3 窗口比较

同组的两个比较通道组合使用可实现窗口比较，并分别由 CMP2、CMP4 完成对比较结果的监视、滤波、中断和输出等操作。

以 CMP1 和 CMP2 组合为例，设定步骤如下：

1. 关闭 CMP1, 2 的模块停止功能（`PWR_FCG3.bit8` 写 0）。
2. 设定 `CMP1_PMSR` 寄存器的 CVSL 位，选择 CMP1 的正端电压；  
设定 `CMP1_PMSR` 寄存器的 RVSL 位，选择 CMP1 的负端电压作为窗口下限。
3. 设定 `CMP2_PMSR` 寄存器的 CVSL 位，选择 CMP2 的正端电压；  
(注意 CMP1 和 CMP2 的正端输入必须一致)  
设定 `CMP2_PMSR` 寄存器的 RVSL 位，选择 CMP2 的负端电压作为窗口上限。
4. 设定 `CMP2_MDR` 寄存器的 WDEN 位为 1，选择窗口比较模式。
5. 设定 `CMP1_MDR` 和 `CMP2_MDR` 寄存器的 HYST 位，选择迟滞电压；  
(注意迟滞电压带来的窗口变化)  
设定 `CMP1_MDR` 和 `CMP2_MDR` 寄存器的 CENA 位为 1，启动比较器。
6. 等待稳定时间 tCMP。（具体数值请参考产品数据手册）

7. 设定 CMP2\_FIR 寄存器的 FCKS 位，设置滤波采样周期数；
8. 设定 CMP2\_FIR 寄存器的 EDGS、CIEN 位，设置中断。
9. 设定 CMP2\_OCR 寄存器的 COPS、CPOE 位，设置外部输出；
10. 设定 CMP1\_OCR 和 CMP2\_OCR 寄存器的 COEN 位为 1，允许比较器输出。

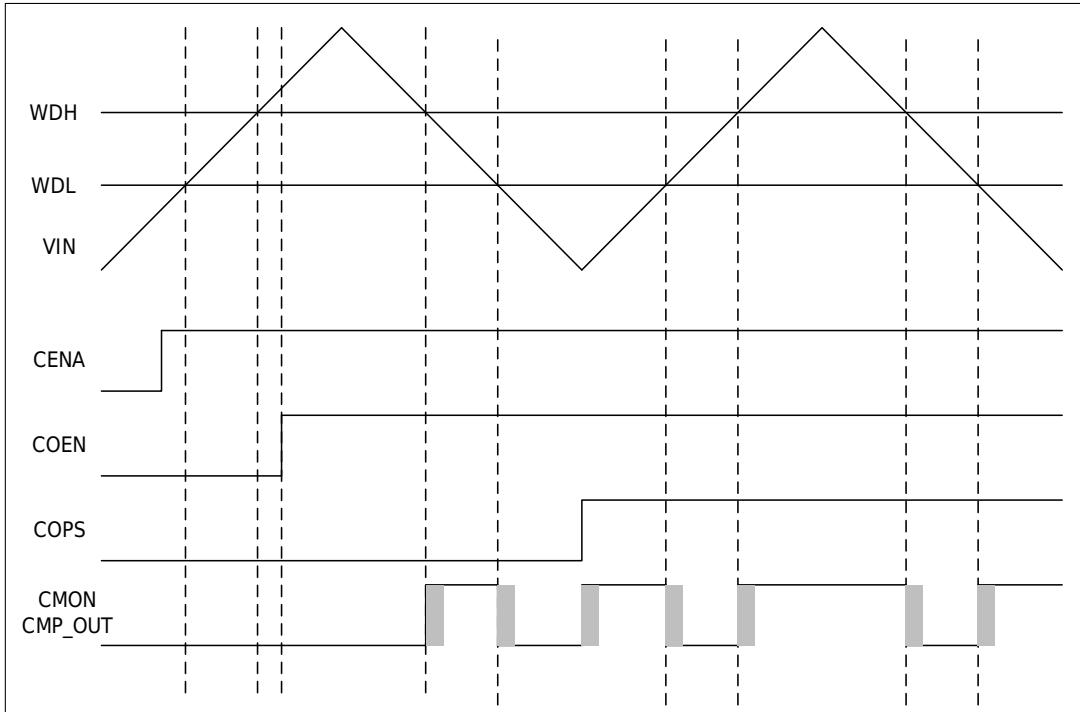


图 16-3 窗口比较示意图

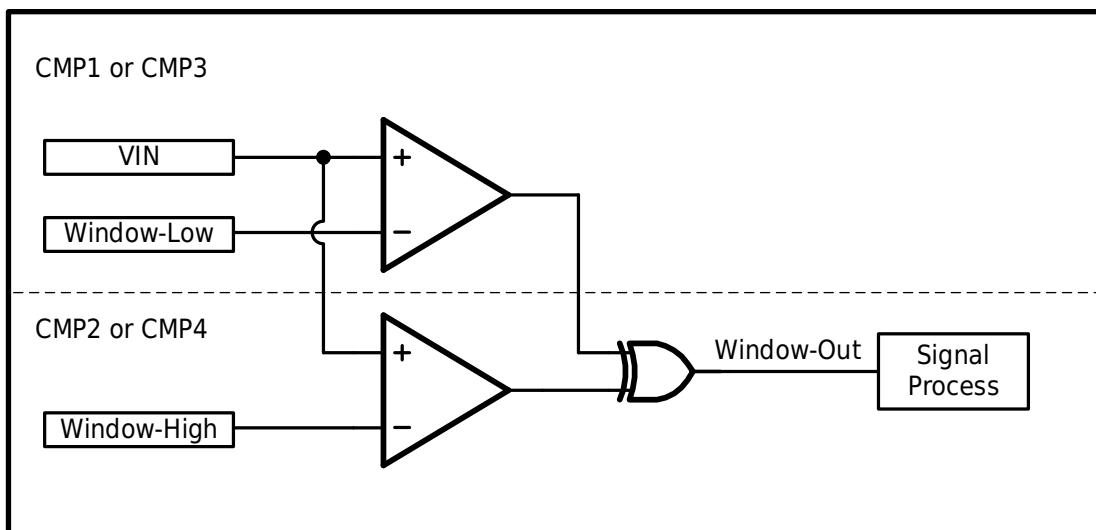


图 16-4 窗口比较原理图

如图 16-3，在正相输出时，CMP 允许输出后如果正端电压 VIN 处于两个负端电压（WDH，WDL）组成的窗口内，CMP2\_OMR 寄存器的 CMON 位为 1；如果 VIN 处于窗口外，则 CMON 位为 0。在反相输出时 CMON 的结果与正相输出时相反。如果使用数字滤波器，由于同步和滤波采样，CMON 的变化沿会相对晚至少 6 个 CMPCLK 周期（图中灰色部分所示）。

### 16.3.4 空白窗口

可以将定时器输出的 PWM 信号作为空白窗口(Blank-Window,又称定时器窗口)信号对比较结果 CMPO 进行输出控制，用于定时器 PWM 过零检测等应用场景。

窗口输出有电平和边沿两种模式。

#### ■ 电平模式

启动窗口输出 (CMPx\_OCR 寄存器的 BWEN 位为 1) 并且选择电平模式 (CMPx\_OCR 寄存器的 BWMD 位为 0) 时，通过 CMPx\_BWSR1 寄存器的 CTWS 位选择一个或多个定时器 PWM 作为 Blank-Window 信号，并通过 CTWP 位为各个 Blank-Window 信号选择有效电平。这样窗口输出 CMPOA 在 Blank-Window 有效时会被固定成由 CMPx\_OCR 寄存器的 BWOL 位设定的状态。

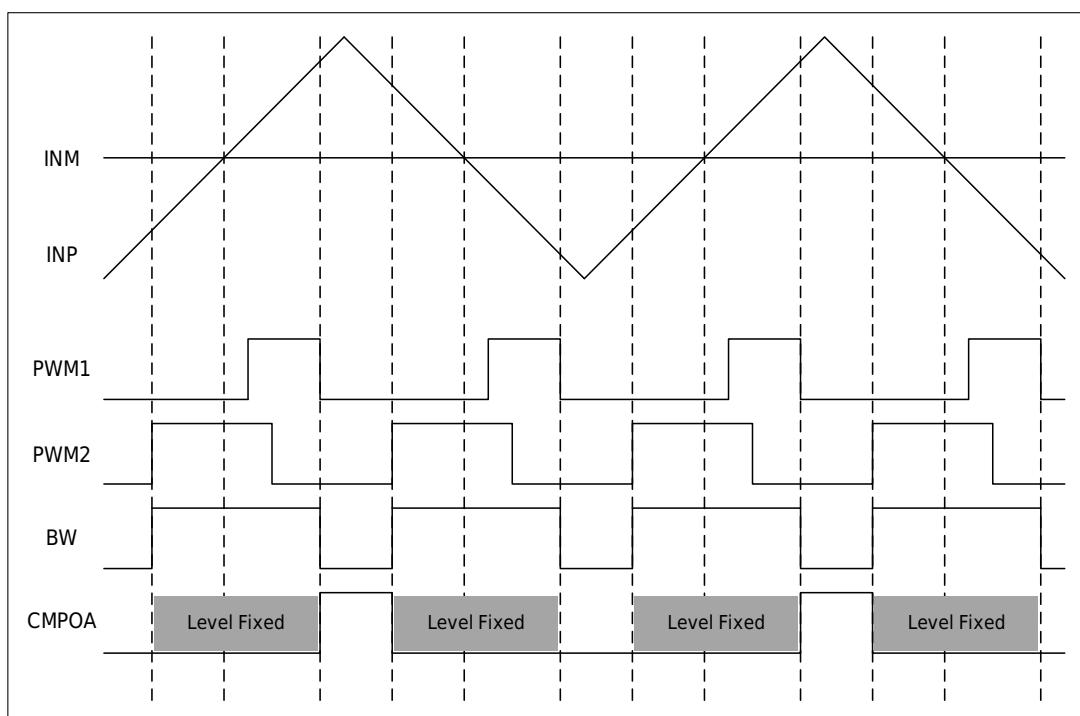


图 16-5 窗口输出示意图一

如图 16-5,选择 PWM1,2 作为 Blank-Window 并且都是高电平有效,同时 Blank-Window 有效时 CMPOA 输出状态为低电平。于是在 PWM1,2 有一个为高电平时 CMPOA 输出就被固定成低电平,而在 PWM1,2 都为低电平时 CMPOA 则正常输出比较结果。

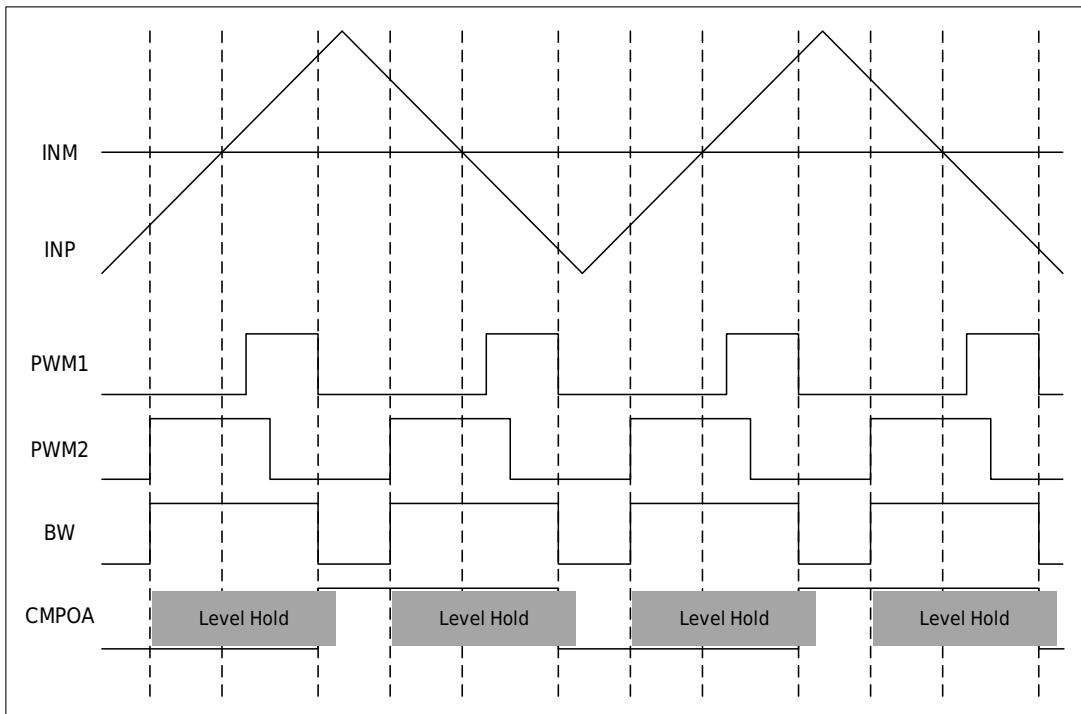


图 16-6 窗口输出示意图二

如图 16-6,选择 PWM1,2 作为 Blank-Window 并且都是高电平有效,同时 Blank-Window 有效时 CMPOA 输出状态保持。于是 PWM1,2 有一个为高电平时 CMPOA 就保持前面的状态;而在 PWM1,2 都为低电平时 CMPOA 则正常输出比较结果,但会晚 1~3 个 CMPCLK 周期,这是由于同步和采样造成的。

#### ■ 边沿模式

启动窗口输出 (CMPx\_OCR 寄存器的 BWEN 位为 1) 并且选择边沿模式 (CMP\_OCR 寄存器的 BWMD 位为 1) 时,通过 CMPx\_BWSR1 寄存器的 CTWS 位选择一个或多个定时器 PWM 作为触发源产生 Blank-Window 信号,通过 CMPx\_BWSR2 寄存器的 TWEG 位选择触发源的有效边沿。这样,窗口输出 CMPOA 在触发源的有效边沿处会被固定成由 CMPx\_OCR 寄存器的 BWOL 位设定的状态,并维持由 CMPx\_BWSR2 寄存器的 MSKW 位设定的 N 个 CMPCLK 周期。

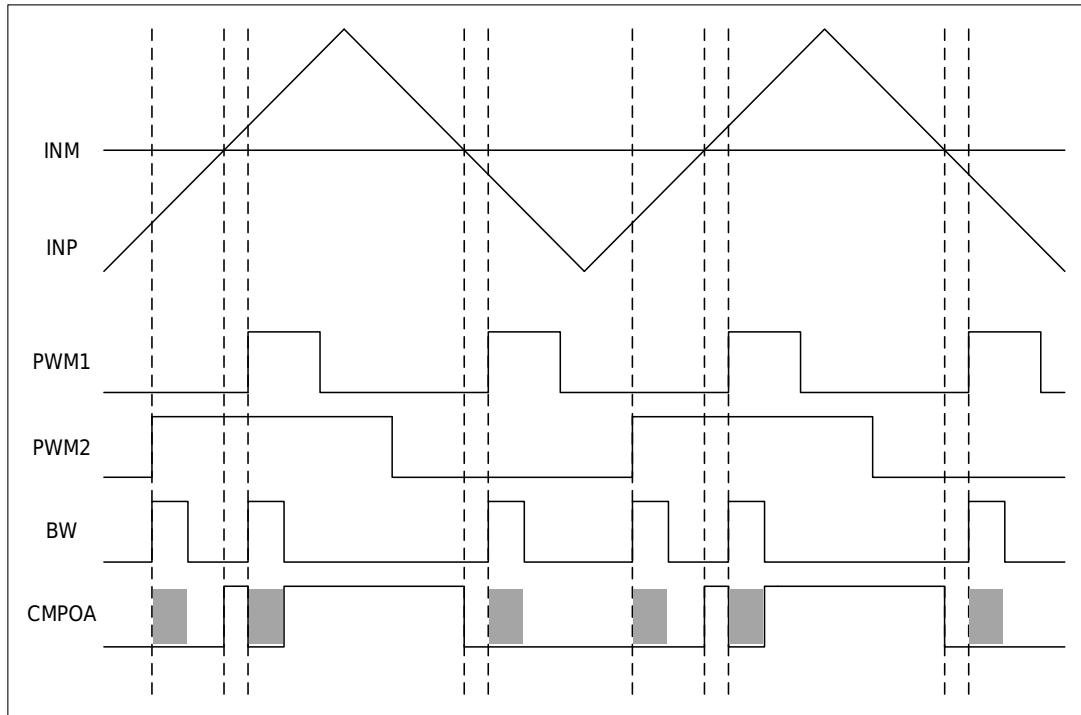


图 16-7 窗口输出示意图三

如图 16-7，选择 PWM1,2 作为触发源并且为上升沿有效，同时 Blank-Window 有效时 CMPOA 输出状态为低电平。于是，在 PWM1,2 的每个上升沿都会产生宽度可设的 Blank-Window 信号。Blank-Window 有效时 CMPOA 输出被固定成了低电平，Blank-Window 无效时 CMPOA 则正常输出比较结果。

### 16.3.5 迟滞功能

为了避免噪声信号所引起的假输出，每个比较通道都带有可配置的迟滞功能并通过 CMPx\_MDR 的 HYST 位选择迟滞电压。HYST 位设为 000b 时迟滞功能关闭。

### 16.3.6 噪声滤波

每个比较通道各包含一个噪声滤波器对比较器输出 CMPOA 进行滤波。噪声滤波器会在每个采样时钟周期对 CMPOA 进行采样，如果连续采样到三次相同的值则在下一个采样时钟周期将该值输出到 COUT，否则 COUT 保持不变。采样时钟可以通过 CMPx\_FIR 寄存器的 FCKS 位选择，当 FCKS 位设为 00b 时噪声滤波器关闭。

### 16.3.7 中断事件

每个比较通道可由输出 COUT 产生一个中断请求。使用时必须将 CMPx\_FIR 寄存器的 CIEN 位设置成 1，并通过 EDGS 位（2'b00 以外的值）设定 COUT 产生中断的边沿。由于 COUT 是原始比较结果经过极性选择、空白窗口、噪声滤波后输出的，使用时请注意实际发生中断的边沿。

COUT 还可以唤醒停止低功耗模式。在进入停止模式前必须设定中断有效（CIEN=1）并且关闭数字滤波和边沿检测，系统将在 COUT 由低变高时被唤醒并产生一次中断。系统回到工作模式后，请先设定中断无效（CIEN=0）后再进行后续操作。

外设触发事件和中断请求的产生条件相同，但必须通过目标外设的硬件触发源寄存器启动或关闭事件触发功能，与 CIEN 位无关。

### 16.3.8 外部输出

可以将 COUT 输出到 CMP 外部，如 CMPx\_OUT 引脚以及 EMB、TIMER 等外设。CMPx\_OCR 寄存器的 CPOE 位用来使能外部输出。由于 CMPx\_OUT 对应的引脚默认均为通用端口（GPIO），使用时还需要将其设定为 CMPx\_OUT 功能。

## 16.4 注意事项

### 16.4.1 模块停止功能

CMP 有模块停止功能，通过设定模块停止寄存器可将模块的数字部分关闭。

CMP1 和 CMP2 共用一个模块停止寄存器位 PWR\_FCG3.bit8。

CMP3 和 CMP4 共用一个模块停止寄存器位 PWR\_FCG3.bit9。

所有比较通道初始均为停止状态，设定模块工作时才可以访问各自的寄存器。相关寄存器设定请参考低功耗章节。

### 16.4.2 模块停止时的功耗

CMP 在工作状态进入到模块停止状态时，比较器将继续保持工作，功耗等同于工作状态。如需进一步降低功耗，请将 CMPx\_MDR 寄存器的 CENA 位清“0”。

### 16.4.3 停止低功耗模式时的功耗

芯片进入停止低功耗模式时，如果 CMP 处于工作状态，进入停止低功耗模式后将继续保持工作，功耗等同于进入停止低功耗模式之前的水平。如需进一步降低功耗，请在进入停止低功耗模式前将 CMPx\_MDR 寄存器的 CENA 位清“0”。

## 16.5 寄存器说明

CMP1 基准地址：0x40038400

CMP2 基准地址：0x40038500

CMP3 基准地址：0x4003B000

CMP4 基准地址：0x4003B100

表 16-4 CMP 寄存器一览表

寄存器名	符号	偏移地址	位宽	复位值
比较器工作模式寄存器	CMPx_MDR	0x00	8	0x00
比较器滤波中断寄存器	CMPx_FIR	0x01	8	0x00
比较器输出控制寄存器	CMPx_OCR	0x02	8	0x00
比较器结果监视寄存器	CMPx_OMR	0x03	8	0x00
比较器输入选择寄存器	CMPx_PMSR	0x04	32	0x00000000
比较器输出空白窗口设定寄存器1	CMPx_BWSR1	0x10	32	0x00000000
比较器输出空白窗口设定寄存器2	CMPx_BWSR2	0x14	16	0x0000

表中 x=1~4。

所有寄存器在软件复位和看门狗复位发生时将保持。

### 16.5.1 比较器工作模式寄存器 (CMPx\_MDR, x=1~4)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
Reserved		WDEN	HYST[2:0]		CENA		
位	标记	位名	功能				读写
b7~b5	Reserved	-	读出时为“0”，写入时写“0”				R/W
b4	WDEN	窗口比较使能	WDEN只存在于CMP2/4中。设为窗口比较后，比较结果的滤波、边沿检测以及输出等控制均由CMP2/4完成。 0：独立比较 1：窗口比较				R/W
b3~b1	HYST[2:0]	迟滞电压选择	以下列出的均为典型值。 000: 0mv (无迟滞) 001: 10mv 010: 20mv 011: 30mv 100: 40mv 101: 50mv 110: 60mv 111: 70mv				R/W
b0	CENA	比较器工作许可	0：电压比较器停止 1：电压比较器工作 每次将CENA由“0”设为“1”后，请等待tCMP（参考产品数据手册），确保CMP稳定后再进行后续操作。				R/W

## 16.5.2 比较器滤波中断寄存器 (CMPx\_FIR, x=1~4)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
CRF	CFF	EDGS[1:0]		CIEN		FCKS[2:0]	
<hr/>							
位	标记	位名	功能			读写	
b7	CRF	比较器输出上升沿标志	0: 未检测到比较器输出COUT的上升沿 1: 检测到比较器输出COUT的上升沿 CRF置位后需通过写1清零。			R/W	
b6	CFF	比较器输出下降沿标志	0: 未检测到比较器输出COUT的下降沿 1: 检测到比较器输出COUT的下降沿 CFF置位后需通过写1清零。			R/W	
b5~b4	EDGS[1:0]	比较器输出边沿检测选择	0 0: 不检测比较器输出COUT的边沿 0 1: 检测比较器输出COUT的上升沿 1 0: 检测比较器输出COUT的下降沿 1 1: 检测比较器输出COUT的上升沿和下降沿 改变EDGS[1:0]可能会引起中断或者外设触发事件，因此请在中断禁止或者外设触发功能无效状态下设定该寄存器。寄存器设定后，请清除相应的中断标志。			R/W	
b3	CIEN	比较器中断许可	0: 禁止比较器中断 1: 允许比较器中断			R/W	
b2~b0	FCKS[2:0]	比较器结果滤波采样选择	0 0 0: 不使用噪声滤波器 0 0 1: 使用噪声滤波器，通过PCLK0进行采样 0 1 0: 使用噪声滤波器，通过(PCLK0)/2进行采样 0 1 1: 使用噪声滤波器，通过(PCLK0)/4进行采样 1 0 0: 使用噪声滤波器，通过(PCLK0)/8进行采样 1 0 1: 使用噪声滤波器，通过(PCLK0)/16进行采样 1 1 0: 使用噪声滤波器，通过(PCLK0)/32进行采样 1 1 1: 使用噪声滤波器，通过(PCLK0)/64进行采样 请在比较器输出禁止（即COEN位为“0”）时改写FCKS[2:0]。 FCKS[2:0]从“00b”切换到其他值时，请使用经过4次采样后的滤波输出作为中断请求或外设触发事件。 改变FCKS[2:0]可能会引起中断或者外设触发事件，因此请在中断禁止或者外设触发功能无效状态下设定该寄存器。寄存器设定后，请清除相应的中断标志。			R/W	

### 16.5.3 比较器输出控制寄存器 (CMPx\_OCR, x=1~4)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
	BWOL[1:0]	BWMD	BWEN	Reserved	CPOE	COPS	COEN

位	标记	位名	功能	读写
b7~b6	BWOL[1:0]	Blank-Window输出电平	0: Blank-Window有效（输出关闭）期间比较器输出低电平 1: Blank-Window有效（输出关闭）期间比较器输出高电平 1x: Blank-Window有效（输出关闭）期间比较器输出保持	R/W
b5	BWMD	Blank-Window模式	0: 电平模式 1: 边沿模式	R/W
b4	BWEN	Blank-Window使能	0: 关闭BLANK-WINDOW功能 1: 打开BLANK-WINDOW功能	R/W
b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2	CPOE	比较器外部输出允许	0: 禁止输出到EMB/TIMER/VCOUT引脚 1: 允许输出到EMB/TIMER/VCOUT引脚	R/W
b1	COPS	比较器输出极性选择	0: 比较器正相输出 1: 比较器反相输出  请在比较器输出禁止（即COEN位为“0”）时改写COPS。改变COPS位可能会引起中断或者外设触发事件，因此请在中断禁止或者外设触发功能无效状态下设定该寄存器。寄存器设定后，请清除相应的中断标志。	R/W
b0	COEN	比较器输出允许	0: 禁止比较器输出（比较器输出固定成低电平） 1: 允许比较器输出	R/W

### 16.5.4 比较器结果监视寄存器 (CMPx\_OMR, x=1~4)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
Reserved							CMON

位	标记	位名	功能	读写
b7~b1	Reserved	-	读出时为“0”，写入时写“0”	R/W
b0	CMON	比较结果监视	普通比较模式时 0: 正端电压低于负端电压 1: 正端电压高于负端电压  窗口比较模式时（只对CMP2/4有效） 0: 正端电压在窗口外 1: 正端电压在窗口内	R

### 16.5.5 比较器输入选择寄存器 (CMPx\_PMSR, x=1~4)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved							
b23	b22	b21	b20	b19	b18	b17	b16
Reserved				CVSL[3:0]			
b15	b14	b13	b12	b11	b10	b9	b8
Reserved						RVSL[9:8]	
b7	b6	b5	b4	b3	b2	b1	b0
RVSL[7:0]							

位	标记	位名	功能	读写
b31~b20	Reserved	-	读出时为“0”，写入时写“0”	R/W
b19~b16	CVSL[3:0]	正端输入选择	选择CMP的正端输入电压。 0000: 无正端电压输入 0001: 选择INP1作为正端电压 (所选输入见表 16-2) 0010: 选择INP2作为正端电压 (所选输入见表 16-2) 0100: 选择INP3作为正端电压 (所选输入见表 16-2) 1000: 选择INP4作为正端电压 (所选输入见表 16-2) 其他: 设定禁止 注1	R/W
b15~b10	Reserved	-	读出时为“0”，写入时写“0”	R/W
b9~b0	RVSL[9:0]	负端输入选择	选择CMP的负端输入电压。 0000000000: 无负端电压输入 0000000001: 选择INM1作为负端电压 (所选输入见表 16-2) 0000000010: 选择INM2作为负端电压 (所选输入见表 16-2) 0000000100: 选择INM3作为负端电压 (所选输入见表 16-2) 0000001000: 选择INM4作为负端电压 (所选输入见表 16-2) 0000010000: 选择INM5作为负端电压 (所选输入见表 16-2) 0000100000: 选择INM6作为负端电压 (所选输入见表 16-2) 0001000000: 选择INM7作为负端电压 (所选输入见表 16-2) 0010000000: 选择INM8作为负端电压 (所选输入见表 16-2) 0100000000: 选择INM9作为负端电压 (所选输入见表 16-2) 1000000000: 选择INM10作为负端电压 (所选输入见表 16-2) 其他: 设定禁止 注1	R/W

**注 1:** 请在比较器输出禁止 (即 COEN 位为“0”) 时改写。每次改写后请等待 300ns 再允许比较器输出 (即 COEN 位设为“1”)。改写可能会引起中断或者外设触发事件，因此请在中断禁止或者外设触发功能无效状态下设定该寄存器。寄存器设定后，请清除相应的中断标志。

**16.5.6 比较器输出空白窗口设定寄存器 1 (CMPx\_BWSR1, x=1~4)**

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
CTWP15	CTWP14	CTWP13	CTWP12	CTWP11	CTWP10	CTWP9	CTWP8
b23	b22	b21	b20	b19	b18	b17	b16
CTWP7	CTWP6	CTWP5	CTWP4	CTWP3	CTWP2	CTWP1	CTWP0
b15	b14	b13	b12	b11	b10	b9	b8
CTWS15	CTWS14	CTWS14	CTWS12	CTWS11	CTWS10	CTWS9	CTWS8
b7	b6	b5	b4	b3	b2	b1	b0
CTWS7	CTWS6	CTWS5	CTWS4	CTWS3	CTWS2	CTWS1	CTWS0

位	标记	位名	功能	读写
b31~b16	CTWP15~0	窗口信号极性	Blank-Window为电平模式 (CMP_OCR.BWMD=0) 时设定用于关闭比较器输出的窗口信号的有效电平。(仅对应的CTWS为1时该位的设定有效) 0: 窗口信号为高电平时比较器输出关闭 1: 窗口信号为低电平时比较器输出关闭	R/W
b15~b0	CTWS15~0	窗口信号使能	Blank-Window有效 (CMP_OCR.BWEN=1) 时选择用于关闭比较器输出的窗口信号。(各bit对应的窗口信号见表 16-5) 0: 窗口信号无效 1: 窗口信号有效	R/W

表 16-5 定时器窗口 PWM 一览表

CTWS bit	CMP1	CMP2	CMP3	CMP4
15	TIM4_1_OWL	TIM4_1_OWL	TIM4_1_OWL	TIM4_1_OWL
14	TIM4_1_OWH	TIM4_1_OWH	TIM4_1_OWH	TIM4_1_OWH
13	TIM4_1_OVL	TIM4_1_OVL	TIM4_1_OVL	TIM4_1_OVL
12	TIM4_1_OVH	TIM4_1_OVH	TIM4_1_OVH	TIM4_1_OVH
11	TIM4_1_OUL	TIM4_1_OUL	TIM4_1_OUL	TIM4_1_OUL
10	TIM4_1_OUH	TIM4_1_OUH	TIM4_1_OUH	TIM4_1_OUH
9	TIM6_7_PWMA	TIM6_8_PWMA	TIM6_7_PWMB	TIM6_8_PWMB
8	TIM6_5_PWMA* <sup>1</sup>	TIM6_6_PWMA* <sup>1</sup>	TIM6_5_PWMB* <sup>1</sup>	TIM6_6_PWMB* <sup>1</sup>
7	TIM6_3_PWMA* <sup>1</sup>	TIM6_4_PWMA* <sup>1</sup>	TIM6_3_PWMB* <sup>1</sup>	TIM6_4_PWMB* <sup>1</sup>
6	TIM6_1_PWMA* <sup>1</sup>	TIM6_2_PWMA* <sup>1</sup>	TIM6_1_PWMB* <sup>1</sup>	TIM6_2_PWMB* <sup>1</sup>
5	TIMA_2_PWM3	TIMA_4_PWM3	TIMA_3_PWM3	TIMA_4_PWM3
4	TIMA_2_PWM2	TIMA_4_PWM2	TIMA_3_PWM2	TIMA_4_PWM2
3	TIMA_2_PWM1	TIMA_4_PWM1	TIMA_3_PWM1	TIMA_4_PWM1
2	TIMA_1_PWM3	TIMA_3_PWM3	TIMA_1_PWM3	TIMA_2_PWM3
1	TIMA_1_PWM2	TIMA_3_PWM2	TIMA_1_PWM2	TIMA_2_PWM2
0	TIMA_1_PWM1	TIMA_3_PWM1	TIMA_1_PWM1	TIMA_2_PWM1

注 1：可选高精度 PWM 模式。

### 16.5.7 比较器输出空白窗口设定寄存器 2 (CMPx\_BWSR2, x=1~4)

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	-	-	TWEG[1:0]	
b7	b6	b5	b4	b3	b2	b1	b0
MSKW[7:0]							

位	标记	位名	功能	读写
b15~b10	Reserved	-	读出时为“0”，写入时写“0”	R/W
b9~b8	TWEG[1:0]	窗口信号边沿	Blank-Window为边沿模式 (CMP_OCR.BWMD=1) 时设定用于关闭比较器输出的窗口信号的有效边沿。 0 0：无有效边沿 0 1：窗口信号的上升沿处比较器输出关闭 1 0：窗口信号的下降沿处比较器输出关闭 1 1：窗口信号的变化沿处比较器输出关闭	
b7~b0	MSKW	BLANK-WINDOW有效宽度	Blank-Window为边沿模式 (CMP_OCR.BWMD=1) 时设定关闭比较器输出的宽度。 可设0~255之间的任意值。 当检测到窗口信号发生TWEG位设定的边沿时，比较器输出将关闭并持续MSKW位设定的CMPCLK周期数。	R/W

## 17 模数转换模块 (ADC)

### 17.1 简介

12 位 ADC 是一种采用逐次逼近方式的模拟数字转换器。本 MCU 搭载 3 个 ADC 单元，单元 1 和 2 支持 24 个通道，单元 3 支持 25 个通道，可以转换来自外部引脚、以及芯片内部的模拟信号。模拟输入通道可以任意组合成一个序列，一个序列可以进行单次扫描转换，或连续扫描转换。支持对任意指定通道进行连续多次转换并对转换结果进行平均。ADC 模块还搭载模拟看门狗功能，对任意指定通道的转换结果进行监视，检测其是否超出用户设定的范围。

#### ADC 主要特性

##### ■ 高性能

- 可配置 12 位、10 位和 8 位分辨率
- ADC 数字接口时钟 PCLK4 和转换时钟 PCLK2 (也称作 ADCLK) 的频率比可设置为 1:1、2:1、4:1、8:1、1:2、1:4  
PCLK2 可选与系统时钟 HCLK 异步的 PLL 时钟，此时频率比  
 $PCLK4:PCLK2=1:1$   
PCLK2 频率最高支持 60MHz
- 采样率：2.5MSPS (PCLK2=60MHz, 12 位，采样 11 周期，变换 13 周期)
- 各通道采样时间独立编程
- 各通道独立数据寄存器
- 数据寄存器可配置左/右对齐方式
- 连续多次转换平均功能
- 模拟看门狗，监视转换结果
- 不使用时可以将 ADC 模块设定成停止状态

##### ■ 模拟输入通道

- 总计 32 个外部模拟输入，单个 ADC 单元最大支持 25 通道
- 3 个内部模拟输入：内部基准电压、CMP 参考电压、AVCC 分压

##### ■ 转换开始条件

- 软件设置转换开始
- 周边外设同步触发转换开始
- 外部引脚触发转换开始

##### ■ 转换模式

- 2 个扫描序列 A、B，可任意指定单个或多个通道
- 序列 A 单次扫描
- 序列 A 连续扫描

- 双序列扫描，序列 A、B 独立选择触发源，序列 B 优先级高于 A
- 协同工作模式（适用于具有两个或三个 ADC 的设备）
- 中断与事件信号输出
  - 序列 A 扫描结束中断和事件 ADC\_EOCA
  - 序列 B 扫描结束中断和事件 ADC\_EOCB
  - 模拟看门狗 0 比较中断和事件 ADC\_CMP0
  - 模拟看门狗 1 比较中断和事件 ADC\_CMP1
  - 上述的 4 个事件输出都可启动 DMA，仅搭载了 DMA 的产品支持

## 17.2 ADC 系统框图

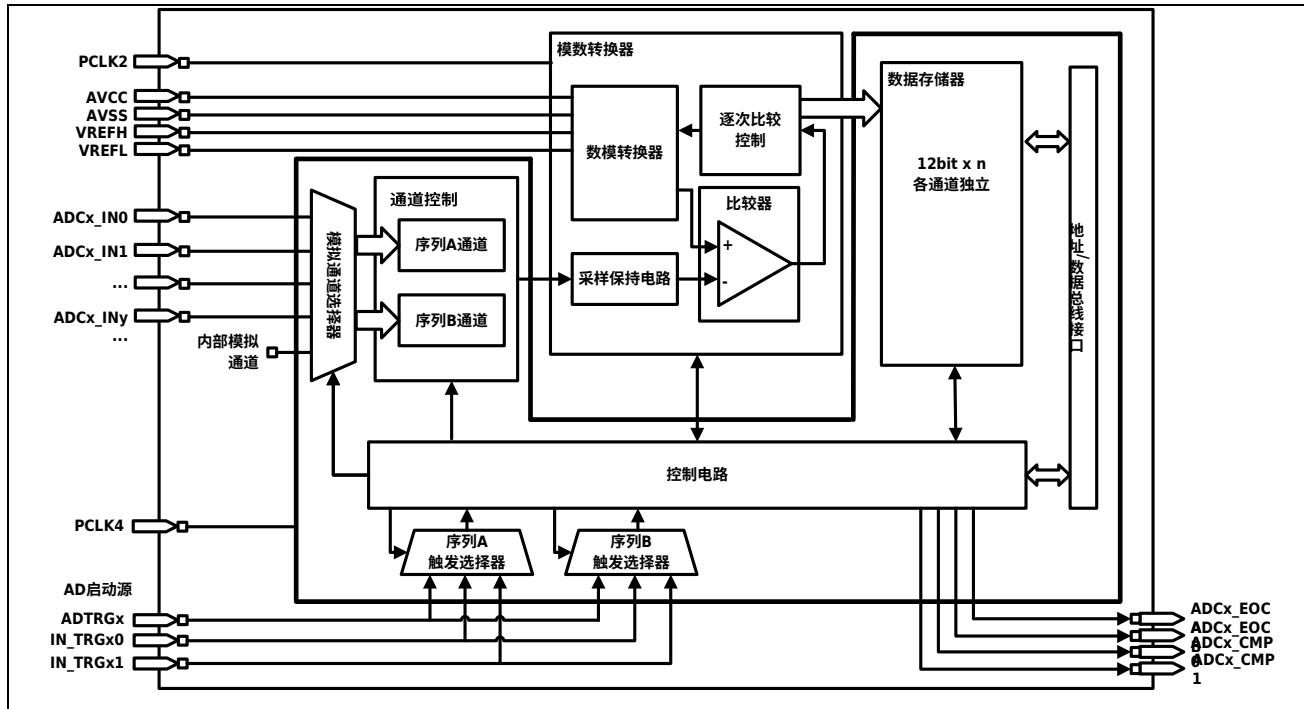


图 17-1 ADC 框图

本芯片搭载了 3 个 ADC 模块单元，每个单元的配置有所不同，具体参考下表：

表 17-1 各 ADC 单元规格 (HC32F472KCHI/HC32F472KEHI)

项目	单元1 (ADC1)	单元2 (ADC2)	单元3 (ADC3)
电源	AVCC		
	AVSS		
基准电压	VREFL		
	VREFH		
模拟通道 *1	CH0	ADC123_IN0	ADC123_IN0
	CH1	ADC123_IN1	ADC123_IN1
	CH2	ADC123_IN2	ADC123_IN2
	CH3	ADC123_IN3	ADC123_IN3
	CH4	ADC12_IN4	ADC3_IN4
	CH5	ADC12_IN5	ADC3_IN5
	CH6	ADC12_IN6	ADC3_IN6
	CH7	ADC12_IN7	ADC3_IN7
	CH8	ADC123_IN8	ADC123_IN8
	CH9	ADC123_IN9	ADC123_IN9
	CH10	ADC123_IN10	ADC123_IN10
	CH11	ADC123_IN11	ADC123_IN11
	CH12	ADC123_IN12	ADC123_IN12
	CH13	ADC123_IN13	ADC123_IN13
	CH14	ADC12_IN14	ADC3_IN14
	CH15	ADC12_IN15	ADC3_IN15
	CH16	ADC123_IN16	ADC123_IN16
	CH17	ADC12_IN17	ADC3_IN17
	CH18	ADC12_IN18	ADC3_IN18
	CH19	ADC12_IN19	ADC3_IN19
	CH20	ADC12_IN20	ADC3_IN20
	CH21	内部基准电压 ADVREF	ADC3_IN21
	CH22	内部VREF模块输出电压 VREFINT	内部基准电压 ADVREF
	CH23	AVCC分压AVCCDIVMON	内部VREF模块输出电压 VREFINT
	CH24	-	AVCC分压 AVCCDIVMON
可编程增益放大器PGA	PGA	ADC12_IN7	ADC123_IN8
硬件触发源	外部引脚	ADTRG1	ADTRG2
	片内周边	IN_TRG10	IN_TRG20
		IN_TRG11	IN_TRG21
			IN_TRG31

表 17-2 各 ADC 单元规格 (HC32F472JCTI/HC32F472JETI/HC32F472PCTI/HC32F472PETI)

项目	单元1 (ADC1)	单元2 (ADC2)	单元3 (ADC3)
电源	AVCC		
	AVSS		
基准电压	VREFL		
	VREFH		
模拟通道 *1	CH0	ADC123_IN0	ADC123_IN0
	CH1	ADC123_IN1	ADC123_IN1
	CH2	ADC123_IN2	ADC123_IN2
	CH3	ADC123_IN3	ADC123_IN3
	CH4	ADC12_IN4	ADC3_IN4
	CH5	ADC12_IN5	ADC3_IN5
	CH6	ADC12_IN6	ADC3_IN6
	CH7	ADC12_IN7	ADC3_IN7
	CH8	ADC12_IN8	ADC3_IN8
	CH9	ADC12_IN9	-
	CH10	ADC123_IN10	ADC123_IN10
	CH11	ADC123_IN11	ADC123_IN11
	CH12	ADC123_IN12	ADC123_IN12
	CH13	ADC123_IN13	ADC123_IN13
	CH14	ADC12_IN14	-
	CH15	ADC12_IN15	-
	CH16	-	ADC3_IN16
	CH17	-	ADC3_IN17
	CH18	-	ADC3_IN18
	CH19	-	ADC3_IN19
	CH20	ADC12_IN20	ADC3_IN20
	CH21	内部基准电压 ADVREF	ADC3_IN21
	CH22	内部VREF模块输出电压 VREFINT	内部基准电压 ADVREF
	CH23	AVCC分压AVCDIVMON	内部VREF模块输出电压 VREFINT
	CH24	-	AVCC分压 AVCDIVMON
可编程增益 放大器PGA	PGA	ADC12_IN7	ADC123_IN8
硬件触发源	外部引脚	ADTRG1	ADTRG2
	片内周边	IN_TRG10	IN_TRG20
		IN_TRG11	IN_TRG21
			IN_TRG30
			IN_TRG31

**注意：**ADC 中虚拟通道 CH0~CH15 与物理通道 ADCx\_INy (y=0~15) (实际的模拟输入源) 可以设置寄存器进行自由映射，本表所示为复位后默认的映射关系。

## 17.3 功能说明

### 17.3.1 ADC 时钟

ADC 模块需要使用 2 个时钟：数字接口时钟 PCLK4，模拟电路时钟 PCLK2。PCLK4 与 PCLK2 是同步关系，频率比率可设置为 1: 1, 2: 1, 4: 1, 8: 1, 1: 2, 1: 4。

PCLK2 可以选择与系统时钟 HCLK 异步的 PLL 时钟源，此时 PCLK4 与 PCLK2 相同，为同步同频率关系。

**注意：**

- PCLK2 的频率请设置在 1MHz~60MHz 以内。

### 17.3.2 通道选择

ADC 模块支持通道映射，即模块中的虚拟通道与实际的物理通道间的映射。虚拟通道是指 ADC 模块中假定的通道，如寄存器 ADC\_CHSELRA 设置为 0x1 表示序列 A 选择转换 CH0，这个 CH0 就是虚拟通道，而寄存器 ADC\_DR0，是虚拟通道 CH0 的转换结果寄存器。物理通道是指实际存在的模拟通道，即外部引脚的模拟输入 ADCx\_INy 以。虚拟通道与物理通道的映射可以通过寄存器 ADC\_CHMUXR 进行配置，具体参考寄存器说明(只有前 16 个通道支持映射)。本章中若无特别说明，通道 n 或 CHn 均表示虚拟通道。

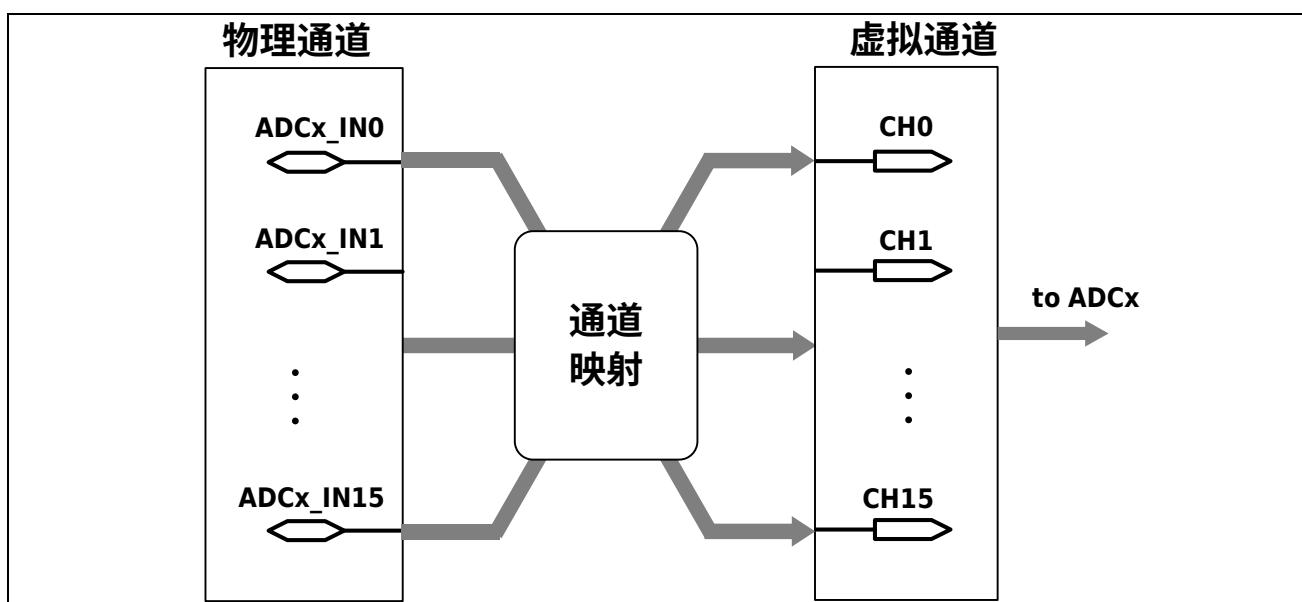


图 17-2 通道映射示意图

ADC 模块有多个通道，可配置为两个序列：序列 A，序列 B 进行转换。序列 A 和 B 配有独立的通道选择寄存器 ADC\_CHSELRA, ADC\_CHSELRB。寄存器每位代表一个通道，如 bit0 位写 1 表示转换 CH0，写 0 表示不转换 CH0。两个序列可独立选择任意 1 个或多个通道进行转换。例如：ADC\_CHSELRA 设

置为 0x0055，ADCHSELRB 设置为 0x0002，则序列 A 的触发条件发生时，将依次转换 CH0, CH2, CH4 和 CH6 这 4 个通道。序列 B 的触发条件发生时，将转换 CH1 这一个通道。

其中，复位后内部模拟通道是处于关闭状态，使用这些通道前请参考电源控制(PWC)高精度参考电压(VREF) 章节，设置相应的寄存器，允许内部模拟电压输出。

另外，大部分的物理通道可以输入到多个 ADC 模块单元，如 ADC123\_IN0 可输入到 ADC\_1,2,3 三个单元，ADC12\_IN4 可输入 ADC\_1,2 两单元。结合多 ADC 协同工作模式以及通道映射功能，可以方便的对指定通道实现高采样率的转换。

#### 注意：

- 不要在序列 A 和 B 中选择相同的通道。对于不存在的通道，请不要设置相应的寄存器，保持其复位后的状态。

### 17.3.3 触发源选择

序列 A, 序列 B 独立选择触发源。可选择的触发源包括外部端口 ADTRGx, 内部事件 IN\_TRGx0, IN\_TRGx1。其中，端口 ADTRGx 下降沿输入有效。IN\_TRGx0, IN\_TRGx1 由寄存器 ADC\_TRGSEL0,1 设置，可以选择芯片内部丰富的事件源。此外，写寄存器 ADC\_STR 可生成序列 A 软件触发信号，序列 A 软件触发只能在 ADC 处于待机状态时使用。软件触发与触发源选择寄存器 ADC\_TRGSR 的设置无关。

### 17.3.4 序列 A 单次扫描模式

ADC 控制寄存器 ADC\_CR0.MS[1:0]设为 00b 选择序列 A 单次扫描模式。

该模式下，当寄存器 ADC\_TRGSR 选择的序列 A 启动条件发生，或者 ADC\_STR.STRT 位写 1 软件触发，ADC 启动，对序列 A 通道选择寄存器 ADC\_CHSELRA 中选择的所有通道依次进行采样和转换，转换结果存入相应的数据寄存器 ADC\_DR 中。ADC 转换过程中 ADC\_STR.STRT 保持为 1，当所有通道转换结束后自动清 0，ADC 进入转换待机状态，等待下次触发条件的发生。

当所有通道转换结束时，序列 A 转换结束标志位 ADC\_ISR.EOCAF 置 1，并产生序列 A 转换结束事件 ADC\_EOCA，可以用此事件启动 DMA。若 ADC\_ICR.EOCAIEN 为 1，中断许可的状态下，同时还产生序列 A 转换结束中断请求。

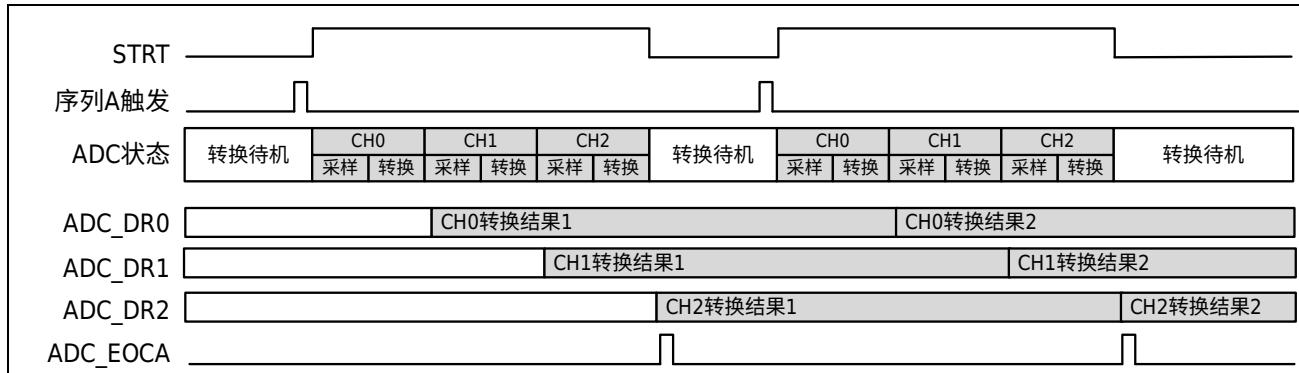


图 17-3 序列 A 单次扫描模式

序列 A 单次扫描模式的软件流程：

- 确认 ADC\_STR.STRT 为 0，ADC 处于转换待机状态。
- ADC 控制寄存器 ADC\_CR0.MS[1:0]设为 00b 选择序列 A 单次扫描模式。
- 设置序列 A 通道选择寄存器 ADC\_CHSELRA。
- 设置采样时间寄存器 ADC\_SSTR。
- ADC\_STR.STRT 写 1 软件触发序列 A，或者设置寄存器 ADC\_TRGSR 选择序列 A 触发条件。
- 查询序列 A 转换结束标志位 EOCAF。
- 读取各通道数据寄存器 ADC\_DR。
- 写 0 清除 EOCAF 标志位，为下次转换准备。

上面的步骤 6~8 的 CPU 查询方式也可以替换成中断方式，利用 ADC\_EOCA 中断来处理转换数据。或者利用 ADC\_EOCA 事件启动 DMA 读取数据。

### 17.3.5 序列 A 连续扫描模式

ADC 控制寄存器 ADC\_CR0.MS[1:0]设为 01b 选择序列 A 连续扫描模式。

序列 A 连续扫描模式与序列 A 单次扫描模式动作类似，不同点在于，连续模式在所用通道转换结束后不是进入转换待机状态，而是重新开始转换序列 A。STRT 位也不会自动清 0。

当需要停止连续扫描时，对 STRT 位写 0，并读 STRT 确认为 0 以判断 ADC 进入转换待机状态。

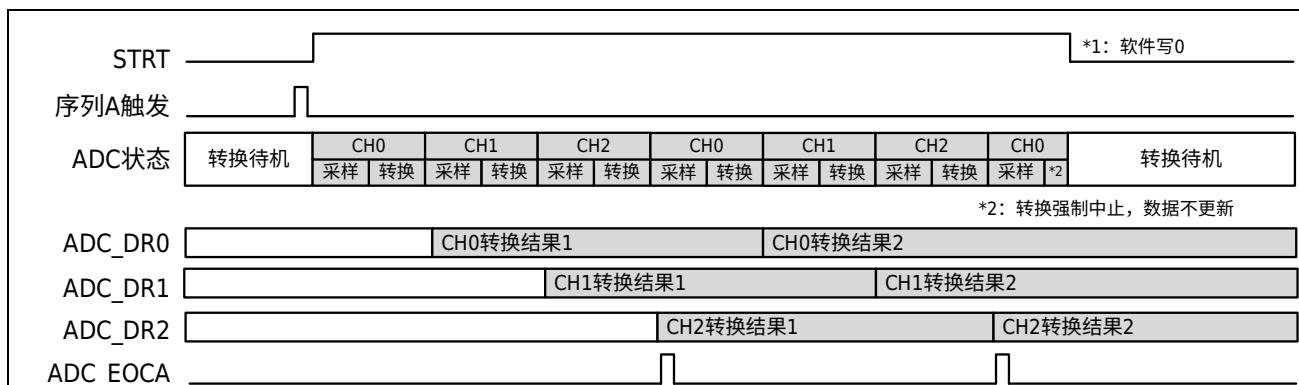


图 17-4 连续扫描

序列 A 连续扫描模式的软件流程：

1. 确认 ADC\_STR.STRT 为 0，ADC 处于转换待机状态。
2. ADC 控制寄存器 ADC\_CR0.MS[1:0]设为 01b 选择序列 A 连续扫描模式。
3. 设置序列 A 通道选择寄存器 ADC\_CHSELRA。
4. 设置采样时间寄存器 ADC\_SSTR。
5. ADC\_STR.STRT 写 1 软件触发序列 A，或者设置寄存器 ADC\_TRGSR 选择序列 A 触发条件。
6. 查询序列 A 转换结束标志位 EOCAF。
7. 读取各通道数据寄存器 ADC\_DR。
8. 写 0 清除 EOCAF 标志位，为下次转换准备。
9. 不需要继续转换时，对 START 位写 0，并读 START 确认为 0 以判断 ADC 进入转换待机状态。

上面的步骤 6~8 的查询方式也可以替换成中断方式，利用 ADC\_EOCA 中断来处理转换数据。或者利用 ADC\_EOCA 事件启动 DMA 读取数据。

**注意：**

- 由于是连续转换，每次扫描的间隔比较短，特别是只选择 1 个通道转换时。推荐使用 ADC\_EOCA 事件启动 DMA 读取数据，避免查询方式下处理不及时导致数据丢失。

### 17.3.6 双序列扫描模式

ADC 控制寄存器 ADC\_CR0.MS[1:0]设为 10b 或 11b 选择双序列扫描模式，即序列 A 与序列 B 都可由各自选择的触发条件来启动扫描。

当 MS[1:0]=10b 时，序列 A 和 B 等效于两个独立的单次扫描序列。MS[1:0]=11b 序列 A 为连续扫描模式，B 为单次扫描模式。

序列 A 由 ADC\_TRGSR.TRGSELA[2:0]选择触发源，由 ADC\_CHSELRA 选择转换的通道。序列 B 由 ADC\_TRGSR.TRGSELB[2:0]选择触发源，由 ADC\_CHSELRB 选择转换的通道。

当序列 A 全部通道转换结束时，序列 A 转换结束标志位 ADC\_ISR.EOCAF 置 1，并产生序列 A 转换结束事件 ADC\_EOCA，若 ADC\_ISCR.EOCAIEN 为 1，中断许可的状态下，同时产生序列 A 转换结束中断请求。当序列 B 全部通道转换结束时，序列 B 转换结束标志位 ADC\_ISR.EOCBF 置 1，并产生序列 B 转换结束事件 ADC\_EOCB，若 ADC\_ISCR.EOCBIEN 为 1，中断许可的状态下，同时产生序列 B 转换结束中断请求。

双序列扫描模式下，当序列 A 与序列 B 发生竞争时，序列 B 将会被优先处理，即序列 B 优先级高于序列 A。具体情况请参看下表。

表 17-3 序列 A 和 B 的各种竞争

ADC转换	触发信号发生	处理方式	
		ADC_CR1.RSCHSEL=0	ADC_CR1.RSCHSEL=1
序列A转换过程中	序列A触发	触发信号无效	
	序列B触发	1) 序列A的转换被中断, 开始序列B转换 2) 序列B的转换完成后, 序列A从被中断的通道开始继续转换	1) 序列A的转换被中断, 开始序列B转换 2) 序列B的转换完成后, 序列A从第一个通道开始重新转换
序列B转换过程中	序列A触发	序列B全部通道转换完成后, 开始序列A转换	
	序列B触发	触发信号无效	
AD空闲中, 序列A, B同时触发		序列B先启动, 全部通道转换完成后, 开始序列A转换	

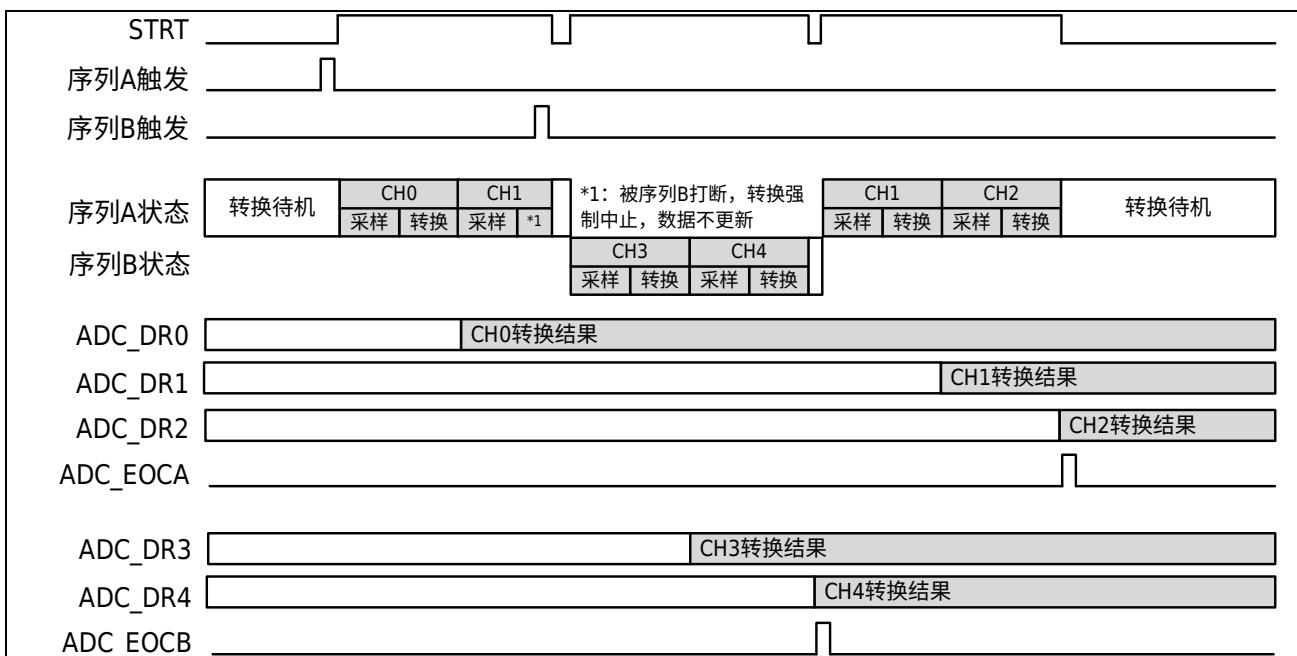


图 17-5 双序列扫描模式（序列 A 从被中断通道重新启动）

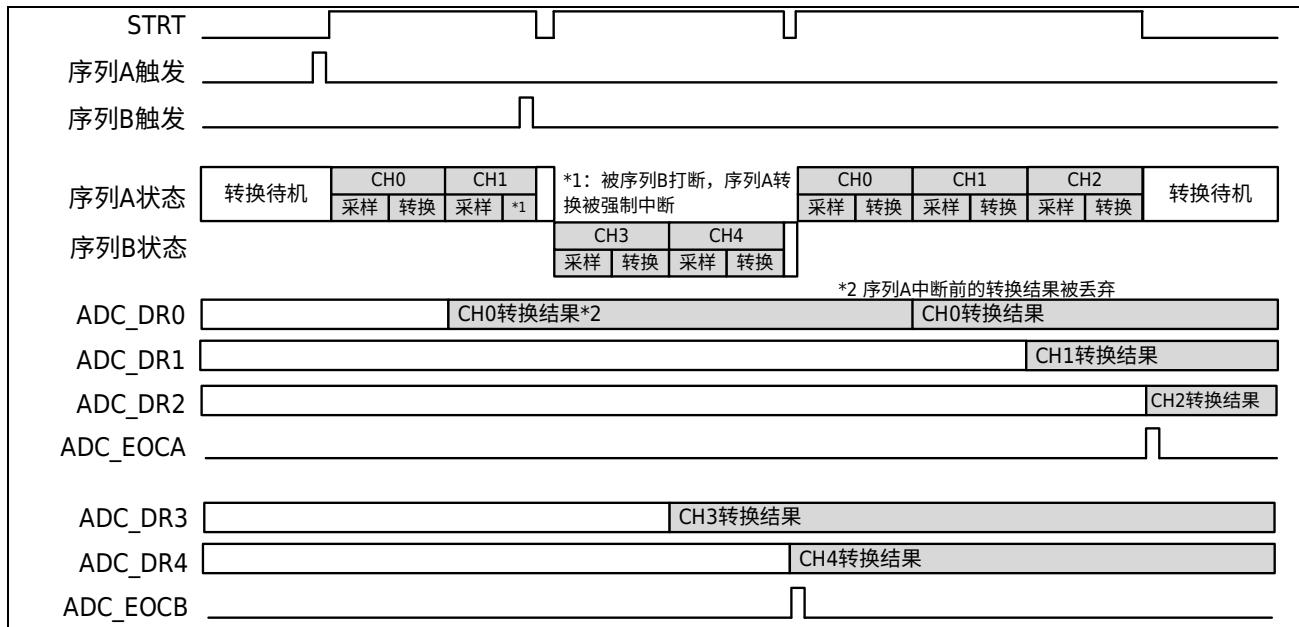


图 17-6 双序列扫描模式（序列 A 从第一个通道重新启动）

双序列扫描模式的软件流程：

1. 确认 ADC\_STR.STRT 为 0，ADC 处于转换待机状态。
2. ADC 控制寄存器 ADC\_CR0.MS[1:0]设为 10b 或 11b 选择双序列扫描模式。
3. 设置寄存器 ADC\_CR1.RSCHSEL 选择序列 A 被打断后启动方式。
4. 设置序列 A 通道选择寄存器 ADC\_CHSELRA。
5. 设置序列 B 通道选择寄存器 ADC\_CHSELRB。
6. 设置采样时间寄存器 ADC\_SSTR。
7. 设置寄存器 ADC\_TRGSR 选择序列 A 和 B 触发条件。
8. 通过查询 EOCAF, EOCBF, 或者 ADC\_EOCA, ADC\_EOCB 中断, 或者启动 DMA 在序列 A 或 B 转换结束后处理转换数据。

**注意：**

- 不要在序列 A 和 B 中选择相同的通道。序列 A 和 B 不要选择相同的触发源。

### 17.3.7 模拟看门狗功能

模拟看门狗功能是指在通道的 ADC 转换结束时对转换结果进行比较，本 ADC 支持 2 个比较窗口：比较窗口 0、比较窗口 1。以比较窗口 0 为例，如下图所示，若转换结果在保护区域内，则生成看门狗比较中断和事件 ADC\_CMP0。

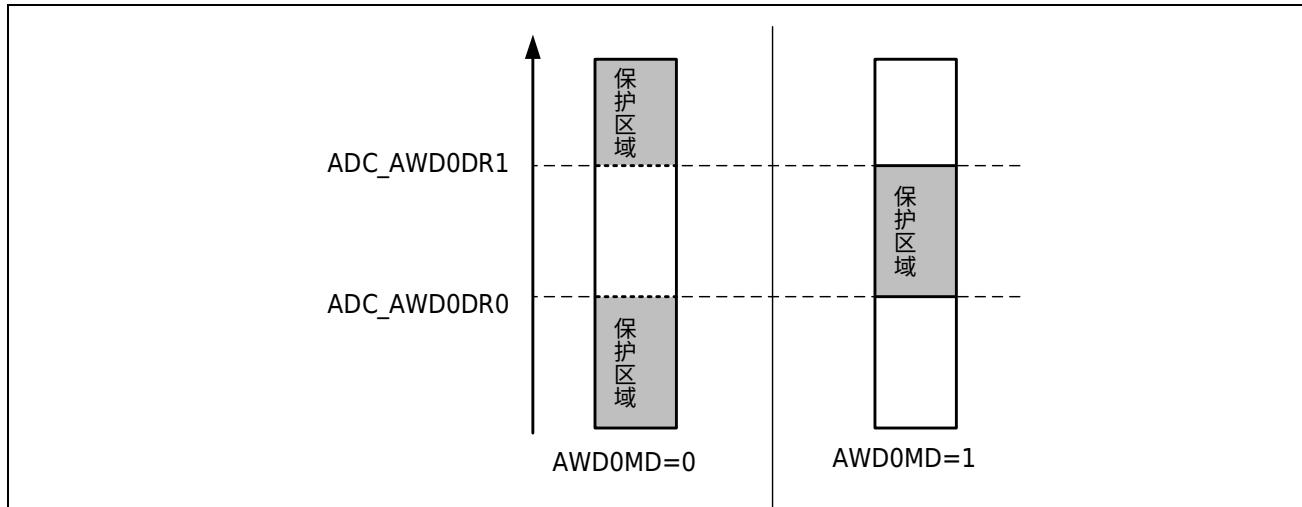


图 17-7 模拟看门狗保护区域（比较模式）

使用模拟看门狗功能的软件流程，以比较窗口 0 为例：

1. 设置阈值寄存器 AD\_AWD0DR0, ADC\_AWD0DR1；
2. 设置比较通道寄存器 ADC\_AWD0CHSR，选择需要比较的通道；
3. 设置 ADC\_AWDCR.AWD0MD 选择比较模式；
4. 设置 ADC\_AWDCR.AWD0IEN 中断许可位；
5. 设置 ADC\_AWDCR.AWD0EN 允许模拟看门狗 0 功能；
6. 根据前文，设置扫描模式，启动 AD 进行转换；
7. 在 ADC\_CMP0 中断中，或 ADC 转换结束后，查询比较状态寄存器 ADC\_AWDSR.AWD0F，对比较结果进行做相应的处理。

比较窗口 1 的使用方法与比较窗口 0 相同。

两个比较窗口可以组合使用。当窗口组合功能有效时，在窗口 1 选的通道转换结束时，比较中断 ADC\_CMP1 输出的不再是窗口 1 单独的比较结果，而是按照设置，对窗口 0 和 1 的比较结果进行逻辑或、逻辑与或者逻辑异或后产生的组合结果。使用窗口组合比较功能的软件流程与窗口单独使用时相似，需要在设置好窗口 0 和 1 之后，追加设置 ADC\_AWDCR.AWDCM[1:0]寄存器选择组合方式。

### 17.3.8 模拟输入的采样时间和转换时间

在单次扫描模式中，ADC 转换可选择软件设置，内部触发 IN\_TRGx0,1 和外部引脚触发 ADTRGx 启动方式。在扫描转换延迟时间  $t_D$  后，ADC 模块才开始对模拟通道进行采样和转换，全部转换结束后经过

转换结束延迟时间  $t_{ED}$  后进入待机状态，一次扫描才最终完成。连续扫描模式与单次扫描相似，只是在序列的第二次以及之后的启动时没有  $t_D$  时间。

单个通道的转换时间  $t_{CONV}=t_{SPL}+t_{CMP}$ 。其中  $t_{SPL}$  表示模拟输入的采样时间，可以根据输入阻抗设置寄存器 ADC\_SSTR 调整采样周期数。 $t_{CMP}$  表示逐次比较时间，12 位精度 13 个 PCLK2，10 位精度 11 个 PCLK2，8 位精度 9 个 PCLK2。

一次扫描转换的时间  $t_{SCAN}=t_D+\sum t_{CONV}+t_{ED}$ 。其中  $\sum t_{CONV}$  表示所有扫描通道的转换时间总和，由于可以独立设置采样时间  $t_{SPL}$ ，各通道的转换时间  $t_{CONV}$  可以不同。

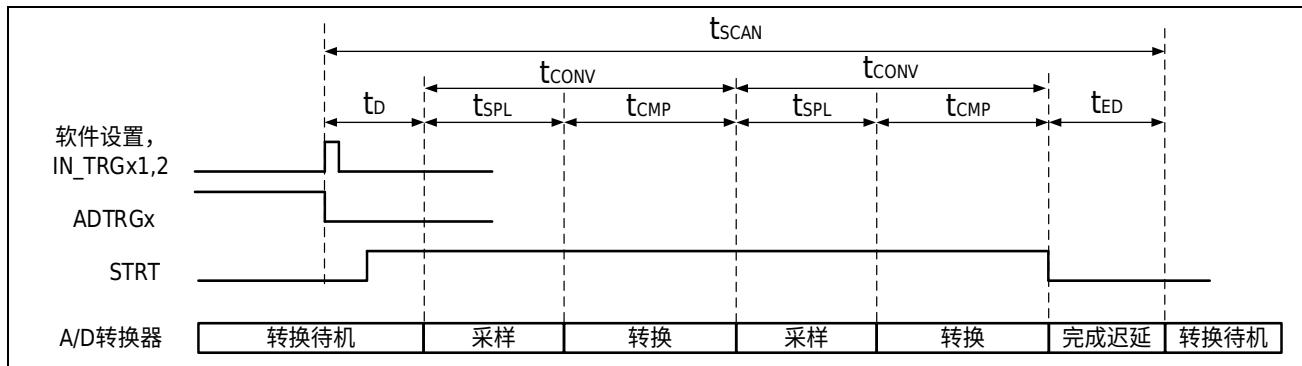


图 17-8 ADC 转换时间

表 17-4 AD 转换时间

标记	说明	条件			
		同步周边触发	异步周边触发 *注	外部引脚触发	软件触发
$t_D$	扫描开始	ADC空闲中,启动转换	4 PCLK2	3 PCLK4 + 4 PCLK2 + 1 PCLK4_SYNC	2 PCLK4 + 4 PCLK2
	处理时间	序列A转换中被打断,启动序列B转换	5 PCLK2	4 PCLK4 + 5 PCLK2 + 1 PCLK4_SYNC	2 PCLK4 + 5 PCLK2
$t_{CONV}$	$t_{SPL}$	采样时间	ADSSTRx.SST[7:0] x PCLK2		
	$t_{CMP}$	12位分辨率	13 PCLK2		
		10位分辨率	11 PCLK2		
		8位分辨率	9 PCLK2		
$t_{ED}$	扫描完成处理时间		3 PCLK2		
$t_{TD}$	最小连续触发时间间隔		$\sum t_{CONV} + 6 \text{ PCLK2}$		

#### 注意：

- 异步周边触发是指 ADC 模块选择与系统时钟异步的 PLL 时钟动作时周边触发的情况。此时周边模块时钟与 ADC 模块时钟为异步关系。PCLK4\_SYNC 表示 ADC 模块原来的同步时钟(即 CMU\_SCFG 设置的时钟)，此时的 PCLK4, PCLK2 相同，均为异步的 PLL 时钟。

### 17.3.9 ADC 数据寄存器自动清除功能

当 ADC\_CR0.CLREN 为“1”，ADC 转换数据寄存器 ADC\_DR 被 CPU 或者 DMA 读取后将自动被清除为“0x0000”。

使用此功能能够检测到数据寄存器 ADC\_DR 是否被更新。以下将举例说明。

- 当 ADC\_CR0.CLREN 为“0”，自动清除功能禁止的情况下，待测模拟量（0x0222）由于某种原因未被转换或结果未被更新到数据寄存器 ADC\_DR 中，ADC\_DR 寄存器继续保持前次转换值（0x0111）。ADC 转换完成中断处理中将读取未被更新的（0x0111）。为检测 ADC 转换值是否有效，需要额外将以前的转换值存储到 RAM 中，通过对比转换结果来判断。
- 如果 ADC\_CR0.CLREN 为“1”，自动清除功能许可的情况下，前次的转换结果（0x0111）被 CPU 或 DMA 读取后，ADC\_DR 寄存器将自动被清除为“0x0000”，此后进行 ADC 转换后，如果转换结果未被正确传送到 ADC\_DR 寄存器中，ADC\_DR 寄存器将保持“0x0000”，这时，如果在中断处理中读出了“0x0000”，将很容易判断 ADC 转换数据是否正确被存储。

### 17.3.10 转换数据平均计算功能

ADC 转换平均计算功能是指对同一通道进行连续进行 2,4,8,16,32,64,128 或 256 次转换，并将转换结果进行平均后保存到数据寄存器的功能。使用平均计算功能可以去除一定的噪声成分使转换结果更加精确。

寄存器 ADC\_CR0.AVCNT[2:0]设置连续转换的次数，寄存器 ADC\_AVCHSEL 选择任意一个或多个需要平均的通道。

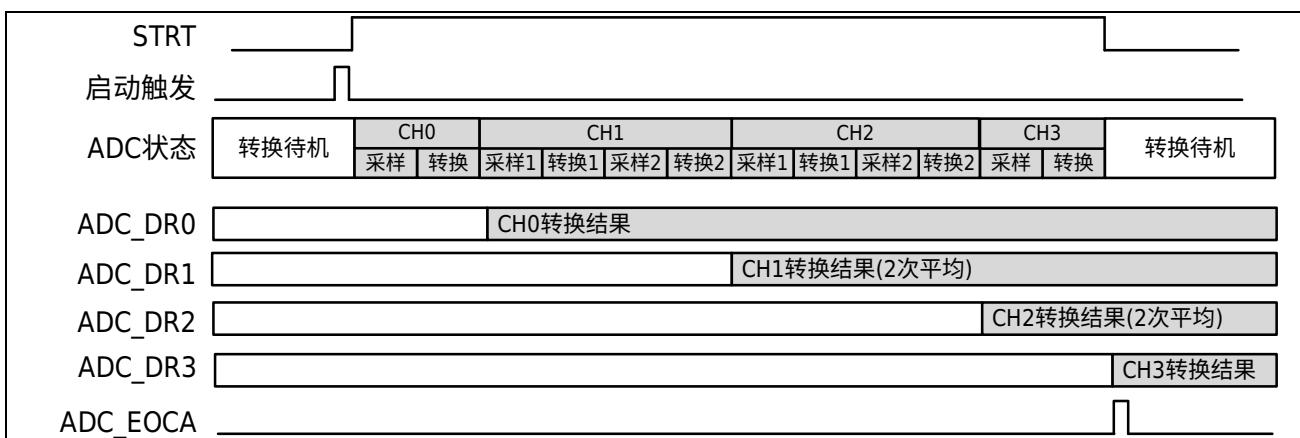


图 17-9 平均功能有效时的转换动作

图 17-9 中，序列 A 单次扫描模式，选择转换 CH0~3 这 4 个通道，其中 CH1、2 设置为 2 次平均模式。在扫描过程中，CH1、2 都会进行连续两次转换，并将平均后的结果保存至对应通道的数据寄存器 ADC\_DR1、2 中。

### 17.3.11 可编程增益放大器 PGA

本 MCU 搭载了可编程增益放大器 PGA，可以设置寄存器 ADC\_PGACR，有效 PGA 电路并选择增益倍数，增益范围 x2~x32 可选择。此时，模拟输入先经过 PGA 电路进行放大，然后再输入到 ADC 模块进行转换。

使用 PGA 前请先设置寄存器 PWC\_FCG3.CMBIAS 允许 PGA，并等待 2us 的启动稳定时间。

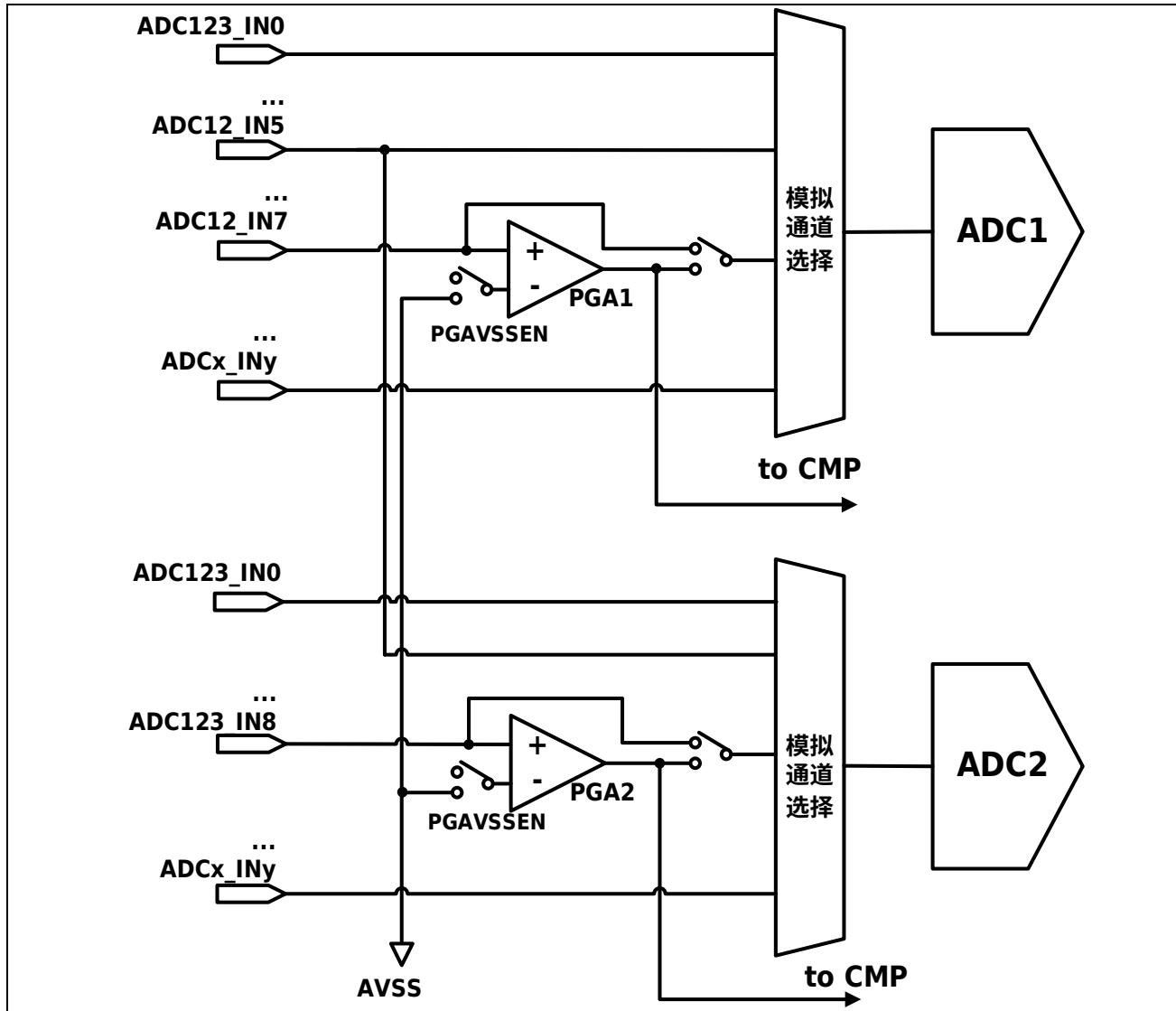


图 17-10 AD 与 PGA 通道示意图

如上图, ADC1 连接了 PGA1, 与物理通道 ADC12\_IN7 对应。ADC2 连接了 PGA2, 与物理通道 ADC123\_IN8 对应。ADC3 不支持 PGA。共 2 个 PGA 通道。

### 17.3.12 多 ADC 协同工作模式

在搭载有两个或三个 ADC 模块的芯片上，可以使用 ADC 协同工作模式。

在 ADC 协同工作模式下，ADC1 作为主控单元，通过 ADC1 的触发信号来同步 ADC2 和 ADC3 的转换。即 ADC2 和 ADC3 的序列 A 触发源选择寄存器 ADC\_TRGSR.TRGSEL[2:0]设置无效。所有 ADC 模块均由 ADC1 的序列 A 触发源选择寄存器选定的触发源来触发。该模式下 ADC\_STR.START 寄存器写 1 不会启动转换，即软件启动无效。

使用协同工作模式时，请禁止序列 B 动作，以免打乱同步。

可以设置 ADC1, ADC2 两个 ADC 模块协同工作，也可以 ADC1, ADC2, ADC3 三个 ADC 模块协同工作。根据产品具体规格，ADC3 可能未搭载。

ADC 可配置成以下四种协同工作模式：

- 单次并行触发模式
- 单次延迟触发模式
- 循环并行触发模式
- 循环延迟触发模式

#### 单次并行触发模式

ADC1 的序列 A 触发条件同时触发处于协同工作模式的所有 ADC 模块，且只触发一次。

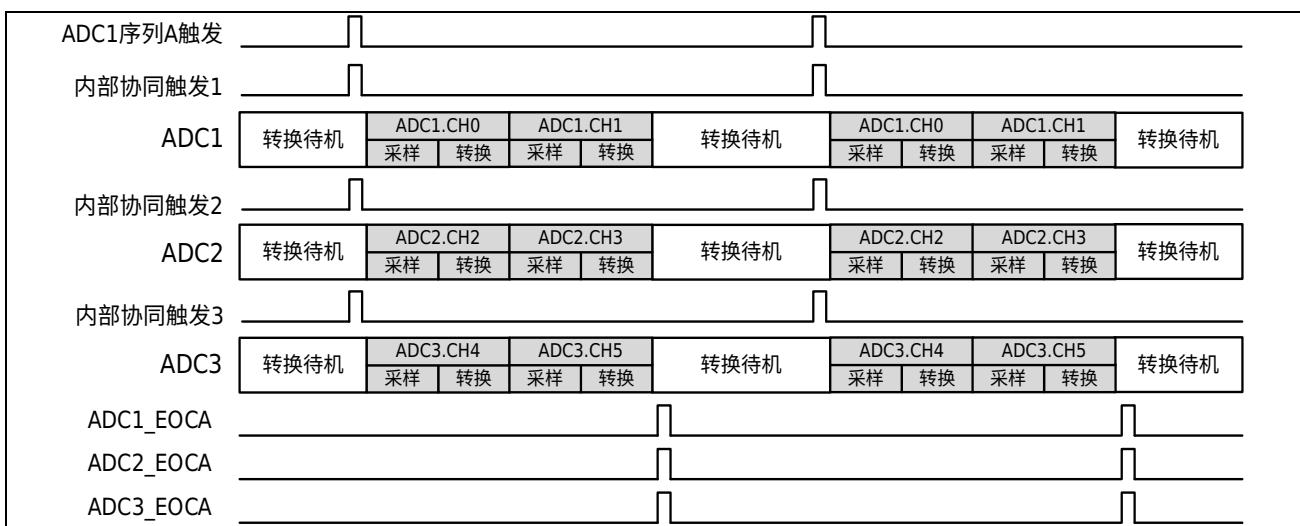


图 17-11 单次并行触发模式（三 ADC）

#### 注意：

- 禁止多个 ADC 同时对同一个模拟输入进行转换，一个模拟通道同一时间只能给一个 ADC 模块采样，否则精度不做保证，下同。

本模式的软件设定流程如下：

1. 协同工作许可寄存器 ADC\_SYNCCR.SYNCEN 写 0，确认协同工作无效
2. 设置 ADC1 模块
  - a) 确认 ADC1\_STR.STRT 为 0，ADC1 处于转换待机状态
  - b) 设置控制寄存器 ADC1\_CR0.MS[1:0] 为 00b：序列 A 单次扫描模式，或 01b：序列 A 连续扫描模式
  - c) 设置序列 A 通道选择寄存器 ADC1\_CHSELRA
  - d) 设置采样时间寄存器 ADC1\_SSTR
  - e) 设置序列 A 触发源选择寄存器 ADC1\_TRGSR
3. 设置 ADC2 模块
  - a) 确认 ADC2\_STR.STRT 为 0，ADC2 处于转换待机状态
  - b) 设置控制寄存器 ADC2\_CR0.MS[1:0]，通道选择寄存器 ADC2\_CHSELRA，通道采样时间寄存器 ADC2\_SSTR

**注意：**

- 为保证 ADC2 与 ADC1 的同步工作，上述寄存器尽量与 ADC1 的寄存器设置相同的值。具体通道无需相同，只要保持通道数，以及对应通道的采样时间一致即可。
4. 设置 ADC3 模块（三 ADC 协同工作时）
    - a) 确认 ADC3\_STR.STRT 为 0，ADC3 处于转换待机状态
    - b) 设置控制寄存器 ADC3\_CR0.MS[1:0]，通道选择寄存器 ADC3\_CHSELRA，通道采样时间寄存器 ADC3\_SSTR

**注意：**

- 同 ADC2，为保证 ADC3 与 ADC1 的同步工作，上述寄存器尽量与 ADC1 的寄存器设置相同的值。
5. 设置协同模式控制寄存器 ADC\_SYNCCR.SYNCMD[2:0]，写 010b：ADC1, ADC2 两 ADC 协同工作。或者写 011b：ADC1, ADC2, ADC3 三 ADC 协同工作
  6. 协同工作许可寄存器 ADC\_SYNCCR.SYNCEN 写 1，协同工作有效
  7. 等待 ADC1 序列 A 触发源输入，在 ADC1, 2, 3 完成转换后处理结果

### 单次延迟触发模式

ADC1 的序列 A 触发条件触发 ADC1 之后，经过设定的延迟后触发 ADC2 启动转换，再经过设定的延迟后触发 ADC3 启动转换，每个 ADC 模块只触发一次。

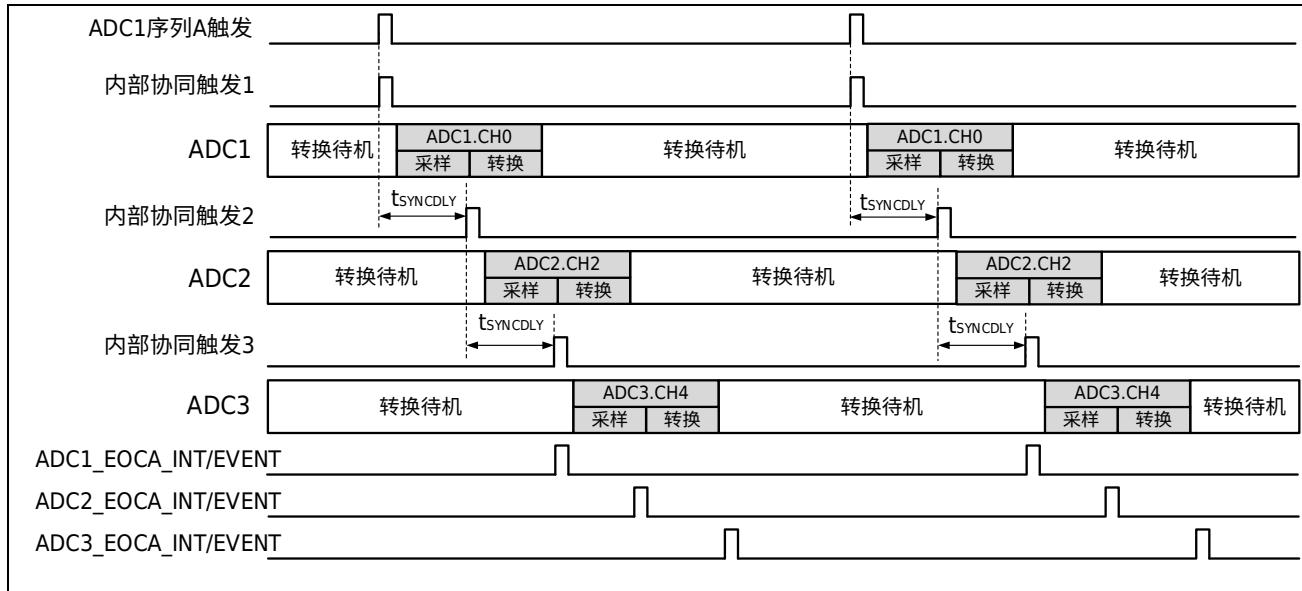


图 17-12 单次延迟触发模式（三 ADC）

#### 注意：

- ADC1 序列 A 触发第一次输入后，ADC3 协同触发发生前，再次输入 ADC1 序列 A 触发将被忽略。
- 若各 ADC 单元转换的是同一模拟通道，需要错开采样时间，即延迟时间  $t_{SYNCDLY}$  与通道开采样时间  $t_{SPL}$  需满足： $t_{SYNCDLY} > t_{SPL}$ 。

本模式的软件设定流程如下：

1. 协同工作许可寄存器 ADC\_SYNCCR.SYNCEN 写 0，确认协同工作无效。
2. 设置 ADC1, 2, 3 模块（参考单次平行模式）。
3. 设置协同模式控制寄存器 ADC\_SYNCCR.SYNCDLY[7:0]，设置两个 ADC 的启动延迟。
4. 设置协同模式控制寄存器 ADC\_SYNCCR.SYNCMD[2:0]，写 000b：ADC1, ADC2 两 ADC 协同工作。或者写 001b：ADC1, ADC2, ADC3 三 ADC 协同工作。
5. 协同工作许可寄存器 ADC\_SYNCCR.SYNCEN 写 1，协同工作有效。
6. 等待 ADC1 序列 A 触发源输入，在 ADC1, 2, 3 完成转换后处理结果。

## 循环并行触发模式

ADC1 的序列 A 触发条件同时触发处于协同工作模式的所有 ADC 模块，且之后每经过指定延迟之后会再次同时触发所有 ADC 模块。直至用户主动软件停止 ADC1 模块，或者禁止协同工作模式。

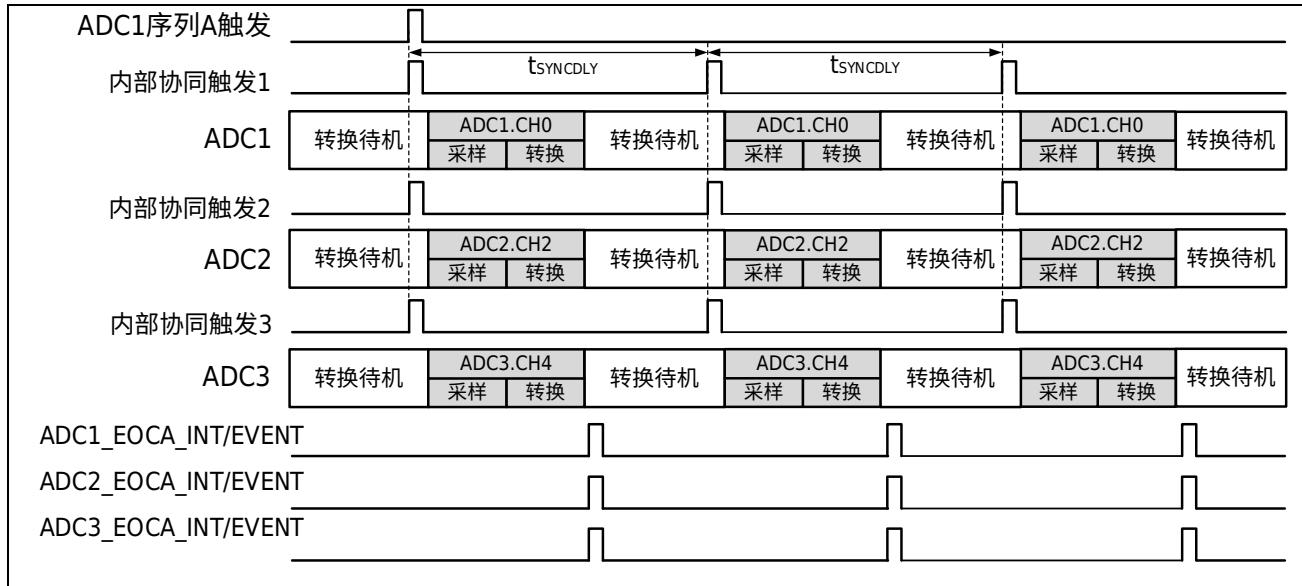


图 17-13 循环并行触发模式（三 ADC）

### 注意：

- 延迟时间  $t_{SYNCDLY}$  与一次扫描转换的时间  $t_{SCAN}$  需满足： $t_{SYNCDLY} > t_{SCAN}$ 。

本模式的软件设定流程如下：

1. 协同工作许可寄存器 ADC\_SYNCCR.SYNCEN 写 0，确认协同工作无效。
2. 设置 ADC1, 2, 3 模块，参考单次并行模式。ADC\_CR0.MS[1:0]设置为 00b：序列 A 单次扫描模式。
3. 设置协同模式控制寄存器 ADC\_SYNCCR.SYNCDLY[7:0]，设置每次并行触发的延迟。
4. 设置协同模式控制寄存器 ADC\_SYNCCR.SYNCMD[2:0]，写 110b：ADC1, ADC2 两 ADC 协同工作。或者写 111b：ADC1, ADC2, ADC3 三 ADC 协同工作。
5. 协同工作许可寄存器 ADC\_SYNCCR.SYNCEN 写 1，协同工作有效。
6. 等待 ADC1 序列 A 触发源输入，在 ADC1, 2, 3 完成转换后处理结果。

### 循环延迟触发模式

ADC1 的序列 A 触发条件触发 ADC1 之后，每经过设定的延迟后，依次循环不断触发 ADC2，ADC3，ADC1，ADC2...，直至用户主动软件停止 ADC1 模块，或者禁止协同工作模式。

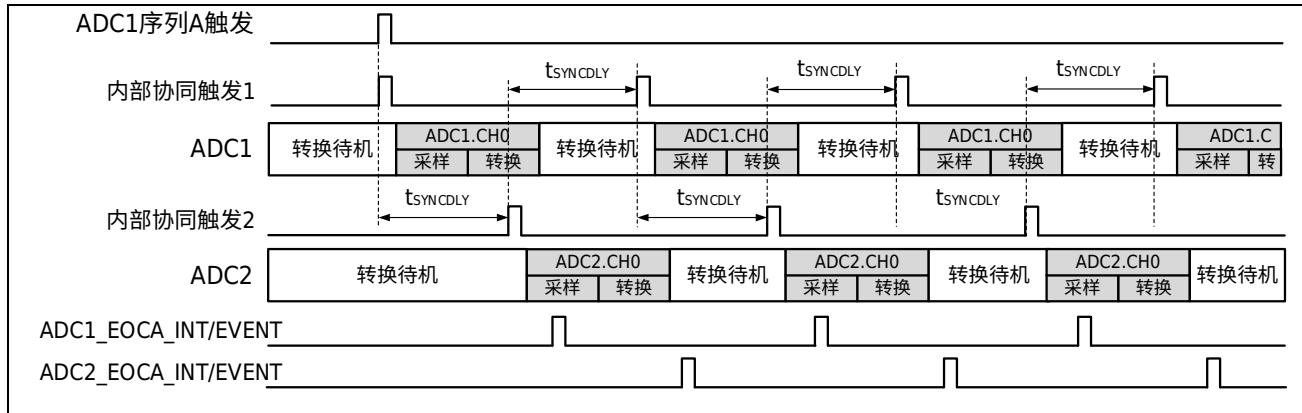


图 17-14 循环延迟触发模式（两 ADC）

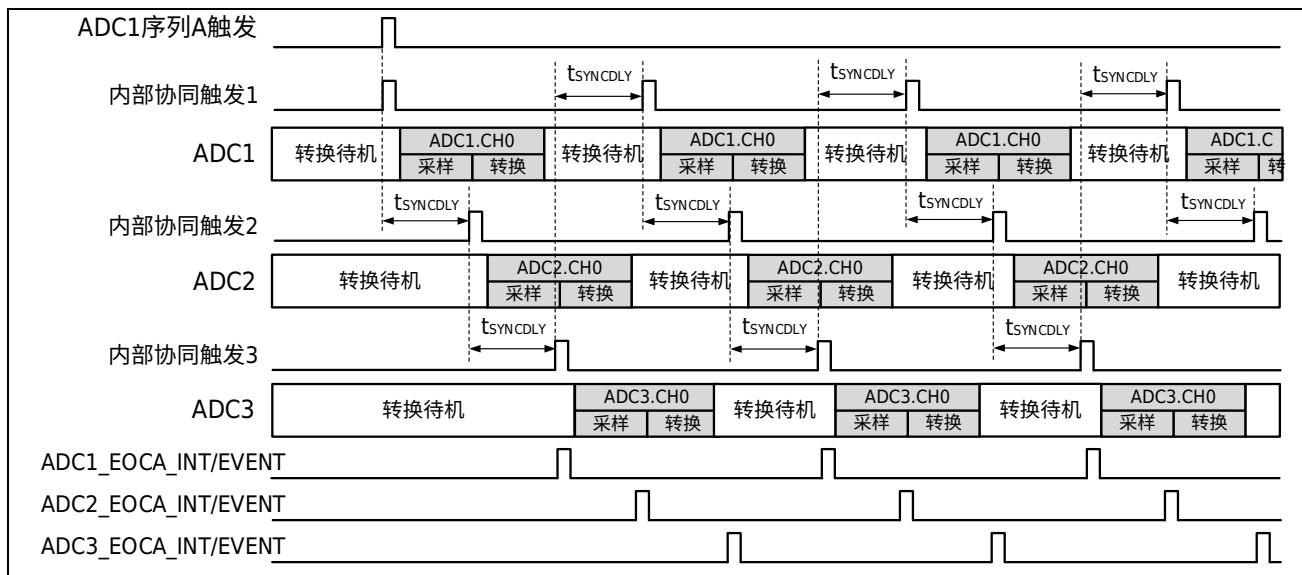


图 17-15 循环延迟触发模式（三 ADC）

#### 注意：

- 两 ADC 协同工作时，延迟时间  $t_{SYNCDLY}$  与一次扫描转换的时间  $t_{SCAN}$  需满足：  
 $t_{SYNCDLY} > t_{SCAN}/2$ 。三 ADC 协同工作时需满足： $t_{SYNCDLY} > t_{SCAN}/3$ 。同时，如 ADC1，ADC2，ADC3 转换的是同一模拟通道，还需要错开采样时间，即  $t_{SYNCDLY} > t_{SPL}$ 。

本模式的软件设定流程如下：

1. 协同工作许可寄存器 ADC\_SYNCCR.SYNCEN 写 0，确认协同工作无效。
2. 设置 ADC1, 2, 3 模块，参考循环并行触发模式。
3. 设置协同模式控制寄存器 ADC\_SYNCCR.SYNCDLY[7:0]，设置每次触发的延迟。
4. 设置协同模式控制寄存器 ADC\_SYNCCR.SYNCMD[2:0]，写 100b: ADC1, ADC2 两 ADC 协同工作。或者写 101b: ADC1, ADC2, ADC3 三 ADC 协同工作。
5. 协同工作许可寄存器 ADC\_SYNCCR.SYNCEN 写 1，协同工作有效。
6. 等待 ADC1 序列 A 触发源输入，在 ADC1, 2, 3 完成转换后处理结果。

### 17.3.13 中断和事件信号输出

ADC 模块可以产生以下四种事件输出，每个事件发生时，若对应的中断许可寄存器设置为有效时，同时输出中断申请。

1. 序列 A 扫描结束 ADC\_EOCA，对应中断许可寄存器 ADC\_ICR.EOCAIEN
2. 序列 B 扫描结束 ADC\_EOCB，对应中断许可寄存器 ADC\_ICR.EOCBIEN
3. 模拟看门狗 0 ADC\_CMP0，对应中断许可寄存器 ADC\_AWDCR.AWD0IEN
4. 模拟看门狗 1 ADC\_CMP1，对应中断许可寄存器 ADC\_AWDCR.AWD1IEN

上述四种事件输出，可以启动其他片内周边模块，包括启动 DMA 传送。利用 DMA 传送可以连续读取 ADC 转换结果，不需要软件干预，完全由硬件实现，降低 CPU 的负荷。DMA 的设定请参考 DMA 说明章节。事件信号输出和中断使能位的控制无关，只要条件发生就会输出。

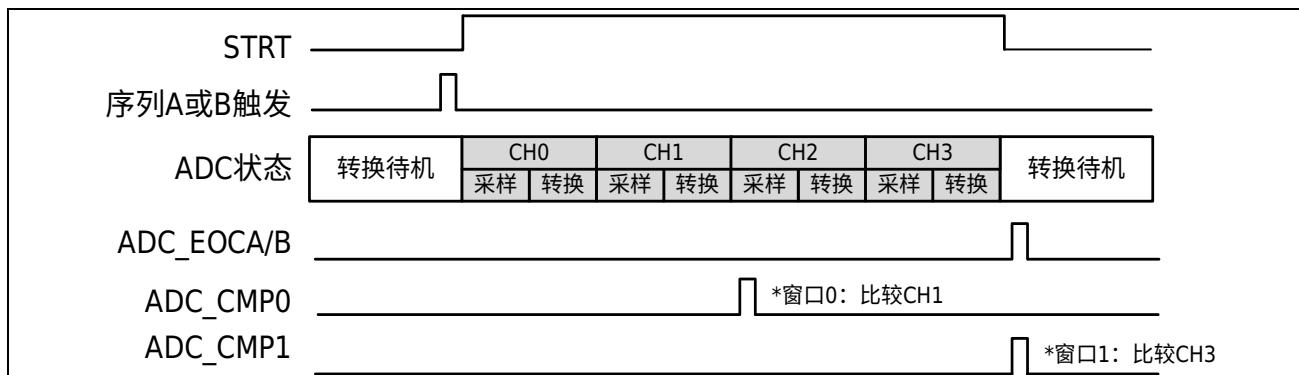


图 17-16 ADC 中断和事件输出时序

## 17.4 寄存器说明

### 17.4.1 寄存器一览

表 17-5 ADC 寄存器一览

单元 1 BASE\_ADDR: 0x40040000

单元 2 BASE\_ADDR: 0x40040400

单元 3 BASE\_ADDR: 0x40040800

寄存器名	符号	偏移地址	位宽	复位值
ADC启动寄存器	ADC_STR	0x00	8	0x00
ADC控制寄存器0	ADC_CR0	0x02	16	0x0000
ADC控制寄存器1	ADC_CR1	0x04	16	0x0000
ADC转换开始触发寄存器	ADC_TRGSR	0x0a	16	0x0000
ADC通道选择寄存器A	ADC_CHSELRA	0x0c	32	0x00000000
ADC通道选择寄存器B	ADC_CHSELRB	0x10	32	0x00000000
ADC平均通道选择寄存器	ADC_AVCHSELR	0x14	32	0x00000000
ADC采样周期寄存器	ADC_SSTRx[注]	0x20+x	8	0x0b
	ADC_SSTRL	0x30	8	0x0b
ADC通道映射控制寄存器0	ADC_CHMUXR0	0x38	16	0x3210
ADC通道映射控制寄存器1	ADC_CHMUXR1	0x3a	16	0x7654
ADC通道映射控制寄存器2	ADC_CHMUXR2	0x3c	16	0xba98
ADC通道映射控制寄存器3	ADC_CHMUXR3	0x3e	16	0xfedc
ADC中断状态寄存器	ADC_ISR	0x44	8	0x00
ADC中断许可寄存器	ADC_ICR	0x45	8	0x03
ADC中断状态复位寄存器	ADC_ISCLRR	0x46	8	0x00
ADC协同模式控制寄存器	ADC_SYNCCR	0x4c	16	0x0c00
ADC数据寄存器	ADC_DRy[注]	0x50+2*y	16	0x0000
模拟看门狗控制寄存器	ADC_AWDCR	0xa0	16	0x0000
模拟看门狗状态寄存器	ADC_AWDSR	0xa2	8	0x00
模拟看门狗状态复位寄存器	ADC_AWDSCLRR	0xa3	8	0x00
模拟看门狗窗口0阀值寄存器	ADC_AWD0DR0	0xa4	16	0x0000
	ADC_AWD0DR1	0xa6	16	0xffff
模拟看门狗窗口0比较通道选择寄存器	ADC_AWD0CHSR	0xa8	8	0x00
模拟看门狗窗口1阀值寄存器	ADC_AWD1DR0	0xac	16	0x0000
	ADC_AWD1DR1	0xae	16	0xffff
模拟看门狗窗口1比较通道选择寄存器	ADC_AWD1CHSR	0xb0	8	0x00
可编程增益放大器控制寄存器	ADC_PGACR	0xc0	8	0x00
可编程增益放大器参考地选择寄存器	ADC_PGAVSSENR	0xc4	8	0x00

**注：**

ADC\_SSTRx 中的 “x” 表示通道，ADC1/ADC2/ADC3 中 x=0~15。

ADC\_DRy 中的 “y” 表示通道，ADC1 中 y=0~23、ADC2 中 y=0~23、ADC3 中 y=0~24。

### 17.4.2 ADC 启动寄存器 ADC\_STR

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0				
—	—	—	—	—	—	—	STRT				
<hr/>											
位	标记	位名	功能								
b7-b1	Reserved	—	读出时为0，写入时写0								
<hr/>											
0: 停止转换 1: 开始转换 置“1”条件： (1) 软件设置 (2) 选择的触发条件发生 (3) ADC转换中 清“0”条件： (1) 软件清“0” (2) 转换结束后自动清“0” 注意： - STRT为0 (ADC空闲中) 时写1产生软件触发，启动序列A - STRT为1 (ADC动作中) 时写1无效。 - STRT为1时写0表示强制停止AD转换。若ADC_TRGSR设置了0x0以外的值且不希望ADC再启动，请先将ADC_TRGSR设置为0，再对STRT写0。 - STRT为0时写0无效。											
b0	STRT	AD转换开始									
<hr/>											

### 17.4.3 ADC 控制寄存器 0 ADC\_CRO

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8
—	—	—	—	—	AVCNT[2:0]		
b7	b6	b5	b4	b3	b2	b1	b0
DFMT	CLREN	ACCSEL[1:0]			—	—	MS[1:0]
<hr/>							
位	标记	位名	功能				R/W
b15-b12	Reserved	—	读出时为0，写入时写0				R/W
			0 0 0: 连续转换2次平均 0 0 1: 连续转换4次平均 0 1 0: 连续转换8次平均 0 1 1: 连续转换16次平均 1 0 0: 连续转换32次平均 1 0 1: 连续转换64次平均 1 1 0: 连续转换128次平均 1 1 1: 连续转换256次平均				
b10-b8	AVCNT[2:0]	次数选择					R/W
b7	DFMT	数据格式	0: 转换数据右对齐 1: 转换数据左对齐				
b6	CLREN	数据寄存器自动清除	0: 自动清除禁止 1: 自动清除许可	注意：CLREN位设定后，寄存器ADC_DRx将在CPU、DMA等读取后被自动清除。自动清除功能主要用于检测数据寄存器是否更新。			
b5-b4	ACCSEL[1:0]	分辨率选择	0 0: 12位分辨率 0 1: 10位分辨率 1 0: 8位分辨率 1 1: 设定禁止				
b3-b2	Reserved	—	读出时为0，写入时写0				
b1-b0	MS[1:0]	模式选择	0 0: 序列A单次扫描模式，序列B无效 0 1: 序列A连续扫描模式，序列B无效 1 0: 序列A单次扫描模式，序列B单次扫描模式 1 1: 序列A连续扫描模式，序列B单次扫描模式				

#### 注意：

- 请在 ADC\_STR.START 为“0”时设置本寄存器。

#### 17.4.4 ADC 控制寄存器 1 ADC\_CR1

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8
—	—	—	—	—	—	—	—
b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	RSCHSEL	—	—

位	标记	位名	功能	R/W
b15-b3	Reserved	—	读出时为0, 写入时写0	R/W
b2	RSCHSEL	序列A重启通道选择	0: 被序列B中断后, 序列A重启时从被中断通道开始继续扫描 1: 被序列B中断后, 序列A重启时从被第一个通道开始重新扫描	R/W
b1-b0	Reserved	—	读出时为0, 写入时写0	R/W

##### 注意：

- 请在 ADC\_STR.START 为“0”时设置本寄存器。

### 17.4.5 ADC 转换开始触发寄存器 ADC\_TRGSR

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8
TRGENB	—	—	—	—	—	TRGSELB[1]	TRGSELB[0]
b7	b6	b5	b4	b3	b2	b1	b0
TRGENA	—	—	—	—	—	TRGSELA[1]	TRGSELA[0]

位	标记	位名	功能	R/W
b15	TRGENB	序列B触发使能	0: 序列B片内或外部引脚触发禁止 1: 序列B片内或外部引脚触发许可 注意：选择外部引脚触发有效。如果ADTRGx由“High”变为“Low”，检测到下降沿，则扫描转换开始，请保持“Low”1.5*PCLK4周期以上。	R/W
b14-b10	Reserved	—	读出时为0，写入时写0	R/W
b9-b8	TRGSELB[1:0]	序列B触发条件选择	在序列B有效模式下(ADC_CR0.MS[1]=1)，作为序列B的触发条件 00b: ADTRGx 01b: IN_TRGx0 10b: IN_TRGx1 11b: IN_TRGx0 + IN_TRGx1 注意：只在序列B有效模式下有效。其他模式设定无效。 两次触发的间隔必须大于或等于扫描周期t <sub>SCAN</sub> ，若小于则触发无效。	R/W
b7	TRGENA	序列A触发使能	0: 序列A片内或外部引脚触发禁止 1: 序列A片内或外部引脚触发许可 注意：选择外部引脚触发有效。如果ADTRGx由“High”变为“Low”，检测到下降沿，则扫描转换开始，请保持“Low”1.5*PCLK4周期以上。	R/W
b6-b2	Reserved	—	读出时为0，写入时写0	R/W
b1-b0	TRGSELA[1:0]	序列A触发条件选择	序列A的触发条件。 00b: ADTRGx (x=1~3,代表ADC单元编号) 01b: IN_TRGx0 10b: IN_TRGx1 11b: IN_TRGx0 + IN_TRGx1 注意： ADC空闲中对ADC_STR.START写1软件触发，无视TRGENA, TRGSELA[1:0]的设定，直接开始ADC转换。 两次触发的间隔必须大于或等于扫描周期t <sub>SCAN</sub> ，若小于则触发无效。	R/W

#### 注意：

- 在 ADC\_STR.START 为“0”时设置本寄存器。

### 17.4.6 ADC 通道选择寄存器 A ADC\_CHSELRA

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
CHSELRA[31:24]							
b23	b22	b21	b20	b19	b18	b17	b16
CHSELRA[23:16]							
b15	b14	b13	b12	b11	b10	b9	b8
CHSELRA[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
CHSELRA[7:0]							

位	标记	位名	功能	R/W
序列A的通道选择，每一位代表一个通道，CHSELRA[x]代表通道CHx，可选择任意组合。				
b31-b0	CHSELRA[31:0]	转换通道选择	0: 未选择对应通道 1: 选择对应通道	R/W
不存在通道的对应位为Reserved位，读出时为0，写入时写0。 注意：请不要在序列A和序列B中选择相同的通道。				

#### 注意：

- 请在 ADC\_STR STRT 为“0”时设置本寄存器。

### 17.4.7 ADC 通道选择寄存器 B ADC\_CHSELRB

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
CHSELB[31:24]							
b23	b22	b21	b20	b19	b18	b17	b16
CHSELB[23:16]							
b15	b14	b13	b12	b11	b10	b9	b8
CHSELB[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
CHSELB[7:0]							

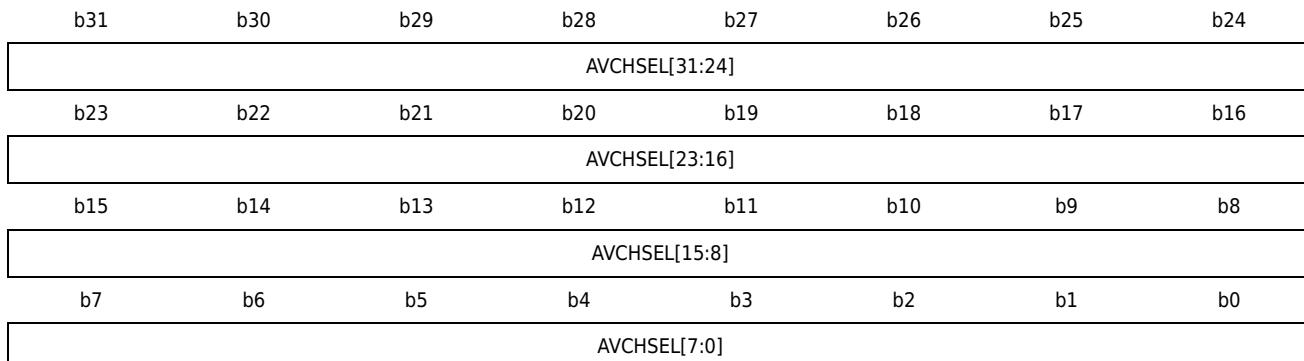
位	标记	位名	功能	R/W
序列B的通道选择，每一位代表一个通道，CHSELB[x]代表通道CHx，可选择任意组合。				
只有在双序列扫描模式是才有效。				
b31-b0	CHSELB[31:0]	转换通道选择	0: 未选择对应通道 1: 选择对应通道  不存在通道的对应位为Reserved位，读出时为0，写入时写0。 注意：请不要在序列A和序列B中选择相同的通道。	R/W

#### 注意：

- 请在 ADC\_STR STRT 为“0”时设置本寄存器。

### 17.4.8 ADC 平均通道选择寄存器 ADC\_AVCHSELR

复位值：0x00000000



位	标记	位名	功能	R/W
每一位代表一个通道，AVCHSEL[x]代表通道CHx，可选择任意组合。				
b31-b0	AVCHSEL[31:0]	平均通道选择	不存在通道的对应位为Reserved位，读出时为0，写入时写0。  注意：当AVCHSEL与ADC_CHSELRA或ADC_CHSELB的对应的通道同时被选择时，则该通道在扫描时将连续执行设定次数ADC转换，并对转换结果进行平均计算后更新入数据寄存器。如果对应通道AVCHSEL未被设定，该通道将执行普通一次转换。	R/W

#### 注意：

- 请在 ADC\_STR STRT 为“0”时设置本寄存器。

### 17.4.9 ADC 采样周期寄存器 ADC\_SSTRx, x=0~15/ADC\_SSTRL

复位值：0x0B

b7	b6	b5	b4	b3	b2	b1	b0					
位	标记	位名	功能					R/W				
SST[7:0]												
采样周期数可被设为5到255个周期。												
b7-b0	SST[7:0]	采样周期数	通道CH0~15由ADC_SSTRx, x=0~15来设置，其他通道由ADC_SSTRL来设置。	注意：PCLK2频率为50MHz时，一个采样周期为20ns，初始的转换状态有11个采样周期。当外部输入阻抗R <sub>Ain</sub> 太大采样时间不足或者PCLK2频率低时可以设置寄存器进行调整采样时间。采样时间不要少于5周期。	SST ≥ (R <sub>Ain</sub> +R <sub>ADC</sub> ) *C <sub>ADC</sub> *ln(2 <sup>N+2</sup> )*f <sub>ADC</sub> +1	其中：R <sub>Ain</sub> 表示外部输入阻抗 (Ω)， R <sub>ADC</sub> 表示内部采样开关电阻 (Ω)， C <sub>ADC</sub> 表示内部采样和保持电容 (F)， N表示AD分辨率 (12/10/8)， f <sub>ADC</sub> 表示PCLK2频率 (Hz)。具体参考电器特性相关说明。	R/W					

#### 注意：

- 请在 ADC\_STR.STRT 为“0”时设置本寄存器。
- 内部基准电压的采样时间不要小于 1us。

## 17.4.10 ADC 通道映射控制寄存器 ADC\_CHMUXR

ADC\_CHMUXR0 复位值: 0x3210

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH03MUX[3:0]				CH02MUX[3:0]				CH01MUX[3:0]				CH00MUX[3:0]			

ADC\_CHMUXR1 复位值: 0x7654

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH07MUX[3:0]				CH06MUX[3:0]				CH05MUX[3:0]				CH04MUX[3:0]			

ADC\_CHMUXR2 复位值: 0xba98

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH11MUX[3:0]				CH10MUX[3:0]				CH09MUX[3:0]				CH08MUX[3:0]			

ADC\_CHMUXR3 复位值: 0xfedc

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH15MUX[3:0]				CH14MUX[3:0]				CH13MUX[3:0]				CH12MUX[3:0]			

位	标记	位名	功能	R/W
不存在的通道对应位读出时为0，写入时写0				
对不同的ADC单元CHx的映射关系如下：				
CHxMUX[3:0] 通道x映射选择 x=0~15	设定值	ADC1 映射对象	ADC2 映射对象	ADC3 映射对象
	0x0	ADC123_IN0	ADC123_IN0	ADC123_IN0
	0x1	ADC123_IN1	ADC123_IN1	ADC123_IN1
	0x2	ADC123_IN2	ADC123_IN2	ADC123_IN2
	0x3	ADC123_IN3	ADC123_IN3	ADC123_IN3
	0x4	ADC12_IN4	ADC12_IN4	ADC3_IN4
	0x5	ADC12_IN5	ADC12_IN5	ADC3_IN5
	0x6	ADC12_IN6	ADC12_IN6	ADC3_IN6
	0x7	ADC12_IN7	ADC12_IN7	ADC3_IN7
	0x8	ADC123_IN8	ADC123_IN8	ADC123_IN8
	0x9	ADC123_IN9	ADC123_IN9	ADC123_IN9
	0xa	ADC123_IN10	ADC123_IN10	ADC123_IN10
	0xb	ADC123_IN11	ADC123_IN11	ADC123_IN11
	0xc	ADC123_IN12	ADC123_IN12	ADC123_IN12
	0xd	ADC123_IN13	ADC123_IN13	ADC123_IN13
	0xe	ADC12_IN14	ADC12_IN14	ADC3_IN14
	0xf	ADC12_IN15	ADC12_IN15	ADC3_IN15

注意：请不要设置到不存在的模拟输入上。

### 注意：

- 请在 ADC\_STR.START 为“0”时设置本寄存器。

**17.4.11 ADC 中断状态寄存器 ADC\_ISR**

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	SASTPDF	—	—	EOCBF	EOCAF

位	标记	位名	功能	R/W
b7-b5	Reserved	—	读出时为0，写入时写0	R/W
b4	SASTPDF	序列A被打断标志	双序列扫描模式，序列A扫描过程中被高优先级的序列B打断时置1 本寄存器位为只读位	R
b3-b2	Reserved	—	读出时为0，写入时写0	R/W
b1	EOCBF	序列B转换完成标志	序列B所选通道全部扫描完成后置1 本寄存器位为只读位	R
b0	EOCAF	序列A转换完成标志	序列A所选通道全部扫描完成后置1 本寄存器位为只读位	R

**17.4.12 ADC 中断许可寄存器 ADC\_ICR**

复位值：0x03

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	EOCBIEN	EOCAIEN

位	标记	位名	功能	R/W
b7-b2	Reserved	—	读出时为0，写入时写0	R/W
b1	EOCBIEN	序列B转换完成中断使能	0: 序列B转换完成中断禁止 1: 序列B转换完成中断许可	R/W
b0	EOCAIEN	序列A转换完成中断使能	0: 序列A转换完成中断禁止 1: 序列A转换完成中断许可	R/W

### 17.4.13 ADC 中断状态复位寄存器 ADC\_ISCLRR

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	CLRSASTPDF	—	—	CLREOCBF	CLREOCAF

位	标记	位名	功能	R/W
b7-b5	Reserved	—	读出时为0，写入时写0	R/W
b4	CLRSASTPDF	序列A被打断标志复位	写0无任何效果，写1复位SASTPDF状态位，读出永远为0	R/W
b3-b2	Reserved	—	读出时为0，写入时写0	R/W
b1	CLREOCBF	序列B转换完成标志复位	写 0 无任何效果，写 1 复位 EOCBF 状态位，读出永远为0	R/W
b0	CLREOCAF	序列A转换完成标志复位	写 0 无任何效果，写 1 复位 EOCAF 状态位，读出永远为0。	R/W

### 17.4.14 ADC 协同模式控制寄存器 ADC\_SYNCCR

复位值：0x0c00

b15	b14	b13	b12	b11	b10	b9	b8
SYNCMD[7:0]							
b7	b6	b5	b4	b3	b2	b1	b0
—	SYNCMD[2]	SYNCMD[1]	SYNCMD[0]	—	—	—	SYNCEN
<hr/>							
位	标记	位名	功能	R/W			
延迟触发模式时，两个ADC的启动延迟时间 $t_{SYNCDLY}$ 。 0x1表示 $t_{SYNCDLY} = 1 \times PCLK2$ , 0xff表示 $t_{SYNCDLY} = 255 \times PCLK2$ 注意：在SYNCEN为“0”时设置本寄存器。请不要写入0x00。 根据各ADC的采样时间，转换时间，设置合理的延迟时间，避免多个ADC同时处于采样状态引起的误差增大，避免ADC还未转换结束就再次发生触发，导致同步失败。推荐设置如下：							
b15-b8	SYNCMD[7:0]	同步延迟时间	单次延迟触发模式： $t_{SYNCDLY} > t_{SPL}$ 两ADC循环延迟触发模式： $t_{SYNCDLY} > t_{SPL}$ , 且 $t_{SYNCDLY} > t_{SCAN}/2$ 三ADC循环延迟触发模式： $t_{SYNCDLY} > t_{SPL}$ , 且 $t_{SYNCDLY} > t_{SCAN}/3$ 单次并行触发模式：本寄存器设置无效。 循环并行触发模式： $t_{SYNCDLY} > t_{SCAN}$	R/W			
b7	Reserved	—	读出时为0, 写入时写0	R/W			
SYNCMD[2] 0: 单次触发 1: 循环触发 SYNCMD[1] 0: 延迟触发模式 1: 并行触发模式 SYNCMD[0] 0: ADC1和ADC2同步工作, ADC3独立工作 1: ADC1, ADC2和ADC3同步工作 注意：在SYNCEN为“0”时设置本寄存器。使用单次触发时，请将需要同步的ADC设置为序列A单次扫描，或序列A连续扫描模式。使用循环触发模式时，请将ADC设置为序列A单次扫描模式。							
b6-b4	SYNCMD[2:0]	同步模式选择	0: ADC1和ADC2同步工作, ADC3独立工作 1: ADC1, ADC2和ADC3同步工作 注意：在SYNCEN为“0”时设置本寄存器。使用单次触发时，请将需要同步的ADC设置为序列A单次扫描，或序列A连续扫描模式。使用循环触发模式时，请将ADC设置为序列A单次扫描模式。	R/W			
b3-b1	Reserved	—	读出时为0, 写入时写0	R/W			
0: 同步模式无效 1: 同步模式有效 注意：							
b0	SYNCEN	同步模式许可	同步模式只支持序列A。在SYNCEN写1之前，请将参与同步的几个ADC的序列B关闭 (ADC_CR0.MS[1]=0)，并给序列A选择相同数目的通道，设置相同的通道采样时间 ADC_SSTRx。以避免各ADC扫描时间 $t_{SCAN}$ 不一致，引起后续同步失败。 软件对ADC1_STR.START写0强制停止转换时，SYNCEN自动清0。	R/W			

#### 注意：

- 本寄存器只在主控 ADC (即 ADC1) 中搭载，其他 ADC 单元中无此寄存器。

### 17.4.15 ADC 数据寄存器 ADC\_DRy, ADC1 y=0~23, ADC2 y=0~23, ADC3 y=0~24

ADC\_DR 寄存器是用于存储各通道 ADC 转换数据的只读寄存器。复位值为 0x0000

根据数据对齐方式和转换分辨率，转换结果数据存储方式有所不同。

数据右对齐-12 位分辨率

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	AD[11:0]											

数据右对齐-10 位分辨率

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		
0	0	0	0	0	0	AD[9:0]											

数据右对齐-8 位分辨率

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0				
0	0	0	0	0	0	0	0	AD[7:0]											

数据左对齐-12 位分辨率

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AD[11:0]												0	0	0	0

数据左对齐-10 位分辨率

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AD[9:0]												0	0	0	0

数据左对齐-8 位分辨率

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AD[7:0]												0	0	0	0

### 17.4.16 模拟看门狗控制寄存器 ADC\_AWDCR

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8
—	—	—	—	—	—	AWDCM[1:0]	
b7	b6	b5	b4	b3	b2	b1	b0
—	AWD1MD	AWD1IEN	AWD1EN	—	AWD0MD	AWD0IEN	AWD0EN

---

位	标记	位名	功能	R/W
b15-b10	Reserved	—	读出时为0, 写入时写0	R/W
b9-b8	AWDCM[1:0]	看门狗窗口组合选择	00: 窗口组合无效, ADC_CMP1输出窗口1独立比较结果 01: 窗口组合有效, ADC_CMP1输出窗口0与窗口1比较结果的逻辑或 10: 窗口组合有效, ADC_CMP1输出窗口0与窗口1比较结果的逻辑与 11: 窗口组合有效, ADC_CMP1输出窗口0与窗口1比较结果的逻辑异或 注意: 使用窗口组合功能时需要窗口0和窗口1都使能, 即AWD0EN写1, AWD1EN写1。 若窗口0与窗口1选择的比较通道不相同, 请保证扫描转换过程中, 窗口1所选通道在窗口0所选通道之后转换。ADC_CMP1中断或事件在窗口1所选通道转换结束时输出。	R/W
b7	Reserved	—	读出时为0, 写入时写0	R/W
b6	AWD1MD	看门狗窗口1比较模式	窗口1保护区域选择 0: 转换结果<AWD1DR0, 或转换结果>AWD1DR1 1: 转换结果≥AWD1DR0, 且转换结果≤AWD1DR1。	R/W
b5	AWD1IEN	看门狗窗口1中断使能	0: 看门狗窗口1比较中断ADC_CMP1无效 1: 看门狗窗口1比较中断ADC_CMP1有效	R/W
b4	AWD1EN	看门狗窗口1比较功能使能	0: 看门狗窗口1比较功能无效 1: 看门狗窗口1比较功能有效	R/W
b3	Reserved	—	读出时为0, 写入时写0	R/W
b2	AWD0MD	看门狗窗口0比较模式	窗口0保护区域选择 0: 转换结果<AWD0DR0, 或转换结果>AWD0DR1 1: 转换结果≥AWD0DR0, 且转换结果≤AWD0DR1。	R/W
b1	AWD0IEN	看门狗窗口0中断使能	0: 看门狗窗口0比较中断ADC_CMP0无效 1: 看门狗窗口0比较中断ADC_CMP0有效	R/W
b0	AWD0EN	看门狗窗口0比较功能使能	0: 看门狗窗口0比较功能无效 1: 看门狗窗口0比较功能有效	R/W

**17.4.17 模拟看门狗状态寄存器 ADC\_AWDSR**

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	AWDCMF	—	—	AWD1F	AWD0F

位	标记	位名	功能	R/W
b7-b5	Reserved	—	读出时为0, 写入时写0	R/W
b4	AWDCMF	看门狗窗口组合比较状态位	看门狗窗口组合比较功能有效（即AWDCM[1:0]=01b/10b/11b）时，窗口1所选通道转换结束时，窗口0与窗口1比较结果满足组合条件时置1。 对本寄存器位写无效。	R
b3-b2	Reserved	—	读出时为0, 写入时写0	R/W
b1	AWD1F	看门狗窗口1比较状态位	窗口1所选通道转换结束时，转换结果满足比较条件时置1。 对本寄存器位写无效。	R
b0	AWD0F	看门狗窗口0比较状态位	窗口0所选通道转换结束时，转换结果满足比较条件时置1。 对本寄存器位写无效。	R

**17.4.18 模拟看门狗状态复位寄存器 ADC\_AWDSCLRR**

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	CLRAWDCMF	—	—	CLRAWD1F	CLRAWD0F

位	标记	位名	功能	R/W
b7-b5	Reserved	—	读出时为0, 写入时写0	R/W
b4	CLRAWDCMF	看门狗窗口组合比较状态位复位	写0无任何效果, 写1复位AWDCMF状态位, 读出永远为0	R/W
b3-b2	Reserved	—	读出时为0, 写入时写0	R/W
b1	CLRAWD1F	看门狗窗口1比较状态复位	写0无任何效果, 写1复位AWD1F状态位, 读出永远为0	R/W
b0	CLRAWD0F	看门狗窗口0比较状态复位	写0无任何效果, 写1复位AWD1F状态位, 读出永远为0	R/W

### 17.4.19 模拟看门狗阀值寄存器 ADC\_AWD0DR0、ADC\_AWD0DR1、ADC\_AWD1DR0、ADC\_AWD1DR1

复位值：ADC\_AWD0DR0=0x0000, ADC\_AWD0DR1=0xffff

ADC\_AWD1DR0=0x0000, ADC\_AWD1DR1=0xffff

b15	b14	b13	b12	b11	b10	b9	b8
AWDDR[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
AWDDR[7:0]							
位	标记	位名	功能	R/W			
b15-b0	AWDDR[15:0]	比较数据	比较数据	R/W			

AWD0DR0 设置窗口 0 低阀值， AWD0DR1 设置窗口 0 高阀值。

AWD1DR0 设置窗口 1 低阀值， AWD1DR1 设置窗口 1 高阀值。

AWD0DR0、AWD0DR1、AWD1DR0、AWD1DR1 根据对齐方式（数据右对齐或左对齐），分辨率（12 位、10 位或者 8 位）会有差异。

- 数据右对齐-12 位分辨率 低 12 位[11:0]可用
- 数据右对齐-10 位分辨率 低 10 位[9:0]可用
- 数据右对齐-8 位分辨率 低 8 位[7:0]可用
- 数据左对齐-12 位分辨率 高 12 位[15:4]可用
- 数据左对齐-10 位分辨率 高 10 位[15:6]可用
- 数据左对齐-8 位分辨率 高 8 位[15:8]可用

### 17.4.20 模拟看门狗比较通道选择寄存器 ADC\_AWD0CHSR、ADC\_AWD1CHSR

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
—							
位	标记	位名	功能	R/W			
b7-b5	Reserved	—	读出时为0，写入时写0	R/W			
b4-b0	AWDCH[4:0]	看门狗比较通道选择	ADC_AWD0CHSR选择窗口0的比较通道，ADC_AWD1CHSR选择窗口1的比较通道。 0x00: CH0 0x01: CH1 以此类推 注意：请不要设置成不存在的通道	R/W			

### 17.4.21 可编程增益放大器控制寄存器 ADC\_PGACR

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
PGAGAIN[3:0]				PGACTL[3:0]			
<hr/>							
位	标记	位名	功能				
b7-b4	PGAGAIN[3:0]	增益设定	0 0 0 0: × 2.000				
			0 0 0 1: × 2.133				
			0 0 1 0: × 2.286				
			0 0 1 1: × 2.667				
			0 1 0 0: × 2.909				
			0 1 0 1: × 3.2				
			0 1 1 0: × 3.556				
			0 1 1 1: × 4.000				
			1 0 0 0: × 4.571				
			1 0 0 1: × 5.333				
			1 0 1 0: × 6.4				
			1 0 1 1: × 8				
			1 1 0 0: × 10.667				
			1 1 0 1: × 16				
			1 1 1 0: × 32				
注意：其他值禁止设定							
b3-b0	PGACTL[3:0]	放大器控制	0000: 放大器无效				
			1110: 放大器有效，信号按PGAGAIN[3:0]设定值放大				
注意：禁止设定上述以外的值。							

#### 注意：

- 单元 1 (ADC1) 支持 1 个 PGA 通道，参考表 17-1。寄存器 ADC1\_PGACR 对应控制 PGA1。
- 单元 2 (ADC2) 支持 1 个 PGA 通道，参考表 17-1。寄存器 ADC2\_PGACR 对应控制 PGA2。
- 单元 3 (ADC3) 不支持 PGA，无此寄存器。

### 17.4.22 可编程增益放大器参考地选择寄存器 ADC\_PGA\_VSSEN\_R

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	PGAVSSEN

位	标记	位名	功能	R/W
b7-b1	Reserved	—	读出时为0，写入时写0	R/W
b0	PGAVSSEN	PGA放大器参考地使能	0：禁止模拟地AVSS作为PGA参考地 1：使用模拟地AVSS作为PGA参考地	R/W

#### 注意：

- 单元 1 (ADC1) 支持 1 个 PGA 通道，参考表 17-1。寄存器 ADC1\_PGACR 对应控制 PGA1。
- 单元 2 (ADC2) 支持 1 个 PGA 通道，参考表 17-1。寄存器 ADC2\_PGACR 对应控制 PGA2。
- 单元 3 (ADC3) 不支持 PGA，无此寄存器。

## 17.5 使用注意事项

### 17.5.1 数据寄存器读取时注意事项

ADC 数据寄存器 ADC\_DR 请以半字单位访问。请不要以字节单位访问数据寄存器。

### 17.5.2 扫描完成中断处理注意事项

当对同一通道进行连续两次扫描转换时，从第一次转换完成中断处理到第二次转完成中断处理期间，如果 CPU 没有及时将第一次的转换数据读取，那么第二次的转换数据会将第一次的转换数据覆盖。

### 17.5.3 模块停止与低功耗设定的注意事项

通过设定寄存器 PWC\_FCG，可以设定 ADC 模块停止，降低功耗。ADC 初始状态为停止状态。需要 ADC 模块工作时请先设定 PWC\_FCG 寄存器相应位取消停止，并等待 1us 以后再启动 ADC 转换。

在设置模块停止前，请先确认 ADC 处于转换停止中，即 ADC\_STR.STRT 位为“0”。

在设置系统进入停止模式（STOP）前，请先将 ADC 设定为模块停止模式。

具体请参考低功耗说明章节。

### 17.5.4 ADC 转换模拟通道输入的引脚设定

当芯片引脚设定为 ADC 模拟通道输入时，请先禁用对应引脚的数字功能(PCRxy.DDIS)。参考【通用 IO (GPIO)】章节。

### 17.5.5 噪声控制

为防止浪涌等异常电压破坏模拟输入引脚，推荐使用[数据手册中电气特性章节](#)所示保护电路。

## 18 数模转换器 (DAC)

### 18.1 简介

HC32F472 搭载了 4 个数模转换器 DAC1~4。每个 DAC 均包含两个 12 位的 DAC 转换通道，可以独立转换也可以同步转换。模拟输出配有输出缓冲器，可以在没有外部运放时直接驱动外部负载。

DAC 主要特性如下：

- 8 个独立的 12 位 DAC 转换通道
- 转换数据支持左对齐或者右对齐模式
- 同一个 DAC 的两个转换通道可实现同步转换
- 独立的参考电压 VREFH 输入可提高转换精度
- 每个转换通道配有输出缓冲器
- 使用输出缓冲器时可选择 1.0x 或者 2.0x 输出增益
- 输出可用于电压比较器 (CMP) 的负端输入
- ADC 转换优先模式可减少对 ADC 转换的干扰
- 软件复位/看门狗复位发生时 DAC 输出保持

## 18.2 功能框图

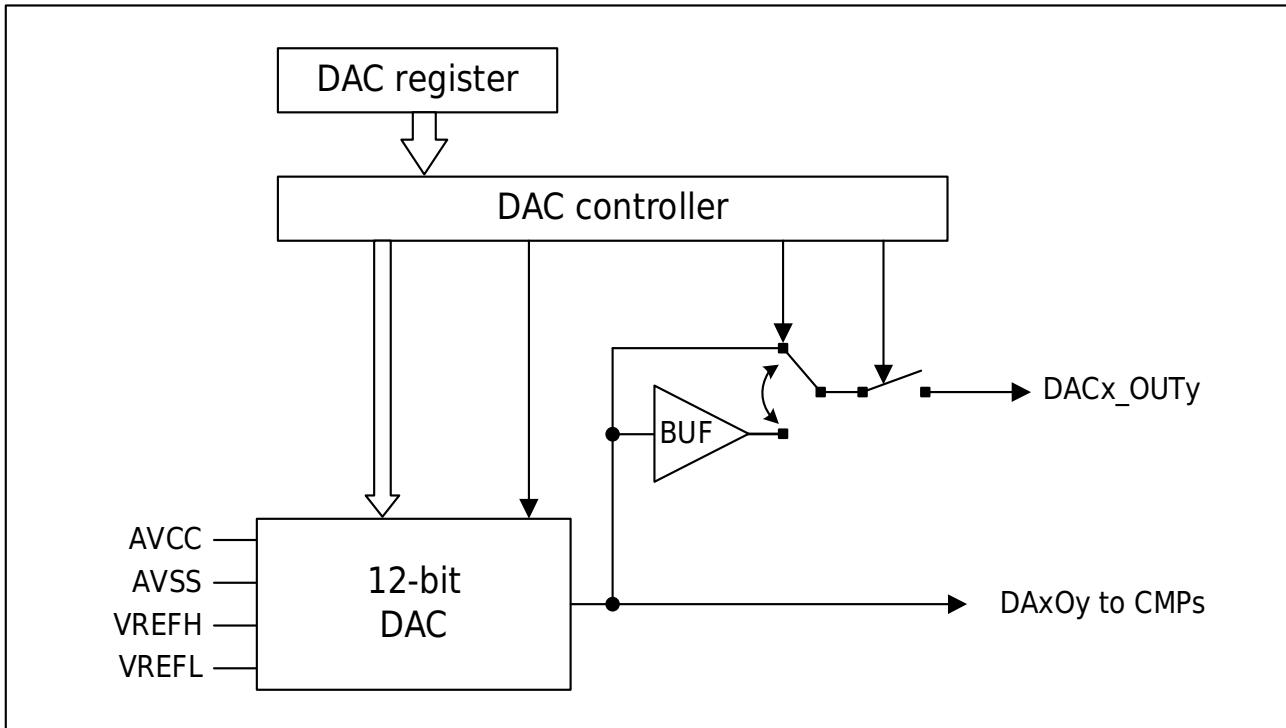


图 18-1 DAC 转换通道框图

注：x 表示 DAC 编号，取值范围为 1~4；y 表示 DAC 转换通道编号，取值范围为 1~2。以下同。

表 18-1 DAC 输入输出引脚

引脚名称	输入/输出	功能
AVCC	输入	模拟电源
AVSS	输入	模拟地
VREFH <sup>注1</sup>	输入	参考正电压
VREFL <sup>注1</sup>	输入	参考负电压（外部接模拟地）
DACx_OUTy	输出	模拟电压输出

注 1：部分封装没有 VREFH/VREFL 端子，内部分别与 AVCC/AVSS 相连。

## 18.3 功能说明

### 18.3.1 DAC 转换

每个转换通道都可以独立进行 DAC 转换。将 `DACx_DACR.DAyE` 设置成 1 时，对应通道的 DAC 转换开始并从 `DACx_OUTy` 端口输出模拟电压。如果仅将输出的模拟电压作为 CMP 的输入电压，可用 `DACx_DAOCR` 寄存器将 `DACx_OUTy` 输出关闭并将外部端口释放。

下面是一个使用 `DAC1_ch1` 进行单通道转换的例子，动作时序见图 18-2。

1. 设置 DAC 转换数据 (`DAC1_DADR1`) 和数据格式 (`DAC1_DACR.DPSEL`)。
2. 设置 `DAC1_DACR.DA1E` 为 1 后 DAC 转换开始。经过转换时间  $t_{DCONV}$  后，转换结果从端口 `DAC1_OUT1` 输出并保持到 `DAC1_DADR1` 被改写或者 `DAC1_DACR.DA1E` 被设置成 0。
3. `DAC1_DACR.DA1E` 为 1 时，改写 `DADR` 的值将触发一次新的 DAC 转换。同样，经过转换时间  $t_{DCONV}$  后，从端口 `DAC1_OUT1` 输出新的转换结果。
4. 设置 `DAC1_DACR.DA1E` 位为 0，关闭 `DAC1_ch1`，`DAC1_OUT1` 输出高阻态。

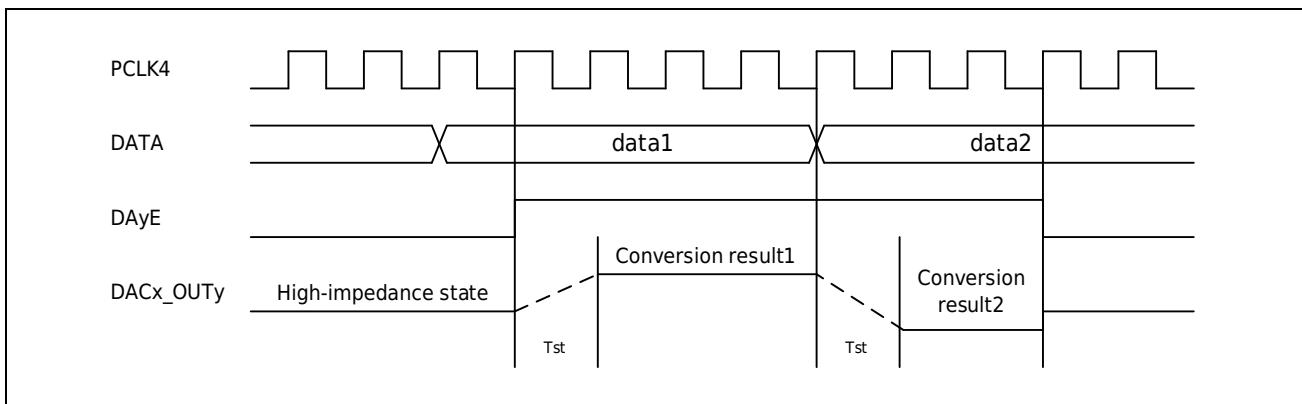


图 18-2 DAC 转换示意图

### 18.3.2 同步转换

通过转换数据的同步更新可以实现同一个 DAC 单元的两个转换通道的同步转换。

下面是一个使用 `DAC1` 进行两通道同步转换的例子。

1. 设置 DAC 转换数据 (`DAC1_DADR1,2`) 和数据格式 (`DAC1_DACR.DPSEL`)。
2. 设置 `DAC1_DACR.DAE` 为 1，两个通道的 DAC 转换同步开始。经过转换时间  $t_{DCONV}$  后，转换结果分别从端口 `DAC1_OUT1,2` 输出并一直保持到 `DADR` 被改写或者 `DAC1_DACR.DAE` 被设置成 0。
3. `DAC1_DACR.DAE` 为 1 的状态下，同时改写 `DAC1_DADR1,2` 的值将同步触发两个通道开始一次新的 DAC 转换。同样，经过转换时间  $t_{DCONV}$  后，从 `DAC1_OUT1,2` 输出新的转换结果。
4. 设置 `DAC1_DACR.DAE` 为 0，两个通道同步关闭，`DAC1_OUT1,2` 输出高阻态。

### 18.3.3 输出电压

每个 DAC 通道可以输出 DAxOy, DACx\_OUTy 两路电压。DAxOy 仅供内部使用, DACx\_OUTy 可以输出到外部引脚。

使用输出缓冲器且 DACx\_DACR.DAOGSy = 0, 或者禁止输出缓冲器时,

$$DAxOy = \text{ConversionData} / 4096 \times VREFH$$

$$DACx\_OUTy = \text{ConversionData} / 4096 \times VREFH$$

使用输出缓冲器且 DACx\_DACR.DAOGSy = 1 时,

$$DAxOy = \text{ConversionData} / 4096 \times VREFH$$

$$DACx\_OUTy = \text{ConversionData} / 2048 \times VREFH$$

DAxOy 在 DAC 通道使能时一直有输出, DACx\_OUTy 则在 DACx\_DAOCR.DAODISy = 1 时被禁止输出。

DAxOy 和 DACx\_OUTy 都可以用于 CMP 的模拟输入。

### 18.3.4 ADC 转换优先模式

DAC 开始 DAC 转换时模拟电源上可能出现短暂的浪涌电流, 这会对正在进行中的 ADC 转换产生干扰。

ADC 转换优先模式通过改变 DAC 转换数据的更新时序, 有效地避免了这种情况的发生。

设置 DACx\_DAADPCR.ADOPEN 为 1, DAC 进入 ADC 转换优先模式。这时, 如果在 ADC 转换期间改写 DACx\_DADRY 的值, DAC 转换并不会立即开始, 而是要等到 ADC 转换完成后才开始。也就是说, 从改写 DADR 到实际开始 DAC 转换最长需要等待一次 ADC 转换的时间。因此, 在此期间 DADR 的值与模拟输出值并不一致。但是, 如果改写 DACx\_DADRY 时 ADC 处于停止状态 (ADC\_STR.START 为 0), DAC 转换则会在 2 个 ADCLK 周期后开始。DACx\_DAADPCR.ADCSLn (n=0~2) 用于选择 ADC 转换优先的 ADC 通道。

下面以 DAC1\_ch1 为例, 说明 ADC 转换优先模式的设定步骤, 动作时序见图 18-3。

1. 确认 ADC 处于停止状态, 设置 DAC1\_DAADPCR.ADCSL。
  2. 确认 ADC 处于停止状态, 设置 DAC1\_DACR.DA1E 为 1。
  3. 将转换数据写入 DAC1\_DADR1。
- 转换数据 A 被写入 DAC1\_DADR1 时, ADC\_STR.START 为 0, ADC 处于停止状态, DAC 转换在 2 个 ADCLK 周期后开始。
  - 转换数据 B 被写入 DAC1\_DADR1 时, ADC\_STR.START 为 1, ADC 正在进行 ADC 转换, DAC 转换要到 ADC 转换完成后才会开始。而在 DAC 转换开始前转换数据 C 已被写入 DAC1\_DADR1, 因此最终被转换的是转换数据 C, 而不是转换数据 B。

为避免转换数据丢失，请先查看 DAC 转换状态寄存器 (DACx\_DAADPCR.DAySF)，确认当前数据转换完成后改写 DADR 的值。

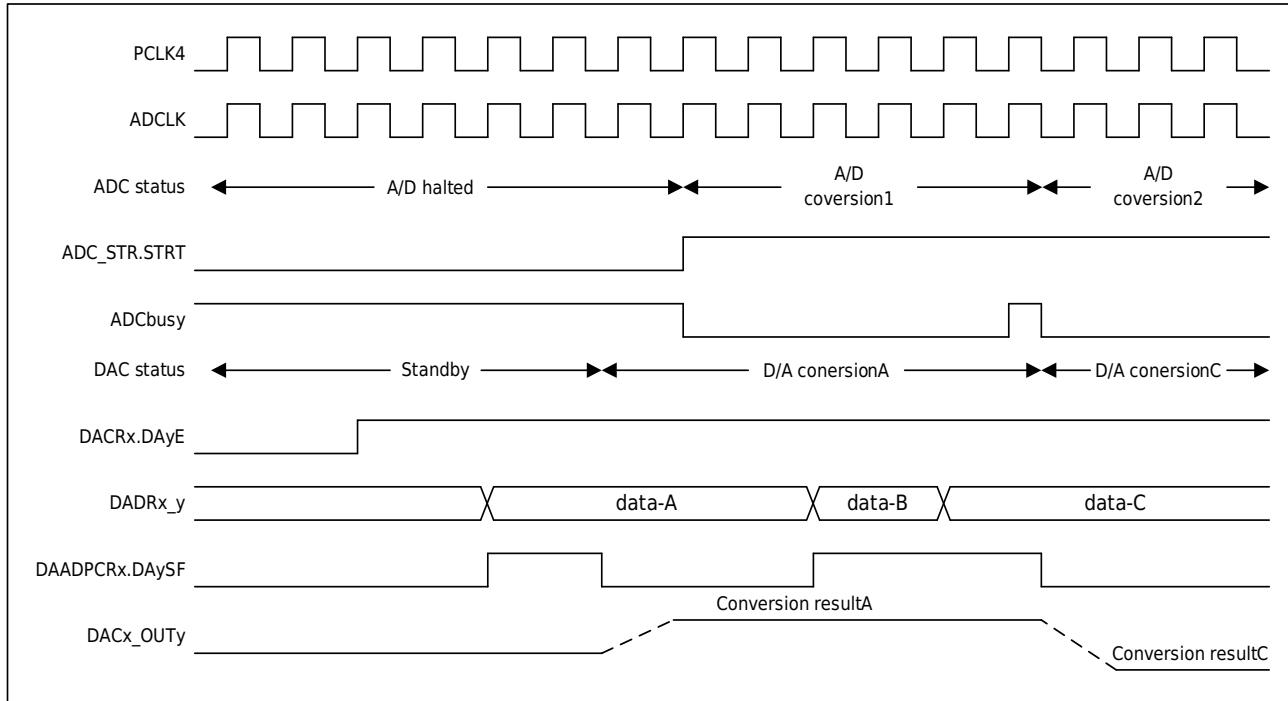


图 18-3 ADC 转换优先模式动作示意图

## 18.4 注意事项

### 18.4.1 模块停止功能

可以使用模块停止控制寄存器设定 DAC 有效或者停止。系统复位后 DAC 都是停止的，需要解除模块停止状态后才可以访问寄存器。详见低功耗模式章节。

### 18.4.2 停止低功耗模式

如果系统是在 DAC 转换时进入停止低功耗模式的，模拟输出将被保持，这时流过模拟电源的电流与 DAC 转换时相同。如果需要进一步减小停止模式下的功耗，请将 DACx\_DACR 的 DAyE 和 DAE 位全部设为 0。

### 18.4.3 掉电低功耗模式

如果系统是在 DAC 转换时进入掉电低功耗模式的，模拟输出将被置为高阻态。

### 18.4.4 使用输出缓冲器

使用输出缓冲器时请使用以下的初始化程序。

1. 给 DACx\_DADRx 写全 0。
2. 设置 DACx\_DACR.DAAMPy 为 1。
3. 设置 DACx\_DACR.DAE 或者 DAyE 为 1。
4. 等待 3us 后将变换数据写入 DACx\_DADRx。

关闭 DAC 可以使缓冲器停止工作。若要再次使用缓冲器，请重复步骤 1~4。

## 18.5 寄存器说明

表 18-2 DACx 寄存器一览表

DAC1 基准地址: 0x40041000

DAC2 基准地址: 0x40041400

DAC3 基准地址: 0x40041800

DAC4 基准地址: 0x40041C00

寄存器名	符号	偏移地址	位宽	复位值
DACx 数据寄存器1	DACx_DADR1	0x00	16	0x0000
DACx 数据寄存器2	DACx_DADR2	0x02	16	0x0000
DACx 控制寄存器	DACx_DACR	0x04	16	0x0000
DACx ADC转换优先控制寄存器	DACx_DAADPCR	0x06	16	0x0000
DACx 模拟输出控制寄存器	DACx DAOCR	0x1c	16	0x0000

表中 x=1~4。

所有寄存器在软件复位和看门狗复位发生时将保持。

### 18.5.1 DAC 数据寄存器 (DACx\_DADRx, y=1~2)

DACR.DPSEL=0 (数据右对齐) 时

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0			
-	-	-	-	DADR[11:0]														

位	标记	位名	功能	读写
b15~b12	Reserved	-	读时读出0, 写时请写0	R/W
b11~b0	DADR[11:0]	变换数据	变换数据设定值	R/W

DACR.DPSEL=1 (数据左对齐) 时

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0			
DADR[11:0]															-	-	-	-

位	标记	位名	功能	读写
b15~b4	DADR[11:0]	变换数据	变换数据设定值	R/W
b3~b0	Reserved	-	读时读出0, 写时请写0	R/W

DACx\_DADRx 用于存放 DAC 变换的数据，每个通道有一个。只要允许 DAC 转换，DACx\_DADRx 中的值就被转换成模拟电压并从模拟端口输出。对同一单元的 DACx\_DADR1 进行 32 位操作可实现双通道同步转换。

### 18.5.2 DAC 控制寄存器 (DACx\_DACR)

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	DAA MP2	DAA MP1	DP SEL	-	-	-	DAO GS2	DAO GS1	DA2 E	DA1 E	DAE

位	标记	位名	功能	读写
b15~b11	Reserved	-	读出时为“0”，写入时写“0”	R/W
b10	DAAMP2*注	通道2输出缓冲器使能	0: 禁止通道2输出缓冲器 1: 启动通道2输出缓冲器	R/W
b9	DAAMP1*注	通道1输出缓冲器使能	0: 禁止通道1输出缓冲器 1: 启动通道1输出缓冲器	R/W
b8	DPSEL	DADR数据对齐模式选择	0: DADR数据右对齐 1: DADR数据左对齐	R/W
b7~b5	Reserved	-	读出时为“0”，写入时写“0”	R/W
b4	DAOGS2*注	通道2电压输出增益选择	DAAMP2=1时此位设定有效 0: 1.0x增益 1: 2.0x增益	R/W
b3	DAOGS1*注	通道1电压输出增益选择	DAAMP1=1时此位设定有效 0: 1.0x增益 1: 2.0x增益	R/W
b2	DA2E	通道2使能	DAE=0时此位设定有效 0: 通道2禁止 1: 通道2使能	R/W
b1	DA1E	通道1使能	DAE=0时此位设定有效 0: 通道1禁止 1: 通道1使能	R/W
b0	DAE	总使能	0: 通道1, 2禁止 1: 通道1, 2使能	R/W

DACR 用于控制 DAC 转换和 DAC 输出。详见表 18-3。

通道禁止时不管 DAAMP1 是什么值，该通道的模拟输出也被禁止，端口呈现高阻态。

ADC 转换优先模式时，请在 ADC\_STR STRT=0 的状态下设定 DA1E, DA2E 和 DAE 位。同时为了避免 ADC 被意外启动，请将 ADC 的触发选择设定为软件触发。

注：HC32F472JCTI/HC32F472JETI/HC32F472PCTI/HC32F472PETI 不支持此功能，请将此寄存器位设定为 “0”。

### 18.5.3 DAC 模拟输出控制寄存器 (DACx\_DAOCR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DAO DIS2	DAO DIS1	-	-	-	-	-	-	-	-	-	-	-	-	-	-

位	标记	位名	功能	读写
b15	DAODIS2	DACx_OUT2端口 输出控制	1: 禁止DACx_OUT2端口输出模拟电压 0: 允许DACx_OUT2端口输出模拟电压	R/W
b14	DAODIS1	DACx_OUT1端口 输出控制	1: 禁止DACx_OUT1端口输出模拟电压 0: 允许DACx_OUT1端口输出模拟电压	R/W
b13~b0	Reserved	-	读出时为“0”，写入时写“0”	R/W

表 18-3 DAC 转换和模拟输出控制

DAE	DA1E/ DA2E	DAAMP1/ DAAMP2	DAOGS1/ DAOGS2	DAODIS1/ DAODIS2	DAC转换	输出缓冲器	DAC_OUT 输出	DAO 输出
0	0	任意	任意	任意	停止	停止	高阻态	高阻态
	1	0	任意	0	启动	停止	普通输出	普通输出
				1	启动	停止	高阻态	普通输出
	1	0	0	0	启动	启动	1.0x放大输出	普通输出
				1	禁止设定	禁止设定	禁止设定	禁止设定
				1	启动	启动	2.0x放大输出	普通输出
				1	禁止设定	禁止设定	禁止设定	禁止设定
	1	0	任意	0	启动	停止	普通输出	普通输出
				1	启动	停止	高阻态	普通输出
	1	0	0	0	启动	启动	1.0x放大输出	普通输出
				1	禁止设定	禁止设定	禁止设定	禁止设定
	1	0	任意	0	启动	停止	普通输出	普通输出
				1	启动	停止	高阻态	普通输出
	1	0	0	0	启动	启动	1.0x放大输出	普通输出
				1	禁止设定	禁止设定	禁止设定	禁止设定
	1	0	任意	0	启动	启动	2.0x放大输出	普通输出
				1	禁止设定	禁止设定	禁止设定	禁止设定

### 18.5.4 DAC ADC 转换优先控制寄存器 (DACx\_DAADPCR)

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADP EN	-	-	-	-	-	DA2 SF	DA1 SF	-	-	-	-	-	ADC SL3	ADC SL2	ADC SL1

位	标记	位名	功能	读写
b15	ADPEN	ADC转换优先使能	0: ADC转换优先模式无效 1: ADC转换优先模式有效	R/W
b14~b10	Reserved	-	读出时为“0”，写入时写“0”	R/W
b9	DA2SF	通道2数据更新状态	本标志位只在ADC转换优先模式时变化 0: 通道2数据更新完 1: 通道2数据更新中	R/W
b8	DA1SF	通道1数据更新状态	本标志位只在ADC转换优先模式时变化 0: 通道1数据更新完 1: 通道1数据更新中	R/W
b7~b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2	ADCSL3	ADC_3优先选择	0: 不选择ADC_3作为ADC转换优先对象 1: 选择ADC_3作为ADC转换优先对象	R/W
b1	ADCSL2	ADC_2优先选择	0: 不选择ADC_2作为ADC转换优先对象 1: 选择ADC_2作为ADC转换优先对象	R/W
b0	ADCSL1	ADC_1优先选择	0: 不选择ADC_1作为ADC转换优先对象 1: 选择ADC_1作为ADC转换优先对象	R/W

DACx\_DAADPCR 用于 ADC 转换优先模式的控制。

ADPEN 设定成 1 时 ADC 转换优先模式有效，请在设定 ADPEN 的同时选择作为优先对象的 ADC 单元。

DA1SF、DA2SF 为 DA 转换状态标志位，反映当前该通道数据寄存器中的数据是否已被转换。

## 19 温度传感器 (OTS)

### 19.1 简介

温度传感器 (On-chip Temperature Sensor, 以下简称 OTS) 可以获取芯片内部的温度，以支持系统的可靠性操作。OTS 提供了一组与温度相关的数字量，通过计算可以得到温度值。不使用时可以通过模块停止功能将其关闭以减少系统功耗。

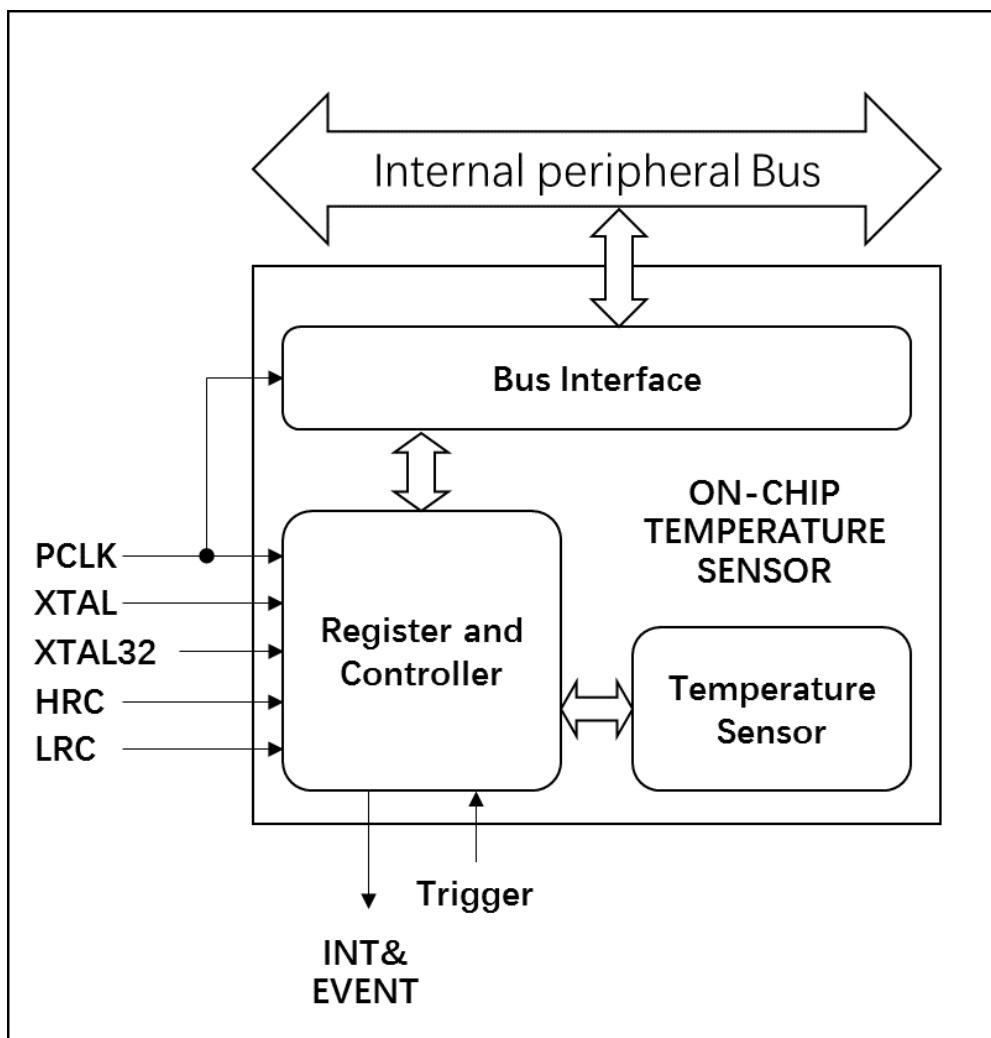


图 19-1 OTS 功能框图

## 19.2 使用说明

使用 OTS 获取芯片内部温度之前，先要对其进行定标，且测量精度与定标方式有关，请参考下述定标实验的说明进行定标。使用时请先关闭模块停止功能并启动内部低速时钟 LRC,同时请根据使用情况选择启动内部高速时钟 HRC、外部高速时钟 XTAL 以及外部低速时钟 XTAL32。将寄存器 OTS\_CTL.OTSST 设置成“1”后测温开始，测温完成后 OTSST 位会自动清零。确认 OTSST 为“0”后，读取寄存器 OTS\_DR1,2 以及 OTS\_ECR 中的温度参数，并利用以下计算公式求得温度值。

$$T = K \times (1/D1 - 1/D2) \times E_{hrc} + M$$

### 【参数说明】

T：温度（°C）

K：温度斜率（通过定标实验确定）

D1：温度数据 1（从寄存器 OTS\_DR1 读取）

D2：温度数据 2（从寄存器 OTS\_DR2 读取）

E<sub>hrc</sub>：HRC 频率误差补偿数据（从寄存器 OTS\_ECR 读取）

M：温度偏移量（通过定标实验确定）

寄存器 OTS\_CTL.OTSCK 用来选择测温时钟。选择 HRC 动作时，其频率误差可能会影响到最终计算出的温度的精度。为了消除这一误差，请在测温前启动 XTAL32 并在定标和计算温度时使用 E<sub>hrc</sub>。选择 XTAL 动作时请忽略 E<sub>hrc</sub> 并将其示为常量 1。

### 【定标实验】

请在两个确定的温度下进行定标实验，计算 K 和 M。

$$K = (T2 - T1) / (A2 - A1)$$

$$M = T1 - K \times A1 = T2 - K \times A2$$

T1：实验温度 1

T2：实验温度 2

$$A1: (1/D1_{T1} - 1/D2_{T1}) \times E_{hrcT1}$$

D1<sub>T1</sub>, D2<sub>T1</sub>, E<sub>hrcT1</sub> 分别是在温度 T1 下测到的 D1, D2, E<sub>hrc</sub>；

$$A2: (1/D1_{T2} - 1/D2_{T2}) \times E_{hrcT2}$$

D1<sub>T2</sub>, D2<sub>T2</sub>, E<sub>hrcT2</sub> 分别是在温度 T2 下测到的 D1, D2, E<sub>hrc</sub>；

作为参考，芯片内部预置了三组温度数据，可任意选择两个温度点的数据进行定标。由于测试环境温度会有偏差，预置数据仅供参考使用。

表 19-1 OTS 预置温度数据

测试温度	读取方式		测试条件
	温度数据 1 (D1)	温度数据 2 (D2)	
T <sub>a</sub> = -40°C	OTS_PDR3[15:0]	OTS_PDR3[31:16]	XTAL=8MHz
T <sub>a</sub> = 25°C	OTS_PDR1[15:0]	OTS_PDR1[31:16]	XTAL=8MHz
T <sub>a</sub> = 125°C	OTS_PDR2[15:0]	OTS_PDR2[31:16]	XTAL=8MHz

E<sub>hrc</sub> 是为了消除 HRC 本身的误差对测量温度影响的一个校准值，可以从寄存器 OTS\_ECR 中读取，使用方法如下。

表 19-2 Ehrc 使用及设定方法

动作时钟	定标实验 (计算 K,M)		温度测量
(OTSCK)	使用预置温度数据	重新获取温度数据	(计算 T)
XTAL	E <sub>hrc</sub> =1	E <sub>hrc</sub> =1	E <sub>hrc</sub> =1
HRC	E <sub>hrc</sub> =f <sub>hrc</sub> (MHz)/0.032768	E <sub>hrc</sub> =OTS_ECR[15:0]	E <sub>hrc</sub> =OTS_ECR[15:0]

f<sub>hrc</sub> 为 HRC 的标准频率，即 16MHz 或者 20MHz。

寄存器 OTS\_CTL.TSSTP 用来选择测温完成后是否关闭模拟温度传感器。TSSTP 初始值为“0”，表示在一次测温完成后将模拟温度传感器置于开启状态，这样在下一次测温时将自动跳过模拟温度传感器由关闭到开启时的稳定时间。如需在每次测温后关闭模拟温度传感器，请将 TSSTP 设置成“1”。

可以通过其他的外设事件触发开始测温，使用时请将触发源的触发目标设置成 OTS。测温完成时也可以产生触发其他的外设启动的事件，使用时请设定寄存器 OTS\_TRGSEL 选择触发目标。使用测温完成中断时请将寄存器 OTS\_CTL.OTSIE 设置为“1”。

### 19.3 寄存器说明

表 19-3 OTS 寄存器一览表

基准地址：0x4004A800

寄存器名	符号	偏移地址	位宽	复位值
OTS控制寄存器	OTS_CTL	0x00	16	0x0000
OTS数据寄存器1	OTS_DR1	0x02	16	0x0000
OTS数据寄存器2	OTS_DR2	0x04	16	0x0000
OTS误差补偿寄存器	OTS_ECR	0x06	16	0x0000

基准地址：0x40010600

寄存器名	符号	偏移地址	位宽	复位值
OTS预置数据寄存器1	OTS_PDR1	0xe4	32	不定值
OTS预置数据寄存器2	OTS_PDR2	0xf4	32	不定值
OTS预置数据寄存器3	OTS_PDR3	0xf8	32	不定值

**19.3.1 OTS 控制寄存器 (OTS\_CTL)**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	TSST_P	OTS_IE	OTS_CK	OTS_ST

复位值: 0x0000

位	标记	位名	功能	读写
b15~b4	Reserved	-	读时读出0, 写时请写0	R/W
b3	TSSTP	关闭模拟温度传感器	选择测温结束后是否自动关闭模拟温度传感器 0: 不关闭模拟温度传感器 1: 关闭模拟温度传感器	R/W
b2	OTSIE	中断使能位	0: 禁止测温结束中断请求 1: 允许测温结束中断请求	R/W
b1	OTSCK	时钟选择位	0: 选择外部高速时钟 (XTAL) 动作 1: 选择内部高速时钟 (HRC) 动作	R/W
b0	OTSST	测温开始位	0: 停止测温 1: 开始测温 置“1”条件: (1) 软件置“1” (2) 硬件触发置“1” 清“0”条件: (1) 软件清“0” (2) 测温结束后自动清“0”	R/W

**19.3.2 OTS 数据寄存器 1 (OTS\_DR1)**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TSDC[15:0]															

复位值: 0x0000

位	标记	位名	功能	读写
b15~b0	TSDC[15:0]	温度数据D1	温度数据D1 测温完成后自动更新。请确认OTS_CTL.OTSST为“0”后再读取。	R

### 19.3.3 OTS 数据寄存器 2 (OTS\_DR2)

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TSDC[15:0]															

复位值: 0x0000

位	标记	位名	功能	读写
b15~b0	TSDC[15:0]	温度数据D2	温度数据D2 测温完成后自动更新。请确认OTS_CTL.OTSST为“0”后再读取。	R

### 19.3.4 OTS 误差补偿寄存器 (OTS\_ECR)

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TSEC[15:0]															

复位值: 0x0000

位	标记	位名	功能	读写
b15~b0	TSEC[15:0]	误差系数	误差系数Ehrc 测温完成后自动更新。请确认OTS_CTL.OTSST为“0”后再读取。	R

### 19.3.5 OTS 预置温度数据寄存器 (OTS\_PDR1,2,3)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TSPD2[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TSPD1[15:0]															

复位值: 不定值

位	标记	位名	功能	读写
b31~b16	TSPD2[15:0]	预置温度数据D2	预置温度数据，供定标时使用。 该数据为8MHz时测得，使用前请先与用由OTSK选择的动作时钟的实际频率进行换算。 $D2 = TSPD2 \times f_{OTSK} / 8$ , $f_{OTSK}$ 为实际动作时钟的频率，单位MHz。	R
b15~b0	TSPD1[15:0]	预置温度数据D1	预置温度数据，供定标时使用。 该数据为8MHz时测得，使用前请先与用由OTSK选择的动作时钟的实际频率进行换算。 $D1 = TSPD1 \times f_{OTSK} / 8$ , $f_{OTSK}$ 为实际动作时钟的频率，单位MHz。	R

## 20 高级控制定时器 (Timer6)

### 20.1 简介

高级控制定时器 6 (Timer6) 是一个 16 位计数宽度的高性能定时器，能在各种复杂应用场景中提供丰富、灵活的搭配组合和各种中断、事件、PWM 输出。该定时器支持锯齿波和三角波两种波形模式，可生成各种 PWM 波形（单边对齐独立 PWM、双边对称独立 PWM、双边对称互补 PWM、双边非对称 PWM 等）；单元间可实现软件同步和硬件同步（同步启动、停止、清零、刷新等）；各基准值寄存器支持缓存功能（单级缓存和双级缓存）；支持脉宽测量和周期测量；支持 2 相正交编码和 3 相正交编码；支持 EMB 控制。本系列产品中搭载 10 个单元的 Timer6。

### 20.2 基本框图

Timer6 基本的功能及特性如表 20-1 所示。

表 20-1 Timer6 的基本功能及特性

波形模式	锯齿波（递加、递减计数）、三角波（递加递减计数）
基本功能	• 捕获输入
	• 软件同步
	• 硬件同步
	• 缓存功能
	• 脉宽测量
	• 周期测量
	• 正交编码计数
	• 通用 PWM 输出
	• EMB 控制
中断输出	计数比较匹配中断
	计数周期匹配中断
	死区时间错误中断
事件输出	计数比较匹配事件
	计数周期匹配事件

Timer6 的基本框图如图 20-1 所示。框图中所示“ $<t>$ ”表示单元编号，即“ $<t>$ ”为 1~10，本章节后文提到“ $<t>$ ”时均指单元编号，不再赘述。

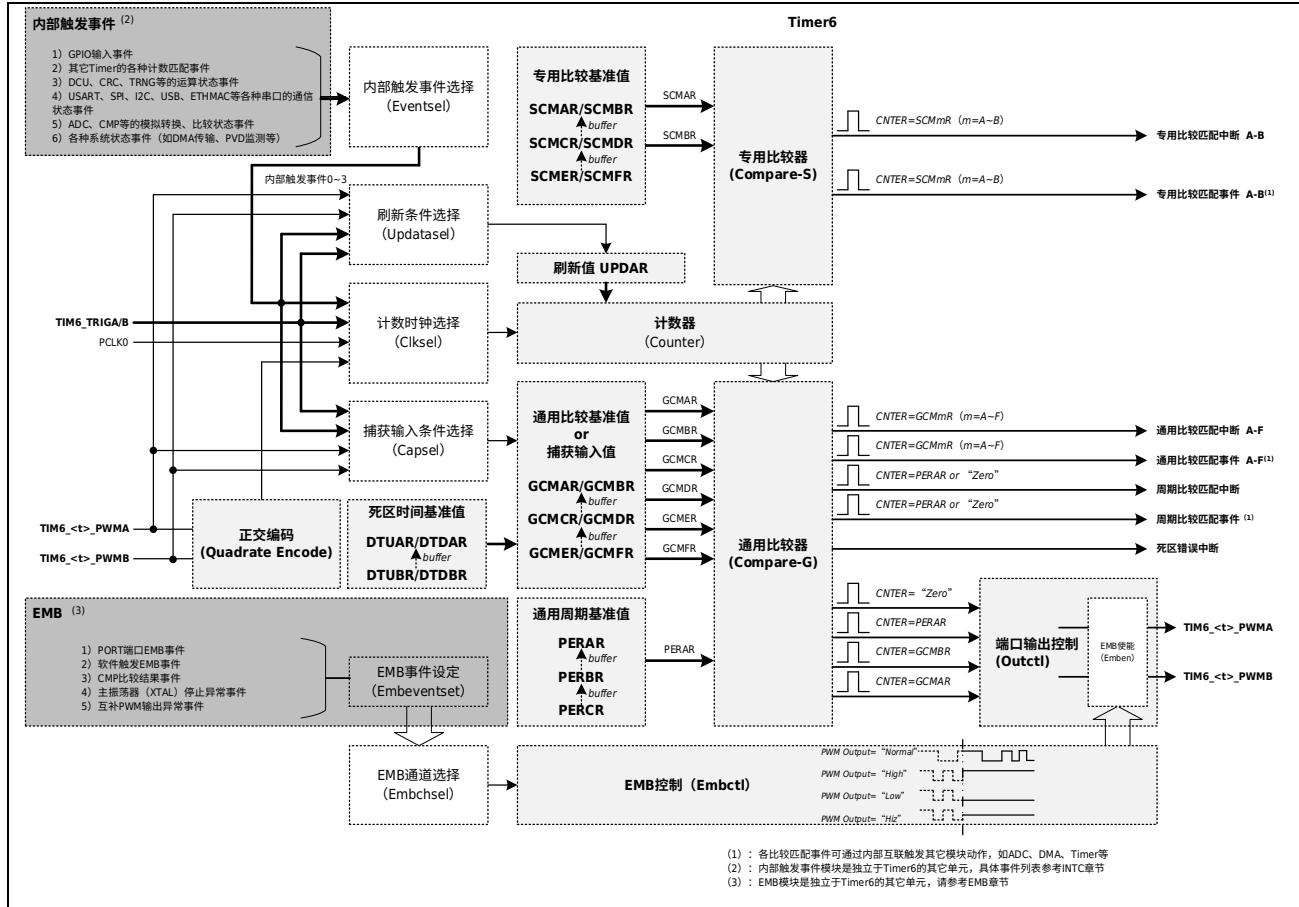


图 20-1 Timer6 基本框图

表 20-2 所示为 Timer6 的输入输出端口列表。

表 20-2 Timer6 端口列表

端口名	方向	功能
TIM6_<t>_PWMA	in or out	1) 正交编码计数时钟输入端口或捕获输入端口或比较输出端口 2) 硬件启动、停止、清零、刷新条件输入端口
TIM6_<t>_PWMB		
TIM6_TRIGA	in	1) 硬件计数时钟输入端口或捕获输入端口 2) 硬件启动、停止、清零、刷新条件输入端口
TIM6_TRIGB		
TIM6_TRIGC		
TIM6_TRIGD		

## 20.3 功能说明

### 20.3.1 波形模式

Timer6 有 2 种基本计数波形模式，锯齿波模式和三角波模式。两种波形模式的基本波形如图 20-2、图 20-3 所示。

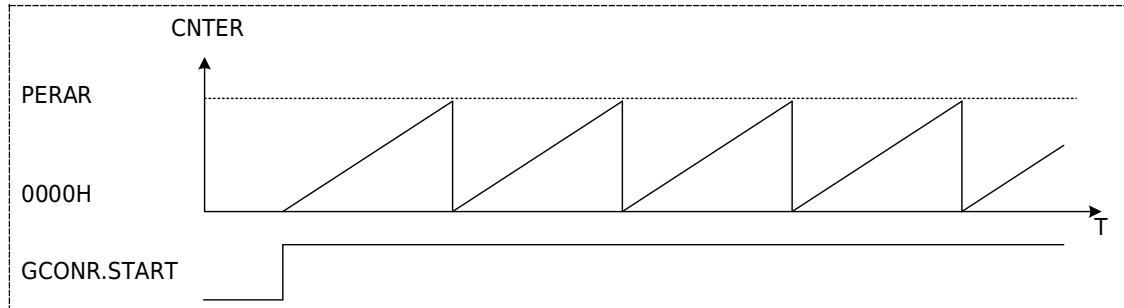


图 20-2 锯齿波波形（递加计数）

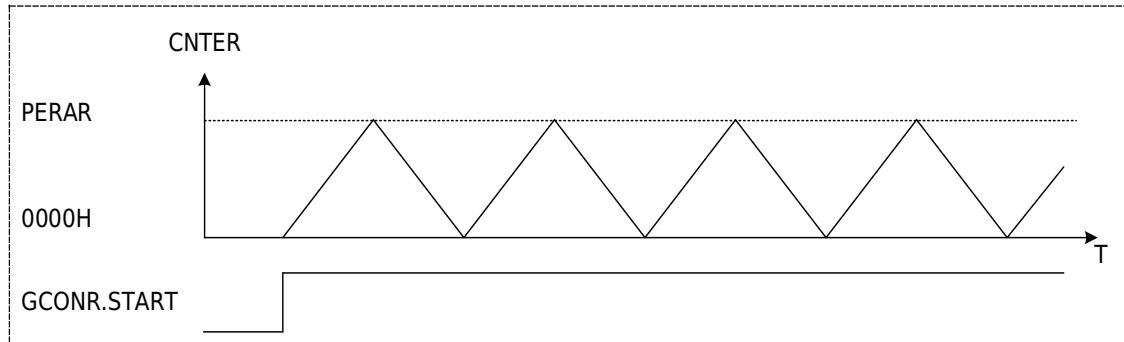


图 20-3 三角波波形

### 20.3.2 时钟源选择

Timer6 的计数时钟可以有以下几种选择：

- a) PCLK0 及 PCLK0 的 2、4、8、16、32、64、128、256、512、1024 分频 (GCONR.CKDIV[3:0] 设定)
- b) 内部触发事件输入 0~3 (HCUPR[11:8]或 HCDOR[11:8]设定)
- c) TIM6\_TRIGA~D 的端口输入 (HCUPR[23:16]或 HCDOR[23:16]设定)
- d) TIM6\_<t>\_PWMA 和 TIM6\_<t>\_PWMB 的正交编码关系输入 (HCUPR[7:0]或 HCDOR[7:0]设定)

计数时钟源选择 a 时为软件计数模式，计数时钟源选择 b、c、d 时为硬件计数模式。上述描述可以看到，b、c、d 时钟互相独立，可分别设定有效或无效，并且当选择 b、c、d 时钟时，a 时钟自动无效。

### 20.3.3 计数方向

Timer6 的定时器计数方向可通过软件方式改变。不同波形模式时，改变计数方向的方法略有不同。

#### 20.3.3.1 锯齿波计数方向

锯齿波模式时，计数方向可在定时器计数中或停止时设定。

在递加计数中时，设定 GCONR.DIR=0 (递减计数)，则定时器计数到上溢后变为递减计数模式；在递减计数中时，设定 GCONR.DIR=1 (递加计数)，则定时器计数到下溢后变为递加计数模式。

在计数停止时，设定 GCONR.DIR 位，计数开始后直至上溢或下溢时，GCONR.DIR 的设定反映到计数中。

#### 20.3.3.2 三角波计数方向

三角波模式时，设定计数方向无效，计数至计数峰点或计数谷点时，自动改变计数方向。

### 20.3.4 比较输出

每个单元的 Timer6 有 2 个比较输出端口 (TIM6\_<t>\_PWMA、TIM6\_<t>\_PWMB)，可在计数值与比较基准值比较匹配时输出指定的电平。GCMAR、GCMBR 寄存器对应了 TIM6\_<t>\_PWMA、TIM6\_<t>\_PWMB 的计数比较基准值。当定时器的计数值和 GCMA<B>R 相等时，TIM6\_<t>\_PWMA 或 TIM6\_<t>\_PWMB 端口输出指定电平。

TIM6\_<t>\_PWMA 端口的计数开始时的电平、计数停止时的电平、计数溢出时的电平、计数比较匹配时的电平等，可通过端口控制寄存器 (PCNAR) 的 PCNAR.STACA、PCNAR.STPCA、PCNAR.OVFCA、PCNAR.UDFCA、PCNAR.CMA<B>U<D>A、PCNAR.FORCA 位设定。

TIM6\_<t>\_PWMB 端口的计数开始时的电平、计数停止时的电平、计数溢出时的电平、计数比较匹配时的电平等，可通过端口控制寄存器 (PCNBR) 的 PCNBR.STACB、PCNBR.STPCB、PCNBR.OVFCB、PCNBR.UDFCB、PCNBR.CMA<B>U<D>B、PCNBR.FORCB 位设定。图 20-4 为比较输出的动作例。

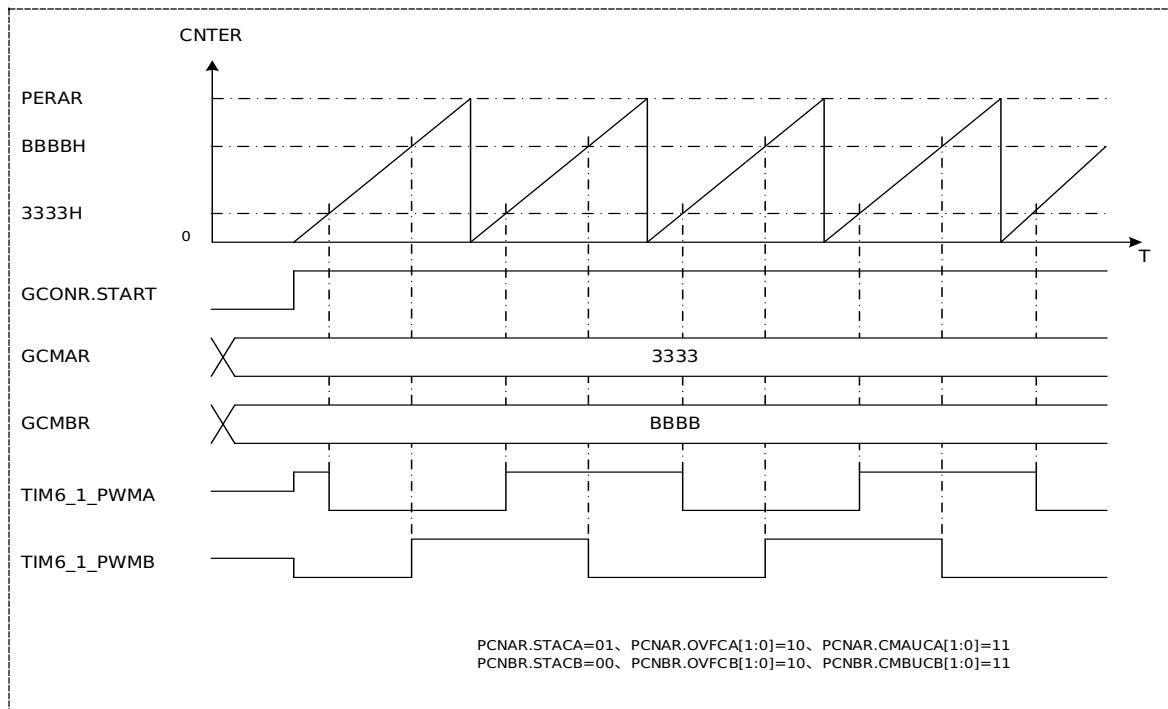


图 20-4 比较输出动作

### 20.3.5 捕获输入

每个单元都具有捕获输入功能，具备 2 组捕获输入寄存器（GCMAR、GCMBR），用于保存捕获到的当前计数值。设定端口控制寄存器（PCNA<B>R）的 CAPMDA<B>位为 1，捕获输入功能变为有效。当设定了对应的捕获输入条件且该条件有效时，当前的计数值就被保存到相应的捕获寄存器（GCMAR、GCMBR）中，同时，状态标志寄存器（STFLR）的 CMA<B>F 位为 1。

各单元的每组捕获输入的条件可以是内部触发事件输入 0~3、TIM6\_TRIGA~D 的端口输入、TIM6\_<t>\_PWMA 或 TIM6\_<t>\_PWMB 的端口输入等，具体的条件选择可通过硬件捕获事件选择寄存器（HCPAR、HCPBR）来设定。图 20-5 为捕获输入的动作例。

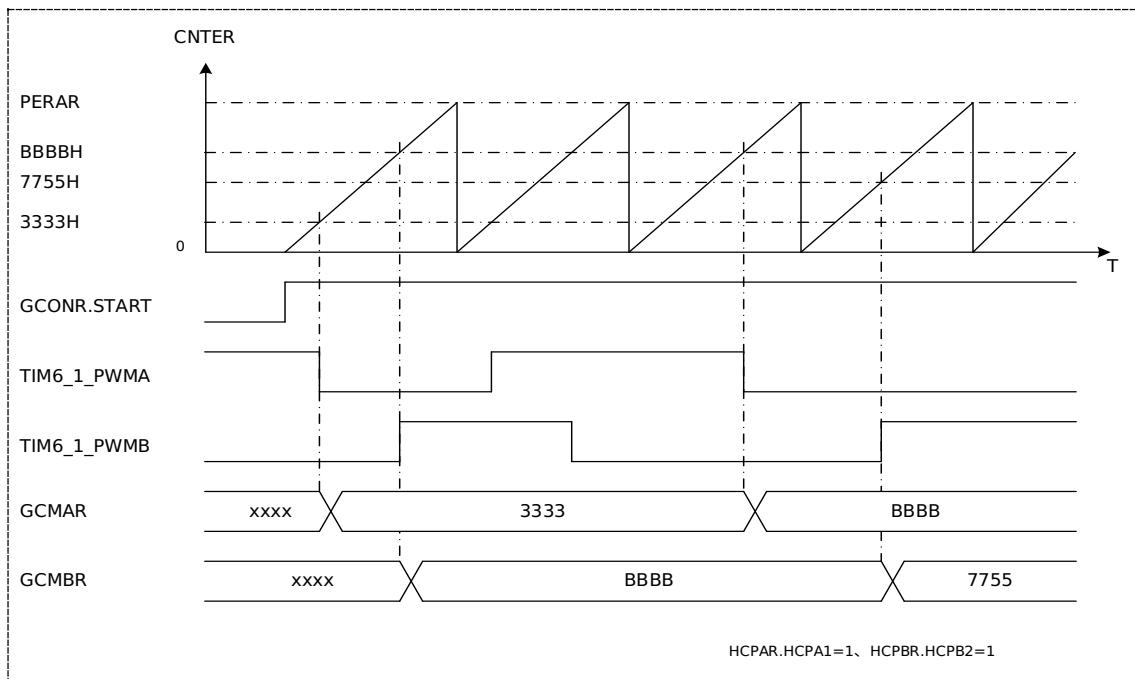
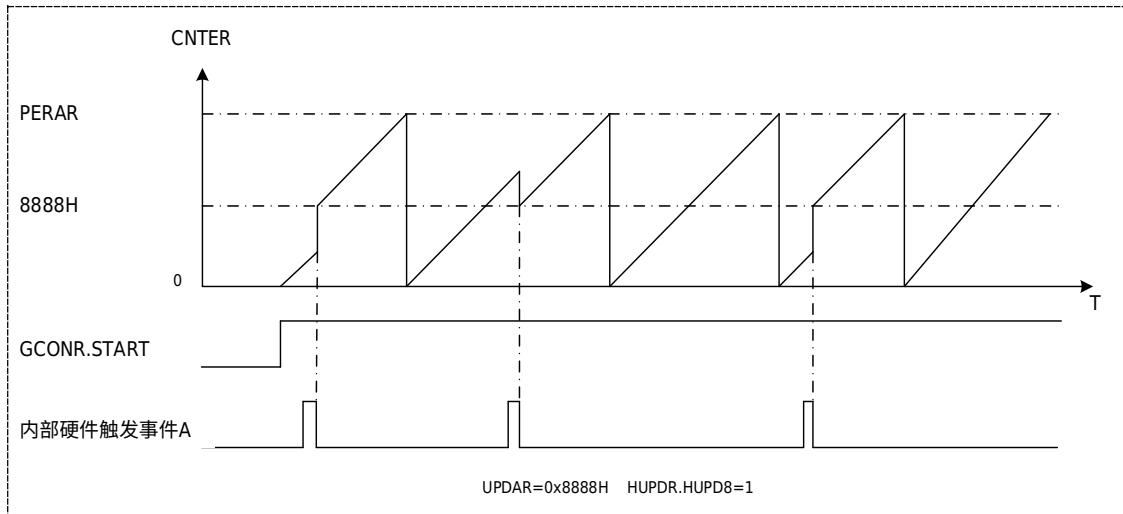


图 20-5 捕获输入动作

### 20.3.6 计数器刷新

Timer6 具有一个通用刷新值寄存器 (UPDAR)，可以在计数停止或计数中时，对通用计数值寄存器 (COUNTER) 的计数值实时进行刷新。

计数值的刷新条件由硬件刷新事件选择寄存器 (HUPDR) 或软件同步刷新控制寄存器 (SUPDR) 的对应位设定。当设定的刷新事件有效时，通用计数值寄存器 (COUNTER) 的值更新为通用刷新值寄存器 (UPDAR) 中指定的值。图 20-6 是锯齿波模式在计数中的硬件刷新动作例。



## 20.3.7 软件同步

### 20.3.7.1 软件同步启动

各单元可通过设定软件同步启动控制寄存器（SSTAR）的相关位，实现目标单元计数器（CNTER）的同步启动。

### 20.3.7.2 软件同步停止

各单元可通过设定软件同步停止控制寄存器（SSTPR）的相关位，实现目标单元计数器（CNTER）的同步停止。

### 20.3.7.3 软件同步清零

各单元可通过设定软件同步清零控制寄存器（SCLRR）的相关位，实现目标单元计数器（CNTER）的同步清零。

### 20.3.7.4 软件同步刷新

各单元可通过设定软件同步刷新控制寄存器（SUPDR）的相关位，实现目标单元计数器（CNTER）的同步刷新。

如图 20-7 所示、若设定 SSTAR.SSTA1=SSTAR.SSTA2=SSTAR.SSTA3=1，即可实现单元 1~3 的软件同步启动、设定 SSTPR.SSTP1=SSTPR.SSTP2=SSTPR.SSTP3=1，即可实现单元 1~3 的软件同步停止。

软件同步动作相关寄存器（SSTAR、SSTPR、SCLRR、SUPDR）是一组独立于单元外、各个单元间共用的寄存器，这组寄存器的各个位只在写 1 时有效，写 0 无效。在读取 SSTAR 寄存器时，会读出各个单元的定时器状态（计数停止或计数中），在读取 SSTPR、SCLRR 或 SUPDR 时，会读出零。

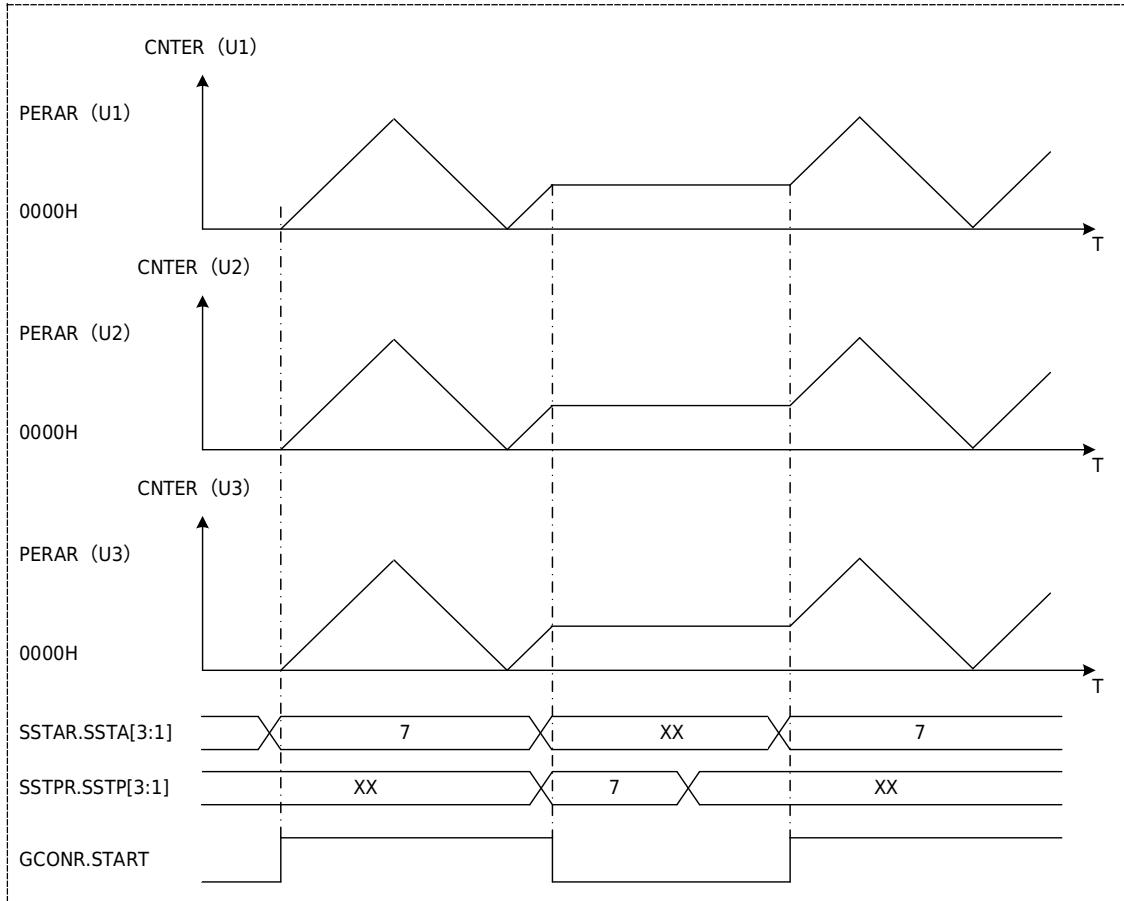


图 20-7 软件同步动作

### 20.3.8 硬件同步

每个单元除独立拥有 2 个通用输入端口 (TIM6\_<t>\_PWMA、TIM6\_<t>\_PWMB) 外，还共同拥有 4 个通用外部触发输入端口 (TIM6\_TRIGA、TIM6\_TRIGB、TIM6\_TRIGC、TIM6\_TRIGD) 及 4 个内部触发事件输入条件，可实现单元间的硬件同步动作。

内部触发事件的事件源可通过硬件触发事件选择寄存器 (HTSSR0~3) 中对应的编号设定来选择，具体的事件对应关系请参考 INTC 章节。使用内部触发功能时，需要先将功能时钟控制寄存器 0 (PWC\_FCG0) 的外围电路触发功能使能位置 0。

#### 20.3.8.1 硬件同步启动

各单元均可选择用硬件方式启动定时器，选择相同硬件启动条件的单元即可在启动条件有效时实现同步启动。具体的硬件启动条件由硬件启动事件选择寄存器 (HSTAR) 的设定来决定。

#### 20.3.8.2 硬件同步停止

各单元均可选择用硬件方式停止定时器，选择相同硬件停止条件的单元即可在停止条件有效时实现同步停止。具体的硬件停止条件由硬件停止事件选择寄存器 (HSTPR) 的设定来决定。

### 20.3.8.3 硬件同步清零

各单元均可选择用硬件方式清零定时器，选择相同硬件清零条件的单元即可在清零条件有效时实现同步清零。具体的硬件清零条件由硬件清零事件选择寄存器（HCLRR）的设定来决定。

### 20.3.8.4 硬件同步刷新

各单元均可选择用硬件方式刷新定时器，选择相同硬件刷新条件的单元即可在刷新条件有效时实现同步刷新。具体的硬件刷新条件由硬件刷新事件选择寄存器（HUPDR）的设定来决定。

### 20.3.8.5 硬件同步捕获输入

各单元均可选择用硬件方式实现捕获输入功能，选择相同捕获输入功能条件的单元即可在捕获输入功能条件有效时实现同步捕获输入。具体的硬件捕获输入功能条件由硬件捕获事件选择寄存器（HCPAR、HCPBR）的设定来决定。

### 20.3.8.6 硬件同步计数

各单元均可选择用硬件输入作为 CLOCK 进行计数，选择相同硬件计数条件的单元即可在硬件计数时钟有效时实现同步计数。具体的硬件计数条件由硬件递加事件选择寄存器（HCUPR）和硬件递减事件选择寄存器（HCDOR）的设定来决定。

选择硬件同步计数功能时，只是选择了外部输入时钟源，不影响定时器的启动、停止、清零动作。定时器的启动、停止、清零等还需要单独设定。

图 20-8 所示是单元 1~3 的硬件同步动作例。

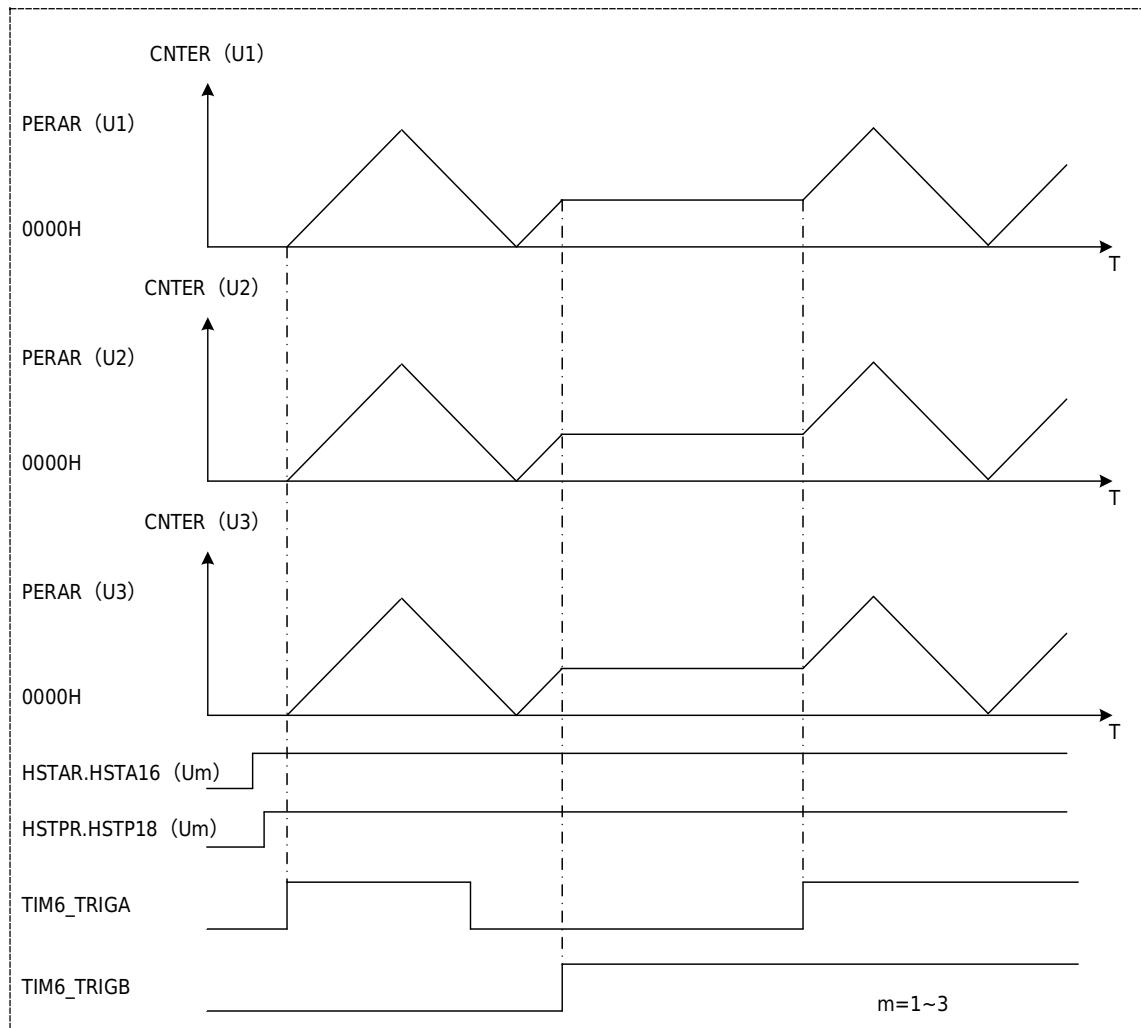


图 20-8 硬件同步动作

### 20.3.9 脉宽测量

在使用 TIM6\_<t>\_TRIGA~D 端口的硬件触发相关功能（参考硬件同步章节）时，每个单元都可以实现 2 路独立的脉宽测量功能。

例如，将计数器的硬件启动条件设为 TIM6\_<t>\_TRIGA 的上升沿，硬件清零条件、停止条件和 GCMAR 寄存器的捕获输入条件均设为 TIM6\_<t>\_TRIGA 的下降沿，就可以实现连续的脉冲宽度测量。对应动作如图 20-9 所示。

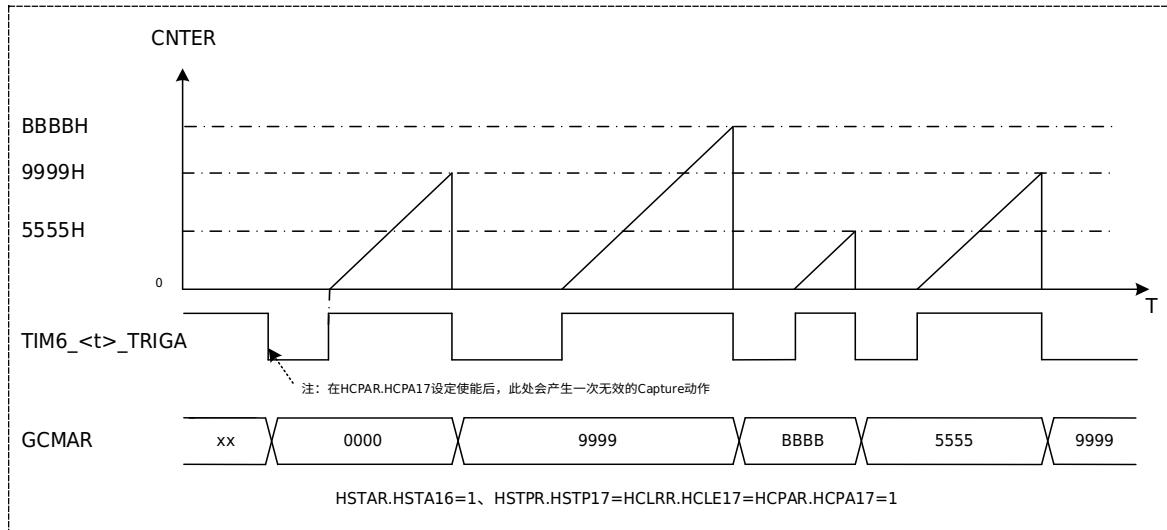


图 20-9 脉宽测量

### 20.3.10 周期测量

在使用 TIM6\_<t>\_TRIGA~D 端口的硬件触发相关功能（参考硬件同步章节）时，每个单元都可以实现 2 路独立的周期测量功能。

例如，将计数器的硬件启动条件、硬件清零条件、GCMBR 寄存器的捕获输入条件均设为 TIM6\_<t>\_TRIGB 的上升沿，就可以实现连续的周期测量。对应动作如图 20-10 所示。

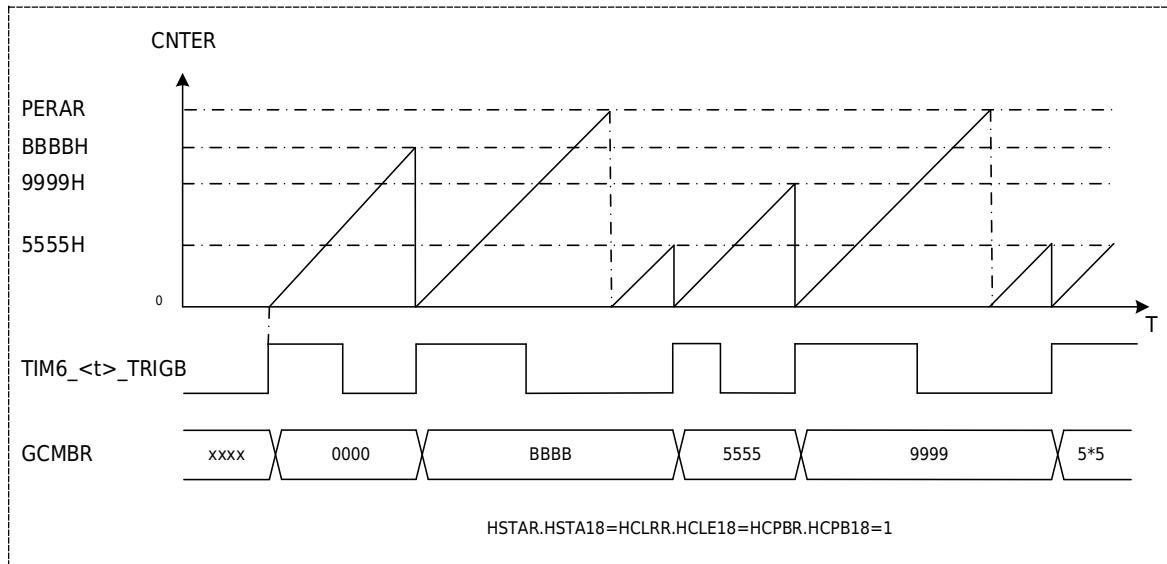


图 20-10 周期测量

## 20.3.11 缓存功能

Timer6 的通用周期基准值、通用比较基准值、专用比较基准值、死区时间基准值等都具有缓存功能，可在计数期间硬件实现周期变化、占空比变化、死区变化等。周期基准值、通用比较基准值和专用比较基准值等有单缓存、双缓存功能，死区时间基准值有单缓存功能。

### 20.3.11.1 单缓存动作

单缓存动作是指通过设定缓存控制寄存器（BCONR.BENA<B><P><SPA><SPB>=1、BCONR.BSEA<B><P><SPA><SPB>=0）、死区控制寄存器（DCONR.DTBENU<D>=1），在缓存传送时间点，选择发生以下事件：

- a) 通用周期基准值缓存寄存器（PERBR）的值自动传送到通用周期基准值寄存器（PERAR）中
- b) 通用比较基准值缓存寄存器（GCMCR、GCMDR）的值自动传送到通用比较基准值寄存器（GCMAR、GCMBR）中（比较输出时）
- c) 通用比较基准值寄存器（GCMAR、GCMBR）的值自动传送到通用比较基准值缓存寄存器（GCMCR、GCMDR）中（捕获输入时）
- d) 专用比较基准值缓存寄存器（SCMCR、SCMDR）的值自动传送到专用比较基准值寄存器（SCMAR、SCMBR）中
- e) 死区时间基准值缓存寄存器（DTUBR、DTDBR）的值自动传送到死区时间基准值寄存器（DTUAR、DTDAR）中

如图 20-11 所示，是单元 1 比较输出动作（PCNAR.CAPMDA=0）时、通用比较基准值寄存器的单缓存方式的时序图。从图中可以看到，在计数期间改变通用比较基准值寄存器（GCMAR）的值可以调整输出占空比，改变通用周期基准值寄存器（PERAR）的值可以调整输出周期。

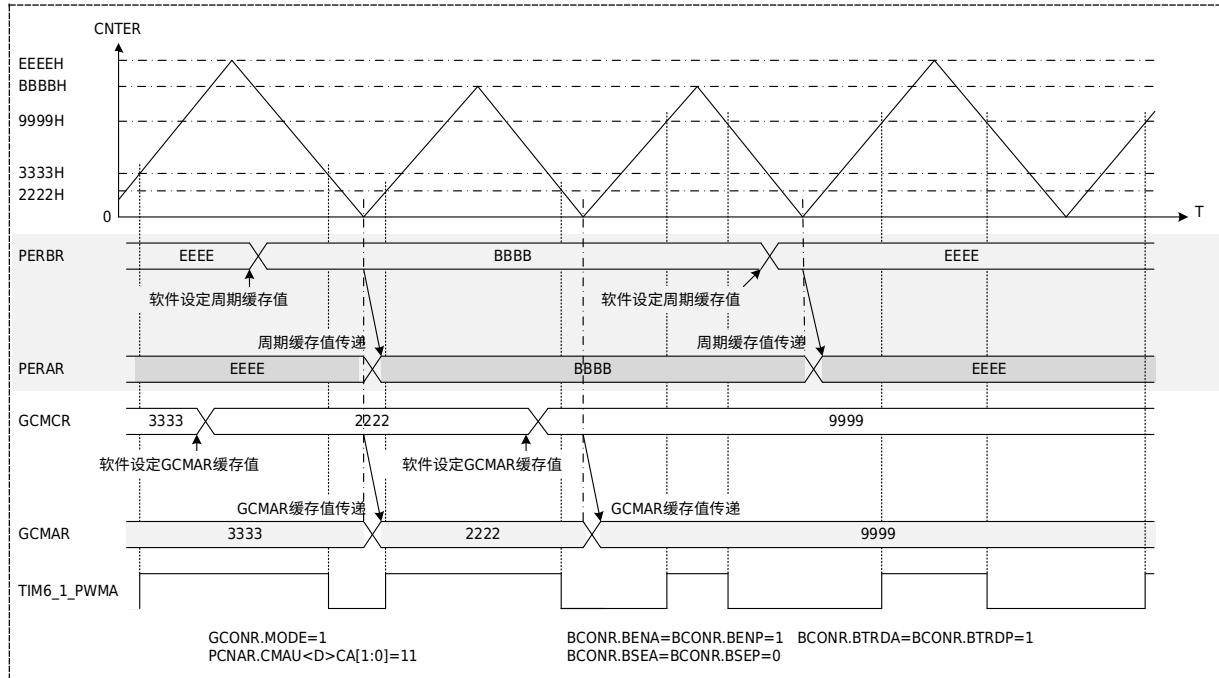


图 20-11 单缓存方式比较输出时序

### 20.3.11.2 双缓存动作

双缓存动作是指通过设定缓存控制寄存器（ $BCONR.BENA<B><P><SPA><SPB>=1$ 、 $BCONR.BSEA<B><P><SPA><SPB>=1$ ），在缓存传送时间点，选择发生以下事件：

- a) 通用周期基准值缓存寄存器（PERBR）的值自动传送到通用周期基准值寄存器（PERAR）中、通用周期基准值双缓存寄存器（PERCR）的值自动传送到通用周期基准值缓存寄存器（PERBR）中
  - b) 通用比较基准值缓存寄存器（GCMCR、GCMDR）的值自动传送到通用比较基准值寄存器（GCMAR、GCMBR）、通用比较基准值双缓存寄存器（GCMER、GCMFR）的值自动传送到通用比较基准值缓存寄存器（GCMCR、GCMDR）中（比较输出时）
  - c) 通用比较基准值缓存寄存器（GCMCR、GCMDR）的值自动传送到通用比较基准值双缓存寄存器（GCMER、GCMFR）中、通用比较基准值寄存器（GCMAR、GCMBR）的值自动传送到通用比较基准值缓存寄存器（GCMCR、GCMDR）中（捕获输入时）
  - d) 专用比较基准值缓存寄存器（SCMCR、SCMDR）的值自动传送到专用比较基准值寄存器（SCMAR、SCMBR）中、专用比较基准值双缓存寄存器（SCMER、SCMFR）的值自动传送到专用比较基准值缓存寄存器（SCMCR、SCMDR）中
- 图 20-12 所示，是内部触发事件 0 触发捕获输入时、双缓存方式的时序图。

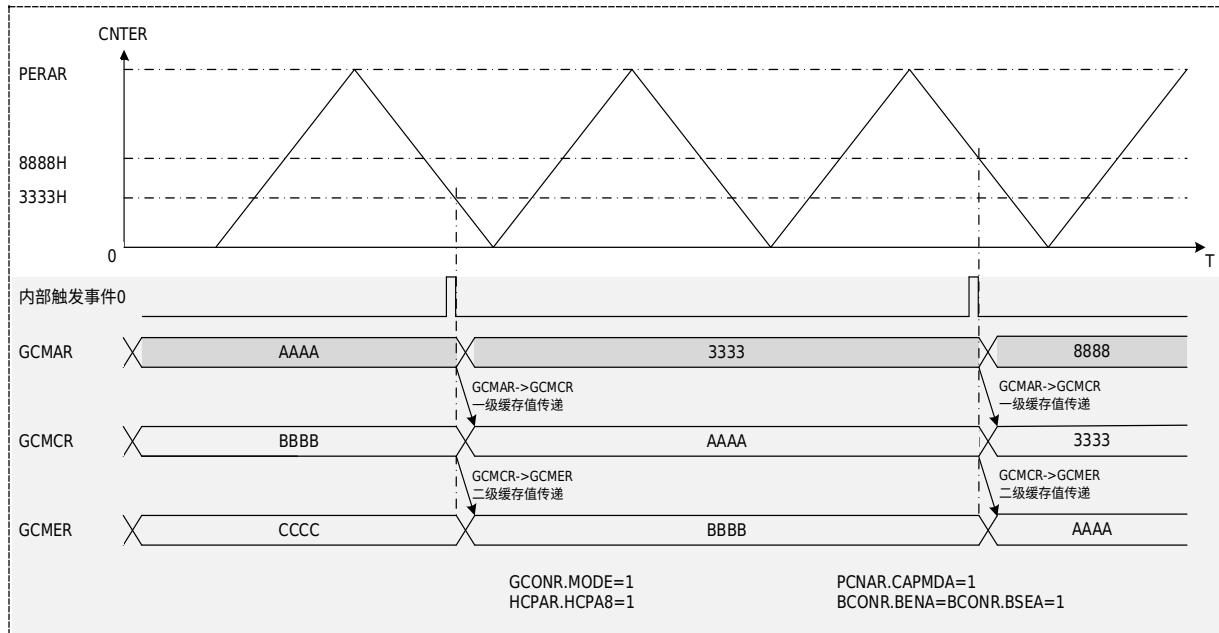


图 20-12 双缓存方式捕获输入时序

### 20.3.11.3 缓存传送节点

#### 比较输出（锯齿波）缓存传送

在缓存功能有效 ( $\text{BCONR.BENA} < \text{B} > < \text{P} > < \text{SPA} > < \text{SPB} > = \text{DCONR.DTBENU} < \text{D} > = 1$ ) && 比较输出计数 ( $\text{PCNA} < \text{B} > \text{R.CAPMDA} < \text{B} > = 0$ ) && 锯齿波计数模式 ( $\text{GCONR.MODE} = 0$ ) 时，通用周期基准值、通用比较基准值、专用比较基准值、死区时间基准值的缓存传送发生在递加计数上溢点或递减计数下溢点。

如图 20-13 所示，是锯齿波递加计数时的缓存动作。

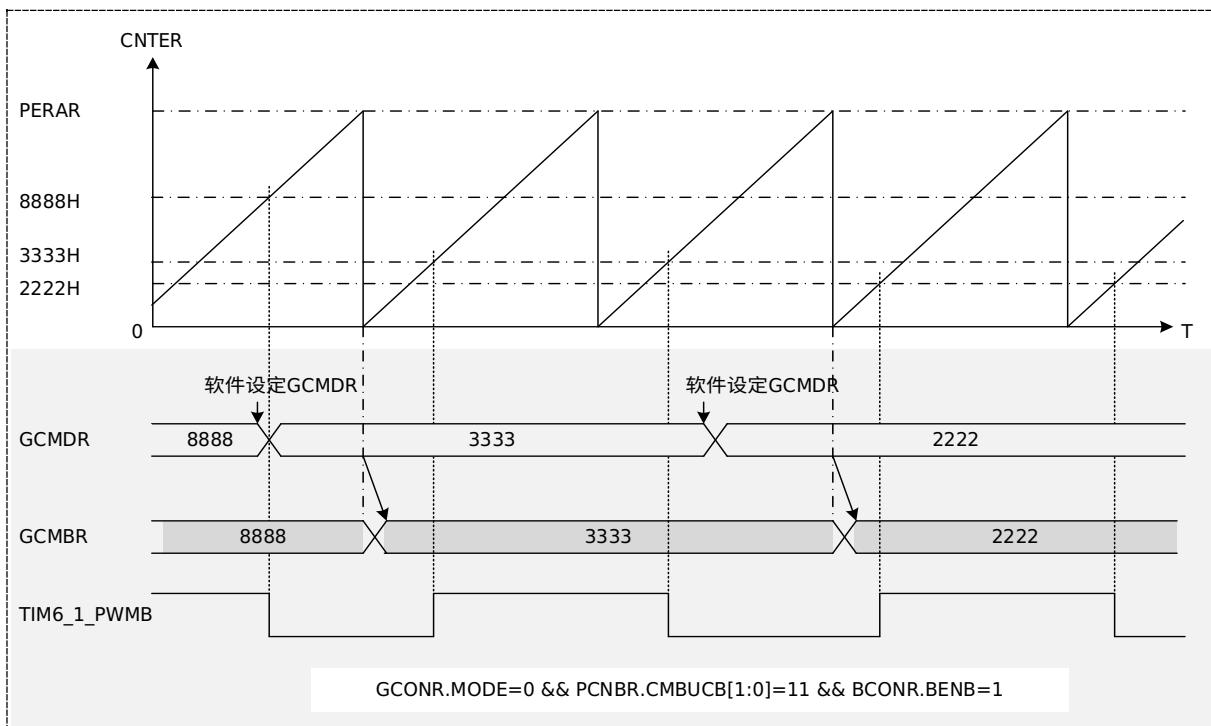


图 20-13 锯齿波模式时计数缓存动作

**注意：**

- 在锯齿波计数模式时，若产生清零动作，也视为一次计数溢出。各基准值会根据相应的缓存动作设定状况（单缓存、双缓存等）发生一次缓存传送。
- 在硬件计数模式时，若产生清零动作，也视为一次计数溢出。通用周期基准值、通用比较基准值会根据相应的缓存动作设定状况（单缓存、双缓存、上溢传送、下溢传送等）发生一次缓存传送，其它基准值不发生缓存传送。

**比较输出（三角波）缓存传送**

在缓存功能有效 ( $\text{BCONR.BENA} < \text{B} > < \text{P} > < \text{SPA} > < \text{SPB} > = \text{DCONR.DTBENU} < \text{D} > = 1$ ) && 比较输出计数 ( $\text{PCNA} < \text{B} > \text{R.CAPMDA} < \text{B} > = 0$ ) && 三角波计数模式 ( $\text{GCONR.MODE} = 1$ ) 时，通用周期基准值、通用比较基准值、专用比较基准值、死区时间基准值的缓存传送时间点由对应的寄存器控制位决定。当  $\text{BCONR.BTRUA} < \text{B} > < \text{P} > < \text{SPA} > < \text{SPB} > = 1$  或  $\text{DCONR.DTBTRU} = 1$  时，在计数器计数到三角波的峰点时，发生对应的缓存传送；当  $\text{BCONR.BTRDA} < \text{B} > < \text{P} > < \text{SPA} > < \text{SPB} > = 1$  或  $\text{DCONR.DTBTRD} = 1$  时，在计数器计数到三角波的谷点时，发生对应的缓存传送。

如图 20-14 所示，是三角波计数到谷点时的缓存动作，图 20-15 所示，是三角波计数到峰点和谷点时均发生缓存动作的示意图。

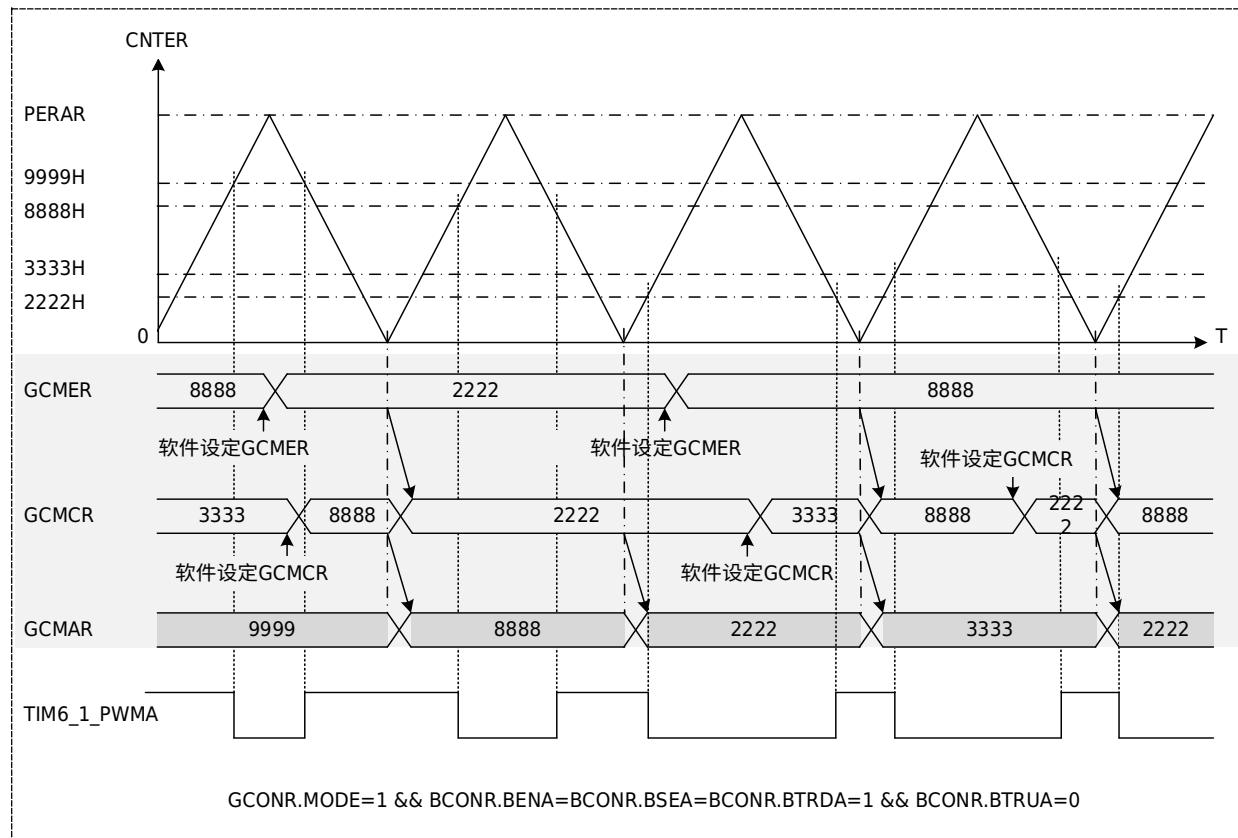


图 20-14 三角波模式时计数缓存动作 1

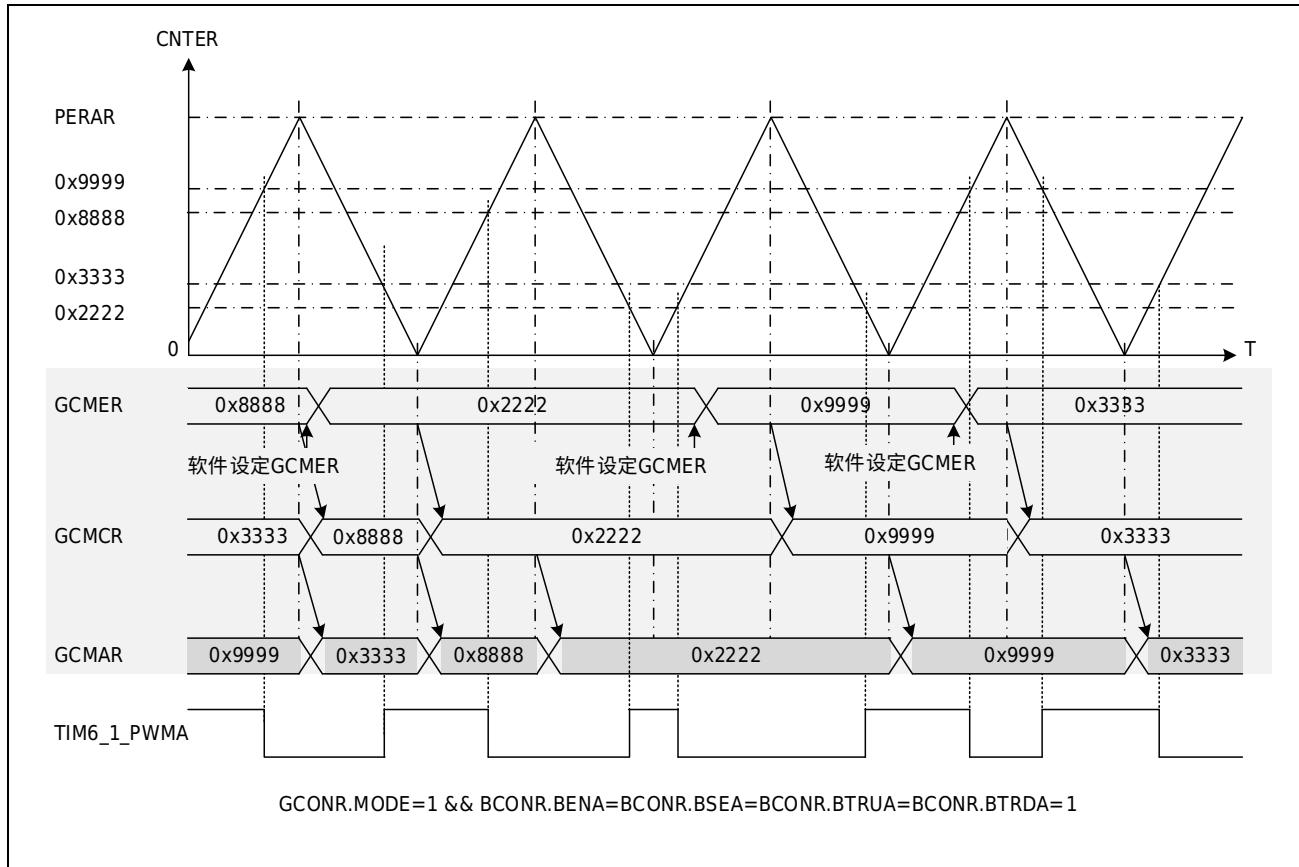


图 20-15 三角波模式时计数缓存动作 2

### 捕获输入缓存传送

在捕获输入动作 ( $PCNA<2>R.CAPMDA<2>=1$ ) 有效时，通用比较基准值支持缓存功能。若缓存设定有效 ( $BCONR.BENA<2>=1$ )，在捕获输入动作点会发生一次缓存传送。捕获输入动作可选择单缓存功能或双缓存功能（由  $BCONR.BSEA<2>$  设定）。

### 20.3.12 数字滤波

TIM6\_<t>\_PWMA、TIM6\_<t>\_PWMB、TIM6\_TRIGA~D 端口输入都具有数字滤波功能。TIM6\_<t>\_PWMA、TIM6\_<t>\_PWMB 通过设定滤波控制寄存器 (FCNGR) 的相关使能位开启对应端口的滤波功能，滤波有效时的滤波基准时钟也可通过滤波控制寄存器 (FCNGR) 设定；TIM6\_TRIGA~D 端口是一组单元间共用的端口通过设定滤波控制寄存器 (FCNTR) 的相关使能位开启对应端口的滤波功能，滤波有效时的滤波基准时钟也可通过滤波控制寄存器 (FCNTR) 设定。

在滤波采样基准时钟采样到端口上 3 次一致的电平时，该电平被当作有效电平传送到模块内部；小于 3 次一致的电平会被当作外部干扰滤掉，不传送到模块内部。其动作例如图 20-16 所示。

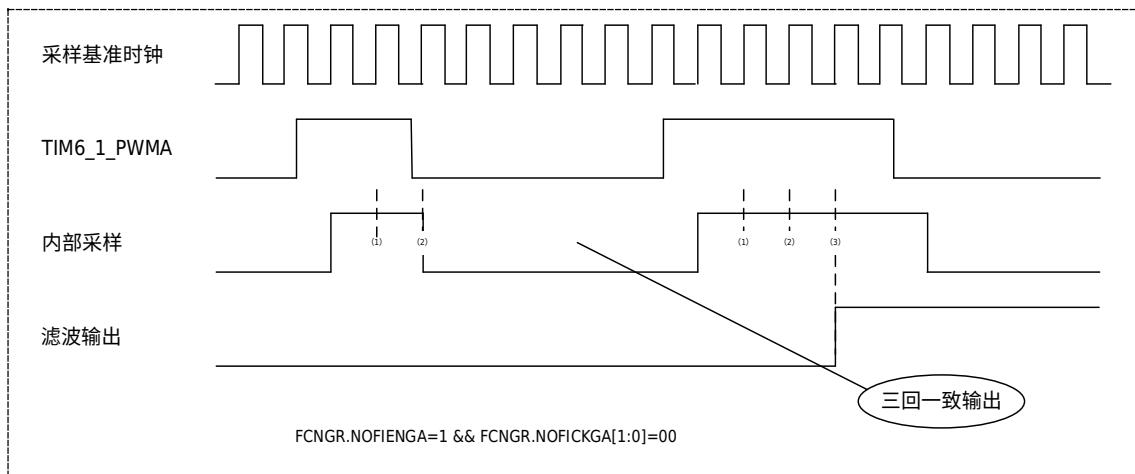


图 20-16 捕获输入端口的滤波功能

### 20.3.13 通用 PWM 输出

#### 20.3.13.1 单边对齐独立 PWM 输出

在锯齿波计数模式 (GCONR.MODE=0) 时，每个单元的 2 个端口 TIM6\_<t>\_PWMA、TIM6\_<t>\_PWMB 都能独立的输出 PWM 波。在计数溢出点若设定同样的电平变化，则可实现单边对齐独立 PWM 输出。如下图 20-17 所示。

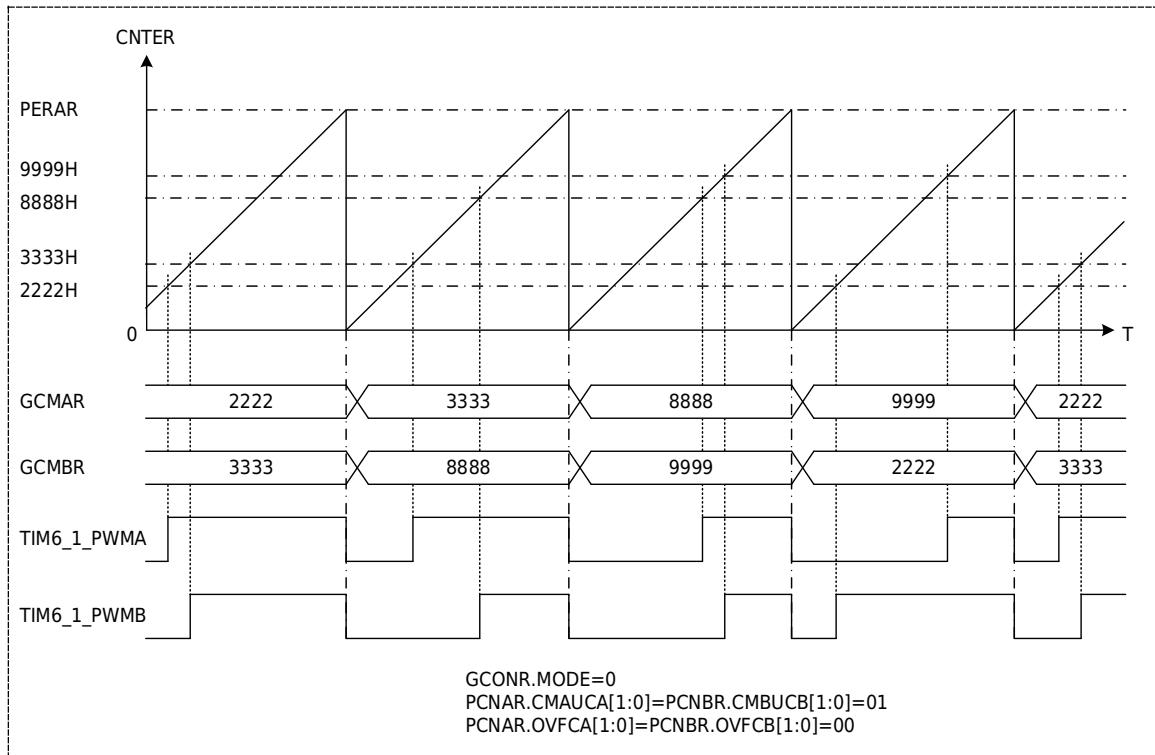


图 20-17 单边对齐独立 PWM

### 20.3.13.2 双边对称独立 PWM 输出

在三角波计数模式 (GCONR.MODE=1) 时，每个单元的 2 个端口 TIM6\_<t>\_PWMA、TIM6\_<t>\_PWMB 都能独立的输出 PWM 波。在计数比较匹配点若设定电平变化、在计数峰点和谷点设定电平不变化，则可实现双边对称独立 PWM 输出。如图 20-18 所示，是 TIM6\_<t>\_PWMA 端口和 TIM6\_<t>\_PWMB 端口实现双边对称独立输出 PWM。

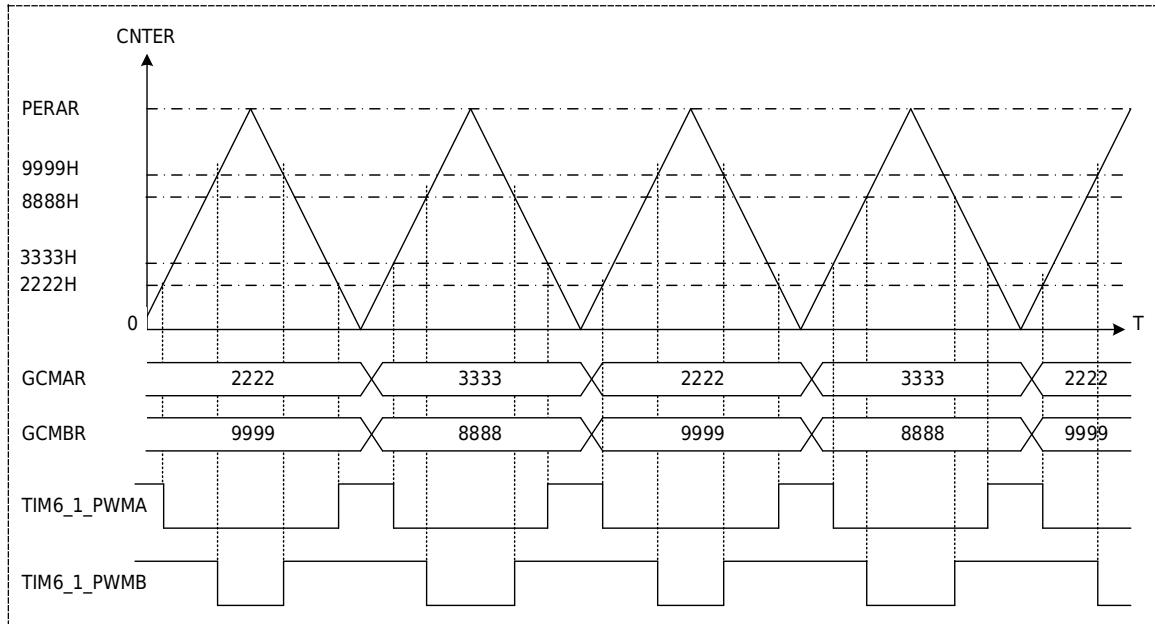


图 20-18 双边对称独立 PWM

### 20.3.13.3 双边对称互补 PWM 输出

在三角波计数模式 (GCONR.MODE=1) 时，设定好端口在计数启动、比较匹配、计数溢出时的电平状态，可实现 TIM6\_<t>\_PWMA 和 TIM6\_<t>\_PWMB 端口上输出一对互补 PWM 波形。

根据 GCMBR 基准值的赋值方式不同，双边对称互补 PWM 的输出可分为“软件设定 GCMBR 互补 PWM 输出”和“硬件设定 GCMBR 互补 PWM 输出”。

#### 软件设定 GCMBR 互补 PWM 输出

软件设定方式是指在三角波模式下，用于 TIM6\_<t>\_PWMB 端口波形输出的通用比较基准值寄存器 (GCMBR) 直接由 CPU 等写入，与 GCMAR 的值没有直接关系。

图 20-19 为软件设定 GCMBR 互补 PWM 波的输出例。

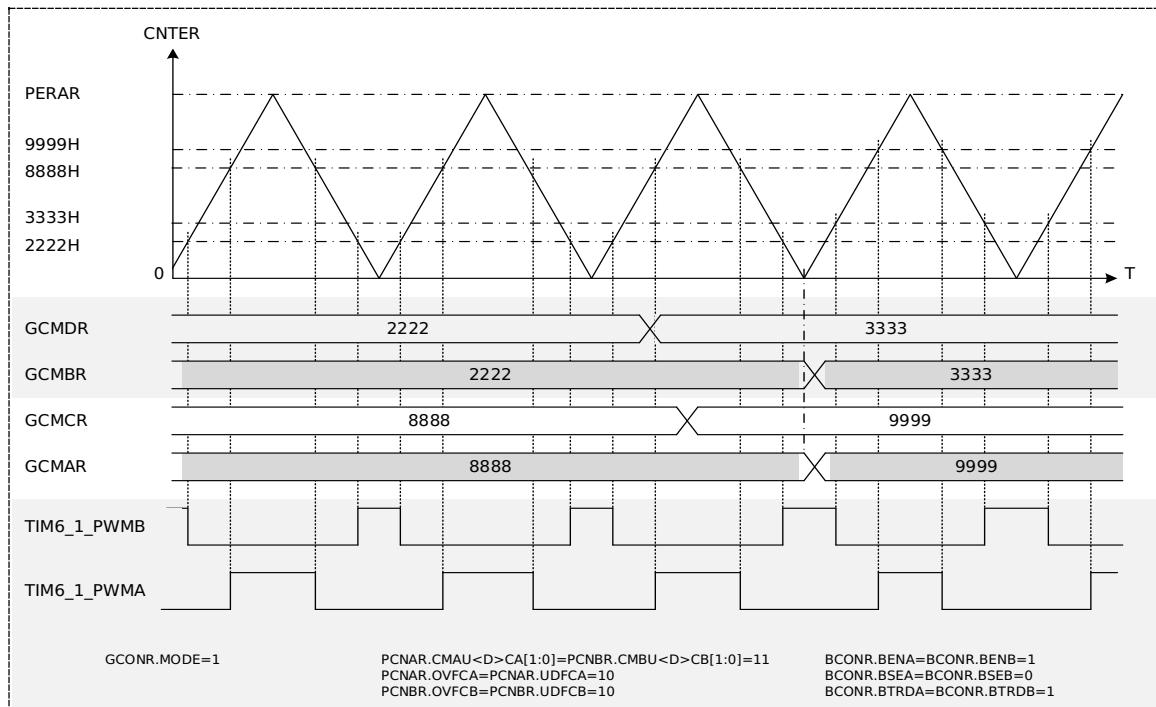


图 20-19 软件设定 GCMBR 互补 PWM 波输出

## 硬件设定 GCMBR 互补 PWM 输出

硬件设定方式是指在三角波模式下，用于 TIM6\_<t>\_PWMB 端口波形输出的通用比较基准值寄存器（GCMBR）的值由通用比较基准值寄存器（GCMAR）和死区时间基准值寄存器（DTU<D>AR）的值运算决定。

图 20-20 为硬件设定 GCMBR 互补 PWM 波输出例。

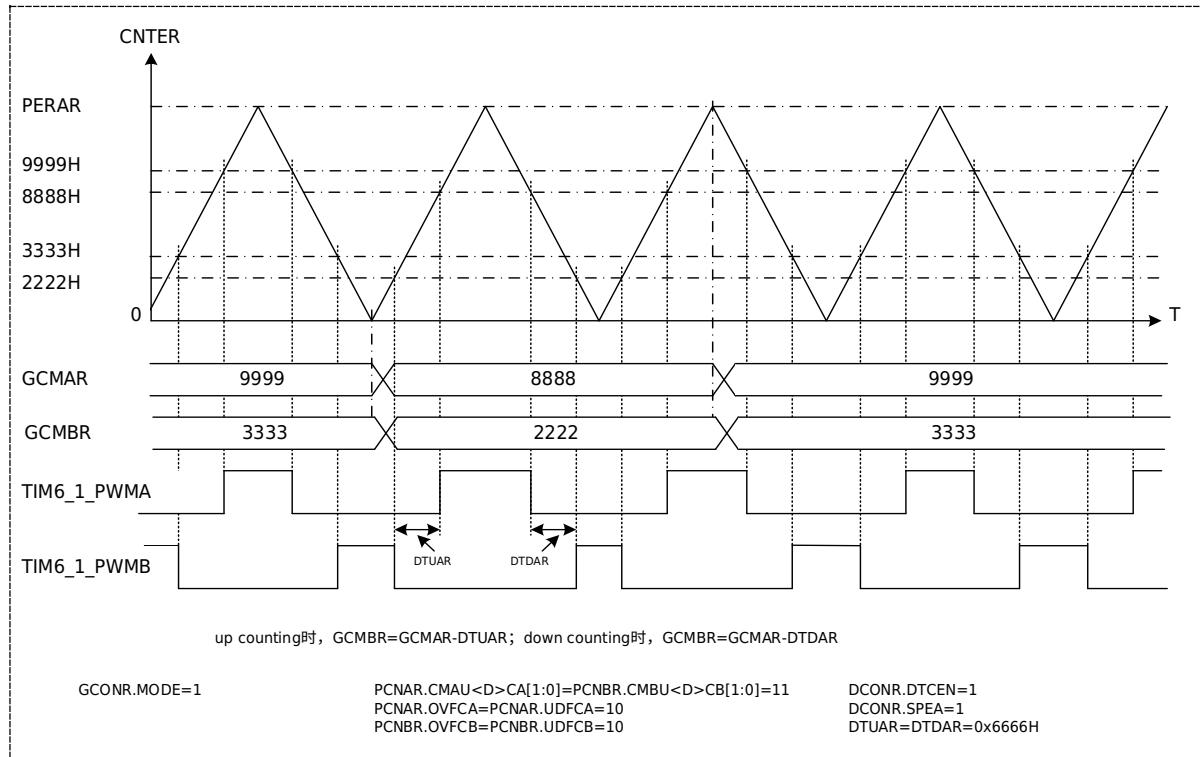


图 20-20 硬件设定 GCMBR 互补 PWM 波输出

### 20.3.13.4 双边非对称 PWM 输出

在三角波计数模式 (GCONR.MODE=1) 时，每个单元的 2 个端口 TIM6\_<t>\_PWMA、TIM6\_<t>\_PWMB 都能独立的输出 PWM 波，且每个端口都能根据通用比较基准值寄存器（GCMAR、GCMBR）的基准进行对应的电平输出变化。若在计数器的向上计数期间和向下计数期间，TIM6\_<t>\_PWMA 的端口电平变化分别由 GCMAR、GCMBR 基准值的比较结果控制，则在 TIM6\_<t>\_PWMA 端口上可实现双边非对称 PWM 输出。如图 20-21 所示，是 TIM6\_<t>\_PWMA 端口的双边非对称 PWM 输出。

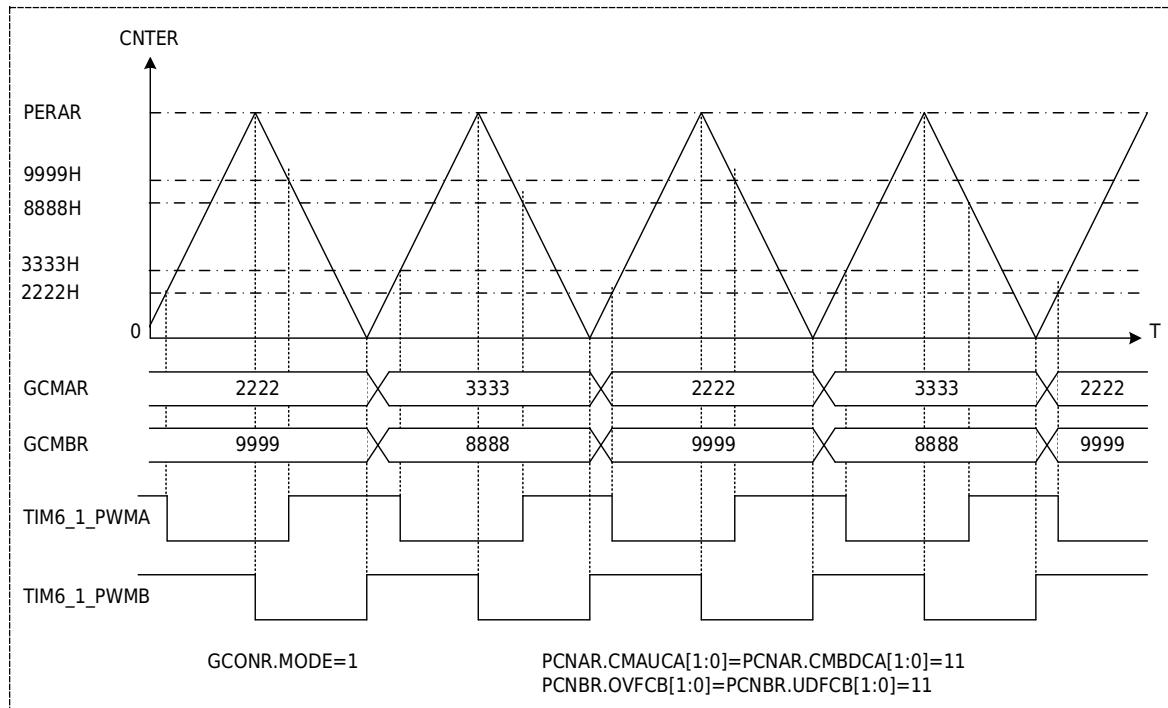


图 20-21 双边非对称 PWM 输出

### 20.3.13.5 单元间多相 PWM 输出

每个单元的 TIM6\_<t>\_PWMA、TIM6\_<t>\_PWMB 端口都能输出 2 相 PWM 波，多个单元间组合，同时结合软件、硬件同步动作就可实现多相 PWM 波输出。如图 20-22，单元 1、单元 2、单元 3 组合输出 6 相单边对齐独立 PWM；如图 20-23，单元 1、单元 2、单元 3 组合输出 3 相双边对称互补 PWM。

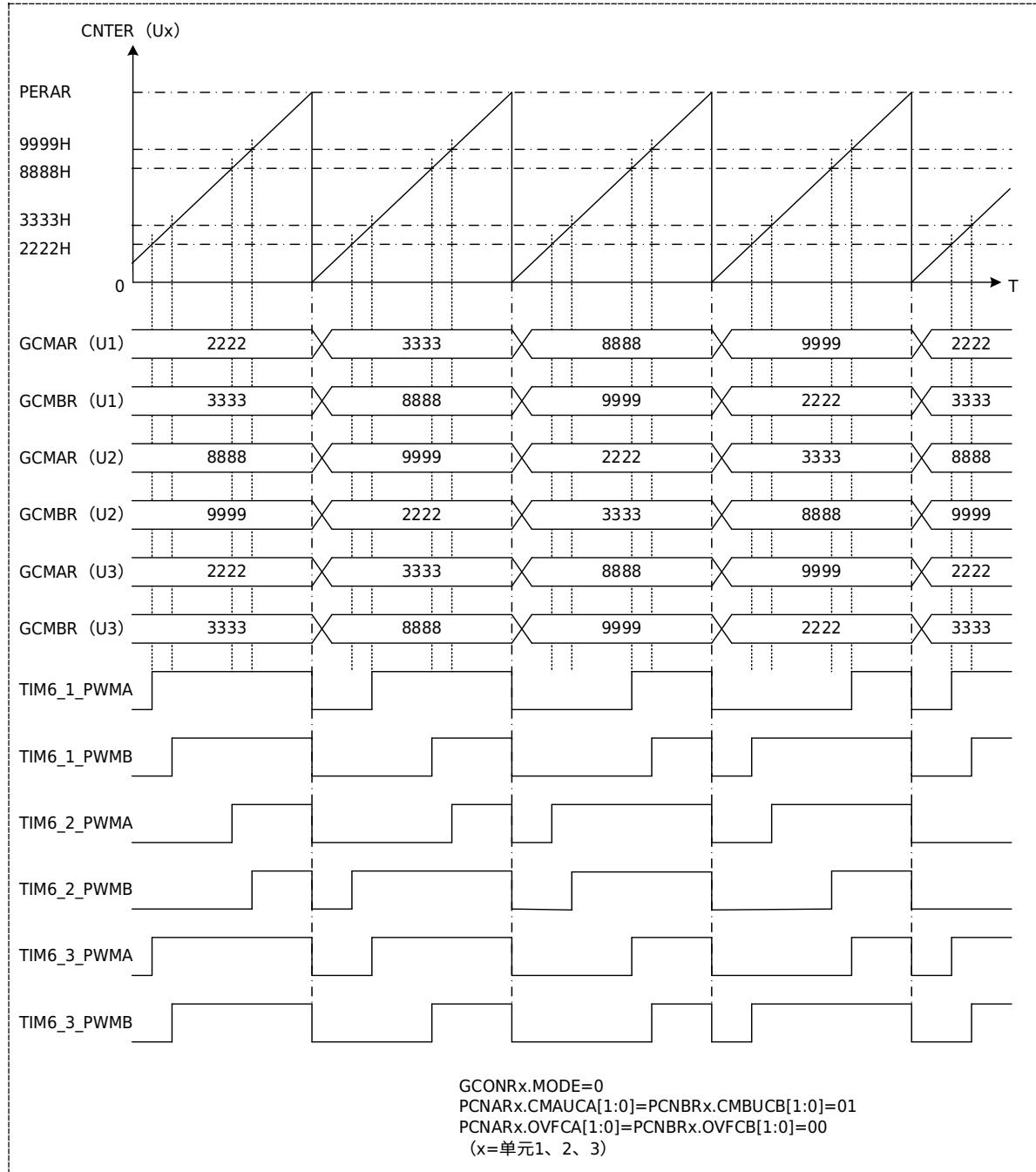


图 20-22 6 相单边对齐独立 PWM

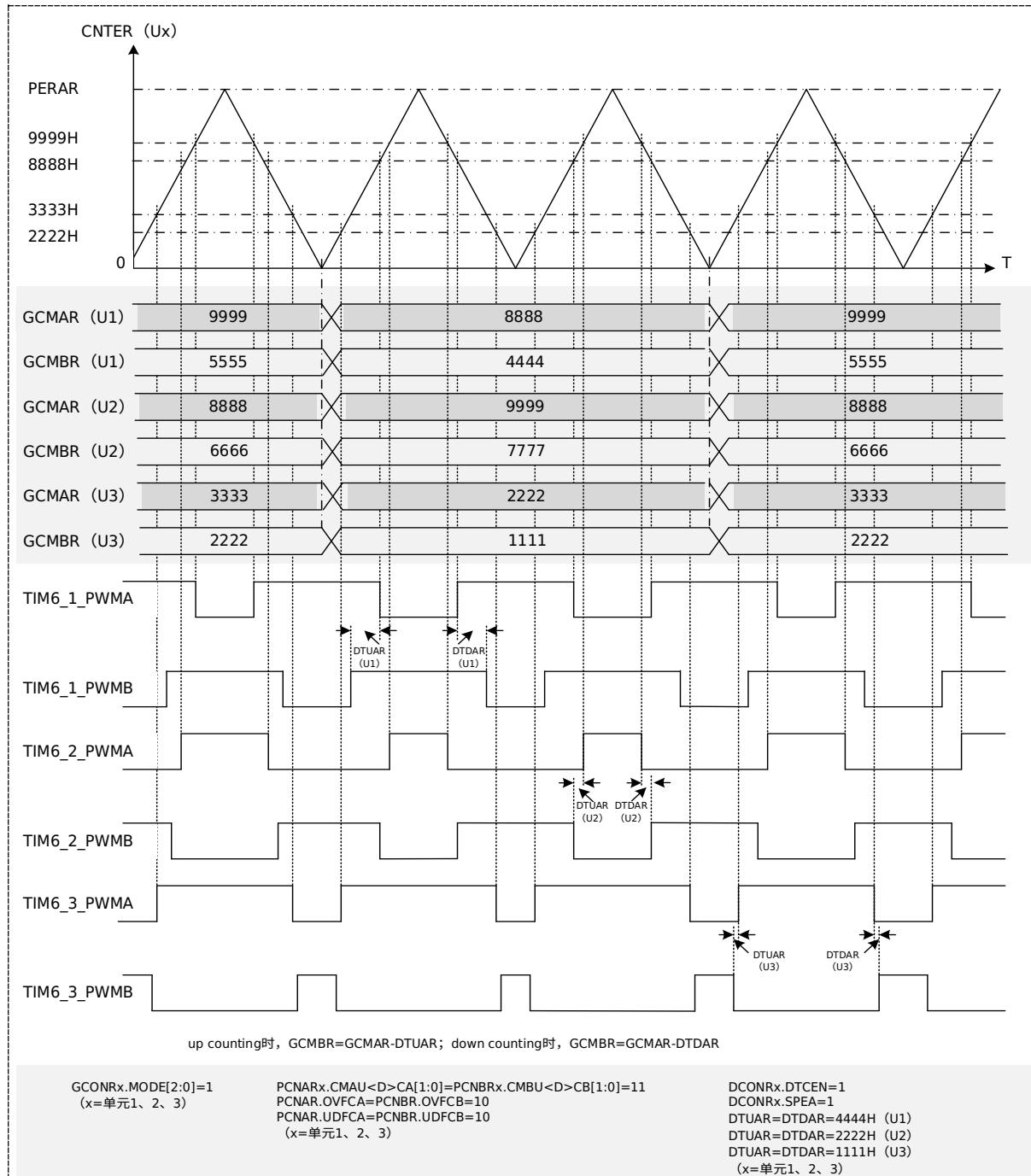


图 20-23 带死区时间 3 相双边对称互补 PWM

### 20.3.14 周期间隔响应

Timer6 的 2 个专用比较基准值寄存器（SCMAR、SCMBR），在计数比较匹配时可分别输出专用比较匹配中断 A 信号、专用比较匹配中断 B 信号到 INTC 产生对应的中断；同时可分别输出专用比较匹配事件 A 信号、专用比较匹配事件 B 信号，用于和其它模块关联动作，多用于启动 ADC 等。

该中断和事件的请求信号可以每间隔几个周期后产生一次有效的请求信号，即实现周期间隔响应。该功能通过设定有效周期寄存器（VPERR）的 VPERR.PCNTE[1:0]位和 VPERR.SPPERIA/B 位使能。设定 VPERR.PCNTS[2:0]位来指定每隔多少个周期请求信号有效一次，其它周期内即使计数值和专用比较基准值寄存器 SCMAR 或 SCMBR 的值相等，也不会输出有效的请求信号。

该功能有效后，各波形模式下的周期匹配中断和周期匹配事件也只在专用比较匹配中断和事件输出的有效周期里（下图中 STFLR.VPERNUM=0 的周期）输出。图 20-24 所示是周期间隔有效请求信号的动作例。

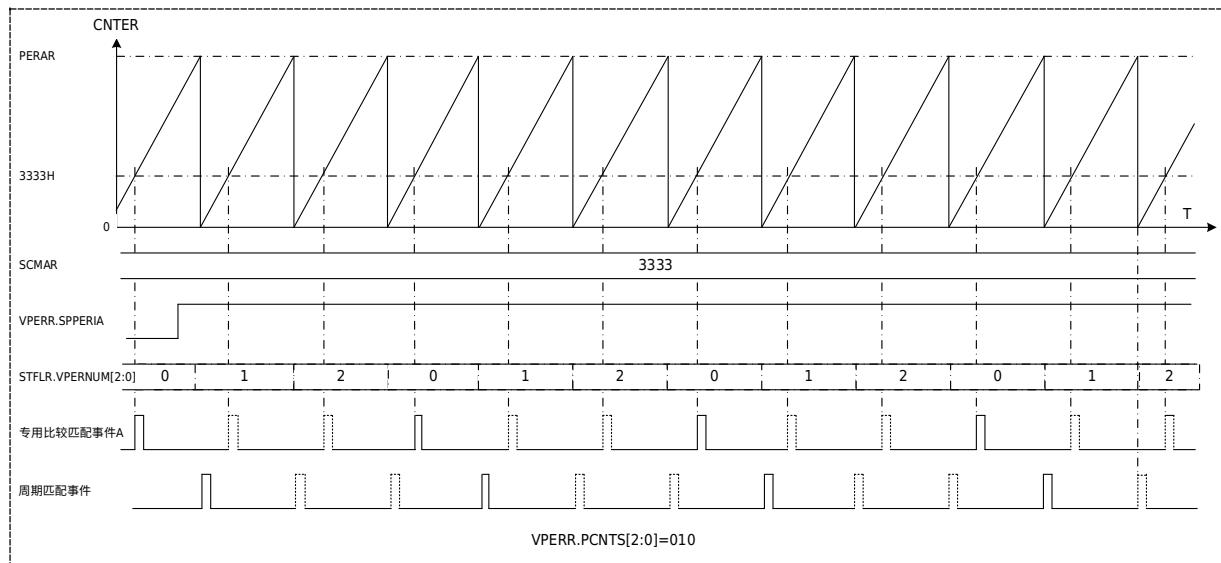


图 20-24 周期间隔有效请求信号动作

### 20.3.15 正交编码计数

将 TIM6\_<t>\_PWMA 输入看作 AIN 输入、TIM6\_<t>\_PWMB 输入看作 BIN 输入、TIM6\_TRIGA~D 中的任意一个输入看作 ZIN 输入，Timer6 就可以实现三路输入的正交编码计数。

一个单元的 AIN、BIN 单独动作可以实现位置计数模式；两个单元的 AIN、BIN、ZIN 组合动作可以实现公转计数模式，一个单元用于位置计数，一个单元用于公转计数。

公转计数模式时，Timer6 的 10 个单元可任意组合选择做位置计数单元或公转计数单元，但在 Z 相屏蔽功能有效时组合固定，单元 1、2 组合，单元 1 作为位置计数单元，单元 2 作为公转计数单元，分别实现位置计数和公转计数；单元 3、4 组合，单元 3 作为位置计数单元，单元 4 作为公转计数单元，分别实现位置计数和公转计数；单元 5、6 组合，单元 5 作为位置计数单元，单元 6 作为公转计数单元，分别实现位置计数和公转计数；单元 7、8 组合，单元 7 作为位置计数单元，单元 8 作为公转计数单元，单元 9、10 组合，单元 9 作为位置计数单元，单元 10 作为公转计数单元分别实现位置计数和公转计数。

AIN 和 BIN 的计数条件是通过设定硬件递加事件选择寄存器（HCUPR）和硬件递减事件选择寄存器（HCDOR）中 TIM6\_<t>\_PWMA 和 TIM6\_<t>\_PWMB 的正交关系实现；ZIN 的输入动作通过设定位置单元的硬件清零事件选择寄存器（HCLRR）实现位置计数单元的位置定时器清零、通过设定公转单元的硬件递加事件选择寄存器（HCUPR）实现公转计数单元的公转定时器计数。

### 20.3.15.1 位置计数模式

正交编码位置计数模式，是指根据 AIN、BIN 的输入实现基本计数功能、相位差计数功能和方向计数功能。

#### 基本计数

基本计数动作是根据 AIN 或 BIN 端口的输入时钟进行计数，如下图 20-25 所示。

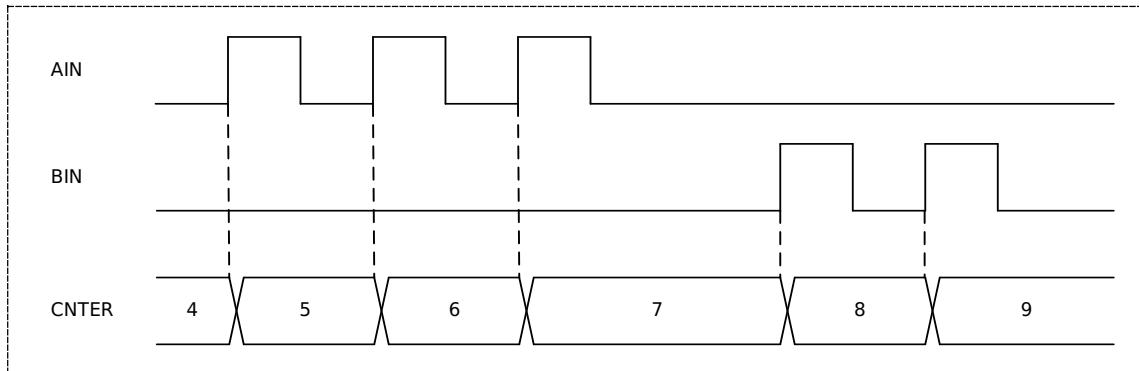


图 20-25 位置模式-基本计数

#### 相位差计数

相位差计数是指根据 AIN 和 BIN 的相位关系进行计数。根据设定的不同，可以实现 1 倍计数、2 倍计数、4 倍计数等，如下图 20-26~图 20-28 所示。

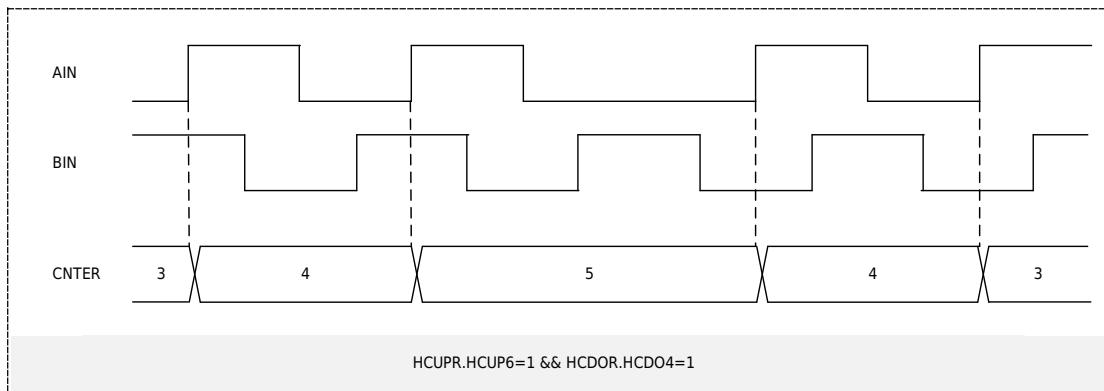


图 20-26 位置计数模式-相位差计数（1 倍计数）

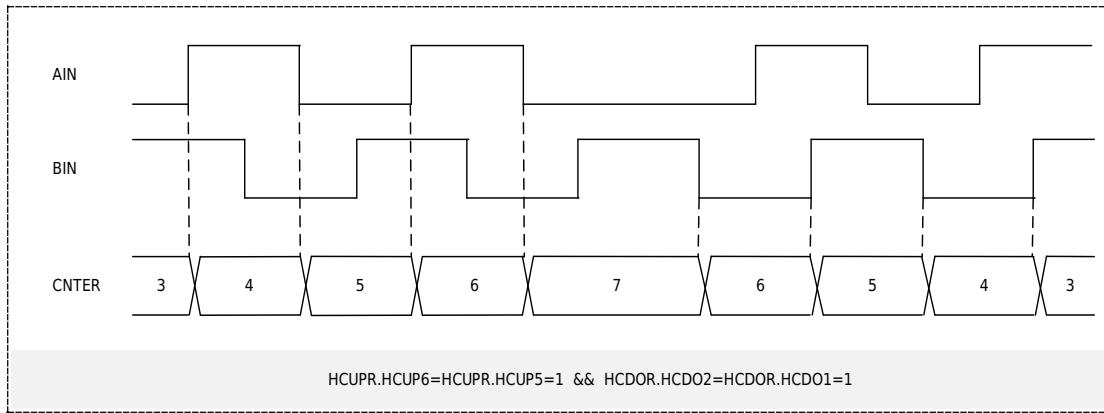


图 20-27 位置计数模式-相位差计数（2 倍计数）

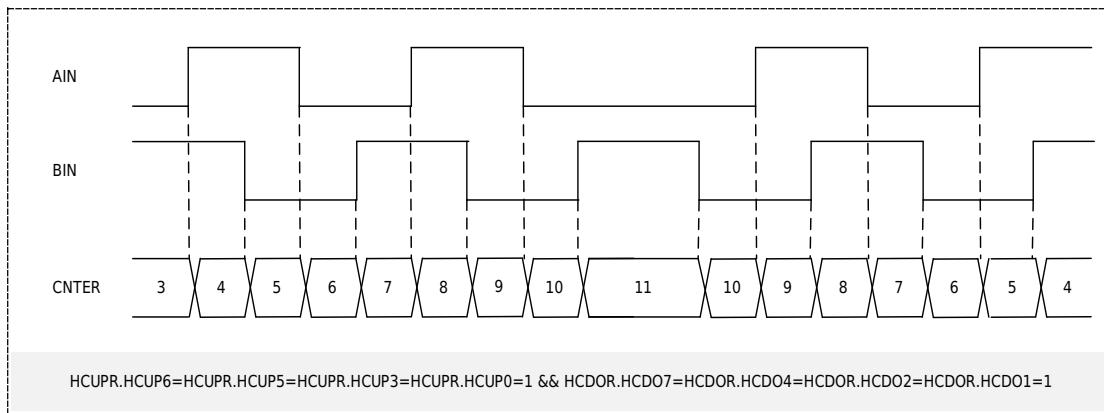


图 20-28 位置计数模式-相位差计数（4 倍计数）

## 方向计数

方向计数是指将 AIN 的输入状态设定为方向控制，将 BIN 的输入作为时钟计数，如下图 20-29 所示。

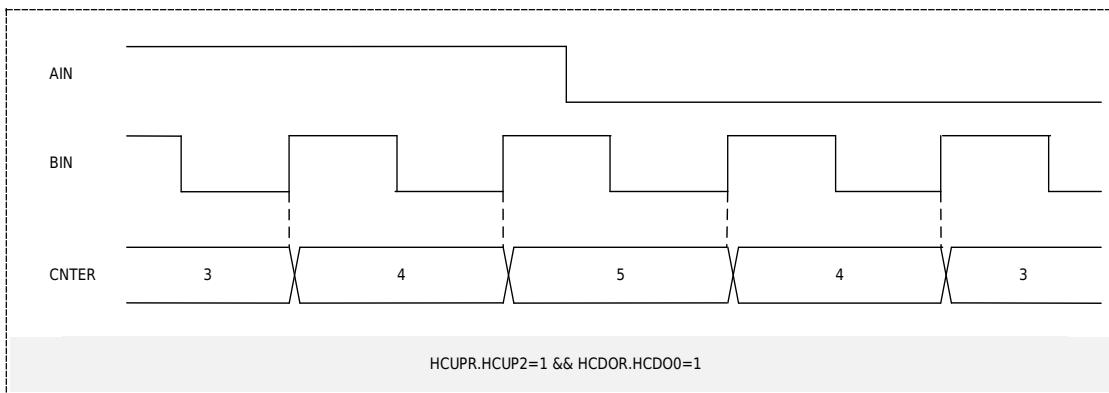


图 20-29 位置计数模式-方向计数

### 20.3.15.2 公转计数模式

正交编码公转计数模式，是指在 AIN、BIN 计数的基础上，加入 ZIN 的输入事件以实现对公转圈数等的判断。公转计数模式时根据公转定时器的计数方式，可实现 Z 相计数功能、位置溢出计数功能和混合计数功能。

#### Z 相计数

Z 相计数是指根据 ZIN 的输入，公转计数单元进行计数，同时将位置计数单元清零的计数动作。如下图 20-30 所示。

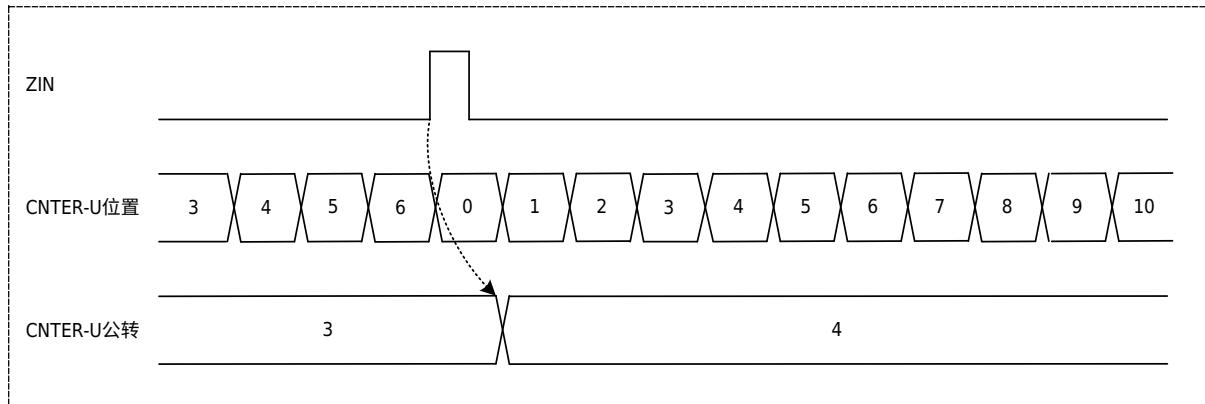


图 20-30 公转计数模式-Z 相计数

#### 位置溢出计数

位置溢出计数是指位置计数单元计数发生上溢或下溢时，产生一个溢出事件，从而触发公转计数单元的定时器进行一次计数（在该计数方式时 ZIN 的输入不进行公转计数单元的计数动作和位置计数单元的清零动作）。

位置计数单元的溢出事件通过内部触发事件接口选通实现公转计数单元计数，即可实现位置溢出计数。公转计数单元的硬件递加（递减）事件选择寄存器（HCUPR 或 HCDOR）的递加（递减）事件选择 Bit8~Bit11 中的 1 位，同时将对应的硬件触发事件选择寄存器（HTSSR0~3）中的事件编号设定为位置计数单元的上溢或下溢事件。具体事件编号参考 INTC 章节。如下图 20-31 所示。

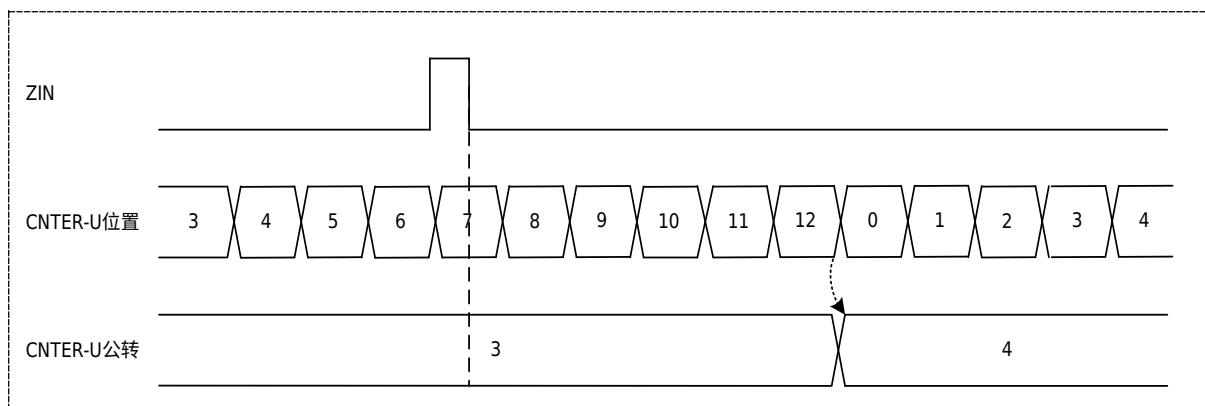


图 20-31 公转计数模式-位置溢出计数

## 混合计数

混合计数是指上述 Z 相计数和位置溢出计数两种计数方式合并起来的计数动作，其实现方式也是上述两种计数方式的组合。如下图 20-32 所示。

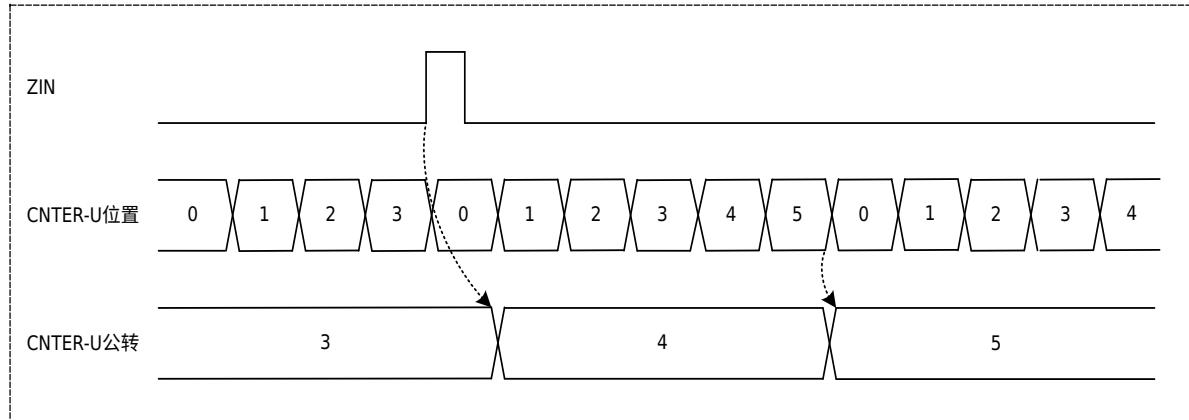


图 20-32 公转计数模式-混合计数

### 20.3.15.3 Z 相动作屏蔽

在公转计数模式的 Z 相计数功能或混合计数功能时，可以设定在位置定时器的上溢点或下溢点后的几个周期内（GCONR.ZMSKVAL[0:1]设定），将 ZIN 的有效输入屏蔽，不进行公转计数单元的计数和位置计数单元的清零。

位置计数单元的通用控制寄存器（GCONR）的 GCONR.ZMSKPOS 为 1 时，位置计数单元的 Z 相屏蔽功能使能，Z 相屏蔽的周期数由 GCONR.ZMSKVAL 设定；公转计数单元的通用控制寄存器（GCONR）的 GCONR.ZMSKREV 为 1 时，公转计数单元的 Z 相屏蔽功能使能。

图 20-33 是公转计数模式混合计数时，在位置计数单元计数上溢后的 4 个计数周期内有 ZIN 相输入时，ZIN 相输入的动作无效，即公转计数单元不计数、位置计数单元不清零；之后再来的 ZIN 相输入正常动作。

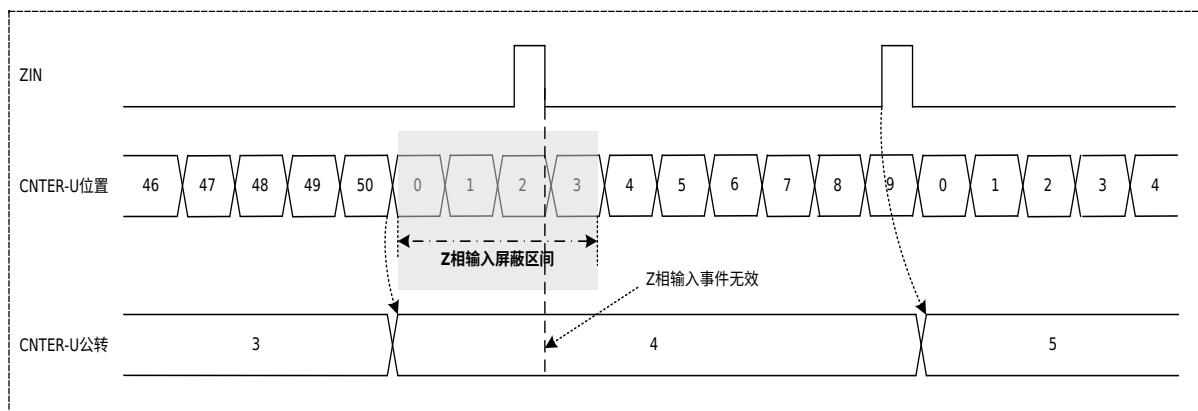


图 20-33 公转计数模式-混合计数 Z 相屏蔽动作例 1

图 20-34 是公转计数模式混合计数时，在位置计数单元计数上溢后的第 3 个周期，计数方向发生变化，此时设定的 4 个周期的屏蔽周期变为无效（实际 ZIN 相屏蔽功能维持了 3 个周期），开始向下计数。在

位置计数单元发生计数下溢后，ZIN 相屏蔽功能重新开启，维持 4 个周期后变为无效。在 ZIN 相屏蔽期间，ZIN 相的输入功能无效，即公转计数单元不计数、位置计数单元不清零；之后再来的 ZIN 相输入正常动作。

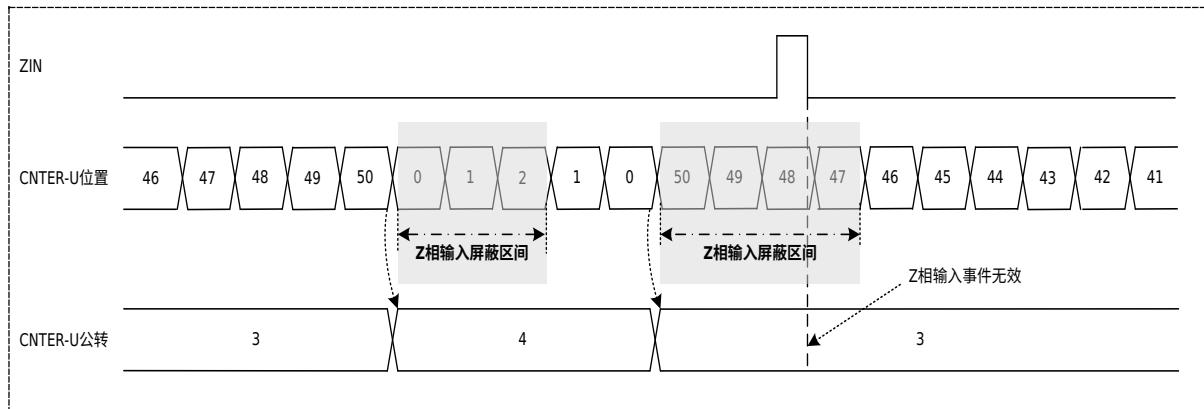


图 20-34 公转计数模式-混合计数 Z 相屏蔽动作例 2

### 20.3.16 EMB 控制

Timer6 可以对端口的输出状态进行保护控制，在发生异常时将端口状态固定成预先设定好的安全状态。所有单元具有 6 个共用的端口输出控制接口，每个单元通过端口控制寄存器 (PCNA<B>R.EMBSA<B>) 的设定选择要使用的 EMB 事件连接接口，这个接口连接 EMB 模块输出的 1 组 EMB 事件。同时接口上选通的异常状况事件可从 EMB 侧设定（参见 EMB 章节），当这些接口上监测到异常状况时，可以实现对通用 PWM 输出的控制。

端口在正常输出期间，若监测到从 EMB 过来的 EMB 事件，则端口的输出状态可变为预先设定好的状态。通用 PWM 输出端口在 EMB 异常事件发生时，端口状态可以变为输出高阻态、输出低电平或输出高电平（根据 PCNA<B>R.EMBCA<B>的设定决定）。例如，若 PCNAR.EMBCA=01 设定时，则在 TIM6\_<t>\_PWMA 端口正常输出期间，若产生 EMB 事件，则 TIM6\_<t>\_PWMA 端口上输出变为高阻态。

在 EMB 事件无效后（从 EMB 模块连接到 Timer6 的异常事件消失，信号变为正常电平），PWM 端口的输出可以自动恢复到正常的输出。此时，可以通过端口控制寄存器的设定选择立即恢复 PWM 正常输出 (PCNA<B>R.EMBRA<B>=00；这种方式称之为 One Shot 方式释放) 或在计数到下一个溢出点之后再恢复 PWM 的正常输出 (PCNA<B>R.EMBRA<B>=01、10、11；这种方式称之为 Cycle By Cycle 方式释放)。

### 20.3.17 典型应用例

下面描述几种典型应用情况下，Timer6 相关寄存器的基本设定，供用户参考。

#### 20.3.17.1 基本计数及中断动作

- a) 设定通用周期基准值 (PERAR)
- b) 设定需要的比较基准值，包括通用比较基准值 (GCMAR~GCMFR)、专用比较基准值 (SCMAR~SCMBR) 等
- c) 设定需要的中断使能位，包括计数上溢中断 (ICONR.INTENOVF)、计数下溢中断 (ICONR.INTENUDF)、计数匹配中断 (ICONR.INTENA~F, ICONR.INTENSAU, ICONR.INTENSAD, ICONR.INTENSBU, ICONR.INTENSB) 等
- d) 设定内部计数时钟分频 (GCONR.CKDIV[3:0])
- e) 设定波形模式 (GCONR.MODE)
- f) 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定)
- g) 启动计数器 (GCONR.START=1)

#### 20.3.17.2 比较输出及中断动作

- a) 设定通用周期基准值 (PERAR)
- b) 设定各通道的比较基准值，包括通用比较基准值 A (GCMAR)、通用比较基准值 B (GCMBR)
- c) 设定需要的中断使能位，包括计数上溢中断 (ICONR.INTENOVF)、计数下溢中断 (ICONR.INTENUDF)、计数匹配中断 (ICONR.INTENA~B) 等
- d) 设定各通道在不同计数状态时的端口输出状态 (参考 PCNAR 或 PCNBR 的 bit17~bit0 相关控制)
- e) 设定内部计数时钟分频 (GCONR.CKDIV[3:0])
- f) 设定波形模式 (GCONR.MODE)
- g) 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定)
- h) 设定各通道比较输出模式 (PCNAR.CAPMDA=0, PCNBR.CAPMDB=0)
- i) 设定各通道输出使能 (PCNAR.OUTENA=1, PCNBR.OUTENB=1)
- j) 启动计数器 (GCONR.START=1)

#### 20.3.17.3 捕获输入及中断动作

- a) 设定通用周期基准值 (PERAR)
- b) 设定需要的中断使能位，包括计数上溢中断 (ICONR.INTENOVF)、计数下溢中断 (ICONR.INTENUDF)、捕获输入中断 (ICONR.INTENA~B) 等
- c) 设定各通道的捕获输入外部条件 (参考 HCPAR 或 HCPBR 的全部有效控制位。该有效控制位互相独立，可以同时选择多个均作为某通道的捕获输入条件)

- d) 设定内部计数时钟分频 (GCONR.CKDIV[3:0])
- e) 设定波形模式 (GCONR.MODE)
- f) 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定)
- g) 设定捕获输入模式 (PCNAR.CAPMDA=1、PCNBR.CAPMDB=1)
- h) 启动计数器 (GCONR.START=1)
- i) 等待捕获输入条件产生，读取对应通道的捕获输入值 (GCMAR 或 GCMBR) 或等待相应的中断产生

#### 20.3.17.4 缓存传送动作 (周期基准值)

- a) 设定需要的通用周期基准值 (PERAR、PERBR、PERCR)
- b) 设定单、双缓存传送方式 (BCONR.BSEP)
- c) 设定缓存传送时间点 (BCONR.BTRUP、BCONR.BTRDP，这两个控制位互相独立，可以同时选择，均作为缓存传送时间点) (该设定仅在三角波模式时有效，在锯齿波模式时无效)
- d) 设定内部计数时钟分频 (GCONR.CKDIV[3:0])
- e) 设定波形模式 (GCONR.MODE)
- f) 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定)
- g) 设定缓存功能有效 (BCONR.BENP=1)
- h) 启动计数器 (GCONR.START=1)
- i) 等待对应的缓存传送时间点，发生缓存动作 (PERBR->PERAR (BCONR.BSEP=0 时)、PERCR->PERBR->PERAR (BCONR.BSEP=1 时))

#### 20.3.17.5 缓存传送动作 (通用比较基准值)

- a) 设定需要的通用比较基准值 (GCMAR、GCMCR、GCMER、GCMBR、GCMDR、GCMFR)
- b) 设定各通道单、双缓存传送方式 (BCONR.BSEA、BCONR.BSEB)
- c) 设定各通道缓存传送时间点 (BCONR.BTRUA、BCONR.BTRDA、BCONR.BTRUB、BCONR.BTRDB，每个通道的两个控制位互相独立，可以同时选择，均作为缓存传送时间点) (该设定仅在三角波模式时有效，在锯齿波模式时无效)
- d) 设定内部计数时钟分频 (GCONR.CKDIV[3:0])
- e) 设定波形模式 (GCONR.MODE)
- f) 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定)
- g) 设定各通道缓存功能有效 (BCONR.BENA=1、BCONR.BENB=1)
- h) 启动计数器 (GCONR.START=1)
- i) 等待各通道所设定的对应缓存传送时间点，发生缓存动作 (GCMCR->GCMAR (BCONR.BSEA=0 时)、GCMER->GCMCR->GCMAR (BCONR.BSEA=1 时)、GCMDR->GCMBR (BCONR.BSEB=0 时)、GCMFR->GCMDR->GCMBR (BCONR.BSEB=1 时))

### 20.3.17.6 缓存传送动作（专用比较基准值）

- a) 设定需要的专用比较基准值 (SCMAR、SCMCR、SCMER、SCMBR、SCMDR、SCMFR)
- b) 设定各通道单、双缓存传送方式 (BCONR.BSESPA、BCONR.BSESPB)
- c) 设定各通道缓存传送时间点 (BCONR.BTRUSPA、BCONR.BTRDSPA、BCONR.BTRUSPB、  
BCONR.BTRDSPB, 每个通道的两个控制位互相独立, 可以同时选择, 均作为缓存传送时间点)  
(该设定仅在三角波模式时有效, 在锯齿波模式时无效)
- d) 设定内部计数时钟分频 (GCONR.CKDIV[3:0])
- e) 设定波形模式 (GCONR.MODE)
- f) 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定)
- g) 设定各通道缓存功能有效 (BCONR.BENSPA=1、BCONR.BENSPB=1)
- h) 启动计数器 (GCONR.START=1)
- i) 等待各通道所设定的对应缓存传送时间点, 发生缓存动作 (SCMCR->SCMAR  
(BCONR.BSESPA=0 时)、SCMER->SCMCR->SCMAR (BCONR.BSESPA=1 时)、  
SCMDR->SCMBR (BCONR.BSESPB=0 时)、SCMFR->SCMDR->SCMBR (BCONR.BSESPB=1 时))

### 20.3.17.7 缓存传送动作（死区基准值）

- a) 设定需要的死区时间基准值 (DTUAR、DTUBR、DTDAR、DTDBR)
- b) 设定缓存传送时间点 (DCONR.DTBTRU、DCONR.DTBTRD, 这两个控制位互相独立, 可以同时  
选择, 均作为缓存传送时间点) (该设定仅在三角波模式时有效, 在锯齿波模式时无效)
- c) 设定内部计数时钟分频 (GCONR.CKDIV[3:0])
- d) 设定波形模式 (GCONR.MODE)
- e) 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定)
- f) 设定缓存功能有效 (DCONR.DTBENU=1、DCONR.DTBEND=1)
- g) 设定硬件死区功能有效 (DCONR.DTCEN=1)
- h) 启动计数器 (GCONR.START=1)
- i) 等待对应缓存传送时间点, 发生缓存动作 (DTUBR->DTUAR、DTDBR->DTDAR)

### 20.3.17.8 同步启动动作（软件方式）

- a) 参考[基本计数及中断动作]章节的 a~f 步骤, 对需要同步启动的各个单元做设定
- b) 同步启动计数器 (设定 SSTAR 寄存器的对应位为 1, 每个单元对应一个寄存器位)

### 20.3.17.9 同步启动动作（硬件方式）

- a) 设定通用周期基准值 (PERAR)
- b) 设定需要的比较基准值, 包括通用比较基准值 (GCMAR~GCMFR)、专用比较基准值  
(SCMAR~SCMBR) 等

- c) 设定需要的中断使能位，包括计数上溢中断（ICONR.INTENOVF）、计数下溢中断（ICONR.INTENUDF）、计数匹配中断（ICONR.INTENA~F、ICONR.INTENSAU、ICONR.INTENSAD、ICONR.INTENSBU、ICONR.INTENSBD）等
- d) 设定硬件启动条件（通过 HSTAR.HSTAx 选择， $x=8\sim31$ ）
- e) 设定硬件启动使能（HSTAR.STAS=1）
- f) 重复上述 a~e 步骤对需要同步启动的各个单元做设定（需要同步启动的各个单元中，步骤 d 的设定要一致）
- g) 等待设定的触发事件产生，确认各个单元的计数器同步启动

#### 20.3.17.10 正交编码计数动作（2相）

- a) 设定通用周期基准值（PERAR）
- b) 设定需要的比较基准值，包括通用比较基准值（GCMAR~GCMFR）、专用比较基准值（SCMAR~SCMBR）等
- c) 设定需要的中断使能位，包括计数上溢中断（ICONR.INTENOVF）、计数下溢中断（ICONR.INTENUDF）、计数匹配中断（ICONR.INTENA~F、ICONR.INTENSAU、ICONR.INTENSAD、ICONR.INTENSBU、ICONR.INTENSBD）等
- d) 设定需要的硬件向上计数条件（通过 HCUPR.HCUPx 选择， $x=0\sim7$ ）
- e) 设定需要的硬件向下计数条件（通过 HCDOR.HCDOx 选择， $x=0\sim7$ ）
- f) 启动计数器（GCONR.START=1）
- g) 等待设定的正交编码计数事件产生，确认计数器正常计数

#### 20.3.17.11 正交编码计数动作（3相）

- a) 参考[正交编码计数动作（2相）]章节的 a~e 步骤，对位置计数单元做设定
- b) 设定位臵计数单元的硬件清零条件（通过 HCLRR.HCLRx 选择， $x=16\sim31$ ）
- c) 设定位臵计数单元硬件清零使能（HCLRR.CLES=1）
- d) 设定公转计数单元的通用周期基准值（PERAR）
- e) 设定公转计数单元的比较基准值，包括通用比较基准值（GCMAR~GCMFR）、专用比较基准值（SCMAR~SCMBR）等
- f) 设定公转计数单元需要的中断使能位，包括计数上溢中断（ICONR.INTENOVF）、计数下溢中断（ICONR.INTENUDF）、计数匹配中断（ICONR.INTENA~F、ICONR.INTENSAU、ICONR.INTENSAD、ICONR.INTENSBU、ICONR.INTENSBD）等
- g) 设定公转计数单元的硬件向上计数条件 1（ZIN 相输入）（通过 HCUPR.HCUPx 选择， $x=16\sim31$ ，此处的设定事件应和步骤 b 中位臵计数单元设定的事件一致）
- h) 设定公转计数单元的硬件向上计数条件 2（位臵计数单元的上溢事件输入）（通过 HCUPR.HCUP8 选择内部硬件触发事件 0）

- i) 设定公转计数单元的硬件向下计数条件（位置计数单元的下溢事件输入）(通过 HCDOR.HCDO9 选择内部硬件触发事件 1)
- j) 设定 HTSSR0 中的触发源编号为位置计数单元的计数上溢事件（该上溢事件编号参考 INTC 章节）
- k) 设定 HTSSR1 中的触发源编号为位置计数单元的计数下溢事件（该下溢事件编号参考 INTC 章节）
- l) 启动公转计数单元计数器 (GCONR.START=1)
- m) 启动位置计数单元计数器 (GCONR.START=1)
- n) 等待设定的 AIN、BIN、ZIN 相计数事件产生，确认计数器正常计数

#### 20.3.17.12 单路 PWM 输出

- a) 参考[比较输出及中断动作]章节 a~j 步骤的设定（每个单元内部的 2 个 PWM 通道 TIM6\_<t>\_PWMA 和 TIM6\_<t>\_PWMB 的输出状态均可独立设定，形成 2 个互不相关的单路 PWM 输出）

#### 20.3.17.13 互补 PWM 输出（软件死区）

- a) 设定通用周期基准值 (PERAR)
- b) 设定通用比较基准值 A (GCMAR)、通用比较基准值 B (GCMBR)
- c) 设定需要的中断使能位，包括计数上溢中断 (ICONR.INTENOVF)、计数下溢中断 (ICONR.INTENUDF)、计数匹配中断 (ICONR.INTENA~B) 等
- d) 设定在不同计数状态时的端口输出状态（参考 PCNAR 和 PCNBR 的 bit17~bit0 相关控制，结合 GCMAR 和 GCMBR 的设定值，需要保证 2 个 PWM 输出之间形成互补死区）
- e) 设定内部计数时钟分频 (GCONR.CKDIV[3:0])
- f) 设定波形模式为三角波模式 (GCONR.MODE=1)
- g) 设定比较输出模式 (PCNAR.CAPMDA=0、PCNBR.CAPMDB=0)
- h) 设定输出使能 (PCNAR.OUTENA=1、PCNBR.OUTENB=1)
- i) 启动计数器 (GCONR.START=1)

#### 20.3.17.14 互补 PWM 输出（硬件死区）

- a) 设定通用周期基准值 (PERAR)
- b) 设定通用比较基准值 A (GCMAR)、死区时间基准值 (DTUAR、DTDAR)
- c) 设定需要的中断使能位，包括计数上溢中断 (ICONR.INTENOVF)、计数下溢中断 (ICONR.INTENUDF)、计数匹配中断 (ICONR.INTENA~B)、死区错误中断 (ICONR.INTENDTE) 等
- d) 设定在不同计数状态时的端口输出状态（参考 PCNAR 和 PCNBR 的 bit17~bit0 相关控制，结合 GCMAR、DTUAR 和 DTDAR 的设定值，需要保证 2 个 PWM 输出之间形成互补死区）

- e) 设定内部计数时钟分频 (GCONR.CKDIV[3:0])
- f) 设定波形模式为三角波模式 (GCONR.MODE=1)
- g) 设定各通道比较输出模式 (PCNAR.CAPMDA=0、PCNBR.CAPMDB=0)
- h) 设定各通道输出使能 (PCNAR.OUTENA=1、PCNBR.OUTENB=1)
- i) 设定硬件死区功能有效 (DCONR.DTCEN=1)
- j) 启动计数器 (GCONR.START=1)

#### 20.3.17.15 EMB 监控及中断动作

- a) 参考[互补 PWM 输出（软件死区）]章节的 a~h 步骤或[互补 PWM 输出（硬件死区）]章节 a~i 步骤，对互补 PWM 输出动作进行设定
- b) 设定 EMB 事件发生时 PWM 端口的状态 (PCNAR.EMBCA、PCNBR.EMBCB) (根据系统应用的不同，选择相应的保护状态)
- c) 设定 EMB 事件变为无效时 PWM 端口恢复正常输出的时间点 (PCNAR.EMBRA、PCNBR.EMBRB)
- d) 设定选择从 EMB 模块输入的 EMB 事件源 (PCNAR.EMBSA、PCNBR.EMBSB)
- e) 设定 EMB 模块的相关寄存器 (包括 EMB 中断许可寄存器 (EMB\_INTEN0~5)、EMB 控制寄存器 1/2 (EMB\_CTL1/2\_0~5)、EMB 控制 PWM 输出释放方式选择寄存器 (EMB\_RLSSEL0~5) 等)
- f) 启动计数器 (GCONR.START=1)，EMB 模块实时监控系统状态

### 20.3.18 功能汇总表

Timer6 的锯齿波模式和三角波 A、B 模式下，主要功能的汇总表如下表 20-3 所示。

表 20-3 不同模式下的功能对比表

PWM输出功能			锯齿波	三角波	相关主要寄存器
独立 PWM 输出	端口状态控制	启动时	支持	支持	PCNA<B>R.STACA<B>
		停止时	支持	支持	PCNA<B>R.STPCA<B>
		上溢时	支持	支持	PCNA<B>R.OVFCA<B>
		下溢时	支持	支持	PCNA<B>R.UDFCA<B>
		计数匹配时 (Up Counting)	支持	支持	PCNA<B>R.CMA<B>UCA<B>
	缓存传送	计数匹配时 (Down Counting)	支持	支持	PCNA<B>R.CMA<B>DCA<B>
		周期基准值	单缓存	支持	PERBR->PERAR
			双缓存	支持	PERCR->PERBR PERBR->PERAR
		比较基准值	单缓存	支持	GCMDR->GCMBR GCMCR->GCMAR
			双缓存	支持	GCMFR->GCMDR GCMER->GCMCR GCMDR->GCMBR GCMCR->GCMAR
	EMB			支持	PCNA<B>R.EMBCA<B>
互补 PWM 输出	端口状态控制	启动时	支持	支持	PCNA<B>R.STACA<B>
		停止时	支持	支持	PCNA<B>R.STPCA<B>
		上溢时	支持	支持	PCNA<B>R.OVFCA<B>
		下溢时	支持	支持	PCNA<B>R.UDFCA<B>
		计数匹配时 (Up Counting)	支持	支持	PCNA<B>R.CMA<B>UCA<B>
		计数匹配时 (Down Counting)	支持	支持	PCNA<B>R.CMA<B>DCA<B>
	缓存传送	周期基准值	单缓存	支持	PERBR->PERAR
			双缓存	支持	PERCR->PERBR PERBR->PERAR
		比较基准值	单缓存	支持	GCMDR->GCMBR GCMCR->GCMAR
			双缓存	支持	GCMFR->GCMDR GCMER->GCMCR GCMDR->GCMBR GCMCR->GCMAR
		死区基准值	单缓存	支持	DTUBR->DTUAR DTDBR->DTDAR
	无死区PWM输出			支持	GCMAR=GCMBR
	带死区PWM输出	软件方式	支持	支持	GCMAR≠GCMBR
		硬件方式	不支持	支持	GCMBR=GCMAR-DTUAR GCNBR=GCMAR-DTDAR
	EMB			支持	PCNA<B>R.EMBCA<B>

## 20.4 中断及事件说明

### 20.4.1 中断输出

Timer6 含有 6 个通用计数比较匹配中断（含 2 个捕获输入中断）、2 个专用计数比较匹配中断、2 个计数周期匹配中断、1 个死区时间错误中断。

#### 20.4.1.1 计数比较匹配中断

通用比较基准值寄存器（GCMAR~GCMFR）共计 6 个，可分别与计数值比较产生比较匹配。计数比较匹配时，状态标志寄存器（STFLR）中的 STFLR.CMAF~STFLR.CMFF 位分别会被置为 1。此时若设定中断控制寄存器（ICONR）的 INTENA~INTENF 中相应位为 1 使能中断，则对应的中断请求（TMR6\_<t>\_GCMA~F）也会被触发。

在硬件捕获事件选择寄存器（HCPAR、HCPBR）选择的捕获输入有效条件产生时，捕获输入动作发生。此时若设置中断控制寄存器（ICONR）的 INTENA 或 INTENB 位为 1 使能中断，则对应的中断请求（TMR6\_<t>\_GCMA~B）被触发。

2 个专用比较基准值寄存器（SCMAR~SCMBR）也可分别与计数值比较产生比较匹配。计数比较匹配时，状态标志寄存器（STFLR）中的 STFLR.CMSPAF~CMSPBF 位分别会被置为 1。此时若设定中断控制寄存器（ICONR）的 INTENSAU<D>或 INTENSBU<D>中相应位为 1 使能中断，则对应的中断请求（TMR6\_<t>\_SCMA~B）也会被触发。

#### 20.4.1.2 计数周期匹配中断

锯齿波递加计数至上溢点、锯齿波递减计数至下溢点、三角波计数至谷点或三角波计数至峰点时，状态标志寄存器（STFLR）的 STFLR.OVFF 或 STFLR.UDFF 位会被置为 1。此时若设定中断控制寄存器（ICONR）的 ICONR.INTENOVF 或 ICONR.INTENUDF 位使能中断，则在对应的时间点可触发计数周期匹配中断（TMR6\_<t>\_GOVF 和 TMR6\_<t>\_GUDF）。

#### 20.4.1.3 死区时间错误中断

死区时间基准值寄存器（DTU<D>AR）的值加载到通用比较基准值寄存器（GCMBR）中时，若超过周期限制，则会产生死区时间错误，状态标志寄存器（STFLR）的 STFLR.DTEF 位会被置为 1。此时若设置中断控制寄存器（ICONR）的 INTENDTE 位使能中断，则会在该时刻触发死区时间错误中断（TMR6\_<t>\_GDTE）。

### 20.4.2 事件输出

在时钟计数过程中，若产生周期匹配事件（锯齿波的上溢点或下溢点、三角波的计数峰点或谷点）、通用计数比较匹配事件、专用计数比较匹配事件时，会产生相应的事件输出信号，用于选择触发别的模块，如 ADC、DMA 等。

下图是单元 1 的通用比较匹配中断 A~F&&事件 A~F、专用比较匹配中断 A~B&&事件 A~B、周期匹配中断&&事件的动作例。

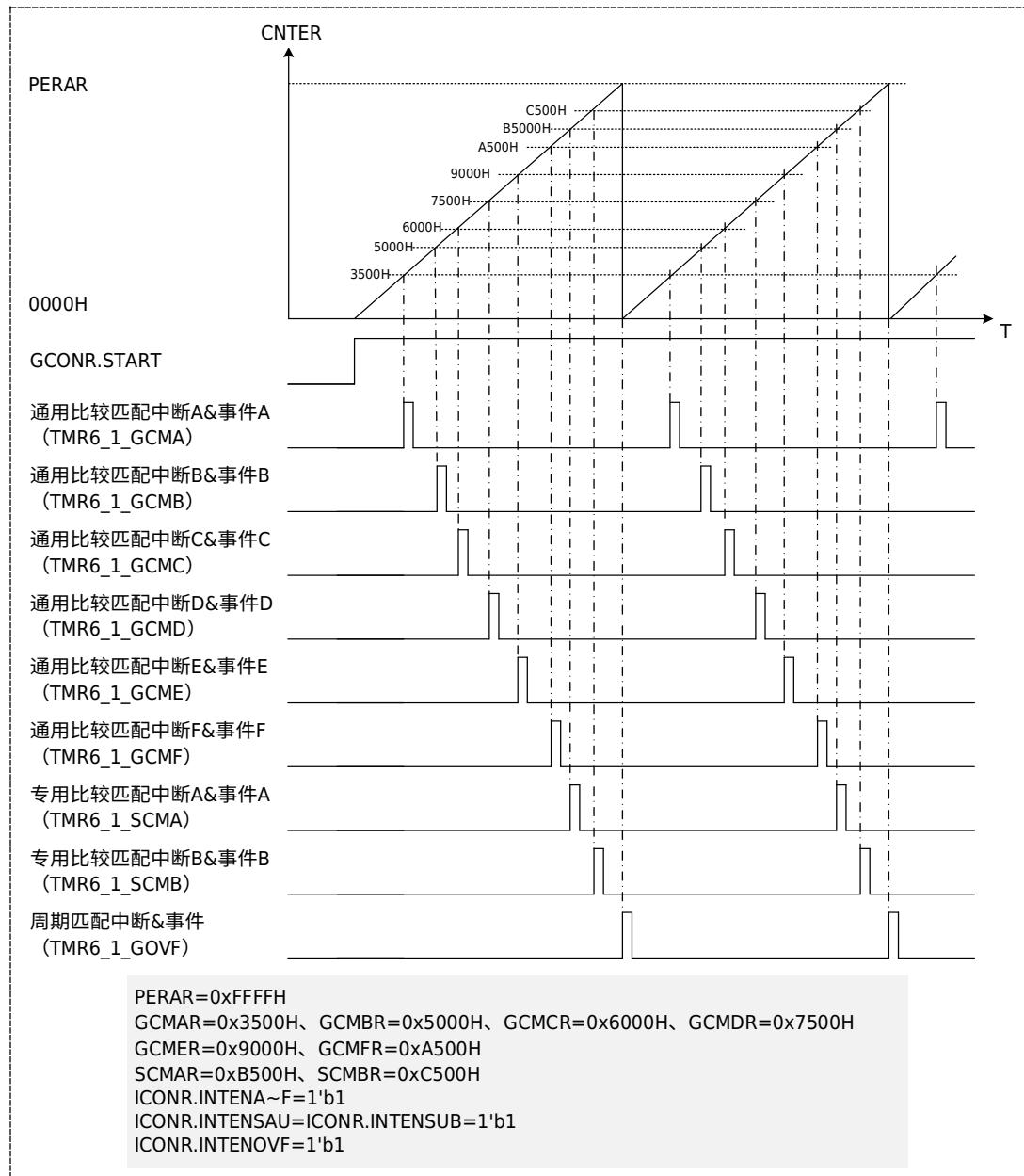


图 20-35 锯齿波模式时中断&事件输出例

## 20.5 寄存器说明

表 20-4 所示，为 Timer6 模块的寄存器列表。

表 20-4 Timer6 寄存器列表

BASE ADDR:

0x4003C000 (U1)、0x4003C400 (U2)、0x4003C800 (U3)、0x4003CC00 (U4)

0x4003D000 (U5)、0x4003D400 (U6)、0x4003D800 (U7)、0x4003DC00 (U8)

0x4003E000 (U9)、0x4003E400 (U10)

寄存器名	符号	偏移量	位宽	复位值
通用计数值寄存器	TMR6_CNTER	0x0000	32	0x0000 0000
通用刷新值寄存器	TMR6_UPDAR	0x0004	32	0x0000 0000
通用周期基准值寄存器A	TMR6_PERAR	0x0040	32	0x0000 FFFF
通用周期基准值寄存器B	TMR6_PERBR	0x0044	32	0x0000 FFFF
通用周期基准值寄存器C	TMR6_PERCR	0x0048	32	0x0000 FFFF
通用比较基准值寄存器A	TMR6_GCMAR	0x0080	32	0x0000 FFFF
通用比较基准值寄存器B	TMR6_GCMBR	0x0084	32	0x0000 FFFF
通用比较基准值寄存器C	TMR6_GCMCR	0x0088	32	0x0000 FFFF
通用比较基准值寄存器D	TMR6_GCMDR	0x008C	32	0x0000 FFFF
通用比较基准值寄存器E	TMR6_GCMER	0x0090	32	0x0000 FFFF
通用比较基准值寄存器F	TMR6_GCMFR	0x0094	32	0x0000 FFFF
专用比较基准值寄存器A	TMR6_SCMAR	0x00C0	32	0x0000 FFFF
专用比较基准值寄存器B	TMR6_SCMBR	0x00C4	32	0x0000 FFFF
专用比较基准值寄存器C	TMR6_SCMCR	0x00C8	32	0x0000 FFFF
专用比较基准值寄存器D	TMR6_SCMDR	0x00CC	32	0x0000 FFFF
专用比较基准值寄存器E	TMR6_SCMER	0x00D0	32	0x0000 FFFF
专用比较基准值寄存器F	TMR6_SCMFR	0x00D4	32	0x0000 FFFF
死区时间基准值寄存器UA	TMR6_DTUAR	0x0100	32	0x0000 FFFF
死区时间基准值寄存器DA	TMR6_DTDAR	0x0104	32	0x0000 FFFF
死区时间基准值寄存器UB	TMR6_DTUBR	0x0108	32	0x0000 FFFF
死区时间基准值寄存器DB	TMR6_DTDDBR	0x010C	32	0x0000 FFFF
通用控制寄存器	TMR6_GCONR	0x0140	32	0x0000 0002
中断控制寄存器	TMR6_ICONR	0x0144	32	0x0000 0000
缓存控制寄存器	TMR6_BCONR	0x0148	32	0x0000 0000
死区控制寄存器	TMR6_DCONR	0x014C	32	0x0000 0000
端口控制寄存器A	TMR6_PCNAR	0x0154	32	0x0000 0000
端口控制寄存器B	TMR6_PCNBR	0x0158	32	0x0000 0000
滤波控制寄存器G	TMR6_FCNGR	0x015C	32	0x0000 0000

寄存器名	符号	偏移量	位宽	复位值
有效周期寄存器	TMR6_VPERR	0x0160	32	0x0000 0000
状态标志寄存器	TMR6_STFLR	0x0164	32	0x8000 0000
硬件启动事件选择寄存器	TMR6_HSTAR	0x0180	32	0x0000 0000
硬件停止事件选择寄存器	TMR6_HSTPR	0x0184	32	0x0000 0000
硬件清零事件选择寄存器	TMR6_HCLRR	0x0188	32	0x0000 0000
硬件刷新事件选择寄存器	TMR6_HUPDR	0x018C	32	0x0000 0000
硬件捕获事件选择寄存器A	TMR6_HCPAR	0x0190	32	0x0000 0000
硬件捕获事件选择寄存器B	TMR6_HCPBR	0x0194	32	0x0000 0000
硬件递加事件选择寄存器	TMR6_HCUPR	0x0198	32	0x0000 0000
硬件递减事件选择寄存器	TMR6_HCDOR	0x019C	32	0x0000 0000
滤波控制寄存器T	TMR6_FCNTTR	U1_base+0x03EC	32	0x0000 0000
软件同步启动控制寄存器	TMR6_SSTAR	U1_base+0x03F0	32	0x0000 0000
软件同步停止控制寄存器	TMR6_SSTPR	U1_base+0x03F4	32	0x0000 0000
软件同步清零控制寄存器	TMR6_SCLRR	U1_base+0x03F8	32	0x0000 0000
软件同步刷新控制寄存器	TMR6_SUPDR	U1_base+0x03FC	32	0x0000 0000

**注意：**

- 触发选择寄存器 (TMR6\_TRGSEL0~3) 是 4 个独立于单元的寄存器，为 10 个单元 Timer6 所共有。
- 软件同步寄存器 (TMR6\_SSTAR、TMR6\_SSTPR、TMR6\_SCLLR、TMR6\_SUPDR) 是 4 个独立于单元的寄存器，为 10 个单元 Timer6 所共有。
- 滤波控制寄存器 (TMR6\_FCNTTR) 是 1 个独立于单元的寄存器，为 10 个单元 Timer6 所共有。

### 20.5.1 通用计数值寄存器 (TMR6\_CNTER)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为“0”	R											
b15~b0	CNT[15:0]	计数值	当前定时器的计数值	R/W											

### 20.5.2 通用刷新值寄存器 (TMR6\_UPDAR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UPDA[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为“0”	R											
b15~b0	UPDA[15:0]	刷新值	设定要更新进定时器的刷新值	R/W											

### 20.5.3 通用周期基准值寄存器 (TMR6\_PERmR) (m=A~C)

复位值: 0x0000FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PERA-C[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为“0”	R											
b15~b0	PERA-C[15:0]	计数周期值	设定每轮计数的计数周期值及对应缓存值	R/W											

### 20.5.4 通用比较基准值寄存器 (TMR6\_GCMmR) (m=A~F)

复位值: 0x0000FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GCMA-F[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为“0”	R											
b15~b0	GCMA-F[15:0]	计数比较基准值	比较基准值设定，与计数值相等时匹配信号有效	R/W											

### 20.5.5 专用比较基准值寄存器 (TMR6\_SCmR) (m=A~F)

复位值: 0x0000FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SCMA-F[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为“0”	R											
b15~b0	SCMA-F[15:0]	专用比较基准值	设定比较基准值及缓存值	R/W											

### 20.5.6 死区时间基准值寄存器 (TMR6\_DTmnR) (m=D、U&&n=A、B)

复位值: 0x0000FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DTUA-B[15:0]/DTDA-B[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为“0”	R											
b15~b0	DTU/DA-B[15:0]	死区时间值	死区时间设定值及缓存值	R/W											

## 20.5.7 通用控制寄存器 (TMR6\_GCONR)

复位值: 0x00000002

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved										ZMSK VAL[1:0]	ZMSK POS	ZMSK REV			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved										OV STP	CKDIV[3:0]	-	MODE	DIR	START
<hr/>															
位	标记	位名	功能	读写											
b31~b20	Reserved	-	读出时为“0”,写入时写“0”	R/W											
正交编码Z相输入被屏蔽的计数周期值															
b19~b18	ZMSKVAL[1:0]	Z相输入屏蔽周期数	00: Z相输入屏蔽功能无效 01: 位置计数上溢后或下溢后的4个计数周期内的Z相输入被屏蔽 10: 位置计数上溢后或下溢后的8个计数周期内的Z相输入被屏蔽 11: 位置计数上溢后或下溢后的16个计数周期内的Z相输入被屏蔽	R/W											
b17	ZMSKPOS	Z相输入位置定时器选择	0: Z相输入时该单元作为位置定时器, 在屏蔽周期期间内位置定时器清零功能正常动作 1: Z相输入时该单元作为位置定时器, 在屏蔽周期期间内位置定时器清零功能被屏蔽	R/W											
b16	ZMSKREV	Z相输入公转定时器选择	0: Z相输入时该单元作为公转定时器, 在屏蔽周期期间内公转定时器计数功能正常动作 1: Z相输入时该单元作为公转定时器, 在屏蔽周期期间内公转定时器计数功能被屏蔽	R/W											
b15~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b8	OVSTP	计数溢出停止控制	0: 计数器在计数上溢或下溢后, 继续计数 1: 计数器在计数上溢或下溢后, 停止计数	R/W											
0000: PCLK0 0001: PCLK0/2 0010: PCLK0/4 0011: PCLK0/8 0100: PCLK0/16															
b7~b4	CKDIV[2:0]	计数时钟选择	0101: PCLK0/32 0110: PCLK0/64 0111: PCLK0/128 1000: PCLK0/256 1001: PCLK0/512 1010: PCLK0/1024	R/W											
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b2	MODE	计数模式	0: 锯齿波模式 1: 三角波模式	R/W											
b1	DIR	计数方向	0: 递减计数 1: 递加计数	R/W											
b0	START	定时器启动	0: 定时器关闭 1: 定时器启动	R/W											
注: 该位在软件停止条件或硬件停止条件有效时, 会自动变为0															

## 20.5.8 中断控制寄存器 (TMR6\_ICONR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved												INTEN SBD	INTEN SBU	INTEN SAD	INTEN SAU
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved						INTEN DTE	INTEN UDF	INTEN OVF	INTEN F	INTEN E	INTEN D	INTEN C	INTEN B	INTEN A	
<hr/>															
位	标记	位名	功能	读写											
b31~b20	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b19	INTENSBD	专用向下计数中断使能B	0: 向下计数期间, SCMBR寄存器和计数值相等时, 该中断无效 1: 向下计数期间, SCMBR寄存器和计数值相等时, 该中断使能	R/W											
b18	INTENSBU	专用向上计数中断使能B	0: 向上计数期间, SCMBR寄存器和计数值相等时, 该中断无效 1: 向上计数期间, SCMBR寄存器和计数值相等时, 该中断使能	R/W											
b17	INTENSAD	专用向下计数中断使能A	0: 向下计数期间, SCMAR寄存器和计数值相等时, 该中断无效 1: 向下计数期间, SCMAR寄存器和计数值相等时, 该中断使能	R/W											
b16	INTENSAU	专用向上计数中断使能A	0: 向上计数期间, SCMAR寄存器和计数值相等时, 该中断无效 1: 向上计数期间, SCMAR寄存器和计数值相等时, 该中断使能	R/W											
b15~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b8	INTENDTE	死区时间错误中断使能	0: 死区时间错误时, 该中断无效 1: 死区时间错误时, 该中断使能	R/W											
b7	INTENUDF	下溢中断使能	0: 计数器计数下溢时, 该中断无效 1: 计数器计数下溢时, 该中断使能	R/W											
b6	INTENOVF	上溢中断使能	0: 计数器计数上溢时, 该中断无效 1: 计数器计数上溢时, 该中断使能	R/W											
b5	INTENF	计数匹配中断使能F	0: GCMFR寄存器与计数值相等时, 该中断无效 1: GCMFR寄存器与计数值相等时, 该中断使能	R/W											
b4	INTENE	计数匹配中断使能E	0: GCMER寄存器与计数值相等时, 该中断无效 1: GCMER寄存器与计数值相等时, 该中断使能	R/W											
b3	INTEND	计数匹配中断使能D	0: GCMDR寄存器与计数值相等时, 该中断无效 1: GCMDR寄存器与计数值相等时, 该中断使能	R/W											
b2	INTENC	计数匹配中断使能C	0: GCMCR寄存器与计数值相等时, 该中断无效 1: GCMCR寄存器与计数值相等时, 该中断使能	R/W											
b1	INTENB	计数匹配中断使能B	0: GCMBR寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断无效 1: GCMBR寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断使能	R/W											
b0	INTENA	计数匹配中断使能A	0: GCMAR寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断无效 1: GCMAR寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断使能	R/W											

## 20.5.9 缓存控制寄存器 (TMR6\_BCONR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								BTRD SPB	BTRU SPB	BSE SPB	BEN SPB	BTRD SPA	BTRU SPA	BSE SPA	BEN SPA
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	BTRD P	BTRU P	BSE P	BEN P	BTRD B	BTRU B	BSE B	BEN B	BTRD A	BTRU A	BSE A	BEN A

位	标记	位名	功能	读写
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	BTRDSPB	专用比较基准值缓存传送时间设定DB	0: 三角波计数模式计数到谷点时, 缓存值不传送 1: 三角波计数模式计数到谷点时, 发生一次缓存值传送 注: 锯齿波模式时, 不需要该位控制	R/W
b22	BTRUSPB	专用比较基准值缓存传送时间设定UB	0: 三角波计数模式计数到峰点时, 缓存值不传送 1: 三角波计数模式计数到峰点时, 发生一次缓存值传送 注: 锯齿波模式时, 不需要该位控制	R/W
b21	BSESPB	专用比较基准值缓存传送选择B	0: 单缓存传送 (SCMDR->SCMBR) 1: 双缓存传送 (SCMFR->SCMDR->SCMBR)	R/W
b20	BENSPB	专用比较基准值缓存传送B	0: 缓存传送无效 1: 缓存传送使能	R/W
b19	BTRDSPA	专用比较基准值缓存传送时间设定DA	0: 三角波计数模式计数到谷点时, 缓存值不传送 1: 三角波计数模式计数到谷点时, 发生一次缓存值传送 注: 锯齿波模式时, 不需要该位控制	R/W
b18	BTRUSPA	专用比较基准值缓存传送时间设定UA	0: 三角波计数模式计数到峰点时, 缓存值不传送 1: 三角波计数模式计数到峰点时, 发生一次缓存值传送 注: 锯齿波模式时, 不需要该位控制	R/W
b17	BSESPA	专用比较基准值缓存传送选择A	0: 单缓存传送 (SCMCR->SCMAR) 1: 双缓存传送 (SCMER->SCMCR->SCMAR)	R/W
b16	BENSPA	专用比较基准值缓存传送A	0: 缓存传送无效 1: 缓存传送使能	R/W
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	BTRDP	周期值缓存传送时间设定D	0: 三角波计数模式计数到谷点时, 缓存值不传送 1: 三角波计数模式计数到谷点时, 发生一次缓存值传送 注: 锯齿波模式时, 不需要该位控制	R/W
b10	BTRUP	周期值缓存传送时间设定U	0: 三角波计数模式计数到峰点时, 缓存值不传送 1: 三角波计数模式计数到峰点时, 发生一次缓存值传送 注: 锯齿波模式时, 不需要该位控制	R/W
b9	BSEP	周期值缓存传送选择	0: 单缓存传送 (PERBR->PERAR) 1: 双缓存传送 (PERCR->PERBR->PERAR) 注: 传送时间点跟计数模式无关, 只在锯齿波的上溢点、下溢点或三角波的波谷	R/W
b8	BENP	周期值缓存传送	0: 缓存传送无效 1: 缓存传送使能	R/W
b7	BTRDB	通用比较值缓存传送时间设定DB	0: 三角波计数模式计数到谷点时, 缓存值不传送 1: 三角波计数模式计数到谷点时, 发生一次缓存值传送 注: 锯齿波模式时, 不需要该位控制	R/W

b6	BTRUB	通用比较值缓存传送时间设定UB	0: 三角波计数模式计数到峰点时, 缓存值不传送 1: 三角波计数模式计数到峰点时, 发生一次缓存值传送 注: 锯齿波模式时, 不需要该位控制	R/W
b5	BSEB	通用比较值缓存传送选择B	比较输出功能时: 0: 单缓存传送 (GCMDR->GCMBR) 1: 双缓存传送 (GCMFR->GCMDR->GCMBR) 捕获输入功能时: 0: 单缓存传送 (GCMBR->GCMDR) 1: 双缓存传送 (GCMBR->GCMDR->GCMFR)	R/W
b4	BENB	通用比较值缓存传送B	0: 缓存传送无效 1: 缓存传送使能	R/W
b3	BTRDA	通用比较值缓存传送时间设定DA	0: 三角波计数模式计数到谷点时, 缓存值不传送 1: 三角波计数模式计数到谷点时, 发生一次缓存值传送 注: 锯齿波模式时, 不需要该位控制	R/W
b2	BTRUA	通用比较值缓存传送时间设定UA	0: 三角波计数模式计数到峰点时, 缓存值不传送 1: 三角波计数模式计数到峰点时, 发生一次缓存值传送 注: 锯齿波模式时, 不需要该位控制	R/W
b1	BSEA	通用比较值缓存传送选择A	比较输出功能时: 0: 单缓存传送 (GCMCR->GCMAR) 1: 双缓存传送 (GCMER->GCMCR->GCMAR) 捕获输入功能时: 0: 单缓存传送 (GCMAR->GCMCR) 1: 双缓存传送 (GCMAR->GCMCR->GCMER)	R/W
b0	BENA	通用比较值缓存传送A	0: 缓存传送无效 1: 缓存传送使能	R/W

## 20.5.10 死区控制寄存器 (TMR6\_DCONR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								DTB TRD	DTB TRU	DTB END	DTB ENU	-	-	SEPA	DTC EN
<hr/>															
位	标记	位名	功能	读写											
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b7	DTBTRD	死区时间值缓存 传送时间设定D	0: 三角波计数模式计数到谷点时, 死区缓存值不传送 1: 三角波计数模式计数到谷点时, 发生一次死区缓存值传送 注: 锯齿波模式时, 不需要该位控制	R/W											
b6	DTBTRU	死区时间值缓存 传送时间设定U	0: 三角波计数模式计数到峰点时, 死区缓存值不传送 1: 三角波计数模式计数到峰点时, 发生一次死区缓存值传送 注: 锯齿波模式时, 不需要该位控制	R/W											
b5	DTBEND	死区时间值缓存 传送D	0: 缓存传送无效 1: 缓存传送使能 (DTDBR->DTDAR)	R/W											
b4	DTBENU	死区时间值缓存 传送U	0: 缓存传送无效 1: 缓存传送使能 (DTUBR->DTUAR)	R/W											
b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b1	SEPA	分离设定	0: DTUAR和DTDAR分别设定 1: DTDAR的值和DTUAR的值自动相等	R/W											
b0	DTCEN	死区功能	0: 死区功能无效 1: 死区功能有效	R/W											

## 20.5.11 端口控制寄存器 (TMR6\_PCNAR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CAP MDA	-	-	OUT ENA	-		EMBS A[2:0]		EMBR A[1:0]		EMBC A[1:0]	-	-		FORC A[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMBDC A[1:0]		CMBUC A[1:0]		CMADC A[1:0]		CMAUC A[1:0]		UDFC A[1:0]		OVFC A[1:0]		STPC A[1:0]		STAC A[1:0]	

位	标记	位名	功能	读写
b31	CAPMDA	功能模式选择A	0: 比较输出功能 1: 捕获输入功能	R/W
b30~b29	Reserved	-	读出时为“0”,写入时写“0”	R/W
b28	OUTENA	输出使能A	0: Timer6功能时的TIM6_<t>_PWMA端口输出无效 1: Timer6功能时的TIM6_<t>_PWMA端口输出有效	R/W
b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26~b24	EMBSA[2:0]	EMB事件时端口状态选择A	000: 选择EMB的group0控制事件	
			001: 选择EMB的group1控制事件	
			010: 选择EMB的group2控制事件	
			011: 选择EMB的group3控制事件	R/W
			100: 选择EMB的group4控制事件	
			101: 选择EMB的group5控制事件	
b23~b22	EMBRA[1:0]	EMB释放方式选择A	00: EMBSA[2:0]选择的EMB group控制事件无效时, 立即释放 TIM6_<t>_PWMA端口 (One Shot)	
			01: EMBSA[2:0]选择的EMB group控制事件无效时, 等到计数器 计数到上溢时释放TIM6_<t>_PWMA端口 (Cycle By Cycle 1)	
			10: EMBSA[2:0]选择的EMB group控制事件无效时, 等到计数器 计数到下溢时释放TIM6_<t>_PWMA端口 (Cycle By Cycle 2)	R/W
			11: EMBSA[2:0]选择的EMB group控制事件无效时, 等到计数器 计数到上溢或下溢时释放TIM6_<t>_PWMA端口 (Cycle By Cycle 3)	
b21~b20	EMBCA[1:0]	EMB事件时端口状态设定A	00: 发生EMBSA[2:0]选择的EMB group控制事件时, TIM6_<t>_PWMA端口正常输出	
			01: 发生EMBSA[2:0]选择的EMB group控制事件时, TIM6_<t>_PWMA端口输出高阻态	
			10: 发生EMBSA[2:0]选择的EMB group控制事件时, TIM6_<t>_PWMA端口输出低电平	R/W
			11: 发生EMBSA[2:0]选择的EMB group控制事件时, TIM6_<t>_PWMA端口输出高电平	
b19~b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17~b16	FORCA[1:0]	强制端口状态设定A	0x: 设定无效定	
			10: 下周期开始, TIM6_<t>_PWMA端口输出设定为低电平	
			11: 下周期开始, TIM6_<t>_PWMA端口输出设定为高电平	
			注1: 下周期是指硬件计数模式或锯齿波计数到上溢点或下溢点、 三角波计数到谷点 注2: 该寄存器位可用于实现PWM输出占空比0%或100%的控制	R/W
b15~b14	CMBDCA[1:0]	向下计数&&比较值	00: 在向下计数期间, 定时器计数值与GCMR相等时,	R/W

		B匹配时端口状态设 定A	TIM6_<t>_PWMA端口输出设定为低电平 01: 在向下计数期间, 定时器计数值与GCMBR相等时, TIM6_<t>_PWMA端口输出设定为高电平 10: 在向下计数期间, 定时器计数值与GCMBR相等时, TIM6_<t>_PWMA端口输出保持先前状态 11: 在向下计数期间, 定时器计数值与GCMBR相等时, TIM6_<t>_PWMA端口输出设定为反转电平	
b13~b12	CMBUCA[1:0]	向上计数&&比较值 B匹配时端口状态设 定A	00: 在向上计数期间, 定时器计数值与GCMBR相等时, TIM6_<t>_PWMA端口输出设定为低电平 01: 在向上计数期间, 定时器计数值与GCMBR相等时, TIM6_<t>_PWMA端口输出设定为高电平 10: 在向上计数期间, 定时器计数值与GCMBR相等时, TIM6_<t>_PWMA端口输出保持先前状态 11: 在向上计数期间, 定时器计数值与GCMBR相等时, TIM6_<t>_PWMA端口输出设定为反转电平	R/W
b11~b10	CMADCA[1:0]	向下计数&&比较值 A匹配时端口状态设 定A	00: 在向下计数期间, 定时器计数值与GCMAR相等时, TIM6_<t>_PWMA端口输出设定为低电平 01: 在向下计数期间, 定时器计数值与GCMAR相等时, TIM6_<t>_PWMA端口输出设定为高电平 10: 在向下计数期间, 定时器计数值与GCMAR相等时, TIM6_<t>_PWMA端口输出保持先前状态 11: 在向下计数期间, 定时器计数值与GCMAR相等时, TIM6_<t>_PWMA端口输出设定为反转电平	R/W
b9~b8	CMAUCA[1:0]	向上计数&&比较值 A匹配时端口状态设 定A	00: 在向上计数期间, 定时器计数值与GCMAR相等时, TIM6_<t>_PWMA端口输出设定为低电平 01: 在向上计数期间, 定时器计数值与GCMAR相等时, TIM6_<t>_PWMA端口输出设定为高电平 10: 在向上计数期间, 定时器计数值与GCMAR相等时, TIM6_<t>_PWMA端口输出保持先前状态 11: 在向上计数期间, 定时器计数值与GCMAR相等时, TIM6_<t>_PWMA端口输出设定为反转电平	R/W
b7~b6	UDFCA[1:0]	计数下溢端口状态 设定A	00: 计数下溢时, TIM6_<t>_PWMA端口输出设定为低电平 01: 计数下溢时, TIM6_<t>_PWMA端口输出设定为高电平 10: 计数下溢时, TIM6_<t>_PWMA端口输出保持先前状态 11: 计数下溢时, TIM6_<t>_PWMA端口输出设定为反转电平	R/W
b5~b4	OVFCA[1:0]	计数上溢端口状态 设定A	00: 计数上溢时, TIM6_<t>_PWMA端口输出设定为低电平 01: 计数上溢时, TIM6_<t>_PWMA端口输出设定为高电平 10: 计数上溢时, TIM6_<t>_PWMA端口输出保持先前状态 11: 计数上溢时, TIM6_<t>_PWMA端口输出设定为反转电平	R/W
b3~b2	STPCA[1:0]	计数停止端口状态 设定A	00: 计数停止时, TIM6_<t>_PWMA端口输出设定为低电平 01: 计数停止时, TIM6_<t>_PWMA端口输出设定为高电平 10: 计数停止时, TIM6_<t>_PWMA端口输出保持先前状态 11: 计数停止时, TIM6_<t>_PWMA端口输出保持先前状态	R/W
b1~b0	STACA[1:0]	计数开始端口状态 设定A	00: 计数开始时, TIM6_<t>_PWMA端口输出设定为低电平 01: 计数开始时, TIM6_<t>_PWMA端口输出设定为高电平 10: 计数开始时, TIM6_<t>_PWMA端口输出保持先前状态 11: 计数开始时, TIM6_<t>_PWMA端口输出保持先前状态	R/W

## 20.5.12 端口控制寄存器 (TMR6\_PCNBR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CAPMDB	-	-	OUTENB	-		EMBSB[2:0]		EMBRB[1:0]		EMBCB[1:0]	-	-	FORCB[1:0]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMBDCB[1:0]		CMBUCB[1:0]		CMADC[1:0]		CMAUCB[1:0]		UDFCB[1:0]		OVFCB[1:0]		STPCB[1:0]		STACB[1:0]	

位	标记	位名	功能	读写
b31	CAPMDB	功能模式选择B	0: 比较输出功能 1: 捕获输入功能	R/W
b30~b29	Reserved	-	读出时为“0”,写入时写“0”	R/W
b28	OUTENB	输出使能B	0: Timer6功能时的TIM6_<t>_PWMB端口输出无效 1: Timer6功能时的TIM6_<t>_PWMB端口输出有效	R/W
b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26~b24	EMBSB[2:0]	EMB事件时端口状态选择B	000: 选择EMB的group0控制事件 001: 选择EMB的group1控制事件 010: 选择EMB的group2控制事件 011: 选择EMB的group3控制事件 100: 选择EMB的group4控制事件 101: 选择EMB的group5控制事件	R/W
			00: EMBSB[2:0]选择的EMB group控制事件无效时, 立即释放TIM6_<t>_PWMB端口 (One Shot)	
			01: EMBSB[2:0]选择的EMB group控制事件无效时, 等到计数器计数到上溢时释放TIM6_<t>_PWMB端口 (Cycle By Cycle 1)	
			10: EMBSB[2:0]选择的EMB group控制事件无效时, 等到计数器计数到下溢时释放TIM6_<t>_PWMB端口 (Cycle By Cycle 2)	
			11: EMBSB[2:0]选择的EMB group控制事件无效时, 等到计数器计数到上溢或下溢时释放TIM6_<t>_PWMB端口 (Cycle By Cycle 3)	
			00: 发生EMBSB[2:0]选择的EMB group控制事件时, TIM6_<t>_PWMB端口正常输出 01: 发生EMBSB[2:0]选择的EMB group控制事件时, TIM6_<t>_PWMB端口输出高阻态	
b21~b20	EMBCB[1:0]	EMB事件时端口状态设定B	10: 发生EMBSB[2:0]选择的EMB group控制事件时, TIM6_<t>_PWMB端口输出低电平 11: 发生EMBSB[2:0]选择的EMB group控制事件时, TIM6_<t>_PWMB端口输出高电平	R/W
			00: 设定无效定	
			10: 下周期开始, TIM6_<t>_PWMB端口输出设定为低电平 11: 下周期开始, TIM6_<t>_PWMB端口输出设定为高电平	
b17~b16	FORCB[1:0]	强制端口状态设定B	注1: 下周期是指硬件计数模式或锯齿波计数到上溢点或下溢点、三角波计数到谷点 注2: 该寄存器位可用于实现PWM输出占空比0%或100%的控制	R/W
			00: 在向下计数期间, 定时器计数值与GCMBR相等时,	
b15~b14	CMBDCB[1:0]	向下计数&&比较值		R/W

		B匹配时端口状态设 定B	TIM6_<t>_PWMB端口输出设定为低电平 01: 在向下计数期间, 定时器计数值与GCMBR相等时, TIM6_<t>_PWMB端口输出设定为高电平 10: 在向下计数期间, 定时器计数值与GCMBR相等时, TIM6_<t>_PWMB端口输出保持先前状态 11: 在向下计数期间, 定时器计数值与GCMBR相等时, TIM6_<t>_PWMB端口输出设定为反转电平	
b13~b12	CMBUCB[1:0]	向上计数&&比较值 B匹配时端口状态设 定B	00: 在向上计数期间, 定时器计数值与GCMBR相等时, TIM6_<t>_PWMB端口输出设定为低电平 01: 在向上计数期间, 定时器计数值与GCMBR相等时, TIM6_<t>_PWMB端口输出设定为高电平 10: 在向上计数期间, 定时器计数值与GCMBR相等时, TIM6_<t>_PWMB端口输出保持先前状态 11: 在向上计数期间, 定时器计数值与GCMBR相等时, TIM6_<t>_PWMB端口输出设定为反转电平	R/W
b11~b10	CMADCB[1:0]	向下计数&&比较值 A匹配时端口状态设 定B	00: 在向下计数期间, 定时器计数值与GCMAR相等时, TIM6_<t>_PWMB端口输出设定为低电平 01: 在向下计数期间, 定时器计数值与GCMAR相等时, TIM6_<t>_PWMB端口输出设定为高电平 10: 在向下计数期间, 定时器计数值与GCMAR相等时, TIM6_<t>_PWMB端口输出保持先前状态 11: 在向下计数期间, 定时器计数值与GCMAR相等时, TIM6_<t>_PWMB端口输出设定为反转电平	R/W
b9~b8	CMAUCB[1:0]	向上计数&&比较值 A匹配时端口状态设 定B	00: 在向上计数期间, 定时器计数值与GCMAR相等时, TIM6_<t>_PWMB端口输出设定为低电平 01: 在向上计数期间, 定时器计数值与GCMAR相等时, TIM6_<t>_PWMB端口输出设定为高电平 10: 在向上计数期间, 定时器计数值与GCMAR相等时, TIM6_<t>_PWMB端口输出保持先前状态 11: 在向上计数期间, 定时器计数值与GCMAR相等时, TIM6_<t>_PWMB端口输出设定为反转电平	R/W
b7~b6	UDFCB[1:0]	计数下溢端口状态 设定B	00: 计数下溢时, TIM6_<t>_PWMB端口输出设定为低电平 01: 计数下溢时, TIM6_<t>_PWMB端口输出设定为高电平 10: 计数下溢时, TIM6_<t>_PWMB端口输出保持先前状态 11: 计数下溢时, TIM6_<t>_PWMB端口输出设定为反转电平	R/W
b5~b4	OVFCB[1:0]	计数上溢端口状态 设定B	00: 计数上溢时, TIM6_<t>_PWMB端口输出设定为低电平 01: 计数上溢时, TIM6_<t>_PWMB端口输出设定为高电平 10: 计数上溢时, TIM6_<t>_PWMB端口输出保持先前状态 11: 计数上溢时, TIM6_<t>_PWMB端口输出设定为反转电平	R/W
b3~b2	STPCB[1:0]	计数停止端口状态 设定B	00: 计数停止时, TIM6_<t>_PWMB端口输出设定为低电平 01: 计数停止时, TIM6_<t>_PWMB端口输出设定为高电平 10: 计数停止时, TIM6_<t>_PWMB端口输出保持先前状态 11: 计数停止时, TIM6_<t>_PWMB端口输出保持先前状态	R/W
b1~b0	STBCB[1:0]	计数开始端口状态 设定B	00: 计数开始时, TIM6_<t>_PWMB端口输出设定为低电平 01: 计数开始时, TIM6_<t>_PWMB端口输出设定为高电平 10: 计数开始时, TIM6_<t>_PWMB端口输出保持先前状态 11: 计数开始时, TIM6_<t>_PWMB端口输出保持先前状态	R/W

## 20.5.13 滤波控制寄存器 (TMR6\_FCNGR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
位	标记	位名	功能	读写											
b31~b7	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b6~b5	NOFICKGB[1:0]	滤波采样基准时钟选择GB	00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64	R/W											
b4	NOFIENGB	捕获输入端口滤波GB	0: 该单元TIM6_<t>_PWMB输入端口滤波功能无效 1: 该单元TIM6_<t>_PWMB输入端口滤波功能使能	R/W											
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b2~b1	NOFICKGA[1:0]	滤波采样基准时钟选择GA	00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64	R/W											
b0	NOFIENGA	捕获输入端口滤波GA	0: 该单元TIM6_<t>_PWMA输入端口滤波功能无效 1: 该单元TIM6_<t>_PWMA输入端口滤波功能使能	R/W											

## 20.5.14 滤波控制寄存器 (TMR6\_FCNTR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	NOFI CKTD[1:0]	NOFI ENTD	-	NOFI CKTC[1:0]	NOFI ENTC	-	NOFI CKTB[1:0]	NOFI ENTB	-	NOFI CKTA[1:0]	NOFI ENTA				
位	标记	位名	功能	读写											
b31~b15	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b14~b13	NOFICKTD[1:0]	滤波采样基准时钟选择TD	00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64	R/W											
b12	NOFIENTD	捕获输入端口滤波TD	0: TIM6_TRIGD端口输入滤波功能无效 1: TIM6_TRIGD端口输入滤波功能使能	R/W											
b11	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b10~b9	NOFICKTC[1:0]	滤波采样基准时钟选择TC	00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64	R/W											
b8	NOFIENTC	捕获输入端口滤波TC	0: TIM6_TRIGC端口输入滤波功能无效 1: TIM6_TRIGC端口输入滤波功能使能	R/W											
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b6~b5	NOFICKTB[1:0]	滤波采样基准时钟选择TB	00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64	R/W											
b4	NOFIENTB	捕获输入端口滤波TB	0: TIM6_TRIGB端口输入滤波功能无效 1: TIM6_TRIGB端口输入滤波功能使能	R/W											
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b2~b1	NOFICKTA[1:0]	滤波采样基准时钟选择TA	00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64	R/W											
b0	NOFIENTA	捕获输入端口滤波TA	0: TIM6_TRIGA端口输入滤波功能无效 1: TIM6_TRIGA端口输入滤波功能使能	R/W											

## 20.5.15 有效周期寄存器 (TMR6\_VPERR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Reserved										PCNTS[2:0]		PCNTE[1:0]				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Reserved					SP PERIB	SP PERIA	Reserved									

位	标记	位名	功能	读写
b31~b21	Reserved	-	读出时为“0”,写入时写“0”	R/W
b20~b18	PCNTS[2:0]	有效周期选择	000: 有效周期选择功能无效 001: 每隔1个周期有效一次 010: 每隔2个周期有效一次 011: 每隔3个周期有效一次 100: 每隔4个周期有效一次 101: 每隔5个周期有效一次 110: 每隔6个周期有效一次 111: 每隔7个周期有效一次	R/W
b17~b16	PCNTE[1:0]	有效周期计数条件选择	00: 有效周期选择功能无效 01: 锯齿波计数上、下溢点或三角波波谷做为计数条件 10: 锯齿波计数上、下溢点或三角波波峰做为计数条件 11: 锯齿波计数上、下溢点或三角波波谷、波峰做为计数条件	R/W
b15~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9	SPPERIB	专用信号有效周期选择B	0: 有效周期选择功能无效 1: 有效周期选择功能使能	R/W
b8	SPPERIA	专用信号有效周期选择A	0: 有效周期选择功能无效 1: 有效周期选择功能使能	R/W
b7~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

## 20.5.16 状态标志寄存器 (TMR6\_STFLR)

复位值: 0x80000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DIRF	-	-	-	CMBF2	CMAF2	-	-	VPERNUM[2:0]	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	CMSBDF	CMSBUF	CMSADF	CMSAUF	DTEF	UDFF	OVFF	CMFF	CMEF	CMDF	CMCF	CMBF	CMAF

位	标记	位名	功能	读写
b31	DIRF	计数方向	0: 递减计数 1: 递加计数	R
b30~b28	Reserved	-	读出时为“0”,写入时写“0”	R
b27	CMBF2	再次捕获B	0: 计数匹配B标志CMBF为1后未发生TIM6_<t>_PWMB捕获完成 动作 1: 计数匹配B标志CMBF为1后再次发生TIM6_<t>_PWMB捕获完成动作	R/W
b26	CMAF2	再次捕获A	0: 计数匹配A标志CMAF为1后未发生TIM6_<t>_PWMA捕获完成 动作 1: 计数匹配A标志CMAF为1后再次发生TIM6_<t>_PWMA捕获完成动作	R/W
b25~24	Reserved	-	读出时为“0”,写入时写“0”	R
b23~b21	VPERNUM[2:0]	周期次数	有效周期选择功能使能时, 计数后的周期次数	R
b20~b13	Reserved	-	读出时为“0”,写入时写“0”	R
b12	CMSBDF	向下计数专用比较基准值匹配B	0: 向下计数时, SCMBR寄存器的值与计数值不相等 1: 向下计数时, SCMBR寄存器的值与计数值相等	R/W
b11	CMSBUF	向上计数专用比较基准值匹配B	0: 向上计数时, SCMBR寄存器的值与计数值不相等 1: 向上计数时, SCMBR寄存器的值与计数值相等	R/W
b10	CMSADF	向下计数专用比较基准值匹配A	0: 向下计数时, SCMAR寄存器的值与计数值不相等 1: 向下计数时, SCMAR寄存器的值与计数值相等	R/W
b09	CMSAUF	向上计数专用比较基准值匹配A	0: 向上计数时, SCMAR寄存器的值与计数值不相等 1: 向上计数时, SCMAR寄存器的值与计数值相等	R/W
b8	DTEF	死区时间错误	0: 未发生死区时间错误 1: 发生死区时间错误	R
b7	UDFF	下溢匹配	0: 未发生锯齿波下溢或三角波计数到谷点 1: 发生锯齿波下溢或三角波计数到谷点	R/W
b6	OVFF	上溢匹配	0: 未发生锯齿波上溢或三角波计数到峰点 1: 发生锯齿波上溢或三角波计数到峰点	R/W
b5	CMFF	计数匹配F	0: GCMFR寄存器的值与计数值不相等 1: GCMFR寄存器的值与计数值相等	R/W
b4	CMEF	计数匹配E	0: GCMER寄存器的值与计数值不相等 1: GCMER寄存器的值与计数值相等	R/W
b3	CMDF	计数匹配D	0: GCMDR寄存器的值与计数值不相等 1: GCMDR寄存器的值与计数值相等	R/W
b2	CMCF	计数匹配C	0: GCMCR寄存器的值与计数值不相等 1: GCMCR寄存器的值与计数值相等	R/W

			0: GCMBR寄存器的值与计数值不相等，且未发生 TIM6_<t>_PWMB捕获完成动作 1: GCMBR寄存器的值与计数值相等，或发生TIM6_<t>_PWMB 捕获完成动作	R/W
b1	CMBF	计数匹配B	0: GCMAR寄存器的值与计数值不相等，且未发生 TIM6_<t>_PWMA捕获完成动作 1: GCMAR寄存器的值与计数值相等，或发生TIM6_<t>_PWMA 捕获完成动作	R/W
b0	CMAF	计数匹配A		

## 20.5.17 硬件启动事件选择寄存器 (TMR6\_HSTAR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
								HSTA 23	HSTA 22	HSTA 21	HSTA 20	HSTA 19	HSTA 18	HSTA 17	HSTA 16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HSTA 11	HSTA 10	HSTA 9	HSTA 8	STAS	-	-	-	HSTA 3	HSTA 2	HSTA 1	HSTA 0
<hr/>															
位	标记	位名	功能	读写											
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b23	HSTA23	硬件启动条件23	条件: TIM6_TRIGD端口上采样到下降沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W											
b22	HSTA22	硬件启动条件22	条件: TIM6_TRIGD端口上采样到上升沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W											
b21	HSTA21	硬件启动条件21	条件: TIM6_TRIGC端口上采样到下降沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W											
b20	HSTA20	硬件启动条件20	条件: TIM6_TRIGC端口上采样到上升沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W											
b19	HSTA19	硬件启动条件19	条件: TIM6_TRIGB端口上采样到下降沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W											
b18	HSTA18	硬件启动条件18	条件: TIM6_TRIGB端口上采样到上升沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W											
b17	HSTA17	硬件启动条件17	条件: TIM6_TRIGA端口上采样到下降沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W											
b16	HSTA16	硬件启动条件16	条件: TIM6_TRIGA端口上采样到上升沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W											
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b11	HSTA11	硬件启动条件11	条件: 内部硬件触发事件3有效 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W											
b10	HSTA10	硬件启动条件10	条件: 内部硬件触发事件2有效 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W											
b9	HSTA9	硬件启动条件9	条件: 内部硬件触发事件1有效 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W											
b8	HSTA8	硬件启动条件8	条件: 内部硬件触发事件0有效 0: 条件匹配时, 硬件启动无效	R/W											

1: 条件匹配时，硬件启动有效				
b7	STAS	硬件启动使能	0: 硬件启动无效 1: 硬件启动有效  注：硬件启动有效时，SSTAR的设定无效	R/W
b6~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	HSTA3	硬件启动条件3	条件：TIM6_<t>_PWMB端口上采样到下降沿  0: 条件匹配时，硬件启动无效 1: 条件匹配时，硬件启动有效	R/W
b2	HSTA2	硬件启动条件2	条件：TIM6_<t>_PWMB端口上采样到上升沿  0: 条件匹配时，硬件启动无效 1: 条件匹配时，硬件启动有效	R/W
b1	HSTA1	硬件启动条件1	条件：TIM6_<t>_PWMA端口上采样到下降沿  0: 条件匹配时，硬件启动无效 1: 条件匹配时，硬件启动有效	R/W
b0	HSTA0	硬件启动条件0	条件：TIM6_<t>_PWMA端口上采样到上升沿  0: 条件匹配时，硬件启动无效 1: 条件匹配时，硬件启动有效	R/W

**20.5.18 硬件停止事件选择寄存器 (TMR6\_HSTPR)**

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
								HSTP 23	HSTP 22	HSTP 21	HSTP 20	HSTP 19	HSTP 18	HSTP 17	HSTP 16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HSTP 11	HSTP 10	HSTP 9	HSTP 8	STPS	-	-	-	HSTP 3	HSTP 2	HSTP 1	HSTP 0

位	标记	位名	功能	读写
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	HSTP23	硬件停止条件23	条件: TIM6_TRIGD端口上采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b22	HSTP22	硬件停止条件22	条件: TIM6_TRIGD端口上采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b21	HSTP21	硬件停止条件21	条件: TIM6_TRIGC端口上采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b20	HSTP20	硬件停止条件20	条件: TIM6_TRIGC端口上采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b19	HSTP19	硬件停止条件19	条件: TIM6_TRIGB端口上采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b18	HSTP18	硬件停止条件18	条件: TIM6_TRIGB端口上采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b17	HSTP17	硬件停止条件17	条件: TIM6_TRIGA端口上采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b16	HSTP16	硬件停止条件16	条件: TIM6_TRIGA端口上采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	HSTP11	硬件停止条件11	条件: 内部硬件触发事件3有效 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b10	HSTP10	硬件停止条件10	条件: 内部硬件触发事件2有效 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b9	HSTP9	硬件停止条件9	条件: 内部硬件触发事件1有效 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b8	HSTP8	硬件停止条件8	条件: 内部硬件触发事件0有效 0: 条件匹配时, 硬件停止无效	R/W

			1: 条件匹配时, 硬件停止有效 0: 硬件停止无效	
b7	STPS	硬件停止使能	1: 硬件停止有效 注: 硬件停止有效时, SSTPR的设定无效	R/W
b6~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	HSTP3	硬件停止条件3	条件: TIM6_<t>_PWMB端口上采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b2	HSTP2	硬件停止条件2	条件: TIM6_<t>_PWMB端口上采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b1	HSTP1	硬件停止条件1	条件: TIM6_<t>_PWMA端口上采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b0	HSTP0	硬件停止条件0	条件: TIM6_<t>_PWMA端口上采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W

## 20.5.19 硬件清零事件选择寄存器 (TMR6\_HCLRR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HCLE 23	HCLE 22	HCLE 21	HCLE 20	HCLE 19	HCLE 18	HCLE 17	HCLE 16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HCLE 11	HCLE 10	HCLE 9	HCLE 8	CLES	-	-	-	HCLE 3	HCLE 2	HCLE 1	HCLE 0
<hr/>															
位	标记	位名	功能	读写											
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b23	HCLE23	硬件清零条件23	条件: TIM6_TRIGD端口上采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W											
b22	HCLE22	硬件清零条件22	条件: TIM6_TRIGD端口上采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W											
b21	HCLE21	硬件清零条件21	条件: TIM6_TRIGC端口上采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W											
b20	HCLE20	硬件清零条件20	条件: TIM6_TRIGC端口上采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W											
b19	HCLE19	硬件清零条件19	条件: TIM6_TRIGB端口上采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W											
b18	HCLE18	硬件清零条件18	条件: TIM6_TRIGB端口上采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W											
b17	HCLE17	硬件清零条件17	条件: TIM6_TRIGA端口上采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W											
b16	HCLE16	硬件清零条件16	条件: TIM6_TRIGA端口上采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W											
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b11	HCLE11	硬件清零条件11	条件: 内部硬件触发事件3有效 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W											
b10	HCLE10	硬件清零条件10	条件: 内部硬件触发事件2有效 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W											
b9	HCLE9	硬件清零条件9	条件: 内部硬件触发事件1有效 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W											
b8	HCLE8	硬件清零条件8	条件: 内部硬件触发事件0有效 0: 条件匹配时, 硬件清零无效	R/W											

			1: 条件匹配时, 硬件清零有效	
b7	CLES	硬件清零使能	0: 硬件清零无效 1: 硬件清零有效	R/W
			注: 硬件清零有效时, SCLR的设定无效	
b6~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	HCLE3	硬件清零条件3	条件: TIM6_<t>_PWMB端口上采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b2	HCLE2	硬件清零条件2	条件: TIM6_<t>_PWMB端口上采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b1	HCLE1	硬件清零条件1	条件: TIM6_<t>_PWMA端口上采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b0	HCLE0	硬件清零条件0	条件: TIM6_<t>_PWMA端口上采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W

## 20.5.20 硬件刷新事件选择寄存器 (TMR6\_HUPDR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HUPD23	HUPD22	HUPD21	HUPD20	HUPD19	HUPD18	HUPD17	HUPD16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HUPD11	HUPD10	HUPD9	HUPD8	UPDS	-	-	-	HUPD3	HUPD2	HUPD1	HUPD0
<hr/>															
位	标记	位名	功能	读写											
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b23	HUPD23	硬件刷新条件23	条件: TIM6_TRIGD端口上采样到下降沿 0: 条件匹配时, 硬件刷新无效 1: 条件匹配时, 硬件刷新有效	R/W											
b22	HUPD22	硬件刷新条件22	条件: TIM6_TRIGC端口上采样到上升沿 0: 条件匹配时, 硬件刷新无效 1: 条件匹配时, 硬件刷新有效	R/W											
b21	HUPD21	硬件刷新条件21	条件: TIM6_TRIGC端口上采样到下降沿 0: 条件匹配时, 硬件刷新无效 1: 条件匹配时, 硬件刷新有效	R/W											
b20	HUPD20	硬件刷新条件20	条件: TIM6_TRIGB端口上采样到上升沿 0: 条件匹配时, 硬件刷新无效 1: 条件匹配时, 硬件刷新有效	R/W											
b19	HUPD19	硬件刷新条件19	条件: TIM6_TRIGB端口上采样到下降沿 0: 条件匹配时, 硬件刷新无效 1: 条件匹配时, 硬件刷新有效	R/W											
b18	HUPD18	硬件刷新条件18	条件: TIM6_TRIGB端口上采样到上升沿 0: 条件匹配时, 硬件刷新无效 1: 条件匹配时, 硬件刷新有效	R/W											
b17	HUPD17	硬件刷新条件17	条件: TIM6_TRIGA端口上采样到下降沿 0: 条件匹配时, 硬件刷新无效 1: 条件匹配时, 硬件刷新有效	R/W											
b16	HUPD16	硬件刷新条件16	条件: TIM6_TRIGA端口上采样到上升沿 0: 条件匹配时, 硬件刷新无效 1: 条件匹配时, 硬件刷新有效	R/W											
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b11	HUPD11	硬件刷新条件11	条件: 内部硬件触发事件3有效 0: 条件匹配时, 硬件刷新无效 1: 条件匹配时, 硬件刷新有效	R/W											
b10	HUPD10	硬件刷新条件10	条件: 内部硬件触发事件2有效 0: 条件匹配时, 硬件刷新无效 1: 条件匹配时, 硬件刷新有效	R/W											
b9	HUPD9	硬件刷新条件9	条件: 内部硬件触发事件1有效 0: 条件匹配时, 硬件刷新无效 1: 条件匹配时, 硬件刷新有效	R/W											
b8	HUPD8	硬件刷新条件8	条件: 内部硬件触发事件0有效 0: 条件匹配时, 硬件刷新无效	R/W											

			1: 条件匹配时, 硬件刷新有效 0: 硬件刷新无效	
b7	UPDS	硬件刷新使能	1: 硬件刷新有效 注: 硬件刷新有效时, SUPDR的设定无效	R/W
b6~b4	Reserved	-	读出时为“0”,写入时写“0” 条件: TIM6_<t>_PWMB端口上采样到下降沿	R/W
b3	HUPD3	硬件刷新条件3	0: 条件匹配时, 硬件刷新无效 1: 条件匹配时, 硬件刷新有效	R/W
b2	HUPD2	硬件刷新条件2	条件: TIM6_<t>_PWMB端口上采样到上升沿 0: 条件匹配时, 硬件刷新无效 1: 条件匹配时, 硬件刷新有效	R/W
b1	HUPD1	硬件刷新条件1	条件: TIM6_<t>_PWMA端口上采样到下降沿 0: 条件匹配时, 硬件刷新无效 1: 条件匹配时, 硬件刷新有效	R/W
b0	HUPD0	硬件刷新条件0	条件: TIM6_<t>_PWMA端口上采样到上升沿 0: 条件匹配时, 硬件刷新无效 1: 条件匹配时, 硬件刷新有效	R/W

## 20.5.21 硬件捕获事件选择寄存器 A (TMR6\_HCPAR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HCP A23	HCP A22	HCP A21	HCP A20	HCP A19	HCP A18	HCP A17	HCP A16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HCP A11	HCP A10	HCP A9	HCP A8	-	-	-	-	HCP A3	HCP A2	HCP A1	HCP A0
<hr/>															
位	标记	位名	功能	读写											
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b23	HCPA23	硬件捕获A条件23	条件: TIM6_TRIGD端口上采样到下降沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W											
b22	HCPA22	硬件捕获A条件22	条件: TIM6_TRIGC端口上采样到上升沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W											
b21	HCPA21	硬件捕获A条件21	条件: TIM6_TRIGC端口上采样到下降沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W											
b20	HCPA20	硬件捕获A条件20	条件: TIM6_TRIGB端口上采样到上升沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W											
b19	HCPA19	硬件捕获A条件19	条件: TIM6_TRIGB端口上采样到下降沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W											
b18	HCPA18	硬件捕获A条件18	条件: TIM6_TRIGB端口上采样到上升沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W											
b17	HCPA17	硬件捕获A条件17	条件: TIM6_TRIGA端口上采样到下降沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W											
b16	HCPA16	硬件捕获A条件16	条件: TIM6_TRIGA端口上采样到上升沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W											
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b11	HCPA11	硬件捕获A条件11	条件: 内部硬件触发事件3有效 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W											
b10	HCPA10	硬件捕获A条件10	条件: 内部硬件触发事件2有效 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W											
b9	HCPA9	硬件捕获A条件9	条件: 内部硬件触发事件1有效 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W											
b8	HCPA8	硬件捕获A条件8	条件: 内部硬件触发事件0有效 0: 条件匹配时, 硬件捕获A无效	R/W											

1: 条件匹配时，硬件捕获A有效				
b7~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	HCPA3	硬件捕获A条件3	条件：TIM6_<t>_PWMB端口上采样到下降沿 0: 条件匹配时，硬件捕获A无效 1: 条件匹配时，硬件捕获A有效	R/W
b2	HCPA2	硬件捕获A条件2	条件：TIM6_<t>_PWMA端口上采样到上升沿 0: 条件匹配时，硬件捕获A无效 1: 条件匹配时，硬件捕获A有效	R/W
b1	HCPA1	硬件捕获A条件1	条件：TIM6_<t>_PWMA端口上采样到下降沿 0: 条件匹配时，硬件捕获A无效 1: 条件匹配时，硬件捕获A有效	R/W
b0	HCPA0	硬件捕获A条件0	条件：TIM6_<t>_PWMA端口上采样到上升沿 0: 条件匹配时，硬件捕获A无效 1: 条件匹配时，硬件捕获A有效	R/W

## 20.5.22 硬件捕获事件选择寄存器 B (TMR6\_HCPBR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HCP B23	HCP B22	HCP B21	HCP B20	HCP B19	HCP B18	HCP B17	HCP B16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HCP B11	HCP B10	HCP B9	HCP B8	-	-	-	-	HCP B3	HCP B2	HCP B1	HCP B0
<hr/>															
位	标记	位名	功能	读写											
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b23	HCPB23	硬件捕获B条件23	条件: TIM6_TRIGD端口上采样到下降沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W											
b22	HCPB22	硬件捕获B条件22	条件: TIM6_TRIGD端口上采样到上升沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W											
b21	HCPB21	硬件捕获B条件21	条件: TIM6_TRIGC端口上采样到下降沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W											
b20	HCPB20	硬件捕获B条件20	条件: TIM6_TRIGC端口上采样到上升沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W											
b19	HCPB19	硬件捕获B条件19	条件: TIM6_TRIGB端口上采样到下降沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W											
b18	HCPB18	硬件捕获B条件18	条件: TIM6_TRIGB端口上采样到上升沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W											
b17	HCPB17	硬件捕获B条件17	条件: TIM6_TRIGA端口上采样到下降沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W											
b16	HCPB16	硬件捕获B条件16	条件: TIM6_TRIGA端口上采样到上升沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W											
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b11	HCPB11	硬件捕获B条件11	条件: 内部硬件触发事件3有效 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W											
b10	HCPB10	硬件捕获B条件10	条件: 内部硬件触发事件2有效 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W											
b9	HCPB9	硬件捕获B条件9	条件: 内部硬件触发事件1有效 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W											
b8	HCPB8	硬件捕获B条件8	条件: 内部硬件触发事件0有效 0: 条件匹配时, 硬件捕获B无效	R/W											

			1: 条件匹配时, 硬件捕获B有效	
b7~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	HCPB3	硬件捕获B条件3	条件: TIM6_<t>_PWMB端口上采样到下降沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W
b2	HCPB2	硬件捕获B条件2	条件: TIM6_<t>_PWMB端口上采样到上升沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W
b1	HCPB1	硬件捕获B条件1	条件: TIM6_<t>_PWMA端口上采样到下降沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W
b0	HCPB0	硬件捕获B条件0	条件: TIM6_<t>_PWMA端口上采样到上升沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W

## 20.5.23 硬件递加事件选择寄存器 (TMR6\_HCUPR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
					Reserved			HC UP23	HC UP22	HC UP21	HC UP20	HC UP19	HC UP18	HC UP17	HC UP16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HC UP11	HC UP10	HC UP9	HC UP8	HC UP7	HC UP6	HC UP5	HC UP4	HC UP3	HC UP2	HC UP1	HC UP0
<hr/>															
位	标记	位名	功能	读写											
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b23	HCUP23	硬件递加条件23	条件: TIM6_TRIGD端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W											
b22	HCUP22	硬件递加条件22	条件: TIM6_TRIGD端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W											
b21	HCUP21	硬件递加条件21	条件: TIM6_TRIGC端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W											
b20	HCUP20	硬件递加条件20	条件: TIM6_TRIGC端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W											
b19	HCUP19	硬件递加条件19	条件: TIM6_TRIGB端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W											
b18	HCUP18	硬件递加条件18	条件: TIM6_TRIGB端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W											
b17	HCUP17	硬件递加条件17	条件: TIM6_TRIGA端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W											
b16	HCUP16	硬件递加条件16	条件: TIM6_TRIGA端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W											
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b11	HCUP11	硬件递加条件11	条件: 内部硬件触发事件3有效 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W											
b10	HCUP10	硬件递加条件10	条件: 内部硬件触发事件2有效 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W											
b9	HCUP9	硬件递加条件9	条件: 内部硬件触发事件1有效 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W											
b8	HCUP8	硬件递加条件8	条件: 内部硬件触发事件0有效 0: 条件匹配时, 硬件递加无效	R/W											

			1: 条件匹配时, 硬件递加有效
b7	HCUP7	硬件递加条件7	条件: TIM6_<t>_PWMB端口为高电平时, TIM6_<t>_PWMA端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
b6	HCUP6	硬件递加条件6	条件: TIM6_<t>_PWMB端口为高电平时, TIM6_<t>_PWMA端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
b5	HCUP5	硬件递加条件5	条件: TIM6_<t>_PWMB端口为低电平时, TIM6_<t>_PWMA端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
b4	HCUP4	硬件递加条件4	条件: TIM6_<t>_PWMB端口为低电平时, TIM6_<t>_PWMA端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
b3	HCUP3	硬件递加条件3	条件: TIM6_<t>_PWMA端口为高电平时, TIM6_<t>_PWMB端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
b2	HCUP2	硬件递加条件2	条件: TIM6_<t>_PWMA端口为高电平时, TIM6_<t>_PWMB端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
b1	HCUP1	硬件递加条件1	条件: TIM6_<t>_PWMA端口为低电平时, TIM6_<t>_PWMB端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
b0	HCUP0	硬件递加条件0	条件: TIM6_<t>_PWMA端口为低电平时, TIM6_<t>_PWMB端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效

## 20.5.24 硬件递减事件选择寄存器 (TMR6\_HCDOR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HC DO23	HC DO22	HC DO21	HC DO20	HC DO19	HC DO18	HC DO17	HC DO16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HC DO11	HC DO10	HC DO9	HC DO8	HC DO7	HC DO6	HC DO5	HC DO4	HC DO3	HC DO2	HC DO1	HC DO0
<hr/>															
位	标记	位名	功能	读写											
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b23	HCDO23	硬件递减条件23	条件: TIM6_TRIGD端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W											
b22	HCDO22	硬件递减条件22	条件: TIM6_TRIGD端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W											
b21	HCDO21	硬件递减条件21	条件: TIM6_TRIGC端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W											
b20	HCDO20	硬件递减条件20	条件: TIM6_TRIGC端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W											
b19	HCDO19	硬件递减条件19	条件: TIM6_TRIGB端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W											
b18	HCDO18	硬件递减条件18	条件: TIM6_TRIGB端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W											
b17	HCDO17	硬件递减条件17	条件: TIM6_TRIGA端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W											
b16	HCDO16	硬件递减条件16	条件: TIM6_TRIGA端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W											
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b11	HCDO11	硬件递减条件11	条件: 内部硬件触发事件3有效 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W											
b10	HCDO10	硬件递减条件10	条件: 内部硬件触发事件2有效 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W											
b9	HCDO9	硬件递减条件9	条件: 内部硬件触发事件1有效 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W											
b8	HCDO8	硬件递减条件8	条件: 内部硬件触发事件0有效 0: 条件匹配时, 硬件递减无效	R/W											

			1: 条件匹配时, 硬件递减有效
b7	HCDO7	硬件递减条件7	条件: TIM6_<t>_PWMB端口为高电平时, TIM6_<t>_PWMA端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
b6	HCDO6	硬件递减条件6	条件: TIM6_<t>_PWMB端口为高电平时, TIM6_<t>_PWMA端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
b5	HCDO5	硬件递减条件5	条件: TIM6_<t>_PWMB端口为低电平时, TIM6_<t>_PWMA端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
b4	HCDO4	硬件递减条件4	条件: TIM6_<t>_PWMB端口为低电平时, TIM6_<t>_PWMA端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
b3	HCDO3	硬件递减条件3	条件: TIM6_<t>_PWMA端口为高电平时, TIM6_<t>_PWMB端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
b2	HCDO2	硬件递减条件2	条件: TIM6_<t>_PWMA端口为高电平时, TIM6_<t>_PWMB端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
b1	HCDO1	硬件递减条件1	条件: TIM6_<t>_PWMA端口为低电平时, TIM6_<t>_PWMB端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
b0	HCDO0	硬件递减条件0	条件: TIM6_<t>_PWMA端口为低电平时, TIM6_<t>_PWMB端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效

## 20.5.25 软件同步启动控制寄存器 (TMR6\_SSTAR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				SSTA10	SSTA9	SSTA8	SSTA7	SSTA6	SSTA5	SSTA4	SSTA3	SSTA2	SSTA1		
<hr/>															
位	标记	位名	功能	读写											
b31~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b9	SSTA10	单元10软件启动	0: 软件启动无效 1: 软件启动使能	R/W											
b8	SSTA9	单元9软件启动	0: 软件启动无效 1: 软件启动使能	R/W											
b7	SSTA8	单元8软件启动	0: 软件启动无效 1: 软件启动使能	R/W											
b6	SSTA7	单元7软件启动	0: 软件启动无效 1: 软件启动使能	R/W											
b5	SSTA6	单元6软件启动	0: 软件启动无效 1: 软件启动使能	R/W											
b4	SSTA5	单元5软件启动	0: 软件启动无效 1: 软件启动使能	R/W											
b3	SSTA4	单元4软件启动	0: 软件启动无效 1: 软件启动使能	R/W											
b2	SSTA3	单元3软件启动	0: 软件启动无效 1: 软件启动使能	R/W											
b1	SSTA2	单元2软件启动	0: 软件启动无效 1: 软件启动使能	R/W											
b0	SSTA1	单元1软件启动	0: 软件启动无效 1: 软件启动使能	R/W											

## 20.5.26 软件同步停止控制寄存器 (TMR6\_SSTPR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				SSTP10	SSTP9	SSTP8	SSTP7	SSTP6	SSTP5	SSTP4	SSTP3	SSTP2	SSTP1		
<hr/>															
位	标记	位名	功能	读写											
b31~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b9	SSTP10	单元10软件停止	0: 软件停止无效 1: 软件停止使能	R/W											
b8	SSTP9	单元9软件停止	0: 软件停止无效 1: 软件停止使能	R/W											
b7	SSTP8	单元8软件停止	0: 软件停止无效 1: 软件停止使能	R/W											
b6	SSTP7	单元7软件停止	0: 软件停止无效 1: 软件停止使能	R/W											
b5	SSTP6	单元6软件停止	0: 软件停止无效 1: 软件停止使能	R/W											
b4	SSTP5	单元5软件停止	0: 软件停止无效 1: 软件停止使能	R/W											
b3	SSTP4	单元4软件停止	0: 软件停止无效 1: 软件停止使能	R/W											
b2	SSTP3	单元3软件停止	0: 软件停止无效 1: 软件停止使能	R/W											
b1	SSTP2	单元2软件停止	0: 软件停止无效 1: 软件停止使能	R/W											
b0	SSTP1	单元1软件停止	0: 软件停止无效 1: 软件停止使能	R/W											

## 20.5.27 软件同步清零控制寄存器 (TMR6\_SCLRR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				SCLE10	SCLE9	SCLE8	SCLE7	SCLE6	SCLE5	SCLE4	SCLE3	SCLE2	SCLE1		
<hr/>															
位	标记	位名	功能	读写											
b31~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b9	SCLE10	单元10软件清零	0: 软件清零无效 1: 软件清零使能	R/W											
b8	SCLE9	单元9软件清零	0: 软件清零无效 1: 软件清零使能	R/W											
b7	SCLE8	单元8软件清零	0: 软件清零无效 1: 软件清零使能	R/W											
b6	SCLE7	单元7软件清零	0: 软件清零无效 1: 软件清零使能	R/W											
b5	SCLE6	单元6软件清零	0: 软件清零无效 1: 软件清零使能	R/W											
b4	SCLE5	单元5软件清零	0: 软件清零无效 1: 软件清零使能	R/W											
b3	SCLE4	单元4软件清零	0: 软件清零无效 1: 软件清零使能	R/W											
b2	SCLE3	单元3软件清零	0: 软件清零无效 1: 软件清零使能	R/W											
b1	SCLE2	单元2软件清零	0: 软件清零无效 1: 软件清零使能	R/W											
b0	SCLE1	单元1软件清零	0: 软件清零无效 1: 软件清零使能	R/W											

## 20.5.28 软件同步刷新控制寄存器 (TMR6\_SUPDR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				SUPD10	SUPD9	SUPD8	SUPD7	SUPD6	SUPD5	SUPD4	SUPD3	SUPD2	SUPD1		
<hr/>															
位	标记	位名	功能	读写											
b31~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b9	SUPD10	单元10软件刷新	0: 软件刷新无效 1: 软件刷新使能	R/W											
b8	SUPD9	单元9软件刷新	0: 软件刷新无效 1: 软件刷新使能	R/W											
b7	SUPD8	单元8软件刷新	0: 软件刷新无效 1: 软件刷新使能	R/W											
b6	SUPD7	单元7软件刷新	0: 软件刷新无效 1: 软件刷新使能	R/W											
b5	SUPD6	单元6软件刷新	0: 软件刷新无效 1: 软件刷新使能	R/W											
b4	SUPD5	单元5软件刷新	0: 软件刷新无效 1: 软件刷新使能	R/W											
b3	SUPD4	单元4软件刷新	0: 软件刷新无效 1: 软件刷新使能	R/W											
b2	SUPD3	单元3软件刷新	0: 软件刷新无效 1: 软件刷新使能	R/W											
b1	SUPD2	单元2软件刷新	0: 软件刷新无效 1: 软件刷新使能	R/W											
b0	SUPD1	单元1软件刷新	0: 软件刷新无效 1: 软件刷新使能	R/W											

## 20.6 使用注意事项

- 1) 计数器在计数中时 CPU 不能对 CNTER 寄存器执行写入动作，只能在计数器停止时对 CNTER 寄存器执行写入动作。
- 2) GCMCR~GCMFR 在对应的缓存功能有效时，其中断输出、事件输出无效。
- 3) 在使用双边对称互补 PWM 输出功能时，用户需计算、设定好相关的基准值和端口输出状态，保证端口为预期的输出状态。
- 4) 在一些特定场合，对计数器的控制事件有可能同时出现，此时，计数器（CNTER）的变化控制优先级如下表 20-5 所示。

表 20-5 计数器（CNTER）控制优先级

计时器控制事件	计数器（CNTER）值	优先级
CPU对CNTER执行写操作	CPU的写入值	高
刷新事件（软件刷新或硬件刷新）	UPDAR中的值	↓
清零事件（软件清零或硬件清零）	硬件计数或锯齿波模式时	根据计数方向变为0或PERAR
	三角波模式时	0
硬件计数	正常计数值	↓
三角波模式计数	正常计数值	↓
锯齿波模式计数	正常计数值	低

- 5) 在一些特定场合，对 PWM 端口输出的控制事件有可能同时出现，此时，两个端口（TIM6\_<t>\_PWMA 和 TIM6\_<t>\_PWMB）的输出变化控制优先级分别如下表 20-6 所示、表 20-7 所示。

表 20-6 PWMA 端口输出控制优先级

PWMA端口输出控制事件	端口状态	优先级
EMB选择的事件发生	PCNAR.EBCA设定状态	高
设定强制端口输出	PCNAR.FORCA设定状态	↓
计数开始时	PCNAR.STACA设定状态	↓
锯齿波向上计数时	CNTER=PERAR	PCNAR.OVFCA设定状态
	CNTER=GCMAR	PCNAR.CMAUCA设定状态
	CNTER=GCMBR	PCNAR.CMBUCA设定状态
锯齿波向下计数时	CNTER=0	PCNAR.UDFCA设定状态
	CNTER=GCMAR	PCNAR.CMADCA设定状态
	CNTER=GCMBR	PCNAR.CMBDCA设定状态
三角波向上计数时	CNTER=GCMAR	PCNAR.CMAUCA设定状态
	CNTER=GCMBR	PCNAR.CMBUCA设定状态
	CNTER=PERAR	PCNAR.OVFCA设定状态
三角波向下计数时	CNTER=GCMAR	PCNAR.CMADCA设定状态
	CNTER=GCMBR	PCNAR.CMBDCA设定状态

PWMA端口输出控制事件		端口状态	优先级
	CNTER=0	PCNAR.UDFCA设定状态	↓
计数停止时		PCNAR.STPCA设定状态	低

表 20-7 PWMB 端口输出控制优先级

PWMB端口输出控制事件		端口状态	优先级
EMB选择的事件发生		PCNBR.EMBCB设定状态	高
设定强制端口输出		PCNBR.FORCB设定状态	↓
计数开始时		PCNBR.STACB设定状态	↓
锯齿波向上计数时	CNTER=PERAR	PCNBR.OVFCB设定状态	↓
	CNTER=GCMBR	PCNBR.CMBUCB设定状态	↓
	CNTER=GCMAR	PCNBR.CMAUCB设定状态	↓
锯齿波向下计数时	CNTER=0	PCNBR.UDFCB设定状态	↓
	CNTER=GCMBR	PCNBR.CMBDCB设定状态	↓
	CNTER=GCMAR	PCNBR.CMADCB设定状态	↓
三角波向上计数时	CNTER=GCMBR	PCNBR.CMBUCB设定状态	↓
	CNTER=GCMAR	PCNBR.CMAUCB设定状态	↓
	CNTER=PERAR	PCNBR.OVFCB设定状态	↓
三角波向下计数时	CNTER=GCMBR	PCNBR.CMBDCB设定状态	↓
	CNTER=GCMAR	PCNBR.CMADCB设定状态	↓
	CNTER=0	PCNBR.UDFCB设定状态	↓
计数停止时		PCNBR.STPCB设定状态	低

## 21 通用控制定时器 (Timer4)

### 21.1 简介

通用控制定时器 4 (Timer4) 是一个用于三相电机控制的定时器模块，提供各种不同应用的三相电机控制方案。该定时器支持三角波和锯齿波两种波形模式，可生成各种 PWM 波形；支持缓存功能；支持 EMB 控制。本系列产品中搭载 1 个单元的 Timer4。

### 21.2 基本框图

Timer4 基本的功能及特性如表 21-1 所示。

表 21-1 Timer4 的基本功能及特性

波形模式	锯齿波、三角波
基本功能	• 递加、递减计数方向
	• 缓存功能
	• 通用 PWM 输出
	• 通用比较匹配事件触发其他外设
	• 上溢、下溢、重载计数匹配事件触发其他外设
	• 专用比较匹配事件触发其他外设
	• 其他外设启动计数
	• EMB 控制
中断类型	通用比较匹配中断
	计数周期匹配中断
	重载计数匹配中断
	专用比较匹配中断

图 21-1 所示，描述了通用控制定时器 Timer4 的基本架构。框图中所示“”表示单元编号，即“”为 1，本章节后文提到“”时均指单元编号，不再赘述。

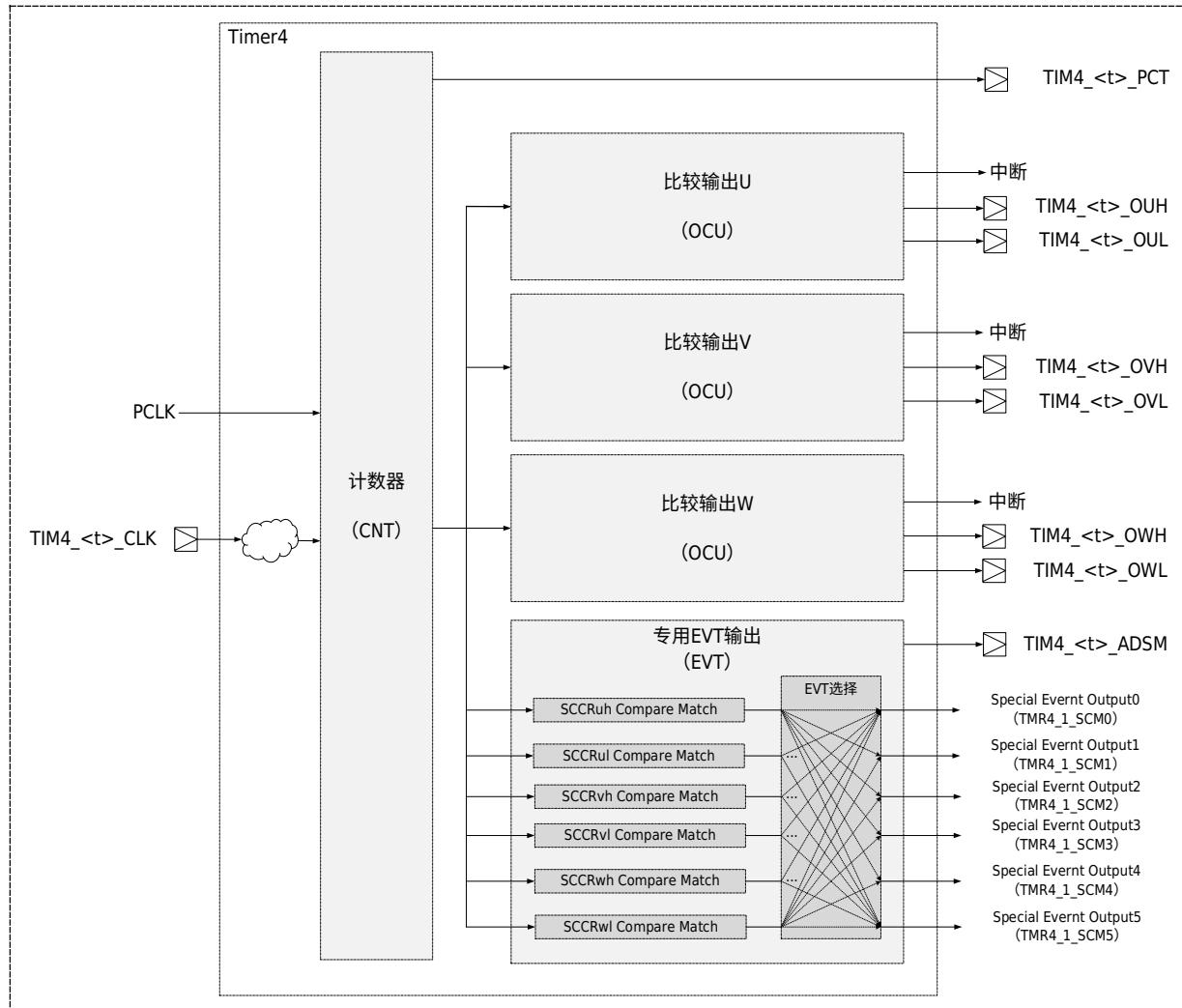


图 21-1 Timer4 基本框图

表 21-2 所示为 Timer4 的输入输出端口列表。

表 21-2 Timer4 端口列表

端口名	方向	功能
TIM4_<t>_CLK	输入	计数时钟输入端口
TIM4_<t>_PCT	输出	PWM周期输出监测
TIM4_<t>_ADSM	输出	专用事件输出监测
TIM4_<t>_OUH	输出	PWM输出端口
TIM4_<t>_OUL		
TIM4_<t>_OVH		
TIM4_<t>_OVL		
TIM4_<t>_OWH		
TIM4_<t>_OWL		

## 21.3 功能说明

### 21.3.1 基本动作

#### 21.3.1.1 波形模式

Timer4 有 2 种基本计数波形模式，锯齿波模式和三角波模式。根据 CCSR.MODE 的设定可分别实现两种波形模式。下图 21-2 和图 21-3 分别是锯齿波和三角波的波形图。

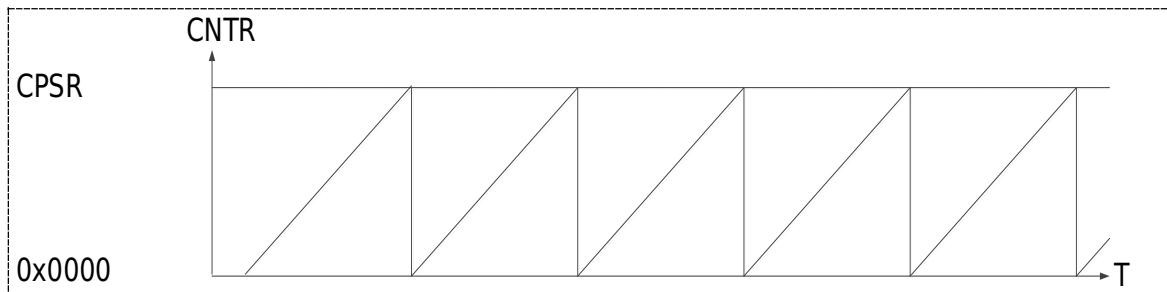


图 21-2 Timer4 锯齿波波形

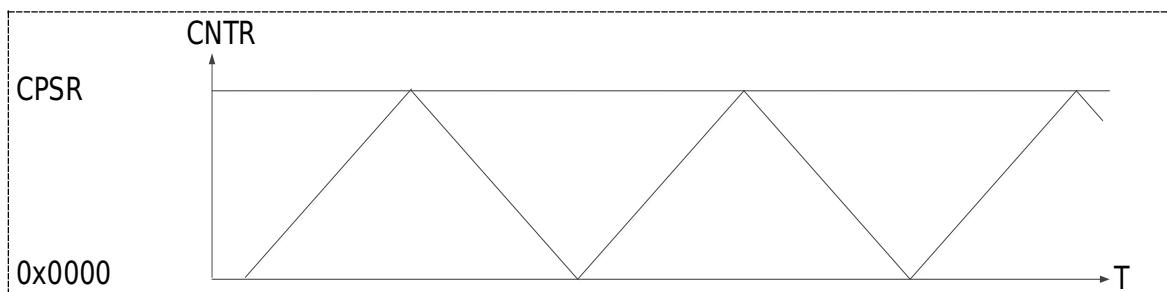


图 21-3 Timer4 三角波波形

#### 21.3.1.2 计数动作

1) 锯齿波计数操作与控制流程如图 21-4 所示。

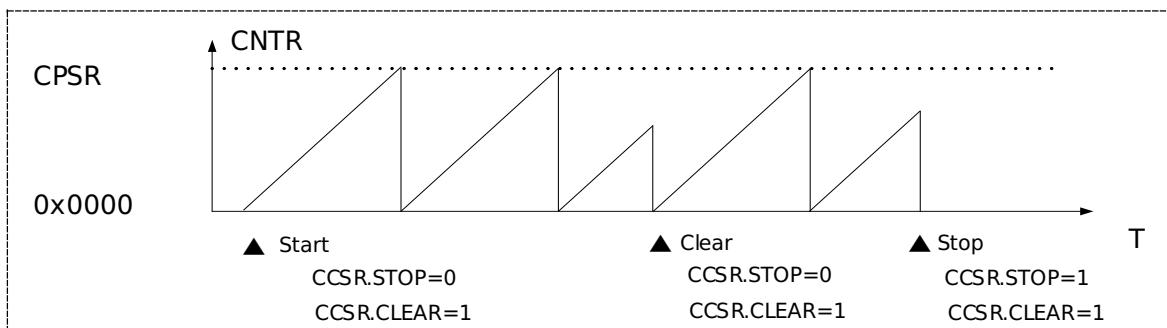


图 21-4 Timer4 锯齿波模式计数动作

- 设置模式 CCSR.MODE=0。
- 设置计数峰值 CPSR 寄存器。
- 写入 CCSR.STOP=0 和 CCSR.CLEAR=1，计数器计数值（CNTR）被初始化到 0x0000 并启动计数操作。计数器值从 0x0000 开始递增计数，当达到峰值（CPSR），计数值归 0x0000，依

次重复此操作。

- d) 计数周期=  $(CPSR+1) \times$ 计数时钟周期
  - e) 计数过程中, 写入 CCSR.STOP=0 和 CCSR.CLEAR=1 可以将计数值初始化到 0x0000 并继续进行计数操作。
  - f) 计数过程中, 写入 CCSR.STOP=1 和 CCSR.CLEAR=1 可以将计数值初始化到 0x0000 并停止计数操作。
- 2) 三角波计数操作与控制流程如图 21-5 所示。

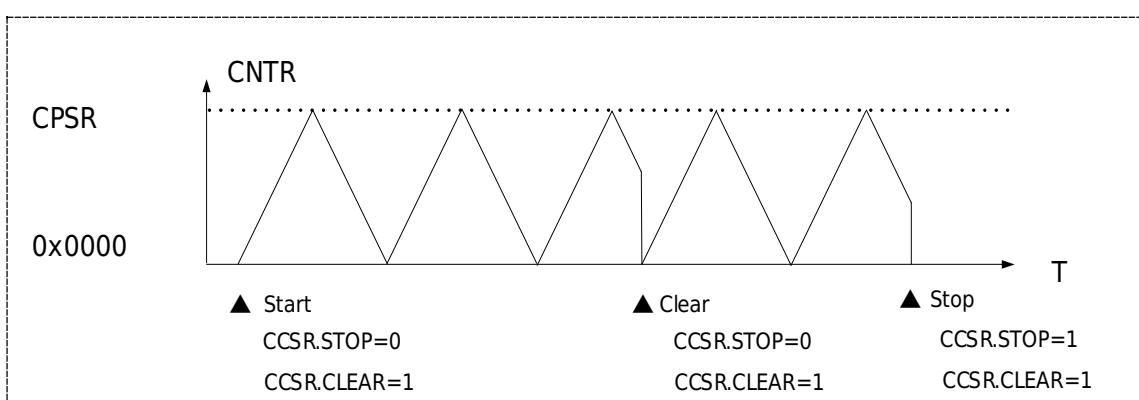


图 21-5 Timer4 三角波模式计数动作

- a) 设置模式 CCSR.MODE=1。
- b) 设置计数峰值 CPSR 寄存器。
- c) 写入 CCSR.STOP=0 和 CCSR.CLEAR=1, 计数器计数值 (CNTR) 被初始化到 0x0000 并启动计数操作。计数器值从 0x0000 开始做递增计数, 在到达计数峰值之前做递增计数; 当达到峰值 (CPSR) 时, 计数器开始做递减计数, 直到计数值归 0x0000; 之后重新进行递增计数操作, 依次重复此操作。
- d) 计数周期=  $(CPSR) \times 2 \times$ 计数时钟周期
- e) 计数过程中, 写入 CCSR.STOP=0 和 CCSR.CLEAR=1 可以将计数值初始化到 0x0000 并重新进行递增计数操作, 之后重复上述操作。
- f) 计数过程中, 写入 CCSR.STOP=1 和 CCSR.CLEAR=1 可以将计数值初始化到 0x0000 并停止计数操作。

### 21.3.1.3 比较输出

1) 下图 21-6 是锯齿波计数模式下的比较输出模块的波形输出示例。

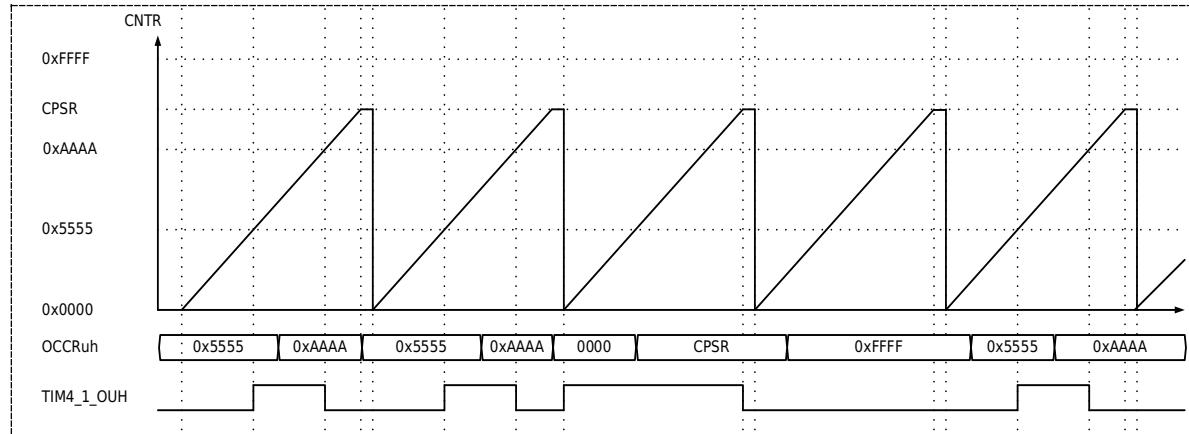


图 21-6 锯齿波模式波形输出例

2) 下图 21-7 是三角波计数模式下的比较输出模块的波形输出示例。

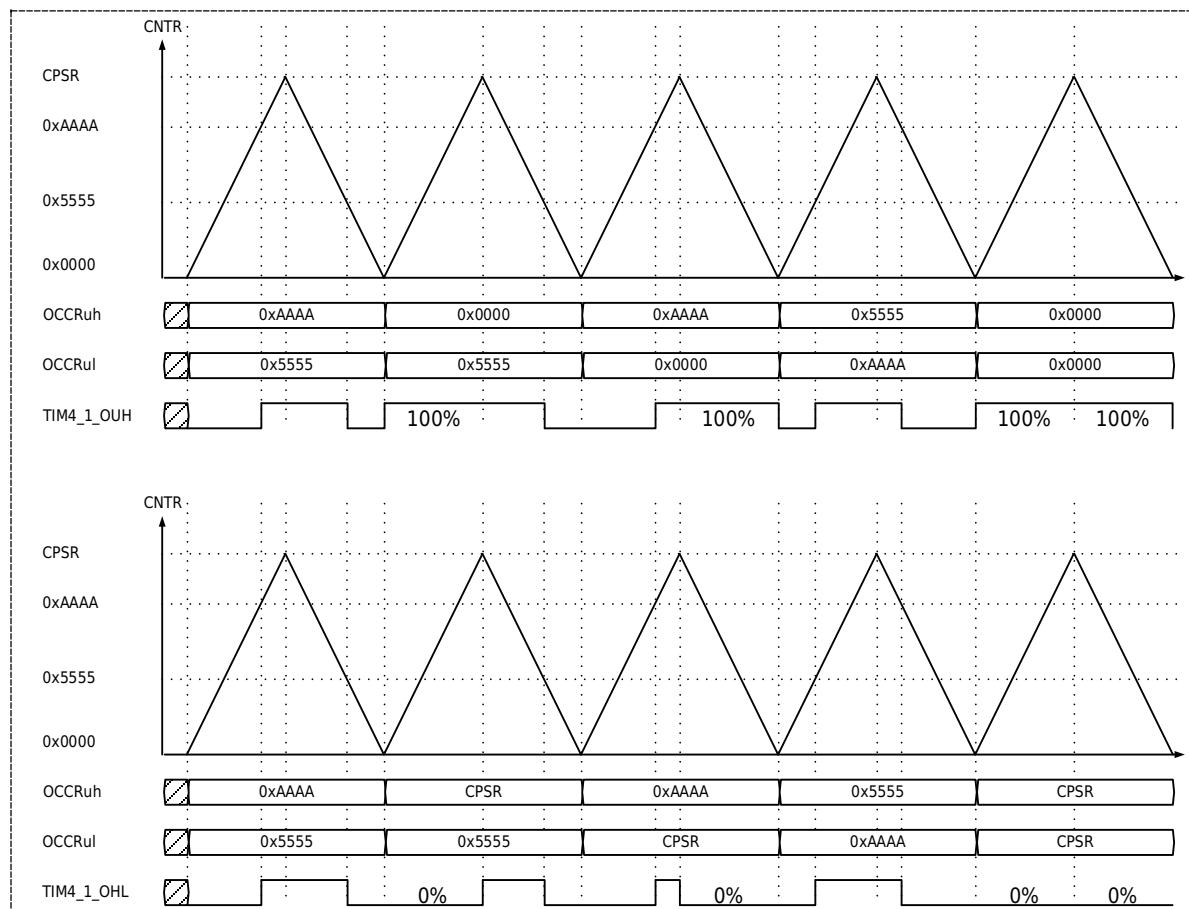


图 21-7 三角波模式波形输出例

## 21.3.2 缓存功能

Timer4 的周期基准寄存器 (CPSR)、通用比较基准寄存器 (OCCR)、通用模式控制寄存器 (OCMR)、专用比较基准寄存器 (SCCR) 和专用模式控制寄存器 (SCMR) 都具有缓存功能。

### 21.3.2.1 周期基准寄存器缓存功能

CPSR 具有缓存功能寄存器，写入的计数峰值数据先写入到缓冲寄存器。在以下条件下，数据从缓冲器寄存器传输到 CPSR 寄存器中。

- 缓冲功能禁止时 (CCSR.BUFEN=0)，写入数据立即从缓冲寄存器传输到 CPSR 寄存器。
- 缓冲功能使能时 (CCSR.BUFEN=1)，在计数器停止时 (CCSR.STOP=1) 或者计数器的计数值为 "0x0000" 时，数据从缓冲寄存器传输到 CPSR 寄存器。

**注意：**

- 从 CPSR 读取数据时，读取的并非 CPSR 缓冲寄存器的值，而是 CPSR 寄存器的值。当缓冲功能使能时，在传输完成之前所读取的值不是最近写入的值，而是上一次写入的 CPSR 值。

图 21-8 所示，当锯齿波模式且禁止缓冲功能时修改计数峰值 CPSR 的操作。

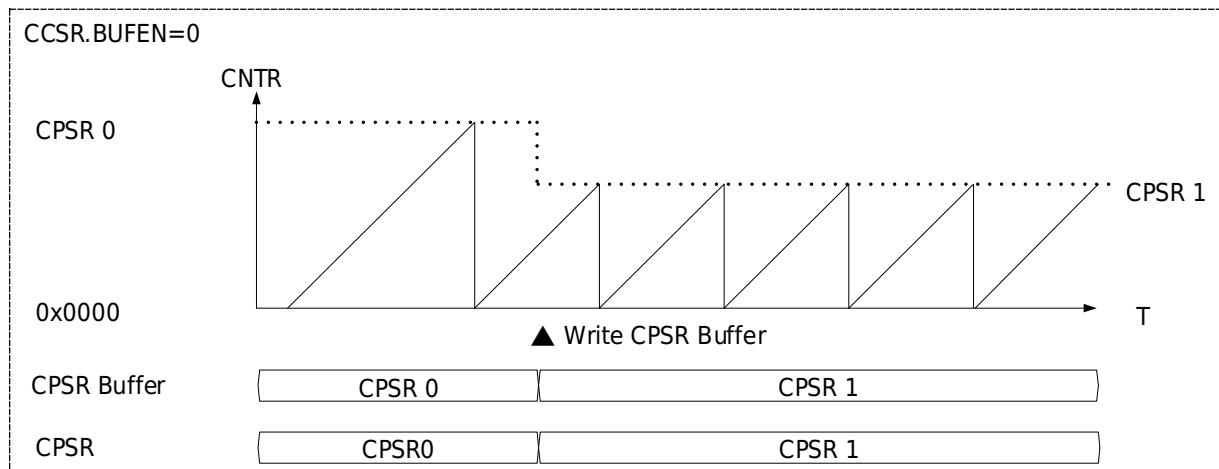


图 21-8 缓存无效时修改锯齿波计数周期

**注意：**

- 当禁止缓冲功能时，写入数据立即从缓冲寄存器传输到 CPSR 计数器，计数周期在写入操作完成后立即改变。在此情况下，如果写入的值小于计数器当前计数值，计数器将持续进行递增计数，直到到达 "0xFFFF"，因此，需特别注意此种状况。

图 21-9 所示，当锯齿波模式且使能缓冲功能时修改计数峰值 CPSR 的操作。

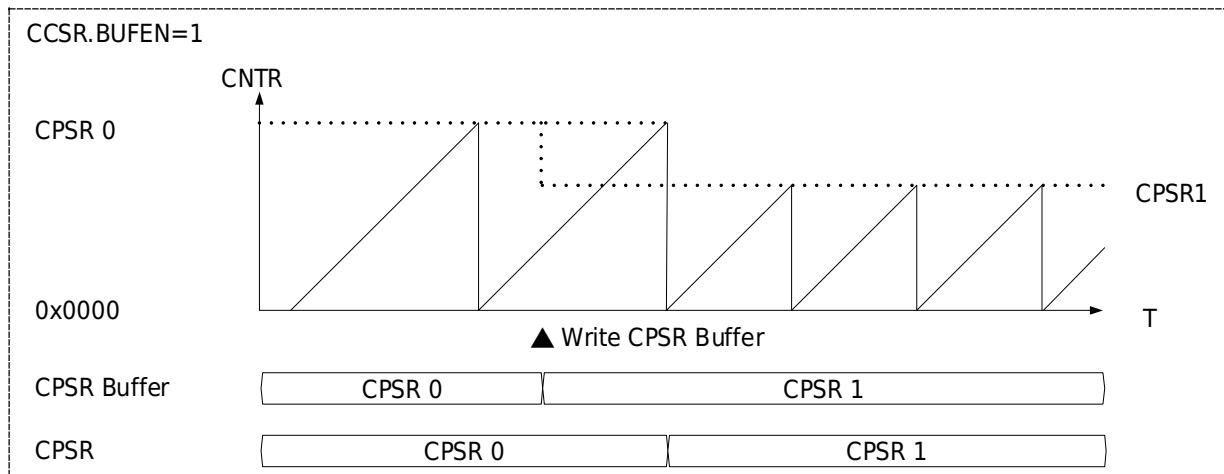


图 21-9 缓存使能时修改锯齿波计数周期

如图所示，当使能缓冲功能时，在计数器停止时或者计数器的计数值为"0x0000"时，写入的数据从缓冲寄存器传输到 CPSR 寄存器。

图 21-10 所示，当三角波模式且使能缓冲功能时修改计数峰值 CPSR 的操作。

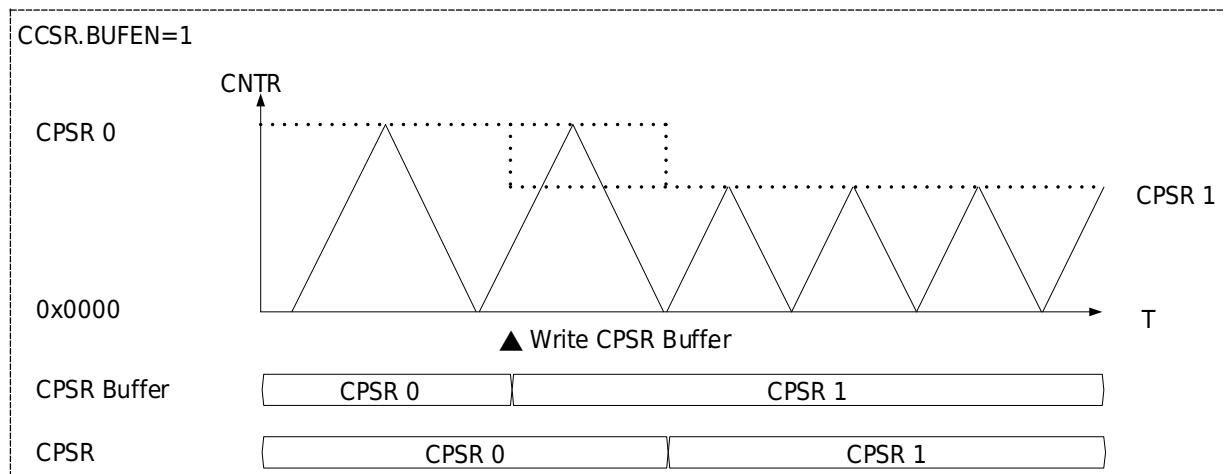


图 21-10 缓存使能时修改三角波计数周期

如图所示，三角波计数模式下，当使能缓冲功能时，在计数器停止时或者计数器的下一次计数值为"0x0000"时，写入的数据从缓冲寄存器传输到 CPSR 寄存器。计数器周期的改变在写入操作完成后的下一个计数器周期开始。

### 21.3.2.2 通用比较寄存器缓存功能

通用比较基准值寄存器（OCCR）和通用模式控制寄存器（OCMR）均有缓冲寄存器功能，当使能缓冲功能时，在指定的传输时刻传输加载到 OCCR 和 OCMR 寄存器。OCCR 缓冲功能可以用于计数过程中同步改变比较值，OCMR 缓冲功能可以用于计数过程中计数上溢点（锯齿波）、计数谷点或峰点（三角波）同步改变内部 PWM 输出。

- 当输出比较与计数器周期间隔响应功能的链接传输禁止时，缓冲值在设定的计数状态时加载

到寄存器。此时加载情况与计数器周期间隔计数器无关。

图 21-11 所示，是通用输出比较 OCCR 缓冲功能使能，计数器零值时加载（OCER.CxBUFEN=01），计数器周期间隔响应链接禁止（OCER.LMCx=0）时的波形（ $x=L$  或  $H$ ）。

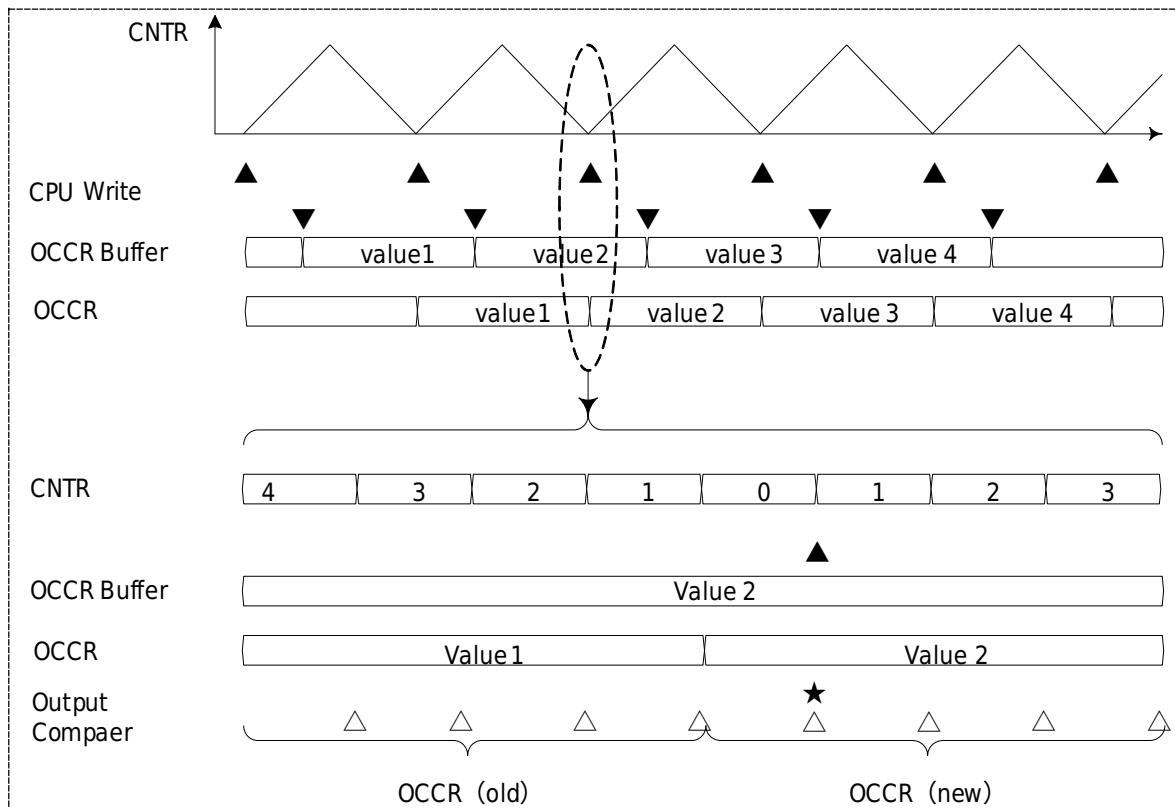


图 21-11 OCCR 缓冲数据传输（周期间隔响应链接禁止时）

图中的上半部分是全局示意图，下半部分是传输操作时放大图。

计数器为三角波计数模式，下溢中断在标志▲（计数谷点）时刻产生。在▼时刻，CPU 改写 OCCR 寄存器，写入的数据存储在 OCCR 缓冲寄存器中。之后当计数器计数值为 0x0000 时，进行数据加载操作，数据从缓冲寄存器到 OCCR 寄存器，同时产生中断标志 IRQZF。

在△时刻，输出比较根据设定的 OCCR 寄存器值与计数值匹配事件，执行改变 PWM 输出和置位 OCSR.OCFx 位 ( $x=L$  或  $H$ )。在时刻★（计数值=0x0000）之后，端口输出根据新的 OCCR 数据执行操作。在时刻★之前根据原 OCCR 数据执行操作。

图中示意了 OCCR 缓冲寄存器在计数谷点的传输操作，OCMR 的缓冲寄存器的缓存传送操作与之类似；同理，在计数峰点的传输操作也类似。新的数据在传输时刻之后即刻生效（新写入数据将控制 PWM 输出和中断标志置位）。

- b) 当计数器周期间隔响应链接使能时，缓冲值在设定的计数状态且周期间隔计数器计数值为 0 时执行缓冲寄存器传输操作。

图 21-12 所示，是通用输出比较 OCCR 缓冲功能使能、计数器零值时加载（OCER.CxBUFEN=01），计数器周期间隔响应链接使能（OCER.LMCx=1）时的波形（ $x=L$  或  $H$ ）。

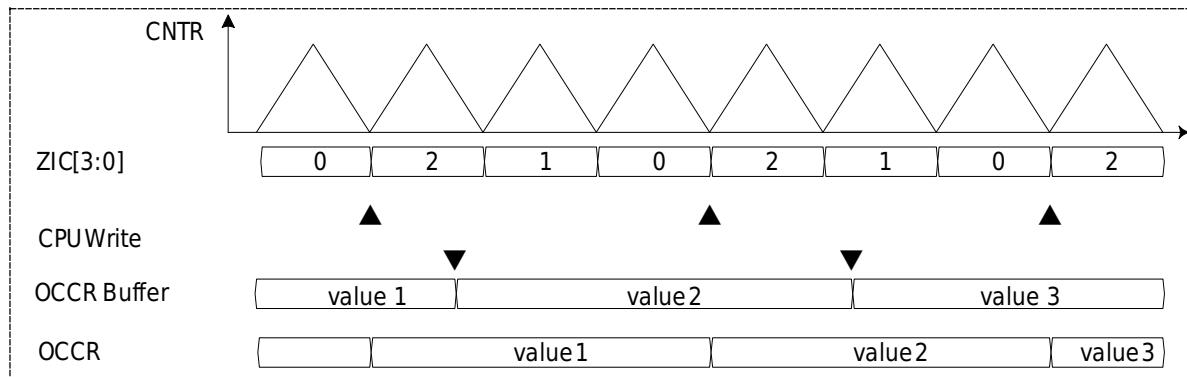


图 21-12 OCCR 缓冲数据传输（周期间隔响应链接使能）

计数器为三角波计数模式，周期间隔计数器（CVPR.ZIC）从 2 到 0 做递减计数，下溢中断在标志▲（计数谷点）时刻产生。在▼时刻，CPU 改写 OCCR 寄存器，写入的数据存储在 OCCR 缓冲寄存器中。之后当计数器计数值为 0x0000 且周期间隔计数器（CVPR.ZIC）为 0 时，进行数据加载操作，数据从缓冲寄存器到 OCCR 寄存器，同时产生中断标志 IRQZF。

图中示意了 OCCR 缓冲寄存器在计数谷点的传输操作，OCMR 的缓冲寄存器的缓存传送操作与之类似；同理，在计数峰点的传输操作也类似。新的数据在传输时刻之后即刻生效（新写入数据将控制 PWM 输出和中断标志置位）。

使用通道链接操作模式时，同时使能 OCCRh 和 OCCRu 缓冲功能可以产生各种不同的 PWM 输出波形。图 21-13 示例了在输出比较寄存器 OCCRh 和 OCCRu 保持不变的情况下，改变 OCMR 寄存器值来产生不同的输出波形 TIM4\_<t>\_OUL 方式的情形。

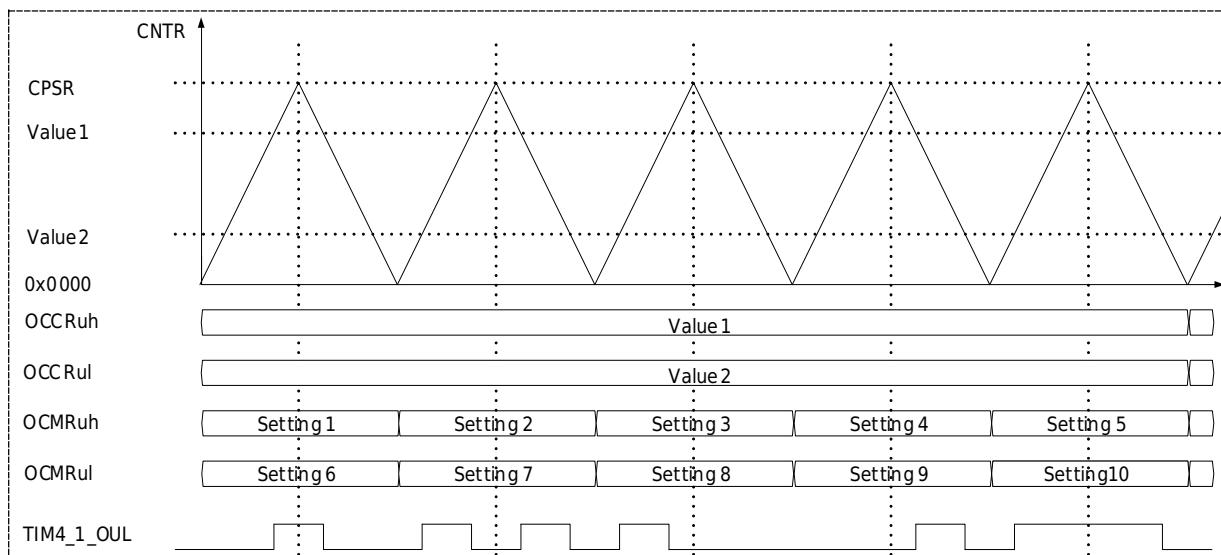


图 21-13 输出比较缓冲数据传输（OCMR 缓冲使能）

### 21.3.2.3 专用比较寄存器缓存功能

专用比较基准寄存器（SCCR）和在专用模式控制寄存器（SCMR）都带有缓冲功寄存器。当使能缓冲功能时，CPU 写入 SCCR 和 SCMR 缓冲寄存器的值在设定的计数器状态下加载到 SCCR 和 SCMR 寄存器。

- a) 当禁止计数器周期间隔响应链接传输时，缓冲传输操作仅与计数器状态有关，不受计数器周期间隔计数器影响。

图 21-14 所示，是使能寄存器缓冲功能，禁止计数器周期间隔响应链接传输（SCSR.LMC=0），计数器零值（SCSR.BUFEN=01）时传送到 SCCR 和 SCMR 寄存器。

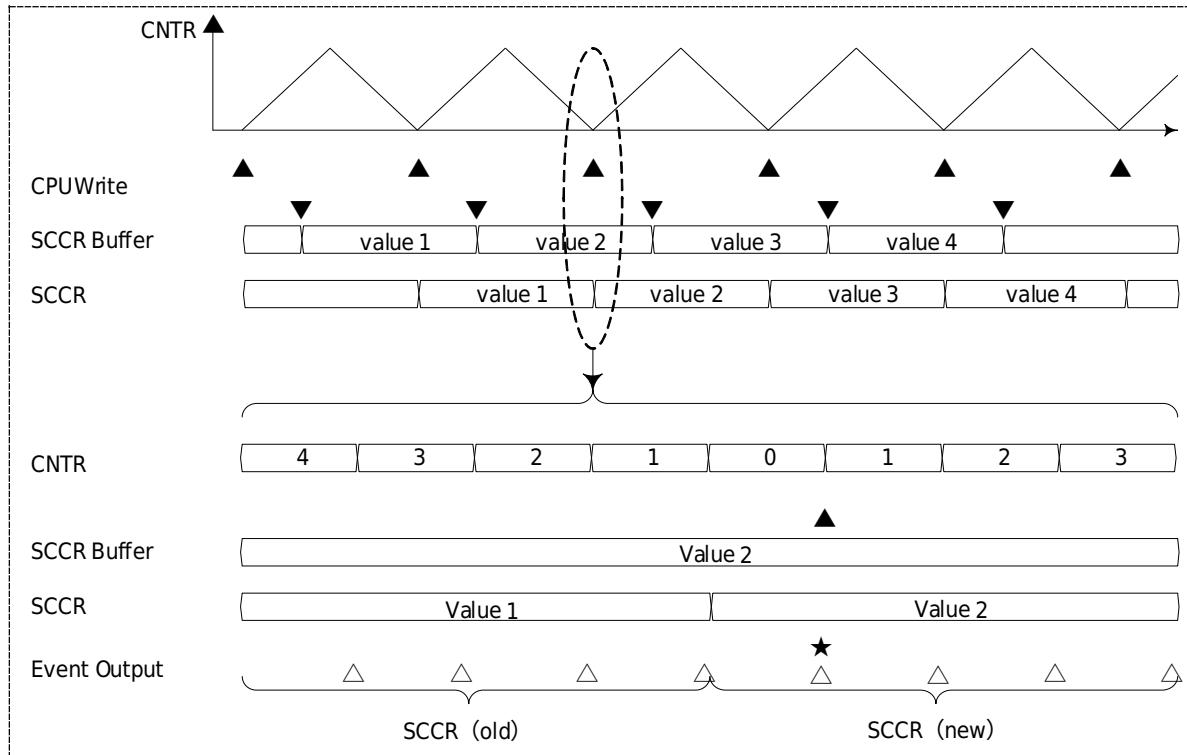


图 21-14 SCCR 缓冲传输操作（周期间隔响应链接传输禁止时）

图中的上半部分是全局示意图，下半部分缓冲寄存器传输操作的局部放大图。

计数器为三角波计数模式，下溢中断在标志▲（计数谷点）时刻产生。在▼时刻，CPU 改写 SCCR 寄存器，写入的数据存储在 SCCR 缓冲寄存器中。之后当计数器计数值为 0x0000 时，进行数据加载操作，数据从缓冲寄存器到 SCCR 寄存器，同时产生中断标志 IRQZF。

在△时刻，根据设定的 SCCR 寄存器值与计数值进行比较操作。在时刻★（计数值=0x0000）之后，专用事件输出信号根据新的 SCCR 数据执行操作。在时刻★之前根据 SCCR 数据执行操作。

图中示意了 SCCR 缓冲寄存器在计数谷点的传输操作，SCMR 的缓冲寄存器的缓存传送操作与之类似；同理，在计数峰点的传输操作也类似。新的数据在传输时刻之后即刻生效（新写入数据将专用事件输出信号和中断标志置位）。

- b) 当计数器周期间隔响应链接使能时，缓冲值在设定的计数状态且周期间隔计数器计数值为 0

时执行缓冲寄存器传输操作。

图 21-15 所示，是使能 SCCR 缓冲功能、计数器零值时加载（SCSR.BUFEN=01），计数器周期期间隔响应链接传输使能时（SCSR.LMC=1）的示意图。

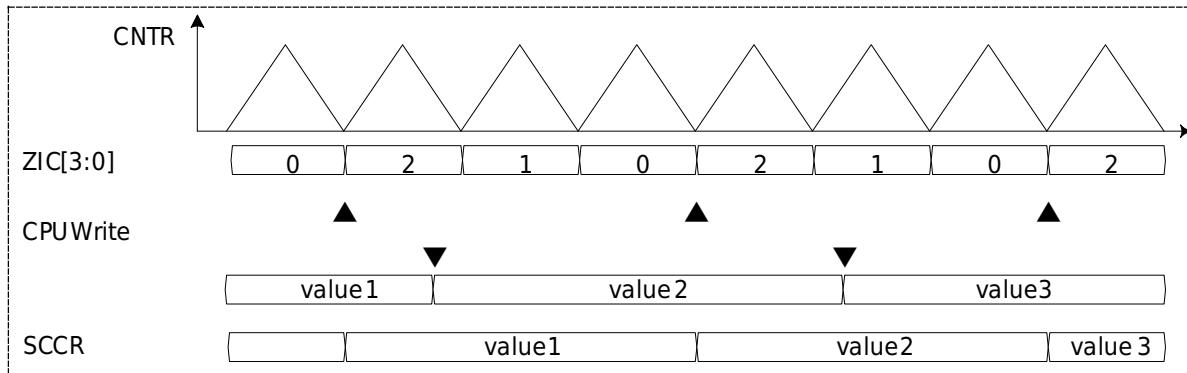


图 21-15 SCCR 缓冲传输操作（周期间隔响应链接传输使能时）

计数器为三角波计数模式，周期间隔计数器（CVPR.ZIC）从 2 到 0 做递减计数，下溢中断在标志▲（计数谷点）时刻产生。在▼时刻，CPU 改写 SCCR 寄存器，写入的数据存储在 SCCR 缓冲寄存器中。之后当计数器计数值为 0x0000 且周期间隔计数器（CVPR.ZIC）为 0 时，进行数据加载操作，数据从缓冲寄存器到 SCCR 寄存器，同时产生中断标志 IRQZF。

### 21.3.3 通用 PWM 输出

#### 21.3.3.1 独立 PWM 输出

在直通模式（POCR.PWMMD=00）下，分别设定 OCCRxh、OCCRxl 的基准值和 OCMRxh、OCMRxl 的端口输出状态（ $x=u, v, w$ ），可实现不同的 PWM 输出。此时，每个端口的 PWM 输出都独立控制。使用独立 PWM 输出模式前，需要设定 MOE、OExy 位（寄存器 PSCR）以使能 Timer4 的输出（ $x=u, v, w, y=h, l$ ），详细描述请参考【PWM 状态控制寄存器（TMR4\_PSCR）】。图 21-16 和图 21-17 分别是单元 1 的锯齿波和三角波下的独立 PWM 输出例。

**注意：**

- 直通模式是指通用比较基准寄存器（OCCRxh、OCCRxl）的值发生比较匹配所产生的内部输出信号（in\_opxh、in\_opxl）直接输出至对应的端口（TIM4\_<t>\_OXH、TIM4\_<t>\_OXL）上（ $X=U, V, W, x=u, v, w$ ）。

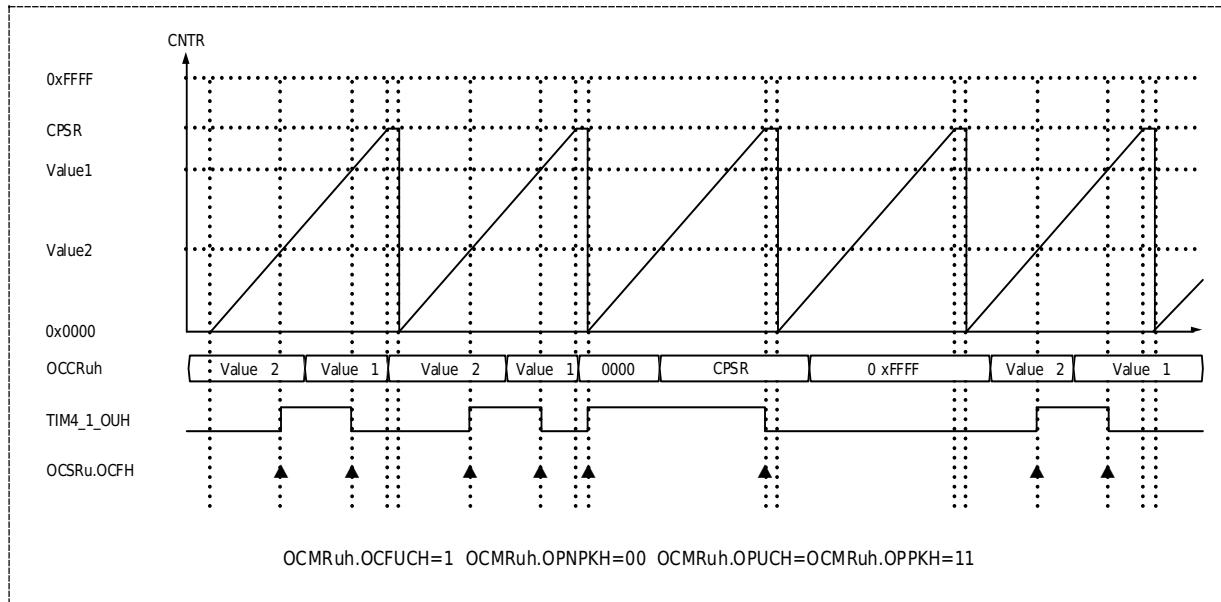


图 21-16 锯齿波独立 PWM 输出例

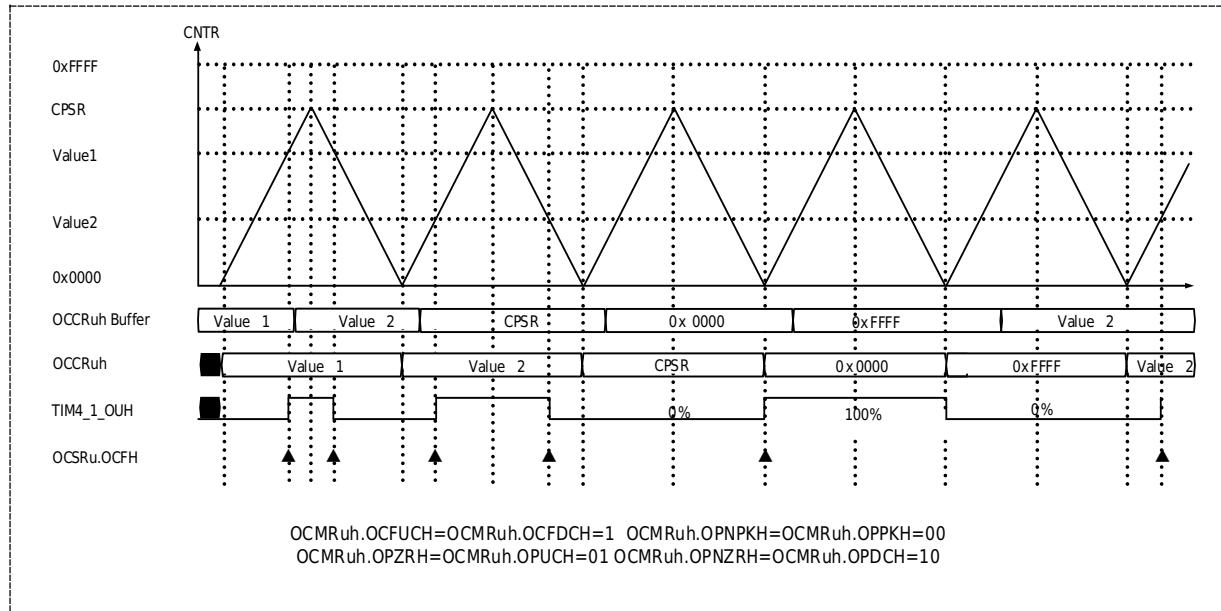


图 21-17 三角波独立 PWM 输出例

### 21.3.3.2 扩展 PWM 输出

在直通模式 (POCR.PWMMD=00) 下, TIM4\_<t>\_OXL 的端口输出状态也可由 OCMRxI 寄存器中的扩展位 (bit32~16) 决定, 该扩展位的设定与 OCRRxh 基准值相关, 从而在 TIM4\_<t>\_OXL 端口上实现扩展 PWM 输出 ( $X=U、V、W, x=u、v、w$ )。使用扩展 PWM 输出模式前, 需要设定 MOE、OExy 位 (寄存器 PSCR) 以使能 Timer4 的输出 ( $x=u、v、w, y=h、l$ ), 详细描述请参考【PWM 状态控制寄存器 (TMR4\_PSCR)】。图 21-18 所示是该方式下 TIM4\_<t>\_OUH 和 TIM4\_<t>\_OUL 端口的 PWM 输出。

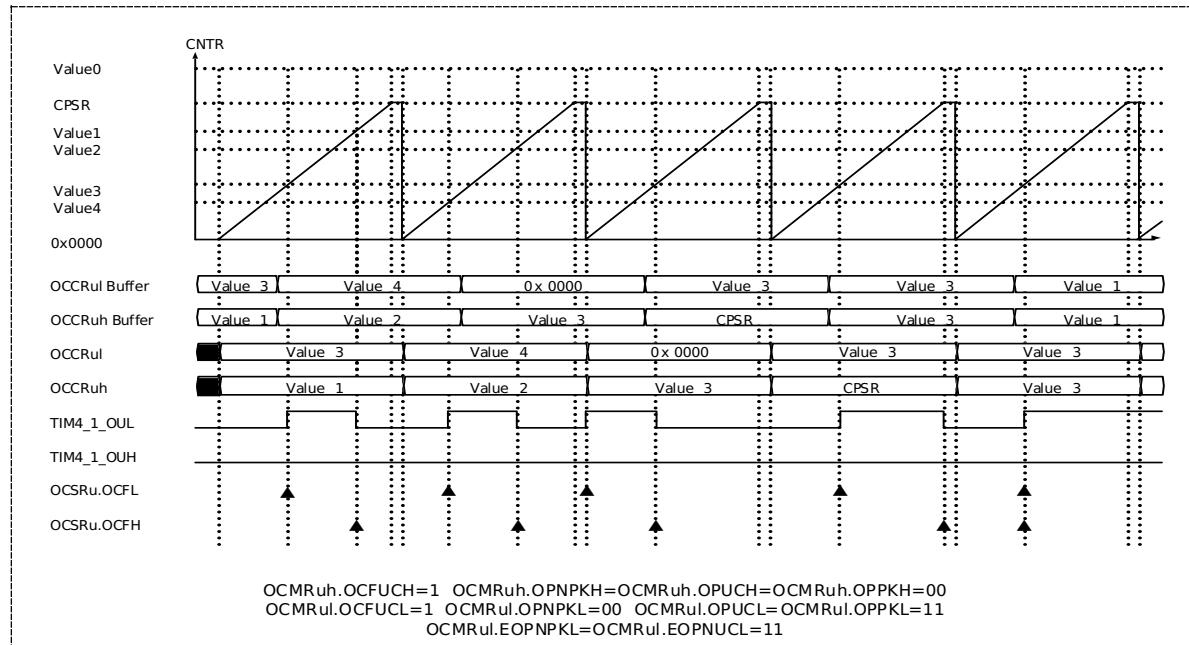


图 21-18 三角波扩展 PWM 输出

**注意：**

- 独立 PWM 输出方式下, TIM4\_<t>\_OXL 的端口状态由 OCMRxI 寄存器的 bit15~bit4 位决定, 仅与 OCCRxI 基准值有关 ( $X=U、V、W, x=u、v、w$ )。

**21.3.3.3 互补 PWM 输出****软件设定实现互补 PWM 输出**

在直通模式 (POCR.PWMMD=00) 下, 直接设定 OCCRxh 和 OCCRxI ( $x=u、v、w$ ) 基准值实现一对互补 PWM 波形输出至端口, 3 组端口同样方式设定即可实现 3 相互补 PWM 输出。如图 21-19 所示。

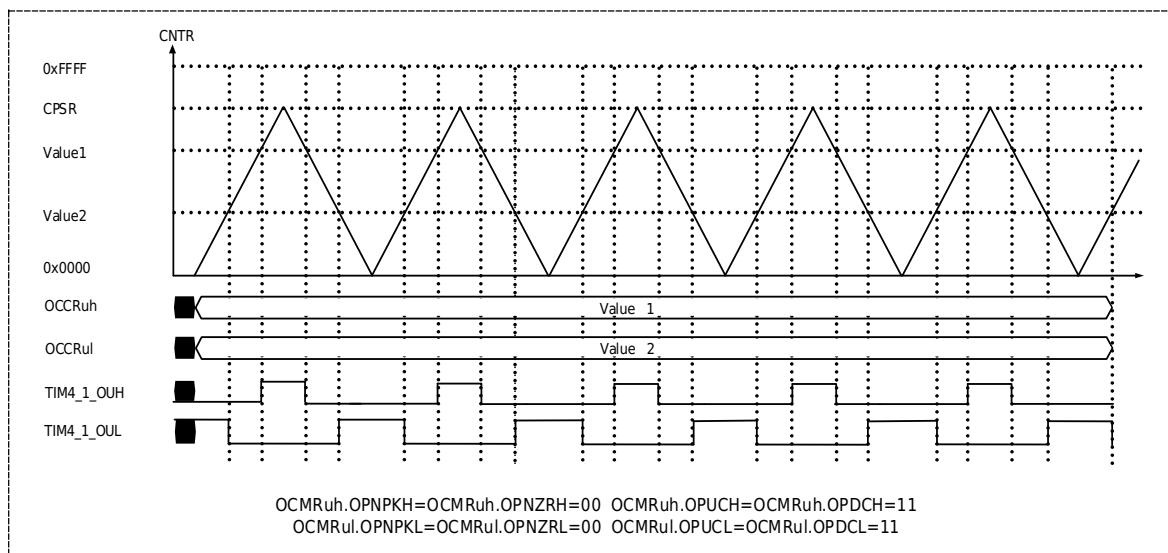


图 21-19 软件实现互补 PWM 输出

## 硬件设定实现互补 PWM 输出

在死区定时器模式 (POCR.PWMMD=01) 下，通用比较基准寄存器 (OCCR<sub>Xl</sub>) 的值发生比较匹配产生的内部输出信号 (in\_op<sub>Xl</sub>) 和 PWM 死区控制寄存器 (PDAR/PDBR) 的设定值通过时序偏移，以硬件方式实现互补 PWM 输出。

在该模式下，TIM4\_<t>\_OXH 端口输出的极性与 in\_op<sub>Xl</sub> 相同，TIM4\_<t>\_OXL 端口输出的极性与 in\_op<sub>Xl</sub> 相反 ( $X=U、V、W, x=u、v、w$ )。图 21-20 所示是死区定时器模式下的互补 PWM 输出例。

如果检测到 in\_op<sub>Xl</sub> 上升沿，TIM4\_<t>\_OXL 输出变为低电平，死区计数器加载 PDBRx 寄存器的设定值并启动递减计数，当计数值变为 0x0000 时，计数器停止并使 TIM4\_<t>\_OXH 输出高电平；如果检测到 in\_op<sub>Xl</sub> 下降沿，TIM4\_<t>\_OXH 输出变为低电平，死区计数器加载 PDARx 寄存器的设定值并启动递减计数，当计数值变为 0x0000 时，计数器停止并使 TIM4\_<t>\_OXL 输出高电平 ( $X=U、V、W, x=u、v、w$ )。

通过设置 PWM 死区控制寄存器 PDAR 和 PDBR，输出上升和下降变化的死区时间可以相应设定。

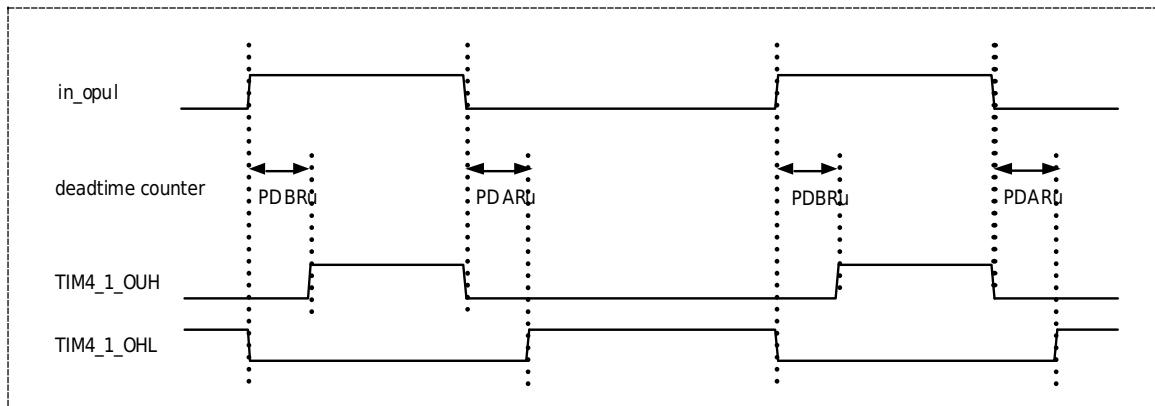


图 21-20 死区定时器模式下的互补 PWM 输出

in\_opxl 的高电平脉冲宽度小于 PDBR 设定的死区时间时，仅 TIM4\_<t>\_OXL 输出变为低电平。TIM4\_<t>\_OXL 输出电平变为高的条件是当 in\_opxl 下降沿之后经过 PDAR 寄存器设定的死区时间之后。在此情况下，TIM4\_<t>\_OXH 输出将持续保持为低电平。

in\_opxl 的低电平脉冲宽度小于 PDAR 设定的死区时间时，仅 TIM4\_<t>\_OXH 输出变为低电平。TIM4\_<t>\_OXH 输出电平变为高的条件是当 in\_opxl 上升沿之后经过 PDBR 寄存器设定的死区时间之后。在此情况下，TIM4\_<t>\_OXL 输出将持续保持为低电平 ( $X=U, V, W$ ,  $x=u, v, w$ )。如下图 21-21 所示。

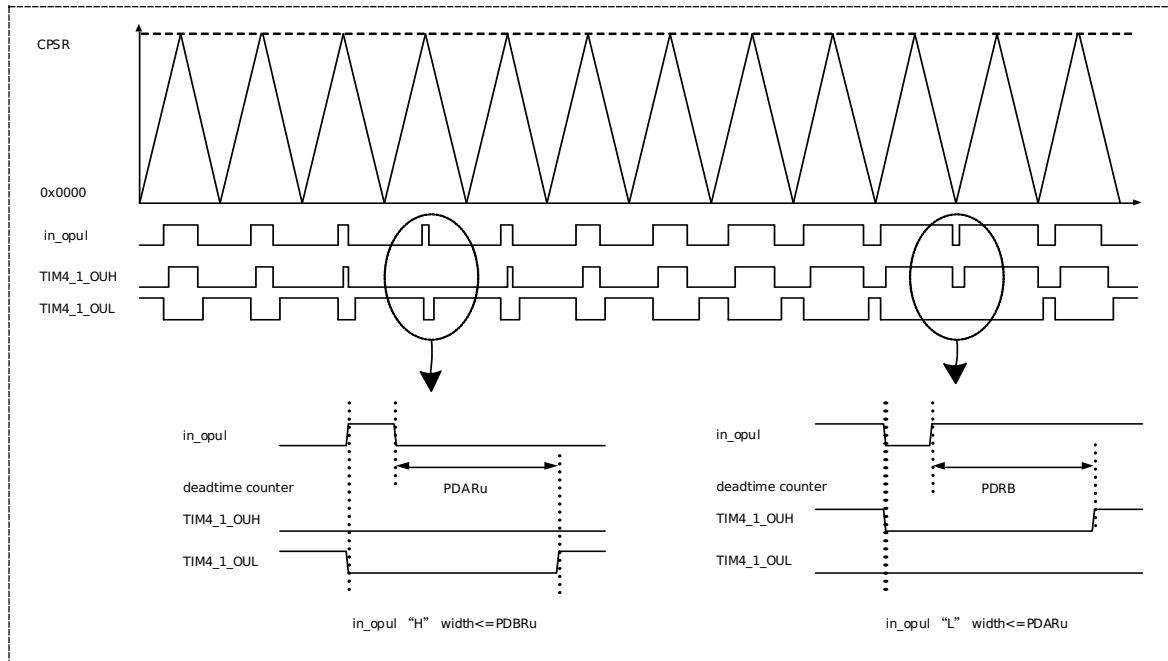


图 21-21 脉宽异常时的死区定时器模式下的波形输出

在上述硬件实现死区输出方式的基础上，还可以对内部的比较匹配信号 (in\_opxl) 进行脉冲宽度的监测，从而实现对 in\_opxl 信号的滤波控制。这种 in\_opxl 带脉宽滤波的死区输出实现方式叫死区计数器滤波模式 (POCR.PWMMD=10) ( $x=u, v, w$ )。

在死区计数器滤波模式下，滤波宽度由 PWM 滤波控制寄存器 (PFSRn) 的设定值决定。当 in\_opxl 的脉冲宽度大于寄存器 PFSRn 设定的时间时，滤波计数器将 in\_opxl 信号推迟 PFSR 设定的时间后输出，再经过死区定时器模式所描述的方式产生互补 PWM 输出 ( $x=u, v, w$ )。

如果检测到信号 in\_opxl 上升沿，滤波计数器加载 PFSR 寄存器值并启动计量 in\_opxl 的高电平宽度，当 in\_opxl 的高电平脉冲宽度大于寄存器 PFSR 设定的时间时，经过 PFSR 所设定时间之后，TIM4\_<t>\_OXL 输出变为低电平，死区计数器加载 PDBR 寄存器的设定值并启动递减计数，当计数值变为 0x0000 时，计数器停止并使 TIM4\_<t>\_OXH 输出高电平；如果检测到信号 in\_opxl 下降沿，滤波计数器加载 PFSR 寄存器值并启动计量 in\_opxl 的低电平宽度，当 in\_opxl 的低电平脉冲宽度大于寄存器 PFSR 设定的时间时，经过 PFSR 所设定时间之后，TIM4\_<t>\_OXH 输出变为低电平，死区计数器加载 PDAR 寄存器的设定值并启动递减计数，当计数值变为 0x0000 时，计数器停止并使 TIM4\_<t>\_OXL 输出变为低电平。

出高电平。当 in\_opxl 的电平脉冲宽度小于寄存器 PFSR 设定的时间时，输出 TIM4\_<t>\_OXH 和 TIM4\_<t>\_OXL 将保持不变（X=U、V、W，x=u、v、w）。

图 21-22 所示，是死区计数器滤波模式下的互补 PWM 输出例。

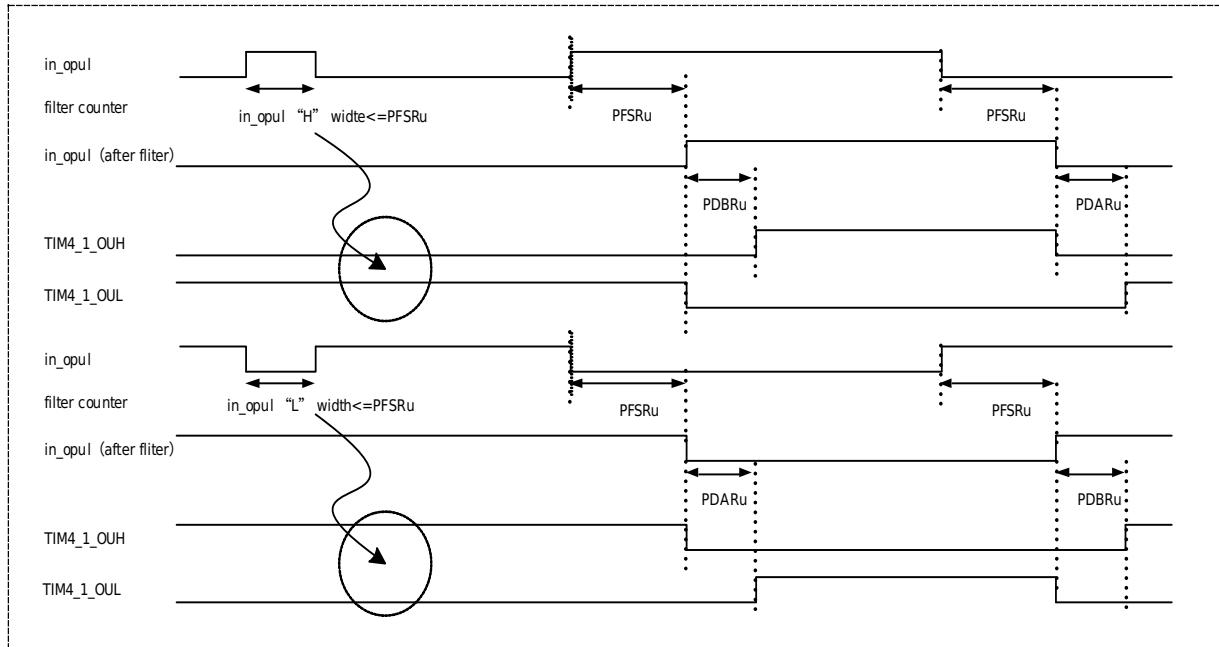


图 21-22 死区定时器滤波模式下的互补 PWM 输出

#### 注意：

- 使用软件或硬件互补 PWM 输出模式前，需要设定 MOE、OE<sub>xy</sub> 位（寄存器 PSCR）以使能 Timer4 的输出（x=u、v、w，y=h、l），详细描述请参考【PWM 状态控制寄存器】。

#### 21.3.4 周期间隔响应

下溢中断屏蔽计数器用于降低（掩盖）下溢标志位（CCSR.IRQZF）的被置位次数。下溢中断屏蔽计数器（CVPR.ZIC[3:0]）作为递减计数器操作，在开始时载入 CVPR.ZIM[3:0]所设定的值，当 CVPR.ZIC[3:0] = “0”时，下溢标志位(CCSR.IRQZF)被置位为“1”。

上溢中断屏蔽计数器用于降低（掩盖）上溢标志位（CCSR.IRQPF）的被置位次数。上溢中断屏蔽计数器（CVPR.PIC[3:0]）作为递减计数器操作，在开始时载入 CVPR.PIM[3:0]所设定的值，当 CVPR.PIC[3:0] = “0”时，上溢标志位(CCSR.IRQPF)被置位为“1”。

如下图 21-23 所示，是周期间隔响应时的 IRQZF、IRQPF 的置位时序图。

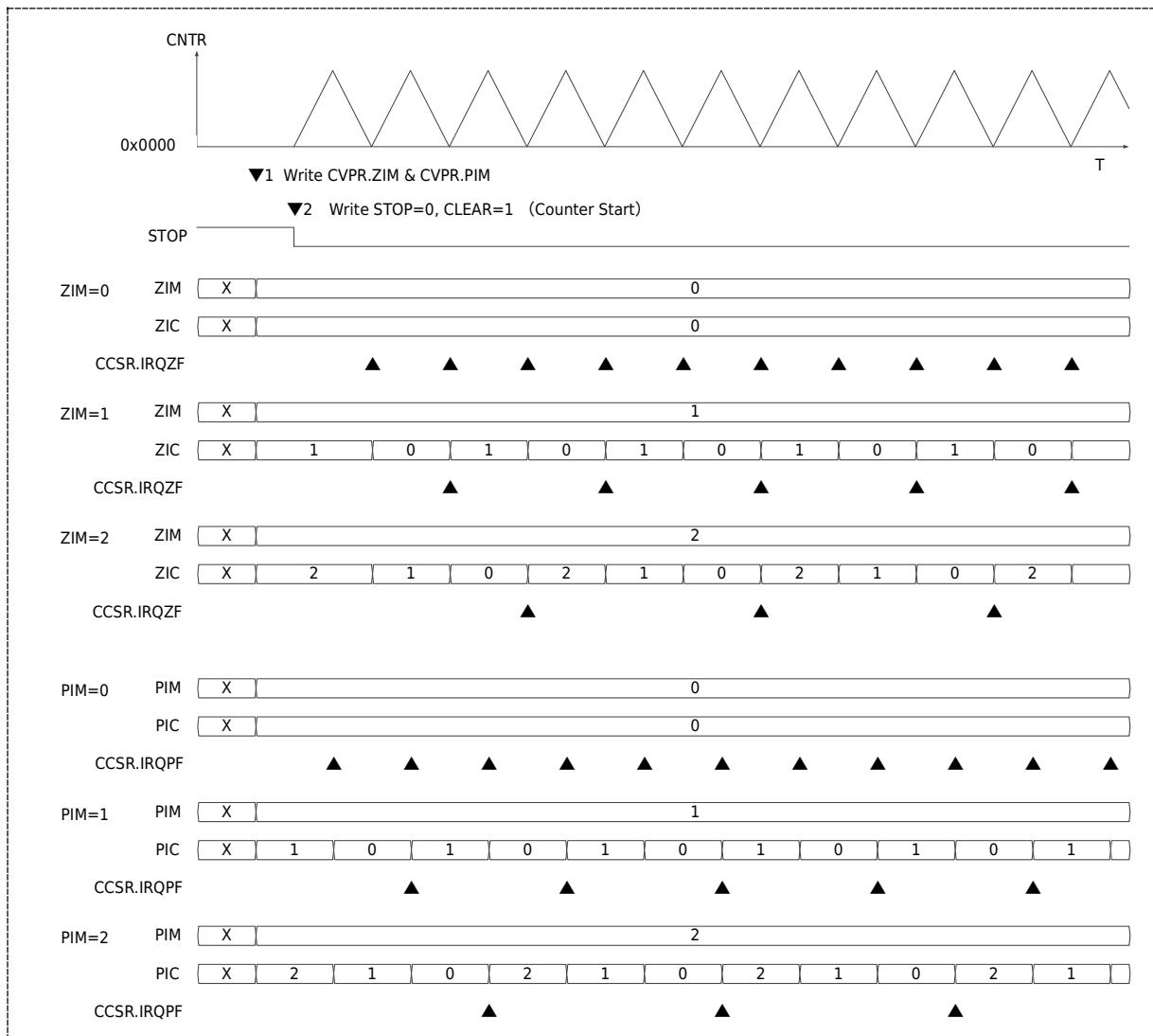


图 21-23 周期间隔响应时序图

▼1 计数器停止时，写入 CVPR.ZIM 和 CVPR.PIM 初始值，初始值立即反映在内部计数器(CVPR.ZIC、CVPR.PIC)。

▼2 初始化并启动计数器(STOP=0 且 CLEAR=1)，计数器在总线复位或者从软件初始化 CLEAR=1 后从零开始进行递增计数，此刻 CCSR.IRQZF 标志不会立即置位，之后每当中断屏蔽计数器的计数值为 0x0000 且计数器的计数值为 0x0000 和 CPSR 时，标志▲为 CCSR.IRQZF 或 CCSR.IRQPF 置位的时刻。

#### 注意：

- 在计数器运行中，写入 CVPR.ZIM 和 CVPR.PIM，该设置值不会立即反应到中断屏蔽计数器 (CVPR.ZIC 和 CVPR.PIC) 中。如果写入软复位 (CLEAR=1)，写入的 CVPR.ZIM 和 CVPR.PIM 值将立即加载为中断屏蔽计数器的初值。

专用比较基准寄存器 (SCCRm) 的比较匹配事件 (专用事件输出) 也具有周期间隔响应功能。如下图 21-24 所示, 是专用事件输出信号的周期间隔响应输出图。

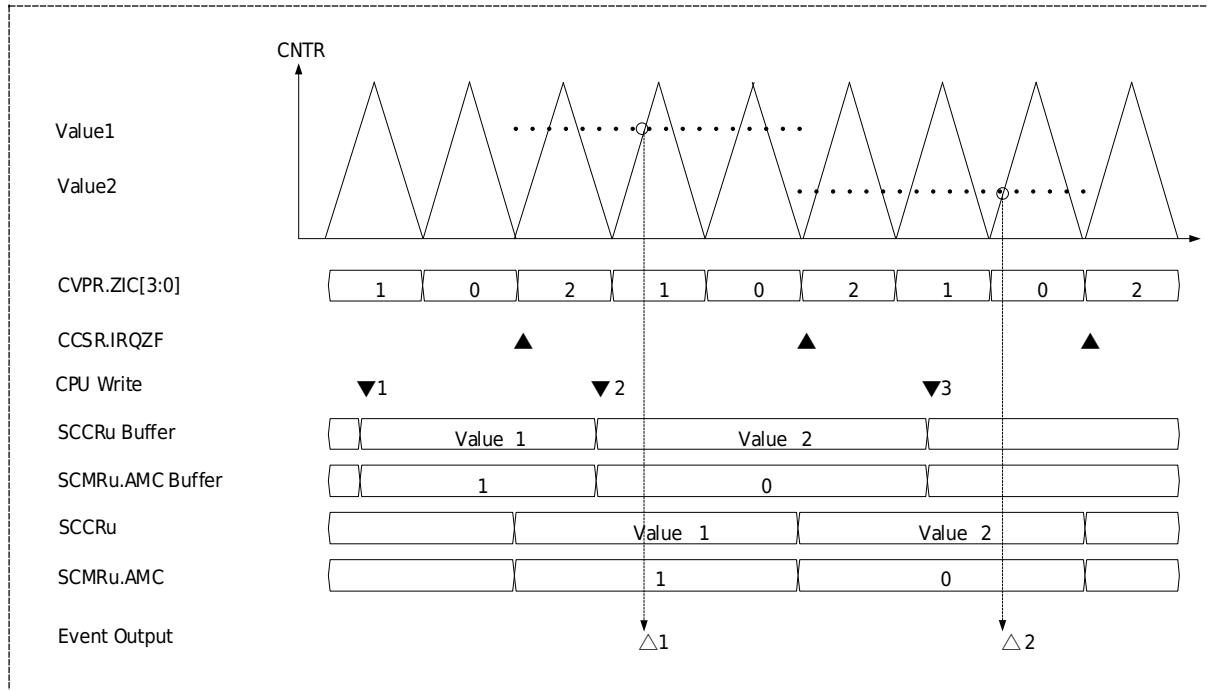


图 21-24 专用事件输出信号周期间隔响应输出

计数器是三角波计数模式,下溢中断屏蔽计数器(CVPR.ZIC)从 2-0 做递减计数。下溢中断在时刻▲产生。

在时刻▼1, 向 SCCR 缓冲寄存器写入 value 1。同时, MZCE=1、MPCE=0、AMC=0001 写入 SCMR 缓冲寄存器。之后, 缓冲寄存器 SCCR 和 SCMR 传送操作在▲时刻执行。因为 MZCE=1 且 AMC=1 且 计数值=SCCR=Value 1, 在时刻△1, 专用事件输出信号置位。

在时刻▼2, 向 SCCR 缓冲寄存器写入 value 2。同时, MZCE=1、MPCE=0、AMC=0000 写入 SCMR 缓冲寄存器。之后, 缓冲寄存器 SCCR 和 SCMR 传送操作在▲时刻执行。因为 MZCE=1 且 AMC=0 且 计数值=SCCR=Value 2, 在时刻△2, 专用事件输出信号置位。

### 21.3.5 EMB 控制

每个 Timer4 单元都有一个输出无效事件接口, 连接 EMB 模块输出的 EMB 事件。该接口上选通的异常状况事件可从 EMB 侧设定 (参见【紧急刹车模块 (EMB)】章节)。

各单元内 3 组 PWM 端口在正常输出期间, 若监测到从 EMB 过来的 EMB 异常事件, PSCR.MOE 位由硬件异步清零, 各 PWM 端口的输出状态变为预先设定好的状态。根据 PSCR.OSxy 的设定, 该预设端口状态可以为输出高阻态、输出低电平或输出高电平 ( $x=U, V, W, y=H, L$ )。

例如, 若设定 PSCR.OSUH=01 时, 在 Timer4 的 TIM4\_<t>\_OUH 端口正常输出期间产生 EMB 事件, 则 TIM4\_<t>\_OUH 端口上输出变为高阻态。

触发 EMB 异常事件的条件无效后，用户可通过写 EMB 状态复位寄存器(EMB\_STATCLR0)来清除 EMB 异常事件，若此时 PSCR.AOE=1 则 PSCR.MOE 位将在下一个计数下溢点由硬件自动置 1（此处的下溢点为计数器的实际计数零点，不受寄存器 CVPR 的设定值影响），随后 PWM 端口恢复到正常输出；若此时 PSCR.AOE=0 则需要软件置位 PSCR.MOE，以恢复 PWM 端口到正常输出。

需要注意，硬件上控制 PWM 输出使能使用的是 MOE 的异步信号，当 MOE 的值发生变化后，需要等待几个时钟周期才能在总线上读取到实际的值。

关于 EMB 的详细操作方法，请参考【紧急刹车模块（EMB）】章节的描述。

### 21.3.6 监测输出

每个 Timer4 单元都有一个 PWM 周期输出监测端口 TIM4\_<t>\_PCT 和专用事件输出监测端口 TIM4\_<t>\_ADSM。这些监测端口输出到外部，用户可根据端口的变化信息对应用系统进行灵活地控制。

PWM 周期输出监测端口 TIM4\_<t>\_PCT 用来监测当前计数器的计数状态。在三角波模式时，每次计数到波峰或波谷时，监测端口 TIM4\_<t>\_PCT 上就发生一次翻转。图 21-25 是计数方向信号的示例。

专用事件输出监测端口 TIM4\_<t>\_ADSM 可以用来监测内部计数器的专用比较事件。通过专用扩展控制寄存器（SCER）的 SCER.EVTRS 位选择需要监测的专用比较输出通道，在该通道上的比较基准值和计数器产生比较匹配时，监测端口 TIM4\_<t>\_ADSM 上就会置位（输出为高）；在该通道上的计数器计数归零时（锯齿波溢出和三角波计数到波谷），监测端口 TIM4\_<t>\_ADSM 上就会复位（输出为低）。

需要注意，当配置寄存器 TMR4\_SCSRm 同时使能向上计数 EVT 和向下计数 EVT 条件下（TMR4\_SCSRm.UEN=1 且 TMR4\_SCSRm.DEN=1），输出到外部端口的同步信号只对向上计数 EVT 有效。

图 21-26 是比较启动模式下（SCSRm.EVTMS=0）输出到端口的同步信号示例，向上计数 EVT 有效后 TIM4\_<t>\_ADSM 置位，计数器下溢后 TIM4\_<t>\_ADSM 清零。

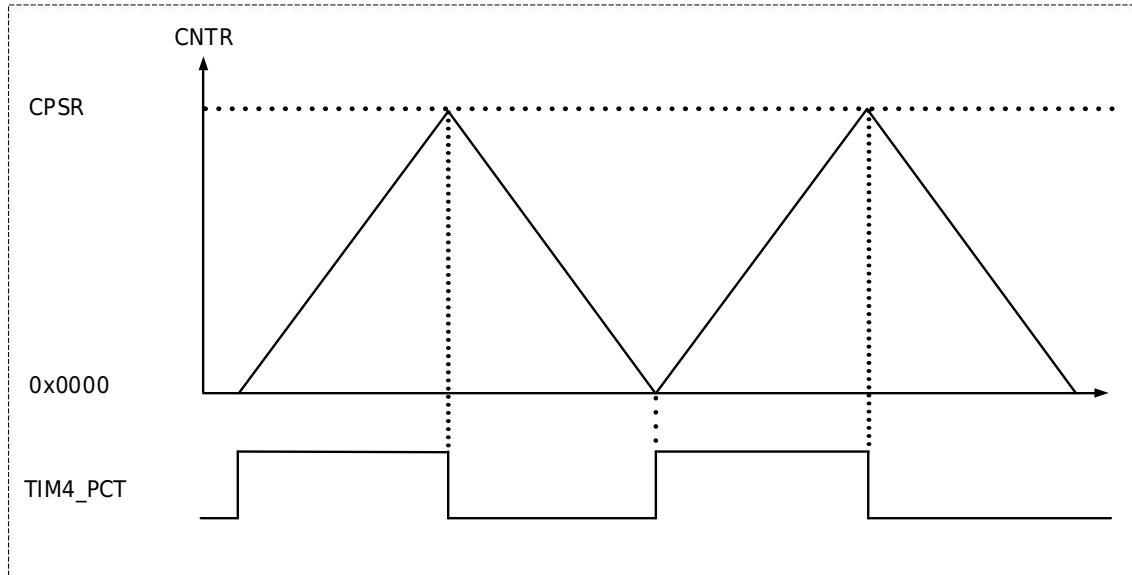


图 21-25 PWM 周期输出监测示例

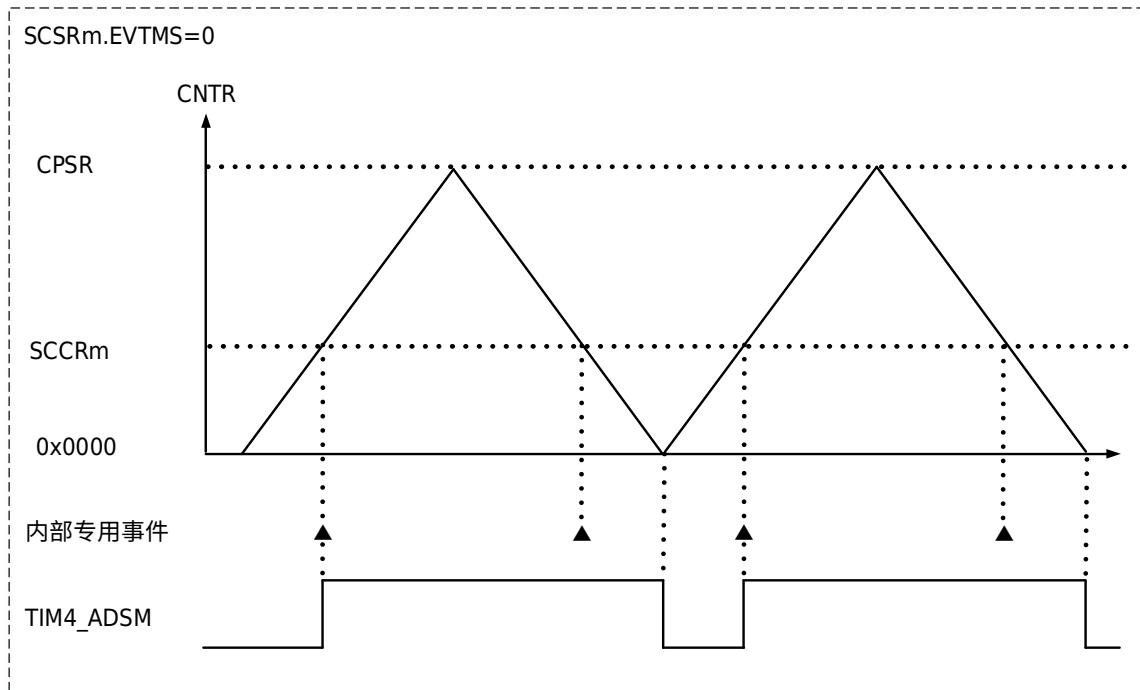


图 21-26 比较启动模式下专用事件输出 TIM4\_&lt;t&gt;\_ADSM 时序

### 21.3.7 硬件启动

Timer4 可以通过内部硬件触发事件来实现计数器的启动功能。

Timer4 有一个内部硬件触发源，事件源可通过内部触发事件选择寄存器（HTSSR）中对应的编号设定来选择，具体的事件对应关系请参考 INTC 章节。使用内部硬件触发功能时，需要先将功能时钟控制寄存器（PWR\_FCG0）的外围电路触发功能使能位置 1。

## 21.4 中断及事件说明

### 21.4.1 通用计数比较匹配中断及事件

通用比较基准值寄存器（OCCRm）共计 6 个，可分别与计数值比较产生比较匹配有效信号。计数比较匹配时，通用控制状态寄存器（OCSRn）中的 OCSRn.OCFH、OCSRn.OCFL 位分别会被置为 1。此时若设定 OCSRn.OCIEH、OCSRn.OCIEL 使能中断，则对应的中断请求（TMR4\_<t>\_GCMmn, m=U、V、W；n=H、L）也会被触发。

通用计数比较匹配事件可以用于触发别的模块，事件编号请参考【中断控制器（INTC）】【中断事件表】章节。

### 21.4.2 计数周期匹配中断及事件

锯齿波递加计数至上溢点、锯齿波递减计数至下溢点、三角波计数至谷点或三角波计数至峰点时，控制状态寄存器（CCSR）的 CCSR.IRQPF 或 CCSR.IRQZF 位会被置为 1。此时若设置 CCSR.IRQPEN 或 CCSR.IRQZEN 位使能中断，则在对应的时间点可触发计数周期匹配中断（TMR4\_<t>\_GOVF 和 TMR4\_<t>\_GUDF）。

计数周期匹配事件可以用于触发别的模块，事件编号请参考【中断控制器（INTC）】【中断事件表】章节。

### 21.4.3 重载计数匹配中断及事件

在重载功能有效时，PWM 滤波控制寄存器（PFSRn）与计数值比较产生比较匹配信号。计数比较匹配时，重载控制状态寄存器（RCSR）中的 RCSR.RTIFU、RCSR.RTIFV、RCSR.RTIFW 位分别会被置为 1。此时若设定 RCSR.RTIDU、RCSR.RTIDV、RCSR.RTIDW 中断屏蔽无效，则对应的重载计数匹配中断请求（TMR4\_<t>\_RLOm, m=U、V、W）也会被触发。

重载计数匹配事件可以用于触发别的模块，事件编号请参考【中断控制器（INTC）】【中断事件表】章节。

### 21.4.4 专用比较匹配事件

Timer4 的 6 个专用比较基准寄存器（SCCRm）对应产生 6 个专用事件输出信号可以用于选择触发别的模块，如启动 ADC 等。

时钟计数过程中，若专用比较基准值（SCCRm）发生计数比较匹配事件，产生相应的有效请求信号，该请求信号可以被配置到任意的事件 EVT 输出信号上（由 SCSR.EVTOS 位设定）用于触发其它模块。此时若设定 SCIR.ITENO~5 使能中断，则对应的专用比较匹配中断请求（TMR4\_1\_SCM0~5）也会被触发。

该事件请求信号的输出可以选择比较启动模式或延时启动模式。在比较启动模式时 (SCSR.EVTMS=0)，产生 SCCR 的计数比较匹配事件后，专用事件输出信号直接输出；在延时启动模式时 (SCSR.EVTMS=1)，产生 OCCRxh 或 OCCRxI (由 SCSR.EVTDS 位选择；x=u、v、w) 的计数比较匹配事件后，经过 SCCR 设定的基准周期时间后，专用事件输出信号输出。下图 21-27 所示，是延时启动模式下的专用事件输出信号的请求输出例。

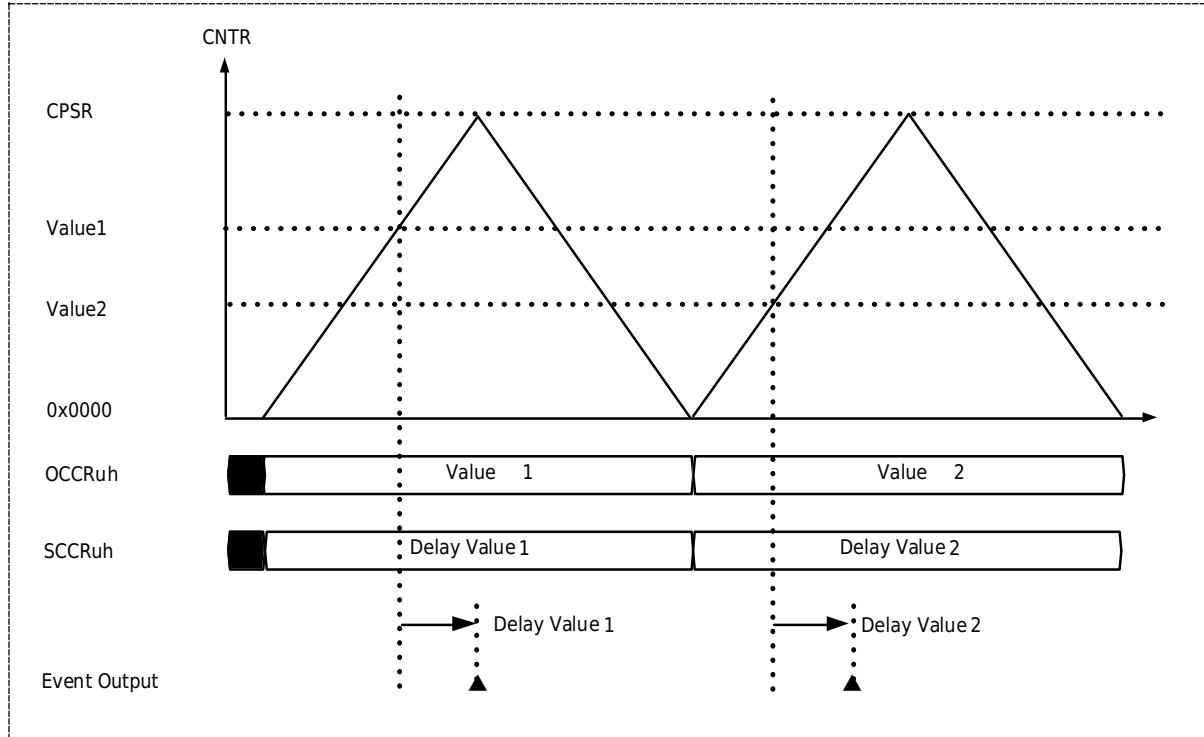


图 21-27 延时启动模式下专用事件输出信号的输出时序

**注意：**

- 在延迟计数运行中，如果再次发生 OCCR 与计数器匹配事件，延迟计数器重新加载计数值并重新进行递减计数。因此，如果 OCCR 匹配事件时间间隔小于设定的延迟时间 SCCR 时，专用事件输出的请求信号可能一直不会产生。

## 21.5 寄存器说明

表 21-3 所示，为 Timer4 模块的寄存器列表。

表 21-3 寄存器列表

BASE ADDR: 0x4003\_8000 (U1)

寄存器名	符号	偏移量	位宽	复位值
计数值寄存器	TMR4_CNTR	0x0044	16	0x0000
周期基准寄存器	TMR4_CPSR	0x0040	16	0xFFFF
控制状态寄存器	TMR4_CCSR	0x0048	16	0x0040
有效周期寄存器	TMR4_CVPR	0x004A	16	0x0000
PWM状态控制寄存器	TMR4_PSCR	0x004C	32	0x05550000
通用比较基准寄存器UH	TMR4_OCCRuh	0x0000	16	0x0000
通用比较基准寄存器UL	TMR4_OCCRul	0x0004	16	0x0000
通用比较基准寄存器VH	TMR4_OCCRvh	0x0008	16	0x0000
通用比较基准寄存器VL	TMR4_OCCRvl	0x000C	16	0x0000
通用比较基准寄存器WH	TMR4_OCCRwh	0x0010	16	0x0000
通用比较基准寄存器WL	TMR4_OCCRwl	0x0014	16	0x0000
通用控制状态寄存器U	TMR4_OCSRu	0x0018	16	0xFF00
通用控制状态寄存器V	TMR4_OCSRv	0x001C	16	0xFF00
通用控制状态寄存器W	TMR4_OCSRw	0x0020	16	0xFF00
通用扩展控制寄存器U	TMR4_OCERu	0x001A	16	0x0000
通用扩展控制寄存器V	TMR4_OCERv	0x001E	16	0x0000
通用扩展控制寄存器W	TMR4_OCERw	0x0022	16	0x0000
通用模式控制寄存器UH	TMR4_OCMRuh	0x0024	16	0x0000
通用模式控制寄存器UL	TMR4_OCMRul	0x0028	32	0x00000000
通用模式控制寄存器VH	TMR4_OCMRvh	0x002C	16	0x0000
通用模式控制寄存器VL	TMR4_OCMRvl	0x0030	32	0x00000000
通用模式控制寄存器WH	TMR4_OCMRwh	0x0034	16	0x0000
通用模式控制寄存器WL	TMR4_OCMRwl	0x0038	32	0x00000000
专用比较基准寄存器UH	TMR4_SCCRuh	0x00B0	16	0x0000
专用比较基准寄存器UL	TMR4_SCCRul	0x00B4	16	0x0000
专用比较基准寄存器VH	TMR4_SCCRvh	0x00B8	16	0x0000
专用比较基准寄存器VL	TMR4_SCCRvl	0x00BC	16	0x0000
专用比较基准寄存器WH	TMR4_SCCRwh	0x00C0	16	0x0000
专用比较基准寄存器WL	TMR4_SCCRwl	0x00C4	16	0x0000
专用控制状态寄存器UH	TMR4_SCSRuh	0x00C8	16	0x0000
专用控制状态寄存器UL	TMR4_SCSRul	0x00CC	16	0x0000
专用控制状态寄存器VH	TMR4_SCSRvh	0x00D0	16	0x0000

寄存器名	符号	偏移量	位宽	复位值
专用控制状态寄存器VL	TMR4_SCSRvl	0x00D4	16	0x0000
专用控制状态寄存器WH	TMR4_SCSRwh	0x00D8	16	0x0000
专用控制状态寄存器WL	TMR4_SCSRwl	0x00DC	16	0x0000
专用模式控制寄存器UH	TMR4_SCMRuh	0x00CA	16	0xFF00
专用模式控制寄存器UL	TMR4_SCMRul	0x00CE	16	0xFF00
专用模式控制寄存器VH	TMR4_SCMRvh	0x00D2	16	0xFF00
专用模式控制寄存器VL	TMR4_SCMRvl	0x00D6	16	0xFF00
专用模式控制寄存器WH	TMR4_SCMRwh	0x00DA	16	0xFF00
专用模式控制寄存器WL	TMR4_SCMRwl	0x00DE	16	0xFF00
专用扩展控制寄存器	TMR4_SCER	0x00E0	16	0xFF00
PWM基本控制寄存器U	TMR4_POCRu	0x0098	16	0xFF00
PWM基本控制寄存器V	TMR4_POCRv	0x009C	16	0xFF00
PWM基本控制寄存器W	TMR4_POCRw	0x00A0	16	0xFF00
PWM滤波控制寄存器U	TMR4_PFSRu	0x0082	16	0x0000
PWM滤波控制寄存器V	TMR4_PFSRv	0x008A	16	0x0000
PWM滤波控制寄存器W	TMR4_PFSRw	0x0092	16	0x0000
PWM死区控制寄存器AU	TMR4_PDARu	0x0084	16	0x0000
PWM死区控制寄存器BU	TMR4_PDBRu	0x0086	16	0x0000
PWM死区控制寄存器AV	TMR4_PDARv	0x008C	16	0x0000
PWM死区控制寄存器BV	TMR4_PDBRv	0x008E	16	0x0000
PWM死区控制寄存器AW	TMR4_PDARw	0x0094	16	0x0000
PWM死区控制寄存器BW	TMR4_PDBRw	0x0096	16	0x0000
重载控制状态寄存器	TMR4_RCSR	0x00A4	16	0x0000
专用中断控制寄存器	TMR4_SCIR	0x00E4	16	0xFF00
专用标志控制寄存器	TMR4_SCFR	0x00E8	16	0xFF00

**注意：**

- 下述寄存器详细描述中，m=uh、ul、vh、vl、wh、wl，n=u、v、w。m 所指寄存器分别在功能实现时对应端口 TIM4\_<t>\_OUH、TIM4\_<t>\_OUL、TIM4\_<t>\_OVH、TIM4\_<t>\_OVL、TIM4\_<t>\_OWH、TIM4\_<t>\_OWL 的输出控制等；n 所指寄存器分别在功能实现时对应端口 TIM4\_<t>\_OUx、TIM4\_<t>\_OVx、TIM4\_<t>\_OWx 的输出控制等，其中 x=H 或 L，且 H 或 L 的具体控制在这些寄存器中有相对应的对称位。

### 21.5.1 计数值寄存器 (TMR4\_CNTR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNTR[15:0]															
位	标记	位名		功能											读写
b15~b0	CNTR[15:0]	计数器当前值		在计数停止时，通过向该寄存器写入值，可初始化计数器计数值 在计数中时，该位指示当前计数器计数值 注：在计数中时，不能向该寄存器写入值											R/W

### 21.5.2 周期基准寄存器 (TMR4\_CPSR)

复位值: 0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CPSR[15:0]															
位	标记	位名		功能											读写
b15~b0	CPSR[15:0]	通用周期基准值		计数器的计数周期值 注：从本地址区域读取数据时，读取的并非缓冲器寄存器的值，而是CPSR寄存器的值											R/W

### 21.5.3 控制状态寄存器 (TMR4\_CCSR)

复位值: 0x0040

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ECKEN	-	HST	-	IRQZF	IRQZEN	IRQPF	IRQOPEN	IRQPEN	BUFEN	STOP	MODE	CLEAR			CKDIV[3:0]

位	标记	位名	功能	读写
b15	ECKEN	时钟源选择	0: 内部PCLK1时钟 1: 外部TIM4_<t>_CLK端口输入时钟  注1: 该位在计数器停止时设定 注2: 使用外部输入时钟进行操作的场合, 计数器操作开始后, 最初的外部输入时钟的边沿, 无论是上升沿还是下降沿都被忽略, 计数操作从之后的边沿开始	R/W
b14	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b13	HST	硬件启动使能	条件: TMR4_TRGSEL寄存器中指定的事件发生 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效, 计数器开始计数	R/W
b12	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b11	IRQZF	下溢状态	0: 未发生计数下溢 1: 发生计数下溢  注1: 当使用周期间隔响应功能时, 该位的置位条件由CVPR所设定的周期间隔计数器设定 注2: 当计数器被总线复位或者写入CLEAR=“1”, IRQZF位将不会被置位	R/W
B10	IRQZEN	下溢中断使能	0: 禁止IRQZF产生中断到CPU 1: 允许IRQZF产生中断到CPU	R/W
b9	IRQPF	上溢状态	0: 未发生计数上溢 1: 发生计数上溢  注1: 当使用周期间隔响应功能时, 该位的置位条件由CVPR所设定的周期间隔计数器设定。 注2: 当计数器被总线复位或者写入CLEAR=“1”, IRQZF位将不会被置位	R/W
b8	IRQOPEN	上溢中断使能	0: 禁止IRQPF产生中断到CPU 1: 允许IRQPF产生中断到CPU	R/W
b7	BUFEN	缓存使能	0: 禁止CPSR缓存功能 1: 使能CPSR缓存功能	R/W
b6	STOP	计数器使能	0: 计数器启动 1: 计数器停止	R/W
b5	MODE	波形模式	0: 锯齿波模式 (仅支持向上计数) 1: 三角波模式	R/W
b4	CLEAR	计数器清零	0: 无操作 1: 计数器清零  注: 该位读出时始终为0	R/W
b3~b0	CKDIV	计数时钟分频	该位指示基本计数器的计数时钟分频 0000: 计数时钟为PCLK0 0001: 计数时钟为PCLK0/2	R/W

0010: 计数时钟为PCLK0/4

0011: 计数时钟为PCLK0/8

0100: 计数时钟为PCLK0/16

0101: 计数时钟为PCLK0/32

0110: 计数时钟为PCLK0/64

0111: 计数时钟为PCLK0/128

1000: 计数时钟为PCLK0/256

1001: 计数时钟为PCLK0/512

1010: 计数时钟为PCLK0/1024

其他: 设定禁止

注: 时钟源为内部PCLK0时, 分频有效

---

### 21.5.4 有效周期寄存器 (TMR4\_CVPR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIC[3:0]				ZIC[3:0]				PIM[3:0]				ZIM[3:0]			

位	标记	位名	功能	读写
b15~b12	PIC[3:0]	上溢中断屏蔽状态	当前被屏蔽的上溢中断个数	R
b11~b8	ZIC[3:0]	下溢中断屏蔽状态	当前被屏蔽的下溢中断个数	R
b7~b4	PIM[3:0]	上溢中断屏蔽设定	设置屏蔽的上溢中断个数	R/W
b3~b0	ZIM[3:0]	上溢中断屏蔽设定	设置屏蔽的下溢中断个数	R/W

### 21.5.5 通用比较基准寄存器 (TMR4\_OCCRm)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OCCR[15:0]															

位	标记	位名	功能	读写
b15~b0	OCCR[15:0]	通用比较基准值	通用比较基准值	R/W
			注: 从本地址区域读取数据时, 读取的并非缓冲器寄存器的值, 而是OCCR寄存器的值	

### 21.5.6 通用控制状态寄存器 (TMR4\_OCSRn)

复位值: 0xFF00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		
Reserved								OCFL	OCFH	OCIEL	OCIEH	OCPL	OCPH	OCEL	OCEH		
b15~b8	Reserved								读写								
b15~b8	读出时为“1”,写入时写“1”								R/W								
b7	OCFL	0: 计数器计数值与OCCRxl设定值不相等 1: 计数器计数值与OCCRxl设定值相等 (x=u、v、w) 注: 该位必须在OCEL=1时有效								R/W							
b6	OCFH	0: 计数器计数值与OCCRxh设定值不相等 1: 计数器计数值与OCCRxh设定值相等 (x=u、v、w) 注: 该位必须在OCEH=1时有效								R/W							
b5	OCIEL	0: OCFL置位时, 不发生中断 1: OCFL置位时, 发生中断								R/W							
b4	OCIEH	0: OCFH置位时, 不发生中断 1: OCFH置位时, 发生中断								R/W							
b3	OCPL	0: OCEL=0时, TIM4_<t>_OxL上输出低电平 1: OCEL=0时, TIM4_<t>_OxL上输出高电平 (m=U、V、W)								R/W							
b2	OCPH	0: OCEH=0时, TIM4_<t>_OxH上输出低电平 1: OCEH=0时, TIM4_<t>_OxH上输出高电平 (m=U、V、W)								R/W							
b1	OCEL	0: 比较输出无效, TIM4_<t>_OxL端口状态由OCPL决定 1: 比较输出有效, TIM4_<t>_OxL端口状态由OCMRyl的设定和 OCFL状态决定 (x=U、V、W, y=u、v、w)								R/W							
b0	OCEH	0: 比较输出无效, TIM4_<t>_OxH端口状态由OCPH决定 1: 比较输出有效, TIM4_<t>_OxH端口状态由OCMRyh的设定和 OCFH状态决定 (x=U、V、W, y=u、v、w)								R/W							

### 21.5.7 通用扩展控制寄存器 (TMR4\_OCERn)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	MCECL	MCECH	LMM_L	LMM_H	LMC_L	LMC_H	MLBUFEN[1:0]	MHBUFEN[1:0]	CLBUFEN[1:0]	CHBUFEN[1:0]				
位	标记	位名	功能												读写
b15~b14	Reserved	-	读出时为“0”,写入时写“0”												R/W
b13	MCECL	扩展控制使能L	(x=u、v、w) 注1：只能在输出比较功能无效 (OCSR.OCEL=0) 时修改该位 注2：具体请参考TMR4_OCMRm寄存器说明的注意事项												R/W
b12	MCECH	扩展控制使能H	(x=u、v、w) 注1：只能在输出比较功能无效 (OCSR.OCEH=0) 时修改该位 注2：具体请参考TMR4_OCMRm寄存器说明的注意事项												R/W
b11	LMML	周期间隔响应功能链接L	0：禁止计数值与OCCRxl匹配条件的扩展控制功能 1：使能计数值与OCCRxl匹配条件的扩展控制功能 注1：只能在输出比较功能无效 (OCSR.OCEL=0) 时修改该位 注2：具体请参考TMR4_OCMRm寄存器说明的注意事项												R/W
b10	LMMH	周期间隔响应功能链接H	0：周期间隔响应功能链接无效，OCMRxl的缓存传送由MLBUFEN设定决定 1：周期间隔响应功能链接有效，OCMRxl的缓存传送在MLBUFEN设定的基础上，还必须满足CVPR.PIC[3:0]=0000 (计数上溢时) 或CVPR.ZIC[3:0]=0000 (计数下溢时) (x=u、v、w)												R/W
b9	LMCL	周期间隔响应功能链接L	0：周期间隔响应功能链接无效，OCMRxl的缓存传送由MLBUFEN设定决定 1：周期间隔响应功能链接有效，OCMRxl的缓存传送在MLBUFEN设定的基础上，还必须满足CVPR.PIC[3:0]=0000 (计数上溢时) 或CVPR.ZIC[3:0]=0000 (计数下溢时) (x=u、v、w)												R/W
b8	LMCH	周期间隔响应功能链接H	0：周期间隔响应功能链接无效，OCMRxl的缓存传送由MLBUFEN设定决定 1：周期间隔响应功能链接有效，OCMRxl的缓存传送在MLBUFEN设定的基础上，还必须满足CVPR.PIC[3:0]=0000 (计数上溢时) 或CVPR.ZIC[3:0]=0000 (计数下溢时) (x=u、v、w)												R/W
b7~b6	MLBUFEN[1:0]	OCMRxl缓存传送	00：OCMRxl缓存寄存器的值直接写入OCMRxl 01：OCMRxl缓存寄存器的值在计数下溢时写入OCMRxl 10：OCMRxl缓存寄存器的值在计数上溢时写入OCMRxl 11：OCMRxl缓存寄存器的值在计数下溢或上溢时写入OCMRxl (x=u、v、w)												R/W

b5~b4	MHBUFEN[1:0]	OCMRxh缓存转送	00: OCMRxh缓存寄存器的值直接写入OCMRxh 01: OCMRxh缓存寄存器的值在计数下溢时写入OCMRxh 10: OCMRxh缓存寄存器的值在计数上溢时写入OCMRxh R/W 11: OCMRxh缓存寄存器的值在计数下溢或上溢时写入OCMRxh (x=u、v、w)
b3~b2	CLBUFEN[1:0]	OCCRxl缓存传送	00: OCCRxl缓存寄存器的值直接写入OCCRxl 01: OCCRxl缓存寄存器的值在计数下溢时写入OCCRxl 10: OCCRxl缓存寄存器的值在计数上溢时写入OCCRxl R/W 11: OCCRxl缓存寄存器的值在计数下溢或上溢时写入OCCRxl (x=u、v、w)
b1~b0	CHBUFEN[1:0]	OCCRxh缓存传送	00: OCCRxh缓存寄存器的值直接写入OCCRxh 01: OCCRxh缓存寄存器的值在计数下溢时写入OCCRxh 10: OCCRxh缓存寄存器的值在计数上溢时写入OCCRxh R/W 11: OCCRxh缓存寄存器的值在计数下溢或上溢时写入OCCRxh (x=u、v、w)

## 21.5.8 通用模式控制寄存器 (TMR4\_OCMRm)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OPN ZRH[1:0]	OPN PKH[1:0]	OP ZRH[1:0]	OP UCH[1:0]	OP PKH[1:0]	OP DCH[1:0]	OP ZRH	OCF UCH	OCF PKH	OCF DCH						

注：该寄存器位描述用于 OCMRuh、OCMRvh、OCMRwh 时

位	标记	位名	功能	读写
b15~b14	OPNZRH[1:0]	下溢点端口状态H	条件：计数下溢 & OCCRxh计数不匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyH端口位保持不变 01: 满足条件时, TIM4_<t>_OyH端口位输出高电平 10: 满足条件时, TIM4_<t>_OyH端口位输出低电平 11: 满足条件时, TIM4_<t>_OyH端口位输出反转 (y=U、V、W)	R/W
b13~b12	OPNPKH[1:0]	上溢点端口状态H	条件：计数上溢 & OCCRxh计数不匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyH端口位保持不变 01: 满足条件时, TIM4_<t>_OyH端口位输出高电平 10: 满足条件时, TIM4_<t>_OyH端口位输出低电平 11: 满足条件时, TIM4_<t>_OyH端口位输出反转 (y=U、V、W)	R/W
b11~b10	OPZRH[1:0]	下溢点端口状态H	条件：计数下溢 & OCCRxh计数匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyH端口位保持不变 01: 满足条件时, TIM4_<t>_OyH端口位输出高电平 10: 满足条件时, TIM4_<t>_OyH端口位输出低电平 11: 满足条件时, TIM4_<t>_OyH端口位输出反转 (y=U、V、W)	R/W
b9~b8	OPUCH[1:0]	向上计数端口状态H	条件：计数器向上计数 & OCCRxh计数匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyH端口位保持不变 01: 满足条件时, TIM4_<t>_OyH端口位输出高电平 10: 满足条件时, TIM4_<t>_OyH端口位输出低电平 11: 满足条件时, TIM4_<t>_OyH端口位输出反转 (y=U、V、W)	R/W
b7~b6	OPPKH[1:0]	上溢点端口状态H	条件：计数上溢 & OCCRxh计数匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyH端口位保持不变 01: 满足条件时, TIM4_<t>_OyH端口位输出高电平 10: 满足条件时, TIM4_<t>_OyH端口位输出低电平 11: 满足条件时, TIM4_<t>_OyH端口位输出反转 (y=U、V、W)	R/W
b5~b4	OPDCH[1:0]	向下计数端口状态H	条件：计数器向下计数 & OCCRxh计数匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyH端口位保持不变 01: 满足条件时, TIM4_<t>_OyH端口位输出高电平 10: 满足条件时, TIM4_<t>_OyH端口位输出低电平 11: 满足条件时, TIM4_<t>_OyH端口位输出反转 (y=U、V、W)	R/W
b3	OCFZRH	下溢点OCFH状态H	条件：计数下溢 & OCCRxh计数匹配 (x=u、v、w)	R/W

			0: 满足条件时, OCSR.OCFH位保持不变 1: 满足条件时, OCSR.OCFH位置位	
b2	OCFUCH	向上计数OCFH状态H	条件: 计数器向上计数 & OCCRxh计数匹配 ( $x=u, v, w$ ) 0: 满足条件时, OCSR.OCFH位保持不变 1: 满足条件时, OCSR.OCFH位置位	R/W
b1	OCFPKH	上溢点OCFH状态H	条件: 计数上溢 & OCCRxh计数匹配 ( $x=u, v, w$ ) 0: 满足条件时, OCSR.OCFH位保持不变 1: 满足条件时, OCSR.OCFH位置位	R/W
b0	OCFDCH	向下计数OCFH状态H	条件: 计数器向下计数 & OCCRxh计数匹配 ( $x=u, v, w$ ) 0: 满足条件时, OCSR.OCFH位保持不变 1: 满足条件时, OCSR.OCFH位置位	R/W

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EOPN ZRL[1:0]		EOPN PKL[1:0]		EOP ZRL[1:0]		EOP UCL[1:0]		EOP PKL[1:0]		EOP DCL[1:0]		EOPN UCL[1:0]		EOPN DCL[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OPN ZRL[1:0]		OPN PKL[1:0]		OP ZRL[1:0]		OP UCL[1:0]		OP PKL[1:0]		OP DCL[1:0]		OCF ZRL		OCF UCL	

注: 该寄存器位描述用于 OCMLu1、OCMRv1、OCMRw1 时

位	标记	位名	功能	读写
b31~b30	EOPNZRL[1:0]	扩张下溢点端口状态L	条件: 计数下溢 & OCCRxI计数不匹配 & OCCRxh计数匹配 ( $x=u, v, w$ ) 00: 满足条件时, TIM4_<t>_OyL端口位保持不变 01: 满足条件时, TIM4_<t>_OyL端口位输出高电平 10: 满足条件时, TIM4_<t>_OyL端口位输出低电平 11: 满足条件时, TIM4_<t>_OyL端口位输出反转 ( $y=U, V, W$ )	R/W
b29~b28	EOPNPKL[1:0]	扩张上溢点端口状态L	条件: 计数上溢 & OCCRxI计数不匹配 & OCCRxh计数匹配 ( $x=u, v, w$ ) 00: 满足条件时, TIM4_<t>_OyL端口位保持不变 01: 满足条件时, TIM4_<t>_OyL端口位输出高电平 10: 满足条件时, TIM4_<t>_OyL端口位输出低电平 11: 满足条件时, TIM4_<t>_OyL端口位输出反转 ( $y=U, V, W$ )	R/W
b27~b26	EOPZRL[1:0]	扩张下溢点端口状态L	条件: 计数下溢 & OCCRxI计数匹配 & OCCRxh计数匹配 ( $x=u, v, w$ ) 00: 满足条件时, TIM4_<t>_OyL端口位保持不变 01: 满足条件时, TIM4_<t>_OyL端口位输出高电平 10: 满足条件时, TIM4_<t>_OyL端口位输出低电平 11: 满足条件时, TIM4_<t>_OyL端口位输出反转 ( $y=U, V, W$ )	R/W
b25~b24	EOPUCL[1:0]	扩张向上计数端口状态L	条件: 计数器向上计数 & OCCRxI计数匹配 & OCCRxh计数匹配 ( $x=u, v, w$ ) 00: 满足条件时, TIM4_<t>_OyL端口位保持不变 01: 满足条件时, TIM4_<t>_OyL端口位输出高电平	R/W

			10: 满足条件时, TIM4_<t>_OyL端口位输出低电平 11: 满足条件时, TIM4_<t>_OyL端口位输出反转 (y=U、V、W)	
b23~b22	EOPPKL[1:0]	扩张上溢点端口状态L	条件: 计数上溢 & OCCRxI计数匹配 & OCCRxh计数匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyL端口位保持不变 01: 满足条件时, TIM4_<t>_OyL端口位输出高电平 10: 满足条件时, TIM4_<t>_OyL端口位输出低电平 11: 满足条件时, TIM4_<t>_OyL端口位输出反转 (y=U、V、W)	R/W
b21~b20	EOPDCL[1:0]	扩张向下计数端口状态L	条件: 计数器向下计数 & OCCRxI计数匹配 & OCCRxh计数匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyL端口位保持不变 01: 满足条件时, TIM4_<t>_OyL端口位输出高电平 10: 满足条件时, TIM4_<t>_OyL端口位输出低电平 11: 满足条件时, TIM4_<t>_OyL端口位输出反转 (y=U、V、W)	R/W
b19~b18	EOPNUCL[1:0]	扩张向上计数端口状态L	条件: 计数器向上计数 & OCCRxI计数不匹配 & OCCRxh计数匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyL端口位保持不变 01: 满足条件时, TIM4_<t>_OyL端口位输出高电平 10: 满足条件时, TIM4_<t>_OyL端口位输出低电平 11: 满足条件时, TIM4_<t>_OyL端口位输出反转 (y=U、V、W)	R/W
b17~b16	EOPNDCL[1:0]	扩张向下计数端口状态L	条件: 计数器向下计数 & OCCRxI计数不匹配 & OCCRxh计数匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyL端口位保持不变 01: 满足条件时, TIM4_<t>_OyL端口位输出高电平 10: 满足条件时, TIM4_<t>_OyL端口位输出低电平 11: 满足条件时, TIM4_<t>_OyL端口位输出反转 (y=U、V、W)	R/W
b15~b14	OPNZRL[1:0]	下溢点端口状态L	条件: 计数下溢 & OCCRxI计数不匹配 & OCCRxh计数不匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyL端口位保持不变 01: 满足条件时, TIM4_<t>_OyL端口位输出高电平 10: 满足条件时, TIM4_<t>_OyL端口位输出低电平 11: 满足条件时, TIM4_<t>_OyL端口位输出反转 (y=U、V、W)	R/W
b13~b12	OPNPKL[1:0]	上溢点端口状态L	条件: 计数上溢 & OCCRxI计数不匹配 & OCCRxh计数不匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyL端口位保持不变 01: 满足条件时, TIM4_<t>_OyL端口位输出高电平 10: 满足条件时, TIM4_<t>_OyL端口位输出低电平 11: 满足条件时, TIM4_<t>_OyL端口位输出反转 (y=U、V、W)	R/W
b11~b10	OPZRL[1:0]	下溢点端口状态L	条件: 计数下溢 & OCCRxI计数匹配 & OCCRxh计数不匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyL端口位保持不变	R/W

			01: 满足条件时, TIM4_<t>_OyL端口位输出高电平 10: 满足条件时, TIM4_<t>_OyL端口位输出低电平 11: 满足条件时, TIM4_<t>_OyL端口位输出反转 (y=U、V、W)	
b9~b8	OPUCL[1:0]	向上计数端口状态L	条件: 计数器向上计数 & OCCRxI计数匹配 & OCCRxh计数不匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyL端口位保持不变 01: 满足条件时, TIM4_<t>_OyL端口位输出高电平 10: 满足条件时, TIM4_<t>_OyL端口位输出低电平 11: 满足条件时, TIM4_<t>_OyL端口位输出反转 (y=U、V、W)	R/W
b7~b6	OPPKL[1:0]	上溢点端口状态L	条件: 计数上溢 & OCCRxI计数匹配 & OCCRxh计数不匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyL端口位保持不变 01: 满足条件时, TIM4_<t>_OyL端口位输出高电平 10: 满足条件时, TIM4_<t>_OyL端口位输出低电平 11: 满足条件时, TIM4_<t>_OyL端口位输出反转 (y=U、V、W)	R/W
b5~b4	OPDCL[1:0]	向下计数端口状态L	条件: 计数器向下计数 & OCCRxI计数匹配 & OCCRxh计数不匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyL端口位保持不变 01: 满足条件时, TIM4_<t>_OyL端口位输出高电平 10: 满足条件时, TIM4_<t>_OyL端口位输出低电平 11: 满足条件时, TIM4_<t>_OyL端口位输出反转 (y=U、V、W)	R/W
b3	OCFZRL	下溢点OCFL状态L	条件: 计数下溢 & OCCRxI计数匹配 (x=u、v、w) 0: 满足条件时, OCSR.OCFL位保持不变 1: 满足条件时, OCSR.OCFL位置位	R/W
b2	OCFUCL	向上计数OCFL状态L	条件: 计数器向上计数 & OCCRxI计数匹配 (x=u、v、w) 0: 满足条件时, OCSR.OCFL位保持不变 1: 满足条件时, OCSR.OCFL位置位	R/W
b1	OCFPKL	上溢点OCFL状态L	条件: 计数上溢 & OCCRxI计数匹配 (x=u、v、w) 0: 满足条件时, OCSR.OCFL位保持不变 1: 满足条件时, OCSR.OCFL位置位	R/W
b0	OCFDCL	向下计数OCFL状态L	条件: 计数器向下计数 & OCCRxI计数匹配 (x=u、v、w) 0: 满足条件时, OCSR.OCFL位保持不变 1: 满足条件时, OCSR.OCFL位置位	R/W

### 注意:

- 从本地址区域读取数据时, 读取的并非缓冲器寄存器的值, 而是 OCMR 寄存器的值。
- TIM4\_<t>\_OyL 可以由 OCCRxI 与计数器的计数值(独立操作模式), 或者 OCCRxh 与计数器的计数值且 OCCRxI 与计数器的计数值(链接操作模式)来决定。向寄存器 OCMLRI 的 bit[31:20] 和 bit[15:4]写入相同的 12 位值, 同时将 OCMLRI[19:16]写入“0000”, 此时, TIM4\_<t>\_OyL 输出将不受 OCCRxh 的影响, 而仅由 OCCRxI 决定。此模式叫做独立操作模式---通道 yH 由 OCCRxh 决定, 通道 yL 由 OCCRxI 决定配置。如果不满足上述条件的独立操作模式, 就是链接操作模式---通道 yL 输出同时受 OCCRxh 和 OCCRxI 影响。

( $x=u, v, w$ ,  $y=U, V, W$ )

- 下面 2 种情况满足时，也认为是符合 OCMRxh.OPPKH[7:6]设定的条件 ( $x=u, v, w$ ):
  - 1) CCSR.MODE=1 && 计数上溢 && OCCRxh=0xFFFF
  - 2) OCERx.MCECH=1 && 计数上溢 && OCCRxh $\geq$ CNTR
- 下面 2 种情况满足时，也认为是符合 OCMRxl.OPPKL[7:6]设定的条件 ( $x=u, v, w$ ):
  - 1) CCSR.MODE=1 && 计数上溢 && OCCRxl=0xFFFF
  - 2) OCERx.MCECL=1 && 计数上溢 && OCCRxl $\geq$ CNTR

### 21.5.9 专用比较基准寄存器 (TMR4\_SCCRm)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SCCR[15:0]															
位	标记	位名		功能											读写
b15~b0	SCCR[15:0]	专用比较基准值 (比较启动模式) 或延迟基准值 (延迟启动模式) 注: 从本地址区域读取数据时, 读取的并非缓冲器寄存器的值, 而是SCCR寄存器的值											R/W		

## 21.5.10 专用控制状态寄存器 (TMR4\_SCSRm)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ZEN	UEN	PEN	DEN	-	-	EVT DS	EVT MS	-	-	LMC	EVTOS[2:0]	BUFEN[1:0]			
<hr/>															
位	标记	位名	功能												读写
<hr/>															
b15	ZEN	下溢点EVT使能	0: 计数下溢时, EVT无操作 1: 计数下溢时: EVTMS=0&SCCR比较匹配&SCMR设定匹配时, EVT启动输出 EVTMS=1&OCCR比较匹配&SCMR设定匹配时, EVT延时模式启动												R/W
b14	UEN	向上计数EVT使能	0: 向上计数时, EVT无操作 1: 向上计数时: EVTMS=0&SCCR比较匹配&SCMR设定匹配时, EVT启动输出 EVTMS=1&OCCR比较匹配&SCMR设定匹配时, EVT延时模式启动												R/W
b13	PEN	上溢点EVT使能	0: 计数上溢时, EVT无操作 1: 计数上溢时: EVTMS=0&SCCR比较匹配&SCMR设定匹配时, EVT启动输出 EVTMS=1&OCCR比较匹配&SCMR设定匹配时, EVT延时模式启动												R/W
b12	DEN	向下计数EVT使能	0: 向下计数时, EVT无操作 1: 向下计数时: EVTMS=0&SCCR比较匹配&SCMR设定匹配时, EVT启动输出 EVTMS=1&OCCR比较匹配&SCMR设定匹配时, EVT延时模式启动												R/W
b11~b10	Reserved	-	读出时为“0”, 写入时写“0”												R/W
b9	EVTDS	EVT延时对象选择	0: 在延时启动模式时, OCCR <sub>xh</sub> 作为延时比较匹配对象 1: 在延时启动模式时, OCCR <sub>xl</sub> 作为延时比较匹配对象 (x=u、v、w) 注: 该位在EVTMS=0时无效												R/W
b8	EVTMS	EVT模式选择	0: 比较启动模式 (CNTR和SCCR的比较结果触发) 1: 延时启动模式 (比较匹配事件经SCCR延迟后触发)												R/W
b7~b6	Reserved	-	读出时为“0”, 写入时写“0”												R/W
b5	LMC	周期间隔响应功能链接	0: 周期间隔响应功能链接无效, SCCR的缓存传送由BUFEN设定决定 1: 周期间隔响应功能链接有效, SCCR的缓存传送在BUFEN设定的基础上, 还必须满足CVPR.PIC[3:0]=0000 (计数上溢时) 或 CVPR.ZIC[3:0]=0000 (计数下溢时)												R/W
b4~b2	EVTOS[2:0]	EVT输出选择	000: Special Event 0的EVT输出有效 001: Special Event 1的EVT输出有效 010: Special Event 2的EVT输出有效 011: Special Event 3的EVT输出有效 100: Special Event 4的EVT输出有效 101: Special Event 5的EVT输出有效 请不要设定其他值												R/W

---

b1~b0	BUFEN[1:0]	SCCR&SCMR缓存传送	00: SCCR、SCMR缓存寄存器的值直接写入SCCR、SCMR	
			01: SCCR、SCMR缓存寄存器的值在计数下溢时写入SCCR、SCMR	
			10: SCCR、SCMR缓存寄存器的值在计数上溢时写入SCCR、SCMR	R/W
			11: SCCR、SCMR缓存寄存器的值在计数下溢或上溢时写入SCCR、SCMR	

---

### 21.5.11 专用扩展控制寄存器 (TMR4\_SCER)

复位值: 0xFF00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Reserved										PCTS	EVTRS[2:0]					
位	标记	位名										功能				读写
b15~b4	Reserved	-										读出时为“0”,写入时写“0”				R/W
b3	PCTS	计数方向 输出使能										计数器工作在三角波模式时, 控制计数方向输出到端口 0: 禁止计数方向信号输出到TIM4_<t>_PCT端口 1: 允许计数方向信号输出到TIM4_<t>_PCT端口				
b2~b0	EVTRS[2:0]	专用事件输出选择										000: 禁止专用事件输出到端口 001: Special Event 0的EVT输出至TIM4_<t>_ADSM端口 010: Special Event1的EVT输出至TIM4_<t>_ADSM端口 011: Special Event 2的EVT输出至TIM4_<t>_ADSM端口 100: Special Event 3的EVT输出至TIM4_<t>_ADSM端口 101: Special Event 4的EVT输出至TIM4_<t>_ADSM端口 110: Special Event 5的EVT输出至TIM4_<t>_ADSM端口 111: 设定禁止				R/W

### 21.5.12 专用模式控制寄存器 (TMR4\_SCMRm)

复位值: 0xFF00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0			
Reserved										MPCE	MZCE	-	-	AMC[3:0]				
位	标记	位名										功能				读写		
b15~b8	Reserved	-										读出时为“1”,写入时写“1”				R/W		
b7	MPCE	周期间隔响应使能										0: 禁止AMC与CVPR.PIC比较 1: 使能AMC与CVPR.PIC比较				R/W		
b6	MZCE	周期间隔响应使能										0: 禁止AMC与CVPR.ZIC比较 1: 使能AMC与CVPR.ZIC比较				R/W		
b5~b4	Reserved	-										读出时为“0”,写入时写“0”				R/W		
b3~b0	AMC[3:0]	专用事件输出周期 间隔值										该位设定专用事件输出功能时的周期间隔值, 在AMC和CVPR.PIC或 CVPR.ZIC相等时, 专用事件输出功能有效				R/W		

#### 注意:

- 从本地址区域读取数据时, 读取的并非缓冲器寄存器的值, 而是 SCMR 寄存器的值。

**21.5.13 PWM 基本控制寄存器 (TMR4\_POCRn)**

复位值: 0xFF00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0							
Reserved								LVLS[1:0]		PWMMMD[1:0]		-	DIVCK[2:0]									
<hr/>																						
位	标记	位名								功能												
b15~b8	Reserved	-								读出时为“1”,写入时写“1”												
b7~b6	LVLS[1:0]	PWM输出极性控制								00: TIM4_<t>_OxH和TIM4_<t>_OxL的输出都不反转 01: TIM4_<t>_OxH和TIM4_<t>_OxL的输出都反转 10: TIM4_<t>_OxH的输出反转, TIM4_<t>_OxL的输出不反转 11: TIM4_<t>_OxH的输出不反转, TIM4_<t>_OxL的输出反转								R/W				
b5~b4	PWMMMD[1:0]	PWM输出模式								00: 直通模式 01: 死区定时器模式 10: 死区定时器滤波模式 11: 设定禁止						R/W						
b3	Reserved	-								读出时为“0”,写入时写“0”						R/W						
b2~b0	DIVCK[2:0]	计数时钟分频								该位指示滤波计数器和死区计数器的计数时钟分频 000: 计数时钟为PCLK0 001: 计数时钟为PCLK0/2 010: 计数时钟为PCLK0/4 011: 计数时钟为PCLK0/8 100: 计数时钟为PCLK0/16 101: 计数时钟为PCLK0/32 110: 计数时钟为PCLK0/64 111: 计数时钟为PCLK0/128								R/W				

## 21.5.14 PWM 状态控制寄存器 (TMR4\_PSCR)

复位值: 0x05550000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16				
Reserved		OSWL[1:0]		OSWH[1:0]		OSVL[1:0]		OSVH[1:0]		OSUL[1:0]		OSUH[1:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0				
Reserved				AOE	MOE	ODT[1:0]		OEWL	OEWH	OEVL	OEVH	OEUL	OEUH						
位	标记	位名	功能	读写															
b31~b28	Reserved	-	读出时为“0”,写入时写“0”	R/W															
b27~b26	OSWL[1:0]	TIM4_<t>_OWL 输出禁止状态	条件: 1. 发生EMB事件; 2. 软件写MOE=0; 3. MOE=1&OEWL=0 00: 满足以上条件之一时, TIM4_<t>_OWL端口正常输出 01: 满足以上条件之一时, TIM4_<t>_OWL端口输出为Hi-z 10: 满足以上条件之一时, TIM4_<t>_OWL端口输出固定为低电平 11: 满足以上条件之一时, TIM4_<t>_OWL端口输出固定为高电平 注: 只能在Timer4停止 (CCSR.STOP=1) 时修改该位	R/W															
b25~b24	OSWH[1:0]	TIM4_<t>_OWH 输出禁止状态	条件: 1. 发生EMB事件; 2. 软件写MOE=0; 3. MOE=1&OEWH=0 00: 满足以上条件之一时, TIM4_<t>_OWH端口正常输出 01: 满足以上条件之一时, TIM4_<t>_OWH端口输出为Hi-z 10: 满足以上条件之一时, TIM4_<t>_OWH端口输出固定为低电平 11: 满足以上条件之一时, TIM4_<t>_OWH端口输出固定为高电平 注: 只能在Timer4停止 (CCSR.STOP=1) 时修改该位	R/W															
b23~b22	OSVL[1:0]	TIM4_<t>_OVL 输出禁止状态	条件: 1. 发生EMB事件; 2. 软件写MOE=0; 3. MOE=1&OEVL=0 00: 满足以上条件之一时, TIM4_<t>_OVL端口正常输出 01: 满足以上条件之一时, TIM4_<t>_OVL端口输出为Hi-z 10: 满足以上条件之一时, TIM4_<t>_OVL端口输出固定为低电平 11: 满足以上条件之一时, TIM4_<t>_OVL端口输出固定为高电平 注: 只能在Timer4停止 (CCSR.STOP=1) 时修改该位	R/W															
b21~b20	OSVH[1:0]	TIM4_<t>_OVH 输出禁止状态	条件: 1. 发生EMB事件; 2. 软件写MOE=0; 3. MOE=1&OEVH=0 00: 满足以上条件之一时, TIM4_<t>_OVH端口正常输出 01: 满足以上条件之一时, TIM4_<t>_OVH端口输出为Hi-z 10: 满足以上条件之一时, TIM4_<t>_OVH端口输出固定为低电平 11: 满足以上条件之一时, TIM4_<t>_OVH端口输出固定为高电平 注: 只能在Timer4停止 (CCSR.STOP=1) 时修改该位	R/W															
b19~b18	OSUL[1:0]	TIM4_<t>_OUL 输出禁止状态	条件: 1. 发生EMB事件; 2. 软件写MOE=0; 3. MOE=1&OEUL=0 00: 满足以上条件之一时, TIM4_<t>_OUL端口正常输出 01: 满足以上条件之一时, TIM4_<t>_OUL端口输出为Hi-z 10: 满足以上条件之一时, TIM4_<t>_OUL端口输出固定为低电平 11: 满足以上条件之一时, TIM4_<t>_OUL端口输出固定为高电平 注: 只能在Timer4停止 (CCSR.STOP=1) 时修改该位	R/W															
b17~b16	OSUH[1:0]	TIM4_<t>_OUH 输出禁止状态	条件: 1. 发生EMB事件; 2. 软件写MOE=0; 3. MOE=1&OEUH=0 00: 满足以上条件之一时, TIM4_<t>_OUH端口正常输出	R/W															

---

01: 满足以上条件之一时, TIM4_<t>_OUH端口输出为Hi-z 10: 满足以上条件之一时, TIM4_<t>_OUH端口输出固定为低电平 11: 满足以上条件之一时, TIM4_<t>_OUH端口输出固定为高电平 注: 只能在Timer4停止 (CCSR.STOP=1) 时修改该位			
b15~b10	Reserved	-	读出时为“0”,写入时写“0” R/W
b9	AOE	自动输出使能	0: EMB事件清除后MOE位只能软件置1以恢复PWM正常输出 1: EMB事件清除后MOE位在下一个计数下溢点由硬件自动置1, PWM输出恢复正常 注: 置位EMB_STATCLR或清零EMB_SOE可以清除EMB事件
b8	MOE	主输出使能	0: TIM4_<t>_Oxy端口输出由OSxy位指定的状态 (x=U、V、W, y=H、L) 1: OExy位设定为使能时TIM4_<t>_Oxy端口输出使能 (x=U、V、W, y=H、L) 注: EMB事件发生后此位由硬件立即清零。EMB事件清除后, 依据AOE位的设置此位可以软件置1或硬件自动置1。
b7~b6	ODT	端口使能位生效时间	条件: 端口使能位OExy的值改变后 (x=U、V、W, y=H、L) 0X: 满足条件时, 立即生效 R/W 10: 满足条件时, 计数器下次下溢时生效 11: 满足条件时, 计数器下次上溢时生效
b5	OEWL	TIM4_<t>_OWL输出使能	0: TIM4_<t>_OWL端口输出状态由OSWL位设定 1: MOE=1时, TIM4_<t>_OWL端口输出正常PWM波形 R/W
b4	OEWH	TIM4_<t>_OWH输出使能	0: TIM4_<t>_OWH端口输出状态由OSWH位设定 1: MOE=1时, TIM4_<t>_OWH端口输出正常PWM波形 R/W
b3	OEVL	TIM4_<t>_OVL输出使能	0: TIM4_<t>_OVL端口输出状态由OSVL位设定 1: MOE=1时, TIM4_<t>_OVL端口输出正常PWM波形 R/W
b2	OEVH	TIM4_<t>_OVH输出使能	0: TIM4_<t>_OVH端口输出状态由OSVH位设定 1: MOE=1时, TIM4_<t>_OVH端口输出正常PWM波形 R/W
b1	OEUL	TIM4_<t>_OUL输出使能	0: TIM4_<t>_OUL端口输出状态由OSUL位设定 1: MOE=1时, TIM4_<t>_OUL端口输出正常PWM波形 R/W
b0	OEUH	TIM4_<t>_OUH输出使能	0: TIM4_<t>_OUH端口输出状态由OSUH位设定 1: MOE=1时, TIM4_<t>_OUH端口输出正常PWM波形 R/W

下表为 PWM 端口输出状态与寄存器设定值的关系说明。

表 21-4 PWM 端口输出状态与寄存器设定值

MOE位	OExy位	OSxy位	TIM4_<t>_Oxy输出状态
1	0	00	正常输出 (Timer4驱动)
		01	Hi-z
		10	低电平
		11	高电平
	1	X	正常输出 (Timer4驱动)
0	X	00	正常输出 (Timer4驱动)
		01	Hi-z
		10	低电平
		11	高电平

### 21.5.15 PWM 滤波控制寄存器 (TMR4\_PFSRn)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
PFSR[15:0]																
位	标记	位名										功能				读写
b15~b0	PFSR[15:0]	滤波初始值										注: 当PWM波形输出模式不选死区定时器滤波模式时, 16位滤波计数器用作的16位重载计数器, 此时16位滤波计数器可以周期性产生中断输出, 此功能与PWM波形生成器功能无关。				R/W

### 21.5.16 PWM 死区控制寄存器 (TMR4\_PDA<B>Rn)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
PDA/BR[15:0]																
位	标记	位名										功能				读写
b15~b0	PDA/BR[15:0]	死区初始值										死区计数初始值				R/W

### 21.5.17 重载控制状态寄存器 (TMR4\_RCSR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RTS W	RTE W	RTIC W	RTIF W	RTS V	RTE V	RTIC V	RTIF V	RTS U	RTE U	RTIC U	RTIF U	-	RTID W	RTID V	RTID U
<hr/>															
位	标记	位名	功能												
<hr/>															
b15	RTSW	重载计数器停止W	0: 无操作 1: 停止重载计数器W并清除RTIFW 注: 该位在读出时始终为0												
b14	RTEW	重载计数器启动W	0: 写0无效, 读出为0表示重载计数器W已停止 1: 写1启动重载计数器W, 读出为1表示重载计数器W已启动												
b13	RTICW	清零计数匹配状态W	0: 无操作 1: 清零RTIFW标志位 注: 该位在读出时始终为0												
b12	RTIFW	计数匹配状态W	0: 重载计数器计数值与PFSRw发生未比较匹配 1: 重载计数器计数值与PFSRw发生比较匹配												
b11	RTSV	重载计数器停止V	0: 无操作 1: 停止重载计数器V并清除RTIFV 注: 该位在读出时始终为0												
b10	RTEV	重载计数器启动V	0: 写0无效, 读出为0表示重载计数器V已停止 1: 写1启动重载计数器V, 读出为1表示重载计数器V已启动												
b9	RTICV	清零计数匹配状态V	0: 无操作 1: 清零RTIFV标志位 注: 该位在读出时始终为0												
b8	RTIFV	计数匹配状态V	0: 重载计数器计数值与PFSRv发生未比较匹配 1: 重载计数器计数值与PFSRv发生比较匹配												
b7	RTSU	重载计数器停止U	0: 无操作 1: 停止重载计数器U并清除RTIFU 注: 该位在读出时始终为0												
b6	RTEU	重载计数器启动U	0: 写0无效, 读出为0表示重载计数器U已停止 1: 写1启动重载计数器U, 读出为1表示重载计数器U已启动												
b5	RTICU	清零计数匹配状态U	0: 无操作 1: 清零RTIFU标志位 注: 该位在读出时始终为0												
b4	RTIFU	计数匹配状态U	0: 重载计数器计数值与PFSRU发生未比较匹配 1: 重载计数器计数值与PFSRU发生比较匹配												
b3	Reserved	-	读出时为“0”, 写入时写“0”												
b2	RTIDW	重载中断屏蔽W	0: 重载功能有效时, 重载中断W输出有效 1: 重载功能有效时, 重载中断W输出无效												
b1	RTIDV	重载中断屏蔽V	0: 重载功能有效时, 重载中断V输出有效 1: 重载功能有效时, 重载中断V输出无效												
b0	RTIDU	重载中断屏蔽U	0: 重载功能有效时, 重载中断U输出有效 1: 重载功能有效时, 重载中断U输出无效												

**21.5.18 专用中断控制寄存器 (TMR4\_SCIR)**

复位值: 0xFF00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved										ITEN5	ITEN4	ITEN3	ITEN2	ITEN1	ITENO
<hr/>															
位	标记	位名	功能	读写											
b15~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b5	ITEN5	专用匹配wl中断使能	0: 专用计数比较匹配5中断不使能 1: 专用计数比较匹配5中断使能	R/W											
b4	ITEN4	专用匹配wh中断使能	0: 专用计数比较匹配4中断不使能 1: 专用计数比较匹配4中断使能	R/W											
b3	ITEN3	专用匹配vl中断使能	0: 专用计数比较匹配3中断不使能 1: 专用计数比较匹配3中断使能	R/W											
b2	ITEN2	专用匹配vh中断使能	0: 专用计数比较匹配2中断不使能 1: 专用计数比较匹配2中断使能	R/W											
b1	ITEN1	专用匹配ul中断使能	0: 专用计数比较匹配1中断不使能 1: 专用计数比较匹配1中断使能	R/W											
b0	ITENO	专用匹配uh中断使能	0: 专用计数比较匹配0中断不使能 1: 专用计数比较匹配0中断使能	R/W											

**21.5.19 专用标志控制寄存器 (TMR4\_SCFR)**

复位值: 0xFF00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved										SF5	SF4	SF3	SF2	SF1	SF0
<hr/>															
位	标记	位名	功能	读写											
b15~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b5	SF5	专用计数匹配标志位wl	0: 未发生专用计数比较匹配5事件 1: 发生专用计数比较匹配5事件	R/W											
b4	SF4	专用计数匹配标志位wh	0: 未发生专用计数比较匹配4事件 1: 发生专用计数比较匹配4事件	R/W											
b3	SF3	专用计数匹配标志位vl	0: 未发生专用计数比较匹配3事件 1: 发生专用计数比较匹配3事件	R/W											
b2	SF2	专用计数匹配标志位vh	0: 未发生专用计数比较匹配2事件 1: 发生专用计数比较匹配2事件	R/W											
b1	SF1	专用计数匹配标志位ul	0: 未发生专用计数比较匹配1事件 1: 发生专用计数比较匹配1事件	R/W											
b0	SF0	专用计数匹配标志位uh	0: 未发生专用计数比较匹配0事件 1: 发生专用计数比较匹配0事件	R/W											

## 22 紧急刹车模块 (EMB)

### 22.1 简介

紧急刹车模块是在满足一定条件时产生控制事件输出给定时器，以控制定时器停止向外部电机输出 PWM 信号的功能模块，下列要因用于产生控制事件：

- 外部端口输入电平变化
- PWM 输出端口电平发生同相（同高或同低）
- 电压比较器比较结果
- 系统错误发生
- 写寄存器软件控制

EMB 结构框图如图 22-1 所示。

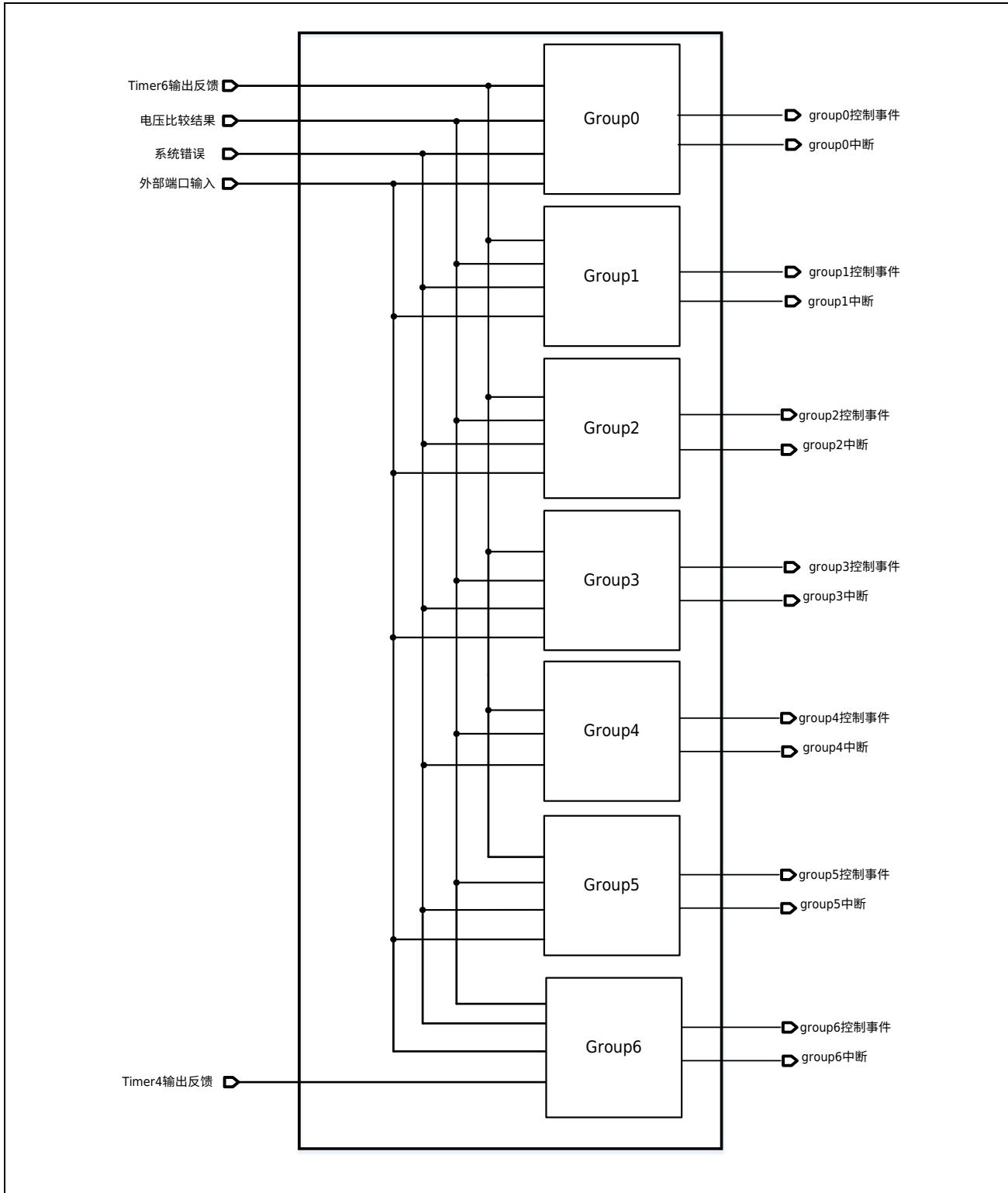


图 22-1 EMB 结构框图

## 22.2 功能描述

### 22.2.1 概述

EMB 用于向具有 PWM 功能的定时器模块(Timer4,Timer6)在满足一定条件时输出一个控制事件信号，通知定时器模块关闭当前 PWM 输出。EMB 模块具有 7 个 group，其中 group0~group5 用于控制 Timer6，由 Timer6 寄存器设定选择使用，group0~group5 用于控制 Timer6，group6 用于控制 Timer4。

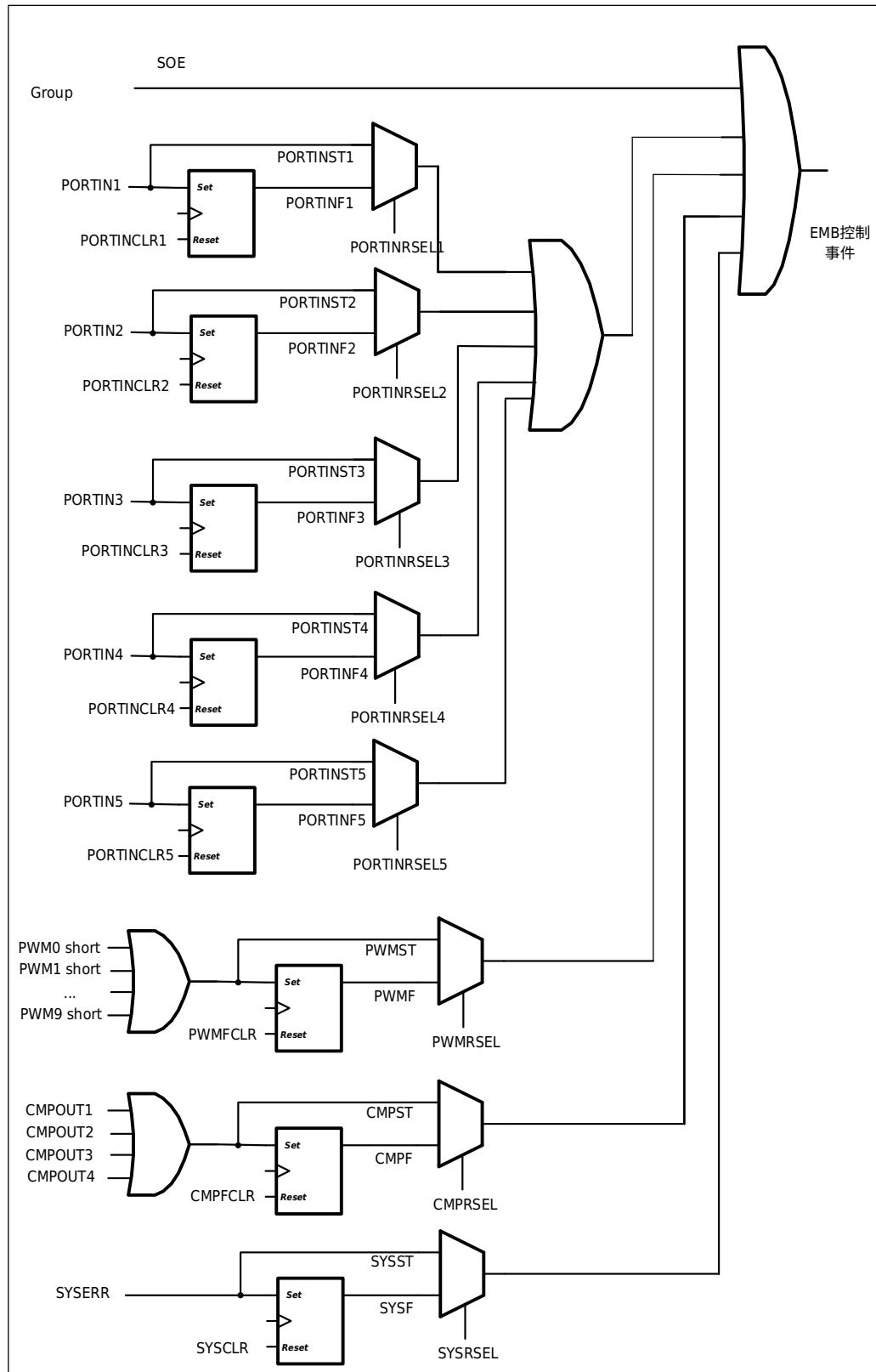


图 22-2 EMB 通道功能框图

## 22.2.2 外部端口输入电平变化时控制 PWM 信号输出

EMB 共有 5 个外部端口用于实现当输入电平变化时控制 PWM 信号输出。每个 group 能够独立设置 5 个外部端口中的 1 个或多个有效，端口分配如下表所示。

表 22-1 端口分配表

端口	PIN 名
PORT1	PA6、PA15、PA12、PE15、PB12、PA14、PB8
PORT2	PA15、PA11、PA6、PC3
PORT3	PA0、PA9、PE7、PC5
PORT4	PB11、PB5、PE8、PA1、PF2
PORT5	PC7、PA10、PD2、PB4

使用外部端口输入电平变化时控制 PWM 信号输出时，首先将使能位，即 EMB 控制寄存器 EMB\_CTL1\_x.PORTINENy ( $x=0\sim6, y=1\sim5$ ) 置为有效，同时通过 EMB\_CTL1\_x.INVSEL y ( $x=0\sim6, y=1\sim5$ ) 设置在端口电平为高 (INVSEL=0) 时 或 端口电平为低 (INVSEL=1) 时产生控制信号。根据需要通过 EMB\_CTL2\_x.NFENy ( $x=0\sim6, y=1\sim5$ ) 和 EMB\_CTL2\_x.NFSELy ( $x=0\sim6, y=1\sim5$ ) 使能滤波功能和设置滤波时钟。

滤波器根据滤波时钟对输入信号进行采样后，采用三次比较一致的方式进行滤波，即当滤波时钟采样到端口上 3 次一致的电平时，该电平被当作有效电平传送到模块内部；小于 3 次一致的电平会被当作外部干扰滤掉，不传送到模块内部。

当端口上产生符合条件的有效电平时，EMB 状态寄存器的端口输入状态 EMB\_STATx.PORTINSTy ( $x=0\sim6, y=1\sim5$ ) 被置位，同时端口输入控制标志位 EMB\_STATx.PORTINFy ( $x=0\sim6, y=1\sim5$ ) 被置位。当中断许可寄存器 EMB\_INTENx.PORTINEN ( $x=0\sim6$ ) = 1 时将产生中断。

EMB 的控制事件通过 EMB 释放方式选择寄存器 EMB\_RLSSELx.PORTINRSELy ( $x=0\sim6, y=1\sim5$ ) 选择由 EMB\_STATx.PORTINSTy ( $x=0\sim6, y=1\sim5$ ) 或者 EMB\_STATx.PORTINFy ( $x=0\sim6, y=1\sim5$ ) 向 Timer6 和 Timer4 输出。Timer6 和 Timer4 在接到控制事件后，可根据寄存器设定将输出端口置为高电平，低电平或高阻态。

释放控制时，如果 EMB\_RLSSELx.PORTINRSELy ( $x=0\sim6, y=1\sim5$ ) = 1，当端口输入电平转为无效时，EMB\_STATx.PORTINSTy ( $x=0\sim6, y=1\sim5$ ) 将自动清零，同时控制事件将立即释放。如果 EMB\_RLSSELx.PORTINRSELy ( $x=0\sim6, y=1\sim5$ ) = 0，当端口输入电平转为无效后，需要写 EMB 状态复位寄存器 EMB\_STATCLRx .PORTINFCLR ( $x=0\sim6$ ) 使 EMB\_STATx.PORTINF ( $x=0\sim6$ ) 清 0，将控制事件释放。

### 22.2.3 PWM 输出端口电平发生同相（同高或同低）时停止 PWM 信号输出

使用时首先将使能位，即 EMB 控制寄存器 EMB\_CTL1\_x.PWMSEN y(x=0~5,y=0~9 或 x=4~5,y=0~2) 置为有效，同时通过 EMB 控制寄存器 EMB\_CTL2\_x.PWMLV y(x=0~5,y=0~9 或 x=4~5,y=0~2) 选择需要监控的有效电平。

完成设置后，EMB 监控 Timer6 和 Timer4 的互补 PWM 输出信号，当输出信号出现同高或同低情况时，EMB 状态寄存器的 PWM 输出状态 EMB\_STATx.PWMST(x=0~6) 被置位，同时 PWM 输出同相位控制标志位 EMB\_STATx.PWMF(x=0~6) 被置位。当中断许可寄存器 EMB\_INTENx.PWMEN (x=0~6)=1 时将产生中断。

EMB 的控制事件通过 EMB 释放方式选择寄存器 EMB\_RLSSELx.PWMRSEL(x=0~6) 选择由 EMB\_STATx.PWMST (x=0~6) 或者 EMB\_STATx.PWMF (x=0~6) 向 Timer6 和 Timer4 输出。Timer6 和 Timer4 在接到控制事件后可根据寄存器设定将输出端口置为高电平，低电平或高阻态。

释放控制时，如果 EMB\_RLSSELx.PWMRSEL(x=0~6)=1，当 PWM 输出同高或同低状态解除，EMB\_STATx.PWMST(x=0~6) 将自动清零，同时控制事件将立即释放。如果 EMB\_RLSSELx.PWMST(x=0~6)=0，当 PWM 输出同高或同低状态解除后，需要写 EMB 状态复位寄存器 EMB\_STATCLRx.PWMFCLR (x=0~6) 使 EMB\_STATx.PWMF (x=0~6) 清 0 后，将控制事件释放。

group0-5 可用于监控 Timer6 的互补 PWM 输出信号，group6 用于监控 Timer4\_1 的互补 PWM 输出信号。

表 22-2 EMB group 对照表

端口名	功能	对应 group	EMB_CTL1 控制位	EMB_CTL2 控制位
TIM6_m_PWMA(m=1~10)	Timer6的互补PWM输出信号	group0	PWMSEN[9:0]	PWMLV[9:0]
TIM6_m_PWMB(m=1~10)		group1		
TIM4_1_OUH	Timer4_1的互补PWM输出信号	group6	PWMSEN[2]	PWMLV[2]
TIM4_1_OUL			PWMSEN[1]	PWMLV[1]
TIM4_1_OVH			PWMSEN[0]	PWMLV[0]
TIM4_1_OVL				
TIM4_1_OWH				
TIM4_1_OWL				

### 22.2.4 根据电压比较器比较结果停止 PWM 信号输出

EMB 的每个 group 能够根据电压比较器的 4 组比较结果向 Timer6 和 Timer4 发送控制事件信号。电压比较器输出结果的设定请参考【电压比较器 (CMP)】章节。

使用时首先将使能位，即 EMB 控制寄存器 EMB\_CTL1\_x. CMPENy(x=0~6)(y=1~4)置为有效。当电压比较器比较结果标志位被置起时，EMB 状态寄存器的电压比较器状态 EMB\_STATx.CMPST(x=0~6)被置位，同时 EMB 电压比较器控制标志位 EMB\_STATx.CMPF(x=0~6)被置位。当中断许可寄存器 EMB\_INTENx. CMPINTEN(x=0~6)=1 时将产生中断。

EMB 的控制事件通过 EMB 释放方式选择寄存器 EMB\_RLSSELx.CMPRSEL(y=0~6)选择由 EMB\_STATx.CMPST(x=0~6)或者 EMB\_STATx.CMPF (x=0~6)向 Timer6 和 Timer4 输出。Timer6 和 Timer4 在接到控制事件信号后可根据寄存器设定将输出端口置为高电平，低电平或高阻态。

释放控制时，如果 EMB\_RLSSELx. CMPRSEL(x=0~6)=1，当电压比较器结果转为无效时，EMB\_STATx.CMPST(x=0~6) 将自动清零，同时控制事件将立即释放。如果 EMB\_RLSSELx.CMPRSEL(y=0~6)=0，当电压比较器结果转为无效后，需要写 EMB 状态复位寄存器 EMB\_STATCLRx .CMPFCLR(x=0~6)使 EMB\_STATx.CMPF (x=0~6)清 0 后，将控制事件释放。

### 22.2.5 系统错误发生时停止 PWM 信号输出

EMB 的每个 group 能够在系统错误发生时向 Timer6 和 Timer4 发送通知信号。系统错误信号来自外部振荡器停止、SRAM 奇偶校验错误以及 ECC 校验错误、Cortex-M4 锁定信号和 PVD 检测。

使用时首先将使能位，即 EMB 控制寄存器 EMB\_CTL1\_x. SYSEN (x=0~6)置为有效,设置 EMB\_CTL1\_x. PVDEN、LOCKUPEN、SRAMPYERREN、SRAMERREN、OSCSTPEN 位，使能需要检测系统错误源。当选择的系统错误源标志被置起时，EMB 状态寄存器的系统错误状态 EMB\_STATx.SYSST(x=0~6)被置位，同时 EMB 系统错误控制标志位 EMB\_STATx.SYSF(x=0~6)被置位。当中断许可寄存器 EMB\_INTENx. SYSINTEN(x=0~6)=1 时还将产生中断。

EMB 的控制事件通过 EMB 释放方式选择寄存器 EMB\_RLSSELx.SYSRSEL(y=0~6)选择由 EMB\_STATx.SYSST(x=0~6)或者 EMB\_STATx.SYSF (x=0~6)向 Timer6 和 Timer4 输出。Timer6 和 Timer4 在接到控制事件信号后可根据寄存器设定将输出端口置为高电平，低电平或高阻态。

释放控制时，如果 EMB\_RLSSELx. SYSRSEL(x=0~6)=1，当系统错误标志转为无效时 EMB\_STATx.SYSST(x=0~6) 将自动清零，同时控制事件将立即释放。如果 EMB\_RLSSELx.SYSRSEL(y=0~6)=0，当系统错误标志转为无效后，需要写 EMB 状态复位寄存器 EMB\_STATCLRx .SYSFCLR(x=0~6)使 EMB\_STATx.SYSF (x=0~6)清 0 后，将控制事件释放。

### 22.2.6 写寄存器软件控制 PWM 信号输出

EMB 的软件输出使能控制寄存器(EMB\_SOEx)(x=0~6)能够允许用户通过软件直接置位和复位的方式向 Timer6 和 Timer4 发送控制信号，软件控制 PWM 输出时不会产生中断请求。

## 22.3 寄存器说明

表 22-3 寄存器一览

名称	英文缩写	说明	偏移地址
EMB控制寄存器1	EMB_CTL1	各个PWM输出控制事件的使能	0x0
EMB控制寄存器2	EMB_CTL2	端口输入控制事件的滤波以及选择PWM反馈信号的有效电平	0x4
EMB软件输出使能控制寄存器	EMB_SOE	软件产生PWM输出控制事件	0x8
EMB状态寄存器	EMB_STAT	表示PWM输出控制的状态	0xC
EMB状态复位寄存器	EMB_STATCLR	清除PWM输出控制的状态	0x10
EMB中断许可寄存器	EMB_INTEN	中断使能	0x14
EMB控制PWM输出释放方式选择寄存器	EMB_RSEL	选择各个PWM输出控制事件的释放方式	0x18

### 22.3.1 EMB 控制寄存器 1\_0~5(EMB\_CTL1\_0~5)

该寄存器为单次写入寄存器，即复位之后仅能够写入一次

地址：0x40017C00, 0x40017C20, 0x40017C40, 0x40017C60, 0x40017C680, 0x40017C6A0

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PVD EN	LOC KUPE N	SRA MPY ERR EN	SRA MER REN	OSC STPE N	INVSE L5	INVSE L4	INVSE L3	INVSE L2	INVSE L1	Rese rved	PORT INEN 5	PORT INEN 4	PORT INEN 3	PORT INEN 2	PORT INEN 1
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Rese rved	PWM SEN 9	PWM SEN 8	PWM SEN 7	PWM SEN 6	PWM SEN5	PWM SEN4	PWM SEN3	PWM SEN2	PWM SEN1	PWM SEN0	SYSE N	CMP EN4	CMP EN3	CMP EN2	CMP EN1

位	标记	位名	功能	读写
b31	PVDEN	PVD检测控制使能	0: PVD检测结果输出控制无效 1: PVD检测结果输出控制有效	R/W
b30	LOCKUPEN	Cortex-M4 LOCKUP控制使能	0: LOCKUP发生时输出控制无效 1: LOCKUP发生时输出控制有效	R/W
b29	SRAMPYERREN	SRAM 奇偶校验错误控制使能	0: SRAM 奇偶校验错误发生时输出控制无效 1: SRAM 奇偶校验错误发生时输出控制有效	R/W
b28	SRAMERREN	SRAM ECC校验错误控制使能	0: SRAM ECC校验错误发生时输出控制无效 1: SRAM ECC校验错误发生时输出控制有效	R/W
b27	OSCSTPEN	振荡器停止控制使能	0: 振荡器停止发生时输出控制无效 1: 振荡器停止发生时输出控制有效	R/W
b26	INVSEL5	端口5输入有效电平选择	0: 高电平有效 1: 低电平有效	R/W
b25	INVSEL4	端口4输入有效电平选择	0: 高电平有效 1: 低电平有效	R/W
b24	INVSEL3	端口3输入有效电平选择	0: 高电平有效 1: 低电平有效	R/W
b23	INVSEL2	端口2输入有效电平选择	0: 高电平有效 1: 低电平有效	R/W
b22	INVSEL1	端口1输入有效电平选择	0: 高电平有效 1: 低电平有效	R/W
b21	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b20	PORTINEN5	端口5输入控制使能	0: 端口输入控制无效 1: 端口输入控制有效	R/W
b19	PORTINEN4	端口4输入控制使能	0: 端口输入控制无效 1: 端口输入控制有效	R/W
b18	PORTINEN3	端口3输入控制使能	0: 端口输入控制无效 1: 端口输入控制有效	R/W
b17	PORTINEN2	端口2输入控制使能	0: 端口输入控制无效 1: 端口输入控制有效	R/W
b16	PORTINEN1	端口1输入控制使能	0: 端口输入控制无效 1: 端口输入控制有效	R/W

b15	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b14~b5	PWMSEN[9:0]	TIM6_m_PWMA/B(m=1~10) 短路输出控制使能	对应10组PWM反馈输入 0: 短路时输出控制无效 1: 短路时输出控制有效	R/W
b4	SYSEN	系统错误控制使能	0: 系统错误发生时输出控制无效 1: 系统错误发生时输出控制有效	R/W
b3	CMPEN4	CMP4电压比较器比较结果控制 使能	0: 电压比较器比较结果输出控制无效 1: 电压比较器比较结果输出控制有效	R/W
b2	CMPEN3	CMP3电压比较器比较结果控制 使能	0: 电压比较器比较结果输出控制无效 1: 电压比较器比较结果输出控制有效	R/W
b1	CMPEN2	CMP2电压比较器比较结果控制 使能	0: 电压比较器比较结果输出控制无效 1: 电压比较器比较结果输出控制有效	R/W
b0	CMPEN1	CMP1电压比较器比较结果控制 使能	0: 电压比较器比较结果输出控制无效 1: 电压比较器比较结果输出控制有效	R/W

### 22.3.2 EMB 控制寄存器 1\_6(EMB\_CTL1\_6)

该寄存器为单次写入寄存器，即复位之后仅能够写入一次

地址：0x40017CC0

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PVDE N	LOCK UPEN	SRAM PYER REN	SRAM ERRE N	OSCS TPEN	INVSE L5	INVSE L4	INVSE L3	INVSE L2	INVSE L1	Reser ved	PORTI NEN5	PORTI NEN4	PORTI NEN3	PORTI NEN2	PORTI NEN1
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								PWM SEN2	PWM SEN1	PWM SEN0	SYSE N	CMPE N4	CMPE N3	CMPE N2	CMPE N1

位	标记	位名	功能	读写
b31	PVDEN	PVD检测控制使能	0: PVD检测结果输出控制无效 1: PVD检测结果输出控制有效	R/W
b30	LOCKUPEN	Cortex-M4 LOCKUP控制使能	0: LOCKUP发生时输出控制无效 1: LOCKUP发生时输出控制有效	R/W
b29	SRAMPYERREN	SRAM 奇偶校验错误控制使能	0: SRAM 奇偶校验错误发生时输出控制无效 1: SRAM 奇偶校验错误发生时输出控制有效	R/W
b28	SRAMERREN	SRAM ECC校验错误控制使能	0: SRAM ECC校验错误发生时输出控制无效 1: SRAM ECC校验错误发生时输出控制有效	R/W
b27	OSCSTPEN	振荡器停止控制使能	0: 振荡器停止发生时输出控制无效 1: 振荡器停止发生时输出控制有效	R/W
b26	INVSEL5	端口5输入有效电平选择	0: 高电平有效 1: 低电平有效	R/W
b25	INVSEL4	端口4输入有效电平选择	0: 高电平有效 1: 低电平有效	R/W
b24	INVSEL3	端口3输入有效电平选择	0: 高电平有效 1: 低电平有效	R/W
b23	INVSEL2	端口2输入有效电平选择	0: 高电平有效 1: 低电平有效	R/W
b22	INVSEL1	端口1输入有效电平选择	0: 高电平有效 1: 低电平有效	R/W
b21	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b20	PORTINEN5	端口5输入控制使能	0: 端口输入控制无效 1: 端口输入控制有效	R/W
b19	PORTINEN4	端口4输入控制使能	0: 端口输入控制无效 1: 端口输入控制有效	R/W
b18	PORTINEN3	端口3输入控制使能	0: 端口输入控制无效 1: 端口输入控制有效	R/W
b17	PORTINEN2	端口2输入控制使能	0: 端口输入控制无效 1: 端口输入控制有效	R/W
b16	PORTINEN1	端口1输入控制使能	0: 端口输入控制无效 1: 端口输入控制有效	R/W
b15~b8	Reserved	-	读时读出“0”，写入时请写“0”	R/W

b7	PWMSEN2	TIM4_m_OUH/L 短路输出 控制使能 (m=1)	0: 短路时输出控制无效 1: 短路时输出控制有效	R/W
b6	PWMSEN1	TIM4_m_OVH/L 短路输出 控制使能 (m=1)	0: 短路时输出控制无效 1: 短路时输出控制有效	R/W
b5	PWMSEN0	TIM4_m_OWH/L 短路输出 控制使能 (m=1)	0: 短路时输出控制无效 1: 短路时输出控制有效	R/W
b4	SYSEN	系统错误控制使能	0: 系统错误发生时输出控制无效 1: 系统错误发生时输出控制有效	R/W
b3	CMPEN4	CMP4电压比较器比较结果控制使能	0: 电压比较器比较结果输出控制无效 1: 电压比较器比较结果输出控制有效	R/W
b2	CMPEN3	CMP3电压比较器比较结果控制使能	0: 电压比较器比较结果输出控制无效 1: 电压比较器比较结果输出控制有效	R/W
b1	CMPEN2	CMP2电压比较器比较结果控制使能	0: 电压比较器比较结果输出控制无效 1: 电压比较器比较结果输出控制有效	R/W
b0	CMPEN1	CMP1电压比较器比较结果控制使能	0: 电压比较器比较结果输出控制无效 1: 电压比较器比较结果输出控制有效	R/W

### 22.3.3 EMB 控制寄存器 2\_0~5(EMB\_CTL2\_0~5)

该寄存器为单次写入寄存器，即复位之后仅能够写入一次

地址：0x40017C04, 0x40017C24, 0x40017C44, 0x40017C64, 0x40017C84, 0x40017CA4

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res erve d	NFEN5	NFSEL5[1:0]	NFEN4	NFSEL4[1:0]	NFEN3	NFSEL3[1:0]	NFEN2	NFSEL2[1:0]	NFEN1	NFSEL1[1:0]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					PWML V9	PWML V8	PWML V7	PWML V6	PWML V5	PWML V4	PWML V3	PWML V2	PWML V1	PWML V0	

位	标记	位名	功能	读写
b31	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b30	NFEN5	端口5输入数字滤波器使能	0: 滤波器无效 1: 滤波器有效	R/W
b29~b28	NFSEL5[1:0]	端口5数字滤波器滤波时钟选择	00: 使用总线时钟滤波 01: 使用总线时钟的8分频滤波 10: 使用总线时钟的32分频滤波 11: 使用总线时钟的128分频滤波	R/W
b27	NFEN4	端口4输入数字滤波器使能	0: 滤波器无效 1: 滤波器有效	R/W
b26~b25	NFSEL4[1:0]	端口4数字滤波器滤波时钟选择	00: 使用总线时钟滤波 01: 使用总线时钟的8分频滤波 10: 使用总线时钟的32分频滤波 11: 使用总线时钟的128分频滤波	R/W
b24	NFEN3	端口3输入数字滤波器使能	0: 滤波器无效 1: 滤波器有效	R/W
b23~b22	NFSEL3[1:0]	端口3数字滤波器滤波时钟选择	00: 使用总线时钟滤波 01: 使用总线时钟的8分频滤波 10: 使用总线时钟的32分频滤波 11: 使用总线时钟的128分频滤波	R/W
b21	NFEN2	端口2输入数字滤波器使能	0: 滤波器无效 1: 滤波器有效	R/W
b20~b19	NFSEL2[1:0]	端口2数字滤波器滤波时钟选择	00: 使用总线时钟滤波 01: 使用总线时钟的8分频滤波 10: 使用总线时钟的32分频滤波 11: 使用总线时钟的128分频滤波	R/W
b18	NFEN1	端口1输入数字滤波器使能	0: 滤波器无效 1: 滤波器有效	R/W
b17~b16	NFSEL1[1:0]	端口1数字滤波器滤波时钟选择	00: 使用总线时钟滤波 01: 使用总线时钟的8分频滤波 10: 使用总线时钟的32分频滤波 11: 使用总线时钟的128分频滤波	R/W
b15~b10	Reserved	-	读时读出“0”，写入时请写“0”	R/W

---

b9~b0	PWMLV[9:0]	TIM6_m_PWMA/B(m=1~10) 输出有效电平选择	对应10组PWM反馈输入 0：低电平为有效电平 1：高电平为有效电平	R/W
-------	------------	-----------------------------------	------------------------------------------	-----

---

### 22.3.4 EMB 控制寄存器 2\_6(EMB\_CTL2\_6)

该寄存器为单次写入寄存器，即复位之后仅能够写入一次

地址：0x40017CC4

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res erve d	NFEN 5	NFSEL5[1:0]	NFEN4	NFSEL4[1:0]	NFEN3	NFSEL3[1:0]	NFEN 2	NFSEL2[1:0]	NFEN 1	NFSEL1[1:0]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												PWML V2	PWML V1	PWML V0	

位	标记	位名	功能	读写
b31	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b30	NFEN5	端口5输入数字滤波器使能	0: 滤波器无效 1: 滤波器有效	R/W
b29~b28	NFSEL5[1:0]	端口5数字滤波器滤波时钟选择	00: 使用总线时钟滤波 01: 使用总线时钟的8分频滤波 10: 使用总线时钟的32分频滤波 11: 使用总线时钟的128分频滤波	R/W
b27	NFEN4	端口4输入数字滤波器使能	0: 滤波器无效 1: 滤波器有效	R/W
b26~b25	NFSEL4[1:0]	端口4数字滤波器滤波时钟选择	00: 使用总线时钟滤波 01: 使用总线时钟的8分频滤波 10: 使用总线时钟的32分频滤波 11: 使用总线时钟的128分频滤波	R/W
b24	NFEN3	端口3输入数字滤波器使能	0: 滤波器无效 1: 滤波器有效	R/W
b23~b22	NFSEL3[1:0]	端口3数字滤波器滤波时钟选择	00: 使用总线时钟滤波 01: 使用总线时钟的8分频滤波 10: 使用总线时钟的32分频滤波 11: 使用总线时钟的128分频滤波	R/W
b21	NFEN2	端口2输入数字滤波器使能	0: 滤波器无效 1: 滤波器有效	R/W
b20~b19	NFSEL2[1:0]	端口2数字滤波器滤波时钟选择	00: 使用总线时钟滤波 01: 使用总线时钟的8分频滤波 10: 使用总线时钟的32分频滤波 11: 使用总线时钟的128分频滤波	R/W
b18	NFEN1	端口1输入数字滤波器使能	0: 滤波器无效 1: 滤波器有效	R/W
b17~b16	NFSEL1[1:0]	端口1数字滤波器滤波时钟选择	00: 使用总线时钟滤波 01: 使用总线时钟的8分频滤波 10: 使用总线时钟的32分频滤波 11: 使用总线时钟的128分频滤波	R/W
b15~b3	Reserved	-	读时读出“0”，写入时请写“0”	R/W

b2	PWMLV2	TIM4_m_OUH/L输出有效电平选择 (m=1)	0: 低电平为有效电平 1: 高电平为有效电平	R/W
b1	PWMLV1	TIM4_m_OVH/L输出有效电平选择 (m=1)	0: 低电平为有效电平 1: 高电平为有效电平	R/W
b0	PWMLV0	TIM4_m_OWH/L输出有效电平选择 (m=1)	0: 低电平为有效电平 1: 高电平为有效电平	R/W

### 22.3.5 EMB 软件输出使能控制寄存器(EMB\_SOEx)(x=0~6)

地址： 0x40017C08, 0x40017C28, 0x40017C48, 0x40017C68, 0x40017C88, 0x40017CA8,  
0x40017CC8

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
SOE															
位	标记	位名	功能	读写											
b31~1	Reserved	-	读时读出“0”，写入时请写“0”	R/W											
b0	SOE	软件控制输出	0: PWM正常输出 1: PWM停止输出	R/W											

### 22.3.6 EMB 状态寄存器(EMB\_STATx) (x=0~6)

地址：0x40017C0C, 0x40017C2C, 0x40017C4C, 0x40017C6C, 0x40017C8C, 0x40017CAC,  
0x40017CCC

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Reserved														PORTI NST5	PORTI NST4	PORTI NST3
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
PORTI NST2	PORTI NST1	Reser ved	PORTI NF5	PORTI NF4	PORTI NF3	PORTI NF2	PORTI NF1	SYSST	CMPST	PWMS T	Reser ved	SYSF	CMPF	PWMS F	Reser ved	

位	标记	位名	功能	读写
b31~19	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b18	PORTINST5	端口5输入控制状态	0：端口5输入控制处于无效电平状态 1：端口5输入控制处于有效电平状态	R
b17	PORTINST4	端口4输入控制状态	0：端口4输入控制处于无效电平状态 1：端口4输入控制处于有效电平状态	R
b16	PORTINST3	端口3输入控制状态	0：端口3输入控制处于无效电平状态 1：端口3输入控制处于有效电平状态	R
b15	PORTINST2	端口2输入控制状态	0：端口2输入控制处于无效电平状态 1：端口2输入控制处于有效电平状态	R
b14	PORTINST1	端口1输入控制状态	0：端口1输入控制处于无效电平状态 1：端口1输入控制处于有效电平状态	R
b13~12	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b12	PORTINF5	端口5输入控制标志位	EMB_STATx. PORTINST5为1时置位 EMB_STATCLRx. PORTINFCLR5写1时清0	R
b11	PORTINF4	端口4输入控制标志位	EMB_STATx. PORTINST4为1时置位 EMB_STATCLRx. PORTINFCLR4写1时清0	R
b10	PORTINF3	端口3输入控制标志位	EMB_STATx. PORTINST3为1时置位 EMB_STATCLRx. PORTINFCLR3写1时清0	R
b9	PORTINF2	端口2输入控制标志位	EMB_STATx. PORTINST2为1时置位 EMB_STATCLRx. PORTINFCLR2写1时清0	R
b8	PORTINF1	端口1输入控制标志位	EMB_STATx. PORTINST1为1时置位 EMB_STATCLRx. PORTINFCLR1写1时清0	R
b7	SYSST	系统错误状态	0：系统错误未发生 1：系统错误发生	R
b6	CMPST	电压比较器状态	电压比较器比较结果	R
b5	PWMST	PWM输出状态	0：没有发生PWM输出同相 1：发生PWM输出同相	R
b4	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b3	SYSF	系统错误标志位	EMB_STATx. SYSST为1时置位 EMB_STATx. SYSFCLR写1时清0	R
b2	CMPF	电压比较器控制标志位	EMB_STATx. CMPST为1时置位 EMB_STATCLRx. CMPFCLR写1时清0	R
b1	PWMSF	PWM输出同相位控制	EMB_STATx. PWMST为1时置位	R

标志位		EMB_STATCLR <sub>x</sub> . PWM_SCLR写1时清0	
b0	Reserved	-	读时读出“0”，写入时请写“0” R/W

### 22.3.7 EMB 状态复位寄存器(EMB\_STATCLR $x$ ) ( $x=0\sim6$ )

地址：0x40017C10, 0x40017C30, 0x40017C50, 0x40017C70, 0x40017C90, 0x40017CB0, 0x40017CD0

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved	PORTIN FCLR5	PORTIN FCLR4	PORTIN FCLR3	PORTIN FCLR2	PORTIN FCLR1		Reserved		SYSFC LR	CMPF CLR	PWMS FCLR	Reser ved			
<hr/>															
位	标记	位名	功能	读写											
b31~b13	Reserved	-	读时读出“0”，写入时请写“0”	W											
b12	PORTINFCLR5	复位 EMB_STAT.PORTINF5	0: 无任何效果 1: 当EMB_STAT.PORTINST5=0时，将EMB_STAT.PORTINF5清0 读该寄存器位无任何效果	W											
b11	PORTINFCLR4	复位 EMB_STAT.PORTINF4	0: 无任何效果 1: 当EMB_STAT.PORTINST4=0时，将EMB_STAT.PORTINF4清0 读该寄存器位无任何效果	W											
b10	PORTINFCLR3	复位 EMB_STAT.PORTINF3	0: 无任何效果 1: 当EMB_STAT.PORTINST3=0时，将EMB_STAT.PORTINF3清0 读该寄存器位无任何效果	W											
b9	PORTINFCLR2	复位 EMB_STAT.PORTINF2	0: 无任何效果 1: 当EMB_STAT.PORTINST2=0时，将EMB_STAT.PORTINF2清0 读该寄存器位无任何效果	W											
b8	PORTINFCLR1	复位 EMB_STAT.PORTINF1	0: 无任何效果 1: 当EMB_STAT.PORTINST1=0时，将EMB_STAT.PORTINF1清0 读该寄存器位无任何效果	W											
b7~b4	Reserved	-	读时读出“0”，写入时请写“0”	W											
b3	SYSFCLR	复位 EMB_STAT.SYSF	0: 无任何效果 1: 当EMB_STAT.SYSST=0时，将EMB_STAT.SYSF清0 读该寄存器位无任何效果	W											
b2	CMPFCLR	复位 EMB_STAT.CMPF	0: 无任何效果 1: 当EMB_STAT.CMPST=0时，将EMB_STAT.CMPF清0 读该寄存器位无任何效果	W											
b1	PWMSCLR	复位 EMB_STAT.PWMSF	0: 无任何效果 1: 当EMB_STAT.PWMST=0时，将EMB_STAT.PWMSF清0 读该寄存器位无任何效果	W											
b0	Reserved	-	读时读出“0”，写入时请写“0”	W											

### 22.3.8 EMB 中断许可寄存器(EMB\_INTENx)(x=0~6)

地址： 0x40017C14, 0x40017C34, 0x40017C54, 0x40017C74, 0x40017C94, 0x40017CB4,  
0x40017CD4

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved	PORTIN TEN5	PORTIN TEN4	PORTIN TEN3	PORTIN TEN2	PORTIN TEN1		Reserved		SYSIN TEN	CMPIN TEN	PWMS INTEN		Reser ved		

位	标记	位名	功能	读写
b31~b13	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b12	PORTINTEN5	端口5输入控制中断使能	0: 端口5输入控制不产生中断 1: 端口5输入控制产生中断	R/W
b11	PORTINTEN4	端口4输入控制中断使能	0: 端口4输入控制不产生中断 1: 端口4输入控制产生中断	R/W
b10	PORTINTEN3	端口3输入控制中断使能	0: 端口3输入控制不产生中断 1: 端口3输入控制产生中断	R/W
b9	PORTINTEN2	端口2输入控制中断使能	0: 端口2输入控制不产生中断 1: 端口2输入控制产生中断	R/W
b8	PORTINTEN1	端口1输入控制中断使能	0: 端口1输入控制不产生中断 1: 端口1输入控制产生中断	R/W
b7~b4	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b3	SYSINTEN	系统错误控制中断使能	0: 系统错误控制不产生中断 1: 系统错误控制产生中断	R/W
b2	CMPINTEN	电压比较器比较结果控制中断使能	0: 电压比较器比较结果控制不产生中断 1: 电压比较器比较结果控制产生中断	R/W
b1	PWMSINTEN	PWM输出同相位控制中断使能	0: PWM输出同相位控制不产生中断 1: PWM输出同相位控制产生中断	R/W
b0	Reserved	-	读时读出“0”，写入时请写“0”	R/W

### 22.3.9 EMB 控制 PWM 输出释放方式选择寄存器(EMB\_RLSSELx)(x=0~6)

地址： 0x40017C18, 0x40017C38, 0x40017C58, 0x40017C78, 0x40017C98, 0x40017CB8,  
0x40017CD8

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved	PORTINRSEL5	PORTINRSEL4	PORTINRSEL3	PORTINRSEL2	PORTINRSEL1	Reserved				SYSRSEL	CMPRSEL	PWMRSEL	Reserved		

位	标记	位名	功能	读写
b31~b12	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b11	PORTINRSEL5	端口输入5控制释放方式选择	0: EMB_STATx. PORTINF5=0时释放PWM输出控制 1: EMB_STATx. PORTINST5=0时释放PWM输出控制	R/W
b11	PORTINRSEL4	端口输入4控制释放方式选择	0: EMB_STATx. PORTINF4=0时释放PWM输出控制 1: EMB_STATx. PORTINST4=0时释放PWM输出控制	R/W
b10	PORTINRSEL3	端口输入3控制释放方式选择	0: EMB_STATx. PORTINF3=0时释放PWM输出控制 1: EMB_STATx. PORTINST3=0时释放PWM输出控制	R/W
b9	PORTINRSEL2	端口输入2控制释放方式选择	0: EMB_STATx. PORTINF2=0时释放PWM输出控制 1: EMB_STATx. PORTINST2=0时释放PWM输出控制	R/W
b8	PORTINRSEL1	端口输入1控制释放方式选择	0: EMB_STATx. PORTINF1=0时释放PWM输出控制 1: EMB_STATx. PORTINST1=0时释放PWM输出控制	R/W
b7~b4	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b3	SYSRSEL	系统错误控制释放方式选择	0: EMB_STATx. SYSF=0时释放PWM输出控制 1: EMB_STATx. SYSYST=0时释放PWM输出控制	R/W
b2	CMPRSEL	比较器比较结果控制释放方式选择	0: EMB_STATx. CMPF=0时释放PWM输出控制 1: EMB_STATx. CMPST=0时释放PWM输出控制	R/W
b1	PWMRSEL	PWM输出控制释放方式选择	0: EMB_STATx. PWMSF=0时释放PWM输出控制 1: EMB_STATx. PWMST=0时释放PWM输出控制	R/W
b0	Reserved	-	读时读出“0”，写入时请写“0”	R/W

## 23 通用定时器 (TimerA)

### 23.1 简介

通用定时器 A (TimerA) 是一个具有 16/32 位计数宽度、4 路 PWM 输出的定时器。该定时器支持三角波和锯齿波两种波形模式，可生成各种 PWM 波形（单边对齐 PWM、双边对称 PWM）；支持计数器同步启动；比较基准值寄存器支持缓存功能；支持单元间级联计数；支持 2 相正交编码计数和 3 相正交编码计数。本系列产品搭载 6 个单元 TimerA (U1~2 为 32bit 定时器；U3~U6 为 16bit 定时器)，最大可实现 24 路 PWM 输出。

### 23.2 基本框图

TimerA 基本的功能及特性如表 23-1 所示。

表 23-1 TimerA 的基本功能及特性

波形模式	锯齿波、三角波
基本功能	• 递加、递减计数方向
	• 同步启动计数器
	• 基准值缓存功能
	• 32位级联计数
	• 正交编码计数
	• 4路PWM输出
	• 比较匹配事件输出
中断类型	• 比较匹配中断
	• 周期匹配中断

TimerA 的基本框图如图 23-1 所示。图中“<t>”为单元编号，即“<t>”为 1~6，本章节后文提到“<t>”时均指单元编号，不再赘述。

本系列产品搭载的 6 个单元的 TimerA，其中单元 1~单元 4 的总线时钟和计数时钟均为 PCLK0 (此时，框图中所示 PCLK 是指 PCLK0)；单元 5~单元 6 的总线时钟和计数时钟为 PCLK1 (此时，框图中所示 PCLK 是指 PCLK1)。各单元的计数分频时钟源及端口数字滤波的采样基准时钟源与其总线时钟和计数时钟相同。

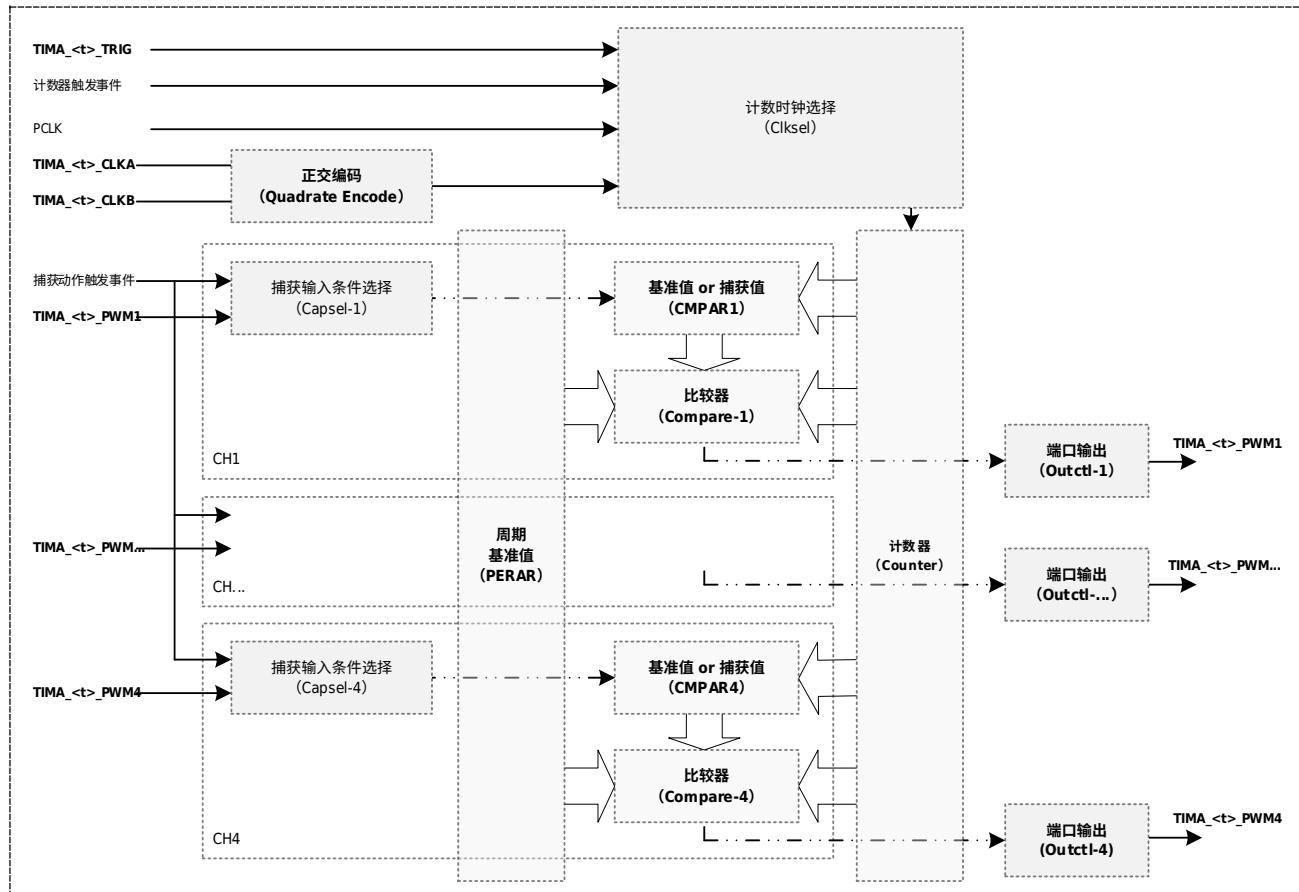


图 23-1 TimerA 基本框图

表 23-2 所示为 TimerA 的输入输出端口列表。

表 23-2 TimerA 端口列表

端口名	方向	功能
TIMA_<t>_PWMM <sub>m</sub>	in or out	捕获输入事件端口或 PWM 输出端口 ( $m=1\sim 4$ )
TIMA_<t>_CLKA	in	正交编码计数事件输入端口
TIMA_<t>_CLKB		
TIMA_<t>_TRIG	in	硬件触发启动、停止、清零事件输入端口

## 23.3 功能说明

### 23.3.1 波形模式

TimerA 有 2 种基本计数波形模式，锯齿波模式和三角波模式。两种波形模式的基本波形如图 23-2、图 23-3 所示。

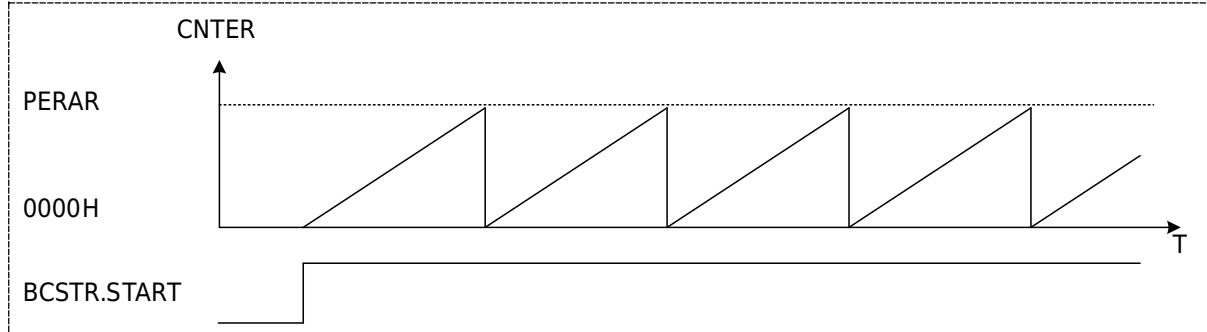


图 23-2 锯齿波波形 (递加计数)

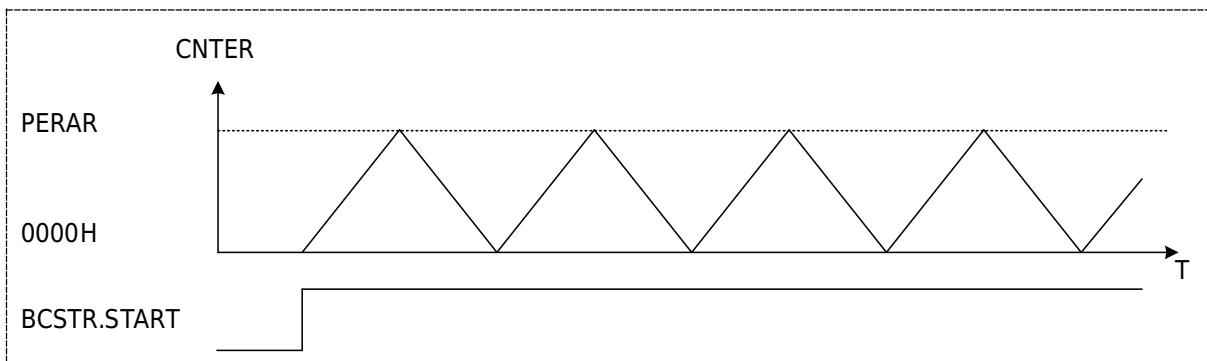


图 23-3 三角波波形

### 23.3.2 时钟源选择

TimerA 的计数时钟可以有以下几种选择：

- PCLK 的 1、2、4、8、16、32、64、128、256、512、1024 分频 (BCCTRL.CKDIV[3:0]设定)
- TIMA\_<t>\_TRIG 端口事件输入 (HCUPR[9:8]或 HCDOR[9:8]设定)
- 内部计数器触发事件输入 (HCUPR[10]或 HCDOR[10]设定)
- 对称单元的计数上溢或计数下溢事件输入 (HCUPR[12:11]或 HCDOR[12:11]设定)
- TIMA\_<t>\_CLKA、TIMA\_<t>\_CLKB 的端口正交编码输入 (HCUPR[7:0]或 HCDOR[7:0]设定)

计数时钟源选择 a 时为软件计数模式，计数时钟源选择 b、c、d、e 时为硬件计数模式。计数时钟选择 d 时多用于三相正交编码计数的公转计数模式（参见[位置溢出计数]和[混合计数]章节），也可用于级联计数。上述描述可以看到，b、c、d、e 时钟互相独立，可分别设定有效或无效，并且当选择 b、c、d、e 时钟时，a 时钟自动无效。

### 23.3.3 比较输出

各个 TimerA 单元内部均含有 4 个通道的比较输出 (TIMA\_<t>\_PWMn)，可在计数值与比较基准值比较匹配时输出指定的电平。TMRA\_CMPARn 寄存器分别对应了 TIMA\_<t>\_PWMn 输出端口的计数比较基准值。当定时器的计数值和 TMRA\_CMPARn 相等时，TIMA\_<t>\_PWMn 端口输出指定的电平(n=1~4)。

TIMA\_<t>\_PWMn 端口的计数开始时的电平、计数停止时的电平、计数比较匹配时的电平、计数周期匹配时的电平等，可通过端口控制寄存器 (PCONRn) 的 STAC、STPC、CMPC、PERC、FORC 位设定控制 (n=1~4)。图 23-4 为单元 1 的比较输出动作例。

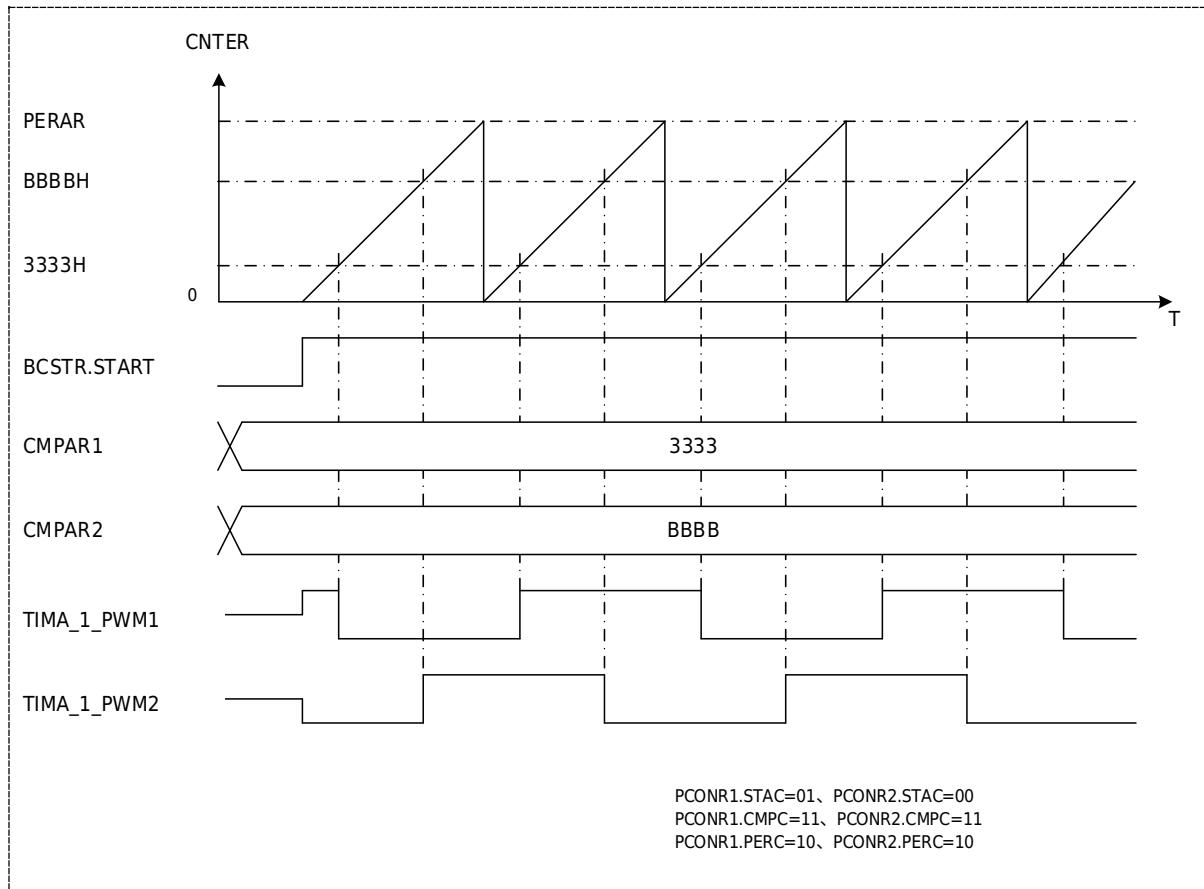


图 23-4 比较输出动作

### 23.3.4 捕获输入

各个 TimerA 单元的每个 PWM 输出通道都具有捕获输入功能，用于保存捕获到的计数值。设定捕获控制寄存器 (CCONRn) 的 CCONR.CAPMD 位为 1，捕获输入功能变为有效。此时选择对应的捕获输入条件且该条件有效时，当前的计数值就被保存到相应的寄存器 (CMPARn) 中 (n=1~4)。

捕获输入条件可以选择内部捕获动作触发事件（通过 HTSSR0~3 寄存器选择，具体参考寄存器说明章节）、TIMA\_<t>\_PWMn 端口输入等，具体的条件选择可通过捕获控制寄存器 (CCONRn) 的 HICP 位来设定 (n=1~4)。图 23-5 为捕获输入动作例。

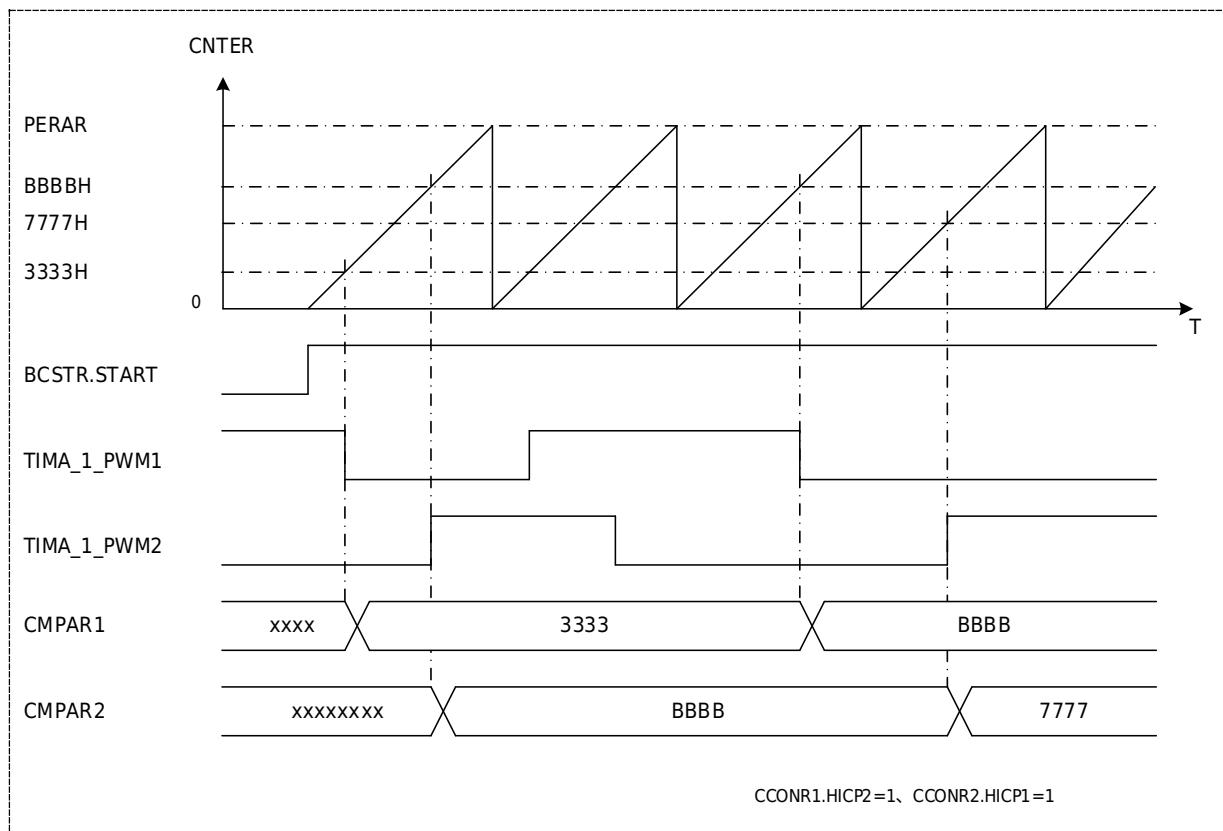


图 23-5 捕获输入动作

### 23.3.5 同步启动

本产品所搭载 6 个单元的 TimerA，可以实现软件同步启动或硬件同步启动。当本单元为单元 m 时，单元 m 可以选择与单元 n 同步启动（当  $m=2、4、6$  时， $n=1、3、5$ ）。

当单元 m 中的 BCCTRL.SYNST 位设定为 1 时，单元 m 与单元 n 的同步启动功能有效。此时，若软件设定单元 n 的 BCCTRL.START 位为 1，被同步单元（单元 m）的计数器开始软件同步计数；若硬件设定单元 n 的 HCONR.HSTA1~0 中任意位为 1，且单元 n 的对应硬件事件发生时，被同步单元（单元 m）的计数器开始硬件同步计数。在选择硬件同步计数启动功能时，被同步单元（单元 m）的 HCONR.HSTA1~0 的对应位也必须设定为有效（当  $m=2、4、6$  时， $n=1、3、5$ ）。

图 23-6 为设定单元 4 的 BCCTRL.SYNST=1 时的软件同步启动例。

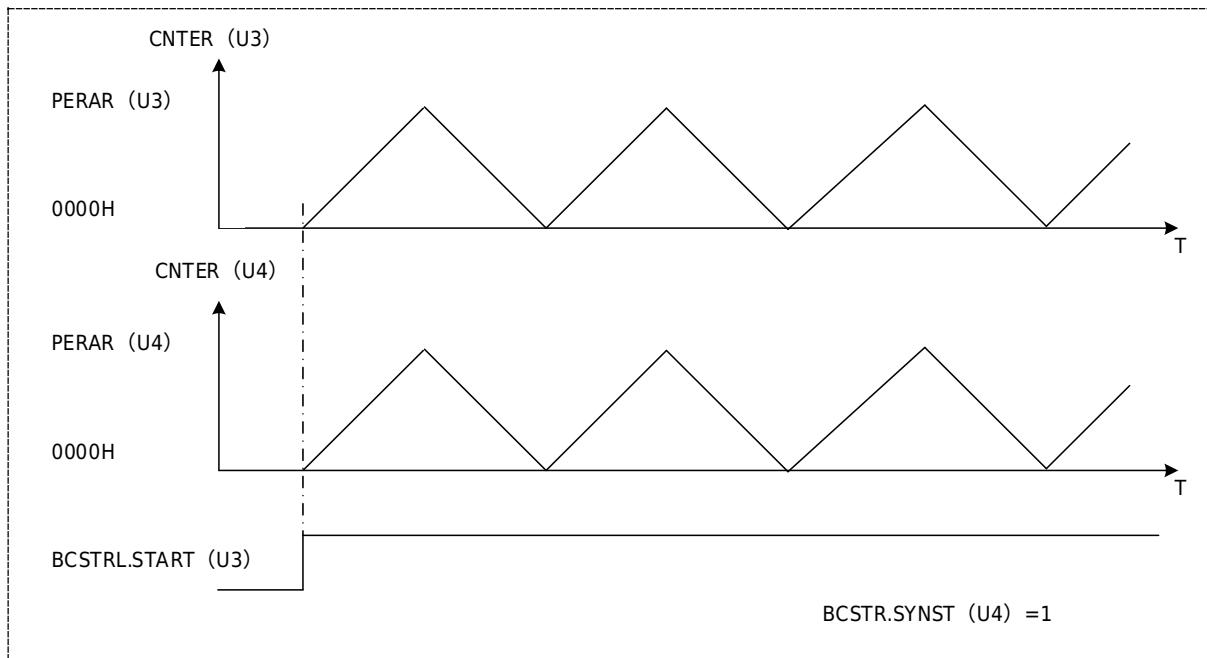


图 23-6 软件同步动作

### 23.3.6 数字滤波

各个单元的 TIMA\_<t>\_CLKA、TIMA\_<t>\_CLKB、TIMA\_<t>\_TRIG、TIMA\_<t>\_PWMr（捕获输入功能时）端口输入都有数字滤波功能。各端口的滤波功能的使能和滤波时钟的选择可通过设定滤波控制寄存器（FCONR）和捕获控制寄存器（CCONRn）的对应位来实现（n=1~4）。

在滤波采样基准时钟采样到端口上 3 次一致的电平时，该电平被当作有效电平传送到模块内部；小于 3 次一致的电平会被当作外部干扰滤掉，不传送到模块内部。图 23-7 所示为 TIMA\_1\_CLKA 端口滤波动作例。

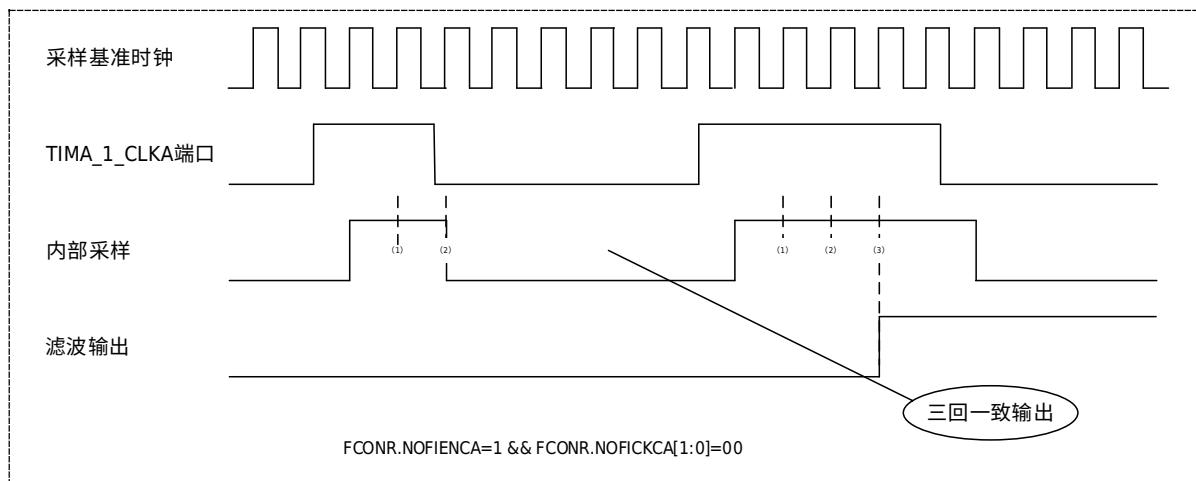


图 23-7 时钟输入端口的滤波功能

### 23.3.7 缓存功能

TimerA 的共计 4 个比较基准寄存器（CMPARn）可以成对实现缓存功能（n=1~4）。即 CMPAR2 作为 CMPAR1 的缓存基准值、CMPAR4 作为 CMPAR3 的缓存基准值。缓存控制寄存器（BCONRm）分别实现对两组缓存功能的控制（m=1~2）。

当缓存控制寄存器（BCONRm）的 BEN 位被置位时，缓存功能变为有效（m=1~2）。计数器计数到特定时间点时就发生一次缓存传送（CMPAR4/2->CMPAR3/1）。该“特定时间点”有以下几种情况：

- 硬件计数模式时，计数到上溢点（BCSTRL.DIR=1 时）或下溢点（BCSTRL.DIR=0 时）
- 锯齿波计数模式（BCSTRL.MODE=0）时，计数器计数到上溢点（BCSTRL.DIR=1 时）或下溢点（BCSTRL.DIR=0 时）
- 三角波计数模式（BCSTRL.MODE=1）时，计数到峰点（BCSTRL.DIR=1 && BCONRn.BSE0=1 时）  
(n=1~2)
- 三角波计数模式（BCSTRL.MODE=1）时，计数到谷点（BCSTRL.DIR=0 && BCONRn.BSE1=1 时）  
(n=1~2)
- 硬件计数模式或锯齿波计数模式时，发生清零动作

下图 23-8 所示，是锯齿波模式时的缓存传送示意图。

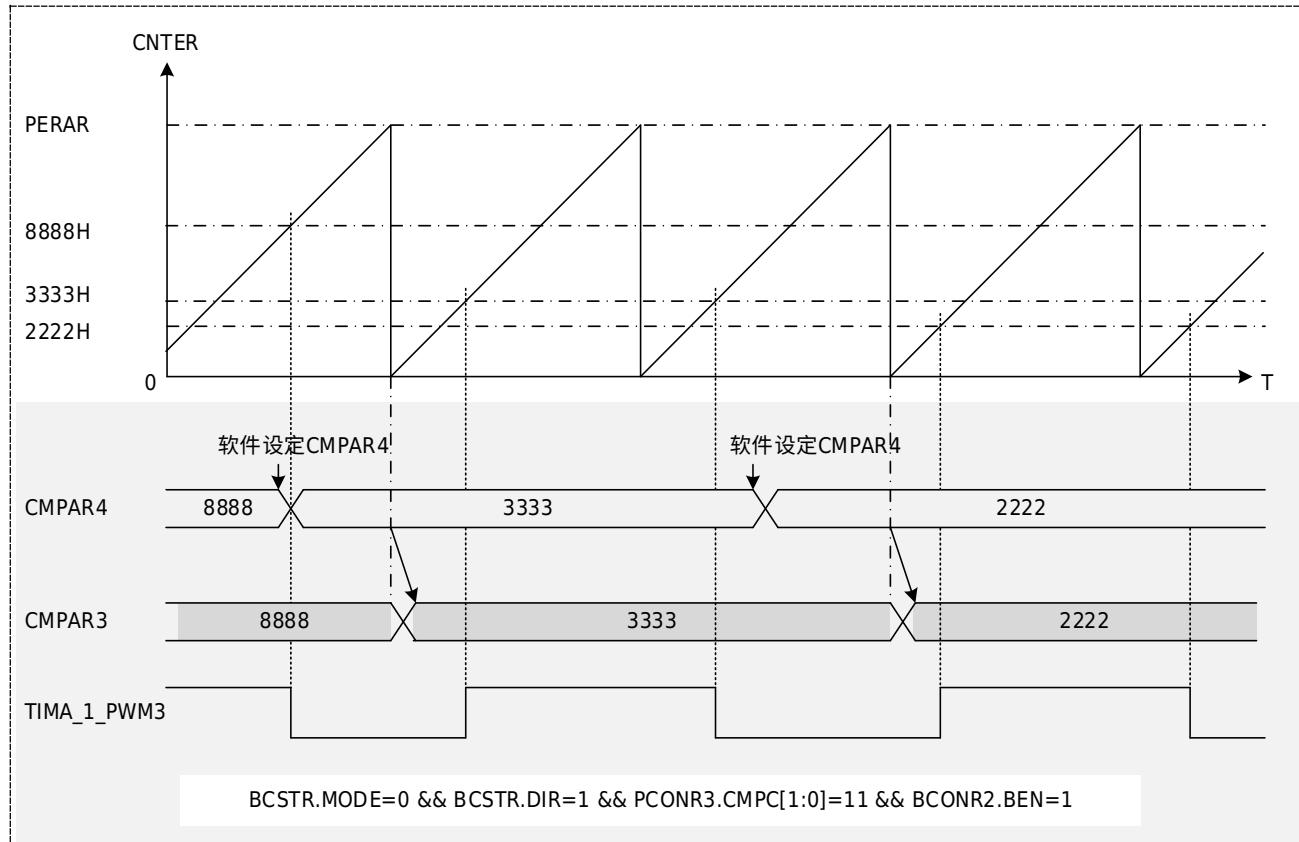


图 23-8 锯齿波模式时缓存动作

### 23.3.8 级联计数

在计数时钟源选择章节中，当时钟源选择 d) 时，本单元的计数时钟就选择为对称单元的计数溢出（计数上溢或计数下溢）事件，此时两个单元级联合并可实现 32 位计数器。在级联计数中，对称单元的 CNTER 为低 16 位计数器，本单元的 CNTER 为高 16 位计数器。

例如，在三角波向上计数模式(BCSTR.MODE=0、BCSTR.DIR=1)时，设定单元 1 的计数时钟为 PCLK，设定单元 2 的计数时钟源为单元 1 的计数上溢事件（单元 2 的 TMRA\_HCUPR.HCUP11=1），启动单元 1、2 计数（先启动单元 2，再启动单元 1）就实现级联计数。单元 1 的 CNTER 位低 16 位计数器，单元 2 的 CNTER 为高 16 位计数器。如图 23-9 所示是单元 1、2 级联计数的示意图。

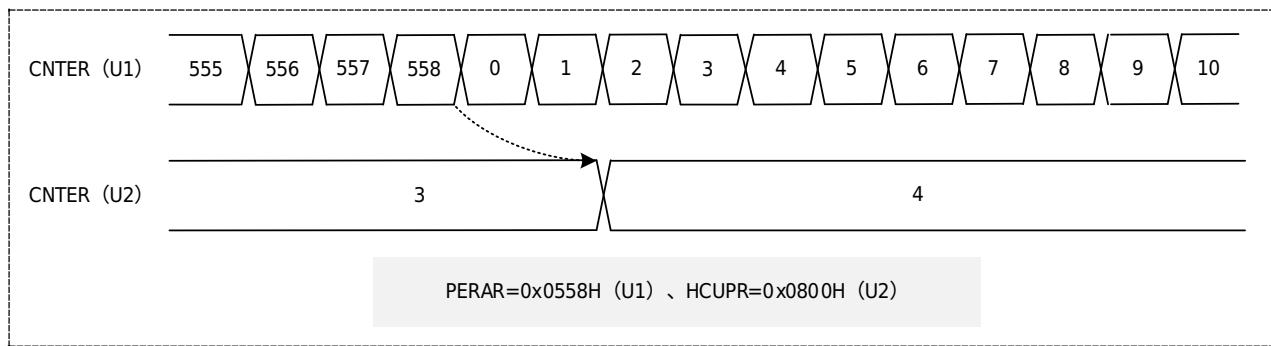


图 23-9 32 位级联计数动作

### 23.3.9 PWM 输出

#### 23.3.9.1 单边对齐 PWM 输出

锯齿波计数模式时，一个单元内的各个通道通过各种端口控制设定，可以实现单边对齐 PWM 输出（在计数周期点对齐）。例如，设定在比较基准值比较匹配时翻转（PCONR.CMPC=11）、在周期基准值比较匹配时翻转（PCONR.PERC=11），就可实现在一个周期内产生单边对齐的 PWM 输出。

图 23-10 所示，是锯齿波模式下，单边对齐 PWM 输出波形例。

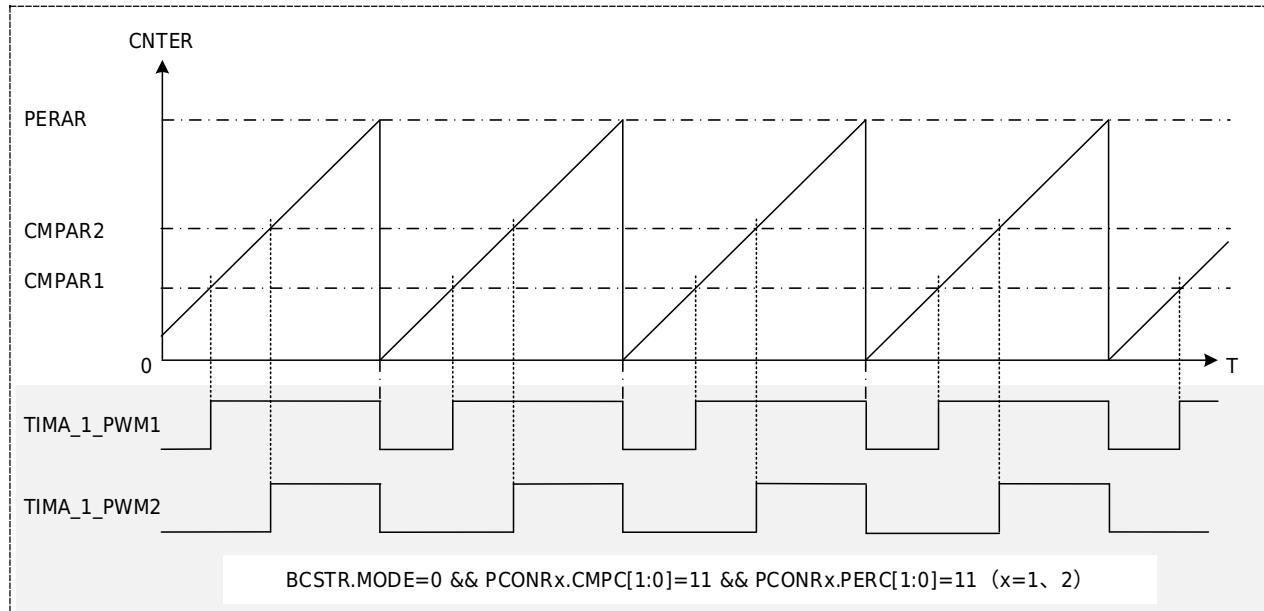


图 23-10 单边对齐 PWM 输出例

#### 23.3.9.2 双边对称 PWM 输出

三角波计数模式时，一个单元内的各个通道通过各种端口控制设定，可以实现双边对称 PWM 输出（以计数峰点对称）。根据通道间的输出关系，可以实现独立的 PWM 输出或互补的 PWM 输出。

图 23-11 所示，是三角波模式下，通道 1、2、3、4 的双边对称 PWM 输出波形例。其中，通道 1、2 可以作为一对互补 PWM 输出。

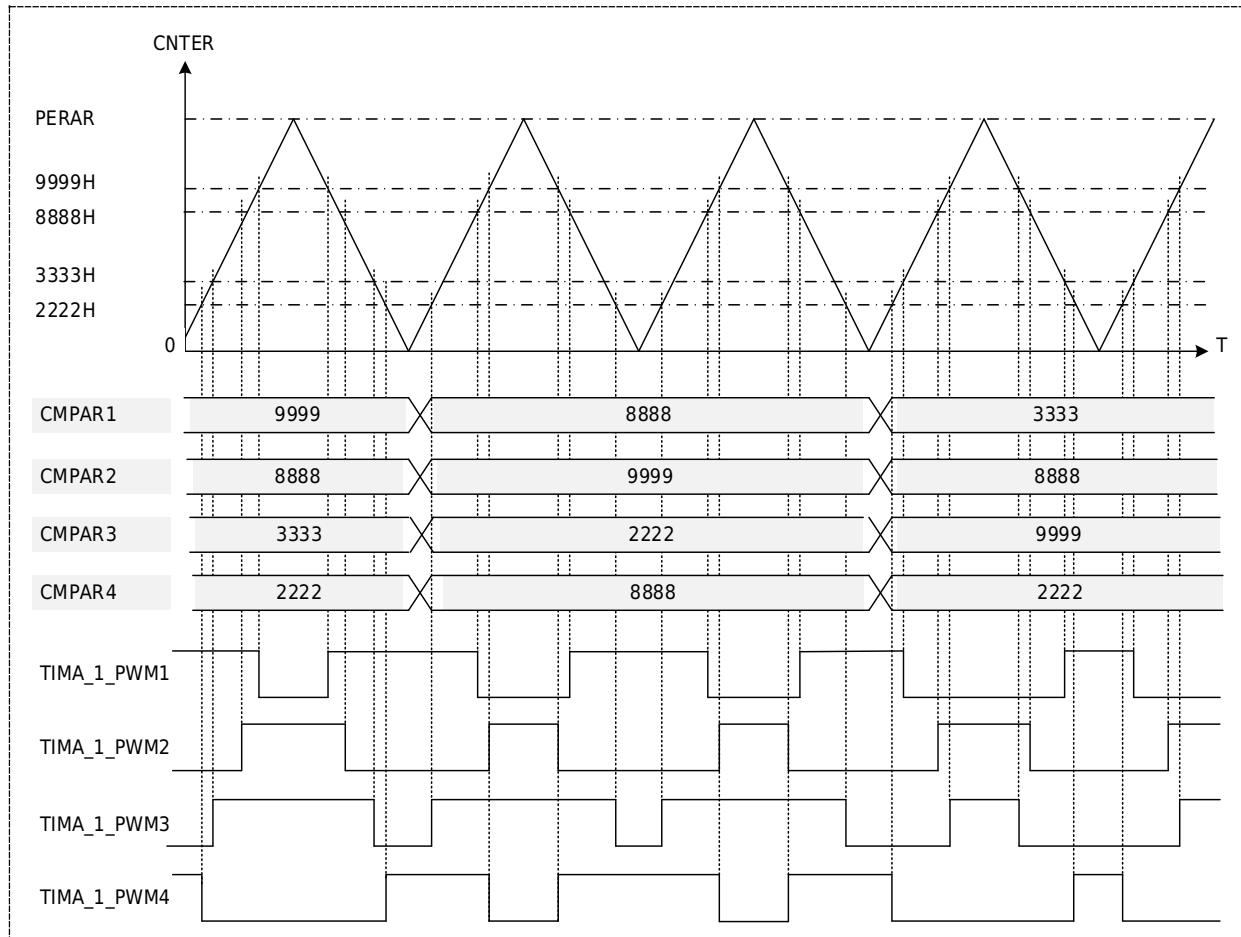


图 23-11 双边对称 PWM 输出例

### 23.3.10 正交编码计数

将 `TIMA_<t>_CLKA` 输入看作 `AIN` 输入、`TIMA_<t>_CLKB` 输入看作 `BIN` 输入、`TIMA_<t>_TRIG` 输入看作 `ZIN` 输入，TimerA 就可以实现三路输入的正交编码计数。

每个单元的 `AIN`、`BIN` 单独动作可以实现位置计数模式；两个单元的 `AIN`、`BIN`、`ZIN` 组合动作可以实现公转计数模式，其中用于位置计数的单元称之为位置计数单元、用于公转计数的单元称之为公转计数单元。公转计数模式时，每两个单元间互相组合（单元 1、2 组合；单元 3、4 组合；单元 5、6 组合；，组合内位置计数单元和公转计数单元可以任意指定。

`AIN` 和 `BIN` 的计数条件使能通过设定硬件递加事件选择寄存器（`HCUPR`）和硬件递减事件选择寄存器（`HCDOR`）中 `TIMA_<t>_CLKA` 和 `TIMA_<t>_CLKB` 的正交关系来实现；`ZIN` 的输入动作通过设定位置计数单元的硬件触发事件选择寄存器（`HCONR`）的清零使能位实现位置定时器清零、通过设定公转单元的硬件递加事件选择寄存器（`HCUPR`）实现公转定时器计数。

### 23.3.10.1 位置计数模式

正交编码位置计数模式，是指根据 AIN、BIN 的输入实现基本计数功能、相位差计数功能和方向计数功能。

#### 基本计数

基本计数动作是根据 AIN 或 BIN 端口的输入时钟进行计数，如下图 23-12 所示。

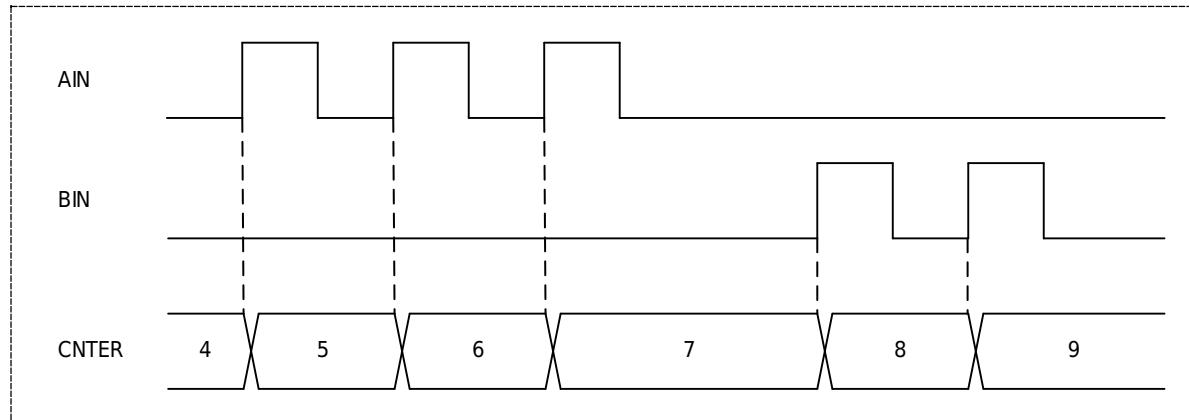


图 23-12 位置模式-基本计数

#### 相位差计数

相位差计数是指根据 AIN 和 BIN 的相位关系进行计数。根据设定的不同，可以实现 1 倍计数、2 倍计数、4 倍计数等，如下图 23-13~图 23-15 所示。

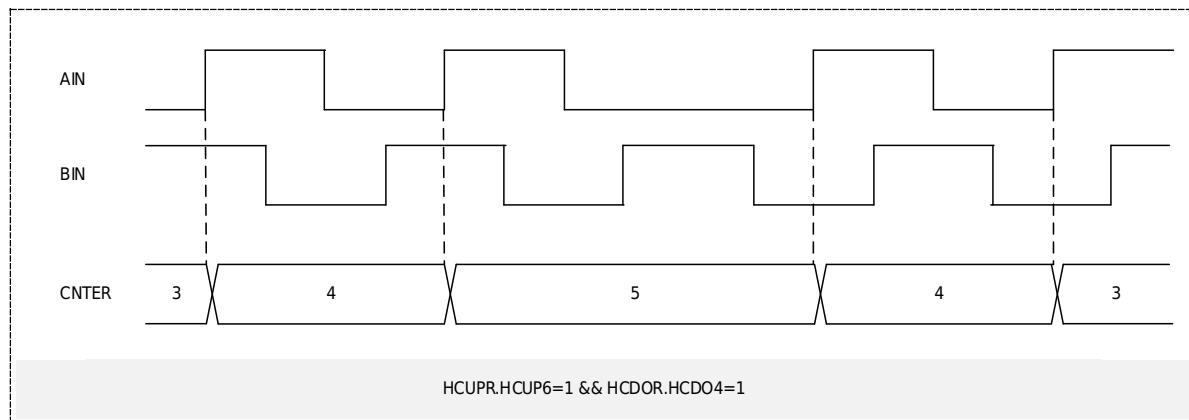


图 23-13 位置计数模式-相位差计数（1 倍计数）

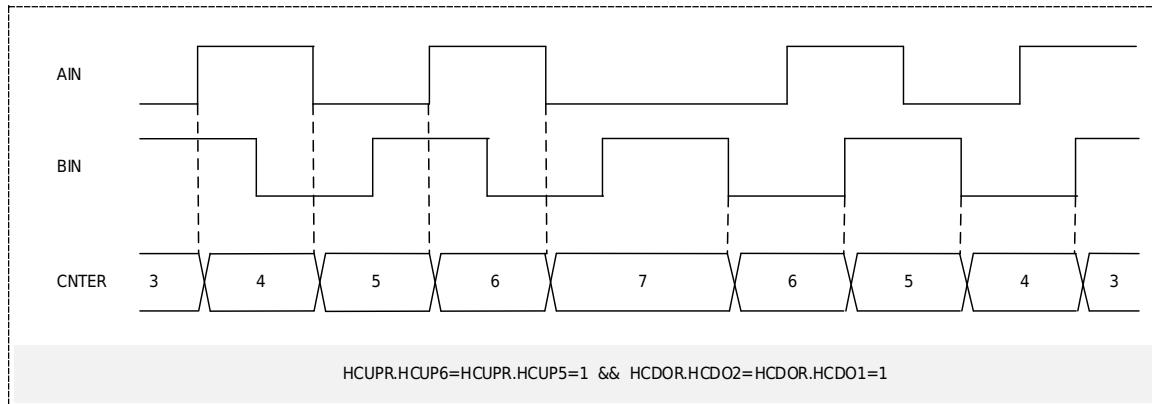


图 23-14 位置计数模式-相位差计数（2 倍计数）

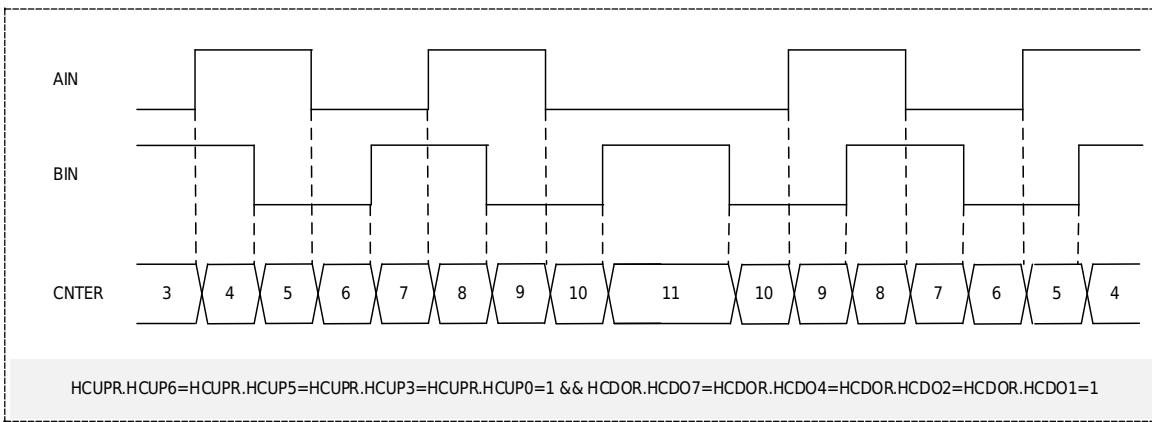


图 23-15 位置计数模式-相位差计数（4 倍计数）

## 方向计数

方向计数是指将 AIN 的输入状态设定为方向控制，将 BIN 的输入作为时钟计数，如下图 23-16 所示。

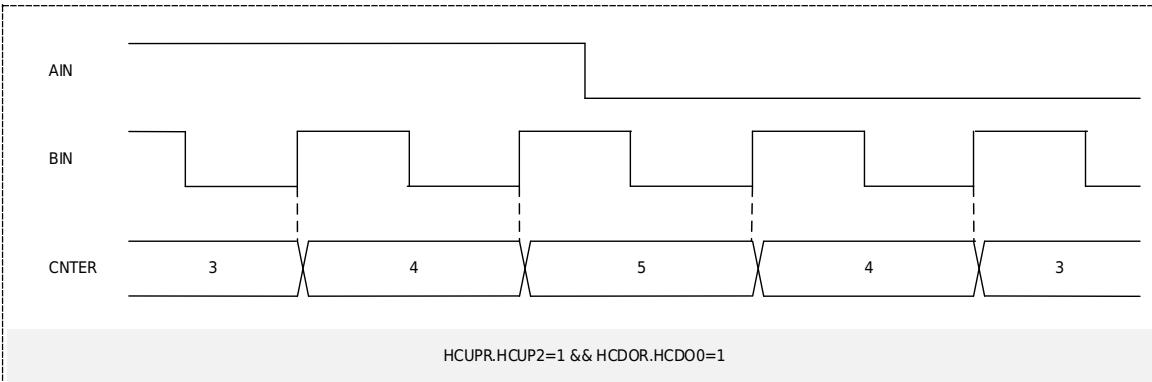


图 23-16 位置计数模式-方向计数

### 23.3.10.2 公转计数模式

正交编码公转计数模式，是指在 AIN、BIN 计数的基础上，加入 ZIN 的输入事件以实现对公转圈数等的判断。公转计数模式时根据公转定时器的计数方式，可实现 Z 相计数功能、位置溢出计数功能和混合计数功能。

#### Z 相计数

Z 相计数是指根据 ZIN 的输入，公转计数单元进行计数，同时将位置计数单元清零的计数动作。如下图 23-17 所示。

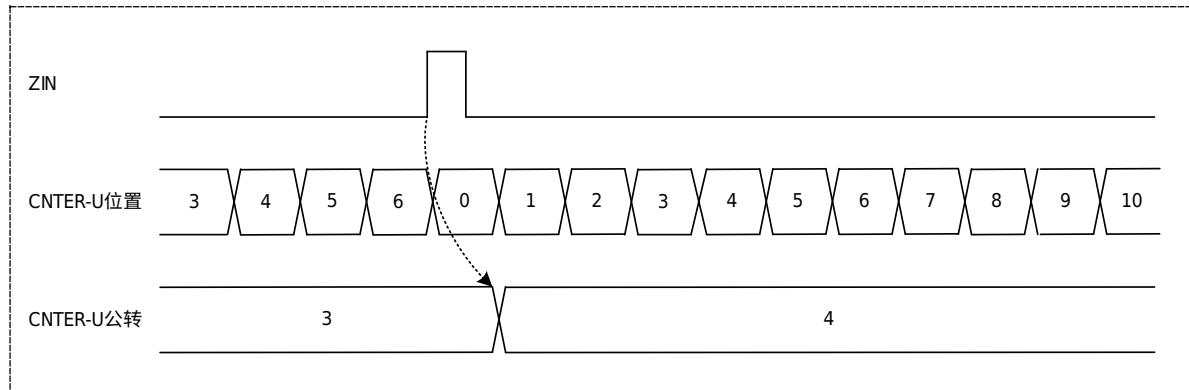


图 23-17 公转计数模式-Z 相计数

#### 位置溢出计数

位置溢出计数是指位置计数单元计数发生上溢或下溢时，产生一个溢出事件，从而触发公转计数单元的定时器进行一次计数（在该计数方式时 ZIN 的输入不进行公转计数单元的计数动作和位置计数单元的清零动作）。

公转计数单元的硬件递加（递减）事件选择寄存器（HCUPR 或 HCDOR）的递加（递减）事件 bit12~11 位使能，位置计数单元的溢出事件就可以触发公转计数单元实现一次计数。如下图 23-18 所示。

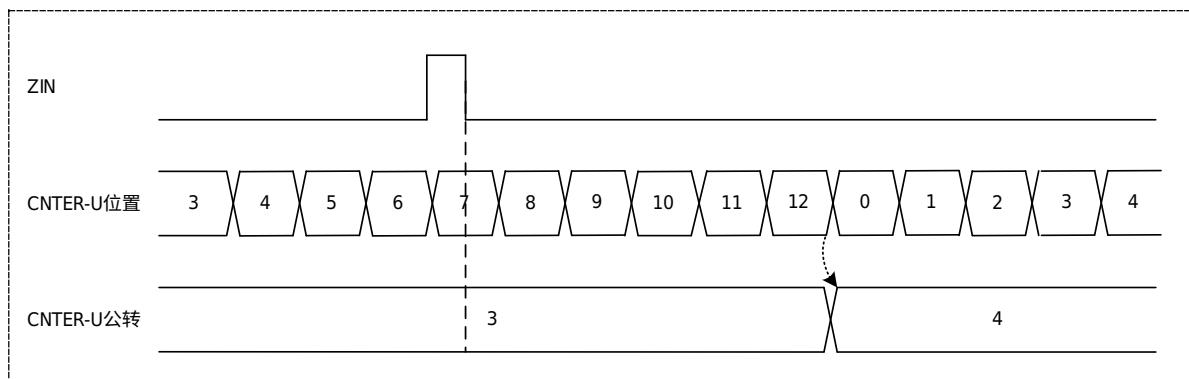


图 23-18 公转计数模式-位置溢出计数

## 混合计数

混合计数是指上述 Z 相计数和位置溢出计数两种计数方式合并起来的计数动作，其实现方式也是上述两种计数方式的组合。如下图 23-19 所示。

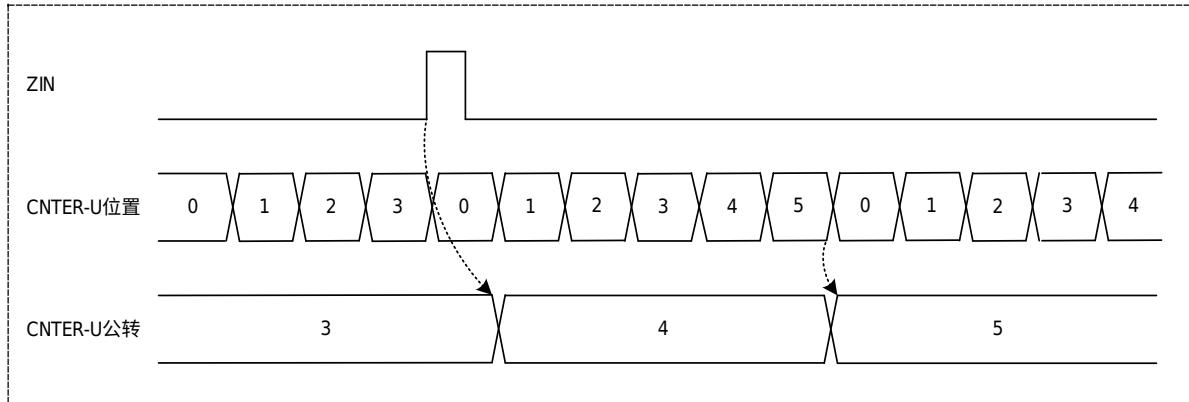


图 23-19 公转计数模式-混合计数

### 23.3.11 红外功能 (IFRP)

红外线接口用来控制发射红外光的 LED，该 LED 可发射红外数据来实现红外遥控，受 TimerA 的 U5 与 U6（第 5 单元与第 6 单元）控制。该功能整合了 TimerA 的 U5 与 U6 的输出来产生红外信号。U5 的 PWM1 输出低频 PWM 信号来产生调制包络信号。U6 的 PWM1 输出高频 PWM 信号来产生载波信号。产生信号前需要使能以上通道的 PWM 输出，并选择 PORT 引脚上的 IFRP 功能。IFRP 的输出信号比输入的 TIMA\_6\_PWM1 高频载波信号有一个 PCLK1 时钟的延迟；两路输入信号的占空比可以改变，当 TIMA\_5\_PWM1 低频调制包络信号电平为高时，IFRP 的输出信号相对于 TIMA\_6\_PWM1 高频载波信号反相；IFRP 的输出信号将保持高频载波信号的完整性，即使 TIMA\_5\_PWM1 低频调制包络信号电平在高低变换后，也不会出现相对于 TIMA\_6\_PWM1 高频载波信号不完整的输出波形的情况。如下图 23-20 所示。

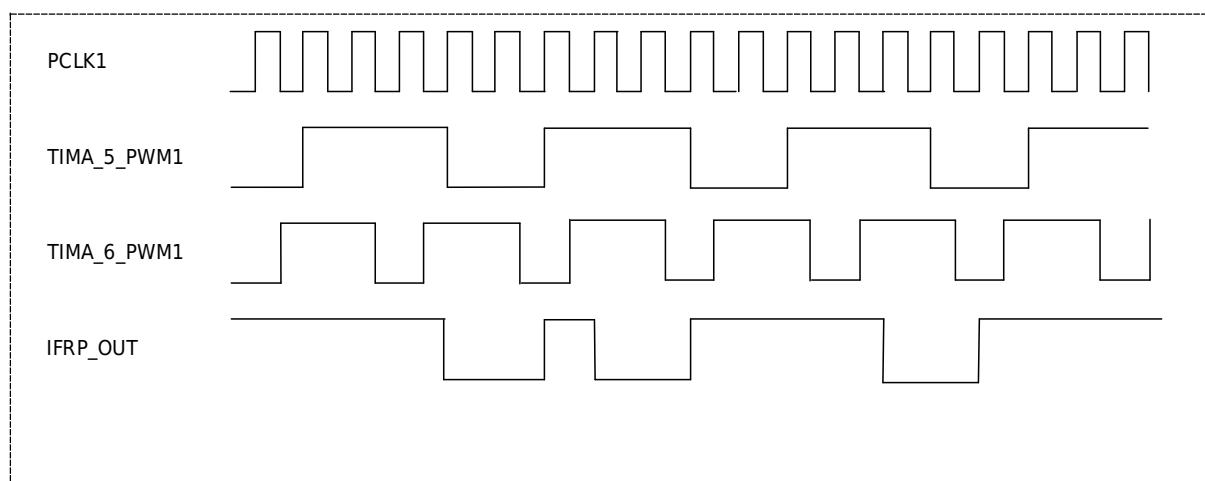


图 23-20 红外输出信号特性示意图

**注意：**

- IFRP 的输出信号由 TimerA 的 U5 与 U6 的 PWM1 决定。
- 为了获得正确的红外信号，U5 应产生低频调制包络信号，U6 应产生高频载波信号。

## 23.4 中断及事件说明

TimerA 含有 3 个中断输出和 3 个事件输出，分别是 1 个比较匹配中断和事件、2 个周期匹配中断和事件。

### 23.4.1 比较匹配中断及事件

比较基准值寄存器（CMPARn）与计数值比较发生比较匹配时，状态标志寄存器（STFLR）中的对应位（STFLR.CMPFn）会被置为 1。此时，若中断控制寄存器（ICONR）的对应位（ICONR.ITENn）设定为 1，则对应的中断请求（TMRA\_<t>\_CMP）会被触发；若事件控制寄存器（ECONR）的对应位（ECONR.ETENn）设定为 1，则对应的事件请求（TMRA\_<t>\_CMP）会被触发（n=1~4）。

捕获控制寄存器（CCONRn）选择的捕获输入有效条件产生时，捕获输入动作发生。此时，若中断控制寄存器（ICONR）的对应位（ICONR.ITENn）设定为 1，则对应的中断请求（TMRA\_<t>\_CMP）被触发；若事件控制寄存器（ECONR）的对应位（ECONR.ETENn）设定为 1，则对应的事件请求（TMRA\_<t>\_CMP）会被触发（n=1~4）。

每个单元内部的 4 个基准值的比较匹配中断和比较匹配事件并非独立输出，比较匹配中断通过“或逻辑”汇总成一个中断输出至中断模块（参见 INTC 章节），比较匹配事件通过“或逻辑”汇总成一个事件输出用于选择触发别的模块。

### 23.4.2 周期匹配中断及事件

锯齿波模式递加计数至上溢点、锯齿波模式递减计数至下溢点、三角波模式计数至谷点或峰点，控制状态寄存器（BCSTRH）的 OVFF 或 UDFF 位会被置为 1。此时，若 BCSTRH.ITENOVF 或 BCSTRH.ITENUDF 位设定为 1 使能中断，则在对应的周期点可触发周期匹配中断（TMRA\_<t>\_OVF 和 TMRA\_<t>\_UDF）输出至中断模块（INTC）；周期匹配事件则无对应的使能位控制，在对应的计数周期点就触发周期匹配事件（TMRA\_<t>\_OVF 和 TMRA\_<t>\_UDF）输出用于选择触发别的模块。

## 23.5 寄存器说明

表 23-3 所示，为 TimerA 模块的寄存器列表。

表 23-3 TimerA 寄存器列表

BASE ADDR:

0x4003A000 (U1)、0x4003A400 (U2)、0x4003A800 (U3)、  
0x4003AC00 (U4)、0x40026000 (U5)、0x40026400 (U6)

寄存器名	符号	偏移量	位宽	复位值
通用计数值寄存器	TMRA_CNTER	0x0000	16	0x0000
周期基准值寄存器	TMRA_PERAR	0x0004	16	0xFFFF
比较基准值寄存器1	TMRA_CMPAR1	0x0040	16	0xFFFF
比较基准值寄存器2	TMRA_CMPAR2	0x0044	16	0xFFFF
比较基准值寄存器3	TMRA_CMPAR3	0x0048	16	0xFFFF
比较基准值寄存器4	TMRA_CMPAR4	0x004C	16	0xFFFF
控制状态寄存器L	TMRA_BCSTRL	0x0080	8	0x02
控制状态寄存器H	TMRA_BCSTRH	0x0081	8	0x00
中断控制寄存器	TMRA_ICONR	0x0090	16	0x0000
事件控制寄存器	TMRA_ECONR	0x0094	16	0x0000
滤波控制寄存器	TMRA_FCONR	0x0098	16	0x0000
状态标志寄存器	TMRA_STFLR	0x009C	16	0x0000
缓存控制寄存器1	TMRA_BCONR1	0x00C0	16	0x0000
缓存控制寄存器2	TMRA_BCONR2	0x00C8	16	0x0000
捕获控制寄存器1	TMRA_CCONR1	0x0100	16	0x0000
捕获控制寄存器2	TMRA_CCONR2	0x0104	16	0x0000
捕获控制寄存器3	TMRA_CCONR3	0x0108	16	0x0000
捕获控制寄存器4	TMRA_CCONR4	0x010C	16	0x0000
端口控制寄存器1	TMRA_PCONR1	0x0140	16	0x0000
端口控制寄存器2	TMRA_PCONR2	0x0144	16	0x0000
端口控制寄存器3	TMRA_PCONR3	0x0148	16	0x0000
端口控制寄存器4	TMRA_PCONR4	0x014C	16	0x0000
硬件触发事件选择寄存器	TMRA_HCONR	0x0084	16	0x0000
硬件递加事件选择寄存器	TMRA_HCUPR	0x0088	16	0x0000
硬件递减事件选择寄存器	TMRA_HCDOR	0x008C	16	0x0000

注意：

- U1~2 单元的通用计数值寄存器、周期基准值寄存器、比较基准值寄存器的位宽均为 32 位，各复位值与 U3~U6 单元不一样。其中通用计数值寄存器的复位值为 0x00000000，周期基准值寄存器、比较基准值寄存器的复位值为 0xFFFFFFFF。

表 23-4 内部触发事件 HTSSR 选择关系对应表

单元数	事件分类	所对应的选择寄存器
U1	计数器触发事件： HCONR、HCUPR、HCDOR 所指的 HTSSR 选择事件	TMRA_TRGSEL0
U2		TMRA_TRGSEL1
U3		TMRA_TRGSEL2
U4		TMRA_TRGSEL3
U5		TMRA_TRGSEL0
U6		TMRA_TRGSEL1
U1	捕获动作触发事件： CCONR 中所指的 HTSSR 选择事件	TMRA_TRGSEL1
U2		TMRA_TRGSEL0
U3		TMRA_TRGSEL3
U4		TMRA_TRGSEL2
U5		TMRA_TRGSEL1
U6		TMRA_TRGSEL0

### 23.5.1 通用计数值寄存器 (TMRA\_CNTER)

U1~2: 复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CNT[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT[15:0]															

位	标记	位名	功能	读写
b31~b0	CNT[31:0]	计数值	当前定时器的计数值	R/W

U3~6: 复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT[15:0]															

位	标记	位名	功能	读写
b15~b0	CNT[15:0]	计数值	当前定时器的计数值	R/W

### 23.5.2 周期基准值寄存器 (TMRA\_PERAR)

U1~2: 复位值: 0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PER[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PER[15:0]															

位	标记	位名	功能	读写
b31~b0	PER[31:0]	计数周期值	设定每轮计数的计数周期值	R/W

U3~6: 复位值: 0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PER[15:0]															

位	标记	位名	功能	读写
b15~b0	PER[15:0]	计数周期值	设定每轮计数的计数周期值	R/W

### 23.5.3 比较基准值寄存器 (TMRA\_CMPARm) (m=1~4)

U1~2: 复位值: 0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CMP[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMP[15:0]															

位	标记	位名	功能	读写
b31~b0	CMP[31:0]	计数比较基准值	设定比较基准值	R/W

U3~6: 复位值: 0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMP[15:0]															

位	标记	位名	功能	读写
b15~b0	CMP[15:0]	计数比较基准值	设定比较基准值	R/W

### 23.5.4 控制状态寄存器 L (TMRA\_BCSTRL)

复位值: 0x02

b7	b6	b5	b4	b3	b2	b1	b0
			CKDIV[3:0]	SYNST	MODE	DIR	START
<hr/>							
位/位域	标记	位名	功能		读写		
b7~b4	CKDIV[3:0]	计数时钟选择	0000: PCLK 0001: PCLK/2 0010: PCLK/4 0011: PCLK/8 0100: PCLK/16 0101: PCLK/32 0110: PCLK/64 0111: PCLK/128 1000: PCLK/256 1001: PCLK/512 1010: PCLK/1024		RW		
b3	SYNST	同步启动使能	0: 对称单元中偶数单元与奇数单元的同步启动无效 1: 对称单元中偶数单元与奇数单元的同步启动有效  <b>注:</b> 对称单元中奇数单元TimerA或无对称单元的TimerA的该位设定无效, 读出时为0		RW		
b2	MODE	计数模式	0: 锯齿波模式 1: 三角波模式		RW		
b1	DIR	计数方向	0: 计数器向下计数 1: 计数器向上计数  <b>注:</b> 三角波模式下, 该位表示计数器方向状态		RW		
b0	START	定时器启动	0: 定时器关闭 1: 定时器启动  <b>注1:</b> 该位在硬件停止条件有效时, 会自动变为0  <b>注2:</b> 对称单元中偶数单元的同步启动功能有效时, 奇数单元软件启动后, 偶数单元的该位也会被置位		RW		

#### 注意:

- 三角波模式下, 定时器动作过程中请勿对该寄存器进行写操作。

### 23.5.5 控制状态寄存器 H (TMRA\_BCSTRH)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
UDFF	OVFF	ITENUDF	ITENOVF		Res		OVSTP

位/位域	标记	位名	功能	读写
b7	UDFF	下溢标志	0: 向下计数时, 未发生计数下溢 1: 向下计数时, 发生计数下溢	RW
b6	OVFF	上溢标志	0: 向上计数时, 未发生计数上溢 1: 向上计数时, 发生计数上溢	RW
b5	ITENUDF	下溢中断使能	0: 计数下溢中断不使能 1: 计数下溢中断使能	RW
b4	ITENOVF	上溢中断使能	0: 计数上溢中断不使能 1: 计数上溢中断使能	RW
b3~b1	Res	保留位	读出时为“0”, 写入时写“0”	RW
b0	OVSTP	计数溢出停止控制	0: 计数器在计数上溢或下溢后, 继续计数 1: 计数器在计数上溢或下溢后, 停止计数	RW

### 23.5.6 中断控制寄存器 (TMRA\_ICONR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												<input type="checkbox"/> IT EN4	<input type="checkbox"/> IT EN3	<input type="checkbox"/> IT EN2	<input type="checkbox"/> IT EN1

位	标记	位名	功能	读写
b15~b4	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b3	ITEN4	计数匹配中断使能4	0: CMPAR4寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断无效 1: CMPAR4寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断使能	R/W
b2	ITEN3	计数匹配中断使能3	0: CMPAR3寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断无效 1: CMPAR3寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断使能	R/W
b1	ITEN2	计数匹配中断使能2	0: CMPAR2寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断无效 1: CMPAR2寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断使能	R/W
b0	ITEN1	计数匹配中断使能1	0: CMPAR1寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断无效 1: CMPAR1寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断使能	R/W

### 23.5.7 事件控制寄存器 (TMRA\_ECONR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Reserved													ET EN4	ET EN3	ET EN2	ET EN1
位	标记	位名	功能													读写
b15~b4	Reserved	-	读出时为“0”,写入时写“0”													R/W
b3	ETEN4	计数匹配事件使能4	0: CMPAR4寄存器与计数值相等时, 或者发生捕获输入事件时, 该事件输出无效 1: CMPAR4寄存器与计数值相等时, 或者发生捕获输入事件时, 该事件输出使能													R/W
b2	ETEN3	计数匹配事件使能3	0: CMPAR3寄存器与计数值相等时, 或者发生捕获输入事件时, 该事件输出无效 1: CMPAR3寄存器与计数值相等时, 或者发生捕获输入事件时, 该事件输出使能													R/W
b1	ETEN2	计数匹配事件使能2	0: CMPAR2寄存器与计数值相等时, 或者发生捕获输入事件时, 该事件输出无效 1: CMPAR2寄存器与计数值相等时, 或者发生捕获输入事件时, 该事件输出使能													R/W
b0	ETEN1	计数匹配事件使能1	0: CMPAR1寄存器与计数值相等时, 或者发生捕获输入事件时, 该事件输出无效 1: CMPAR1寄存器与计数值相等时, 或者发生捕获输入事件时, 该事件输出使能													R/W

### 23.5.8 滤波控制寄存器 (TMRA\_FCONR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	NOFI CKCB[1:0]	NOFI ENCB	-	NOFI CKCA[1:0]	NOFI ENCA	-	-	-	-	-	-	NOFI CKTG[1:0]	NOFI ENTG		
<hr/>															
位	标记	位名	功能	读写											
b15	Reserved	-	读出时为“0”,写入时写“0”	R/W											
			00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64												
b14~b13	NOFICKCB[1:0]	滤波采样基准时钟选择CB		R/W											
b12	NOFIENCB	捕获输入端口滤波CB	0: TIMA_<t>_CLKB端口输入滤波功能无效 1: TIMA_<t>_CLKB端口输入滤波功能使能	R/W											
b11	Reserved	-	读出时为“0”,写入时写“0”	R/W											
			00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64												
b10~b9	NOFICKCA[1:0]	滤波采样基准时钟选择CA		R/W											
b8	NOFIENCA	捕获输入端口滤波CA	0: TIMA_<t>_CLKA端口输入滤波功能无效 1: TIMA_<t>_CLKA端口输入滤波功能使能	R/W											
b7~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W											
			00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64												
b2~b1	NOFICKTG[1:0]	滤波采样基准时钟选择TG		R/W											
b0	NOFIENTG	捕获输入端口滤波TG	0: TIMA_<t>_TRIG输入端口滤波功能无效 1: TIMA_<t>_TRIG输入端口滤波功能使能	R/W											

### 23.5.9 状态标志寄存器 (TMRA\_STFLR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				ICP F4	ICP F3	ICP F2	ICP F1		Reserved		CMP F4	CMP F3	CMP F2	CMP F1	

位	标记	位名	功能	读写
b15-b12	Reserved	-	读出时为“0”,写入时写“0”	R
b11	ICPF4	重复捕获标志4	0: 计数匹配标志位CMPF4为1后未发生捕获完成动作 1: 计数匹配标志位CMPF4为1后再次发生捕获完成动作	R/W
b10	ICPF3	重复捕获标志3	0: 计数匹配标志位CMPF3为1后未发生捕获完成动作 1: 计数匹配标志位CMPF3为1后再次发生捕获完成动作	R/W
b9	ICPF2	重复捕获标志2	0: 计数匹配标志位CMPF2为1后未发生捕获完成动作 1: 计数匹配标志位CMPF2为1后再次发生捕获完成动作	R/W
b8	ICPF1	重复捕获标志1	0: 计数匹配标志位CMPF1为1后未发生捕获完成动作 1: 计数匹配标志位CMPF1为1后再次发生捕获完成动作	R/W
b7-b4	Reserved	-	读出时为“0”,写入时写“0”	R
b3	CMPF4	计数匹配标志4	0: CMPAR4寄存器的值与计数值不相等, 且未发生捕获完成动作 1: CMPAR4寄存器的值与计数值相等, 或发生捕获完成动作	R/W
b2	CMPF3	计数匹配标志3	0: CMPAR3寄存器的值与计数值不相等, 且未发生捕获完成动作 1: CMPAR3寄存器的值与计数值相等, 或发生捕获完成动作	R/W
b1	CMPF2	计数匹配标志2	0: CMPAR2寄存器的值与计数值不相等, 且未发生捕获完成动作 1: CMPAR2寄存器的值与计数值相等, 或发生捕获完成动作	R/W
b0	CMPF1	计数匹配标志1	0: CMPAR1寄存器的值与计数值不相等, 且未发生捕获完成动作 1: CMPAR1寄存器的值与计数值相等, 或发生捕获完成动作	R/W

#### 注意:

- 该寄存器中的标志位只能写入 0, 无法写入 1。

### 23.5.10 缓存控制寄存器 (TMRA\_BCONRm) (m=1~2)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Reserved														BSE1	BSE0	BEN

位	标记	位名	功能	读写
b15~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	BSE1	三角波缓存传送选择1	0: 三角波计数模式计数到谷点时, 缓存值不传送 1: 三角波计数模式计数到谷点时, 缓存值传送, 即: CMPARm -> CMPARn (m=2、4, n=1、3)	R/W
b1	BSE0	三角波缓存传送选择0	0: 三角波计数模式计数到峰点时, 缓存值不传送 1: 三角波计数模式计数到峰点时, 缓存值传送, 即: CMPARm -> CMPARn (m=2、4, n=1、3)	R/W
b0	BEN	缓存使能	0: CMParn基准值的缓存功能无效 1: CMParn基准值的缓存功能有效 (n=1、3)	R/W

**23.5.11 捕获控制寄存器 (TMRA\_CCONRm) (m=1~4)**

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	NOFI CKCP[1:0]	NOFI ENCP	-	-	HICP 4	HICP 3	-	HICP 2	HICP 1	HICP 0	-	-	-	CAP MD	

位	标记	位名	功能	读写
b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14~b13	NOFICKCP[1:0]	滤波采样基准时钟 选择CP	00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64	R/W
b12	NOFIENCP	捕获输入端口滤波 CP	0: TIMA_<t>_PWMr端口输入滤波功能无效 1: TIMA_<t>_PWMr端口输入滤波功能使能 (n=1~4)	R/W
b11~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9	HICP4	捕获输入条件使能4	0: TIMA_<t>_TRIG端口输入采样到下降沿时, 通道m不发生捕获输入动作 1: TIMA_<t>_TRIG端口输入采样到下降沿时, 通道m产生捕获输入动作 注: 该位只有CCONR4寄存器有效。即, 该位有效后且对应事件发生时, 在CCONR4.CAPMD=1的条件下、当前计数器值被捕获保存到CMPAR4, 且STFLR.CMPF4置位	R/W
b8	HICP3	捕获输入条件使能3	0: TIMA_<t>_TRIG端口输入采样到上升沿时, 通道m不发生捕获输入动作 1: TIMA_<t>_TRIG端口输入采样到上升沿时, 通道m产生捕获输入动作 注: 该位只有CCONR4寄存器有效。即, 该位有效后且对应事件发生时, 在CCONR4.CAPMD=1的条件下、当前计数器值被捕获保存到CMPAR4, 且STFLR.CMPF4置位	R/W
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6	HICP2	捕获输入条件使能2	0: TMRA_TRGSEL寄存器中指定的事件发生时, 不发生捕获输入动作 1: TMRA_TRGSEL寄存器中指定的事件发生时, 产生捕获输入动作 注: 具体请参考寄存器说明章节的注意事项说明	R/W
b5	HICP1	捕获输入条件使能1	0: TIMA_<t>_PWMr端口输入采样到下降沿时, 不发生捕获输入动作 1: TIMA_<t>_PWMr端口输入采样到下降沿时, 产生捕获输入动作 (n=1~4)	R/W
b4	HICP0	捕获输入条件使能0	0: TIMA_<t>_PWMr端口输入采样到上升沿时, 不发生捕获输入动作 1: TIMA_<t>_PWMr端口输入采样到上升沿时, 产生捕获输入动作 (n=1~4)	R/W
b3~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	CAPMD	功能模式选择	0: 比较输出功能 1: 捕获输入功能	R/W

### 23.5.12 端口控制寄存器 (TMRA\_PCONRm) (m=1~4)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	OUTEN	-	-	FORC[1:0]	PERC[1:0]		CMPC[1:0]	STPC[1:0]		STAC[1:0]			

位	标记	位名	功能	读写
b15~b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	OUTEN	输出使能	0: PWM输出功能时的TIMA_<t>_PWMn端口输出无效 1: PWM输出功能时的TIMA_<t>_PWMn端口输出有效 (n=1~4)	R/W
b11~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9~b8	FORC[1:0]	强制端口状态设定	0x: 设定无效定 10: 下周期开始, TIMA_<t>_PWMn端口输出设定为低电平 11: 下周期开始, TIMA_<t>_PWMn端口输出设定为高电平 (n=1~4) 注1: 下周期是指硬件计数模式或锯齿波计数到上溢点或下溢点、 三角波计数到谷点 注2: 该寄存器位可用于实现PWM输出占空比0%或100%的控制	R/W
b7~b6	PERC[1:0]	周期值匹配时端口状态设定	00: 计数值与PERAR相等时, TIMA_<t>_PWMn端口输出设定为低电平 01: 计数值与PERAR相等时, TIMA_<t>_PWMn端口输出设定为高电平 10: 计数值与PERAR相等时, TIMA_<t>_PWMn端口输出保持先前状态 11: 计数值与PERAR相等时, TIMA_<t>_PWMn端口输出设定为反转电平 (n=1~4)	R/W
b5~b4	CMPC[1:0]	比较值匹配时端口状态设定	00: 计数值与CMPARn相等时, TIMA_<t>_PWMn端口输出设定为低电平 01: 计数值与CMPARn相等时, TIMA_<t>_PWMn端口输出设定为高电平 10: 计数值与CMPARn相等时, TIMA_<t>_PWMn端口输出保持先前状态 11: 计数值与CMPARn相等时, TIMA_<t>_PWMn端口输出设定为反转电平 (n=1~4)	R/W
b3~b2	STPC[1:0]	计数停止时端口状态设定	00: 计数停止时, TIMA_<t>_PWMn端口输出设定为低电平 01: 计数停止时, TIMA_<t>_PWMn端口输出设定为高电平 10: 计数停止时, TIMA_<t>_PWMn端口输出保持先前状态 11: 计数停止时, TIMA_<t>_PWMn端口输出保持先前状态 (n=1~4)	R/W
b1~b0	STAC[1:0]	计数开始时端口状态设定	00: 计数开始时, TIMA_<t>_PWMn端口输出设定为低电平 01: 计数开始时, TIMA_<t>_PWMn端口输出设定为高电平 10: 计数开始时, TIMA_<t>_PWMn端口输出保持先前状态 11: 计数开始时, TIMA_<t>_PWMn端口输出保持先前状态	R/W

---

(n=1~4)

注：该位设定只在不分频（BCCTRL.CKDIV=4'h0）的情况下有效，其它分频请设定为2'b10或2'b11

---

### 23.5.13 硬件触发事件选择寄存器 (TMRA\_HCONR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HCLE6	HCLE5	HCLE4	HCLE3	-	HCLE2	HCLE1	HCLE0	-	HSTP2	HSTP1	HSTP0	-	HSTA2	HSTA1	HSTA0

位	标记	位名	功能	读写
b15	HCLE6	硬件清零条件6	条件: TIMA_<t>_PWM3端口输入采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b14	HCLE5	硬件清零条件5	条件: TIMA_<t>_PWM3端口输入采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b13	HCLE4	硬件清零条件4	条件: 本单元为单元m时, 单元n的TRIG端口输入采样到下降沿 (当m=1、3、5、7、9、11时, n=2、4、6、8、10、12; 当 m=2、4、6、8、10、12时, n=1、3、5、7、9、11) 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b12	HCLE3	硬件清零条件3	条件: 本单元为单元m时, 单元n的TRIG端口输入采样到上升沿 (当m=1、3、5、7、9、11时, n=2、4、6、8、10、12; 当 m=2、4、6、8、10、12时, n=1、3、5、7、9、11) 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b11	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b10	HCLE2	硬件清零条件2	条件: TMRA_TRGSEL寄存器中指定的事件发生 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效 注: 具体请参考寄存器说明章节的注意事项说明	R/W
b9	HCLE1	硬件清零条件1	条件: TIMA_TRIG端口输入采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b8	HCLE0	硬件清零条件0	条件: TIMA_TRIG端口输入采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b7	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b6	HSTP2	硬件停止条件2	条件: TMRA_TRGSEL寄存器中指定的事件发生 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效 注: 具体请参考寄存器说明章节的注意事项说明	R/W
b5	HSTP1	硬件停止条件1	条件: TIMA_<t>_TRIG端口输入采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b4	HSTP0	硬件停止条件0	条件: TIMA_<t>_TRIG端口输入采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b3	Reserved	-	读出时为“0”, 写入时写“0”	R/W

			条件：TMRA_TRGSEL寄存器中指定的事件发生	
b2	HSTA2	硬件启动条件2	0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
			注：具体请参考寄存器说明章节的注意事项说明	
			条件：	
			1) 本单元TIMA_<t>_TRIG端口输入采样到下降沿（同步启动功能无效） 2) TIMA_n_TRIG端口输入采样到下降沿（同步启动功能有效）	
b1	HSTA1	硬件启动条件1	0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
			注：条件2) 中，当本单元为2、4、6、8、10、12时，n=1、3、5、7、9、11；当本单元为1、3、5、7、9、11时，该功能无效	
			条件：	
			1) 本单元TIMA_<t>_TRIG端口输入采样到上升沿（同步启动功能无效） 2) TIMA_n_TRIG端口输入采样到上升沿（同步启动功能有效）	
b0	HSTA0	硬件启动条件0	0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
			注：条件2) 中，当本单元为2、4、6、8、10、12时，n=1、3、5、7、9、11；当本单元为1、3、5、7、9、11时，该功能无效	

### 23.5.14 硬件递加事件选择寄存器 (TMRA\_HCUPR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	HC UP12	HC UP11	HC UP10	HC UP9	HC UP8	HC UP7	HC UP6	HC UP5	HC UP4	HC UP3	HC UP2	HC UP1	HC UP0

位	标记	位名	功能	读写
b15~b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	HCUP12	硬件递加条件12	条件: 本单元为单元m时, 单元n发生计数下溢 (当m=1、3、5、7、9、11时, n=2、4、6、8、10、12; 当m=2、4、6、8、10、12时, n=1、3、5、7、9、11) 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b11	HCUP11	硬件递加条件11	条件: 本单元为单元m时, 单元n发生计数上溢 (当m=1、3、5、7、9、11时, n=2、4、6、8、10、12; 当m=2、4、6、8、10、12时, n=1、3、5、7、9、11) 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b10	HCUP10	硬件递加条件10	条件: TMRA_TRGSEL寄存器中指定的事件发生 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效 注: 具体请参考寄存器说明章节的注意事项说明	R/W
b9	HCUP9	硬件递加条件9	条件: TIMA_<t>_TRIG端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b8	HCUP8	硬件递加条件8	条件: TIMA_<t>_TRIG端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b7	HCUP7	硬件递加条件7	条件: TIMA_<t>_CLKB端口为高电平时, TIMA_<t>_CLKA端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b6	HCUP6	硬件递加条件6	条件: TIMA_<t>_CLKB端口为低电平时, TIMA_<t>_CLKA端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b5	HCUP5	硬件递加条件5	条件: TIMA_<t>_CLKB端口为低电平时, TIMA_<t>_CLKA端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b4	HCUP4	硬件递加条件4	条件: TIMA_<t>_CLKB端口为高电平时, TIMA_<t>_CLKA端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b3	HCUP3	硬件递加条件3	条件: TIMA_<t>_CLKA端口为高电平时, TIMA_<t>_CLKB端口上采样到下降沿 0: 条件匹配时, 硬件递加无效	R/W

			1: 条件匹配时, 硬件递加有效	
b2	HCUP2	硬件递加条件2	条件: TIMA_<t>_CLKA端口为高电平时, TIMA_<t>_CLKB端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b1	HCUP1	硬件递加条件1	条件: TIMA_<t>_CLKA端口为低电平时, TIMA_<t>_CLKB端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b0	HCUP0	硬件递加条件0	条件: TIMA_<t>_CLKA端口为低电平时, TIMA_<t>_CLKB端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W

### 23.5.15 硬件递减事件选择寄存器 (TMRA\_HCDOR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	HC DO12	HC DO11	HC DO10	HC DO9	HC DO8	HC DO7	HC DO6	HC DO5	HC DO4	HC DO3	HC DO2	HC DO1	HC DO0

位	标记	位名	功能	读写
b15~b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	HCDO12	硬件递减条件12	条件: 本单元为单元m时, 单元n发生计数下溢 (当m=1、3、5、7、9、11时, n=2、4、6、8、10、12; 当m=2、4、6、8、10、12时, n=1、3、5、7、9、11) 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b11	HCDO11	硬件递减条件11	条件: 本单元为单元m时, 单元n发生计数上溢 (当m=1、3、5、7、9、11时, n=2、4、6、8、10、12; 当m=2、4、6、8、10、12时, n=1、3、5、7、9、11) 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b10	HCDO10	硬件递减条件10	条件: TMRA_TRGSEL寄存器中指定的事件发生 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效 注: 具体请参考寄存器说明章节的注意事项说明	R/W
b9	HCDO9	硬件递减条件9	条件: TIMA_<t>_TRIG端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b8	HCDO8	硬件递减条件8	条件: TIMA_<t>_TRIG端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b7	HCDO7	硬件递减条件7	条件: TIMA_<t>_CLKB端口为高电平时, TIMA_<t>_CLKA端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b6	HCDO6	硬件递减条件6	条件: TIMA_<t>_CLKB端口为高电平时, TIMA_<t>_CLKA端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b5	HCDO5	硬件递减条件5	条件: TIMA_<t>_CLKB端口为低电平时, TIMA_<t>_CLKA端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b4	HCDO4	硬件递减条件4	条件: TIMA_<t>_CLKB端口为低电平时, TIMA_<t>_CLKA端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b3	HCDO3	硬件递减条件3	条件: TIMA_<t>_CLKA端口为高电平时, TIMA_<t>_CLKB端口上采样到下降沿 0: 条件匹配时, 硬件递减无效	R/W

			1: 条件匹配时, 硬件递减有效	
b2	HCDO2	硬件递减条件2	条件: TIMA_<t>_CLKA端口为高电平时, TIMA_<t>_CLKB端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b1	HCDO1	硬件递减条件1	条件: TIMA_<t>_CLKA端口为低电平时, TIMA_<t>_CLKB端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b0	HCDO0	硬件递减条件0	条件: TIMA_<t>_CLKA端口为低电平时, TIMA_<t>_CLKB端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W

## 24 通用定时器 (Timer2)

### 24.1 简介

通用定时器 2 (Timer2) 是一个可以实现同步计数、异步计数方式的基本定时器。该定时器内含 2 个通道 (CH-A 和 CH-B)。每个通道均有一个输出端口，可实现基本的方波输出；每个通道均有 2 个输入端口，一个是时钟输入端口，可实现端口异步计数；一个是触发输入端口，可实现定时器启动、停止、清零、计数动作及计数值捕获输入；支持脉宽测量和周期测量。本系列产品中搭载 4 个单元的 Timer2。

### 24.2 基本框图

Timer2 基本的功能及特性如下：

表 24-1 Timer2 的基本功能及特性

基本动作	硬件触发计数器启动、停止、清零、计数及捕获输入
	同步计数、异步计数
	脉宽测量
	周期测量
	方波输出
	事件信号输出
中断类型	计数比较匹配中断
	计数溢出中断

Timer2 的基本框图如图 24-1 所示。框图中所示“ $<t>$ ”表示单元编号，即“ $<t>$ ”为 1~4，本章节后文提到“ $<t>$ ”时均指单元编号，不再赘述。框图中  $\text{TIM2}_{<t>}\text{CLKA/B}$  端口输入上的 ANF 是一个模拟滤波单元，滤波宽度为 40ns（典型条件）。

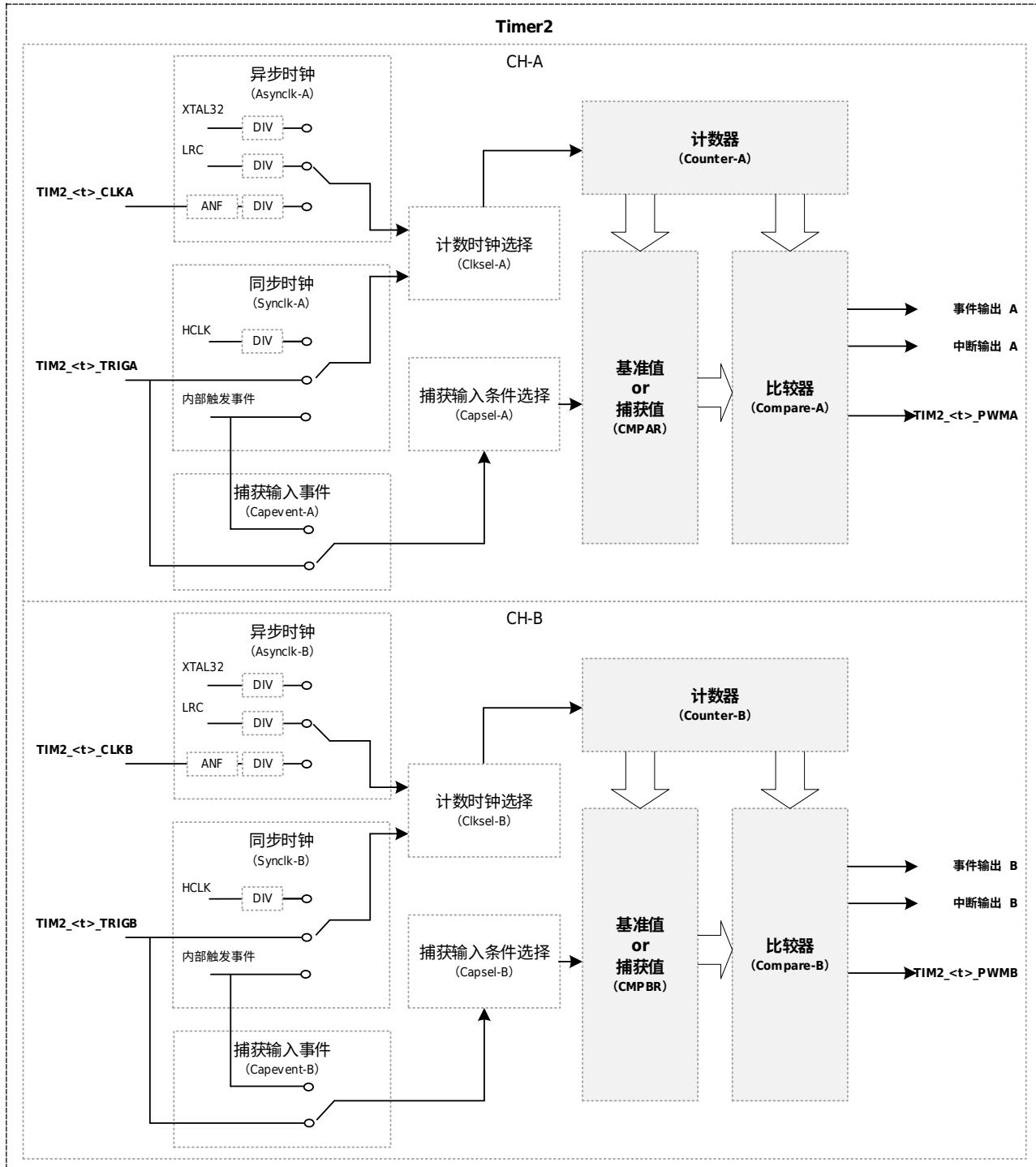


图 24-1 Timer2 基本框图

表 24-2 所示为 Timer2 的输入输出端口列表。

表 24-2 Timer2 端口列表

端口名	方向	功能
TIM2_<t>_CLKA/B	I	异步时钟输入端口
TIM2_<t>_TRIGA/B	I	1) 硬件启动、停止、清零、计数输入端口 2) 捕获输入端口
TIM2_<t>_PWMA/B	O	方波输出端口

## 24.3 功能说明

### 24.3.1 时钟源选择

Timer2 的计数方式可以选择同步计数方式或异步计数方式。

同步计数方式是指定时器的计数时钟和总线访问时钟（寄存器读写操作时钟）有同步时序关系；异步计数方式是指定时器的计数时钟和总线访问时钟（寄存器读写操作时钟）是非同步的时序关系。异步计数方式时对寄存器读操作时，定时器等的状态可能正在发生变化、读出不可预期的状态。因此，在异步计数方式时，寄存器读操作须在计数停止状态下实现。

#### 24.3.1.1 同步计数时钟源

同步计数方式时 ( $BCONR.SYNNSA<B>=0$ )，时钟源可以有以下几种选择 ( $BCONR.SYNCLK A<B>[1:0]$  或  $BCONR.SYNCLK A<B> T[1:0]$  设定)：

- a) PCLK1 及 PCLK1 的 2、4、8、16、32、64、128、256、512、1024 分频作为同步计数时钟 ( $BCONR.SYNCLK A<B>[1:0]=00$  &  $BCONR.CKDIV A<B>[3:0]$  设定)
- b) TIM2\_<t>\_TRIGA、TIM2\_<t>\_TRIGB 端口的输入有效选择沿被内部 PCLK1 采样后作为同步计数时钟 ( $BCONR.SYNCLK A<B>[1:0]=01$  或 10)
- c) 内部硬件触发事件输入作为同步计数时钟 ( $BCONR.SYNCLK A<B>[1:0]=11$ )
- d) Timer6 的计数上溢事件 ( $BCONR.SYNCLKA<B>T[0]=1$ )
- e) Timer6 的计数下溢事件 ( $BCONR.SYNCLKA<B>T[1]=1$ )

计数时钟源选择 a 时为软件计数模式，计数时钟源选择 b、c、d、e 时为硬件计数模式。上述描述可以看到，b、c、d、e 时钟互相独立，可分别设定有效或无效，并且当选择 b、c、d、e 时钟时，a 时钟自动无效。选择 d、e 时钟源时，请设定 PCLK0 和 PCLK1 同频。

### 24.3.1.2 异步计数时钟源

异步计数方式时 (BCONR.SYNS A<B>=1) , 时钟源可以有以下几种选择 (BCONR.ASYNCLK A<B> [1:0] 设定选) :

- a) LRC 时钟源输入及其 2、4、8、16、32、64、128、256、512、1024 分频作为异步计数时钟 (BCONR.ASYNCLK A<B>=00 & BCONR.CKDIV A<B> [3:0] 设定)
- b) XTAL32 时钟源输入及其 2、4、8、16、32、64、128、256、512、1024 分频作为异步计数时钟 (BCONR.ASYNCLK A<B>=01 & BCONR.CKDIV A<B> [3:0] 设定)
- c) TIM2\_<t>\_CLKA、TIM2\_<t>\_CLKB 时钟输入端口及其 2、4、8、16、32、64、128、256、512、1024 分频作为异步计数时钟 (BCONR.ASYNCLK A<B>=10 & BCONR.CKDIV A<B> [3:0] 设定)

### 24.3.2 比较输出

Timer2 有 2 个输出端口 (TIM2\_<t>\_PWMA、TIM2\_<t>\_PWMB) , 可以实现基本的方波输出。

CMPAR、CMPBR 寄存器对应了 TIM2\_<t>\_PWMA、TIM2\_<t>\_PWMB 的计数比较基准值。当定时器的计数值 CNTAR 和 CMPAR 相等时, TIM2\_<t>\_PWMA 端口输出指定的电平; 当定时器的计数值 CNTBR 和 CMPBR 相等时, TIM2\_<t>\_PWMB 端口输出指定的电平。

TIM2\_<t>\_PWMA/B 端口的计数起始电平、停止电平、计数比较匹配时的电平等, 可由端口控制寄存器 (PCONR) 的 PCONR.STACA<B>[1:0]、PCONR.STPCA<B>[1:0]、PCONR.CMPCA<B>[1:0] 位设定。图 24-2 为比较输出动作例。

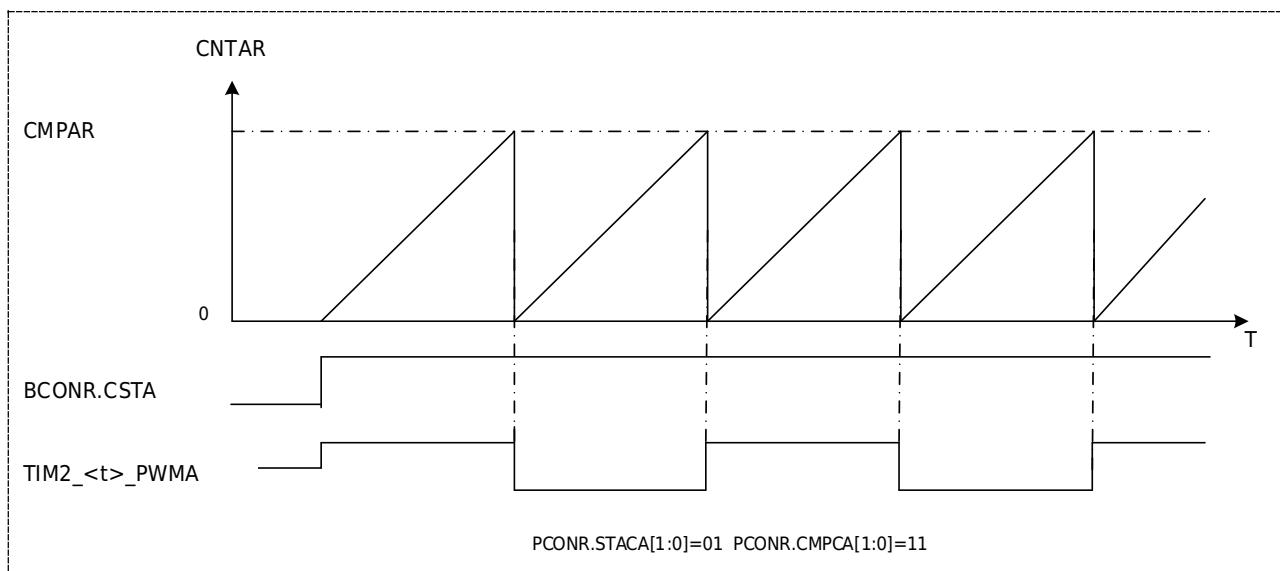


图 24-2 比较输出动作

### 24.3.3 硬件触发

Timer2 的硬件触发动作有两类，一个是 TIM2\_<t>\_TRIGA/B 端口的输入事件，另一个是内部硬件触发事件。这两类事件均可以实现计数器的启动、停止、清零及硬件捕获输入功能。各种功能组合设定也可实现脉宽测量、周期测量等。

TIM2\_<t>\_TRIGA/B 端口输入类型选择（上升沿或下降沿）及功能选择（启动、停止、清零、捕获输入）、内部硬件触发事件的功能选择（启动、停止、清零、捕获输入）可通过硬件控制寄存器（HCONR）的相应位设定。

Timer2 有一个内部硬件触发源，事件源可通过触发选择寄存器（HTSSR）中对应的编号设定来选择，具体的事件对应关系请参考 INTC 章节。使用内部硬件触发功能时，需要先将功能时钟控制寄存器（PWC\_FCG0）的外围电路触发功能使能位置 1。

#### 24.3.3.1 启动停止清零

每个通道的硬件启动、停止、清零条件由硬件控制寄存器（HCONR）的相关设定来决定。图 24-3 是硬件启动、清零的动作例。

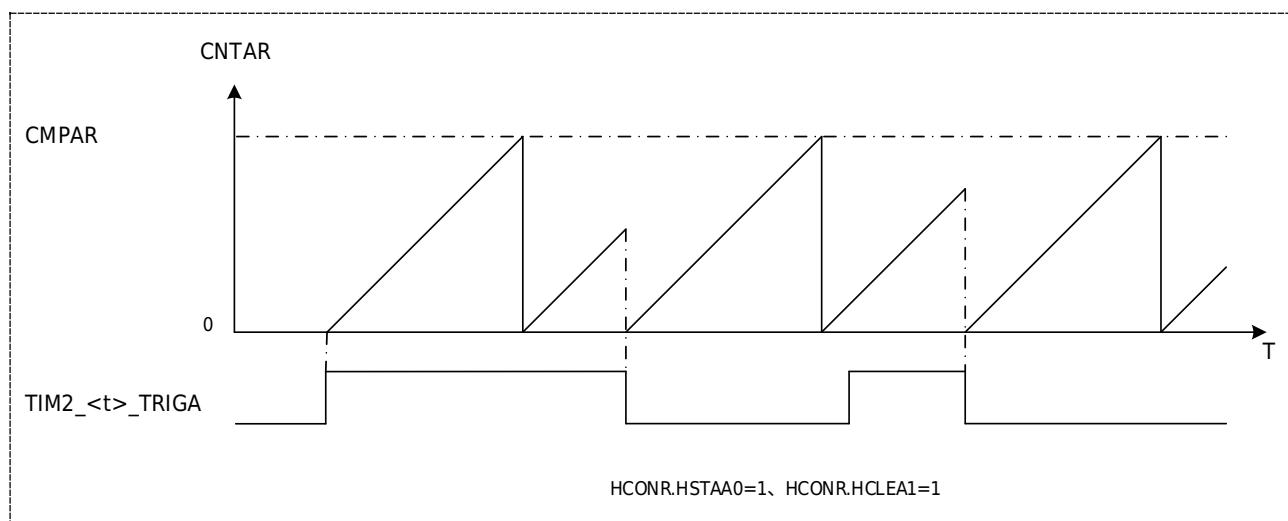


图 24-3 硬件启动、清零动作

### 24.3.3.2 捕获输入动作

TIM2\_<t>\_TRIGA/B 端口输入或内部触发事件作为捕获输入条件时，可实现捕获输入功能。当硬件控制寄存器 (HCONR) 选择的捕获输入条件有效时，当前的计数值就被保存到相应的寄存器 (CMPAR、CMPBR) 中。图 24-4 为捕获输入的动作例。

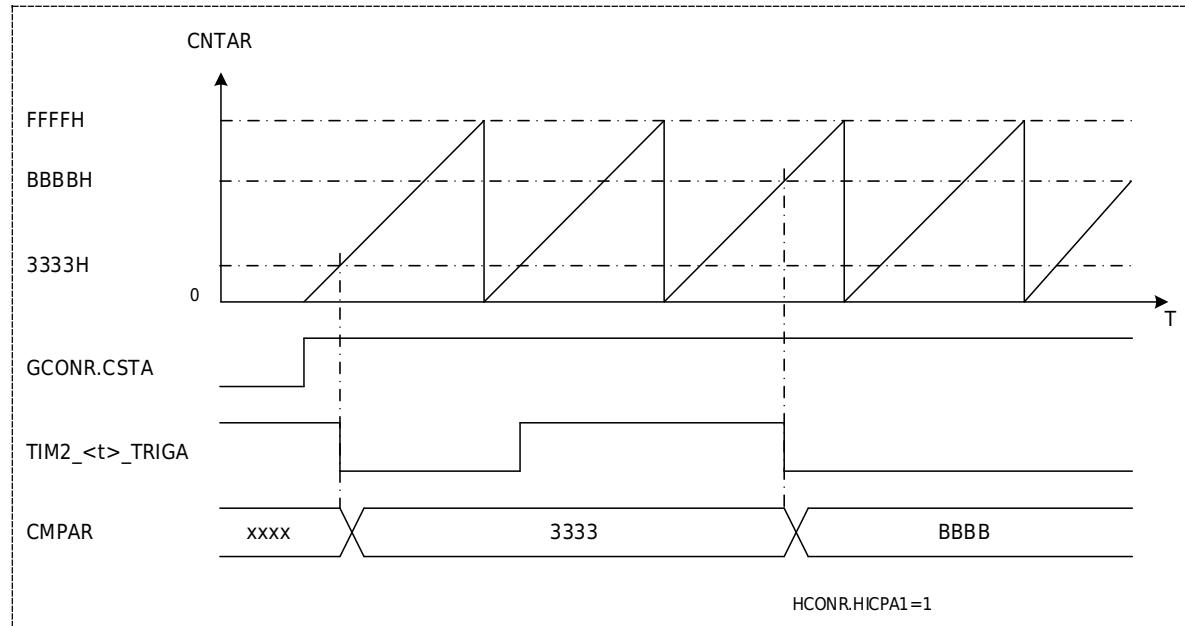


图 24-4 捕获输入动作

### 24.3.3.3 脉宽测量

将 A 通道的硬件启动条件设为 TIM2\_<t>\_TRIGA 的上升沿，硬件清零条件、停止条件和捕获输入条件均设为 TIM2\_<t>\_TRIGA 的下降沿，就可以实现连续的脉冲宽度测量。B 通道可实现同样功能。对应动作如图 24-5 所示。

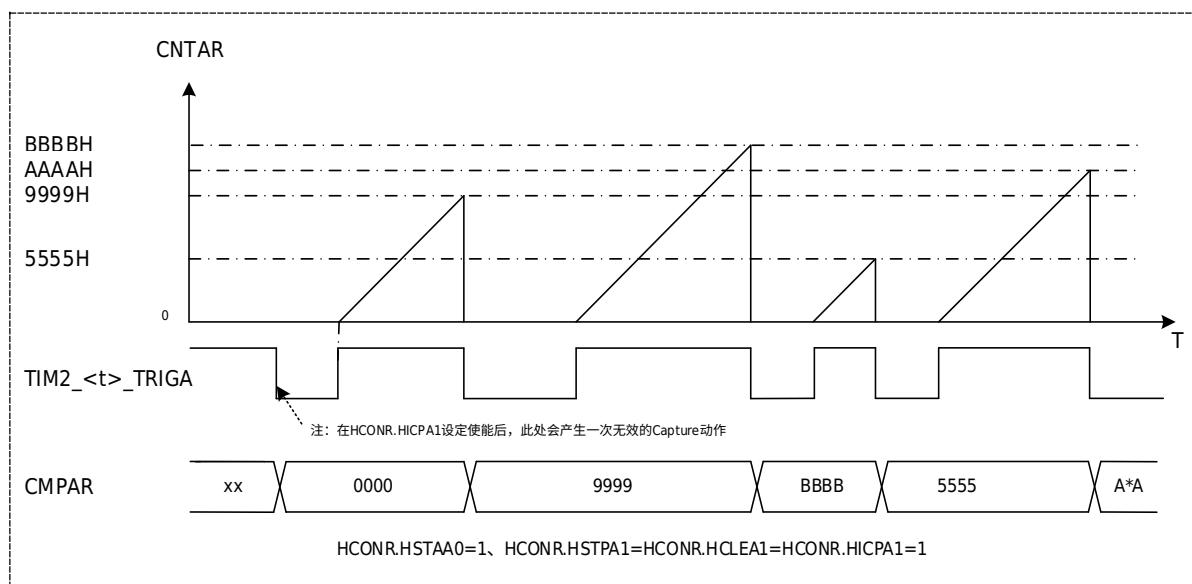


图 24-5 脉宽测量

### 24.3.3.4 周期测量

将 A 通道的硬件启动条件、硬件清零条件和捕获输入条件设为 TIM2\_<t>\_TRIGA 的相同沿（上升沿或下降沿），就可以实现连续的周期宽度测量。B 通道可实现同样功能。对应动作如图 24-6 所示。

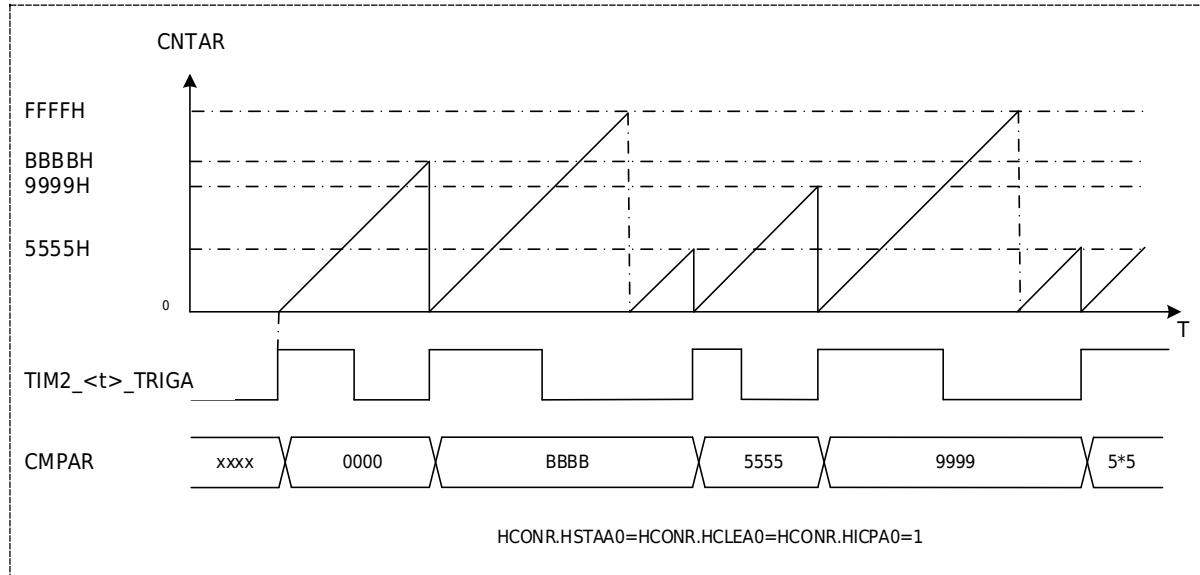


图 24-6 周期测量

### 24.3.4 数字滤波

Timer2 的 TRIG 输入端口 (TIM2\_<t>\_TRIGA/B) 具有数字滤波功能。可通过设定端口控制寄存器 (PCONR.NOFIENA<B>) 的使能位开启对应端口的滤波功能。滤波用的基准时钟也通过端口控制寄存器 (PCONR.NOFICKA<B>[1:0]) 设定。

在滤波采样基准时钟采样到端口上 3 次一致的电平时，该电平被当作有效电平传送到模块内部；小于 3 次一致的电平会被当作外部干扰滤掉，不传送到模块内部。其动作例如图 24-7 所示。

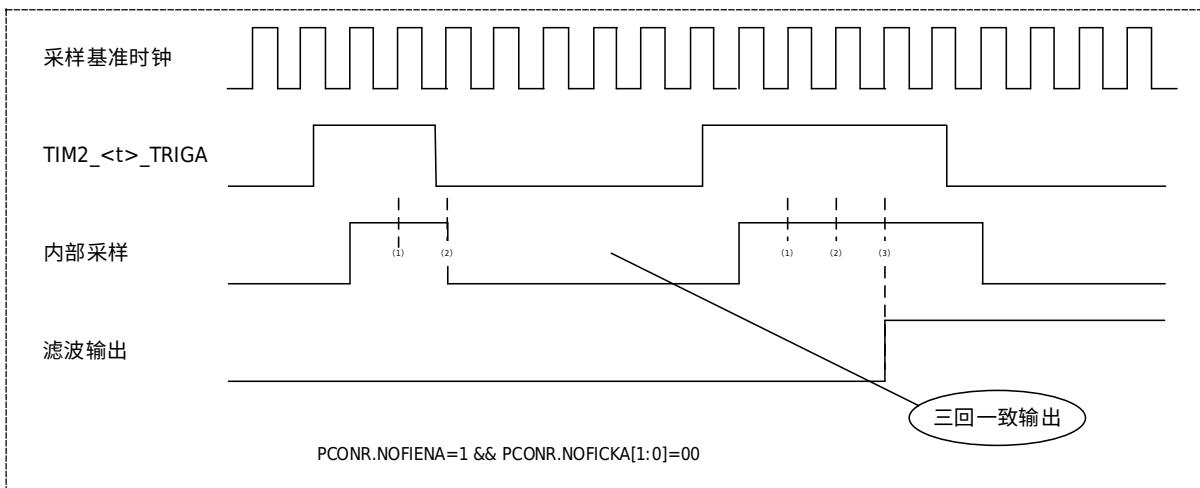


图 24-7 TRIG 输入端口的数字滤波

## 24.4 中断及事件说明

### 24.4.1 中断输出

一个 Timer2 含有 4 个中断，分别是通道 A 和通道 B 的计数比较匹配中断或捕获输入中断、通道 A 和通道 B 的计数溢出中断。

基准值寄存器（CMPAR、CMPBR）共计 2 个，可分别与计数值寄存器（CNTAR、CNTBR）比较产生比较匹配有效信号。计数比较匹配时，状态标志寄存器（STFLR）中的 STFLR.CMFA<B>位分别会被置为 1。此时若设定中断控制寄存器（ICONR）的 ICONR.CMENA<B>位使能中断，则对应的中断请求（TMR2\_<t>\_CMPm, m=A、B）也会被触发。

在硬件控制寄存器（HCONR）选择的捕获输入有效条件产生时，可以产生相应的捕获输入动作。此时若设定中断控制寄存器（ICONR）的 ICONR.CMENA<B>位使能中断，则对应的中断请求（TMR2\_<t>\_CMPm, m=A、B）被触发。

计数值寄存器（CNTAR、CNTBR）计数到 0xFFFF 时发生计数溢出事件，此时状态标志寄存器（STFLR）中的 STFLR.OVFA<B>位会被置为 1。若设定中断控制寄存器（ICONR）的 ICONR.OVENA<B>位使能中断，则对应的中断请求（TMR2\_<t>\_OVFm, m=A、B）也会被触发。

在选择异步计数模式（BCONR.ASYNCLKA[1:0]=2'b10 除外）时，单元 1 的基准值寄存器（CMPAR）产生的比较匹配中断可用于在低功耗模式时对系统进行唤醒，具体请参考【中断控制器（INTC）】章节。

### 24.4.2 事件输出

一个 Timer2 含有 4 个事件，分别是通道 A 和通道 B 的计数比较匹配事件或捕获输入事件、通道 A 和通道 B 的计数溢出事件。

在计数过程中发生计数比较匹配或捕获输入动作或计数溢出时，会产生相应的事件请求（TMR2\_<t>\_CMPm 或 TMR2\_<t>\_OVFm, m=A、B）输出信号，可以用于选择触发其它模块。

## 24.5 寄存器说明

表 24-3 所示，为 Timer2 模块的寄存器列表。

表 24-3 Timer2 寄存器列表

BASE ADDR:

0x40024800 (U1)、0x40024C00 (U2)、0x40025000 (U3)、0x40025400 (U4)

寄存器名	符号	偏移量	位宽	复位值
计数值寄存器	TMR2_CNTAR	0x0000	32	0x00000000
计数值寄存器	TMR2_CNTBR	0x0004	32	0x00000000
基准值寄存器	TMR2_CMPAR	0x0008	32	0x0000FFFF
基准值寄存器	TMR2_CMPBR	0x000C	32	0x0000FFFF
基本控制寄存器	TMR2_BCONR	0x0010	32	0x00000000
中断控制寄存器	TMR2_ICONR	0x0014	32	0x00000000
端口控制寄存器	TMR2_PCONR	0x0018	32	0x00000000
硬件控制寄存器	TMR2_HCONR	0x001c	32	0x00000000
状态标志寄存器	TMR2_STFLR	0x0020	32	0x00000000

### 24.5.1 计数值寄存器 (TMR2\_CNTmR) (m=A、B)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNTA<B>[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为“0”	R											
b15~b0	CNTA<B>[15:0]	计数值	当前定时器的计数值	R/W											

### 24.5.2 基准值寄存器 (TMR2\_CMPmR) (m=A、B)

复位值: 0x0000FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMPA<B>[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为“0”	R											
b15~b0	CMPA<B>[15:0]	基准值	设定计数基准值，产生Compare Match事件	R/W											

### 24.5.3 基本控制寄存器 (TMR2\_BCONR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	SYN CLKBT[1:0]	ASYN CLKB[1:0]	SYN CLKB[1:0]				CKDIV B[3:0]		SYNS B	-	CAP MDB	CST B		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	SYN CLKAT[1:0]	ASYN CLKA[1:0]	SYN CLKA[1:0]				CKDIV A[3:0]		SYNS A	-	CAP MDA	CST A		

位	标记	位名	功能	读写
b31~b30	Reserved	-	读出时为“0”,写入时写“0”	R/W
b29	SYNCLKBT[1]	同步计数同时钟源选择BT	条件: Timer6 的单元 n 发生计数下溢 0: 条件匹配时, Timer2 单元 m 同步计数无效 1: 条件匹配时, Timer2 单元m进行一次同步计数 (当 m=1、2、3、4 时, n=2、4、6、8)	R/W
b28	SYNCLKBT[0]	同步计数同时钟源选择BT	条件: Timer6 的单元 n 发生计数上溢 0: 条件匹配时, Timer2 单元 m 同步计数无效 1: 条件匹配时, Timer2 单元m进行一次同步计数 (当m=1、2、3、4时, n=2、4、6、8)	R/W
b27~b26	ASYNCLKB[1:0]	异步计数同时钟源选择B	00: LRC 01: XTAL32 10: TIM2_<t>_CLKB时钟输入 11: 设定禁止	R/W
b25~b24	SYNCLKB[1:0]	同步计数同时钟源选择B	00: PCLK1 01: TIM2_<t>_TRIGB上升沿 (内部PCLK1同步) 10: TIM2_<t>_TRIGB下降沿 (内部PCLK1同步) 11: 内部触发事件输入	R/W
b23~b20	CKDIVB[3:0]	计数时钟分频选择B	计数时钟分频选择: 0000: 时钟源 0001: 时钟源/2 0010: 时钟源/4 0011: 时钟源/8 0100: 时钟源/16 0101: 时钟源/32 0110: 时钟源/64 0111: 时钟源/128 1000: 时钟源/256 1001: 时钟源/512 1010: 时钟源/1024 请不要设定其它值 注: 被分频的时钟源可以是异步计数时的各种时钟源、同步计数时的PCLK1	R/W
b19	SYNSB	计数方式选择B	0: 同步计数方式 1: 异步计数方式	R/W
b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17	CAPMDB	功能模式选择B	0: 比较输出功能 1: 捕获输入功能	R/W

b16	CSTB	定时器启动B	0: 通道B定时器关闭 1: 通道B定时器启动  注: 该位在硬件触发停止条件有效时, 会自动变为0	R/W
b15~b14	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b13	SYNCLKAT[1]	同步计数同时钟源选择AT	条件: Timer6 的单元 n 发生计数下溢 0: 条件匹配时, Timer2 单元 m 同步计数无效 1: 条件匹配时, Timer2 单元 m 进行一次同步计数 (当 m=1、2、3、4 时, n=1、3、5、7)	R/W
b12	SYNCLKAT[0]	同步计数同时钟源选择AT	条件: Timer6 的单元 n 发生计数上溢 0: 条件匹配时, Timer2 单元 m 同步计数无效 1: 条件匹配时, Timer2 单元 m 进行一次同步计数 (当 m=1、2、3、4 时, n=1、3、5、7)	R/W
b11~b10	ASYNCLKA[1:0]	异步计数同时钟源选择A	00: LRC 01: XTAL32 10: TIM2_<t>_CLKA时钟输入 11: 设定禁止	R/W
b9~b8	SYNCLKA[1:0]	同步计数同时钟源选择A	00: PCLK1 01: TIM2_<t>_TRIGA上升沿 (内部HCLK同步) 10: TIM2_<t>_TRIGA下降沿 (内部HCLK同步) 11: 内部触发事件输入	R/W
b7~b4	CKDIVA[3:0]	计数时钟分频选择A	计数时钟分频选择: 0000: 时钟源 0001: 时钟源/2 0010: 时钟源/4 0011: 时钟源/8 0100: 时钟源/16 0101: 时钟源/32 0110: 时钟源/64 0111: 时钟源/128 1000: 时钟源/256 1001: 时钟源/512 1010: 时钟源/1024 请不要设定其它值  注: 被分频的时钟源可以是异步计数时的各种时钟源、同步计数时的PCLK1	R/W
b3	SYNSA	计数方式选择A	0: 同步计数方式 1: 异步计数方式	R/W
b2	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b1	CAPMDA	功能模式选择A	0: 比较输出功能 1: 捕获输入功能	R/W
b0	CSTA	定时器启动A	0: 通道A定时器关闭 1: 通道A定时器启动  注: 该位在硬件触发停止条件有效时, 会自动变为0	R/W

#### 24.5.4 中断控制寄存器 (TMR2\_ICONR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved												OVENB	CMENB		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												OVENA	CMENA		
<hr/>															
位	标记	位名	功能	读写											
b31~b18	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b17	OVENB	计数溢出中断使能 B	0: 计数值 (CNTBR) =0xFFFF 时, 该中断无效 1: 计数值 (CNTBR) =0xFFFF 时, 该中断使能	R/W											
b16	CMENB	计数匹配中断使能 B	0: CMPBR 寄存器与计数值 (CNTBR) 相等时, 或者发生捕获输入事件时, 该中断无效 1: CMPBR 寄存器与计数值 (CNTBR) 相等时, 或者发生捕获输入事件时, 该中断使能	R/W											
b15~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b1	OVENA	计数溢出中断使能 A	0: 计数值 (CNTAR) =0xFFFF 时, 该中断无效 1: 计数值 (CNTAR) =0xFFFF 时, 该中断使能	R/W											
b0	CMENA	计数匹配中断使能 A	0: CMPAR 寄存器与计数值 (CNTAR) 相等时, 或者发生捕获输入事件时, 该中断无效 1: CMPAR 寄存器与计数值 (CNTAR) 相等时, 或者发生捕获输入事件时, 该中断使能	R/W											

## 24.5.5 端口控制寄存器 (TMR2\_PCONR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	NOFI CKB[1:0]	NOFI ENB	-	-	-	OUT ENB	-	-	CMP CB[1:0]	STP CB[1:0]	STA CB[1:0]				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	NOFI CKA[1:0]	NOFI ENA	-	-	-	OUT ENA	-	-	CMP CA[1:0]	STP CA[1:0]	STA CA[1:0]				

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“0”,写入时写“0”	R/W
b30~b29	NOFICKB	滤波采样基准时钟选择 B	00: 时钟源 01: 时钟源/4 10: 时钟源/16 11: 时钟源/64 注: 该时钟源为异步计数时的各种时钟源、同步计数时的 PCLK1	R/W
b28	NOFIENB	触发端口滤波 B	0: TIM2_<t>_TRIGB 输入端口滤波功能无效 1: TIM2_<t>_TRIGB 输入端口滤波功能使能	R/W
b27~b25	Reserved	-	读出时为“0”,写入时写“0”	R/W
b24	OUTENB	端口输出使能 B	0: TIM2_<t>_PWMB 端口输出无效 1: TIM2_<t>_PWMB 端口输出有效	R/W
b23~b22	Reserved	-	读出时为“0”,写入时写“0”	R/W
b21~b20	CMPCB[1:0]	比较值匹配时端口状态设定 B	00: 计数值 (CNTBR) 与 CMPBR 相等时, TIM2_<t>_PWMB 端口输出设定为低电平 01: 计数值 (CNTBR) 与 CMPBR 相等时, TIM2_<t>_PWMB 端口输出设定为高电平 10: 计数值 (CNTBR) 与 CMPBR 相等时, TIM2_<t>_PWMB 端口输出保持先前状态 11: 计数值 (CNTBR) 与 CMPBR 相等时, TIM2_<t>_PWMB 端口输出设定为反转电平	R/W
b19~b18	STPCB[1:0]	计数停止时端口状态设定 B	00: 计数停止时, TIM2_<t>_PWMB 端口输出设定为低电平 01: 计数停止时, TIM2_<t>_PWMB 端口输出设定为高电平 10: 计数停止时, TIM2_<t>_PWMB 端口输出保持先前状态 11: 计数停止时, TIM2_<t>_PWMB 端口输出保持先前状态	R/W
b17~b16	STACB[1:0]	计数启动时端口状态设定 B	00: 计数启动时, TIM2_<t>_PWMB 端口输出设定为低电平 01: 计数启动时, TIM2_<t>_PWMB 端口输出设定为高电平 10: 计数启动时, TIM2_<t>_PWMB 端口输出保持先前状态 11: 计数启动时, TIM2_<t>_PWMB 端口输出保持先前状态 注: 该位设定只在不分频 (BCONR.CKDIVB=4'h0) 的情况下有效, 其它分频请设定为 2'b10 或 2'b11	R/W
b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14~b13	NOFICKA	滤波采样基准时钟选择 A	00: 时钟源 01: 时钟源/4 10: 时钟源/16 11: 时钟源/64 注: 该时钟源为异步计数时的各种时钟源、同步计数时的 PCLK1	R/W
b12	NOFIENA	触发端口滤波 A	0: TIM2_<t>_TRIGA 输入端口滤波功能无效	R/W

			1: TIM2_<t>_TRIGA 输入端口滤波功能使能	
b11~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8	OUTENA	端口输出使能 A	0: TIM2_<t>_PWMA 端口输出无效 1: TIM2_<t>_PWMA 端口输出有效	R/W
b7~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5~b4	CMPCA[1:0]	比较值匹配时端口状态设定 A	00: 计数值 (CNTAR) 与 CMPAR 相等时, TIM2_<t>_PWMA 端口输出设定为低电平 01: 计数值 (CNTAR) 与 CMPAR 相等时, TIM2_<t>_PWMA 端口输出设定为高电平 10: 计数值 (CNTAR) 与 CMPAR 相等时, TIM2_<t>_PWMA 端口输出保持先前状态 11: 计数值 (CNTAR) 与 CMPAR 相等时, TIM2_<t>_PWMA 端口输出设定为反转电平	R/W
b3~b2	STPCA[1:0]	计数停止时端口状态设定 A	00: 计数停止时, TIM2_<t>_PWMA 端口输出设定为低电平 01: 计数停止时, TIM2_<t>_PWMA 端口输出设定为高电平 10: 计数停止时, TIM2_<t>_PWMA 端口输出保持先前状态 11: 计数停止时, TIM2_<t>_PWMA 端口输出保持先前状态	R/W
b1~b0	STACA[1:0]	计数启动时端口状态设定 A	00: 计数启动时, TIM2_<t>_PWMA 端口输出设定为低电平 01: 计数启动时, TIM2_<t>_PWMA 端口输出设定为高电平 10: 计数启动时, TIM2_<t>_PWMA 端口输出保持先前状态 11: 计数启动时, TIM2_<t>_PWMA 端口输出保持先前状态 注: 该位设定只在不分频 (BCONR.CKDIVA=4'h0) 的情况下有效, 其它分频请设定为 2'b10 或 2'b11	R/W

## 24.5.6 硬件控制寄存器 (TMR2\_HCONR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	HICP B2	HICP B1	HICP B0	-	HCLE B2	HCLE B1	HCLE B0	-	HSTP B2	HSTP B1	HSTP B0	-	HSTA B2	HSTA B1	HSTA B0
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	HICP A2	HICP A1	HICP A0	-	HCLE A2	HCLE A1	HCLE A0	-	HSTP A2	HSTP A1	HSTP A0	-	HSTA A2	HSTA A1	HSTA A0

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“0”,写入时写“0”	R/W
b30	HICPB2	硬件捕获输入条件 B2	条件：内部硬件触发事件有效 0: 条件匹配时，硬件捕获输入无效 1: 条件匹配时，硬件捕获输入有效	R/W
b29	HICPB1	硬件捕获输入条件 B1	条件：TIM2_<t>_TRIGB 端口上采样到下降沿 0: 条件匹配时，硬件捕获输入无效 1: 条件匹配时，硬件捕获输入有效	R/W
b28	HICPB0	硬件捕获输入条件 B0	条件：TIM2_<t>_TRIGB 端口上采样到上升沿 0: 条件匹配时，硬件捕获输入无效 1: 条件匹配时，硬件捕获输入有效	R/W
b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26	HCLEB2	硬件清零条件 B2	条件：内部硬件触发事件有效 0: 条件匹配时，硬件清零无效 1: 条件匹配时，硬件清零有效	R/W
b25	HCLEB1	硬件清零条件 B1	条件：TIM2_<t>_TRIGB 端口上采样到下降沿 0: 条件匹配时，硬件清零无效 1: 条件匹配时，硬件清零有效	R/W
b24	HCLEB0	硬件清零条件 B0	条件：TIM2_<t>_TRIGB 端口上采样到上升沿 0: 条件匹配时，硬件清零无效 1: 条件匹配时，硬件清零有效	R/W
b23	Reserved	-	读出时为“0”,写入时写“0”	R/W
b22	HSTPB2	硬件停止条件 B2	条件：内部硬件触发事件有效 0: 条件匹配时，硬件停止无效 1: 条件匹配时，硬件停止有效	R/W
b21	HSTPB1	硬件停止条件 B1	条件：TIM2_<t>_TRIGB 端口上采样到下降沿 0: 条件匹配时，硬件停止无效 1: 条件匹配时，硬件停止有效	R/W
b20	HSTPB0	硬件停止条件 B0	条件：TIM2_<t>_TRIGB 端口上采样到上升沿 0: 条件匹配时，硬件停止无效 1: 条件匹配时，硬件停止有效	R/W
b19	Reserved	-	读出时为“0”,写入时写“0”	R/W
b18	HSTAB2	硬件启动条件 B2	条件：内部硬件触发事件有效 0: 条件匹配时，硬件启动无效 1: 条件匹配时，硬件启动有效	R/W
b17	HSTAB1	硬件启动条件 B1	条件：TIM2_<t>_TRIGB 端口上采样到下降沿 0: 条件匹配时，硬件启动无效 1: 条件匹配时，硬件启动有效	R/W

			条件: TIM2_<t>_TRIGB 端口上采样到上升沿	
b16	HSTAB0	硬件启动条件 B0	0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W
b15	Reserved	-	读出时为“0”, 写入时写“0”	R/W
			条件: 内部硬件触发事件有效	
b14	HICPA2	硬件捕获输入条件 A2	0: 条件匹配时, 硬件捕获输入无效 1: 条件匹配时, 硬件捕获输入有效	R/W
			条件: TIM2_<t>_TRIGA 端口上采样到下降沿	
b13	HICPA1	硬件捕获输入条件 A1	0: 条件匹配时, 硬件捕获输入无效 1: 条件匹配时, 硬件捕获输入有效	R/W
			条件: TIM2_<t>_TRIGA 端口上采样到上升沿	
b12	HICPA0	硬件捕获输入条件 A0	0: 条件匹配时, 硬件捕获输入无效 1: 条件匹配时, 硬件捕获输入有效	R/W
b11	Reserved	-	读出时为“0”, 写入时写“0”	R/W
			条件: 内部硬件触发事件有效	
b10	HCLEA2	硬件清零条件 A2	0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
			条件: TIM2_<t>_TRIGA 端口上采样到下降沿	
b9	HCLEA1	硬件清零条件 A1	0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
			条件: TIM2_<t>_TRIGA 端口上采样到上升沿	
b8	HCLEA0	硬件清零条件 A0	0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b7	Reserved	-	读出时为“0”, 写入时写“0”	R/W
			条件: 内部硬件触发事件有效	
b6	HSTPA2	硬件停止条件 A2	0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
			条件: TIM2_<t>_TRIGA 端口上采样到下降沿	
b5	HSTPA1	硬件停止条件 A1	0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
			条件: TIM2_<t>_TRIGA 端口上采样到上升沿	
b4	HSTPA0	硬件停止条件 A0	0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b3	Reserved	-	读出时为“0”, 写入时写“0”	R/W
			条件: 内部硬件触发事件有效	
b2	HSTAA2	硬件启动条件 A2	0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W
			条件: TIM2_<t>_TRIGA 端口上采样到下降沿	
b1	HSTAA1	硬件启动条件 A1	0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W
			条件: TIM2_<t>_TRIGA 端口上采样到上升沿	
b0	HSTAA0	硬件启动条件 A0	0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W

### 24.5.7 状态标志寄存器 (TMR2\_STFLR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved												ICPB	OVFB	CMFB	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												ICPA	OVFA	CMFA	
<hr/>															
位	标记	位名	功能	读写											
b31~b19	Reserved	-	读出时为“0”,写入时写“0”	R											
b18	ICPB	重复捕获标志B	0: 计数匹配标志位CMFB为1后未发生捕获输入动作 1: 计数匹配标志位CMFB为1后再次发生捕获输入动作	R/W											
b17	OVFB	计数溢出标志B	0: 计数值 (CNTBR) 未计数到0xFFFF 1: 计数值 (CNTBR) 计数到0xFFFF	R/W											
b16	CMFB	计数匹配标志B	0: CMPBR寄存器的值与计数值 (CNTBR) 不相等且未发生捕获输入动作 1: CMPBR寄存器的值与计数值 (CNTBR) 相等或发生捕获输入动作	R/W											
b15~b3	Reserved	-	读出时为“0”,写入时写“0”	R											
b2	ICPA	重复捕获标志A	0: 计数匹配标志位CMFA为1后未发生捕获输入动作 1: 计数匹配标志位CMFA为1后再次发生捕获输入动作	R/W											
b1	OVFA	计数溢出标志A	0: 计数值 (CNTAR) 未计数到0xFFFF 1: 计数值 (CNTAR) 计数到0xFFFF	R/W											
b0	CMFA	计数匹配标志A	0: CMPAR寄存器的值与计数值 (CNTAR) 不相等且未发生捕获输入动作 1: CMPAR寄存器的值与计数值 (CNTAR) 相等或发生捕获输入动作	R/W											

#### 注意：

- 该寄存器中的标志位只能写入 0，无法写入 1。

## 24.6 使用注意事项

- 在异步计数动作时,需先设定 BCONR.ASYNCLKA<B>位选择异步时钟源,再设定 BCONR.SYNSA<B>位选择异步计数方式, 然后再启动 Timer2。
- 在选择异步计数的情况下, 修改计数值(CNTA<B>R)、基准值 (CMPA<B>R)、启动位 (BCONR.CSTA<B>)、状态位 (STFLR.CMFA<B>、STFLR.OVFA<B>、STFLR.ICPA<B>) 时, Timer2 从接收到写动作后经过 3 个异步计数时钟才将修改值写入对应的寄存器中。
- 在选择异步计数的情况下, 连续对计数值(CNTA<B>R)、基准值 (CMPA<B>R)、启动位 (BCONR.CSTA<B>)、状态位 (STFLR.CMFA<B>、STFLR.OVFA<B>、STFLR.ICPA<B>) 进行写动作时, 需间隔至少 3 个异步计数时钟。
- 在选择异步计数的情况下, 请将 BCONR.SYNCLKA<B>[1:0]设定为 0。

## 25 通用定时器 (Timer0)

### 25.1 简介

通用定时器 0 (Timer0) 是一个可以实现同步计数、异步计数方式的基本定时器。该定时器内含 2 个通道 (CH-A 和 CH-B)，可以在计数期间产生比较匹配事件。该事件可以触发中断，也可作为事件输出来控制其它模块等。本系列产品中搭载 2 个单元的 Timer0。

### 25.2 基本框图

Timer0 的基本框图如图 25-1 所示。

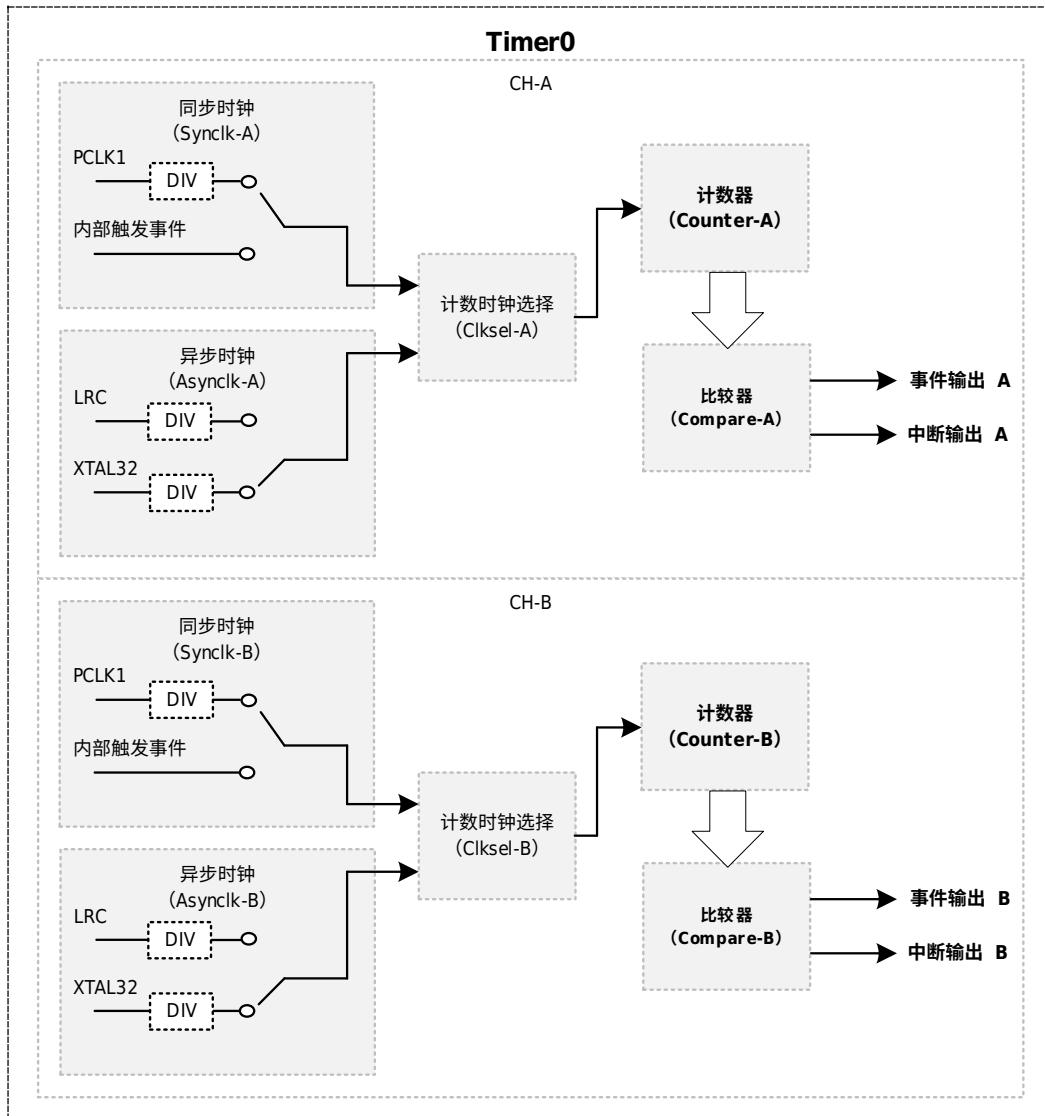


图 25-1 Timer0 基本框图

## 25.3 功能说明

### 25.3.1 时钟源选择

Timer0 的计数方式可以选择同步计数方式或异步计数方式。

同步计数方式是指定时器的计数时钟和总线访问时钟（寄存器读写操作时钟）有同步时序关系；异步计数方式是指定时器的计数时钟和总线访问时钟（寄存器读写操作时钟）是非同步的时序关系。异步计数方式时对寄存器读操作时，定时器等的状态可能正在发生变化、读出不可预期的状态。因此，在异步计数方式时，寄存器读操作须在计数停止状态下实现。

#### 25.3.1.1 同步计数时钟源

同步计数方式时 (BCONR.SYNSA<B>=0)，时钟源可以有以下几种选择 (BCONR.SYNCLKA<B>设定)：

- PCLK1 及 PCLK1 的 2、4、8、16、32、64、128、256、512、1024 分频作为同步计数时钟 (BCONR.SYNCLK A<B>=0 & BCONR.CKDIV A<B> [3:0]设定)
- 内部硬件触发事件输入作为同步计数时钟 (BCONR.SYNCLK A<B>=1)

#### 25.3.1.2 异步计数时钟源

异步计数方式时 (BCONR.SYNS A<B>=1)，时钟源可以有以下几种选择 (BCONR.ASYNCLK A<B>设定选择)：

- LRC 时钟源输入及其 2、4、8、16、32、64、128、256、512、1024 分频作为异步计数时钟 (BCONR.ASYNCLK A<B>=0 & BCONR.CKDIV A<B> [3:0]设定)
- XTAL32 时钟源输入及其 2、4、8、16、32、64、128、256、512、1024 分频作为异步计数时钟 (BCONR.ASYNCLK A<B>=1 & BCONR.CKDIV A<B> [3:0]设定)

### 25.3.2 基本计数

Timer0 的每个通道可设定基准计数值，在计数值和基准值相等时产生计数比较匹配事件。如图 25-2 所示。

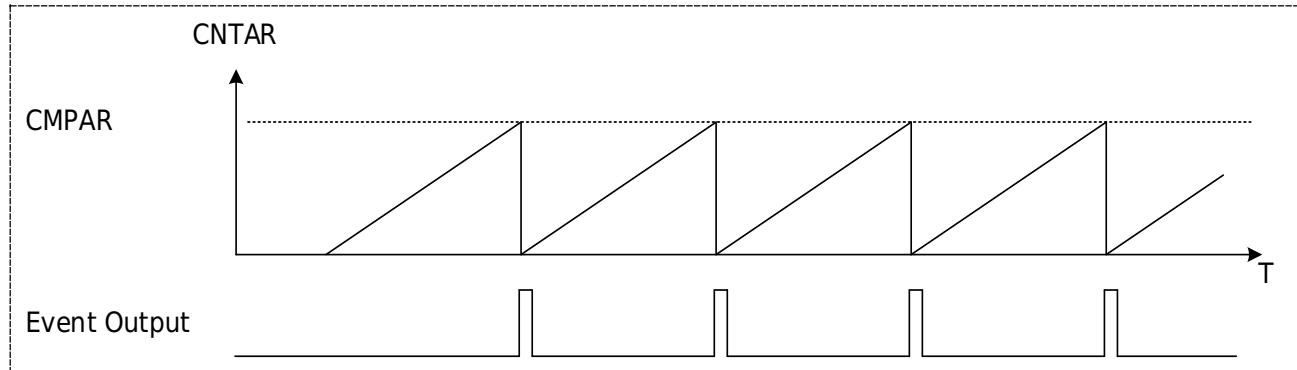


图 25-2 Timer0 计数时序图

### 25.3.3 硬件触发

Timer0 的 2 个通道有一个共用的内部硬件触发源，可以通过基本控制寄存器（BCONR）的相关设定来控制定时器的状态（计数、启动、停止、清零）以及捕获输入动作等。

该硬件触发源的源选择通过向硬件触发选择寄存器（HTSSR）中输入对应的编号来实现，具体的事件对应关系请参考 INTC 章节。使用内部硬件触发功能时，需要先将功能时钟控制寄存器 0（PWC\_FCG0）的外围电路触发功能位使能。

## 25.4 中断及事件说明

### 25.4.1 中断输出

一个 Timer0 含有 2 个中断输出，分别是通道 A 和通道 B 的计数比较匹配中断或捕获输入中断。

基准值寄存器（CMPAR、CMPBR）共计 2 个，可分别与计数值寄存器（CNTAR、CNTBR）比较产生比较匹配有效信号。计数比较匹配时，状态标志寄存器（STFLR）中的 STFLR.CMFA<B>位分别会被置为 1。此时若设定基本控制寄存器（BCONR）的 BCONR.INTENA<B>位使能中断，则对应的中断请求（TMR0\_m\_CMPn, m=1、2; n=A、B）也会被触发。

在内部硬件触发输入作为捕获输入条件时，可以产生相应的捕获输入动作。此时若设定基本控制寄存器（BCONR）的 BCONR.INTENA<B>位使能中断，则对应的中断请求（TMR0\_m\_CMPn, m=1、2; n=A、B）被触发。

在选择异步计数模式时，单元 1 的基准值寄存器（CMPAR）产生的比较匹配中断可用于在低功耗模式时对系统进行唤醒，具体请参考【中断控制器（INTC）】章节。

### 25.4.2 事件输出

一个 Timer0 含有 2 个事件输出，分别是通道 A 和通道 B 的计数比较匹配事件或捕获输入事件。

在计数过程中发生计数比较匹配或捕获输入动作时，会分别产生相应的事件请求（TMR0\_m\_CMPn, m=1、2; n=A、B）输出信号，可以用于选择触发其它模块。

## 25.5 寄存器说明

表 25-1 所示，为 Timer0 模块的寄存器列表。

表 25-1 Timer0 寄存器列表

BASE ADDR: 0x40024000 (U1)、0x40024400 (U2)

寄存器名	符号	偏移量	位宽	复位值
计数值寄存器	TMR0_CNTAR	0x0000	32	0x00000000
计数值寄存器	TMR0_CNTBR	0x0004	32	0x00000000
基准值寄存器	TMR0_CMPAR	0x0008	32	0x0000FFFF
基准值寄存器	TMR0_CMPBR	0x000c	32	0x0000FFFF
基本控制寄存器	TMR0_BCONR	0x0010	32	0x00000000
状态标志寄存器	TMR0_STFLR	0x0014	32	0x00000000

### 25.5.1 计数值寄存器 (TMR0\_CNTmR) (m=A~B)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNTA<B>[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为“0”	R											
b15~b0	CNTA<B>[15:0]	计数值	当前定时器的计数值	R/W											

### 25.5.2 基准值寄存器 (TMR0\_CMPmR) (m=A~B)

复位值: 0x0000FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMPA<B>[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为“0”	R											
b15~b0	CMPA<B>[15:0]	基准值	设定计数基准值，产生Compare Match事件	R/W											

### 25.5.3 基本控制寄存器 (TMR0\_BCONR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
HICPB B	HCLEB B	HSTPB B	HSTAB B	-	ASYNCLKB CLKB	SYNCLKB CLKB	SYNSB SB		CKDIVB[3:0]	-	INTENB ENB	CAPMDB MDB	CSTB CSTB		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HICPA A	HCLEA A	HSTPA A	HSTAA A	-	ASYNCLKA CLKA	SYNCLKA CLKA	SYNSA SA		CKDIVA[3:0]	-	INTENA ENA	CAPMDA MDA	CSTA CSTA		

位	标记	位名	功能	读写
b31	HICPB	硬件触发捕获输入B	条件：内部硬件触发事件有效 0: 条件匹配时，捕获输入无效 1: 条件匹配时，捕获输入有效	R/W
b30	HCLEB	硬件触发清零B	条件：内部硬件触发事件有效 0: 条件匹配时，定时器清零无效 1: 条件匹配时，定时器清零有效	R/W
b29	HSTPB	硬件触发停止B	条件：内部硬件触发事件有效 0: 条件匹配时，定时器停止无效 1: 条件匹配时，定时器停止有效	R/W
b28	HSTAB	硬件触发启动B	条件：内部硬件触发事件有效 0: 条件匹配时，定时器启动无效 1: 条件匹配时，定时器启动有效	R/W
b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26	ASYNCLKB	通道B异步计数时钟源选择	0: LRC 1: XTAL32	R/W
b25	SYNCLKB	通道B同步计数时钟源选择	0: PCLK1 1: 内部硬件触发事件	R/W
b24	SYNSB	通道B计数方式选择	0: 同步计数方式 1: 异步计数方式	R/W
通道B计数时钟分频选择：				
			0000: 时钟源 0001: 时钟源/2 0010: 时钟源/4 0011: 时钟源/8 0100: 时钟源/16 0101: 时钟源/32	
b23~b20	CKDIVB[3:0]	通道B计数时钟分频选择	0110: 时钟源/64 0111: 时钟源/128 1000: 时钟源/256 1001: 时钟源/512 1010: 时钟源/1024	R/W
			请不要设定其它值 注：被分频的时钟源可以是异步计数时的各种时钟源、 同步计数时的PCLK1	
b19	Reserved	-	读出时为“0”,写入时写“0”	R/W
b18	INTENB	计数匹配中断使能B	0: CMPB寄存器与计数值 (CNTBR) 相等时，或者发生 捕获输入事件时，该中断无效	R/W

1: CMPBR寄存器与计数值（CNTBR）相等时，或者发生 捕获输入事件时，该中断使能			
b17	CAPMDB	功能模式选择B	0: 比较输出功能 1: 捕获输入功能
b16	CSTB	定时器启动	0: 通道B定时器关闭 1: 通道B定时器启动 注：该位在硬件触发停止条件有效时，会自动变为0
b15	HICPA	硬件触发捕获输入A	条件：内部硬件触发事件有效 0: 条件匹配时，捕获输入无效 1: 条件匹配时，捕获输入有效
b14	HCLEA	硬件触发清零A	条件：内部硬件触发事件有效 0: 条件匹配时，定时器清零无效 1: 条件匹配时，定时器清零有效
b13	HSTPA	硬件触发停止A	条件：内部硬件触发事件有效 0: 条件匹配时，定时器停止无效 1: 条件匹配时，定时器停止有效
b12	HSTAA	硬件触发启动A	条件：内部硬件触发事件有效 0: 条件匹配时，定时器启动无效 1: 条件匹配时，定时器启动有效
b11	Reserved	-	读出时为“0”，写入时写“0”
b10	ASYNCLKA	通道A异步计数时钟源选择	0: LRC 1: XTAL32
b9	SYNCLKA	通道A同步计数时钟源选择	0: PCLK1 1: 内部硬件触发事件
b8	SYNSA	通道A计数方式选择	0: 同步计数方式 1: 异步计数方式
通道A计数时钟分频选择： 0000: 时钟源 0001: 时钟源/2 0010: 时钟源/4 0011: 时钟源/8 0100: 时钟源/16 0101: 时钟源/32 0110: 时钟源/64 0111: 时钟源/128 1000: 时钟源/256 1001: 时钟源/512 1010: 时钟源/1024 请不要设定其它值 注：被分频的时钟源可以是异步计数时的各种时钟源、 同步计数时的PCLK1			
b7~b4	CKDIVA[3:0]	通道A计数时钟分频选择	0110: 时钟源/64 0111: 时钟源/128 1000: 时钟源/256 1001: 时钟源/512 1010: 时钟源/1024 请不要设定其它值 注：被分频的时钟源可以是异步计数时的各种时钟源、 同步计数时的PCLK1
b3	Reserved	-	读出时为“0”，写入时写“0”
b2	INTENA	计数匹配中断使能A	0: CMPAR寄存器与计数值（CNTAR）相等时，或者发生 捕获输入事件时，该中断使能 1: CMPAR寄存器与计数值（CNTAR）相等时，或者发生 捕获输入事件时，该中断使能
b1	CAPMDA	功能模式选择A	0: 比较输出功能 1: 捕获输入功能

---

b0	CSTA	定时器启动	0: 通道A定时器关闭 1: 通道A定时器启动	R/W
注：该位在硬件触发停止条件有效时，会自动变为0				

---

**注意：**

- 该寄存器中提到的内部硬件触发事件 (bit31~bit28 和 bit15~bit12) 及异步计数时的 XTAL32 时钟源 (bit26 和 bit10)，在 USART 模块的 TIMEOUT 功能有效时均由 USART 模块提供输入，具体请参考【通用同步异步收发器 (USART)】章节介绍。

## 25.5.4 状态标志寄存器 (TMR0\_STFLR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved														ICPB	CMFB
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved														ICPA	CMFA
<hr/>															
位	标记	位名	功能	读写											
b31~b18	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b17	ICPB	重复捕获B	0: 计数匹配B标志位CMFB为1后未发生捕获输入动作 1: 计数匹配B标志位CMFB为1后再次发生捕获输入动作	R/W											
b16	CMFB	计数匹配B	0: CMPBR寄存器的值与计数值 (CNTBR) 不相等且未发生捕获输入动作 1: CMPBR寄存器的值与计数值 (CNTBR) 相等或发生捕获输入动作	R/W											
b15~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b2	ICPA	重复捕获A	0: 计数匹配A标志位CMFA为1后未发生捕获输入动作 1: 计数匹配A标志位CMFA为1后再次发生捕获输入动作	R/W											
b0	CMFA	计数匹配A	0: CMPAR寄存器的值与计数值 (CNTAR) 不相等且未发生捕获输入动作 1: CMPAR寄存器的值与计数值 (CNTAR) 相等或发生捕获输入动作	R/W											

### 注意:

- 该寄存器中的标志位只能写入 0, 无法写入 1。

## 25.6 使用注意事项

- 在异步计数动作时,需先设定 BCONR.ASYNCLKA<B>位选择异步时钟源,再设定 BCONR.SYNSA<B>位选择异步计数方式, 然后再启动 Timer0。
- 在选择异步计数的情况下, 修改计数值(CNTA<B>R)、基准值 (CMPA<B>R)、启动位 (BCONR.CSTA<B>)、状态位 (STFLR.CMFA<B>)、重复捕获状态位 (STFLR.ICPA<B>) 时, Timer0 从接收到写动作后经过 6 个异步计数时钟才将修改值写入对应的寄存器中。
- 在选择异步计数的情况下, 连续对计数值(CNTA<B>R)、基准值 (CMPA<B>R)、启动位 (BCONR.CSTA<B>)、状态位 (STFLR.CMFA<B>)、重复捕获状态位 (STFLR.ICPA<B>) 进行写动作时, 需间隔至少 6 个异步计数时钟。
- 在选择异步计数的情况下, 请将 BCONR.SYNCLKA<B>设定为 0。

## 26 实时时钟 (RTC)

### 26.1 简介

实时时钟 (RTC) 是一个以 BCD 码格式保存时间信息的计数器。记录从 00 年到 99 年间的具体日历时间。支持 12/24 小时两种时制，根据月份和年份自动计算日数 28、29（闰年）、30 和 31 日。表 26-1 所示是其基本特性。

表 26-1 RTC 的基本规格

计数时钟源	外部低速振荡器 (32.768kHz) 内部低速振荡器 (32.768kHz) 外部引脚时钟 XTAL小数分频时钟
基本功能	<ul style="list-style-type: none"><li>BCD码表示秒、分、时、日、周、月、年时间</li><li>软件启动或停止</li><li>12/24时制可选、闰年自动识别</li><li>可编程闹钟</li><li>分布式/均匀式补偿1Hz时钟输出</li><li>时钟误差补偿功能</li></ul>
中断	周期中断 闹钟中断

## 26.2 基本框图

RTC 的基本框图如图 26-1 所示。

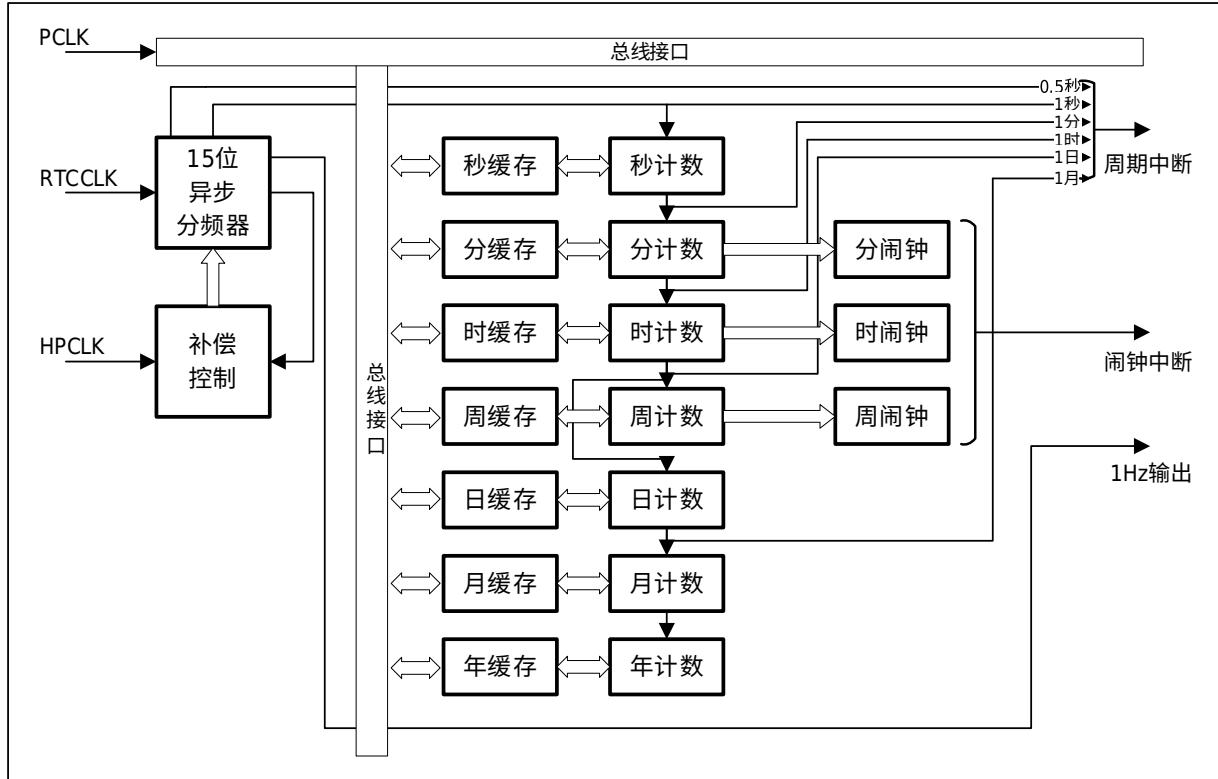


图 26-1 RTC 的基本框图

## 26.3 功能说明

### 26.3.1 上电设定

RTC 寄存器值可以通过上电复位或设定 RTC\_CR0.RESET 复位，在设定控制寄存器、日历初始值、闹钟设置等之后，启动 RTC。RTC 启动之后，其他外部各种复位请求都不能复位 RTC，RTC 会一直处于工作状态。可以设定控制寄存器的 RTC\_CR1.START 位为“0”停止 RTC 工作。RTC 在进行设定时，必须保证时钟源已稳定。

### 26.3.2 RTC 计数开始设定

- 上电后，设定 RTC\_CR0.RESET=0，确认 RESET 位为“0”后，设定 RTC\_CR0.RESET=1，复位所有寄存器；
- 设定 RTC\_CR1.START=0，停止计数；
- 设定系统时钟寄存器，打开外部低速振荡器，再设定 RTC\_CR3，选择 RTC 计数时钟源；
- 设定 RTC\_CR1，设定时制、周期、1Hz 时钟输出；
- 设定秒，分，时，周，日，月，年的日历计数寄存器；
- 需要进行时钟误差补偿时，设定计数时钟误差补偿寄存器 RTC\_ERRCRL，RTC\_ERRCRH；
- 清除寄存器 RTC\_CR2 中的标志寄存器位，并使能中断；
- 设定 RTC\_CR1.START=1，计数开始。

### 26.3.3 系统低功耗模式切换

在 RTC 计数开始后，系统立即切换为低功耗模式时，请执行下列任意一种确认后再进行模式切换。

- 在 RTC\_CR1.START=1 设定后，经过 2 个以上的 RTC 计数时钟后再进行模式切换。
- 在 RTC\_CR1.START=1 设定后，设定 RTC\_CR2.RWREQ=1，查询 RTC\_CR2.RWEN=1。设定日历计数寄存器，再设定 RTC\_CR2.RWREQ=0，查询 RTC\_CR2.RWEN=0，进行模式切换。

### 26.3.4 读出计数寄存器

- 在 RTC\_CR1.START=1 设定后，经过 2 个以上的 RTC 计数时钟后，设定 RTC\_CR2.RWREQ=1，进行日历寄存器读请求；
- 查询直到 RTC\_CR2.RWEN=1；
- 读出全部或者部分秒，分，时，周，日，月，年计数寄存器值；
- 设定 RTC\_CR2.RWREQ=0；
- 查询直到 RTC\_CR2.RWEN=0。

### 26.3.5 写入计数寄存器

- 在 RTC\_CR1.START=1 设定后，经过 2 个以上的 RTC 计数时钟后，设定 RTC\_CR2.RWREQ=1，进行日历寄存器写请求；
- 查询直到 RTC\_CR2.RWEN=1；
- 写入全部或者部分秒，分，时，周，日，月，年计数寄存器值；
- 设定 RTC\_CR2.RWREQ=0。注意，须在 1 秒内完成所有写操作；
- 查询直到 RTC\_CR2.RWEN=0。

### 26.3.6 闹钟设定

- 设定 RTC\_CR2.ALME=0，闹钟禁止；
- 设定 RTC\_CR2.ALMIE=1，闹钟中断许可；
- 分闹钟 RTC\_ALMMIN，时闹钟 RTC\_ALMHOUR，周闹钟 RTC\_ALMEEK 设定；
- 设定 RTC\_CR2.ALME=1，闹钟许可；
- 等待闹钟中断；
- 闹钟发生，RTC\_CR2.ALMF=1，进入闹钟中断处理。

### 26.3.7 时钟误差补偿

由于外部低速晶振在各种温度条件下存在偏差，在需要得到高精度的计数结果时，需要对误差进行补偿。补偿方法参照【时钟误差补偿寄存器（RTC\_ERRCRH、RTC\_ERRCRL）】。

### 26.3.8 1Hz 输出

RTC 可输出 1Hz 时钟，提供三种精度输出方式，第一种，无时钟补偿的普通精度 1Hz 输出；第二种，每 32 秒内平均补偿的分布式补偿 1Hz 输出和第三种每秒补偿的均匀式补偿 1Hz 输出。当时钟误差补偿功能有效 RTC\_ERRCRH.COMPEN=1 时可选择分布式补偿 1Hz 输出和均匀式补偿 1Hz 输出。其中，

**普通精度的 1Hz 输出设定如下：**

- 设定 RTC\_CR0.RESET=0，确认 RESET 位为“0”后，设定 RTC\_CR0.RESET=1，复位日历计数寄存器；
- 设定 RTC\_CR1.START=0，计数停止；
- 1Hz 输出引脚设定；
- RTC\_CR1.ONEHZOE=1，时钟输出许可；
- 设定 RTC\_CR1.START=1，计数开始；
- 等待 2 个计数周期以上；
- 1Hz 输出开始。

**分布式补偿 1Hz 输出设定如下：**

- 设定 RTC\_CR0.RESET=0，确认 RESET 位为“0”后，设定 RTC\_CR0.RESET=1，复位日历计数寄存器；
- 设定 RTC\_CR1.START=0，计数停止；
- 1Hz 输出引脚设定；
- RTC\_CR1.ONEHZOE=1，时钟输出许可；
- 时钟误差补偿寄存器 RTC\_ERRCRL.COMP[7:0] 与 RTC\_ERRCRH.COMP[8] 补偿数设定；
- 时钟误差补偿寄存器 RTC\_ERRCRH.COMPEN=1，误差补偿有效；
- 设定 RTC\_CR1.START=1，计数开始；
- 等待 2 个计数周期以上；
- 1Hz 输出开始。

**均匀式补偿 1Hz 输出设定如下：**

- 设定 RTC\_CR0.RESET=0，确认 RESET 位为“0”后，设定 RTC\_CR0.RESET=1，复位日历计数寄存器；
- 设定 RTC\_CR1.START=0，计数停止；
- RTC 输出引脚设定；
- RTC\_CR1.ONEHZOE=1，时钟输出许可；
- RTC\_CR1.ONEHZSEL=1，选择输出均匀式补偿 1Hz 时钟；
- 时钟误差补偿寄存器 RTC\_ERRCRL.COMP[7:0] 与 RTC\_ERRCRH.COMP[8] 补偿数设定；
- 时钟误差补偿寄存器 RTC\_ERRCRH.COMPEN=1，精度补偿有效；
- 设定 RTC\_CR1.START=1，计数开始；
- 等待 2 个计数周期以上；
- 1Hz 输出开始。

## 26.4 中断说明

RTC 支持 3 种中断类型。计时闹钟中断、定周期中断。

### 26.4.1 闹钟中断

闹钟中断 RTC\_ALM，在控制寄存器 2 (RTC\_CR2) 的 ALMIE=1 并且控制寄存器 2 (RTC\_CR2) 的 ALME=1 时，若当前日历时间与分闹钟寄存器 (RTC\_ALMMIN)、时闹钟寄存器 (RTC\_ALMHOUR)、周闹钟寄存器 (RTC\_ALMWEEK) 相等时，触发闹钟中断。闹钟配置独立的标志寄存器位 RTC\_CR2.ALMF，对 RTC\_CR2.ALMF 位写“0”清除闹钟标志。

### 26.4.2 定周期中断

定周期中断 RTC\_PRD，控制寄存器 2 (RTC\_CR2) 的 PRDIE=1 时，选择的周期发生后，触发定周期唤醒中断。定周期中断配置独立的标志寄存器 RTC\_CR2.PRDF，可通过对 RTC\_CR2.PRDF 位写“0”清除定周期标志。

## 26.5 寄存器说明

表 26-2 所示，为 RTC 模块的寄存器列表。

表 26-2 寄存器列表

寄存器地址：0x4004C000

寄存器名	符号	偏移量	位宽	复位值
控制寄存器0	RTC_CR0	0x0000	8	0x00
控制寄存器1	RTC_CR1	0x0004	8	0x00
控制寄存器2	RTC_CR2	0x0008	8	0x00
控制寄存器3	RTC_CR3	0x000C	8	0x00
秒计数寄存器	RTC_SEC	0x0010	8	0x00
分计数寄存器	RTC_MIN	0x0014	8	0x00
时计数寄存器	RTC_HOUR	0x0018	8	0x12
周计数寄存器	RTC_WEEK	0x001C	8	0x00
日计数寄存器	RTC_DAY	0x0020	8	0x00
月计数寄存器	RTC_MON	0x0024	8	0x00
年计数寄存器	RTC_YEAR	0x0028	8	0x00
分闹钟寄存器	RTC_ALMMIN	0x002C	8	0x00
时闹钟寄存器	RTC_ALMHOUR	0x0030	8	0x12
周闹钟寄存器	RTC_ALMEEK	0x0034	8	0x00
时钟误差补偿寄存器	RTC_ERRCRH	0x0038	8	0x00
时钟误差补偿寄存器	RTC_ERRCRL	0x003C	8	0x20

### 26.5.1 控制寄存器 0 (RTC\_CR0)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
			Res				
位	标记	位名	功能				读写
b7~b1	Reserved	-	读出时为“0”,写入时写“0”				R/W
b0	RESET	RTC日历计数器复位	写入状态 0: 初始化寄存器无效 1: 初始化寄存器有效 初始话所有的RTC寄存器。 读出状态 0: 正常计数状态或RTC软件复位结束 1: RTC处于复位状态				R/W

### 26.5.2 控制寄存器 1 (RTC\_CR1)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0																																
START	ONEHZSEL	ONEHZOE	Res	AMPM	PRDS[2:0]																																		
位	标记	位名	功能				读写																																
b7	START	RTC计数开始	0: RTC计数停止 1: RTC计数开始				R/W																																
b6	ONEHZSEL	1Hz输出选择	0: 分布式补偿1Hz输出 1: 均匀式补偿1Hz输出 注意: RTC_ERRCRH.COMPEN=1时, 该位设定有效。				R/W																																
b5	ONEHZOE	1Hz输出许可	0: 1Hz输出禁止 1: 1Hz输出许可				R/W																																
b4	Reserved	-	读出时为“0”,写入时写“0”				R/W																																
b3	AMPM	时制选择	0: 12小时时制 1: 24小时时制				R/W																																
周期选择设定:																																							
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <th>PRDS[2]</th><th>PRDS[1]</th><th>PRDS[0]</th><th>周期选择</th></tr> <tr> <td>0</td><td>0</td><td>0</td><td>不选择</td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>每0.5秒周期</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>每1秒周期</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>每1分周期</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>每1时周期</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>每1日周期 (每日00时00分00秒)</td></tr> <tr> <td>1</td><td>1</td><td>X</td><td>每1月周期 (每月1日00时00分00秒)</td></tr> </table>								PRDS[2]	PRDS[1]	PRDS[0]	周期选择	0	0	0	不选择	0	0	1	每0.5秒周期	0	1	0	每1秒周期	0	1	1	每1分周期	1	0	0	每1时周期	1	0	1	每1日周期 (每日00时00分00秒)	1	1	X	每1月周期 (每月1日00时00分00秒)
PRDS[2]	PRDS[1]	PRDS[0]	周期选择																																				
0	0	0	不选择																																				
0	0	1	每0.5秒周期																																				
0	1	0	每1秒周期																																				
0	1	1	每1分周期																																				
1	0	0	每1时周期																																				
1	0	1	每1日周期 (每日00时00分00秒)																																				
1	1	X	每1月周期 (每月1日00时00分00秒)																																				
注意: 在START=1计数过程中写入周期选择时, 为防止误动作请将周期中断 许可关闭。并且在写入后应将相关标志位清除。																																							

### 26.5.3 控制寄存器 2 (RTC\_CR2)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
位	标记	位名	功能	读写			
b7	ALME	ALMIE	闹钟功能使能	0: 闹钟功能禁止 1: 闹钟功能许可			
b6	ALMIE	PRDIE	闹钟中断使能	0: 闹钟中断禁止 1: 闹钟中断许可			
b5	PRDIE	-	周期中断使能	0: 周期中断禁止 1: 周期中断许可			
b4	Reserved	-	读出时为“0”,写入时写“0”	R/W			
b3	ALMF	闹钟标志	0: 闹钟不匹配 1: 闹钟匹配 注意: 在ALME=1时有效, 闹钟匹配时, 一个计数时钟后置“1”。写“0”时清除标志, 写“1”无效。	R/W			
b2	PRDF	周期标志	0: 周期不发生 1: 周期发生 注意: 设定周期发生后, 该位置“1”。写“0”时清除标志, 写“1”无效	R/W			
b1	RWEN	读出/写入允许	0: 读出/写入禁止 1: 读出/写入允许 注意: 日历寄存器读写允许标志。在读出/写入前请确认该位是否为“1”。日历寄存器包括秒, 分, 时, 周, 日, 月, 年计数寄存器。	R/W			
b0	RWREQ	读出/写入请求	0: 正常计数模式 1: 读出/写入请求 注意: 在读出/写入日历寄存器时请将该位置“1”, 请求读写, 由于计数器在连续计数, 请在1秒的时间内完成读出/写入操作并将该位清“0”。	R/W			

## 26.5.4 控制寄存器 3 (RTC\_CR3)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
位 标记 位名 功能 读写							
b7	RCKSEL	Res	LRCEN		Res		
b7	RCKSEL	RTC计数时钟选择		0: 选择外部低速发振器XTAL32时钟 1: 选择其他计数时钟 可选择LRC, 端子输入, XTAL小数分频时钟, 选择这些计数时钟时需要配置PWRC6.RTCCKSEL1[1:0]寄存器位。		R/W	
b6	Reserved	-		读出时为“0”, 写入时写“0”		R/W	
b5	Reserved	-		读出时为“0”, 写入时写“0”		R/W	
b4	LCREN	内部低速发振器使能	0: 内部低速发振器LRC停止 1: 内部低速发振器LRC工作 注意: 低速发振器作为RTC时钟源时, 请设定LCREN位使能。			R/W	
b3~b1	Reserved	-		读出时为“0”, 写入时写“0”		R/W	
b0	Reserved	-		读出时为“0”, 写入时写“0”		R/W	

## 26.5.5 秒计数寄存器 (RTC\_SEC)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
位 标记 位名 功能 读写							
	Res	SECD[2:0]		SECU[3:0]			
b7	Reserved	-		读出时为“0”, 写入时写“0”		R/W	
b6~b4	SECD[2:0]	秒十位		秒十位计数值		R/W	
b3~b0	SECU[3:0]	秒个位		秒个位计数值		R/W	

表示 0-59 秒, 采用十进制计数。请写入十进制 0-59 的 BCD 码, 写入错误值时, 写入值将被忽略。

## 26.5.6 分计数寄存器 (RTC\_MIN)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res		MIND[2:0]			MINU[3:0]		

位	标记	位名	功能	读写
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6~b4	MIND[2:0]	分十位	分十位计数值	R/W
b3~b0	MINU[3:0]	分个位	分个位计数值	R/W

表示 0-59 分，采用十进制计数。请写入十进制 0-59 的 BCD 码，写入错误值时，写入值将被忽略。

## 26.5.7 时计数寄存器 (RTC\_HOUR)

复位值: 0x12

b7	b6	b5	b4	b3	b2	b1	b0
Res		HOURD[1:0]			HOURU[3:0]		

位	标记	位名	功能	读写
b7~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5~b4	HOURD[1:0]	时十位	时十位计数值	R/W
b3~b0	HOURU[3:0]	时个位	时个位计数值	R/W

24 小时时制时，表示 0-23 小时。12 小时时制时，b5=0 表示 AM，则 01~12 表示上午；b5=1 表示 PM，则 21~32 表示下午。

请根据控制位 AMPM 的值，设定正确十进制的 0~23 或者 01~12,21~32 的 BCD 码。写入超出范围的值将被忽略。

具体时间表示参考下表：

24小时时制	AMPM=1 寄存器表示	12小时时制	AMPM=0 寄存器表示
时间		时间	
00时	00H	AM 12时	12H
01时	01H	AM 01时	01H
02时	02H	AM 02时	02H
03时	03H	AM 03时	03H
04时	04H	AM 04时	04H
05时	05H	AM 05时	05H
06时	06H	AM 06时	06H
07时	07H	AM 07时	07H
08时	08H	AM 08时	08H
09时	09H	AM 09时	09H
10时	10H	AM 10时	10H
11时	11H	AM 11时	11H
12时	12H	PM 12时	32H
13时	13H	PM 01时	21H
14时	14H	PM 02时	22H
15时	15H	PM 03时	23H
16时	16H	PM 04时	24H
17时	17H	PM 05时	25H
18时	18H	PM 06时	26H
19时	19H	PM 07时	27H
20时	20H	PM 08时	28H
21时	21H	PM 09时	29H
22时	22H	PM 10时	30H
23时	23H	PM 11时	31H

### 26.5.8 日计数寄存器 (RTC\_DAY)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res		DAYD[1:0]		DAYU[3:0]			

位	标记	位名	功能	读写
b7~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5~b4	HOURD[1:0]	日十位	日十位计数值	R/W
b3~b0	HOURU[3:0]	日个位	日个位计数值	R/W

十进制表示 1~31 日，自动计算闰年和月份。具体表示如下：

月份	日计数表示
2月 (普通年)	01~28
2月 (闰年)	01~29
4、6、9、11月	01~30
1、3、5、7、8、10、12月	01~31

### 26.5.9 周计数寄存器 (RTC\_WEEK)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
		Res			WEEK[2:0]		

位	标记	位名	功能	读写
b7~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~b0	WEEK[2:0]	周	周计数值	R/W

十进制 0~6 表示周日~周六。请写入正确的十进制 0~6 的 BCD 码，写入其他值，将被忽略。周计数值对应关系如下：

周	周计数表示
周日	00H
周一	01H
周二	02H
周三	03H
周四	04H
周五	05H
周六	06H

## 26.5.10月计数寄存器 (RTC\_MON)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0		
	Res			MON[4:0]					

位	标记	位名	功能	读写
b7~b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4~b0	MON[4:0]	月	月计数值	R/W

十进制 1~12 表示 1~12 月。请写入正确的十进制 1~12 的 BCD 码，写入其他值，将被忽略。

## 26.5.11年计数寄存器 (RTC\_YEAR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0		
	YEARD[3:0]			YEARU[3:0]					

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7~b4	YEARD[3:0]	年十位	年十位计数值	R/W
b3~b0	YEARU[3:0]	年个位	年个位计数值	R/W

十进制 0~99 表示 0~99 年。根据月进位计数。自动计算闰年如：00、04、08、...、92、96 等。请写入正确的十进制年计数值，写入错误值将被忽略。

## 26.5.12分闹钟寄存器 (RTC\_ALMMIN)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0		
Res		ALMMIND[2:0]		ALMMINU[3:0]					

位	标记	位名	功能	读写
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6~b4	ALMMIND[3:0]	分闹钟十位	分闹钟十位匹配值	R/W
b3~b0	ALMMINU[3:0]	分闹钟个位	分闹钟个位匹配值	R/W

请设定十进制 0~59 的 BCD 码。

## 26.5.13 时闹钟寄存器 (RTC\_ALMHOUR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res		ALMHOURD[1:0]		ALMHOURU[3:0]			

位	标记	位名	功能	读写
b7~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5~b4	ALMHOURD[1:0]	时闹钟十位	时闹钟十位匹配值	R/W
b3~b0	ALMHOURU[3:0]	时闹钟个位	时闹钟个位匹配值	R/W

请根据时制设定正确的闹钟匹配值。

## 26.5.14 周闹钟寄存器 (RTC\_ALMWEEK)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res		ALMWEEK[6:0]					

位	标记	位名	功能	读写
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6~b0	ALMWEEK[6:0]	周闹钟	b0~b6分别对应周日~周六, 对应为置“1”时, 代表每周该日闹钟有效。如, b0=1, b5=1代表周日和周五闹钟设定有效。	R/W

请根据时制设定正确的闹钟匹配值。

## 26.5.15 时钟误差补偿寄存器 (RTC\_ERRCRH、RTC\_ERRCRL)

复位值: 0x00

### RTC\_ERRCRH

b7	b6	b5	b4	b3	b2	b1	b0
COMPEN	Res						COMP[8]

复位值: 0x20

### RTC\_ERRCRL

b7	b6	b5	b4	b3	b2	b1	b0
COMP[7:0]							

### RTC\_ERRCRH

位	标记	位名	功能	读写
b7	COMPEN	补偿使能	0: 时钟误差补偿无效 1: 时钟误差补偿有效	R/W
B6~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	COMP[8]	补偿值	与COMP[7:0]共同设定补偿值	R/W

### RTC\_ERRCRL

位	标记	位名	功能	读写																																																																																																																																																																			
通过补偿值设定, 可针对每秒进行+/-0.96ppm的精度补偿。补偿值为9位带小数点的2的补码, 后5位为小数部分。可补偿范围-275.5ppm~+212.9ppm。最小分辨率0.96ppm。具体补偿精度请参考下表:																																																																																																																																																																							
b7~b0	COMP[7:0]	补偿值	<table border="1"> <thead> <tr> <th colspan="10">补偿值设定</th> <th>补偿数</th> </tr> <tr> <th>COMPEN</th> <th colspan="9">COMP[8:0]</th> <th></th> </tr> </thead> <tbody> <tr> <td rowspan="16">1</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>-275.5ppm</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>-274.6ppm</td> </tr> <tr> <td>~</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>-30.5ppm</td> </tr> <tr> <td>~</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>-0.96ppm</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0ppm</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>+0.96ppm</td> </tr> <tr> <td>~</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>+30.5ppm</td> </tr> <tr> <td>~</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>+212.0ppm</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>+212.9ppm</td> </tr> <tr> <td>0</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>无效</td> </tr> </tbody> </table>	补偿值设定										补偿数	COMPEN	COMP[8:0]										1	1	0	0	0	0	0	0	0	0	-275.5ppm	1	0	0	0	0	0	0	0	1	-274.6ppm	~	~	~	~	~	~	~	~	~	~	0	0	0	0	0	0	0	0	0	-30.5ppm	~	~	~	~	~	~	~	~	~	~	0	0	0	0	1	1	1	1	1	-0.96ppm	0	0	0	1	0	0	0	0	0	0ppm	0	0	0	1	0	0	0	0	1	+0.96ppm	~	~	~	~	~	~	~	~	~	~	0	0	1	0	0	0	0	0	0	+30.5ppm	~	~	~	~	~	~	~	~	~	~	0	1	1	1	1	1	1	1	0	+212.0ppm	0	1	1	1	1	1	1	1	1	+212.9ppm	0	X	X	X	X	X	X	X	X	无效	R/W
补偿值设定										补偿数																																																																																																																																																													
COMPEN	COMP[8:0]																																																																																																																																																																						
1	1	0	0	0	0	0	0	0	0	-275.5ppm																																																																																																																																																													
	1	0	0	0	0	0	0	0	1	-274.6ppm																																																																																																																																																													
	~	~	~	~	~	~	~	~	~	~																																																																																																																																																													
	0	0	0	0	0	0	0	0	0	-30.5ppm																																																																																																																																																													
	~	~	~	~	~	~	~	~	~	~																																																																																																																																																													
	0	0	0	0	1	1	1	1	1	-0.96ppm																																																																																																																																																													
	0	0	0	1	0	0	0	0	0	0ppm																																																																																																																																																													
	0	0	0	1	0	0	0	0	1	+0.96ppm																																																																																																																																																													
	~	~	~	~	~	~	~	~	~	~																																																																																																																																																													
	0	0	1	0	0	0	0	0	0	+30.5ppm																																																																																																																																																													
	~	~	~	~	~	~	~	~	~	~																																																																																																																																																													
	0	1	1	1	1	1	1	1	0	+212.0ppm																																																																																																																																																													
	0	1	1	1	1	1	1	1	1	+212.9ppm																																																																																																																																																													
	0	X	X	X	X	X	X	X	X	无效																																																																																																																																																													

补偿计算说明:

当默认状态下直接输出 1Hz 时钟, 通过测定该时钟的精度, 计算补偿目标值。

假设实际测定值为 0.9999888Hz, 则:

$$\text{实际发振频率} = 32768 \times 0.9999888 \approx 32767.63$$

$$\begin{aligned}\text{补偿目标值} &= (\text{实际发振频率} - \text{目标频率}) / \text{目标频率} \times 10^6 \\ &= (32767.63 - 32768) / 32768 \times 10^6 \\ &= -11.29\text{ppm}\end{aligned}$$

设定值计算：

$$\text{COMP[8:0]} = \left( \frac{\text{补偿目标值[ppm]} \times 2^{15}}{10^6} \right)_{\text{取2的补码}} + 0001.00000B$$

如果补偿目标值为+20.3ppm，计算相应的寄存器值如下：

$$\begin{aligned}\text{COMP[8:0]} &= (20.3 \times 2^{15} / 10^6)_{\text{取2的补码}} + 0001.00000B \\ &= (0.6651904)_{\text{取2的补码}} + 0001.00000B \\ &= 0000.10101B + 0001.00000B \\ &= 0001.10101B\end{aligned}$$

如果补偿目标值为-20.3ppm，计算相应的寄存器值如下：

$$\begin{aligned}\text{COMP[8:0]} &= (-20.3 \times 2^{15} / 10^6)_{\text{取2的补码}} + 0001.00000B \\ &= (-0.6651904)_{\text{取2的补码}} + 0001.00000B \\ &= 1111.01011B + 0001.00000B \\ &= 0000.01011B\end{aligned}$$

## 27 看门狗计数器 (WDT/ SWDT)

### 27.1 简介

看门狗计数器有两个，一种是计数时钟源为专用内部 RC (SWDTLRC:10KHz) 的专用看门狗计数器 (SWDT)，另一种是计数时钟源为 PCLK3 的通用看门狗计数器 (WDT)。专用看门狗和通用看门狗是 16 位递减计数器，用来监测由于外部干扰或不可预见的逻辑条件造成应用程序背离正常的运行而产生的软件故障。

两个看门狗都支持窗口功能。在计数开始前可预设窗口区间，计数值位于窗口区间时，可刷新计数器，计数重新开始。基本特性如表 27-1。

表 27-1 看门狗计数器的基本特性

计数时钟	SWDT: SWDTLRC的1/16/32/64/128/256/2048分频 WDT: PCLK3的4/64/128/256/512/1024/2048/8192分频
最长溢出时间	SWDT:3.72hour(max)      WDT:10.7s (PCLK3=50MHz)
计数模式	递减计数
窗口功能	可设定窗口区间，定义刷新动作的允许区间
启动方式	1) 硬件启动 2) 软件启动
停止条件	1) 复位中 2) 下溢，刷新错误发生时 再开始：硬件启动模式下，复位或中断请求输出后自动开始 软件启动模式下，再次设定刷新寄存器
中断/复位条件	1) 计数下溢 2) 刷新错误

## 27.2 功能说明

### 27.2.1 启动看门狗

看门狗计数器的启动方式有两种：硬件启动方式和软件启动方式。

硬件启动方式是指启动时从主闪存区域读取看门狗计数器的设定信息（ICG0 寄存器），计数器自动开始计数；软件启动方式是指设定控制寄存器后，写刷新寄存器完成刷新动作，计数器开始计数。

### 27.2.2 硬件启动方式

ICG0 寄存器的位 16(WDTAUTS)、位 0(SWDTAUTS)为 0 时，为硬件启动方式。选择硬件启动方式时，WDT\_CR 和 SWDT\_CR 寄存器的相关设定信息无效。

硬件启动方式时，在复位期间将 ICG0 寄存器里的 WDT/SWDT 相关设定（计数时钟、窗口设定值、计数周期等）载入到 WDT/SWDT 的模块中，复位之后、计数器按照设定自动开始计数。图 27-1 为硬件启动方式的动作例。

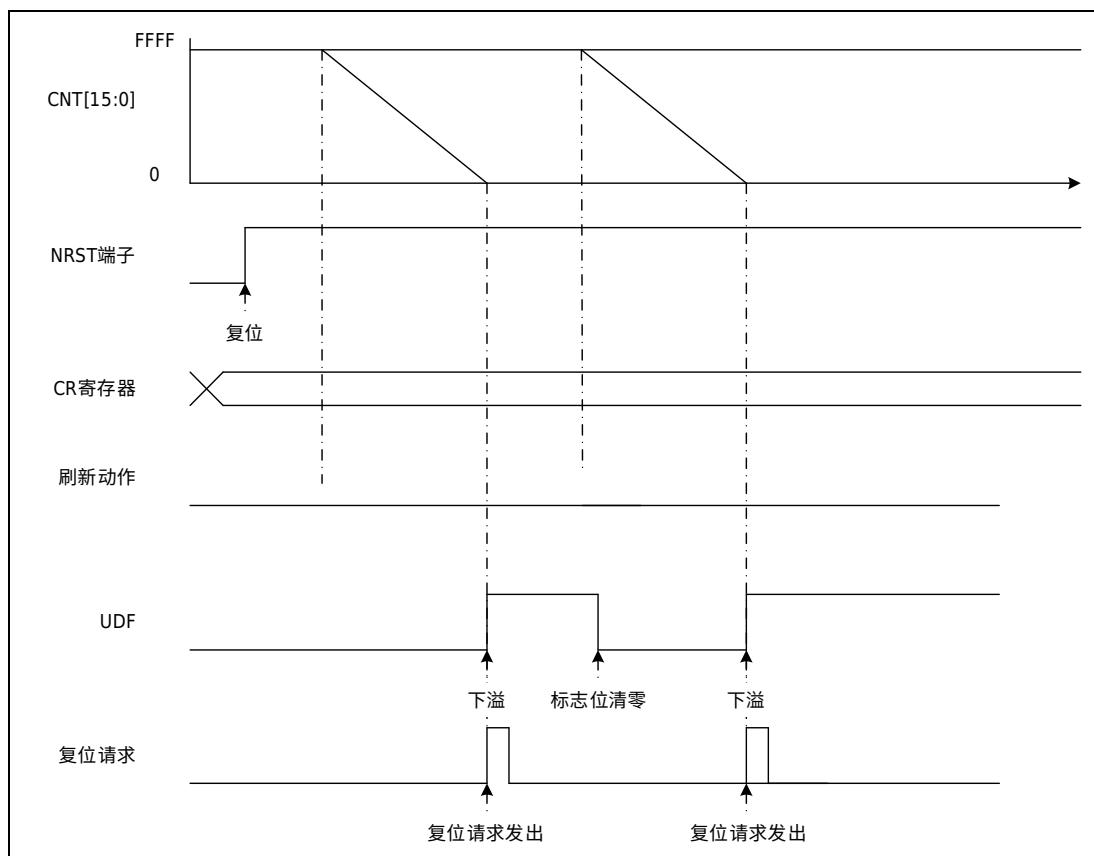


图 27-1 硬件启动例

### 27.2.3 软件启动方式

ICG0 寄存器的位 16(WDTAUTS) , 位 0(SWDTAUTS)为 1 时, 通过设定刷新寄存器的方式启动 WDT/SWDT 为软件启动方式。复位后, 设定 WDT\_CR/SWDT\_CR 寄存器中的计数时钟、窗口设定值、计数周期等, 然后执行刷新动作, 计数器就开始计数。WDT\_CR/SWDT\_CR 设定只能 1 次, 再次设定写入值无效。图 27-2 为软件启动方式的动作例。

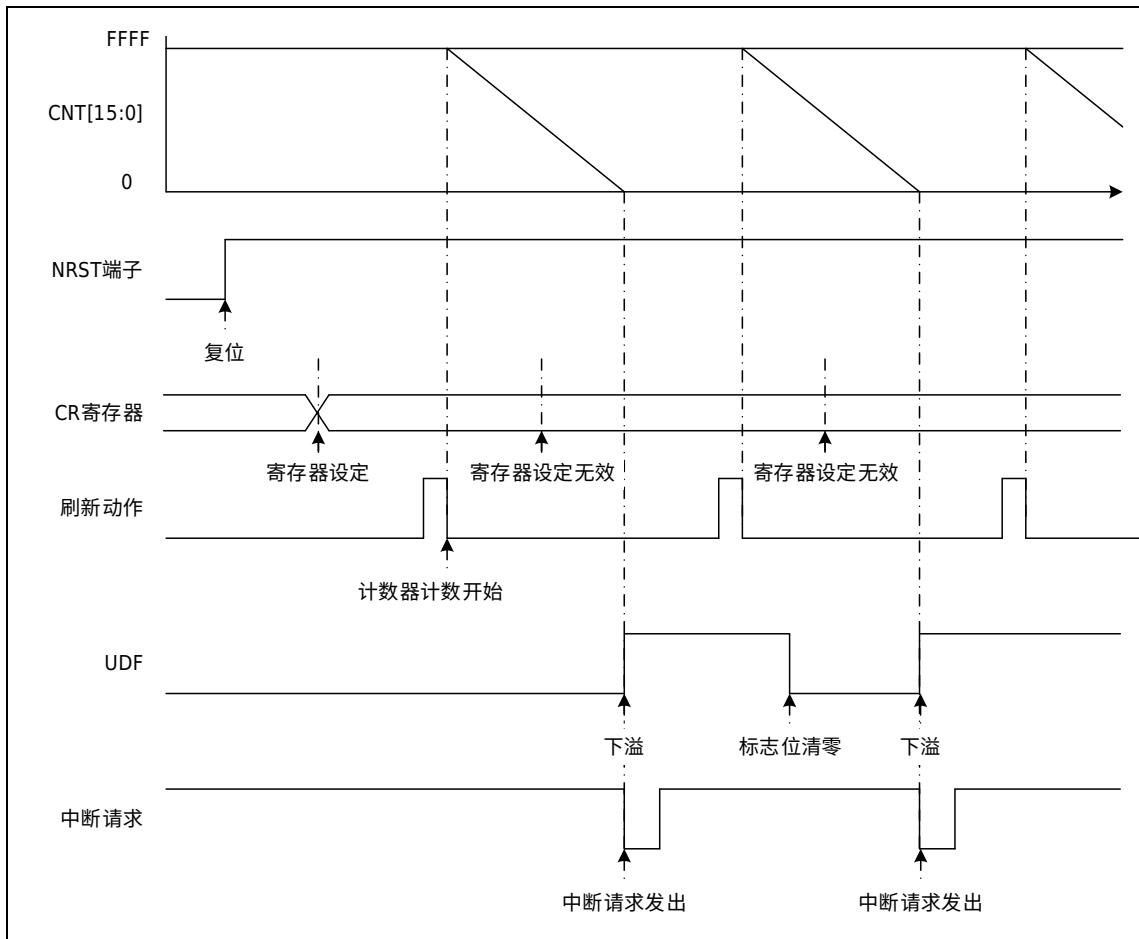


图 27-2 软件启动例

## 27.2.4 刷新动作

(S)WDT\_RR 寄存器中先写 0x0123、再写 0x3210 完成一次刷新动作，WDT/SWDT 的计数器就开始计数(软件启动)或重新开始计数。

(S)WDT\_RR 寄存器在写 0x0123、0x3210 之间，若对发生对其他寄存器访问或读取(S)WDT\_RR 寄存器等，不影响正常的刷新动作。

如图 27-3 所示动作示例。

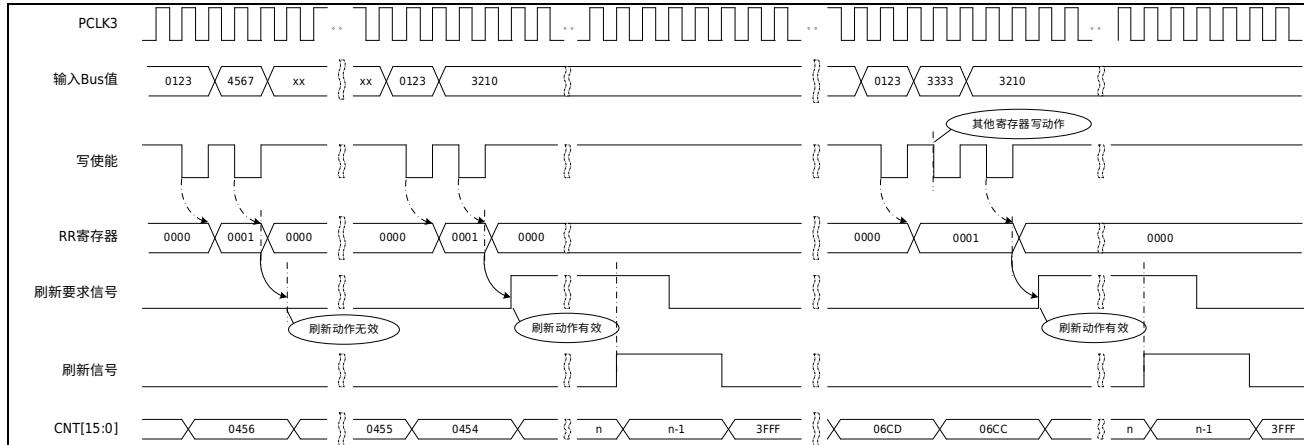


图 27-3 各种刷新动作时序示例 (动作确认，刷新要求信号的下降沿等)

刷新动作需要 4 个计数周期完成计数值的更新，所以请在刷新下位窗口和下溢位置的提前 4 个计数值完成刷新寄存器的写入。计数值的确认请读取状态寄存器。

## 27.2.5 标志位

刷新错误标志位和计数下溢标志位在中断请求的情况下会保持。当进入中断后，可以通过查询标志位来确认中断原因。标志位清零：先读“1”再写“0”。

刷新错误或者计数下溢标志位置位时，硬件启动模式看门狗计数不停止；软件启动模式看门狗计数停止。对标志位写“0”时，SWDT 最多需要经过 3 个 SWDTLRC 和 2 个 PCLK3 时间后，寄存器位才能被清零；WDT 最多需要经过 5 个 PCLK3 时间后，寄存器位才能被清零。另，在发生刷新错误或者下溢错误的一定时间内，对标志位读“1”清零无效，这段时间为：1 个计数周期+2 个 SWDTLRC(SWDT 模块)；1 个计数周期+2 个 PCLK3 (WDT 模块)。

## 27.2.6 中断复位

WDT/SWDT 在计数器计数下溢或刷新错误时，可以选择产生中断请求或复位请求。硬件启动模式下，通过 ICG0 的 WDTITS/SWDTITS 位，来决定产生中断请求还是复位请求。软件启动模式下，通过设定 WDT\_CR/SWDT\_CR 寄存器 ITS 位，来决定产生中断请求还是复位请求。

WDT/SWDT 的中断复位产生条件有两种。一种是计数器计数产生下溢；一种是在刷新允许区间之外执行刷新动作，产生刷新错误。

### 27.2.7 计数下溢

如图 27-4 例，在递减计数到零时产生下溢。

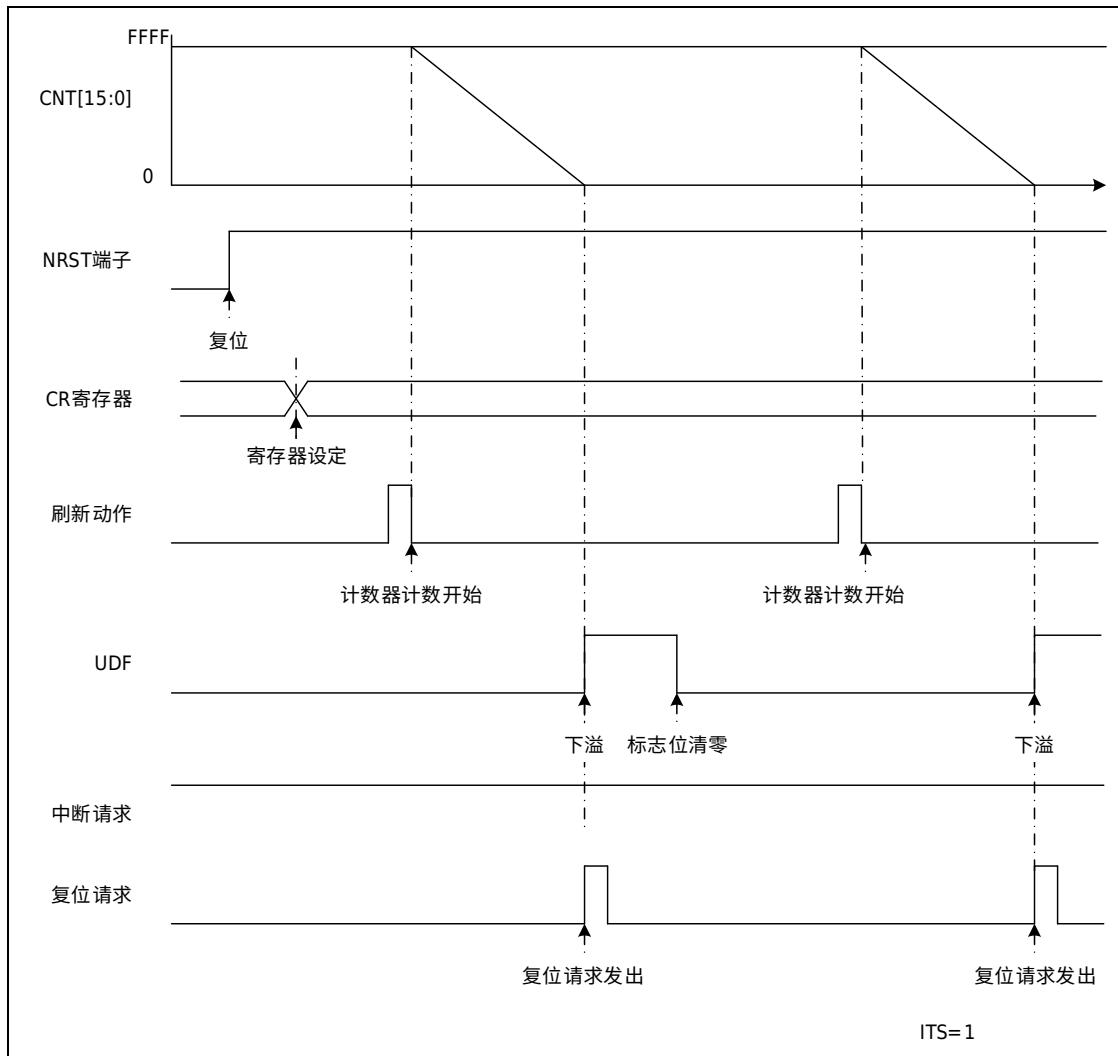


图 27-4 计数器下溢动作例

### 27.2.8 刷新错误

在设定了窗口区间后，只有在窗口区间内执行刷新动作时计数器才会被刷新、重新开始计数，在窗口区间外执行刷新动作时产生刷新错误。图 27-5 为刷新动作例。

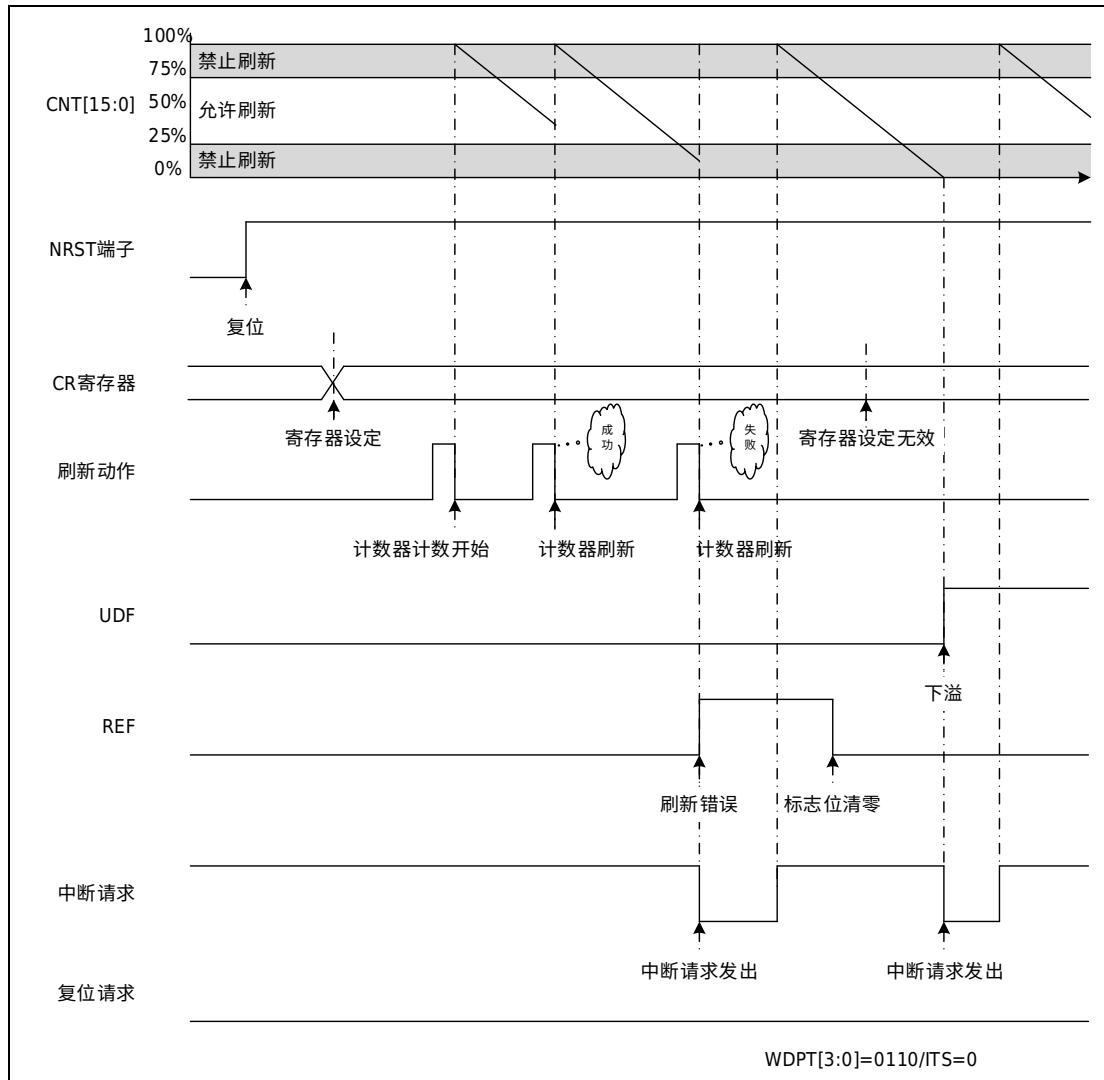


图 27-5 计数器刷新动作例

## 27.3 寄存器说明

表 27-2 所示，为 WDT 和 SWDT 模块的寄存器列表。

表 27-2 寄存器列表

WDT\_BASE\_ADDR: 0x40049000

SWDT\_BASE\_ADDR: 0x40049400

寄存器名	符号	偏移地址	位宽	复位值
SWDT控制寄存器	SWDT_CR	0x00	32	0x8001_0FF3
SWDT状态寄存器	SWDT_SR	0x04	32	0x0000_0000
SWDT刷新寄存器	SWDT_RR	0x08	32	0x0000_0000
WDT控制寄存器	WDT_CR	0x00	32	0x8001_0FF3
WDT状态寄存器	WDT_SR	0x04	32	0x0000_0000
WDT刷新寄存器	WDT_RR	0x08	32	0x0000_0000

### 27.3.1 控制寄存器 (SWDT\_CR、WDT\_CR)

复位值: 0x8001\_0FF3

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ITS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SLPOFF
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	WDPT		CKS[3:0]		-	-	PERI[1:0]					

位	标记	位名	功能	读写
b31	ITS	刷新错误/溢出 中断/ 复位选择	0: 中断请求 1: 复位请求	R/W
b30~b17	Reserved	-	读出时为“0”,写入时写“0”	R/W
b16	SLPOFF	(S)WDT在低功耗模式 下计数禁止	WDT_CR: 0: WDT在sleep模式下计数许可 1: WDT在sleep模式下计数禁止 SWDT_CR: 0: SWDT在sleep/stop模式下计数许可 1: SWDT在sleep/stop模式下计数禁止	R/W
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11~b8	WDPT[3:0]	刷新允许区域计数值 百分比	0000: 0%~100% 0001: 0%~25% 0010: 25%~50% 0011: 0%~50% 0100: 50%~75% 0101: 0%~25%,50%~75% 0110: 25%~75% 0111: 0%~75% 1000: 75%~100% 1001: 0%~25%,75%~100% 1010: 25%~50%,75%~100% 1011: 0%~50%,75%~100% 1100: 50%~100% 1101: 0%~25%,50%~100% 1110: 25%~100% 1111: 0%~100%	R/W
b7~b4	CKS[3:0]	计数时钟	WDT_CR: 0010: PCLK3/4 0110: PCLK3/64 0111: PCLK3/128 1000: PCLK3/256 1001: PCLK3/512 1010: PCLK3/1024 1011: PCLK3/2048 1101: PCLK3/8192 其余值: 预留功能 SWDT_CR:	R/W

0000: SWDTCLK  
0100: SWDTCLK/16  
0101: SWDTCLK/32  
0110: SWDTCLK/64  
0111: SWDTCLK/128  
1000: SWDTCLK/256  
1011: SWDTCLK/2048  
其它值: 预留

b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
			00: 256 cycle	
b1~b0	PERI[1:0]	计数周期	01: 4096 cycle 10: 16384 cycle 11: 65536 cycle	R/W

### 27.3.2 状态寄存器 (SWDT\_SR、WDT\_SR)

复位值：0x0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	REF	UDF
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT[15:0]															

位	标记	位名	功能	读写
b31~b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17	REF	刷新错误标志	0: 没有刷新错误 1: 发生刷新错误 对该位读出1后写入0, 该位清零。	R/W
b16	UDF	计数下溢标志	0: 没有计数下溢 1: 发生计数下溢 对该位读出1后写入0, 该位清零。	R/W
b15~b0	CNT[15:0]	计数值	计数器当前计数值	R

### 27.3.3 刷新寄存器 (SWDT\_RR、WDT\_RR)

复位值：0x0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RF[15:0]															

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b0	RF[15:0]	刷新值	依次写入0x0123、0x3210之后, 完成刷新动作。 当寄存器写入0x0123h后, 读出寄存器为0x00000001;其余情况读出值都是0x00000000。	R/W

## 27.4 使用注意事项

SWDT 动作时，周边时钟 PCLK3 的动作频率必须大于或等于计数时钟频率的 4 倍，即 PCLK3 频率 $\geq$ 计数时钟频率  $\times 4$ 。

## 28 通用同步异步收发器 (USART)

### 28.1 简介

本产品搭载通用串行收发器模块 (USART) 6 个单元。通用串行收发器模块 (USART) 能够灵活地与外部设备进行全双工数据交换；本 USART 支持通用异步串行通信接口 (UART)，时钟同步通信接口，智能卡接口 (ISO/IEC7816-3) 和 LIN 通信接口。支持调制解调器操作 (CTS/RTS 操作)，多处理器操作。与 Timer0 模块配合支持 UART 接收 TIMEOUT 功能。USART\_1 支持通过 RX 线唤醒 STOP 模式功能。

具体功能分配如下：

- UART：全通道支持
- 多处理器通信：全通道支持
- 时钟同步通信：全通道支持
- RX 线唤醒 STOP 模式功能：USART\_1 支持
- 小数波特率：全通道支持
- LIN：USART\_3, USART\_6 支持
- 智能卡：USART\_1, USART\_2, USART\_4, USART\_5 支持
- UART 接收超时功能：USART\_1, USART\_2, USART\_4, USART\_5 支持

#### USART 主要特性：

- 支持全双工异步通信，全双工时钟同步通信
- 支持 LIN 总线
- 支持智能卡接口 (ISO/IEC7816-3)
- 发送器和接收器具有独立使能位
- 内置双缓冲器
- LSB/MSB 可选
- 支持调制解调器操作 (CTS/RTS)
- 传输标志：发送数据寄存器空，发送数据完成，接收数据寄存器满，接收错误标志，UART 接收超时标志，LIN 唤醒信号检出标志，LIN 间隔段检出标志，LIN 总线错误标志

#### UART 主要特性：

- 数据长度可编程:8 位/9 位
- 校验功能可配置：奇校验/偶校验/无校验
- 停止位可配置：1 位/2 位
- 时钟源可选：内部时钟源(内部波特率生成器生成的时钟)/外部时钟源( USARTn\_CK 管脚输入的时钟)
- 接收错误：校验错误，帧错误，上溢错误

- 支持多个处理器间通信
- 内置数字滤波器
- 支持接收数据 TIMEOUT 功能
- 单元 1 支持停止模式唤醒功能
- 支持全双工/半双工通信方式

**时钟同步模式主要特性：**

- 数据长度：8 位
- 接收错误：上溢错误
- 时钟源：内部时钟源(内部波特率生成器生成的时钟)/外部时钟源( USARTn\_CK 管脚输入的时钟)
- 支持全双工通信方式

**智能卡接口的主要特性：**

- 数据长度：8 位
- 检测到校验错误时能自动送出错误信号
- 支持数据重发

**LIN 的主要特性：**

- 数据长度：8 位
- 支持唤醒信号的检测
- 支持 10/11 位同步间隔段（BF）的检测
- 支持同步段测量，寄存器记录测量值
- 支持 10/11/13/14 位的间隔段（BF）发送
- 支持总线冲突检测
- 支持回环模式

## 28.2 USART 系统框图

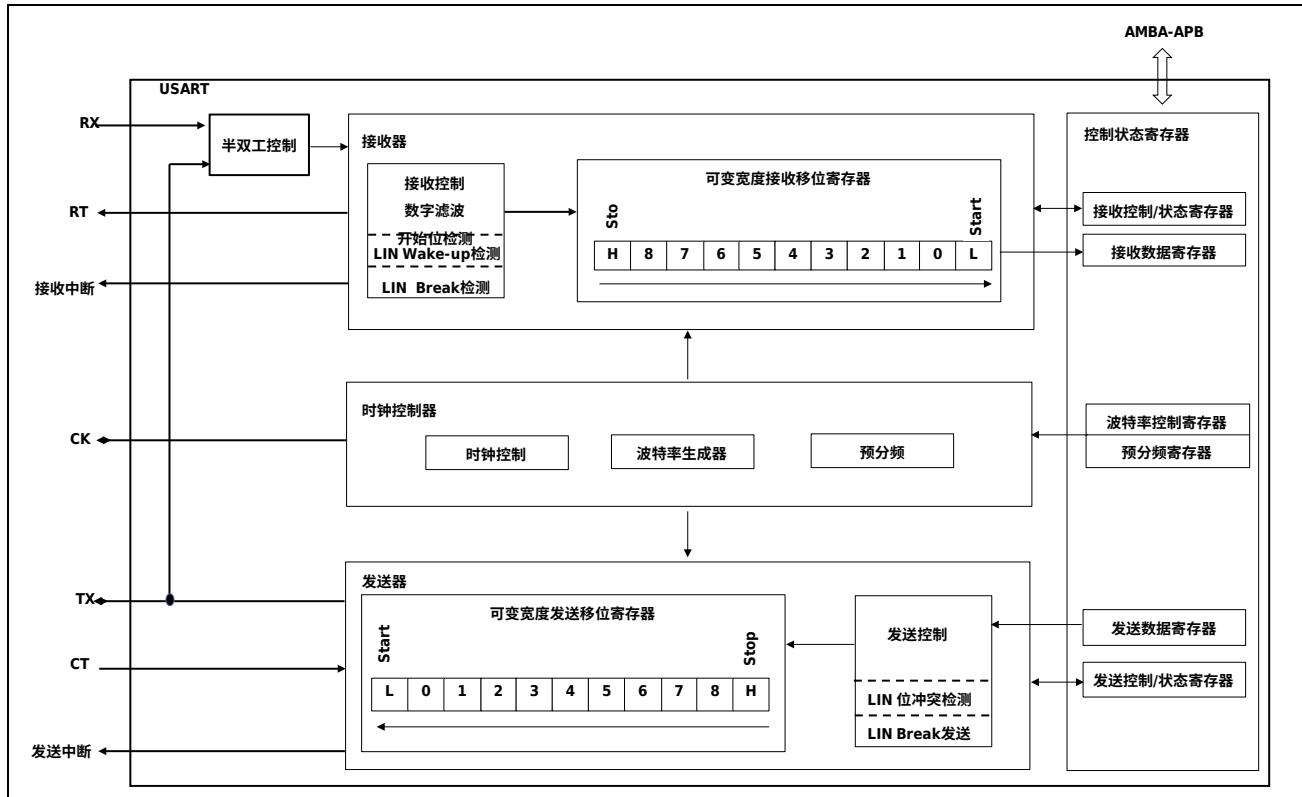


图 28-1 USART 系统框图

## 28.3 管脚说明

表 28-1 USART 管脚说明

管脚名	方向	功能描述
USARTn_CK	输入输出	时钟
USARTn_TX	输入输出	发送数据管脚 半双工时也作为接收数据管脚
USARTn_RX	输入	接收数据管脚
USARTn_CTS	输入	调制解调器操作管脚 清除发送管脚
USARTn_RTS	输出	调制解调器操作管脚 请求发送管脚

n:1~6

## 28.4 功能说明

本章将对 UART，多处理器通信，智能卡，时钟同步模式和 LIN 的功能详细说明。

### 28.4.1 UART

#### 28.4.1.1 时钟

UART 可以选择内部波特率生成器生成的时钟(内部时钟源)或 USARTn\_CK 管脚输入的时钟(外部时钟源)作为通信的时钟源。

##### 内部时钟源

USARTn\_CR2.CLKC[1:0]位设定为 00b 或者 01b 时选择时钟源为内部时钟源即内部波特率生成器生成的时钟。

USARTn\_CR2.CLKC[1:0]=00b 时 USARTn\_CK 管脚不作为时钟管脚使用，可以作为普通 IO 使用。

USARTn\_CR2.CLKC[1:0]=01b 时从 USARTn\_CK 管脚输出与通信波特率相同频率的时钟。

内部波特率生成器的时钟源由 USARTn\_PR.PSC[1:0]位的设定选择为 PCLK, PCLK/4, PCLK/16, PCLK/64。

##### 外部时钟源

USARTn\_CR2.CLKC[1:0]位设定为 10b 或者 11b 时选择时钟源为从 USARTn\_CK 管脚输入的外部时钟，输入时钟的频率为波特率的 16 倍(USARTn\_CR1.OVER8=0)或者 8 倍(USARTn\_CR1.OVER8=1)。

##### 最高波特率

内部时钟源时，内部波特率生成器生成的波特率计算公式为：

$$B = \frac{C}{8 \times (2 - OVER8) \times (DIV\_Integer + 1)}$$

B : 波特率 单位：Mbps

C : USARTn\_PR.PSC[1:0]位设定的时钟 (PCLK,PCLK/4,PCLK/16,PCLK/64) 单位：MHz

OVER8 : USARTn\_CR1.OVER8 设定值

DIV\_Integer : USARTn\_BRR.DIV\_Integer 设定值

最高波特率为 PCLK/8(Mbps)。

外部时钟源时，外部输入 UART 时钟的最高频率要求为 PCLK(MHz)/4，所以时钟源为外部输入时钟时最高波特率为 PCLK/64(Mbps) (USARTn\_CR1.OVER8=0 时) 或者 PCLK/32 (Mbps) (USARTn\_CR1.OVER8=1 时)。

需要注意的是，UART 最高通信波特除了以上描述的基于 PCLK 的计算方法外，还需要参考电气特性章节规定的最高通信波特率。

#### 28.4.1.2 数据格式

UART 模式时一帧数据是由开始位，数据位，校验位和停止位组成。

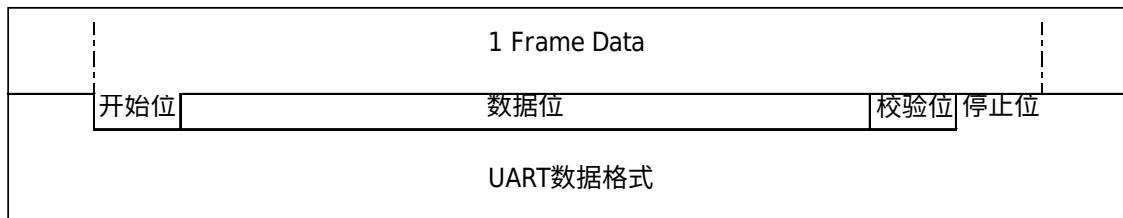


图 28-2 UART 数据格式

##### 开始位

开始位固定有一位的低电平构成。

##### 数据位

数据位可以配置成 8 位或者 9 位。

##### 校验位

校验位可以配置成 1 位偶校验或 1 位奇校验或无校验位。

##### 停止位

停止位固定为高电平，可以配置成 1 位或者 2 位。

#### 28.4.1.3 调制解调器操作

调制解调器操作包括 CTS 功能和 RTS 功能。USARTn\_CR3.RTSE=1 时 RTS 功能有效，USARTn\_CR3.CTSE=1 时 CTS 功能有效。

##### CTS 功能

CTS 功能是通过 USARTn\_CTS 管脚的输入来控制数据的发送，只有当 USARTn\_CTS 管脚输入低电平时才可以发送数据，发送数据过程中如果 USARTn\_CTS 输入高电平，正在发送的数据不受影响。

##### RTS 功能

RTS 功能是指通过 USARTn\_RTS 管脚输出低电平，请求对方发送数据。

USARTn\_RTS 管脚输出低电平需要满足以下全部条件：

- 接收使能(USARTn\_CR1.RE=1)，且不正在接收数据
- USARTn\_TDR.RDR 寄存器中没有未读取的接收数据
- 无任何接收错误，包括帧错误，校验错误和上溢错误

#### 28.4.1.4 发送器

发送器可发送 8 位或 9 位的数据，具体取决于 USARTn\_CR1.M 位的设定值。

发送器使能位 (USARTn\_CR1.TE) 置 1，写入发送数据后，发送数据在 TX 管脚上串行输出；相应的时钟脉冲可以选择在 USARTn\_CK 管脚输出或者不输出。

发送数据的顺序为：开始位->数据位(MSB/LSB)->校验位(有或者无)->停止位。

发送数据寄存器 TDR 和内部的发送移位寄存器组成双缓冲器结构，可以连续发送数据。

通过发送数据寄存器空中断或者 DMA 写入发送数据时，为保证发送的正确性，一次请求只能写入一次数据。

#### 发送数据设定步骤

1. 将 USARTn\_CR1 寄存器设定为复位值
2. 设定 UART 所需要使用的管脚
3. 通过 USARTn\_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn\_CR1, USARTn\_CR2, USARTn\_CR3 寄存器
5. 设定 USARTn\_PR 选择预分频值，USARTn\_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. 使能发送器 (USARTn\_CR1. TE=1)，如果需要使用发送数据寄存器空中断，则设置 USARTn\_CR1. TXEIE=1
7. 等待发送数据寄存器空，写通信数据到 USARTn\_TDR.TDR，数据传输到发送移位寄存器，发送开始  
(CTS 功能有效时，USARTn\_CTS 输入为低电平时数据传输到发送移位寄存器，发送开始)
8. 如果需要连续发送数据时，重复步骤 7
9. 通过确认 USARTn\_SR.TC 位确认发送是否完成。连续发送数据并使用发送中断的情况，可通过 TI 中断写入最后一个发送数据，并将 USARTn\_CR1. TXEIE 写 0, USARTn\_CR1. TCIE 写 1，最后一帧数据发送结束后，产生发送完成中断

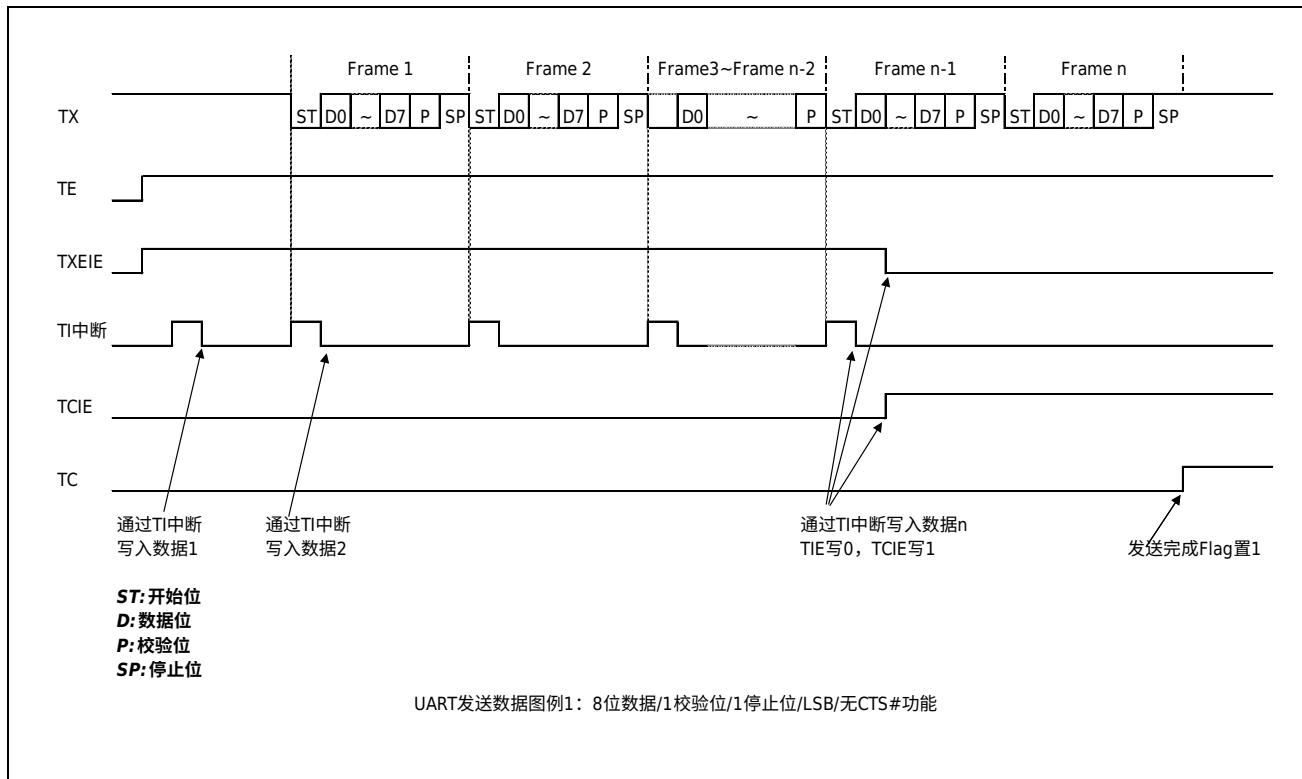


图 28-3 UART 发送数据图例 1

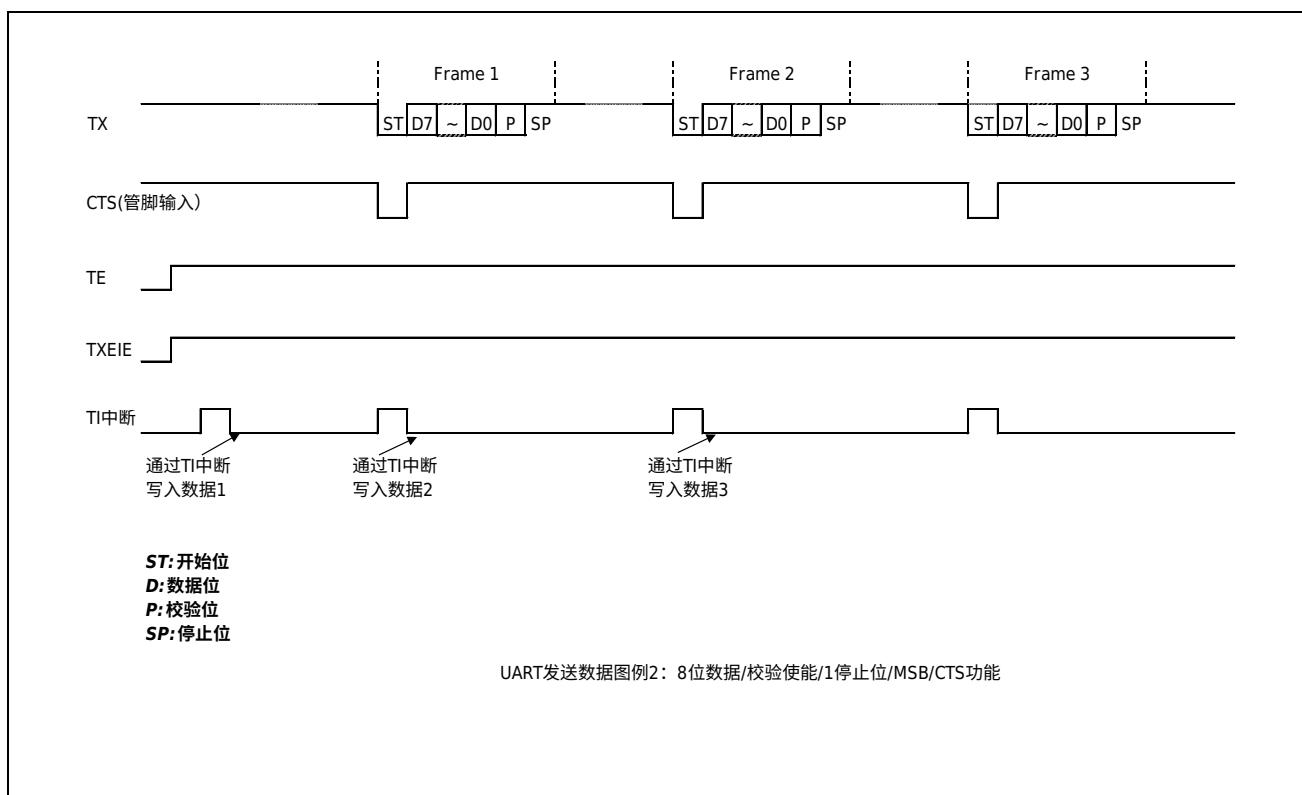


图 28-4 UART 发送数据图例 2

## 发送器中断

UART 模式发送器支持两种中断，发送数据寄存器空中断 TI 和发送完成中断 TCI。

TXEIE=1， USARTn\_TDR.TDR 寄存器的值传送到发送移位寄存器时 TI 中断发生。

TCIE=1，发送数据的最后一位时 USARTn\_TDR.TDR 寄存器没有更新则 TCI 中断发生。

### 28.4.1.5 接收器

接收器可接收 8 位或 9 位的数据，具体取决于 USARTn\_CR1.M 位的设定值。接收器使能位 (USARTn\_CR.RE) 置 1 并检测到开始位后，RX 管脚上数据接收到接收移位寄存器，收满一帧数据，数据从接收移位寄存器传送到接收数据寄存器 USARTn\_RDR.RDR。

接收数据的顺序为：开始位->数据位(MSB/LSB)->校验位(有或者无)->停止位。

接收数据寄存器 USARTn\_RDR.RDR 寄存器和内部的接收移位寄存器组成双缓冲器结构，可以连续接收数据。

通过接收数据寄存器满中断或者 DMA 读取接收数据时，一次请求只能读取一次数据。

## 开始位检测

开始位检测可以选择低电平方式或下降沿方式，具体取决于 USARTn\_CR1.SBS 位，USARTn\_CR1.SBS=0 时为低电平检测，USARTn\_CR1.SBS=1 时为下降沿检测。

## 采样和接收容差

检测到开始条件（低电平或下降沿）后，USART 会基于内部基本时钟对接收数据进行时钟同步，从而开始数据接收。

数据的采样在数据中央，USARTn\_CR1.OVER8=0 时在第 8 个内部基本时钟采样，USARTn\_MR.OVER8=1 时在第 4 个内部基本时钟采样。

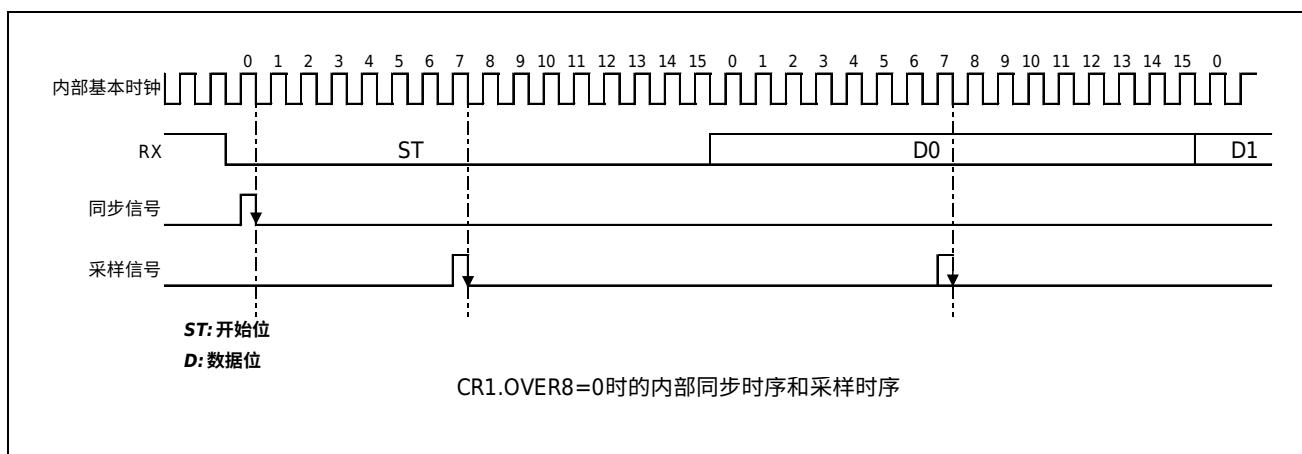


图 28-5 UART 内部同步和采样时序

仅当总时钟系统偏差小于 UART 接收器的容差时，UART 异步接收器才能正常工作。影响总偏差的因素包括：

- 发送器误差引起的偏差（其中还包括发送器本地振荡器的偏差）
- 接收器的波特率量化引起的误差
- 接收器本地振荡器的偏差
- 传输线路引起的偏差

对于正确接收数据，UART 异步接收器所容许的最大偏差值具体取决于以下选项：

- 数据长度 FL。FL 由 USART\_CR1 寄存器中 M 位定义的 8 或 9 数据位和 PCE 位定义的校验使能位决定
- 由 USART\_CR1 寄存器中 OVER8 位定义的 8 倍或 16 倍过采样
- 由 USART\_CR1 寄存器中 FBME 位定义的是否使用小数波特率

表 28-2 DIV\_Fraction 为 0 时 UART 接收器的容差

FL	OVER8位=0	OVER8位=1
10	4.375%	3.75%
11	3.97%	3.41%
12	3.646%	3.125%

表 28-3 DIV\_Fraction 不为 0 时 UART 接收器的容差

FL	OVER8位=0	OVER8位=1
10	3.88%	3%
11	3.53%	2.73%
12	3.23%	2.5%

在特殊情况下，当 RX 管脚接收到数据长度 FL 时间的连续高电平时，表 29-2 和表 29-3 中指定的数据可能会略微不同。

## 接收数据设定步骤

1. 将 USARTn\_CR1 寄存器设定为复位值
2. 设定 UART 所需要使用的管脚
3. 通过 USARTn\_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn\_CR1, USARTn\_CR2, USARTn\_CR3 寄存器
5. 设定 USARTn\_PR 选择预分频值, USARTn\_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. 使能接收器 (USARTn\_CR1. RE=1), 如果需要使用接收中断, 则设置 USARTn\_CR1. RIE=1
7. 当检测到开始位后, 接收器将数据接收到接收移位寄存器, 并检查校验位和停止位
  - 校验错误时, 接收到的数据传送到 USARTn\_RDR.RDR 寄存器中并置位 USARTn\_SR.PE 标志
  - 停止位不为高电平时, 发生帧错误, 接收到的数据传送到 USARTn\_RDR.RDR 寄存器中并置位 USARTn\_SR.FE 标志
  - 发生上溢错误时, 数据丢失并置位 USARTn\_SR.ORE 标志
  - 无错误发生时, 接收到的数据传送到 USARTn\_RDR.RDR 寄存器中, 并置位 USARTn\_SR.RXNE 标志, 读取接收到的数据后重复步骤 7 可连续接收数据

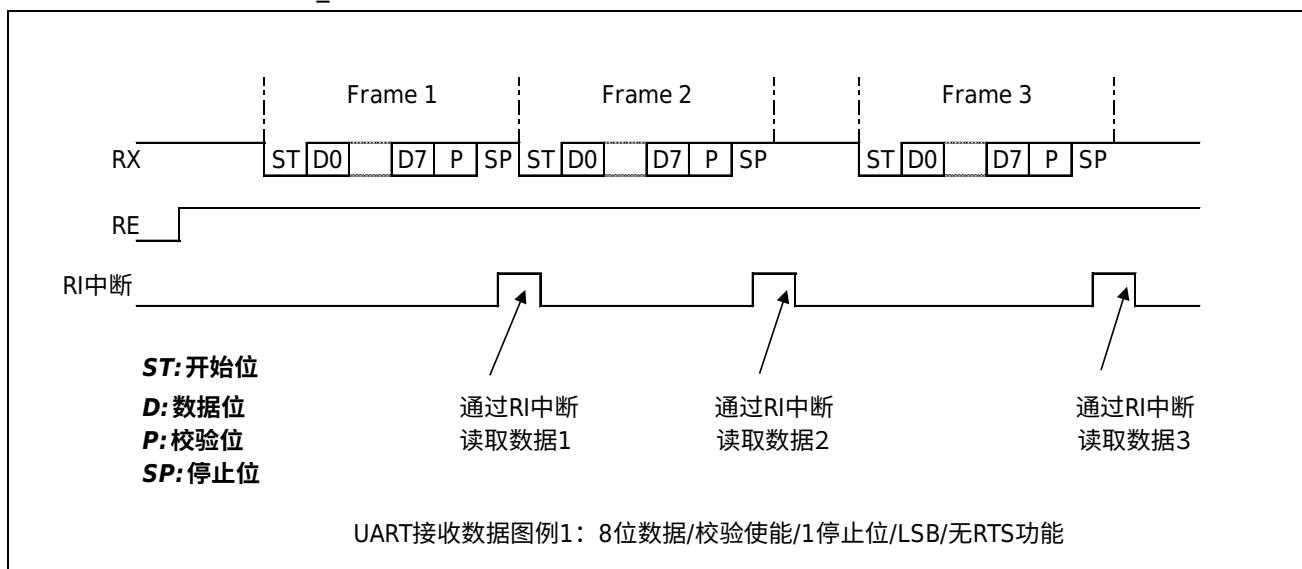


图 28-6 UART 接收数据图例 1

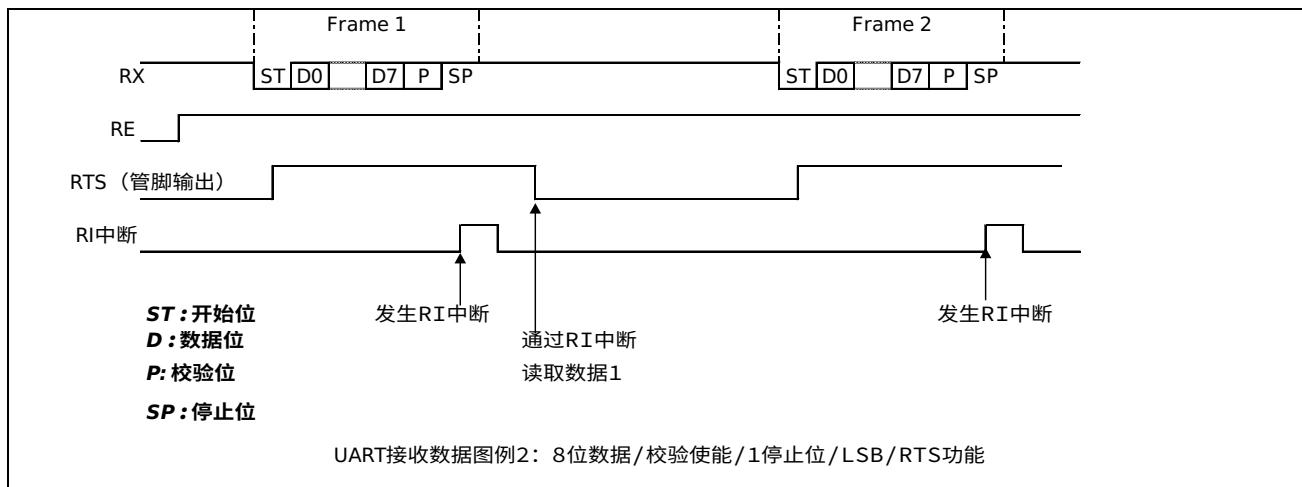


图 28-7 UART 接收数据图例 2

### 错误处理

接收数据时有三种类型的接收错误，分别为上溢错误(USARTn\_SR.ORE)，校验错误(USARTn\_SR.PE)和帧错误(USARTn\_SR.FE)。发生任何一种接收错误都不能再进行数据的接收。可以通过将所有的错误标志清零来重启数据接收，清零方法是写对应清零寄存器。

上溢错误发生的条件是 USARTn\_RDR.RDR 寄存器值未被读取的情况下又接收到一帧新的数据，所以应该在收到当前帧最后一位之前将接收到的前一帧数据读取。

校验错误发生的条件是发生了奇偶校验错误。

帧错误发生的条件是停止位为低电平，2个停止位的情况也只检查第一个停止位。

发生上溢错误时接收到的数据丢失，RI 中断不发生。

发生校验错误时接收到的数据传送给 USARTn\_RDR.RDR，RI 中断不发生。

发生帧错误时接收到的数据传送给 USARTn\_RDR.RDR，RI 中断不发生。

### 接收器中断

UART 模式接收器支持两种中断，接收数据寄存器满中断 RI 和接收错误中断 EI。

USARTn\_CR.RIE=1，未发生任何接收错误，数据从接收移位寄存器传送到接收数据寄存器时 RI 中断发生。

USARTn\_CR.RIE=1，接收过程中发生上溢错误，校验错误或者帧错误时 EI 中断发生。

#### 28.4.1.6 UART 接收 TIMEOUT 功能

UART 接收数据停止位被检测时 TIMEOUT 计数器启动，经过设定的 TIMEOUT 时间（设定单位为接收位数）后未检测到下一帧接收数据时，发生 TIMEOUT，如果此时 CR1.RE=1，则 TIMEOUT 状态位 USARTn\_SR.RTOF 置位，如果此时 USARTn\_CR1.RE=0，则等待 USARTn\_CR1.RE=1 后 TIMEOUT 状态位 USARTn\_SR.RTOF 置位。

TIMEOUT 计数器采用 Timer0 模块的计数器，具体对应关系如下：

USART\_1: Timer0 Unit1 A 通道

USART\_2: Timer0 Unit1 B 通道

USART\_4: Timer0 Unit2 A 通道

USART\_5: Timer0 Unit2 B 通道

#### TIMEOUT 功能 Timer0 比较计数器值设定

Timer0 为 16 位计数器，计数时钟最大可以选择 1024 分频，TMR0\_CMPA<B>R 值设定计算公式如下：

$$\text{CMPA } < \text{B} > \text{R} = \frac{\text{RTB}}{2^{\text{CKDIVA } < \text{B} >}} - \alpha$$

CMPA<B>R:TMR0\_CMPA<B>R 寄存器值，计算结果请向上进位取整。

$\alpha$ : Timer0 异步计数同步电路带来的迟延

计数时钟不分频时， $\alpha = 7$

计数时钟 2 分频时， $\alpha = 5$

计数时钟 4、8、16 分频时， $\alpha = 3$

计数时钟 32 分频及以上， $\alpha = 2$

RTB: Receive Timeout Bits，最小值 = 接收数据帧长度 +  $\alpha \times 2^{\text{CKDIVA } < \text{B} >}$

实际发生 TIMEOUT 的时间与 RTB 值之间有一定误差，误差  $\leq 2^{\text{CKDIVA } < \text{B} >}$ 。

CKDIRA<B>: TMR0.BCONR.CKDIVA<B>位寄存器值

#### TIMEOUT 功能设定步骤

1. 将 USARTn\_CR1 寄存器设定为复位值；
2. 设定 UART 所需要使用的管脚；
3. 通过 USARTn\_CR2.CLK[1:0] 位选择时钟源(如果选择内部时钟源时需设置 CR2.CLKC[0]=1)；
4. 设定 USARTn\_CR1, USARTn\_CR2, USARTn\_CR3 寄存器；
5. 设定 USARTn\_PR 选择预分频值，USARTn\_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)；
6. USARTn\_CR1.RTOE=1，如果需要使用中断，则设定 USARTn\_CR1.RTOIE=1；
7. 设置 TMR0.BCONR.CSTA<B>=0；

8. 设置 TMR0\_CNTA<B>R 为 0，设置 TMR0\_CMPA<B>R 寄存器和 TMR0.BCONR.CKDIVA<B>寄存器决定 TIMETOU 时间；
9. 设置 TMR0.BCONR.HCLEA<B>=1, TMR0.BCONR.HSTAA<B>=1, TMR0.BCONR.ASYNCLKA<B>=1, TMR0.BCONR.SYNCLKA<B>=1, TMR0.BCONR.SYNSA<B>=1;
10. 使能接收器 (USARTn\_CR1.RE=1)，如果需要使用接收中断，则设置 USARTn\_CR1.RIE=1；
11. 检测到 TIMEOUT 后按照以下步骤设置关闭 TMR0 定时器和清除 USARTn\_SR.RTOF 状态位。同时设置 TMR0.BCONR.SYNCLKA<B>=1 和 TMR0.BCONR.SYNA<B>=0，再同时设置 TMR0.BCONR.CSTA<B>=0, TMR0.BCONR.SYNCLKA<B>和 TMR0.BCONR.SYNSA<B>=1，通过写 USARTn\_CR1.CRTOF 清除 USARTn\_SR.RTOF 状态位。

#### 28.4.1.7 RX 线唤醒停止模式功能

UART 通信空闲时，可以让系统进入停止模式以节省电流消耗，在不改变 UART PORT 设定的情况下，UART 单元 1 可以通过 RX 线来唤醒系统的停止模式。具体步骤如下：

1. UART 通信空闲时，设定 USART\_1\_WUPI 中断向量以及 INT\_WUPEN. RXWUEN 位使能 UART 接收信号线唤醒停止模式功能。
2. 系统进入停止模式。
3. 系统检测到 RX 线下降沿时，从停止模式返回，在 USART\_1\_WUPI 中断处理程序中关闭该功能。

需要注意的是，当通信方需要唤醒本系统时，需要发送一帧唤醒数据（建议为 0x00），该数据不会被 UART 接收且不置位相关的标志。并且通信方需要经过系统停止模式唤醒需要的时间后再进行 UART 通信数据的传输。

UART RX 线唤醒功能，可以对 RX 线上的噪声进行过滤，详细信息请参考 USART1\_NFC 寄存器。

#### 28.4.1.8 UART 半双工通信方式

UART 模式支持单线半双工模式，UART 模式下通过设定 USARTn\_CR3.HDSEL=1 启用单线半双工模式。

单线半双工模式时：

- TX 和 RX 线从内部相连接，不再使用 RX 管脚。
- 无数据传输且发送功能禁止时，TX 管脚处于释放状态。因此使用单线半双工模式时，通过上拉 TX 线来避免无数据传输时的浮空输入。
- 数据传输且发送功能使能时，TX 管脚输出为高电平。

除此以外，半双工模式与正常 UART 模式通信相似。需要注意的是，发送过程不会被硬件封锁，只要数据在 USARTn\_CR1.TE=1 时写入，发送就会进行。因此线路上的冲突必须有软件进行管理。

#### 28.4.1.9 UART 中断和事件

表 28-4 UART 中断/事件表

功能名称	使能位 (仅中断)	标志位	可否作为事件源
接收错误中断	RIE	ORE,FE,PE	可
接收数据寄存器满中断	RIE	RXNE	可
发送数据寄存器空中断	TXEIE	TXE	可
发送完成中断	TCIE	TC	可
发送完成中断	TENDIE	TEND	不可
TIMEOUT中断	RTOIE	RTOF	可
RX线唤醒停止模式中断	INT_WUPEN.RXWUEN	-	不可

#### 28.4.2 多处理器通信

##### 28.4.2.1 功能简介

多处理器通信模式是指多个处理器间共用通信线的一种通信方式，处理器分为发送站和接收站，每个接收站都有自己固有的 ID。发送站发送数据的类型有接收站 ID 和通信数据两种。通过在数据格式中添加 MPB 位来区分当前发送的是接收站的 ID 还是通信数据。MPB 位为 0 时当前帧为通信数据，MPB 位为 1 时当前帧为接收站的 ID。所有接收站都能接收发送站发送的 ID 并与自己的 ID 比较，如果一致，则接收数据，不一致则进入静默模式(既不接收数据也不置位接收相关标志)直到再次接收到 ID。

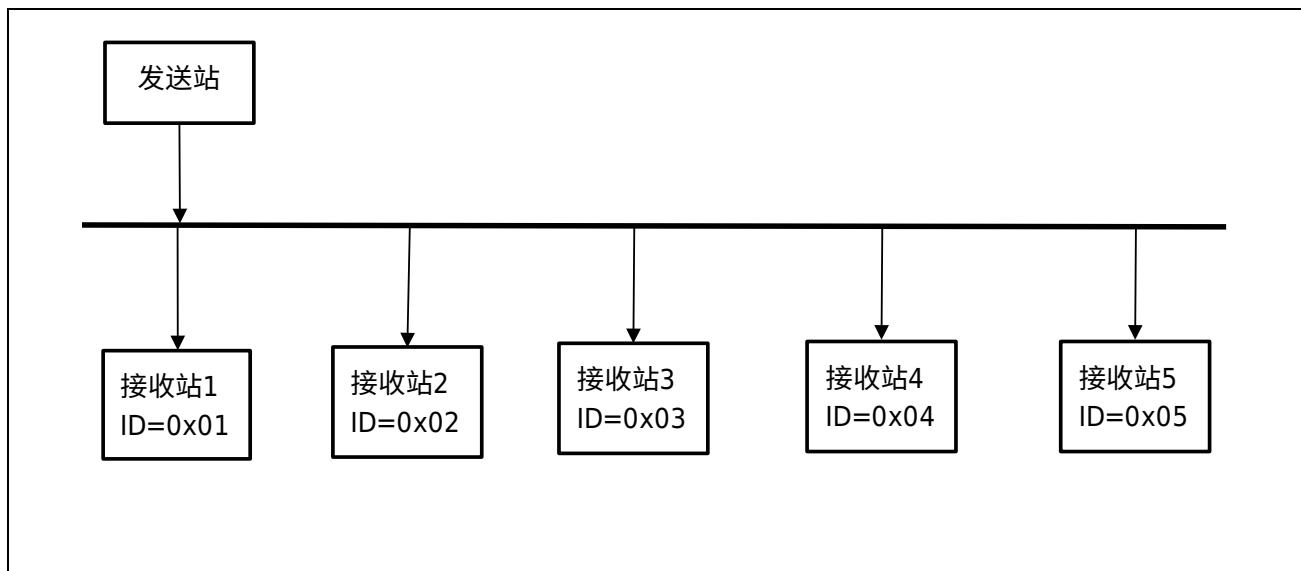


图 28-8 多处理器通信图例

#### 28.4.2.2 数据格式



ST: 开始位 (1位低电平)

D: 数据位 (可配置成 8 位或 9 位)

MPB: 多处理器位 (1: ID、0: 数据)

SP: 停止位 (可配置 1~2 位高电平)

图 28-9 多处理器模式数据格式

#### 28.4.2.3 动作说明

多处理器模式时校验位功能无效，增加了多处理器位功能，其余功能如时钟，中断等与 UART 模式相同。

##### 发送站动作

1. 将 USARTn\_CR1 寄存器设定为复位值
2. 设定所需要使用的管脚
3. 通过 USARTn\_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn\_CR1, USARTn\_CR2, USARTn\_CR3 寄存器
5. 设定 USARTn\_PR 选择预分频值, USARTn\_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. 使能发送器 (USARTn\_CR1. TE=1)，如果需要使用发送数据寄存器空中断，则设置 USARTn\_CR1. TXEIE=1
7. 等待发送数据寄存器空，设定 USARTn\_TDR.MPID 位为 1(发送 ID),写 ID 值到 USARTn\_TDR，发送 ID  
(CTS 功能有效时，USARTn\_CTS 输入为低电平时数据传输到发送移位寄存器，发送开始)
8. 设定 USARTn\_TDR.MPID 位为 0(发送数据)，写数据到 USARTn\_TDR，发送数据  
(CTS 功能有效时，USARTn\_CTS 输入低电平时数据传输到发送移位寄存器，发送开始)
9. 如果需要连续发送数据，重复步骤 8，如果需要改变 ID 后再发送数据重复 7 和 8
10. 通过确认 USARTn\_SR.TC 位确认发送是否完成。连续发送数据并使用发送中断的情况，可通过 TI 中断写入最后一个发送数据，并将 USARTn\_CR1. TXEIE 写 0, USARTn\_CR1. TCIE 写 1，最后一帧数据发送结束后，产生发送完成中断

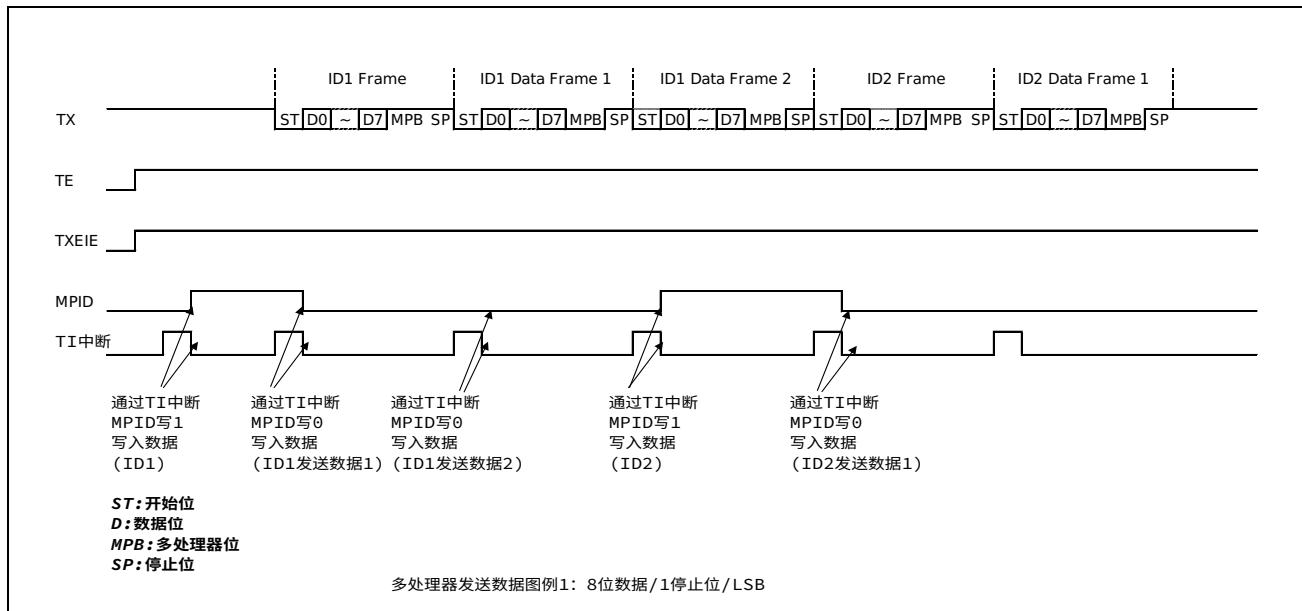


图 28-10 多处理器模式发送数据图例

### 接收站动作

在多处理器模式时，接收站必须保证能收到每一个 ID 数据，并与自身的 ID 比较，如果一致则接收数据，不一致则进入静默模式(不接收数据，也不置位接收相关标志)，直到接收到下一个 ID 数据。通过 USARTn\_CR1.SLME 位来实现这个功能。

USARTn\_CR1.SLME=0 时正常接收数据。

USARTn\_CR1.SLME=1 时，除非接收到 MPB 位为 1 ( ID ) 的数据，否则不接收数据，不发生 RI 中断，错误标志 FE,ORE 也不置位。当接收到 MPB 位为 1 的数据时( ID )，USARTn\_CR1.SLME 位自动清零，正常接收数据和发生中断。

动作步骤：

1. 将 USARTn\_CR1 寄存器设定为复位值
2. 设定所需要使用的管脚
3. 通过 USARTn\_CR1.CLKC[1:0]位选择时钟源
4. 设定 USARTn\_CR1, USARTn\_CR2, USARTn\_CR3 寄存器
5. 设定 USARTn\_PR 选择预分频值，USARTn\_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. USARTn\_CR1.RE=1, USARTn\_CR1.SLME=1(等待接收 ID) , 如果使用接收中断，则设定 USARTn\_CR1.RIE=1
7. 当检测到开始位时，接收器将数据接收到接收移位寄存器，并检查 USARTn\_SR.MPB 位
8. 如果 USARTn\_SR.MPB=1, USARTn\_CR1.SLME 位自动清零，正常接收数据，软件比较接收的 ID 与自身的 ID
  - 如果 ID 一致，则正常接收数据，发生中断，进行错误检测，与 UART 接收数据相同

- 如果 ID 不一致，软件再次将 USARTn\_CR1.SLME 位写 1，重复 8 的动作

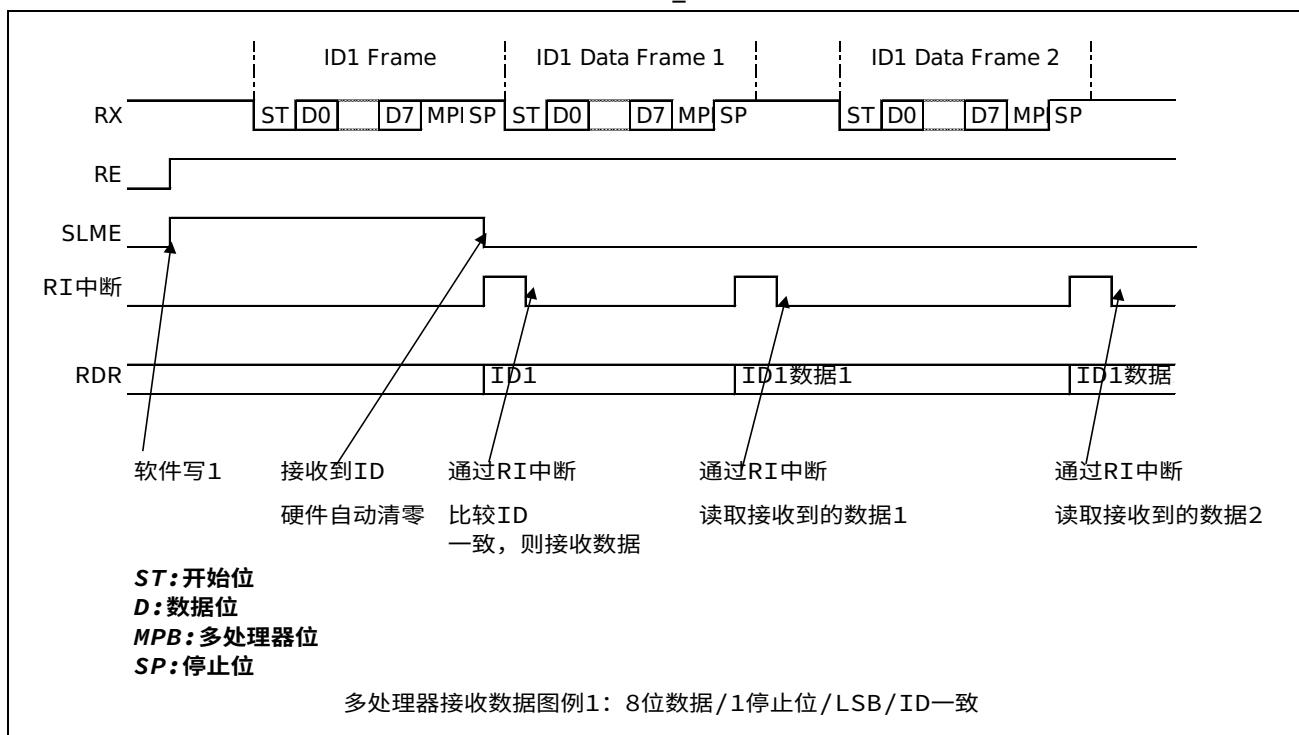


图 28-11 多处理器模式接收数据图例 1

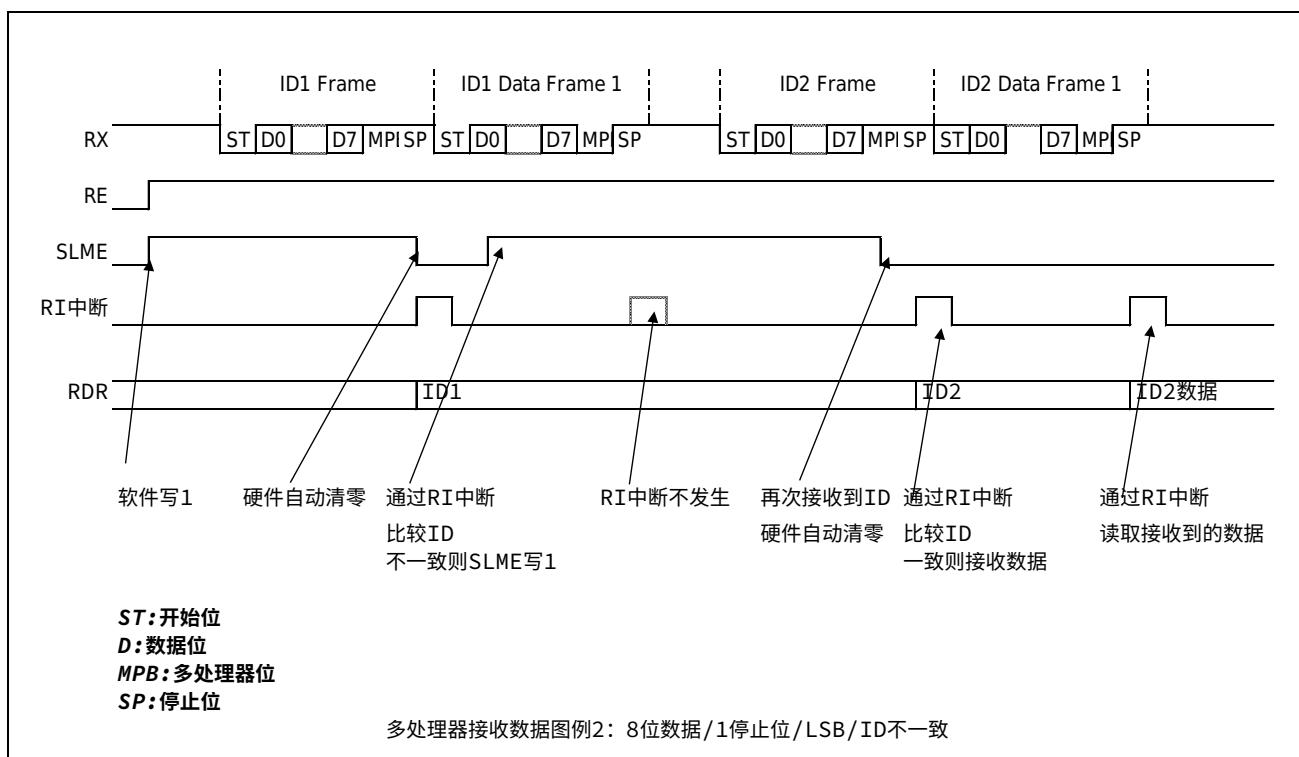


图 28-12 多处理器模式接收数据图例 2

#### 28.4.2.4 中断和事件

多处理器模式除了无校验错误外，中断处理与 UART 模式相同。

表 28-5 多处理器模式中断/事件表

功能名称	使能位（仅中断）	标志	可否作为事件源
接收错误中断	RIE	ORE,FE	可
接收数据寄存器满中断	RIE	RXNE	可
发送数据寄存器空中断	TXEIE	TXE	可
发送完成中断	TCIE	TC	可
发送完成中断	TENDIE	TEND	不可

#### 28.4.3 UART-LIN

##### 28.4.3.1 功能简介

LIN 是 Local Interconnect Network 的简称，是为降低汽车网络成本的低速（1~20kbps）串行通信协议。

##### 28.4.3.2 LIN 数据格式

LIN 的一帧数据是由开始位+8 位数据+1 停止位组成，以 LSB 方式发送和接收数据。LIN 总线上的数据行为如下图所示。

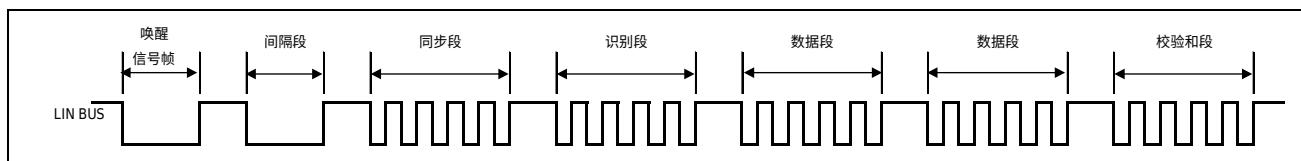


图 28-13 LIN 总线数据行为

##### 28.4.3.3 LIN 发送

###### 唤醒信号帧的发送

通过发送数据 0x80 对应信号帧的发送。

###### 间隔段发送

同步间隔段表示一帧数据的开始，本产品硬件支持 LIN 间隔段的发送。

LIN 间隔段低电平的宽度由寄存器 USARTn\_CR2.SBKL 寄存器设定。

USARTn\_CR2.SBKM=0 时，USARTn\_CR2.SBK 写 1，硬件自动送出间隔段。

USARTn\_CR2.SBKM=1 时，USARTn\_CR2.SBK 写 1 后，写数据 0x00 到发送数据寄存器 USARTn\_TDR.TDR 开始发送间隔段。

间隔段发送完成后，USARTn\_CR2.SBK 自动清零。

## 同步段发送

通过发送数据 0x55 实现同步段数据的发送。

## 总线错误检测

USARTn\_CR2.BEE=1 时,硬件检测到总线上的数据和发送的数据不一致时,置位 USARTn\_SR.BE 标志, USARTn\_CR2.BEIE=1 时,产生对应的中断。

### 28.4.3.4 LIN 接收

#### 唤醒信号帧的检测

USARTn\_CR2.WKUPE=1 时,硬件自动检测唤醒信号,检测到 RX 线低电平宽度上大于等于 2.5 位数据宽度 (19.2Kbps 时为 130uS) 时,置位 USARTn\_SR.WKUP 标志, USARTn\_CR2.WKUPIE=1 时,产生对应的中断。

USARTn\_CR2.WKUPE 寄存器仅在系统需要等待唤醒信号时设定为 1,其它情况需要设定为 0。

#### 同步间隔段的检测

本产品硬件支持同步间隔段检测,当检测到通信线低电平宽度大于等于 USARTn\_CR2.LBDL 寄存器设定的值,并检测到 break delimiter 时,置位 USARTn\_SR.LBD 标志, USARTn\_CR2.LBDIE=1 时,产生对应的中断。

需要注意的是,同步间隔段检测时,会置位 USARTn\_SR.FE 标志,清除 USARTn\_SR.LBD 标志的同时,需要清除 USARTn\_SR.FE 标志,并读取 USARTn\_RDR.RDR 寄存器(读取值为 0x00)。

#### 同步段的检测与波特率测量

当检测到间隔段后,硬件自动测量同步段的频率。测量计数器的时钟由 USARTn\_PR.LBMPSC 位设定,计数器的值保存在 USARTn\_LBMC 寄存器中,选择的计数器时钟频率除以 USARTn\_LBMC 寄存器值即可得到主节点的波特率。

需要注意的是,USARTn\_LBMC 寄存器的读取需要在同步段接收完成后再读取。

#### 接收数据设定步骤

1. 将 USARTn\_CR1 寄存器设定为复位值
2. 设定 UART 所需要使用的管脚
3. 通过 USARTn\_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn\_CR1(RE=1,RIE=0), USARTn\_CR2, USARTn\_CR3 寄存器

(以下步骤 5~6 为唤醒信号帧的检测,如无必要则跳过)

5. 设定 USARTn\_CR2.WKUP=1, 等待唤醒信号
6. 检测到唤醒信号, USARTn\_CR2.WKUP=0, 等待同步段间隔段

(以下步骤 7 为同步段间隔场检测，如无必要则跳过)

7. 检测到同步间隔断后，清除 USARTn\_SR.PE/FE/LBD 标志，并确认接收到的数据为 0x00。

(以下步骤 8~9 为同步段频率测量与通信波特率计算，如无必要则跳过)

8. 如果需要使用接收中断，则设置 USARTn\_CR1.RIE=1
9. 检测到同步间隔场，并完成同步段接收后，读取 USARTn\_LBMC 寄存器计算波特率，并确认接收到的数据为 0x55

(以下步骤 10~12 为接收识别段，数据以及校验和，过程与 UART 接收数据过程相同)

10. 接收识别段
11. 接收数据
12. 接收校验和

#### 28.4.3.5 LIN 中断和事件

表 28-6 LIN 中断/事件表

功能名称	使能位（仅中断）	标志位	可否作为事件源
错误中断	RIE	ORE,FE	可
	BEIE	BE	可
唤醒信号/间隔段检测中断	WKUPIE	WKUP	可
	LBD	LDBIE	可
接收数据寄存器满中断	RIE	RXNE	可
发送数据寄存器空中断	TXEIE	TXE	可
发送完成中断	TCIE	TC	可
发送完成中断	TENDIE	TEND	不可

## 28.4.4 智能卡

### 28.4.4.1 连接示意图

支持 ISO/IEC 7816-3 规定的智能卡通信协议。下图为智能卡模式的连接示意图。

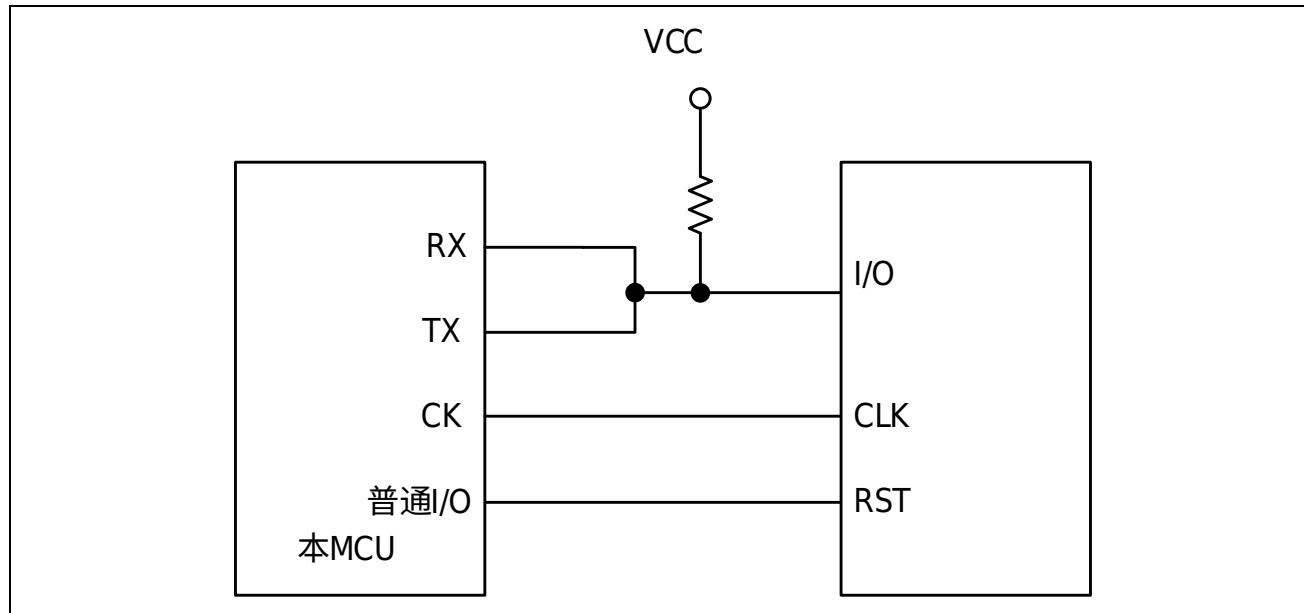


图 28-14 智能卡连接示意图

### 28.4.4.2 时钟

#### 内部时钟源

智能卡模式时只能使用内部波特率生成器生成的时钟作为时钟源。

一位数据传输的基本时钟数为 USARTn\_CR3.BCN[2:0]设定值。

通过设定寄存器 USARTn\_CR2.CLKC[1:0]位控制智能卡模式的时钟输出。

#### 最高波特率

内部时钟源时，内部波特率生成器生成的波特率计算公式为：

$$B = \frac{C}{2 \times BCN \times (DIV\_Integer + 1)}$$

B : 波特率 单位：Mbps

C : USARTn\_PR.PSC[1:0]位设定的时钟 (PCLK,PCLK/4,PCLK/16,PCLK/64) 单位：MHz

DIV\_Integer : USARTn\_BRR.DIV\_Integer 设定值

BCN : USARTn\_CR3.BCN 寄存器设定值

当 C 为 PCLK, DIV\_Integer=0, BCN=0 时，波特率为最高波特率为 PCLK/64(Mbps)。

### 采样和接收容差

检测到 RX 的下降后，USART 会基于内部基本时钟对接收数据进行时钟同步，从而开始数据接收。接收数据将在数据中央被采样。

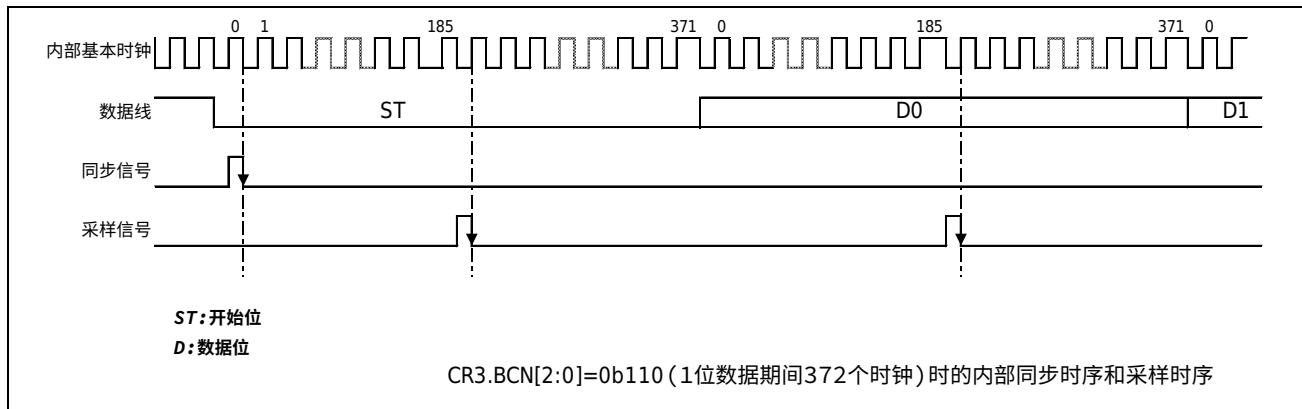


图 28-15 智能卡模式同步时序和采样时序图

接收容差的计算公式如下：

$$RM[\%] = |0.5 \times \left(1 - \frac{1}{BCN}\right) - 9.5CFD| \times 100$$

RM : 接收容差

BCN : 一位数据传输所需要的时钟数 (USARTn.CR3.BCN[2:0]设定值)

CFD : 时钟频率偏差

#### 28.4.4.3 数据格式

智能卡模式时一帧数据由起始位，数据位和校验位组成。

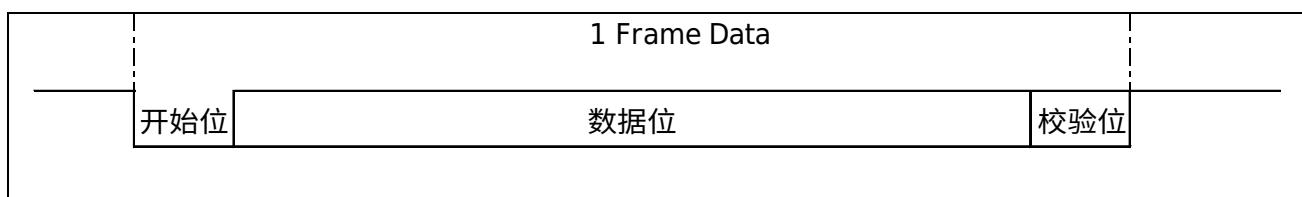


图 28-16 智能卡模式数据格式

##### 起始位

起始位固定有一位低电平构成。

##### 数据位

数据位固定为 8 位数据。

##### 校验位

校验位需配置成 1 位偶校验。

#### 28.4.4.4 智能卡的初始化设定步骤

1. 将 USARTn\_CR1 寄存器设定为复位值
2. 设定所需要使用的管脚
3. 状态寄存器确认， USARTn\_SR 寄存器设定为复位值
4. 设定 USARTn\_CR1, USARTn\_CR2, USARTn\_CR3 寄存器
5. 设定 USARTn\_PR 选择预分频值， USARTn\_BRR 寄存器设定通信波特率
6. USARTn\_CR2.CLKC[1:0]位设定时钟控制
7. USARTn\_CR1 寄存器(TE,RE,RIE,TXEIE 位)设定，除了自测外， TE 和 RE 不要同时设定为 1

由发送模式切换到接收模式，或者由接收模式切换到发送模式时，需要重新设定上述步骤 1 到步骤 7。

#### 28.4.4.5 智能卡模式动作说明

智能卡模式时， TI 中断(发送数据空中断)的标志位为 USARTn\_SR.TC 位。 USARTn\_SR.TC=1 而且 USARTn\_CR1.TXEIE=1 时产生 TI 中断。

##### 功能概述

发送数据时两帧数据间 (从校验位结束到下一帧的起始位开始)有 2etu(Elementary Time Unit)以上的保护时间。

发送数据时如果检测到接收方发送的错误信号则经过 2etu 后自动重发数据。

接收数据发生校验错误时，发送 1etu 的低电平即错误信号，错误信号送出的时序为接收开始经过 10.5etu。

##### 发送说明

1. 一帧数据发送完成后，如果检测到接收方发送的错误信号，则 USARTn\_SR.FE 置 1 (如果 USARTn\_CR.RIE=1, 则发生错误中断)， USARTn\_SR.TC 标志不置 1，数据自动重发。 USARTn\_SR.FE 位必须在接受到下一帧校验位之前清零。
2. 一帧数据无错误发送完成后， USARTn\_SR.TC 标志置位， USARTn\_CR1.TXEIE=1 时发生 TI 中断。再次写入数据，则可实现连续发送数据。

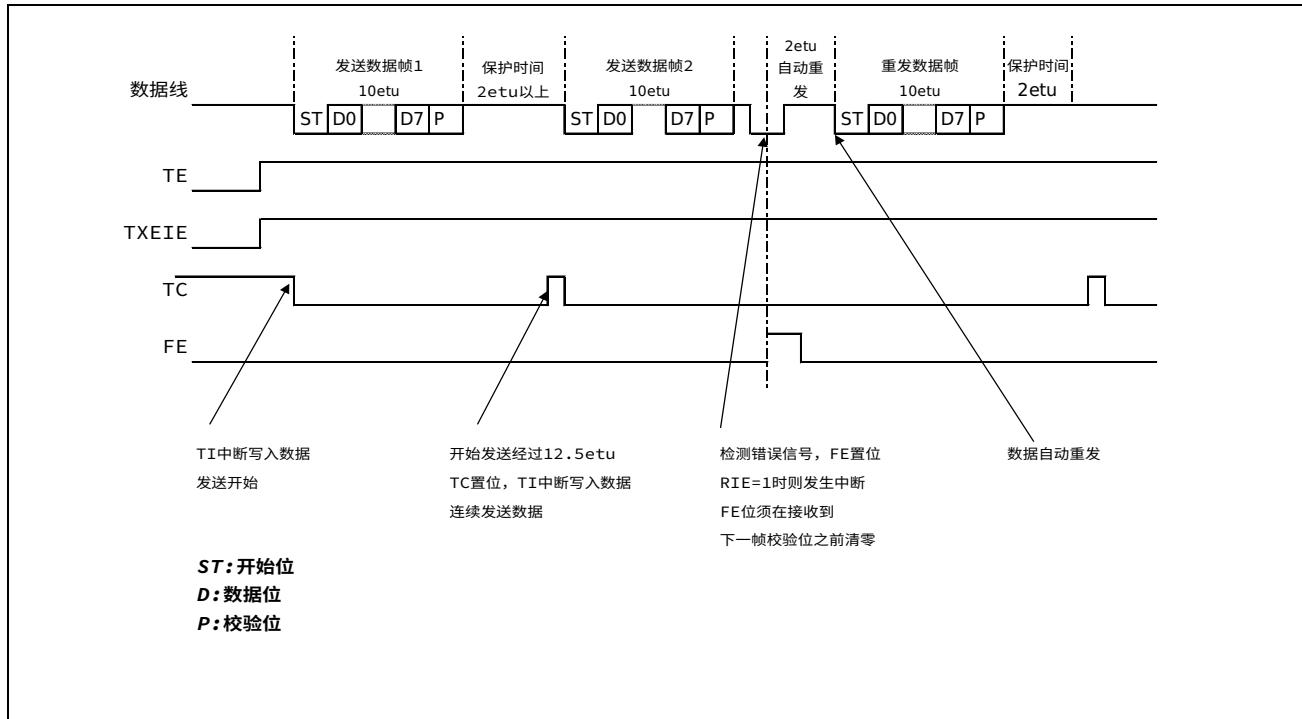


图 28-17 智能卡模式发送数据图例

### 接收说明

1. 接收数据时，如果检测到校验错误，则 USARTn\_SR.PE 置位，中断使能时，发生 EI 中断。USARTn\_SR.PE 位需要在接收到下一帧校验位之前清零。
2. 发生校验错误时，会发送 1etu 的低电平，即错误信号，要求发送方重新发送数据。
3. 正常接收数据，可以通过 RI 中断读取接收到的数据，连续接收。
4. 接收数据时，检测上溢错误。

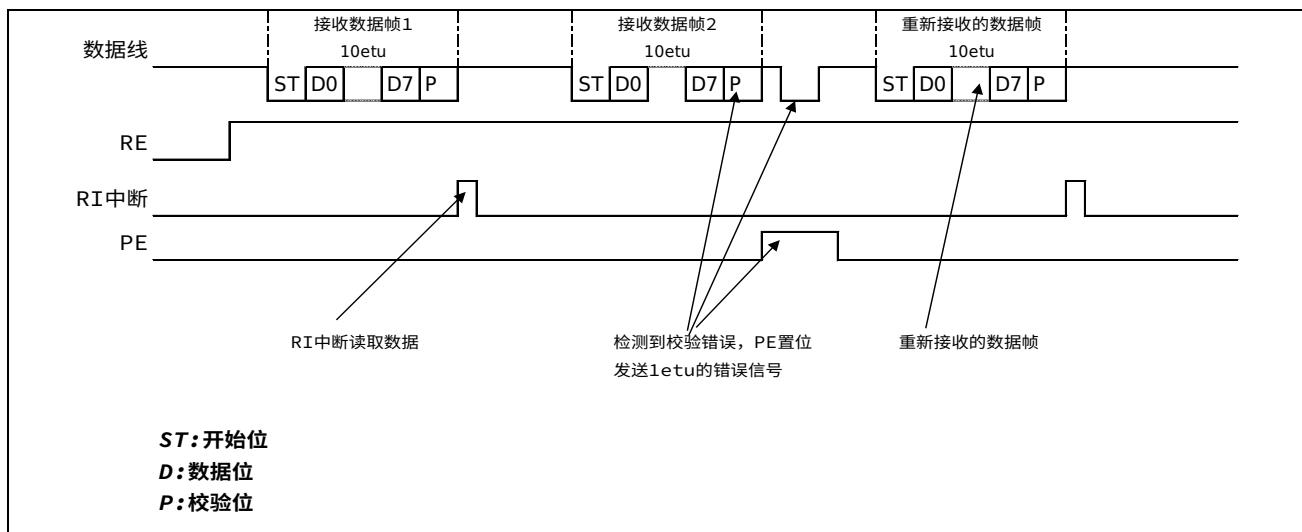


图 28-18 智能卡模式接收数据图例

#### 28.4.4.6 中断和事件

表 28-7 智能卡模式中断/事件表

功能名称	使能位（仅中断）	标志位	可否作为事件源
错误中断	RIE	ORE,PE,FE	可
接收数据满中断	RIE	RXNE	可
发送数据空中断	TXEIE	TC	可

#### 28.4.5 时钟同步模式

##### 28.4.5.1 时钟

时钟同步模式可以选择内部波特率生成器生成的时钟(内部时钟源)或 USARTn\_CK 管脚输入的时钟(外部时钟源)作为通信的时钟源。

###### 内部时钟源

同步时钟从 USARTn\_CK 管脚输出，一帧数据输出 8 个时钟脉冲，既不发送数据也不接收数据时，时钟输出固定为高电平。

###### 外部时钟源

外部时钟源即从 USARTn\_CK 管脚输入时钟作为通信时钟。

###### 最高波特率

内部时钟源时，内部波特率生成器生成的波特率计算公式为：

$$B = \frac{C}{4 \times (\text{DIV\_Integer} + 1)}$$

B：波特率 单位：MBps

C：USARTn\_PR.PSC[1:0]位设定的时钟 (PCLK,PCLK/4,PCLK/16,PCLK/64) 单位：MHz

DIV\_Integer：USARTn\_BRR.DIV\_Integer 设定值

内部时钟源时，当 C 为 PCLK, DIV\_Integer=1 时，最高波特率为 PCLK/8(MBps)。注意同步模式时 DIV\_Integer 禁止设置为 0。

外部时钟源时，外部输入时钟的最大频率要求为 PCLK(MHz)/6，所以最高波特率为 PCLK/6(MBps)。

需要注意的是，同步模式最高通信波特除了以上描述的基于 PCLK 的计算方法外，还需要参考电气特性章节规定的最高通信波特率。

#### 28.4.5.2 数据格式

时钟同步模式一帧数据固定有 8 位组成，一帧数据的发送和接收需要 8 个同步时钟脉冲。发送数据时数据在同步时钟的下降沿送出，接收数据时数据在同步时钟的上升沿被采样。

同步时钟在没有数据传输时固定为高电平，最后一位发送完后，通信线保持最后一位的值。

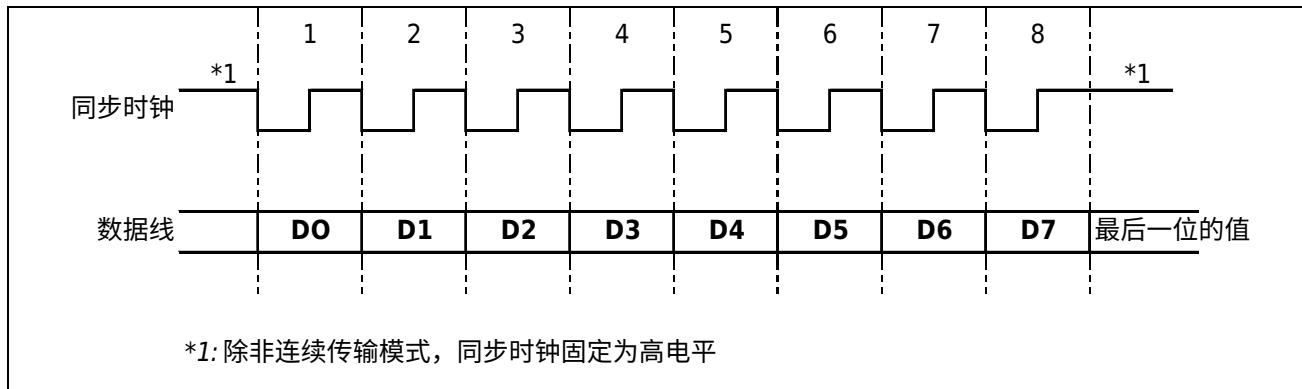


图 28-19 时钟同步模式数据格式

#### 28.4.5.3 调制解调器操作

调制解调器操作包括 CTS 功能和 RTS 功能。USARTn\_CR3.RTSE=1 时 RTS 功能有效，USARTn\_CR3.CTSE=1 时 CTS 功能有效。

##### CTS 功能

CTS 功能是通过 USARTn\_CTS 管脚的输入来控制数据的发送，只有当 USARTn\_CTS 管脚输入低电平时才可以发送数据，发送数据过程中如果 USARTn\_CTS 输入高电平，正在发送的数据不受影响。

##### RTS 功能

RTS 功能是指通过 USARTn\_RTS 管脚输出低电平，请求对方发送数据。

USARTn\_RTS 管脚输出低电平需要满足以下全部条件：

- 接收使能(USARTn\_CR1.RE=1)，且不正在接收数据
- USARTn\_RDR.RDR 寄存器中没有未读取的数据(USARTn\_CR1.RE=1 时)
- USARTn\_TDR.TDR 更新完成(USARTn\_CR1.TE=1 时)
- 无任何接收错误

如果不能同时满足以上全部条件，USARTn\_RTS 则输出高电平。

#### 28.4.5.4 发送器

发送器使能位 (USARTn\_CR1.TE) 置 1 时，发送移位寄存器中的数据在 USARTn\_TX 管脚串行输出，相应的时钟脉冲在 USARTn\_CK 管脚输出。

发送数据寄存器 USARTn\_TDR.TDR 和内部的发送移位寄存器组成双缓冲器结构，可以连续发送数据。

通过发送数据寄存器空中断或者 DMA 写入发送数据时，为保证发送的正确性，一次请求只能写入一次数据。

### 发送数据设定步骤

1. 将 USARTn\_CR1, USARTn\_SR1 寄存器设定为复位值
2. 设定所需要使用的管脚
3. 通过 USARTn\_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn\_CR1, USARTn\_CR2, USARTn\_CR3 寄存器
5. 设定 USARTn\_PR 选择预分频值, USARTn\_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. 使能发送器 (USARTn\_CR1. TE=1)，如果需要使用发送数据寄存器空中断，则设置 USARTn\_CR1. TXEIE=1
7. 等待发送数据寄存器空，写通信数据到 USARTn\_TDR.TDR，数据传输到发送移位寄存器，发送开始  
(CTS 功能有效时，USARTn\_CTS 输入为低电平时数据传输到发送移位寄存器，发送开始)
8. 如果需要连续发送数据时，重复步骤 7
9. 通过确认 USARTn\_SR.TC 位确认发送是否完成。连续发送数据并使用发送中断的情况，可通过 TI 中断写入最后一个发送数据，并将 USARTn\_CR1. TXEIE 写 0, USARTn\_CR1. TCIE 写 1，最后一个数据发送结束后，产生发送完成中断

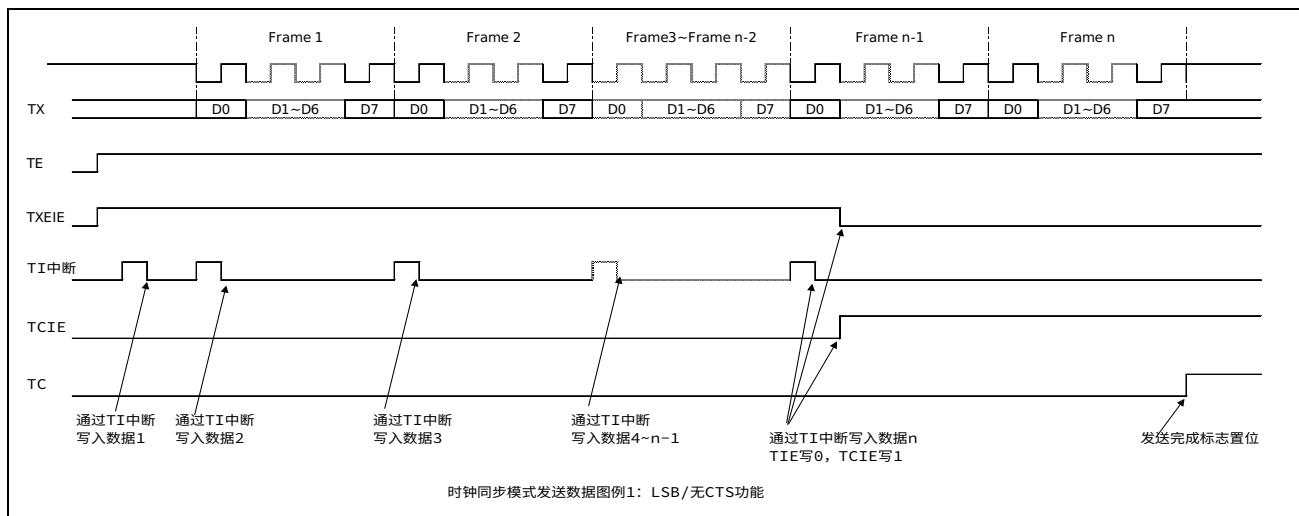


图 28-20 时钟同步模式发送数据图例 1

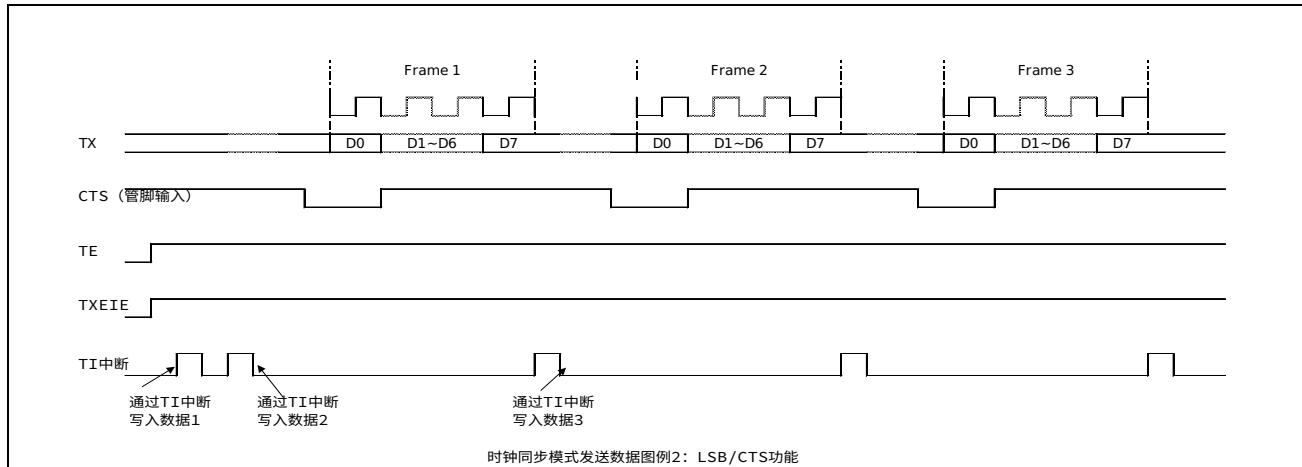


图 28-21 时钟同步模式发送数据图例 2

### 发送器中断

时钟同步模式发送器支持两种中断，发送数据寄存器空中断 TI 和发送完成中断 TCI。

TXEIE=1，USARTn\_TDR.TDR 寄存器的值传送到发送移位寄存器时 TI 中断发生。

TCIE=1，发送数据的最后一位时 USARTn\_TDR.TDR 寄存器没有更新则 TCI 中断发生。

### 28.4.5.5 接收器

#### 接收数据设定步骤

1. 将 USARTn\_CR1, USARTn\_SR 寄存器设定为复位值
2. 设定所需要使用的管脚
3. 通过 USARTn\_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn\_CR1, USARTn\_CR2, USARTn\_CR3 寄存器
5. 设定 USARTn\_PR 选择预分频值，USARTn\_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. 使能接收器 (USARTn\_CR1.RE=1)，如果需要使用接收中断，则设置 USARTn\_CR1.RIE=1  
(使用 RTS 功能时，RE=1 后 USARTn\_RTS 输出低电平)
7. 同步于输入同步时钟或者内部生成的同步时钟开始接收数据，接收数据到接收移位寄存器。
  - 发生上溢错误时，数据丢失并置位 USARTn\_SR.ORE 标志
  - 无错误发生时，接收到的数据传送到 USARTn\_RDR.RDR 寄存器中，置位 USARTn\_SR.RXNE 标志，将当前接收到的数据在接收到下一帧数据最后一位前读取后重复步骤 7 可实现连续接收数据功能。  
(使用 RTS 功能时，数据读取后 USARTn\_RTS 输出低电平)

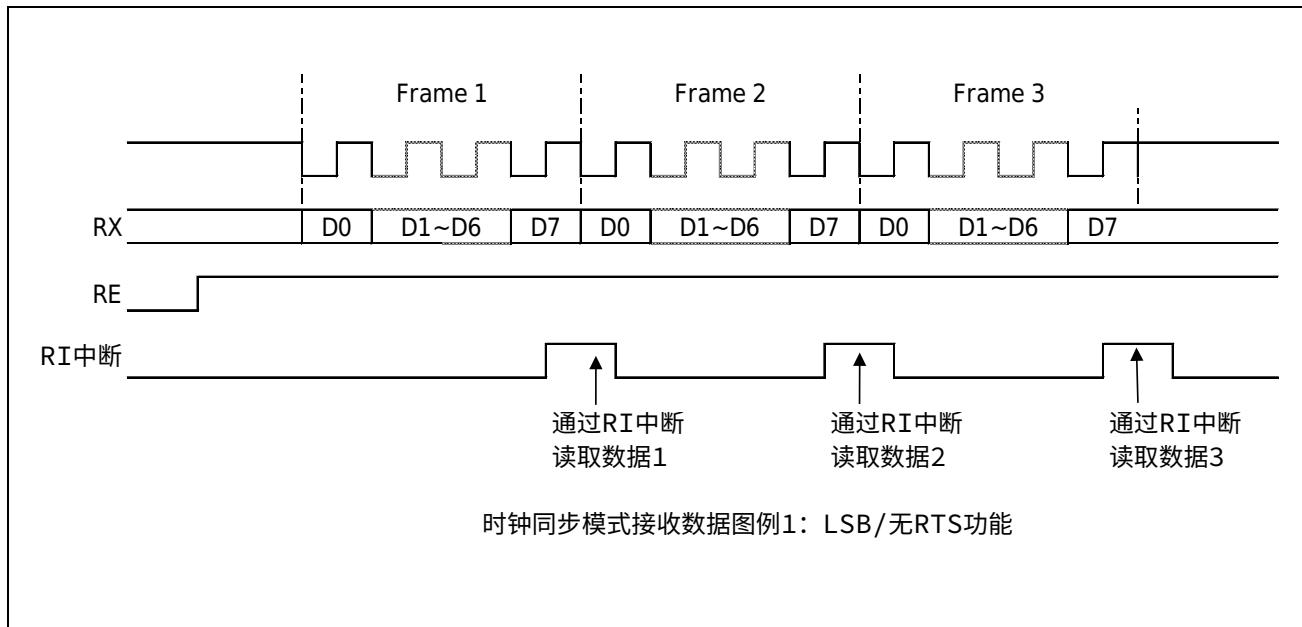


图 28-22 时钟同步模式接收数据图例 1

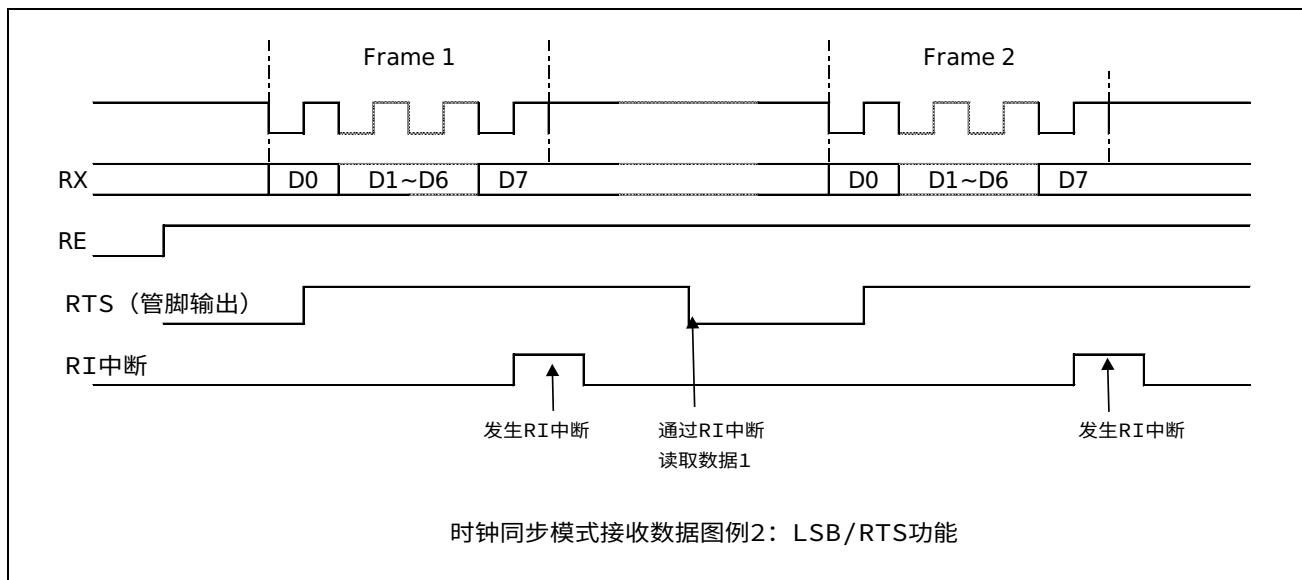


图 28-23 时钟同步模式接收数据图例 2

### 错误处理

时钟同步模式接收数据时接收错误为上溢错误 (USARTn\_SR.ORE)。发生接收错误不能再进行数据的接收和发送。可以通过将错误标志清零来重启数据传输，清除方法是写对应的清零寄存器。

上溢错误发生的条件是 USARTn\_RDR.RDR 值未被读取的情况下又收到新的数据，所以应该在收到当前帧最后一位之前将前一帧收到的数据读取。发生上溢错误时接收到的数据丢失，RI 中断不发生。

### 接收器中断

时钟同步模式接收器支持两种中断，接收数据寄存器满中断 RI 和接收错误中断 EI。

RIE=1，数据从接收移位寄存器传送到接收数据寄存器时 RI 中断发生。

RIE=1，接收数据发生错误（上溢错误）时 EI 中断发生。

#### 28.4.5.6 同时发送接收数据

USART 时钟同步模式支持全双工动作，同时发送接收数据。同时发送接收数据时需要一个命令把 RE,TE,RIE,TXEIE 写 1，其它设定流程与发送器和接收器相同。

#### 28.4.5.7 时钟同步模式中断和事件

表 28-8 时钟同步模式中断/事件表

中断名称	使能位(仅中断)	标志位	可否作事件源
接收错误中断	RIE	ORE	可
接收数据寄存器满中断	RIE	RXNE	可
发送数据寄存器空中断	TXEIE	TXE	可
发送完成中断	TCIE	TC	可
发送完成中断	TENDIE	TEND	不可

#### 28.4.6 数字滤波功能

USARTn\_CR1.NFE=1 时，内置数字滤波器功能有效。数字滤波器仅在 UART 模式时有效，可以除去接收数据线 RX 上的噪音。

内置数字滤波器可以滤除小于一位数据的 3/16(USARTn\_CR1.OVER8=0) 宽度或者 3/8 宽度(USARTn\_CR1.OVER8=1)的噪音。

如果数字滤波器的时钟停止后再开始时，数字滤波器从时钟停止时保持的状态继续工作。

USARTn\_CR.TE=0 且 USARTn\_CR.RE=0，将数字滤波器内部的 Flip-Flop 状态复位为 1。

## 28.4.7 中断

下表给出 USART 整体中断：

表 28-9 USART 整体中断一览表

中断名称	符号	说明
错误中断	USART_n_EI n=1~6	UART/多处理器/智能卡/时钟同步模式接收错误中断 LIN错误中断
接收数据寄存器满中断	USART_n_RI n=1~6	UART/多处理器/LIN/智能卡/时钟同步模式接收数据寄存器满中断 LIN错误中断
发送数据寄存器空中断	USART_n_TI n=1~6	UART/多处理器/LIN/智能卡/时钟同步模式发送数据寄存器空中断
发送完成中断	USART_n_TCI n=1~6	UART/多处理器/LIN/智能卡/时钟同步模式发送完成中断 (向发送数据寄存器写入发送数据可清中断)
发送完成中断	USART_n_TENDI n=1~6	UART/多处理器/LIN/智能卡/时钟同步模式发送完成中断 (向寄存器CR1.CTEND位写入1或向发送数据寄存器写入发送数据可清中断)
唤醒信号/间隔段检测中断	USART_n_BRWKWP1 n=3/6	LIN 唤醒信号/间隔段检测中断
UART接收TIMEOUT中断	USART_n_RTO n=1/2/4/5	UART接收TIMEOUT中断
RX线唤醒停止模式中断	USART_1_WUPI	USART_1 RX线唤醒停止模式中断

## 28.5 寄存器说明

本章将详细描述 USART 模块控制状态等相关寄存器，需要注意的是功能寄存器位只有该通道配置了对应的功能才有效，否则默认为 0。

基地址如下：

USART\_1\_BASE\_ADDR:0x4001CC00

USART\_2\_BASE\_ADDR:0x4001D000

USART\_3\_BASE\_ADDR:0x4001D400

USART\_4\_BASE\_ADDR:0x40020C00

USART\_5\_BASE\_ADDR:0x40021000

USART\_6\_BASE\_ADDR:0x40021400

表 28-10 USART 寄存器一览表

寄存器名	偏移地址	复位值
状态寄存器 (USART_SR)	0x00	0x0000 00C0
发送数据寄存器 (USART_TDR)	0x04	0x01FF
接收数据寄存器 (USART_RDR)	0x06	0x0000
波特率寄存器 (USART_BRR)	0x08	0x0000 FFFF
控制寄存器1 (USART_CR1)	0x0C	0x8000 0000
控制寄存器2 (USART_CR2)	0x10	0x0000 0600
控制寄存器3 (USART_CR3)	0x14	0x0000 00000
预分频寄存器 (USART_PR)	0x18	0x0000 0000
LIN波特率测量计数寄存器 (USART_LBMC)	0x1C	0x0000 0000

## 28.5.1 状态寄存器 (USART\_SR)

USART Status Register

偏移地址：0x00

复位值：0x000000C0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	MPB
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	TEN D	LBD	WKU P	RTOF	TXE	TC	RXN E	BE	ORE	-	FE	PE
<hr/>															
位	标记	位名	功能												读写
b31~b17	Reserved	-	读出时为“0”,写入时写“0”												R/W
多处理器位标志															
b16	MPB	多处理器位	0: 当前接收数据为通信数据 1: 当前接收数据为ID 注意: MPB位只在多处理器模式时有效												R
b15~b12	Reserved	-	读出时为“0”,写入时写“0”												R/W
发送完成标志位															
b11	TEND	发送完成标志	0: 发送数据中 1: 发送数据完成 UART模式, 时钟同步模式 TEND置位条件 • 送出一帧数据的最后一一位时, 发送数据寄存器的值没有被更新 TEND清零条件 • TE=1时, 向发送数据寄存器写入发送数据 • 清零寄存器CR1.CTEND位写入1 智能卡模式 TEND置位条件 • 最后1字节的数据送出后经过特定时间后, FE=0而且发送数据寄存器的值没有被更新。 TEND置位的具体时序为: 校验位送出后经过2.5位时间 TEND清零条件 • TE=1时, 向发送数据寄存器写入发送数据 • 清零寄存器CR1.CTEND位写入1												R
LIN 间隔段检出标志 (仅USART_3/USART_6有效)															
b10	LBD	LIN 间隔段检出标志	0: 未检测到间隔段 1: 检测到间隔段 LBD置位条件 CR2.WKUPE=0时, 检测到RX线低电平宽度上大于等于CR2.LBDL寄存器设定的宽度 后, 并检测到break delimiter时 清零条件 清零寄存器CR1.CLBD位写入1 注意: RE=0并不能复位LBD位												R
b9	WKUP	LIN Wakeup信号标志	LIN Wakeup信号检出标志 (仅USART_3/USART_6有效) 0: 未检测到Wakeup信号												R

			1: 检测到Wakeup信号 WKUP置位条件 CR2.WKUPE=1时, 检测到RX线低电平宽度上大于等于2.5位数据宽度 (19.2Kbps时为130uS) 清零条件 清零寄存器CR1.CWKUP位写入1 注意: RE=0并不能复位WKUP位 检测到WKUP信号为1后, 表示本LIN节点已经从休眠模式退出, 设定寄存器CR2.WKUPE=0后, 进行后续间隔场检测等操作。	
b8	RTOF	UART接收 TIMEOUT标志位	UART接收TIMEOUT标志位 (仅USART_1/USART_2/ USART_4/ USART_5有效) 0: 无UART接收TIMEOUT 1: 发生UART接收TIMEOUT RTOF置位条件 • 从检测到最后一帧数据的STOP位起经过设定时间后没有检测到新的接收数据 RTOF清零条件 • 清零寄存器CR1.CRTOF位写入 注意: RTOF由硬件置1, 且只有在CR1.RE=1且CR1.RTOE=1时置1。CR1.RE=0时, TIMEOUT功能有效, 但RTOF不置1。	R
b7	TXE	发送数据寄存器空	发送数据寄存器空标志 TXE位在UART/时钟同步模式/LIN模式时有效。 0: 数据未传输到移位寄存器, 发送数据寄存器非空 1: 数据传输到移位寄存器, 发送数据寄存器空 注意: TXE位由硬件置1和清0, 数据未传输到移位寄存器时硬件将TXE清0, 数据传输到移位寄存器时硬件将TXE置1	R
b6	TC	发送完成标志	发送完成标志位 0: 发送数据中 1: 发送数据完成 UART模式, 时钟同步模式 TC置位条件 • TE=0发送禁止时 • 送出一帧数据的最后一一位时, 发送数据寄存器的值没有被更新 TC清零条件 • TE=1时, 向发送数据寄存器写入发送数据 智能卡模式 TC置位条件 • TE=0发送禁止时 • 最后1字节的数据送出后经过特定时间后, FE=0而且发送数据寄存器的值没有被更新。 TC置位的具体时序为: 校验位送出后经过2.5位时间 TC清零条件 • TE=1时, 向发送数据寄存器写入发送数据 注意: TE位由0变化为1时TC为保持为1	R
b5	RXNE	接收数据寄存器不为空	接收数据寄存器不为空标志 0: 未接收到数据 1: 准备好读取接收到的数据 注意: RXNE位由硬件置1和清0, 准备好读取接收到的数据时硬件将RXNE置1, 读取接收数据后硬件将RXNE清0	R
b4	BE	LIN总线检错误标	LIN总线检错误标志 (仅USART_3/USART_6有效)	R

		志	<p>0: 无总线错误 1: 发生总线错误</p> <p>BE置位条件</p> <ul style="list-style-type: none"> <li>发送数据时, 检测到总线数据和发送数据不一致, 检测位置为每一位的13/16处(16位过采样)</li> </ul> <p>BE清零条件</p> <ul style="list-style-type: none"> <li>清零寄存器CR1.CBE位写入1</li> </ul> <p>注意: TE=0并不能复位BE位</p> <p>BE=1时, 正在发送的数据发送完成后, 不会开始新的数据发送, 需要软件清除BE标志, 重启发送动作</p>	
b3	ORE	接收上溢错误	<p>接收上溢错误标志位</p> <p>0: 无接收上溢错误 1: 发生接收上溢错误</p> <p>ORE置位条件</p> <ul style="list-style-type: none"> <li>接收数据寄存器未被读取的情况下, 又接收到一帧新的数据</li> </ul> <p>ORE清零条件</p> <ul style="list-style-type: none"> <li>清零寄存器CR1.CORE位写入1</li> </ul> <p>注意: RE=0并不能复位ORE位</p> <p>ORE=1之前接收到的数据会被保持, ORE=1时接收到的数据会丢掉</p> <p>ORE=1后不能继续接收数据, 时钟同步模式下也不能发送数据</p>	R
b2	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b1	FE	接收帧错误	<p>接收帧错误标志位</p> <p>0: 无接收帧错误 1: 发生接收帧错误</p> <p>UART模式</p> <p>FE置位条件</p> <ul style="list-style-type: none"> <li>接收的数据帧的停止位为低电平, 两个停止位的情况只检查第一个停止位</li> </ul> <p>FE清零条件</p> <ul style="list-style-type: none"> <li>清零寄存器CR1.CFE位写入1</li> </ul> <p>注意: UART模式时, RE=0并不能复位PE位</p> <p>FE=1时收到的数据会保留但是RI中断不会发生, FE=1后不能继续接收数据</p> <p>智能卡模式</p> <p>FE置位条件</p> <ul style="list-style-type: none"> <li>采样到低电平错误信号标志</li> </ul> <p>FE清零条件</p> <ul style="list-style-type: none"> <li>清零寄存器CLR.CFE位写入1</li> </ul> <p>注意: 智能卡模式时, RE=0并不能复位FE位</p>	R
b0	PE	接收数据校验	<p>接收数据校验错误标志</p> <p>0: 无接收数据校验错误 1: 发生接收数据校验错误</p> <p>PE置位条件</p> <ul style="list-style-type: none"> <li>接收数据发生奇偶校验错误时</li> </ul> <p>PE清零条件</p> <ul style="list-style-type: none"> <li>清零寄存器CR1.CPE位写入1</li> </ul> <p>注意: RE=0并不能复位PE位</p> <p>PE=1时收到的数据会保留但是RI中断不会发生, PE=1后不能继续接收数据</p>	R

## 28.5.2 发送数据寄存器 (USART\_TDR)

USART Transmit Data Register

偏移地址：0x04

复位值：0x01FF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0					
-	-	-	-	-	-	MPID	TDR[8:0]													

位	标记	位名	功能	读写
b15~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9	MPID	多处理器模式ID位	多处理器模式时,发送通信数据或者发送ID的选择位 0: 发送数据 1: 发送ID 注意: MPID位只在多处理器模式时有效,其他模式须设定为复位值	R/W
b8~b0	TDR[8:0]	发送数据寄存器	发送数据寄存器 注意: 最高位TDR[8]仅在UART模式而且数据长度设定为9位时有效	R/W

## 28.5.3 接收数据寄存器 (USART\_RDR)

USART Receive Data Register

偏移地址：0x06

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0						
-	-	-	-	-	-	-	-	RDR[8:0]													

位	标记	位名	功能	读写
b15~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8~b0	RDR[8:0]	接收数据寄存器	接收数据寄存器 注意: 最高位RDR[8]仅在UART模式而且数据长度设定为9位时有效	R

## 28.5.4 波特率寄存器 (USART\_BRR)

USART Bit Rate Register

偏移地址：0x08

复位值：0x0000FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DIV_Integer [7:0]										-	DIV_Fraction[6:0]				
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b15~b8	DIV_Integer[7:0]	整数分频寄存器	整数分频寄存器	R/W											
b7	Reserved	-	写入时需写入复位值	R/W											
b6~b0	DIV_Fraction[6:0]	小数分频寄存器	注意：DIV_Fraction[6:0]只能在TE=0&RE=0(发送/接收禁止)时设定	R/W											
			注意：DIV_Fraction[6:0]只能在TE=0&RE=0(发送/接收禁止)时设定，而且仅在FBME=1时设定值有效	R/W											

表 28-11 波特率计算公式 (小数波特率无效 FBME=0)

模式	波特率计算公式	误差E(%)计算公式
UART模式 多处理器模式	$B = \frac{C}{8 \times (2 - OVER8) \times (DIV\_Integer + 1)}$	$E(\%) = \left\{ \frac{C}{8 \times (2 - OVER8) \times (DIV\_Integer + 1) \times B} - 1 \right\} \times 100$
时钟同步模式	$B = \frac{C}{4 \times (DIV\_Integer + 1)}$	-
智能卡模式	$B = \frac{C}{2 \times BCN \times (DIV\_Integer + 1)}$	$E(\%) = \left\{ \frac{C}{2 \times BCN \times (DIV\_Integer + 1) \times B} - 1 \right\} \times 100$

B : 波特率 单位：Mbps

C : PR.PSC[1:0]位设定的时钟 单位：MHz

BCN : CR3.BCN 寄存器设定值

表 28-12 波特率计算公式 (小数波特率有效 FBME=1)

模式	波特率计算公式	误差E(%)计算公式
UART模式 多处理器模式	$B = \frac{C \times (128 + DIV\_Fraction)}{8 \times (2 - OVER8) \times (DIV\_Integer + 1) \times 256}$	$E(\%) = \left\{ \frac{C \times (128 + DIV\_Fraction)}{8 \times (2 - OVER8) \times (DIV\_Integer + 1) \times 256 \times B} - 1 \right\} \times 100$
时钟同步模式	$B = \frac{C \times (128 + DIV\_Fraction)}{4 \times (DIV\_Integer + 1) \times 256}$	
智能卡模式	$B = \frac{C \times (128 + DIV\_Fraction)}{2 \times BCN \times (DIV\_Integer + 1) \times 256}$	$E(\%) = \left\{ \frac{C \times (128 + DIV\_Fraction)}{2 \times BCN \times (DIV\_Integer + 1) \times 256 \times B} - 1 \right\} \times 100$

B : 波特率 单位：Mbps

C : PR.PSC[1:0]位设定的时钟 单位：MHz

BCN : CR3.BCN 寄存器设定值

## 28.5.5 控制寄存器 1 (USART\_CR1)

USART Control Register 1

偏移地址：0x0C

复位值：0x80000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SBS	NFE	FBME	ML	-	-	CTE ND	MS	CLB D	CWK UP	CBE	CRTO F	COR E	-	CFE	CPE
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OVER 8	-	-	M	-	PCE	PS	TEN DIE	TXEI E	TCIE	RIE	SLME	TE	RE	RTOI E	RTOE

位	标记	位名	功能	读写
b31	SBS	UART模式接收数据时，开始位检测方式设定位	0: 开始位检测方式为RX管脚低电平 1: 开始位检测方式为RX管脚下降沿 注意：非UART模式时SBS位须保持复位值 SBS位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b30	NFE	数字滤波使能位	0: 禁止数字滤波功能 1: 使能数据滤波功能 注意：非UART模时NFE位必须保持复位值 NFE位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b29	FBME	小数波特率功能使能	0: 禁止 1: 使能 注意：FBME位只能在TE=0&RE=0(发送/接收禁止)时设定 LINEN=1时不支持设定FBME=1	R/W
b28	ML	MSB/LSB选择位	0: LSB方式 1: MSB方式 注意：ML位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b27~b26	Reserved	-	读出时为“0”,写入时写“0”	R/W
b25	CTEND	TEND清零位	0:不清零TEND标志 1:清零TEND标志 注意：CTEND位写1清除TEND标志	R/W
b24	MS	通信模式选择位	0: UART模式 1: 时钟同步模式 注意：MS位只能在TE=0&RE=0(发送/接收禁止)时设定，智能卡模式MS需写入复位值	R/W
b23	CLBD	LBD清零位	0:不清零LBD标志 1:清零LBD标志 注意：CLBD位写1清除LBD标志，读取时返回0	R/W
b22	CWKUP	WKUP清零位	WKUP清零位（仅USART_3/USART_6有效）	R/W

				0:不清零WKUP标志 1:清零WKUP标志 注意：CWKUP位写1清除WKUP标志，读取时返回0
b21	CBE	BE清零位	0:不清零BE标志 1:清零BE标志 注意：CBE位写1清除BE标志，读取时返回0	R/W
b20	CRTOF	RTOF清零位	0:不清零RTOF标志 1:清零RTOF标志 注意：CRTOF位写1清除RTOF标志，读取时返回0	R/W
b19	CORE	ORE标志清零位	0: 不清零ORE标志 1: 清零ORE标志 注意：CORE位写1清除ORE标志，读取时返回0	R/W
b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17	CFE	FE标志清零位	FE标志清零位 0:不清零FE标志 1:清零FE标志 注意：CFE位写1清除FE标志，读取时返回0	R/W
b16	CPE	PE标志清零位	PE标志清零位 0:不清零PE标志 1:清零PE标志 注意：CPE位写1清除PE标志，读取时返回0	R/W
b15	OVER8	UART过采样模式	UART过采样模式设定，即一位数据传输期间的基本时钟数 0: 16位 1: 8位 注意：非UART模式时OVER8位必须保持复位值 OVER8位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b14~b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	M	数据长度设定位	UART模式时，发送/接收数据长度设定位 0: 8位 1: 9位 注意：非UART模式时M位必须保持复位值 M位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10	PCE	校验使能位	UART模式时，奇偶校验使能位 0: 无校验 1: 校验 注意：智能卡模式时PCE位必须为1，时钟同步模式时PCE位必须保持复位值 PCE位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b9	PS	校验位	UART模式时，奇偶校验选择位 0: 偶校验 1: 奇校验 注意：PS位只能在TE=0&RE=0(发送/接收禁止)时设定，PS位只在PCE=1时有效	R/W
b8	TENDIE	发送完成中断使能位	发送完成中断使能位 0: TENDI中断请求无效，TENDI中断不发生	R/W

			1: TENDI中断请求使能, TENDI中断发生	
b7	TXEIE	发送数据寄存器 空中断使能位	发送数据寄存器空中断使能位 0: TI中断请求无效, TI中断不发生 1: TI中断请求有效, TI中断发生 注意: TE=0时写TXEIE=1, 则需要等待TE=1时发生TI中断 TE=1时写TXEIE=1, 则需要等待SR.TC=1时才能写入	R/W
b6	TCIE	发送完成中断使能位	发送完成中断使能位 0: TCI中断请求无效, TCI中断不发生 1: TCI中断请求使能, TCI中断发生 注意: 非智能卡模式读取有效, 智能卡模式读取为0	R/W
b5	RIE	接收中断使能位	接收中断使能位 0: 接收中断请求无效, RI和EI中断不发生 1: 接收中断请求有效, RI和EI中断发生	R/W
b4	SLME	静默模式使能位	多处理器操作时, 静默模式使能位 0: 正常模式 1: 静默模式 注意: SLME位只在UART多处理器模式时有效, 其他模式时此位必须保持复位值。	R/W
b3	TE	发送器使能位	发送器使能位 0: 发送器禁止 1: 发送器使能 注意: 时钟同步模式时TE位只能在TE=0&RE=0(发送/接收禁止)时写1。	R/W
b2	RE	接收器使能位	接收器使能位 0: 接收器禁止 1: 接收器使能 注意: 时钟同步模式时RE位只能在TE=0&RE=0(发送/接收禁止)时写1	R/W
b1	RTOIE	UART TIMEOUT中 断使能位	UART TIMEOUT中断使能位 (仅USART_1/USART_2/ USART_4/USART_5有效) 0: UART TIMEOUT中断请求无效, RTOI中断不发生 1: UART TIMEOUT中断请求有效, RTOI中断发生 非智能卡模式/非时钟同步模式读取有效, 智能卡模式/时钟同步模式读取为0	R/W
b0	RTOE	UART TIMEOUT功 能使能位	UART TIMEOUT功能使能位 (仅USART_1/USART_2/ USART_4/USART_5有效) 0: UART TIMEOUT功能禁止 1: UART TIMEOUT功能使能 非智能卡模式/非时钟同步模式读取有效, 智能卡模式/时钟同步模式读取为0	R/W

## 28.5.6 控制寄存器 2 (USART\_CR2)

USART Control Register 2

偏移地址：0x10

复位值：0x00000600

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	SBK M	SBK
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	LINE N	STOP	CLKC[1:0]	-	-	WKU PE	SBKL[1:0]	LBDL	LBDI E	BEE	BEIE	WKU PIE	MPE		

位	标记	位名	功能	读写
b31~b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17	SBKM	间隔段发送模式	间隔段发送模式选择 (仅USART_3/USART_6有效) 0: SBK位写1自动送出间隔段 1: SBK位写1后, 写0x00到USARTn_TDR.TDR寄存器, 开始发送间隔段 注意: 时钟同步模式时RE位只能在TE=0&RE=0(发送/接收禁止)时写1	R/W
b16	SBK	间隔段发送使能位	间隔段发送使能位 (仅USART_3/USART_6有效) 0: 不发送间隔段 1: 根据SBKM位设定发送间隔段 SBK SBKM 间隔段发送 0 0 不发送 0 1 不发送 1 0 SBK写1时, 自动发送间隔段, 发送完成后, SBK自动清零 1 1 写数据0x00到USARTn_TDR.TDR, 开始发送间隔段, 发送完成后, SBK自动清零	R/W
b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14	LINEN	LIN功能使能	UART模式时, LIN功能使能位 (仅USART_3/USART_6有效) 0: 禁止LIN功能 1: 使能LIN功能 注意: LINEN位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b13	STOP	停止位设定位	UART模式时, 停止位长度设定位 0: 1停止位 1: 2停止位 注意: 非UART模式时STOP位必须保持复位值 STOP位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b12~b11	CLKC[1:0]	时钟控制位	<b>USART模式</b> 00b: 时钟源为内部波特率生成器生成的时钟, 时钟不输出到USARTn_CK管脚, USARTn_CK管脚可以当作普通IO使用 01b: 时钟源为内部波特率生成器生成的时钟, 时钟输出到USARTn_CK管脚, 输出时钟频率和波特率相同 10b or 11b: 时钟源为外部输入时钟, 输入时钟的频率为波特率的16倍 (OVER8=0)或者8倍(OVER8=1) <b>时钟同步模式</b> 00b or 01b: 时钟源为内部波特率生成器生成的时钟, 输出到USARTn_CK管脚 10b or 11b: 时钟源为外部输入时钟, 输入时钟的频率和波特率相同	R/W

**智能卡模式**

00b：时钟源为内部波特率生成器生成的时钟，时钟不输出到CK管脚，CK管脚可

以当作普通IO使用

01b：时钟源为内部波特率生成器生成的时钟，时钟输出到CK管脚

10b or 11b：设定禁止

注意：CLKC[1:0]位只能在TE=0&RE=0(发送/接收禁止)时设定

b10~b9	Reserved	-	需保持复位值11b,写入时写11b	R/W
b8	WKUPE	LIN唤醒信号检测使能位	<p>LIN唤醒信号检测使能位（仅USART_3/USART_6有效）</p> <p>0: 禁止唤醒信号检测 1: 使能唤醒信号检测</p> <p>注意：仅在系统需要唤醒信号检测时，设置WKUPE为1，检测到唤醒信号后，需要软件清零WKUPE</p>	R/W
b7~b6	SBKL[1:0]	LIN间隔段低电平宽度设定位	<p>LIN间隔段低电平宽度设定位（仅USART_3/USART_6有效）</p> <p>00b: 10位 01b: 11位 10b: 13位 11b: 14位</p> <p>注意：非UART-LIN模式时SBKL位必须保持复位值 SBKL位只能在TE=0&amp;RE=0(发送/接收禁止)时设定</p>	R/W
b5	LBDL	LIN间隔段低电平宽度检测阈值设定位	<p>LIN间隔段低电平宽度检测阈值设定位（仅USART_3/USART_6有效）</p> <p>0: ≥10位 1: ≥11位</p> <p>注意：非UART-LIN模式时LBDL位必须保持复位值 LBDL位只能在TE=0&amp;RE=0(发送/接收禁止)时设定</p>	R/W
b4	LBDIE	LIN间隔段检测中断使能位	<p>LIN间隔段检测中断使能位（仅USART_3/USART_6有效）</p> <p>0: LIN间隔段检测中断请求无效，不发生中断 1: LIN间隔段检测中断请求有效，发生中断</p> <p>注意：非UART-LIN模式时LBDIE位必须保持复位值</p>	R/W
b3	BEE	LIN总线错误检测功能使能位	<p>LIN总线错误检测功能使能位（仅USART_3/USART_6有效）</p> <p>0: 禁止LIN总线错误功能检测功能 1: 使能LIN总线错误功能检测功能</p> <p>注意：非UART-LIN模式时BEE位必须保持复位值 BEE位只能在TE=0&amp;RE=0(发送/接收禁止)时设定</p>	R/W
b2	BEIE	LIN总线错误中断使能位	<p>LIN总线错误中断使能位（仅USART_3/USART_6有效）</p> <p>0: LIN总线错误中断请求无效，不发生中断 1: LIN总线错误中断请求有效，发生中断</p> <p>注意：非UART-LIN模式时LBDIE位必须保持复位值</p>	R/W
b1	WKUPIE	LIN唤醒信号检测中断使能位	<p>LIN唤醒信号检测中断使能位（仅USART_3/USART_6有效）</p> <p>0: LIN唤醒信号检测中断请求无效，不发生中断 1: LIN唤醒信号检测中断请求有效，发生中断</p> <p>注意：非UART-LIN模式WKUPIE位必须保持复位值</p>	R/W
b0	MPE	多处理器功能使能位	<p>UART模式时，多处理器功能使能位</p> <p>0: 禁止 1: 使能</p> <p>注意：非UART模式时MPE位必须保持复位值 MP位只能在TE=0&amp;RE=0(发送/接收禁止)时设定</p>	R/W

### 28.5.7 控制寄存器 3 (USART\_CR3)

USART Control Register 3

偏移地址：0x14

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	BCN[2:0]	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b15	b14
-	-	-	-	-	-	CTSE	RTSE	-	-	SCE_N	LOO_P	HDS_EL	-	-	-

位	标记	位名	功能	读写
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
			智能卡模式时, 设定一位数据传输期间的基本时钟数	
			BCN[2:0]设定值 一位数据传输期间的基本时钟数	
		000b	32	
		001b	64	
		010b	设定禁止	
		011b	128	
b23~b21	BCN[2:0]	基本时钟数	100b 设定禁止	R/W
			101b 256	
			110b 372	
		111b	设定禁止	
			注意: 非智能卡模式时BCN[2:0]位必须保持复位值	
			BCN[2:0]位只能在TE=0&RE=0(发送/接收禁止)时设定	
			USART_3/USART_6不支持该功能, 读出为0	
b20~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
			CTS功能使能位	
b9	CTSE	CTS功能使能位	0: CTS功能无效 1: CTS功能有效	R/W
			注意: CTSE位只能在TE=0&RE=0(发送/接收禁止)时设定	
			RTS功能使能位	
b8	RTSE	RTS功能使能位	0: RTS功能无效 1: RTS功能有效	R/W
			注意: RTSE位只能在TE=0&RE=0(发送/接收禁止)时设定	
b7~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
			智能卡模式使能位	
			0: 禁止智能卡模式 1: 使能智能卡模式	
b5	SCEN	智能卡模式使能位	注意: 非智能卡模式时SCEN位必须保持复位值	R/W
			SCEN位只能在TE=0&RE=0(发送/接收禁止)时设定	
			USART_3/USART_6不支持该功能, 读出为0	
b4	LOOP	LIN 回环模式使能位	LIN 回环模式使能位	
			0: 正常模式 1: 回环模式	R/W
			注意: LOOP位只能在TE=0&RE=0(发送/接收禁止)时设定	
b3	HDSEL	UART单线半双工模式	UART单线半双工模式使能位	R/W

使能位	0: UART全双工模式 1: UART半双工模式 注意: HDSEL位只能在TE=0&RE=0(发送/接收禁止)时设定
b2~b0      Reserved      -	读出时为“0”,写入时写“0”      R/W

## 28.5.8 预分频寄存器 (USART\_PR)

USART prescaler register

偏移地址：0x18

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

位	标记	位名	功能	读写
b31~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3~b2	LBMPSC[1:0]	LIN波特率测量计数器 时钟源选择	LIN波特率测量计数器 时钟源选择 (仅USART_3/USART_6有效) 00: PCLK 01: PCLK/2 10: PCLK/4 11: PCLK/8 注意: LBMPSC[1:0]位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b1~b0	PSC[1:0]	预分频器值	内部时钟源时, 预分频器分频值选择位 00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64 注意: PSC[1:0]位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W

### 28.5.9 LIN 波特率测量计数寄存器 (USART\_LBMC)

USART LIN Baudrate Measuring Counter

偏移地址：0x1C

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LBMC[15:0]															

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b0	LBMC[15:0]	LIN波特率测量结果计数器	LIN波特率测量结果计数器（仅USART_3/USART_6有效） LIN从节点，用USARTn_PR.LBMPSC寄存器选择的时钟源去测量得到的主节点发送的同步段的频率。 USARTn_PR.LBMPSC选择的计数时钟频率除以LBMC的值，即可得到测量到的LIN通信波特率。 注意：LBMC的值在同步段接收完成后读取才有意义	R

## 28.5.10 USART1 滤波控制寄存器 (USART1\_NFC)

USART1 Noise Filting Control Counter

地址: 0x4005541C

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

-	-	-	-	-	-	-	-	-	-	-	-	-	-	USA RT1 NFE	USASRT1_NFS [1:0]
---	---	---	---	---	---	---	---	---	---	---	---	---	---	-------------------	----------------------

位	标记	位名	功能	读写
b31~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	USART1_NFE	USART1滤波使能寄存器	USART_1滤波使能寄存器 该寄存器用于控制STOP模式下，RX线上的模拟滤波器的开关 0: 模拟滤波器关 1: 模拟滤波器开，滤波范围请参考USART1_NFS[1:0]的设定	R/W
b1~b0	USART1_NFS	USART1滤波选择寄存器	USART_1滤波选择寄存器 该寄存器用于控制STOP模式下，RX线上的模拟滤波器的滤波范围 00b: 滤波宽度档位1 01b: 滤波宽度档位2 10b: 滤波宽度档位3 11b: 滤波宽度档位4 各档位具体值请参考“电气特性 USART1 STOP模式下RX滤波特性”章节。	R/W

## 28.6 使用注意事项

### 28.6.1 UART 注意事项

#### 发送器

UART 模式发送器发送禁止时(USARTn\_CR1.TE=0)，则 TX 管脚可以当作普通 IO 使用，可以设定输出值和方向。如果输出 0，则会使接收方产生帧错误，从而中断数据传输。如果输出 1，则使接收方检测不到开始位从而无法开始数据传输。

#### 接收器

UART 模式产生帧错误时，可以软件检测后续 RX 线是否为低电平，从而判断发送方是否想中断传输。

如果接收数据开始位检测方式为低电平检测，则在清除错误标志后继续接收全为低电平数据，接收错误会再次发生。

### 28.6.2 时钟同步模式注意事项

- 使用外部输入时钟发送数据时，USARTn\_TDR.TDR 的更新需要在时钟输入之前完成，写入数据后，至少需要等待一位数据时间再输入时钟。
- 连续发送数据时，下一帧数据需要在当前帧最后一位发送前完成更新。

### 28.6.3 其他注意事项

- 为了防止发送禁止时 TX 通信线 Hi-Z 状态，可以采用以下方法：
  - 通信线上拉
  - 发送数据结束时，USARTn\_CR1.TE=0 之前，将 TX 管脚设为普通 IO 输出
  - 发送数据开始前，USARTn\_CR1.TE=1 之后，将 IO 设为 TX 功能

## 29 集成电路总线（I2C）

### 29.1 简介

I2C（集成电路总线）用作微控制器和 I2C 串行总线之间的接口。提供多主模式功能，可以控制所有 I2C 总线的协议、仲裁。支持标准模式、快速模式。还支持 SMBus 总线。

#### I2C 主要特性：

- I2C 总线方式、SMBUS 总线方式可选。主机模式、从机模式可选。自动确保与传送速率相对于的各种准备时间、保持时间和总线空闲时间。
- 标准模式最大 100Kbps、快速模式最大 400Kbps、FM+模式最大 1Msps。
- 自动生成开始条件、重新开始条件和停止条件，并能检测到总线的开始条件，重新开始条件和停止条件。
- 最大支持 128 个从机模式地址。可同时设定 7 位地址格式和 10 位地址格式。能检测到广播呼叫地址、SMBus 主机地址、SMBus 设备默认地址、SMBus 报警地址。
- 2 级发送 FIFO 和 2 级接收 FIFO。
- 发送时可以自动判定应答位。接收时可以自动发送应答位。
- 握手功能。
- 仲裁功能。
- 超时功能，可以检测 SCL 时钟长时间停止。
- SCL 输入和 SDA 输入内置数字滤波器，滤波能力可编程。
- 通信错误，接收数据满，发送数据空，一帧发送结束，地址匹配一致中断。

## 29.2 I2C 系统框图

### 29.2.1 系统框图

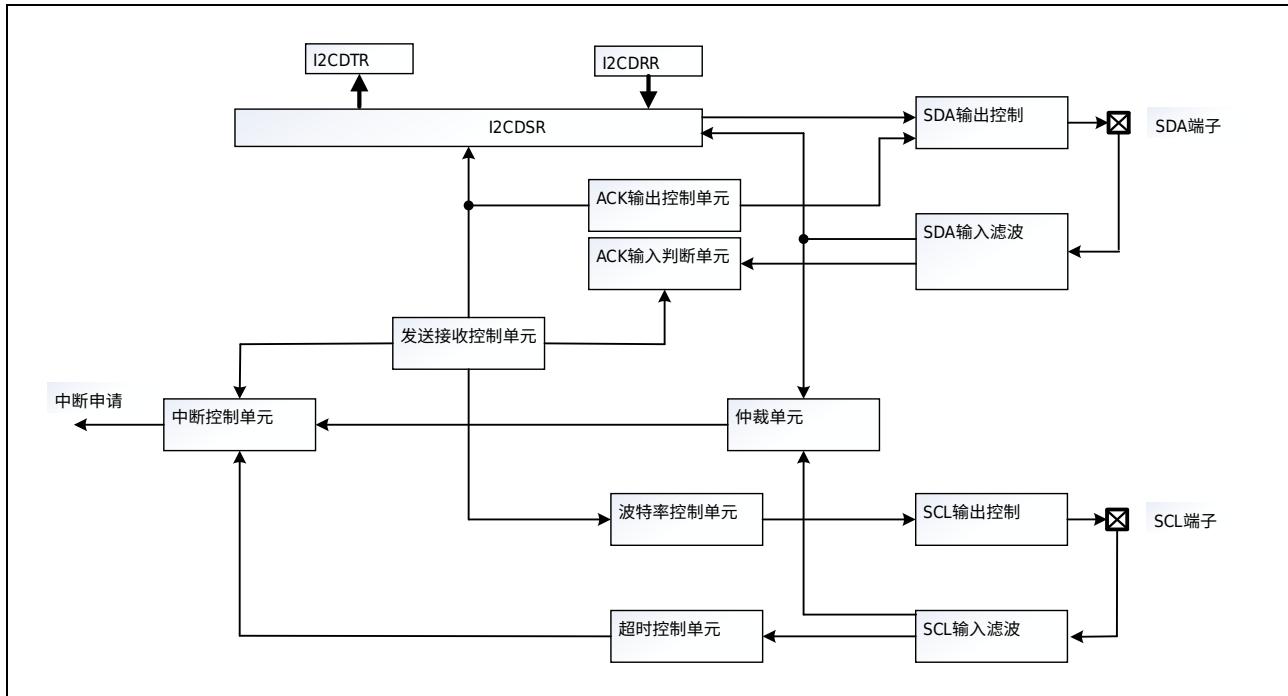


图 29-1 I2C 系统框图

## 29.2.2 结构图

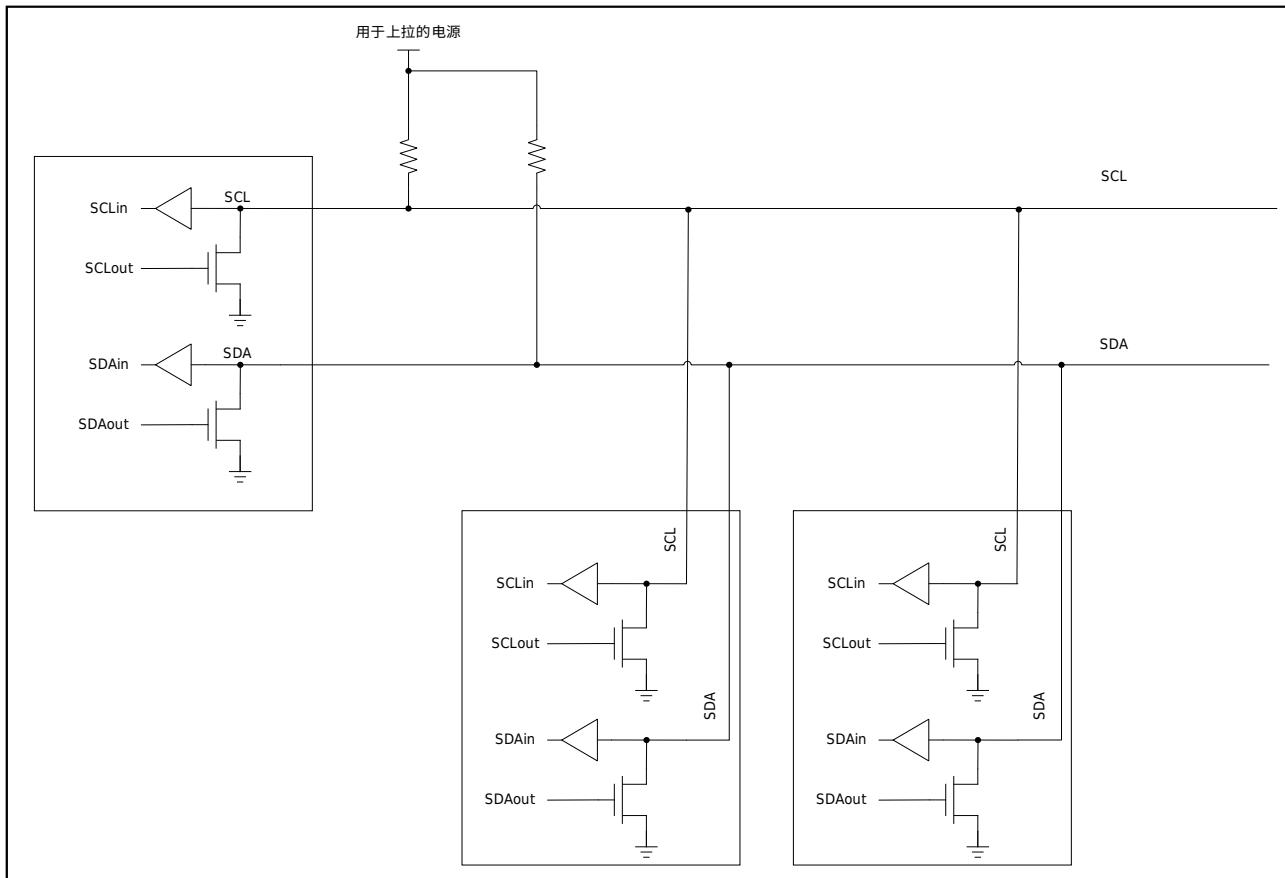


图 29-2 I<sup>2</sup>C 总线的结构例

表 29-1 输入/输出引脚

引脚名	输入/输出	功能
SCL	输入/输出	时钟输入/输出引脚
SDA	输入/输出	数据输入/输出引脚

当选择 I<sup>2</sup>C 总线时，SCL/SDA 输入电平为 Schmitt 电平（兼容 CMOS 电平）。当选择 SMBus 时，请通过设定【通用 IO (GPIO)】章节中的 PCRxy 寄存器 (PCRxy.CINSEL) 将 SCL/SDA 输入电平设定为 CMOS 电平。

## 29.3 动作说明

本节提供了 I<sup>2</sup>C 模块功能的描述。

### 29.3.1 I<sup>2</sup>C 协议

I<sup>2</sup>C 总线由一根时钟线 (SCL)，一根数据线 (SDA) 构成。所有的连接器件必须是漏极开路输出。SCL，SDA 线外接上拉电阻。电阻阻值取决于系统应用。

通常情况下，一个完整的通信过程包括下列 4 部分：

1. 开始条件
2. 地址传送
3. 数据传送
4. 停止条件

下图是 I<sup>2</sup>C 总线的时序图。

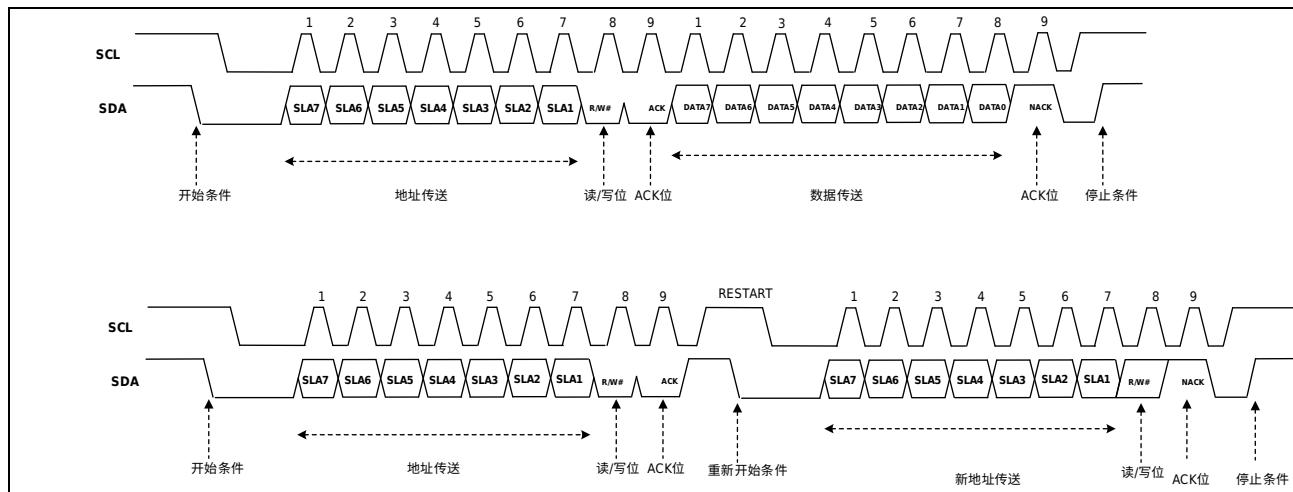


图 29-3 I<sup>2</sup>C 总线的时序图

#### 29.3.1.1 开始条件

当总线上的主机都不驱动总线，总线进入空闲状态。SCL 和 SDA 都为高电平。总线空闲状态下总线上设备都可以通过发送开始条件启动通信。

在 I<sup>2</sup>C\_SR.BUSY 标志为“0”（总线空闲）的状态下如果将 START 位置“1”，就发行开始条件。如果检测到开始条件，就自动将 I<sup>2</sup>C\_SR.BUSY 标志和 I<sup>2</sup>C\_SR.STARTF 标志置“1”，并且自动将 START 位置“0”。此时，如果在 START 位为“1”的状态下发送的 SDA 信号和 SDA 线的信号状态相同，并且检测到开始条件，就视为通过 START 位正确地发行了开始条件，在将 I<sup>2</sup>C\_SR.MSL 位和 I<sup>2</sup>C\_SR.TRA 位自动置“1”后变为主控发送模式。另外，I<sup>2</sup>C\_SR.TEMPTYF 因 TRA 位为“1”而自动变为“1”。接下来把从机地址写入 I<sup>2</sup>C\_DTR 寄存器，发送地址。

### 29.3.1.2 地址传送

开始条件或者重新开始条件后面的帧是地址帧，用于指定主机通信的对象地址。在发送停止条件之前，指定的从机一直有效。

地址帧的高 7 位为从机地址。地址帧第 8 位来决定数据帧传送的方向。

- 7 位寻址模式见下图[7 位地址格式]

主机发送模式，主机发送地址帧第 8 位为 0

主机接收模式，主机发送地址帧第 8 位为 1

- 10 位寻址模式见下图[10 位地址格式]

主机发送模式，主机第一帧发送头序列（11110XX0，其中 XX 表示 10 位地址的高两位），然后第二帧发送低八位从机地址。

主机接受模式，主机第一帧发送头序列（11110XX0，其中 XX 表示 10 位地址的高两位），然后第二帧发送低八位从机地址。接下来会发送一个重新开始条件，然后再发送一帧头序列（11110XX1，其中 XX 表示 10 位地址的高两位）。

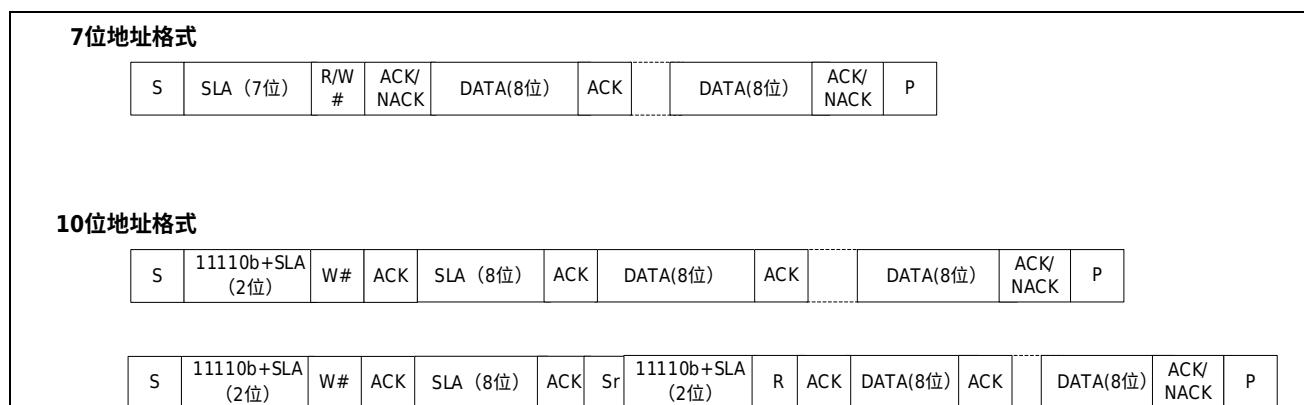


图 29-4 I2C 总线的数据格式

S : 表示开始条件。

SLA : 表示从机地址。

R/W# : 表示发送和接收的方向。当 R/W# 为“1”时，将数据从从机发送到主机；当 R/W# 为“0”时，将数据从主机发送到从机。

Sr : 表示重新开始条件。

DATA : 表示发送和接收的数据。

P : 表示停止条件。

### 29.3.1.3 数据传送

地址匹配一致后，总线上的主机根据 R/W 定义的方向一帧一帧的传送数据。

所有的地址帧后传送的数据都视为数据帧。即使是 10 位地址格式的低 8 位地址也视为数据帧。

数据帧的长度是 8 位。SCL 的低电平 SDA 变化，SCL 的高电平 SDA 保持，每个时钟周期发送一位数据。数据帧后的第 9 个时钟是应答位，是接收方向发送方传送的握手信号。

如果总线上从机接收数据，在第 9 个时钟周期不响应主机，从机必须发送 NACK。如果总线上主机接收数据，第 9 个周期发送 NACK，从机接收到 NACK，从机停止发送数据。

无论主机还是从机发送了 NACK，数据传送终止。主机可以做下列任一动作：

- 发送停止条件释放总线
- 发送重新开始条件开始一个新的通信。

#### 主机发送数据

在主机发送模式中，主机输出 SCL 时钟和发送数据，从机接收数据并返回应答。主机发送数据运行时序例如下图所示。

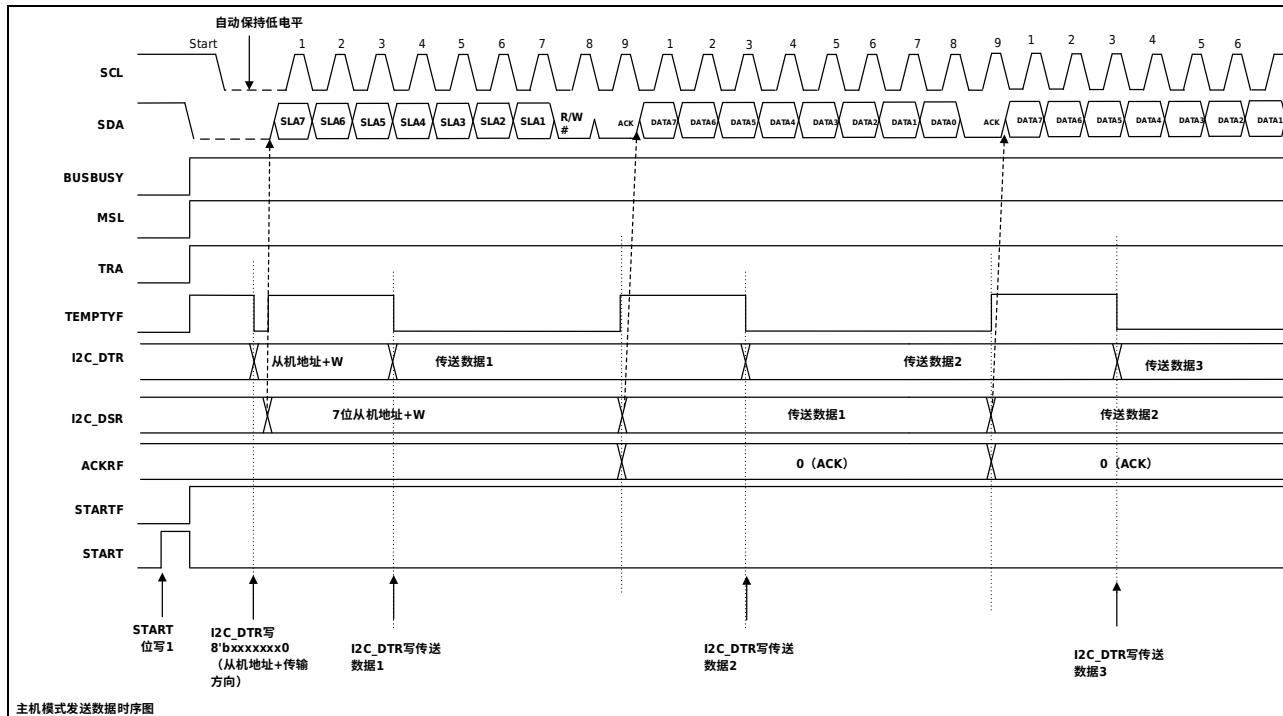


图 29-5 7 位地址格式的主机发送数据时序图（例）

## 主机接收数据

在主机接收模式中，主机输出 SCL 时钟，接收从机数据并返回应答。主机接收数据的运行时序例如下图所示。

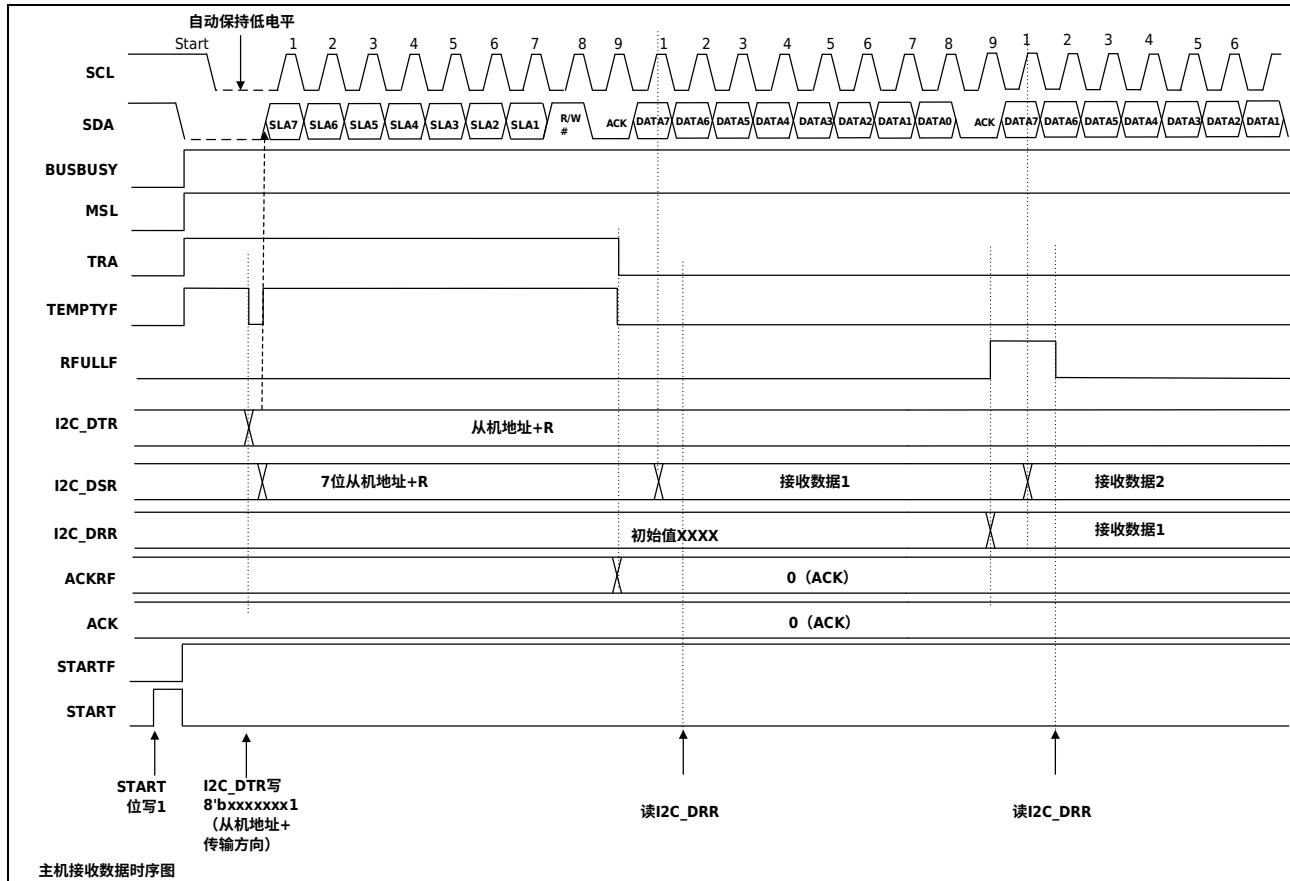


图 29-6 7 位地址格式的主机接收数据的时序图（例）

## 从机发送数据

在从机发送模式中，接收来自主机的 SCL 时钟，本产品为从机发送数据，并且接收主机返回应答。从机发送数据的运行时序例如下图所示。

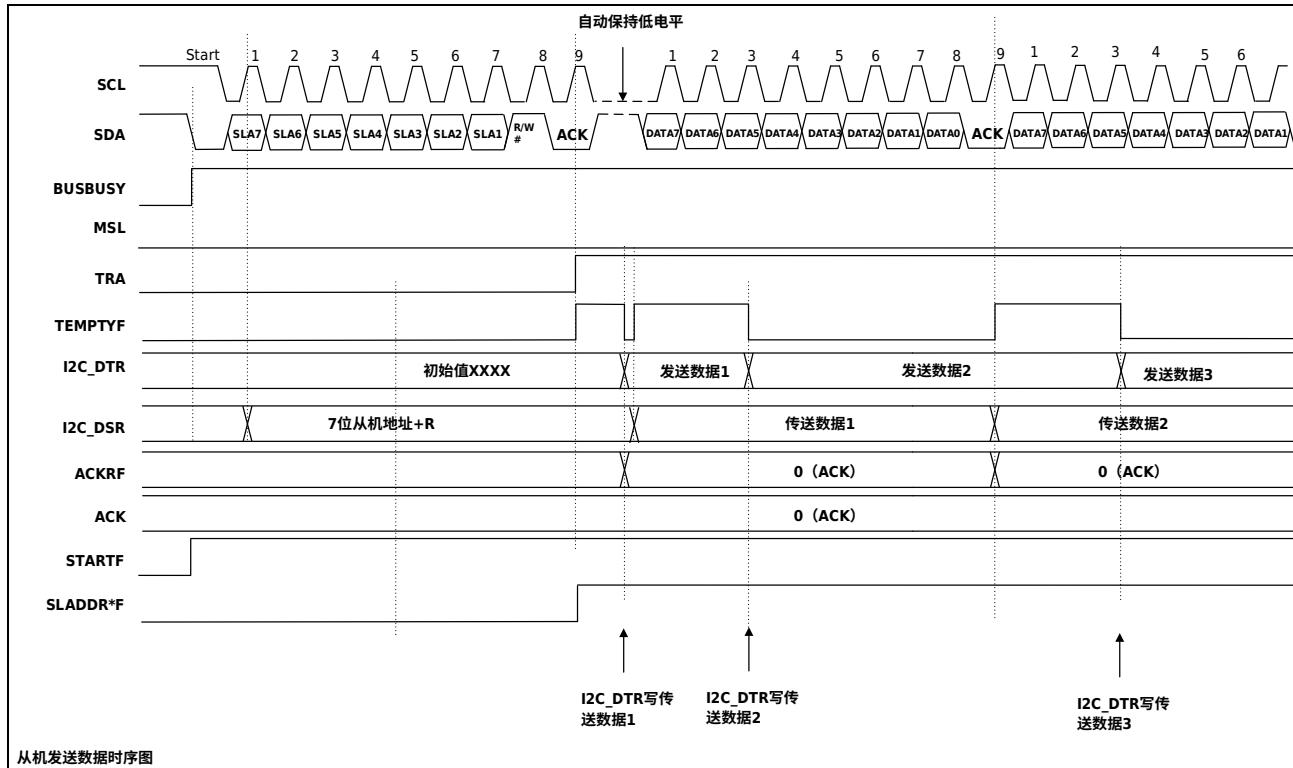


图 29-7 7 位地址格式的从机发送模式时序图（例）

## 从机接收数据

在从机接收模式中，接收来自主机的 SCL 时钟和数据，接收完数据后返回应答。从机接收数据的运行时序例如下图所示。

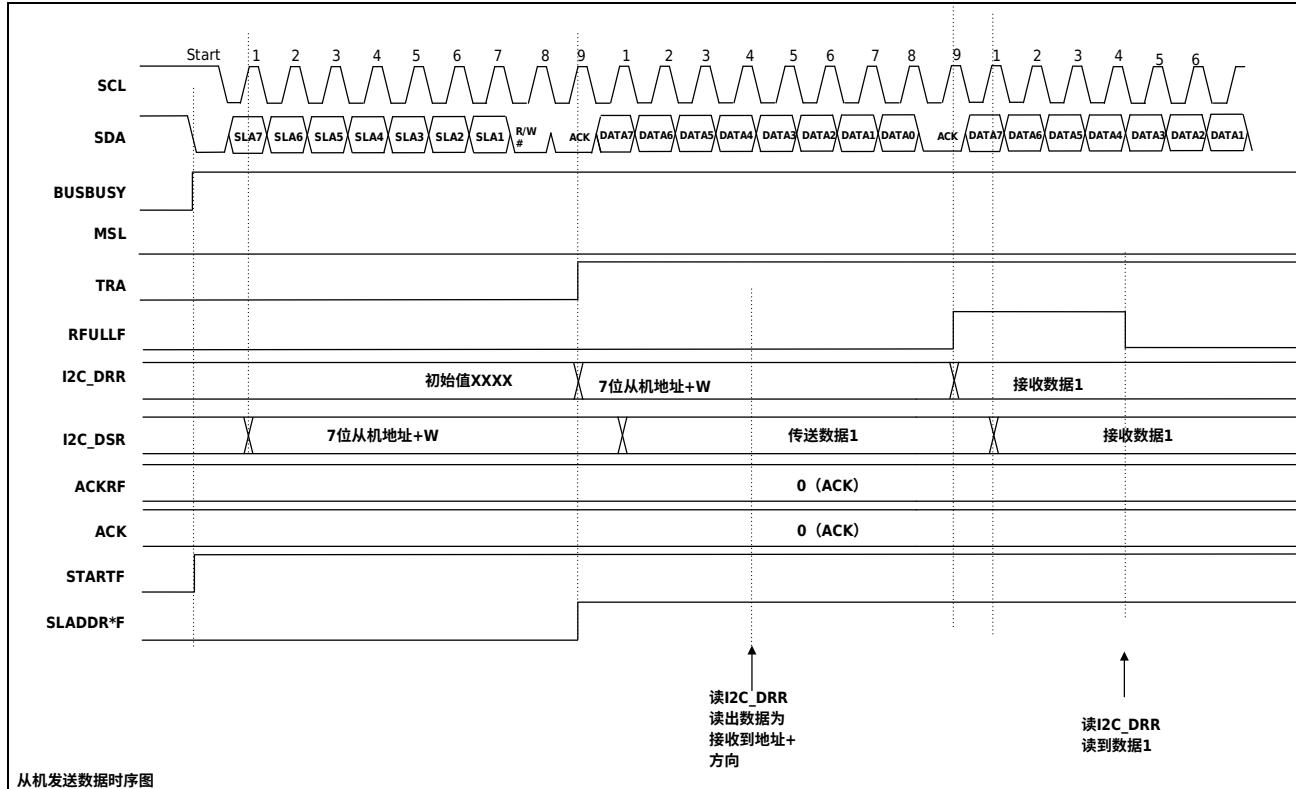


图 29-8 7 位地址格式从机接收模式时序图（例）

### 29.3.1.4 停止条件

通过 I2C\_CR1.STOP 位发行停止条件。

在 I2C\_SR.BUSY 标志为“1”（总线忙）并且 I2C\_SR.MSL 位为“1”（主机模式）的状态下将 STOP 位置“1”，就发行停止条件。

### 29.3.1.5 重新开始条件

通过 I2C\_CR1.RSTART 位生成重新开始条件。

在 I2C\_SR.BUSY 标志为“1”（总线忙）并且 I2C\_SR.MSL 位为“1”（主机模式）的状态下 RSTART 位置“1”，就生成行重新开始条件。

通过重新开始条件，主机可以在不释放 BUS 权的情况下，切换发送/接收模式。也可以在不释放 BUS 权的情况下和另一个从机建立通信。

### 29.3.1.6 SCL 时钟同步

在多主机模式中使用 I2C 总线时，有可能因和其他主机的竞争而使 SCL 时钟和 SCL 时钟发生冲突。如果 SCL 时钟发生冲突，主机就需要与 SCL 时钟同步，而且需要逐位进行 SCL 时钟的同步。当检测到 SCL 线的上升沿并且在对 I2C\_CCR.SHIGHW 寄存器设定的高电平进行计数的过程中，如果因其他主机的 SCL 时钟输出而使 SCL 线下降，就在检测到 SCL 线的下降沿时中止高电平宽度的递增计数，并且在 SCL 线被驱动为低电平的同时开始对 I2C\_CCR.SLOWW 设定的低电平宽度进行递增计数，在结束低电平宽度的计数时结束 SCL 线的低电平驱动，并释放 SCL 线。此时，如果其他主机的 SCL 时钟的低电平宽度大于 SLOWW 设定的低电平宽度，就延长 SCL 时钟的低电平宽度。当其他主机结束低电平输出时，释放 SCL 线并且 SCL 时钟上升。因此，在发生 SCL 时钟输出冲突时，SCL 时钟的高电平宽度与短时钟同步，低电平宽度与长时钟同步。

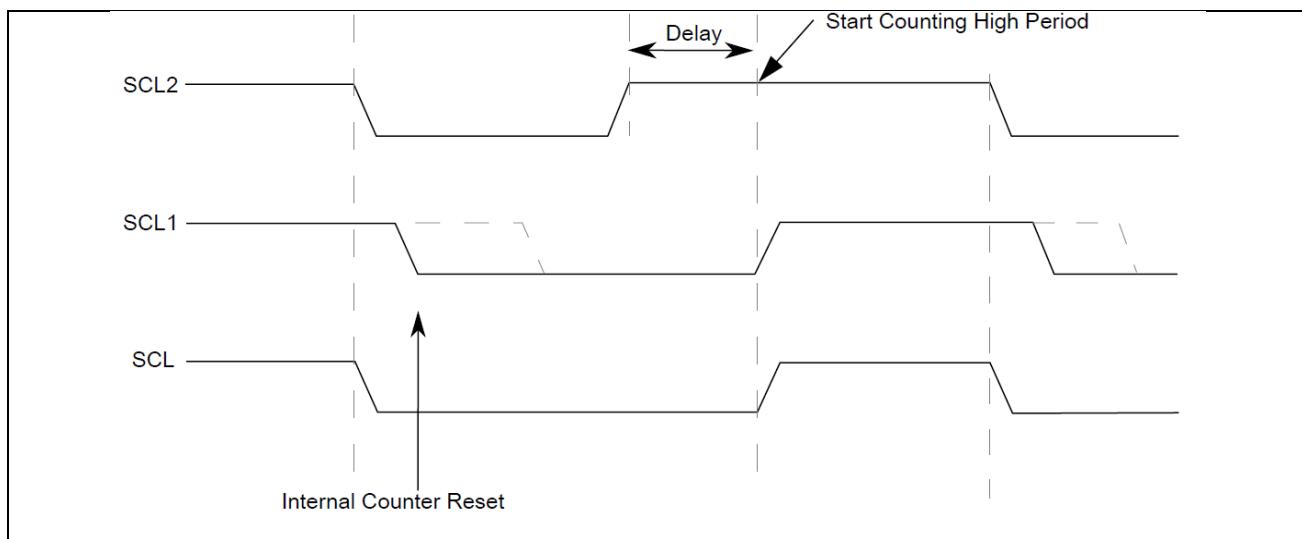


图 29-9 SCL 同步时序

### 29.3.1.7 仲裁

I2C 总线是一个真正的多主机总线，允许多个主机连接。

如果两个或两个以上的主机试图同时控制总线，SCL 时钟同步过程决定了总线时钟。总线时钟的低周期取决于最长的低电平时钟，高周期取决于最短的高电平时钟。高电平采集到的数据决定了仲裁结果。发送的 SDA 输出为高电平输出（SDA 引脚为高阻抗状态）而检测到 SDA 线为低电平时，就产生仲裁失败。I2C\_SR.AROLF 位会硬件置“1”。如果发生主机仲裁失败，就立即转移到从机接收模式。此时，如果包括广播地址在内的从机地址匹配，就继续从机模式的运行。

### 29.3.1.8 握手

数据传送过程中通过 SCL 时钟同步机制实现握手。从机在传送完一帧数据后（包含 ACK 位），将 SCL 时钟线维持在低电平。在这种情况下，SCL 时钟的低电平让主机进入等待状态，直到从机释放 SCL 线。

【从机发送模式】

- 在发送模式中 (I2C\_SR.TRA 位=1)，如果移位寄存器 (I2C\_DSR 寄存器) 为空状态并且未写发送数据 (I2CDTR 寄存器)，就在第 9 时钟和下次传送的第 1 个时钟的低电平区间自动保持 SCL 线的低电平，动作时序如下图所示。

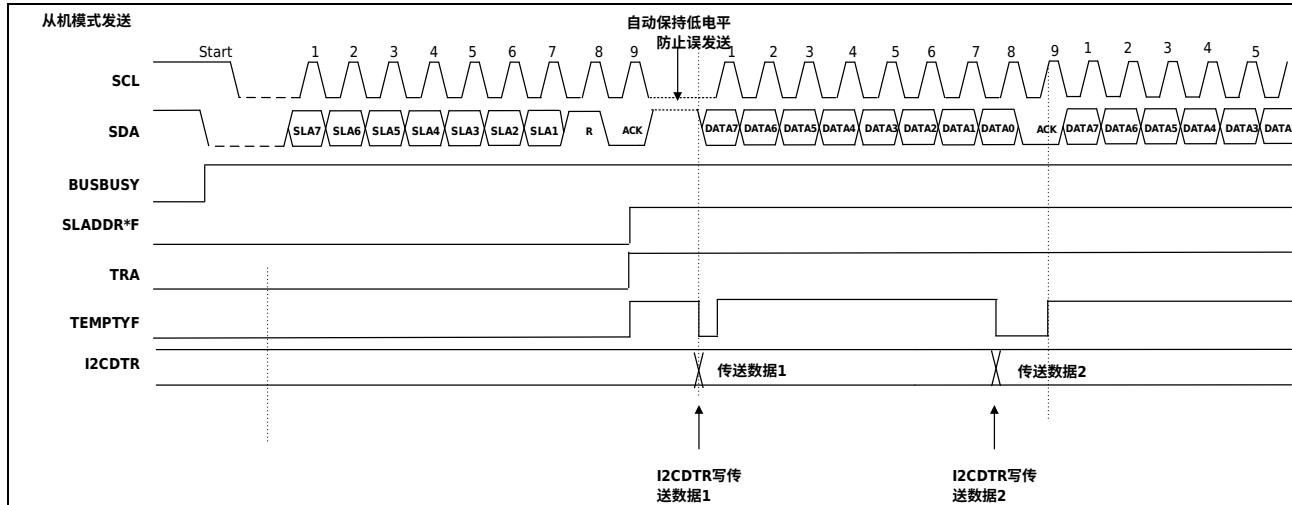


图 29-10 从机发送时序图 (1)

- 在 I2C\_SR.NACKF 标志变为“1”或者将最后的发送数据写到 I2C\_DTR 寄存器后，在 I2C\_SR.TEMPTYF 标志为“1”的状态下等到 I2C\_SR.TENDF 标志变为“1”。当 I2C\_SR.NACKF 标志或者 TENDF 标志为“1”时，在第 9 个时钟下降后将 SCL 线保持为低电平。此时必须通过读 I2C\_DRR 寄存器来结束通信，从而释放 SCL 线。

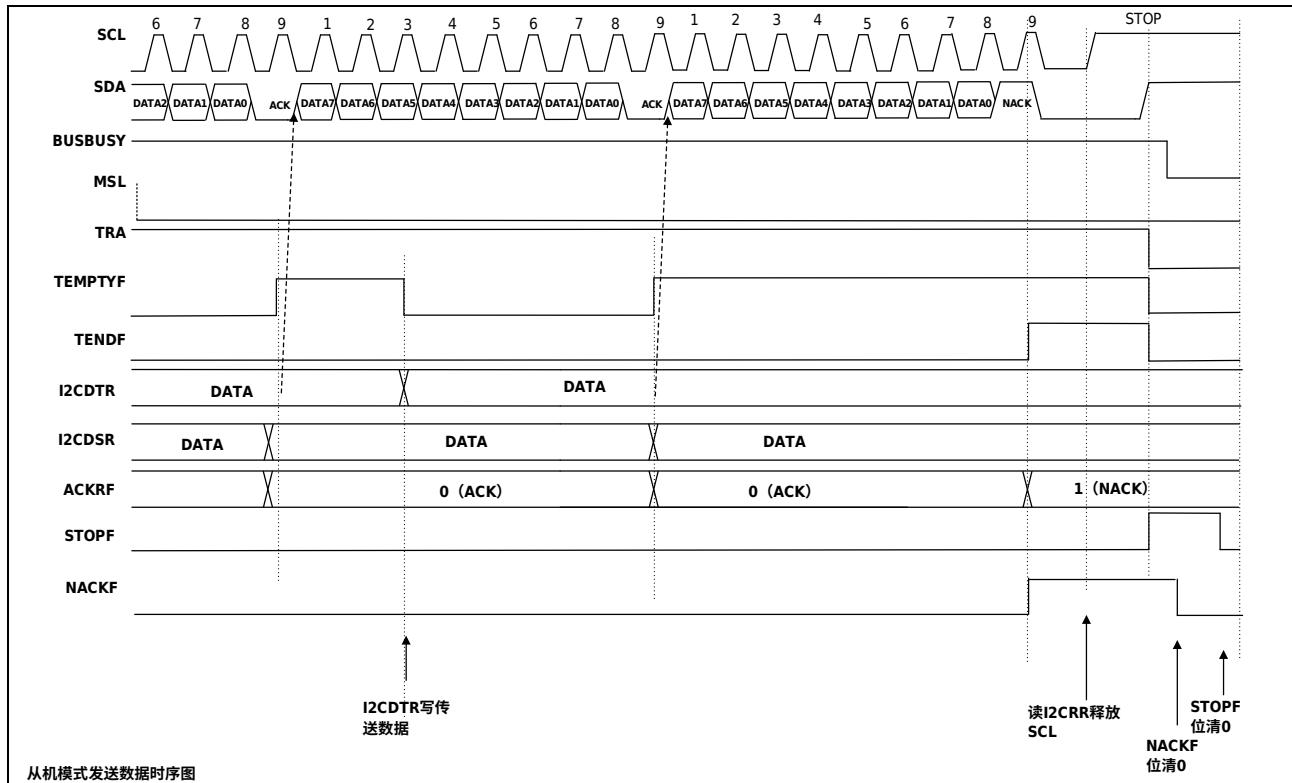


图 29-11 从机发送时序图 (2)

### 【从机接收模式】

如果在接收模式中 (I2C\_SR.TRA 位=0) 并且接收数据满 (I2C\_SR.RFULLF 标志=1) 的状态下，因至少推迟 1 个传送帧读接收数据 (I2C\_DRR 寄存器) 等而发生响应处理延迟，就在开始下一个数据接收前，在第 8 个 SCL 和第 9 个 SCL 时钟之间自动保持 SCL 线的低电平，动作时序如下图所示。

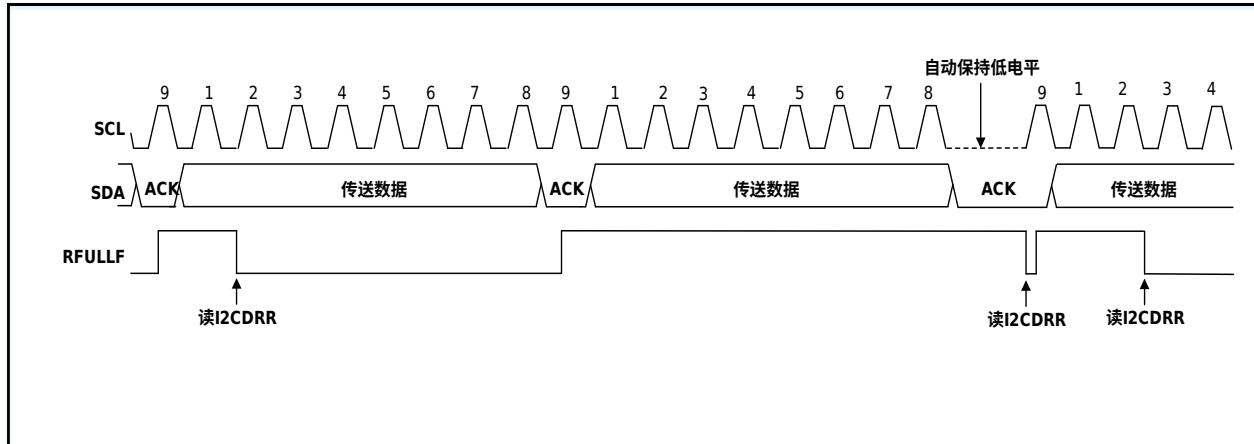


图 29-12 从机接收时序图

### 【快速 ACK/NACK】

SMBUS 通信中，利用系统内置的 CRC 运算器，计算 SMBUS 的数据包错误码（PEC）或者检查接收的数据。在检查 PEC 码的过程中，在最后字节根据是否匹配发送 ACK 或者 NACK。这就必须在接收的最后字节的 SCL 的第 8 个时钟的下降沿将 SCL 保持低电平。以此来满足软件处理时间。软件根据计算结果，写 I2C\_CR1.ACK 位来解除 SCL 低电平。快速 ACK/NACK 通过 I2C\_CR3.FACKEN 位控制，动作时序如下图所示。

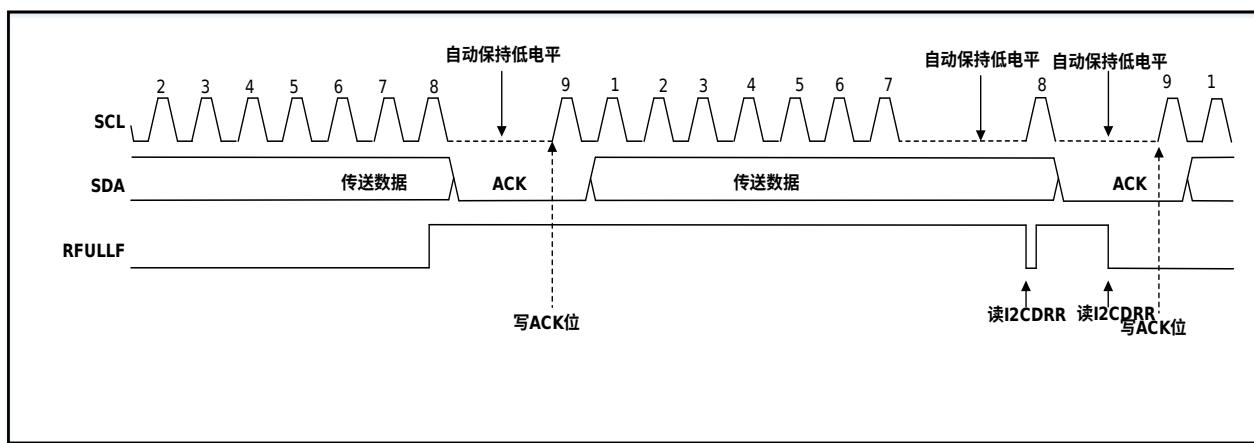


图 29-13 快速 ACK/NACK 时序图

## 29.3.2 地址匹配

作为从机，可以设定除广播地址和主机通知地址以外的 2 种地址，从机地址能设定 7 位地址或者 10 位地址格式。

### 29.3.2.1 从机地址匹配

本 I2C 总线能设定 2 种从机地址，有分别对应的从机地址检测功能。当 SLADDR1EN、SLADDR0EN 为“1”时，能检测到 I2C\_SLR1、I2C\_SLR0 寄存器设定的从属地址。

如果设定的从机地址匹配一致，就在 SCL 时钟的第 9 个时钟的下降沿将对应的 SLADDR1F、SLADDR0F 置“1”，然后根据后续的 R/W# 位将 I2C\_SR.RFULLF 标志或者 I2C\_SR.TEMPTYF 标志置“1”。由此，就能产生接收数据满中断或者发送数据空中断，并且能通过确认 I2C\_SR.SLADDR1F、SLADDR0F 标志判断指定了哪个从机地址。

I2C\_SR.SLADDR1F、SLADDR0F 标志变为“1”的时序分别如下图所示。

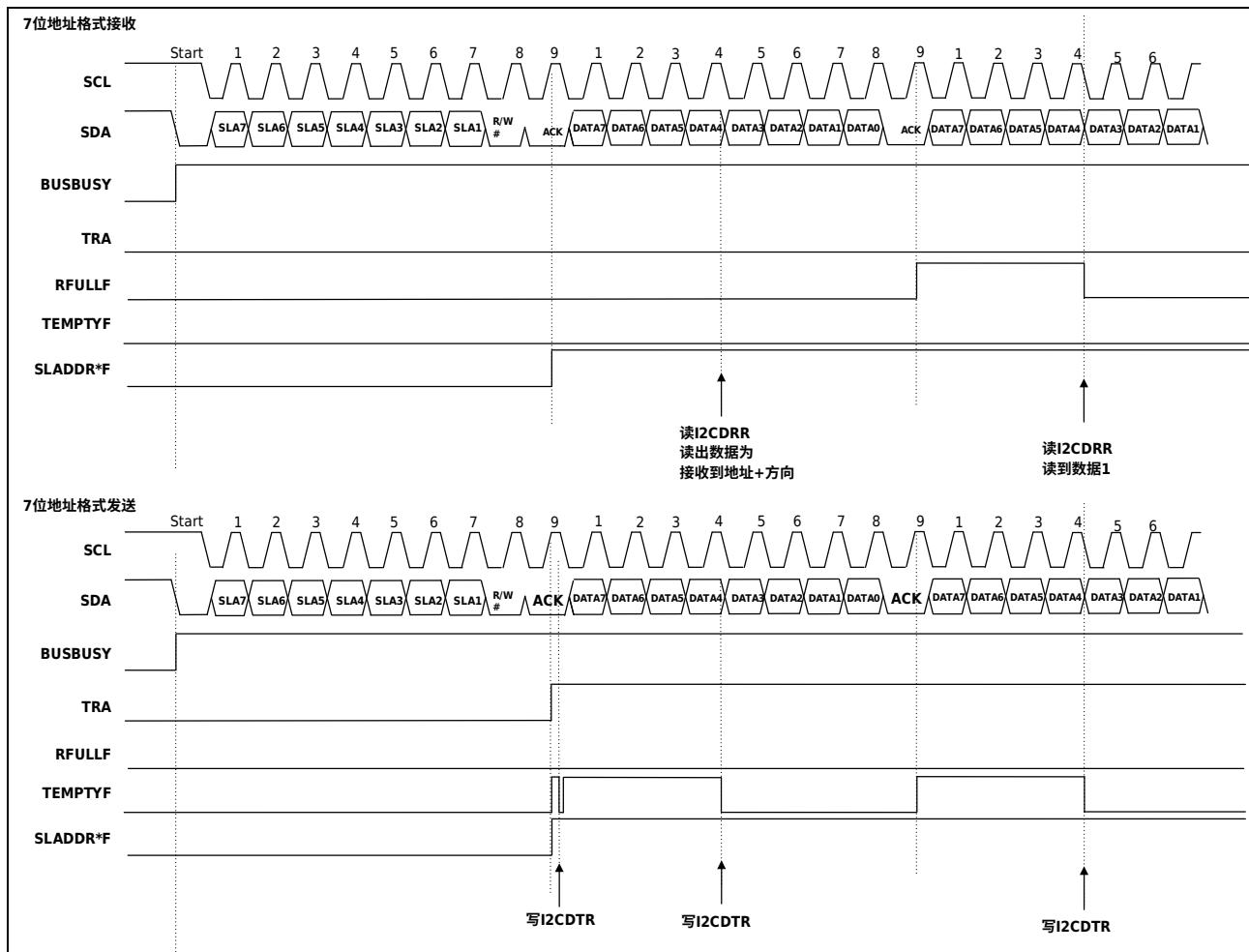


图 29-14 选择 7 位地址格式时的时序

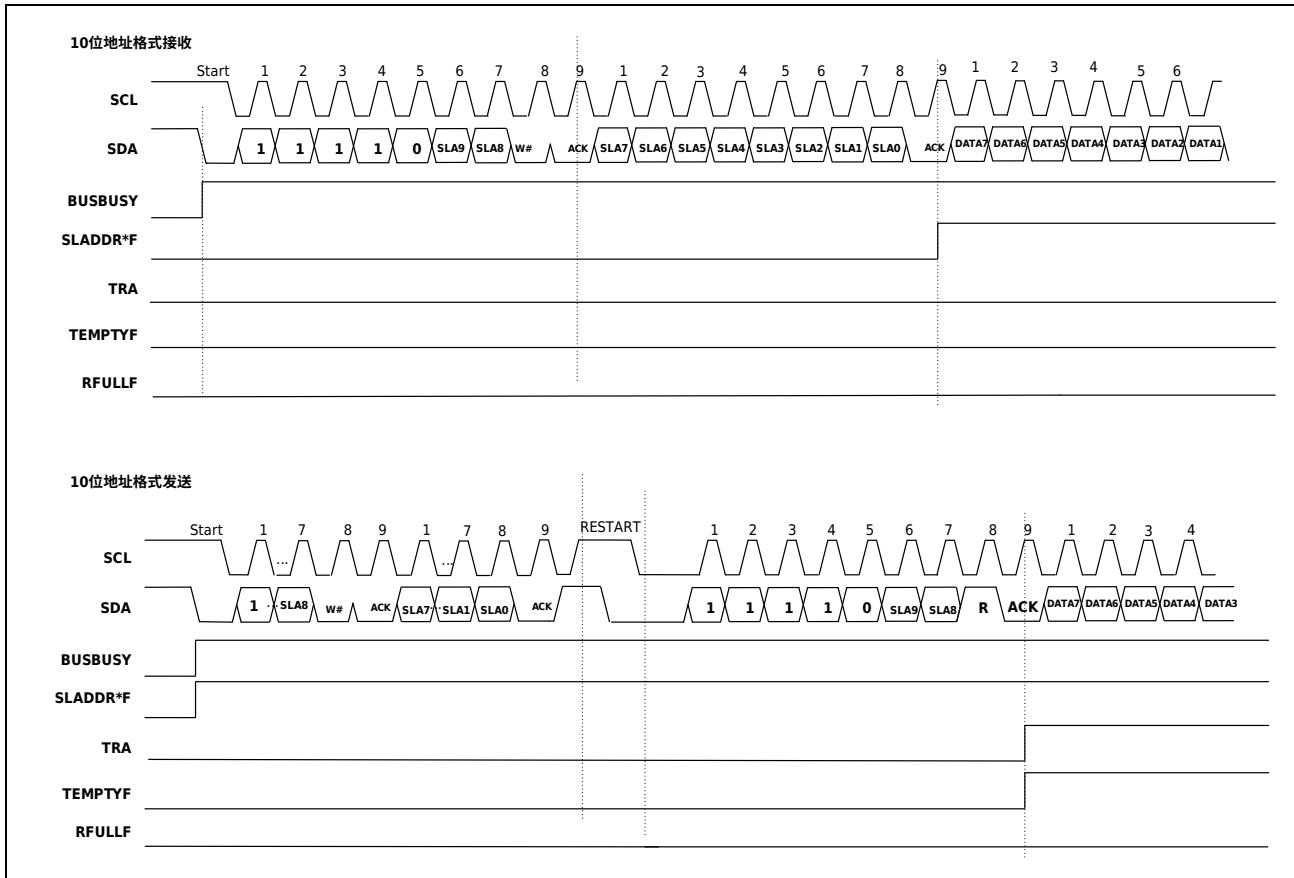


图 29-15 选择 10 位地址格式时的时序

### 29.3.2.2 广播地址匹配

当 I2C\_CR1.GCEN 位为“1”时，能检测广播地址 (0000 000b+0[W])。

但是开始条件或者重新开始条件后的地址为 0000 000b+1[R] (开始字节)，就将此地址视为 All“0”的从机地址而不视为广播地址。

如果匹配到广播地址，就在 SCL 时钟的第 9 个时钟的下降沿将 I2C\_SR.GCF 标志置“1”。

广播地址匹配一致后的运行和普通的从机接收运行相同。

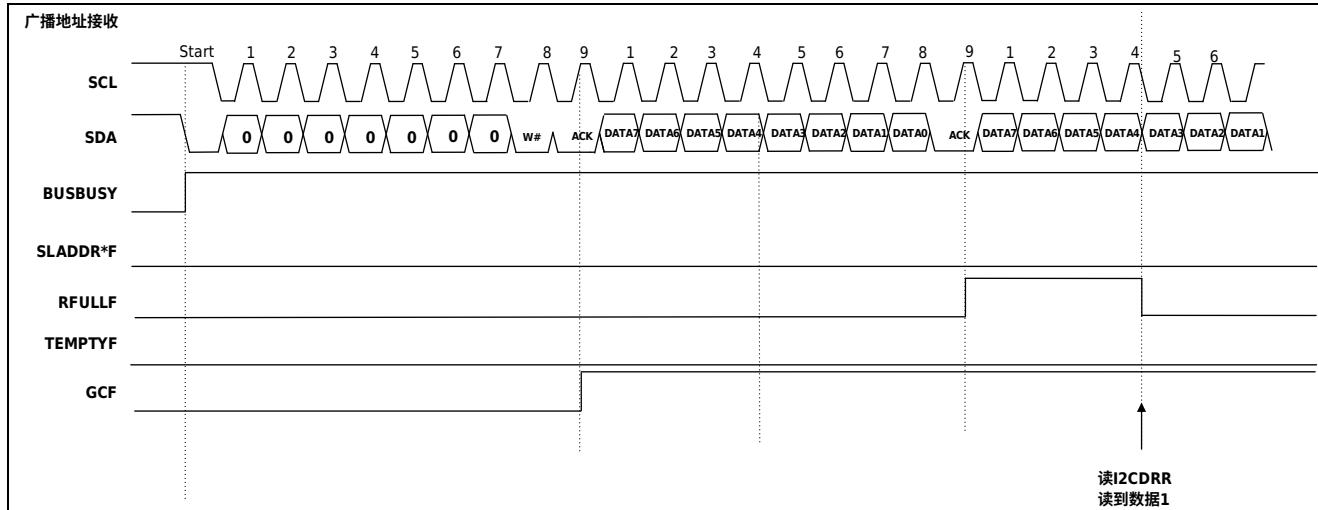


图 29-16 广播地址匹配时序图

### 29.3.2.3 SMBus 主机地址匹配

本产品具有 SMBus 运行时的主机地址检测功能。如果在 I2C\_CR1.SMBUS 位为“1”时将 I2C\_CR1.SMBHOSTEN 位置“1”，就能在从机接收模式 (I2C\_CR1.MSL 位 TRA 位为“00b”) 中检测主机地址 (0001 000b)。

如果检测到 SMBUS 主机地址，就在 SCL 时钟的第 9 个时钟的下降沿将 I2C\_SR.SMBHOSTF 标志置“1”。

即使接在 SMBUS 主机地址 (0001 000b) 后面的位是 Rd 位 (R/W# 位接收到“1”)，也能检测 SMBUS 主机地址。SMBUS 主机地址检测后的运行和普通的从机模式运行相同。

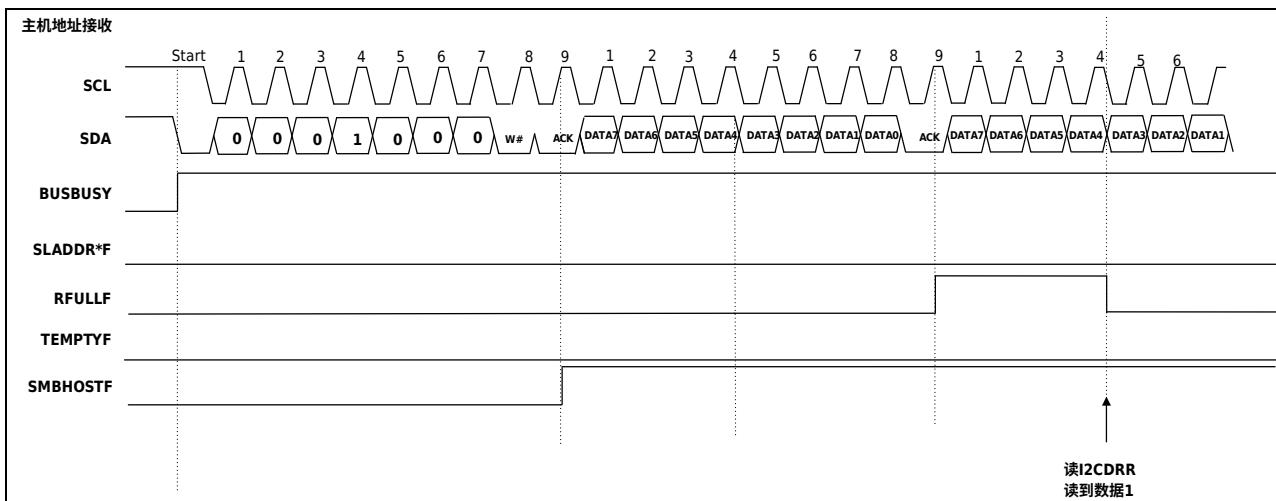


图 29-17 SMBus 主机地址匹配时序图

### 29.3.2.4 SMBus 报警响应地址匹配

本产品具有 SMBus 运行时的报警响应地址检测功能。如果在 I2C\_CR1.SMBUS 位为“1”时将 I2C\_CR1.SMBARLERTEN 位置“1”，就能在从机接收模式（I2C\_CR1.MSL 位 TRA 位为“00b”）中检测 SMBUS 报警响应地址（0001 100b）。

如果检测到 SMBUS 报警响应地址，就在 SCL 时钟的第 9 个时钟的下降沿将 I2C\_SR.SMBARLERTF 标志置“1”。

SMBUS 报警响应地址检测后的运行和普通的从机模式运行相同。

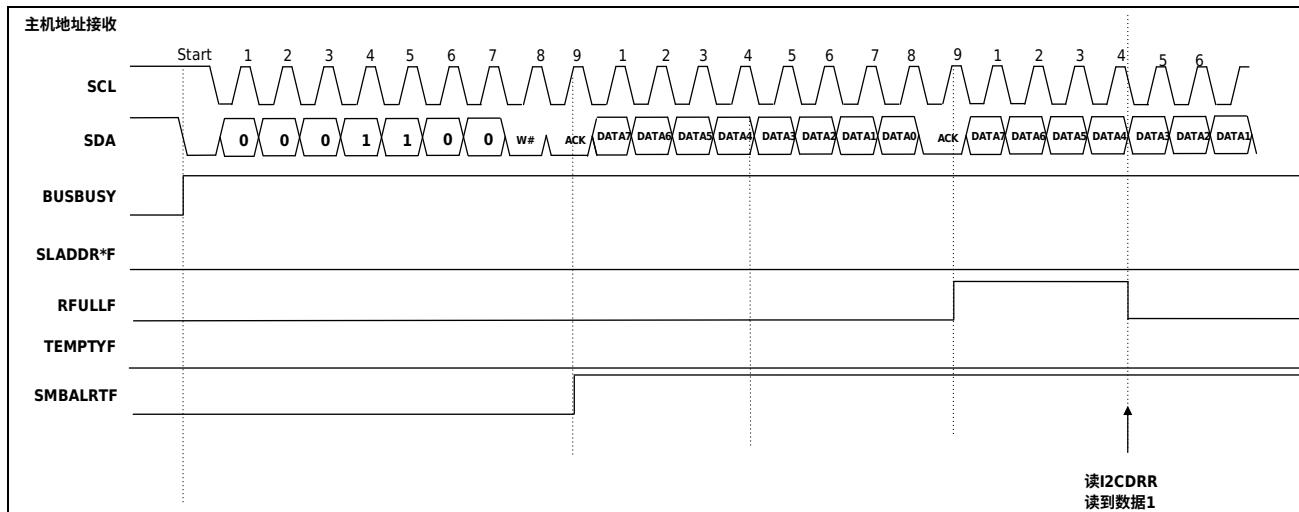


图 29-18 SMBus 报警响应地址匹配时序图

### 29.3.2.5 SMBus 默认地址匹配

本产品具有 SMBus 运行时的默认地址检测功能。如果在 I2C\_CR1.SMBUS 位为“1”时将 I2C\_CR1.SMBDEFAULTEN 位置“1”，就能在从机接收模式（I2C\_CR1.MSL 位 TRA 位为“00b”）中检测 SMBUS 默认地址（1100 001b）。

如果检测到 SMBUS 默认地址，就在 SCL 时钟的第 9 个时钟的下降沿将 I2C\_SR.SMBDEFAULTF 标志置“1”。

SMBUS 默认地址检测后的运行和普通的从机模式运行相同。

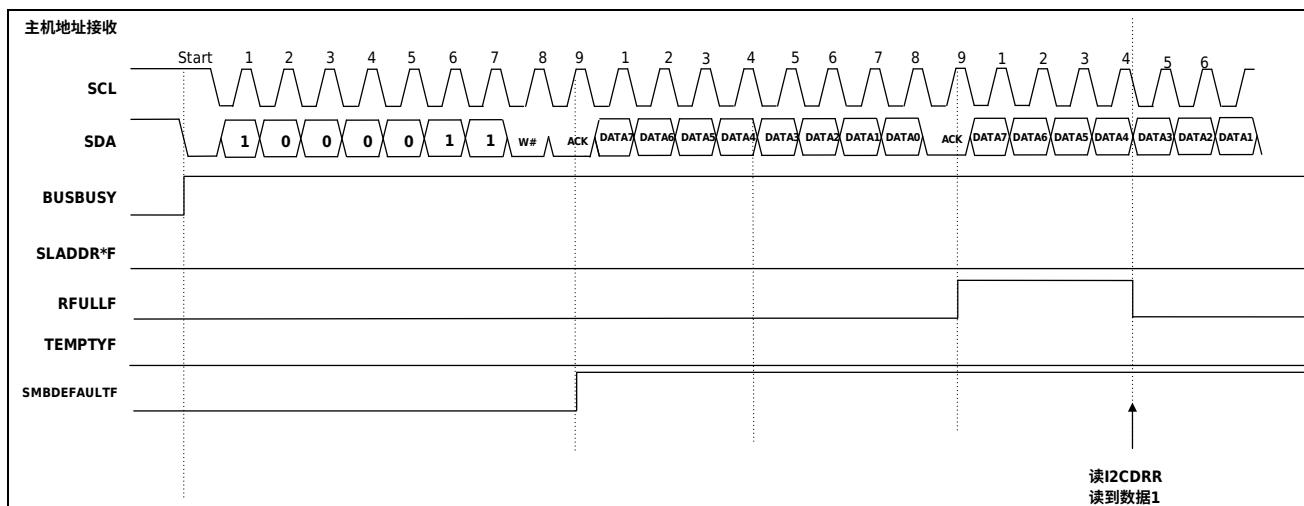


图 29-19 SMBus 默认地址时序图

### 29.3.3 SMBus 动作

此 I2C 接口能进行以 SMBus (Ver.2.0) 为基准的通信。要进行 SMBus 通信时，必须将 I2C\_CR1.SMBUS 位置“1”。必须通过设定 I2C\_CCR 寄存器，将传送速度设定在 SMBus 规格的 10kbps~ 100kbps 范围内。

#### 29.3.3.1 SMBus 超时测量

##### 1) SCL 电平超时测量

在总线忙的状态下，能通过检测到 SCL 线的低电平或者高电平被固定了一定的时间以上，并且检测到总线的异常状态。

超时检测功能监视 SCL 线的状态，通过内部计数器对高电平或者低电平的时间进行计数。如果 SCL 线有变化（上升/ 下降），就对内部计数器进行复位，否则就继续进行计数。如果在 SCL 线没有变化的状态下内部计数器计数到 TOUTHIGH/TOUTLOW 设定值，就能检测到超时并且通知总线的异常状态。

对于内部计数器的计数，能通过设定 HTMOUT、LTMOUT 位选择是在 SCL 线的低电平还是在高电平的状态下进行计数，或者在低电平和高电平的状态下都进行计数。如果将 HTMOUT、LTMOUT 位都置“0”，就不进行内部计数。

## 2) 从机的超时测量

SMBus 通信的从属设备需要测量以下所示的区间（超时间隔：TLOW:SEXT）。

- 开始条件到停止条件的区间

在通过从属设备进行超时测量时，使用开始条件检测中断和停止条件检测中断并且通过芯片定时器，测量从检测到开始条件到检测到停止条件的时间。此超时的测量时间必须在 SMBus 规格的时钟低电平的累积时间[ 从属设备]TLOW:SEXT: 25ms (max) 以内。

如果定时器测量的时间超过 SMBus 规格的时钟 Low 电平检测的超时 TTIMEOUT: 25ms (min)，从机就需要释放总线。

## 3) 主机的超时测量

SMBus 通信的主控设备需要测量以下所示的区间（超时间隔：TLOW:MEXT）。

- 开始条件到应答位的区间
- 应答位到下一个应答位的区间
- 应答位到停止条件的区间

在主机进行超时测量时，使用开始条件检测中断、停止条件检测中断以及发送结束中断或者接收数据满中断，通过芯片定时器测量各区间的时间。此超时的测量时间必须在 SMBus 规格的时钟低电平的累积时间[ 主机]TLOW:MEXT: 10ms (max) 以内，开始条件到停止条件的全部 TLOW:MEXT 的累加结果必须在 TLOW:SEXT: 25ms (max) 以内。

如果定时器测量的时间超过 SMBus 规格的时钟低电平的累积时间[ 主控设备]TLOW:MEXT: 10ms (max)，或者各测量时间的累加结果超过 SMBus 规格的时钟低电平检测的超时 TTIMEOUT: 25ms (min)，主机就需要中止处理。在主机发送时，必须立即中止发送（写 I2C\_DTR 寄存器）。通过发行停止条件中止主机的处理。

### 29.3.3.2 数据包错误码（PEC）

通信中，利用 CPU 运算 CRC，发送 SMBus 的数据包错误码（PEC）或者检查接收数据。

### 29.3.4 复位

具有对通信模块进行复位的功能。有 2 种复位，1 种是对包括 ICCR2.BBSY 标志在内的全部寄存器进行初始化的复位，另一种是在保持各种设定值的状态下解除从属地址匹配状态以及对内部计数器进行初始化等的内部复位。

复位后，必须将 I2C\_CR1.SWRST 位置“0”。

因为无论进行哪种复位都要解除 SCL 引脚/SDA 引脚的输出状态而变为高阻抗状态，所以也能用于解除总线意外停机状态。

从属模式中的复位会引起与主控设备的不同步，因此尽量避免使用。必须注意：在复位（I2C\_CR1.PE 位和 I2C\_CR1.SWRST 位为“01b”）过程中不能监视开始条件等的总线状态。

### 29.3.5 中断和事件信号输出

I2C 具有 4 种中断和用于触发启动其他外围电路的事件输出供用户选择。包括：通信错误的发生（仲裁失败检测、NACK 检测、超时检测、开始条件检测、停止条件检测）、接收结束、发送数据空、发送结束。

表 29-2 中断一览表

名称	中断源	中断标志	中断条件
I2C_EEI	通信错误/通信事件	ARLOF	ARLOF=1&ARLOIE=1
		SLADDR0F	SLADDR0F=1& SLADDR0IE=1
		SLADDR1F	SLADDR1F=1& SLADDR1IE=1
		SMBALRTF	SMBALRTF =1& SMBALRTIE=1
		SMBHOSTF	SMBHOSTF =1& SMBHOSTFIE=1
		SMBDEFAULTF	SMBDEFAULTF =1& SMBDEFAULTIE=1
		GENCALLF	GENCALLF =1& GENCALLIE=1
		NACKF	NACKF=1&NACKIE-1
		TMOUTF	TMOUTF=1&TMOUTIE=1
		STARTF	STARTF=1&STARTIE=1
I2C_RXI	接收数据满	RFULLF	RFULLF=1&RFULLIE=1 & ~FEN
	接收FIFO数据非空	RFREQF	RFREQF= 1 & RFREQIE=1 & FEN
I2C_TXI	发送数据空	TEMPTYF	TEMPTYF=1&TEMPTYIE=1
I2C_TEI	发送结束	TENDF	TENDF=1&TENDIE=1

表 29-3 事件信号输出一览表

名称	事件源	事件条件
I2C_EEI	通信错误/通信时间	ARLOF=1
		SLADDR0F=1
		SLADDR1F=1
		SMBALRTF=1
		SMBHOSTF=1
		SMBDEFAULTF=1
		GENCALLF=1
		NACKF=1
		TMOUTF=1
		STARTF=1
I2C_RXI	接收数据满	RFULLF=1
	接收数据非空	RFREQF=1
I2C_TXI	发送数据空	TEMPTYF=1
I2C_TEI	发送结束	TENDF=1

注：I2C\_RXI、I2C\_TXI、I2C\_TEI、I2C\_EEI 事件作为 AOS 源时可以触发 DMA 模块动作，不能触发其他 AOS 目标模块。

### 29.3.6 可编程数字滤波

SCL 引脚和 SDA 引脚的状态经由模拟滤波器电路和数字滤波器进入内部。数字滤波器电路的框图下图所示。

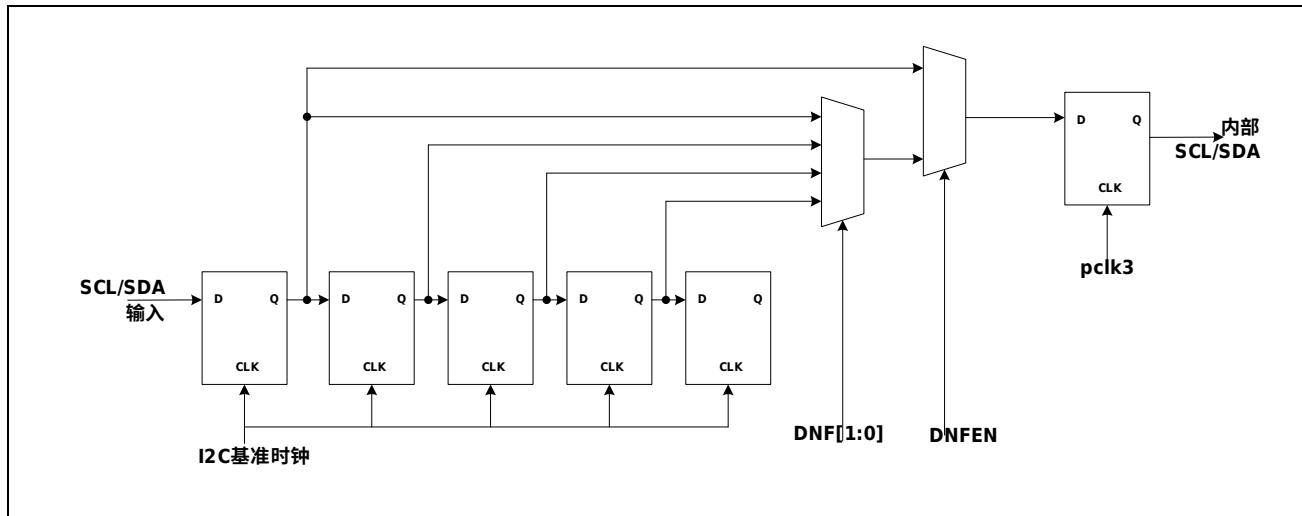


图 29-20 数字滤波电路框图

内部数字滤波器电路由 4 段串联的触发电路和匹配检测电路构成。

通过 I2C\_FLTR.DNF 位选择数字滤波器的有效段数，根据选择的有效段数，噪声消除能力为 1~4 个 I2C 周期。

在 I2C 内部时钟的下降沿对 SCL 引脚的输入信号（或者 SDA 引脚的输入信号）进行采样，如果 I2C\_FLTR.DNF 位设定的有效段数的触发电路输出全部匹配，就将该电平作为内部信号进行传输，否则就保持原来的值。

## 29.4 应用软件设定 I2C 初始化流程

在开始发送或者接收数据时，必须按下述步骤进行初始化。

1. PE 位设定为 0。
2. SWRST 设定为 1，通信复位
3. PE 位设定为 1，内部状态复位
4. 设定从机地址格式和地址
5. 设定波特率
6. 根据需要设定控制寄存器功能及中断
7. SWRST 位设定为 0，解除内部状态复位。
8. 初始化结束。可发送接收数据。

## 29.5 寄存器说明

表 29-4 寄存器一览表

I2C1 基准地址: 0x4003B400

I2C2 基准地址: 0x4003B800

I2C3 基准地址: 0x4003BC00

寄存器名	符号	偏移地址	位宽	复位值
I2C控制寄存器1	I2C_CR1	0x00	32	0x0000 0040
I2C控制寄存器2	I2C_CR2	0x04	32	0x0000 0000
I2C控制寄存器3	I2C_CR3	0x08	32	0x0000 0006
I2C控制寄存器4	I2C_CR4	0x0C	32	0x0030 0307
I2C从机地址寄存器0	I2C_SLR0	0x10	32	0x0000 1000
I2C从机地址寄存器1	I2C_SLR1	0x14	32	0x0000 0000
I2C SCL电平超时控制寄存器	I2C_SLTR	0x18	32	0xFFFF FFFF
I2C状态寄存器	I2C_SR	0x1C	32	0x0000 0000
I2C状态清零寄存器	I2C_CLR	0x20	32	0x0000 0000
I2C数据发送寄存器	I2C_DTR	0x24	8	0xFF
I2C数据接收寄存器	I2C_DRR	0x28	8	0x00
I2C数据移位寄存器	I2C_DSR	-	-	-
I2C时钟控制寄存器	I2C_CCR	0x2C	32	0x0000 1F1F
I2C滤波控制寄存器	I2C_FLTR	0x30	32	0x0000 0010
I2C FIFO控制寄存器	I2C_FSTR	0x34	32	0x0000 0000
I2C从机地址接收寄存器	I2C_SLVADRR	0x38	32	0x0000 0000

## 29.5.1 I2C 控制寄存器 1(I2C\_CR1)

复位值：0x00000040

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SWR ST	-	-	-	-	ACK	STOP	STAR T	REST ART	ENG C	-	SMBH OSTE N	SMBD EFAUL TEN	SMBA LRTEN	SMB US	PE

位	标记	位名	功能	读写									
b31-16	Reserved	-	读出时为“0”,写入时写“0”	R/W									
b15	SWRST	软件复位	0: 解除复位 1: 软件复位 本位与PE位组合, 选择内部状态复位或者通信复位 <table border="1" style="margin-left: 20px;"> <tr> <td>SWRST</td><td>PE</td><td>复位内容</td></tr> <tr> <td>1</td><td>0</td><td>通信复位: I2C内部的全部寄存器及内部状态复位。</td></tr> <tr> <td>1</td><td>1</td><td>内部状态复位: I2C_SR, I2C_DSR 寄存器和内部状态机进行复位</td></tr> </table>	SWRST	PE	复位内容	1	0	通信复位: I2C内部的全部寄存器及内部状态复位。	1	1	内部状态复位: I2C_SR, I2C_DSR 寄存器和内部状态机进行复位	R/W
SWRST	PE	复位内容											
1	0	通信复位: I2C内部的全部寄存器及内部状态复位。											
1	1	内部状态复位: I2C_SR, I2C_DSR 寄存器和内部状态机进行复位											
b14-11	Reserved	-	读出时为“0”,写入时写“0”	R/W									
b10	ACK	发送应答	0: 应答位发送“0” (发送ACK) 1: 应答位发送“1” (发送NACK)	R/W									
b9	STOP	停止条件生成位	0: 不生成停止条件 1: 生成停止条件 此位可软件置1和清0。 硬件清0条件: 检测到停止条件 仲裁失败 检测到开始条件 通信复位	R/W									
b8	START	起始条件生成位	0: 不生成起始条件 1: 生成起始条件 此位可软件置1和清0。 硬件清0条件: 检测到开始条件 仲裁失败时 通信复位	R/W									
b7	RESTART	重复起始条件生成位	0: 不生成重复起始条件 1: 生成重复起始条件 此位可软件置1和清0。 硬件清0条件: 1) 检测到开始条件 2) 仲裁失败时 3) 通信复位	R/W									
b6	ENGC	广播呼叫时能	0: 广播地址检测无效 1: 广播地址检测有效	R/W									

b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4	SMBHOSTEN	允许匹配SMBUS主机地址位	0: 禁止匹配SMBUS主机地址 1: 允许匹配SMBUS主机地址	R/W
b3	SMBDEFAULTEN	允许匹配SMBUS默认地址位	0: 禁止匹配SMBUS默认地址 1: 允许匹配SMBUS默认地址	R/W
b2	SMBALRTEN	允许匹配SMBUS报警响应地址位	0: 禁止SMBUS报警响应地址 1: 允许SMBUS报警响应地址	R/W
b1	SMBUS	SMBUS/I2C总线模式选择位	0: I2C总线模式 1: SMBUS总线模式	R/W
b0	PE	I2C功能使能	0: I2C功能禁止 1: I2C功能允许	R/W
本位与SWRST位组合, 选择内部状态复位或者通信复位				

## 29.5.2 I2C 控制寄存器 2(I2C\_CR2)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	SMBALRTIE	SMBHOSTIE	SMBDEFALTIIE	GENCALLIE	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	TMOUTIE	-	NACKIE	RFREQIE	-	ARLOIE	-	TEMPTYIE	RFULLIE	-	STOPIE	TENDIE	SLADDR1IE	SLADDR0IE	STARIE

位	标记	位名	功能	读写
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	SMBALRTIE	SMBUS报警响应地址匹配一致中断允许	0: SMBUS报警响应地址匹配一致中断禁止 1: SMBUS报警响应地址匹配一致中断允许	R/W
b22	SMBHOSTIE	SMBUS主机地址匹配一致中断允许	0: SMBUS主机地址匹配一致中断禁止 1: SMBUS主机地址匹配一致中断允许	R/W
b21	SMBDEFAULTIE	SMBUS默认地址匹配一致中断允许	0: SMBUS默认地址匹配一致中断禁止 1: SMBUS默认地址匹配一致中断允许	R/W
b20	GENCALLIE	广播呼叫地址匹配一致中断允许	0: 广播呼叫地址匹配一致中断禁止 1: 广播呼叫地址匹配一致中断允许	R/W
b19~b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14	TMOUTIE	超时中断允许	0: 超时中断禁止 1: 超时中断允许	R/W
b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	NACKIE	NACK中断允许	0: 接收到NACK中断禁止 1: 接收到NACK中断允许	R/W
b11	RFREQIE	FIFO接收请求中断使能	0: 接收FIFO中断请求禁止 1: 接收FIFO中断请求使能	R/W
b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9	ARLOIE	仲裁失败中断允许	0: 仲裁失败中断禁止 1: 仲裁失败中断允许	R/W
b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	TEMPTYIE	发送数据空中断允许位	0: 发送数据空中断禁止 1: 发送数据空中断允许	R/W
b6	RFULLIE	接收数据满中断允许位	0: 接收数据满中断禁止 1: 接收数据满中断允许	R/W
b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4	STOPIE	停止条件中断允许	0: 总线检测到停止条件中断禁止 1: 总线检测到停止条件中断允许	R/W
b3	TENDIE	发送一帧数据结束中断允许位	0: 发送一帧数据结束中断禁止 1: 发送一帧数据结束中断允许	R/W
b2	SLADDR1IE	从机地址1匹配一致中断允许	0: 从机地址1匹配一致中断禁止 1: 从机地址1匹配一致中断允许	R/W
b1	SLADDR0IE	从机地址0匹配一致中断允许	0: 从机地址0匹配一致中断禁止 1: 从机地址0匹配一致中断允许	R/W

---

b0	STARTIE	开始条件/重新开始条件中断允许	0: 总线检测到开始条件中断禁止 1: 总线检测到开始条件中断允许	R/W
----	---------	-----------------	--------------------------------------	-----

---

### 29.5.3 I2C 控制寄存器 3(I2C\_CR3)

复位值：0x00000006

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	FACK EN	-	-	-	-	HTM OUT	LTM OUT	TMOU TEN

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	FACKEN	RFULLF标志位置位时间点选择	0: 在SCL时钟的第9个时钟上升时, RFULLF标志位为“1”。(在第8个时钟的下降沿, SCL线不保持低电平) 1: 在SCL时钟的第8个时钟上升时, RFULLF标志位为“1”。(在第8个时钟的下降沿, SCL线保持低电平) 通过写ACK位来解除保持的低电平。	R/W
b6~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	HTMOUT	高电平超时检测允许	0: 在SCL线为高电平时, 禁止超时检测。 1: 在SCL线为高电平时, 允许超时检测。	R/W
b1	LTMOUT	低电平超时检测允许	0: 在SCL线为低电平时, 禁止超时检测。 1: 在SCL线为低电平时, 允许超时检测。	R/W
b0	TMOUTEN	超时功能允许位	0: 检测SCL电平超时功能禁止 1: 检测SCL电平超时功能允许	R/W

## 29.5.4 I2C 控制寄存器 4(I2C\_CR4)

复位值：0x00301b07

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	BUSFREE_E CLREN	-	BUSWAIT AIT	-	-	-	-	-	-	-	-	-	-

位	标记	位名	功能	读写
b31~b22	Reserved	-	读出时为“0”,写入时写“0”	R/W
b21~b20	Reserved	-	读出时为“1”,写入时写“1”	R/W
b19~b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	BUSFREE_CLREN	总线结束状态清除位	0: 禁止I2C总线STOP标志对I2C_SLVADDR从机接收地址寄存器 以及I2C_SR状态寄存器中的SMBALRTF, SMBHOSTF, SMBDEFAULTF, GENCALLF, MSL, SLADDR1F, SLADDR0F, STARTF位进行清除 1: 使能I2C总线STOP标志对I2C_SLVADDR从机接收地址寄存器 以及I2C_SR状态寄存器中的SMBALRTF, SMBHOSTF, SMBDEFAULTF, GENCALLF, MSL, SLADDR1F, SLADDR0F, STARTF位进行清除	R/W
b11	Reserved	-	读出时为“1”,写入时写“1”	R/W
b10	BUSWAIT	总线等待位	0: I2C_DRR接收满、I2C_DSR为空时, 在第9个时钟和下次传送的第1个时钟之间不保持低电平, 继续接收下一个数据。 1: I2C_DRR接收满、I2C_DSR为空时, 在第9个时钟和下次传送的第1个时钟之间保持低电平, 通过读I2C_DRR寄存器来解除保持的低电平。	R/W
b9~b8	Reserved	-	读出时为“1”,写入时写“1”	R/W
b7~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~b0	Reserved	-	读出时为“1”,写入时写“1”	R/W

## 29.5.5 I2C 从机地址寄存器 0(I2C\_SLR0)

复位值：0x00001000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16				
-	-	-	-	-	MASK0 EN	MSLADDR0[9:0]													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0				
ADD RMO D0	-	-	SLAD DROE N	-	-	SLADDR0[9:0]													

位	标记	位名	功能	读写
b31~b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26	MASK0EN	从机地址0掩码有效位	0: 从机地址寄存器0掩码设定值无效 1: 从机地址寄存器0掩码设定值有效	
b25~b24	MSLADDR0[9:8]	10位从机地址的高位 掩码地址	设定从机地址0的高位掩码地址。 当MASK0EN位为“0”时，MSLADDR0[9:8]设定的掩码无效 当MASK0EN位为“1”时，MSLADDR0[9:8]设定的掩码有效 其有效范围与ADDRMOD0所设值成正相关即： 当ADDRMOD0位为“0”时，MSLADDR0[9:8]位无效。 当ADDRMOD0位为“1”时，MSLADDR0[9:8]为10位从机地址的高 两位掩码地址	
b23~b16	MSLADDR0[7:0]	7位地址/10位地址的 低位掩码地址	设定从机地址0的低位掩码地址。 当MASK0EN位为“0”时，MSLADDR0[9:0]设定的掩码值无效 当MASK0EN位为“1”时，MSLADDR0[9:0] 设定的掩码值有效，其 有效范围与ADDRMOD0所设值成正相关即： 当ADDRMOD0位为“0”时，MSLADDR0[7:1]为7位从机地址掩码， MSLADDR0[0]位无效。 当ADDRMOD0位为“1”时，MSLADDR0[7:0]为10位从机地址的低8 位地址掩码。	
b15	ADDRMOD0	7位/10位地址格式选 择位	0: 选择7位地址格式 1: 选择10位地址格式	R/W
b14~b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	SLADDR0EN	从机地址0有效位	0: 从机地址寄存器0设定值无效 1: 从机地址寄存器0设定值有效	R/W
b11~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9~b8	SLADDR0[9:8]	10位从机地址的高位	设定从机地址。 当ADDRMOD0位为“0”时，此位设定无效。 当ADDRMOD0位为“1”时，此位作为10位从机地址的高两位。	R/W
b7~b0	SLADDR0[7:0]	7位地址/10位地址的 低位	设定从机地址。 当ADDRMOD0位为“0”时，SLADDR0[7:1]为7位从机地址。 SLADDR0[0]位无效。 当ADDRMOD0位为“1”时，SLADDR0[7:0]为10位从机地址的低8 位地址。	R/W

## 29.5.6 I2C 从机地址寄存器 1(I2C\_SLR1)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16				
-	-	-	-	-	MASK1EN	MLADDR1[9:0]													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0				
ADD RMO D1	-	-	SLAD DR1E N	-	-	SLADDR1[9:0]													

位	标记	位名	功能	读写
b31~b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26	MASK1EN	掩码使能位	0: 禁止I2C从机地址1的掩码功能 1: 使能I2C从机地址1的掩码功能	R/W
b25~b24	MSLADDR1[9:8]	10位从机地址掩码的高位	设定从机地址1的高位掩码地址。 当MASK1EN位为“0”时，MSLADDR1[9:8]设定的掩码无效 当MASK1EN位为“1”时，MSLADDR1[9:8]设定的掩码有效 其有效范围与ADDRMOD1所设值成正相关即： 当ADDRMOD1位为“0”时，MSLADDR1[9:8]位无效。 当ADDRMOD1位为“1”时，MSLADDR1[9:8]为10位从机地址的高两位掩码地址。	
b23~b16	MSLADDR1[7:0]	7位地址/10位地址掩码的低位	设定从机地址1的低位掩码地址。 当MASK1EN位为“0”时，MSLADDR1[9:0]设定的掩码值无效 当MASK1EN位为“1”时，MSLADDR1[9:0] 设定的掩码值有效，其有效范围与ADDRMOD1所设值成正相关即： 当ADDRMOD1位为“0”时，MSLADDR1[7:1]为7位从机地址掩码，MSLADDR1[0]位无效。 当ADDRMOD1位为“1”时，MSLADDR1[7:0]为10位从机地址的低8位地址掩码。	
b15	ADDRMOD1	7位/10位地址格式选择位	0: 选择7位地址格式 1: 选择10位地址格式	R/W
b14~b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	SLADDR1EN	从机地址1有效位	0: 从机地址寄存器1设定值无效 1: 从机地址寄存器1设定值有效	R/W
b11~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9~b8	SLADDR1[9:8]	10位从机地址的高位	设定从机地址。 当ADDRMOD1位为“0”时，此位设定无效。 当ADDRMOD1位为“1”时，此位作为10位从机地址的高两位。	R/W
b7~b0	SLADDR1[7:0]	7位地址/10位地址的低位	设定从机地址。 当ADDRMOD1位为“0”时，SLADDR1[7:1]为7位从机地址。 SLADDR1[0]位无效。 当ADDRMOD1位为“1”时，SLADDR1[7:0]为10位从机地址的低8位地址。	R/W

### 29.5.7 I2C SCL 电平超时控制寄存器(I2C\_SLTR)

复位值：0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TOUTHIGH[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TOUTLOW[15:0]															

位	标记	位名	功能	读写
b31~b16	TOUTHIGH	SCL高电平超时周期	TOUTHIGH设定SCL高电平超时周期。 SCL高电平超时时间=TOUTHIGH*I2C基准时钟周期	R/W
b15~b0	TOUTLOW	SCL低电平超时周期	TOUTLOW设定SCL低电平超时周期。 SCL低电平超时时间=TOUTLOW*I2C基准时钟周期	R/W

## 29.5.8 I2C 状态寄存器(I2C\_SR)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RF REQ	-	TFST[1:0]	RF FULL	TF EMP TY	TF FULL	TF EMP TY	SMB ALRT F	SMB HOST F	SMB DEFA ULTF	GEN CALL F	-	TRA	BUS Y	MSL	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	TMOU TF	-	NAC KF	-	ACK RF	ARL OF	-	TEM PTYF	RFUL LF	-	STOP F	TEN DF	SLAD DR1F	SLAD DROF	STAR TF

位	标记	位名	功能	读写
b31	RFREQ	接收FIFO请求标志位	0: 当前接收FIFO无有效数据 1: 当前接收FIFO有有效数据	R
b30	Reserved	-	读出时为“0”,写入时写“0”	R
b29~b28	TFST[1:0]	发送FIFO状态标志位	00: 当前发送FIFO已空 01: 当前发送FIFO有1 byte有效数据 10: 当前发送FIFO有2 byte有效数据 11: 保留位	R
b27	RFFULL	接收FIFO满标志位	0: 当前接收FIFO未满 1: 当前接收FIFO已满	R
b26	RFEMPTY	接收FIFO空标志位	0: 当前接收FIFO未空 1: 当前接收FIFO已空	R
b25	TFFULL	发送FIFO满标志位	0: 当前发送FIFO未满 1: 当前发送FIFO已满	R
b24	TFEMPTY	发送FIFO空标志位	0: 当前发送FIFO未空 1: 当前发送FIFO已空	R
b23	SMBALRTF	SMBUS报警响应地址匹 配一致标志位	0: 未匹配到SMBUS报警响应地址 1: 检测到主机地址 置“1”条件: 接收到的地址和0001 100b匹配一致 清“0”条件: SMBALRTFCLR写“1” 检测到停止条件 通信复位	R
b22	SMBHOSTF	SMBUS主机地址匹 配一致标志位	0: 未匹配到SMBUS主机地址 1: 匹配到SMBUS主机地址 地址匹配一致条件如下: 置“1”条件: 接收到的地址和0001 000b匹配一致 清“1”条件: SMBHOSTFCLR写“1” 检测到停止条件 通信复位以及内部复位	R
b21	SMBDEFAULTF	SMBUS默认地址匹 配一致标志位	0: 未匹配到SMBUS默认地址 1: 匹配到SMBUS默认地址 置“1”条件: 接收到的地址和1100 001b匹配一致	R

			清“0”条件： SMBDEFAULTCLR写“1” 检测到停止条件 通信复位																
b20	GENCALLF	广播呼叫地址匹配一致 标志	0：未匹配到广播呼叫地址 1：匹配到广播呼叫地址 置“1”条件： 当接收的从机地址与广播呼叫地址（All“0”）匹配一致时 清“0”条件： GENCALLFCLR写“1” 检测到停止条件 通信复位	R															
b19	Reserved	-	读出时为“0”，写入时写“0”	R															
b18	TRA	发送接收选择位	此位表示选择发送数据还是接收数据。 0：接收数据 1：发送数据 此位可软件置1和清0。 硬件置“1”条件 检测到开始条件 主机模式下，发送的R/W位为0 从机模式下，地址匹配且接收的R/W位为1 硬件清“0”条件 检测到停止条件 主机模式下，发送的R/W位为1 从机模式下，地址匹配且接收的R/W位为0 通信复位	R/W															
b17	BUSY	总线忙标志位	0：空闲状态，总线上无通信 1：占有状态，总线正在通信 置“1”条件： 检测到总线上开始条件 清“0”条件： 检测到总线停止条件 通信复位	R															
b16	MSL	主从机选择位	此位表示主机还是从机。 0：从机模式 1：主机模式 通过和TRA位的组合，表示I2C的运行模式。 <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <th>MSL</th> <th>TRA</th> <th>I2C运行模式</th> </tr> <tr> <td>0</td> <td>0</td> <td>从机接收模式</td> </tr> <tr> <td>0</td> <td>1</td> <td>从机发送模式</td> </tr> <tr> <td>1</td> <td>0</td> <td>主机接收模式</td> </tr> <tr> <td>1</td> <td>1</td> <td>主机发送模式</td> </tr> </table> 此位可软件置1和清0。 硬件置“1”条件 START位为1的状态下，检测到开始条件 硬件清“0”条件 1) 检测到停止条件 2) 仲裁失败	MSL	TRA	I2C运行模式	0	0	从机接收模式	0	1	从机发送模式	1	0	主机接收模式	1	1	主机发送模式	R/W
MSL	TRA	I2C运行模式																	
0	0	从机接收模式																	
0	1	从机发送模式																	
1	0	主机接收模式																	
1	1	主机发送模式																	

## 3) 通信复位

b15	Reserved	-	读出时为“0”,写入时写“0”  0: 未检测到SCL电平超时 1: SCL电平超时 置“1”条件: I2C_SLTR设定的周期内, SCL未翻转 清“1”条件: TMOUTFCLR写“1” 通信复位	R
b13	Reserved	-	读出时为“0”,写入时写“0”  0: 未接收到NACK 1: 接收到NACK 置“1”条件: 发送模式下, 接收到NACK 清“0”条件: NACKFCLR写“1” 通信复位	R
b11	Reserved	-	读出时为“0”,写入时写“0”  0: 接收到应答位为“0” (接收ACK) 1: 接收到应答位为“1” (接收NACK) 置“1”条件: 发送模式下, 接收到NACK 清“0”条件: 发送模式下, 接收到ACK 通信复位	R
b10	ACKRF	接收应答位	0: 未发生仲裁失败 1: 仲裁失败 置“1”条件: 仲裁失败 清“0”条件: ARLOFCLR写“1” 通信复位	R
b9	ARLOF	仲裁失败标志位	仲裁失败 清“0”条件: ARLOFCLR写“1” 通信复位	R
b8	Reserved	-	读出时为“0”,写入时写“0”  0: I2C_DTR寄存器满 1: I2C_DTR寄存器空 置“1”条件: I2C_DTR数据传送到I2C_DSR TRA位置1 清“0”条件: 写I2C_DTR TRA位清0 通信复位	R
b7	TEMPTYF	发送数据空标志位	0: I2C_DRR寄存器空 1: I2C_DRR寄存器满 置“1”条件: TRA位置1 清“0”条件: 写I2C_DTR TRA位清0 通信复位	R
b6	RFULLF	接收数据满标志位	0: I2C_DRR寄存器空 1: I2C_DRR寄存器满 置“1”条件: 接收的数据从I2C_DSR传送到I2C_DRR 清“0”条件:	R

			读I2C_DRR RFULLFCLR写“1” 通信复位	
b5	Reserved	-	读出时为“0”,写入时写“0”  0: 总线未检测到停止条件 1: 总线检测到停止条件 置“1”条件: 1) 检测到停止条件 清“0”条件: STOPFCLR写“1” 通信复位	R
b4	STOPF	停止条件标志位	0: I2C_DSR寄存器发送中 1: I2C_DSR寄存器发送结束 置“1”条件: TEMPTYF=1的条件下, SCL的第9个上升沿此位置“1”	
b3	TENDF	发送数据结束标志位	清“0”条件: 检测到停止条件 写I2C_DTR TENDFCLR写“1” 通信复位	R
b2	SLADDR1F	从机地址寄存器1匹配一致标志	0: 未检测到从机地址寄存器1一致地址 1: 检测到从机地址寄存器1一致地址 置“1”条件: 当I2C_SLR1.ADDRMOD1位为“0”时, 接收到的从机地址和I2C_SLR1.SLADDR1[7:1]匹配时。 当I2C_SLR1.ADDRMOD1位为“1”时, 接收10位从机地址的第一个字节地址与11110b+I2C_SLR1.SLADDR1[9:8]匹配一致并且第二个字节地址与I2C_SLR1.SLADDR1[7:0]匹配一致。 清“0”条件: 检测到停止条件 SLADDR1FCLR写“1” 通信复位	R
b1	SLADDR0F	从机地址寄存器0匹配一致标志	0: 未检测到从机地址寄存器0一致地址 1: 检测到从机地址寄存器0一致地址 置“1”条件: 当I2C_SLR0.ADDRMOD0位为“0”时, 接收到的从机地址和I2C_SLR0.SLADDR0[7:1]匹配时。 当I2C_SLR0.ADDRMOD0位为“1”时, 接收10位从机地址的第一个字节地址与11110b+I2C_SLR0.SLADDR0[9:8]匹配一致并且第二个字节地址与I2C_SLR0.SLADDR0[7:0]匹配一致。 清“0”条件: 检测到停止条件 SLADDR0FCLR写“1” 通信复位	R
b0	STARTF	开始条件/重新开始条件标志位	0: 总线未检测到开始条件 1: 总线检测到开始条件 置“1”条件 1) 检测到开始条件	R

---

清“0”条件

- 1) 检测到停止条件
  - 2) STARTFCLR写“1”
  - 3) 通信复位
-

### 29.5.9 I2C 状态清零寄存器(I2C\_CLR)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	SMB ALRT FCLR	SMB HOST FCLR	SMB DEFA ULTF CLR	GEN CALL FCLR	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	TMO UTFC LR	-	NAC KFCLR	-	RF REQ CLR	ARL OFCLR	-	-	RFUL LFCLR	-	STOP FCLR	TEN DFCLR	SLA DDR1 FCLR	SLA DDR0 FCLR	STAR TFCLR

位	标记	位名	功能	读写
b31~b24	Reserved	-	写入时写“0”	W
b23	SMBALRTFCLR	SMBUS报警响应地址 匹配一致标志清零位	写“1”清除SMBALRTF标志位	W
b22	SMBHOSTFCLR	SMBUS主机地址匹配 一致标志清零位	写“1”清除SMBHOSTF标志位	W
b21	SMBDEFAULTFCLR	SMBUS默认地址匹配 一致标志清零位	写“1”清除SMBDEFAULTF标志位	W
b20	GENCALLFCLR	广播呼叫地址匹配一 致标志	写“1”清除GENCALLF标志位	W
b19~b15	Reserved	-	写入时写“0”	W
b14	TMOUTFCLR	超时标志位	写“1”清除TMOUTF标志位	W
b13	Reserved	-	写入时写“0”	W
b12	NACKFCLR	NACK标志位	写“1”清除NACKF标志位	W
b11	Reserved	-	写入时写“0”	W
b10	RFREQCLR	接收FIFO数据请求标 志位	写“1”清除RFREQF	W
b9	ARLOFCLR	仲裁失败标志位	写“1”清除ARLOF标志位	W
b8~b7	Reserved	-	写入时写“0”	W
b6	RFULLFCLR	接收数据满标志位	写“1”清除RFULLF标志位	W
b5	Reserved	-	写入时写“0”	W
b4	STOPFCLR	停止条件标志位	写“1”清除STOPF标志位	W
b3	TENDFCLR	发送数据结束标志位	写“1”清除TENDF标志位	W
b2	SLADDR1FCLR	从机地址寄存器1匹 配一致标志清零位	写“1”清除SLADDR1F标志位	W
b1	SLADDR0FCLR	从机地址寄存器1匹 配一致标志清零位	写“1”清除SLADDR0F标志位	W
b0	STARTFCLR	开始条件/重新开始条 件标志清零位	写“1”清除STARTF标志位	W

## 29.5.10 I2C 数据发送寄存器(I2C\_DTR)

复位值：0xFF

b7	b6	b5	b4	b3	b2	b1	b0
DT[7:0]							

如果 I2C\_DSR 寄存器为空，就将写在 I2C\_DTR 寄存器的发送数据传送到 I2C\_DSR 寄存器，发送模式时开始发送数据到 SDA 上。

I2C\_DSR 寄存器和 I2C\_DTR 寄存器是双缓冲结构，在 I2C\_DSR 寄存器数据发送过程中，如果预先写 I2C\_DTR 寄存器的数据，就能进行连续发送数据。

I2C\_DTR 寄存器可读可写。请在发送数据空中断要求发生时，仅写一次 I2C\_DTR 寄存器。

如果当前 I2C 工作在 FIFO 模式时，访问 I2C\_DTR 寄存器地址可以直接访问发送 FIFO，如果当前 TEMPTYF 有效，针对 I2C\_DTR 寄存器地址的写操作可以将当前的总线数据写入发送 FIFO。当发送 FIFO 有效并且当前 I2C 允许发送则可以将发送 FIFO 数据传送到 I2C\_DSR 寄存器并开始发送数据到 SDA 上。

发送数据源的选择通过 I2C 中 FIFO 状态寄存器 (I2C\_FSTR) 里面的 FEN 位来选择。

当 FEN 为'1'时选择外部数据总线到 FIFO 数据通路以及 FIFO 到 I2C\_DSR 的数据通路。

当 FEN 为'0'时选择外部数据总线到 I2C\_DTR 寄存器通路以及 I2C\_DTR 寄存器到 I2C\_DSR 的数据通路。

默认选择非 FIFO 模式。

### 29.5.11 I2C 数据接收寄存器(I2C\_DRR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
DR[7:0]							

如果接收到 1 帧数据，就能将接收数据从移位寄存器 (I2C\_DSR) 转存到 I2C\_DRR 寄存器，进而可以进入到下一个数据接收状态。

I2C\_DSR 寄存器和 I2C\_DRR 寄存器是双缓冲结构，在 I2C\_DSR 寄存器数据接收过程中，如果读取了 I2C\_DRR 寄存器的数据，就能进行连续接收数据。

禁止对 I2C\_DRR 寄存器写。请在接收数据满中断要求发生时，仅读一次 I2C\_DRR 寄存器。

在 I2C\_SR.RFULLF 标志位为“1”的状态下，如果不读取 I2C\_DRR 寄存器的数据，而立即接收下一个数据，SCL 时钟将在该数据接收的第 7 个或第 8 个下降沿后（参考图 29-12 和图 29-13）自动保持为低电平，直至 I2C\_DRR 寄存器的数据被读取。

I2C 工作在 FIFO 模式时，访问 I2C\_DRR 寄存器地址可以直接访问接收 FIFO。

当 I2C 工作在 FIFO 模式时，如果接收到 1 帧数据，就能将接收数据从移位寄存器 (I2C\_DSR) 转存到接收 FIFO。当接收 FIFO 收到有效数据之后将输出 RFREQ 请求状态信号，此时通过针对 I2C\_DRR 寄存器地址的读操作就可以直接访问接收 FIFO 中的数据。

I2C 工作在 FIFO 模式时，在 I2C\_SR.RFULLF 标志位为“1”的状态下（即接收 FIFO 已经存储了 2byte 数据，此时 FIFO 已满），如果不读取接收 FIFO 中的数据而立即接收下一个数据，SCL 时钟将在该数据接收的第 7 个或第 8 个下降沿后自动保持为低电平，直至接收 FIFO 的数据被读取。

I2C 接收通路的选择通过 I2C 中 FIFO 状态寄存器里面的 FEN 位来选择。

当 FEN 为‘1’时选择 I2C\_DSR 到接收 FIFO 数据通路以及接收 FIFO 到数据总线的通路。

当 FEN 为‘0’时选择 I2C\_DSR 到 I2C\_DRR 的数据通路以及 I2C\_DRR 到数据总线的通路。

默认选择非 FIFO 功能。

### 29.5.12 I2C 数据移位寄存器(I2C\_DSR)

b7	b6	b5	b4	b3	b2	b1	b0
DSR							

I2C\_DSR 寄存器用于发送和接收数据的移位寄存器。I2C\_DSR 寄存器不可读也不可写。

在数据发送时，将发送数据从 I2C\_DTR 寄存器传送到 I2C\_DSR 寄存器，从 SDA 引脚发送数据。在数据接收时，一旦 1 帧数据接收结束，就将数据从 I2C\_DSR 寄存器传送到 I2C\_DRR 寄存器。

## 29.5.13 I2C 时钟控制寄存器(I2C\_CCR)

复位值：0x00001F1F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	FMPL USEN	-	-	-	-	-	-	FREQ[2:0]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SHIGHW[7:0]										SLOWW[7:0]					
<hr/>															
位	标记	位名	功能	读写											
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b23	FMPLUSEN	Fast plus模式使能位	0: Fast plus模式禁止 1: Fast plus模式使能	R/W											
b22~b19	Reserved	-	读出时为“1”,写入时写“1”	R/W											
b18-b16	FREQ[2:0]	I2C基准时钟频率设定位	0 0 0: I2C基准时钟频率=PCLK0/1	R/W											
			0 0 1: I2C基准时钟频率=PCLK0/2												
			0 1 0: I2C基准时钟频率=PCLK0/4												
			0 1 1: I2C基准时钟频率=PCLK0/8												
			1 0 0: I2C基准时钟频率=PCLK0/16												
			1 0 1: I2C基准时钟频率=PCLK0/32												
			1 1 0: I2C基准时钟频率=PCLK0/64												
			1 1 1: I2C基准时钟频率=PCLK0/128												
b15~b8	SHIGHW[7:0]	设定SCL高电平宽度位	设定SCL时钟的高电平宽度	R/W											
b7~b0	SLOWW[7:0]	设定SCL低电平宽度位	设定SCL时钟的低电平宽度	R/W											

### SHIGHW 位 (设定 SCL 高电平宽度位)

在主机模式下，SHIGHW 是用于设定 SCL 时钟的高电平宽度。在从机模式下，设定无效。

### SLOWW 位 (设定 SCL 低电平宽度位)

SLOWW 是用于设定 SCL 时钟的低电平宽度。在从机模式下，设定值要大于数据准备时间。数据准备时间 (tSU;DAT) 250ns (~100kbps: 标准模式)、100ns (~400kbps: 快速模式)、50ns (~1Mbps: FM+模式)

波特率：

DNFE=0,FREQ=000

波特率=1/{[(SHIGHW+3)+(SLOWW+3)]/ΦI2C+SCL 上升时间+SCL 下降时间}

DNFE=1,FREQ=000

波特率=1/{[(SHIGHW+3+滤波能力)+(SLOWW+3+滤波能力)]/ΦI2C+SCL 上升时间+SCL 下降时间}

DNFE=0,FREQ!=000

波特率=1/{[(SHIGHW+2)+(SLOWW+2)]/ΦI2C+SCL 上升时间+SCL 下降时间}

DNFE=1,FREQ!=000

波特率=1/{[(SHIGHW+2+滤波能力)+(SLOWW+2+滤波能力)]/ $\Phi$ I2C+SCL 上升时间+SCL 下降时间}

**注意：**

- SCL 线的上升时间[tr] 和下降时间[tf] 取决于总线的总电容量[C<sub>b</sub>] 和上拉电阻[R<sub>p</sub>]，详细内容请参照 I2C 总线协议。
- 上述公式中的  $\Phi$ I2C 指 I2C\_CCR.FREQ 设定的 I2C 基准时钟周期。

## 29.5.14 I2C 滤波控制寄存器(I2C\_FLTR)

复位值：0x00000010

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	ANFEN	DNFEN	-	-	DNF[1:0]	

位	标记	位名	功能	读写
b31~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5	ANFEN	模拟滤波功能允许位	0: 模拟滤波功能禁止 1: 模拟滤波功能允许	R/W
b4	DNFEN	数字滤波功能允许位	0: 数字滤波功能禁止 1: 数字滤波功能允许	R/W
b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1-b0	DNF[1:0]	数字滤波器滤波能力选择	00: 滤波能力1个I2C基准时钟周期 01: 滤波能力2个I2C基准时钟周期 10: 滤波能力3个I2C基准时钟周期 11: 滤波能力4个I2C基准时钟周期	R/W

### 29.5.15 I2C FIFO 控制寄存器 (I2C\_FSTR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	RFST[1:0]	TFST[1:0]	NACK_TFFLUSH	RFFLUSH	TFFLUSH	FEN		

位	标记	位名	功能	读写
b31~b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7~b6	RFST[1:0]	接收FIFO状态位	00: FIFO空 01: FIFO半满 10: FIFO满 11: 保留	R
b5~b4	TFST[1:0]	发送FIFO状态位	00: FIFO空 01: FIFO半空 10: FIFO满 11: 保留位	R
b3	NACK_TFFLUSH	NACK复位发送FIFO	0: 从模式下接收到NACK后不执行TFIFO FLUSH 1: 从模式下接收到NACK后执行TFIFO FLUSH	R/W
b2	RFFLUSH	RX FIFO指针复位	0: 写入无效 1: 复位RX FIFO指针	R/W
b1	TFFLUSH	TX FIFO指针复位	0: 写入无效 1: 复位TX FIFO指针	R/W
b0	FEN	FIFO功能允许位	0: FIFO功能禁止 1: FIFO功能允许	R/W

### 29.5.16 I2C 从机地址接收寄存器 (I2C\_SLVADRR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-							SLVADR			

位	标记	位名	功能	读写
b31~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9~b0	SLVADR	从机地址接收寄存器	6:0: 7位从机接收地址 9:0: 10位从机接收地址	R

## 30 串行外设接口（SPI）

### 30.1 简介

本产品搭载 4 个通道的串行外设接口 SPI，支持高速全双工串行同步传输，方便地与外围设备进行数据交换。用户可根据需要进行三线/四线，主机/从机及波特率范围的设置。

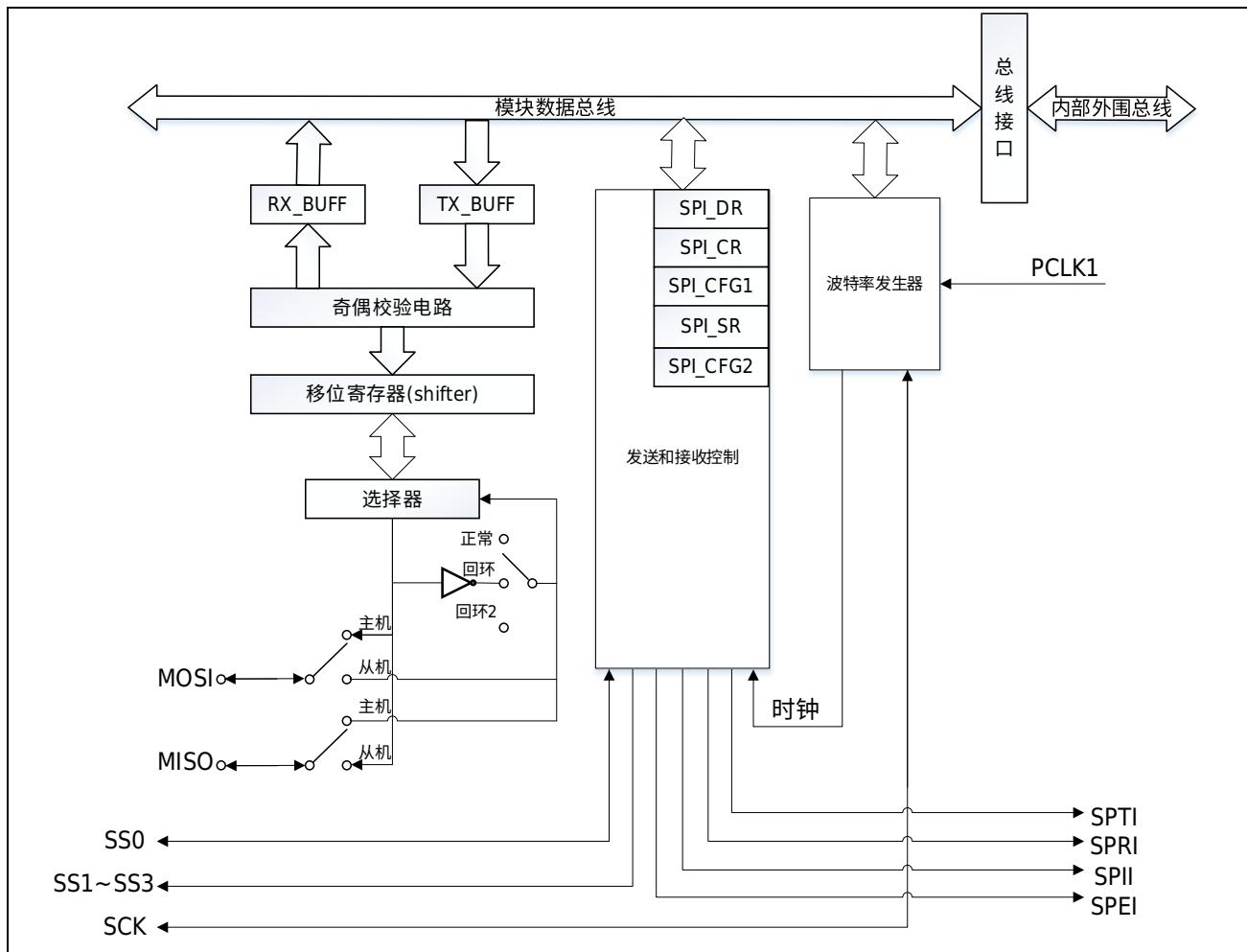
SPI 主要特性：

表 30-1 SPI 特性要点

要点	描述
通道数	1通道
串行通信功能	<ul style="list-style-type: none"><li>支持4线式SPI模式和3线式时钟同步运行模式</li><li>支持全双工和只发送两种通信方式</li><li>可调整通信时钟SCK的极性和相位</li></ul>
数据格式	<ul style="list-style-type: none"><li>可选择数据移位顺序:MSB开始/LSB开始</li><li>可选择数据宽度:4/5/6/7/8/9/10/11/12/13/14/15/16/20/24/32位</li><li>单次最多可传送或接收4帧宽度为32位的数据</li></ul>
波特率	<ul style="list-style-type: none"><li>主机模式下可通过内置专用波特率发生器对波特率进行调整，波特率范围为 PCLK1 的 2 分频~256 分频</li><li>从机模式下允许的最大波特率为 PCLK1 的 6 分频</li></ul>
数据缓冲	<ul style="list-style-type: none"><li>带有16字节的数据缓冲区域</li><li>支持双重缓冲</li></ul>
错误监测	<ul style="list-style-type: none"><li>模式故障错误监测</li><li>数据过载错误监测</li><li>数据欠载错误监测</li><li>奇偶校验错误监测</li></ul>
片选信号控制	<ul style="list-style-type: none"><li>每个通道配置四根片选信号线</li><li>可对片选信号和通信时钟的相对时序关系进行调整</li><li>可对连续两次通信之间的片选信号无效时间进行调整</li><li>极性可调</li></ul>
主机模式下的传输控制	<ul style="list-style-type: none"><li>通过将数据写入数据寄存器启动传输</li><li>通信自动挂起功能</li></ul>
中断	<ul style="list-style-type: none"><li>接收数据区域已满</li><li>发送数据区域已空</li><li>SPI 错误（模式/过载/欠载/奇偶校验）</li><li>SPI 空置</li><li>传输完成（仅为事件源）</li></ul>
低功耗控制	可设置模块停止
其他功能	<ul style="list-style-type: none"><li>SPI 初始化功能</li></ul>

**注意：**

- 在主接收模式使用通信自动挂起功能时，由于通信时钟停止，将不会发生过载错误。详情请参考【过载错误】。

**30.2 SPI 系统框图****图 30-1 系统框图****30.3 管脚说明****表 30-2 管脚说明**

管脚名	端口方向	功能
SCK	输入/输出	通信时钟管脚
MOSI	输入/输出	主机数据传输管脚
MISO	输入/输出	从机数据传输管脚
SS0	输入/输出	从机选择输入/输出管脚
SS1	输出	从机选择输出管脚
SS2	输出	从机选择输出管脚
SS3	输出	从机选择输出管脚

## 30.4 SPI 动作系统说明

### 30.4.1 主机模式的管脚状态

SPI 工作在主机模式下时，各个管脚的状态如下表 30-3 所示。

表 30-3 主机模式时 SPI 管脚状态说明

模式		管脚名	管脚状态(PFS.ODS=0)	管脚状态(PFS.ODS=1)
SPI动作 (SPIMDS=0)	主机模式 (MSTR=1、 MODFE=0)	SCK	CMOS输出	OD输出
		SS0~SS3	CMOS输出	OD输出
		MOSI	CMOS输出	OD输出
		MISO	输入	输入
时钟同步运行 (SPIMDS=1)	主机模式 (MSTR=1)	SCK	CMOS输出	OD输出
		SS0~SS3 (不使用)	Hi-Z (可作为通用I/O)	Hi-Z (可作为通用I/O)
		MOSI	CMOS输出	OD输出
		MISO	输入	输入

**注意：**管脚输入类型请设定为 CMOS 输入，输出请设定为高驱动力模式，设定请参照【通用控制寄存器 (PCRxy)】。

### 30.4.2 从机模式的管脚状态

SPI 工作在从机模式下时，各个管脚的状态如下表 30-4 所示。

表 30-4 从机模式时 SPI 管脚状态说明

模式		管脚名	管脚状态 (PFS.ODS=0)	管脚状态(PFS.ODS=1)
SPI动作 (SPIMDS=0)	从机模式 (MSTR=0、 MODFE=0)	SCK	输入	输入
		SS0	输入	输入
		SS1~SS3 (不使 用)	Hi-Z (可作为通用I/O)	Hi-Z (可作为通用I/O)
		MOSI	输入	输入
		MISO (注2)	CMOS输出/Hi-Z	OD输出/Hi-Z
时钟同步运行 (SPIMDS=1)	从机模式 (MSTR=0)	SCK	输入	输入
		SS0~SS3 (不使 用)	Hi-Z (可作为通用I/O)	Hi-Z (可作为通用I/O)
		MOSI	输入	输入
		MISO	CMOS输出	OD输出

**注意：**管脚输入类型请设定为 CMOS 输入，输出请设定为高驱动力模式，设定请参照【通用控制寄存器 (PCRxy)】。

### 30.4.3 SPI 系统连接实例

#### 主机模式

在主机-多从机模式的 SPI 系统结构中，主机驱动 SCK、MOSI 和 SS0 ~ SS3。在 SPI 从机设备 0~3 中，当某一从机的 SS 输入为有效电平时，该从机设备驱动 MISO。

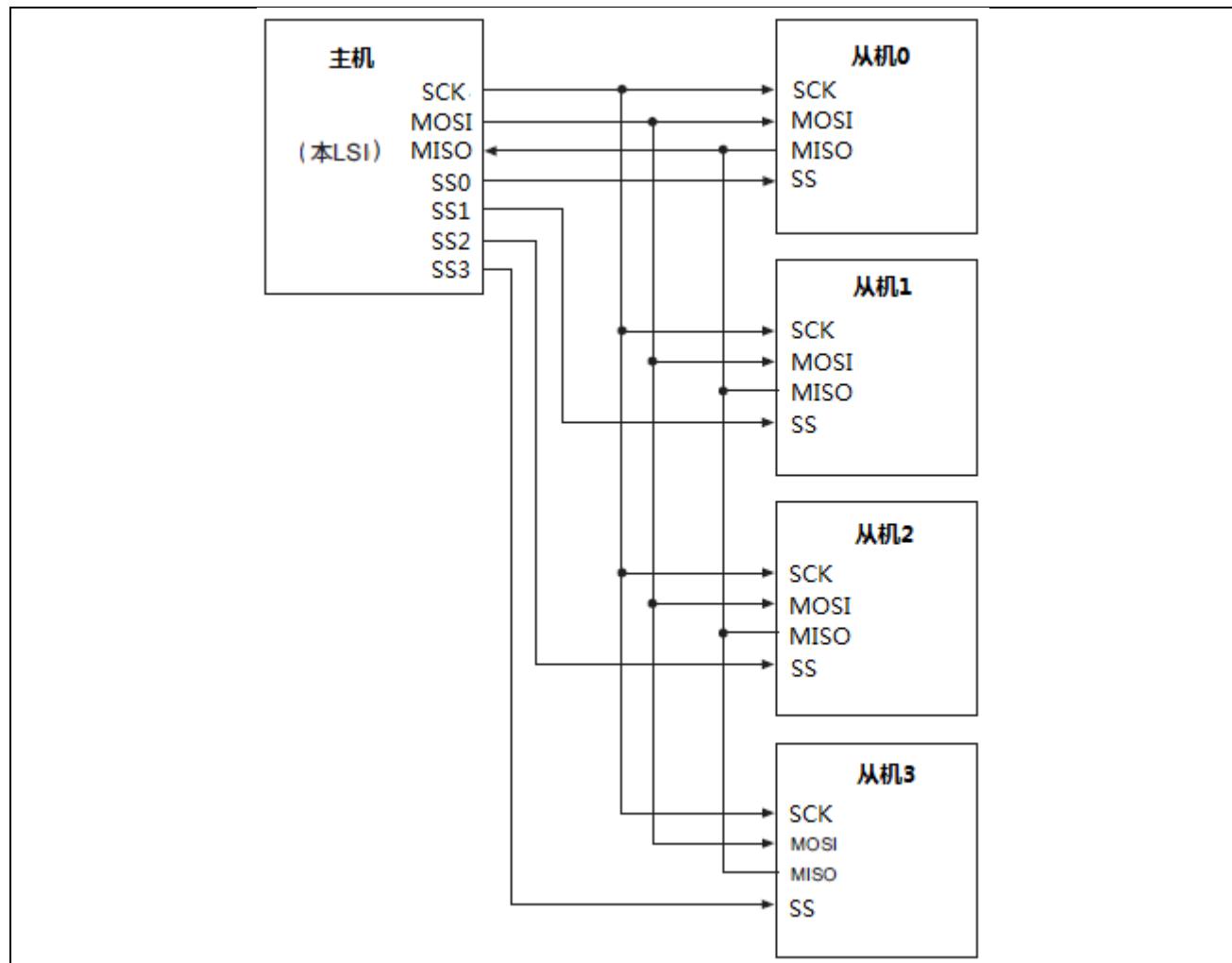


图 30-2 主机模式结构

### 时钟同步运行

在用作时钟同步运行动作的 SPI 系统结构中，主机设备驱动 SCK 和 MOSI，从机设备驱动 MISO。SS 管脚不作使用。

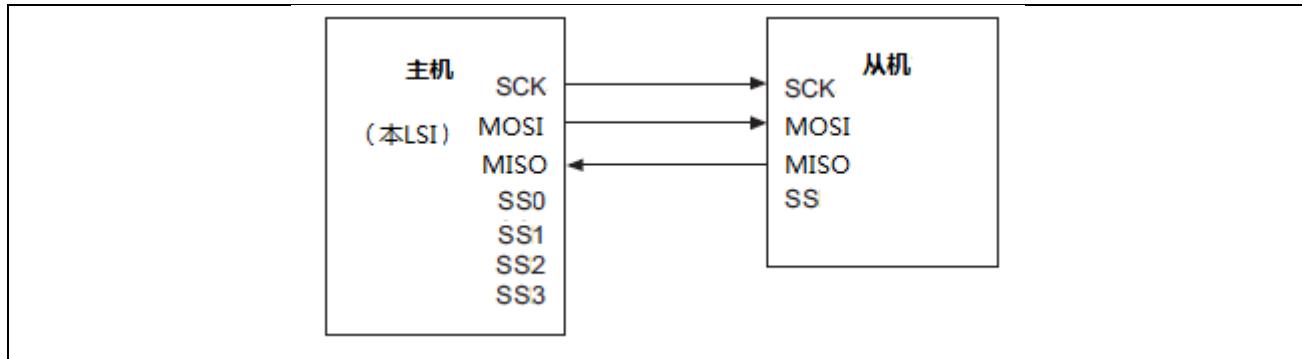


图 30-3 三线式时钟同步运行

## 30.5 数据通信说明

### 30.5.1 波特率

主机模式下，SPI 时钟由内部波特率发生器提供；从机模式下，时钟为 SCK 脚输入。

波特率取决于 SPI\_CFG2.MBR[1:0]位和 SPI\_CFG1.CLKDIV[3:0]的设置。其计算方法如下面公式所示，公式中 N 为 MBR[1:0] 位的设定值，范围是 0~3，M 为 CLKDIV[3:0]位的设定值，范围是 0~15。f<sub>PCLK1</sub> 为 PCLK1 时钟频率。

$$\text{波特率} = \frac{f_{\text{PCLK1}}}{(M+1) \cdot 2^{N+1}}$$

表 30-5 部分设定值的位速

MBR[1:0] 位 的设定值	CLKDIV[3:0] 位 的设定值	分 频 比	波特率			
			PCLK1=5MHz	PCLK1=10MHz	PCLK1=20MHz	PCLK1=40MHz
0	0	2	2.50Mbps	5.00Mbps	10.0Mbps	20.0Mbps
0	1	4	1.25Mbps	2.50Mbps	5.00Mbps	10.0Mbps
0	3	8	625kbps	1.25Mbps	2.50Mbps	5.00Mbps
0	7	16	313kbps	625kbps	1.25Mbps	2.50Mbps
0	15	32	156kbps	313kbps	625kbps	1.25Mbps
1	15	64	78kbps	156kbps	313kbps	625kbps
2	15	128	39kbps	78kbps	156kbps	313kbps
3	15	256	20kbps	39kbps	78kbps	156kbps

### 30.5.2 数据格式

SPI 的数据格式取决于 SPI 命令寄存器 SPI\_CFG2 和 SPI 控制寄存器 SPI\_CR 中的奇偶校验许可位 PAE 的设定值。SPI 将数据寄存器 SPI\_DR 中从 LSB 位开始一定长度（数据长度由寄存器 SPI\_CFG2 中的 DSIZE[3:0]位设置）的数据作为传送对象进行处理，与 MSB/LSB 移位顺序无关。

SPI\_CFG2.DSIZE[3:0]决定数据的位宽，位宽范围为 4~32 位，SPI\_CR.PAE 决定数据最末位，PAE 为 1 时最末位作为奇偶校验位，为 0 时则是数据本身最低位。如图 30-4 所示。

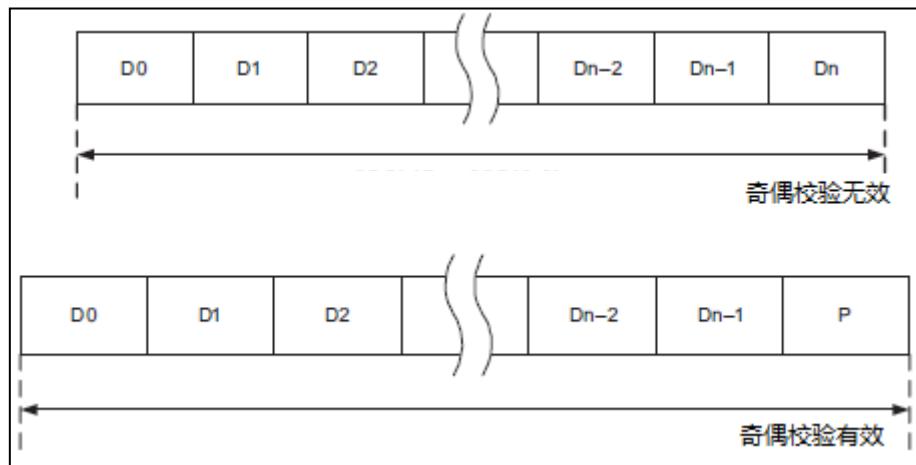


图 30-4 数据格式

SPI 数据发送时，传送数据先进入发送缓冲器 (TX\_BUFF)，再将 TX\_BUFF 的数据复制到移位寄存器 (shifter)，shifter 依次发出数据；SPI 数据接收时，数据从 shifter 依次移入，移入完成后再将 shifter 的数据复制到接收缓冲器 (RX\_BUFF)。

数据传输时，根据移位顺序控制位 SPI\_CFG2.LSBF 和奇偶校验控制位 SPI\_CR.PAE 的设置分为 4 种情况：

- MSB 先传，奇偶校验无效

发送时，数据 d31~d0 按照顺序从 TX\_BUFF 复制到 shifter，按照 d31~d0 的顺序从 shifter 的最高位移出；

接收时，数据 d31~d0 从 shifter 的最低位移入，等到数据全部移入后再将数据复制到 RX\_BUFF。

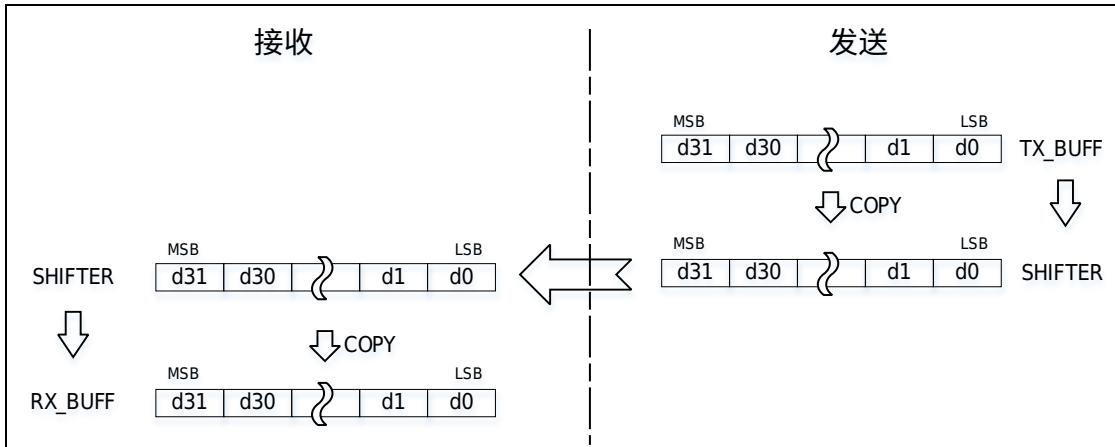


图 30-5 MSB 先传，奇偶校验无效

#### ■ LSB 先传，奇偶校验无效

发送时，数据  $d_{31} \sim d_0$  按照  $d_0 \sim d_{31}$  的顺序从 TX\_BUFF 复制到 shifter，按照  $d_0 \sim d_{31}$  的顺序从 shifter 的最高位移出；

接收时，数据  $d_0 \sim d_{31}$  从 shifter 的最低位移入，等到数据全部移入后，按照  $d_{31} \sim d_0$  的顺序从 shifter 复制到 RX\_BUFF。

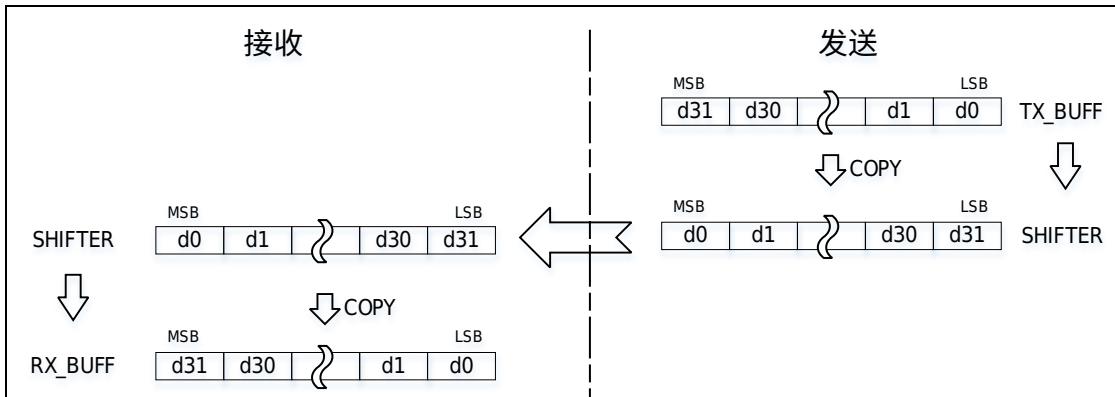


图 30-6 LSB 先传，奇偶校验无效

**■ MSB 先传，奇偶校验有效时**

发送时，先根据 d31~d1 的值计算出奇偶校验位 P 的值，然后用 P 代替 d0，按照 d31~d1，P 的顺序复制到 shifter，按照 d31~P 的顺序从 shifter 的最高位移出；

接收时，数据 d31~P 从 shifter 的最低位移入，在数据复制到 shifter 时，进行奇偶校验。最后再将数据复制到 RX\_BUFF。

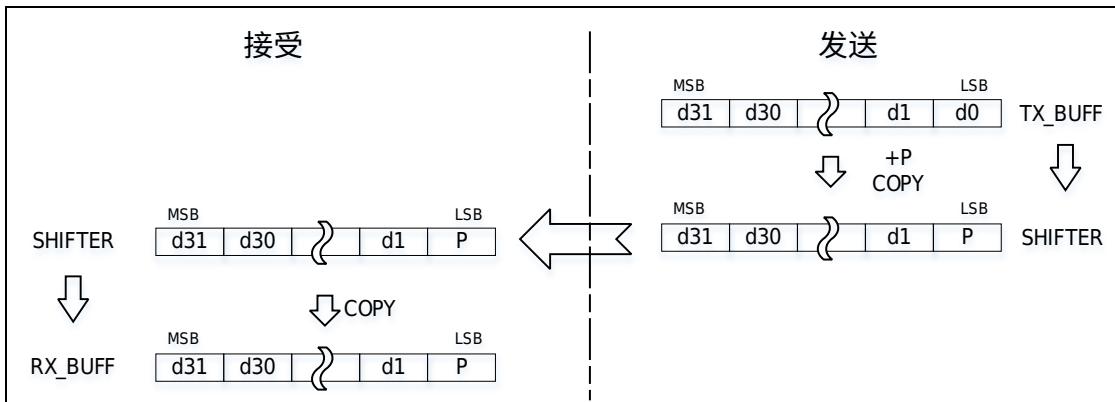


图 30-7 MSB 先传，奇偶校验有效

**■ LSB 先传，奇偶校验有效时**

发送时，先根据 d30~d0 的值计算出奇偶校验位 P 的值，然后用 P 代替 d31，按照 d0~P 的顺序从 TX\_BUFF 复制到 shifter，按照 d0~P 的顺序从 shifter 的最高位移出；

接收时，数据 d0~P 从 shifter 的最低位移入，在数据复制到 shifter 时，进行奇偶校验。数据 d0~P 在复制时重新排列，按照 P~d0 的顺序复制到 RX\_BUFF。

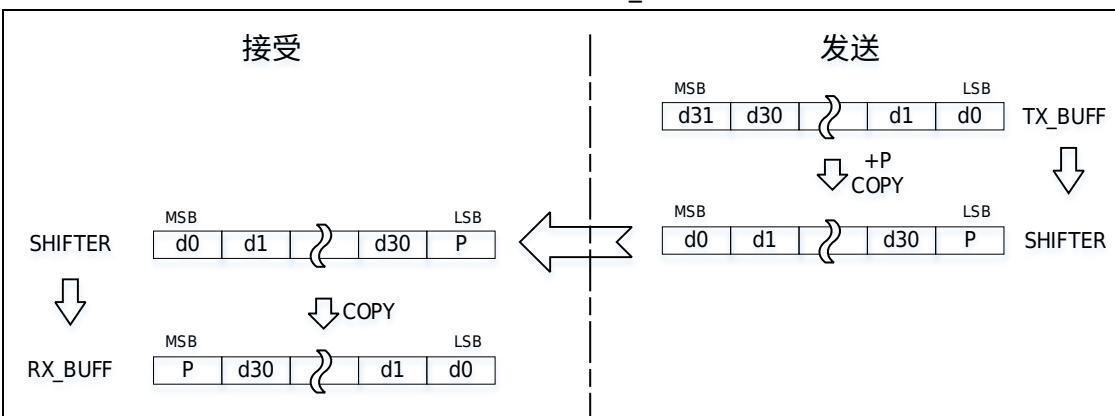


图 30-8 LSB 先传，奇偶校验有效

### 30.5.3 传送格式

#### ■ CPHA=0 的情况

当 SPI\_CFG2.CPHA 位为“0”时，SPI 在 SCK 的奇数边沿进行数据采样，偶数边沿进行数据更新。

图 30-9 是 CPHA=0 时 SPI 的传送时序图。当 SS<sub>i</sub> 信号的输入电平变为有效电平时，MOSI/MISO 开始更新传输数据。在 SS<sub>i</sub> 信号变为有效后的第一个 SCK 信号边沿进行第一次数据采样，在此之后，每过一个 SCK 周期对数据进行一次采样。每次采样后的 1/2 个 SCK 周期时 MOSI/MISO 信号上数据进行更新。CPOL 位的设定值不影响 SCK 信号的运行时序，而只影响信号的极性。

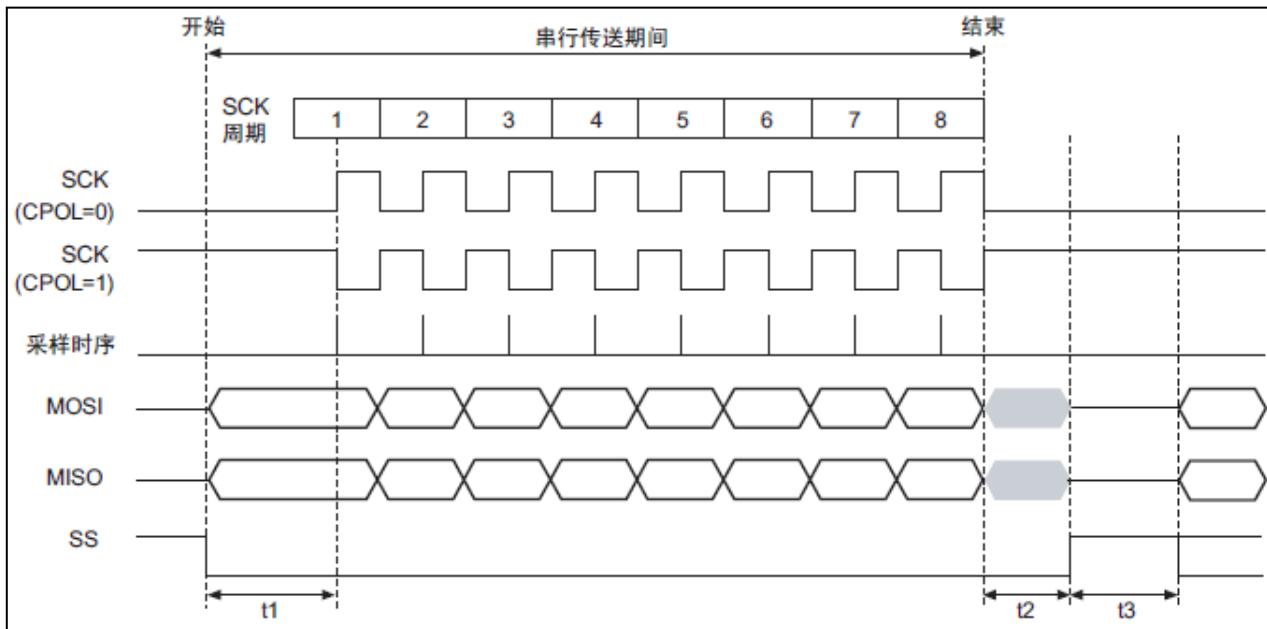


图 30-9 数据传送格式图 (CPHA=0)

上图中 t1 表示从 SS 信号有效到 SCK 振荡的间隔时间 (SCK 延迟时间，可通过设置 SPI\_CFG1.MSSI[2:0]和 SPI\_CFG2.MSSIE 进行控制)，t2 表示从 SCK 振荡停止到 SS 信号变成无效的间隔期间 (SS 无效延迟时间，它可以通过设置 SPI\_CFG1.MSSDL[2:0]和 SPI\_CFG2.MSSDLE 来控制)，t3 表示在串行传送结束后到下次传送开始的最小等待时间 (下次存取延迟，可以通过设置 SPI\_CFG1.MIDI[2:0]和 SPI\_CFG2.MIDIE 来控制)。t1、t2 和 t3 由 SPI 系统上的主机设备进行控制。详细可参考【SPI 运行模式时的主机动作】这一章节。

#### ■ CPHA=1 的情况

当 SPI\_CFG2.CPHA 位为“1”时，SPI 在 SCK 的奇数边沿进行数据更新，偶数边沿进行数据采样。

图 30-10 是 CPHA=1 时 SPI 的传送时序图。MOSI/MISO 在 SS<sub>i</sub> 信号变为有效电平后的第一个 SCK 信号边沿时开始传输数据的更新。在此之后，每过一个 SCK 周期对数据进行一次更新。每次更新后的 1/2 个 SCK 周期时对数据进行采样。SPI\_CFG2.CPOL 位的设定值不影响 SCK 信号的运行时序而只影响信号的极性。

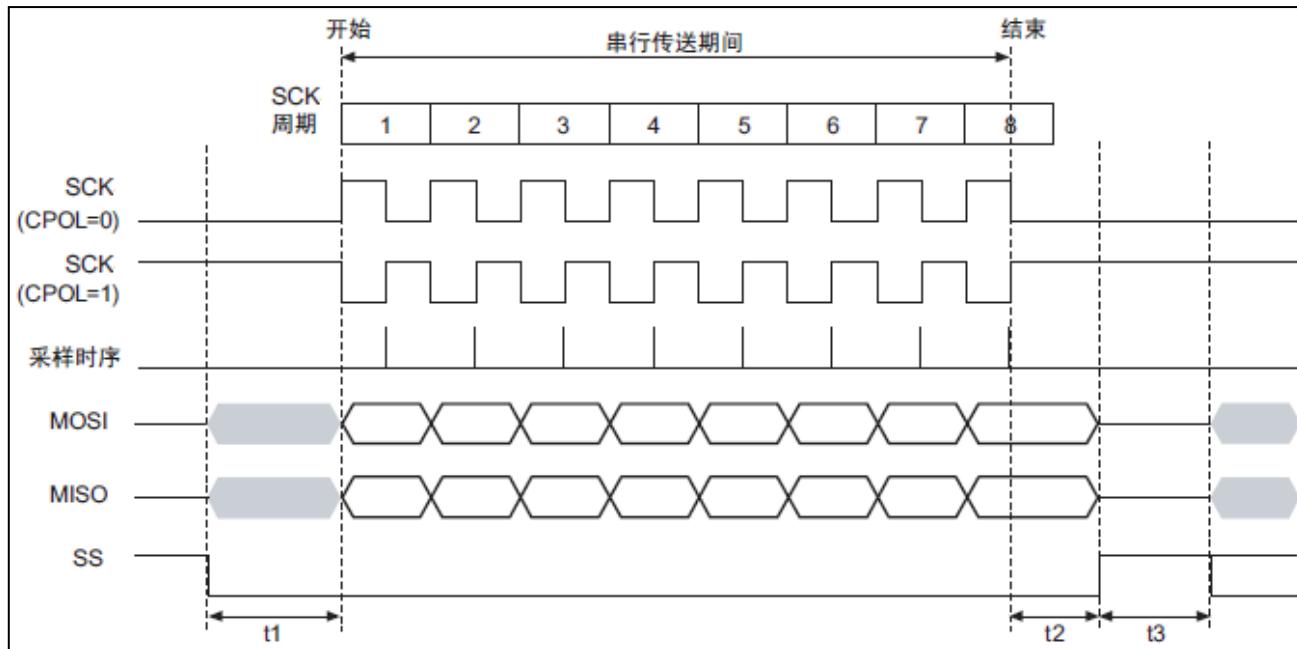


图 30-10 数据传送格式 (CPHA=1)

t1、t2、t3 和 CPHA 位为 0 的情况相同。

### 30.5.4 通信方式

本 SPI 有全双工同步串行通信和只发送串行通信两种通信方式，可通过 SPI 控制寄存器（SPI\_CR）的 TXMDS 位进行选择。

#### ■ 全双工同步串行通信方式

当 SPI\_CR.TXMDS 位为“0”时，SPI 运行在全双工同步串行通信方式。如图 30-11 所示，SPI\_CFG1.FTHLV[1:0] 位为“00b”，SPI\_CFG2.CPHA 位为“1”并且 SPI\_CFG2.CPOL 位为“0”，SPI 进行 8 位串行传送。

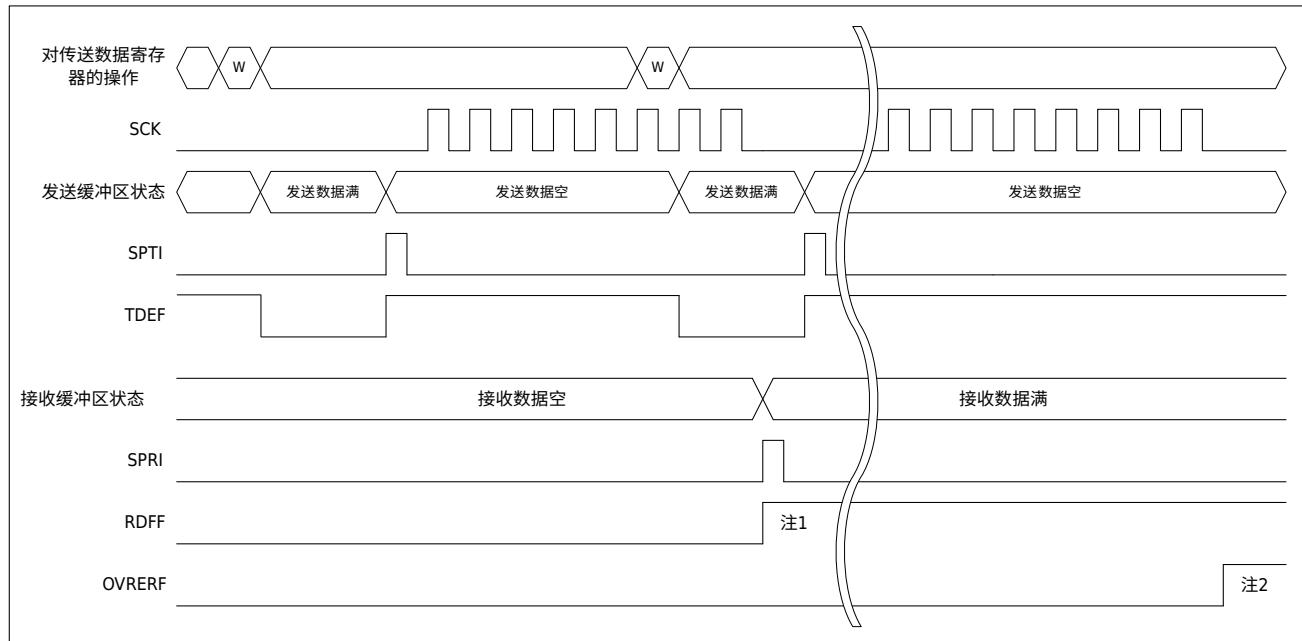


图 30-11 全双工同步串行通信

#### 注：

- 当本次串行传输结束时，如果接收数据缓冲寄存器为空，SPI 将会把接收到的数据从移位寄存器复制到接收数据缓冲寄存器中，接收数据缓冲寄存器满的标志位被置成 1（RDFF），并产生一个接收数据满的中断请求（SPRI）。
- 当本次串行传输结束时，如果接收数据缓冲寄存器中还保持着上次收到的数据而没有被系统读取，SPI 会将数据过载标志位 OVRERF 置成 1，本次数据接收无效，接收移位寄存器中的数据将被丢弃。

### ■ 只发送通信方式

当 SPI\_CR.TXMDS 位为“1”时，SPI 运行在只发送通信方式。如图 30-12 所示，SPI\_CFG1.FTHLV[1:0]位为“00b”，SPI\_CFG2.CPHA 位为“1”并且 SPI\_CFG2.CPOL 位为“0”，SPI 进行 8 位串行传送。

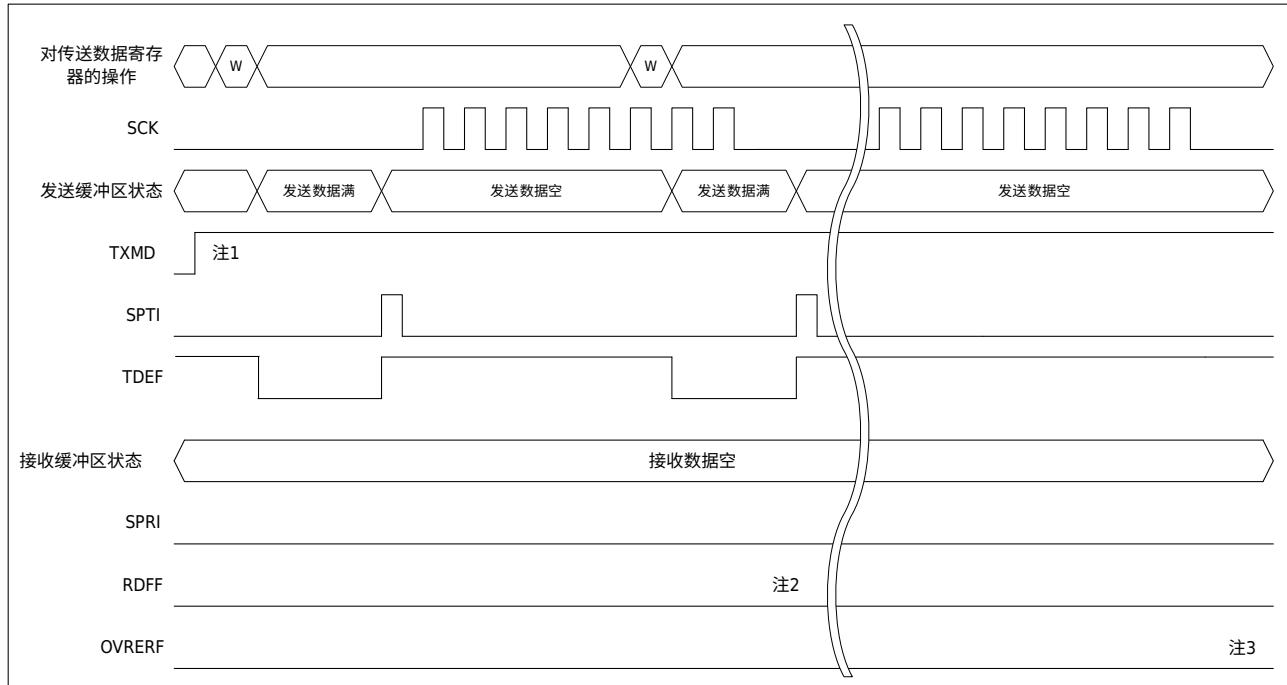


图 30-12 只进行发送通信

### 注：

- 在设置进入只发送通信方式之前，请确保接收缓冲寄存器中没有未读取的数据（即 RDFF 为 0）且未出现数据过载错误（即 OVRERF 为 0）。
- 在只发送通信方式下，当本次串行传输结束时，即使接收数据缓冲寄存器为空，也不会有数据被接收，RDFF 将始终保持 0 的状态。
- 在只发送通信方式下，由于接收数据缓冲寄存器始终为空，所以不会发生数据过载错误，OVRERF 标志位始终保持 0 的状态。

### 30.5.5 通信间隔时间

#### ■ 常规设定下的通信间隔时间

当 SPI\_CFG1.CTMDS 位为“0”时，SPI 使用常规通信设定。连续两次通信之间的间隔时间可以通过 SPI\_CFG1.MIDI[2:0]和 SPI\_CFG2.MIDIE 来进行设定，最小间隔时间为  $1*SCK+2*PCLK1$ 。如图 30-13 所示，SPI\_CFG2.CPHA 位为“1”并且 SPI\_CFG2.CPOL 位为“0”，SPI 进行串行传送。

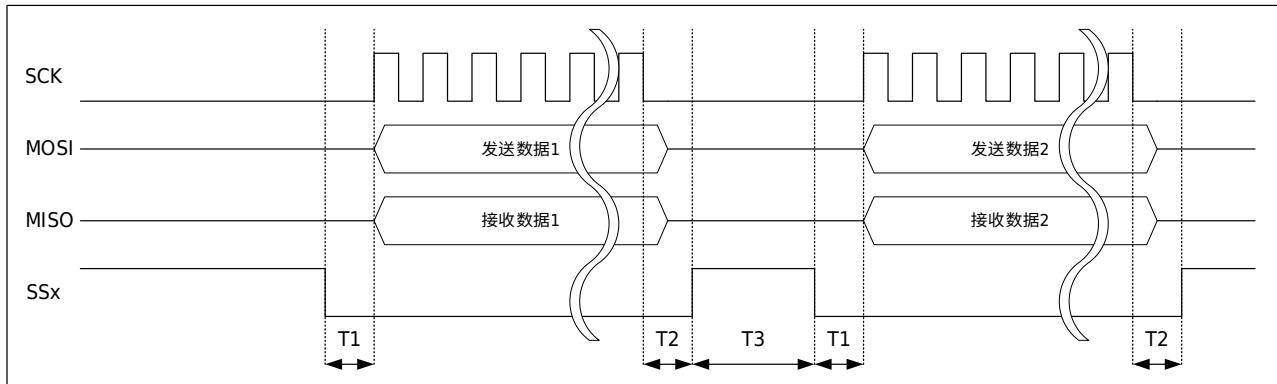


图 30-13 常规通信

在上图中， $t_1$  表示从 SS 信号有效到 SCK 第一边沿的时间间隔（SCK 延迟时间，可通过设置 SPI\_CFG1.MSSI[2:0]和 SPI\_CFG2.MSSIE 进行控制）， $t_2$  表示从 SCK 最后边沿到 SS 信号无效的时间间隔（SS 无效延迟时间，它可以通过设置 SPI\_CFG1.MSSDL[2:0]和 SPI\_CFG2.MSSDLE 来控制）， $t_3$  表示从本次传输结束到下一次传输开始的最小等待时间（下次存取数据延迟，可以通过设置 SPI\_CFG1.MIDI[2:0]和 SPI\_CFG2.MIDIE 来控制）。

### ■ 连续通信设定下的通信间隔时间

当 SPI\_CFG1.CTMDS 位为“1”时，SPI 使用连续通信设定。可实现完全连续不间断的数据通信。

如图 30-14 所示，SPI\_CFG2.CPHA 位为“1”并且 SPI\_CFG2.CPOL 位为“0”，SPI 进行串行传送。

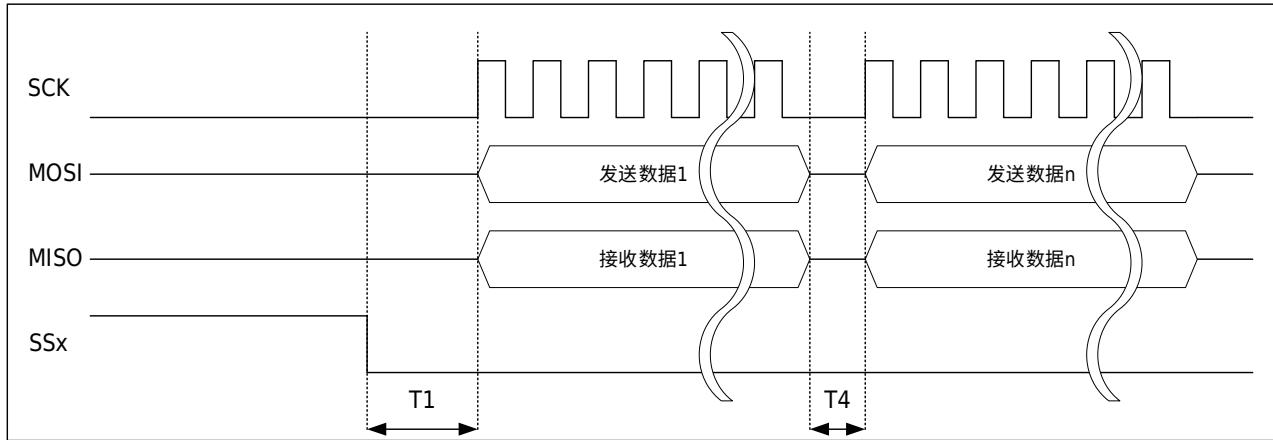


图 30-14 连续通信

在上图中，t1 表示从 SS 信号有效到 SCK 第一边沿的时间间隔（SCK 延迟时间，可通过设置 SPI\_CFG1.MSSI[2:0]和 SPI\_CFG2.MSSIE 进行控制），t4 表示每次传输之间的时间间隔，可通过设置 SPI\_CFG1.MSSI[2:0]和 SPI\_CFG2.MSSIE 进行控制。t1 和 t4 之间的关系定义为  $t4=t1-1*SCK$ 。

注：

- 连续通信模式下，主机如果没有新的数据写入发送缓冲寄存器，则将进入暂停状态，SCK 停止输出，SS 信号保持有效状态，等待新的发送数据写入发送缓冲寄存器后继续下一帧数据的传输。
- 连续通信模式下，从机如果没有新的数据写入发送缓冲寄存器，则将发生欠载错误，SPI\_SR.MODFERF 和 SPI\_SR.UDRERF 将置 1，详细请参考关于欠载错误的描述。
- 时钟同步模式下的连续通讯，如果使用 CPHA=0 的传送格式，在要保证数据在 CPHA 设置成 0 之前写入。
- 使用 CPHA=0 的传送格式时，从机需要在上一帧数据发送结束前向发送缓冲寄存器写入新的数据，否则无论主机是否开始下一帧数据的通讯，从机都将发生欠载错误。
- 连续通信模式下，也可以通过 MSSI 来调整上一帧数据最后一个 SCK 和下一帧数据第一个 SCK 之间的时间间隔。

## 30.6 运行说明

### 30.6.1 运行模式概要

本 SPI 支持 4 线式 SPI 模式和 3 线式时钟同步运行模式。每个运行模式下都可作为主机或者从机进行串行通信。设置 SPI 控制寄存器 SPI\_CR 中的 MSTR 和 SPIMDS 位设定 SPI 的模式。SPI 的模式和 SPI\_CR 寄存器设定的关系以及各模式概要如表 30-6 所示。

表 30-6 SPI 模式和寄存器设定关系

模式	主机 (SPI运行)	从机 (SPI运行)	主机 (时钟同步运行)	从机 (时钟同步运行)
MSTR位的设定	1	0	1	0
SPIMDS位的设定	0	0	1	1
SCK信号	输出	输入	输出	输入
MOSI信号	输出	输入	输出	输入
MISO信号	输入	输出/Hi-Z	输入	输出
SS0信号	输出	输入	Hi-Z (不使用)	Hi-Z (不使用)
SS1~SS3信号	输出	Hi-Z	Hi-Z (不使用)	Hi-Z (不使用)
SS极性变更功能	有	有	-	-
最大传输速率	~PCLK1/2	~PCLK1/6	~PCLK1/2	~PCLK1/6
时钟源	内部波特率发生器	SCK输入	内部波特率发生器	SCK输入
时钟极性	2种	2种	2种	2种
时钟相位	2种	2种	2种	1种 (CHPA=1)
开始传送位	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB
传送数据长度	4~32位	4~32位	4~32位	4~32位
SCK延迟控制	有	无	有	无
SS无效延迟控制	有	无	有	无
下次存取延迟控制	有	无	有	无
传送启动方法	通过发送缓冲器空中断请求, 写发送缓冲器	SS输入有效或SCK时钟边沿	通过发送缓冲器空中断请求, 写发送缓冲器	SCK振荡
模式故障错误	无	有	无	有
发送缓冲器空检测	有	有	有	有
接收缓冲器满检测	有 (注1)	有 (注1)	有 (注1)	有 (注1)
过载错误检测	有 (注1)	有 (注1)	有 (注1)	有 (注1)
奇偶校验错误检测	有 (注1、注2)	有 (注1、注2)	有 (注1、注2)	有 (注1、注2)
欠载错误检测	有	无	有	无

注:

- 当 SPI\_CR.TXMDS 位为“1”时, 不进行接收缓冲器满的检测、过载错误的检测和奇偶校验错误的检测。

- 当 SPI\_CR.PAE 位为“0”时，不进行奇偶校验错误的检测。

### 30.6.2 SPI 运行模式时的主机动作

#### ■ 作为主机时的动作说明

在 SPI 数据发送缓冲寄存器 (TX\_BUFF) 为空的状态下 (状态寄存器 SPI\_SR 中的 TDEF 标志位为 1)，将格式控制寄存器 SPI\_CFG1 的 FTHLV[1:0] 位所设帧数长度的数据写到 SPI 数据寄存器 (SPI\_DR) 后，SPI 会将 SPI\_DR 数据更新至 TX\_BUFF 中。此时若移位寄存器 (shifter) 为空，SPI 将 TX\_BUFF 的数据复制到移位寄存器开始串行传送。

当发送数据复制到 shifter 后，SPI 将会 shifter 状态改为满状态；当串行传送结束，改为空状态。shifter 的状态无法进行读取。

当 SPI 发送完最后采样时序所需的 SCK 边沿后，本次串行传送结束，结束时间与 SPI\_CFG2.CPHA 位无关。在接收缓冲器 (RX\_BUFF) 为空的情况下，串行传送结束后 SPI 将 shifter 中的数据复制到 RX\_BUFF 中，可通过数据寄存器 SPI\_DR 进行读取。

最后的采样时序取决于传送数据的位长，主控模式的 SPI 数据长度取决于 SPI\_CFG2.DSIZE[3:0] 位的设定值，SS 输出引脚的极性取决于 SPI\_CFG1 寄存器的设定值。有关 SPI 传送格式的详细内容，请参照 32.5.3 传送格式。

#### ■ SPI 主机模式的初始化

1. 设置通信配置寄存器 1 (SPI\_CFG1)，包括有波特率的设定，使用帧数的设定，各种延迟时间的设定等。
2. 设置通信配置寄存器 2 (SPI\_CFG2)，包括有 SS 电平设定，数据移位顺序设定，各种延迟的允许位的设定，数据格式及时钟极性相位的设定等。
3. 如需要使用中断，请设置系统的中断寄存器。
4. 如需要使用 DMA，请设置 DMA 的相关寄存器。
5. 设定输入输出管脚。
6. 设定 SPI 控制寄存器 SPI\_CR，包括有模式及运行方式的设定，自诊断功能的设定，奇偶校验的设定等。
7. 确认 SPI\_CR 寄存器的设置。
8. 清除各种标志位。
9. 设置中断许可位。
10. 将控制寄存器 SPI\_CR 的 SPE 位设置成 1，动作开始。

### 30.6.3 SPI 运行模式时的从机动作

#### ■ SPI 作为从机时的动作说明

当 SPI\_CFG2.CPHA 位为 0 时，如果 SPI 检测到 SS0 输入信号变为有效电平，就需要开始向 MISO 输出信号驱动有效数据。因此，在 CPHA 位为 0 时，将 SS0 输入信号电平从无效变为有效视为开始串行传送的触发信号。

当 CPHA 位为“1”时，如果在 SS0 输入信号为有效电平的状态下 SPI 检测到最初的 SCK 边沿，就需要开始向 MISO 输出信号驱动有效数据。因此，在 CPHA 位为“1”时，将 SS0 信号处于有效电平状态下的首个 SCK 边沿视为开始串行传送的触发信号。

如果 SPI 在 shifter 为空的状态下检测到串行传送的开始，就将 shifter 改为满状态，并且不能在串行传送过程中将数据从 TX\_BUFF 传送到 shifter。如果在开始串行传送前 shifter 已满，SPI 就保持 shifter 的满状态。

如果 SPI 检测到最后采样时序的 SCK 边沿，则本次串行传输结束，结束时间与 SPI\_CFG2.CPHA 位无关。在 RX\_BUFF 为空的情况下，串行传送结束后 SPI 将 shifter 的接收数据复制到 RX\_BUFF。可通过访问 SPI\_DR 读取该数据。SPI 在串行传送结束后将 shifter 改为空状态，该状态与 RX\_BUFF 的状态无关。

在串行传送期间，如果 SPI 检测到 SS0 输入信号无效，就发生模式故障错误。

最后的采样时序取决于传送数据的位长，从属模式的 SPI 的数据长度取决于 SPI\_CFG2.DSIZE[3:0] 位的设定值，SS0 输入信号的极性取决于 SPI\_CFG1.SS0PV 位的设定值。有关 SPI 传送格式的详细内容，请参照 32.5.3 传送格式。

#### 注意：

- 当 SPI\_CFG2.CPHA 位为“0”时，将 SS0 输入信号电平从无效变为有效视为开始串行传送的触发信号。由于在从机模式的结构中 SS0 输入信号被固定为有效状态，此时 SPI 将无法正常开始串行传送。因此，在 SS0 输入信号被固定为有效状态的结构中，要使从机模式的 SPI 正常进行发送和接收，必须将 CPHA 位设置为“1”。如果需要将 CPHA 位设成“0”，则不能固定 SS0 输入信号。

#### ■ SPI 从机模式的初始化

1. 设定通信配置寄存器 1 (SPI\_CFG1)，主要包括使用帧数的设定。
2. 设置通信配置寄存器 2 (SPI\_CFG2)，包括有传输速率，数据格式及时钟极性相位的设定等。
3. 如需要使用中断，请设置系统的中断寄存器。
4. 如需要使用 DMA，请设置 DMA 的相关寄存器。
5. 设定输入输出管脚。
6. 设定 SPI 控制寄存器 SPI\_CR，包括有模式及运行方式的设定，自诊断功能的设定，奇偶校验的设定等。
7. 确认 SPI\_CR 寄存器的设置。

8. 清除各种标志位。
9. 设置中断许可位。
10. 将控制寄存器 SPI\_CR 的 SPE 位设置成 1，动作开始。

### 30.6.4 时钟同步运行模式中的主机动作

当 SPI 控制寄存器 SPI\_CR 中的 SPIMDS 位为 1 时，SPI 处于时钟同步运行模式。在该模式动作时，SPI 只使用 SCK、MOSI 和 MISO 这 3 个管脚进行通信，SSI 管脚被释放可用于普通 I/O 功能。

尽管时钟同步运行模式时不使用 SSI 管脚，但模块内部的运行和 SPI 运行模式是相同的。

#### ■ SPI 作为主机时的动作说明

在 SPI 数据发送缓冲寄存器 (TX\_BUFF) 为空的状态下（状态寄存器 SPI\_SR 中的 TDEF 标志位为 0），将格式控制寄存器 SPI\_CFG1 的 FTHLV[1:0] 位所设帧数长度的数据写到 SPI 数据寄存器 (SPI\_DR) 后，SPI 会将 SPI\_DR 数据更新至 TX\_BUFF 中。此时若移位寄存器 (shifter) 为空，SPI 将 TX\_BUFF 的数据复制到移位寄存器开始串行传送。

当发送数据复制到 shifter 后，SPI 将会 shifter 状态改为满状态；当串行传送结束，改为空状态。shifter 的状态无法进行读取。

当 SPI 发送完最后采样时序所需的 SCK 边沿后，本次串行传送结束，结束时间与 SPI\_CFG2.CPHA 位无关。在接收缓冲器 (RX\_BUFF) 为空的情况下，串行传送结束后 SPI 将 shifter 中的数据复制到 RX\_BUFF 中，可通过数据寄存器 SPI\_DR 进行读取。

最后的采样时序取决于传送数据的位长，主控模式的 SPI 数据长度取决于 SPI\_CFG2.DSIZE[3:0] 位的设定值。有关 SPI 传送格式的详细内容，请参照 32.5.3 传送格式。

#### ■ 时钟同步运行模式时主机的初始化设置

1. 设置通信配置寄存器 1 (SPI\_CFG1)，包括有波特率的设定，使用帧数的设定，各种迟延时间的设定等
2. 设置通信配置寄存器 2 (SPI\_CFG2)，包括有数据移位顺序设定，各种延迟的允许位的设定，数据格式及时钟极性相位的设定等。
3. 如需要使用中断，请设置系统的中断寄存器。
4. 如需要使用 DMA，请设置 DMA 的相关寄存器。
5. 设定输入输出管脚。
6. 设定 SPI 控制寄存器 SPI\_CR，包括有模式及运行方式的设定，自诊断功能的设定，奇偶校验的设定等。
7. 确认 SPI\_CR 寄存器的设置。
8. 清除各种标志位。
9. 设置中断许可位。
10. 将控制寄存器 SPI\_CR 的 SPE 位设置成 1，动作开始。

### 30.6.5 时钟同步运行模式中的从机动作

#### ■ SPI 作为从机时的动作说明

当 SPI\_CFG2.CPHA 位为 0 时，需要 SPI 检测到 SS0 输入信号变为有效电平作为开始串行通信的触发信号。由于时钟同步运行模式下不使用 SS0 管脚，因此，在 CPHA 位为 0 时无法进行正常通信。

当 CPHA 位为“1”时，如果在 SS0 输入信号为有效电平的状态下 SPI 检测到最初的 SCK 边沿，就需要开始向 MISO 输出信号驱动有效数据。由于时钟同步运行模式下不使用 SS0 管脚，因此，在 CPHA 位为“1”时，将首个 SCK 边沿视为开始串行传送的触发信号。

如果 SPI 在 shifter 为空的状态下检测到串行传送的开始，就将 shifter 改为满状态，并且不能在串行传送过程中将数据从 TX\_BUFF 传送到 shifter。如果在开始串行传送前 shifter 已满，SPI 就保持 shifter 的满状态。

如果 SPI 检测到最后采样时序的 SCK 边沿，则本次串行传输结束。在 RX\_BUFF 为空的情况下，串行传送结束后 SPI 将 shifter 的接收数据复制到 RX\_BUFF。可通过访问 SPI\_DR 读取该数据。SPI 在串行传送结束后将 shifter 改为空状态，该状态与 RX\_BUFF 的状态无关。

最后的采样时序取决于传送数据的位长，从属模式的 SPI 的数据长度取决于 SPI\_CFG2.DSIZE[3:0]位的设定值。

#### ■ 时钟同步运行模式时从机的初始化设置

1. 设定通信配置寄存器 1 (SPI\_CFG1)，主要包括使用帧数的设定。
2. 设置通信配置寄存器 2 (SPI\_CFG2)，包括有传输速率，数据格式及时钟极性相位的设定等。
3. 如需要使用中断，请设置系统的中断寄存器。
4. 如需要使用 DMA，请设置 DMA 的相关寄存器。
5. 设定输入输出管脚。
6. 设定 SPI 控制寄存器 SPI\_CR，包括有模式及运行方式的设定，自诊断功能的设定，奇偶校验的设定等。
7. 确认 SPI\_CR 寄存器的设置。
8. 清除各种标志位。
9. 设置中断许可位。
10. 将控制寄存器 SPI\_CR 的 SPE 位设置成 1，动作开始。

### 30.6.6 几种 SPI 动作的处理流程

#### ■ SPI 作为主机时的数据传送处理流程

1. 等待数据发送缓冲寄存器空的中断或通过轮询方式确认数据发送缓冲寄存器处于空状态。
2. 向数据寄存器 SPI\_DR 写入要发送的数据。
3. 重复①②步骤直到最后一个数据发送完成。
4. 将发送数据寄存器空中断的允许位 TXIE 清零，同时将 SPI 闲置状态中断允许位 IDIE 设为 1。
5. 发送 SPI 闲置状态中断。
6. 将 SPE 置 0，停止 SPI 动作，同时将 IDIE 清零。

#### ■ 数据接收处理流程

1. 等待数据接收缓冲寄存器满的中断或者通过轮询方式确认数据接收缓冲寄存器处于满状态。
2. 通过访问 SPI\_DR 从接收缓冲寄存器读取数据。
3. 重复①②步骤直到最后一个接收数据被读取。
4. 将数据接收缓冲寄存器满的中断允许位 RXIE 清零。

#### ■ 通信错误处理流程

1. 等待通信错误中断或者通过轮询方式确认通信错误标志位 (MODFERF/OVERRF/UDRERF/PERF) 被置成 1。
2. 确认 SSO 状态，排除模式故障错误。
3. 将 SPE 清零，停止 SPI 动作。
4. 通过错误标志位确定通信错误种类，进行通讯错误处理。
5. 将错误标志位清零。
6. 启动 SPI，重新开始通信。

## 30.7 奇偶校验位自诊断

奇偶校验电路由传输数据的奇偶校验位和接收数据的错误检测部分构成。可按照下图所示流程使用自诊断功能对奇偶校验电路进行故障诊断。

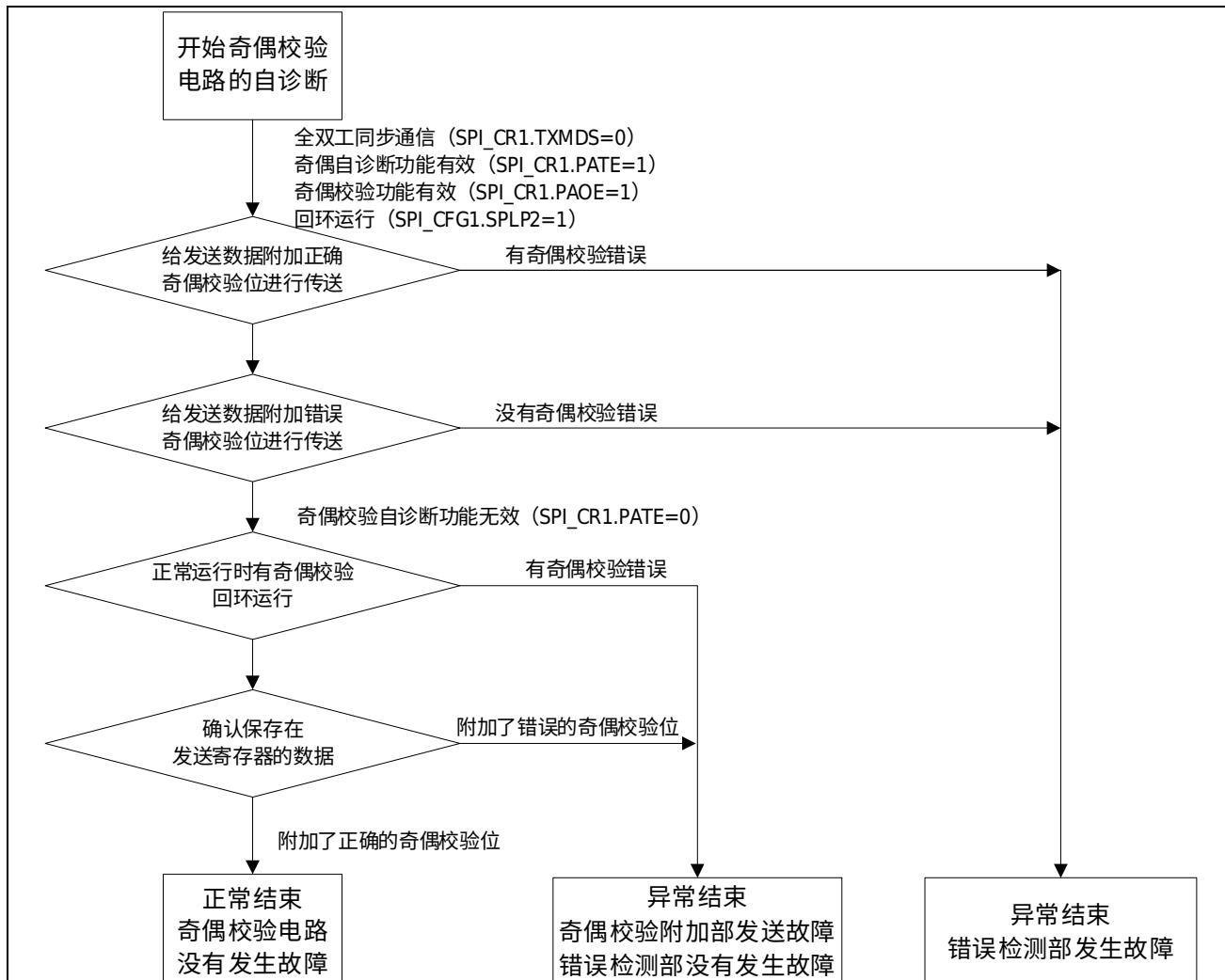


图 30-15 奇偶校验流程

## 30.8 错误检测

在正常的 SPI 串行传送中，系统通过对 SPI\_DR 寄存器写操作进行数据串行发送，通过对 SPI\_DR 寄存器的读操作获得串行接收的数据。但由于发送/接收缓冲器的状态以及串行传送开始/结束时的 SPI 的状态，某些情况下有可能会发生异常传送。当异常传送发生时，SPI 会将此次传输检测为欠载错误、过载错误、奇偶校验错误或者模式故障错误。异常传送和 SPI 错误检测的对应关系如下表 30-7 所示。

表 30-7 错误检测对应表

序号	发生条件	SPI运行	检测错误
①	在发送缓冲器满的状态下写SPI_DR寄存器	• 保持发送缓冲器内容	无
②		• 写数据丢失	
③	在接收缓冲器为空的状态下读SPI_DR寄存器	输出上次的串行接收数据	无
④	从机模式中，发送数据未转移到移位寄存器的状态下开始串行传送	• 中止串行传输	欠载错误
		• 丢失发送和接收数据	
		• 停止驱动MISO输出信号	
		• 停止SPI功能	
⑤	从机模式：SS0管脚有效电平宽度没有达到数据传输所需时间。	• 中止传输	模式错误
⑥	在接收缓冲器满的状态下结束串行传送	• 丢失发送和接收数据	
⑦	在进行全双工同步串行通信，并且奇偶校验功能有效的情况下，接收到错误的奇偶校验位	• 停止SPI功能	
⑧	在接收缓冲器满的状态下读SPI_DR寄存器	• 保持接收缓冲器内容	过载错误
⑨		• 接收数据丢失	
⑩	在进行全双工同步串行通信，并且奇偶校验功能有效的情况下，接收到错误的奇偶校验位	奇偶校验错误标志有效	奇偶校验错误
⑪		奇偶校验错误标志无效	

①描述的情况，SPI 不发生检测错误。为防止在写数据到 SPI\_DR 寄存器的过程中发生数据遗漏，必须通过发送缓冲器空的中断把数据写入 SPI\_DR 寄存器。同理，②的情况下，SPI 也不发生检测错误。为防止无关数据被读入，SPI\_DR 的数据读取必须通过接收寄存器满的中断请求来进行。

### 30.8.1 欠载错误

MSTR 位为“0”时，SPI 以从机状态运行，如果 SPE 置为“1”后，SS0 管脚接收到有效电平前，传送数据还未准备好，则 SPI 发生欠载错误，SPI\_SR.MODFERF 和 SPI\_SR.UDRERF 标志将被置为“1”。

当检测到欠载错误后，SPI 将停止驱动信号输出，同时把 SPI\_CR.SPE 置为“0”。

监测欠载错误可通过直接访问 SPI\_SR 寄存器，或者使用 SPI 错误中断读取 SPI\_SR 等方式进行。如果不使用错误中断，请使用轮询方式来监测欠载错误。

当 SPI\_SR.MODFERF 为“1”时，系统禁止对 SPE 位写“1”。要将 SPI\_CR.SPE 置成“1”使能 SPI 功能必须先将 MODFERF 标志清零。

### 30.8.2 模式错误

SPI 在主机模式时，SPI\_CR.MODFE 请勿设定为“1”。在从机模式时，当 SSI 有效电平宽度没有达到传输数据所需时间时，发生模式故障，SPI\_SR.MODFERF 置“1”，SPI\_CR.SPE 置为“0”。需要传送时，对 SPI\_SR.MODFERF 清零后，再把 SPI\_CR.SPE 置“1”。

### 30.8.3 过载错误

如果在接收缓冲器满的状态下结束串行传送，SPI 发生过载错误，SPI\_SR.OVRERF 标志被置成“1”。因为在 OVRERF 标志为“1”的状态下 SPI 不会把移位寄存器的数据复制到接收缓冲器，所以接收缓冲器中保存的是发生错误前的接收数据。需要在 OVRERF 标志为 1 的状态下读 SPI\_SR 寄存器后才能给 OVRERF 标志写“0”。

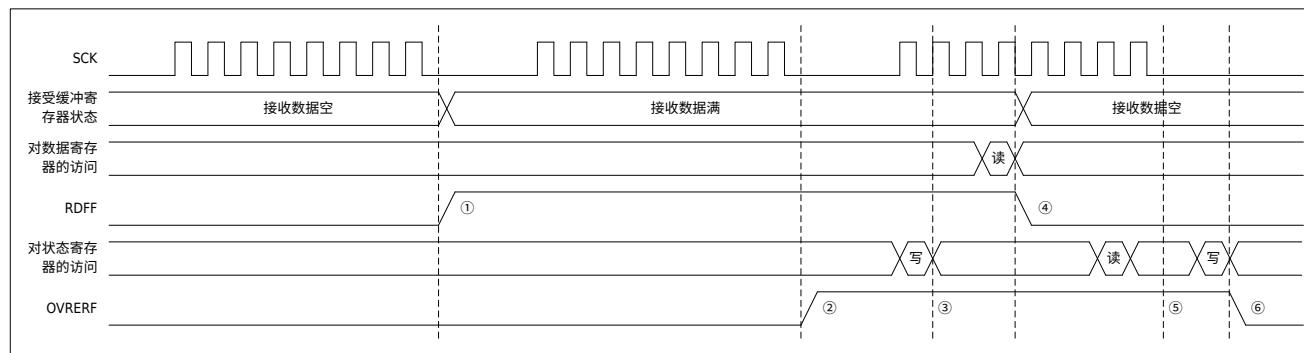


图 30-16 过载错误处理

以下说明在图中的①~⑥所示的时序中标志的运行内容：

- ① 在接收缓冲器空的状态下结束串行传送，SPI 正常动作，将移位寄存器的数据复制到接收缓冲寄存器中，并将 RDFF 标志置“1”。
- ② 在接收缓冲器满的状态下结束串行传送，SPI 检测到过载错误并且将 OVRERF 标志置“1”。SPI 将不会把移位寄存器的数据复制到接收缓冲器。即使 PAE 位为“1”时，也不检测奇偶校验错误。
- ③ 清零 OVRERF 标志位。
- ④ 对数据寄存器 SPI\_DR 进行读取访问，SPI 就能读到接收缓冲器的数据。RDFF 标志变为 0。即使此时接收缓冲器状态为空，OVRERF 标志也不会变为“0”。
- ⑤ 在 OVRERF 标志为“1”的状态（过载错误）下结束串行传送，SPI 将不会把移位寄存器的数据复制到接收缓冲寄存器，也不产生接收缓冲器满中断，RDFF 标志保持为“0”。即使 PAE 位为“1”，也不检测奇偶校验错误。在发生过载错误的状态下，如果在没有将接收数据从移位寄存器复制到接收缓冲器的情况下结束串行传送，SPI 就判断为移位寄存器处于空的状态，允许将数据从发送缓冲寄存器器传送到移位寄存器。
- ⑥ 只有在将 OVRERF 标志变为“0”后才可以进行正常的接收运行。

监测过载错误能通过直接访问 SPI\_SR 寄存器或者使用 SPI 错误中断访问 SPI\_SR 寄存器等方式进行。在进行串行传送时，必须通过诸如读取完 SPI\_DR 寄存器后立即读取 SPI\_SR 寄存器等方法，尽早检测到过载错误的发生。

在主机模式下如果启用通信自动挂起功能（将 SPI\_CR.CSUSPE 位设成“1”），则 SPI 将在发生过载错误前的最后一个采样周期将通信时钟暂停，此时由于移位寄存器尚未完成最后一位的接收，SPI 保持在正常通信状态，过载错误不会发生。在通信时钟暂停期间可以对接收缓冲寄存器进行读取，读取后接收缓冲寄存器状态变为空，SPI 再重启通信时钟完成最后一位的数据接收。详细动作可参考下图 30-17 和图 30-18。

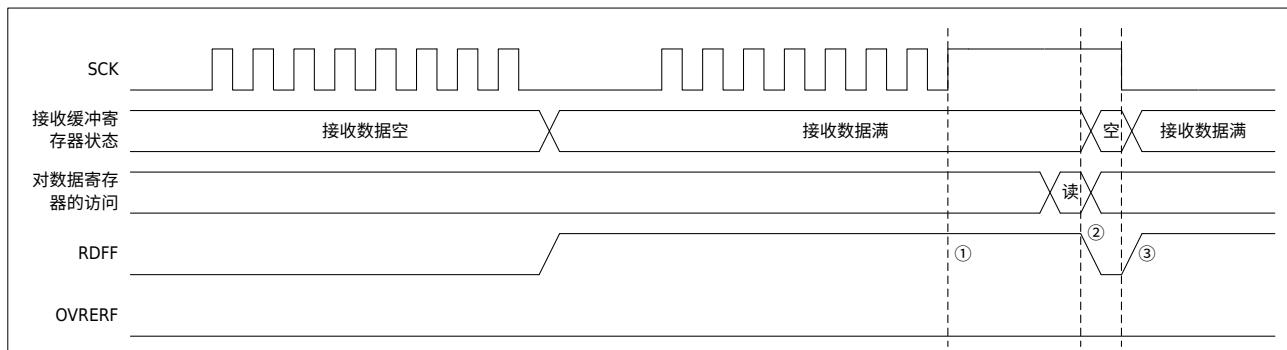


图 30-17 启用时钟自动停止功能时的动作示意图 (CPHA=1)

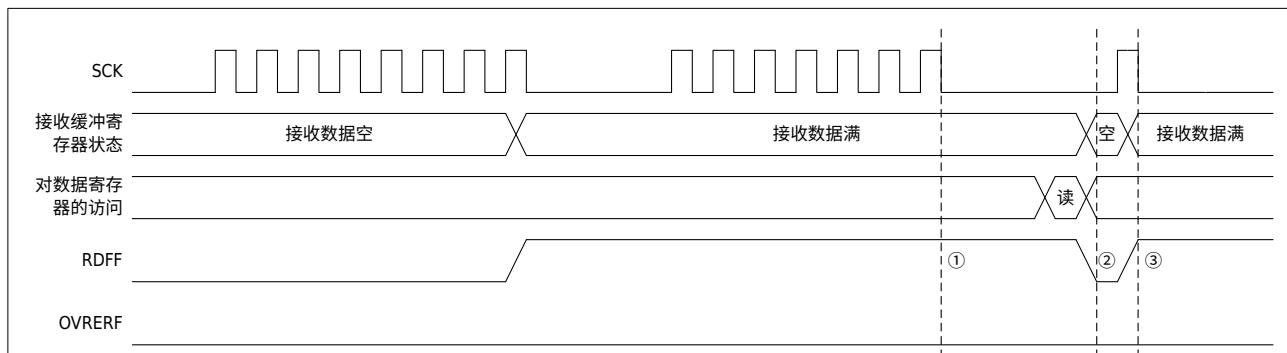


图 30-18 启用时钟自动停止功能时的动作示意图 (CPHA=0)

以下说明在图中的①~③所示的时序中标志的运行内容：

- ① 当处于接收缓冲寄存器满状态时，SPI 在最后一位数据接收前暂停通信时钟。此时将不会发生过载错误。
- ② 当通过访问 SPI\_DR 读取接收缓冲寄存器中的数据后，接收缓冲寄存器变为空状态，RDFF 标志为清零，SPI 重启通信时钟完成最后一位的数据通信。
- ③ 最后一位数据通信完成，接收缓冲寄存器再次变为满状态，RDFF 标志置“1”，可通过访问 SPI\_DR 读取接收到的数据。

### 30.8.4 奇偶校验错误

在 SPI\_CR.TXMDS 位为“0”并且 SPI\_CR.PAE 位为“1”的状态下，SPI 将在全双工同步串行通信结束时进行奇偶校验。当 SPI 检测到接收数据有奇偶校验错误时，将 SPI\_SR.PERF 标志置“1”。在 SPI\_SR.OVRERF 位为“1”的状态下，因为 SPI 不会将移位寄存器的数据复制到接收缓冲器，所以不对接收数据进行奇偶校验错误的检测。要将 PERF 标志清零时，需要在 PERF 标志为“1”的状态下读 SPI\_SR 寄存器后再给 PERF 标志写“0”。

OVRERF 标志和 PERF 标志的运行实例如下图 30-19 所示。在图中的例子中，SPI 在 SPI\_CR.TXMDS 位为“0”并且 SPI\_CR.PAE 位为“1”的状态下进行全双工同步串行通信的 8 位串行传送。

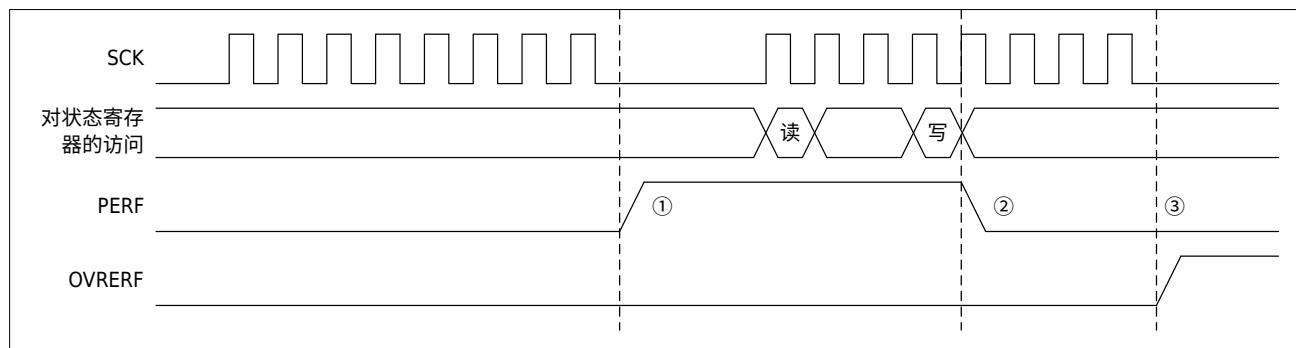


图 30-19 奇偶校验错误

以下说明在图中①~③所示的时序中标志的运行内容：

- ① SPI 未检测到过载错误，串行传送正常结束。SPI 将移位寄存器的数据复制到接收缓冲器。此时，SPI 对接收数据进行奇偶校验。如果检测到奇偶校验错误，则将 PERF 标志置 1。
- ② 清零 RERF 标志。
- ③ SPI 检测到过载错误，此时 SPI 不会将移位寄存器的数据复制到接收缓冲器，SPI 不会对数据进行奇偶校验，不会发生奇偶校验错误。

可通过直接访问 SPI\_SR 寄存器或者经 SPI 错误中断读取 SPI\_SR 寄存器等方式监测奇偶校验错误的发生。在进行串行传送时，必须通过访问状态寄存器 SPI\_SR 等方法，尽早监测到奇偶校验错误的发生。

## 30.9 SPI 的初始化

通过写操作或者模式故障错误检测将 SPE 位清零，可以使 SPI 功能无效并且对部分 SPI 功能进行初始化。如果发生系统复位，则对全部 SPI 功能进行初始化。

### 30.9.1 清除 SPE 位进行初始化

当 SPI\_CR.SPE 位为 0 时，SPI 进行以下的初始化操作：

- 中止正在进行的串行传送。
- 若处于从机状态则停止驱动输出信号（状态变为 Hi-Z）。
- 对 SPI 内部状态进行初始化。
- 清空发送缓冲寄存器，SPI\_SR.TDEF 标志置为 1。

通过将 SPE 位清零来进行初始化时，不会对 SPI 的控制位进行初始化。因此，只要重新将 SPE 位置为 1，就能以和初始化前相同的传送模式启动 SPI。

清除 SPE 位不会对错误标志位和序列状态进行初始化。因此，在即使在 SPE 被清零以后，也能通过读取接收缓冲器的数据来确认 SPI 传送时的错误发生状况。

由于清除 SPE 位会清空发送缓冲寄存器，并将 SPI\_SR.TDEF 标志置为 1。因此，如果在初始化后将 SPI\_CR.TXIE 位设成 1，就会产生 SPI 发送缓冲寄存器为空的中断。要避免系统发生此中断，必须在将 SPE 位清零的同时也将 TXIE 位设成 0。

### 30.9.2 系统复位初始化

通过系统复位进行初始化，将对 SPI 所有的控制位、状态位和数据寄存器进行初始化。

## 30.10 中断

SPI 的接收缓冲器满、发送缓冲器空、模式故障(过载、欠载、奇偶校验错误)和 SPI 空闲即可做为中断源也可做为内部触发源，传送完成只可以作为内部触发源。

过载、欠载和奇偶校验错误的中断被集成为 SPI 错误中断 SPEI，所以需要通过标志判断实际发生的中断源。SPI 中断/内部触发源的具体说明如表 30-8 所示。一旦条件成立，就产生相应的中断/内部触发请求。对于接收缓冲器满和发送缓冲器空的中断源，需要通过数据传送改变缓冲器状态来清除。

表 30-8 SPI 中断源说明

中断/内部触发源	略称	条件
接收缓冲器满	SPRI	在SPI_CR.RXIE位为“1”的状态下接收缓冲器变满时
发送缓冲器空	SPTI	在SPI_CR.TXIE位为“1”的状态下发送缓冲器变空时
SPI错误（过载、欠载、奇偶校验错误）	SPEI	在 SPI_CR.EIE 位 为 “1” 的 状 态 下 SPI_SR.OVRERF 、 SPI_SR.PERF 或 SPI_SR.MODFERF 及 SPI_SR.UDRERF 标志变为“1”时
SPI空闲	SPII	在SPI_CR.IDIE位为“1”的状态下 主机模式：数据传输完成或者SPI_CR.SPE由1写为0。 从机模式：SPI_CR.SPE由1写为0。
传送完成	SPEND	主机模式：数据传输完成。 从机模式：四线式，SSL管脚为无效电平时；三线式，SCK管脚时钟达到数据传输所需时钟数。

## 30.11 可供使用的事件触发源

SPI 产生的可供使用的事件触发源主要有以下几种：

- 数据发送缓冲寄存器空
- 数据接收缓冲寄存器满
- SPI 通信错误（包括过载，欠载，奇偶校验等错误）
- SPI 处于闲置状态
- SPI 通讯结束

用户可将上述事件触发源对应的向量写入不同的触发对象寄存器实现各种事件触发功能。

上述事件触发源对应的向量请参考【中断控制器（INTC）】。

## 30.12 寄存器说明

表 30-9 寄存器一览

寄存器基准地址： SPI1\_BASE: 0x4001C000; SPI2\_BASE: 0x4001C400

SPI3\_BASE: 0x4001C800; SPI4\_BASE: 0x40020000

寄存器名	偏移地址	复位值
SPI数据寄存器SPI_DR	0x00	0x0000 0000
SPI控制寄存器SPI_CR	0x04	0x0000 0000
SPI通信配置寄存器1 SPI_CFG1	0x0C	0x0000 0010
SPI状态寄存器SPI_SR	0x14	0x0000 0020
SPI通信配置寄存器2 SPI_CFG2	0x18	0x0000 0F1D

### 30.12.1 SPI 数据寄存器 (SPI\_DR)

复位值： 0x0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SPD[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SPD[15:0]															
位	标记	位名	功能	读写											
b31~b0	SPD[31:0]	串行数据	SPI数据存储	R/W											

### 30.12.2 SPI 控制寄存器 (SPI\_CR)

复位值：0x0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PAE	PAOE	PATE	MODFE	IDIE	RXIE	TXIE	EIE	CSUSPE	SPE	SPLPBK2	SPLPBK	MSTR	-	TXMDS	SPIMDS

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15	PAE	奇偶校验允许	0: 发送数据不附加奇偶校验位, 接收数据不进行奇偶校验 1: 发送数据附加奇偶校验位, 接收数据进行奇偶校验 (SPI_CR.TXMDS=0); 发送数据附加奇偶校验位, 接收数 据不进行奇偶校验 (SPI_CR.TXMDS=1)	R/W
b14	PAOE	奇偶校验模式选择	0: 选择偶校验进行发送和接收 1: 选择奇校验进行发送和接收	R/W
b13	PATE	奇偶校验自诊断	0: 奇偶校验自诊断功能无效 1: 奇偶校验自诊断功能有效	R/W
b12	MODFE	模式故障错误检测允许	0: 禁止模式故障错误检测 1: 允许模式故障错误检测	R/W
b11	IDIE	SPI空闲中断允许	0: 禁止空闲中断请求产生 1: 允许空闲中断请求产生	R/W
b10	RXIE	SPI接收中断允许	0: 禁止SPI接收中断请求产生 1: 允许SPI接收中断请求产生	R/W
b9	TXIE	SPI发送中断允许	0: 禁止SPI发送中断请求产生 1: 允许SPI发送中断请求产生	R/W
b8	EIE	SPI错误中断允许	0: 禁止SPI错误中断请求产生 1: 允许SPI错误中断请求产生	R/W
b7	CSUSPE	通讯自动挂起功能允许	0: 通讯自动挂起功能无效 1: 通讯自动挂起功能有效	R/W
b6	SPE	SPI功能允许	0: SPI功能无效 1: SPI功能有效	R/W
b5	SPLPBK2	SPI回环2位	0: 正常模式 1: 回环模式 (发送数据=接收数据)	R/W
b4	SPLPBK	SPI回环位	0: 正常模式 1: 回环模式 (发送数据的反相=接收数据)	R/W
b3	MSTR	SPI主从模式选择	0: 从机模式 1: 主机模式	R/W
b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	TXMDS	通信模式选择	0: 全双工同步串行通信 1: 只进行发送串行通信	R/W
b0	SPIMDS	SPI模式选择	0: SPI运行 (4线式) 1: 时钟同步运行 (3线式)	R/W

### 30.12.3 SPI 通信配置寄存器 1 (SPI\_CFG1)

复位值：0x0010

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	MIDI[2:0]	-		MSSDL[2:0]	-		MSSI[2:0]	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CLKDIV[3:0]	SS3PV	SS2PV	SS1PV	SS0PV	-	SPRDTD	-	-	-	-	CTMDS	FTHLV[1:0]			

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“0”,写入时写“0”	R/W
b30~b28	MIDI[2:0]	主机下次存取数据间隔 闲置时间设定位	b30~b28 0 0 0: 1 个SCK+2个PCLK1 0 0 1: 2 个SCK+2个PCLK1 0 1 0: 3 个SCK+2个PCLK1 0 1 1: 4 个SCK+2个PCLK1 1 0 0: 5 个SCK+2个PCLK1 1 0 1: 6 个SCK+2个PCLK1 1 1 0: 7 个SCK+2个PCLK1 1 1 1: 8 个SCK+2个PCLK1	R/W
b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26~b24	MSSDL[2:0]	主机SS无效延迟设定位	b26~b24 0 0 0: 1 个SCK 0 0 1: 2 个SCK 0 1 0: 3 个SCK 0 1 1: 4 个SCK 1 0 0: 5 个SCK 1 0 1: 6 个SCK 1 1 0: 7 个SCK 1 1 1: 8 个SCK	R/W
b23	Reserved	-	读出时为“0”,写入时写“0”	R/W
b22~b20	MSSI[2:0]	主机SS闲置时间设定位	b22~b20 0 0 0: 1 个SCK 0 0 1: 2 个SCK 0 1 0: 3 个SCK 0 1 1: 4 个SCK 1 0 0: 5 个SCK 1 0 1: 6 个SCK 1 1 0: 7 个SCK 1 1 1: 8 个SCK	R/W
b19~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b12	CLKDIV	通讯基准时钟选择位	b15~b12 0 0 0 0: pclk/2 0 0 0 1: pclk/4 0 0 1 0: pclk/6 0 0 1 1: pclk/8 0 1 0 0: pclk/10 0 1 0 1: pclk/12 0 1 1 0: pclk/14 0 1 1 1: pclk/16 1 0 0 0: pclk/18 1 0 0 1: pclk/20	R/W

			1 0 1 0: pcclk/22 1 0 1 1: pcclk/24 1 1 0 0: pcclk/26 1 1 0 1: pcclk/28 1 1 1 0: pcclk/30 1 1 1 1: pcclk/32	
b11	SS3PV	SS3信号极性设定	0: SS3信号的Low电平有效 1: SS3信号的High电平有效	R/W
b10	SS2PV	SS2信号极性设定	0: SS2信号的Low电平有效 1: SS2信号的High电平有效	R/W
b9	SS1PV	SS1信号极性设定	0: SS1信号的Low电平有效 1: SS1信号的High电平有效	R/W
b8	SS0PV	SS0信号极性设定	0: SS0信号的Low电平有效 1: SS0信号的High电平有效	R/W
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6	SPRD TD	数据寄存器读取对象选择	0: SPI_DR 读接收缓冲器 1: SPI_DR 读发送缓冲器(必须在TDEF=1时才可以读取)	R/W
b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4	Reserved	-	读出时为“1”,写入时写“1”	R/W
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	CTMDS	连续通信模式选择	0: 常规通信模式 1: 连续通信模式  b1~b0 0 0: 1 帧 0 1: 2 帧 1 0: 3 帧 1 1: 4 帧	R/W
b1~b0	FTHLV[1:0]	帧数设定位	0 1: 2 帧 1 0: 3 帧 1 1: 4 帧	R/W

### 30.12.4 SPI 状态寄存器 (SPI\_SR)

复位值：0x0020

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	RDFF	-	TDEF	UDRERF	PERF	MODFERF	IDLNF	OVRERF

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	RDFF	接收缓冲器满标志	0: 接收缓冲器数据未满 1: 接收缓冲器数据已满 硬件置位, 清零	R
b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5	TDEF	发送缓冲器空标志	0: 发送缓冲器有数据 1: 发生缓冲器无数据 硬件置位, 清零	R
b4	UDRERF	欠载错误标志	0: 未发生欠载错误 1: 欠载错误发生 (MODFERF=1) 当MODFERF=0, 此位会被初始化 硬件置位后, 读取1写0, 状态位清零。	R/W
b3	PERF	奇偶校验错误标志	0: 未发生奇偶校验错误 1: 发生奇偶校验错误 硬件置位后, 读取1写0, 状态位清零	R/W
b2	MODFERF	模式故障错误标志	0: 未发送模式故障错误 1: 发生模式故障错误 硬件置位后, 读取1写0, 状态位清零	R/W
b1	IDLNF	SPI空闲标志	0: SPI为空闲状态 1: SPI为传送状态 硬件置位, 清零	R
b0	OVRERF	过载错误标志	0: 未发生过载错误 1: 发生过载错误 硬件置位后, 读取1写0, 状态位清零	R/W

### 30.12.5 SPI 通信配置寄存器 2 (SPI\_CFG2)

复位值：0x0F1D

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MSSIE	MSSDLE	MIDIE	LSBF	DSIZE[3:0]				SSA[2:0]			-	MBR[1:0]	CPOL	CPHA	

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15	MSSIE	SCK延迟允许	0: SCK 延迟为1个SCK 1: SCK 延迟为SSI的设定值	R/W
b14	MSSDLE	SS无效延迟允许	0: SS 无效延迟为1个SCK 1: SS 无效延迟为SSDL的设定值	R/W
b13	MIDIE	SPI下次存取延迟允许	0: 下次存取延迟为1个SCK+2个PCLK1 1: 下次存取延迟为MIDI的设定值	R/W
b12	LSBF	SPI LSB first 位	0: MSB first 1: LSB first	R/W
b11~b8				
0 0 0 0: 4位				
0 0 0 1: 5位				
0 0 1 0: 6位				
0 0 1 1: 7位				
0 1 0 0: 8位				
0 1 0 1: 9位				
0 1 1 0: 10位				
0 1 1 1: 11位				
1 0 0 0: 12位				
1 0 0 1: 13位				
1 0 1 0: 14位				
1 0 1 1: 15位				
1 1 0 0: 16位				
1 1 0 1: 20位				
1 1 1 0: 24位				
1 1 1 1: 32位				
b7~b2				
0 0 0: SS0				
0 0 1: SS1				
0 1 0: SS2				
0 1 1: SS3				
1 x x: 禁止设定				
b4	Reserved	-	读出时为“1”,写入时写“1”	R/W
b3~b2				
0 0: 选择基准通讯时钟				
0 1: 选择基准通讯时钟的2分频				
1 0: 选择基准通讯时钟的4分频				
1 1: 选择基准通讯时钟的8分频				

b1	CPOL	SCK极性设定位	0: 空闲时的SCK为Low电平 1: 空闲时的SCK为High电平	R/W
b0	CPHA	SCK相位设定位	0: 在奇数边沿进行数据采样，在偶数边沿数据发生变化 1: 在奇数边沿数据发生变化，在偶数边沿进行数据采样	R/W

## 31 四线式串行外设接口（QSPI）

### 31.1 简介

四线式串行外设接口（QSPI）是一个存储器控制模块，主要用于和带 SPI 兼容接口的串行 ROM 进行通信。其对象主要包括有串行闪存，串行 EEPROM 以及串行 FeRAM。

表 31-1 QSPI 主要规格

参数	规格
通道数	1通道
SPI	<ul style="list-style-type: none"><li>支持扩展SPI，二线式SPI和四线式SPI等多种协议</li><li>支持SPI模式0和SPI模式3</li><li>地址线宽度可选择8位/16位/24位/32位</li></ul>
时序调整	可通过时序调整以支持各种串行闪存
闪存读取	<ul style="list-style-type: none"><li>支持多种读取方式<ul style="list-style-type: none"><li>a. 标准读/快速读</li><li>b. 二线式输出快速读取/二线式输入输出快速读取</li><li>c. 四线式输出快速读取/四线式输入输出快速读取</li></ul></li><li>自由设置指令</li><li>数量可调的虚拟周期</li><li>16字节的预读取功能</li><li>状态查询功能</li><li>SPI总线周期延长功能</li><li>XIP控制功能</li></ul>
直接通信功能	灵活而广泛的支持大量串行闪存软件控制指令，包括擦，写，ID读取及掉电控制等等。
中断源	QSPI总线错误中断
模块停止功能	可通过模块停止降低功耗

## QSPI结构

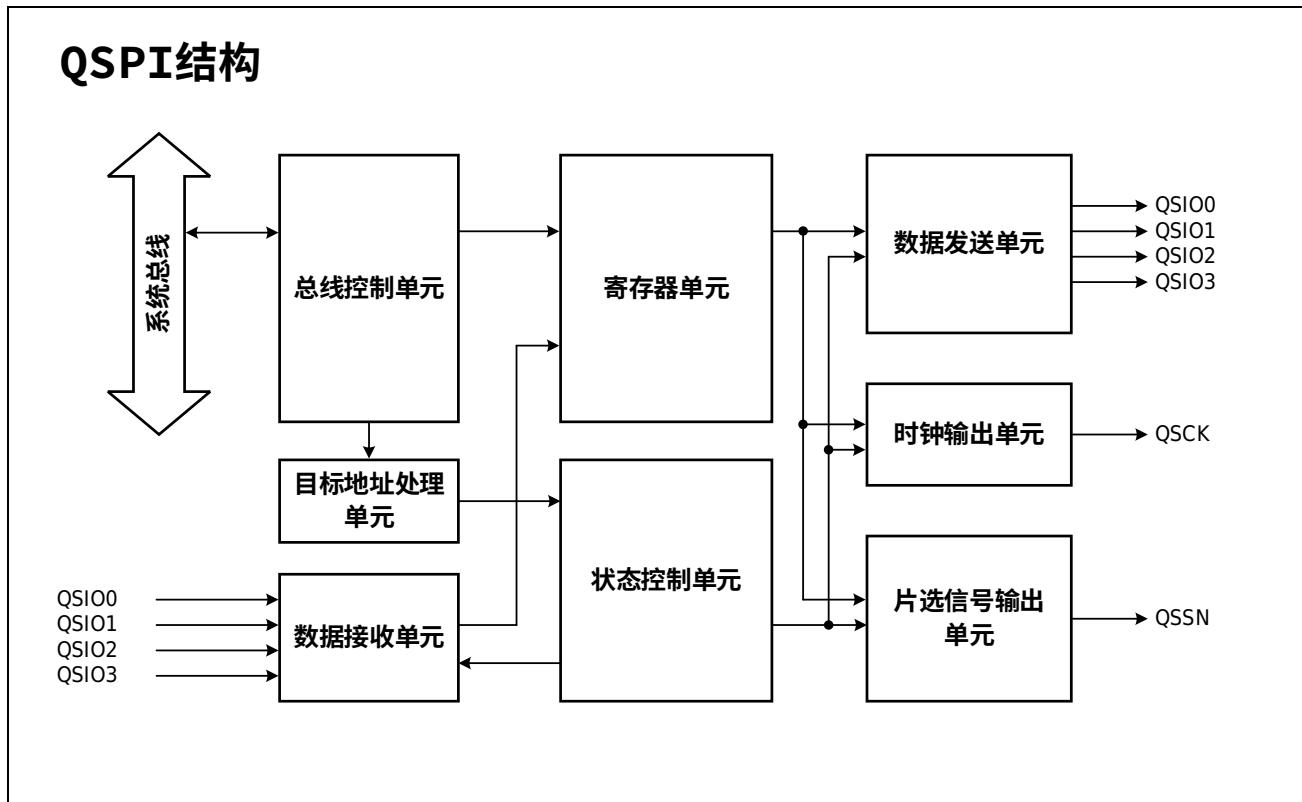


图 31-1 QSPI 的模块构成图

表 31-2 QSPI 管脚

管脚名	输入/输出	功能描述
QSCK	输出	QSPI时钟输出管脚
QSSN	输出	QSPI从机选择管脚
QSIO0	输入/输出	数据线0
QSIO1	输入/输出	数据线1
QSIO2	输入/输出	数据线2
QSIO3	输入/输出	数据线3

## 31.2 内存映射

### 31.2.1 内部总线空间

串行闪存及相关的控制寄存器在 AHB 总线空间的位置由总体的地址范围配置来决定。

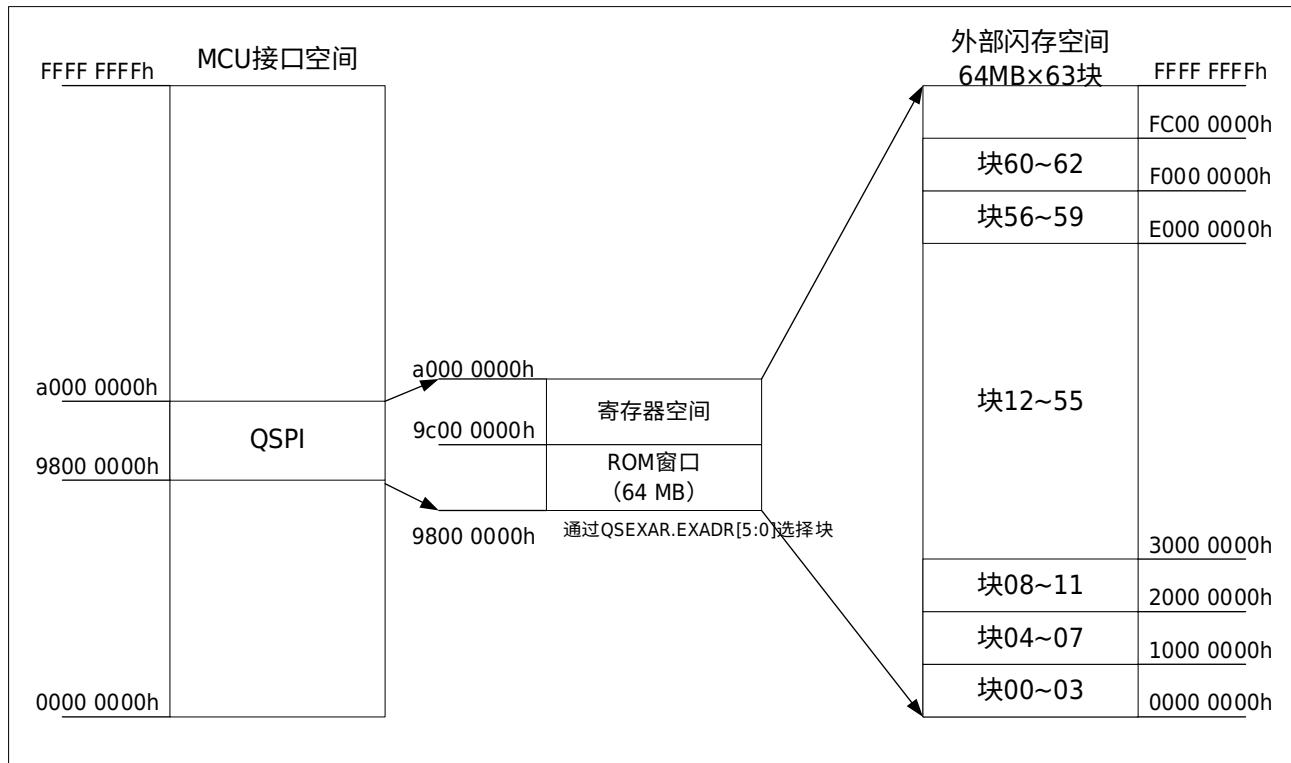


图 31-2 默认区域设定及 AHB 总线空间内存映射关系图

### 31.2.2 ROM 空间及总线的地址宽度

QSPI 的拥有 32 位地址总线宽度来配合串行闪存。每当对 QSPI 的 ROM 空间进行读访问时，QSPI 总线自动开始工作，将从串行闪存内读到的数据传送过来。

QSPI 不仅仅只能使用 32 位地址总线宽度，还可以通过设置 QSFCR 寄存器内的 AWSL[1:0]来选择使用 8 位/16 位/24 位地址总线宽度。

如果选择了 8 位/16 位/24 位的地址总线宽度，那么只有地址与之匹配的低位空间可以被正常访问，也就是说，访问 QSPI 中高位的串行闪存镜像空间将会反复出现低位空间的内容。

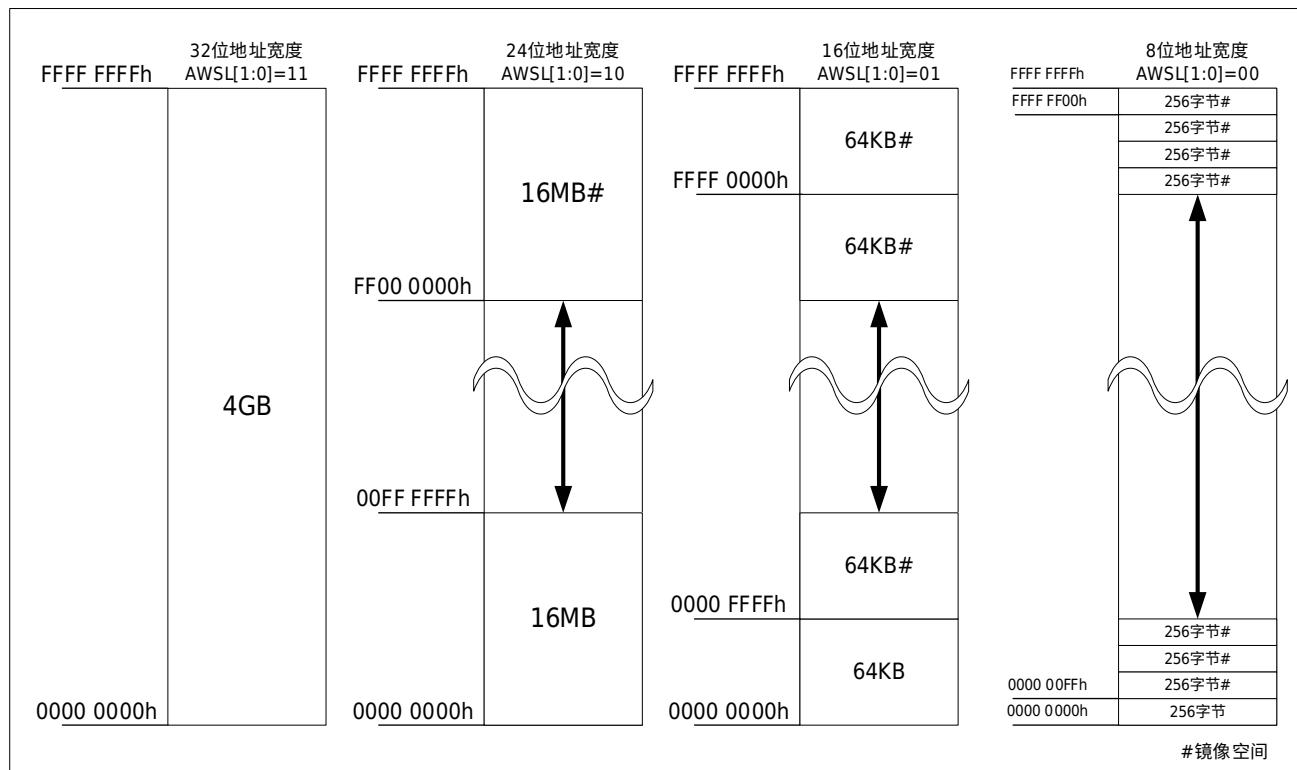


图 31-3 QSPI-ROM 空间内存映像图

#### 注意：

- 地址总线宽度可以通过设置 QSFCR 寄存器内的 AWSL[1:0]来选择使用 8 位/16 位/24 位/32 位。

## 31.3 QSPI 总线

### 31.3.1 SPI 协议

本 QSPI 支持扩展式 SPI，二线式 SPI 和四线式 SPI 三种协议。初始的默认协议是扩展式 SPI 协议。可通过设置 QSCR 寄存器中的 DPRSL[1:0]/APRSL[1:0]/IPRSL[1:0]位来分别配置各个阶段的协议。扩展式 SPI 协议只用 QSIO0 管脚单线进行指令输出，之后的地址及数据则根据具体的读取模式指令使用单线式/二线式/四线式输出。

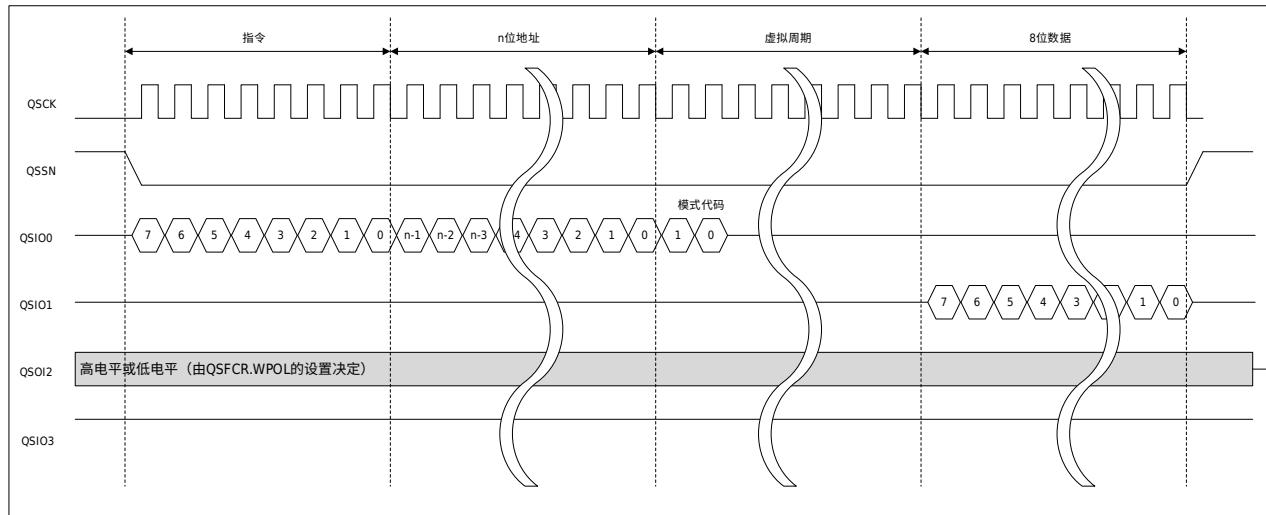


图 31-4 扩展式 SPI 协议动作示意图 1 (快速读模式)

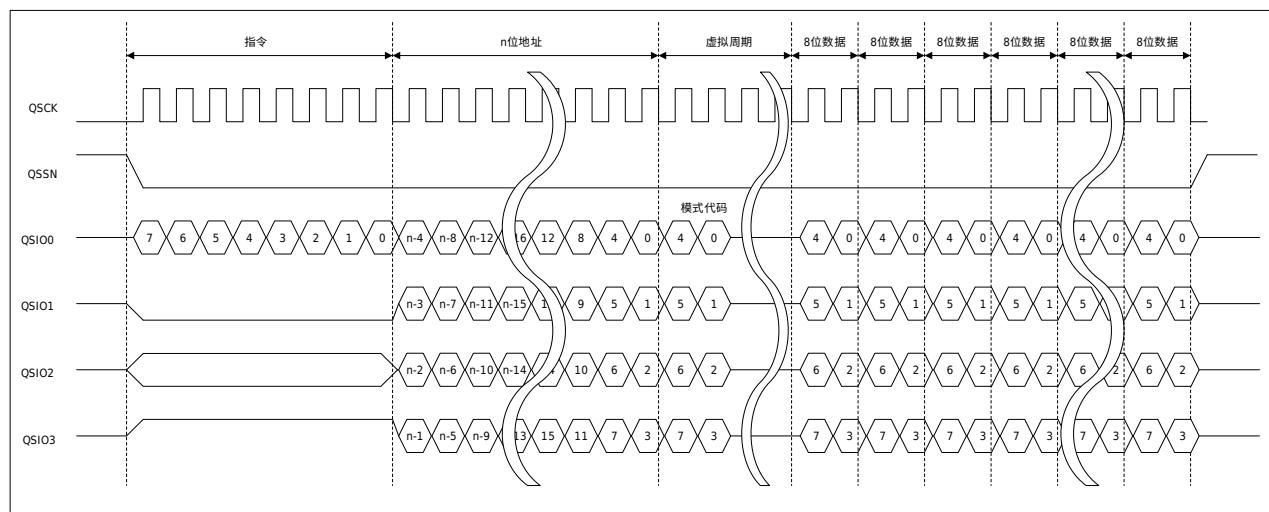


图 31-5 扩展式 SPI 协议动作示意图 2 (四线式输入输出快速读模式)

二线式 SPI 协议使用 QSIO0, QSIO1 两个管脚实行相应的操作，包括发出指令，地址，接收数据等等。

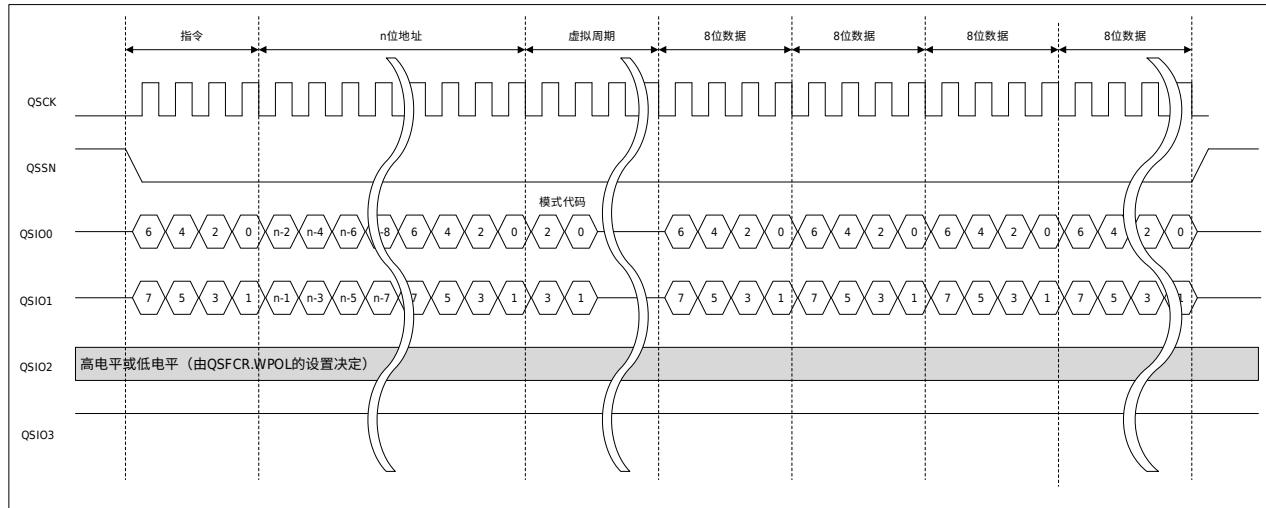


图 31-6 二线式 SPI 协议动作示意图（快速读模式）

四线式 SPI 协议使用 QSI00, QSI01, QSI02, QSI03 四个管脚实行发出指令, 地址, 接收数据等所有相关操作。

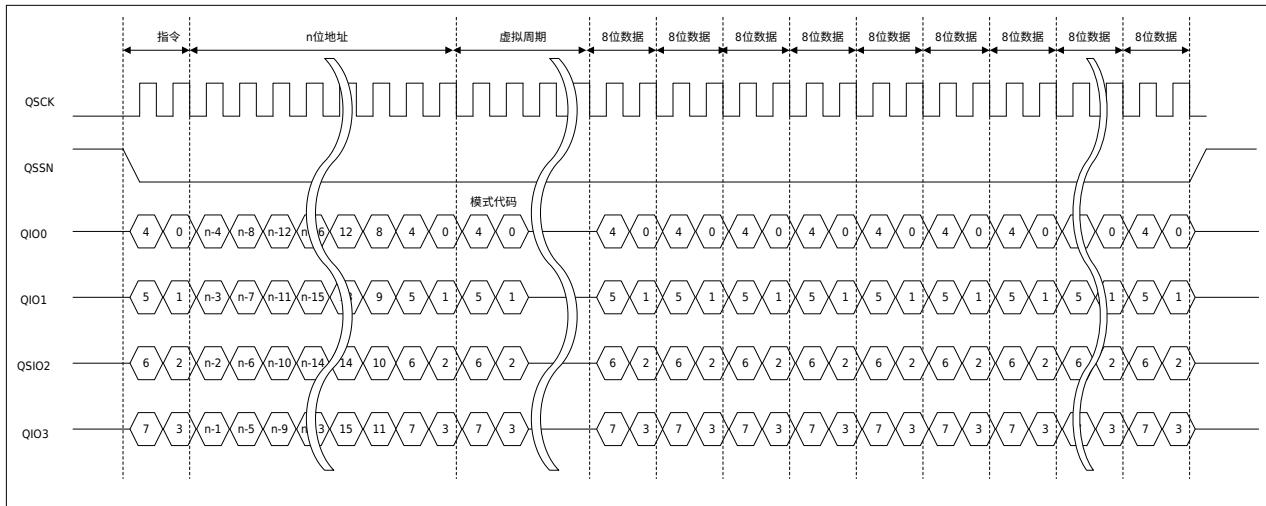


图 31-7 四线式 SPI 协议动作示意图（快速读模式）

### 31.3.2 SPI 模式

SPI 模式有模式 0 和模式 3 两种，可通过设置 QSCR 寄存器中的 SPIMD3 位来实现模式切换。SPI 模式 0 和模式 3 的区别在于待机状态时 QSCK 的电平不同。在 SPI 模式 0 时 QSCK 的待机电平为低电平，而模式 3 时待机电平则是高电平。

串行数据在串行时钟的下降沿从 QSPI 输出并在上升沿被读入外部闪存。而外部闪存则在串行时钟的下降沿输出串行数据并在下一个时钟的下降沿被 QSPI 读入。

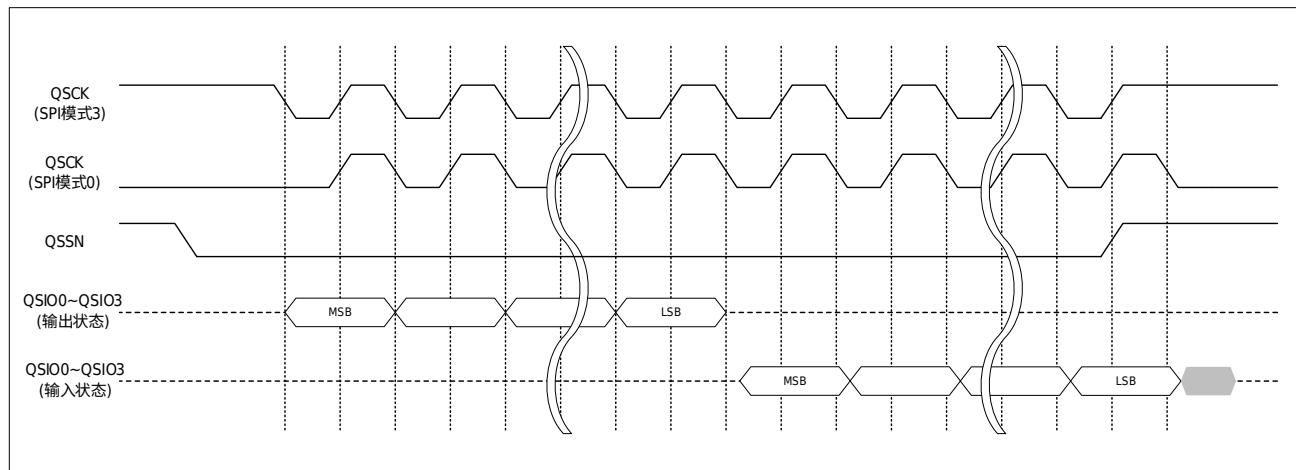


图 31-8 串行接口的基本时序图

## 31.4 QSPI 总线的时序调整

QSPI 总线信号可以通过寄存器进行时序微调，微调后的时序可更好的应用于各种 QSPI 总线访问，无论是 ROM 访问模式还是直接通信模式。

### 31.4.1 QSPI 总线基准时钟

QSPI 总线的基准时钟是由 HCLK 经过分频后得到的。通过设置 QSCR 寄存器的 DIV[5:0]位可以选择 HCLK 的 2 分频到 64 分频等多种时钟源作为 QSPI 总线的基准时钟。

表 31-3 QSPI 总线基准时钟选择一览表

DIV[5:0]	分频比	实际动作频率	DIV[5:0]	分频比	实际动作频率
		(HCLK=200MHz)			(HCLK=200MHz)
6'b000000	2	100.00	6'b100000	33	6.06
6'b000001	2	100.00	6'b100001	34	5.88
6'b000010	3	66.67	6'b100010	35	5.71
6'b000011	4	50.00	6'b100011	36	5.56
6'b000100	5	40.00	6'b100100	37	5.41
6'b000101	6	33.33	6'b100101	38	5.26
6'b000110	7	28.57	6'b100110	39	5.13
6'b000111	8	25.00	6'b100111	40	5.00
6'b001000	9	22.22	6'b101000	41	4.88
6'b001001	10	20.00	6'b101001	42	4.76
6'b001010	11	18.18	6'b101010	43	4.65
6'b001011	12	16.67	6'b101011	44	4.55
6'b001100	13	15.38	6'b101100	45	4.44
6'b001101	14	14.29	6'b101101	46	4.35
6'b001110	15	13.33	6'b101110	47	4.26
6'b001111	16	12.50	6'b101111	48	4.17
6'b010000	17	11.76	6'b110000	49	4.08
6'b010001	18	11.11	6'b110001	50	4.00
6'b010010	19	10.53	6'b110010	51	3.92
6'b010011	20	10.00	6'b110011	52	3.85
6'b010100	21	9.52	6'b110100	53	3.77
6'b010101	22	9.09	6'b110101	54	3.70
6'b010110	23	8.70	6'b110110	55	3.64
6'b010111	24	8.33	6'b110111	56	3.57
6'b011000	25	8.00	6'b111000	57	3.51
6'b011001	26	7.69	6'b111001	58	3.45
6'b011010	27	7.41	6'b111010	59	3.39

DIV[5:0]	分频比	实际动作频率	DIV[5:0]	分频比	实际动作频率
		(HCLK=200MHz)			(HCLK=200MHz)
6'b011011	28	7.14	6'b111011	60	3.33
6'b011100	29	6.90	6'b111100	61	3.28
6'b011101	30	6.67	6'b111101	62	3.23
6'b011110	31	6.45	6'b111110	63	3.17
6'b011111	32	6.25	6'b111111	64	3.13

### 31.4.2 SPI 总线基准时钟

当基准时钟选择了 HCLK 的偶数倍分频时，QSCK 信号的高低电平时间是一致的，如果选择了奇数倍分频，那么 QSCK 信号的高电平时间将比低电平多一个 HCLK 周期。

如果希望选择奇数倍分频时的 QSCK 也能输出 50%左右占空比的时钟信号，可以将 QSFCR 寄存器中 DUTY 位设置成 1。通过这个设置，QSCK 信号的上升沿的输出时间将比调整前晚半个 HCLK 周期，下降沿输出时间保持不变。由此可以得到占空比为 50%的 QSCK 信号。当基准时钟选择 HCLK 的偶数倍分频时，DUTY 位请设置为 0。初始状态下 DUTY 位默认为 1。

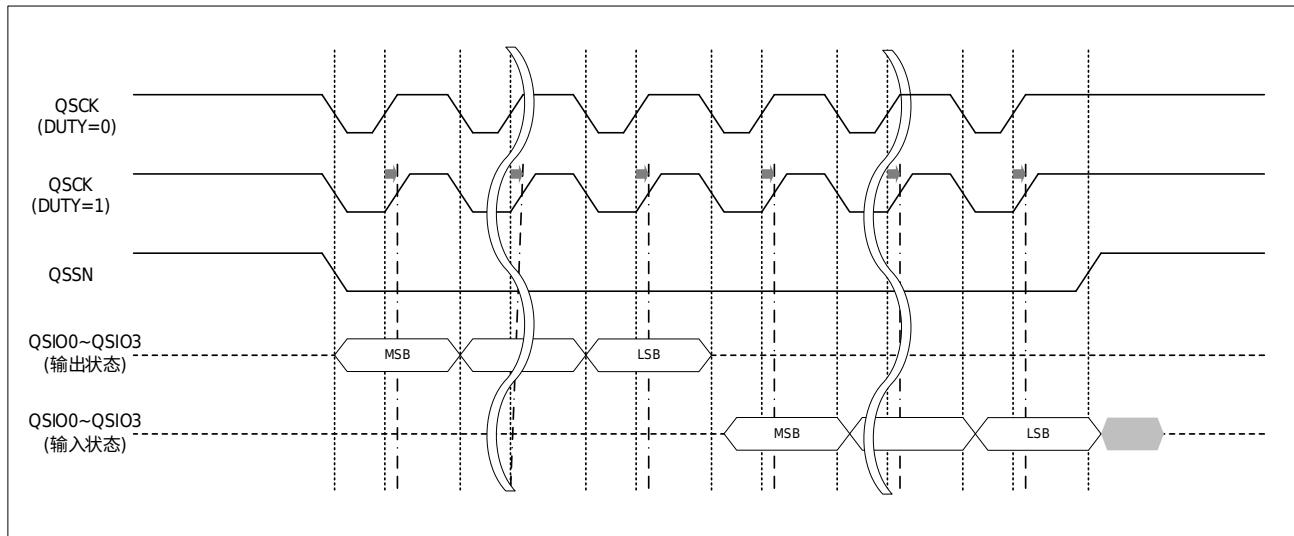


图 31-9 基准时钟选择 HCLK 三分频时输出时钟占空比补正示意图

### 31.4.3 QSSN 信号最小高电平宽度

为满足串行闪存所要求的取消时间要求，相邻的两次 QSPI 总线周期之间 QSSN 信号必须保持足够长时间的高电平状态（即闲置状态）。通过设置 QSCSCR 寄存器中的 SSHW[3:0]位可以选择 QSSN 的最小高电平宽度，选择范围为 1 个到 16 个 QSPI 基准时钟周期。

### 31.4.4 QSSN 的建立时间

从 QSSN 信号开始输出低电平（即变为有效状态）到 QSCK 信号输出第一个上升沿之间的时间称为 QSSN 的建立时间，该时间可以通过寄存器设置进行配置以满足外接串行闪存的要求。设置寄存器 QSFCR 的 SSNLD 位来选择 QSSN 的建立时间是 0.5 个或是 1.5 个 QSPI 基准时钟周期。该设置同样可用于配置数据管脚从数据输出许可到 QSCK 信号输出第一个上升沿之间的建立时间，可根据需要合理运用。

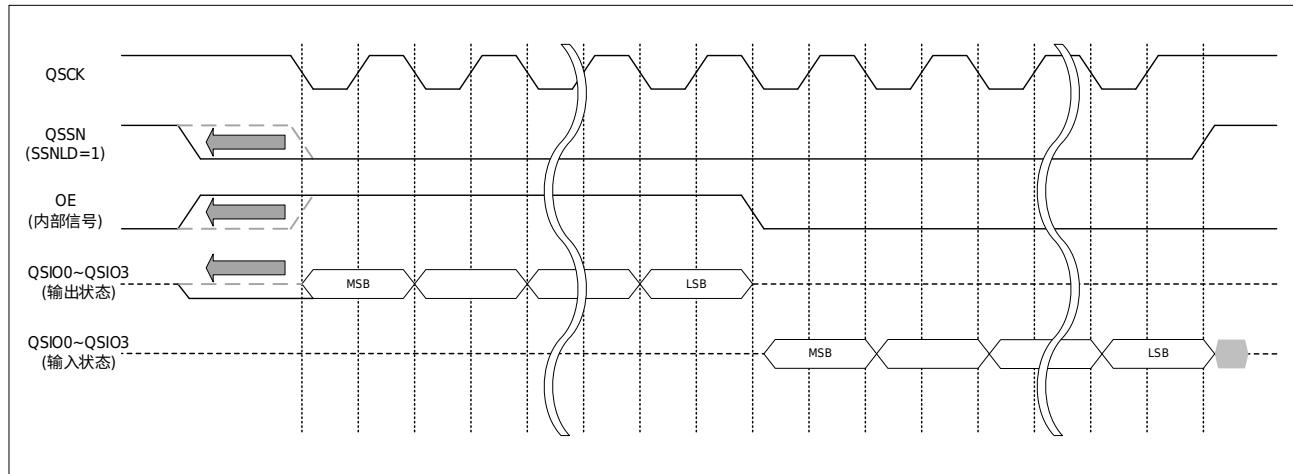


图 31-10 QSSL 建立时间配置示意图

### 31.4.5 QSSN 的保持时间

从 QSCK 信号输出最后一个上升沿到 QSSN 信号开始输出高电平（即变为闲置状态）之间的时间称为 QSSN 的保持时间，该时间可以通过寄存器设置进行配置以满足外接设备的要求。设置寄存器 QSFCR 的 SSNHD 位来选择 QSSN 的保持时间是 0.5 个或是 1.5 个 QSPI 基准时钟周期。

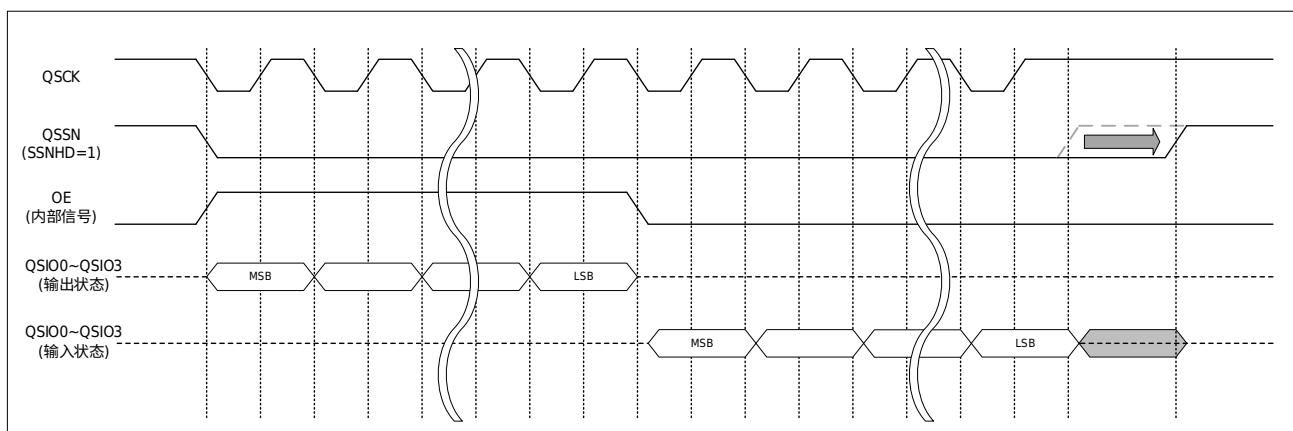


图 31-11 QSSN 保持时间配置示意图

### 31.4.6 串行数据接收延迟

串行闪存的数据和 QSCK 的下降沿同步输出，QSPI 在下一个 QSCK 的下降沿接收该数据。从串行闪存开始输出数据到该数据被 QSPI 接收之间的这段时间称之为接收延迟。QSPI 在第一个数据接收周期之前加入了一个延迟调整周期。站在串行闪存角度，该周期可视为对手方数据接收周期的增加。该迟延调整周期只会在数据接收动作时产生。

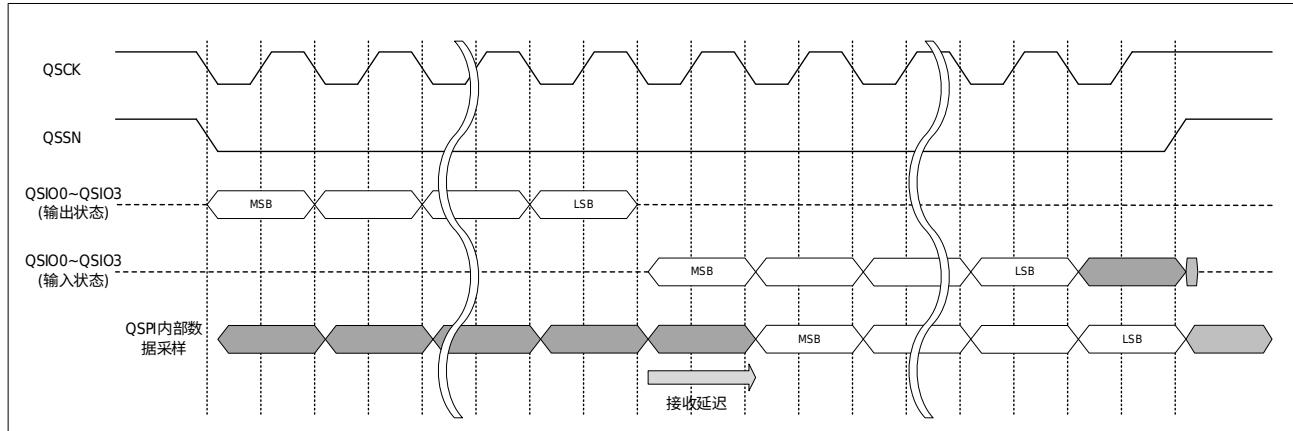


图 31-12 数据接收延迟示意图

## 31.5 用于 ROM 访问的 SPI 指令介绍

### 31.5.1 现有的 QSPI-ROM 指令参考

表 31-4 参考指令一览表

模式名	指令代码	说明
4-byte指令模式	标准读	8'h13
	快速读	8'h0c
	二线式输出快速读	8'h3c
	二线式输入输出快速读	8'hbc
	四线式输出快速读	8'h6c
	四线式输入输出快速读	8'hec
	退出4-byte指令模式	8'hb7
3-byte指令模式	标准读	8'h03
	标准读	8'h0b
	快速读	8'h0b
	二线式输出快速读	8'h3b
	二线式输入输出快速读	8'hbb
	四线式输出快速读	8'h6b
	四线式输入输出快速读	8'heb
	进入4-byte指令模式	8'he9
—	写模式	8'h06

对串行闪存进行访问时，指令需通过指令寄存器 QSCCMD 来进行设置。

### 31.5.2 标准读指令

标准读指令是绝大多数串行闪存都支持的一种常用读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，随即 QSPI 就输出该指令的指令代码 (03h/13h) \*1，紧接着输出目标地址，该地址的宽度可以通过 QSFCR 寄存器中的 AWSL[1:0]位来设置。然后就可以接收到数据。QSPI 的初始状态选择的指令就是标准读指令。

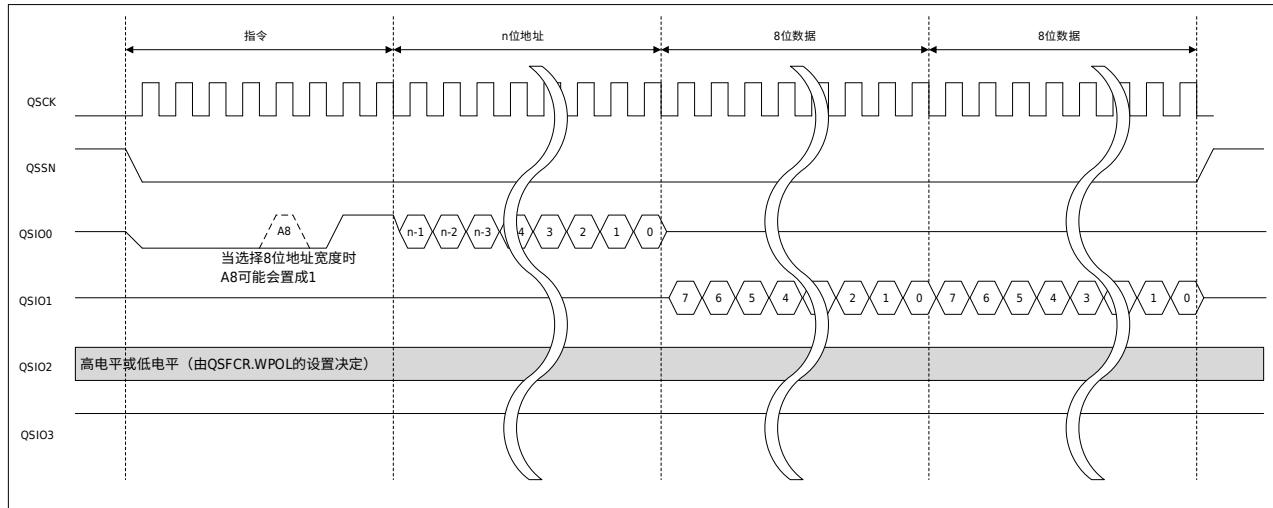


图 31-13 标准读总线周期示意图

### 31.5.3 快速读指令

快速读指令是一种支持更快的通信时钟的读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，随即 QSPI 就输出该指令的指令代码（0Bh/0Ch），紧接着输出目标地址，地址宽度可以通过 QSFCR 寄存器中的 AWSL[1:0]位来设置。地址输出后是一定数量的虚拟周期，其具体数量由 QSFCR 寄存器中的 DMCYCN[3:0]决定。紧接着就是数据的接收。

虚拟周期最初的两个周期用于决定是否选择 XIP 模式。当选择 XIP 模式时，本次传输所使用的指令将被应用于下一个 SPI 总线周期，在下一个 SPI 总线周期时指令传输部分将会被省略。详细可参考【XIP 控制】。

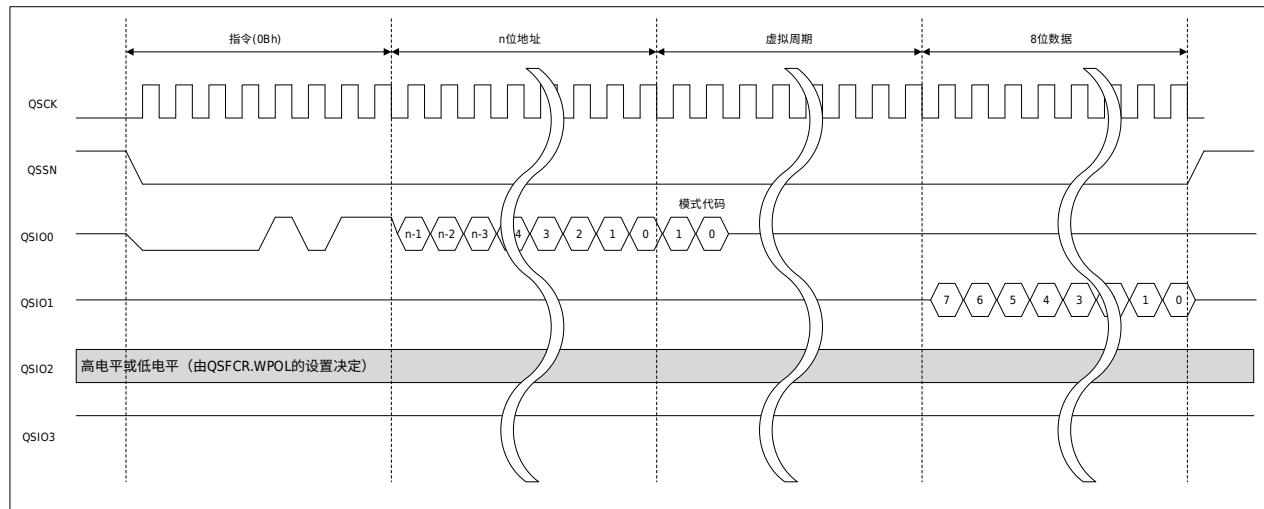


图 31-14 快速读总线周期示意图

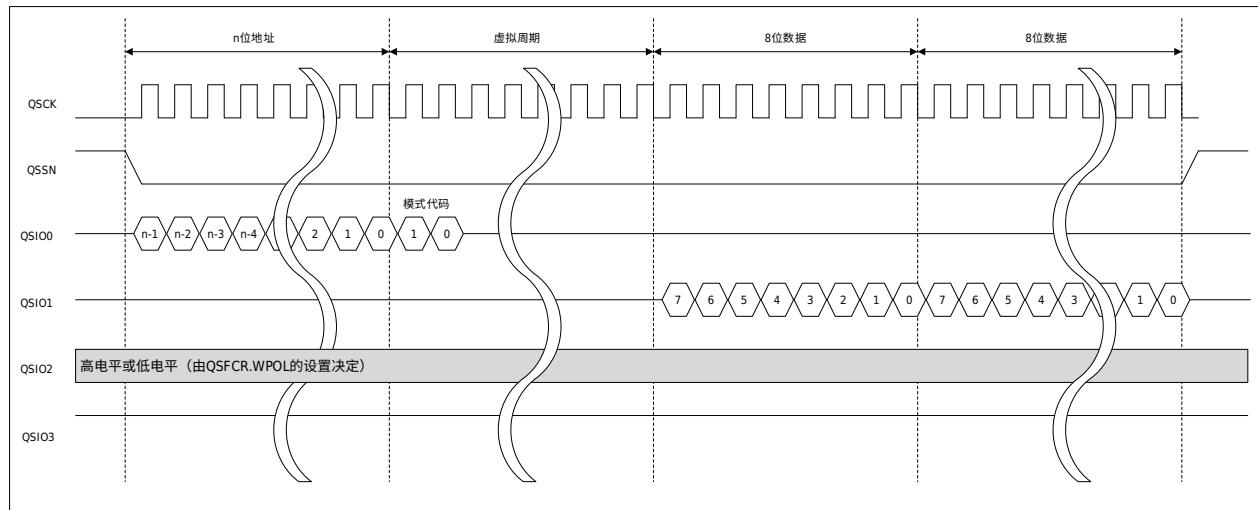


图 31-15 选择 XIP 模式的快速读总线周期示意图

**注意：**

- 要使用快速读指令请确保使用支持快速读功能的串行闪存。

**31.5.4 二线式输出快速读指令**

二线式输出快速读是一种使用两根信号线进行数据接收的读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSIO0 管脚输出该指令的指令代码（3Bh/3Ch）和目标地址，地址宽度可以通过 QSFCR 寄存器中的 AWSL[1:0]位来设置。在这之后是一定数量的虚拟周期，其具体数量由 QSFCR 寄存器中的 DMCYCN[3:0]决定。然后开始通过 QSIO0 和 QSIO1 两根管脚进行数据接收。偶位数据在 QSIO0 接收，奇位在 QSIO1。

虚拟周期最初的两个周期用于决定是否选择 XIP 模式。当选择 XIP 模式时，本次传输所使用的指令将被应用于下一个 QSPI 总线周期，在下一个 QSPI 总线周期时指令传输部分将会被省略。详细可参考【XIP 控制】。

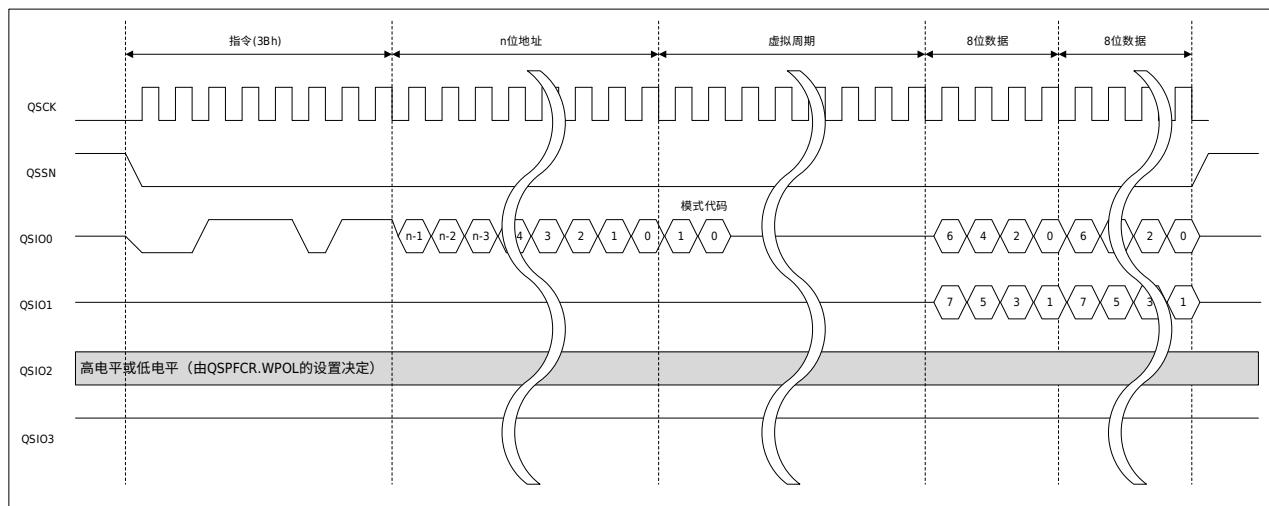


图 31-16 二线式输出快速读总线周期示意图

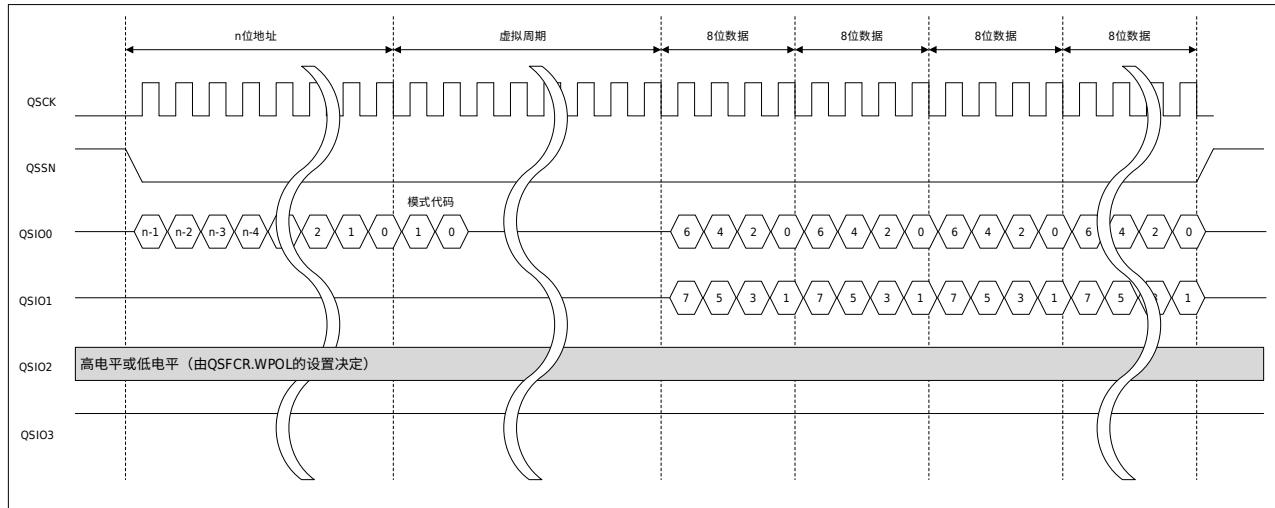


图 31-17 选择 XIP 模式的二线式输出快速读总线周期示意图

**注意：**

- 要使用二线式输出快速读指令请确保使用支持该功能的串行闪存。

### 31.5.5 二线式输入输出快速读指令

二线式输入输出快速读是一种使用两根信号线进行地址发送和数据接收的读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSIO0 管脚输出该指令的指令代码（BBh/BCh）。在这之后 QSPI 从 QSIO0 和 QSIO1 两个管脚输出目标地址，地址宽度可以通过 QSFCR 寄存器中的 AWSL[1:0]位来设置。在这之后是一定数量的虚拟周期，其具体数量由 QSFCR 寄存器中的 DMCYCN[3:0]决定。然后开始通过 QSIO0 和 QSIO1 两根管脚进行数据接收。偶数位的数据使用 QSIO0 管脚，奇数位使用 QSIO1 管脚。

虚拟周期最初的两个周期用于决定是否选择 XIP 模式。当选择 XIP 模式时，本次传输所使用的指令将被应用于下一个 QSPI 总线周期，在下一个 QSPI 总线周期时指令传输部分将会被省略。详细可参考【XIP 控制】。

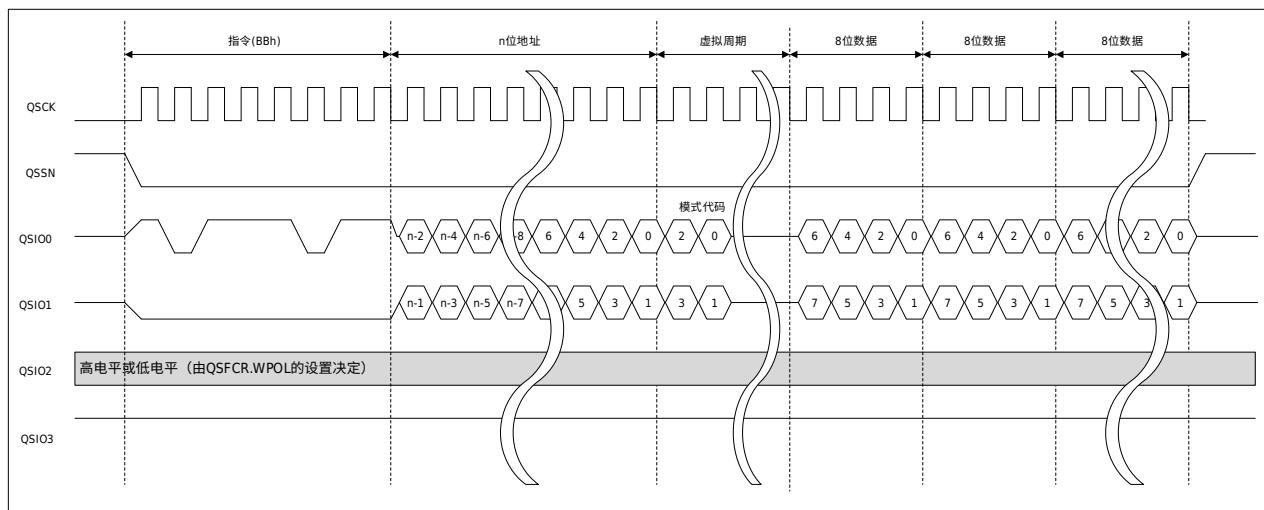


图 31-18 二线式输入输出快速读总线周期示意图

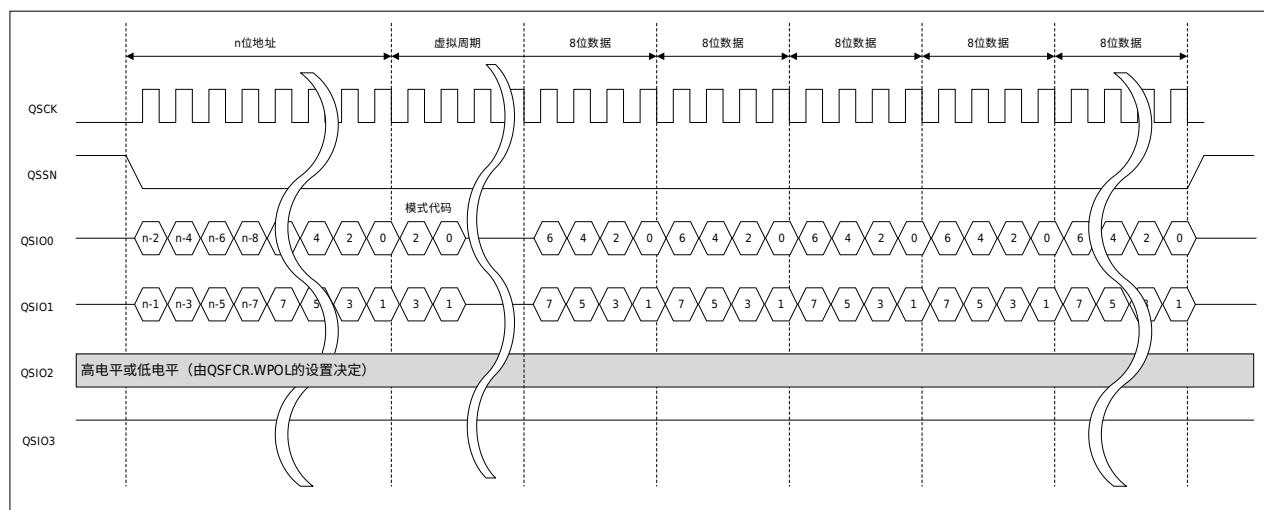


图 31-19 选择 XIP 模式的二线式输入输出快速读总线周期示意图

#### 注意：

- 要使用二线式输入输出快速读指令请确保使用支持该功能的串行闪存。

### 31.5.6 四线式输出快速读指令

四线式输出快速读是一种使用四根信号线进行数据接收的读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSI00 管脚输出该指令的指令代码（6Bh/6Ch）和目标地址，地址宽度可以通过 QSFCR 寄存器中的 AWSL[1:0]位来设置。在这之后是一定数量的虚拟周期，其具体数量由 QSFCR 寄存器中的 DMCYCN[3:0]决定。然后开始通过 QSI00, QSI01, QSI02 和 QSI03 四根管脚进行数据接收。

虚拟周期最初的两个周期用于决定是否选择 XIP 模式。当选择 XIP 模式时，本次传输所使用的指令将被应用于下一个 SPI 总线周期，在下一个 SPI 总线周期时指令传输部分将会被省略。详细可参考【XIP 控制】。

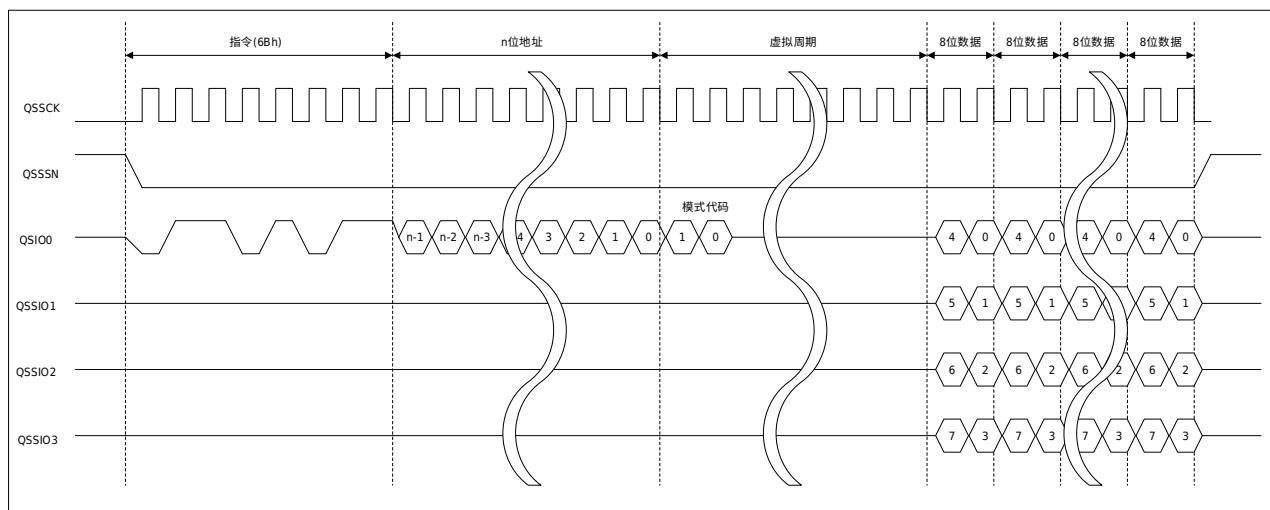


图 31-20 四线式输出快速读总线周期示意图

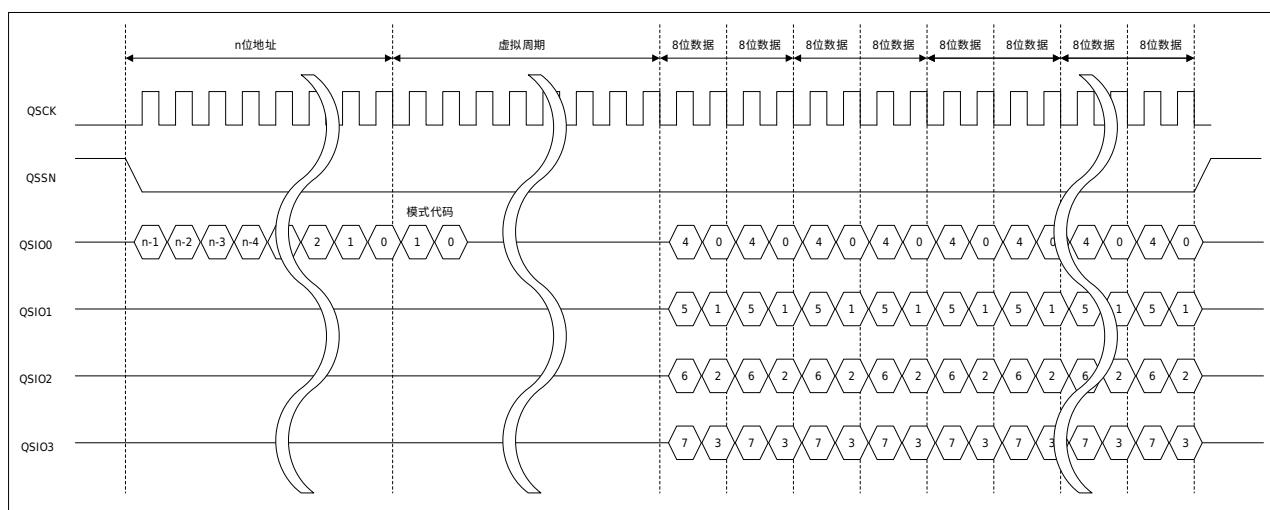


图 31-21 选择 XIP 模式的四线式输出快速读总线周期示意图

**注意：**

- 要使用四线式输出快速读指令请确保使用支持该功能的串行闪存。

### 31.5.7 四线式输入输出快速读指令

四线式输入输出快速读是一种使用四根信号线进行地址发送和数据接收的读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSIO0 管脚输出该指令的指令代码（EBh/ECh）。在这之后 QSPI 从 QSIO0, QSIO1, QSIO2 和 QSIO3 四个管脚输出目标地址，地址宽度可以通过 QSFCR 寄存器中的 AWSL[1:0]位来设置。在这之后是一定数量的虚拟周期，其具体数量由 QSFCR 寄存器中的 DMCYCN[3:0]决定。然后开始通过 QSIO0, QSIO1, QSIO2 和 QSIO3 四根管脚进行数据接收。

虚拟周期最初的两个周期用于决定是否选择 XIP 模式。当选择 XIP 模式时，本次传输所使用的指令将被应用于下一个 QSPI 总线周期，在下一个 QSPI 总线周期时指令传输部分将会被省略。详细可参考【XIP 控制】。

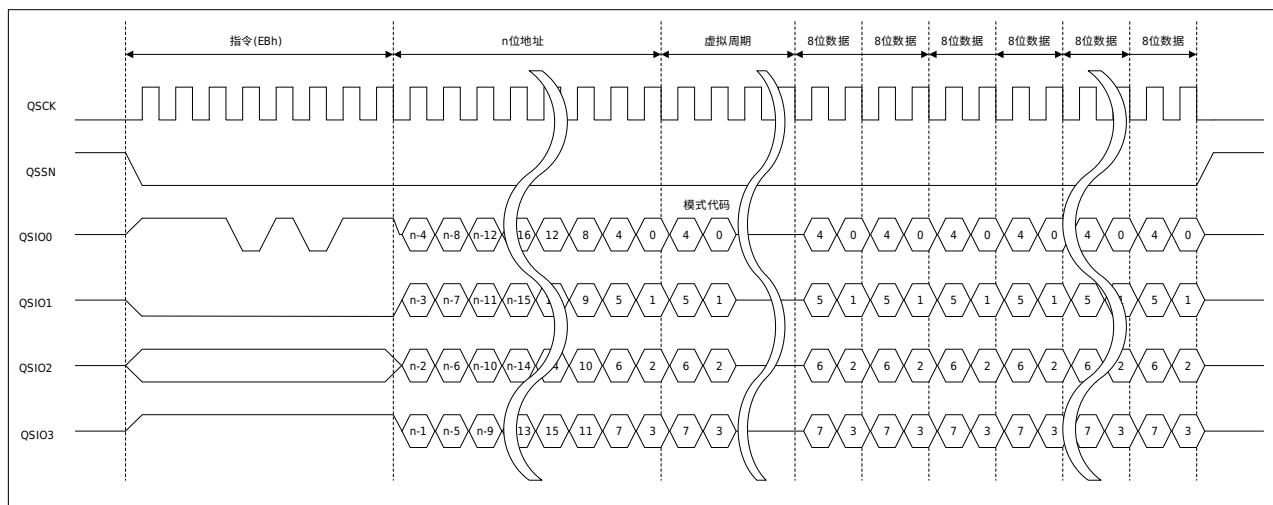


图 31-22 四线式输入输出快速读总线周期示意图

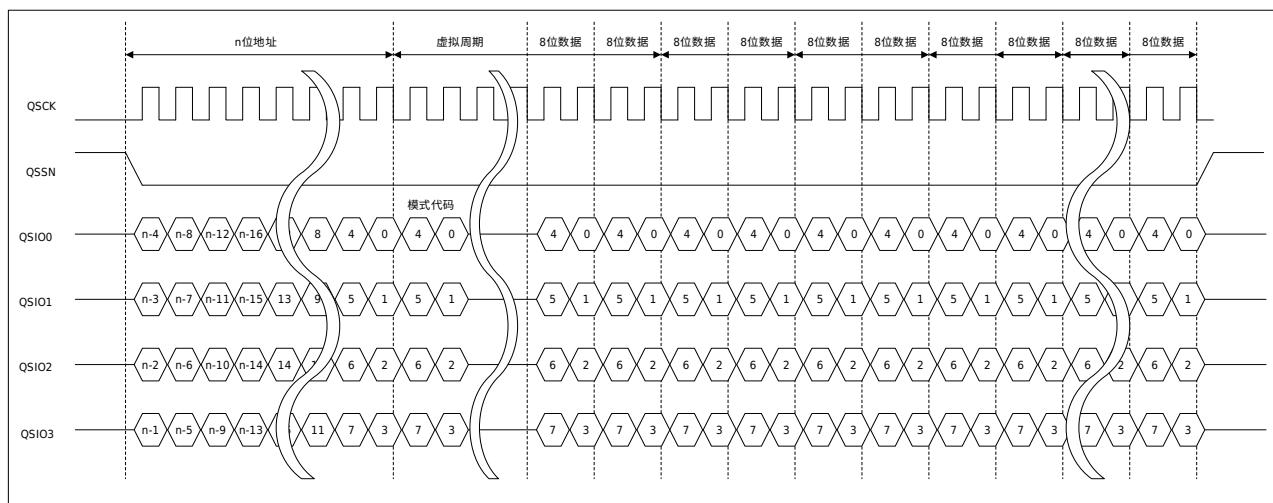


图 31-23 选择 XIP 模式的四线式输入输出快速读总线周期示意图

#### 注意：

- 要使用四线式输入输出快速读指令请确保使用支持该功能的串行闪存。

### 31.5.8 进入 4-Byte 模式指令

进入 4-Byte 模式指令可以将串行闪存的地址宽度设置成 4 个字节。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSIO0 管脚输出该指令的指令代码（B7h）。

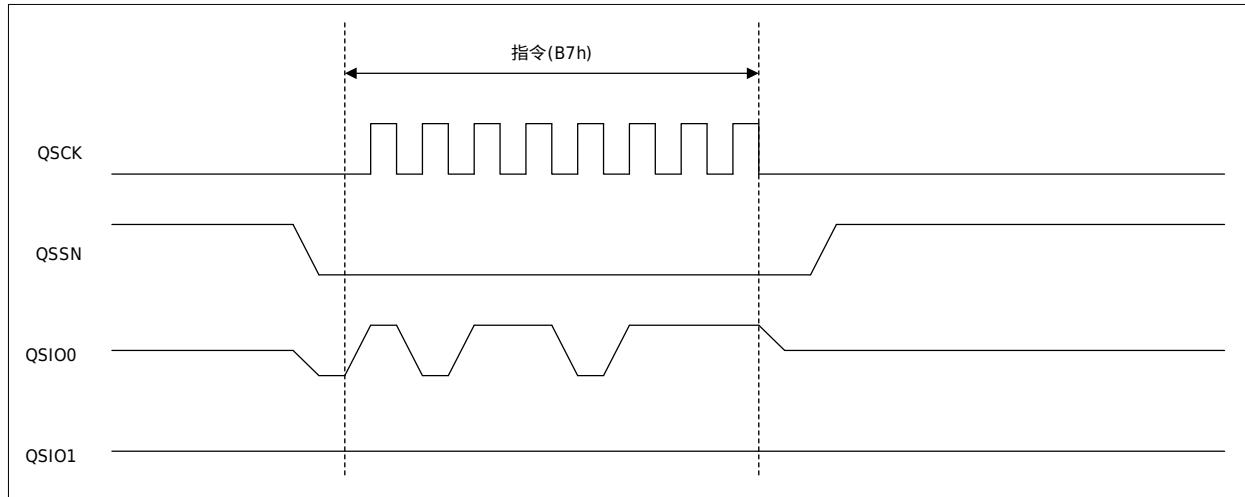


图 31-24 进入 4-Byte 模式指令总线周期示意图

**注意：**

- 无论串行闪存是否处于 4-Byte 或是 3-Byte 模式，该指令都可发布。

### 31.5.9 退出 4-Byte 模式指令

退出 4-Byte 模式指令可以将串行闪存的地址宽度设置成 3 个字节。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSIO0 管脚输出该指令的指令代码（E9h）。

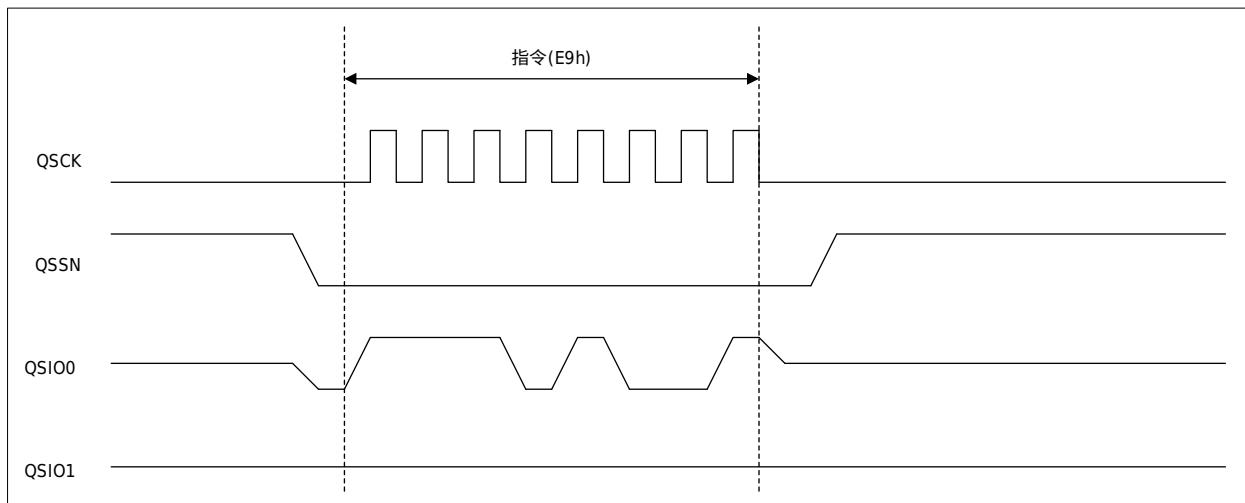


图 31-25 退出 4-Byte 模式指令总线周期示意图

**注意：**

- 无论串行闪存是否处于 4-Byte 或是 3-Byte 模式，该指令都可发布。

### 31.5.10 写许可指令

写许可指令允许改变串行闪存的地址宽度。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSI00 管脚输出该指令的指令代码（06h）。

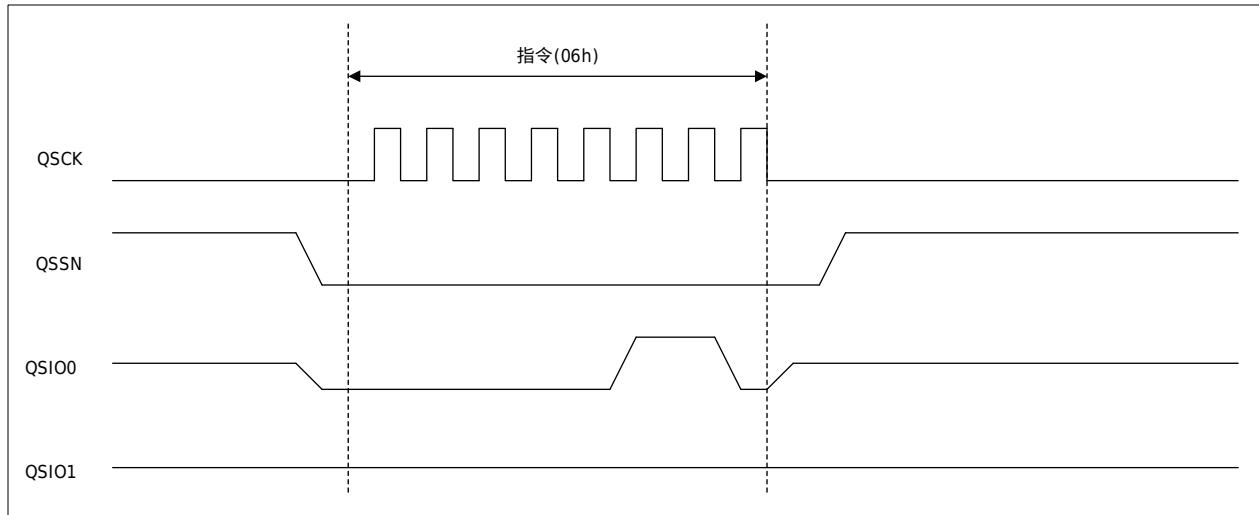


图 31-26 写许可指令总线周期示意图

## 31.6 QSPI 总线周期的安排

### 31.6.1 独立转换的单一闪存读取

对于 ROM 的单一读取指令会独立的从芯片内部总线周期一对一的转换为 QSPI 总线周期。当一个 ROM 的读取总线周期被检测到时，QSSN 信号会置为有效状态，从而启动一个 QSPI 总线周期。当接收完串行闪存的数据后，QSSN 信号变成无效状态，该 QSPI 总线周期宣告完成。

当另一个 ROM 的读取总线周期被检测到时，QSSN 信号会在确保无效保持时间已经超过最小无效保持宽度后再度置为有效状态，新一个 QSPI 总线周期开始。

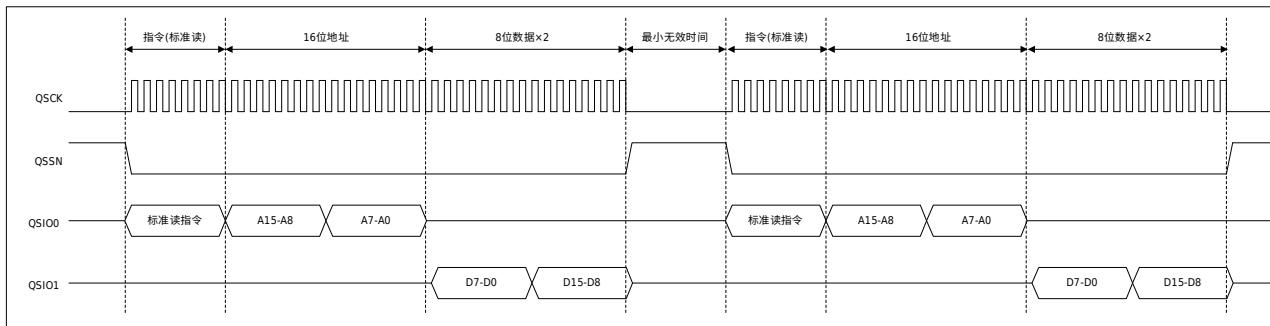


图 31-27 独立转换的单一闪存数据读取操作示意图

### 31.6.2 使用预读取功能进行闪存读取

对于诸如 CPU 指令或是数据块的传输，系统通常是以一个顺次递增的闪存地址顺序进行数据的读取。串行闪存具有连续数据传输能力而不需要再次发送指令代码和地址。但如果 MCU 发布的内部总线周期

是独立转换的话，QSPI 总线周期也被分成了各自独立的个体，导致无法有效利用串行闪存的这种连续数据传输的优点。对此，QSPI 提供了预读取功能来进行连续的数据接收。

通过设置 QSCR 寄存器中的 PFE 位为 1 激活预读取功能。当该功能有效后，数据会被连续接收并储存到缓冲区而不需要等待另一个闪存读取要求。当 MCU 发出一个闪存读取操作时，QSPI 将会对访问地址进行匹配。如果匹配成功，则将对应位置的缓冲区的数据传给 MCU，若匹配失败，则缓冲区的数据将会被丢弃并重新发出一个新的 QSPI 总线周期。

预读取的缓冲区最大可存储 16 个字节的数据，除此以外，还有 2 个字节的数据接收缓冲区也可以存储预读取的数据，当所有的缓冲区数据取满后，QSPI 总线周期结束。当缓冲区数据被读取后产生新的缓冲空间，QSPI 会自动开始一个新的 QSPI 总线周期来恢复预读取动作。

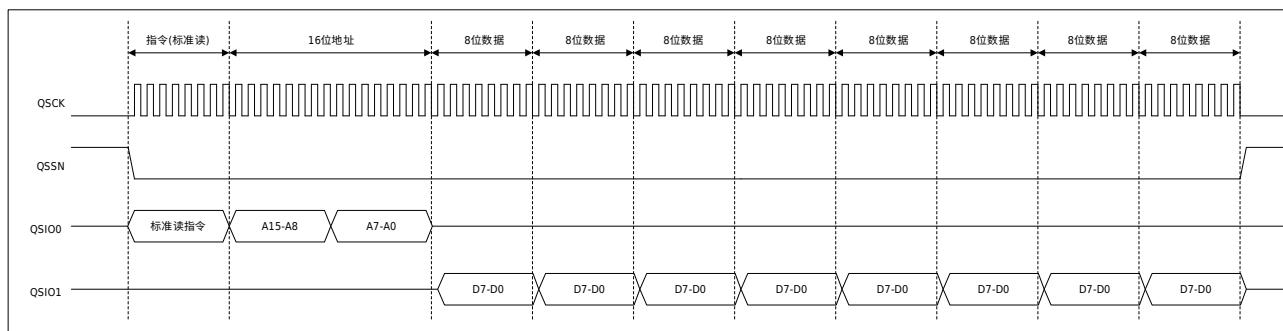


图 31-28 预读取功能有效时数据读取操作示意图

### 31.6.3 预读取的终止

在预读取传送过程中如果发生了对串行闪存其他地址的 ROM 读取总线周期，那么原预读取动作将会被终止并开始一个新的 QSPI 总线周期。通常情况下预读取动作会在当前字节传输完成之后终止，但如果将 QSCR 寄存器中的 PFSAE 位设置成 1 的话，QSPI 会即时停止预读取动作而不会等到当前字节传输完成。使用这一功能需要串行闪存器件支持即时停止动作功能。

### 31.6.4 预读取状态监测

从一个低速串行闪存读取数据会增加系统的负荷,这是因为内部总线需要一直处在等待状态直到收信的 QSPI 总线周期完成。QSPI 提供了预读取状态监测功能来降低这种负荷。

在预读取状态寄存器 QSSR 中，PFAN 位显示了当前的预读取工作状态，PFFUL 位表示预读取数据缓冲区已经放满，而 PFNUM[4:0]则显示了目前已经读取到缓冲区的数据的字节数。通过这些状态位可以很方便的通过一条 CPU 指令来确定当前的预读取状况。

**注意：**

- 当执行一段预读取状态监测程序时，请将该程序代码放置在对象串行闪存区域以外或者启用指令缓存。否则预读取对象将频繁的切换在对象数据区域和指令区域之间，失去了预读取的意义，而监测程序也会因为预读取始终无法完成而进入无限循环状态。

### 31.6.5 使用 QSPI 总线周期延长功能进行闪存读取

如果对 QSCSCR 寄存器中的 SSNW[1:0]进行设置 00 以外的值，QSPI 总线周期将会在接收完数据后处于保持状态等待下一个数据的读取。这时 QSSN 信号将保持低电平的有效状态，而 QSCK 则处于停止状态。如果下一条闪存读取指令到来时，如果读取对象地址是紧接着当前地址顺序递增的，则 QSPI 会将 QSCK 重新启动直接进行数据接受。如果是非连续的地址的话，QSSN 信号会被置成高电平的无效状态将原先保持的 QSPI 总线周期结束，然后重新开始一个新的 SPI 总线周期。

这一功能可以避免系统在对连续递增地址进行不连续读取时反复发送指令代码和地址，从而提高读取效率。

QSPI 总线周期的延长时间可以通过 SSNW[1:0]设置，当超过所设置的时间仍未发生下一次的读取时，QSSN 会自动置成高电平的无效状态，结束 QSPI 总线周期。如果 SSNW[1:0]设成 11 的话，QSSN 会被无限期延长，QSPI 总线周期将始终处在保持状态，但这样做会增加串行闪存的功耗。

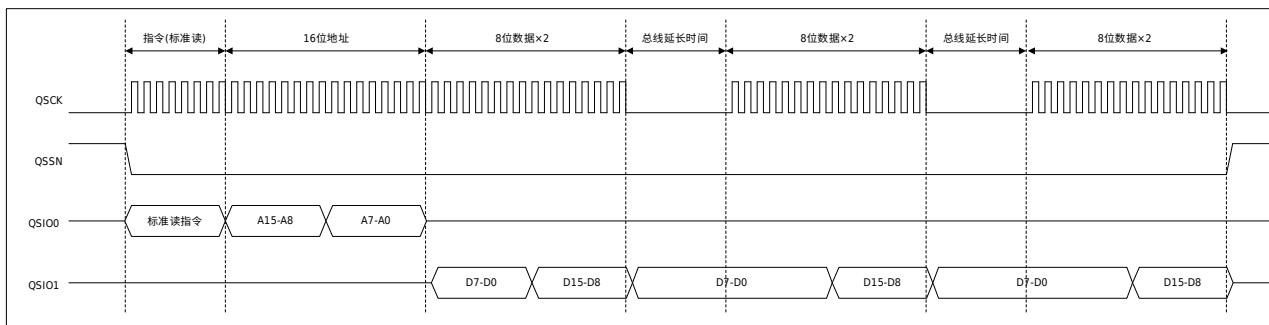


图 31-29 使用 QSPI 总线周期延长功能的数据读取操作示意图

### 31.7 XIP 控制

一些串行闪存器件可以通过省略接收读取指令来降低延迟时间。该机能可通过虚拟周期期间发送的模式代码来选择。

在快速度指令时的虚拟周期期间，QSPI 在最初的两个周期通过发送 XIP 模式代码来控制串行闪存的 XIP 模式。不同的串行闪存其 XIP 模式代码也不相同，可通过寄存器 QSXCMD 的 XIPMC[7:0]位进行针对性设置。具体参照下图 31-30。

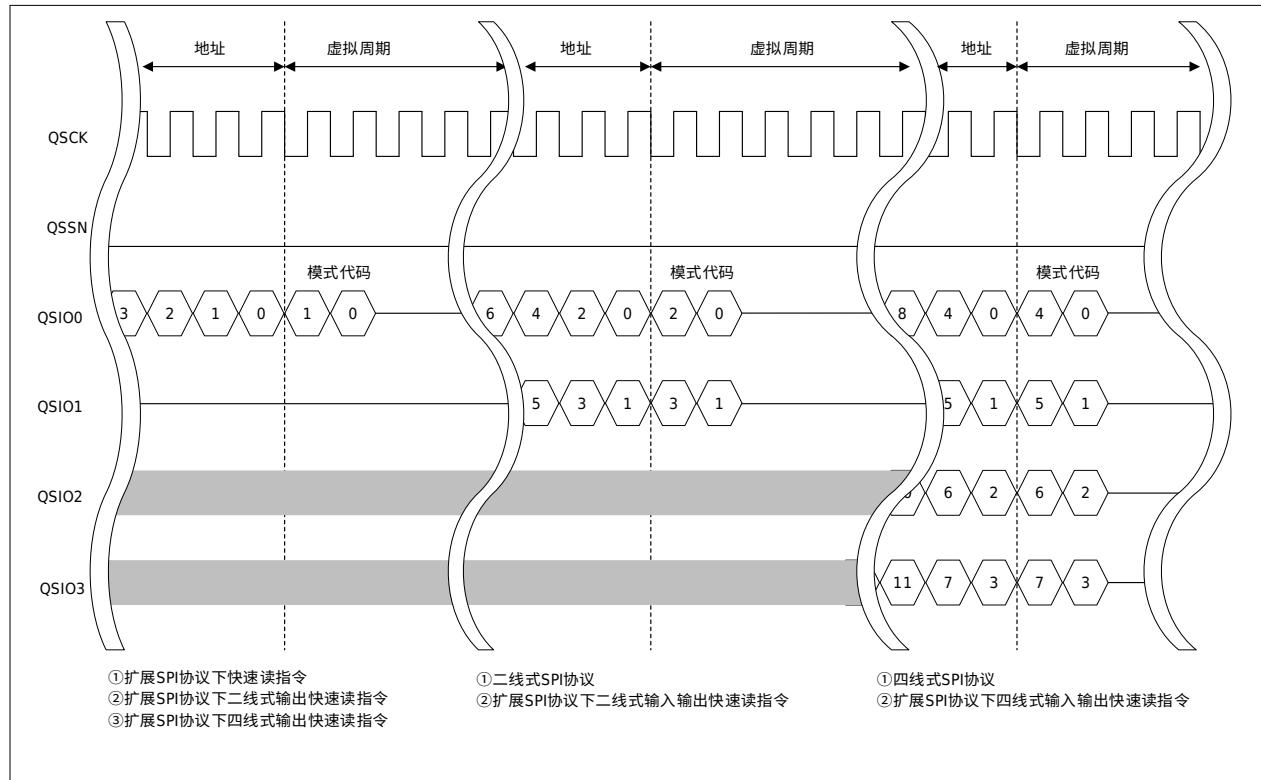


图 31-30 XIP 模式控制示意图

### 31.7.1 XIP 模式的设置

在将对应串行闪存的 XIP 模式代码写入寄存器 QSXCMD 的 XIPMC[7:0]且寄存器 QSCR 的 XIPE 位设置成 1 的情况下，当下一个快速读指令发生时，所设置的模式代码将在虚拟周期的前两个周期期间传送给对象串行闪存，在模式代码接收完成后，串行闪存及其控制部就启动 XIP 模式。可以通过访问 QSSR 的 XIPF 位来确认是否已经进入 XIP 模式。

**注意：**

- 启动串行闪存的 XIP 模式需要在 QSXCMD[7:0]中设置相应的模式代码，控制部分的 XIP 模式只需要将 XIPE 位置成 1 就可以，而与 QSXCMD[7:0]的值无关。

### 31.7.2 XIP 模式的退出

在将对应串行闪存的退出 XIP 模式的代码写入寄存器 QSXCMD 的 XIPMC[7:0]且寄存器 QSCR 的 XIPE 位设置成 0 的情况下，当下一个快速读指令发生时，所设置的退出模式代码将在虚拟周期的前两个周期期间传送给对象串行闪存，在退出模式代码接收完成后，串行闪存及其控制部的 XIP 模式被终止。可以通过访问 QSSR 的 XIPF 位来确认是否已经退出 XIP 模式。

**注意：**

- 退出串行闪存的 XIP 模式需要在 QSXCMD[7:0]中设置相应的退出模式代码，控制部分的 XIP 模式只需要将 XIPE 位清零就可以，而与 QSXCMD[7:0]的值无关。

## 31.8 QSIO2 和 QSIO3 的管脚状态

QSIO2 和 QSIO3 管脚状态取决于 QSCR 寄存器中 MDSEL[2:0]位所设置的串行读取模式。

表 31-5 QSIO2 和 QSIO3 的管脚状态

QSCR寄存器 MDSEL[2:0]位	QSIO2状态	QSIO3状态	备注
000			标准读（初始状态）
001			快速读
010	输出状态，输出电平由QSFCR寄存器的WPOL位决定，初始输出为低电平	输出高电平	二线式输出快速读
011			二线式输入输出快速读
100	作为第三根数据线进行输入或输出动作，待机状态为Hi-Z	作为第四根数据线进行输入或输出动作，待机状态为Hi-Z	四线式输出快速读
101			四线式输入输出快速读
110			自定义协议标准读
111	参照各个阶段具体的协议设置	参照各个阶段具体的协议设置	自定义协议快速读

注意：

- QSIO2 管脚也可用作串行闪存的 WP#功能。
- QSIO3 管脚也可用作串行闪存 HOLD#或 RESET#功能。

## 31.9 直接通信模式

### 31.9.1 关于直接通信模式

QSPI 可以通过自动将 MCU 的外部 ROM 读取总线周期转换为 QSPI 总线周期来对串行闪存进行读取。但串行闪存还有很多不同的追加功能，诸如 ID 信息读取，擦除，写入及状态信息读取等。这些功能并没有一套标准的指令来进行设置，并且随着串行闪存新功能的迅速增加，硬件层面上的对应变得愈发的困难。

针对这种情况，QSPI 提供了直接通信模式，用户可通过软件直接对串行闪存进行控制。由此模式软件可以产生任意所需的 QSPI 总线周期。

### 31.9.2 直接通信模式的设置

将 QSCR 寄存器的 DCOME 位设成 1 可以进入直接通信模式。一旦进入直接通信模式，将无法进行通常的闪存读取操作，如果要进行常规的闪存读取，需要将 DCOME 位清零退出直接通信模式。

注意：

- 如果处于 XIP 模式，则需要先退出 XIP 模式再启动直接通信模式。

### 31.9.3 直接通信模式下 QSPI 总线周期的生成

直接通信模式下一个完整的 QSPI 总线周期从对寄存器 QSDCOM 的 DCOM[7:0]第一次操作开始直到对 QSCR 寄存器的进行一次写操作后结束。在这期间可以对 DCOM[7:0]进行多次操作，对 DCOM[7:0]的写会转换为一次 QSPI 总线的单字节的数据传送，而对 DCOM[7:0]的读则会转换成一次 QSPI 总线的单字节的数据接收。

从对寄存器 QSCOM 的 DCOM[7:0]第一次操作开始到最后对 QSCR 寄存器的写操作为止，这期间 QSSN 信号始终保持低电平的有效状态。

直接通信模式不支持多线式动作。

**注意：**

- 在直接通信模式下是无法对 QSCR 和 QSDCOM 以外的寄存器进行写操作的。对其他寄存器的写操作将会退出直接通信模式。使用这种方式退出可能导致不可预料的情况出现所以不推荐使用。

## 31.10 中断

当直接通信模式下检测到对 ROM 的读访问操作时，QSSR 寄存器的 RAER 位被置成 1，这时候将产生 QSPI 总线错误中断。该中断请求将会一直被保留直到 RAER 位被清零。详细请参考【中断控制器（INTC）】。

## 31.11 使用上的注意事项

### 31.11.1 QSPI 寄存器的设置顺序

在系统操作中可以动态的对 QSPI 控制寄存器进行设置或更改。但是不注意寄存器的设置顺序可能会导致 QSPI 总线周期在寄存器还没有完全设置完成时就开始，因此请仔细配置寄存器的设置顺序以避免这类情况的发生。

### 31.11.2 模块停止信号的设置

QSPI 在系统复位后处于模块停止状态，只有在将模块停止控制寄存器中的 QSPI 模块停止信号清零号才可以对寄存器进行设置。具体请参考【动作模式与低功耗模式】。

## 31.12 寄存器说明

表 31-6 QSPI 寄存器一览表

寄存器基准地址：0x9C000000

寄存器名	偏移地址	复位值
QSPI控制寄存器QSCR	0x0000	0x003F0000
QSPI片选控制寄存器QSCSCR	0x0004	0x0000000F
QSPI格式控制寄存器QSFCR	0x0008	0x000080B3
QSPI状态寄存器QSSR	0x000C	0x00008000
QSPI直接通信指令寄存器QSDCOM	0x0010	0xXXXXXXXX
QSPI指令代码寄存器QSCCMD	0x0014	0x00000000
QSPIXIP模式代码寄存器QSXCMD	0x0018	0x000000FF
QSPI标志清除寄存器QSSR2（只写）	0x0024	-
QSPI外部地址寄存器QSEXAR	0x0804	0x00000000

### 31.12.1 QSPI 控制寄存器 (QSCR)

b31	b30	b29	b28	b27	b26	b25	b24
—	—	—	—	—	—	—	—
b23	b22	b21	b20	b19	b18	b17	b16
—	—			DIV[5:0]			
b15	b14	b13	b12	b11	b10	b9	b8
—	—	DPRSL[1:0]		APRSL[1:0]		IPRSL[1:0]	
b7	b6	b5	b4	b3	b2	b1	b0
SPIMD3	XIPE	DCOME	PFSAE	PFE		MDSEL[2:0]	

位	标记	位名	功能	读写
b31~b22	Reserved	—	读出时为“0”,写入时写“0”	R/W
b21~b16	DIV[5:0]	基准时钟选择位	串行接口基准时钟选择 b5 b4 b3 b2 b1 b0 0 0 0 0 0 0: 2个HCLK周期 0 0 0 0 0 1: 2个HCLK周期* 0 0 0 0 1 0: 3个HCLK周期 0 0 0 0 1 1: 4个HCLK周期* 0 0 0 1 0 0: 5个HCLK周期 0 0 0 1 0 1: 6个HCLK周期* 0 0 0 1 1 0: 7个HCLK周期 0 0 0 1 1 1: 8个HCLK周期* 0 0 1 0 0 0: 9个HCLK周期 0 0 1 0 0 1: 10个HCLK周期* 0 0 1 0 1 0: 11个HCLK周期 0 0 1 0 1 1: 12个HCLK周期* 0 0 1 1 0 0: 13个HCLK周期 0 0 1 1 0 1: 14个HCLK周期* 0 0 1 1 1 0: 15个HCLK周期 0 0 1 1 1 1: 16个HCLK周期* 0 1 0 0 0 0: 17个HCLK周期 0 1 0 0 0 1: 18个HCLK周期 0 1 0 0 1 0: 19个HCLK周期 0 1 0 0 1 1: 20个HCLK周期 0 1 0 1 0 0: 21个HCLK周期 0 1 0 1 0 1: 22个HCLK周期 0 1 0 1 1 0: 23个HCLK周期 0 1 0 1 1 1: 24个HCLK周期 0 1 1 0 0 0: 25个HCLK周期 0 1 1 0 0 1: 26个HCLK周期 0 1 1 0 1 0: 27个HCLK周期 0 1 1 0 1 1: 28个HCLK周期 0 1 1 1 0 0: 29个HCLK周期 0 1 1 1 0 1: 30个HCLK周期 0 1 1 1 1 0: 31个HCLK周期 0 1 1 1 1 1: 32个HCLK周期	R/W

1 0 0 0 0 0: 33个HCLK周期  
 1 0 0 0 0 1: 34个HCLK周期\*  
 1 0 0 0 1 0: 35个HCLK周期  
 1 0 0 0 1 1: 36个HCLK周期\*  
 1 0 0 1 0 0: 37个HCLK周期  
 1 0 0 1 0 1: 38个HCLK周期\*  
 1 0 0 1 1 0: 39个HCLK周期  
 1 0 0 1 1 1: 40个HCLK周期\*  
 1 0 1 0 0 0: 41个HCLK周期  
 1 0 1 0 0 1: 42个HCLK周期\*  
 1 0 1 0 1 0: 43个HCLK周期  
 1 0 1 0 1 1: 44个HCLK周期\*  
 1 0 1 1 0 0: 45个HCLK周期  
 1 0 1 1 0 1: 46个HCLK周期\*  
 1 0 1 1 1 0: 47个HCLK周期  
 1 0 1 1 1 1: 48个HCLK周期\*  
 1 1 0 0 0 0: 49个HCLK周期  
 1 1 0 0 0 1: 50个HCLK周期  
 1 1 0 0 1 0: 51个HCLK周期  
 1 1 0 0 1 1: 52个HCLK周期  
 1 1 0 1 0 0: 53个HCLK周期  
 1 1 0 1 0 1: 54个HCLK周期  
 1 1 0 1 1 0: 55个HCLK周期  
 1 1 0 1 1 1: 56个HCLK周期  
 1 1 1 0 0 0: 57个HCLK周期  
 1 1 1 0 0 1: 58个HCLK周期  
 1 1 1 0 1 0: 59个HCLK周期  
 1 1 1 0 1 1: 60个HCLK周期  
 1 1 1 1 0 0: 61个HCLK周期  
 1 1 1 1 0 1: 62个HCLK周期  
 1 1 1 1 1 0: 63个HCLK周期  
 1 1 1 1 1 1: 64个HCLK周期

b15~b14	Reserved	—	读出时为“0”,写入时写“0”	R/W
b13~b12	DPRSL[1:0]	数据接收阶段SPI协议选择	数据接收阶段SPI协议选择。 b1 b0 0 0: 扩展式SPI协议 0 1: 二线式SPI协议 1 0: 四线式SPI协议 1 1: 设定禁止	R/W
b11~b10	APRSL[1:0]	地址发送阶段SPI协议选择	地址发送阶段SPI协议选择。 b1 b0 0 0: 扩展式SPI协议 0 1: 二线式SPI协议 1 0: 四线式SPI协议 1 1: 设定禁止	R/W
b9~b8	IPRSL[1:0]	指令发送阶段SPI协议选择	指令发送阶段SPI协议选择。 b1 b0 0 0: 扩展式SPI协议 0 1: 二线式SPI协议	R/W

			1 0: 四线式SPI协议 1 1: 设定禁止	
b7	SPIMD3	SPI模式选择	SPI模式选择 0: SPI模式0 1: SPI模式3	R/W
b6	XIPE	XIP模式许可	0: XIP模式禁止 1: XIP模式许可	R/W
b5	DCOME	直接通信许可	QSPI总线通信模式选择 0: ROM访问模式 1: 直接通信模式	R/W
b4	PFSAE	预读取即时停止许可	选择重置预读取动作的位置 0: 当前的预读取动作在字节边界中止 1: 当前的预读取动作即时中止	R/W
b3	PFE	预读取许可	预读取功能有效/无效选择 0: 预读取功能无效 1: 预读取功能有效	R/W
b2~b0	MDSEL[2:0]	QSPI读取模式选择	串行接口读取模式选择 b2 b1 b0 0 0 0: 标准读 0 0 1: 快速读 0 1 0: 二线式输出快速读 0 1 1: 二线式输入输出快速读 1 0 0: 四线式输出快速读 1 0 1: 四线式输入输出快速读 1 1 0: 自定义标准读 1 1 1: 自定义快速读	R/W

### 31.12.2 QSPI 片选控制寄存器 (QSCSCR)

b31	b30	b29	b28	b27	b26	b25	b24
—	—	—	—	—	—	—	—
b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—	—	—
b15	b14	b13	b12	b11	b10	b9	b8
—	—	—	—	—	—	—	—
b7	b6	b5	b4	b3	b2	b1	b0
—	—	SSNW[1:0]		SSHW[3:0]			

位	标记	位名	功能	读写
b31~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5~b4	SSNW[1:0]	QSSN有效时间延长设置	QSPI总线访问后QSSN有效时间延长功能选择 b5 b4 0 0: 不延长QSSN有效时间 0 1: 将QSSN有效时间延长32个QSCK周期 1 0: 将QSSN有效时间延长128个QSCK周期 1 1: 将QSSN有效时间无限延长	R/W
b3~b0	SSHW[3:0]	QSSN最小无效时间设置	QSSN信号最小无效时间选择 b3 b2 b1 b0 0 0 0 0: 1个QSCK周期 0 0 0 1: 2个QSCK周期 0 0 1 0: 3个QSCK周期 0 0 1 1: 4个QSCK周期 0 1 0 0: 5个QSCK周期 0 1 0 1: 6个QSCK周期 0 1 1 0: 7个QSCK周期 0 1 1 1: 8个QSCK周期 1 0 0 0: 9个QSCK周期 1 0 0 1: 10个QSCK周期 1 0 1 0: 11个QSCK周期 1 0 1 1: 12个QSCK周期 1 1 0 0: 13个QSCK周期 1 1 0 1: 14个QSCK周期 1 1 1 0: 15个QSCK周期 1 1 1 1: 16个QSCK周期	R/W

### 31.12.3 QSPI 格式控制寄存器 (QSFCR)

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	-	-	-	-
b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8
DUTY	-	-	-	DMCYCN[3:0]			
b7	b6	b5	b4	b3	b2	b1	b0
-	WPOL	SSNLD	SSNHD	-	4BIC	AWSL[1:0]	

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15	DUTY	占空比补正	QSCK输出波形占空比补正 0: 不进行占空比补正 1: 将QSCK的上升沿滞后0.5个HCLK周期 (当QSCK选择的频率是HCLK的奇数倍时有效)	R/W
b14~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11~b8	DMCYCN[3:0]	虚拟周期设置	使用快速读指令时虚拟周期数量选择 b3 b2 b1 b0 0 0 0 0: 3个QSCK周期 0 0 0 1: 4个QSCK周期 0 0 1 0: 5个QSCK周期 0 0 1 1: 6个QSCK周期 0 1 0 0: 7个QSCK周期 0 1 0 1: 8个QSCK周期 0 1 1 0: 9个QSCK周期 0 1 1 1: 10个QSCK周期 1 0 0 0: 11个QSCK周期 1 0 0 1: 12个QSCK周期 1 0 1 0: 13个QSCK周期 1 0 1 1: 14个QSCK周期 1 1 0 0: 15个QSCK周期 1 1 0 1: 16个QSCK周期 1 1 1 0: 17个QSCK周期 1 1 1 1: 18个QSCK周期	R/W
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6	WPOL	WP管脚输出电平设置	WP管脚 (QIO2) 电平设置 0: 低电平 1: 高电平	R/W
b5	SSNLD	QSSN信号输出时间延迟设置	QSSN信号输出时序选择 0: 比QSCK第一个上升沿提前0.5个QSCK输出QSSN 1: 比QSCK第一个上升沿提前1.5个QSCK输出QSSN	R/W
b4	SSNHD	QSSN信号释放时间延迟设置	QSSN信号释放时序选择 0: 比QSCK最后一个上升沿滞后0.5个QSCK释放QSSN 1: 比QSCK最后一个上升沿滞后1.5个QSCK释放QSSN	R/W

b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	4BIC	4字节地址读指令 代码选择	当地址宽度为4字节时的读指令代码选择 0: 不使用4字节地址读指令代码 1: 使用4字节地址读指令代码	R/W
b1~b0	AWSL[1:0]	地址宽度选择	串行接口地址宽度选择 b1 b0 0 0: 1字节 0 1: 2字节 1 0: 3字节 1 1: 4字节	R/W

### 31.12.4 QSPI 状态寄存器 (QSSR)

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	-	-	-	-
b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8
PFAN	PFFUL	-		PFNUM[4:0]			
b7	b6	b5	b4	b3	b2	b1	b0
RAER	XIPF	-	-	-	-	-	BUSY

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15	PFAN	预读取动作状态	0: 预读取处于动作状态 1: 预读取处于停止状态	R
b14	PFFUL	预读取缓冲区状态	0: 预读取缓冲区有剩余空间 1: 预读取缓冲区数据已满	R
b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12~b8	PFNUM[4:0]	预读取缓冲区已存储数据字节数	显示 b4 b3 b2 b1 b0 0 0 0 0 0: 0个字节 0 0 0 0 1: 1个字节 0 0 0 1 0: 2个字节 0 0 0 1 1: 3个字节 0 0 1 0 0: 4个字节 0 0 1 0 1: 5个字节 0 0 1 1 0: 6个字节 0 0 1 1 1: 7个字节 0 1 0 0 0: 8个字节 0 1 0 0 1: 9个字节 0 1 0 1 0: 10个字节 0 1 0 1 1: 11个字节 0 1 1 0 0: 12个字节 0 1 1 0 1: 13个字节 0 1 1 1 0: 14个字节 0 1 1 1 1: 15个字节 1 0 0 0 0: 16个字节 1 0 0 0 1: 17个字节 1 0 0 1 0: 18个字节 其余设定无效	R
b7	RAER*1	ROM访问错误标志	直接通信模式下发生ROM访问的错误标志位 0: 未检测到发生ROM访问 1: 检测到发生ROM访问	R/W
b6	XIPF	XIP模式标志	XIP模式状态信号	R

			0: 非XIP模式 1: XIP模式	
b5~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	BUSY	总线忙标志	直接通信模式下QSPI总线工作状态标志位 0: 总线空闲, 无串行传输进程 1: 总线忙, 串行传输进程进行中	R

\*1: RAER 需通过 QSSR2 的 RAERCLR 位清零

### 31.12.5 QSPI 指令代码寄存器 (QSCCMD)

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	-	-	-	-
b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	-	-	-	-
b7	b6	b5	b4	b3	b2	b1	b0
RIC [7:0]							

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7~b0	RIC[7:0]	替换指令代码	用于替代默认指令的串行闪存指令代码	R/W

### 31.12.6 QSPI 直接通信指令寄存器 (QSDCOM)

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	-	-	-	-
b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	-	-	-	-
b7	b6	b5	b4	b3	b2	b1	b0
DCOM[7:0]							

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7~b0	DCOM[7:0]	直接通信模式指令	直接通信模式时的接口，通过QSPI总线进行直接通信。对该接口的读写访问会被转化成一个相应的QSPI总线周期。该接口只在直接通信模式下有效，ROM访问模式下该接口禁止访问。	R/W

### 31.12.7 QSPI XIP 模式代码寄存器 (QSXCMD)

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	-	-	-	-
b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	-	-	-	-
b7	b6	b5	b4	b3	b2	b1	b0
XIPMC [7:0]							

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7~b0	XIPMC[7:0]	XIP模式代码	用于串行闪存的模式代码。(设置XIP模式)	R/W

### 31.12.8 QSPI 标志清除寄存器 (QSSR2)

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	-	-	-	-
b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	-	-	-	-
b7	b6	b5	b4	b3	b2	b1	b0
RAERCLR	-	-	-	-	-	-	-

位	标记	位名	功能	读写
b31~b8	Reserved	-	写入时写“0”	W
b7	RAERCLR	RAER清除	写入1时将QSSR中的RAER位清零	W
b6~b0	Reserved	-	写入时写“0”	W

### 31.12.9 QSPI 外部扩展地址寄存器 (QSEXAR)

b31	b30	b29	b28	b27	b26	b25	b24
EXADR[5:0]						—	—
b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—	—	—
b15	b14	b13	b12	b11	b10	b9	b8
—	—	—	—	—	—	—	—
b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	—

位	标记	位名	功能	读写
b31~b26	EXADR[5:0]	外部扩展地址代码	QSPI外部地址高6位设置，配合QSPI的ROM访问窗口地址最大可访问64MB×63块的外部ROM空间	R/W
b25~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

## 32 可编程逻辑阵列 (PLA)

### 32.1 简介

可配置逻辑阵列为外部引脚、ADC 和定时器提供 256 个可编程数字逻辑操作，而无需 CPU 干预。本模块实现 16 个独立的 PLA 单元。每个 PLA 单元支持 GPIO 端口的输出。

#### PLA 主要特性：

- 16 个独立的 PLA 单元，每个 PLA 单元具有两个输入多路选择器，支持 28 个输入信号，包括外部引脚、定时器通道、ADC 和 PLA 输出。
- 在每个 PLA 单元中实现一个查找表 (LUT)，提供 256 个可编程数字逻辑功能。
- 16 个 PLA 单元可以组合并支持复杂的逻辑操作。

## 32.2 PLA 模块框图

### 32.2.1 模块框图

PLA 模块主要由以下部分组成：

- 通过 APB 接口对 PLA 配置
- 16 个 PLA 单元

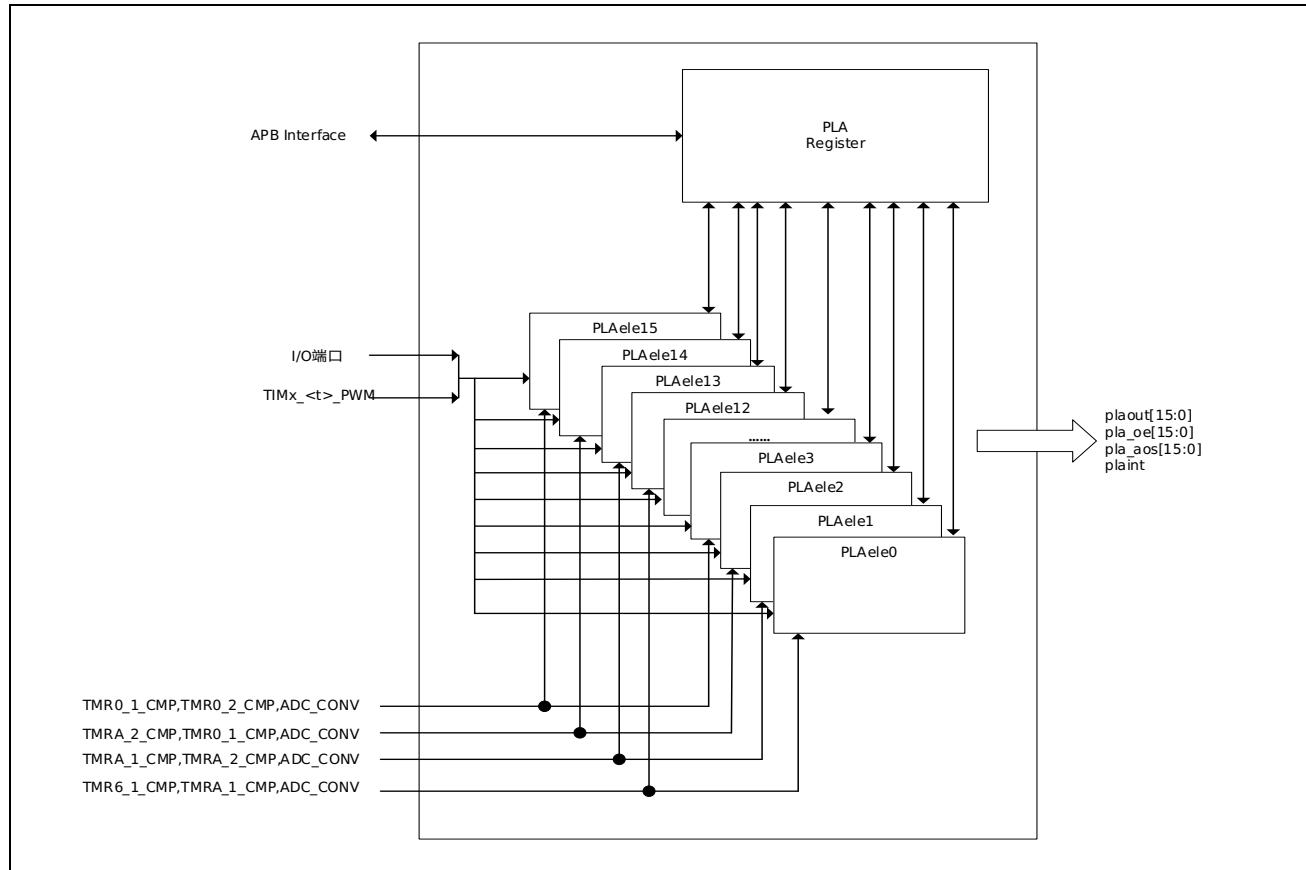


图 32-1 PLA 模块框图

### 32.2.2 动作说明

本节提供了 PLA 模块功能的描述。模块包含十六个相同的 PLA 单元，图 32-2 显示了 PLA 单元的框图，每个 PLA 单元包含两个多路选择器和一个 LUT。

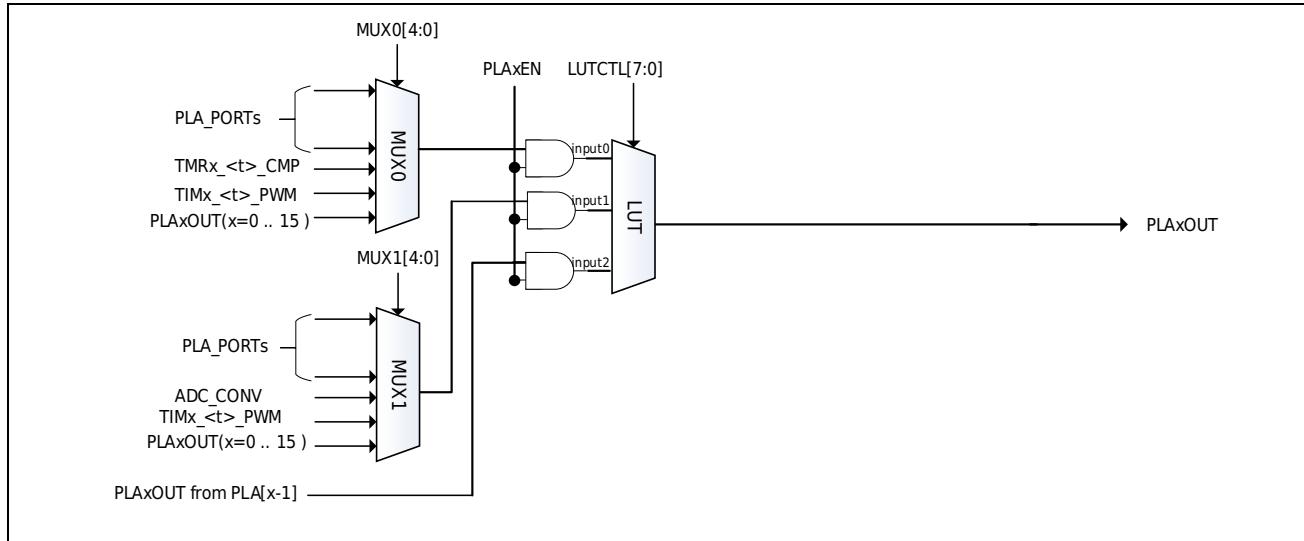


图 32-2 PLA 单元框图

### 32.2.3 PLA 输入多路选择器

每个 PLA 单元包含两个多路选择器：MUX0 和 MUX1。每个多路选择器的输入可以是：外部引脚（PLA\_PORTS），定时器比较事件触发信号（TIMx\_<t>\_CMP），定时器通道（TIMx\_<t>\_PWM），ADC 转换信号（ADC\_CONV）和 PLA 单元输出（PLAx OUT）。当另一个 PLA 单元的输出作为 MUX 的输入时，可以实现一个复杂的组合逻辑运算。

表 32-1 和表 32-2 显示了 PLAxMUX0 和 PLAxMUX1 的输入选择。

表 32-1 PLAxMUX0 输入选择

MUX0[4:0]	PLA0MUX0	PLA1MUX0	PLA2MUX0	PLA3MUX0	PLA4MUX0	PLA5MUX0	PLA6MUX0	PLA7MUX0	PLA8MUX0	PLA9MUX0	PLA10MUX0	PLA11MUX0	PLA12MUX0	PLA13MUX0	PLA14MUX0	PLA15MUX0
00000	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT												
00001	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT												
00010	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT												
00011	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT												
00100	TMR_A_1_CMP	TMR_A_2_CMP	TMR_B_1_TRIG	TMR_B_2_TRIG												
00101	TIM6_7_PWMA	TIM6_7_PWMA	TIM6_8_PWMA	TIM6_9_PWMA	TIM6_7_PWMA	TIM6_7_PWMA	TIM6_8_PWMA	TIM6_9_PWMA	TIM6_7_PWMA	TIM6_8_PWMA	TIM6_9_PWMA	TIM6_7_PWMA	TIM6_8_PWMA	TIM6_9_PWMA	TIM6_7_PWMA	TIM6_8_PWMA
00110	TIM6_8_PWMA	TIM6_10_PWMA	TIM6_10_PWMA	TIM_A_1_PWM1	TIM6_8_PWMA	TIM6_10_PWMA	TIM6_10_PWMA	TIM_A_1_PWM1	TIM6_8_PWMA	TIM6_10_PWMA	TIM_A_1_PWM1	TIM6_8_PWMA	TIM6_10_PWMA	TIM6_10_PWMA	TIM6_10_PWM1	TIM6_10_PWM1
00111	TIM6_9_PWMA	TIM_A_1_PWM1	TIM_A_1_PWM2	TIM_A_1_PWM2	TIM6_9_PWMA	TIM_A_1_PWM1	TIM_A_1_PWM2	TIM6_9_PWMA	TIM_A_1_PWM1	TIM_A_1_PWM2	TIM6_9_PWMA	TIM_A_1_PWM1	TIM_A_1_PWM2	TIM6_9_PWMA	TIM_A_1_PWM1	TIM_A_1_PWM2
01000	PLAIN0_(PA15)	PLAIN4_(PB6)	PLAIN0_(PA15)	PLAIN20_(PF4)												
01001	PLAIN2_(PB4)	PLAIN5_(PB7)	PLAIN20_(PB15)	PLAIN2_(PB4)	PLAIN5_(PB7)	PLAIN20_(PB4)	PLAIN2_(PB7)	PLAIN5_(PB7)	PLAIN20_(PB4)	PLAIN2_(PB7)	PLAIN5_(PB7)	PLAIN20_(PB4)	PLAIN2_(PB7)	PLAIN5_(PB7)	PLAIN20_(PB4)	PLAIN2_(PB7)
01010	PLAIN4_(PB6)	PLAIN8_(PB0)	PLAIN8_(PB0)	PLAIN6_(PB8)	PLAIN4_(PB6)	PLAIN8_(PB8)	PLAIN6_(PB8)	PLAIN4_(PB6)	PLAIN8_(PB8)	PLAIN6_(PB8)	PLAIN4_(PB6)	PLAIN8_(PB8)	PLAIN6_(PB8)	PLAIN8_(PB0)	PLAIN6_(PB8)	PLAIN8_(PB8)
01011	PLAIN6_(PB8)	PLAIN10_(PB2)	PLAIN9_(PB1)	PLAIN7_(PB9)												
01100	PLAIN8_(PB0)	PLAIN12_(PA9)	PLAIN14_(PA11)	PLAIN10_(PB2)												
01101	PLAIN10_(PB2)	PLAIN13_(PA10)	PLAIN15_(PA12)	PLAIN11_(PA8)	PLAIN10_(PB2)	PLAIN13_(PA10)	PLAIN15_(PA12)	PLAIN11_(PA8)	PLAIN10_(PA2)	PLAIN13_(PA10)	PLAIN15_(PA12)	PLAIN11_(PA8)	PLAIN10_(PA2)	PLAIN13_(PA10)	PLAIN15_(PA12)	PLAIN11_(PA8)
01110	PLAIN12_(PA9)	PLAIN16_(PF0)	PLAIN16_(PF0)	PLAIN18_(PA0)	PLAIN12_(PA9)	PLAIN16_(PF0)	PLAIN18_(PA0)	PLAIN12_(PA9)	PLAIN16_(PF0)	PLAIN16_(PA9)	PLAIN18_(PA0)	PLAIN12_(PA9)	PLAIN16_(PF0)	PLAIN16_(PA9)	PLAIN16_(PF0)	PLAIN18_(PA0)
01111	PLAIN14_(PA11)	PLAIN18_(PA0)	PLAIN17_(PF1)	PLAIN19_(PA1)												
10000	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT												
10001	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT												
10010	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT												
10011	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT												
10100	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT												
10101	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT												
10110	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT												
10111	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT												
11000	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT												
11001	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT												
11010	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT												
11011	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT												

表 32-2 PLA<sub>x</sub>MUX1 输入选择

MUX1[4:0]	PLA0MUX1	PLA1MUX1	PLA2MUX1	PLA3MUX1	PLA4MUX1	PLA5MUX1	PLA6MUX1	PLA7MUX1	PLA8MUX1	PLA9MUX1	PLA10MUX1	PLA11MUX1	PLA12MUX1	PLA13MUX1	PLA14MUX1	PLA15MUX1
00000	PLA0 OUT	PLA0 OUT	PLA0 OUT	PLA0 OUT	PLA0 OUT	PLA0 OUT	PLA0 OUT	PLA0 OUT	PLA0 OUT	PLA0 OUT	PLA0 OUT	PLA0 OUT	PLA0 OUT	PLA0 OUT	PLA0 OUT	PLA0 OUT
00001	PLA1 OUT	PLA1 OUT	PLA1 OUT	PLA1 OUT	PLA1 OUT	PLA1 OUT	PLA1 OUT	PLA1 OUT	PLA1 OUT	PLA1 OUT	PLA1 OUT	PLA1 OUT	PLA1 OUT	PLA1 OUT	PLA1 OUT	PLA1 OUT
00010	PLA2 OUT	PLA2 OUT	PLA2 OUT	PLA2 OUT	PLA2 OUT	PLA2 OUT	PLA2 OUT	PLA2 OUT	PLA2 OUT	PLA2 OUT	PLA2 OUT	PLA2 OUT	PLA2 OUT	PLA2 OUT	PLA2 OUT	PLA2 OUT
00011	PLA3 OUT	PLA3 OUT	PLA3 OUT	PLA3 OUT	PLA3 OUT	PLA3 OUT	PLA3 OUT	PLA3 OUT	PLA3 OUT	PLA3 OUT	PLA3 OUT	PLA3 OUT	PLA3 OUT	PLA3 OUT	PLA3 OUT	PLA3 OUT
00100	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV
00101	TIM6_10_PWMA	TIM6_8_PWMA	TIM6_7_PWMA	TIM6_7_PWMA	TIM6_10_PWMA	TIM6_8_PWMA	TIM6_7_PWMA	TIM6_10_PWMA	TIM6_8_PWMA	TIM6_7_PWMA	TIM6_7_PWMA	TIM6_10_PWMA	TIM6_8_PWMA	TIM6_7_PWMA	TIM6_7_PWMA	TIM6_7_PWMA
00110	TIMA_1_PWM1	TIM6_9_PWMA	TIM6_9_PWM	TIM6_8_PWM	TIMA_1_PWM1	TIM6_9_PWM	TIM6_8_PWM	TIMA_1_PWM1	TIM6_9_PWM	TIM6_9_PWM	TIM6_8_PWM	TIMA_1_PWM1	TIM6_9_PWM	TIM6_8_PWM	TIM6_8_PWM	TIM6_8_PWM
00111	TIMA_1_PWM2	TIMA_1_PWM2	TIMA_1_PWM1	TIM6_10_PWM	TIMA_1_PWM2	TIMA_1_PWM2	TIMA_1_PWM1	TIM6_10_PWM	TIMA_1_PWM2	TIMA_1_PWM2	TIMA_1_PWM1	TIM6_10_PWM	TIMA_1_PWM2	TIMA_1_PWM2	TIMA_1_PWM1	TIM6_10_PWM
01000	PLAIN1 (PB3)	PLAIN6 (PB8)	PLAIN2 (PB4)	PLAIN0 (PA15)	PLAIN1 (PB3)	PLAIN6 (PB4)	PLAIN2 (PA15)	PLAIN0 (PB3)	PLAIN1 (PB8)	PLAIN2 (PB4)	PLAIN0 (PA15)	PLAIN1 (PB3)	PLAIN6 (PB8)	PLAIN2 (PB4)	PLAIN0 (PA15)	PLAIN2 (PA15)
01001	PLAIN3 (PB5)	PLAIN7 (PB9)	PLAIN3 (PB5)	PLAIN1 (PB3)	PLAIN7 (PB9)	PLAIN3 (PB5)	PLAIN1 (PB3)	PLAIN3 (PB5)	PLAIN7 (PB9)	PLAIN1 (PB5)	PLAIN1 (PB3)	PLAIN3 (PB5)	PLAIN7 (PB9)	PLAIN3 (PB5)	PLAIN1 (PB3)	PLAIN1 (PB3)
01010	PLAIN5 (PB7)	PLAIN9 (PB1)	PLAIN10 (PB2)	PLAIN4 (PB6)	PLAIN9 (PB1)	PLAIN10 (PB2)	PLAIN4 (PB6)	PLAIN5 (PB6)	PLAIN9 (PB1)	PLAIN10 (PB2)	PLAIN4 (PB6)	PLAIN5 (PB2)	PLAIN9 (PB1)	PLAIN10 (PB2)	PLAIN4 (PB6)	PLAIN4 (PB6)
01011	PLAIN6 (PB8)	PLAIN11 (PA8)	PLAIN11 (PA8)	PLAIN5 (PB7)	PLAIN6 (PB8)	PLAIN11 (PA8)	PLAIN5 (PB8)	PLAIN6 (PA8)	PLAIN11 (PB8)	PLAIN11 (PA8)	PLAIN5 (PB7)	PLAIN6 (PB8)	PLAIN11 (PA8)	PLAIN11 (PA8)	PLAIN5 (PB7)	PLAIN5 (PB7)
01100	PLAIN9 (PB1)	PLAIN14 (PA11)	PLAIN12 (PA9)	PLAIN8 (PB0)	PLAIN9 (PA11)	PLAIN14 (PA9)	PLAIN12 (PA0)	PLAIN8 (PB0)	PLAIN9 (PA11)	PLAIN14 (PA9)	PLAIN12 (PA0)	PLAIN8 (PB0)	PLAIN9 (PA11)	PLAIN14 (PA11)	PLAIN12 (PA9)	PLAIN8 (PB0)
01101	PLAIN11 (PA8)	PLAIN15 (PA12)	PLAIN13 (PA10)	PLAIN9 (PB1)	PLAIN11 (PA8)	PLAIN15 (PA10)	PLAIN13 (PA12)	PLAIN9 (PA8)	PLAIN11 (PA8)	PLAIN15 (PA12)	PLAIN13 (PA10)	PLAIN9 (PA8)	PLAIN11 (PA8)	PLAIN15 (PA10)	PLAIN9 (PA8)	PLAIN9 (PA8)
01110	PLAIN13 (PA10)	PLAIN17 (PF1)	PLAIN18 (PA0)	PLAIN16 (PF0)	PLAIN13 (PA10)	PLAIN17 (PF1)	PLAIN18 (PA0)	PLAIN16 (PF0)	PLAIN13 (PA10)	PLAIN17 (PF1)	PLAIN18 (PA0)	PLAIN16 (PF0)	PLAIN13 (PA10)	PLAIN17 (PF1)	PLAIN18 (PA0)	PLAIN16 (PF0)
01111	PLAIN15 (PA12)	PLAIN19 (PA1)	PLAIN19 (PA1)	PLAIN17 (PF1)	PLAIN15 (PA12)	PLAIN19 (PA1)	PLAIN19 (PA1)	PLAIN17 (PF1)	PLAIN15 (PA12)	PLAIN19 (PA1)	PLAIN19 (PA1)	PLAIN17 (PF1)	PLAIN15 (PA12)	PLAIN19 (PA1)	PLAIN19 (PA1)	PLAIN17 (PF1)
10000	PLA4 OUT	PLA4 OUT	PLA4 OUT	PLA4 OUT	PLA4 OUT	PLA4 OUT	PLA4 OUT	PLA4 OUT	PLA4 OUT	PLA4 OUT	PLA4 OUT	PLA4 OUT	PLA4 OUT	PLA4 OUT	PLA4 OUT	PLA4 OUT
10001	PLA5 OUT	PLA5 OUT	PLA5 OUT	PLA5 OUT	PLA5 OUT	PLA5 OUT	PLA5 OUT	PLA5 OUT	PLA5 OUT	PLA5 OUT	PLA5 OUT	PLA5 OUT	PLA5 OUT	PLA5 OUT	PLA5 OUT	PLA5 OUT
10010	PLA6 OUT	PLA6 OUT	PLA6 OUT	PLA6 OUT	PLA6 OUT	PLA6 OUT	PLA6 OUT	PLA6 OUT	PLA6 OUT	PLA6 OUT	PLA6 OUT	PLA6 OUT	PLA6 OUT	PLA6 OUT	PLA6 OUT	PLA6 OUT
10011	PLA7 OUT	PLA7 OUT	PLA7 OUT	PLA7 OUT	PLA7 OUT	PLA7 OUT	PLA7 OUT	PLA7 OUT	PLA7 OUT	PLA7 OUT	PLA7 OUT	PLA7 OUT	PLA7 OUT	PLA7 OUT	PLA7 OUT	PLA7 OUT
10100	PLA8 OUT	PLA8 OUT	PLA8 OUT	PLA8 OUT	PLA8 OUT	PLA8 OUT	PLA8 OUT	PLA8 OUT	PLA8 OUT	PLA8 OUT	PLA8 OUT	PLA8 OUT	PLA8 OUT	PLA8 OUT	PLA8 OUT	PLA8 OUT
10101	PLA9 OUT	PLA9 OUT	PLA9 OUT	PLA9 OUT	PLA9 OUT	PLA9 OUT	PLA9 OUT	PLA9 OUT	PLA9 OUT	PLA9 OUT	PLA9 OUT	PLA9 OUT	PLA9 OUT	PLA9 OUT	PLA9 OUT	PLA9 OUT
10110	PLA10 OUT	PLA10 OUT	PLA10 OUT	PLA10 OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT
10111	PLA11 OUT	PLA11 OUT	PLA11 OUT	PLA11 OUT	PLA11 OUT	PLA11 OUT	PLA11 OUT	PLA11 OUT	PLA11 OUT	PLA11 OUT	PLA11 OUT	PLA11 OUT	PLA11 OUT	PLA11 OUT	PLA11 OUT	PLA11 OUT
11000	PLA12 OUT	PLA12 OUT	PLA12 OUT	PLA12 OUT	PLA12 OUT	PLA12 OUT	PLA12 OUT	PLA12 OUT	PLA12 OUT	PLA12 OUT	PLA12 OUT	PLA12 OUT	PLA12 OUT	PLA12 OUT	PLA12 OUT	PLA12 OUT
11001	PLA13 OUT	PLA13 OUT	PLA13 OUT	PLA13 OUT	PLA13 OUT	PLA13 OUT	PLA13 OUT	PLA13 OUT	PLA13 OUT	PLA13 OUT	PLA13 OUT	PLA13 OUT	PLA13 OUT	PLA13 OUT	PLA13 OUT	PLA13 OUT
11010	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT
11011	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT

### 32.2.4 查找表 (LUT) 控制

每个 PLA 单元都有一个 LUT, LUT 引入了 256 种逻辑组合功能, 通过 PLAx\_LUTCTL 寄存器的 LUTCTL[7:0] 位控制。由 input2、input1、input0 三位共同控制查找表的输出, 如表 33-3 所示。

表 32-3 LUT 控制

input2	input1	input0	Output
0	0	0	LUTCTL[0]
0	0	1	LUTCTL[1]
0	1	0	LUTCTL[2]
0	1	1	LUTCTL[3]
1	0	0	LUTCTL[4]
1	0	1	LUTCTL[5]
1	1	0	LUTCTL[6]
1	1	1	LUTCTL[7]

例如, 要实现 {input2<sup>^</sup>input1<sup>^</sup>input0} 的逻辑时, 需设置 LUTCTL[7:0]= 8'b10010110。

应用实例：欲实现 pla\_ele0 单元的 {input2<sup>^</sup>input1<sup>^</sup>input0} 的逻辑, 先对 PLA 及 GPIO 进行配置:

将 PSPCR 设定为 0x0000 以关闭 JTAG 等特殊功能,

将 PFSR(0,15) 设定为 0x9、PLAIN0 设定为 PA15,

将 PFSR(1,3) 设定为 0x9、PLAIN1 设定为 PB3,

将 PFSR(0,2) 设定为 0xa、PLA0OUT 设定为 PA2,

PLA0\_CTL=0x000000C0, 选择 LUT 结果直接输出, 允许 PLA\_OUT0 输出,

PLA0\_LUTCTL=0x00000096, 实现 {input2<sup>^</sup>input1<sup>^</sup>input0} 的逻辑,

PLA0\_MUXS=0x00000088, 选通 MUX0 01000 和 MUX1 01000,

PLA\_GCTL=0x00000001, 使能 PLA0 单元。

### 32.2.5 PLA 输出

每个 PLA 单元都有一个 GPIO 端口作为其输出，该输出是 LUT 的结果。

表 32-4 PLA 输出端口对应表

引脚名称	输入/输出	功能
PA2/PB3	O	PLA0OUT
PB0/PB4/PA7	O	PLA1OUT
PA10/PB5/PC10	O	PLA2OUT
PC13/PA11/PA15/PB8	O	PLA3OUT
PC0/PA3	O	PLA4OUT
PC1/PA4	O	PLA5OUT
PA5/PC2	O	PLA6OUT
PA6/PC3	O	PLA7OUT
PA13	O	PLA8OUT
PF6	O	PLA9OUT
PF7	O	PLA10OUT
PA4	O	PLA11OUT
PC4	O	PLA12OUT
PC5	O	PLA13OUT
PC11	O	PLA14OUT
PC12	O	PLA15OUT

### 32.3 寄存器说明

表 32-5 寄存器一览

BASE\_ADDR: 0x4004FC00

寄存器名	符号	偏移地址	位宽	复位值
全局控制寄存器	PLA_GCTL	0x00	32	0x00000000
状态寄存器	PLA_STAT	0x0C	32	0x00000000
多路选择器选择寄存器	PLAx_MUXS	0x10 + 0x0C * x	32	0x00000000
查找表控制寄存器	PLAx_LUTCTL	0x14 + 0x0C * x	32	0x00000000
控制寄存器	PLAx_CTL	0x18 + 0x0C * x	32	0x00000000

注\*1：地址计算公式中 x= 0~15

### 32.3.1 全局控制寄存器 (PLA\_GCTL)

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PLA1 SEN	PLA1 4EN	PLA1 3EN	PLA1 2EN	PLA1 1EN	PLA1 0EN	PLA9 EN	PLA8 EN	PLA7 EN	PLA6 EN	PLA5 EN	PLA4 EN	PLA3 EN	PLA2 EN	PLA1 EN	PLA0 EN
-															
位	标记		位名		功能								读写		
b31~b16	Reserved		-		读出时为 0, 写时请写 0								R/W		
b15	PLA15EN		PLA15 单元使能		0: 禁止 PLA15 单元 1: 允许 PLA15 单元								R/W		
b14	PLA14EN		PLA14 单元使能		0: 禁止 PLA14 单元 1: 允许 PLA14 单元								R/W		
b13	PLA13EN		PLA13 单元使能		0: 禁止 PLA13 单元 1: 允许 PLA13 单元								R/W		
b12	PLA12EN		PLA12 单元使能		0: 禁止 PLA12 单元 1: 允许 PLA12 单元								R/W		
b11	PLA11EN		PLA11 单元使能		0: 禁止 PLA11 单元 1: 允许 PLA11 单元								R/W		
b10	PLA10EN		PLA10 单元使能		0: 禁止 PLA10 单元 1: 允许 PLA10 单元								R/W		
b9	PLA9EN		PLA9 单元使能		0: 禁止 PLA9 单元 1: 允许 PLA9 单元								R/W		
b8	PLA8EN		PLA8 单元使能		0: 禁止 PLA8 单元 1: 允许 PLA8 单元								R/W		
b7	PLA7EN		PLA7 单元使能		0: 禁止 PLA7 单元 1: 允许 PLA7 单元								R/W		
b6	PLA6EN		PLA6 单元使能		0: 禁止 PLA6 单元 1: 允许 PLA6 单元								R/W		
b5	PLA5EN		PLA5 单元使能		0: 禁止 PLA5 单元 1: 允许 PLA5 单元								R/W		
b4	PLA4EN		PLA4 单元使能		0: 禁止 PLA4 单元 1: 允许 PLA4 单元								R/W		
b3	PLA3EN		PLA3 单元使能		0: 禁止 PLA3 单元 1: 允许 PLA3 单元								R/W		
b2	PLA2EN		PLA2 单元使能		0: 禁止 PLA2 单元 1: 允许 PLA2 单元								R/W		
b1	PLA1EN		PLA1 单元使能		0: 禁止 PLA1 单元 1: 允许 PLA1 单元								R/W		
b0	PLAOEN		PLAO 单元使能		0: 禁止 PLAO 单元 1: 允许 PLAO 单元								R/W		

### 32.3.2 状态寄存器 (PLA\_STAT)

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PLA1 5OUT	PLA1 4OUT	PLA1 3OUT	PLA1 2OUT	PLA1 1OUT	PLA1 0OUT	PLA9 OUT	PLA8 OUT	PLA7 OUT	PLA6 OUT	PLA5 OUT	PLA4 OUT	PLA3 OUT	PLA2 OUT	PLA1 OUT	PLA0 OUT
-															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为 0	R											
b15	PLA15OUT	PLA15 单元输出状态	0: PLA15 单元当前逻辑电平为低 1: PLA15 单元当前逻辑电平为高	R											
b14	PLA14OUT	PLA14 单元输出状态	0: PLA14 单元当前逻辑电平为低 1: PLA14 单元当前逻辑电平为高	R											
b13	PLA13OUT	PLA13 单元输出状态	0: PLA13 单元当前逻辑电平为低 1: PLA13 单元当前逻辑电平为高	R											
b12	PLA12OUT	PLA12 单元输出状态	0: PLA12 单元当前逻辑电平为低 1: PLA12 单元当前逻辑电平为高	R											
b11	PLA11OUT	PLA11 单元输出状态	0: PLA11 单元当前逻辑电平为低 1: PLA11 单元当前逻辑电平为高	R											
b10	PLA10OUT	PLA10 单元输出状态	0: PLA10 单元当前逻辑电平为低 1: PLA10 单元当前逻辑电平为高	R											
b9	PLA9OUT	PLA9 单元输出状态	0: PLA9 单元当前逻辑电平为低 1: PLA9 单元当前逻辑电平为高	R											
b8	PLA8OUT	PLA8 单元输出状态	0: PLA8 单元当前逻辑电平为低 1: PLA8 单元当前逻辑电平为高	R											
b7	PLA7OUT	PLA7 单元输出状态	0: PLA7 单元当前逻辑电平为低 1: PLA7 单元当前逻辑电平为高	R											
b6	PLA6OUT	PLA6 单元输出状态	0: PLA6 单元当前逻辑电平为低 1: PLA6 单元当前逻辑电平为高	R											
b5	PLA5OUT	PLA5 单元输出状态	0: PLA5 单元当前逻辑电平为低 1: PLA5 单元当前逻辑电平为高	R											
b4	PLA4OUT	PLA4 单元输出状态	0: PLA4 单元当前逻辑电平为低 1: PLA4 单元当前逻辑电平为高	R											
b3	PLA3OUT	PLA3 单元输出状态	0: PLA3 单元当前逻辑电平为低 1: PLA3 单元当前逻辑电平为高	R											
b2	PLA2OUT	PLA2 单元输出状态	0: PLA2 单元当前逻辑电平为低 1: PLA2 单元当前逻辑电平为高	R											
b1	PLA1OUT	PLA1 单元输出状态	0: PLA1 单元当前逻辑电平为低 1: PLA1 单元当前逻辑电平为高	R											
b0	PLA0OUT	PLA0 单元输出状态	0: PLA0 单元当前逻辑电平为低 1: PLA0 单元当前逻辑电平为高	R											

### 32.3.3 多路选择器选择寄存器 (PLAx\_MUXS) (x=0...15)

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MUX0 [4]	MUX1 [4]														
-															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为 0	R/W											
b15	MUX0[4]	多路选择器 0 输入选择的第 4 位	通过这个位选择 MUX0 的输入 参考“13.4.1 输入多路选择器”表 13-1 PLAxMUX0 输入选择	R/W											
b14	MUX1[4]	多路选择器 1 输入选择的第 4 位	通过这个位选择 MUX1 的输入 参考“13.4.1 输入多路选择器”表 13-2 PLAxMUX1 输入选择	R/W											
b13~b8	Reserved	-	读出时为 0, 写入时写 0	R/W											
b7~b4	MUX0[3:0]	多路选择器 0 输入选择的第 3~0 位	通过这些位选择 MUX0 的输入 参考“13.4.1 输入多路选择器”表 13-1 PLAxMUX0 输入选择	R/W											
b3~b0	MUX1[3:0]	多路选择器 1 输入选择的第 3~0 位	通过这些位选择 MUX1 的输入 参考“13.4.1 输入多路选择器”表 13-2 PLAxMUX1 输入选择	R/W											

### 32.3.4 查找表控制寄存器 (PLAx\_LUTCTL) (x=0...15)

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
<hr/>															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
<hr/>										LUTCTL[7:0]					

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为 0,写入时写 0	R/W
b7~b0	LUTCTL[7:0]	LUT 控制位	这些位控制 input2、input1、input0 的哪个逻辑函数会对输出产生影响。 例如: IN1 IN0: LUTCTL=8'b11101110 IN2&(IN1^IN0):LUTCTL=8'b01100000	R/W

### 32.3.5 控制寄存器 (PLAx\_CTL) (x=0...15)

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
<hr/>															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
<hr/>										OSEL	OEN	-	-	-	-

位	标记	位名	功能	读写
b31~b7	Reserved	-	读出时为 0,写入时写 0	R/W
b7	OSEL	LUT 使能	0: LUT 的结果不作为 PLAxOUT 的输出 1: LUT 的结果作为 PLAxOUT 的输出	R/W
b6	OEN	输出使能	0: PLAxOUT 输出禁止 1: PLAxOUT 输出允许	R/W
b5~b0	Reserved	-	读出时为 0,写入时写 0	R/W

## 32.4 注意事项

在配置 PLA 寄存器时一定要 32 位读写，以确定该位寄存器设定与用户要求一致。

在用户更改 LUTCTL 值时需要经过仔细计算并配合 PLAx\_MUXS 的输入选择，避免出现 Combinal Loop 情况；在修改 PLA 寄存器时需要先将 PLA\_GCTL 寄存器清零，以防止出现 Combinal Loop 情况。

## 33 高精度参考电压 (VREF)

### 33.1 简介

VREF 模块支持 AVCC 分压输出、VREFINT 输出和 VREFH 输出。VREF 模块包含专用 BGR 电路 (Band Gap Reference)、CS 电路 (Current Source)。

AVCC 分压输出值有 AVCC\*1/2、AVCC\*2/5、AVCC\*1/3、AVCC\*1/4。可通过配置寄存器 AVCCMONS[1:0] 选择分压输出值。输出电压用作 ADC 模块自动诊断。

VREFINT 可同时输出 VREFINT(1.1V)及其分压 VREFINT\*3/4、VREFINT\*1/2、VREFINT\*1/4。输出电压用作比较器 CMP 的基准电压。

内部 VREF 电路输出内部参考电压 VREFH，作为 ADC/DAC 的参考电压。内部 VREF 电路关闭时，VREFH 可以通过管脚接通外部参考电压，当内部 VREF 电路关闭时，VREFH 也可以短接至 VSS 放电。

VREFH 引脚在某些封装上与 AVCC 引脚在一个 package 中进行了双重连接时，在这些封装中，内部参考电压缓冲器不可用，必须保持禁用。

### 33.2 功能描述

#### AVCC 分压电路

该模块输出 AVCC 的分压值，如下图所示。

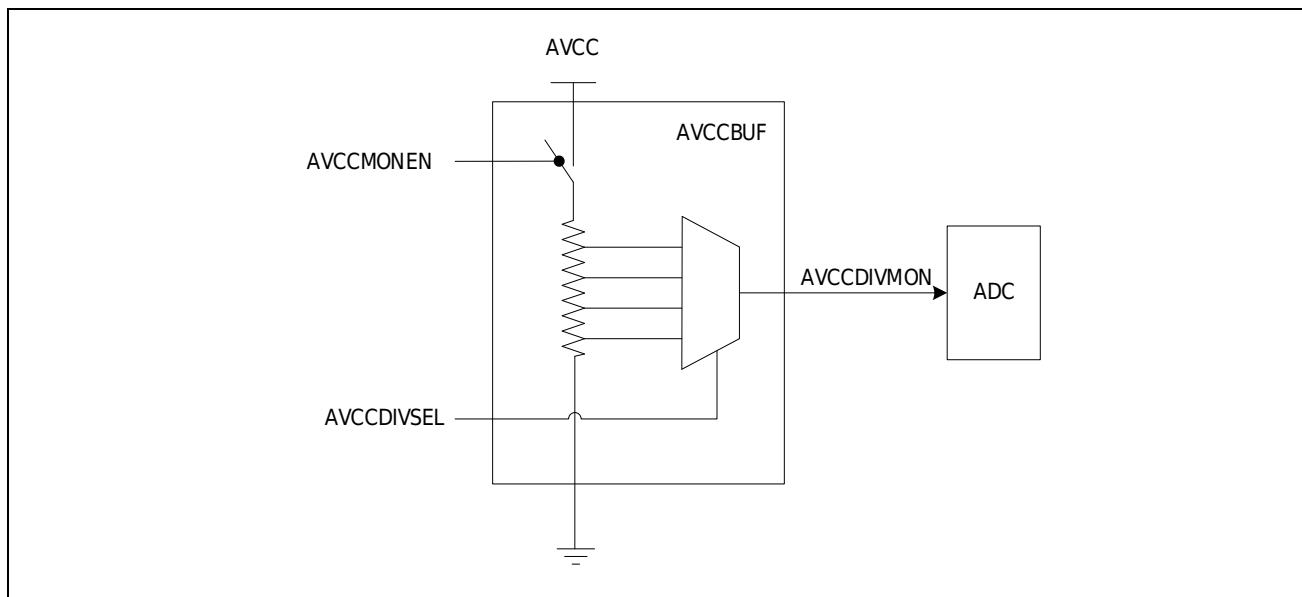


图 33-1 AVCC 分压电路

AVCCMON 输出值设置：

1. 设置寄存器 VREF\_CSCR.CSEN 为 1，使能 CS 电流源。
2. Delay 20us。
3. 设置寄存器 VREF\_CR.AVCCMONS[1:0]选择 AVCCMON 输出挡位。
4. 设置寄存器 VREF\_CR AVCCMONEN 为 1，使能 AVCC 分压电路。
5. 等待 20us 稳定时间后，输出电压用作 ADC 模块自动诊断。

如下表所示：

表 33-1 AVCCMON

AVCCMONS[1:0]	AVCCMON电压值
00	AVCC*1/4
01	AVCC*1/3
10	AVCC*2/5
11	AVCC*1/2

### VREFINT 电路

该模块用于输出 VREFINT 电压及其分压值，如图所示。

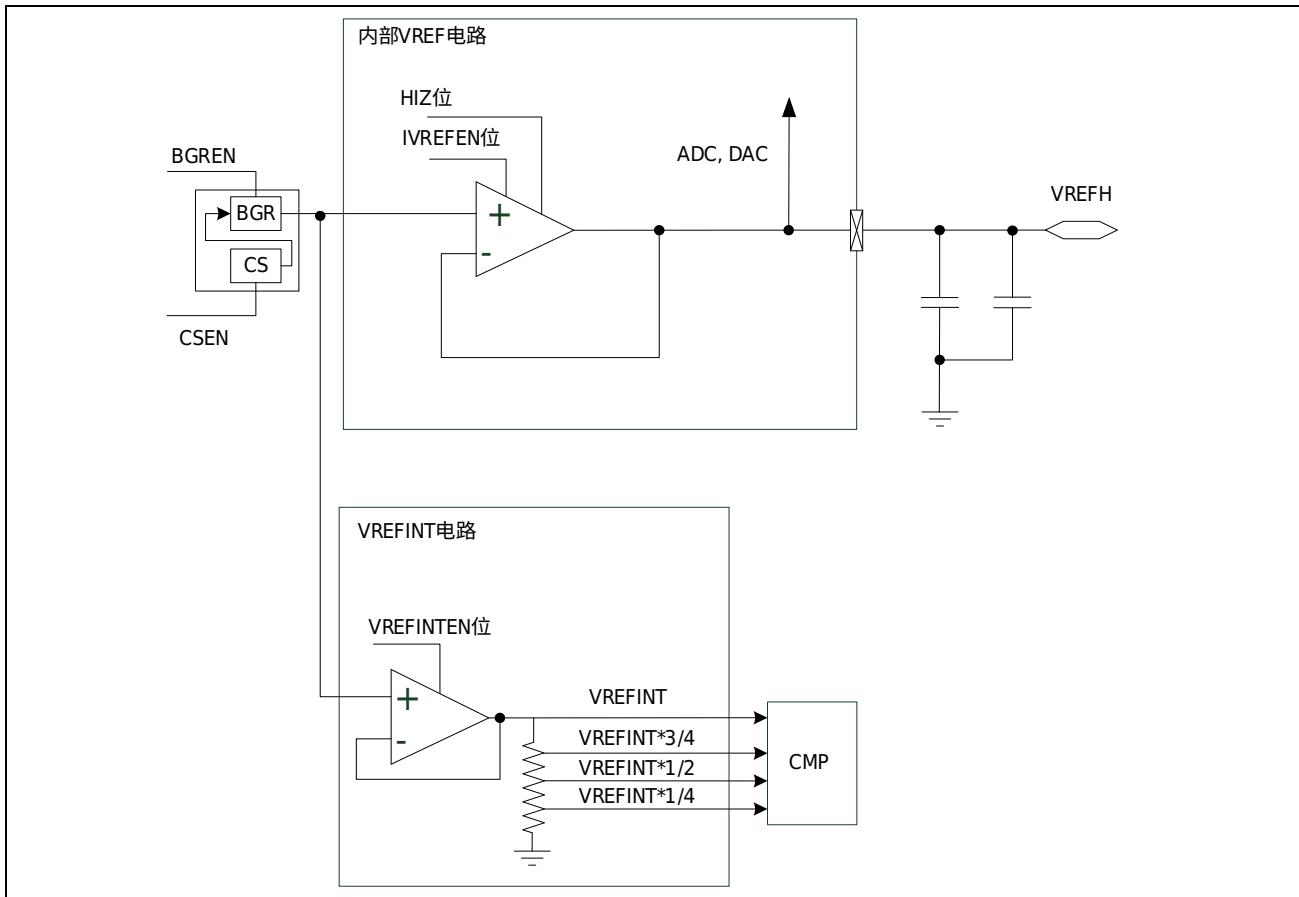


图 33-2 内部 VREF 电路与 VREFINT 电路原理图

VREFINT 输出值设置：

1. 设置 VREF\_CSCR.CSEN 为 1，使能 CS 电路。
2. Delay 20us。
3. 设置 VREF\_BGRCR.BGREN 为 1，使能 BGR 电路。
4. Delay 100us。
5. 设置 VREF\_CR.VREFINTEN 为 1，VREFINT 电路输出使能。
6. 等待 10us 稳定时间后，即可作为 CMP 的比较电压。

## 内部 VREF 电路

配置 IVREFEN 和 HIZ，可以使内部 VREF 电路有三种工作模式，如下表所示：

表 33-2 VREF 电路模式

HIZ	IVREFEN	模式
0	0	外部参考电压模式 内部VREF关闭，VREFH引脚外部输入
1	0	VREFH短接VSS放电
x	1	内部参考电压模式， 内部VREF使能，VREFH引脚连接内部VREF电路

VREF 电路有内部参考电压模式和外部参考电压模式。

### ■ 内部参考电压模式

1. 设置 VREF\_CSCR.CSEN 为 1，使能 CS 电路。
2. Delay 20us。
3. 设置 VREF\_BGRCR.BGREN 为 1，使能 BGR 电路。
4. Delay 100us。
5. 设置 VREF\_CR.IVREFEN 为 1，使能内部 VREF 电路。
6. 参照电气规格，等待稳定时间。内部 VREF 电路可以作为基准电压使用。

### ■ 外部参考电压模式

1. 设置 VREF\_CR.IVREFEN 为 0，关闭内部 VREF 电路。
2. 设置 VREF\_CR.HIZ 为 0，VREFH 引脚作为输入。
3. VREFH 管脚输入电压作为基准电压使用。

### ■ VREFH 短接 VSS 放电

1. 设置 VREF\_CR.IVREFEN 为 0，关闭内部 VREF 电路。
2. 设置 VREF\_CR.HIZ 为 1，VREFH 引脚短接至 VSS。

### 33.3 寄存器说明

表 33-3 寄存器一览表

BASE\_ADDR: 0x4004F800

寄存器名	符号	偏移地址	位宽	复位值
BGR控制寄存器	VREF_BGRCR	0x00	32	0x00000000
电流源控制寄存器	VREF_CSCR	0x04	32	0x00000000
VREF控制寄存器	VREF_CR	0x08	32	0x00000030

#### 33.3.1 BGR 控制寄存器 (VREF\_BGRCR)

偏移地址：0x00

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved							
b23	b22	b21	b20	b19	b18	b17	b16
Reserved							
b15	b14	b13	b12	b11	b10	b9	b8
Reserved							
b7	b6	b5	b4	b3	b2	b1	b0
Reserved	Bgren						

位	标记	位名	功能	读写
b31-1	Reserved	-	读出时为“0”，写入时写“0”	R/W
b0	Bgren	BGR电路控制位	0：关闭BGR电路 1：使能BGR电路，作为VREFINT及内部VREF电路基准。	R/W

### 33.3.2 电流源控制寄存器 (VREF\_CSCR)

偏移地址：0x04

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved							
b23	b22	b21	b20	b19	b18	b17	b16
Reserved							
b15	b14	b13	b12	b11	b10	b9	b8
Reserved							
b7	b6	b5	b4	b3	b2	b1	b0
Reserved	CSEN						
0	0	0	0	0	0	0	0

位	标记	位名	功能	读写
b31-1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	CSEN	CS电路控制位	0: 关闭CS电路 1: 使能CS电路, 输出电流至BGR模块。	R/W

### 33.3.3 VREF 控制寄存器 (VREF\_CR)

偏移地址：0x08

复位值：0x0000 0030

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b7	b6	b5	b4	b3	b2	b1	b0
Reserved	Reserved	AVCCMONS[1:0]		HIZ	IVREFEN	VREFINTEN	AVCCMONEN

位	标记	位名	功能	读写
b31-b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
			该位用于控制AVCC分压电路挡位选择 00: AVCCMON为AVCC*1/4 01: AVCCMON为AVCC*1/3 10: AVCCMON为AVCC*2/5 11: AVCCMON为AVCC*1/2	
b5-b4	AVCCMONS[1:0]	AVCC分压控制位		R/W
b3	HIZ	高阻抗模式	0: VREFH引脚高阻抗 1: VREFH引脚连接至内部VREF电路	
b2	IVREFEN	内部VREF电路使能	该位用于控制内部VREF电路开关 0: 外部参考电压模式, 内部VREF关闭 1: 内部参考电压模式, 内部VREF电路使能	R/W
b1	VREFINTEN	VREFINT电路使能	该位用于控制VREFINT电路开关 0: 关闭VREFINT电路 1: 使能VREFINT电路	R/W
b0	AVCCMONEN	AVCC分压电路使能	该位用于控制AVCC分压电路开关 0: 关闭AVCC分压电路 1: 使能AVCC分压电路	R/W

## 34 USB2.0 全速模块 (USBFS)

### 34.1 USBFS 简介

USB 全速 (USBFS) 控制器为便携式设备提供了一套 USB 通信解决方案。USBFS 控制器支持主机模式和设备模式，且芯片内部集成全速 PHY。主机模式下，USBFS 控制器支持全速 (FS, 12Mb/s) 和低速 (LS, 1.5Mb/s) 收发器，而设备模式下则仅支持全速 (FS, 12Mb/s) 收发器。USBFS 控制器支持 USB 2.0 协议所定义的所有四种传输方式 (控制传输、批量传输、中断传输和同步传输)。遵循协议如下：

- Universal Serial Bus Revision 2.0 Specification
- USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007
- Errata for USB 2.0 ECN: Link Power Management (LPM) - 7/2007

### 34.2 USBFS 主要特性

主要分为三类：通用特性、主机模式特性和设备模式特性。

#### 34.2.1 通用特性

- 内置片上 USB2.0 全速 PHY
- 支持主机模式和设备模式
- 支持 FS SOF 以及低速“Keep-alive”令牌并具有以下功能：
  - SOF 脉冲管脚输出功能
  - SOF 脉冲可作为芯片内部事件源去触发 TIMER, DMA 等模块工作
  - 可配置帧周期
  - 可配置的帧结束中断
- 模块内嵌 DMA，并可软件配置 AHB 突发传输类型
- 具备省电功能，例如 USB 挂起，停止 RAM 时钟，停止 PHY 域时钟
- 具有采用高级 FIFO 控制的 1.25KB 专用 RAM
  - 可以将 RAM 空间划分为不同的 FIFO，以便灵活有效的使用 RAM
  - 每个 FIFO 可存储多个数据包
  - 动态分配存储区
  - FIFO 的大小可配置成为 2 的幂次方值，以便连续使用存储单元
- 一帧之内可以不需要应用程序干预，以达到最大 USB 带宽
- 可根据 ID 线的电平自动确定主机模式或者设备模式

### 34.2.2 主机模式特性

- 主机模式支持 USB2.0 全速 (FS, 12Mb/s) 和低速 (LS, 1.5Mb/s) 传输
- 需要通过外部电源芯片生成 VBUS 电压
- 多达 12 主机通道 (管道): 每个通道都可以动态实现重新配置, 可支持任何类型的 USB 传输
- 内置硬件调度器, 可:
  - 在周期性硬件队列中存储多达 8 个中断加同步传输请求
  - 在非周期性硬件队列中存储多达 8 个控制加批量传输请求
- 管理一个共享 RX FIFO、一个周期性 TX FIFO 和一个非周期性 TX FIFO, 以有效使用 USB 数据 RAM

### 34.2.3 设备模式特性

- 从机模式支持 USB2.0 全速 (FS, 12Mb/s) 传输。
- 1 个双向控制端点 0
- 5 个 OUT 端点, 可以配置为支持批量传输、中断传输或同步传输
- 5 个 IN 端点, 可以配置为支持批量传输、中断传输或同步传输
- 包含 6 个发送 FIFOs (每个 IN 端点配备一个发送 FIFO) 和一个接收 FIFO (由所有的 OUT 端点共享)
- 支持远程唤醒功能。
- 支持软断开功能
- VBUS PIN 支持 5V 耐压。

### 34.3 USBFS 系统框图

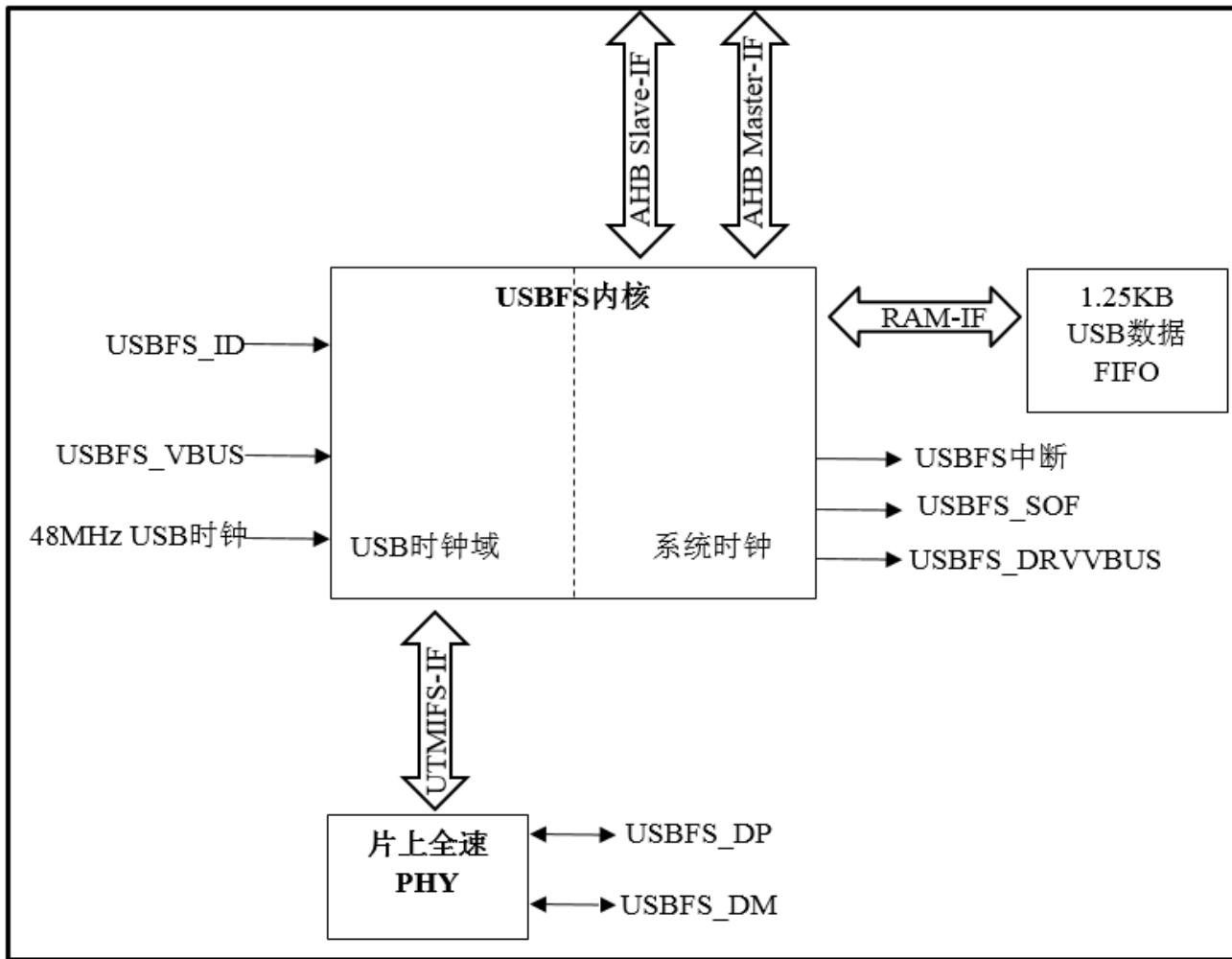


图 34-1 USBFS 系统框图

### 34.4 USBFS 管脚说明

表 34-1 USBFS 管脚说明

管脚名	方向	适用模式	对应管脚	功能描述
USBFS_VBUS	输入	设备	PA9	电源端口, 5V耐压
USBFS_DP	输入/输出	主机、设备	PA12	差分数据D+信号
USBFS_DM	输入/输出	主机、设备	PA11	差分数据D-信号
USBFS_DRVVBUS	输出	主机	PB8	外部电源芯片使能信号
USBFS_ID	输入	主机	PA10	USB A-B器件识别信号
USBFS_SOF	输出	主机、设备	PA8	SOF输出脉冲信号

由于 USBFS\_DP 和 USBFS\_DM 管脚和通用 GPIO 复用，在使用 USBFS 时，建议关闭其对应管脚的数字功能，USBFS\_DP 和 USBFS\_DM 功能为模拟功能，与对应的 PFSR 寄存器设定无关。具体请参考【通用 IO (GPIO)】章节。另外 USBFS 功能不使用时，USBFS\_DP 和 USBFS\_DM 管脚对应的数字功能管脚翻转时，会产生额外的电流消耗。

## 34.5 USBFS 功能说明

### 34.5.1 USBFS 时钟以及工作模式

USBFS 所使用的时钟需要配置为 48MHz，该 48MHz 时钟由内部 PLL 电路产生，PLL 时钟源需选择外部高速振荡器，使用 USBFS 模块前，需要在 CMU 模块内配置好 USBFS 时钟。

USBFS 可以作为主机或者设备使用，并且包含一个片上全速 PHY。

上拉和下拉电阻已经集成在片上全速 PHY 的内部，并且 USBFS 可以根据当前模式和连接状态自动选择。

USBFS 工作时，VCC 电压范围为 3.0~3.6V。

### 34.5.2 USBFS 模式决定

USBFS 决定当前工作模式有以下两种方法：

方法 1：根据 USBFS\_ID 线的状态自动识别，检测到 USBFS\_ID 线为高电平时，模块工作在设备模式，检测到 USBFS\_ID 线为低电平时，模块工作在主机状态。

方法 2：强制主机/设备模式，通过设定寄存器 USBFS\_GUSBCFG 的 FDMOD 或者 FHMOD 位为 1，使模块忽略 USBFS\_ID 线的电平而强制工作在设备或者主机模式。

### 34.5.3 USBFS 主机功能

#### 34.5.3.1 主机功能简介

当 USBFS 工作在主机模式下时，VBUS 为 USB 协议所规定的 5V 电源引脚。内部 PHY 不支持提供 5V 电源，所以需要外接 USB 电源芯片为设备供电。USBFS\_DRVVBUS 用来使能外接 USB 电源芯片，外接电源芯片的过电流检出可以通过本 MCU 的外部中断 IRQ 实现。主机模式下 USBFS\_VBUS 可以作为 GPIO 使用。

典型的 USB 主机模式系统构建图如下：

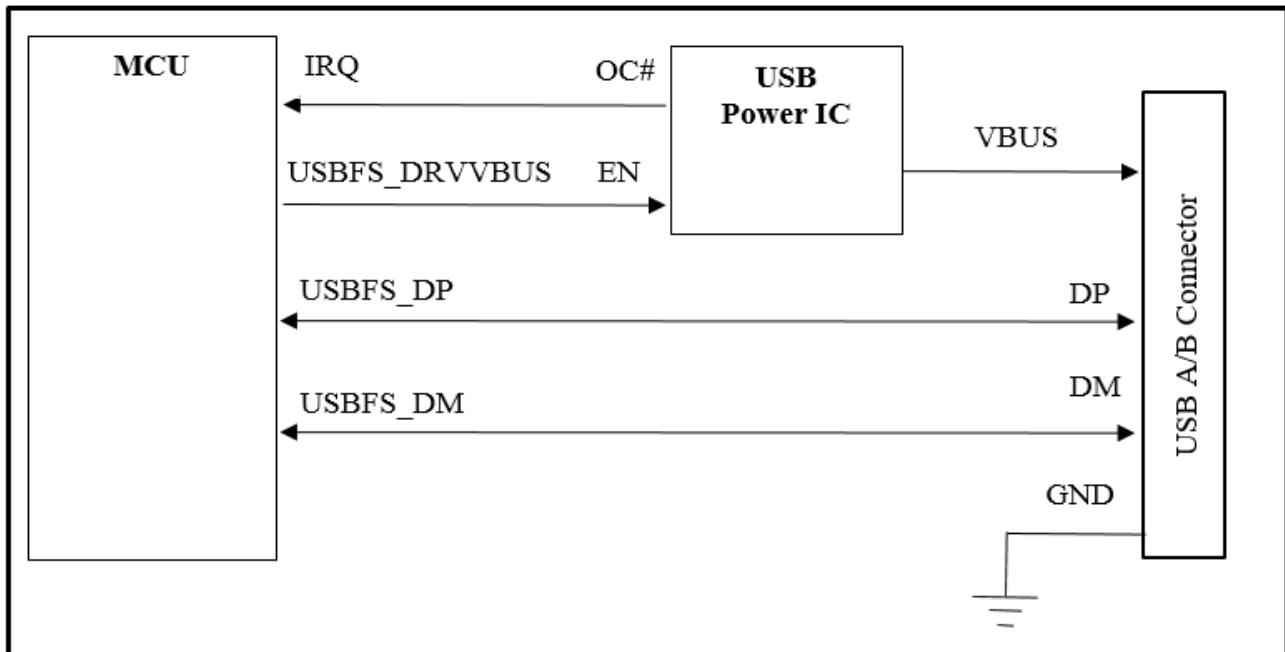


图 34-2 USBFS 主机模式系统构建图

#### 34.5.3.2 主机端口供电

本 MCU 不能输出 5V 以提供 VBUS。为此，必须在微控制器以外添加 USB 电源芯片或基本电源开关（如应用电路板提供 5V 电源）来驱动 5V VBUS 线。外部电源芯片可通过任何 GPIO 输出或者 USBFS\_DRVVBUS 驱动。当应用程序确定使用 GPIO 来控制外部器件提供 VBUS，仍须将主机端口控制和状态寄存器中的端口电源位（USBFS\_HPRT 中的 PPWR 位）置 1。

#### 34.5.3.3 主机检测设备连接和断开

USB 设备将在连接后立即被检测到。USBFS 模块将发出主机端口中断信号，该中断由主机端口控制和状态寄存器中的设备连接位（USBFS\_HPRT 中的 PCDET 位）触发。

设备断开事件将触发断开连接检测中断（USBFS\_GINTSTS 中的 DISCINT 位）。

#### 34.5.3.4 主机枚举

检测到设备连接后，若又有新的设备连接进来，主机必须通过向新的设备发送 USB 复位和配置命令来启动枚举过程。

应用程序通过将主机端口控制和状态寄存器中的端口复位位（USBFS\_HPRT 中的 PRST 位）置 1，使该过程最少持续 10ms、最多持续 20 ms，以此通过 USB 驱动 USB 复位信号（单端零）。应用程序计算这个过程的持续时间，然后将端口复位位清零。

USB 复位序列完成后，端口使能/禁止更改位（USBFS\_HPRT 中的 PENCHNG 位）立即触发主机端口中断，进而向应用程序发出通知，指示可从主机端口控制和状态寄存器中的端口速度字段（USBFS\_HPRT

中的 PSPD) 读取枚举的设备速度, 以及主机已经开始驱动 SOF (FS) 或 Keep-alive 令牌 (LS)。此时主机已就绪, 可通过对设备发送命令来完成对设备的枚举。

### 34.5.3.5 主机挂起

应用程序通过将主机端口控制和状态寄存器中的端口挂起位 (USBFS\_HPRT 中的 PSUSP) 置 1 来挂起 USB 总线。USBFS 模块停止发送 SOF 并进入挂起状态。

可由远程设备的自主活动 (远程唤醒) 使总线退出挂起状态。这种情况下, 远程唤醒信号将触发远程唤醒中断 (USBFS\_GINTSTS 中的 WKUPINT 位), 硬件把主机端口控制和状态寄存器中的端口恢复位 (USBFS\_HPRT 中的 PRES 位) 自行复位, 并通过 USB 自动驱动恢复信号。应用程序必须为恢复窗口定时, 然后将端口恢复位清零以退出挂起状态并重新启动 SOF。

如果由主机发起退出挂起状态, 则应用程序必须将端口恢复位置 1 以启动主机端口上的恢复信号, 为恢复窗口定时并最终将端口恢复位清零。

### 34.5.3.6 主机通道

USBFS 模块实现了 12 个主机通道。每个主机通道均可用于 USB 主机传输 (USB 管道)。主机最多能同时处理 8 个传输请求。如果应用程序有 8 个以上的传输请求挂起, 则在通道从之前任务释放后 (即, 接收到传输完成和通道停止中断后), 主机控制器驱动器 (HCD) 必须为未处理的传输请求重新对通道进行分配。

每个主机通道都可配置为支持输入/输出以及周期性/非周期性事务。每个主机通道都使用专用控制 (HCCHARx) 寄存器、传输配置 (HCTSIZx) 寄存器/中断 (HCINTx) 寄存器以及和其相关的中断屏蔽寄存器 (HCINTMSKx)。

#### 主机通道控制

应用程序可通过主机通道 x 特性寄存器 (HCCHARx) 对主机通道作以下控制:

- 通道使能/禁止
- 设置目标 USB 设备的速度: FS/LS
- 设置目标 USB 设备的地址
- 设置与该通道通信的目标 USB 设备上的端点的编号
- 设置该通道上的传输方向: IN/OUT
- 设置该通道上的 USB 传输的类型: 控制/批量/中断/同步
- 设置与该通道通信的设备端点的最大包长
- 设置要进行周期传输的帧: 奇帧/偶帧

## 主机通道传输

主机通道传输大小寄存器(HCTSIZx)允许应用程序对传输大小参数进行编程并读取传输状态。必须在主机通道特性寄存器中的通道使能位置 1 之前完成对此寄存器的设置。使能端点后，数据包计数字段立即变为只读状态，同时 USBFS 模块根据当前传输状态对该字段进行更新。

可对以下传输参数进行编程：

- 以字节为单位的传输大小
- 构成整个传输大小的数据包个数
- 初始数据 PID

## 主机通道状态/中断

主机通道 x 中断寄存器 (HCINTx) 指示端点在出现 USB 和 AHB 相关事件时的状态。当中断寄存器中的主机通道中断位 (USBFS\_GINTSTS 中的 HCINT 位) 置 1 时，应用程序必须读取这些寄存器以获得详细信息。在读取这些寄存器之前，应用程序必须先读取主机全体通道中断 (HCAINT) 寄存器，以获取主机通道 x 中断寄存器的通道编号。应用程序必须将此寄存器中的相应位清零，才能将 HAINT 和 GINTSTS 寄存器中的相应位清零。 USBFS\_HCINTMSK x 寄存器还提供每个通道各中断源的屏蔽位。

主机模块提供以下状态检查和中断产生功能：

- 传输完成中断，指示应用程序 (AHB) 和 USB 端均已完成数据传输
- 通道因传输完成、USB 事务错误或应用程序发出禁止命令而停止
- 相关的发送 FIFO 为半空或全空状态 (IN 端点)
- 接收到 ACK 响应
- 接收到 NAK 响应
- 接收到 STALL 响应
- 由于 CRC 校验失败、超时、位填充错误和错误的 EOP 导致 USB 事务错误
- 串扰错误
- 帧上溢
- 用于数据同步的翻转位出错

### 34.5.3.7 主机调度器

主机模块内置硬件调度器，可自主对应用程序发出的 USB 事务请求重新排序和管理。每一帧开始时，主机都先执行周期性（同步和中断）事务，然后执行非周期性（控制和批量）事务，以符合 USB 规范对同步和中断传输高优先级的保证。

主机通过请求队列（一个周期性请求队列和一个非周期请求队列）处理 USB 事务。每个请求队列最多可存储 8 个条目。每个条目代表一个应用程序发起但还未得到响应的 USB 事务请求，并存储了执行该 USB 事务所用到的 IN 或 OUT 通道的编号，以及其它相关信息。USB 事务请求在队列中的写入顺序决定了事务在 USB 接口上的执行顺序。

每一帧开始时，主机都先处理周期性请求队列，然后处理非周期性请求队列。如果当前帧结束时，计划在当前帧执行的同步或中断类型的 USB 传输事务请求仍处于挂起状态，则主机将发出未完成周期性传输中断（USBFS\_GINTSTS 中的 IPXFR 位）。USBFS 模块负责对周期性和非周期性请求队列的管理。周期性发送 FIFO 和队列状态寄存器 (HPTXSTS) 与非周期性发送 FIFO 和队列状态寄存器 (HNPTXSTS) 都为只读寄存器，应用程序可使用它们 来读取各请求队列的状态。其中包括：

- 周期性（非周期性）请求队列中当前可用的空闲条目数（最多 8 个）
- 周期性（非周期性）TxFIFO（OUT 事务）中当前可用的空闲空间
- IN/OUT 令牌、主机通道编号和其它状态信息

由于每个请求队列最多可存储 8 个 USB 事务请求，因此应用程序可以把主机 USB 事务请求提前发送给调度器；实际的通信最晚会在调度器处理完已挂起的 8 个周期事务和 8 个非周期事务完成之后出现在 USB 总线上。

要向主机调度器（队列）发出事务请求，应用程序必须读取 USBFS\_HPTXSTS 寄存器中的 PTXQSAV 位或 USBFS\_HNPTXSTS 寄存器中的 NPTQXSAV 位，确保周期性（非周期性）请求队列中至少有一个可用空间来存储当前请求。

### 34.5.4 USBFS 设备功能

#### 34.5.4.1 设备功能简介

当 USBFS 工作在设备模式下时，VBUS 为 USB 协议所规定的 5V 电源引脚，为 5V 耐压管脚。本模块始终检测 VBUS 线的电平状态来连接或者断开设备。USBFS\_VBUS 端口需先经过 100 欧姆以上的电阻才能接到 VBUS 电源。

典型的 USB 设备模式系统构建图如下：

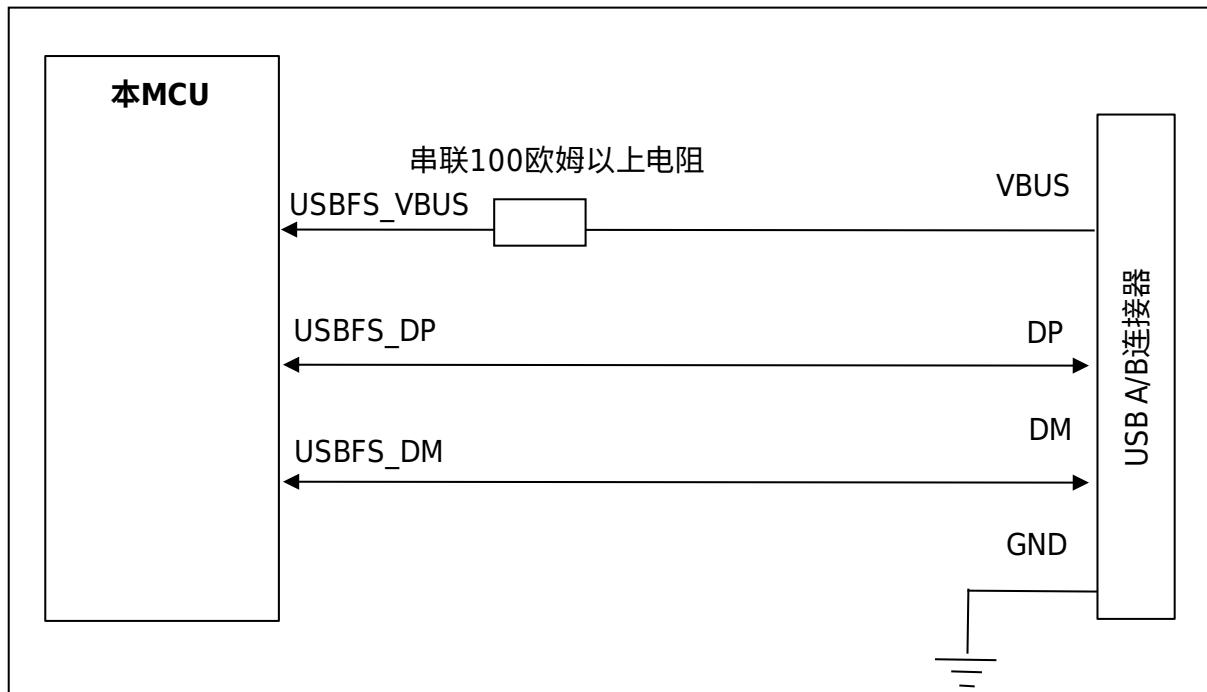


图 34-3 USBFS 设备模式系统构建图

#### 34.5.4.2 设备供电状态

模块检测到 USBFS\_VBUS 为高电平时，就会使 USB 设备进入供电状态。然后，USBFS 自动连接 DP 上拉电阻，发出全速设备与主机相连的信号并生成会话请求中断 (USBFS\_GINTSTS 中的 VBUSVINT 位)，指示进入供电状态。

此外，USBFS\_VBUS 输入还可确保主机在 USB 操作期间提供有效的 VBUS 电平。如果检测到 VBUS 电平为低电平（例如，因电源干扰或主机端口关闭引发），USBFS 将自动断开连接。

供电状态下，USBFS 期望收到来自主机的复位信号。其它 USB 操作则无法执行。收到复位信号后，立即生成检测到复位中断 (USBFS\_GINTST 中的 USBRST)。复位信号结束后，将生成枚举完成中断 (USBFS\_GINTSTS 中的 ENUMDNE 位)，USBFS 随即进入默认状态。

#### 34.5.4.3 设备默认状态

默认状态下，USBFS 期望从主机收到 SET\_ADDRESS 命令。其它 USB 操作则无法执行。当 USB 上解码出有效 SET\_ADDRESS 命令时，应用程序会将相应的地址值写入设备配置寄存器中的设备地址字段（USBFS\_DCFG 中的 DAD 位）。USBF 随即进入地址状态，并准备好以所配置的 USB 地址对主机事务进行应答。

#### 34.5.4.4 设备挂起状态

USBFS 设备持续监视 USB 活动。在 USB 空闲时间达到 3ms 后，将发出早期挂起中断（USBFS\_GINTSTS 中 ESUSP 位），并在 3ms 后由挂起中断（USBFS\_GINTSTS 中的 USBSUSP 位）确认设备进入挂起状态。然后，设备状态寄存器中的设备挂起位（USBFS\_DSTS 中的 SUSPSTS 位）自动置 1，USBFS 随即进入挂起状态。

可通过设备本身退出挂起状态。这种情况下，应用程序会将设备控制寄存器中的远程唤醒信号位（USBFS\_DCTL 中的 RWUSIG 位）置 1，并在 1ms 到 15ms 内将其清零。

但若设备检测到主机发出的恢复信号，将生成恢复中断（USBFS\_GINTSTS 中的 WKUPINT 位），设备挂起位自动清零。

#### 34.5.4.5 设备软断开

供电状态可借助软断开功能通过软件退出。将设备控制寄存器中的软断开位（USBFS\_DCTL 中的 SDIS 位）置 1 即可移除 DP 上拉电阻，此时尽管没有从主机端口实际拔出 USB 电缆，但主机端仍会发生设备断开检测中断。

#### 34.5.4.6 设备端点

##### 端点类别

USBFS 模块实现了以下 USB 端点：

- 控制端点 0：
  - 双向且仅处理控制消息
  - 使用一组单独的存器来处理 IN 和 OUT 事务
  - 专用控制（USBFS\_DIEPCTL0/ USBFS\_DOEPCTL0）寄存器、传输配置（USBFS\_DIEPTSIZ0/ USBFS\_DIEPTSIZ0）寄存器和状态中断（USBFS\_DIEPINTx/ USBFS\_DOEPINT0）寄存器。控制和传输大小寄存器中可用的位组与其它端点中稍有不同
- 5 个 IN 端点
  - 每个端点都可配置为支持同步传输、批量传输或中断传输类型
  - 每个端点都有专用控制（USBFS\_DIEPCTLx）寄存器、传输配置（USBFS\_DIEPTSIZx）寄存器和状态中断（USBFS\_DIEPINTx）寄存器

- 设备 IN 端点通用中断屏蔽寄存器 (USBFS\_DIEPMSK) 可用于使能/禁止所有 IN 端点(包括 EP0)上的同一类端点中断源
- 支持未完成的同步 IN 传输中断 (USBFS\_GINTSTS 中的 IISOIXFR 位)，该中断将在当前帧中至少有一个同步 IN 端点上的传输未完成时触发。该中断和周期性帧中断 (USBFS\_GINTSTS/EOPF) 一起触发
- 5 个 OUT 端点
  - 每个端点都可配置为支持同步传输、批量传输或中断传输类型
  - 每个端点都有专用控制 (USBFS\_DOEPCTLx) 寄存器、传输配置(USBFS\_DOEPTSIZx) 寄存器和状态中断 (USBFS\_DOEPINTx) 寄存器
  - 设备 OUT 端点通用中断屏蔽寄存器 (USBFS\_DOEPMASK) 可用于使能/禁止所有 OUT 端点 (包括 EP0) 上的同一类端点中断源
  - 支持未完成的同步 OUT 传输中断 (USBFS\_GINTSTS 中的 INCOMPISOOUT 位)，该中断将在当前帧中至少有一个同步 OUT 端点上的传输未完成时触发。该中断和周期性帧中断 (USBFS\_GINTSTS/EOPF) 一起触发

## 端点控制

应用程序可通过设备端点 x IN/OUT 控制寄存器 (DIEPCTLx/DOEPCTLx) 对端点采取以下控制：

- 端点使能/禁止
- 在当前配置下激活端点
- 设置 USB 传输类型 (同步、批量和中断)
- 设置支持的数据包大小
- 设置与 IN 端点相关的 Tx-FIFO 编号
- 设置希望收到的或发送时要使用到的 data0/data1 PID (仅限批量/中断传输)
- 设置接收或发送事务时所对应的奇/偶帧 (仅限同步传输)
- 可以设置 NAK 位，从而不论此时 FIFO 的状态如何，都对主机的请求回复 NAK
- 可以设置 STALL 位，使得主机对该端点的令牌都被硬件回复 STALL
- 可以将 OUT 端点设置为侦听模式，即对接收到的数据不进行 CRC 检查

## 端点传输

设备端点 x 传输尺寸寄存器 (DIEPTSIZx/DOEPTSIZx) 允许应用程序对传输尺寸参数进行编程并读取传输状态。必须在端点控制寄存器中的端点使能位置 1 之前完成对此寄存器的设置。使能端点后，这些字段立即变为只读状态，同时 USBFS 模块根据当前传输状态对这些字段进行更新。

可对以下传输参数进行编程：

- 以字节为单位的传输大小
- 构成整个传输的数据包个数

## 端点状态/状态

设备端点 x 中断寄存器 (DIEPINTx/DOEPINTx) 指示端点在出现 USB 和 AHB 相关事件时的状态。当模块中断寄存器中的 OUT 端点中断位或 IN 端点中断位（分别为 USBFS\_GINTSTS 中 OEPINT 位或 USBFS\_GINTSTS 中的 IEPINT 位）置 1 时，应用程序必须读取这些寄存器以获得详细信息。在应用程序读取这些寄存器之前，必须先读取设备全体端点中断(USBFS\_DAINT) 寄存器，以获取设备端点 x 中断寄存器的端点编号。应用程序必须将此寄存器中的相应位清零，才能将 DAINT 和 GINTSTS 寄存器中的相应位清零。

模块提供以下状态检查和中断产生功能：

- 传输完成中断，指示应用程序 AHB 和 USB 端均已完成数据传输
- Setup 阶段已完成（仅针对控制传输类型的 OUT 端点）
- 相关的发送 FIFO 为半空或全空状态（IN 端点）
- NAK 应答已发送到主机（仅针对同步传输的 IN 端点）
- TxFIFO 为空时接收到 IN 令牌（仅针对批量和中断传输类型的 IN 端点）
- 尚未使能端点时接收到 OUT 令牌
- 检测到 babble 错误
- 应用程序关闭端点生效
- 应用程序对端点设置 NAK 生效（仅针对同步传输类型的 IN 端点）
- 接收到 3 个以上连续 setup 数据包（仅针对控制类型的 OUT 端点）
- 检测到超时状况（仅针对控制传输类型的 IN 端点）

### 34.5.5 USBFS SOF 脉冲管脚输出功能

USBFS 在主机和设备模式下都可以监视、跟踪和配置 SOF 帧并且还具备 SOF 脉冲输出功能。SOF 脉冲通过 USBFS\_SOF 管脚输出，输出宽度为 16 个系统时钟周期。

#### 34.5.5.1 主机 SOF

主机模式下，可以在主机帧间隔寄存器 (HFIR) 中对所产生的两个连续 SOF (FS) 或 keep-alive (LS) 令牌期间所出现的 PHY 时钟数进行编程，进而应用程序可对 SOF 帧周期进行控制。帧开始 (USBFS\_GINTSTS 中的 SOF 位) 时都将生成中断。当前帧编号和出现下一个 SOF 前剩余的时间应用程序在主机帧编号寄存器 (HFNUM) 中能够进行跟踪。

使用 USBFS 系统控制寄存器 USBFS\_SYCTLREG 中的 SOFEN 位，可以使任何 SOF 令牌发出的同时产生的、宽度为 16 个系统时钟周期的 SOF 脉冲信号从 USBFS\_SOF 引脚输出。

此外，SOF 脉冲还可以作为内部事件触发 DMA 传输、TIMER 计数等外部模块工作。

### 34.5.5.2 设备 SOF

在设备模式下，USB 每次接收到 SOF 令牌时，都将触发帧开始中断（USBFS\_GINTSTS 中的 SOF 位）。相应的帧编号可从设备状态寄存器（USBFS\_DSTS 中的 FNSOF 位）读取。使用 USBFS 系统控制寄存器 USBFS\_SYCTLREG 中的 SOFEN 位，还可以生成宽度为 16 个系统时钟周期的 SOF 脉冲信号，并使该信号在 USBFS\_SOF 引脚输出，以实现外部可用。

此外，SOF 脉冲还可以作为内部事件触发 DMA 传输、TIMER 计数等外部模块工作。

周期性帧结束中断(GINTSTS/EOPF)用于在经过了 80%、85%、90% 或 95% 的帧间隔时间时通知应用程序，具体取决于设备配置寄存器中的周期性帧间隔字段（USBFS\_DCFG 中的 PFIVL 位）。此功能可用于确定该帧的所有同步通信是否完成。

### 34.5.6 USBFS 功耗控制

不使用 USBFS 模块时，可以通过 CMU 模块停止 USBFS 模块的 HCLK 和 PHY 时钟，从而降低功耗。

使用 USB 模块，但设备 USB 会话未开始或设备未连接时，可以在 USB 挂起状态下使用功率降低技术。

- 停止 PHY 时钟（USBFS\_GCCTL 中的 STPPCLK 位）

将时钟门控控制寄存器中的停止 PHY 时钟位置 1 时，USBFS 全速模块的大多数 48 MHz 内部时钟域均由时钟门控关闭。即使应用程序仍提供时钟输入，也会节省掉模块由于时钟信号翻转带来的动态功耗还会关掉收发器的大部分单元，只有负责检测异步恢复事件或远程唤醒事件的部分还保持工作状态。

- HCLK 门控（USBFS\_GCCTL 中的 GATEHCLK 位）

将时钟门控控制寄存器中的 GATEHCLK 位置 1 时，USBFS 模块内部的大多数系统时钟域均由时钟门控关闭。只有寄存器读取和写入接口保持活动状态。即使应用程序仍提供时钟输入，也会节省掉模块由于时钟信号翻转带来的动态功耗。

为了节省动态功耗，只在 USB 数据 FIFO 被 USBFS 模块访问时为其提供时钟。

### 34.5.7 USBFS 动态更新 USBFS\_HFIR 寄存器

主机模式下，USB 模块具有对帧周期进行动态微调的功能，能够将外部设备与 SOF 帧进行同步。如果 USBFS\_HFIR 寄存器在当前 SOF 帧内发生更改，则将在下一个帧中对 SOF 周期进行相应修正，具体说明请参见图 34-4。

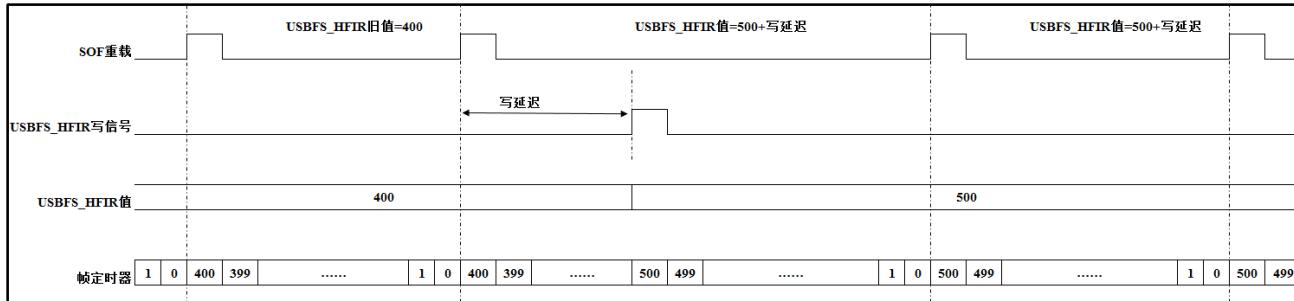


图 34-4 USBFS 动态更新 USBFS\_HFIR 寄存器示意图

### 34.5.8 USBFS 数据 FIFO

USBFS 系统具有 2.5KB 专用 RAM，采用高效的 FIFO 控制机制。USBFS 模块中的数据包 FIFO 控制器模块将 RAM 空间划分为多个 TxFIFO (USB 传输前，应用程序将数据压入其中进行短暂存储) 和单个 Rx FIFO (从 USB 接收到的数据被应用程序读取之前，在其中进行短暂存储)。

RAM 中所构建的 FIFO 的数量与组织方式取决于设备的角色。设备模式下，为每个激活的 IN 端点配置一个 Tx FIFO。FIFO 的大小均由软件配置，以更好地满足应用要求。

### 34.5.9 USBFS 主机 FIFO 架构

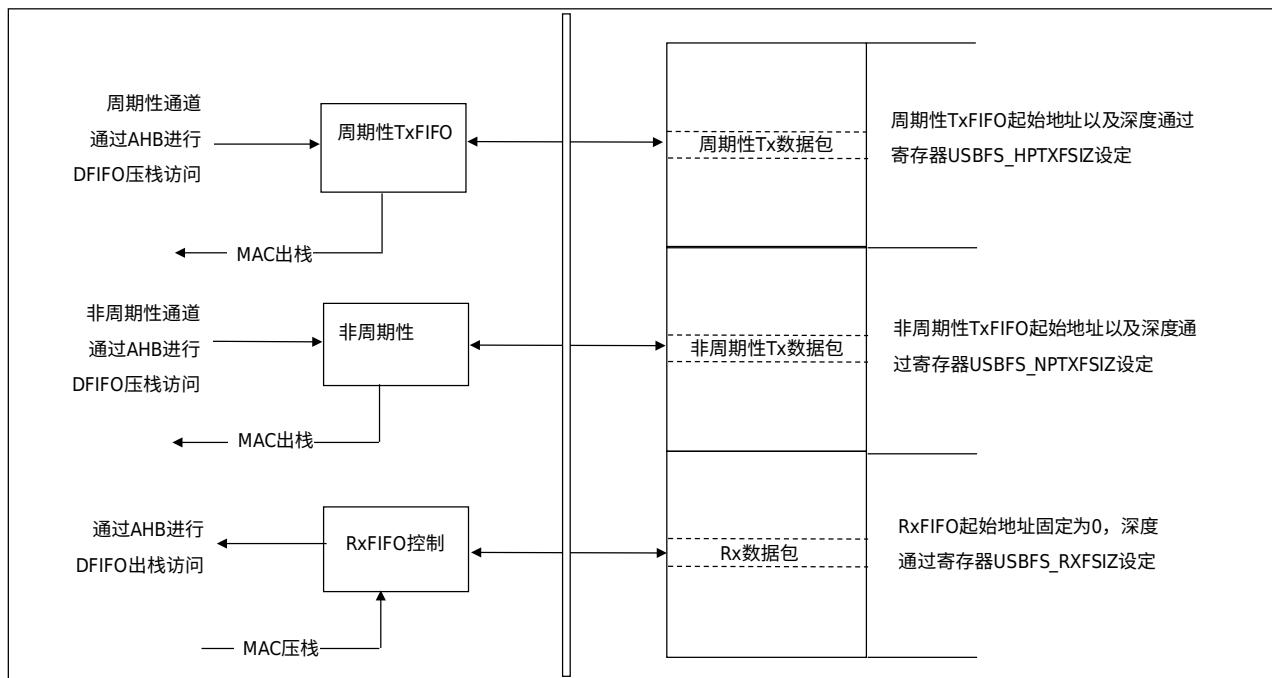


图 34-5 USBFS 主机模式下 FIFO 架构示意图

### 34.5.9.1 主机 RxFIFO

主机使用一个接收 FIFO 处理所有周期和非周期事务。FIFO 用作接收缓冲区以保存从 USB 接收到的数据（接收到的数据包的数据部分），直至这些数据传输到系统存储器。只要 FIFO 中有空间，来自设备 IN 端点的数据包就接收进来并挨个存储。接收到的每个数据包的状态（包含主机目标通道、字节数、数据 PID 和对所接收数据的校验）也存储在 FIFO 中。接收 FIFO 的大小在接收 FIFO 大小寄存器 (GRXFSIZ) 中配置。

单个接收 FIFO 架构使得 USB 主机高效地填充接收数据缓冲区：

- 所有 IN 配置主机通道共享同一个 RAM 缓冲区（共享 FIFO）
  - 对于主机软件驱动的任意序列 IN 令牌，USBFS 模块可将接收 FIFO 填充至限值
- 只要至少有一个数据包在 RxFIFO 中可供读取，应用程序就会接收 Rx FIFO 非空中断。应用程序从接收状态读取和出栈寄存器中读取数据包信息，最后从 RxFIFO 中读出数据。

### 34.5.9.2 主机 TxFIFO

主机使用一个发送 FIFO 处理所有非周期（控制和批量）OUT 事务，使用另一个发送 FIFO 处理所有周期（同步和中断 OUT 事务。FIFO 用作发送缓冲区以保存要通过 USB 发送的数据（发送数据包）。周期（非周期）TxFIFO 的大小在主机周期（非周期）发送 FIFO 大小 (HPTXFSIZ/HNPTXFSIZ) 寄存器中配置。

两个 Tx FIFO 按优先级实施操作，周期性通信的优先级较高，因此在 USB 一帧的时间内首先进行周期性通信。帧起始时，内置的主机调度器先处理周期请求队列，再处理非周期请求队列。

两个发送 FIFO 的架构使得 USB 主机能够对周期和非周期发送数据缓冲区分别进行优化管理：

- 配置为支持周期（非周期）OUT 事务的所有主机通道共享同一个 RAM 缓冲区（共享 FIFO）
- 对于主机软件驱动的任意序列 OUT 令牌，USBFS 模块可将周期性（非周期性）发送 FIFO 填充至限值

只要周期性 TxFIFO 为半空或全空，USBFS 模块就会发出周期性 TxFIFO 空中断 (USBFS\_GINTSTS 中的 PTXFE 位)，具体取决于 AHB 配置寄存器中的周期性 Tx-FIFO 空等级位 (USBFS\_GAHBCFG 中的 PTXFELVL 位) 的值。只要周期性 TxFIFO 和周期性请求队列中均存在空闲空间，应用程序便可提前写入发送数据。可通过读取主机周期性发送 FIFO 和队列状态寄存器 (HPTXSTS) 来了解二者的可用空间。

只要非周期性 TxFIFO 为半空或全空，USBFS 模块就会发出非周期性 TxFIFO 空中断 (USBFS\_GINTSTS 中的 NPTXFE 位)，具体取决于 AHB 配置寄存器中的非周期性 Tx-FIFO 空等级位 (USBFS\_GAHBCFG 中的 TXFELVL 位)。只要非周期性 TxFIFO 和非周期性请求队列中均存在空闲空间，应用程序便可写入发送数据。可通过读取主机非周期性发送 FIFO 和队列状态寄存器 (HNPTXSTS) 来了解二者的可用空间。

### 34.5.10 USBFS 设备 FIFO 架构

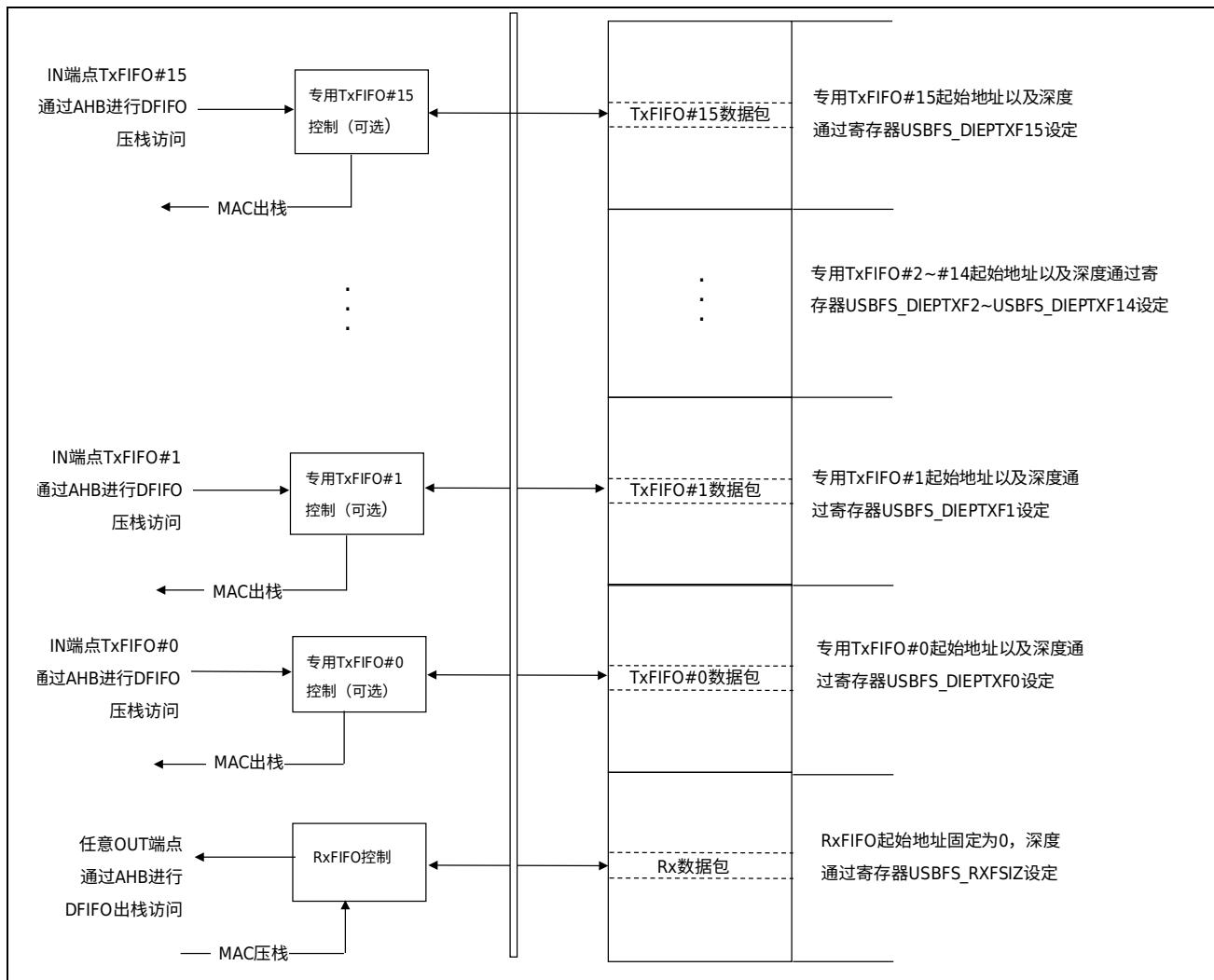


图 34-6 USBFS 设备模式下 FIFO 架构示意图

#### 34.5.10.1 设备 RxFIFO

USBFS 设备使用单个接收 FIFO 接收发送到所有 OUT 端点的数据。只要 RxFIFO 中有空余空间，收到的数据包就挨个填入 RxFIFO。除了有效数据外，接收到的数据包状态（包含 OUT 端点目标编号、字节数、数据 PID 和对所接收数据的验证）也由模块进行存储。没有可用空间时，设备会回复主机事务 NAK 应答并在被寻址的端点上触发中断。接收 FIFO 的大小在接收 FIFO 大小寄存器 (GRXFSIZ) 中配置。

单个接收 FIFO 架构使得 USB 设备更高效地填充接收 RAM 缓冲区：

- 所有 OUT 端点共享同一个 RAM 缓冲区（共享 FIFO）
- 对于任意主机序列 OUT 令牌，USBFS 模块可将接收 FIFO 填充至限值

只要至少有一个数据包在 RxFIFO 中可供读取，应用程序就会一直接收 RxFIFO 非空中断 (USBFS\_GINTSTS 中的 RXFNE 位)。应用程序从接收状态读取和出栈寄存器(GRXSTSP)中读取数据包信息，最后通过读取与端点相关的出栈地址从接收 FIFO 读出相应数据。

### 34.5.10.2 设备 TxFIFO

模块为各个 IN 端点提供了专用的 FIFO。应用程序通过非周期发送 FIFO 大小寄存器 (USBFS\_DIEPTSIZE0) 为 IN 端点 0 配置 FIFO 大小；通过设备 IN 端点发送 FIFOx 寄存器 (DIEPTSIZEx) 为 IN 端点 x 配置 FIFO 大小。

### 34.5.11 USBFS FIFO RAM 分配

#### 34.5.11.1 主机模式

##### 接收 FIFO RAM 分配

状态信息随各个接收数据包写入 FIFO。因此，必须至少为接收数据包分配（最大数据包大小 / 4）+ 1 的空间。如果使能了多个同步通道，则为接收连续数据包分配的空间必须至少为（最大数据包大小 / 4）的两倍 + 1。通常，推荐的空间为（最大数据包 / 4 + 1）的两倍，这样当上一个数据包向 CPU 传送时，USB 可同时接收后续的数据包。

传输完成状态信息和该端点收到的最后一个数据包会一起被写入 FIFO。所以必须为此分配一个位置。

以 DMA 模式运行时，每个主机通道的 DMA 地址寄存器将存储在 FIFO 中，因此需要在 FIFO 中为每个通道保留一个位置存放其地址寄存器。

##### 发送 FIFO RAM 分配

主机非周期性发送 FIFO 所需的最小 RAM 为所支持的所有非周期性 OUT 通道上传输的最大数据包的大小。

通常，推荐的空间为最大数据包大小的两倍，这样当 USB 正在发送当前数据包的同时，AHB 可以往发送 FIFO 填入下一个数据包。

主机周期性发送 FIFO 所需的最小 RAM 为所支持的所有周期性 OUT 通道上传输的最大数据包的大小。如果至少有一个同步 OUT 端点，则空间必须至少为该通道中最大数据包大小的两倍。

以 DMA 模式运行时，每个主机通道的 DMA 地址寄存器将存储在 FIFO 中，因此需要在 FIFO 中为每个通道保留一个位置存放其地址寄存器。

#### 34.5.11.2 设备模式

##### 接收 FIFO RAM 分配

应用程序应为 SETUP 数据包分配 RAM：接收 FIFO 中必须保留 11 个位置以在控制端点上接收 SETUP 数据包。USBFS 模块不会向这些为 SETUP 数据包保留的位置写入任何其它数据。将会为全局 OUT NAK 分配一个位置。状态信息随各个接收数据包写入 FIFO。因此，必须至少为接收数据包分配（最大数据包大小 / 4）+ 1 的空间。如果使能了多个同步端点，则为接收连续数据包分配的空间必须至少为（最大

数据包大小/4) 的两倍+1。通常，推荐的空间为 (最大数据包/4 + 1) 的两倍，这样当上一个数据包向 CPU 传送时，USB 可同时接收后续的数据包。

传输完成状态信息和该端点收到的最后一个数据包会一起被推入 FIFO。通常情况下，推荐为每个 OUT 端点分配一个位置。

### 发送 FIFO RAM 分配

各个 IN 端点发送 FIFO 所需的最小 RAM 空间为该特定 IN 端点的最大数据包大小。

## 34.5.12 USBFS 系统性能

凭借大容量 RAM 缓冲区、高度可配置的 FIFO 大小、通过 AHB 压栈/出栈寄存器进行 32 位 FIFO 快速访问，尤其是高级 FIFO 控制机制可获得最佳 USB 和系统性能。实际上，无论当前 USB 序列如何，USBFS 均可通过该机制高效填充可用的 RAM 空间。借助这些特性：

- 应用程序有足够的裕量来计算并校正 CPU 的负载，从而优化 CPU 带宽利用率：
  - 应用程序可先积累大量发送数据，再通过 USB 发送出去
  - 可带来足够的时间裕量，以从接收 FIFO 读取数据
- USB 模块能够保持全速工作状态，也就是提供最大的全速带宽（尽量多的硬件自动运行，尽量少的软件参与）
  - USB 模块可提前积累大量发送数据供其支配，从而可对 USB 数据发送进行自主管理
  - 接收缓冲区中有大量空白空间，可通过 USB 中的数据自动填满

由于 USBFS 模块能够高效填充 1.25KB RAM 缓冲区且 1.25KB 发送/接收数据足以满足一个全速帧所能容纳的数据量，因此 USB 系统在一帧之内可以无需应用程序干预达到最大 USB 带宽。

### 34.5.13 USBFS 中断和事件

USBFS 中断有以下两类，SOTP 模式唤醒中断 USBFS\_WKUP 和 USBFS 全局中断 USBFS\_GLB。

#### USBFS\_WKUP 中断

USBFS\_WKUP 中断用于 STOP 模式下，通过 USBFS\_DP 或者 USBFS\_DM 唤醒系统的 STOP 模式，该中断使能位为 INT\_WUPEN.USF\_WUEN。

使用 USBFS 唤醒系统 STOP 前，需要确保 USBFS 控制器处于挂起状态，并设定寄存器 USBFS\_SYCTLREG 中的对应的滤波范围和使能滤波功能。

#### USBFS\_GLB 中断

USBFS\_GLB 中断是软件需要处理的主要中断，全局中断的标志位可在 USBFS\_GINTSTS 寄存器读取。

表 34-2 USBFS\_GLB 中断事件表

中断标志	描述	运行模式	内部事件源
WKUPINT	恢复/远程唤醒中断	主机或者设备	-
VBUSVINT	VBUS有效中断	设备	-
DISCINT	断开连接中断	主机	-
CIDSCHG	连接器ID线状态变化中断	主机或者设备	-
PTXFE	周期性Tx FIFO空中断	主机	-
LPMINT	LPM中断	主机或者设备	
HCINT	主机通道中断	主机	-
HPRTINT	主机端口中断	主机	-
DATAFSUSP	数据获取挂起	设备	-
IPXFR/INCOMPISOOUT	未完成周期性传输/未完成OUT同步传输	设备	-
IISOIXFR	未完成IN同步传输	设备	-
OEPINT	OUT端点中断	设备	-
IEPINT	IN端点中断	设备	-
EOPF	周期性帧结束中断	设备	-
ISOODRP	丢弃同步OUT数据包中断	设备	-
ENUMDNE	枚举完成	设备	-
USBRST	USB复位中断	设备	-
USBSUSP	USB挂起中断	设备	-
ESUSP	早期挂起中断	设备	-
GONAKEFF	全局OUT NAK有效中断	设备	-
GINAKEFF	全局非周期性IN NAK有效中断	设备	-
NPTXFE	非周期性Tx FIFO空中断	主机	-
RXFNE	RxFIFO非空中断	主机或者设备	-
SOF	帧起始中断	主机或者设备	是

中断标志	描述	运行模式	内部事件源
MMIS	模式不匹配中断	主机或者设备	-

## 34.6 USBFS 编程模型

### 34.6.1 USBFS 模块初始化

应用程序必须执行模块初始化序列。

模式决定方法请参考【USBFS 模式决定】。

本节介绍了 USBFS 控制器上电后的初始化过程。无论是以主机模式还是设备模式工作，应用程序都必须遵循初始化序列。根据模块配置对所有模块全局寄存器进行初始化：

- 在 USBFS\_GAHBCFG 寄存器中编程以下字段：
  - 全局中断屏蔽位 GINTMSK = 1
  - RxFIFO 非空（USBFS\_GINTSTS 中的 RXFNE 位）
  - 周期性 TxFIFO 空门限
- 在 USBFS\_GUSBCFG 寄存器中编程以下字段：
  - FS 超时校准字段
  - USB 周转时间字段
- 软件必须取消对 USBFS\_GINTMSK 寄存器中以下位的屏蔽：
  - 模式不匹配中断屏蔽
- 通过读取 USBFS\_GINTSTS 中的 CMOD 位，软件可确定 USBFS 控制器是在主机模式还是设备模式下工作。

### 34.6.2 USBFS 主机初始化

要将模块作为主机进行初始化，应用程序必须执行以下步骤：

1. 编程 USBFS\_GINTMSK 寄存器中的 HPRTINT 以将对取消屏蔽。
2. 编程 USBFS\_HCFG 寄存器以选择全速主机。
3. 将 USBFS\_HPRT 中的 PPWR 位编程为 1，给 USB 总线提供 VBUS。
4. 等待 USBFS\_HPRT 中的 PCDET 中断。这表示某设备已连接到主机端口。
5. 将 USBFS\_HPRT 中的 PRST 位编程为 1，在 USB 总线上发出复位信号。
6. 至少等待 10ms，以便完成复位过程。
7. 将 USBFS\_HPRT 中的 PRST 位编程为 0。
8. 等待 USBFS\_HPRT 中的 PENCHNG 中断。
9. 读取 USBFS\_HPRT 中的 PSPD 位以获取枚举速度。
10. 使用所选 PHY 时钟，相应地设置 HFIR 寄存器。

11. 根据步骤 9 中检测到的设备速度编程 USBFS\_HCFG 寄存器中的 FSLSPCS 字段。如果 FSLSPCS 发生更改，则必须执行端口复位。
12. 编程 USBFS\_GRXFSIZ 寄存器以选择接收 FIFO 的大小。
13. 编程 USBFS\_HNPTXFSIZ 寄存器，以选择用于非周期性通信事务的非周期性发送 FIFO 的大小和起始地址。
14. 编程 USBFS\_HPTXFSIZ 寄存器，以选择用于周期性事务的周期性通信发送 FIFO 的大小和起始地址。

要与设备通信，系统软件必须初始化并使能至少一个通道。

### 34.6.3 USBFS 设备初始化

上电期间或者从主机模式切换为设备模式后，应用程序必须执行下列步骤来将模块作为设备进行初始化。

1. 在 USBFS\_DCFG 寄存器中编程以下字段：
  - 设备速度
  - 非零长度状态 OUT 握手信号
2. 编程 USBFS\_GINTMSK 寄存器以取消屏蔽以下中断：
  - USB 复位
  - 枚举完成
  - 早期挂起
  - USB 挂起
  - SOF
3. 等待 USBFS\_GINTSTS 中的 VBUSVINT 中断，表示进入供电状态。
4. 等待 USBFS\_GINTSTS 中的 USBRST 中断。这表示已在 USB 上检测到复位信号，复位过程自接收到此中断后约持续 10ms。
5. 等待 USBFS\_GINTSTS 中的 ENUMDNE 中断。此中断指示 USB 上复位过程结束。接收到此中断时，应用程序必须读取 USBFS\_DSTS 寄存器以确定枚举速度并执行枚举完成时的端点初始化中所列的步骤。

此时，设备已准备好接受 SOF 数据包并在控制端点 0 上执行控制传输。

### 34.6.4 USBFS DMA 模式

USB 使用 AHB 主接口来获取发送数据包数据（AHB 到 USB）和接收数据更新（USB 到 AHB）。AHB 主接口使用经过编程的 DMA 地址（主机模式下的 HCDMAx 寄存器和设备模式下的 DIEPDMAx/DOEPDMAx 寄存器）来访问数据缓冲区。

## 34.6.5 USBFS 主机编程模型

### 34.6.5.1 通道初始化

应用程序必须初始化一个或多个通道，之后才能与所连接的设备通信。

要初始化和使能通道，应用程序必须执行以下步骤：

1. 编程 USBFS\_GINTMSK 寄存器以取消对以下位的中断屏蔽：
  - 用于 OUT 事务的非周期性发送 FIFO 为空（在流水线事务级别工作且数据包计数字段编程值大 1 时适用）。
  - 用于 OUT 事务的非周期性发送 FIFO 为半空（在流水线事务级别工作且数据包计数字段编程值大 1 时适用）。
2. 编程 USBFS\_HAINITMSK 寄存器以使能所选通道中断。
3. 编程 USBFS\_HCINTMSK 寄存器，以使能主机通道中断寄存器中反映的和通信事务有关的中断。
4. 编程所选通道的 USBFS\_HCSTSIZx 寄存器，指定以字节为单位的总传输大小和包括短数据包在内的预期数据包个数。应用程序必须使用初始数据 PID（用于第一个 OUT 事务或预期从第一个 IN 事务获取）编程 PID 字段。
5. 编程所选通道的 USBFS\_HCCHARx 寄存器，指定设备的端点特性，例如类型、速度、方向等。  
(仅当应用程序准备好发送或接收数据包时，才能通过将通道使能位置 1 来使能通道)。

### 34.6.5.2 通道停止

应用程序可以通过编程 USBFS\_HCCHARx 寄存器将 CHDIS 和 CHENA 位置 1 来禁止任何通道。这会使 USBFS 主机清空之前在该通道上发出的请求（如果有）并生成通道停止中断。应用程序在将通道重新分配给其它通信事务之前，必须等待 USBFS\_HCINTx 中的 CHH 中断。USBFS 主机不会中断已在 USB 上启动的通信事务。

禁止通道前，应用程序必须确保非周期性请求队列（禁止非周期性通道时）或周期性请求队列（禁止周期性通道时）中至少有一个空闲空间。应用程序可以在请求队列已满时（禁止通道之前），通过编程 USBFS\_HCCHARx 寄存器将 CHDIS 位置 1 和将 CHENA 位清零，清空请求队列。出现以下任一情况时，应用程序将禁止通道：

1. IN 或 OUT 通道的 USBFS\_HCINTx 中接收到 STALL、TXERR、BBERR 或 DTERR 中断。应用程序在接收到通道停止信号之前，必须能够接收相同通道的其它中断（DTERR、Nak、Data、TXERR）。
2. 接收到 USBFS\_GINTSTS 中的 DISCINT（断开设备连接）中断。（应用程序将禁止所有已使能的通道）。
3. 应用程序在传输正常完成之前将其中止。

在 DMA 模式时，应用程序不能通过写寄存器停止不可分割的周期性传输。

## 34.6.6 USBFS 设备编程模型

### 34.6.6.1 USB 复位时的端点初始化

1. 为所有 OUT 端点将 NAK 位置 1
    - USBFS\_DOEPCTLx 中, SNAK = 1 (对于所有 OUT 端点)
  2. 取消对以下中断位的屏蔽
    - USBFS\_DAINTMSK 中, INEPO=1 (控制 0 IN 端点)
    - USBFS\_DAINTMSK 中, OUTEP0=1 (控制 0 OUT 端点)
    - DOEPMSK 中, STUP=1
    - DOEPMSK 中, XFRC=1
    - DIEPMSK 中, XFRC=1
    - DIEPMSK 中, TOC=1
  3. 为每个 FIFO 设置数据 FIFO RAM
    - 对 USBFS\_GRXFSIZ 寄存器进行编程, 以能够接收控制传输的 OUT 数据和设置数据。该寄存器必须至少等于控制端点 0 的 1 个最大数据包大小+2 个字 (用于控制 OUT 数据包的状态)+10 个字 (用于 SETUP 数据包)。
    - 对 USBFS\_TX0FSIZ 寄存器进行编程 (取决于所选的 FIFO 编号), 以能够发送控制 IN 数据。该寄存器至少必须等于控制端点 0 的 1 个最大数据包大小。
  4. 对端点相关寄存器中的以下字段进行编程, 以使控制 OUT 端点 0 接收 SETUP 数据包
    - USBFS\_DOEPTSIZ0 中的 STUPCNT=3 (接收最多 3 个连续的 SETUP 数据包)
- 此时, 接收 SETUP 数据包所需的所有初始化工作便已完成。

### 34.6.6.2 USB 复位时的端点端点初始化

1. 在枚举完成中断 (USBFS\_GINTSTS 中的 ENUMDNE) 中, 读取 USBFS\_DSTS 寄存器以确定设备的枚举速度。
2. 对 USBFS\_DIEPCTL0 中的 MPSIZ 字段进行编程以设置最大数据包大小。该步骤配置控制端点 0。控制端点的最大数据包大小取决于枚举速度。

此时, 设备已准备好接收 SOF 数据包并配置为在控制端点 0 执行控制传输。

### 34.6.6.3 收到 SetAddress 命令时的端点初始化

本节介绍了应用程序在 SETUP 数据包中接收到 SetAddress 命令时必须执行的操作。

1. 使用在 SetAddress 命令中接收到的设备地址来对 USBFS\_DCFG 寄存器进行编程。
2. 对模块进行编程以发出状态阶段的 IN 数据包。

#### 34.6.6.4 收到 SetConfiguration/SetInterface 命令时的端点初始化

本节介绍了应用程序在 SETUP 包中接收 SetConfiguration 或 SetInterface 命令时必须执行的操作。

1. 接收到 SetConfiguration 命令时，应用程序必须对端点寄存器进行编程，以使用新配置中有效端点的特性来配置这些端点寄存器。
2. 接收到 SetInterface 命令时，应用程序必须对命令指定的端点的端点寄存器进行编程。
3. 在先前配置或其它设置中有效的端点在新的配置或其它设置中无效。必须停用这些无效端点。
4. 使用 USBFS\_DAIINTMSK 寄存器使能有效端点的中断，屏蔽无效端点的中断。
5. 为每个 FIFO 设置数据 FIFO RAM。
6. 配置完所有必需的端点后，应用程序必须对模块进行编程以发送状态阶段的 IN 数据包。

此时，设备模块已可以接收和发送任何类型的数据包。

#### 34.6.6.5 端点激活

本节介绍激活设备端点或者将现有设备端点配置为新类型所需的步骤。

1. 在 USBFS\_DIEPCTLx 寄存器（对于 IN 或双向端点）或 USBFS\_DOEPCTLx 寄存器（对于 OUT 或双向端点）的以下字段中，对所需端点的特性进行编程。
  - 最大数据包大小
  - USB 活动端点位置 1
  - 端点初始数据同步位（对于中断和批量端点）
  - 端点类型
  - TxFIFO 编号
2. 激活端点后，模块便开始解码发送到该端点的令牌，并在收到的令牌有效的情况下回复有效握手信号。

#### 34.6.6.6 端点停用

本节介绍停用现有端点所需的步骤。

1. 在要停用的端点中，将 USBFS\_DIEPCTLx 寄存器（对于 IN 或双向端点）或 USBFS\_DOEPCTLx 寄存器（对于 OUT 或双向端点）中的 USB 活动端点位清零。
2. 停用端点后，模块便会忽略发送到该端点的令牌，从而导致 USB 超时。

## 34.6.7 USBFS 操作模型

### 34.6.7.1 SETUP 和 OUT 数据传输

本节介绍了数据 OUT 传输和 SETUP 事务期间的内部数据流和应用程序操作步骤。

#### 数据包读取

本节介绍如何从接收 FIFO 读取数据包（OUT 数据和 SETUP 数据包）。

1. 捕获到 RXFNE 中断（USBFS\_GINTSTS 寄存器）时，应用程序必须读取接收状态弹出寄存器（USBFS\_GRXSTSP）。
2. 应用程序可以通过写入 RXFNE=0（在 USBFS\_GINTMSK 中）来屏蔽 RXFNE 中断（在 USBFS\_GINTSTS 中），直到它把数据包从接收 FIFO 中读取出来。
3. 如果已接收数据包的字节计数不是 0，则从接收数据 FIFO 中弹出这些数据并存储在存储器中。如果接收到的数据包字节计数为 0，则不会从接收数据 FIFO 中弹出任何数据。
4. 从接收 FIFO 读出的数据包状态有以下几种状态：
  - 全局 OUT NAK：

PKTSTS=全局 OUT NAK，BCNT=0x000，EPNUM 和 DPID 的值无关紧要。  
这些数据表示全局 OUT NAK 位已生效。
  - SETUP 数据包：

PKTSTS=SETUP，BCNT=0x008，EPNUM=控制 EP 编号，DPID=D0。这些数据表示指定端点上收到的 SETUP 数据包现在可从接收 FIFO 中读取。
  - 建立阶段完成：

PKTSTS=建立阶段完成，BCNT=0x0，EPNUM=控制 EP 编号，DPID 值无关紧要。  
这些数据表示指定端点的建立阶段完成并且数据阶段已启动。在此状态条目从接收 FIFO 中弹出后，模块将在该控制 OUT 端点上产生建立中断。
  - OUT 数据包：

PKTSTS=DataOUT，BCNT=接收的 OUT 数据包的大小 (BCNT:0~1024)，EPNUM=收到数据包的端点编号，DPID=实际数据 PID。
  - 数据传输完成：

PKTSTS=OUT 数据传输完成，BCNT=0x0，EPNUM=完成数据传输的 OUT EP 编号，DPID 值无关紧要。  
这些数据表示指定 OUT 端点的 OUT 数据传输完成。在此状态条目从接收 FIFO 中弹出后，模块将在指定的 OUT 端点上引发“传输完成”中断。
5. 从接收 FIFO 中弹出数据后，必须取消对 RXFNE 中断的屏蔽（USBFS\_GINTSTS）。
6. 每次应用程序检测到 USBFS\_GINTSTS 中的 RXFNE 中断时，都将重复步骤 1 到 5。读取空的接收 FIFO 可能导致未定义的模块行为。

## SETUP 事务

本节介绍了模块处理 SETUP 数据包的方式以及应用程序处理 SETUP 事务的顺序。

应用程序要求：

1. 要接收 SETUP 数据包，必须将控制 OUT 端点 STUPCNT 字段 (USBFS\_DOEPTSI $x$ ) 编程为非零值。如果应用程序将 STUPCNT 字段编程为非零值，模块会接收 SETUP 数据包并将其写入接收 FIFO，而不考虑 NAK 状态和 USBFS\_DOEPCTL $x$  中的 EPENA 位设置。控制端点每收到一个 SETUP 数据包后，STUPCNT 字段都会递减。如果在接收 SETUP 数据包之前，未将 STUPCNT 字段编程为适当值，模块仍能接收 SETUP 数据包并使 STUPCNT 字段递减，但应用程序可能无法确定在控制传输的建立阶段中接收的 SETUP 数据包正确数量。
  - 在 USBFS\_DOEPTSI $x$  中，STUPCNT=3
2. 应用程序必须始终在接收数据 FIFO 中分配一些额外空间，以便能够在控制端点上接收连续的最多三个 SETUP 数据包。
  - 预留空间 10 个字。第一个 SETUP 数据包需要 3 个字，“建立阶段完成”状态双字需要 1 个字，还需要 6 个字以存储两个额外的 SETUP 数据包。
  - 每个 SETUP 数据包需要 3 个字以存储 8 个字节的 SETUP 数据和 4 个字节的 SETUP 状态。模块将在接收 FIFO 中保留这些空间。
  - 这段 FIFO 仅用于存储 SETUP 包，绝对不会将该空间用于数据包。
3. 应用程序必须从接收 FIFO 中读取 SETUP 数据包的 2 个字。
4. 应用程序必须从接收 FIFO 中读取并丢弃“建立阶段完成”状态字

内部数据流：

1. 接收到 SETUP 数据包时，模块会将接收到的数据写入接收 FIFO，而不会检查接收 FIFO 中的可用空间，且不考虑端点的 NAK 和 STALL 位设置。
  - 模块会在内部将接收到 SETUP 数据包的控制 IN/OUT 端点的 IN NAK 和 OUTNAK 位置 1。
2. USB 上接收到的每个 SETUP 数据包，模块会将 3 个字的数据写入接收 FIFO，并且将 STUPCNT 字段递减 1。
  - 第一个字包含由模块所使用的内部控制信息
  - 第二个字包含 SETUP 命令的前 4 个字节
  - 第三个字包含 SETUP 命令的最后 4 个字节
3. 当建立阶段结束，数据 IN/OUT 阶段开始时，模块会将一个状态条目（“建立阶段完成”字）写入接收 FIFO，指示建立阶段完成。
4. 在 AHB 端，SETUP 数据包被应用程序读取。
5. 当应用程序从接收 FIFO 中弹出“建立阶段完成”字时，模块将使用 STUP 中断 (USBFS\_DOEPINT $x$ ) 来中断应用程序，指示其可以处理接收到的 SETUP 数据包。

- 模块会将控制 OUT 端点的端点使能位清零。

应用程序编程顺序：

1. 对 USBFS\_DOEPTSI $Zx$  寄存器进行编程。
  - STUPCNT=3
2. 等待 RXFNE 中断 (USBFS\_GINTSTS) 并且从接收 FIFO 中读取数据包。
3. STUP 中断的触发(USBFS\_DOEPINT $x$ ) 表示 SETUP 数据传输成功完成。
  - 发生该中断时，应用程序必须读取 USBFS\_DOEPTSI $Zx$  寄存器以确定接收的 SETUP 数据包数量并处理最后接收的 SETUP 数据包。

处理三个以上连续的 SETUP 数据包：

根据 USB2.0 规范，在 SETUP 数据包错误中，主机通常不会向同一个端点发送 3 个以上连续的 SETUP 数据包。但是，USB2.0 规范并未限制主机可以向同一个端点发送的连续 SETUP 数据包数量。发生这种情况时，USBFS 控制器将生成中断 (USBFS\_DOEPINT $x$  中的 B2BSTUP)。

## 将全局 OUT NAK 置 1

内部数据流：

1. 如果应用程序将全局 OUT NAK (USBFS\_DCTL 中的 SGONAK 位) 置 1，模块将停止向接收 FIFO 中写入 SETUP 数据包以外的数据。无论接收 FIFO 中可用空间大小如何，设备都会对主机发送的非同步 OUT 令牌回复 NAK，而对同步 OUT 数据包直接予以忽略。
2. 模块将全局 OUT NAK 写入接收 FIFO。应用程序必须为此留出足够空间。
3. 当应用程序从接收 FIFO 中弹出全局 OUT NAK 字时，模块会将 GONAKEFF 中断 (USBFS\_GINTSTS) 置 1。
4. 应用程序检测到该中断后，会认为模块处于全局 OUT NAK 模式。应用程序可以通过将 USBFS\_DCTL 中的 SGONAK 位清零来清除该中断。

应用程序编程顺序：

1. 要停止接收任何类型的数据到接收 FIFO 中，应用程序必须通过编程以下字段以将全局 OUT NAK 位置 1。
  - 在 USBFS\_DCTL 中，SGONAK =1
2. 等待 USBFS\_GINTST 中的 GONAKEFF 中断。一旦被触发，该中断表示模块已停止接收 SETUP 数据包以外的任何类型数据。
  - 如果应用程序已将 USBFS\_DCTL 中的 SGONAK 位置 1，则在模块引发 GONAKEFF 中断 (USBFS\_GINTSTS) 之前，应用程序可以接收有效 OUT 数据包。
3. 应用程序可通过对 USBFS\_GINTMSK 寄存器中的 GINAKEFFM 位执行写操作来暂时屏蔽此中断。

- 在 USBFS\_GINTMSK 寄存器中, GINAKEFFM=0
- 4. 当应用程序准备退出全局 OUT NAK 模式时, 必须将 USBFS\_DCTL 中的 SGONAK 位清零。此操作还会清除 GONAKEFF 中断 (USBFS\_GINTSTS)。
- 在 CGONAK 中, USBFS\_DCTL=1
- 5. 如果应用程序在之前已屏蔽此中断, 则必须按以下方式取消对该中断的屏蔽:
  - 在 GINTMSK 中, GINAKEFFM=1

### 将全局 OUT NAK 置 1

应用程序必须使用以下顺序禁止已使能的 OUT 端点。

应用程序编程顺序:

1. 禁止任何 OUT 端点前, 应用程序必须在模块中使能全局 OUT NAK 模式。
  - 在 USBFS\_DCTL 中, SGONAK=1
2. 等待 GONAKEFF 中断(USBFS\_GINTSTS)
3. 通过编程以下字段来禁止 OUT 端点:
  - 在 USBFS\_DOEPCTLx 中, EPDIS=1
  - 在 USBFS\_DOEPCTLx 中, SNAK=1
4. 等待 EPDISD 中断 (USBFS\_DOEPINTx), 该中断表示已完全禁止 OUT 端点。引发 EPDISD 中断时, 模块还会将以下位清零:
  - 在 USBFS\_DOEPCTLx 中, EPDIS=0
  - 在 USBFS\_DOEPCTLx 中, EPENA=0
5. 应用程序必须将全局 OUT NAK 位清零, 以开始从其它未禁止的 OUT 端点接收数据。
  - 在 USBFS\_DCTL 中, SGONAK=0

### 通用非同步 OUT 数据传输

本节介绍一种常规非同步 OUT 数据传输 (控制、批量或中断)。

应用程序要求:

1. 建立 OUT 传输前, 应用程序必须在存储器中分配一个缓冲区, 以容纳要作为 OUT 传输的一部分而接收的所有数据。
2. 对于 OUT 传输, 端点的传输大小寄存器中的传输大小字段必须是端点的最大数据包大小的倍数 (且以字对齐)。
  - 传输大小[EPNUM] =  $n \times (\text{MPSIZ}[EPNUM] + 4 - (\text{MPSIZ}[EPNUM] \bmod 4))$
  - 数据包计数[EPNUM] = n
  - $n > 0$
3. 发生 OUT 端点中断时, 应用程序必须读取端点的传输大小寄存器以计算存储器中有效数据量。接收的有效数据量可能小于编程的传输大小。

- 存储器中的有效数据量 = 应用程序设置的初始传输量 - 模块更新后的剩余传输量
- 接收到 USB 数据包数 = 应用程序设置的初始数据包数 - 模块更新后的剩余数据包数

内部数据流：

1. 应用程序必须在端点相关寄存器中设置传输大小和数据包计数字段，将 NAK 位清零，并使能端点来接收数据。
2. NAK 位清零后，模块便开始接收数据并将数据写入接收 FIFO（只要接收 FIFO 中有空间）。对于 USB 上接收的每个数据包，数据包及其状态都会写入接收 FIFO。写入接收 FIFO 的每个数据包（数据量达到最大数据包大小的数据包或短数据包）都会使该端点的数据包计数字段递减 1。
- 收到的数据包若 CRC 无效，则自动被从接收 FIFO 中清除。
- 在 USB 上为数据包回复 ACK 后，模块将丢弃主机因无法检测到 ACK 而重新发送的非同步 OUT 数据包。应用程序不会在具有相同数据 PID 的相同端点上检测到多个连续的 OUT 数据包。在这种情况下，数据包计数不会递减。
- 如果接收 FIFO 中没有空间，则会忽略同步或非同步数据包并且不会将它们写入接收 FIFO。此外，非同步 OUT 令牌将会收到 NAK 握手应答。
- 在上述所有三种情况中，数据包计数都不会递减，因为没有任何数据写入接收 FIFO。
3. 当数据包计数变为 0 或者在端点上接收到短数据包时，该端点的 NAK 位将置 1。NAK 置位 1 后，将忽略同步或非同步数据包并且不会将它们写入接收 FIFO，同时非同步 OUT 令牌会收到 NAK 握手应答。
4. 在数据写入接收 FIFO 后，应用程序将从接收 FIFO 中读取数据并将数据写入外部存储器，一次一个数据包，逐个端点过来。
5. 在 AHB 上向外部存储器写入完每个数据包后，端点的传输大小都会自动减去该数据包的大小。
6. 在以下情况时，OUT 端点的 OUT 数据传输完成状态将写入接收 FIFO：
  - 传输大小为 0 并且数据包计数为 0
  - 写入接收 FIFO 的最后一个 OUT 数据包是短数据包  
(数据包大小:0~最大数据包大小-1)
7. 当应用程序弹出此状态条目 (OUT 数据传输完成)，并生成该端点的传输完成中断，同时清零端点使能位。

应用程序编程顺序：

1. 使用传输大小和相应数据包个数对 USBFS\_DOEPTSIZx 寄存器进行编程。
2. 使用端点特性对 USBFS\_DOEPCTLx 寄存器进行编程，并将 EPENA 和 CNAK 置位 1。
  - 在 USBFS\_DOEPCTLx 中，EPENA=1
  - 在 USBFS\_DOEPCTLx 中，CNAK =1
3. 等待 RXFNE 中断 (在 USBFS\_GINTSTS 中) 并且从接收 FIFO 中读走数据包。

- 此步骤可重复多次，具体取决于传输大小。
4. 触发 XFRC 中断 (USBFS\_DOEPINTx)，以表示非同步 OUT 数据传输成功完成。
  5. 读取 USBFS\_DOEPTSIZx 寄存器，以确定有效数据量。

### 通用同步 OUT 数据传输

本节介绍常规的同步 OUT 数据传输。

应用程序要求：

1. 非同步 OUT 数据传输的所有应用程序要求均适用于同 OUT 数据传输。
2. 对于同步 OUT 数据传输中的传输大小和数据包计数字段，必须始终将其设置为单个帧中可接收的最大数据包大小的数据包数目。同步类型的 OUT 数据传输事务必须在一个帧内完成。
3. 在周期性帧结束 (USBFS\_GINTSTS 中的 EOPF 中断) 之前，应用程序必须从接收 FIFO 中读取所有同步 OUT 数据包（数据条目和状态条目）。
4. 要接收下一帧中的数据，必须在 EOPF(USBFS\_GINTSTS)之后 SOF(USBFS\_GINTSTS)之前使能一个同步 OUT 端点。

内部数据流：

1. 同步 OUT 端点的内部数据流与非同步 OUT 端点的基本相同，但稍有差异。
2. 同步 OUT 端点通过将端点使能位置 1 并将 NAK 位清零来使能时，必须相应地将偶数/奇数帧位置 1。仅当符合以下条件时，模块才会在同步 OUT 端点上接收特定帧中的数据：
  - EONUM (在 USBFS\_DOEPCTLx 中) = FNSOF [0] (在 USBFS\_DSTS 中)
3. 当应用程序从接收 FIFO 中完整地读取一个同步 OUT 数据包（数据和状态）时，模块会根据从接收 FIFO 中读取的最后一个同步 OUT 数据包的数据 PID 更新 USBFS\_DOEPTSIZx 中的 RXDPID 字段。

应用程序编程顺序：

1. 使用传输大小和相应数据包计数对 USBFS\_DOEPTSIZx 寄存器进行编程
2. 使用端点特性对 USBFS\_DOEPCTLx 寄存器进行编程，并将端点使能位、清除 NAK 位和奇数/偶数帧位置 1。
  - EPENA1
  - CNAK=1
  - EONUM= (0: 偶数/1: 奇数)
3. 等待 RXFNE 中断 (在 USBFS\_GINTSTS 中) 并且从接收 FIFO 中读走数据包。
  - 此步骤可重复多次，具体取决于传输大小。
4. XFRC 中断 (在 USBFS\_DOEPINTx 中) 表示同步 OUT 数据传输完成。该中断不一定意味着存储器中的数据是有效的。
5. 对于同步 OUT 传输，应用程序可能并不总会检测到该中断。相反，应用程序可能检测到 USBFS\_GINTSTS 中的 IISOXXFRM 中断。

6. 读取 USBFS\_DOEPTSIZx 寄存器以确定接收的传输大小以及确定帧中接收的数据的有效性。

仅当符合以下条件之一时，应用程序才必须将存储器中接收的数据视为有效数据：

- RXDPID=D0（在 USBFS\_DOEPTSIZx 中）并且接收该有效数据的 USB 数据包数量=1
- RXDPID=D1（在 USBFS\_DOEPTSIZx 中）并且接收该有效数据的 USB 数据包数量=2
- RXDPID=D2（在 USBFS\_DOEPTSIZx 中）并且接收该有效数据的 USB 数据包数量=3

接收该有效数据的 USB 数据包数量=应用程序编程的初始数据包个数·模块更新后的剩余数据包个数。

应用程序可将无效数据包丢弃。

### 不完整的同步 OUT 数据传输

本节介绍了同步 OUT 数据包出现丢包时应用程序编程顺序。

内部数据流：

1. 对于同步 OUT 端点，可能不会始终引发 XFRC 中断（在 USBFS\_DOEPINTx 中）。如果模块丢弃同步 OUT 数据包，则在以下情况下，应用程序可能无法检测到 XFRC 中断（USBFS\_DOEPINTx）：

- 在接收 FIFO 无法容纳完整的 ISO OUT 数据包时，模块将丢弃接收到的 ISO OU 数据
- 接收到的同步 OUT 数据包存在 CRC 错误
- 模块接收到的同步 OUT 令牌损坏
- 应用程序从接收 FIFO 中读取数据的速度非常缓慢

2. 如果模块在所有同步 OU 端点的传输完成前检测到周期性帧结束，将触发未完成同步 OUT 数据中断（USBFS\_GINTSTS 中的 IISOXXFRM），指示至少有一个同步 OUT 端点上未触发 XFRC 中断（在 USBFS\_DOEPINTx 中）。此时，未完成传输的端点仍保持使能，但在 USB 的该端点上，没有进行中的有效传输。

应用程序编程顺序：

1. 硬件触发 IISOXXFRM 中断（USBFS\_GINTSTS）表示当前帧中至少有一个同步 OUT 端点具有未完成的传输。

2. 如果因未从端点完全读取同步 OUT 数据而发生这种情况，应用程序必须确保首先从接收 FIFO 读取走所有同步 OUT 数据（包括数据条目和状态条目），然后再继续处理。

- 从接收 FIFO 读取所有数据后，应用程序即可检测到 XFRC 中断（USBFS\_DOEPINTx）。在此情况下，应用程序必须重新使能端点以接收下一个帧中的同步 OUT 数据。

3. 当应用程序接收到 IISOXXFRM 中断（在 USBFS\_GINTSTS 中）时，应用程序必须读取所有同步 OUT 端点的控制寄存器（USBFS\_DOEPCTLx），以确定哪些端点在当前帧中具有不完整的传输。同时满足以下两个条件时，表示端点传输未完成：

- EONUM 位（在 USBFS\_DOEPCTLx 中）= FNSOF [0]（在 USBFS\_DSTS 中）
- EPENA=1（在 USBFS\_DOEPCTLx 中）

4. 在检测到 SOF 中断（在 USBFS\_GINTSTS 中）前，必须执行完成上一步操作，以确保当前帧编号未发生更改。
5. 对于具有不完整传输的同步 OUT 端点，应用程序必须丢弃存储器中的数据，并通过将 USBFS\_DOEPCTLx 中的 EPDIS 位置 1 来禁止端点。
6. 等待 EPDIS 中断（在 USBFS\_DOEPINTx 中），并且使能端点以在下一帧中接收新数据。
  - 由于模块可能需要一些时间才能禁止端点，因此应用程序在接收到无效同步数据后，可能无法接收下一个帧中的数据。

### 停止非同步 OUT 端点

本节介绍应用程序如何才能停止非同步端点。

1. 将模块置于全局 OUT NAK 模式。
2. 禁止所需的端点
  - 禁止端点时，请设置 STALL=1（在 USBFS\_DOEPCTL 中），而不是将 USBFS\_DOEPCTL 中的 SNAK 位置 1。STALL 位的优先级始终高于 NAK 位。
3. 当应用程序不再需要端点回复 STALL 握手信号时，必须将 STALL 位（在 USBFS\_DOEPCTLx 中）清零。
4. 如果应用程序由于收到主机的 SetFeature.Endpoint Halt 或 ClearFeature.Endpoint Halt 命令来设置或清除端点的 STALL 状态，则必须在该控制端点上的状态阶段传输前，将 STALL 位置 1 或清零。

### 34.6.7.2 IN 数据传输

#### 数据包写入

本节介绍在已使能专用发送 FIFO 的情况下应用程序如何将数据包写入端点 FIFO。

1. 应用程序可以选择轮询模式或中断模式。
    - 在轮询模式下，应用程序通过读取 USBFS\_DTXFSTSx 寄存器来监视端点发送数据 FIFO 的状态，从而确定数据 FIFO 中是否有足够空间。
    - 在中断模式下，应用程序等待 TXFE 中断（在 USBFS\_DIEPINTx 中），然后读取 USBFS\_DTXFSTSx 寄存器以确定数据 FIFO 中是否有足够空间。
    - 要写入单个非零长度的数据包，数据 FIFO 中必须有足够的空间来容纳整个数据包。
    - 要写入零长度的数据包，应用程序不能查看 FIFO 空间。
  2. 如果使用上述方法之一，当应用程序确定有足够的空间来写入发送数据包时，应用程序必须首先对端点控制寄存器进行相应写操作，然后再将数据写入数据 FIFO。通常，应用程序必须对 USBFS\_DIEPCTLx 寄存器执行读-修改-写操作，以避免在将端点使能位置 1 的同时，修改寄存器中的其它内容。
- 如果有足够空间，应用程序可将同一端点的多个数据包写入发送 FIFO。对于周期性 IN 端点，

应用程序只能一次写入一个帧内的多个数据包。只有先前一个帧的通信事务传输完成之后，应用程序才会写入下一个帧内要发送的所有数据包。

### 将 IN 端点 NAK 置 1

内部数据流：

1. 当应用程序将特定端点的 IN NAK 置 1 时，模块将停止端点上的数据发送，而不考虑端点发送 FIFO 中的数据是否可用。
2. 非同步端点收到 IN 令牌，回复 NAK 握手应答。
  - 同步端点收到 IN 令牌，返回零长度数据包
3. 模块在 USBFS\_DIEPINTx 中触发 INEPNE (IN 端点 NAK 有效) 中断以响应 USBFS\_DIEPCTLx 中的 SNAK 位。
4. 应用程序检测到该中断后，便会认为端点处于 IN NAK 模式。应用程序可通过将 USBFS\_DIEPCTLx 中的 CNAK 位置 1 来清除该中断。

应用程序编程顺序：

1. 要在特定 IN 端点上停止发送任何数据，应用程序必须将 IN NAK 位置 1。要将该位置 1，必须编程以下字段。
  - USBFS\_DIEPCTLx 中的 SNAK=1
2. 等待 USBFS\_DIEPINTx 中的 INEPNE 中断触发。该中断表示模块已在端点上停止发送数据。
3. 在应用程序将 NAK 位置 1 但“NAK 有效”中断尚未触发时，模块可以在端点上发送有效 IN 数据。
4. 应用程序可通过写入 DIEPMSK 中的 INEPNEM 位来临时屏蔽该中断。
  - 在 DIEPMSK 中，INEPNEM = 0
5. 要退出端点 NAK 模式，应用程序必须将 USBFS\_DIEPCTLx 中的 NAK 状态位 (NAKSTS) 清零。此操作还会清除 INEPNE 中断 (在 USBFS\_DIEPINTx 中)。
  - 在 USBFS\_DIEPCTLx 中，CNAK=1
6. 如果应用程序已将该中断屏蔽，则必须按以下方式取消屏蔽：
  - 在 DIEPMSK 中，INEPNEM=1

### 禁止 IN 端点

使用以下顺序来禁止先前已使能的特定 IN 端点。

应用程序编程顺序：

1. 应用程序必须先停止在 AHB 上写入数据，之后才能禁止 IN 端点。
2. 应用程序必须将端点设置为 NAK 模式。
  - USBFS\_DIEPCTLx 中的 SNAK=1
3. 等待 USBFS\_DIEPINTx 中的 INEPNE 中断。

4. 将必须禁止的端点的 USBFS\_DIEPCTLx 寄存器中的以下位置 1。
  - USBFS\_DIEPCTLx 中的 EPDIS=1
  - USBFS\_DIEPCTLx 中的 SNAK=1
5. USBFS\_DIEPINTx 中的 EPDISD 中断的触发表示模块已完全禁止指定的端点。在触发中断的同时，模块还会将以下位清零：
  - 在 USBFS\_DIEPCTLx 中，EPENA=0
  - 在 USBFS\_DIEPCTLx 中，EPDIS=0
6. 应用程序必须为周期性 IN EP 读取 USBFS\_DIEPTSIZx 寄存器，以计算端点上有多少数据是在 USB 上发送的。
7. 应用程序必须通过将 USBFS\_GRSTCTL 寄存器中的以下字段置 1，来清空端点发送 FIFO 中的数据：
  - TXFNUM（在 USBFS\_GRSTCTL 中）= 端点发送 FIFO 编号
  - TXFFLSH（在 USBFS\_GRSTCTL 中）= 1

应用程序必须轮询 USBFS\_GRSTCTL 寄存器，直至模块将 TXFFLSH 位清零，这表示 FIFO 清空操作结束。要在该端点上发送新数据，应用程序可以在稍后重新使能该端点。

### 通用非周期性 IN 数据传输

应用程序要求：

1. 建立 IN 传输前，应用程序必须确保组成一次 IN 传输的每个数据包都可以容纳在单个缓冲区中。
2. 对于 IN 传输，端点传输大小寄存器中的传输大小字段表示本次传输的有效数据量，它由多个最大数据包大小和单个短数据包组成。该短数据包在传输结束时发送。
  - 要发送多个最大数据包大小的数据包并在传输结束时外加一个短数据包：

传输大小[EPNUM] =  $x \times \text{MPSIZ}[EPNUM] + sp$   
如果 ( $sp > 0$ )，数据包计数[EPNUM] =  $x + 1$ 。  
否则，数据包计数[EPNUM] =  $x$
  - 要发送单个零长度数据包：

传输大小[EPNUM] = 0  
数据包计数[EPNUM] = 1
  - 要发送多个最大数据包大小的数据包并在传输结束时外加一个零长度数据包，应用程序必须将传输拆分为两个部分。

第一部分发送最大数据包大小的数据包，第二部分仅发送零长度数据包。
3. 使能某个端点进行数据传输后，模块会更新传输大小寄存器。在 IN 传输结束时，应用程序必

须读取传输大小寄存器，以确定送入发送 FIFO 中的数据已有多少通过 USB 发送出去。

4. 送入发送 FIFO 中的数据量 = 应用程序编程的初始传输大小 - 模块更新后的最终传输大小
  - 通过 USB 已经发送的数据量 = (应用程序编程的初始数据包计数 - 模块更新后的最终数据包计数) × MPSIZ[EPNUM]
  - 要通过 USB 发送的剩余数据量 = (应用程序编程的初始传输大小 - 已通过 USB 发送的数据量)

内部数据流：

1. 应用程序必须在特定端点的寄存器中设置传输大小和数据包计数字段，并使能该端点来发送数据。
2. 应用程序还必须向该端点的发送 FIFO 写入必需的数据。
3. 应用程序每向发送 FIFO 写入一个数据包，该端点的传输大小便会自动减去该数据包大小。应用程序持续从存储器获取数据来写入发送 FIFO，直到该端点的传输大小变为 0。向 FIFO 写入数据后，“FIFO 中的数据包数”计数会递增（这是一个 3 位计数，由模块在内部进行维护，每个 IN 端点发送 FIFO 对应一个。在 IN 端点 FIFO 中，模块所维护的最大数据包数始终为八个）。对于零长度数据包，每个 FIFO 均另有一个单独的标志，FIFO 中没有任何数据。
4. 当数据写入发送 FIFO 后，模块会在接收到 IN 令牌时将这些数据送出。每个数据包发送出去并收到回复的 ACK 握手信号后，该端点的数据包计数都会递减 1，直到数据包计数变 0 为止。发生超时时，数据包计数不会递减。
5. 对于零长度数据包（由内部零长度标志指示），模块会针对 IN 令牌发出一个零长度数据包，并递减数据包计数字段的值。
6. 如果接收到 IN 令牌的端点对应的 FIFO 中无数据，且该端点的数据包计数字段为零，则模块会针对该端点生成一个“Tx FIFO 为空时接收到 IN 令牌”(ITTXFE) 中断（前提是该端点的 NAK 位未置 1）。模块在该非同步端点上回复 NAK 握手信号。
7. 模块会在内部使 FIFO 指针重新返回到开头，并且不会生成超时中断。
8. 当传输大小为 0 且数据包计数为 0 时，将生成该端点的传输完成 (XFRC) 中断，同时将端点使能清零。

应用程序编程顺序：

1. 使用传输大小和相应数据包计数对 USBFS\_DIEPTSIZx 寄存器进行编程。
2. 使用端点特性对 USBFS\_DIEPCTLx 寄存器进行编程，并将 CNAK 和 EPENA（端点使能）位置 1。
3. 发送非零长度数据包时，应用程序必须轮询 USBFS\_DTXFSTSx 寄存器（其中 x 为与该端点相关的 FIFO 编号）以确定数据 FIFO 中是否有足够的空间。写入数据前，应用程序也可选用 TXFE 位（在 USBFS\_DIEPINTx 中）。

## 通用周期性 IN 数据传输

本节介绍典型的周期性 IN 数据传输。

应用程序要求：

1. 通用非周期性 IN 数据传输的应用程序要求 1、2、3、4 对周期性 IN 数据传输同样适用（只是对要求 2 稍加修改）。

- 应用程序只能发送若干个最大数据包大小的数据包或若干个最大数据包大小的包，外加传输结束时的一个短数据包。

要发送多个最大数据包大小的数据包并在传输结束时外加一个短数据包，必须满足以下条件：

$$\text{传输大小[EPNUM]} = x \times \text{MPSIZ[EPNUM]} + sp$$

(其中 x 是大于 0 的整数，且 sp 范围是 0 ~ MPSIZ[EPNUM]-1)

如果 (sp > 0)，数据包计数[EPNUM] = x + 1

否则，数据包计数[EPNUM] = x；

$$MCNT[EPNUM] = \text{数据包计数[EPNUM]}$$

- 应用程序无法在传输结束时发送零长度数据包。应用程序可以单独发送一个零长度数据包。
- 要发送单个零长度数据包：

传输大小[EPNUM]=0

数据包计数[EPNUM]=1

MCNT[EPNUM]=数据包计数[EPNUM]

2. 应用程序一次只能安排一帧的数据传输。

- $(MCNT - 1) \times \text{MPSIZ} < XFERSIZ \leq MCNT \times \text{MPSIZ}$

- PKTCNT = MCNT (在 USBFS\_DIEPTSI<sub>Z</sub>x 中)

- 如果 XFERSIZ < MCNT × MPSIZ，则传输的最后一个数据包为短数据包

- 请注意：MCNT 位于 USBFS\_DIEPTSI<sub>Z</sub>x 中、MPSIZ 位于 USBFS\_DIEPCTLx 中、PKTCNT 位于 USBFS\_DIEPTSI<sub>Z</sub>x 中、XFERSIZ 位于 USBFS\_DIEPTSI<sub>Z</sub>x 中

3. 接收到 IN 令牌前，应用程序必须将要在帧中发送的完整数据写入到发送 FIFO 中。在接收到 IN 令牌时，即使发送 FIFO 中该帧要发送的数据只差 1 个双字未写进来，模块也会执行 FIFO 为空时的操作。当发送 FIFO 为空时：

- 同步端点上将回复零长度数据包

- 中断端点上将回复 NAK 握手信号

内部数据流：

1. 应用程序必须在特定端点的寄存器中设置传输大小和数据包计数字段，并使能该端点来发送数据。
2. 应用程序还必须向与该端点相关联的发送 FIFO 写入必需的数据。

3. 应用程序每向发送 FIFO 写入一个数据包，该端点的传输大小便会自动减去该数据包大小。应用程序持续从存储器获取数据来写入发送 FIFO，直到该端点的传输大小变为 0。
4. 当周期性端点接收到 IN 令牌时，模块将开始发送 FIFO 中的数据（如果 FIFO 中有数据）。如果 FIFO 中没有该帧要发送的数据的完整数据包，则模块将为该端点生成一个“TxFIFO 为空时接收到 IN 令牌”中断。
  - 同步端点上将回复零长度数据包
  - 中断端点上将回复 NAK 握手信号
5. 端点的数据包计数会在下列情况下递减 1：
  - 对于同步端点，发送一个零长度或非零长度的数据包时
  - 对于中断端点，在发送 ACK 握手信号时递减
  - 当传输大小和数据包计数均为 0 时，将生成该端点的传输完成中断，同时将端点使能位清零。
6. 在“周期性帧间隔”（由 USBFS\_DCFG 中的 PFIVL 位控制）内，当模块发现任何在当前帧内应为空的同步 IN 端点 FIFO 中的数据还未发送完成时，都会在 USBFS\_GINTSTS 中生成一个 IISOIXFR 中断。

应用程序编程顺序：

1. 使用端点特性对 USBFS\_DIEPCTLx 寄存器进行编程，并将 CNAK 和 EPENA 位置 1。
2. 将需要在下一帧中发送的数据写入发送 FIFO。
3. 硬件触发 ITTXFE 中断（在 USBFS\_DIEPINTx 中）表示应用程序尚未将需要发送的全部数据写入发送 FIFO。
4. 如果在检测到中断前已使能中断端点，则将忽略该中断。如果中断端点未使能，则使能此端点，以便数据能够在收到下一次 IN 令牌时发送出去。
5. 硬件触发 XFRC 中断（在 USBFS\_DIEPINTx 中）时如果 USBFS\_DIEPINTx 中未产生 ITTXFE 中断，则表示成功完成同步 IN 传输。读取 USBFS\_DIEPTSIZx 寄存器时始终得到传输大小=0 且数据包计数=0，则表示所有数据都已通过 USB 发送完毕。
6. 置位 XFRC 中断（在 USBFS\_DIEPINTx 中）时无论是否产生 ITTXFE 中断（在 USBFS\_DIEPINTx 中），都表示成功完成中断 IN 传输。读取 USBFS\_DIEPTSIZx 寄存器时始终得到传输大小=0 且数据包计数=0，则表示所有数据都已通过 USB 发送完毕。
7. 在 USBFS\_GINTSTS 中置位未完成的同步 IN 传输 (IISOIXFR) 中断时如果未产生任何前述中断，则表示在当前帧中模块至少未收到 1 个周期性的 IN 令牌。

### 未完成同步 IN 数据传输

本节介绍应用程序针对未完成同步 IN 数据传输必须执行的操作。

内部数据流：

1. 符合下列条件之一时，即认为同步 IN 传输未完成：

- 模块在至少一个同步 IN 端点上接收到损坏的同步 IN 令牌。此时，应用程序检测到未完成同步 IN 传输中断（USBFS\_GINTSTS 中的 IISOIXFR 位）。
  - 应用程序向发送 FIFO 写入数据的速度过慢，在将完整数据写入 FIFO 之前便接收到 IN 令牌。此时，应用程序在 USBFS\_DIEPINTx 中检测到“Tx FIFO 为空时接收到 IN 令牌”中断。应用程序可忽略此中断，因为最终这将在周期性帧结束时产生一个未完成同步 IN 传输中断（USBFS\_GINTSTS 中的 IISOIXFR 位）。模块会通过 USB 发送一个零长度数据包来响应接收到的 IN 令牌。
2. 应用程序必须尽快停止向发送 FIFO 写入数据。
  3. 应用程序必须将端点的 NAK 位和禁止位置 1。
  4. 模块会禁止该端点，将禁止位清零并触发端点的“端点禁止”中断。

应用程序编程顺序：

1. 应用程序可以在任何同步 IN 端点上忽略 USBFS\_DIEPINTx 中的“Tx FIFO 为空时接收到 IN 令牌”中断，因为最终这将产生一个未完成同步 IN 传输中断（在 USBFS\_GINTSTS 中）。
2. 硬件触发未完成同步 IN 传输中断（在 USBFS\_GINTSTS 中）表示在至少一个同步 IN 端点上存在未完成的同步 IN 传输。
3. 应用程序必须读取所有同步 IN 端点的“端点控制”寄存器来检测存在未完成 IN 数据传输的端点。
4. 应用程序必须停止向与这些端点相关联的“周期性发送 FIFO”写入数据。
5. 对 USBFS\_DIEPCTLx 寄存器中的下列字段进行编程以禁止端点：
  - USBFS\_DIEPCTLx 中的 SNAK=1
  - USBFS\_DIEPCTLx 中的 EPDIS=1
6. 硬件触发 USBFS\_DIEPINTx 中的“端点禁止”中断表示模块已禁止该端点。
  - 此时，应用程序必须清空相关联的发送 FIFO 中的数据，或者通过在下一帧中使能新传输的端点来覆盖 FIFO 中的现有数据。要刷新数据，应用程序必须使用 USBFS\_GRSTCTL 寄存器。

## 停止非同步 IN 端点

本节介绍应用程序如何才能停止非同步端点。

应用程序编程顺序：

1. 禁止要停止的 IN 端点。同时将 STALL 位置 1。
2. USBFS\_DIEPCTLx 中的 EPDIS=1（当端点已使能时）
  - USBFS\_DIEPCTLx 中的 STALL=1
  - STALL 位的优先级始终高于 NAK 位
3. 硬件触发“端点禁止”中断（在 USBFS\_DIEPINTx 中）可以让应用程序知道模块已禁止指定端点。

4. 应用程序必须根据端点类型清空非周期性或周期性发送 FIFO。对于非周期性端点，应用程序必须重新使能另一个无需停止的非周期性端点来发送数据。
5. 当应用程序准备好结束该端点的 STALL 握手信号时，必须将 USBFS\_DIEPCTLx 的 STALL 位清零。
6. 如果应用程序因收到来自主机的 SetFeature.Endpoint Halt 命令或 ClearFeature.Endpoint Halt 命令来设置或清除端点的 STALL 状态，则必须在该控制端点的状态阶段传输之前将 STALL 位置 1 或清零。

特例：停止控制 OUT 端点

如果在控制传输的数据阶段，主机发送的 IN/OUT 令牌数超过 SETUP 数据包指定的值，则模块必须对这些多余的 IN/OUT 令牌回复 STALL。在这种情况下，应用程序必须在控制传输的数据阶段使能 USBFS\_DIEPINTx 的 ITTXFE 中断和 USBFS\_DOEPINTx 的 OTEPDIS 中断（当模块已完成传输 SETUP 数据包指定的数据量后）。随后，当应用程序收到此中断时，必须将相应端点控制寄存器中的 STALL 位置 1 并清除此中断。

## 34.7 寄存器说明

应用程序通过 AHB 从接口对控制和状态寄存器进行读写操作，以此来控制 USBFS 模块。USBFS 模块所有寄存器为 32 位寄存器，其地址按 32 位对齐，因此只能以 32 位的方式访问。

控制和状态寄存器分为以下几类：

- USBFS 系统控制寄存器
- 模块全局寄存器
- 主机模式寄存器
- 设备模式寄存器
- 电源和时钟门控控制寄存器
- 数据 FIFO(DFIFO)访问寄存器

其中 USBFS 系统控制寄存器与其它寄存器基址不同，该寄存器独立于 USBFS 模块以控制 USBFS 模块的相关设定。

只有模块全局寄存器、电源和时钟门控控制寄存器和数据 FIFO 访问寄存器可以在主机和设备模式下进行访问。当 USBFS 模块处以一种模式（主机或者设备）下，应用程序不得以另外一种角色模式访问寄存器，比如主机模式时，访问设备模式的寄存器。如果发生了非法访问，将会产生模式不匹配中断并在模块中断寄存器 USBFS.GINTSTS.NMIS 位反映。当模块从一种角色模式切换到另一种角色模式时，新工作模式下的寄存器必须重新编程为上电复位后的状态。

## 控制状态寄存器存储器映射

主机和设备模式寄存器占用不同的地址。所有寄存器均在 AHB 时钟域内实现。

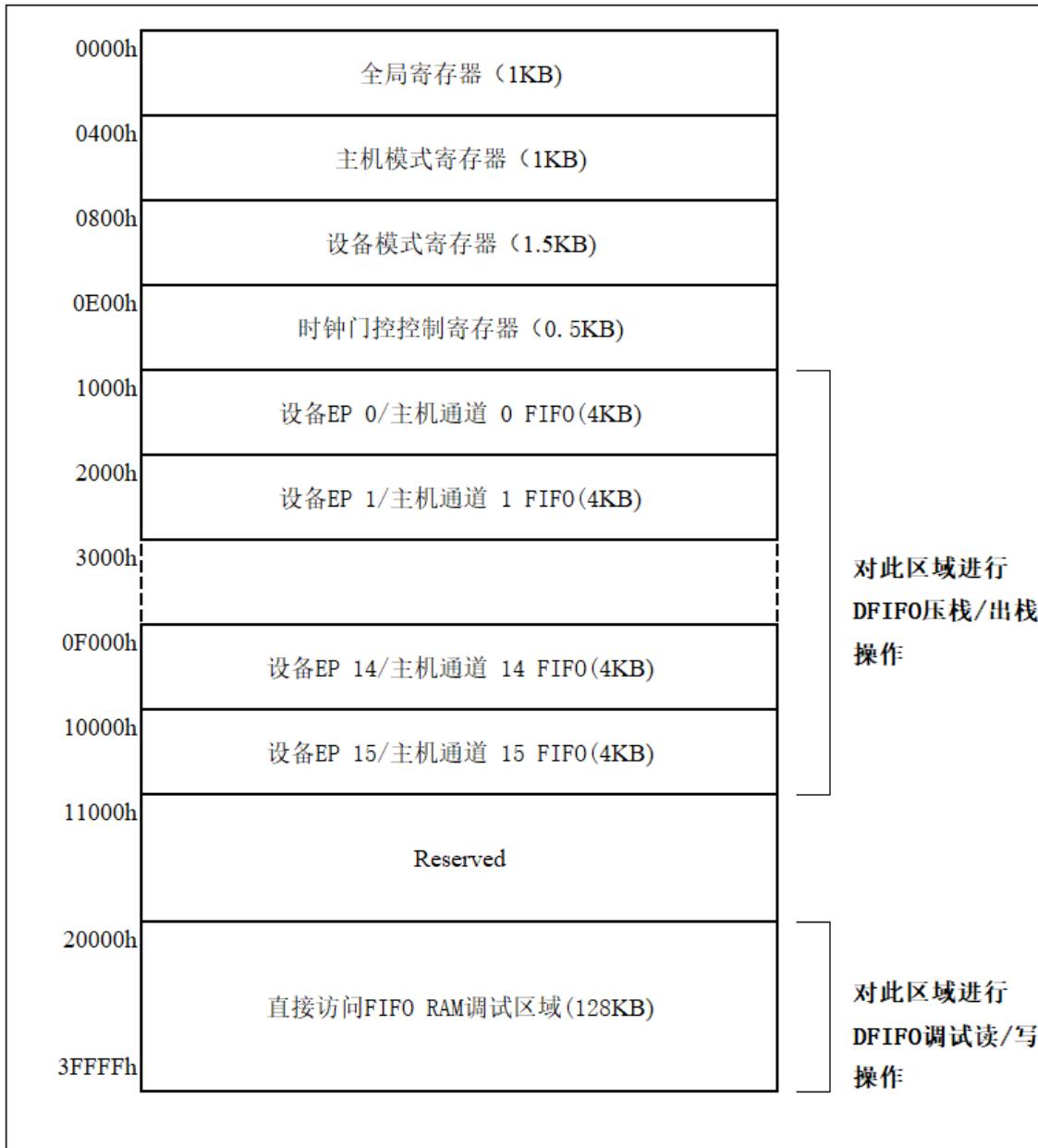


图 34-7 USBFS 控制状态寄存器存储器映射

USBFS 模块寄存器列表以及基址请参考表 34-3~表 34-4 USBFS 寄存器一览表。

表 34-3 USBFS 系统控制寄存器一览表

USB 系统控制寄存器基址: 0x40055400

(USBFS系统控制寄存器) 寄存器名	偏移地址	复位值
USB系统控制寄存器(USB_SYCTLREG)	0x00	0x0000 0000

表 34-4 USBFS 时钟门控控制寄存器一览表

USBFS 模块寄存器基址: 0x40080000

(USBFS全局寄存器) 寄存器名	偏移地址	复位值
USBFS VBUS控制寄存器(USBFS_GVBUSCFG)	0x00	0x00XX 0000
USBFS AHB控制寄存器(USBFS_GAHBCFG)	0x08	0x0000 0000
USBFS USB配置寄存器(USBFS_GUSBCFG)	0x0c	0x0000 1440
USBFS复位寄存器(USBFS_GRSTCTL)	0x10	0x8000 0000
USBFS全局中断状态寄存器(USBFS_GINTSTS)	0x14	0x1400 0020
USBFS全局中断屏蔽寄存器(USBFS_GINTMSK)	0x18	0x0000 0000
USBFS接收状态调试读取寄存器(USBFS_GRXSTSR)	0x1c	0x0000 0000
USBFS状态读取和出栈寄存器(USBFS_GRXSTSP)	0x20	0x0000 0000
USBFS接收FIFO大小寄存器(USBFS_GRXFSIZ)	0x24	0x0000 0280
USBFS 主机非周期性发送FIFO大小寄存器(USBFS_HNPTXFSIZ)/设备端点0发送FIFO大小寄存器(USBFS_DIEPRXF0)	0x28	0x0280 0280
USBFS非周期性发送FIFO/队列状态寄存器(USBFS_HNPTXSTS)	0x2c	0x0008 0280
USBFS模块ID寄存器(USBFS_CID)	0x3c	0x1234 5678
USBFS LPM配置寄存器(USBFS_GLPMCFG)	0x54	0x0000 0000
USBFS周期性发送FIFO大小寄存器(USBFS_HPTXFSIZ)	0x100	0x0280 0500
USBFS设备IN端点x发送FIFO大小寄存器(USBFS_DIEPTXFx)	0x100+x*4(x=1~5)	0x0280 0500+(x-1)*0x280

(USBFS主机控制和状态寄存器) 寄存器名	偏移地址	复位值
USBFS主机配置寄存器(USBFS_HCFG)	0x400	0x0000 0200
USBFS主机帧时间间隔寄存器(USBFS_HFIR)	0x404	0x0000 EA60
USBFS主机帧编号/帧剩余时间间隔寄存器(USBFS_HFNUM)	0x408	0x0000 3FFF
USBFS主机周期性发送FIFO/队列状态寄存器(USBFS_HPTXSTS)	0x410	0x0008 0280
USBFS主机全体通道中断寄存器(USBFS_HAINT)	0x414	0x0000 0000
USBFS主机全体通道中断屏蔽寄存器(USBFS_HAINTMSK)	0x418	0x0000 0000
USBFS主机端口控制和状态寄存器(USBFS_HPRT)	0x440	0x0000 0000
USBFS主机通道x特性寄存器(USBFS_HCCHARx)	0x500+x*0x20(x=0~5)	0x0000 0000
USBFS主机通道x中断寄存器(USBFS_HCINTx)	0x508+x*0x20(x=0~5)	0x0000 0000
USBFS主机通道x中断屏蔽寄存器(USBFS_HCINTx)	0x50c+x*0x20(x=0~5)	0x0000 0000
USBFS主机通道x传输大小寄存器(USBFS_HCTSIZx)	0x510+x*0x20(x=0~5)	0x0000 0000
USBFS主机通道x DMA地址寄存器(USBFS_HCDMAx)	0x514+x*0x20(x=0~5)	0xFFFF XXXX

(USBFS设备控制和状态寄存器) 寄存器名	偏移地址	复位值
USBFS设备配置寄存器(USBFS_DCFG)	0x800	0x0820 0000
USBFS设备控制寄存器(USBFS_DCTL)	0x804	0x0000 0002
USBFS设备状态寄存器(USBFS_DSTS)	0x808	0x0000 0002
USBFS设备IN端点通用中断屏蔽寄存(USBFS_DIEPMSK)	0x810	0x0000 0000
USBFS设备OUT端点通用中断屏蔽寄存器(USBFS_DOEPMSK)	0x814	0x0000 0000
USBFS设备全体端点中断寄存器(USBFS_DAINT)	0x818	0x0000 0000
USBFS设备全体端点中断屏蔽寄存(USBFS_DAINTMSK)	0x81c	0x0000 0000
USBFS设备IN端点FIFO空中断屏蔽寄存器(USBFS_DIEPEMPMSK)	0x834	0x0000 0000
USBFS设备IN端点0控制寄存器(USBFS_DIEPCTL0)	0x900	0x0000 8000
USBFS设备IN端点x控制寄存器(USBFS_DIEPCTLx)	0x900+x*0x20(x=1~5)	0x0000 0000
USBFS设备IN端点x中断寄存器(USBFS_DIEPINTx)	0x908+x*0x20(x=0~5)	0x0000 0080
USBFS设备IN端点x传输大小寄存器(USBFS_DIEPTSIZx)	0x910+x*0x20(x=0~5)	0x0000 0000
USBFS设备IN端点x DMA地址寄存(USBFS_DIEPDMAx)	0x914+x*0x20(x=0~5)	0x0000 0000
USBFS设备IN端点x发送FIFO状态寄存器(USBFS_DTXFSTSx)	0x918+x*0x20(x=0~5)	0x0000 0280
USBFS设备OUT端点0控制寄存器(USBFS_DOEPCTL0)	0xb00	0x0000 8000
USBFS设备OUT端点x控制寄存器(USBFS_DOEPCTLx)	0xb00+x*0x20(x=1~5)	0x0000 0000
USBFS设备OUT端点x中断寄存器(USBFS_DOEPINTx)	0xb08+x*0x20(x=0~5)	0x0000 0000
USBFS设备OUT端点x传输大小寄存(USBFS_DOEPSIZx)	0xb10+x*0x20(x=0~5)	0x0000 0000
USBFS设备OUT端点x DMA地址寄存器(USBFS_DOEPDMAx)	0xb14+x*0x20(x=0~5)	0xXXXXXXXX

(USBFS电源和时钟门控控制寄存器) 寄存器名	偏移地址	复位值
USBFS时钟门控控制寄存器(USBFS_GCCTL)	0xe00	0x0000 0000

### 34.7.1 USBFS 系统控制寄存器

#### 34.7.1.1 USB 系统控制寄存器(USB\_SYCTLREG)

USB System Control Register

偏移地址：0x00

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		
-																	
位	标记	位名	功能													读写	
b31~b19	Reserved	-	必须保持复位值。													R/W	
b18	USBFS_NFE	USBFS滤波使能寄存器	USBFS滤波使能寄存器 该寄存器用于控制STOP模式下，USBFS片上全速PHY DP/DM模拟滤波器的开关													R/W	
			0: 模拟滤波器关 1: 模拟滤波器开，滤波范围请参考USBFS_NFS[1:0]的设定														
b17~b16	USBFS_NFS	USBFS滤波选择寄存器	USBFS滤波选择寄存器 该寄存器用于控制STOP模式下，USBFS片上全速PHY DP/DM模拟滤波器的滤波范围													R/W	
			00b: 滤波宽度档位1 01b: 滤波宽度档位2 10b: 滤波宽度档位3 11b: 滤波宽度档位4 各档位具体值请参考“49.电气特性 49.3.34 USB片上全速PHY STOP模式下滤波特性”章节。														
b15~b2	Reserved	-	必须保持复位值。													R/W	
b1	USBFS_SOFEN	USBFS SOF脉冲输出使能位	USBFS主机发出SOF或者设备成功接收到SOF时，16个系统时钟周期宽度的SOF脉冲从PAD输出使能 0: SOF脉冲不输出 1: SOF脉冲输出 注意：在设备模式和主机模式均可访问。													R/W	
b0	USBFS_DFB	USBFS VBUS/ID管脚内部去抖滤波器旁路使能位	USBFS VBUS/ID管脚模块内部去抖动滤波器旁路使能位 0: 模块内部去抖动滤波器有效 1: 旁路模块内部去抖动滤波器 注意：在设备模式和主机模式均可访问。													R/W	

## 34.7.2 USBFS 全局寄存器

这些寄存器在主机模式和设备模式下都可用，且在这两个模式间切换时无需对其进行重新编程。除非特别说明，否则寄存器描述中的位值以二进制表示。

### 34.7.2.1 USBFS VBUS 控制寄存器(USBFS\_GVBUSCFG)

VBUS Configuration Register

偏移地址：0x00

复位值：0x00XX 0000

该寄存器可用于设定 VBUS 值从而忽略 VBUS 管脚的状态。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16				
Reserved																			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0				
Reserved																			
<table border="1" style="width: 100%;"><tr><td style="width: 100px; padding: 2px;">Reserved</td><td style="width: 10px; padding: 2px;">VBU SVAL</td><td style="width: 10px; padding: 2px;">VBU SOVE N</td><td style="width: 100px; padding: 2px;">Reserved</td></tr></table>																Reserved	VBU SVAL	VBU SOVE N	Reserved
Reserved	VBU SVAL	VBU SOVE N	Reserved																
位	标记	位名	功能	读写															
b31~b8	Reserved	-	必须保持复位值。	R/W															
b7	VBUSVAL	VBUS值	VBUS值(VBUS Value) 用于设定USBFS的VBUS值，当设定为1，且VBUSOVEN设定1后对USBFS完成上电。 注意：仅可在设备模式下访问。	R/W															
b6	VBUSOVEN	VBUS Override 使能	VBUS Override 使能(VBUS Override 使能) 用于将VBUSVAL设定的值反映到USBFS CORE的状态。仅当该位设置为1，VBUSVAL的值才有效。 注意：仅可在设备模式下访问。	R/W															
b5~b0	Reserved	-	必须保持复位值。	R/W															

**34.7.2.2 USBFS AHB 控制寄存器(USBFS\_GAHBCFG)**

AHB Configuration Register

偏移地址：0x08

复位值：0x0000 0000

该寄存器可用于在上电后或更改角色模式时对模块进行配置。该寄存器主要包含 AHB 系统相关的配置参数。

应用程序必须在开始任何 AHB 或 USB 事务前对该寄存器进行编程。请勿在初始编程后更改该寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16						
Reserved																					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0						
Reserved																					
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">PTXF ELVL</td><td style="padding: 2px;">TXFE LVL</td><td style="padding: 2px;">Rese rvred</td><td style="padding: 2px;">DMA EN</td><td style="padding: 2px;">HBSTLEN[3:0]</td><td style="padding: 2px;">GINT MSK</td></tr> </table>																PTXF ELVL	TXFE LVL	Rese rvred	DMA EN	HBSTLEN[3:0]	GINT MSK
PTXF ELVL	TXFE LVL	Rese rvred	DMA EN	HBSTLEN[3:0]	GINT MSK																
位	标记	位名	功能	读写																	
b31~b9	Reserved	-	必须保持复位值。	R/W																	
b8	PTXFELVL	周期性Tx FIFO空门限	周期性Tx FIFO空门限 (Periodic Tx FIFO empty level) 指示何时触发模块中断寄存器中的周期性Tx FIFO空中断位 (USBFS_GINTSTS 中的PTXFE 位)。 0: PTXFE (位于USBFS_GINTSTS) 中断指示周期性Tx FIFO为半空状态 1: PTXFE (位于USBFS_GINTSTS) 中断指示周期性Tx FIFO为全空状态 注意: 仅可在主机模式下访问。	R/W																	
b7	TXFELVL	设备Tx FIFO空门限	设备Tx FIFO空门限 (Tx FIFO empty level) 在设备模式下, 该位指示何时触发IN端点发送FIFO空中断 (USBFS_DIEPINTx 中的TXFE)。 0: TXFE (位于USBFS_DIEPINTx) 中断指示IN端点Tx FIFO为半空状态 1: TXFE (位于USBFS_DIEPINTx) 中断指示IN端点Tx FIFO为全空状态 注意: 仅可在设备模式下访问。	R/W																	
b6	Reserved	-	必须保持复位值。	R/W																	
b5	DMAEN	DMA 使能	DMA 使能 (DMA enable) 0: 模块以从模式运行 1: 模块以DMA模式运行 注意: 在设备模式和主机模式均可访问。	R/W																	
b4~b1	HBSTLEN	批量长度/类型	批量长度/类型 (Burst length/type) 0000b:单次 0001b: INCR 0011b:INCR4 0101b:INCR8 0111b:INCR16 其它值: 保留 注意: 在设备模式和主机模式均可访问。	R/W																	
b0	GINTMSK	全局中断屏蔽	全局中断屏蔽 (Global interrupt mask) 该位用于屏蔽全局中断或对全局中断取消屏蔽。中断状态寄存器由模块进行更新, 与此位的设置无关。 0: 屏蔽应用程序触发的中断	R/W																	

---

1: 取消对应用程序触发的中断的屏蔽

注意：在设备模式和主机模式均可访问。

---

**34.7.2.3 USBFS USB 配置寄存器 (USBFS\_GUSBCFG)**

USBFS USB configuration register

偏移地址：0x00C

复位值：0x0000 01440

该寄存器可用于在上电或更改角色模式后对模块进行配置。其中包含与 USB 和 USB-PHY 相关的配置参数。

应用程序必须在开始任何 AHB 或 USB 事务前对该寄存器进行编程。请勿在初始编程后更改该寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16		
Reserved	FDMOD	FHMOD	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		
Reserved		TRDT[3:0]				Reserved			PHYSEL	Reserved			TOCAL[2:0]				

位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R/W
b30	FDMOD	强制设备模式	<p>强制设备模式 (Force device mode) 向该位写入1时，可将模块强制为设备模式，忽略USBFS_ID输入引脚的状态。</p> <p>0: 正常模式，取决USBFS_ID管脚的输入状态 1: 强制设备模式 将强制位置1后，应用程序必须等待至少25 ms后更改方可生效。 注意：在设备模式和主机模式均可访问。</p>	R/W
b29	FHMOD	强制主机模式	<p>强制主机模式 (Force host mode) 向该位写入1时，可将模块强制为主机模式，忽略USBFS_ID输入引脚的状态。</p> <p>0: 正常模式，取决USBFS_ID管脚的输入状态 1: 强制主机模式 将强制位置1后，应用程序必须等待至少25 ms后更改方可生效。 注意：在设备模式和主机模式均可访问。</p>	R/W
b28~b14	Reserved	-	必须保持复位值。	R/W
b13~b10	TRDT	USB周转时间	<p>USB周转时间 (USB turnaround time) 以PHY时钟数为单位设置周转时间。 要计算TRDT的值，请使用如下公式： <math>TRDT = 4 \times AHB\text{ 时钟} + 1\text{ 个PHY时钟}</math> 例如： 1. 如果AHB时钟频率 = 84 MHz (PHY时钟频率 = 48 MHz)，则TRDT设置为9。 2. 如果AHB时钟频率 = 48 MHz (PHY时钟频率 = 48 MHz)，则 TRDT设置为5。 注意：仅可在设备模式下访问。</p>	R/W
b9~b7	Reserved	-	必须保持复位值。	R/W
b6	PHYSEL	全速串行收发器选择	全速串行收发器选择 (Full Speed serial transceiver select)	R/W

b5~b3	Reserved	-	必须保持复位值。	R/W
			FS 超时校准 (FS timeout calibration) PHY引入的额外延迟包括应用程序在该字段中设置的PHY时钟数，以及模块的全速数据包间超时间隔。不同PHY引入的延迟对数据线状态的影响是不同的。 全速操作的USB标准超时值为16到18（含）个位时间。应用程序必须根据枚举速度编程该字段。每个PHY时钟增加的位时间数为0.25个位时间。 注意：在设备模式和主机模式均可访问。	
b2~b0	TOCAL	FS 超时校准		R/W

**34.7.2.4 USBFS 复位寄存器 (USBFS\_GRSTCTL)**

USBFS reset register

偏移地址：0x10

复位值：0x8000 0000

应用程序通过此寄存器复位模块中的各项硬件特性。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16		
AHBI DL	DMA REQ	CSftR stDon e	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		
Reserved					TXFNUM[4:0]					TXFF LSH	RXFF LSH	Rese rved	FCRS T	HSR ST	CSRS T		

位	标记	位名	功能	读写
b31	AHBIDL	AHB主器件空闲	AHB主器件空闲(AHB master idle) 指示AHB主器件状态机处于空闲情况。 注意：在设备模式和主机模式均可访问。	R
b30	DMAREQ	AHB主器件空闲	DMA请求信号 (DMA request signal) 该位指示DMA请求正在进行中。用于调试。 注意：在设备模式和主机模式均可访问。	R
b29	CSftRstDone	-	复位完成信号 1: 当前所有逻辑复位完成 0: 无复位 注意：此为需要写1清0，即当复位完成后需要通过写'1'去清除复位完成信号	R/W
b28~b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10~b6	TXFNUM	TxFIFO编号	TxFIFO编号(TxFIFO number) 使用TxFIFO刷新位进行FIFO刷新的FIFO编号。只有在模块将TxFIFO刷新位清零后，方可更改此字段。 <ul style="list-style-type: none"> <li>● 00000:</li> <li>— 主机模式下刷新非周期性TxFIFO</li> <li>— 设备模式下刷新Tx FIFO 0</li> </ul> <ul style="list-style-type: none"> <li>● 00001:</li> <li>— 主机模式下刷新周期性TxFIFO</li> <li>— 设备模式下刷新TxFIFO 1</li> </ul> <ul style="list-style-type: none"> <li>● 00010: 设备模式下刷新TxFIFO 2</li> <li>...</li> <li>● 00101: 设备模式下刷新TxFIFO 15</li> <li>● 10000: 在设备模式或主机模式下刷新所有的发送FIFO</li> </ul> 注意：在设备模式和主机模式均可访问。	R/W
b5	TXFFLSH	TxFIFO刷新	TxFIFO刷新 (TxFIFO flush) 此位选择性地刷新一个或所有的发送FIFO，但当模块处理通信事务时无法执行该操作。 只有在确认模块当前未对TxFIFO 执行读写操作后，应用程序方可对此位执行写操作。使用以下寄存器进行确认： <ul style="list-style-type: none"> <li>— 读：NAK有效中断可确保模块当前未对FIFO执行读操作</li> </ul>	R/W

			— 写：USBFS_GRSTCTL中的 AHBIDL位可确保模块当前未对FIFO执行任何写操作 注意：在设备模式和主机模式均可访问。	
b4	RXFFLSH	RxFIFO刷新	RxFIFO刷新(RxFIFO flush) 应用程序可使用此位刷新整个RxFIFO，但必须首先确保模块当前未在处理通信事务。只有在确认模块当前未对RxFIFO执行读写操作后，应用程序方可对此位执行写操作。 应用程序必须等到此位清零后，方可执行其它操作。通常需要等待8个时钟周期（以PHY或AHB时钟中最慢的为准）。 注意：在设备模式和主机模式均可访问。	R/W
b3	Reserved	-	必须保持复位值。	R/W
b2	FCRST	主机帧计数器复位	主机帧计数器复位(Host frame counter reset) 应用程序对该位执行写操作时，模块中的帧数计数器复位。帧计数器复位后，由模块发送的下一个SOF的帧号为 0。 注意：在设备模式和主机模式均可访问。	R/W
b1	HSRST	HCLK域逻辑软复位	HCLK域逻辑软复位 (HCLK soft reset) 应用程序使用此位来刷新 AHB 时钟域中的控制逻辑。仅复位 AHB 时钟域流水线。 FIFO 不通过此位来刷新。 遵照协议终止 AHB 上的事务后，AHB 时钟域中的所有状态机均复位至空闲状态。 AHB 时钟域状态机所使用的 CSR 控制位清零。 要清零该中断，需要将由 AHB 时钟域状态机生成并用于控制中断状态的状态屏蔽位清零。 由于中断状态位并未清零，因此应用程序可以获取在该位置 1 后所发生的所有模块事件的状态。 此位为自清零位，模块将在其中所有必要逻辑复位后将该位清零。该过程需要若干个时钟的时间，具体取决于模块的当前状态。 注意：在设备模式和主机模式均可访问。	R/W
b0	CSRST	模块软复位	模块软复位 (Core soft reset) 按如下所述将HCLK和PCLK域复位： 除以下各位外，将各个中断和所有CSR寄存器位清零： — USBFS_GCCTL中的GATEHCLK位 — USBFS_GCCTL中的STPPCLK位 — USBFS_HCFG 中的FSLSPCS位 — USBFS_DCFG 中的DSPD位 将所有模块状态机（AHB从器件除外）复位至空闲状态，并清空所有发送FIFO 和接收FIFO。 在AHB传输的最后数据阶段结束后，尽快终止AHB主器件上的所有事务。立即终止USB上的所有事务。 应用程序可在需要复位模块时随时对该位执行写操作。 应用程序通过读取本寄存器的第29位来判断当前的复位操作是否完成，当复位完成后应用程序需要同时清除软复位以及复位完成状态信号 软件复位通常在两种情况下使用，一是软件开发期间，二是用户动态更改以上所列USB配置寄存器中的PHY选择位后。用户更改PHY时，将为PHY选择相应的时钟并用于PHY域中。一旦选择了新的时钟，则必须复位PHY域，才能保证	R/W

---

正常运行。

注意：在设备模式和主机模式均可访问。

---

### 34.7.2.5 USBFS 全局中断状态寄存器 (USBFS\_GINTSTS)

USBFS interrupt status register

偏移地址：0x14

复位值：0x14000020

该寄存器用于在当前模式（设备模式或主机模式）下借助系统级别的事件来中断应用程序。

该寄存器中的某些位仅在主机模式下有效，而其它位则仅在设备模式下有效。此外，该寄存器还可指示当前模式。

FIFO 状态中断为只读；如果软件在处理这些中断期间对 FIFO 执行读写操作，则 FIFO 中断标志将自动清零。

使能中断位前，应用程序必须在初始化时将 USBFS\_GINTSTS 寄存器清零，才可以避免在初始化前产生任何中断。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
WKUINT	SESSREQINT	DISCINT	CIDSCHG	LPMINT	PTXF	HCINT	HPRTINT	Reserved	DATAFSUSP	IPXFRR/INCO	IISOIXFR	OEPINT	IEPIN	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EOPF	ISOODRP	ENUMDN	USB	USBSP	ESUSP	Reser	Reser	GONAKEFF	GINAKEFF	NPTXFE	RXFNE	SOF	Reser	MMIS	CMOD

位	标记	位名	功能	读写
b31	WKUINT	检测到恢复/远程唤醒中断	检测到恢复/远程唤醒中断 (Resume/remote wakeup detected interrupt) 在设备模式下，当USB总线上检测到恢复信号时，将触发该中断。 在主机模式下，当USB上检测到远程唤醒时，将触发该中断。 通过软件对该位写1清零。 注意：在设备模式和主机模式均可访问。	R/W
b30	SESSREQINT	SESSION中断请求	USB会话中断请求 0：当前没有会话 1：新的会话请求 注意：仅可在设备模式下访问。	R/W
b29	DISCINT	检测到断开连接中断	检测到断开连接中断 (Disconnect detected interrupt) 当检测到设备断开连接时触发该中断。 通过软件对该位写1清零。 注意：仅可在主机模式下访问。	R/W
b28	CIDSCHG	连接器ID线状态变化中断	连接器ID线状态更改(Connector ID status change) 当连接器ID线状态发生更改时，模块将该位置1。 通过软件对该位写1清零。 注意：在设备模式和主机模式均可访问。	R/W
b27	LPMINT	LPM中断	LPM中断 设备模式 设备收到正确LPM传输且有效应答时该位置1。	R/W

		主机模式	
		主机发送LPM后收到有效应答，或者超过设定的重试次数后该位置1。	
b26	PTXFE	周期性TxFIFO空中断	<p>周期性TxFIFO空中断 (Periodic TxFIFO empty interrupt)</p> <p>当周期性发送FIFO为半空或全空状态，且周期性请求队列中存在可写入至少一个条目的空间时，将触发该中断。该FIFO为半空状态还是全空状态由USBFS_GAHBCFG寄存器中的周期性TxFIFO空级别位 (USBFS_GAHBCFG中的PTXFELVL位) 决定。</p> <p>注意：仅可在主机模式下访问。</p>
b25	HCINT	主机通道中断	<p>主机通道中断 (Host channels interrupt)</p> <p>模块将该位置1时，指示模块中一个通道上存在挂起的中断（在主机模式下）。应用程序必须读取主机USBFS_HAINT寄存器，以确定发生中断的通道的准确编号，然后读取相应的USBFS_HCINTx 寄存器，以确定引发中断的确切原因。应用程序必须先将USBFS_HCINTx寄存器的相应状态位清零，之后才能将该位清零。</p> <p>注意：仅可在主机模式下访问。</p>
b24	HPRTINT	主机端口中断	<p>主机端口中断 (Host port interrupt)</p> <p>模块将该位置1时，指示主机模式下 USBFS 控制器端口的状态发生变化。</p> <p>应用程序必须读取USBFS_HPRT寄存器，以确定引发此中断的确切事件。应用程序必须先将USBFS_HPRT寄存器的相应状态位清零，之后才能将该位清零。</p> <p>注意：仅可在主机模式下访问。</p>
b23	Reserved	-	必须保持复位值。
b22	DATAFSUSP	数据获取挂起	<p>数据获取挂起 (Data fetch suspended)</p> <p>该中断仅在DMA模式下有效。该中断指示，模块因TxFIFO空间或请求队列空间不可用而停止为IN端点获取数据。应用程序将该中断用于端点不匹配算法中。例如，在检测到端点不匹配后，应用程序将执行以下操作：</p> <ul style="list-style-type: none"><li>— 将全局非周期性IN NAK握手信号置 1</li><li>— 禁止IN端点</li><li>— 清空FIFO</li><li>— 根据IN令牌序列学习队列确定令牌序列</li><li>— 重新使能端点</li></ul> <p>— 如果全局非周期性IN NAK已清零但模块尚未为IN端点获取数据，同时又已接收到IN令牌，则清零全局非周期性IN NAK握手信号：模块将产生“FIFO为空时接收到IN令牌”中断。然后，USBFS将NAK响应发送到主机。为避免这种情况的发生，应用程序可以检查USBFS_GINTSTS 中的DATAFSUSP中断，该中断可确保在FIFO存满后再将全局 NAK 握手信号清零。或者，应用程序可以在将全局IN NAK握手信号清零时屏蔽“当 FIFO为空时接收到IN令牌中断。”</p> <p>通过软件对该位写1清零。</p> <p>注意：仅可在设备模式下访问。</p>
b21	IPXFR/ INCOMPISOOUT	未完成周期性传输/ 未完成OUT同步传输	<p>IPXFR:未完成周期性传输 (Incomplete periodic transfer)</p> <p>在主机模式下，如果存在仍处于挂起状态的未完成周期性事务，而这些事务计划在当前帧期间完成，则模块会将该中断位置1。</p> <p>通过软件对该位写1清零。</p> <p>注意：仅可在主机模式下访问。</p> <p>INCOMPISOOUT: 未完成OUT同步传输 (Incomplete isochronous OUT transfer)</p> <p>在设备模式下，模块将该中断置 1 时，指示当前帧中至少有一个同步OUT端</p>

			点上的传输未完成。该中断随该寄存器中的周期性帧结束中断 (EOPF) 位一同触发。 通过软件对该位写1清零。 注意：仅可在设备模式下访问。	
b20	IISOIXFR	未完成IN同步传输	未完成IN同步传输 (Incomplete isochronous IN transfer) 模块将该中断置1时，指示当前帧中至少有一个同步IN端点上的传输未完成。该中断随该寄存器中的周期性帧结束中断 (EOPF) 位一同触发。 通过软件对该位写1清零。 注意：仅可在设备模式下访问。	R/W
b19	OEPINT	OUT端点中断	OUT端点中断(OUT endpoint interrupt) 模块将该位置1时，指示模块中一个OUT端点上存在挂起的中断（在设备模式下）。应用程序必须读取主机USBFS_DAINT寄存器，以确定发生中断的OUT端点的准确编号，然后读取相应的USBFS_DOEPINTx寄存器，以确定引发中断的确切原因。应用程序必须先将相应USBFS_DOEPINTx 寄存器的相关状态位清零，之后才能将该位清零。 注意：仅可在设备模式下访问。	R
b18	IEPINT	IN端点中断	IN端点中断 (IN endpoint interrupt) 模块将该位置1时，指示模块中一个IN端点上存在挂起的中断（在设备模式下）。应用程序必须读取主机USBFS_DAINT寄存器，以确定发生中断的IN端点的准确编号，然后读取相应的USBFS_DIEPINTx寄存器，以确定引发中断的确切原因。应用程序必须先将相应USBFS_DIEPINTx 寄存器的相关状态位清零，之后才能将该位清零。 注意：仅可在设备模式下访问。	R
b17~b16	Reserved	-	必须保持复位值。	R/W
b15	EOPF	周期性帧结束中断	周期性帧结束中断 (End of periodic frame interrupt) 指示当前帧已达到USBFS_DCFG 寄存器中周期性帧间隔字段 (USBFS_DCFG 中的PFIVL位) 所指定的周期。 通过软件对该位写1清零。 注意：仅可在设备模式下访问。	R/W
b14	ISOODRP	丢弃同步OUT数据包中断	丢弃同步OUT数据包中断(Isochronous OUT packet dropped interrupt) 如果由于Rx FIFO空间不足，无法容纳同步OUT端点的最大数据包，从而导致模块无法向Rx FIFO写入同步OUT数据包，模块会将该位置1。 通过软件对该位写1清零。 注意：仅可在设备模式下访问。	R/W
b13	ENUMDNE	枚举完成中断	枚举完成中断 (Enumeration done interrupt) 模块将该位置1时，指示速度枚举已完成。应用程序必须读取USBFS_DSTS 寄存器来获取枚举速度。 通过软件对该位写1清零。 注意：仅可在设备模式下访问。	R/W
b12	USBRST	USB复位中断	USB复位中断(USB reset interrupt) 模块将该位置1时，指示在USB 上检测到复位信号。 通过软件对该位写1清零。 注意：仅可在设备模式下访问。	R/W
b11	USBSUSP	USB挂起中断	USB挂起中断(USB suspend interrupt) 模块将该位置1时，指示在USB上检测到挂起状态。当USB总线上的空闲状态保持3ms，模块便会进入挂起状态。通过软件对该位写1清零。 注意：仅可在设备模式下访问。	R/W
b10	ESUSP	早期挂起中断	早期挂起中断(Early suspend interrupt)	R/W

<p>模块将该位置1时，指示已检测到USB处于空闲状态的时间达到3ms。</p> <p>注意：仅可在设备模式下访问。</p>			
b9~b8	Reserved	-	必须保持复位值。 R/W
b7	GONAKEFF	全局OUT NAK有效中断	<p>全局OUT NAK有效中断(Global OUT NAK effective interrupt)</p> <p>指示USBFS_DCTL寄存器中由应用程序设置的“将全局OUT NAK置1”位 (USBFS_DCTL中的SGONAK位) 已在模块中生效。通过写入 USBFS_DCTL 寄存器中的“将全局OUT NAK清零”位 (USBFS_DCTL中的CGONAK位)，可将该位清零。</p> <p>注意：仅可在设备模式下访问。</p>
b6	GINAKEFF	全局非周期性IN NAK有效中断	<p>全局非周期性IN NAK有效中断(Global IN nonperiodic NAK effective interrupt)</p> <p>指示USBFS_DCTL寄存器中由应用程序设置的“将全局非周期性IN NAK置1”位 (USBFS_DCTL中的SGINAK位) 已在模块中生效。也就是说，模块已对应应用程序设置的全局IN NAK位进行采样，结果已生效。通过清零USBFS_DCTL 寄存器中的“将全局非周期性IN NAK清零”位 (USBFS_DCTL中的CGINAK位)，可将该位清零。此中断不一定表示USB上已发送了一个NAK握手信号。STALL位优先级高于NAK位。</p> <p>注意：仅可在设备模式下访问。</p>
b5	NPTXFE	非周期性TxFIFO空中断	<p>非周期性TxFIFO空中断(Non-periodic TxFIFO empty interrupt)</p> <p>当非周期性TxFIFO为半空或全空状态，且非周期性发送请求队列中至少存在可写入一个条目的空间时，将触发该中断。该 FIFO 为半空状态还是全空状态由USBFS_GAHBCFG寄存器中的非周期性TxFIFO空级别位 (USBFS_GAHBCFG中的TXFELVL位) 决定。</p> <p>注意：仅可在主机模式下访问。</p>
b4	RXFNE	RxFIFO非空中断	<p>RxFIFO非空中断(RxFIFO non-empty interrupt)</p> <p>指示RxFIFO中至少有一个数据包等待读取。 R</p> <p>注意：在主机模式和设备模式均可访问。</p>
b3	SOF	帧起始中断	<p>帧起始中断 (Start of frame interrupt)</p> <p>在主机模式下，模块将该位置1时，指示 USB 上已发送一个 SOF (FS) 或 Keep-Alive (LS)信号。应用程序必须将此位置1才可清除该中断。</p> <p>在设备模式下，模块将该位置1时，指示 USB 上已接收到一个SOF令牌。应 用程序可通过读取设备状态寄存器来获得当前的帧编号。只有在模块以 FS 模式运行时，才会出现此中断。</p> <p>通过软件对该位写1清零。</p> <p>注意：在主机模式和设备模式均可访问。 R/W</p>
b2	Reserved	-	必须保持复位值。 R/W
b1	MMIS	模式不匹配中断	<p>模式不匹配中断(Mode mismatch interrupt)</p> <p>当应用程序尝试访问以下寄存器时，模块将该位置1：</p> <ul style="list-style-type: none"> <li>— 模块运行在设备模式下访问主机模式寄存器</li> <li>— 模块运行在主机模式下访问设备模式寄存器</li> </ul> <p>寄存器访问在AHB上以OKAY响应结束，但该访问在内部被模块忽略并且 不会影响模块运行。</p> <p>通过软件对该位写1清零。</p> <p>注意：在主机模式和设备模式均可访问。 R/W</p>
b0	CMOD	当前工作模式	<p>当前工作模式 (Current mode of operation)</p> <p>指示当前模式。</p> <p>0: 设备模式 1: 主机模式</p>

---

注意：在主机模式和设备模式均可访问。

---

**34.7.2.6 USBFS 全局中断屏蔽寄存器 (USBFS\_GINTMSK)**

USBFS interrupt mask register

偏移地址：0x18

复位值：0x00000000

该寄存器与模块中断寄存器结合使用，以中断应用程序。如果将某个中断位屏蔽，则不会产生与该位相关的中断。

但是，与该中断相对应的模块中断 (USBFS\_GINTSTS) 寄存器位仍会置 1。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
WKUI M	SESS REQI NTMS K	DISCI M	CIDS CHG M	LPMI NTM	PTXF EM	HCIM	HPRT IM	Rese rved	DATA FSUS PM	IPXF RM/ INCO MPIS OCU TM	IISOI XFRM	OEPI M	IEPI M	Rese rved	Rese rved
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EOPF M	ISOO DRPM	ENU MDN EM	USB RSTM	USBS USPM	ESUS PM	Rese rved	Rese rved	GON AKEF FM	GINA KEFF M	NPT XFEM	RXF NEM	SOF M	Rese rved	MMIS M	Rese rved

位	标记	位名	功能	读写
b31	WKUIM	检测到恢复/远程唤醒中断屏蔽	检测到恢复/远程唤醒中断屏蔽 (Resume/remote wakeup detected interrupt mask)  0: 屏蔽中断 1: 使能中断  注意：在主机模式和设备模式均可访问。	R/W
b30	SESSREQINT MSK	会话请求中断屏蔽	会话请求中断屏蔽 (SESSION REQ interrupt mask)  0: 屏蔽中断 1: 使能中断  注意：仅可在设备模式下访问。	R/W
b29	DISCM	检测到断开连接中断屏蔽	检测到断开连接中断屏蔽 (Disconnect detected interrupt mask)  0: 屏蔽中断 1: 使能中断  注意：仅可在主机模式下访问。	R/W
b28	CIDSCHGM	中断连接器ID线状态变化中断 屏蔽	连接器ID线状态更改中断屏蔽(Connector ID status change interrupt mask)  0: 屏蔽中断 1: 使能中断  注意：在设备模式和主机模式均可访问。	R/W
b27	LPMINTM	LPM中断屏蔽	LPM中断屏蔽 (LPM interrupt mask)  0: 屏蔽中断 1: 使能中断	R/W
b26	PTXFEM	周期性Tx FIFO空中断屏蔽	周期性Tx FIFO空中断屏蔽 (Periodic Tx FIFO empty interrupt mask)  0: 屏蔽中断 1: 使能中断  注意：仅可在主机模式下访问。	R/W
b25	HCIM	主机通道中断屏蔽	主机通道中断屏蔽 (Host channels interrupt mask)	R/W

			0: 屏蔽中断 1: 使能中断  注意：仅可在主机模式下访问。	
b24	HPRTIM	主机端口中断屏蔽	主机端口中断屏蔽 (Host port interrupt mask) 0: 屏蔽中断 1: 使能中断  注意：仅可在主机模式下访问。	R/W
b23	Reserved	-	必须保持复位值。	R/W
b22	DATAFSUSPBM	数据获取挂起中断屏蔽	数据获取挂起中断屏蔽 (Data fetch suspended interrupt mask) 0: 屏蔽中断 1: 使能中断  注意：仅可在设备模式下访问。	R/W
b21	IPXFRM/ INCOMPISOOUT	未完成周期性传输中断屏蔽/ 未完成OUT同步传输中断屏蔽	IPXFR:未完成周期性传输中断屏蔽(Incomplete periodic transfer interrupt mask) 0: 屏蔽中断 1: 使能中断  注意：仅可在主机模式下访问。 INCOMPISOOUT: 未完成OUT同步传输中断屏蔽(Incomplete isochronous OUT transfer interrupt mask) 0: 屏蔽中断 1: 使能中断  注意：仅可在设备模式下访问。	R/W
b20	IISOIXFRM	未完成IN同步传输中断屏蔽	未完成IN同步传输中断屏蔽(Incomplete isochronous IN transfer interrupt mask) 0: 屏蔽中断 1: 使能中断  注意：仅可在设备模式下访问。	R/W
b19	OEPIM	OUT端点中断屏蔽	OUT端点中断屏蔽(OUT endpoint interrupt mask) 0: 屏蔽中断 1: 使能中断  注意：仅可在设备模式下访问。	R/W
b18	IEPIM	IN端点中断屏蔽	IN端点中断屏蔽(IN endpoint interrupt mask) 0: 屏蔽中断 1: 使能中断  注意：仅可在设备模式下访问。	R/W
b17~b16	Reserved	-	必须保持复位值。	R/W
b15	EOPFM	周期性帧结束中断屏蔽	周期性帧结束中断屏蔽(End of periodic frame interrupt mask) 0: 屏蔽中断 1: 使能中断  注意：仅可在设备模式下访问。	R/W
b14	ISOODRPM	丢弃同步OUT数据包中断屏蔽	丢弃同步OUT数据包中断屏蔽(Isochronous OUT packet dropped interrupt mask) 0: 屏蔽中断 1: 使能中断  注意：仅可在设备模式下访问。	R/W
b13	ENUMDNEM	枚举完成中断屏蔽	枚举完成中断屏蔽(Enumeration done interrupt mask) 0: 屏蔽中断 1: 使能中断  注意：仅可在设备模式下访问。	R/W

			USB复位中断屏蔽(USB reset interrupt mask)	
b12	USBRSTM	USB复位中断屏蔽	0: 屏蔽中断 1: 使能中断	R/W
			注意：仅可在设备模式下访问。	
			USB挂起中断屏蔽(USB suspend interrupt mask)	
b11	USBSUSPM	USB挂起中断屏蔽	0: 屏蔽中断 1: 使能中断	R/W
			注意：仅可在设备模式下访问。	
			早期挂起中断屏蔽(Early suspend interrupt mask)	
b10	ESUSPM	早期挂起中断屏蔽	0: 屏蔽中断 1: 使能中断	R/W
			注意：仅可在设备模式下访问。	
b9~b8	Reserved	-	必须保持复位值。	R/W
			全局OUT NAK有效中断屏蔽(Global OUT NAK effective interrupt mask)	
b7	GONAKEFFM	全局OUT NAK有效中断屏蔽	0: 屏蔽中断 1: 使能中断	R/W
			注意：仅可在设备模式下访问。	
			全局非周期性IN NAK有效中断屏蔽(Global IN nonperiodic NAK effective interrupt mask)	
b6	GINAKEFFM	全局非周期性IN NAK有效中断屏蔽	0: 屏蔽中断 1: 使能中断	R/W
			注意：仅可在设备模式下访问。	
			非周期性TxFIFO空中断(Non-periodic TxFIFO empty interrupt mask)	
b5	NPTXFEM	非周期性TxFIFO空中断屏蔽	0: 屏蔽中断 1: 使能中断	R/W
			注意：仅可在主机模式下访问。	
			RxFIFO非空中断屏蔽(RxFIFO non-empty interrupt mask)	
b4	RXFNEM	RxFIFO 非空中断屏蔽	0: 屏蔽中断 1: 使能中断	R/W
			注意：在主机模式和设备模式均可访问。	
			帧起始中断屏蔽(Start of frame interrupt mask)	
b3	SOFM	帧起始中断屏蔽	0: 屏蔽中断 1: 使能中断	R/W
			注意：在主机模式和设备模式均可访问。	
b2	Reserved	-	必须保持复位值。	R/W
			模式不匹配中断屏蔽_Mode mismatch interrupt mask)	
b1	MMISM	模式不匹配中断中断屏蔽	0: 屏蔽中断 1: 使能中断	R/W
			注意：在主机模式和设备模式均可访问。	
b0	Reserved	-	必须保持复位值。	R/W

### 34.7.2.7 USBFS 接收状态调试读取 /USBFS 状态读取和出栈寄存器 (USBFS\_GRXSTSR/USBFS\_GRXSTSP)

USBFS Receive status debug read/USBFS status read and pop registers

读取的偏移地址：0x01C

出栈的偏移地址：0x020

复位值：0x0000 0000

读取接收状态调试读取寄存器将返回接收 FIFO 顶部的内容。读取接收状态读取和出栈寄存器将额外弹出 RxFIFO 顶部的数据条目。接收状态内容在主机模式和设备模式下的解释不同。

当接收 FIFO 为空时，模块会忽略对该寄存器的读取或出栈操作，并返回值 0x0000 0000。当模块中断寄存器的接收 FIFO 非空位（USBFS\_GINTSTS 中的 RXFNE 位）置位时，应用程序必须仅弹出接收状态 FIFO。

#### 主机模式：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved										PKTSTS[3:0]			DPID [1]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DPID [0]	BCNT[10:0]										CHNUM[3:0]				
<hr/>															
位	标记	位名	功能	读写											
b31~b21	Reserved	-	必须保持复位值。	R/W											
<hr/>															
<hr/>															
b20~b17	PKTSTS	数据包状态	数据包状态 (Packet status) 指示接收的数据包的状态 0010: 接收到IN数据包 0011: IN传输完成 (触发中断) 0101: 数据同步错误 (触发中断) 0111: 暂停通道 (触发中断) 其他值: 保留	R											
<hr/>															
b16~b15	DPID	数据PID	数据PID (Data PID) 指示接收的数据包的数据PID 00: DATA0 10: DATA1 01: DATA2 11: MDATA	R											
<hr/>															
b14~b4	BCNT	字节计数	字节计数(Byte count) 指示接收的IN数据包的字节数。	R											
<hr/>															
b3~b0	CHNUM	通道编号	通道编号(Channel number) 指示当前接收的数据包所属的通道编号。	R											
<hr/>															

### 设备模式：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved										PKTSTS[3:0]			DPID[1]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DPID[0]	BCNT[11:0]										EPNUM[3:0]				
<hr/>															
位	标记	位名	功能	读写											
b31~b21	Reserved	-	必须保持复位值。	R/W											
<hr/>															
b20~b17	PKTSTS	数据包状态	数据包状态(Packet status) 指示接收的数据包的状态 0001: 全局OUT NAK (触发中断) 0010: 接收到OUT数据包 0011: OUT传输完成 (触发中断) 0100: SETUP事务完成 (触发中断) 0110: 接收到SETUP数据包 其它值: 保留	R											
<hr/>															
b16~b15	DPID	数据PID	数据PID(Data PID) 指示接收的OUT数据包的数据PID 00: DATA0 10: DATA1 01: DATA2 11: MDATA	R											
<hr/>															
b14~b4	BCNT	字节计数	字节计数(Byte count) 指示接收的数据包的字节数。	R											
<hr/>															
b3~b0	EPNUM	端点编号	端点编号(Endpoint number) 指示当前接收的数据包所属的端点编号。	R											

### 34.7.2.8 USBFS 接收 FIFO 大小寄存器 (USBFS\_GRXFSIZ)

USBFS Receive FIFO size register

偏移地址：0x024

复位值：0x0000 0280

此应用程序可以对必须分配给 RxFIFO 的 RAM 大小进行编程。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					RXFD[10:0]										
位	标记	位名	功能										读写		
b31~b11	Reserved	-	必须保持复位值。										R/W		
b10~b0	RXFD	RxFifo深度	RXFD: RxFIFO深度 (RxFIFO depth) 以32位字为单位。 最小值为16 最大值为256 上电复位值为最大Rx数据FIFO深度。										R/W		

### 34.7.2.9 USBFS 主机非周期性发送 FIFO 大小寄存器(USBFS\_HNPTXFSIZ)/端点 0 发送 FIFO 大小(USBFS\_DIEPTXFO)

USBFS Host non-periodic transmit FIFO size register/Device endpoint0 transmit FIFO size register

偏移地址：0x028

复位值：0x02800280

此应用程序可以对必须分配给 TxFIFO 的 RAM 大小进行编程。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
NPTXFD[15:0]/TX0FD[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
NPTXFSA[15:0]/TX0FSA[15:0]															
位	标记	位名	功能	读写											
b31~b16	NPTXFD/ TX0FD	非周期性TxFIFO 深度/端点0 TxFIFO深度	主机模式：NPTXFD 非周期性TxFIFO深度(Non-periodic TxFIFO depth) 以32位字为单位。 最小值为16 最大值为256	R/W											
			设备模式：TX0FD 端点0 TxFIFO深度(Endpoint 0 TxFIFO depth) 以32位字为单位。 最小值为16 最大值为256												
b15~b0	NPTXFSA/ TX0FSA	非周期性发送RAM起始地址/端点0发送RAM起始地址	主机模式：NPTXFSA 非周期性发送RAM起始地址(Non-periodic transmit RAM start address) 此字段包含非周期性发送FIFO RAM的存储器起始地址。	R/W											
			设备模式：TX0FSA 端点0发送RAM起始地址(Endpoint 0 transmit RAM start address) 此字段包含端点0发送FIFO RAM的存储器起始地址。												

### 34.7.2.10 USBFS 非周期性发送 FIFO/队列状态寄存器 (USBFS\_HNPTXSTS)

USBFS Host non-periodic transmit FIFO size register/Device endpoint0 transmit FIFO size register

偏移地址：0x02C

复位值：0x00080280

此只读寄存器包含非周期性 TxFIFO 和非周期性发送请求队列的自由空间信息。

此寄存器仅在主机模式有效，设备模式无效。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserve	NPTXQTOP[6:0]										NPTQXSAR[7:0]				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
NPTXFSAV[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31	Reserved	-	必须保持复位值。	R/W											
b30~b24	NPTXQTOP	非周期性发送请求队列顶部	非周期性发送请求队列顶部 (Top of the non-periodic transmit request queue)	R											
			非周期性发送请求队列中 MAC 目前正在处理的条目。 位30:27: 通道/端点编号(Channel/endpoint number) 位26:25: — 00: IN/OUT 令牌 — 01: 长度为零的发送数据包 — 11: 通道停止命令 位24: 结束 (所选通道/端点的最后一个条目) (Terminate (last entry for selected channel))												
b23~b16	NPTQXSAR	非周期性发送请求队列可用空间	非周期性发送请求队列可用空间 (Non-periodic transmit request queue space available) 指示非周期性发送请求队列中的可用空闲空间大小。 在主机模式下，此队列保存IN和OUT请求。 0: 非周期性发送请求队列已满 1: 1 个位置可用 2: 2 个位置可用 n: n 个位置可用 (其中, n范围: 0~8) 其它值: 保留	R											
			非周期性TxFIFO可用空间 (Non-periodic TxFIFO space available) 指示非周期性TxFIFO中的可用空闲空间大小。 以 32 位字为单位。 00: 非周期性 TxFIFO 已满 01: 1 个字可用 10: 2 个字可用 0xn: n 个字可用 (其中, n范围: 0~256) 其它值: 保留	R											
b15~b0	NPTXFSAV	非周期性TxFIFO可用空间													

### 34.7.2.11 USBFS 模块 ID 寄存器(USBFS\_CID)

USBFS core ID register

偏移地址：0x03C

复位值：0x12345678

该寄存器为可编程用户配置 ID 寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16										
PRODUCT_ID[31:16]																									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0										
PRODUCT_ID[15:0]																									
<table border="1" style="width: 100%;"><thead><tr><th>位</th><th>标记</th><th>位名</th><th>功能</th><th>读写</th></tr></thead><tbody><tr><td>b31~b0</td><td>PRODUCT_ID</td><td>产品ID字段</td><td>产品ID字段(Product ID field) 可通过应用程序编程的ID字段。</td><td>R/W</td></tr></tbody></table>																位	标记	位名	功能	读写	b31~b0	PRODUCT_ID	产品ID字段	产品ID字段(Product ID field) 可通过应用程序编程的ID字段。	R/W
位	标记	位名	功能	读写																					
b31~b0	PRODUCT_ID	产品ID字段	产品ID字段(Product ID field) 可通过应用程序编程的ID字段。	R/W																					

### 34.7.2.12 USBFS LPM 配置寄存器(USBFS\_GLPMCFG)

USBFS LPM configuration register

偏移地址：0x054

复位值：0x00000000

该寄存器为可编程 LPM 配置寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved		ENBS EL	LPMRCNTSTS[2:0]			SNDL PM	LPMRCNT[2:0]			LPMCHIDX[3:0]				L1RS MOK	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SLPS TS	LPMRSP[1:0]		L1DS EN	BSELTHRS[3:0]				L1SS EN	REM WAKE	BSEL[3:0]			LPM ACK	LPME N	

位	标记	位名	功能	读写
b31~b29	Reserved	-	必须保持复位值。	R/W
b28	ENBSEL	BSEL使能位	BSEL使能位 0: 遵循USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007 1: 遵循Errata for USB 2.0 ECN: Link Power Management (LPM) - 7/2007	R/W
b27~b25	LPMRCNTSTS[ 2:0]	LPM重试状态寄存器	LPM重试状态寄存器 记录LPM剩余的重试次数，仅主机模式有效	R
b24	SENDLPM	发送LPM传输	发送LPM传输 0: 不发送LPM传输 1: 发送LPM传输 (EXT 和 LPM 令牌) 当收到设备返回的应答 (ACK,STALL或NYET) 后该位自动清零。 仅主机模式支持。	R/W
b23~b21	LPMRCNT[2:0]	LPM重试次数寄存器	LPM重试次数寄存器 当收到ERROR应答时，主机发送LPM传输的重试次数,直到收到有效的应答 (ACK,STALL或NYET) 仅主机模式支持。	R/W
b20~b17	LPMCHIDX[2: 0]	LPM发送通道索引寄 存器	LPM发送通道索引寄存器 LPM传输所用主机通道索引，硬件自动插入该通道设定的设备地址以及端点信息。 仅主机模式支持。	R/W
b16	L1RSMOK	L1状态位	L1状态位 0: 当前状态不能从L1恢复 1: 当前状态可以从L1恢复	R
b15	SLPSTS	sleep状态位	sleep状态位 设备模式 当设备发送ACK响应LPM传输并经过协议规定的时间后，该位置1，表示进入 SLEEP模式。 当总线状态变化，或者断连，或者发送远程唤醒信号时，该位自动清0，退出 SLEEP模式。 主机模式 主机发送LPM传输，接收到设备响应的ACK应答后，该位置1，表示进入SLEEP 模式。	R

当主机接收到远程唤醒信号，或者主机发起唤醒，或者主机发起复位时，该位自动清0，退出SLEEP模式。 0: Not in L1 1: In L1			
LPM应答 主机接收或者从机发送的LPM应答 00b: ERROR(无应答) 01b: STALL 00b: NYET 00b: ACK			
b14~b13	LPMRSP[1:0]	LPM应答	R
L1 deep sleep使能位 最大限度的降低功耗，应用程序需要将此位设定为1。			
b12	L1DSEN	L1 deep sleep使能位	R
BSEL阈值设定寄存器 设备模式 当设备接收到BSEL值大于等于该寄存器设定时，进入L1。 主机模式 主机发送resume信号的时间。			
b11~b8	BSELTHRS[3:0]	BSEL阈值设定寄存器	R/W
L1 shallow sleep使能位 最大限度的降低功耗，应用程序需要将此位设定为1。			
b7	L1SSEN	L1 shallow sleep使能位	R/W
bRemoteWake值 主机发送或者设备接收到的bRemoteWake值 BSEL寄存器 设备模式 BSEL bmAttribute值，有效应答LPM传输后自动更新。 主机模式 主机发送LPM传输的BSEL值，或者resume信号长度值。			
b6	REMWAKE	bRemoteWake值	R/W
0000b: 125us 0001b: 150us 0010b: 200us 0100b: 400us 0101b: 500us 0110b: 1000us 0111b: 2000us 1000b: 3000us 1001b: 4000us 1010b: 5000us 1011b: 6000us 1100b: 7000us 1101b: 8000us 1110b: 9000us 1111b: 10000us			
b5~b2	BSEL[3:0]	BSEL寄存器	R/W
LPM应答寄存器 1: ACK 即使该位写1，但LPM传输有错误时，不发送ACK应答。 存在PID/CRC错误，ERROR应答 bLinkState不等于0001b，STALL应答			
b1	LPMACK	LPM应答寄存器	R/W

---

存在挂起的数据传输， NYET应答

0: NYET

仅设备模式有效

---

LPM使能位

b0

LPMEN

LPM使能位

0: LPM禁止

R/W

1: LPM使能

---

### 34.7.2.13 USBFS 主机周期性发送 FIFO 大小寄存器(USBFS\_HPTXFSIZ)

USBFS Host periodic transmit FIFO size register

偏移地址：0x100

复位值：0x02800500

此应用程序可以对必须分配给周期 TxFIFO 的 RAM 大小进行编程。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16															
PTXFD[15:0]																														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0															
PTXSA[15:0]																														
<table border="1"><thead><tr><th>位</th><th>标记</th><th>位名</th><th>功能</th><th>读写</th></tr></thead><tbody><tr><td>b31~b16</td><td>PTXFD</td><td>主机周期性Tx FIFO 深度</td><td>主机周期性Tx FIFO深度 (Host periodic Tx FIFO depth) 以32位字为单位。 最小值为16</td><td>R/W</td></tr><tr><td>b15~b0</td><td>PTXSA</td><td>主机周期性Tx FIFO 起始地址</td><td>主机周期性Tx FIFO起始地址(Host periodic Tx FIFO start address) 上电复位值是最大Rx FIFO深度与最大非周期性Tx FIFO深度之和。</td><td>R/W</td></tr></tbody></table>																位	标记	位名	功能	读写	b31~b16	PTXFD	主机周期性Tx FIFO 深度	主机周期性Tx FIFO深度 (Host periodic Tx FIFO depth) 以32位字为单位。 最小值为16	R/W	b15~b0	PTXSA	主机周期性Tx FIFO 起始地址	主机周期性Tx FIFO起始地址(Host periodic Tx FIFO start address) 上电复位值是最大Rx FIFO深度与最大非周期性Tx FIFO深度之和。	R/W
位	标记	位名	功能	读写																										
b31~b16	PTXFD	主机周期性Tx FIFO 深度	主机周期性Tx FIFO深度 (Host periodic Tx FIFO depth) 以32位字为单位。 最小值为16	R/W																										
b15~b0	PTXSA	主机周期性Tx FIFO 起始地址	主机周期性Tx FIFO起始地址(Host periodic Tx FIFO start address) 上电复位值是最大Rx FIFO深度与最大非周期性Tx FIFO深度之和。	R/W																										

### 34.7.2.14 USBFS 设备 IN 端点 x 发送 FIFO 大小寄存器 (USBFS\_DIEPTXF $x$ ) ( $x = 1..5$ )

USBFS device IN endpoint transmit FIFO size register

偏移地址：0x104+(x-1)\*0x4

复位值：0x02800500+(x-1)\*0x280

此应用程序可以对必须分配给设备 TxFIFO 的大小进行编程。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
INEPTXFD[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
INEPTXSA[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	INEPTXFD	IN端点Tx FIFO深度	设备IN端点Tx FIFO深度 (Device IN endpoint Tx FIFO depth) 以32位字为单位。 最小值为16	R/W											
b15~b0	INEPTXSA	IN端点Tx FIFO $x$ RAM起始地址	IN端点Tx FIFO $x$ RAM起始地址 (IN endpoint FIFO $x$ transmit RAM start address) 此字段包含IN端点发送FIFO $x$ 的存储器起始地址。 该地址必须与32位存储器位置对齐。	R/W											

### 34.7.3 USBFS 主机模式寄存器

主机模式寄存器会影响主机模式下的模块操作。在设备模式下不得访问主机模式寄存器，因为产生的结果不明确。

除非特别说明，否则寄存器描述中的位值以二进制表示。

#### 34.7.3.1 USBFS 主机配置寄存器 (USBFS\_HCFG)

USBFS Host configuration register

偏移地址：0x400

复位值：0x00000200

此寄存器将在上电后对模块进行配置。请勿在初始化主机后更改此寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved														FSLS S	FSLSPCS[1: 0]
位	标记	位名	功能	读写											
b31~b3	Reserved	-	必须保持复位值。	R/W											
b2	FSLSS	仅支持FS和LS	应用程序使用此位控制模块的枚举速度。使用此位，应用程序可使模块工作为FS主机，即使所连接的设备支持HS通信也是如此。请勿在初始编程后更改此字段。	R/W											
<p>1: 仅限 FS/LS，即使所连接设备可支持HS</p> <p>FS/LS PHY时钟选择 (FS/LS PHY clock select)</p> <p>当模块处于FS主机模式时</p> <p>01: PHY时钟以48 MHz运行</p> <p>其它值: 保留</p> <p>当模块处于LS主机模式时</p> <p>00: 保留</p> <p>01: 选择48MHz PHY时钟频率</p> <p>10: 选择6MHz PHY时钟频率</p> <p>11: 保留</p> <p>注意: 当设备连上主机时，必须依照所连接设备的速度设置 FSLSPCS (更改此位后，必须进行软件复位)。</p>															
b1~b0	FSLSPCS	FS/LS PHY时钟选择	00: 保留	R/W											
			01: 选择48MHz PHY时钟频率												
			10: 选择6MHz PHY时钟频率												
			11: 保留												

### 34.7.3.2 USBFS 主机帧时间间隔寄存器 (USBFS\_HFIR)

USBFS Host frame interval register

偏移地址：0x404

复位值：0x0000EA60

此寄存器用于存储 USBFS 控制器对已连接设备当前速度所设定的帧间隔信息。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16														
Reserved																													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0														
FRIVL[15:0]																													
<table border="1"><thead><tr><th>位</th><th>标记</th><th>位名</th><th>功能</th><th>读写</th></tr></thead><tbody><tr><td>b31~b26</td><td>Reserved</td><td>-</td><td>必须保持复位值</td><td>R/W</td></tr><tr><td>b15~b0</td><td>FRIVL</td><td>帧间隔</td><td><p>帧间隔 (Frame interval) 应用程序在此字段编程的值用于指定两个连续SOF(FS)或Keep-Alive令牌(LS)之间的时间间隔。此字段包含构成所需帧间隔的PHY时钟数。只有将主机端口控制和状态寄存器的端口使能位 (USBFS_HPRT的PENA位) 置1后，应用程序才能向此寄存器中写入值。如果未对值进行编程，模块将根据在主机配置寄存器的FS/LS PHY时钟选择字段 (USBFS_HCFG中的 FSLSPCS) 中指定的PHY时钟来计算。请勿在初始配置后更改此字段的值。 设定值=帧间隔 (ms) × (PHY 时钟频率) -1 注意：只要应用程序需要更改帧间隔时间，即可对FRIVL位进行修改。</p><p>R/W</p></td><td></td></tr></tbody></table>															位	标记	位名	功能	读写	b31~b26	Reserved	-	必须保持复位值	R/W	b15~b0	FRIVL	帧间隔	<p>帧间隔 (Frame interval) 应用程序在此字段编程的值用于指定两个连续SOF(FS)或Keep-Alive令牌(LS)之间的时间间隔。此字段包含构成所需帧间隔的PHY时钟数。只有将主机端口控制和状态寄存器的端口使能位 (USBFS_HPRT的PENA位) 置1后，应用程序才能向此寄存器中写入值。如果未对值进行编程，模块将根据在主机配置寄存器的FS/LS PHY时钟选择字段 (USBFS_HCFG中的 FSLSPCS) 中指定的PHY时钟来计算。请勿在初始配置后更改此字段的值。 设定值=帧间隔 (ms) × (PHY 时钟频率) -1 注意：只要应用程序需要更改帧间隔时间，即可对FRIVL位进行修改。</p> <p>R/W</p>	
位	标记	位名	功能	读写																									
b31~b26	Reserved	-	必须保持复位值	R/W																									
b15~b0	FRIVL	帧间隔	<p>帧间隔 (Frame interval) 应用程序在此字段编程的值用于指定两个连续SOF(FS)或Keep-Alive令牌(LS)之间的时间间隔。此字段包含构成所需帧间隔的PHY时钟数。只有将主机端口控制和状态寄存器的端口使能位 (USBFS_HPRT的PENA位) 置1后，应用程序才能向此寄存器中写入值。如果未对值进行编程，模块将根据在主机配置寄存器的FS/LS PHY时钟选择字段 (USBFS_HCFG中的 FSLSPCS) 中指定的PHY时钟来计算。请勿在初始配置后更改此字段的值。 设定值=帧间隔 (ms) × (PHY 时钟频率) -1 注意：只要应用程序需要更改帧间隔时间，即可对FRIVL位进行修改。</p> <p>R/W</p>																										

### 34.7.3.3 USBFS 主机帧编号/帧剩余时间寄存器 (USBFS\_HFNUM)

USBFS Host frame interval register

偏移地址：0x408

复位值：0x0000 3FFF

此寄存器用于指示当前帧编号。它还指示当前帧的剩余时间（以PHY时钟数为单位）。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
FTREM[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FRNUM[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	FTREM	帧剩余时间	帧剩余时间 (Frame time remaining) 指示当前帧的剩余时间（以PHY时钟数为单位）。每过去1个PHY时钟，此字段递减1。 当值达到零时，此字段将重新装载帧间隔寄存器中的值，并由模块在USB上发送一个新SOF。	R											
b15~b0	FRNUM	帧编号	帧编号 (Frame number) 当在USB上发送1个新SOF时此字段的值将递增1，当达到0x3FFF时会清零。	R											

**34.7.3.4 USBFS 主机周期性发送 FIFO/队列状态寄存器(USBFS\_HPTXSTS)**

USBFS Host periodic transmit FIFO/queue status register

偏移地址：0x410

复位值：0x00080280

此只读寄存器包含周期性 TxFIFO 和周期性发送请求队列的空闲空间信息。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PTXQTOP[7:0]								PTXQSAV[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PTXFSAVL[15:0]															

位	标记	位名	功能	读写
b31~b24	PTXQTOP	周期性发送请求队列顶部	周期性发送请求队列顶部 (Top of the periodic transmit request queue) 指示周期性Tx请求队列中MAC当前正在处理的项。 该寄存器用于调试。 位 31: 奇数/偶数帧 (Odd/Even frame) — 0: 以偶数帧发送 — 1: 以奇数帧发送 位 30:27: 通道/端点编号 (Channel number) 位 26:25: 类型 (Type) — 00: 输入/输出 — 01: 零长度数据包 — 11: 禁止通道命令 位 24: 结束 (所选通道的最后一个条目) (Terminate (last entry for the selected channel))	R
b23~b16	PTXQSAV	周期性发送请求队列可用空间	周期性发送请求队列可用空间 (Periodic transmit request queue space available) 指示可供写入的周期性发送请求队列的空闲位置的数量。该队列既包含IN请求，又包含OUT请求。 0: 周期性发送请求队列已满 1: 1个位置可用 2: 2个位置可用 n: n个位置可用 (其中, n范围: 0~8) 其它值: 保留	R
b15~b0	PTXFSAVL	周期性发送数据 FIFO可用空间	周期性发送数据FIFO可用空间 (Periodic transmit data FIFO space available) 指示可供写入的周期性 TxFIFO 的空闲位置的数量。 以32位字为单位 0: 周期性TxFIFO已满 1: 1个字可用 2: 2个字可用 n: n个字可用 (其中, n范围: 0~PTXFD) 其它值: 保留	R

### 34.7.3.5 USBFS 主机全体通道中断寄存器 (USBFS\_HINT)

USBFS Host all channels interrupt register

偏移地址：0x414

复位值：0x0000 0000

当通道上有事件发生时，主机全体通道中断寄存器会使用模块中断寄存器中的主机通道中断位（USBFS\_GINTSTS 中的 HCINT 位）中断应用程序。每个通道对应 1 个中断位，最多有 16 个位。

当应用程序通过相应主机通道 x 中断寄存器清零中断时，该寄存器中的位也会清零。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HINT[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	必须保持复位值。	R/W											
b15~b0	HINT	通道中断	通道中断(Channel interrupt) 每个通道对应一位：通道0对应位0，通道15对应位15。	R											

### 34.7.3.6 USBFS 主机全体通道中断屏蔽寄存器(USBFS\_HINTMSK)

USBFS Host all channels interrupt mask register

偏移地址：0x418

复位值：0x0000 0000

主机全体通道中断屏蔽寄存器与主机全体通道中断寄存器结合使用，进而在通道上发生事件时中断应用程序。

每个通道对应 1 个中断屏蔽位，最多有 16 个位。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HINTM[11:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为“0”，写入时写“0”	R/W											
b15~b0	HINTM	通道中断屏蔽	通道中断屏蔽 (Channel interrupt mask) 0：屏蔽中断 1：使能中断 每个通道对应一位：通道0对应位0，通道15对应位15。	R/W											

### 34.7.3.7 USBFS 主机端口控制和状态寄存器 (USBFS\_HPRT)

USBFS Host port control and status register

偏移地址：0x440

复位值：0x0000 0000

该寄存器仅在主机模式下可用。当前，USBFS 主机仅支持一个端口。

该寄存器包含 USB 端口相关的信息，如 USB 复位、使能、挂起、恢复、连接状态。该寄存器中的 PENCHNG/PCDET 位可通过模块中断寄存器中的主机端口中断位（USBFS\_GINTST 中 HPRTINT 位）触发应用程序中断。发生端口中断时，应用程序必须读取该寄存器，并将引起中断的位清零。应用程序必须向该位写入 1 以清除该中断。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved												PSPD[1:0]		Rese rved	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved	PPWR	PLSTS[1:0]	Rese rved	PRST	PSUSP	PRES	Reserved	PENCHNG	PENA	PCDET	PCSTS				
<hr/>															
位	标记	位名	功能	读写											
b31~b19	Reserved	-	必须保持复位值。	R/W											
b18~b17	PSPD	端口速度	端口速度(Port speed) 指示连接到该端口的设备的速度。 00/11: 保留 01: 全速 10: 低速	R											
b16~b13	Reserved	-	必须保持复位值。	R/W											
b12	PPWR	端口电源	端口电源(Port power) 应用程序使用该字段控制该端口的电源。由于本USBFS内置PHY不具备供电能力，所以此为设置为1时，通过USBFS_DRVVBUS使能外部USB电源芯片供电。 0: 掉电 1: 通电	R/W											
b11~b10	PLSTS	端口线状态	指示 USB 数据线的当前逻辑电平 位11: USBFS_DM 的逻辑电平 位10: USBFS_DP 的逻辑电平	R											
b9	Reserved	-	必须保持复位值。	R/W											
b8	PRST	端口复位	端口复位(Port reset) 应用程序将该位置1时，会在该端口上启动复位序列。应用程序必须为复位周期定时，并在复位序列完成后将该位清零。 0: 端口未处于复位状态 1: 端口处于复位状态 应用程序必须将该位置1并最少保持 10 ms，以在端口上启动复位。	R/W											
b7	PSUSP	端口挂起	端口挂起(Port suspend) 应用程序将此位置1以将此端口置于挂起模式。只有此位置1时，模块才会停止发送SOF。要停止PHY时钟，应用程序必须将端口时钟停止位置1，这会使能PHY的挂起输入引脚。	R/W											

此位的读取值反映该端口的当前挂起状态。检测到远程唤醒信号，或者应用程序将此寄存器中的端口复位位或端口恢复位置1后，模块可将此位清零；或应用程序将模块中断寄存器中的恢复/远程唤醒检测中断位或断开连接检测中断位（分别为USBFS\_GINTSTS中的WKUINT或 DISCINT）置1，模块也可将此位清零。

0：端口未处于挂起模式

1：端口处于挂起模式

端口恢复 (Port resume)

应用程序将此位置1以在该端口上驱动恢复信号。模块会持续驱动恢复信号直到应用程序将此位清零。

如模块中断寄存器中的端口恢复/ 远程唤醒检测中断位（USBFS\_GINTSTS中 WKUINT位）指示，如果模块检测到USB远程唤醒序列，则开始驱动恢复信

b6 PRES 端口恢复 号，而无需应用程序进行干预；如果模块检测到断开连接的情况，则将此位清零。 R/W

此位的读取值指示当前模块是否

正在驱动恢复信号。

0：不驱动恢复信号

1：驱动恢复信号

b5~b4 Reserved - 必须保持复位值。 R/W

端口使能/禁止变化 (Port enable/disable change)

b3 PENCHNG 端口使能/禁止变化 该寄存器中的端口使能位2的状态发生变化时，模块将此位置1。通过软件对该位写1清零。 R/W

端口使能(Port enable)

端口执行复位序列后，只能由模块使能，并且可以由过流状况、断开连接状况或应用程序将此位清零来禁止。应用程序无法通过对寄存器执行写操作将此位置1。只能将此位清零来禁止端口。对此位的操作不会触发应用程序的任何中断。

b2 PENA 端口使能 0：禁止端口 R/W

1：使能端口

检测到端口连接 (Port connect detected)

b1 PCDET 检测到端口连接 当检测到设备连接时，模块将此位置1，以使用模块中断寄存器中的主机端口中断位（USBFS\_GINTSTS中的HPRTINT位）触发应用程序的中断。 R/W  
应用程序必须将此位置1才可清除该中断。

端口连接状态(Port connect status)

b0 PCSTS 端口连接状态 0：端口未连接设备 R  
1：端口已连接设备

**34.7.3.8 USBFS 主机通道 x 特性寄存器 (USBFS\_HCCHARx) (x = 0..11)**

USBFS Host channel-x characteristics register

偏移地址: 0x500 + (通道编号 × 0x20)

复位值: 0x0000 0000

该寄存用于设定主机通道特性。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CHE NA	CHDI S	ODD FRM	DAD[6:0]				Reserved			EPTYP[1:0]		LSDEV	Rese rved		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EPDI R	EPNUM[3:0]			MPSIZ[10:0]											

位	标记	位名	功能	读写
b31	CHENA	通道使能	通道使能(Channel enable) 此字段由应用程序软件置1，并由USBFS主机硬件清零。 0: 禁止通道 (Channel disabled) 1: 使能通道	R/W
b30	CHDIS	通道禁止	通道禁止(Channel disable) 应用程序将此位置1以停止通过通道发送/接收数据，即使通过该通道的传输还未完成，停止操作仍然生效。 应用程序必须等待禁止通道的中断以确认通道已经被禁止。	R/W
b29	ODDFRM	奇数帧	奇数帧(Odd frame) 此字段由应用程序置位或复位，以分别指示USBFS主机必须传输奇数帧或偶数帧。此字段只适用于周期性（同步和中断）事务。	R/W
b28~b22	DAD	设备地址	设备地址(Device address) 此字段用于指定要与该主机通信的特定设备。	R/W
b21~b20	Reserved	-	必须保持复位值。	R/W
b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 指示选择的传输类型。 00: 控制 01: 同步 10: 批量 11: 中断	R/W
b17	LSDEV	低速设备	低速设备 (Low-speed device) 此字段由应用程序置 1，表示此通道正在与一个低速设备进行通信。	R/W
b16	Reserved	-	必须保持复位值。	R/W
b15	EPDIR	端点方向	端点方向(Endpoint direction) 指示通信事务的方向是输入还是输出。 0: 输出 1: 输入	R/W
b14-b11	EPNUM	端点编号	端点编号 (Endpoint number) 指示要与该主机通道通信的 USB 设备的端点号。	R/W
b10-b0	MPSIZ	最大数据包大小	最大数据包大小 (Maximum packet size)	R/W

---

指示与该主机通道通信的设备端点的最大数据包大小。

---

**34.7.3.9 USBFS 主机通道 x 中断寄存器 (USBFS\_HCINTx) (x = 0..11)**

USBFS Host channel-x interrupt register

偏移地址：0x508 + (通道编号 × 0x20)

复位值：0x0000 0000

该寄存器指示在出现 USB 和 AHB 相关事件时通道的状态。当模块中断寄存器中的主机通道中断位 (USBFS\_GINTSTS 中的 HCINT 位) 置 1 时，应用程序必须读取该寄存器。在对寄存器执行读操作之前，应用程序必须先读取主机全体通道中断(USBFS\_HAIN) 寄存器，以获取主机通道 x 中断寄存器的准确通道编号。应用程序必须将该寄存器中的相应位清零，才能将 USBFS\_HAIN 和 USBFS\_GINTSTS 寄存器中的对应位清零。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					DTE RR	FRM OR	BBE RR	TXER R	Rese rved	ACK	NAK	STAL L	AHB ERR	CHH	XFRC
位 标记 位名 功能 读写															
b31~b11	Reserved	-	必须保持复位值。										R/W		
b10	DTERR	数据切换错误	数据同步错误 (Data toggle error) 应用程序需通过写1清除该位。										R/W		
b9	FRMOR	帧溢出错误	帧溢出错误(Frame overrun) 应用程序需通过写1清除该位。										R/W		
b8	BBERR	串扰错误	串扰错误(Babble error) 产生串扰事件的典型原因是端点发送了一个数据包，但是数据包长度超过了端点的最大包长。 应用程序需通过写1清除该位。										R/W		
b7	TXERR	通信事务错误	通信事务错误 (Transaction error) 指示 USB 上发生下列错误之一： CRC校验失败 超时 位填充错误 错误的EOP 应用程序需通过写1清除该位。										R/W		
b6	Reserved	-	必须保持复位值。										R/W		
b5	ACK	收到/发出ACK 响应	收到/发出ACK响应(ACK response received/transmitted interrupt) 应用程序需通过写1清除该位。										R/W		
b4	NAK	收到NAK响应	收到NAK响应(NAK response received interrupt) 应用程序需通过写1清除该位。										R/W		
b3	STALL	收到STALL响应	收到STALL响应(STALL response received interrupt) AHB错误 (AHB error)										R/W		
b2	AHBERR	AHB错误	仅当处于内部DMA模式下且AHB读/写操作期间发生AHB错误时才生成此错误。应用程序可通过读取相应的 DMA 通道地址寄存器来获取错误地址。 应用程序需通过写1清除该位。										R/W		

			通道停止(Channel halted)	
b1	CHH	通道停止	因任意USB事务错误或为响应应用程序的禁止请求而导致传输非正常结束。 应用程序需通过写1清除该位。	R/W
			传输完成(Transfer completed)	
b0	XFRC	传输完成	未出现任何错误，正常完成传输。 应用程序需通过写1清除该位。	R/W

### 34.7.3.10 USBFS 主机通道 x 中断屏蔽寄存器 (USBFS\_HCINTMSKx) (x = 0..11)

USBFS Host channel-x interrupt mask register

偏移地址: 0x50C + (通道编号 × 0x20)

复位值: 0x0000 0000

该寄存器用于选择屏蔽主机通道中断。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
位	标记	位名	功能	读写											
b31~b11	Reserved	-	必须保持复位值。	R/W											
b10	DTERRM	数据切换错误中断屏蔽	数据同步错误屏蔽(Data toggle error mask) 0: 屏蔽中断 1: 使能中断	R/W											
b9	FRMORM	帧溢出错误中断屏蔽	帧溢出屏蔽(Frame overrun mask) 0: 屏蔽中断 1: 使能中断	R/W											
b8	BBERRM	串扰错误中断屏蔽	串扰错误屏蔽(Babble error mask) 0: 屏蔽中断 1: 使能中断	R/W											
b7	TXERRM	通信事务错误中断屏蔽	通信事务错误屏蔽(Transaction error mask) 0: 屏蔽中断 1: 使能中断	R/W											
b6	Reserved	-	必须保持复位值。	R/W											
b5	ACKM	收到/发出ACK 响应中断屏蔽	ACK响应接收/发送中断屏蔽 (ACK response received/transmitted interrupt mask) 0: 屏蔽中断 1: 使能中断	R/W											
b4	NAKM	收到NAK响应中断屏蔽	NAK响应接收中断屏蔽(NAK response received interrupt mask) 0: 屏蔽中断 1: 使能中断	R/W											
b3	STALLM	收到STALL响应中断屏蔽	收到STALL响应(STALL response received interrupt mask) 0: 屏蔽中断 1: 使能中断	R/W											
b2	AHBERRM	AHBERR中断屏蔽	AHBERR中断屏蔽 (AHB error mask) 0: 屏蔽中断 1: 使能中断	R/W											
b1	CHHM	通道停止中断屏蔽	通道停止中断屏蔽 (Channel halted mask) 0: 屏蔽中断 1: 使能中断	R/W											
b0	XFRCM	传输完成中断屏蔽	传输完成中断屏蔽 (Transfer completed mask) 0: 屏蔽中断	R/W											

---

1: 使能中断

---

### 34.7.3.11 USBFS 主机通道 x 传输大小寄存器 (USBFS\_HCTSIZx) (x = 0..11)

USBFS Host channel-x transfer size register

偏移地址：0x510 + (通道编号 × 0x20)

复位值：0x0000 0000

该寄存器用于设定主机通道传输大小以及数据 PID。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	DPID[1:0]		PKTCNT[9:0]										XFRSIZ[18:16]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XFRSIZ[15:0]															
位	标记	位名	功能	读写											
b31	Reserved	-	必须保持复位值。	R/W											
数据PID (Data PID) 应用程序在此字段设置数据通信的初始同步PID。 主机在此次传输事务过程中保留该字段的设置。															
b30~b29	DPID	数据PID	00: DATA0 01: 保留 10: DATA1 11: SETUP	R/W											
数据包计数 (Packet count) 应用程序在此字段中设置将要发送或接收的数据包数。 主机每成功发送或接收一个数据包便递减一次计数值。此值达到0后，将中断 应用程序来指示操作正常完成。															
b18~b0	XFRSIZ	传输大小	对于OUT操作，此字段为传输期间主机发送的数据字节数。 对于IN操作，此字段为应用程序保留给传输的缓冲区大小。对于IN事务（周期性和非周期性），应用程序会将此字段编程为最大数据包大小的整数倍。	R/W											

### 34.7.3.12 USBFS 主机通道 xDMA 地址寄存器 (USBFS\_HCDMAX) (x = 0..11)

USBFS Host channel-x DMA address register

偏移地址：0x514 + (通道编号 × 0x20)

复位值：0xXXXX XXXX

该寄存器用于设定主机 DMA 模式时 DMA 地址。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DMAADDR[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DMAADDR[15:0]															

位	标记	位名	功能	读写
b31~b0	DMAADDR	DMA地址	DMA地址(DMA address) 此字段存储主机从设备端点获取数据或往设备端点发送数据所用DMA传输的存储器的地址。每次AHB传输结束，该寄存器都会递增。	R/W

### 34.7.4 USBFS 设备模式寄存器

设备模式寄存器会影响设备模式下的模块操作。在主机模式下不得访问设备模式寄存器，因为产生的结果不明确。

除非特别说明，否则寄存器描述中的位值以二进制表示。

#### 34.7.4.1 USBFS 设备配置寄存器(USBFS\_DCFG)

USBFS Device configuration register

偏移地址：0x800

复位值：0x0820 0000

此寄存器在上电、执行某些控制命令或枚举后，会将模块配置为设备模式。请勿在初始编程后更改该寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved	PFIVL[1:0]						DAD[6:0]					Rese rved	NZLS OHSK		DSPD[1:0]
<hr/>															
位	标记	位名	功能	读写											
b31~b13	Reserved	-	必须保持复位值。	R/W											
<hr/>															
b12~b11	PFIVL	周期性帧间隔	周期性帧间隔(Periodic frame interval) 指示一帧内必须使用周期性帧中断通知应用程序的时间点。此功能可用于确定该帧的所有同步通信是否完成。	R/W											
<hr/>															
b10~b4	DAD	设备地址	设备地址(Device address) 应用程序必须在执行每个SetAddress控制命令后根据命令参数对该字段进行设置。	R/W											
<hr/>															
b3	Reserved	-	必须保持复位值。	R/W											
<hr/>															
b2	NZLSOHSK	非零长度状态OUT握手信号	非零长度状态OUT握手信号 (Non-zero-length status OUT handshake) 在控制传输状态阶段的OUT事务期间，当模块收到非零长度数据包后，应用程序可以使用此字段选择要发送的握手信号。 1: 收到非零长度状态OUT事务时，回复STALL握手信号，收到的 OUT 数据包不发送给应用程序。 0: 将收到的OUT数据包（零长度或非零长度）发送给应用程序，并基于设备端点控制寄存器中端点的NAK和STALL位回复握手信号。	R/W											
<hr/>															
b1~b0	DSPD	设备速度	设备速度 (Device speed) 指示应用程序要求模块进行枚举所采用的速度，或应用程序支持的最大速度。 但是，实际总线速度只有在完成chirp序列后才能确定，同时此速度基于与模块连接的USB主机的速度。	R/W											

---

00: 保留  
01: 保留  
10: 保留  
11: 全速 (USB 1.1收发器时钟为48 MHz)

---

### 34.7.4.2 USBFS 设备控制寄存器(USBFS\_DCTL)

USBFS Device control register

偏移地址：0x804

复位值：0x0000 0002

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved		POP RGD NE	CGO NAK	SGO NAK	CGIN AK	SGIN AK		Reserved		GON STS	GINS TS	SDIS	RWU SIG		

位	标记	位名	功能	读写
b31~b12	Reserved	-	必须保持复位值。	R/W
b11	POPRGDNE	上电编程完成	上电编程完成(Power-on programming done) 应用程序使用此位指示寄存器从掉电模式唤醒后已完成编程。	R/W
b10	CGONAK	清零全局OUT NAK	清零全局OUT NAK(Clear global OUT NAK) 对此位执行写操作会将全局OUT NAK清零。	R/W
b9	SGONAK	置位全局OUT NAK	置位全局OUT NAK(Set global OUT NAK) 对此位执行写操作会将全局OUT NAK置1。 应用程序使用此位在所有OUT端点发送NAK握手信号。 应用程序只有确定模块中断寄存器中全局OUT NAK有效位(USBFS_GINTSTS中GONAKEFF位)已清零时，才可以将此位置1。	R/W
b8	CGINAK	清零全局IN NAK	清零全局IN NAK (Clear global IN NAK) 对此位执行写操作会将全局IN NAK清零。	R/W
b7	SGINAK	置位全局IN NAK	置位全局IN NAK (Set global IN NAK) 对此字段执行写操作会将全局非周期性IN NAK置1。应用程序使用此位使所有非周期性IN端点发送NAK握手信号。 应用程序只有确定模块中断寄存器中全局IN NAK有效位(USBFS_GINTSTS中GINAKEFF位)已清零时，才可以将此位置1。	R/W
b6~b4	Reserved	-	必须保持复位值。	R/W
b3	GONSTS	全局OUT NAK状态	全局OUT NAK状态(Global OUT NAK status) 0: 将根据FIFO状态和NAK和STALL位设置发送握手信号。 1: 无论Rx FIFO中是否还有空闲空间都不接收数据。除SETUP事务之外，对所有收到的数据包回复NAK握手信号。所有同步类型的OUT数据包都将被丢弃。	R
b2	GINSTS	全局IN NAK状态	全局IN NAK状态(Global IN NAK status) 0: 将根据发送FIFO中的数据可用性回复握手信号。 1: 使所有非周期性IN端点回复NAK握手信号，无需考虑发送FIFO中的数据可用性。	R
b1	SDIS	软断连	软断连(Soft disconnect) 应用程序使用该位向USBFS模块发出执行软断开的信号。该位置1时，主机不会看到设备已连接，且该设备也不会接收USB上的信号。在应用程序将此位清零之前，模块会保持断开状态。 0: 正常工作。此位在软断连之后清零，会使主机收到设备已连接的事件。重新连接设备之后，USB主机会重新启动设备枚举。 1: 使主机收到设备断开连接的事件。	R/W

全速时，软断连的最长时间规定如下：

挂起状态：最长时间为1ms+2.5us

空闲状态：2.5us

非空闲或挂起状态：2.5us

发送远程唤醒信号(Remote wakeup signaling)

b0	RWUSIG	发送远程唤醒信号	R/W
----	--------	----------	-----

应用程序将此位置1时，模块会启动远程唤醒信号，以唤醒USB主机。应用程序必须将此位置1以使模块退出挂起状态。根据USB 2.0规范，应用程序必须在将此位置1之后的1 ms到15 ms内将其清零。

### 34.7.4.3 USBFS 设备状态寄存器(USBFS\_DSTS)

USBFS Device status register

偏移地址：0x808

复位值：0x0000 0002

此寄存器指示模块在出现 USB 相关事件时的状态。发生中断时，必须从设备全体中断 (USBFS\_DAINT) 寄存器读取发生中断的端点信息。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								LNSTS[1:0]		FNSOF[13:8]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FNSOF[7:0]								Reserved		EERR	ENUMSPD[1:0]		SUSP STS		

位	标记	位名	功能	读写
b31~b24	Reserved	-	必须保持复位值。	R/W
b23~b22	LNSTS	USB总线状态	LNSTS[1]: D+的逻辑电平 LNSTS[0]: D-的逻辑电平	R
b21~b8	FNSOF	接收SOF的帧编号	接收SOF的帧编号 (Frame number of the received SOF)	R
b7~b4	Reserved	-	必须保持复位值。	R/W
b3	EERR	不定错误	不定错误(Erratic error) 模块将该位置1以报告任何不定错误。 由于不定错误，USBFS控制器会进入挂起状态，并且会USBFS_GINTSTS寄存器的早期挂起位 (USBFS_GINTSTS中的ESUSP 位) 生成一个中断。如果早期挂起中断是由不定错误触发，则应用程序只能执行软断开以恢复通信。	R
b2~b1	ENUMSPD	枚举速度	枚举速度(Enumerated speed) 指示USBFS控制器通过chirp序列检测速度后被枚举成的速度。 01: 保留 10: 保留 11: 全速 (PHY 时钟运行频率为 48 MHz) 其它值: 保留	R
b0	SUSPSTS	挂起状态	挂起状态(Suspend status) 在设备模式下，只要在USB上检测到挂起状态，该位就会置1。当USB 总线上的空闲状态保持3ms，模块便会进入挂起状态。出现以下情况时，模块会退出挂起状态： — USB 数据线上有活动 — 应用程序对USBFS_DCTL寄存器的远程唤醒信号位 (USBFS_DCTL中的RWUSIG位) 执行写操作。	R

**34.7.4.4 USBFS 设备 IN 端点通用中断屏蔽寄存器(USBFS\_DIEPMSK)**

USBFS Device IN endpoint common interrupt mask register

偏移地址：0x810

复位值：0x0000 0000

此寄存器与全体端点的各个 USBFS\_DIEPINTx 寄存器配合使用，以便在每个 IN 端点上生成中断。通过对此寄存器的相应位执行写操作，可屏蔽 USBFS\_DIEPINTx 寄存器中的 IN 端点中断。默认情况下，状态中断都被屏蔽。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved	NAK M								INEP NEM	INEP NMM	TTXF EMSK	TOM	Rese rved	EPD M	XFRC M
位 标记 位名 功能 读写															
b31~b14	Reserved	-	必须保持复位值。										R/W		
b13	NAKM	NAK中断屏蔽	NAK中断屏蔽 (NAK mask) 0: 屏蔽中断 1: 使能中断										R/W		
b12~b7	Reserved	-	必须保持复位值。										R/W		
b6	INEPNEM	IN端点NAK有效中断屏蔽	IN端点NAK有效中断屏蔽 (IN endpoint NAK effective mask) 0: 屏蔽中断 1: 使能中断										R/W		
b5	INEPNMM	EP不匹配时接收到IN令牌中断屏蔽 IN令牌中断屏蔽	EP不匹配时接收到IN令牌中断屏蔽 (IN token received with EP mismatch mask) 0: 屏蔽中断 1: 使能中断										R/W		
b4	TTXFEMSK	TxFIFO为空时接收到IN令牌中断屏蔽	TxFIFO为空时接收到IN令牌中断屏蔽 (IN token received when TxFIFO empty mask) 0: 屏蔽中断 1: 使能中断										R/W		
b3	TOM	超时中断屏蔽 (非同步端点)	超时中断屏蔽 (非同步端点) (Timeout condition mask (Non-isochronous endpoints)) 0: 屏蔽中断 1: 使能中断										R/W		
b2	Reserved	-	必须保持复位值。										R/W		
b1	EPDM	端点禁止中断屏蔽	端点禁止中断屏蔽(Endpoint disabled interrupt mask) 0: 屏蔽中断 1: 使能中断										R/W		
b0	XFRCM	传输完成中断屏蔽	传输完成中断屏蔽(Transfer completed interrupt mask) 0: 屏蔽中断 1: 使能中断										R/W		

**34.7.4.5 USBFS 设备 OUT 端点通用中断屏蔽寄存器(USBFS\_DOEPMSK)**

USBFS Device OUT endpoint common interrupt mask register

偏移地址：0x814

复位值：0x0000 0000

此寄存器与全体端点的各个 USBFS\_DOEPINTx 寄存器配合使用，以便在每个 OUT 端点上生成中断。通过对此寄存器的相应位执行写操作，可屏蔽 USBFS\_DOEPINTx 寄存器中的 OUT 端点中断。默认情况下，状态中断都被屏蔽。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
OTEP DM STUP M Rese rved EPD M XFRC M															
位	标记	位名	功能	读写											
b31~b5	Reserved	-	必须保持复位值。	R/W											
b4	OTEPDM	端点禁止时接收到OUT令牌中断屏蔽 OUT令牌中断屏蔽	端点禁止时接收到OUT令牌中断屏蔽 (OUT token received when endpoint disabled mask) 仅适用于控制 OUT 端点。 0: 屏蔽中断 1: 使能中断	R/W											
b3	STUPM	SETUP阶段完成中 断屏蔽	SETUP阶段完成中断屏蔽(SETUP phase done mask) 仅适用于控制端点。 0: 屏蔽中断 1: 使能中断	R/W											
b2	Reserved	-	必须保持复位值。	R/W											
b1	EPDM	端点禁止中断屏蔽	端点禁止中断屏蔽(Endpoint disabled interrupt mask) 0: 屏蔽中断 1: 使能中断	R/W											
b0	XFRCM	传输完成中断屏蔽	传输完成中断屏蔽(Transfer completed interrupt mask) 0: 屏蔽中断 1: 使能中断	R/W											

### 34.7.4.6 USBFS 设备全体端点中断寄存器 (USBFS\_DAINT)

USBFS Device OUT endpoint common interrupt mask register

偏移地址：0x818

复位值：0x0000 0000

当端点上发生有效事件时，USBFS\_DAINT 寄存器将通过 USBFS\_GINTSTS 寄存器中的设备 OUT 端点中断位或设备 IN 端点中断位（分别为 USBFS\_GINTSTS 中的 OEPINT 或 IEPINT 位）来中断应用程序。每个端点对应一个中断位，OUT 端点和 IN 端点均最多有 16 个中断位。双向端点将使用相应的 IN 和 OUT 中断位。当应用程序将相应设备端点 x 中断寄存器(USBFS\_DIEPINTx/USBFS\_DOEPINTx)中的位置 1 和清零时，此寄存器中的相应位也将置 1 和清零。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
OEPINT[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
IEPINT[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	OEPINT	OUT端点中断位	OUT端点中断位(OUT endpoint interrupt bits) 每个OUT端点对应一位： OUT端点0对应位16，而OUT端点15对应位31。	R/W											
b15~b0	IEPINT	IN端点中断位	IN端点中断位(IN endpoint interrupt bits) 每个IN端点对应一位： IN端点0对应位0，而IN端点15对应位15。	R/W											

### 34.7.4.7 USBFS 设备全体端点中断屏蔽寄存器 (USBFS\_DAINTMSK)

USBFS Device all endpoints interrupt mask register

偏移地址：0x81C

复位值：0x0000 0000

USBFS\_DAINTMSK 寄存器与设备端点中断寄存器结合使用，在设备端点上发生事件时中断应用程序。但是，与该中断相对应的 USBFS\_DAINT 寄存器位仍会置 1。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
OEPINTM[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
IEPINTM[15:0]															
<hr/>															
位	标记	位名	功能	读写											
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															
<hr/>															

### 34.7.4.8 USBFS 设备 IN 端点 FIFO 空中断屏蔽寄存器(USBFS\_DIEPEMPMSK)

USBFS Device IN endpoint FIFO empty interrupt mask register

偏移地址：0x834

复位值：0x0000 0000

此寄存器用于控制 IN 端点 FIFO 空中断的生成。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
INEPTXFEM[5:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	必须保持复位值。	R/W											
b15~b0	INEPTXFEM	IN EP TxFIFO空中断屏蔽位	IN EP Tx FIFO empty interrupt mask bits 这些位用作USBFS_DIEPINTx的屏蔽位。 每个位对应一个IN端点的TXFE中断： IN端点0对应位0，而IN端点15对应位15 0：屏蔽中断 1：使能中断	R/W											

**34.7.4.9 USBFS 设备 IN 端点 0 控制寄存器(USBFS\_DIEPCTL0)**

USBFS Device control IN endpoint 0 control register

偏移地址：0x900

复位值：0x0000 8000

此寄存器用于控制控制传输端点 0。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EPEN A	EPDI S	Reserved	SNA K	CNA K	TXFNUM[3:0]		STAL L	Rese rved	EPTYP[1:0]	NAK STS	Rese rved				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
USB AEP						Reserved									MPSIZ[1:0]

位	标记	位名	功能	读写
b31	EPENA	端点使能	端点使能(Endpoint enable) 应用程序将此位置1以在端点0上启动数据发送。 在此端点上触发以下任一中断之前，模块会将此位清零： — SETUP阶段完成 — 端点禁止 — 传输完成	R/W
b30	EPDIS	端点禁止	端点禁止(Endpoint disable) 即使在该端点上的传输完成之前，应用程序也可将此位置1，以停止端点上的数据发送。应用程序必须等到发生端点禁止中断后，才能将端点视为禁止端点。在端点禁止中断位置1前，模块会将此位清零。只有在该端点的端点使能位置1后，应用程序才可将该位置 1。	R/W
b29~b28	Reserved	-	必须保持复位值。	R/W
b27	SNAK	置位NAK位	置位NAK位(Set NAK) 对此位进行写操作会将端点的NAK位置1。 通过此位，应用程序可以控制端点上NAK握手信号的发送。模块也可在端点接收到SETUP数据包后将该端点的此位置1。	R/W
b26	CNAK	清零NAK位	清零NAK位(Clear NAK) 对此位进行写操作会将端点的NAK位清零。	R/W
b25~b22	TXFNUM	TxFIFO编号	TxFIFO编号(TxFIFO number) 该值设置为分配给IN端点0的FIFO编号。只能使用TX-FIFO0。	R/W
b21	STALL	STALL握手	STALL握手(STALL handshake) 应用程序只能将此位置1，端点接收到SETUP令牌时，模块会将此位清零。如果NAK位、全局IN NAK或全局OUT NAK与此位均置 1，则STALL位优先。	R/W
b20	Reserved	-	必须保持复位值。	R/W
b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 硬件设置为'00'，表示控制类型的端点。	R
b17	NAKSTS	NAK状态	NAK状态(NAK status) 指示以下结果： 0：模块根据FIFO状态回复非NAK握手。 1：模块在此端点上回复NAK握手。 当此位置1时（无论是被应用程序还是被模块），即使TxFIFO中仍有数据可用，模块也会停止发送数据。无论此位如何设置，模块总是通过ACK握手响应	R

SETUP数据包。				
b16	Reserved	-	必须保持复位值。	R/W
b15	USBAEP	USB活动端点	USB活动端点(USB active endpoint) 此位总是置1，指示在所有配置和接口中控制端点0始终处于激活状态。	R
b14~b2	Reserved	-	必须保持复位值。	R/W
最大数据包大小(Maximum packet size) 应用程序必须将此字段编程为当前逻辑端点的最大数据包大小。				
b1~b0	MPSIZ	最大数据包大小	00: 64字节 01: 32字节 10: 16字节 11: 8字节	R/W

### 34.7.4.10 USBFS 设备 IN 端点 x 控制寄存器(USBFS\_DIEPCTLx)(x=1..5)

USBFS Device IN endpoint x control register

偏移地址：0x900 + (端点编号 × 0x20)

复位值：0x0000 0000

应用程序使用此寄存器控制各个逻辑端点（端点 0 除外）的行为。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EPE NA	EPDI S	SOD DFR M	SD0 PID/ SEV NFR M	SNAK	CNA K	TXFNUM[3:0]		STAL L	Rese rved	EPTYP[1:0]	NAK STS	EONU M/ DPI			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
USB AEP		Reserved								MPSIZ[10:0]					

位	标记	位名	功能	读写
b31	EPENA	端点使能	端点使能(Endpoint enable) 应用程序将此位置1以在端点上启动数据发送。 在此端点上触发以下任一中断之前，模块会将此位清零： — 端点禁止 — 传输完成	R/W
b30	EPDIS	端点禁止	端点禁止(Endpoint disable) 即使在该端点上的传输完成之前，应用程序也可将此位置1，以停止端点上的数据发送。应用程序必须等到发生端点禁止中断后，才能将端点视为禁止端点。在端点禁止中断位置1前，模块会将此位清零。只有在该端点的端点使能位置1后，应用程序才可将该位置 1。	R/W
b29	SODDFRM	设置奇数帧	设置奇数帧(Set odd frame) 仅适用于同步IN和OUT端点。 对此字段进行写操作会将偶数/奇数帧 (EONUM) 字段设置为奇数帧。	R/W
b28	SD0PID/ SEVNFRM	设置DATA0 PID/ SEVNFRM	设置DATA0 PID (Set DATA0 PID) 仅适用于中断/批量IN 端点。 对此字段进行写操作会将此寄存器中的端点数据PID(DPID)字段设置为 SEVNFRM: 设置偶数帧 (Set even frame) 仅适用于同步IN端点。 对此字段进行写操作会将偶数/奇数帧(EONUM)字段设置为偶数帧。	R/W
b27	SNAK	置位NAK位	置位NAK位(Set NAK) 对此位进行写操作会将端点的NAK位置 1。 通过此位，应用程序可以控制端点上NAK握手信号的发送。发生传输完成中断时或端点上接收到SETUP后，模块也可以将OUT端点的这个位置 1	R/W
b26	CNAK	清零NAK位	清零NAK位(Clear NAK) 对此位进行写操作会将端点的NAK位清零。	R/W
b25~b22	TXFNUM	TxFIFO编号	TxFIFO编号(TxFIFO number) 这些位用于指定与此端点相关联的 FIFO 编号。必须为每个有效的IN端点设置单独的 FIFO编号。 此字段仅针对IN端点有效。	R/W

			STALL握手(STALL handshake)	
b21	STALL	STALL握手	应用程序将此位置1使得设备对来自 USB 主机的所有令牌都回复STALL。如果 NAK位、全局IN NAK或全局OUT NAK与此位同时置 1，则STALL位优先。只有应用程序能够将此位清零，而模块则不能。	R/W
b20	Reserved	-	必须保持复位值。	R/W
			端点类型(Endpoint type) 以下是这个逻辑端点支持的传输类型。	
b19~b18	EPTYP	端点类型	00: 控制 01: 同步 10: 批量 11: 中断	R
			NAK状态(NAK status) 指示以下结果： 0: 模块根据FIFO状态回复非NAK握手。 1: 模块在此端点上回复NAK握手。	
b17	NAKSTS	NAK状态	当应用程序或模块将此位置1时： 对于非同步IN端点：即使TxFIFO中存在可用数据，模块也会停止通过 IN端点发送任何数据。 对于同步IN端点：即使TxFIFO中存在可用数据，模块也会发送长度为零的数据包。 无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R
			偶数/奇数帧 (Even/odd frame) 仅适用于同步IN端点。 指示模块为此端点发送/接收同步的数据所在的帧的编号。应用程序必须通过此寄存器中的SEVNFRM和SODDFRM字段对偶数/奇数帧编号进行编程，以便此端点发送/接收同步数据。	
b16	EONUM/ DPID	偶数/奇数帧/ 端点数据PID	0: 偶数帧 1: 奇数帧 DPID: 端点数据PID (Endpoint data PID) 仅适用于中断/批量IN端点。 包含此端点上将要接收或发送的数据包的PID。端点激活后，应用程序必须对要在此端点上接收或发送的首个数据包的PID进行编程。应用程序使用SD0PID寄存器字段对DATA0或DATA1 PID进行编程。 0: DATA0 1: DATA1	R
b15	USBAEP	USB活动端点	USB活动端点(USB active endpoint) 指示此端点在当前配置和接口中是否激活。检测到USB复位后，模块会为所有端点（端点0除外）将此位清零。接收到SetConfiguration和 SetInterface命令后，应用程序必须相应地对端点寄存器进行编程并将此位置1。	R/W
b14~b11	Reserved	-	必须保持复位值。	R/W
b10~b0	MPSIZ	最大数据包大小	最大数据包大小(Maximum packet size) 应用程序必须将此字段编程为当前逻辑端点的最大数据包大小。 此值以字节为单位。	R/W

**34.7.4.11 USBFS 设备 IN 端点 x 中断寄存器 (USBFS\_DIEPINTx) (x=0..5)**

USBFS Device IN endpoint x interrupt register

偏移地址: 0x908 + (端点编号 × 0x20)

复位值: 0x0000 0080

此寄存器指示端点在出现 USB 和 AHB 相关事件时的状态。当模块中断寄存器中的 IN 端点中断位 (USBFS\_GINTSTS 中的 IEPINT 位) 置 1 时, 应用程序必须读取此寄存器。在应用程序能够读取此寄存器之前, 必须先读取设备全体端点中断(USBFS\_DAINT)寄存器, 以获取设备端点 x 中断寄存器的准确端点编号。应用程序必须将此寄存器中的相应位清零, 才能将 USBFS\_DAINT 和 USBFS\_GINTSTS 寄存器中的对应位清零。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TXFE	INEPNE	Reserved	TTXFEE	TOC	Reserved	EPDISD	XFRC
<hr/>															
位	标记	位名	功能	读写											
b31~b8	Reserved	-	必须保持复位值。	R/W											
b7	TXFE	发送FIFO为空	发送FIFO为空 (Transmit FIFO empty) 当此端点的TxFIFO为半空或全空时, 此中断被置位。TxFIFO为半空还是全空状态由USBFS_GAHBCFG寄存器中的TxFIFO空白别位 (USBFS_GAHBCFG中的TXFELVL位) 决定。	R											
b6	INEPNE	IN端点NAK有效	INEPNE: IN端点NAK有效 (IN endpoint NAK effective) 当应用程序通过向USBFS_DIEPCTLx 中的CNAK位写入数据来将IN端点 NAK清零时, 此位可被清零。 该中断指示模块已对 (由应用程序或模块) 置1的NAK采样, 结果已生效。该中断指示由应用程序置1的IN端点NAK位已在模块中起作用。 此中断不保证在USB上发送了NAK握手信号。 STALL位的优先级高于 NAK 位。 软件写1也可将此位清零。	R/W											
b5	Reserved	-	必须保持复位值。	R/W											
b4	TTXFEE	TxFIFO为空时接收到IN令牌 到IN令牌	TxFIFO为空时接收到IN令牌 (IN token received when TxFIFO is empty) 仅适用于非周期性 IN 端点。 当和该端点相对应的TxFIFO (周期性/非周期性) 为空时, 接收到IN令牌, 从而产生中断。 通过软件写1清零。	R/W											
b3	TOC	超时	超时条件(Timeout condition) 仅适用于控制IN端点。 指示该端点对最近收到的IN令牌响应超时。 通过软件写1清零。	R/W											
b2	Reserved	-	必须保持复位值。	R/W											
b1	EPDISD	端点禁止中断	端点禁止中断 (Endpoint disabled interrupt) 此位指示该端点已经由应用程序禁止掉。	R/W											

---

通过软件写1清零。

传输完成中断 (Transfer completed interrupt)

b0	XFRC	传输完成中断	此字段指示在此端点上设置的传输已经在USB和AHB上传输完成。 通过软件写1清零。	R/W
----	------	--------	----------------------------------------------	-----

---

### 34.7.4.12 USBFS 设备 IN 端点 0 传输大小寄存器(USBFS\_DIEPTSIZE0)

USBFS Device IN endpoint 0 transfer size register

偏移地址：0x910

复位值：0x0000 0000

在使能端点 0 之前，应用程序必须修改此寄存器。通过设备控制端点 0 控制寄存器中的端点使能位（USBFS\_DIEPCTL0 中的 EPENA）使能端点 0 后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

非零端点使用端点 1~15 的寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved										PKTCNT[1:0]	Reserved				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved										XFRSIZ[6:0]					
位	标记	位名	功能	读写											
b31~b21	Reserved	-	必须保持复位值。	R/W											
b20~b19	PKTCNT	数据包计数	数据包计数 (Packet count) 指示端点0的一次数据传输包含的数据包个数。 每次从TxFIFO读取数据包（最大大小或短数据包）时，此字段将递减。	R/W											
b18~b7	Reserved	-	必须保持复位值。	R/W											
b6~b0	XFRSIZ	传输大小	传输大小(Transfer size) 指示端点0的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。 每次向TxFIFO写入来自外部存储器的数据包时，模块会使此字段递减。	R/W											

**34.7.4.13 USBFS 设备 IN 端点 x 传输大小寄存器(USBFS\_DIEPTSIZx)(x=1..5)**

USBFS Device IN endpoint x transfer size register

偏移地址：0x910 + (端点编号 × 0x20)

复位值：0x0000 0000

在使能该端点之前，应用程序必须修改此寄存器。通过 USBFS\_DIEPCTLx 寄存器中的端点使能位（USBFS\_DIEPCTLx 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	MCNT[1:0]		PKTCNT[9:0]										XFRSIZ[18:16]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XFRSIZ[15:0]															

位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R/W
b30~b29	MCNT	多重计数	多重计数 (Multi count) 对于周期性IN端点，此字段指示在USB上每帧必须发送的数据包数。模块使用此字段计算同步IN端点的数据PID。 01: 1个数据包 10: 2个数据包 11: 3 个数据包	R/W
b28~b19	PKTCNT	数据包计数	数据包计数 (Packet count) 指示该端点上的一次数据传输包含的数据包个数。 每次从TxFIFO读取数据包（最大大小或短数据包）时，此字段将递减。	R/W
b18~b0	XFRSIZ	传输大小	传输大小(Transfer size) 此字段包含当前端点的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。 每次向TxFIFO写入来自外部存储器的数据包时，模块会使此字段递减。	R/W

**34.7.4.14 USBFS 设备 IN 端点 x DMA 地址寄存器 (USBFS\_DIEPDMAx)(x=0..5)**

USBFS Device IN endpoint x transfer size register

偏移地址：0x914 + (端点编号 × 0x20)

复位值：0x0000 0000

该寄存器用于设定设备端点 DMA 模式时 DMA 地址。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16									
DMAADDR[31:16]																								
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0									
DMAADDR[15:0]																								
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding-right: 20px;">位</th> <th style="text-align: left; padding-right: 20px;">标记</th> <th style="text-align: left; padding-right: 20px;">位名</th> <th style="text-align: left; padding-right: 20px;">功能</th> <th style="text-align: right;">读写</th> </tr> </thead> <tbody> <tr> <td style="padding-top: 10px;">b31~b0</td> <td style="padding-top: 10px;">DMAADDR</td> <td style="padding-top: 10px;">DMA地址</td> <td style="padding-top: 10px;">DMA地址(DMA address) 该位包含使用DMA进行端点上数据存储时的外部存储区起始地址。  注意：对于控制端点，该字段所指向的存储区也用于存储控制OUT数据包以及SETUP事务数据包。连续接收到三个以上的SETUP数据包时，存储器中的SETUP数据包将被覆盖。每次进行AHB传输，该寄存器都会递增。应用程序必须设定一个双字对齐地址。</td> <td style="padding-top: 10px; text-align: right;">R/W</td> </tr> </tbody> </table>															位	标记	位名	功能	读写	b31~b0	DMAADDR	DMA地址	DMA地址(DMA address) 该位包含使用DMA进行端点上数据存储时的外部存储区起始地址。  注意：对于控制端点，该字段所指向的存储区也用于存储控制OUT数据包以及SETUP事务数据包。连续接收到三个以上的SETUP数据包时，存储器中的SETUP数据包将被覆盖。每次进行AHB传输，该寄存器都会递增。应用程序必须设定一个双字对齐地址。	R/W
位	标记	位名	功能	读写																				
b31~b0	DMAADDR	DMA地址	DMA地址(DMA address) 该位包含使用DMA进行端点上数据存储时的外部存储区起始地址。  注意：对于控制端点，该字段所指向的存储区也用于存储控制OUT数据包以及SETUP事务数据包。连续接收到三个以上的SETUP数据包时，存储器中的SETUP数据包将被覆盖。每次进行AHB传输，该寄存器都会递增。应用程序必须设定一个双字对齐地址。	R/W																				

**34.7.4.15 USBFS 设备 IN 端点 x 发送 FIFO 状态寄存器 (USBFS\_DTXFSTSx)(x=0..5)**

USBFS Device IN endpoint transmit FIFO status register

偏移地址：0x918 + (端点编号 × 0x20)

复位值：0x0000 0280

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16																																																															
Reserved																																																																														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																															
INEPTFSAV[15:0]																																																																														
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding-right: 20px;">位</th> <th style="text-align: left; padding-right: 20px;">标记</th> <th style="text-align: left; padding-right: 20px;">位名</th> <th style="text-align: left; padding-right: 20px;">功能</th> <th style="text-align: right;">读写</th> </tr> </thead> <tbody> <tr> <td style="padding-top: 10px;">b31~b16</td> <td style="padding-top: 10px;">Reserved</td> <td style="padding-top: 10px;">-</td> <td style="padding-top: 10px;">必须保持复位值。</td> <td style="padding-top: 10px; text-align: right;">R/W</td> </tr> <tr> <td colspan="16" style="text-align: center;">IN端点Tx FIFO可用空间 (IN endpoint Tx FIFO space available)</td></tr> <tr> <td colspan="16" style="text-align: center;">指示端点 Tx FIFO 中的可用空闲空间大小。</td></tr> <tr> <td colspan="16" style="text-align: center;">以32位字为单位：</td></tr> <tr> <td style="padding-top: 10px;">b15~b0</td> <td style="padding-top: 10px;">INEPTFSAV</td> <td style="padding-top: 10px;">IN端点Tx FIFO可用空间</td> <td style="padding-top: 10px;">0x0: 端点Tx FIFO已满 0x1: 1个字可用 0x2: 2个字可用 0xn: n个字可用</td> <td style="padding-top: 10px; text-align: right;">R</td> </tr> </tbody> </table>																位	标记	位名	功能	读写	b31~b16	Reserved	-	必须保持复位值。	R/W	IN端点Tx FIFO可用空间 (IN endpoint Tx FIFO space available)																指示端点 Tx FIFO 中的可用空闲空间大小。																以32位字为单位：																b15~b0	INEPTFSAV	IN端点Tx FIFO可用空间	0x0: 端点Tx FIFO已满 0x1: 1个字可用 0x2: 2个字可用 0xn: n个字可用	R
位	标记	位名	功能	读写																																																																										
b31~b16	Reserved	-	必须保持复位值。	R/W																																																																										
IN端点Tx FIFO可用空间 (IN endpoint Tx FIFO space available)																																																																														
指示端点 Tx FIFO 中的可用空闲空间大小。																																																																														
以32位字为单位：																																																																														
b15~b0	INEPTFSAV	IN端点Tx FIFO可用空间	0x0: 端点Tx FIFO已满 0x1: 1个字可用 0x2: 2个字可用 0xn: n个字可用	R																																																																										

**34.7.4.16 USBFS 设备 OUT 端点 0 控制寄存器(USBFS\_DOEPCTL0)**

USBFS Device control OUT endpoint 0 control register

偏移地址：0xB00

复位值：0x0000 8000

此寄存器用于控制控制传输端点 0。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EPEN A	EPDI S	Reserved	SNA K	CNA K		Reserved			STAL L	SNP M	EPTYP[1:0]	NAK STS	Reser ved		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
USB AEP						Reserved									MPSIZ[1:0]

位	标记	位名	功能	读写
			端点使能(Endpoint enable) 应用程序将此位置1以在端点0上启动数据接收。	
b31	EPENA	端点使能	在此端点上触发以下任一中断之前，模块会将此位清零： — SETUP 阶段完成 — 端点禁止 — 传输完成	R/W
b30	EPDIS	端点禁止	端点禁止(Endpoint disable) 应用程序无法禁止控制OUT端点0。	R
b29~b28	Reserved	-	必须保持复位值。	R/W
b27	SNAK	置位NAK位	置位NAK位(Set NAK) 对此位进行写操作会将端点的NAK位置1。 通过此位，应用程序可以控制端点上NAK握手信号的发送。模块也可在端点接收到SETUP数据包后将该端点的此位置1。	R/W
b26	CNAK	清零NAK位	清零NAK位(Clear NAK) 对此位进行写操作会将端点的NAK位清零。	R/W
b25~b22	Reserved	-	必须保持复位值。	R/W
b21	STALL	STALL握手	STALL握手(STALL handshake) 此端点接收到SETUP令牌时，应用程序只能将此位置1，而模块会将其清零。 如果NAK位、全局OUT NAK与此位同时置1，则STALL位优先。无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R/W
b20	SNPM	监听模式	监听模式 (Snoop mode) 此位用于将端点配置为监听模式。在监听模式下，模块不会在将OUT数据包传输到应用存储区前检查其是否正确。	R/W
b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 硬件设置为'00'，表示控制类型的端点。	R/W
b17	NAKSTS	NAK状态	NAK状态(NAK status) 指示以下结果： 0：模块根据FIFO状态回复非NAK握手。 1：模块在此端点上回复NAK握手。 当应用程序或模块将此位置1时，即使Rx FIFO中存在空间可继续容纳收到的数据包，模块也会停止接收数据。无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R

b16	Reserved	-	必须保持复位值。	R/W
b15	USBAEP	USB活动端点	USB活动端点(USB active endpoint) 此位总是置1，指示在所有配置和接口中控制端点0始终处于激活状态。	R
b14~b2	Reserved	-	必须保持复位值。	R/W
b1~b0	MPSIZ	最大数据包大小	最大数据包大小(Maximum packet size) 控制OUT端点0的最大数据包大小与在控制IN端点0中进行编程的值相同。 00: 64 字节 01: 32 字节 10: 16 字节 11: 8 字节	R/W

**34.7.4.17 USBFS 设备 OUT 端点 x 控制寄存器(USBFS\_DOEPCTLx)(x=1..5)**

USBFS Device OUT endpoint x control register

偏移地址: 0xB00 + (端点编号 × 0x20)

复位值: 0x0000 0000

应用程序使用此寄存器控制各个逻辑端点（端点 0 除外）的行为。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EPEN A	EPDI S	SOD DFRM / SD1P ID	SD0P ID/ SEV NFRM	SNA K	CNA K		Reserved		STAL L	SNP M	EPTYP[1:0]	NAK STS	EON UM/ DPID		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
USB AEP		Reserved									MPSIZ[10:0]				

位	标记	位名	功能	读写
b31	EPENA	端点使能	端点使能(Endpoint enable) 软件置位, USBFS清零 0: 端点除能 1: 端点使能	R/W
b30	EPDIS	端点禁止	端点禁止(Endpoint disable) 即使在该端点上的传送完成之前, 应用程序也可将此位置1, 以停止端点上的数据发送/接收。应用程序必须等到发生端点禁止中断后, 才能将端点视为禁止端点。在端点禁止中断位置1前, 模块会将此位清零。只有在该端点的端点使能位置1后, 应用程序才可将该位置1。	R/W
b29	SD1PID/ SODDFRM	设置DATA1 PID/ 设置奇数帧	设置DATA1 PID 仅适用于中断/批量OUT端点。对此字段进行写操作会将此寄存器中的端点数据PID(DPID) 字段设置为DATA1。 SODDFRM: 设置奇数帧 (Set odd frame)	R/W
b28	SD0PID/ SEVNFRM	设置DATA0 PID/ SEVNFRM	设置DATA0 PID (Set DATA0 PID) 仅适用于中断/批量OUT端点。 对此字段进行写操作会将此寄存器中的端点数据PID(DPID)字段设置为 SEVNFRM: 设置偶数帧 (Set even frame) 仅适用于同步OUT端点。 对此字段进行写操作会将偶数/奇数帧(EONUM)字段设置为偶数帧。	R/W
b27	SNAK	置位NAK位	置位NAK位(Set NAK) 对此位进行写操作会将端点的NAK位置1。 通过此位, 应用程序可以控制端点上NAK握手信号的发送。发生传输完成中断时或端点上接收到SETUP后, 模块也可以将OUT端点的这个位置1。	R/W
b26	CNAK	清零NAK位	清零NAK位(Clear NAK) 对此位进行写操作会将端点的NAK位清零。	R/W
b25~b22	Reserved	-	必须保持复位值。	R/W

			STALL握手(STALL handshake)	
b21	STALL	STALL握手	此端点接收到SETUP令牌时，应用程序只能将此位置1，而模块会将其清零。如果NAK位、全局OUT NAK与此位同时置1，则STALL位优先。只有应用程序能够将此位清零，而模块则不能。	R/W
b20	SNPM	监听模式	监听模式 (Snoop mode) 此位用于将端点配置为监听模式。在监听模式下，模块不会再检查接收数据的正确性。	R/W
b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 以下是这个逻辑端点支持的传输类型。 00: 控制 01: 同步 10: 批量 11: 中断	R/W
b17	NAKSTS	NAK状态	NAK状态 (NAK status) 指示以下结果： 0: 模块根据FIFO状态回复非NAK握手。 1: 模块在此端点上回复NAK握手。 当应用程序或模块将此位置1时： 即使Rx FIFO存在空间可容纳传入数据包，模块也会停止在OUT端点上接收任何数据。 无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R
b16	EONUM/ DPID	偶数/奇数帧/ 端点数据PID	偶数/奇数帧 (Even/odd frame) 仅适用于同步OUT 端点。 指示模块为此端点发送/接收同步的数据所在的帧的编号。应用程序必须通过此寄存器中的SEVNFRM和SODDFRM字段对偶数/奇数帧编号进行编程，以便此端点发送/接收同步数据。 0: 偶数帧 1: 奇数帧 DPID: 端点数据PID (Endpoint data PID) 仅适用于中断/批量OUT端点。 包含此端点上将要接收或发送的数据包的PID。端点激活后，应用程序必须对要在此端点上接收或发送的首个数据包的PID进行编程。应用程序使用SD0PID 寄存器字段对 DATA0或DATA1 PID 进行编程。 0: DATA0 1: DATA1	R
b15	USBAEP	USB活动端点	USB活动端点(USB active endpoint) 指示此端点在当前配置和接口中是否激活。检测到USB复位后，模块会为所有端点（端点0除外）将此位清零。接收到SetConfiguration和 SetInterface命令后，应用程序必须相应地对端点寄存器进行编程并将此位置1。	R/W
b14~b11	Reserved	-	必须保持复位值。	R/W
b10~b0	MPSIZ	最大数据包大小	最大数据包大小(Maximum packet size) 应用程序必须将此字段编程为当前逻辑端点的最大数据包大小。 此值以字节为单位。	R/W

**34.7.4.18 USBFS 设备 OUT 端点 x 中断寄存器 (USBFS\_DOEPINTx) (x=0..5)**

USBFS Device OUT endpoint x interrupt register

偏移地址：0xb08 + (端点编号 × 0x20)

复位值：0x0000 0080

此寄存器指示端点在出现 USB 和 AHB 相关事件时的状态。当 USBFS\_GINTSTS 寄存器中的 OUT 端点中断位 (USBFS\_GINTSTS 中的 OEPINT 位) 置 1 时，应用程序必须读取此寄存器。在应用程序能够读取此寄存器之前，必须先读取 USBFS\_DAINT 寄存器，以获取 USBFS\_DOEPINTx 寄存器的准确端点编号。应用程序必须将此寄存器中的相应位清零，才能将 USBFS\_DAINT 和 USBFS\_GINTSTS 寄存器中的对应位清零。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Reserved																
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Reserved										B2BS TUP	Rese rved	OTE P DIS	STUP	Rese rved	EPDI SD	XFRC
<hr/>																
位	标记	位名	功能	读写												
b31~b7	Reserved	-	必须保持复位值。	R/W												
b6	B2BSTUP	接收到连续的 SETUP数据包	接收到连续的SETUP数据包 (Back-to-back SETUP packets received) 仅适用于控制OUT端点。此位指示该端点已接收到三个以上的连续SETUP数据包。软件写1也可将此位清零。	R/W												
b5	Reserved	-	必须保持复位值。	R/W												
b4	OTEPDIS	端点禁止时接收到 OUT令牌	端点禁止时接收到OUT令牌 (OUT token received when endpoint disabled) 仅适用于控制OUT端点。指示在尚未使能端点时接收到OUT令牌，从而产生中断。通过软件写1清零。	R/W												
b3	STUP	SETUP阶段完成	SETUP 阶段完成 (SETUP phase done) 仅适用于控制OUT端点。指示控制端点的SETUP阶段已完成，当前控制传输中不再接收到连续的 SETUP数据包。在此中断上，应用程序可以对接收到的 SETUP数据包进行解码。 通过软件写1清零。	R/W												
b2	Reserved	-	必须保持复位值。	R/W												
b1	EPDISD	端点禁止中断	端点禁止中断 (Endpoint disabled interrupt) 此位指示该端点已经由应用程序禁止掉。 通过软件写1清零。	R/W												
b0	XFRC	传输完成中断	传输完成中断 (Transfer completed interrupt) 此字段指示在此端点上设置的传输已经在USB和AHB上传输完成。 通过软件写1清零。	R/W												

**34.7.4.19 USBFS 设备 OUT 端点 0 传输大小寄存器(USBFS\_DOEPTSIZ0)**

USBFS Device OUT endpoint 0 transfer size register

偏移地址：0xB10

复位值：0x0000 0000

在使能端点 0 之前，应用程序必须修改此寄存器。通过设备控制端点 0 控制寄存器中的端点使能位（USBFS\_DIEPCTL0 中的 EPENA）使能端点 0 后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

非零端点使用端点 1~15 的寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved	STUPCNT[1:0]		Reserved						PKTCNT	Reserved					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								XFRSIZ[6:0]							
<hr/>															
位	标记	位名	功能	读写											
b31	Reserved	-	必须保持复位值。	R/W											
b30~b29	STUPCNT	SETUP数据包计数	SETUP数据包计数 (SETUP packet count) 此字段指定端点能连续接收的SETUP数据包数量。 01: 1个数据包 10: 2个数据包 11: 3个数据包	R/W											
b28~b20	Reserved	-	必须保持复位值。	R/W											
b19	PKTCNT	数据包计数	数据包计数 (Packet count) 一次传输中应该接收到数据包数量。 在端点使能前，软件设置该位，在传输开始后，每当数据包接收到后，该域数值自动减少。	R/W											
b18~b7	Reserved	-	必须保持复位值。	R/W											
b6~b0	XFRSIZ	传输大小	传输大小(Transfer size) 指示端点0的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。 每次从Rx FIFO读取数据包并将其写入外部存储器时，模块会使此字段递减。	R/W											

### 34.7.4.20 USBFS 设备 OUT 端点 x 传输大小寄存器(USBFS\_DOEPTSIZx)(x=1..5)

USBFS Device OUT endpoint x transfer size register

偏移地址: 0xB10 + (端点编号 × 0x20)

复位值: 0x0000 0000

在使能该端点之前，应用程序必须修改此寄存器。通过 USBFS\_DOEPCTLx 寄存器中的端点使能位（USBFS\_DOEPCTLx 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	RXDPID[1:0]/ STUPCNT[1:0]						PKTCNT[9:0]						XFRSIZ[18:16]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XFRSIZ[15:0]															

位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R/W
b30~b29	RXDPID/ STUPCNT	接收到的数据PID/ SETUP数据包计数	接收到的数据PID (Received data PID) 仅适用于同步 OUT 端点。 这是此端点收到的上一个数据包的 PID。 00: DATA0 01: DATA2 10: DATA1 11: MDATA STUPCNT: SETUP数据包计数 (SETUP packet count) 仅适用于控制OUT端点。 此字段指定端点能连续接收的SETUP数据包数量。 01: 1 个数据包 10: 2 个数据包 11: 3 个数据包	R/W
b28~b19	PKTCNT	数据包计数	数据包计数 (Packet count) 指示该端点上的一次数据传输包含的数据包个数。 每次向Rx FIFO写入数据包 (最大大小或短数据包) 后，此字段将递减。	R/W
b18~b0	XFRSIZ	传输大小	传输大小(Transfer size) 此字段包含当前端点的一次数据传输包含的数据量，以字节为单位。仅当应用 程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端 点的最大数据包大小，以在每个数据包结束时中断。 每次从Rx FIFO读取数据包并将其写入外部存储器时，模块会使此字段递减。	R/W

### 34.7.4.21 USBFS 设备 OUT 端点 x DMA 地址寄存器 (USBFS\_DOEPDMAx)(x=0..5)

USBFS Device OUT endpoint x transfer size register

偏移地址：0xB14 + (端点编号 × 0x20)

复位值：0xFFFF XXXX

该寄存器用于设定设备端点 DMA 模式时 DMA 地址。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DMAADDR[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DMAADDR[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b0	DMAADDR	DMA地址	DMA地址(DMA address) 该位包含使用DMA进行端点上数据发送时的外部存储区起始地址。  注意：对于控制端点，该字段所指向的存储区也用于存储控制OUT数据包以及SETUP事务数据包。连续接收到三个以上的SETUP数据包时，存储器中的SETUP数据包将被覆盖。每次进行AHB传输，该寄存器都会递增。应用程序必须设定一个双字对齐地址。	R/W											

### 34.7.5 USBFS 时钟门控控制寄存器

通过门控时钟控制寄存器控制 HCLK 和 PHY 时钟从而降低功耗。除非特别说明，否则寄存器描述中的位值以二进制表示。

#### 34.7.5.1 USBFS 时钟门控控制寄存器(USBFS\_GCCTL)

偏移地址：0xE00

复位值：0x0000 0000

此寄存器在主机模式和设备模式下均可用。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Reserved																
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Reserved																
SUSP	PHYSLEEP	ENL1GTG	GATEHCLK	STPPCLK												
位	标记	位名	功能	读写												
b31~b8	Reserved	-	必须保持复位值。	R/W												
b7	SUSP	Deep Sleep状态位	Deep Sleep (L1 suspended) 状态位	R												
b6	PHYSLEEP	PHY SLEEP模式状态位	PHY SLEEP模式状态位 (PHY in sleep)	R												
b5	ENL1GTG	L1时钟门控	L1时钟门控 (Enable Sleep Clock Gating) 该位用于表示L1模式时时钟门控控制。	R/W												
b4~b2	Reserved	-	必须保持复位值。	R/W												
b1	GATEHCLK	门控HCLK	门控HCLK (Gate HCLK) 当USB通信挂起或会话无效时，应用程序会将此位置 1，以停止对除AHB总线从接口、主接口和唤醒逻辑之外的模块提供时钟。当USB恢复通信或新会话启动时，应用程序将此位清零。	R/W												
b0	STPPCLK	停止PHY时钟	停止PHY时钟(Stop PHY clock) 当USB通信挂起、会话无效或设备断开连接时，应用程序将此位置1以停止PHY时钟。当USB恢复通信时，应用程序将此位清零。	R/W												

## 35 CAN FD 控制器 (CAN FD)

### 35.1 简介

CAN (Controller Area Network) 总线是一种可以在无主机情况下实现微处理器或者设备之间相互通信的总线标准。

CAN FD 控制器遵循 CAN 总线 CAN2.0(CAN2.0A、CAN2.0B) 和 CAN FD 协议。

CAN 总线控制器可以处理总线上的数据收发，在本产品中，CAN FD 控制器具有 16 组筛选器。筛选器用于为应用程序选择要接收的消息。

CAN FD 控制器中应用程序可通过 1 个高优先级的主发送缓冲器 (Primary Transmit Buffer, 以下简称 PTB) 和 3 个辅发送缓冲器 (Secondary Transmit Buffer, 以下简称 STB) 将发送数据送至总线，由发送调度器决定邮箱发送顺序。通过 8 个接收缓冲器 (Receive Buffer, 以下简称 RB) 获取总线数据。3 个 STB 以及 8 个 RB 可以理解为一个 3 级 FIFO 和一个 8 级 FIFO，FIFO 完全由硬件控制。

CAN FD 总线控制器同时也可以支持时间触发 CAN 通信 (Time-trigger communication)。

#### CAN FD 主要特性：

- 完全支持 CAN2.0A/CAN2.0B/CAN FD 协议。
- CAN2.0 支持最高通信波特率 1Mbit/s
- 支持 1~1/256 的波特率预分频，灵活配置波特率。
- 8 个接收缓冲器
  - FIFO 方式
  - 错误或者不被接收的数据不会覆盖存储的消息
- 1 个高优先主发送缓冲器 PTB
- 3 个副发送缓冲器 STB
  - FIFO 方式
  - 优先级仲裁方式
- 16 组独立的筛选器
  - 支持 11 位标准 ID 和 29 位扩展 ID
  - 可编程 ID CODE 位以及 MASK 位
- PTB/STB 均支持单次发送模式
- 支持静默模式
- 支持回环模式
- 支持捕捉传输的错误种类以及定位仲裁失败位置
- 可编程的错误警告值

- 支持 ISO11898-4 规定时间触发 CAN 以及接收时间戳

## 35.2 CAN FD 系统框图

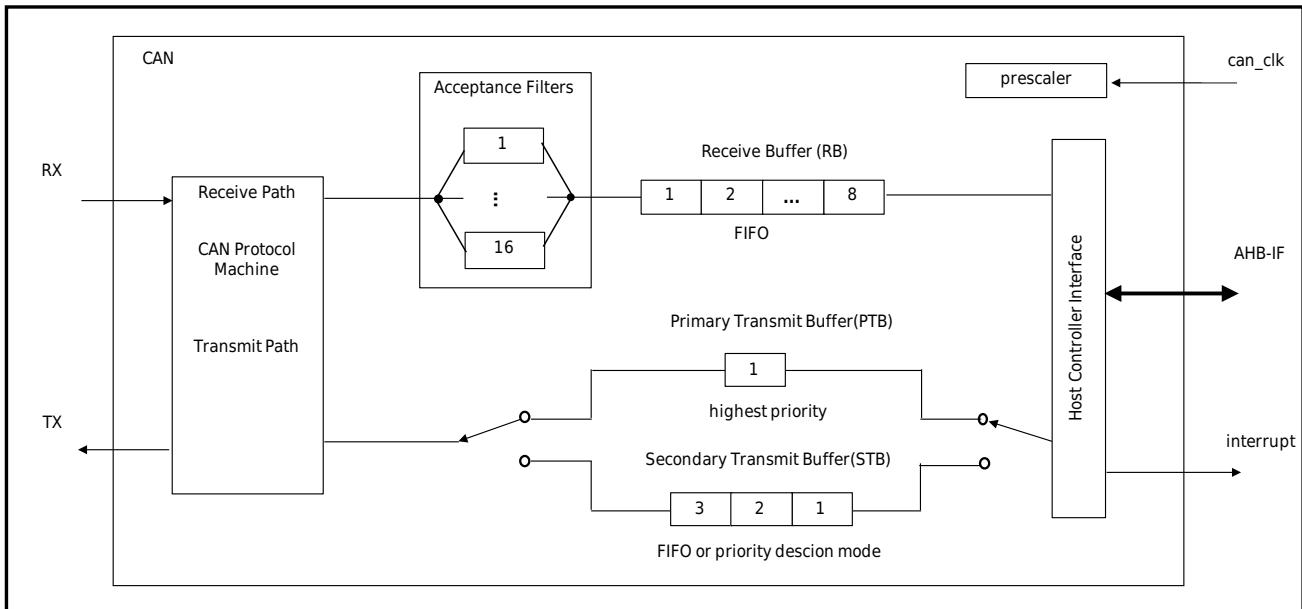


图 35-1 CANFD 系统框图

## 35.3 管脚说明

表 35-1 CAN 管脚说明

管脚名	方向	功能描述
CANx_RX(x=1~3)	输入	CAN接收数据信号
CANx_TX(x=1~3)	输出	CAN发送数据信号
CANx_TST_SAMPLE(x=1~3)	输出	仅用于观测，采样位置（采样点后一周期的通信时钟宽度）
CANx_TST_CLOCK(x=1~3)	输出	仅用于观测，波特率（一位开始前一周期的通信时钟宽度）

## 35.4 功能说明

### 35.4.1 动作模式

使用 CAN FD 功能之前，需先配置 CAN FD 功能使能，详见寄存器章节关于 CAN 全局控制寄存器的说明。CAN FD 控制器存在两个操作模式，复位模式（CAN\_CFG\_STAT.RESET=1）和动作模式（CAN\_CFG\_STAT.RESET=0）。模块初始化时，首先应该在复位模式中设定只能在复位模式下操作的寄存器（详见寄存器说明章节），然后退出复位模式，在动作模式中操作其余寄存器。

### 35.4.2 波特率設定

CAN 通信使用时钟 can\_clk 的时钟源为外部高速振荡器，使用 CAN 模块之前，需要在 CMU 章节设定 CAN 通信时钟。时钟选择时必须满足 PCLK0 (CAN 控制逻辑时钟) 是 can\_clk (CAN 通信时钟) 的 1.5 倍及以上的设定条件。

下图给出 CAN 位时间定义图，虚线上部分为 CAN 协议规定的位时间，虚线下部分为本 CAN 控制器 CAN-CTRL 定义的位时间。其中 segment1 和 segment2 可以通过寄存器 SBT 和 FBT 设定。SBT 寄存器和 FBT 寄存器只能在 CAN\_CFG\_STAT.RESET=1 即 CAN 软件复位时设定。SBT 寄存器用于 CAN2.0 和 CAN FD 的仲裁段，FBT 寄存器用于 CAN FD 数据段。

FD 通信时，建议通信时钟选取 20MHz/40MHz/80MHz。

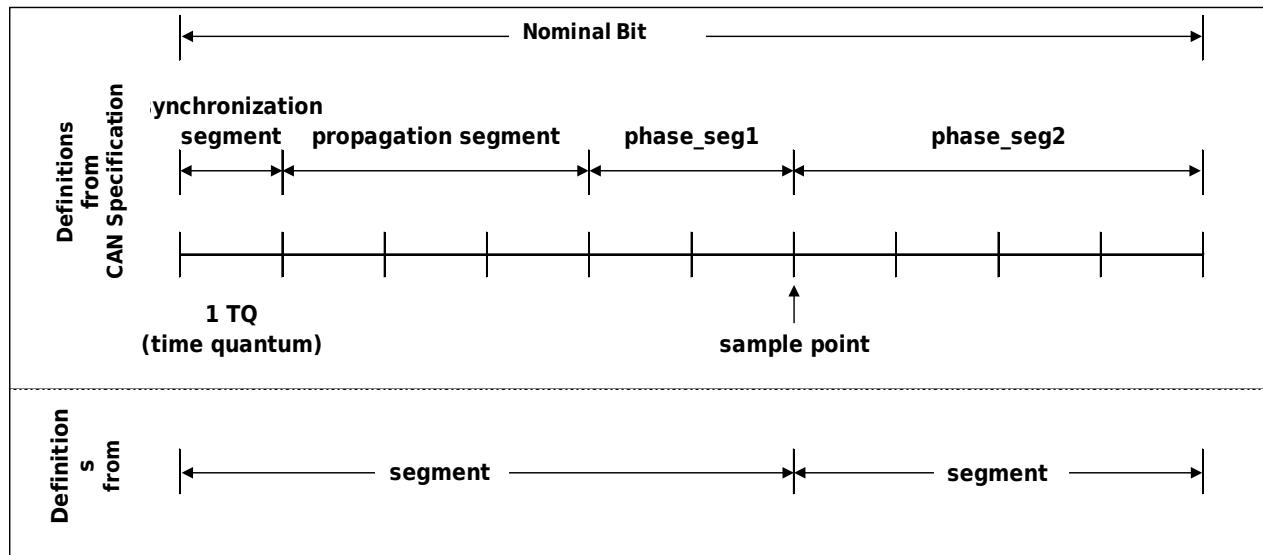


图 35-2 CAN 位时间定义图

TQ 计算方法请参考以下公式，其中 PRESC 通过 SBT 寄存器的 S\_PRESC 位或 FBT 寄存器的 F\_PRESC 位设定。 $f_{can\_clk}$  为 CAN 通信时钟频率。

$$S_{\_TQ} = \frac{S_{\_PRESC} + 1}{f_{can\_clk}}$$

$$F_{TQ} = \frac{F_{PRESC} + 1}{f_{can\_clk}}$$

位时间计算方法请参考以下公式，其中 S\_segment1 和 S\_segment2 通过 SBT 寄存器的 S\_SEG\_1 位和 S\_SEG\_2 位设定, F\_segment1 和 F\_segment2 通过 FBT 寄存器的 F\_SEG\_1 位和 F\_SEG\_2 位设定。

$$\text{Slow_BT} = t_{S\_segment1} + t_{S\_segment2} = ((S\_SEG\_1+2)+(S\_SEG\_2+1)) \times S_{TQ}$$

$$\text{Fast_BT} = t_{F\_segment1} + t_{F\_segment2} = ((F\_SEG\_1+2)+(F\_SEG\_2+1)) \times F_{TQ}$$

表 35-2 CAN 位时间设定规则

位	设定范围			规则
SBT寄存器的S_SEG_1位	[0..63]	CAN2.0 bits	(slow)	SEG_1≥SEG_2+1 SEG_2≥SJW
	[0..63]	CAN FD nominal bits	(slow)	
SBT寄存器的S_SEG_2位	[0..7]	CAN2.0 bits	(slow)	
	[0..31]	CAN FD nominal bits	(slow)	
SBT寄存器的S_SJW位	[0..15]	CAN2.0 bits	(slow)	SEG_1≥SEG_2 SEG_2≥SJW
	[0..15]	CAN FD nominal bits	(slow)	
FBT寄存器的F_SEG_1位	[0..15]	CAN FD data bits	(fast)	
FBT寄存器的F_SEG_2位	[0..7]	CAN FD data bits	(fast)	
FBT寄存器的F_SJW位	[0..7]	CAN FD data bits	(fast)	

以下给出 CANFD 的波特率设定推荐，仅供参考。

PSP : Primary Sample Point

SSP : Secondary Sample Point

Seg 1 : Segment 1

Seg 2 : Segment 2

TDC : Transmitter Delay Compensation

**表 35-3 20MHz 通信时钟时波特率设定建议**

<b>Bit Rate [Mbit/s]</b>	<b>PSP [%]</b>	<b>SSP [%]</b>	<b>Prescaler</b>	<b>Bit Time [TQ]</b>	<b>Seg 1 [TQ]</b>	<b>Seg 2 [TQ]</b>	<b>SJW [TQ]</b>	<b>TDC [CAN 通信时钟]</b>
0.25 (仲裁)	80	-	1	80	64	16	16	-
0.5 (仲裁)	80	-	1	40	32	8	8	-
0.5	80	禁止	1	40	32	8	8	-
1	80	80	1	20	16	4	4	16
2	80	80	1	10	8	2	2	8
4	80	80	1	5	4	1	1	4
5	75	75	1	4	3	1	1	3

**表 35-4 40MHz 通信时钟时波特率设定建议**

<b>Bit Rate [Mbit/s]</b>	<b>PSP [%]</b>	<b>SSP [%]</b>	<b>Prescaler</b>	<b>Bit Time [TQ]</b>	<b>Seg 1 [TQ]</b>	<b>Seg 2 [TQ]</b>	<b>SJW [TQ]</b>	<b>TDC [CAN 通信时钟]</b>
0.25 (仲裁)	80	-	2	80	64	16	16	-
0.5 (仲裁)	80	-	1	80	64	16	16	-
0.5	80	禁止	2	40	32	8	8	-
1	80	80	1	40	32	8	8	32
2	80	80	1	20	16	4	4	16
4	80	80	1	10	8	2	2	8
5	75	75	1	8	6	2	2	6
8	80	80	1	5	4	1	1	4

**表 35-5 80MHz 通信时钟时波特率设定建议**

<b>Bit Rate [Mbit/s]</b>	<b>PSP [%]</b>	<b>SSP [%]</b>	<b>Prescaler</b>	<b>Bit Time [TQ]</b>	<b>Seg 1 [TQ]</b>	<b>Seg 2 [TQ]</b>	<b>SJW [TQ]</b>	<b>TDC [CAN 通信时钟]</b>
0.25 (仲裁)	80	-	4	80	64	16	16	-
0.5 (仲裁)	80	-	2	80	64	16	16	-
0.5	80	禁止	4	40	32	8	8	-
1	80	80	2	40	32	8	8	64
2	80	80	2	20	16	4	4	32
4	80	80	1	20	16	4	4	16
5	75	75	1	16	12	4	4	12
8	80	80	1	10	8	2	2	8

### 35.4.3 发送缓冲器

CAN\_CTRL 提供两种发送缓冲器用于发送数据，主发送数据缓冲器 PTB 和副发送缓冲器 STB。PTB 具有最高的优先级，但只能缓冲一帧数据。STB 优先级比 PTB 低，但可以缓冲 3 帧数据，且 STB 内 3 帧数据可以工作在 FIFO 模式或者优先级仲裁模式。

STB 中的 3 帧说数据可以通过 TCMD 寄存器的 TSALL 位设定为 1 全部发送，在 FIFO 模式下，最先写入的数据先发送，在优先级模式下，ID 小的数据先发送。

PTB 中的数据具有最高优先级，所以 PTB 发送能推迟 STB 发送，但是已经赢得仲裁并开始发送的 STB 不能够被 PTB 发送推迟。

PTB 和 STB 可以通过 TBUF 寄存器进行访问。通过 TCMD 寄存器的 TBSEL 位选择 PTB 或者 STB，TBSEL=0，选择 PTB，TBSEL=1，选择 STB。通过 TCTRL 寄存器的 TSNEXT 位选择 STB 中的下一个 SLOT。对应关系如下图所示：

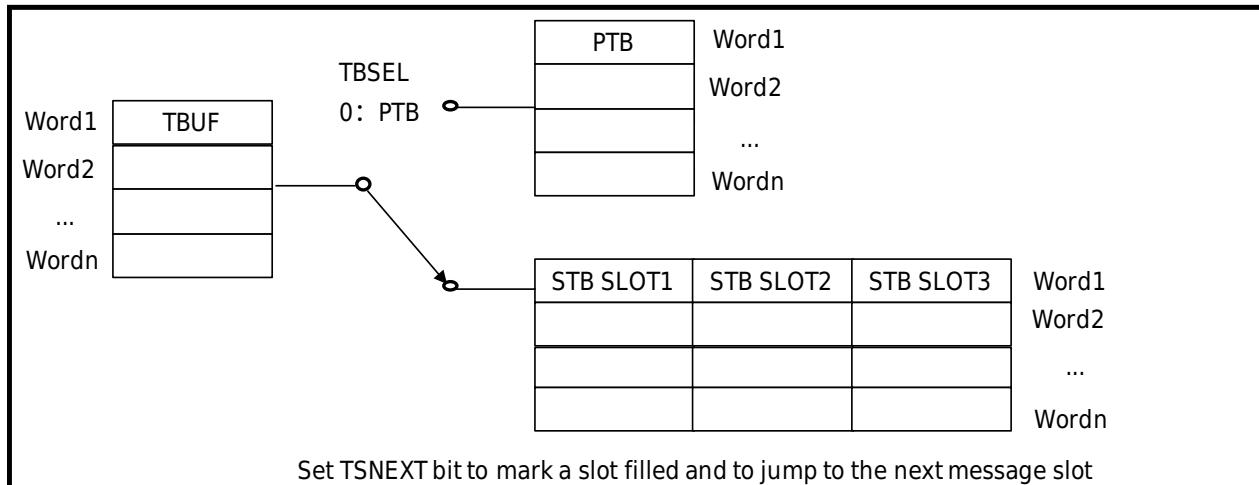


图 35-3 CANFD TBUF 寄存器写发送缓冲器和示意图

### 35.4.4 接收缓冲器

CAN\_CTRL 提供 8 个 SLOT 的接收缓冲器用于存储接收到的数据，该 8 SLOT 的接收缓冲器工作在 FIFO 模式。RB SLOT 通过 RBUF 寄存器来读取接收到的数据，总是最先读取最早接收到的数据，并通过 RCTRL 寄存器的 RREL 设置为 1 释放已经读取的 RB SLOT，并指向下一个 RB SLOT。

通过 RBUF 读取 RB SLOT 示意图如下。

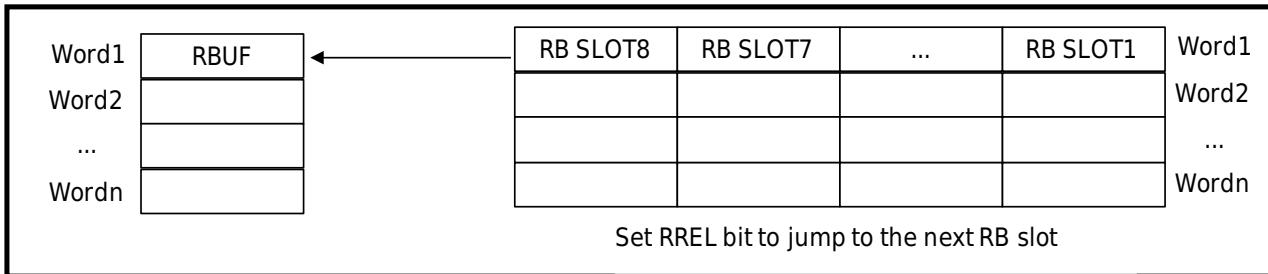


图 35-4 CAN RBUF 寄存器读接收缓冲器示意图

### 35.4.5 接收筛选寄存器组

CAN\_CTRL 提供 16 组 32 位筛选器用于过滤接收到的数据从而降低 CPU 负荷，筛选器可以支持标准格式 11 位 ID 或者扩展格式 29 位 ID。每组筛选器有一个 32 位 ID CODE 寄存器和一个 32 位 ID MASK 寄存器，ID CODE 寄存器用于比较接收到 CAN ID，而 ID MASK 寄存器用于选择比较的 CAN ID 位。对应的 ID MASK 位为 1 时，不比较该位的 ID CODE。

接收到的数据只要通过 16 组筛选器的任意一组，则被接收，接收到的数据存储在 RB 中，否则数据不被接收，也不被存储。

每组筛选器通过 ACFEN 寄存器使能或者禁止。ID CODE 和 ID MASK 通过 ACFCTRL 寄存器的 SELMASK 位设定，SELMASK=0 时，指向 ID CODE，SELMASK=1 时，指向 ID MASK。筛选器通过 ACFCTRL 寄存器的 ACFADR 位选择。ID CODE 和 ID MASK 通过 ACF 寄存器访问且只能在 CFG\_STAT.RESET=1 即 CAN 软件复位时设定。ACF 寄存器访问筛选寄存器组的方式请参考下图。

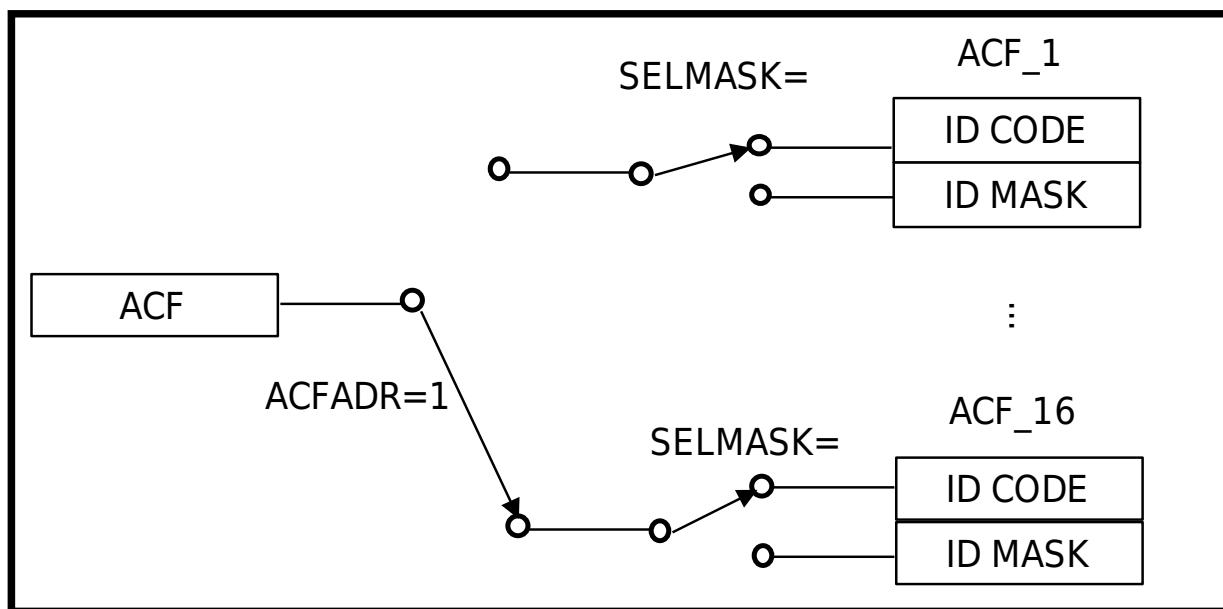


图 35-5 CAN ACF 寄存器访问筛选器组示意图

### 35.4.6 数据发送

在开始发送前必须保证 PTB 或者 STB 待发送的数据已被填充完毕，然后再启动 PTB 或者 STB 发送。发送过程中不允许再次填充数据。

发送数据设定步骤如下：

1. 设定 TBSEL 从 PTB 和 STB 中选择发送 BUF
2. 通过 TBUF 寄存器写需要发送的数据
3. 如果选择的时 STB，设置 TSNEXT=1 以完成全部 STB SLOT 的装载
4. 发送使能
  - PTB 发送使用 TPE
  - STB 发送使用 TSALL 或者 TSONE
5. 发送完成状态确认
  - PTB 发送完成使用 TPIF，TPIE 用于使能 TPIF
  - STB 采用 TSONE 发送完成时使用 TSIF，TSIE 用于使能 TSIF
  - STB 采用 TSALL 发送完成时使用 TSIF，此时需要设定的全部 STB SLOT 数据发送完成后，TSIF 才置位，TSIE 用于使能 TSIF

### 35.4.7 单次数据发送

不需要使用自动重新发送功能时，可以通过寄存器设定为单次发送模式，CFG\_STAT 寄存器的 TPSS 位用于 PTB 的单次发送模式设定，TSSS 位用于 STB 的单次发送模式。数据成功发送时单次发送和正常发送模式时动作相同。但是数据没有成功发送时会出现以下结果：

- TPIF 置位 (TPIE=1)，对应的 BUF SLOT 数据会被清除
- 有错误发送时，KOER 更新，BEIF 置位 (BEIE=1)
- 仲裁失败，ALIF 置位 (ALIE=1)

单次发送模式，不能单独依靠 TPIF 来判断发送完成，需要同 BEIF 和 ALIF 一起判断发送是否完成。

### 35.4.8 取消数据发送

可以通过 TPA 或者 TSA 取消已请求但还没有被执行的数据发送。取消数据发送会出现以下几种情况：

- 仲裁中
  - 节点仲裁失败，则取消数据发送
  - 节点仲裁成功，则继续发送
- 数据发送中
  - 成功发送数据且收到 ACK，对应的标志和状态正常置位。数据发送不取消
  - 成功发送数据但没有收到 ACK，数据发送取消，错误计数器增加

- TSALL=1 设定的发送数据，正在发送的 STB SLOT 数据正常发送，没有开始发送的 STB SLOT 被取消

取消数据发送的结果有以下两种情况：

- TPA 释放 PTB，且使 TPE=0
- TSA 释放一个 STB SLOT 或者全部 STB SLOT 取决是 TSONE 还是 TSALL 使能的发送

### 35.4.9 数据接收

接收筛选器组可以过滤掉不需要的接收数据，减少中断的发生和 RB 的读取，从而降低 CPU 负荷。接收数据设定步骤如下：

1. 设定筛选器组
2. 设定 RFIE，RAFIE 和 AFWL
3. 等待 RFIF 或者 RAFIF
4. 通过 RBUF 从 RB FIFO 中读取最早接收到的数据
5. 设置 RREL=1，选择下一个 RB SLOT
6. 重复 4, 5 直到通过 RSTAT 确认 RB 为空

### 35.4.10 错误处理

CAN\_CTRL 一方面可以自动处理部分错误，比如自动重发数据或者丢弃接收到含有错误的帧，另一方面通过中断将错误向 CPU 报告。

CAN 节点有以下三种错误状态：

- 错误主动：节点检测到错误时自动发送主动错误标志
- 错误被动：节点检测到错误时自动发送被动错误标志
- 节点关闭：关闭状态下此节点不再影响整个 CAN 网络

CAN\_CTRL 提供 TECNT 和 RECNT 两个计数器用于计数错误。TECNT 和 RECNT 计数器按照 CAN 协议规定的规则进行增减。另外提供可编程的 CAN 错误警告 LIMIT 寄存器用于产生错误中断通知 CPU。

CAN 通信过程中有以下 5 种错误类型，错误类型可以通过 EALCAP 寄存器的 KOER 位识别。

- 位错误
- 形式错误
- 填充错误
- 应答错误
- CRC 错误

### 35.4.11 节点关闭

当发送错误数大于 255 时，CAN 节点自动进入节点关闭状态从而不参与 CAN 通信，直到返回到错误主动状态。可以通过 CFG\_STAT 寄存器的 BUSOFF 位确认 CAN 节点关闭状态。BUSOFF 被置位的同时 EIF 中断产生。

CAN 从节点关闭状态恢复到错误主动状态有以下两种方法：

- 上电复位
- 接收到连续 128 个 11 位的隐形位序列（恢复序列）

节点关闭状态下，TECNT 值保持不变，RECNT 用于计数恢复序列。从节点关闭状态恢复后，TECNT 和 RECNT 被复位为 0。

### 35.4.12 仲裁失败位置捕捉

CAN\_CTRL 能够精确捕捉到仲裁失败位的位置并反映到 ALC 寄存器中。ALC 寄存器中保存着最近一次仲裁失败位的位置，如果节点赢得仲裁，则 ALC 位不更新。

**ALC 值定义如下：**

SOF 位后，第一个 ID 数据位 ALC 为 0，第二个 ID 数据位 ALC 为 1，依次类推。因为仲裁只发生在仲裁场内，所以 ALC 的最大值为 31。比如一个标准格式远程帧和一个扩展帧仲裁，扩展帧在 IDE 位失败，则 ALC=12。

### 35.4.13 回环模式

CAN\_CTRL 支持以下两种回环模式：

- 内部回环
- 外部回环

两种回环模式都可以接收自己发出的数据帧，主要用于测试用途。

内部回环模式，模块内部将接收数据线连接到发送数据线，并且发送数据不输出。内部回环模式下，节点会生成自应答信号以避免 ACK 错误。

外部回环模式保持和收发器的连接因此发送的数据仍能出现在 CAN 总线上，在收发器的帮助下，CAN 能收到自己发送的数据。外部回环模式可以通过 RCTRL 寄存器的 SACK 位来决定是否生成自应答信号，SACK=0 时，不生成自应答信号，SACK=1 时，生成自应答信号。

外部回环模式，SACK=0 时，会出现以下两种情况：

- 其它节点也收到本节点发送的数据帧并发送应答信号，该情况下本节点能够成功收发数据
- 如果没有其它节点返回应答信号，则会产生应答错误，会重新发送数据并增加错误计数器。此时推荐采用单次发送模式

从回环模式返回到正常模式时，除了清除模式位以外，还需要软件复位 CAN\_CTRL。

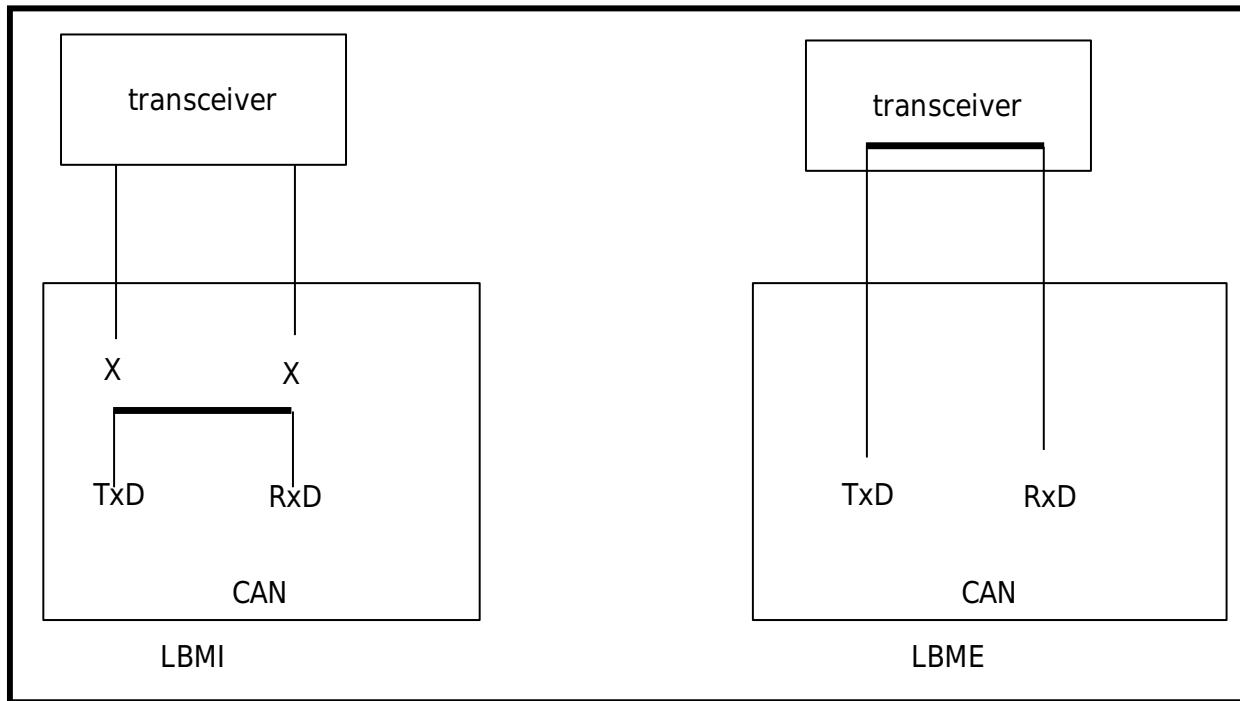


图 35-6 CANFD LBMI 和 LBME 示意图

#### 35.4.14 静默模式

静默模式可以用来监控 CAN 网络数据。在静默模式下，可以从 CAN 总线接收数据，不向总线发送任何数据。将 TCMD 寄存器中的 LOM 置 1，使 CAN 总线控制器进入静默模式，将其清 0 可以离开静默模式。

外部回环模式可以和静默模式组合成外部回环静默模式，此时 CAN 可以认为一个安静的接收者，但在必要的时候可以发送数据。外部回环静默模式下，帧包含自应答信号允许被发送，但是该节点不会产生错误标志和过载帧。

### 35.4.15 软件复位功能

通过设定寄存器 CFG\_STAT 寄存器的 RESET 位为 1，实现软件复位功能，软件复位功能的复位范围如下表所示。

表 35-6 软件复位范围表

寄存器位名	软件复位	备注	寄存器位名	软件复位	备注
ACFADR	否	-	F_SEG_1	是	只能在软件复位时可写
ACODE	否	只能在软件复位时可写	F_SEG_2	是	只能在软件复位时可写
AE_x	否	-	F_SJW	是	只能在软件复位时可写
AFWL	否	-	KOER	是	-
AIF	是	-	LBME	是	-
ALC	是	-	LBMI	是	-
ALIE	否	-	RACTIVE	是	接收立即停止，并不生成 ACK
ALIF	是	-	RAFIE	否	-
AMASK	否	只能在软件复位时可写	RAFIF	是	-
BEIE	否	-	RBALL	是	-
BEIF	是	-	RBUF	是	RB被标记为空，数值不定
BUSOFF	否	通过写1清除	RECNT	否	通过BUSOFF写1清零
EIE_F	否	-	REF_ID	否	-
EIF	否	-	REF_IDE	否	-
EPASS	否	-	RFIE	否	-
EPIE	否	-	RFIF	是	-
EPIF	是	-	RIE	否	-
EWARN	否	-	RIF	是	-
EWL	是	-	ROIE	否	-
FD_ISO	否	只能在软件复位时可写	ROIF	是	-
F_FRESC	否	只能在软件复位时可写	ROM	否	

寄存器位名	软件复位	备注	寄存器位名	软件复位	备注
ROV	是	-	TSMODE	否	
RREL	是	-	TSNEXT	是	-
RSTAT	是		TSONE	是	-
SACK	是	-	TPIE	否	-
SELMASK	否	-	TPIF	是	-
S_PRESC	否	只能在软件复位时可以写	TPSS	是	-
S_SEG_1	否	只能在软件复位时可以写	TSFF	是	所有STB SLOT被标记为

寄存器位名	软件复位	备注	寄存器位名	软件复位	备注
					空
S_SEG_2	否	只能在软件复位时可以写	TSIE	否	-
S_SJW	否	只能在软件复位时可以写	TSIF	是	-
SSPOFF	是	-	TSSS	是	-
TACTIVE	是	发送立即停止	TSSTAT	是	所有STB SLOT被标记为空
TBE	是	-	TTEN	是 是	-
TBF	否	-	TTIF	是	-
TBPTR	否	-	TTIE	否	-
TBSEL	是	-	TTPTR	否	-
TBUF	是	STB被标记为空, 指向PTB	TTTBM	否	-
TDCEN	是	-	TTYPE	否	-
TECNT	否	可通过BUSOFF=1清除	TT_TRIG	否	-
TEIF	是	-	TT_WTRIG	否	-
TPA	是	-	T_PRESC	否	-
TPE	是	-	WTIE	否	-
TSA	是	-	WTIF	是	
TSALL	是	-			

### 35.4.16 向上兼容 CAN-FD 功能

CAN-CTRL 在 CAN FD 功能禁止时即使在包含 CAN FD 网络中收到 CAN FD 的帧, 接收器会自动忽略这些帧, 不返回 ACK, 等到总线空闲时, 再发送或者接收下一个 CAN2.0B 的帧。

### 35.4.17 时间触发 TTCAN

CAN-CTRL 为 ISO11898-4 规定的时间触发通信方式提供部分(lever 1)硬件支持。本章节从以下 5 个部分介绍 TTCAN 功能。

#### 35.4.17.1 TTCAN 模式下的 TBUF 行为

##### TTTBM=1

TTTBM=1 时, PTB 和 STB SLOT 一样组成 TB SLOT, 通过 TBPTR 寄存器指定发送 BUF, 其中 TBPTR=0 时, 指向 PTB, TBPTR=1 是指向 STB SLOT1, 依次类推。主机可以通过 TBE 和 TBF 寄存器来标记发送 BUF SLOT。此时 TBSEL 和 TSNEXT 寄存器无任何意义从而可以被忽略。

TTTBM=1 时, PTB 不具有任何特殊的属性, 和 STB SLOT 一样, 传送完成标志也采用 TSIF。

TTCAN 模式时, 发送 BUF 没有 FIFO 模式和优先级仲裁模式, 同时也只有一个选定的 SLOT 可以发送数据。

TTCAN 模式下，传输开始需要采用时间触发方式，TPE，TSONE，TSALL，TPSS 和 TPA 被固定为 0 且被忽略。

#### TTTB<sub>M</sub>=0

TTTB<sub>M</sub>=0 时，组合使用事件驱动通信和接收时间戳功能。在该模式下，PTB 和 STB 的功能和 TTEN=0 时一致，因此 PTB 始终具有最高的优先级，而 STB 可以工作在 FIFO 模式或者优先级仲裁模式。

#### 35.4.17.2 TTCAN 功能描述

上电后，Time Master 需要根据 ISO 11898-4 协议进行初始化。一个 CAN 网络中，最多可以有 8 个潜在的 Time Master。每一个 Time Master 都具有自己的参考消息 ID（ID 最后 3 位）。这些潜在的 Time Master 根据自己的优先级发送各自的参考消息。

TTEN=1 后，16 位的计数器开始工作，当参考消息被成功接收或者 Time Master 成功发送参考消息时，CAN 控制器将 Sync\_Mark 拷贝给 Ref\_Mark，Ref\_Mark 将 cycle time 设置为 0。成功接收参考消息置位 RIF 标志而成功发送参考消息置位 TPIF 标志或者 TSIF 标志。此时主机需要准备下一个动作的触发条件。

触发条件可以是接收触发。该触发仅触发中断可用于检测期待的消息没有被收到。

触发条件也可以是发送触发。该触发开始发送通过 TTPTR 寄存器指定的 TBUF SLOT 里的数据。如果选定的 TBUF SLOT 被标记为空，则不开始发送，但置位中断标志。

#### 35.4.17.3 TTCAN 时序

CAN\_CTRL 支持 ISO11898-4 level 1。包含的一个 16 位计数器工作在 S\_PRESC，S\_SEG\_1，S\_SEG\_2 定义的位时间下。如果 TTEN=1，则有一个额外的预分频器 T\_PRESC。

一帧数据的 SOF 时，计数器的值为 Sync\_Mark。如果该帧数据为参考消息，则将 Sync\_Mark 拷贝给 Ref\_Mark。cycle time 等于计数器的值减去 Ref\_Mark。该时间用作为接收消息的时间戳或者发送消息的触发时间基准。

#### 35.4.17.4 TTCAN 触发方式

通过 TTTYPE 寄存器定义 TTCAN 的触发方式，TTPTR 寄存器指定发送 SLOT，而 TT\_TRIG 指定触发器的 cycle time。

包含以下五种触发方式：

- 立即触发
- 时间触发
- 单次发送触发
- 发送开始触发
- 发送停止触发

除了立即触发方式外，所有的触发器都使用 TTIF 标志。TTTBM=1 时，只支持时间触发方式。

### 立即触发

通过写 TT\_TRIG 的高位（不在意写入的值），启动触发器。此模式下，TTPTR 选定的 TBUF SLOT 内的数据会立即发送。TTIF 不置位。

### 时间触发

时间触发方式仅通过置位 TTIF 标志产生中断，并无其他功能。如果一个节点期待在特定的时间窗口内收到期待的数据，则可以使用时间触发方式。如果 TT\_TRIG 值小于实际的 cycle time，则 TEIF 置位且无其它动作。

### 单次发送触发

单次发送触发方式用于在执行时间窗口内发送数据。此时，忽略 TSSS 位。

通过 TEW 位设定 ISO11898-4 规定的最多 16 个 cycle time 的 Tick，设定范围为 1~16。如果在规定的发送使能时间窗口内数据没有开始发送，则帧被丢弃。对应的发送 BUF SLOT 被标记为空，并且置位 AIF，对应的发送 BUF 内的数据不会被改写，因为可以通过置位 TPF 再次发送。

如果 TT\_TRIG 值小于实际的 cycle time，则 TEIF 置位且无其它动作。

### 发送开始触发

发送开始触发方式用于仲裁时间窗口内，参与仲裁。TSSS 用于决定是否自动重发或者单次发送模式。如果 TTPTR 寄存器指定的消息没有被成功发送，可以使用发送停止触发来停止该发送。

如果 TT\_TRIG 值小于实际的 cycle time，则 TEIF 置位且无其它动作。

### 发送停止触发

发送停止触发方式用于停止通过发送开始触发方式已经开始的发送。如果发送被停止，则发送帧被舍弃，置位 AIF 并将选定的 TBUF SLOT 标记为空，但 TBUF SLOT 内的数据不会被改写，可以通过置位 TPF 就可以再次发送。

如果 TT\_TRIG 值小于实际的 cycle time 则 TEIF 置位且执行停止。

## 35.4.17.5 TTCAN 触发看门时间

TTCAN 触发看门时间功能类似于看门狗功能，在 TTTBM=1 时使用。用来看门从上次成功接收到参考消息开始的时间。参考消息可以在周期 cycle time 中或者一个事件后被接收，应用程序应该根据具体情况设定合适的看门时间。

如果 cycle count 等于 TT\_WTRIG，则置位 WTIF。通过 WTIE 写 0，关闭看门触发。

如果 TT\_WTRIG 比实际的 cycle time 小，则 TEIF 置位。

### 35.4.18 TDC 和 RDC

CANFD 通信时, 数据的延迟可能超过一位时间, 因此需要补偿。TDC(Transmitter Delay Compensation)和 RDC(Receiver Delay Compensation)用于数据延迟的补偿。其中 TDC 需要软件控制开关, 而 RDC 不需要, 自动有效。TDC 中采用辅助采样点 SSP (Secondary Sample Point) 的方式去补偿数据延迟。如下图:

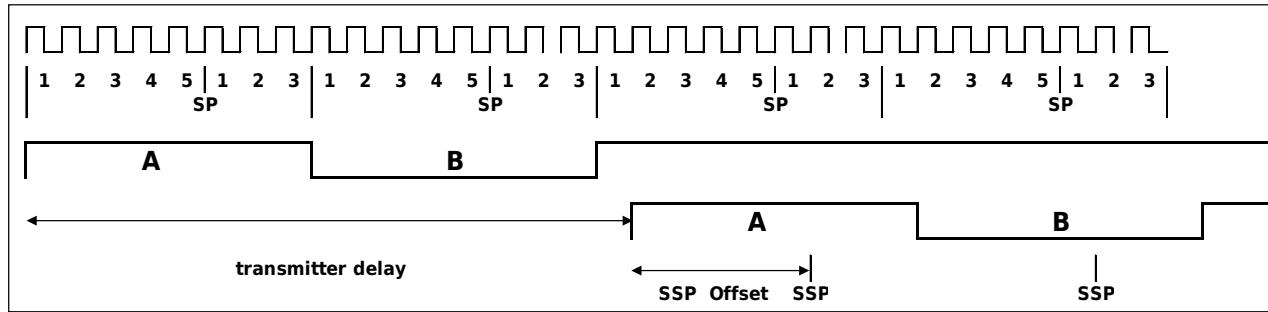


图 35-7 TDC 功能示意图

软件使能 TDC 功能时, 本控制器可以自动确定 transmitter delay, 通过设定寄存器 TDC.SSPOFF 来设定 SSP Offset。建议 SSP Offset 设定值和  $t_{F\_segment1}$  设定值相同。

### 35.4.19 中断

表 35-7 CAN 中断表

符号	中断标志	描述
CAN_x_HOST(x=1~3)	RIF	接收中断
	ROIF	接收上溢中断
	ROIF	接收BUF满中断
	RAFIF	接收BUF将满中断
	TPIF	PTB发送中断
	TSIF	STB发送中断
	EIF	错误中断
	AIF	取消发送中断
	EPIE	错误被动中断
	ALIF	仲裁失败中断
	BEIF	总线错误中断
	WTIF	触发看门中断
	TEIF	触发错误中断
	TTIF	时间触发中断

## 35.5 寄存器说明

表 35-8 CAN 寄存器一览表

CAN 全局控制器地址: 0x4005 5418

寄存器名	符号	地址	位宽	复位值
CAN全局控制寄存器	CAN_SYCTLREG	0x4005 5418	32	0x000 0002

CAN\_1\_BASE\_ADDR: 0x40078000

CAN\_2\_BASE\_ADDR: 0x40071400

CAN\_3\_BASE\_ADDR: 0x40071000

寄存器名	符号	偏移地址	位宽	复位值
CAN接收BUF寄存器	CAN_RBUF	0x00~0x4F	-	0xFFFF XXXX
CAN发送BUF寄存器	CAN_TBUF	0x50~0x97	-	0xFFFF XXXX
CAN配置和状态寄存器	CAN_CFG_STAT	0xA0	8	0x80
CAN命令寄存器	CAN_TCMD	0xA1	8	0x00
CAN发送控制寄存器	CAN_TCTRL	0xA2	8	0x90
CAN接收控制寄存器	CAN_RCTRL	0xA3	8	0x00
CAN接收和发送中断使能寄存器	CAN_RTIE	0xA4	8	0xFE
CAN接收和发送中断标志寄存器	CAN_RTIF	0xA5	8	0x00
CAN错误中断使能和标志寄存器	CAN_ERRINT	0xA6	8	0x00
CAN警告限定寄存器	CAN_LIMIT	0xA7	8	0x1B
CAN Slow位时序寄存器	CAN_SBT	0xA8	32	0x0102 0203
CAN Fast位时序寄存器	CAN_FBT	0xAC	32	0x0102 0203
CAN错误和仲裁失败捕捉寄存器	CAN_EALCAP	0xB0	8	0x00
CAN发送器延迟补偿寄存器	CAN_TDC	0xB1	8	0x00
CAN接收错误计数器寄存器	CAN_RECNT	0xB2	8	0x00
CAN发送错误计数器寄存器	CAN_TECNT	0xB3	8	0x00
CAN筛选器组控制寄存器	CAN_ACFCTRL	0xB4	8	0x00
CAN筛选器组使能寄存器	CAN_ACFEN	0xB6	8	0x01
CAN筛选器组code和mask寄存器	CAN_ACF	0xB8	32	0xFFFF XXXX
TTCAN TB slot指针寄存器	CAN_TBSLOT	0xBE	8	0x00
TTCAN时间触发配置寄存器	CAN_TTCFG	0xBF	8	0x90
TTCAN参考消息寄存器	CAN_REF_MSG	0xC0	32	0xFFFF XXXX
TTCAN触发配置寄存器	CAN_TRG_CFG	0xC4	16	0x0000
TTCAN触发时间寄存器	CAN_TT_TRIG	0xC6	16	0x0000
TTCAN触发看门时间寄存器	CAN_TT_WTRIG	0xC8	16	0xFFFF

### 35.5.1 CAN 全局控制寄存器 (CAN\_SYCTLREG)

CAN System control Register

地址: 0x40055418

复位值: 0x0000 0002

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
<hr/>															
位	标记	位名	功能	读写											
b31~b3	Reserved	-	必须保持复位值	R											
b2	CAN3FDE	CAN_3 FD功能使能位	0: CAN_3 FD功能禁止 1: CAN_3 FD功能使能	R/W											
b1	CAN2FDE	CAN_2 FD功能使能位	0: CAN_2 FD功能禁止 1: CAN_2 FD功能使能	R/W											
b0	CAN1FDE	CAN_1 FD功能使能位	0: CAN_1 FD功能禁止 1: CAN_1 FD功能使能	R/W											

### 35.5.2 CAN 接收 BUF 寄存器 (CAN\_RBUF)

CAN Receive Buffer Registers

偏移地址：0x00

复位值：0xFFFF XXXX

RBUF 寄存器指向最早接收到的 CAN 邮箱的 RB SLOT 地址，RBUF 寄存器可以按照任意顺序读取。

KOER 位即为寄存器 EALCAP.KOER，仅在 RBALL=1 时有意义。

TX 位表示在回环模式下接受到自己发送的消息。

CYCLE\_TIME 位仅在 TTCAN 模式时有效，表示 SOF 开始时的 cycle time。

CAN\_RBUF 仅支持 WORD 访问。

CAN 接收邮箱的数据格式如下：

表 35-9 标准格式 CAN 接收邮箱格式

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
RBUF	ID[7:0]								ID
RBUF+1	-				ID[10:8]				ID
RBUF+2	-								ID
RBUF+3	-								ID
RBUF+4	IDE=0	RTR	FDF	BRS	DLC[3:0]				Control
RBUF+5	KOER[2:0]			TX	-				Status
RBUF+6	CYCLE_TIME[7:0]								TTCAN
RBUF+7	CYCLE_TIME[15:8]								TTCAN
RBUF+8	DATA1								Data
RBUF+9	DATA2								Data
RBUF+10	DATA3								Data
...	...								Data
...	...								Data
...	...								Data
RBUF+70	DATA63								Data
RBUF+71	DATA64								Data

表 35-10 扩展格式 CAN 接收邮箱格式

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
RBUF	ID[7:0]								ID
RBUF+1	ID[15:8] ID[10:8]								ID
RBUF+2	ID[23:16]								ID
RBUF+3	-			ID[28:24]					ID
RBUF+4	IDE=1	RTR	FDF	BRS	DLC[3:0]				Control
RBUF+5	KOER[2:0] TX			TX	-				Status
RBUF+6	CYCLE_TIME[7:0]								TTCAN
RBUF+7	CYCLE_TIME[15:8]								TTCAN
RBUF+8	DATA1								Data
RBUF+9	DATA2								Data
RBUF+10	DATA3								Data
...	...								Data
...	...								Data
...	...								Data
RBUF+70	DATA63								Data
RBUF+71	DATA64								Data

控制位含义如下：

IDE(IDentifier Extension)

0：标准格式

1：扩展格式

RTR(Remote Transmission Request)

0：数据帧

1：远程帧（仅适用于 CAN2.0,CAN FD 时固定为 0）

FDF(CAN FD Frame)

0：CAN2.0 帧

1：CAN FD 帧

BRS(Bit Rate Switch)

0：整帧为低速波特率

1：数据和 CRC 为快速波特率（仅适用于 CAN FD，FDF=0 时固定为 0）

## DLC(Data Length Code)

数据长度码，CAN2.0 设定范围为 0~8，对应数据长度为 0Byte~8Byte，CAN FD 设定范围为 0~15，对应数据长度为 0Byte~64Byte。

表 35-11 DLC 控制位

DLC(二进制)	Frame Type	Playload in Bytes
0000~1000	CAN2.0 和 CAN FD	0~8
1001~1111	CAN2.0	8
1001	CAN FD	12
1010	CAN FD	16
1011	CAN FD	20
1100	CAN FD	24
1101	CAN FD	32
1110	CAN FD	48
1111	CAN FD	64

状态位含义如下：

KOER：同 EALCAP.KOER

TX：回环模式中接收自己发送的数据时此位置 1

### 35.5.3 CAN 发送 BUF 寄存器 (CAN\_TBUF)

CAN Transmit Buffer Registers

偏移地址：0x50

复位值：0xFFFF XXXX

TBUF 寄存器指向下一个空的 CAN 发送 BUF SLOT，TBUF 寄存器可以按照任意顺序读取。通过软件将 TSNEXT 写 1 来标记对应的 TBUF SLOT 已经写入数据，从而指向下一个 TBUF SLOT。

TBUF 只能 WORD 访问。

CAN 发送邮箱的数据格式如下：

表 35-12 标准格式 CAN 发送邮箱格式

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
TBUF	ID[7:0]								ID
TBUF+1	-				ID[10:8]				ID
TBUF+2	-								ID
TBUF+3	-								ID
TBUF+4	IDE=0	RTR	FDF	BRS	DLC[3:0]				Control
TBUF+5	-								-
TBUF+6	-								-
TBUF+7	-								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
...	...								Data
...	...								Data
...	...								Data
TBUF+70	DATA63								Data
TBUF+71	DATA64								Data

表 35-13 扩展格式 CAN 发送邮箱格式

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
TBUF	ID[7:0]								ID
TBUF+1	ID[15:8]								ID
TBUF+2	ID[23:16]								ID
TBUF+3	-			ID[28:24]					ID
TBUF+4	IDE=0	RTR	FDF	BRS	DLC[3:0]				Control
TBUF+5	-								-
TBUF+6	-								-
TBUF+7	-								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
...	...								Data
...	...								Data
...	...								Data
TBUF+70	DATA63								Data
TBUF+71	DATA64								Data

控制位含义请参考 CAN 接收 BUF 寄存器章节说明。

### 35.5.4 CAN 配置和状态寄存器 (CAN\_CFG\_STAT)

CAN Configuration and Status Register

偏移地址：0xA0

复位值：0x80

b7	b6	b5	b4	b3	b2	b1	b0				
RESET	LBME	LBMI	TPSS	TSSS	RACTIVE	TACTIVE	BUSOFF				
<hr/>											
位	标记	位名	功能				读写				
<hr/>											
复位请求位											
0: 不请求局部复位											
1: 请求局部复位											
b7	RESET	复位请求	部分寄存器只能在RESET=1时进行写操作，具体请参考软件复位功能，当该节点进入BUS OFF状态时，硬件自动将RESET位置1。请注意，当RESET=0后需要11个CAN bit times该节点才能参与通信。				R/W				
<hr/>											
外部回环模式使能位											
0: 禁止外部回环模式											
1: 使能外部回环模式											
注意：通信中禁止设定该位。											
<hr/>											
内部回环模式使能位											
0: 禁止内部回环模式											
1: 使能内部回环模式											
注意：通信中禁止设定该位。											
<hr/>											
PTB单次传输模式											
b4	TPSS	PTB单次传输模式	0: 禁止PTB单次传输模式				R/W				
1: 使能PTB单次传输模式											
<hr/>											
STB单次传输模式											
0: 禁止STB单次传输模式											
1: 使能STB单次传输模式											
<hr/>											
接收中状态信号											
b2	RACTIVE	接收中状态信号	0: 非接收中				R				
1: 接收中											
<hr/>											
发送中状态信号											
b1	TACTIVE	发送中状态信号	0: 非发送中				R				
1: 发送中											
<hr/>											
总线关闭状态											
0: 总线有效状态											
b0	BUSOFF	总线关闭状态	1: 总线关闭状态				R/W				
注意：写1可以清零TECNT和RECNT寄存器，但仅限用于调试用途。											
<hr/>											

### 35.5.5 CAN 命令寄存器 (CAN\_TCMD)

CAN Command Register

偏移地址：0xA1

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0	
位	标记	位名	功能				读写	
b7	TBSEL	LOM	-	TPE	TPA	TSONE	TSALL	TSA
b7	TBSEL	发送BUF选择位	发送BUF选择位 (Transmit Buffer Select) 0: PTB 1: STB 当TTEN=1&TTTBM=1时，TBSEL被复位成复位值。 注意：写TBUF寄存器或者TSNEXT位时，此位需要保持定值。				R/W	
b6	LOM	静默模式使能位	静默模式使能位 (Listen Only Mode) 0: 禁止静默模式 1: 使能静默模式 LOM=1&LBME=0时禁止发送。 LOM=1&LBME=1时禁止应答相应接收到的帧以及错误帧，但可以发送数据。 注意：通信中禁止设定该位。				R/W	
b5	Reserved	-	必须保持复位值。 PTB发送使能位 (Transmit Primary Enable) 0: 禁止PTB发送 1: 使能PTB发送 此位使能后，PTB中的Mailbox将在下一个可以发送的位置被发送。已经开始的STB发送将继续，但是下一个等待的STB发送会被延迟到PTB发送完成后再进行。 该位写1后将保持为1直到PTB发送完成或者通过TPA取消发送。软件不能通过写0清除该位。 以下情况TPE被硬件复位成复位值： -RESET=1 -BUSOFF=1 -LOM=1&LBME=0 -TTEN=1&TTTBM=1				R/W	
b3	TPA	PTB发送取消位	PTB发送取消位 (Transmit Primary Abort) 0: 不取消 1: 取消已经通过TPE置1请求但还未开始的PTB发送 该位软件写1但是通过硬件清零。通过写1可以清零TPE位，因此和TPE同时写1。 以下情况TPE被硬件复位成复位值： -RESET=1 -BUSOFF=1 -TTEN=1&TTTBM=1				R/W	
b2	TSONE	发送一帧STB数据设定位	发送一帧STB数据设定位 (Transmit Secondary ONE frame) 0: 不发送 1: 发送一帧STB数据 FIFO模式中，发送最早写入的数据，优先级模式里发送最高优先级的数据				R/W	

该位写1后将保持为1直到STB发送完成或者通过TSA取消发送。软件不能通过写0清除该位。

以下情况TSONE被硬件复位成复位值：

- RESET=1
- BUOFF=1
- LOM=1&LBME=0
- TTEN=1&TTBMM=1

发送所有的STB数据设定位 (Transmit Secondary ALL frame)

0: 不发送

1: 发送STB中所有的数据

b1	TSALL	发送所有的STB数据设定位	该位写1后将保持为1直到STB发送完成或者通过TSA取消发送。软件不能通过写0清除该位。	R/W
b0	TSA	STB发送取消位	以下情况TSALL被硬件复位成复位值：  STB发送取消位 (Transmit Secondary Abort) 0: 不取消 1: 取消已经通过TSONE或者TSALL置1请求但还未开始的STB发送 该位通过软件写1但是通过硬件清零。写1可以清零TSONE或者TSALL位。	R/W

### 35.5.6 CAN 发送控制寄存器 (CAN\_TCTRL)

CAN Transmit Control Register

偏移地址：0xA2

复位值：0x90

b7	b6	b5	b4	b3	b2	b1	b0
FD_ISO	TSNEXT	TSMODE	TTTBM	-	-	TSSTAT[1:0]	
<hr/>							
位	标记	位名	功能	读写			
<hr/>							
b7	FD_ISO	CAN FD ISO模式	CAN FD ISO模式(仅能在CAN_CFG_STAT.RESET=1时写入) 0: Bosch CAN FD模式 1: 11898-1:2015 CAN FD模式	R/W			
b6	TSNEXT	下一个STB SLOT	下一个STB (Transmit buffer Secondary NEXT) 0: 无动作 1: 当前STB SLOT已填充, 指向下一个SLOT 应用程序将TBUF中的数据写完后, 通过置位TSNEXT位标识当前STB SLOT已被填充, 从而硬件将TBUF指向下一个STB SLOT。 被TSNEXT位标识的STB SLOT中的数据可以通过TSONE或者TSALL位发送。 该位通过应用程序写1, 硬件清零。 所有的STB SLOT被填满后, TSNEXT保持为1直到有STB SLOT被释放。 注意: TTCAN模式时此位固定为0。	R/W			
b5	TSMODE	STB发送模式	STB发送模式 (Transmit buffer Secondary operation MODE) 0: FIFO模式 1: 优先级模式 FIFO模式根据数据帧写入的先后顺序发送。 优先级模式根据ID自动判断, ID越小, 优先级越高。 无论何种模式, PTB具有最高的优先级。 注意: TSMODE位只能在STB空时设定。	R/W			
b4	TTTBM	TTCAN BUF模式	TTCAN BUF模式 (TTCAN Transmit Buffer Mode) TTEN=0时, TTTBM被忽略。 0: TSMODE决定, PTB和STB 1: 通过TBPTR和TTPTR设定 TTCAN模式时, 只需要接收时间戳功能时, 此位可以设置为0, 通过TSMODE决定使用PTB还是STB。 注意: TSMODE位只能在STB空时设定。	R/W			
b3~b2	Reserved	-	必须保持复位值。	R/W			
<hr/>							
b1~b0	TSSTAT	STB状态	STB状态 (Transmission Secondary Status bits) TTEN=0 或 TTEN=1 & TTTBM=0 00: STB空 01: STB小于等于半满 10: STB大于半满 11: STB满 TTEN=1 且 TTTBM=1 00: PTB和STB空 01: PTB和STB非满	R			

---

10: 保留

11: PTB和STB满

---

### 35.5.7 CAN 接收控制寄存器 (CAN\_RCTRL)

CAN Receive Control Register

偏移地址：0xA3

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
SACK	ROM	ROV	RREL	RBALL	-	RSTAT[1:0]	
<hr/>							
位	标记	位名	功能			读写	
b7	SACK	自应答	自应答 (Self-ACKnowledge) 0: 无自应答 1: LBME=1时，使能自应答功能			R/W	
b6	ROM	接收BUF上溢模式设定位	接收BUF上溢模式设定位 (Receive buffer Overflow Mode) 0: 最早接收到的数据被覆盖 1: 新接收到的数据不被存储			R/W	
b5	ROV	接收BUF上溢标志位	接收BUF上溢标志位 (Receive buffer OVerflow) 0: 无上溢 1: 上溢，最少有一个数据丢失 通过写RREL为1清零。			R	
b4	RREL	释放接收BUF	释放接收BUF (Receive buffer RElease) 0: 不释放 1: 表示该接收BUF已经被读取过，RBUF寄存器指向下一个RB SLOT。			R/W	
b3	RBALL	接收BUF数据存储所有的数据帧	接收BUF数据存储所有的数据帧 (Receive Buffer stores ALL data frames) 0: 正常模式 1: 存储所有的数据包括有错误的数据。			R/W	
b2	Reserved	-	必须保持复位值。			R/W	
b1~b0	RSTAT	接收BUF状态	接收BUF状态 (Receive buffer STATus) 00: RBUF空 01: RBUF非空但小于AFWL编程值 10: RBUF大于等于AFWL编程值但未满 11: 满 (上溢时保持此值)			R	

### 35.5.8 CAN 接收和发送中断使能寄存器 (CAN\_RTIE)

CAN Receive and Transmit Interrupt Enable Register

偏移地址：0xA4

复位值：0xFE

b7	b6	b5	b4	b3	b2	b1	b0
RIE	ROIE	RFIE	RAFIE	TPIE	TSIE	EIE	TSFF

位	标记	位名	功能	读写
b7	RIE	接收中断使能	接收中断使能 (Receive Interrupt Enable ) 0: 禁止 1: 使能	R/W
b6	ROIE	接收上溢中断使能	接收上溢中断使能 (Receive Overrun Interrupt Enable ) 0: 禁止 1: 使能	R/W
b5	RFIE	接收BUF满中断使能	接收BUF满中断使能 (RB Full Interrupt Enable) 0: 禁止 1: 使能	R/W
b4	RAFIE	接收BUF将满中断使能	接收BUF将满中断使能 (RB Almost Full Interrupt Enable) 0: 禁止 1: 使能	R/W
b3	TPIE	PTB发送中断使能	PTB发送中断使能 (Transmission Primary Interrupt Enable) 0: 禁止 1: 使能	R/W
b2	TSIE	STB发送中断使能	STB发送中断使能 (Transmission Secondary Interrupt Enable) 0: 禁止 1: 使能	R/W
b1	EIE	错误中断使能	错误中断使能 (Error Interrupt Enable) 0: 禁止 1: 使能	R/W
b0	TSFF	发送BUF满标志	TTEN=0 or TTTBM=0: STB满标志 (Transmit Secondary buffer Full Flag) 0: STB SLOT没有被全部填充 1: STB SLOT被全部填充 TTEN=1 and TTTBM=1: TB满标志 (Transmit buffer Full Flag) 0: TBPTR选择的发送BUF没有被全部填充 1: TBPTR选择的发送BUF被全部填充	R

### 35.5.9 CAN 接收和发送中断标志寄存器 (CAN\_RTIF)

CAN Receive and Transmit Interrupt Status Register

偏移地址：0xA5

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
RIF	ROIF	RFIF	RAFIF	TPIF	TSIF	EIF	AIF
<hr/>							
位	标记	位名	功能			读写	
b7	RIF	接收中断标志	接收中断标志 (Receive Interrupt Flag) 0: 未收到数据帧 1: 接收到有效的数据帧或者远程帧 通过应用程序写1清0。			R/W	
b6	ROIF	接收上溢中断标志	接收上溢中断标志 (Receive Overrun Interrupt Flag) 0: 无RB被覆盖 (overwrite) 1: RB至少有一个被覆盖 上溢时ROIF和RFIF同时置1。 通过应用程序写1清0。			R/W	
b5	RFIF	接收BUF满中断标志	接收BUF满中断标志 (RB Full Interrupt Flag) 0: RB FIFO未满 1: RB FIFO满 通过应用程序写1清0。			R/W	
b4	RAFIF	接收BUF将满中断标志	接收BUF将满中断标志 (RB Almost Full Interrupt Flag) 0: 被填充的RB SLOT数目小于AFWL设定值 1: 被填充的RB SLOT数目大于等于AFWL设定值 通过应用程序写1清0。			R/W	
b3	TPIF	PTB发送中断标志	PTB发送中断标志 (Transmission Primary Interrupt Flag) 0: 没有PTB发送完成 1: 请求的PTB发送成功完成 通过应用程序写1清0。 注意: TTCAN模式时, TPIF无效, 仅适用TSIF标志			R/W	
b2	TSIF	STB发送中断标志	STB发送中断标志 (Transmission Secondary Interrupt Flag) 0: 没有STB发送完成 1: 请求的STB发送成功完成 通过应用程序写1清0。 注意: TTCAN模式时, TPIF无效, 仅使用TSIF标志			R/W	
b1	EIF	错误中断标志	错误中断标志 (Error Interrupt Flag) 0: BUSOFF位未发生变化, 或者错误计数器的值与ERROR warning limit设定值的相对关系未发生变化。 1: BUSOFF位发生变化, 或者错误计数器的值与ERROR warning limit设定值的相对关系发生变化。比如错误计数器的值从小于设定值变为大于设定值, 或者从大于设定值变为小于设定值。 通过应用程序写1清0。			R/W	
b0	AIF	取消发送中断标志	取消发送中断标志 (Abort Interrupt Flag)			R/W	

---

0: 未取消发送数据

1: 通过TPA和TSA请求的发送消息被成功取消。

通过应用程序写1清0。

---

### 35.5.10 CAN 错误中断使能和标志寄存器 (CAN\_ERRINT)

CAN ERRor INTerrupt Enable and Flag Register

偏移地址：0xA6

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
EWARN	EPASS	EPIE	EPIF	ALIE	ALIF	BEIE	BEIF

位	标记	位名	功能	读写
b7	EWARN	到达设定的ERROR WARNING LIMIT	到达设定的ERROR WARNING LIMIT (Error WARNING limit reached) 0: RECNT或者TECNT小于EWL设定值 1: RECNT或者TECNT大于等于EWL设定值	R
b6	EPASS	错误被动	错误被动 (Error Passive mode active) 0: 节点是主动错误节点 1: 节点时被动错误节点	R
b5	EPIE	错误被动中断使能	错误被动中断使能 (Error Passive Interrupt Enable) 0: 禁止 1: 使能	R/W
b4	EPIF	错误被动中断标志	错误被动中断标志 (Error Passive Interrupt Flag) 0: 未发生错误主动到错误被动或者错误被动到错误主动的变化 1: 发生错误主动到错误被动或者错误被动到错误主动的变化 通过应用程序写1清0。	R/W
b3	ALIE	仲裁失败中断使能	仲裁失败中断使能 (Arbitration Lost Interrupt Enable) 0: 禁止 1: 使能	R/W
b2	ALIF	仲裁失败中断标志	仲裁失败中断标志 (Arbitration Lost Interrupt Flag) 0: 仲裁成功 1: 仲裁失败 通过应用程序写1清0。	R/W
b1	BEIE	总线错误中断使能	总线错误中断使能 (Bus Error Interrupt Enable) 0: 禁止 1: 使能	R/W
b0	BEIF	总线错误中断标志	总线错误中断标志 (Bus Error Interrupt Flag) 0: 无总线错误 1: 总线错误 通过应用程序写1清0。	R/W

### 35.5.11 CAN Slow 位时序寄存器 (CAN\_SBT)

CAN Slow Bit Timing Register

偏移地址：0xA8

复位值：0x0102 0203

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
				S_PRESC[7:0]				-							S_SJW[6:0]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-				S_SEG_2[6:0]											S_SEG_1[7:0]
<hr/>															
位	标记	位名	功能	读写											
b31~b24	S_PRESC	预分频设定	预分频设定 (S_Prescaler) 此寄存器设定值时将模块通信时钟设定为 (S_PRESC+1) 分频作为TQ。	R/W											
b23	Reserved	-	必须保持复位值。	R											
b22~b16	S_SJW	再同步补偿宽度 时间设定	再同步补偿宽度时间设定 (Bit Timing Segment 2) 再同步补偿宽度时间=(S_SJW+1)*TQ	R/W											
b15	Reserved	-	必须保持复位值。	R											
b14~b8	S_SEG_2	位段2时间设定	位段2时间单元设定 (Bit Timing Segment 2) 位段2时间=(S_SEG_2+1)*TQ	R/W											
b7~b0	S_SEG_1	位段1时间设定	位段1时间单元设定 (Bit Timing Segment 1) 位段1时间=(S_SEG_1+2)*TQ	R/W											

### 35.5.12 CAN Fast 位时序寄存器 (CAN\_FBT)

CAN Fast Bit Timing Register

偏移地址：0xAC

复位值：0x0102 0203

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
F_PRESC[7:0]								--	-	-	-	F_SJW[3:0]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	F_SEG_2[3:0]				-	-	-	F_SEG_1[4:0]				

位	标记	位名	功能	读写
b31~b24	F_PRESC	预分频设定	预分频设定 (Prescaler) 此寄存器设定值时将模块通信时钟设定为 (F_PRESC+1) 分频作为TQ。	R/W
b23~b20	Reserved	-	必须保持复位值。	R
b19~b16	F_SJW	再同步补偿宽度 时间设定	再同步补偿宽度时间设定 (Bit Timing Segment 2) 再同步补偿宽度时间=(F_SJW+1)*TQ	R/W
b15~12	Reserved	-	必须保持复位值。	R
b11~b8	F_SEG_2	位段2时间设定	位段2时间单元设定 (Bit Timing Segment 2) 位段2时间=(F_SEG_2+1)*TQ	R/W
b7~b5	Reserved	-	必须保持复位值。	R
b4~b0	F_SEG_1	位段1时间设定	位段1时间单元设定 (Bit Timing Segment 1) 位段1时间=(F_SEG_1+2)*TQ	R/W

### 35.5.13 CAN 发送器延迟补偿寄存器 (CAN\_TDC)

CAN Transmitter Delay Compensation Register

偏移地址：0xb1

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
TDCEN	SSPOFF[6:0]						

位	标记	位名	功能	读写
b7	TDCEN	发送器延迟补偿功能使能位		
		发送器延迟补偿功能使能位	用于CAN FD BRS=1的情况	R/W
		0: 禁止		
		1: 使能		
b6~b0	SSPOFF	TDC功能辅助采样点位置设定位	TDC功能辅助采样点位置设定位	R/W
		样点位置设定位	设定单位为TQ	

### 35.5.14 CAN 错误和仲裁失败捕捉寄存器 (CAN\_EALCAP)

CAN Error and Arbitration Lost Capture Register

偏移地址：0xB0

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
KOER[2:0]				ALC[4:0]			

位	标记	位名	功能	读写
b7~b5	KOER	错误类别	错误类别 (Kind Of Error) 000: 无错误 001: 位错误 010: 形式错误 011: 填充错误 100: 应答错误 101: CRC错误 110: 其他错误 111: 保留 有错误时KOER位更新，正常发送接收时KOER位保持不变。	R
b4~b0	ALC	仲裁失败位置捕捉	仲裁失败位置捕捉 (Arbitration Lost Capture) 仲裁失败时ALC记录一帧数据中仲裁失败时的位置。	R

### 35.5.15 CAN 警告限定寄存器 (CAN\_LIMIT)

CAN Warning Limits Register

偏移地址：0xA7

复位值：0x1B

b7	b6	b5	b4	b3	b2	b1	b0
AFWL[3:0]				EWL[3:0]			

位	标记	位名	功能	读写
b7~b4	AFWL	接收BUF将满Warning Limit	接收BUF将满Warning Limit (receive buffer Almost Full Warning Limit) 设定值范围为1~8。 AFWL=0无意义，当做AFWL=1处理。	R/W
b3~b0	EWL	Error Waring Limit编程值	Error Waring Limit编程值 (Programmable Error Warning Limit) Error Waring Limit= (EWL+1) *8。 该寄存器设定值影响EIF标志。	R/W

### 35.5.16 CAN 接收错误计数器寄存器 (CAN\_RECNT)

CAN Receive Error CouNT Register

偏移地址：0xB2

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
RECNT[7:0]							

位	标记	位名	功能	读写
b7~b0	RECNT	接收错误计数器	接收错误计数器 (Receive Error CouNT) 接收错误计数器根据CAN协议规定的错误计数增加或者减少。该计数器不存在上溢， 255为最大值。	R

### 35.5.17 CAN 发送错误计数器寄存器 (CAN\_TECNT)

CAN Transmit Error CouNT Register

偏移地址：0xB3

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
TECNT[7:0]							

位	标记	位名	功能	读写
b7~b0	TECNT	发送错误计数器	发送错误计数器根据CAN协议规定的错误计数增加或者减少。该计数器不存在上溢，255为最大值。	R

### 35.5.18 CAN 筛选器组控制寄存器 (CAN\_ACFCTRL)

CAN Acceptance Filter Control Register

偏移地址：0xB4

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
-		SELMASK	-				ACFADR

位	标记	位名	功能	读写
b7~b6	Reserved	-	必须保持复位值。	R
b5	SELMASK	选择筛选器的屏蔽寄存器	选择筛选器的屏蔽寄存器 (SElect acceptance MASK) 0: ACF指向筛选器ID寄存器 1: ACF指向筛选器MASK寄存器 通过ACFADR选择具体的筛选寄存器组	R/W
b4	Reserved	-	必须保持复位值。	R
b3~b0	ACFADR	筛选器地址	筛选器地址 (acceptance filter address) ACFADR指向具体的筛选器，通过SELMASK去区分ID和MASK。 0000: 指向ACF_1 0001: 指向ACF_2 0010: 指向ACF_3 0011: 指向ACF_4 0100: 指向ACF_5 0101: 指向ACF_6 0110: 指向ACF_7 0111: 指向ACF_8 1000: 指向ACF_9 1001: 指向ACF_10 1010: 指向ACF_11 1011: 指向ACF_12 1100: 指向ACF_13 1101: 指向ACF_14 1110: 指向ACF_15 1111: 指向ACF_16	R/W

### 35.5.19 CAN 筛选器组使能寄存器 (CAN\_ACFEN)

CAN Acceptance Filter Enable Register

偏移地址：0xB6

复位值：0x0001

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AE_6	AE_5	AE_4	AE_3	AE_2	AE_1	AE_0	AE_9	AE_8	AE_7	AE_6	AE_5	AE_4	AE_3	AE_2	AE_1

位	标记	位名	功能	读写
b15	AE_16	ACF_16使能	ACF_16使能 (Acceptance Filter 16 Enable) 0: 禁止 1: 使能	R/W
b14	AE_15	ACF_15使能	ACF_15使能 (Acceptance Filter 15 Enable) 0: 禁止 1: 使能	R/W
b13	AE_14	ACF_14使能	ACF_14使能 (Acceptance Filter 14 Enable) 0: 禁止 1: 使能	R/W
b12	AE_13	ACF_13使能	ACF_13使能 (Acceptance Filter 13 Enable) 0: 禁止 1: 使能	R/W
b11	AE_12	ACF_12使能	ACF_12使能 (Acceptance Filter 12 Enable) 0: 禁止 1: 使能	R/W
b10	AE_11	ACF_11使能	ACF_11使能 (Acceptance Filter 11 Enable) 0: 禁止 1: 使能	R/W
b9	AE_10	ACF_10使能	ACF_10使能 (Acceptance Filter 10 Enable) 0: 禁止 1: 使能	R/W
b8	AE_9	ACF_9使能	ACF_9使能 (Acceptance Filter 9 Enable) 0: 禁止 1: 使能	R/W
b7	AE_8	ACF_8使能	ACF_8使能 (Acceptance Filter 8 Enable) 0: 禁止 1: 使能	R/W
b6	AE_7	ACF_7使能	ACF_7使能 (Acceptance Filter 7 Enable) 0: 禁止 1: 使能	R/W
b5	AE_6	ACF_6使能	ACF_6使能 (Acceptance Filter 6 Enable) 0: 禁止 1: 使能	R/W
b4	AE_5	ACF_5使能	ACF_5使能 (Acceptance Filter 5 Enable) 0: 禁止 1: 使能	R/W
b3	AE_4	ACF_4使能	ACF_4使能 (Acceptance Filter 4 Enable) 0: 禁止 1: 使能	R/W
b2	AE_3	ACF_3使能	ACF_3使能 (Acceptance Filter 3 Enable) 0: 禁止 1: 使能	R/W
b1	AE_2	ACF_2使能	ACF_2使能 (Acceptance Filter 2 Enable) 0: 禁止 1: 使能	R/W
b0	AE_1	ACF_1使能	ACF_1使能 (Acceptance Filter 1 Enable) 0: 禁止 1: 使能	R/W

### 35.5.20 CAN 筛选器组 code 和 mask 寄存器 (CAN\_ACF)

CAN Acceptance Filter code and mask Register

偏移地址：0xB8

复位值：0xFFFF XXXX

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16		
-	AIDEE	AIDE	ACODE[28:16] or AMASK[28:16]														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		
ACODE[15:0] or AMASK[15:0]																	
<hr/>																	
位	标记	位名	功能	读写													
b31	Reserved	-	读出值为不定。	R													
b30	AIDEE	IDE位比较使能	IDE位比较使能 (Acceptance mask IDE bit check enable) 尽在SELMASK=1时有效 0: 筛选器接收标准格式和扩展格式帧 1: 筛选器接收AIDE位定义的标准格式或者扩展格式帧	R/W													
b29	AIDE	IDE位MASK	IDE位MASK 0: 筛选器仅接收标准格式 1: 筛选器仅接收扩展格式	R/W													
b28~b0	ACODE/ AMASK	筛选器CODE/ 筛选器MASK	筛选器CODE (acceptance filter code) 通过ACFADR指向具体的筛选器。 SELMASK=0时表示筛选器的CODE。 标准格式时使用位10~位0，扩展格式时使用位29~位0。  筛选器MASK (acceptance filter mask) 通过ACFADR指向具体的筛选器。 SELMASK=1时表示筛选器的MASK。 标准格式时使用位10~位0，扩展格式时使用位29~位0。	R/W													

### 35.5.21 TTCAN TB slot 指针寄存器 (CAN\_TBSLOT)

TTCAN TB Slot Pointer Register

偏移地址：0xBE

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
TBE	TBF	-	-	-			TB PTR[2:0]
<hr/>							
位	标记	位名	功能		读写		
b7	TBE	设置TB为空	设置TB为空 (set TB slot to "empty") 0: 无操作 1: 被TB PTR选择的SLOT被标记为空		当SLOT被标记为空并且TSFF=0时, TBE自动复位为0。 如果设定此位为1时, 被选定的SLOT中存在数据正在发送状态则TBE=1, 则等到发送完成、发送错误或者发送取消后TBE复位为0。 TBE优先级高于TBF。		
b6	TBF	设置TB为已填充	设置TB已填充 (set TB slot to "Filled") 0: 无操作 1: 被TB PTR选择的SLOT被标记为已填充		当SLOT被标记为已填充并且TSFF=1时, TBE自动复位为0。		
b5~b3	Reserved	-	必须保持复位值。				
b2~b0	TB PTR	TB SLOT指针	TB SLOT指针 (Pointer to a TB message slot) 000: 指向PTB 001: 指向STB SLOT1 010: 指向STB SLOT2 011: 指向STB SLOT3 其他: 设定禁止		被指向的TB SLOT可以通过TBUF进行读写访问, 并且可以通过TBE和TBF来标记是否已经被填充。 TTCAN模式时, TB SEL和TSNEXT寄存器无效。 注意: 仅可以在TSFF=0时对该位进行写操作。		

### 35.5.22 TTCAN 时间触发配置寄存器 (CAN\_TTCFG)

TTCAN TB Slot Pointer Register

偏移地址：0xBF

复位值：0x90

b7	b6	b5	b4	b3	b2	b1	b0
WTIE	WTIF	TEIF	TTIE	TTIF	T_PRESC[1:0]		TTEN
<hr/>							
位	标记	位名	功能		读写		
b7	WTIE	触发看门中断使能	触发看门中断使能 (Watch Trigger Interrupt Enable)		R/W		
		使能	0: 禁止 1: 使能				
b6	WTIF	触发看门中断标志	触发看门中断标志 (Watch Trigger Interrupt Flag) 当CYCLE COUNT值=TT_WTRIG设定值时且WTIE=1时，WTIF置位。 通过应用程序写1清0。		R/W		
b5	TEIF	触发错误中断标志	触发错误中断标志 (Trigger Error Interrupt Flag) TT_TTIG设定值小于实际的CYCLE_TIME时，TEIF置位。 通过应用程序写1清0。		R/W		
b4	TTIE	时间触发中断使能	时间触发中断使能 (Time Trigger Interrupt Enable) 0: 禁止 1: 使能		R/W		
b3	TTIF	时间触发中断标志	时间触发中断标志 (Time Trigger Interrupt Flag) 当CYCLE COUNT值=TT_TRIG设定值时且TTIE=1时，TTIF置位。 如果TT_TRIG没有更新，则TTIF只置位1次，下一个基本CYCLE不置位。 通过应用程序写1清0。		R/W		
b2~b1	T_PRESC	TTCAN计数器预分频	TTCAN计数器预分频 (TTCAN Timer PREScaler) 00: SBT寄存器设定的位时间的1分频 01: SBT寄存器设定的位时间的2分频 10: SBT寄存器设定的位时间的4分频 11: SBT寄存器设定的位时间的8分频 注意: T_PRESC可在TTEN=0时进行写操作或者写TTEN=1时同时操作。		R/W		
b0	TTEN	TTCAN使能	TTCAN使能 (Time Trigger Enable) 0: 禁止 1: 使能TTCAN，计数器开始计数。		R/W		

### 35.5.23 TTCAN 参考消息寄存器 (CAN\_REF\_MSG)

TTCAN Reference Message Register

偏移地址：0xC0

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
REF_I DE	-	REF_ID[28:16]													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
REF_ID[15:0]															
<hr/>															
位	标记	位名	功能												读写
b31	REF_IDE	参考消息的IDE位 位	参考消息的IDE位 (REFerence message IDE bit) 0: 标准格式 1: 扩展格式												R/W
b30~b29	Reserved	-	读出值为不定。												R/W
b28~b0	REF_ID	参考消息的ID位	参考消息的ID位 (REFerence message IDentifier) REF_IDE=0: REF_ID[10:0]有效 REF_IDE=1: REF_ID[28:0]有效 REF_ID用于检测参考消息，适用于发送和接收。 检测到参考消息后，当前帧的Sync_Mark则变成Ref_Mark。 REF_ID[2:0]固定为0，并不检查其值，这样最多可以支持8个潜在的time master。 当REF_MSG的最高字节写操作后，则需要等待6个CAN时钟周期以完成REF_MSG向 CAN时钟域的传递。												R/W

### 35.5.24 TTCAN 触发配置寄存器 (CAN\_TRG\_CFG)

TTCAN Reference Message Register

偏移地址：0xC4

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TEW[3:0]		-		TTYPE[2:0]				-			TTPTR[2:0]			
<hr/>															
位	标记	位名	功能	读写											
b15~b12	TEW	发送使能窗口	发送使能窗口 (Transmit Enable Window) 用于TTCAN的单次发送触发模式 (Single Shot Transmit Trigger) ,可以设定TEW+1个cycle time的窗口，发送仅在此窗口内被允许。	R/W											
b11	Reserved	-	必须保持复位值。	R											
b10~b8	TTYPE	触发类型	触发类型 (Trigger Type) 000: 立即触发 (Immediate Trigger for immediate transmission) 001: 时间触发 (Time Trigger for receive triggers) 010: 单次发送触发 (Single Shot Transmit Trigger for exclusive time windows) 011: 发送开始触发 (Transmit Start Trigger for merged arbitrating time windows) 100: 发送停止触发 (Transmit Stop Trigger for merged arbitrating time windows) 其他: 保留 触发时间通过TT_TRIGGER寄存器设定，TB Slot通过TTPTR选择。	R/W											
b7~b3	Reserved	-	必须保持复位值。	R											
b2~b0	TTPTR	发送触发器TB slot指针	发送触发器TB slot指针 (Transmit Trigger TB slot Pointer) 000: 指向PTB 001: 指向STB SLOT1 010: 指向STB SLOT2 011: 指向STB SLOT3 其他: 设定禁止 如果指向的TB SLOT被标记为空，当到达触发时间后，TEIF置位。	R/W											

### 35.5.25 TTCAN 触发时间寄存器 (CAN\_TT\_TRIG)

TTCAN Reference Message Register

偏移地址：0xC6

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TT_TRIG[15:0]															

位	标记	位名	功能	读写
			触发时间 (Trigger Time) 用于指定触发器的cycle time，对于发送触发器来说发送SOF时间大约是TT_TRIG设定	
b15~b0	TT_TRIG	触发时间	值+1 当TT_TRIG的最高字节写操作后，TT_TRIG值开始向CAN时钟域的传递。因此如果 BYTE操作，需先写低字节再写高字节。	R/W

### 35.5.26 TTCAN 触发看门时间寄存器 (CAN\_TT\_WTRIG)

TTCAN Watch Trigger Time Register

偏移地址：0xC8

复位值：0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TT_WTRIG[15:0]															

位	标记	位名	功能	读写
			触发时间 (Trigger Time) 用于指定看门触发器的cycle time。	
b15~b0	TT_TWTRIG	触发时间	当TT_WTRIG的最高字节写操作后，TT_WTRIG值开始向CAN时钟域的传递。因此如果 BYTE操作，需先写低字节再写高字节。	R/W

## 35.6 使用注意事项

### 35.6.1 CAN 总线抗干扰措施

CAN 总线广泛应用于汽车、工业控制等行业，如果 CAN 应用现场电磁环境比较恶劣，存在电路不平衡性、空间电磁场、电网进线等因素，会导致 CAN 总线因辐射、传导干扰而产生大量通信噪声，致使总线错误帧增加、重发频繁，正确数据不能及时到达等情况，严重影响数据通信质量。所以实际应用中应该致力于消除噪声干扰，保障 CAN 总线网络稳定工作。

以下是几类常用的 CAN 总线抗干扰措施（包括但不仅限于）

- 增加 CAN 总线接口电气隔离
- 共收发器的信号地
- 使用屏蔽双绞线缆并正确接地
- 提高 CAN 传输线双绞程度
- 增加信号保护器
- 改进网络拓扑
- 应用层软件抗干扰机制

### 35.6.2 CAN 控制器噪声制约

在 CAN 总线网络中应确保通信的位时间满足标准协议的要求，若引入不满足位时间宽度的噪声干扰，可能引起 CAN 控制器异常动作。

## 36 MDIO 串行总线接口（MDIO）

### 36.1 简介

MDIO 接口包含在 IEEE802.3 协议中，是专用于以太 PHY 的管理的串行总线接口，采用 IEEE 802.3 Clause45。MDIO 接口硬件可以接收完整的 MDIO 帧，而无需软件干预。MDIO 接口硬件也可以在没有软件干预的情况下发送完整的 MDIO 帧，只要在接收读数据帧或读后增量地址帧的周转位（TA）之前提供要发送的数据。为了帮助使用和提供相关数据，在每个完整帧的末尾都会生成中断。如果接收到的 PHYADR 或 DEVADD 与预期值不匹配，则不会对帧执行操作。中断也可以在每个有效的 PHYADR 和 DEVADD 之后生成，以允许在帧内进行更复杂的控制。

### 36.2 模块示意图

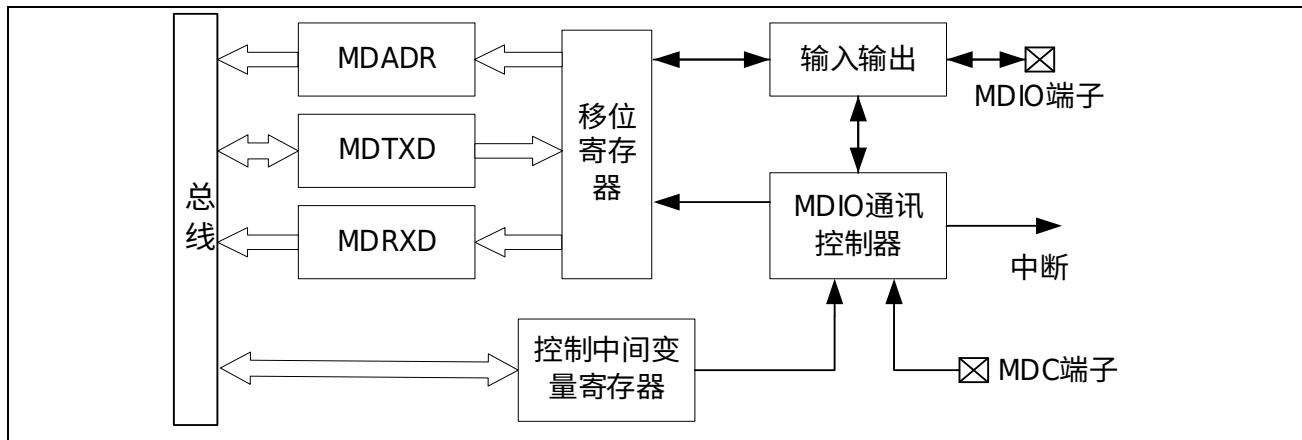


图 36-1 MDIO 模块结构图

### 36.3 功能描述

1. 支持从机模式下最高 4MHz
2. 支持 CFP/CFP2 MSA 管理接口规范
3. 支持多种中断
4. 物理地址可配
  - 软件配置
  - 硬件引脚配置

#### 36.3.1 MDIO 帧结构

MDIO 接口使用 IEEE 802.3 Clause45 中定义的通信数据帧结构。这个帧结构如图 36-2 所示。

每个帧可以是一个地址帧或者一个数据帧。每个帧的总比特长度是 64 位，由 32 位前导和帧命令主体组成。命令主体由六个部分组成，如图 36-2 所示。关于各种帧类型的更多信息见表 36-1。所有值都大端传输。

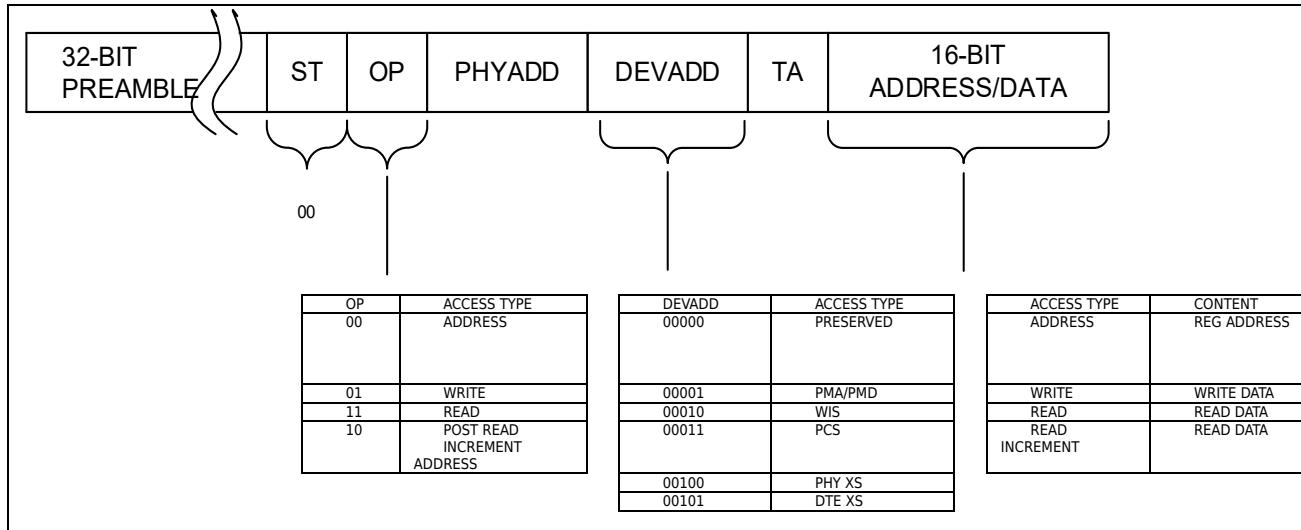


图 36-2 MDIO 帧结构

ST: 起始位 (2bits)

OP: 操作码 (2bits)

PHYADD: 物理端口地址 (5bits)

DEVADD: MDIO 设备地址 (5bits)

TA: 状态转换域 (2bits)

16-BIT ADDRESS/DATA: 地址/数据

对于不同帧类型的帧细节

表 36-1 MDIO 不同类型帧

Frame	idle	Management Frame Fields							Idle
		PRE	ST	OP	PHYADR	DEVADD	TA	Address/Data	
Write Address	Z	1...1	00	00	aaaaaa	aaaaaa	10	aaaaaaaaaaaaaaaaaa	Z
Write Data	Z	1...1	00	01	aaaaaa	aaaaaa	10	ddddddddddddd	Z
Read Data	Z	1...1	00	11	aaaaaa	aaaaaa	z0	ddddddddddddd	Z
Post Read Increment Address	Z	1...1	00	10	aaaaaa	aaaaaa	z0	ddddddddddddd	Z

在闲置情况下，MDC 和 MDIO 不会被驱动为有效。在 TA 的第二位和 16bit 读数据帧以及读后增量地址帧的时候，MDIO 由 MMD 驱动。在其它任何时候，MDIO 由 STA 位驱动。

IDLE (Idle Condition) 闲置

MDIO 的闲置状态为高阻状态。

PRE (Preamble) 前导

在每次处理前，STA (主机) 向 MDIO 发送至少 32 个连续位的序列，一次发送一个位，在 MDC 上具有 32 个响应时钟周期，来建立帧的开始。

ST (Start of Frame) 开始

在前导之后，ST (包含两个 0 位) 指示帧信息的开始。

OP (Operation Code) 操作码

OP 指定要采取的操作，如表 36-2 所示

表 36-2 MDIO 操作码

OP	Descriptions
00	Set the address for a subsequent write or read frame
01	Write to the previously set address
10	Read from the previously set address
11	Read from the previously set address. Then increment the address. Note that user code must increment the address in the MDADR register

PHYADR (Physical Address) 物理地址

这个地址是 5 个位，允许 32 个唯一地址。PHYADR 由 5 个引脚或者软件设置。

DEVADD (Device Address) 设备地址

这个地址是 5 个位，用于选择设备类型。在 CFP 标准中，只有 MDIO 设备地址 1 是支持的。

TA (Turnaround) 状态转换

该时间用于从主机驱动变成由从机驱动。

Address/Data

地址/数据字段位是 16 位。

## 36.4 应用举例

### 36.4.1 应用框图

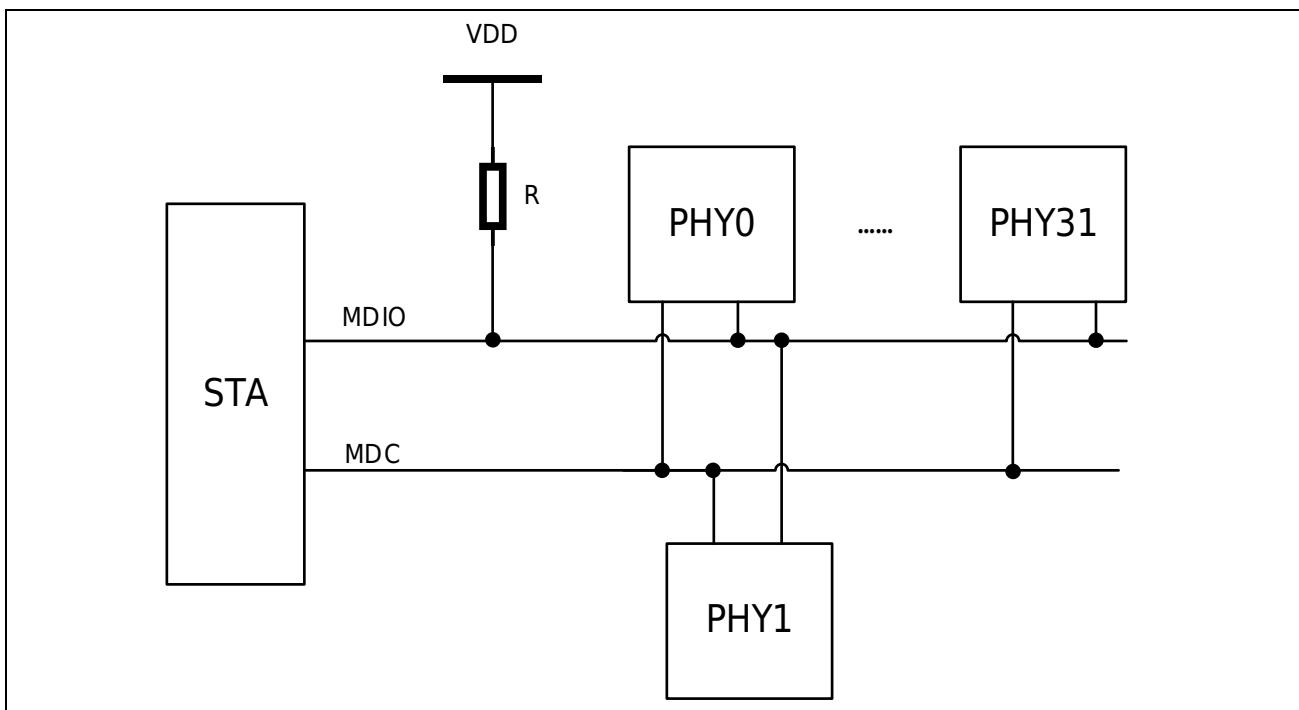


图 36-3 MDIO 应用框图

MDIO 大部分都是硬件完成，只有少部分需要软件配置。

1. 使能端口为 MDIO 的功能。
2. 配置寄存器 MDPHY, MDCON, 和 MDPIN, 得到控制信息。
3. 使能响应的中断寄存器, MDIEN。
4. 写地址帧和写数据帧分别在 MDADR 和 MDRXD 寄存器中接收。
5. 数据必须在读数据帧或读后增量地址帧之前放到 MDTXD, 以便可以自动的插入数据。

MDIO 从机通讯流程:

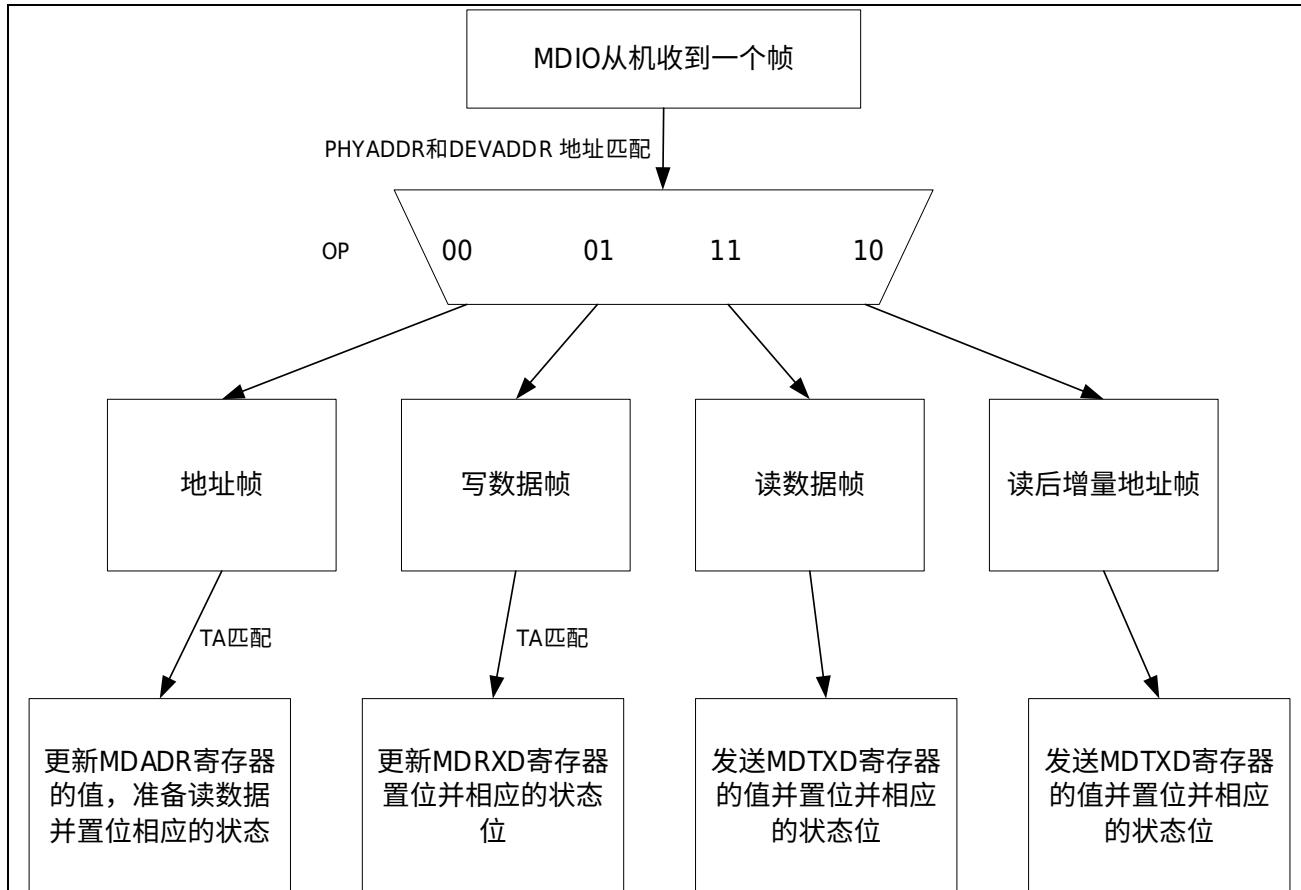


图 36-4 MDIO 从机通讯流程图

在接收到主机发送的写地址帧以后,无论下一帧是写还是读,都可以提前把数据预置到发送数据寄存器里。如果下一帧是读操作的话,MDIO 会将预置的数据发送出去。如果是给了写地址帧,但没有进行读操作,可以在接收完下一次的地址帧之后将新的数据覆盖原来的发送寄存器。

### MDIO 从机接收通信时序：

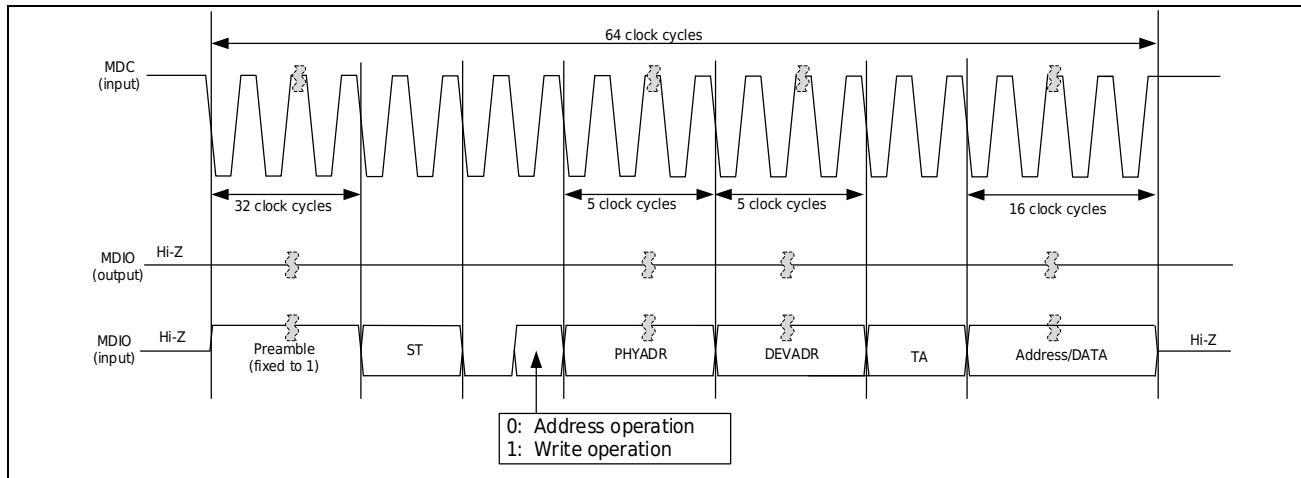


图 36-5 MDIO 从机接收通信时序图

### MDIO 从机发送通信时序：

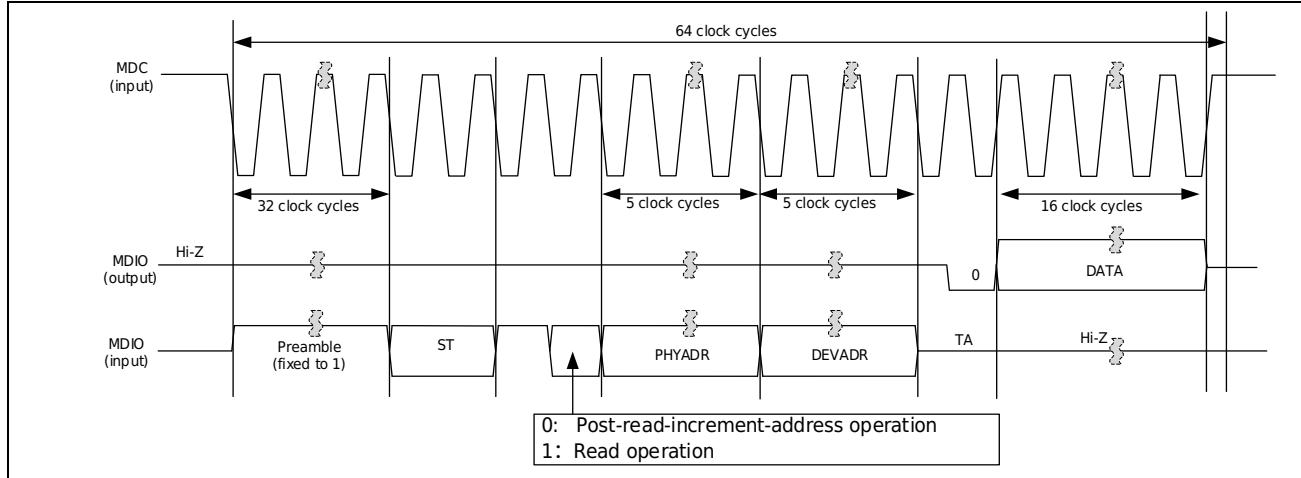


图 36-6 MDIO 从机发送通信时序图

### MDIO 从机连续通信时序：

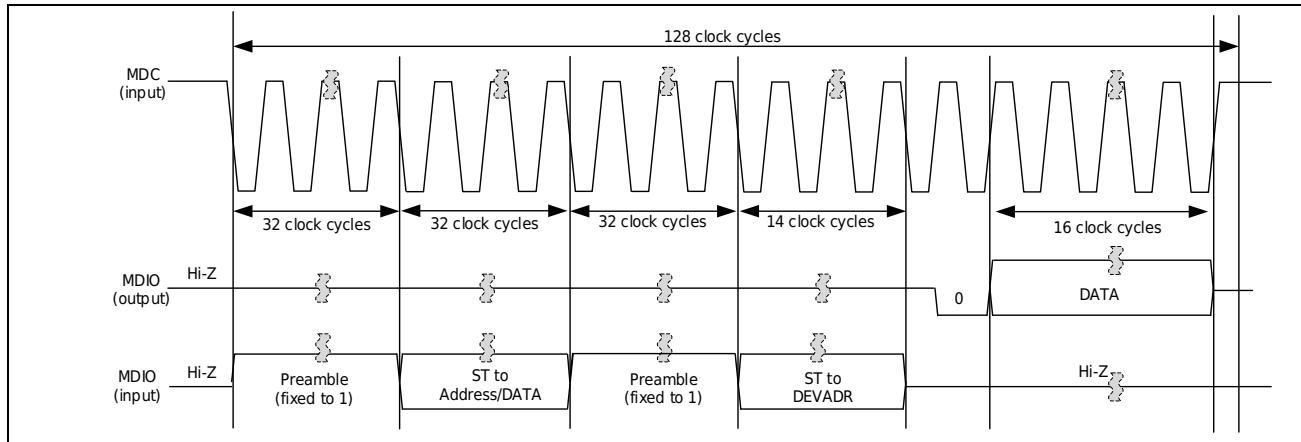


图 36-7 MDIO 从机连续通信时序图

## 36.5 寄存器说明

表 36-3 MDIO 寄存器一览表

MDIO\_BASE\_ADDR:0x40028400

寄存器名	符号	偏移地址	复位值
MDIO模块控制寄存器	MDCON	0x00	0x00000000
MDIO模块控制寄存器1	MDCON1	0x04	0x00000000
MDIO接收帧控制信息寄	MDFRM	0x08	0x00000000
MDIO接收数据寄存器	MDRXD	0x0C	0x00000000
MDIO接收地址寄存器	MDADR	0x10	0x00000000
MDIO传输数据寄存器	MDTXD	0x14	0x00000000
MDIO物理地址寄存器	MDPHY	0x18	0x00000400
MDIO进程信号寄存器	MDSTA	0x1C	0x00000000
MDIO中断许可存器	MDIEN	0x20	0x00000000
MDIO读取物理地址引脚寄存器	MDPIN	0x24	0x00000000
MDIO超时寄存器	MDTO	0x28	0x00000000

### 36.5.1 MDIO 模块控制寄存器(MDCON)

偏移地址：0x00

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved							
b23	b22	b21	b20	b19	b18	b17	b16
Reserved							
b15	b14	b13	b12	b11	b10	b9	b8
Reserved							
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MD_PHYB	MD_RST
0	0	0	0	0	0	0	0

位	标记	位名	功能	读写
b31-b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2	Reserved	-	读出时为“0”，写入时写“0”	R/W
b1	MD_PHYB	物理空间位	0: MDIO物理空间占用5bit 1: MDIO物理空间占用3bit 不用的地址位被忽略	R/W
b0	MD_RST	复位	写“1”将MDIO模块复位 硬件立即再次清掉MD_RST值	R/W

### 36.5.2 MDIO 模块控制寄存器 1(MDCON1)

偏移地址：0x04

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved							
b23	b22	b21	b20	b19	b18	b17	b16
Reserved							
b15	b14	b13	b12	b11	b10	b9	b8
Reserved							
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
Reserved	Reserved	Reserved	Reserved	MD_EN	SEL_EDGE	Reserved	MD_MSDEV
0	0	0	0	0	0	0	0

位	标记	位名	功能	读写
b31-b4	Reserved	-	读出时为“0”，写入时写“0”	R/W
b3	MD_EN	MDIO模块使能位	1: MDIO使能 0: MDIO无效	R/W
b2	SEL_EDGE	超时检测电平选择	1: 选择MDC高电平 0: 选择MDC低电平	R/W
b1	Reserved	-	读出时为“0”，写入时写“0”	R/W
b0	MD_MSDEV	屏蔽dev的比较结果	1: DEVSSDDR比较结果屏蔽 0: DEVADDR比较结果不屏蔽	R/W

### 36.5.3 MDIO 接收帧控制信息寄存器(MDFRM)

偏移地址：0x08

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8
Reserved	Reserved	MD_RTA[13:12]		MD_RDEVAD[11:8]			
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
MD_RDEVAD[7]	MD_RPHYAD[6:2]				MD_ROP[1:0]		
0	0	0	0	0	0	0	0

位	标记	位名	功能	读写
b31-b14	Reserved	-	读出时为“0”，写入时写“0”	R/W
b13-b12	MD_RTA[1:0]	转换位	接收到的帧TA位信息（只支持写地址帧或者写数据帧的TA位）	R
b11-b7	MD_RDEVAD[4:0]	接收设备地址位	接收设备地址	R
b6-b2	MD_RPHYAD[4:0]	接收物理地址位	接收物理地址	R
b1-b0	MD_ROP[1:0]	接收操作码位	接收操作码 00: 地址帧 01: 写入帧 10: 读后地址增量帧 11: 读取帧	R

### 36.5.4 MDIO 接收数据寄存器(MDRXD)

偏移地址：0x0C

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8
MD_RXD[15:8]							
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
MD_RXD[7:0]							
0	0	0	0	0	0	0	0
位	标记	位名	功能	读写			
b31-b16	Reserved		读出时为“0”，写入时写“0”	R/W			
b15-b0	MD_RXD[15:0]	数据接收位	从上一个写入帧 收到的数据	R			

### 36.5.5 MDIO 接收地址寄存器(MDADR)

偏移地址：0x10

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8
MD_ADR[15:8]							
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
MD_ADR[7:0]							
0	0	0	0	0	0	0	0
位	标记	位名	功能	读写			
b31-b16	Reserved		读出时为“0”，写入时写“0”	R/W			
b15-b0	MD_ADR[15:0]	地址接收位	从地址帧 收到的数据	R			

### 36.5.6 MDIO 传输数据寄存器(MDTXD)

偏移地址：0x14

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8
MD_TXD[15:8]							
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
MD_TXD[7:0]							
0	0	0	0	0	0	0	0
位	标记	位名	功能	读写			
b31-b16	Reserved		读出时为“0”，写入时写“0”	R/W			
b15-b0	MD_TXD[15:0]	传输数据位	下个数据帧要 传输的数据	R/W			

### 36.5.7 MDIO 物理地址寄存器(MDPHY)

偏移地址：0x18

复位值：0x00000400

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8
Reserved	MD_DEVADD[14:10]					MD_PHYSEL[9:8]	
0	1	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
MD_PHYSEL[7:5]			MD_PHYSW[4:0]				
0	0	0	0	0	0	0	0

位	标记	位名	功能	读写
b31-b15	Reserved	-	读出时为“0”，写入时写“0”	R/W
b14-b10	MD_DEVADD[4:0]	期待设备地址位	期待的设备地址 通常为01	R/W
b9-b5	MD_PHYSEL[4:0]	选择期待物理地址位	选择期待的物理地址位 0: 设置PHYADR.x=PRTADRx pin 1: 设置PHYADR.x=MD_PHYSW.x	R/W
b4-b0	MD_PHYSW[4:0]	软件提供物理地址位	软件提供物理地址位 根据相应的MD_PHYSEL位选择	R/W

### 36.5.8 MDIO 进程信号寄存器(MDSTA)

偏移地址：0x1C

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved							
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	MD_BUSY						
b15	b14	b13	b12	b11	b10	b9	b8
Reserved	Reserved	Reserved	MD_RBNE	MD_OVR	MD_UDR	MD_TO	MD_TANM
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
MD_PHN	MD_PHYM	MD_DEVN	MD_DEVM	MD_RDF	MD_INCF	MD_ADRF	MD_WRF
0	0	0	0	0	0	0	0

位	标记	位名	功能	读写
b31-b17	Reserved	-	读出时为“0”，写入时写“0”	R/W
b16	MD_BUSY	通信忙中	表示正在通信中。硬件自动清除	R
b13-b15	Reserved	-	读出时为“0”，写入时写“0”	R/W
b12	MD_RBNE	数据接收缓冲区非空标志位	已接收到数据并可以读取。读MDSTA清零	RC(Read Clear)
b11	MD_OVR	数据接收上溢标志位	在MD_RBNE置位的情况下，在接收到TA位后，还没有读MDRXD寄存器，则置1，读MDRXD清零。如果在接收移位寄存器的数据传递给MDRXD之前读MDRXD(即第63个MDC之前)，则当前帧的数据不会丢失。	RC
b10	MD_UDR	数据发送下溢标志位	在数据发送缓冲区为空的情况下，如果在读数据帧/读后地址增量帧的TA位的第二个bit之前没有将数据写到MDTXD寄存器，将会硬件置1。写MDTXD清零。	RC
b9	MD_TO	超时标志	在一个完整的帧的两个位之间的时间，包括前导数据部分，如果已达到设置的超时时间，仍没有收到新的位或者主备的数据位没有发送出去，将会由硬件置1。读MDSTA清零	RC
b8	MD_TANM	TA位不匹配	在写地址或写数据帧TA最后一帧接收完成时，如果接收到的TA与期望的“10”不匹配，则置1，读取MDSTA后清零	RC
b7	MD_PHN	物理地址不匹配	如果物理地址不匹配，在最后置1 读取MDSTA后清掉值	RC
b6	MD_PHYM	物理地址匹配位	如果物理地址匹配，在最后置1 读取MDSTA后清掉值	RC
b5	MD_DEVN	设备地址不匹配	如果设备地址不匹配，在最后置1 读取MDSTA后清掉值	RC
b4	MD_DEVM	设备地址匹配位	如果设备地址匹配，在最后置1	RC

读取MDSTA后清掉值			
b3	MD_RDF	物理、设备地址都匹配读取标志位	如果设备地址和物理地址都匹配,在读取帧的最后置1。 RC 读取MDSTA后清掉值
b2	MD_INCF	物理、设备地址都匹配后地址增加标志位	如果设备地址和物理地址都匹配,在读自增的地址帧的最后置1。 RC 读取MDSTA后清掉值
b1	MD_ADRF	物理、设备地址都匹配后地址标志位	如果设备地址和物理地址都匹配,在地址帧的最后置1。 RC 读取MDSTA后清掉值
b0	MD_WRF	物理、设备地址都匹配后写入标志位	如果设备地址和物理地址都匹配,在写入帧的最后置1。 RC 读取MDSTA后清掉值

### 36.5.9 MDIO 中断许可寄存器(MDIEN)

偏移地址：0x20

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8
Reserved	Reserved	Reserved	MD_RBNEIE	MD_OVRIE	MD_UDRIE	MD_TOIE	MD_TANMIE
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
MD_PHYNIE	MD_PHYMIE	MD_DEVNIE	MD_DEVMIE	MD_RDFIE	MD_INCFIE	MD_ADRFIE	MD_WRFIE
0	0	0	0	0	0	0	0

位	标记	位名	功能	读写
b31-b13	Reserved	-	读出时为“0”，写入时写“0”	R/W
b12	MD_RBNEIE	MD_RBNE有效中断请求位	如果置位，当MD_RBNE有效，发出中断请求	R/W
b11	MD_OVRIE	MD_OVR有效中断请求位	如果置位，当MD_OVR有效，发出中断请求	R/W
b10	MD_UDRIE	MD_UDR有效中断请求位	如果置位，当MD_UDR有效，发出中断请求	R/W
b9	MD_TOIE	MD_TO有效中断请求位	如果置位，当MD_TO有效，发出中断请求	R/W
b8	MD_TANMIE	MD_TANM有效中断请求位	如果置位，当MD_TANM有效，发出中断请求	R/W
b7	MD_PHYNIE	MD_PHN有效中断请求位	如果置位，当MD_PHN有效，发出中断请求	R/W
b6	MD_PHYMIE	MD_PHYM有效中断请求位	如果置位，当MD_PHYM有效，发出中断请求	R/W
b5	MD_DEVNIE	MD_DEVN有效中断请求位	如果置位，当MD_DEVN有效，发出中断请求	R/W
b4	MD_DEVMIE	MD_DEVM有效中断请求位	如果置位，当MD_DEVM有效，发出中断请求	R/W
b3	MD_RDFIE	MD_RDF有效中断请求位	如果置位，当MD_RDF有效，发出中断请求	R/W
b2	MD_INCFIE	MD_INCF有效中断请求位	如果置位，当MD_INCF有效，发出中断请求	R/W
b1	MD_ADRFIE	MD_ADRF有效中断请求位	如果置位，当MD_ADRF有效，发出中断请求	R/W
b0	MD_WRFIE	MD_WRF有效中断请求位	如果置位，当MD_WRF有效，发出中断请求	R/W

### 36.5.10 MDIO 读取物理地址引脚寄存器(MDPIN)

偏移地址：0x24

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
Reserved	Reserved	Reserved	MD_PIN[4:0]				
0	0	0	0	0	0	0	0

位	标记	位名	功能	读写
b31-b5	Reserved	-	读出时为“0”，写入时写“0”	R/W
b4-b0	MD_PIN[4:0]	MD引脚	读取PRTADR引脚的数值	R

### 36.5.11 超时寄存器 (MDTO)

偏移地址: 0x28

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TOEN
b15	b14	b13	b12	b11	b10	b9	b8
TOCNT[15:8]							
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
TOCNT[7:0]							
0	0	0	0	0	0	0	0
位	标记	位名	功能	读写			
b31-b17	Reserved	-	读出时为“0”，写入时写“0”	R/W			
b16	TOEN	超时使能	0: 检测MDC电平超时功能禁止 1: 检测MDC电平超时功能允许	R/W			
b15-b0	TOCNT[15:0]	计数值	超时的计数值 MDIO超时=TOCNT*PCLK	R/W			

## 36.6 使用注意事项

1. 系统时钟是 MDC 的四倍以上。
2. 上溢发生在 MDRXD 有数据没有读走，接收新的数据时。
3. 接收地址帧不受接收数据寄存器是否满的影响。
4. 如果不读 MDSTA 寄存器清掉地址匹配标志位，存在地址匹配和不匹配标志位同时为 1。

## 37 外部存储器控制器 (EXMC)

### 37.1 概要

外部存储器控制器 EXMC (External Memory Controller) 是一个用来访问各种片外存储器，实现数据交换的独立模块。EXMC 通过配置可以把内部的 AMBA 协议接口转换为各种类型的专用片外存储器通信协议接口，包括 SRAM、PSRAM、NOR Flash 等。

### 37.2 基本特性

#### 37.2.1 功能列表

EXMC 基本的功能列表如下所示。

表 37-1 EXMC 的基本功能

支持的外部 存储器类型	• SRAM
	• PSRAM
	• NOR Flash
接口协议	• 支持 AHB 协议与各种外部存储器的接口转换
接口时序	• 各种类型存储器的接口时序可编程
总线位宽	• 16 位、32 位 MEM 总线宽度
总线复用	• SRAM, NOR Flash 和 PSRAM 支持地址线和数据线复用
自动分割	• AHB 总线位宽和外部存储器位宽不匹配时，支持自动分割及字节选择控制

### 37.2.2 控制器架构

图 37-1 为基本的 EXMC 架构框图。SMC 外部存储器控制器独立生成对应协议的接口送至端口。

注：架构图中将 SRAM/PSRAM/NOR Flash 控制器定义为 SMC (Static Memory Controller)。

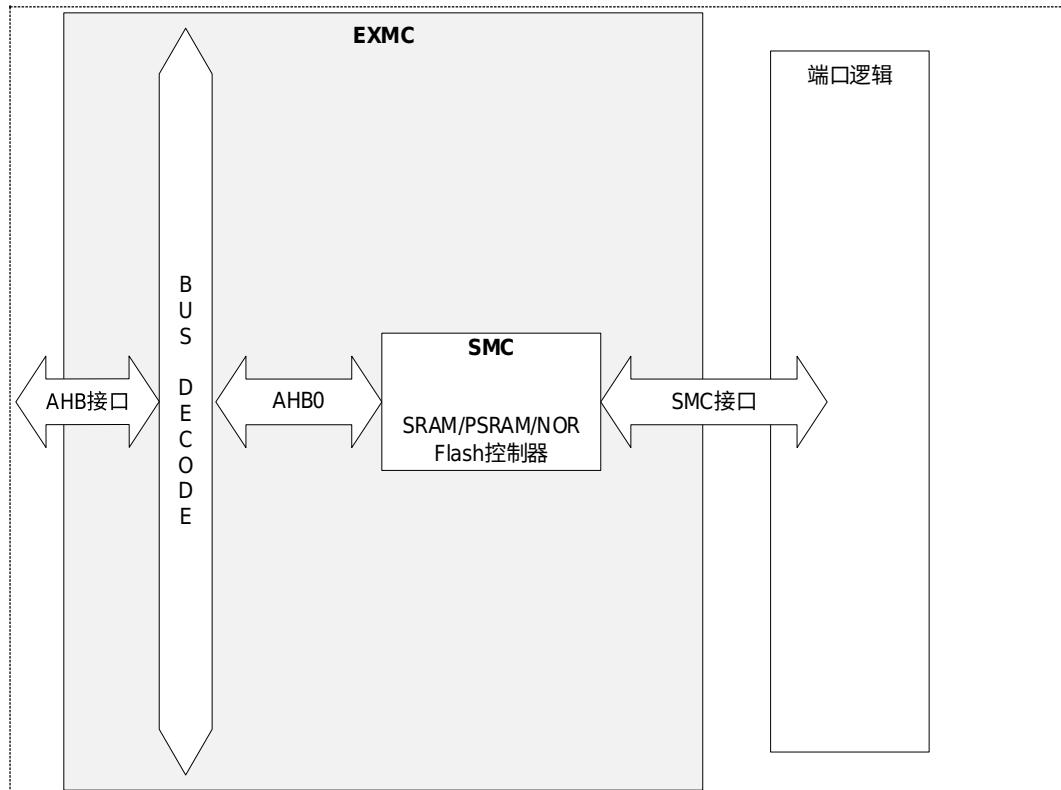


图 37-1 EXMC 架构图

### 37.2.3 基本访问规范

EXMC 是 AHB 总线协议和外部存储器协议的转换接口。在 SMC 的数据传输过程中，AHB 数据宽度和存储器数据宽度可能不相同。为了保证数据传输的一致性，读写访问需要遵从以下规范：

- AHB 访问宽度等于存储器宽度，数据正常访问
- AHB 访问宽度大于存储器宽度，自动将 AHB 访问分割成几个连续的存储器数据宽度的传输
- AHB 访问宽度小于存储器宽度，通过字节选择控制信号 SMC\_BLS[3:0]来访问对应的字节

AHB 访问宽度和各种数据位宽的存储器之间的具体的访问方式参见下表 37-2。

表 37-2 AHB 访问宽度和存储器位宽对应访问方式表

R/W	AHB 访问宽度	外部存储器位宽	注释
R	8	16	
	16		
	32		转换成 2 次 EXMC 读操作
	8	32	
	16		
	32		
W	8	16	使用字节控制信号 BLS[0]
	16		
	32		转换成 2 次 EXMC 写操作
	8	32	使用字节控制信号 BLS[0]
	16		使用字节控制信号 BLS[1:0]
	32		

注：上表仅适用于 SMC 的数据访问

### 37.2.4 地址映射

本产品中定义了总大小为 1GB 的外部存储器访问区间，用于不同类型的外部存储器和内部数据交换，包括 EXMC 和 QSPI 等。该 1GB 的空间按图 37-2 的方案分配，SRAM/PSRAM/NOR Flash (SMC) 的数据访问映射 512MB 的地址空间、QSPI 的控制和数据访问映射 128MB 的地址空间。

**注：**地址空间为 0x98000000~0x9FFFFFFF 的访问控制由 QSPI 模块实现，具体请参考【四线式串行外设接口（QSPI）】章节。

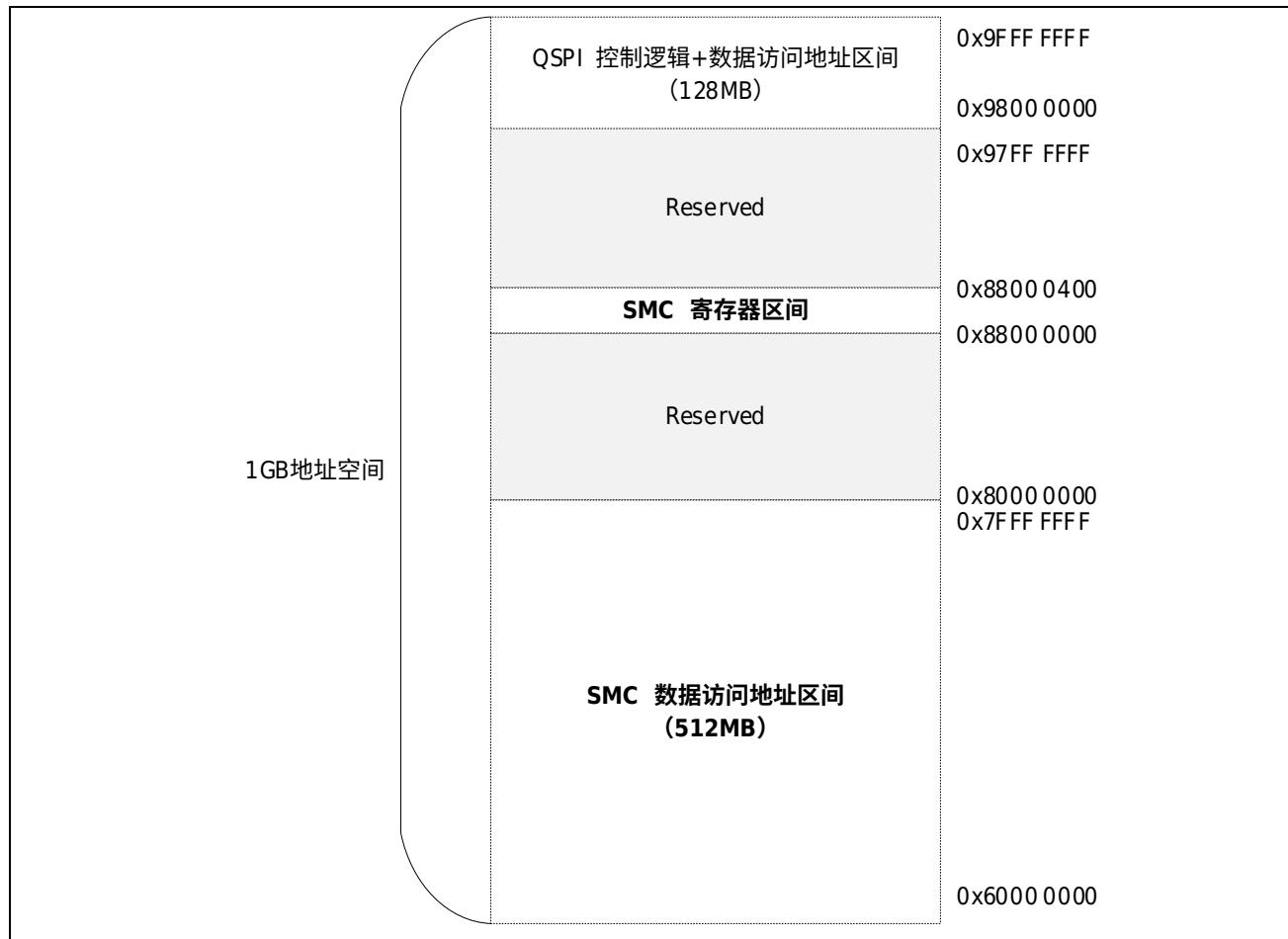


图 37-2 外部空间地址分配

SMC 的地址空间可划分为 4 个大小可编程的 Chip，通过片选控制寄存器（SMC\_CSCR0~1）的设定可以划分各个 Chip 的空间大小。各个 Chip 对应的外部存储器对象可以独立地配置，并对应有不同的 CS（Chip Select）信号输出至外部端口。图 37-3 是 SMC 的 Chip Select 划分，当 CPU 访问的地址空间在某一个 SMC 的 CS 空间时，其对应的 SMC\_CS 信号输出就变为有效电平。

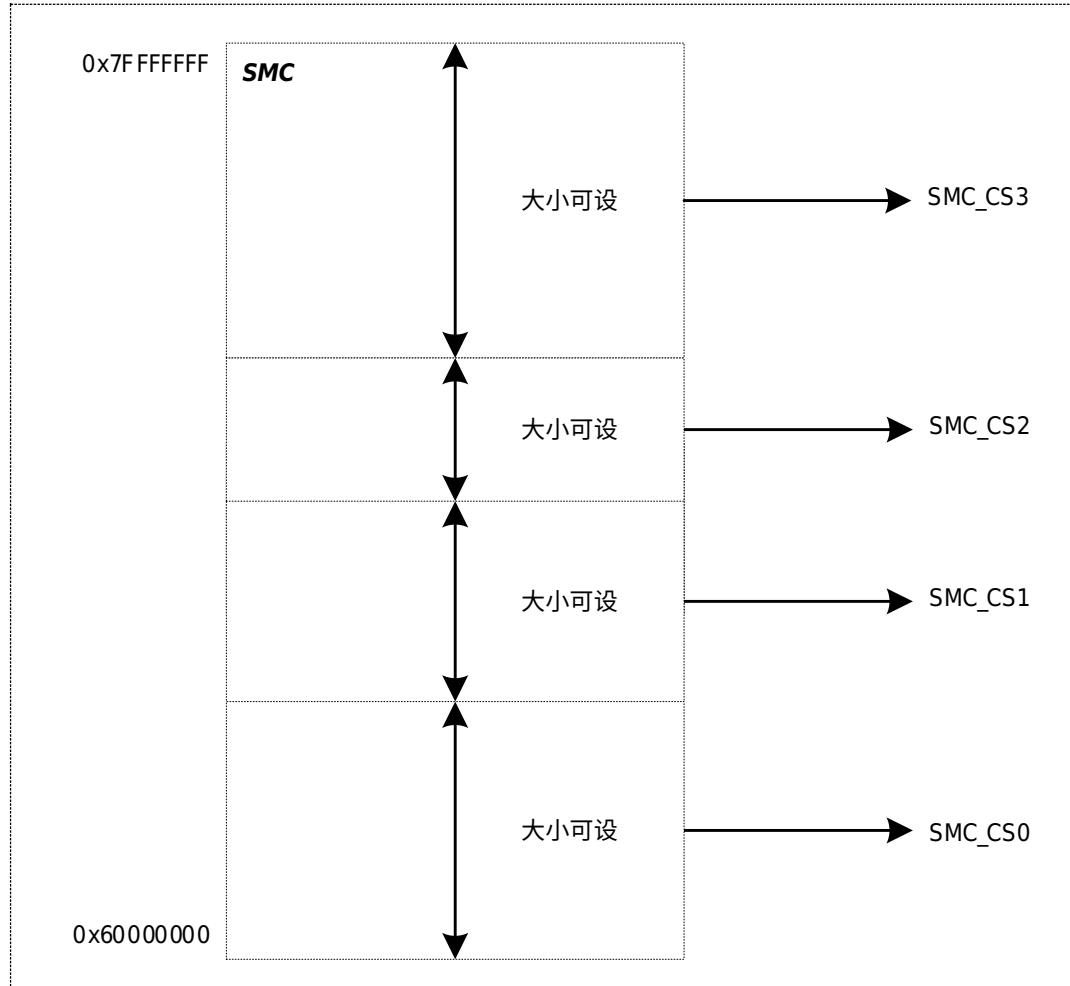


图 37-3 SMC 的地址空间划分

### 37.2.5 协议接口

SMC 对应的存储器的协议接口信号。EXMC 通过 SMC 实现的接口协议和外部存储器进行数据交换。以下为 SMC 存储器所需要的协议接口说明。

#### 37.2.5.1 SMC 协议接口

下表为 SRAM/PSRAM/NOR Flash 控制（SMC）访问所需的协议接口。

表 37-3 SMC 协议接口

协议接口名	方向	有效电平	功能描述
SMC_CLK	O	-	SMC 的时钟输出
SMC_ADD[29:0]	O	-	SMC 的地址输出
SMC_DATA[31:0]	IO	-	SMC 的数据
SMC_WE	O	L	SMC 的写使能
SMC_OE	O	L	SMC 的输出使能
SMC_CS[3:0]	O	L	SMC 的片选信号
SMC_BLS[3:0]	O	L	SMC 的字节选通信号
SMC_ADV	O	L	SMC 的地址锁存信号
SMC_CRE	O	H	SMC 的配置寄存器模式信号
SMC_WAIT	I	L	SMC 的输入等待信号
SMC_BAA	O	L	SMC 的地址提示信号

基于上述 SMC 外部存储器的接口信号表，将芯片的 EXMC 端口功能配置如下表。

表 37-4 EXMC 端口功能分配

引脚名	方向	有效电平	功能描述
			SMC
EXMC_CLK	O	H/L	SMC_CLK
EXMC_ADD29~0	O	H/L	SMC_ADD[29:0]
EXMC_DATA31~0	IO	H/L	SMC_DATA[31:0]
EXMC_WE	O	L	SMC_WE
EXMC_CE7~0	O	L	SMC_BLS[3:0]
	O	L	SMC_CS[3:0]
EXMC_OE	O	L	SMC_OE
EXMC_BAA	O	L	SMC_BAA
EXMC_ADV	O	L	SMC_ADV
EXMC_ALE	O	H	SMC_CRE
EXMC_CLE	O	H	-
EXMC_RB7~0	I	L	-
	I	L	SMC_WAIT

## 37.3 功能说明

### 37.3.1 SMC-SRAM/PSRAM/NOR Flash 控制器

#### 37.3.1.1 SRAM/PSRAM/NOR Flash 介绍

SRAM/NOR Flash 都是具有静态存取功能的存储器，不需要刷新电路即能保存它内部存储的数据。其地址线的宽度决定了存储单元的数量，每一个存储单元上对应存储器存储宽度大小的空间用于存储数据，因此 SRAM/PSRAM/NOR Flash 存储器的容量大小是存储器数据宽度  $\times 2^{\text{地址线宽度}}$ 。

PSRAM 是假静态随机存储器，由于它具有 SRAM 的接口，因此能像 SRAM 一样，与系统内部通过 EXMC 的 SMC 模块实现数据交换。

#### 37.3.1.2 SMC 基本功能

SMC 基本功能特性如下：

- 支持 16 位、32 位外部存储器数据位宽
- AHB 字、半字、字节访问
- 为每个存储器 Chip 提供独立的片选控制
- 字节选择信号输出
- 地址线、数据线复用
- 可编程的协议时序参数
- 可编程速率的自动刷新动作（PSRAM 时使用）
- 具有 2 个 47 位的命令 FIFO
- 具有 4 个 36 位的写数据 FIFO
- 具有 4 个 32 位的读数据 FIFO
- 低功耗管理

### 37.3.1.3 SMC 初始设定

SMC 在上电复位之后的状态图及状态之间的切换如图 37-4 所示。

初始复位时 SMC 为 Low Power 状态；通过状态控制寄存器（SMC\_STCR0 和 SMC\_STCR1）的设定，SMC 可以在 Ready 状态和 Low Power 状态之间进行切换，具体请参考【SMC 低功耗管理】章节。

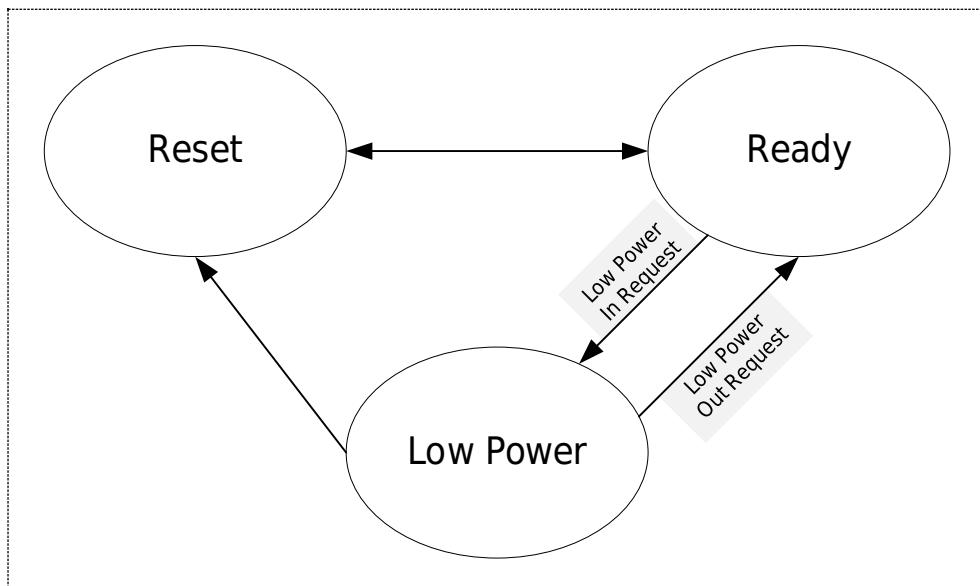


图 37-4 SMC 状态图

SMC 在和外部 SRAM、PSRAM、NOR Flash 等进行通信之前，必须进行初始设定，配置好相关的参数，以保证能进行正确的数据传输。具体的设定顺序参考下图。

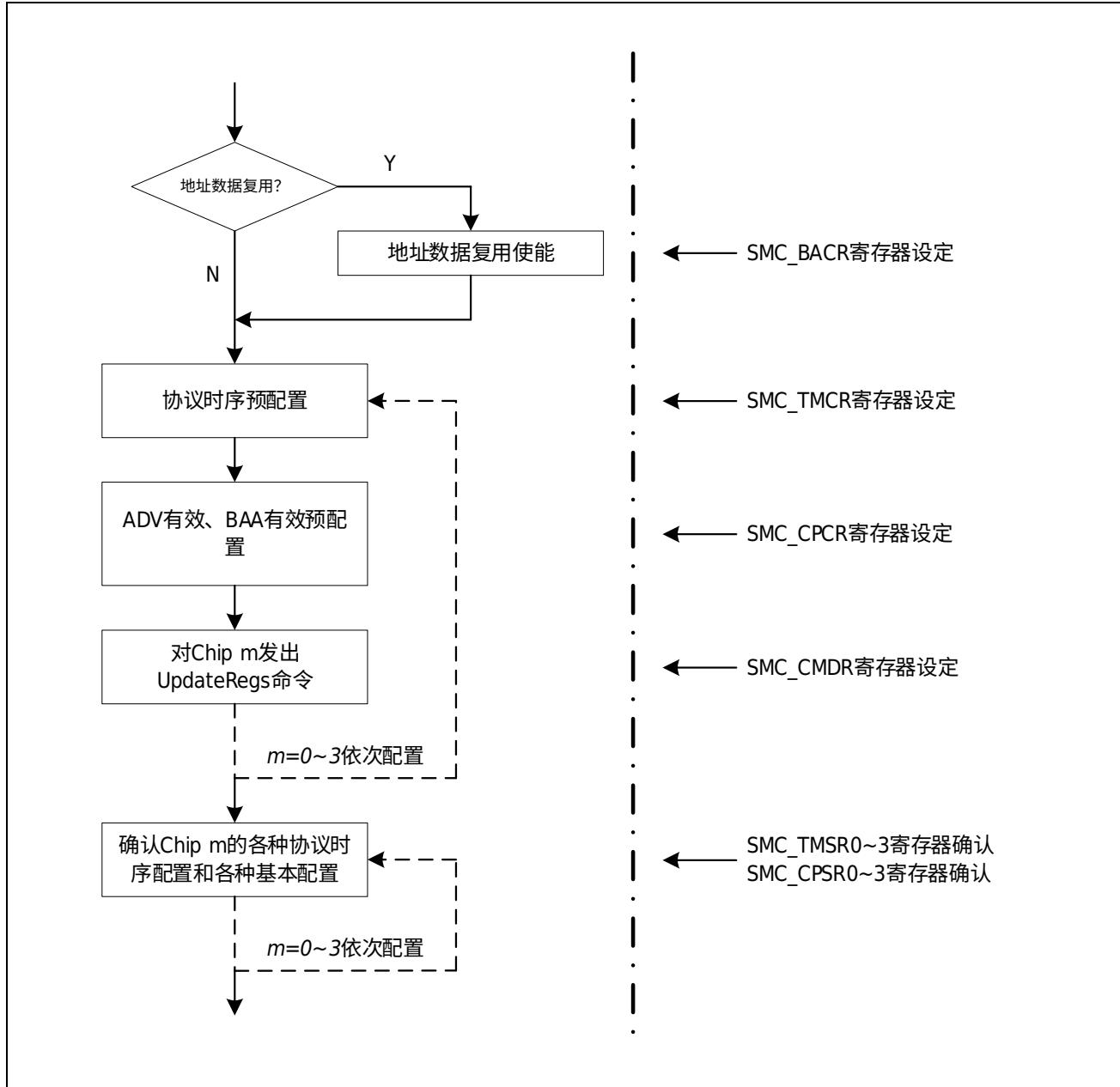


图 37-5 SMC 初始设定流程

### 37.3.1.4 SMC 访问动作

#### 访问方式

SMC 对外部存储器的读写，可以选择同步访问方式或者异步访问方式。

同步访问方式是对应同步访问类型的外部存储器，SMC 在读写控制信号等输出时，时钟信号也同时输出，外部存储器根据 SMC 输出的时钟与 SMC 进行数据传输，即 SMC\_CLK 有效；异步访问方式是对应异步访问类型的外部存储器，SMC 在读写控制信号等输出时，时钟信号不输出（输出始终为低），外部存储器依据其它控制信号实现数据访问。

在同步访问方式时，SMC 和外部存储器间通过 SMC\_WAIT 信号来控制读写动作。在 SMC\_WAIT 信号输入低电平时，SMC 处于等待状态，不执行任何操作，等到 SMC\_WAIT 变为高电平后 SMC 才正常的进行读写访问。

通过设定 CHIP 配置寄存器 (SMC\_CPCR) 的 RSYN、WSYN 位，可选择同步读写方式或异步读写方式。

#### 访问时序

SMC 支持多种类型的存储器的访问，可以把 AHB 的单次读写操作转换成存储器的读写操作。下面列出几种控制器与外部存储器进行数据读写的时序例：

- a. 单次读动作
- b. 地址数据线复用的单次读动作
- c. 单次写动作
- d. 地址数据线复用的单次写动作

表 37-5 和图 37-6~图 37-9 是单次读动作的基本时序图和设定例。

表 37-5 单次读动作基本设定例

基本设定	MW	RSYN	WSYN	BAA	ADV	BLS	- 表示不关注 <set> 表示用户设定值
	<set>	<set>	-	-	0b0	0b0	
时序设定	t_rc	t_wc	t_ceoe	t_wp	t_pc	t_tr	<set> 表示用户设定值
	0b0011	-	0b001	-	-	-	

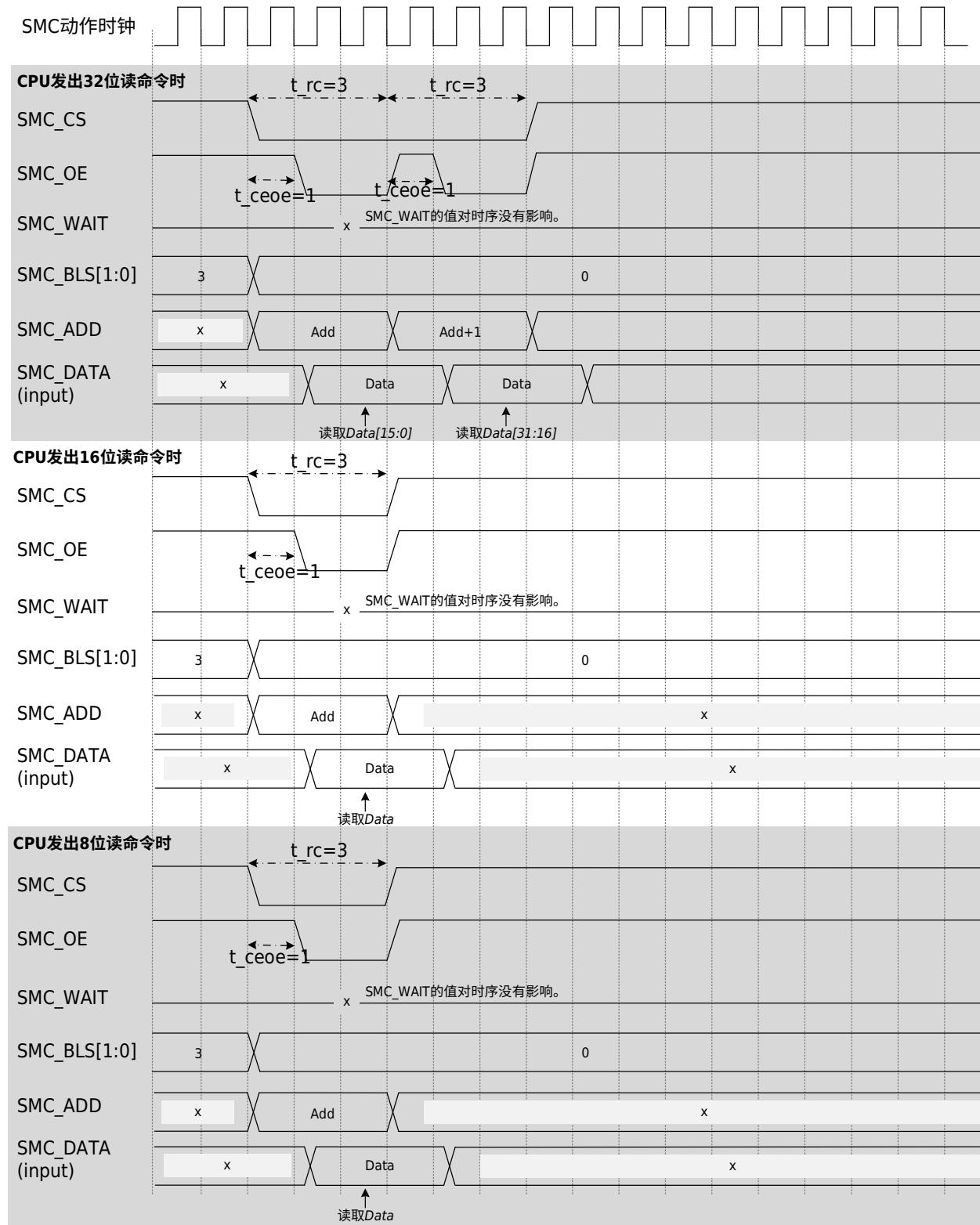


图 37-6 单次读动作基本时序 (异步方式 (RSYN=0) & 16 位位宽 (MW=01))

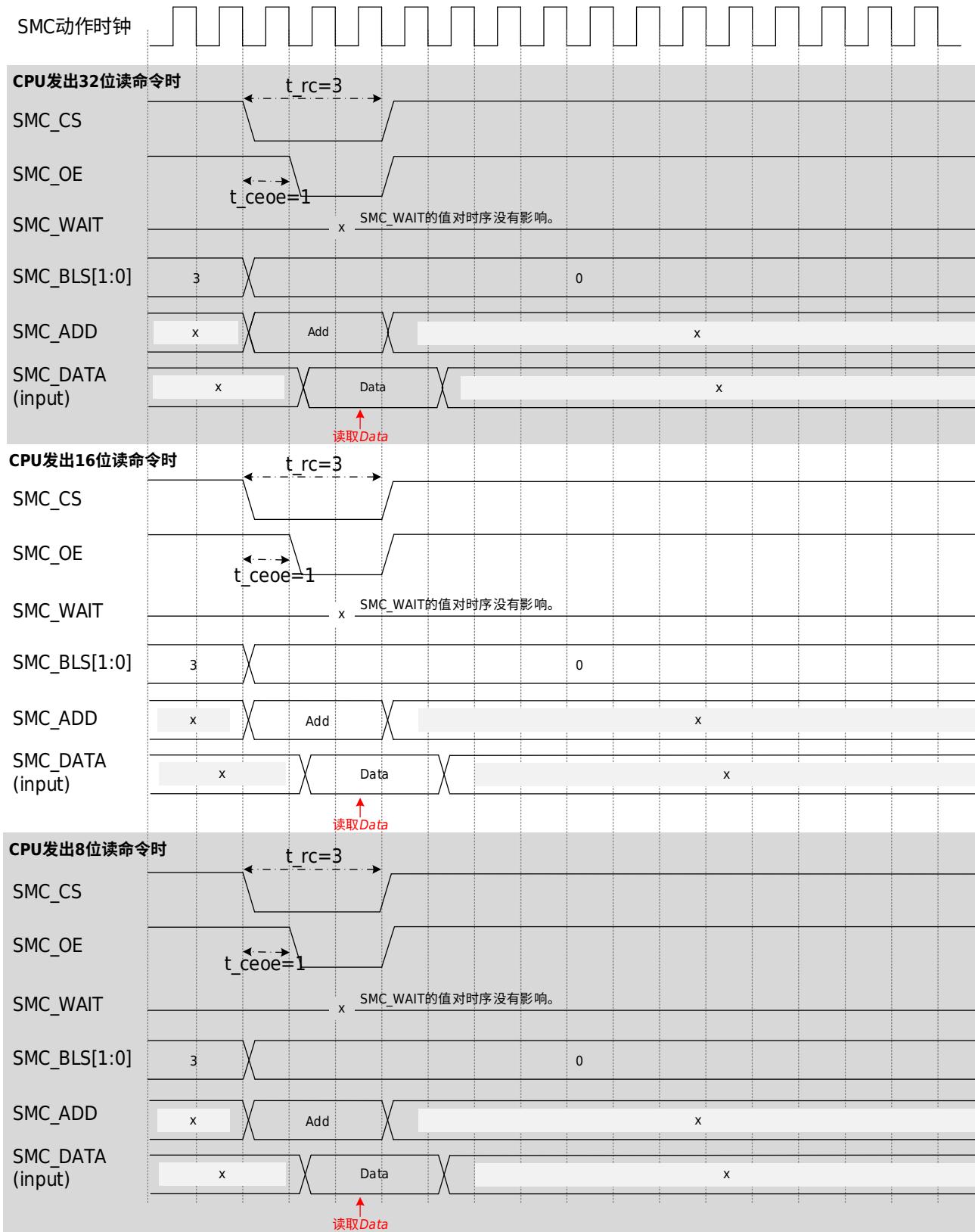


图 37-7 单次读动作基本时序 (异步方式 (RSYN=0) & 32 位位宽 (MW=10))

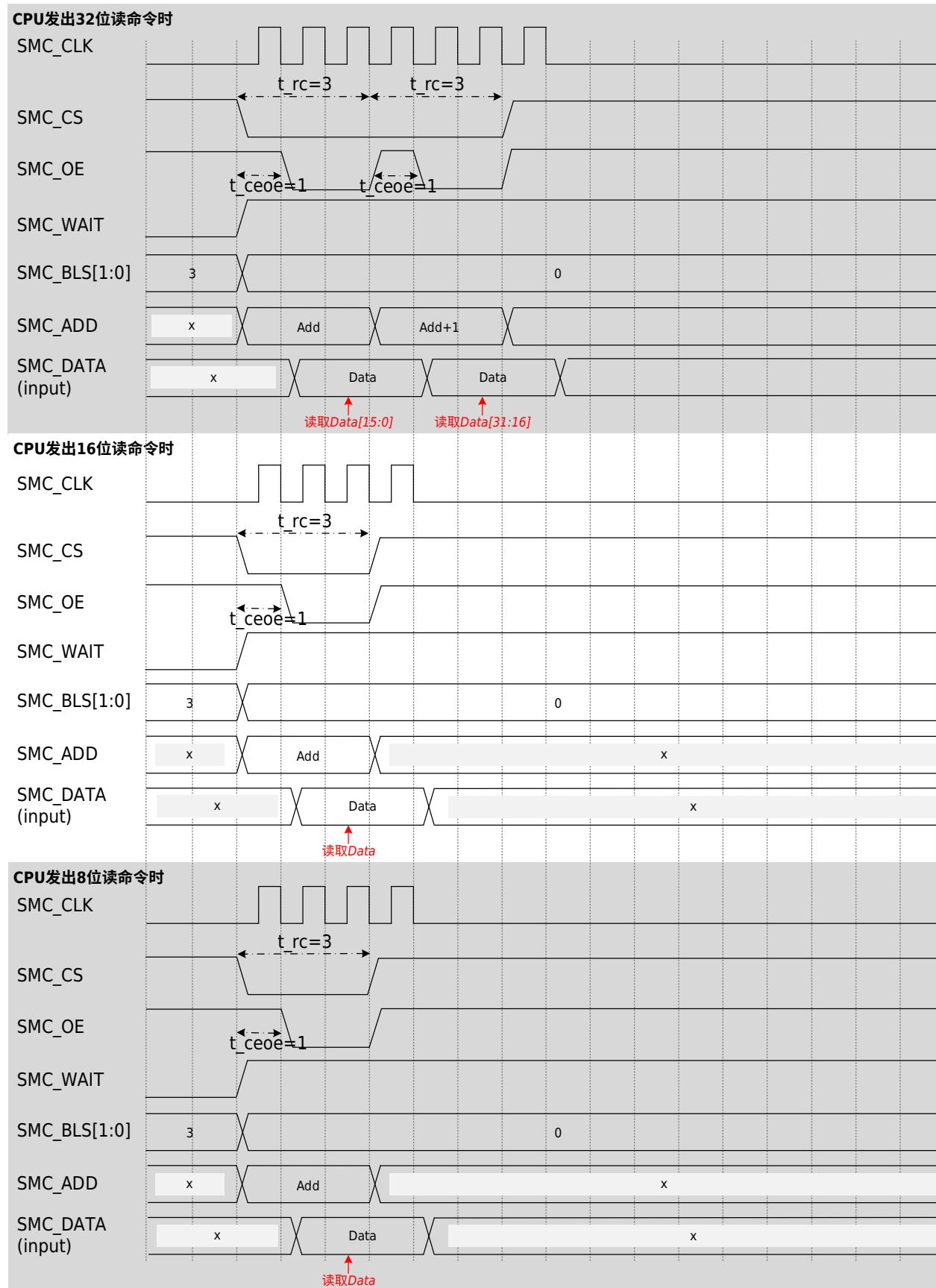


图 37-8 单次读动作基本时序 (同步方式 (RSYN=1) & 16 位位宽 (MW=01))

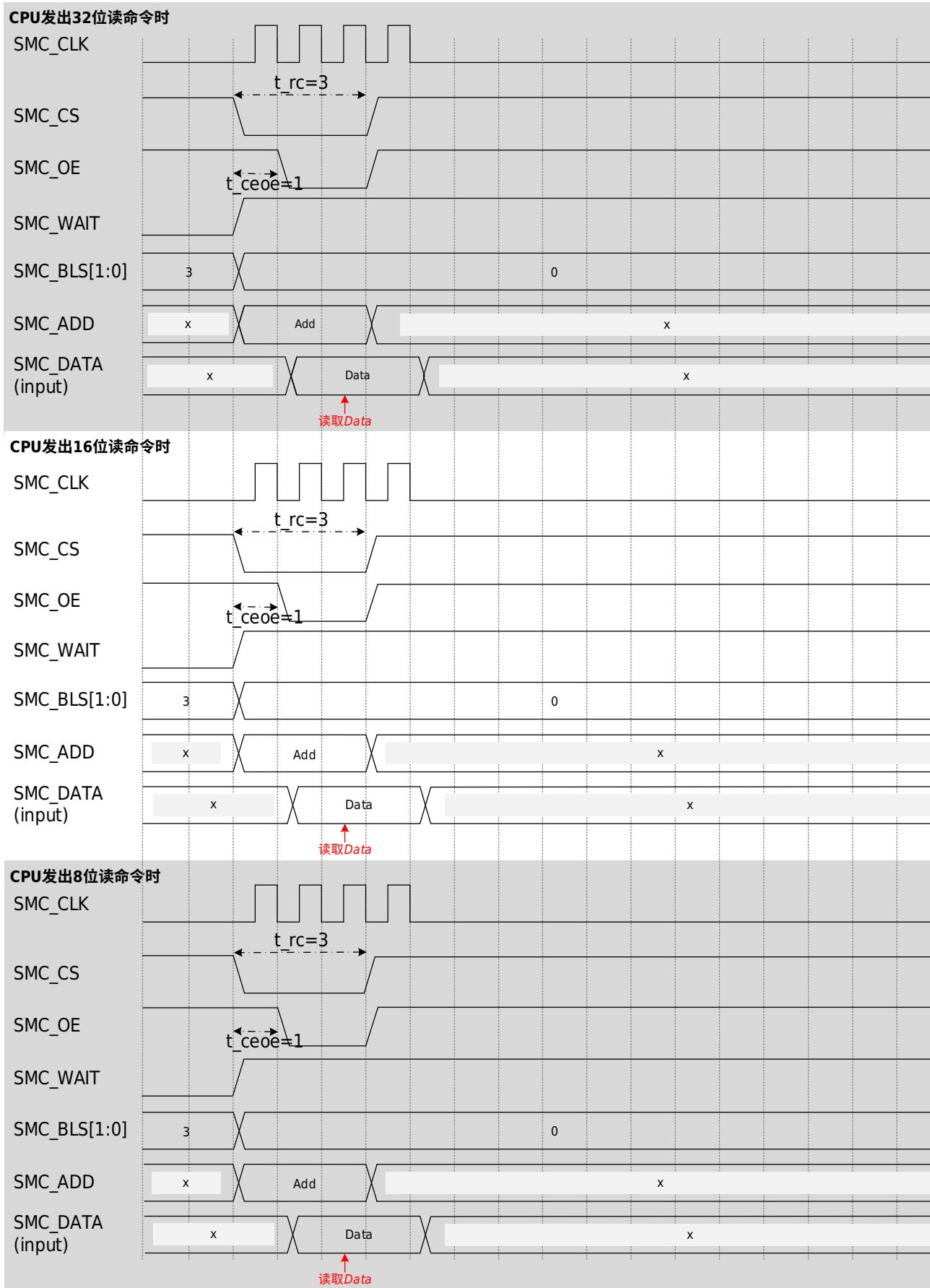


图 37-9 单次读动作基本时序 (同步方式 (RSYN=1) &32 位位宽 (MW=10))

表 37-6 和图 37-10~是地址数据线复用的单次读动作的基本时序图和设定例。

表 37-6 地址数据线复用单次读动作基本设定例

基本设定	MW	RSTN	WSYN	BAA	ADV	BLS	- 表示不关注 <set> 表示用户设定值
	<set>	<set>	-	-	0b1	0b0	
时序设定	t_rc	t_wc	t_ceoe	t_wp	t_pc	t_tr	
	0b0110	-	0b011	-	-	0b010	

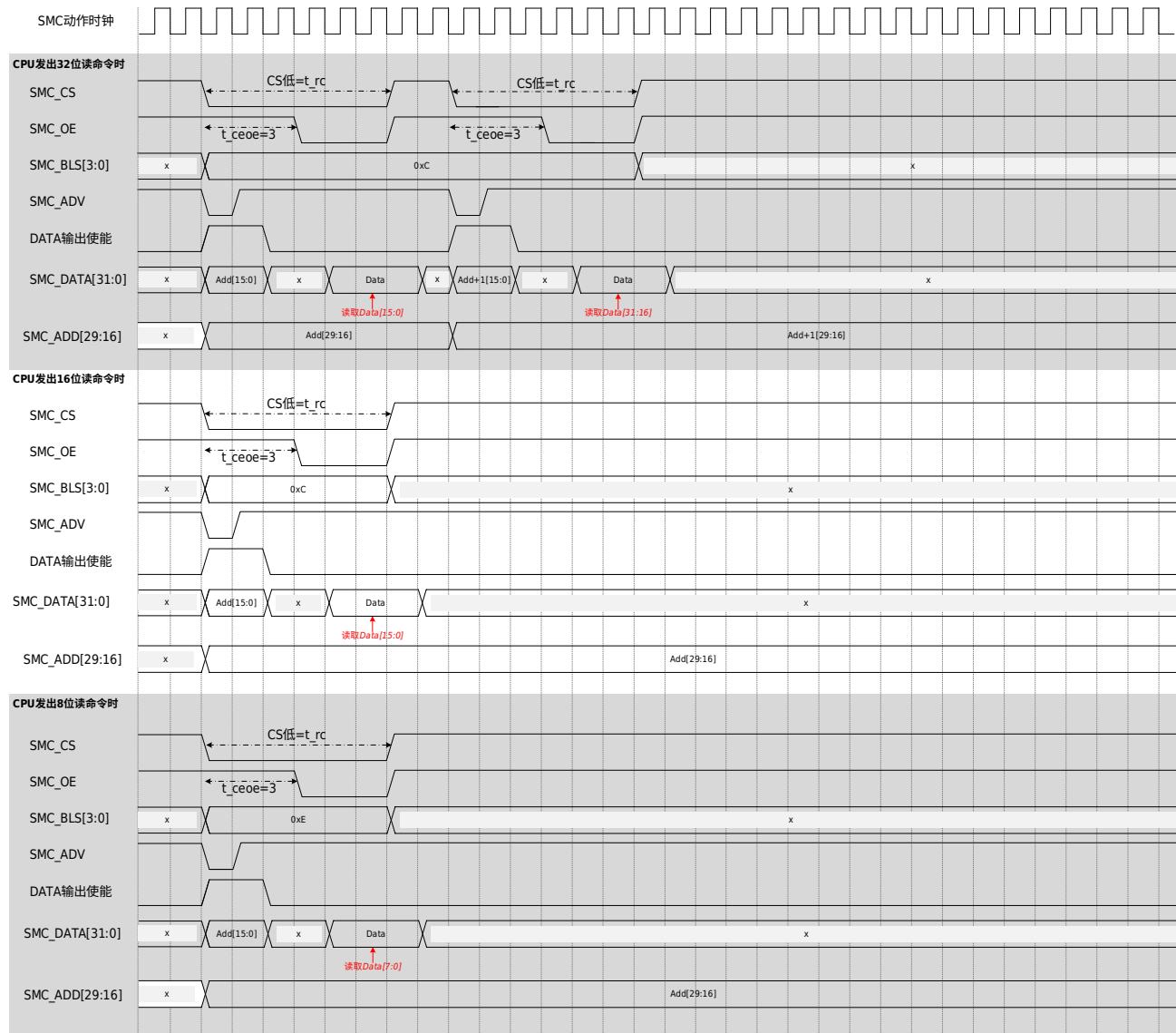


图 37-10 地址数据线复用单次读动作基本时序 (异步方式 (RSTN=0) & 32 位位宽 (MW=10))

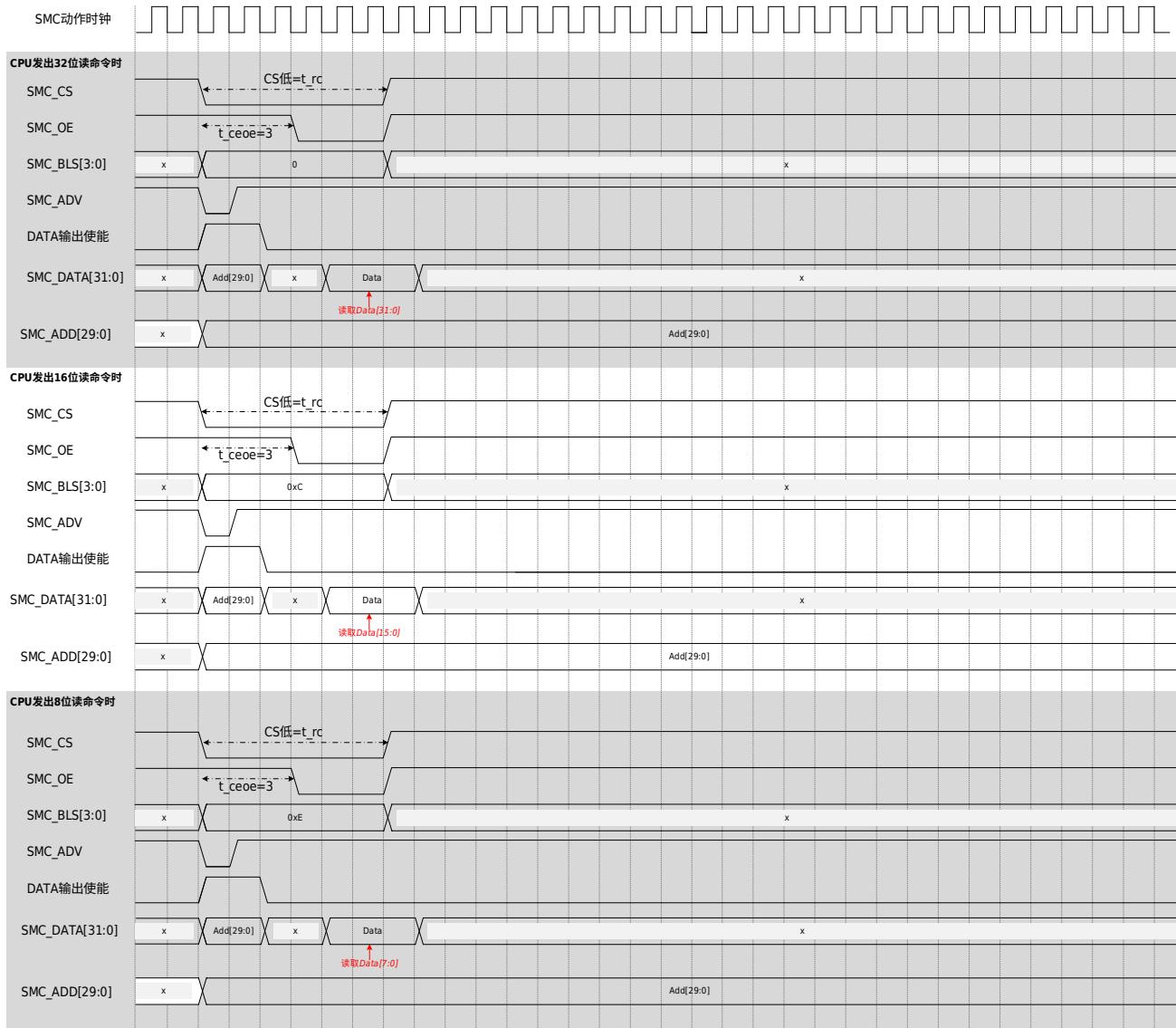


图 37-11 地址数据线复用单次读动作基本时序 (异步方式 (RSYN=0) & 32 位位宽 (MW=10))

表 37-7 和图 37-12~图 37-15 是单次写动作的基本时序图和设定例。

表 37-7 单次写动作基本设定例

基本设定	MW	RSYN	WSYN	BAA	ADV	BLSS	- 表示不关注 <set> 表示用户设定值
	<set>	-	<set>	-	0b0	<set>	
时序设定	t_rc	t_wc	t_ceoe	t_wp	t_pc	t_tr	
	-	0b0100	-	0b010	-	-	

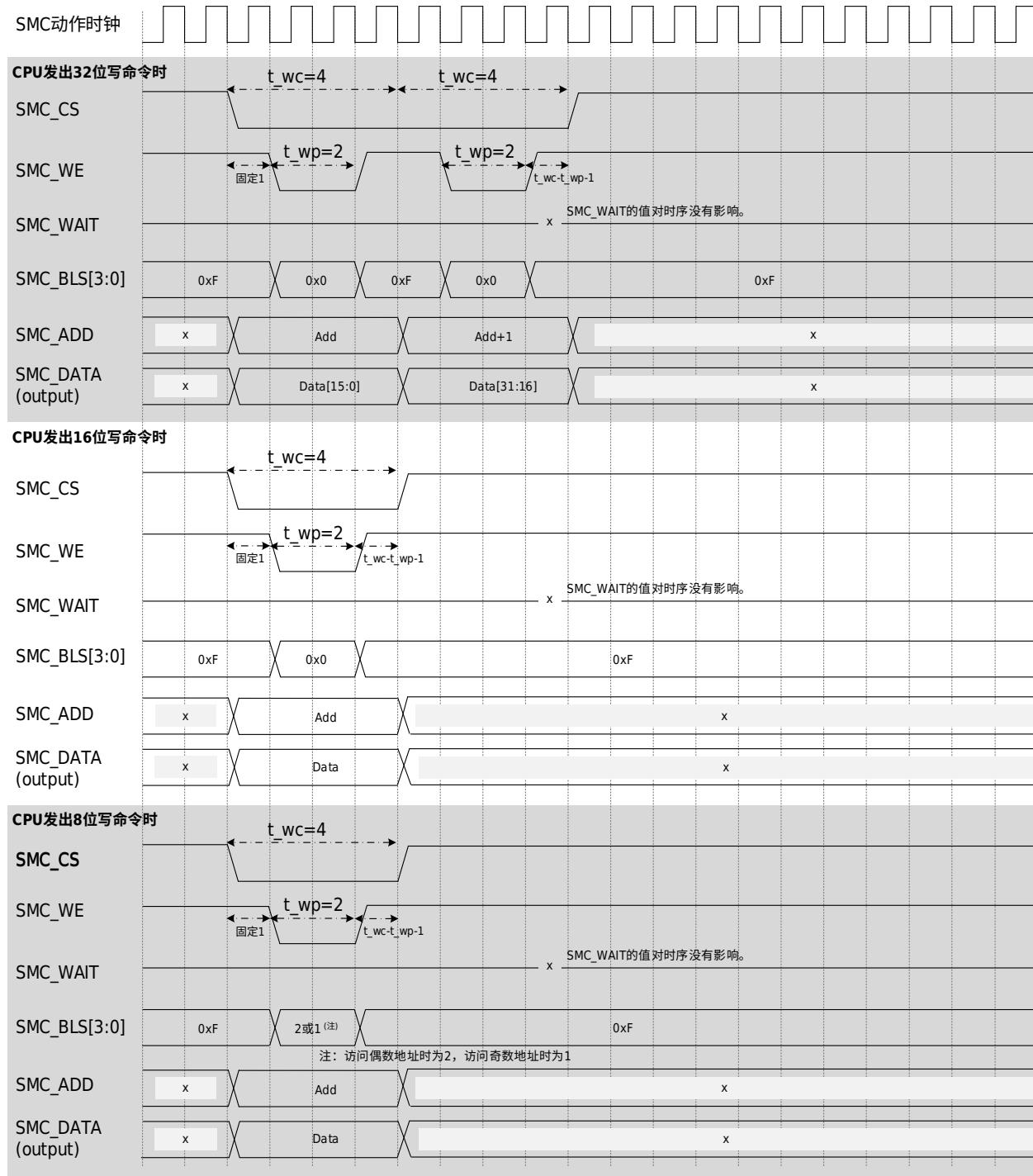


图 37-12 单次写动作基本时序 (异步方式 (WSYN=0) & 16 位位宽 (MW=01) & BLSS=1)

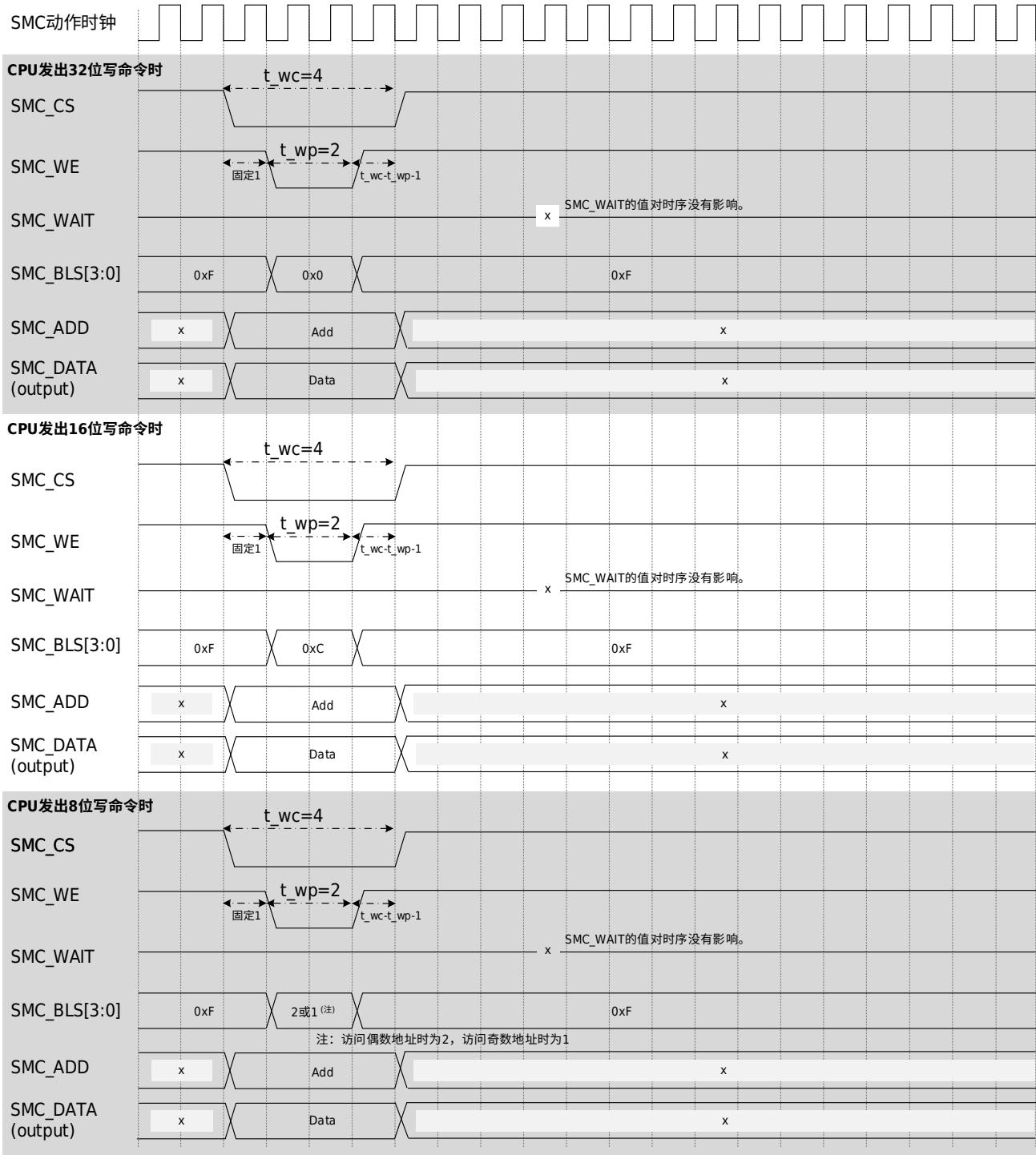


图 37-13 单次写动作基本时序 (异步方式 (WSYN=0) &32 位位宽 (MW=10) & BLSS=1)

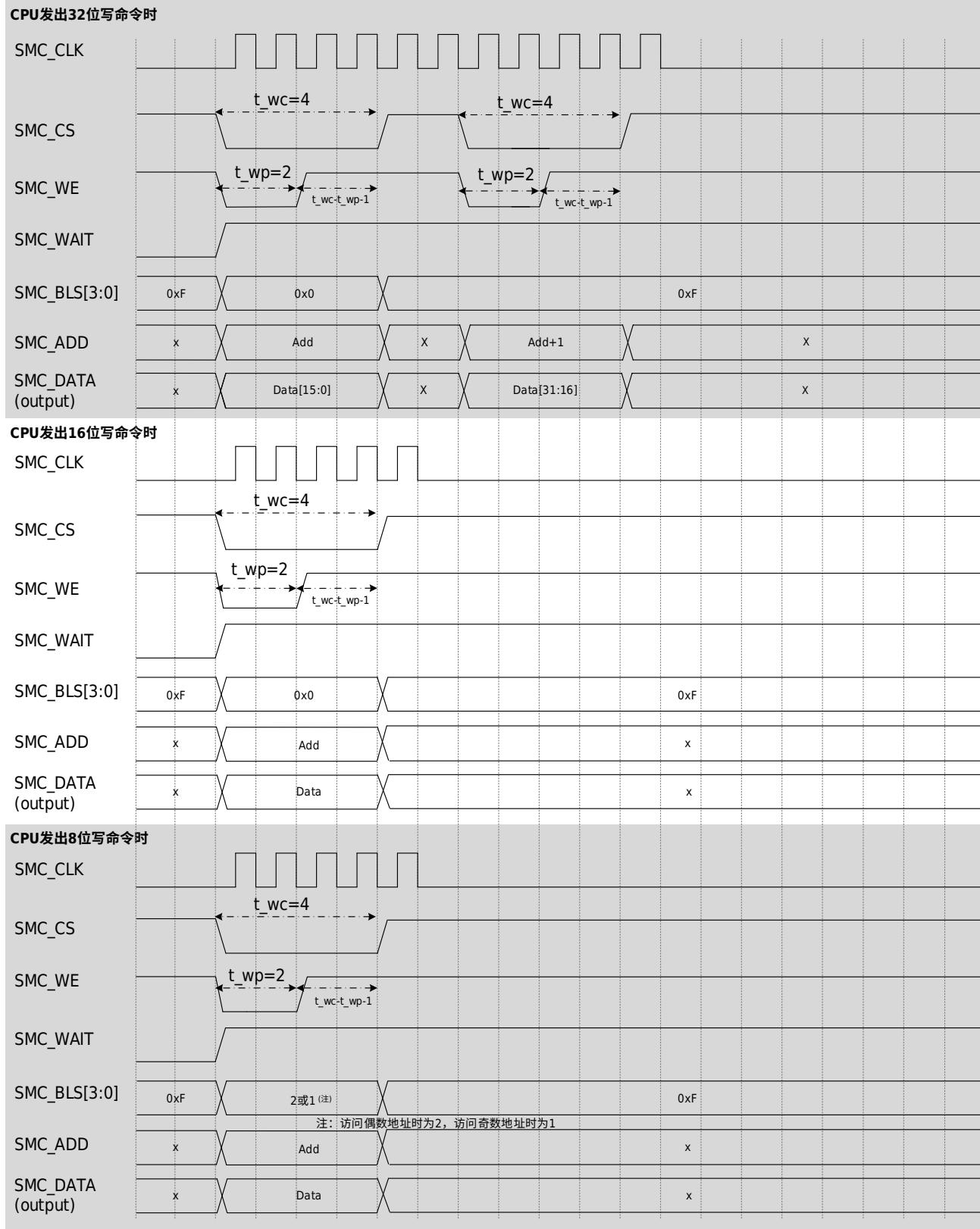


图 37-14 单次写动作基本时序 (同步方式 (WSYN=1) & 16 位位宽 (MW=01) & BLSS=0)

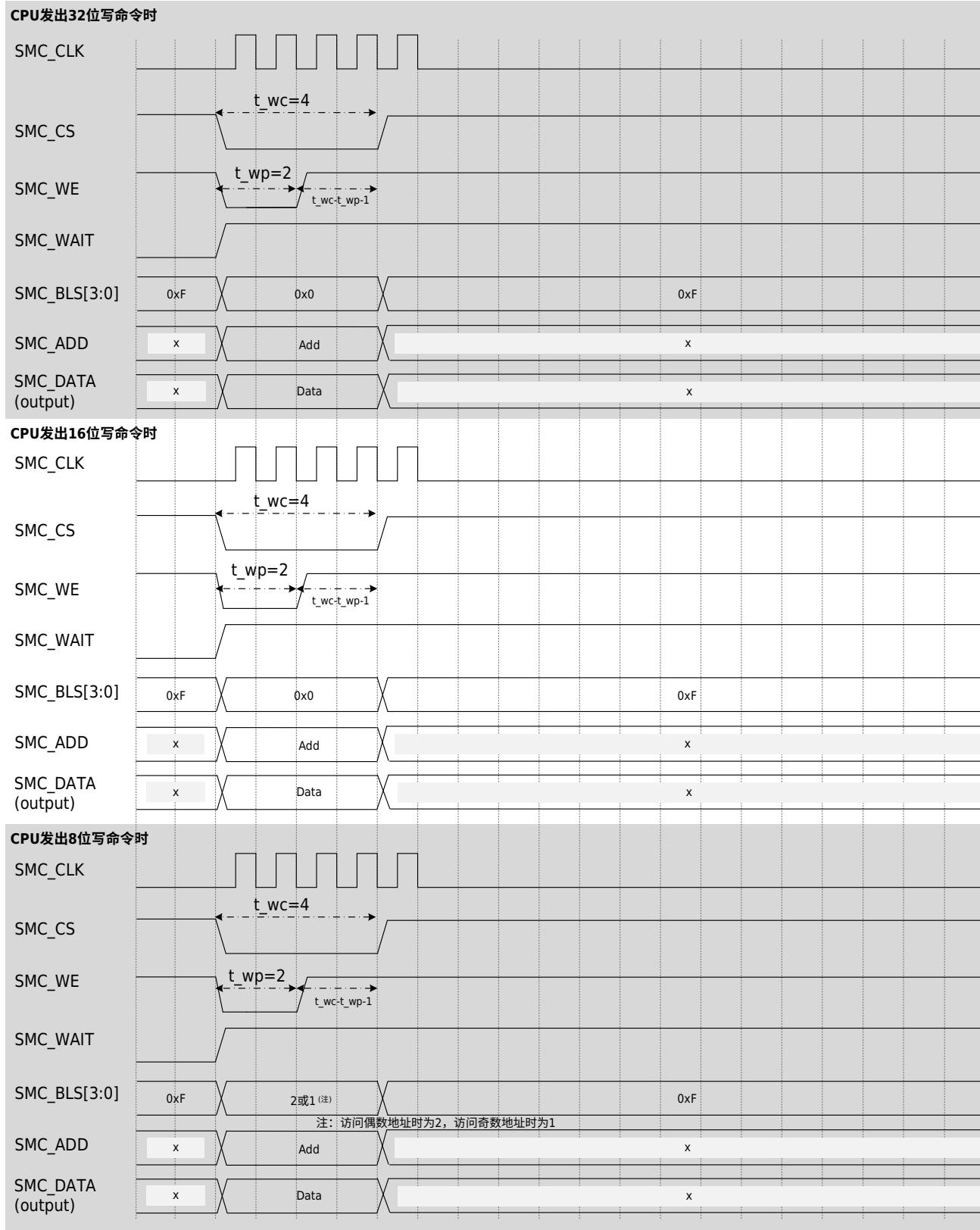


图 37-15 单次写动作基本时序（同步方式（WSYN=1）&32 位位宽（MW=10）& BLSS=0）

表 37-8 和图 37-16 是地址数据线复用的单次写动作的基本时序图和设定例。

**表 37-8 地址数据线复用单次写动作基本设定例**

基本设定	MW	RSYN	WSYN	BAA	ADV	BLS	- 表示不关注 <set> 表示用户设定值
	<set>	-	0	0b0	0b1	0b1	
时序设定	t_rc	t_wc	t_ceoe	t_wp	t_pc	t_tr	
	-	0b0101	-	0b010	-	0b010	

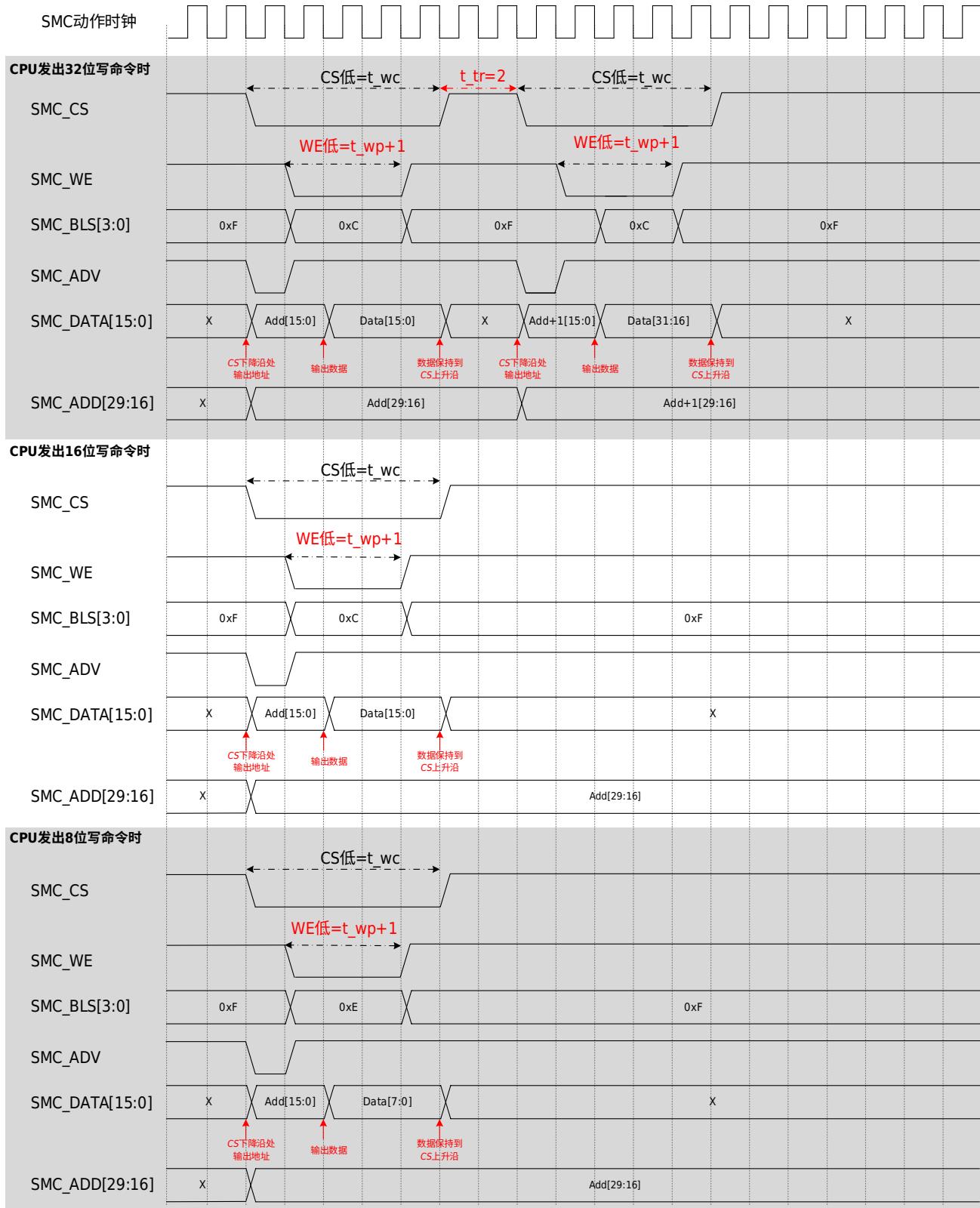


图 37-16 地址数据线复用单次写动作基本时序 (异步方式 (WSYN=0) &amp; 16 位位宽 (MW=01))

表 37-9 和图 37-17 地址数据线复用同步方式单次写动作的基本时序图和设定例。

表 37-9 地址数据线复用同步方式单次写动作基本设定例

基本设定	MW	RSYN	WSYN	BAA	ADVS	BLSS	- 表示不关注 <set> 表示用户设定值
	<set>	-	1	0b0	0b1	0b1	
时序设定	t_rc	t_wc	t_ceoe	t_wp	t_pc	t_tr	
	-	0b0101	-	0b0100	-	0b010	

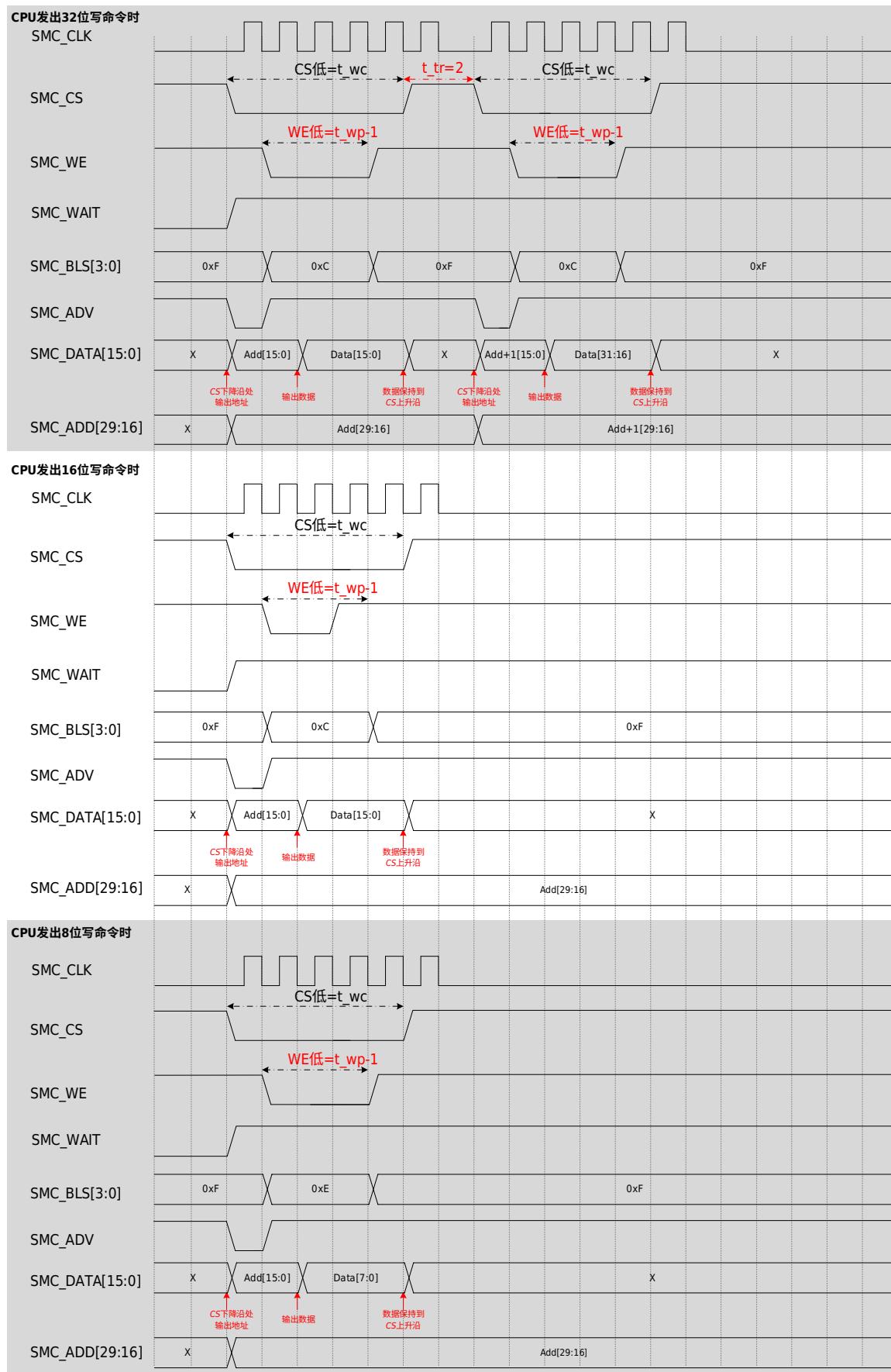


图 37-17 地址数据线复用单次写动作基本时序（同步方式（WSYN=1）& 16 位位宽（MW=01））

## FIFO 管理

SMC 中含有 3 类 FIFO。分别是命令 FIFO、写数据 FIFO、读数据 FIFO。

命令 FIFO：2 个命令 FIFO，用于缓存 2 次 SMC 命令及对应的命令地址、CS 等信息。

写数据 FIFO：4 个写数据 FIFO，用于缓存 AHB 的写数据。

读数据 FIFO：4 个读数据 FIFO，用于缓存从 SRAM/PSRAM/NOR Flash 读入的数据。

### 37.3.1.5 SMC 低功耗管理

SMC 处于 Low Power 状态时，如果将状态控制寄存器（SMC\_STCR1）的 LPWOR 位设定为 1，则 SMC 从 Low Power 状态转为 Ready 状态，SMC 开始正常工作；如果对 SMC 长时间无操作，可将状态控制寄存器（SMC\_STCR0）的 LPWIR 位设定为 1，使 SMC 进入 Low Power 状态，进入 Low Power 状态后，SMC 的内部动作停止，功耗会降低。

## 37.4 寄存器说明

表 37-10 EXMC 寄存器列表

SMC: 0x88000000

寄存器名	符号	偏移量	位宽	复位值
SMC	使能寄存器	SMC_ENAR	0x4005540C	32
	状态寄存器	SMC_STSR	0x0000	32
	命令寄存器	SMC_CMDR	0x0010	32
	状态控制寄存器 0	SMC_STCR0	0x0008	32
	状态控制寄存器 1	SMC_STCR1	0x000c	32
	刷新时间寄存器	SMC_RFTR	0x0020	32
	基本控制寄存器	SMC_BACR	0x0200	32
	片选控制寄存器	SMC_CSCR0	0x0208	32
	片选控制寄存器	SMC_CSCR1	0x020C	32
	CHIP 配置寄存器	SMC_CPCR	0x0018	32
	CHIP 状态寄存器 0	SMC_CPSR0	0x0104	32
	CHIP 状态寄存器 1	SMC_CPSR1	0x0124	32
	CHIP 状态寄存器 2	SMC_CPSR2	0x0144	32
	CHIP 状态寄存器 3	SMC_CPSR3	0x0164	32
	时序配置寄存器	SMC_TMCR	0x0014	32
	时序状态寄存器 0	SMC_TMSR0	0x0100	32
	时序状态寄存器 1	SMC_TMSR1	0x0120	32
	时序状态寄存器 2	SMC_TMSR2	0x0140	32
	时序状态寄存器 3	SMC_TMSR3	0x0160	32

### 37.4.1 SMC-SRAM/PSRAM/NOR Flash 寄存器

#### 37.4.1.1 使能寄存器 (SMC\_ENAR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
SMCEN															
-															
位	标记	位名	功能	读写											
b31~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b1	SMCEN	SMC使能	0: SMC模块的功能端口无效 1: SMC模块的功能端口有效	R/W											
b0	Reserved	-	读出时为“0”,写入时写“0”	R/W											

#### 37.4.1.2 状态寄存器 (SMC\_STSR)

复位值: 0x00000001

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
STATUS															
位	标记	位名	功能	读写											
b31~b1	Reserved	-	读出时为复位值	R											
b0	STATUS	SMC当前状态	0: Ready 1: Low Power	R											

### 37.4.1.3 命令寄存器 (SMC\_CMDR)

复位值: -

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved				CMDCHIP[2:0]				CMD[1:0]		CRES		CMDADD[19:0]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMDADD[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b26	Reserved	-	写入时写“0”	W											
<hr/>															
				CMD命令的Chip对象											
				000: Chip 0											
b25~b23	CMDCHIP[1:0]	CS对象设定	001: Chip 1 010: Chip 2 011: Chip 3 请不要设定其它值	00: 无效 01: MdRegConfig 10: UpdateRegs 11: MdRegConfig & UpdateRegs	W										
<hr/>															
b22~b21	CMD[1:0]	命令输入	0: SMC_CRE输出低电平 1: 在发出MdregConfig命令时, SMC_CRE端口输出高电平	00: 无效 01: MdRegConfig 10: UpdateRegs 11: MdRegConfig & UpdateRegs	W										
b20	CRES	CRE端口设定	当CMD=01或11时, CMDADD[19:0]表示要访问的外部MEM地址	0: SMC_CRE输出低电平 1: 在发出MdregConfig命令时, SMC_CRE端口输出高电平	W										
b19~b0	CMDADD[19:0]	地址对象设定	当CMD=01或11时, CMDADD[19:0]表示要访问的外部MEM地址	当CMD=01或11时, CMDADD[19:0]表示要访问的外部MEM地址	W										

### 37.4.1.4 状态控制寄存器 (SMC\_STCR0)

复位值: -

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
位	标记	位名	功能	读写											
b31~b3	Reserved	-	写入时写“0”	W											
b2	LPWIR	Low Power状态进入请求	0: 无请求 1: Low Power状态进入	W											
b1~b0	Reserved	-	写入时写“0”	W											

### 37.4.1.5 状态控制寄存器 (SMC\_STCR1)

复位值: -

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
位	标记	位名	功能	读写											
b31~b3	Reserved	-	写入时写“0”	W											
b2	LPWOR	Low Power状态退出请求	0: 无请求 1: Low Power状态退出	W											
b1~b0	Reserved	-	写入时写“0”	W											

### 37.4.1.6 刷新时间寄存器 (SMC\_RFTR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved										REFPRD[3:0]					
<hr/>															
位	标记	位名	功能	读写											
b31~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b3~b0	REFPRD[3:0]	刷新周期	刷新周期值设定	R/W											

### 37.4.1.7 基本控制寄存器 (SMC\_BACR)

复位值: 0x00000300

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CKSEL[1:0]	Reserved										MUXMD	Reserved			
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W											
00: 读取锁存SMC_DATA、SMC_WAIT端口数据时, 使用内部EXCLK															
b15~b14	CKSEL[1:0]	采样时钟选择	01: 读取锁存SMC_DATA、SMC_WAIT端口数据时, 使用内部EXCLK取反做时钟 10: 读取锁存SMC_DATA、SMC_WAIT端口数据时, 使用EXMC_CLK端口状态反馈做时钟 11: 设定禁止	R/W											
b13~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b9~b8	Reserved	-	读出时为“1”,写入时写“1”	R/W											
b7~b5	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b4	MUXMD	SMC地址数据复用	0: SMC地址和数据不复用 1: SMC地址和数据复用	R/W											
b3~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W											

### 37.4.1.8 片选控制寄存器 (SMC\_CSCR0)

复位值: 0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ADDMSK3[7:0]								ADDMSK2[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDMSK1[7:0]								ADDMSK0[7:0]							

位	标记	位名	功能	读写
b31~b24	ADDMSK3[7:0]	CS3空间屏蔽地址	CS3空间屏蔽地址设定	R/W
b23~b16	ADDMSK2[7:0]	CS2空间屏蔽地址	CS2空间屏蔽地址设定	R/W
b15~b8	ADDMSK1[7:0]	CS1空间屏蔽地址	CS1空间屏蔽地址设定	R/W
b7~b0	ADDMSK0[7:0]	CS0空间屏蔽地址	CS0空间屏蔽地址设定	R/W

### 37.4.1.9 片选控制寄存器 (SMC\_CSCR1)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ADDMAT3[7:0]								ADDMAT2[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDMAT1[7:0]								ADDMAT0[7:0]							

位	标记	位名	功能	读写
b31~b24	ADDMAT3[7:0]	CS3空间匹配地址	CS3空间匹配地址设定	R/W
b23~b16	ADDMAT2[7:0]	CS2空间匹配地址	CS2空间匹配地址设定	R/W
b15~b8	ADDMAT1[7:0]	CS1空间匹配地址	CS1空间匹配地址设定	R/W
b7~b0	ADDMAT0[7:0]	CS0空间匹配地址	CS0空间匹配地址设定	R/W

### 37.4.1.10 CHIP 配置寄存器 (SMC\_CPCR)

复位值: -

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	BLSS	ADVS	BAAS	MW[1:0]	-	WSYN	-	RSYN							
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	写入时写“0”	W											
b15~b13	Reserved	-	写入时写“0”	W											
b12	BLSS	BLS端口设定	0: BLS的端口状态和CS端口的状态变化同步 1: BLS的端口状态和WE端口的状态变化同步	W											
b11	ADVS	ADV端口设定	0: ADV端口功能无效 1: 地址数据线复用功能有效时, ADV端口功能有效	W											
b10	BAAS	BAA端口设定	0: BAA端口功能无效 1: BAA端口功能有效	W											
b9~b8	MW[1:0]	存储器位宽选择	01: 16位位宽 10: 32位位宽 请不要设定其它值	W											
b7~b5	Reserved	-	写入时写“0”	W											
b4	WSYN	写同步	0: 异步写使能 1: 同步写使能	W											
b3~b1	Reserved	-	写入时写“0”	W											
b0	RSYN	读同步	0: 异步读使能 1: 同步读使能	W											

### 37.4.1.11 CHIP 状态寄存器 (SMC\_CPSR0~3)

复位值: 0xFF000A00

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ADDMAT[7:0]								ADDMSK[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	BLSS	ADVS	BAAS	MW[1:0]	-	-	-	WSYN	-	-	RSYN	-	-	-	-

位	标记	位名	功能	读写
b31~b24	ADDMAT[7:0]	CS空间匹配地址	匹配地址读出	R
b23~b16	ADDMSK[7:0]	CS空间屏蔽地址	屏蔽地址读出	R
b15~b13	-	-	读出为0	R
b12	BLSS	SMC_BLS状态	SMC_BLS状态读出	R
b11	ADVS	SMC_ADV状态	SMC_ADV状态读出	R
b10	BAAS	SMC_BAA状态	SMC_BAA状态读出	R
b9~b8	MW[1:0]	存储器位宽状态	存储器位宽状态读出	R
b7~b5	WBL[2:0]	-	读出为0	R
b4	WSYN	写动作方式状态	写动作方式状态读出	R
b3~b1	RBL[2:0]	-	读出为0	R
b0	RSYN	读动作方式状态	读动作方式状态读出	R

### 37.4.1.12 时序配置寄存器 (SMC\_TMCR)

复位值: -

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								t_tr[2:0]		-	t_pc[2:0]				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	t_wp[2:0]			-	t_ceoe[2:0]			t_wc[3:0]			t_rc[3:0]				

位	标记	位名	功能	读写
b31~b23	Reserved	-	写入时写“0”	W
b22~b20	t_tr[2:0]	t_tr周期设定	t_tr设定	W
b19	Reserved	-	写入时写“0”	W
b18~b16	t_pc[2:0]	t_pc周期设定	t_pc设定	W
b15	Reserved	-	写入时写“0”	W
b14~b12	t_wp[2:0]	t_wp周期设定	t_wp设定	W
b11	Reserved	-	写入时写“0”	W
b10~b8	t_ceoe[2:0]	t_ceoe周期设定	t_ceoe设定	W
b7~b4	t_wc[3:0]	t_wc周期设定	t_wc设定	W
b3~b0	t_rc[3:0]	t_rc周期设定	t_rc设定	W

### 37.4.1.13 时序状态寄存器 (SMC\_TMSR0~3)

复位值: 0x001263CC

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								t_tr[2:0]		-		t_pc[2:0]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	t_wp[2:0]			-	t_ceoe[2:0]			t_wc[3:0]			t_rc[3:0]				
<hr/>															
位	标记	位名	功能	读写											
b31~b23	Reserved	-	读出时为“0”	R											
b22~b20	t_tr[2:0]	t_tr周期状态	t_tr值读出	R											
b19	Reserved	-	读出时为“0”	R											
b18~b16	t_pc[2:0]	t_pc周期状态	t_pc值读出	R											
b15	Reserved	-	读出时为“0”	R											
b14~b12	t_wp[2:0]	t_wp周期状态	t_wp值读出	R											
b11	Reserved	-	读出时为“0”	R											
b10~b8	t_ceoe[2:0]	t_ceoe周期状态	t_ceoe值读出	R											
b7~b4	t_wc[3:0]	t_wc周期状态	t_wc值读出	R											
b3~b0	t_rc[3:0]	t_rc周期状态	t_rc值读出	R											

## 37.5 使用注意事项

- SMC 的相关寄存器只能在 Ready 状态下设定；
- 在使用 SMC 模块对外部 Memory 进行读访问时，Memory Device 在接收到控制器发出的读命令后将数据传送给控制器，控制器对传送回的数据进行采样的时钟可以通过 SMC\_BACR.CKSEL 进行选择，具体设定请参考寄存器章节。选择不同的采样时钟，可以改善通信路径的时间裕量。

下图是同步模式下的 SMC 访问例。

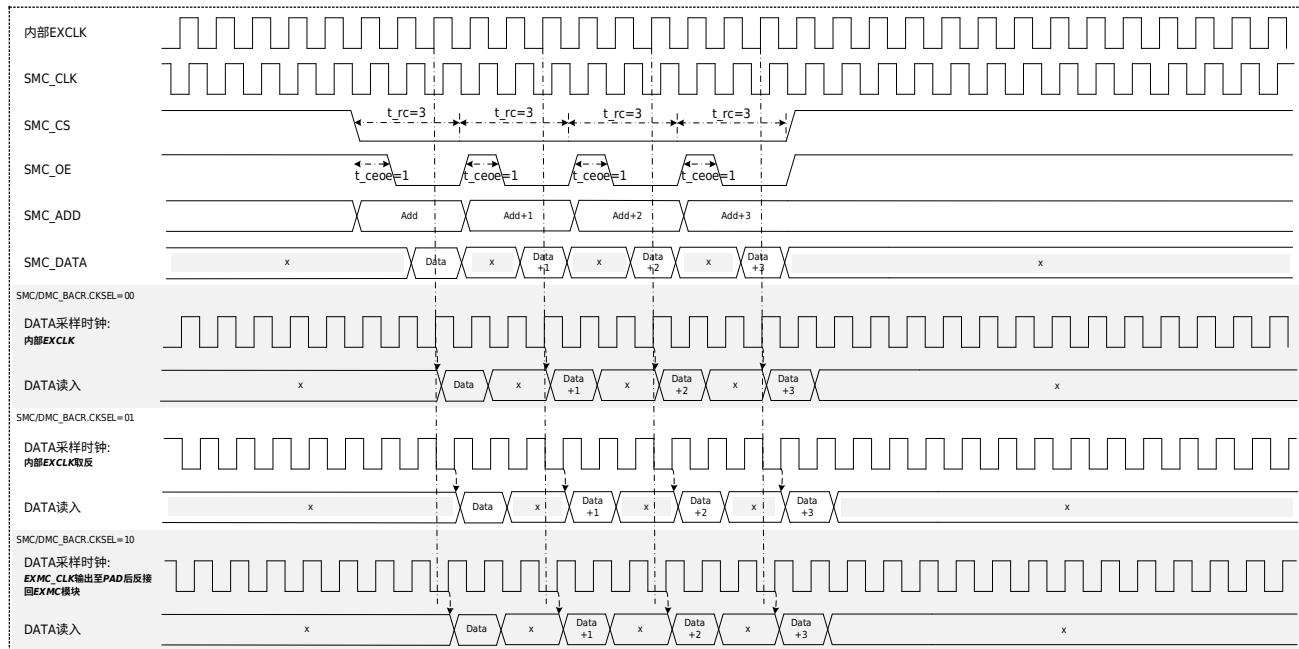


图 37-18 同步模式下 SMC 访问例

## 38 加密协处理模块 (CPM)

### 38.1 简介

加密协处理模块 (CPM) 包括 AES 加解密算法处理器，HASH 安全散列算法，TRNG 真随机数发生器三个子模块。

AES 加解密算法处理器遵循美国国家标准技术研究所 (NIST) 在 2000 年 10 月 2 日正式宣布的新的数据加密标准，分组长度固定为 128 位，而密钥长度支持 128/192/256 位。

HASH 安全散列算法是 SHA-2 版本的 SHA-256 (Secure Hash Algorithm)，符合美国国家标准和技术局发布的国家标准“FIPS PUB 180-3”，可以对长度不超过  $2^{64}$  位的消息产生 256 位的消息摘要输出。支持 HMAC (密钥散列消息认证码) 应用，使用 SHA-256 散列函数对消息进行认证。

TRNG 真随机数发生器是以连续模拟噪声为基础的随机数发生器，提供 64bit 随机数。

### 38.2 加解密算法处理器 (AES)

#### 38.2.1 算法简介

AES (The Advanced Encryption Standard) 是美国国家标准技术研究所 (NIST) 在 2000 年 10 月 2 日正式宣布的新的数据加密标准。

AES 的分组长度固定为 128 位，而密钥长度支持 128、192 和 256 位。对于加密来说，其输入是一个明文分组和一个密钥，输出是一个密文分组；对解密而言，输入是一个密文分组和一个密钥，而输出是一个明文分组。此过程如图 38-1 所示：

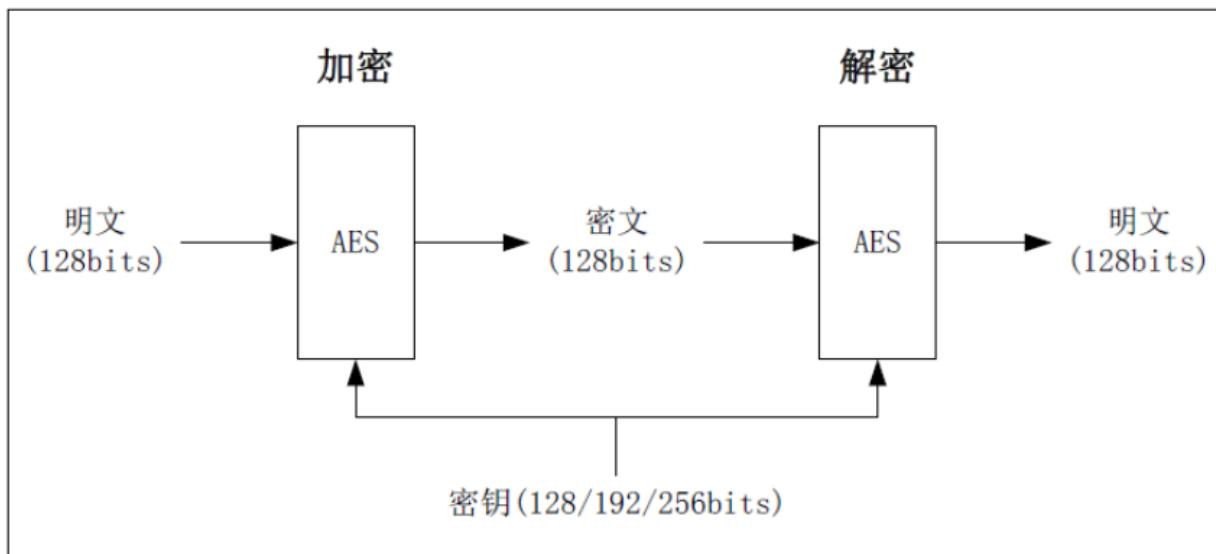


图 38-1 AES 加解密流程示意图

AES 算法处理的基本单位是字节，128 位信息被分成 16 个字节，按顺序复制到一个  $4 \times 4$  的矩阵中，称为状态 (state)，AES 的所有变换都是基于状态矩阵的变换，该矩阵上保存着计算的中间结果。

AES 是一个密钥迭代分组密码，包含了轮变换对状态的重复作用。AES 的轮变换由四个操作组成：SubBytes、ShiftRows、MixColumns、AddRoundKey。其中，SubBytes 包括求每个字节在  $GF(2^8)$  中的模逆元和一个仿射变换；ShiftRows 是一个字节换位，它将状态中的行按照不同的偏移量进行循环移位；MixColumns 对状态各列进行线性变换；AddRoundKey，状态中的各字节与轮密钥进行逐位异或操作。AES 的加密流程如图 38-2 所示：

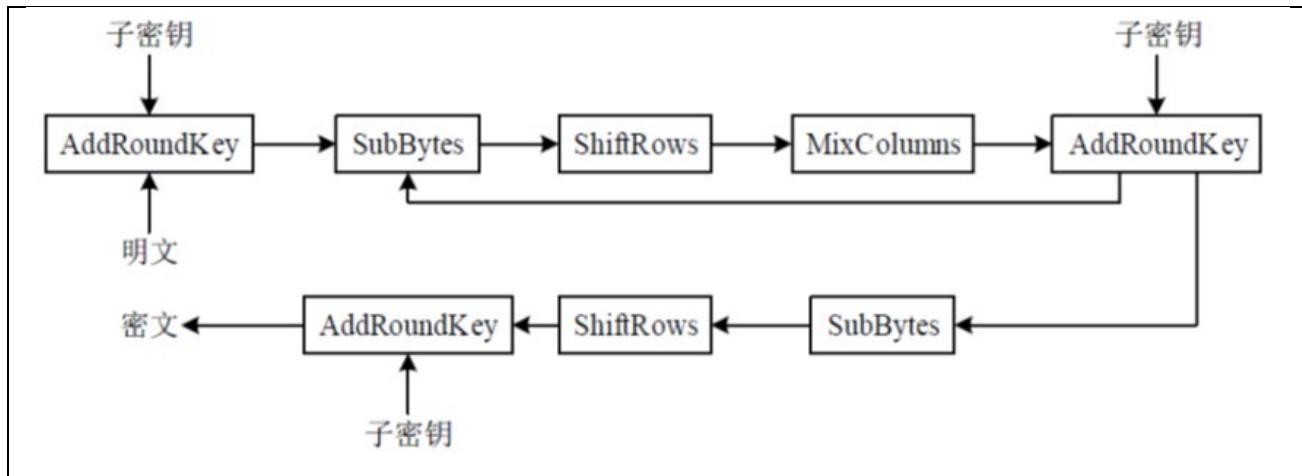


图 38-2 AES 加密流程图

图中所用子密钥需要由初始密钥扩展而来，且密钥的扩展过程和加密过程是同步进行的。

由于明文固定为 128 位，加密过程运行的轮数就取决于密钥的长度。比如，密钥为 128 位时，运行轮数为 10 轮；密钥为 192 位时，运行轮数为 12 轮；密钥为 256 位时，运行轮数为 14 轮。除了最后一轮缺少 MixColumns 变换，其余各轮均进行完整的轮变换操作。

解密流程与加密流程有所区别，首先必须完成所有密钥的扩展，解密过程从扩展的最后一轮子密钥往回使用；然后是轮变换的四个操作变成了相应的逆运算：InvSubBytes、InvShiftRows、InvMixColumns、AddRoundKey。InvSubBytes 中的模逆运算仍然保持，但仿射变换改为逆变换；InvShiftRows 和 InvMixColumns 变成相应的逆变换；AddRoundKey 保持不变。

直接解密流程的轮变换对四个操作的调用顺序为：InvShiftRows、InvSubBytes、AddRoundKey、InvMixColumns，与加密流程的调用顺序不一致，但使用的密钥与加密流程一致；等价解密流程的轮变换对四个操作的调用顺序为：InvSubBytes、InvShiftRows、InvMixColumns、AddRoundKey，与加密流程的调用顺序完全一致，只是每一轮的子密钥需要进行 InvMixColumns 运算。

关于详细的算法表述，可以参见标准《FIPS PUB 197》

### 38.2.2 AES 模块功能描述

- 执行 AES 算法标准的加密流程和解密流程，其执行结果完全符合《FIPS PUB 197》对算法原理的描述；
- 支持 128、192 和 256 位密钥。

### 38.2.3 加密操作流程

AES 的加密操作流程例如下：

- 1) 将待加密的 128 位数据写入数据寄存器 (AES\_DR) 中。
- 2) 将加密密钥写入密钥寄存器 (AES\_KR) 中。
- 3) 根据密钥长度设置 AES\_CR.KEYSIZE。
- 4) 将 AES\_CR.MODE 设置为 0，启动加密模式。
- 5) 向控制寄存器中的 AES\_CR.START 写入 1，启动模块进行运算。  
3)、4) 和 5) 可同时进行。
- 6) 等待 AES\_CR.START 的值恢复为 0，模块运算结束。
- 7) 读取数据寄存器 (AES\_DR)，获得 128 位密文。

### 38.2.4 解密操作流程

AES 的解密操作流程例如下：

- 1) 将待解密的 128 位数据写入数据寄存器 (AES\_DR) 中。
- 2) 将解密密钥写入密钥寄存器 (AES\_KR) 中。
- 3) 根据密钥长度设置 AES\_CR.KEYSIZE。
- 4) 将 AES\_CR.MODE 设置为 1，启动解密模式。
- 5) 向控制寄存器中的 AES\_CR.START 写入 1，启动模块进行运算。  
3)、4) 和 5) 可同时进行。
- 6) 等待 AES\_CR.START 的值恢复为 0，模块运算结束。
- 7) 读取数据寄存器 (AES\_DR)，获得 128 位明文。

### 38.2.5 数据示例

128 位明文:

0xFFEEDCCBAA99887766554433221100

128 位密钥:

0x0F0E0D0C0B0A09080706050403020100

128 位密文:

0x5AC5B47080B7CDD830047B6AD8E0C469

表 38-1 128 位操作寄存器示例

加密前			
寄存器	值 (密钥)	寄存器	值 (明文)
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC

加密后			
寄存器	值 (密钥)	寄存器	值 (密文)
Key0	0x03020100	Data0	0xD8E0C469
Key1	0x07060504	Data1	0x30047B6A
Key2	0x0B0A0908	Data2	0x80B7CDD8
Key3	0x0F0E0D0C	Data3	0x5AC5B470

128 位明文：

0xFFEEDDCCBAA99887766554433221100

192 位密钥：

0x17161514131211100F0E0D0C0B0A09080706050403020100

128 位密文：

0x5AC5B47080B7CDD830047B6AD8E0C469

表 38-2 192 位操作寄存器示例

加密前			
寄存器	值 (密钥)	寄存器	值 (明文)
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC
Key4	0x13121110		
Key5	0x17161514		

加密后			
寄存器	值 (密钥)	寄存器	值 (密文)
Key0	0x03020100	Data0	0xA47CA9DD
Key1	0x07060504	Data1	0xE0DF4C86
Key2	0x0B0A0908	Data2	0xA070AF6E
Key3	0x0F0E0D0C	Data3	0x91710DEC
Key4	0x13121110		
Key5	0x17161514		

128 位明文：

0xFFEEDDCCBAA99887766554433221100

256 位密钥：

0x1F1E1D1C1B1A191817161514131211100F0E0D0C0B0A09080706050403020100

128 位密文：

0x5AC5B47080B7CDD830047B6AD8E0C469

表 38-3 256 位操作寄存器示例

加密前			
寄存器	值 (密钥)	寄存器	值 (明文)
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC
Key4	0x13121110		
Key5	0x17161514		
Key6	0x1B1A1918		
Key7	0x1F1E1D1C		

加密后			
寄存器	值 (密钥)	寄存器	值 (密文)
Key0	0x03020100	Data0	0xCAB7A28E
Key1	0x07060504	Data1	0xBF456751
Key2	0x0B0A0908	Data2	0x9049FCEA
Key3	0x0F0E0D0C	Data3	0x8960494B
Key4	0x13121110		
Key5	0x17161514		
Key6	0x1B1A1918		
Key7	0x1F1E1D1C		

### 38.2.6 运行时间说明

AES 模块从启动一次运算 (AES\_CR.START 写入 1) 到该次运算结束 (AES\_CR.START 恢复到 0) 所需时间如下表所示：

表 38-4 AES 加解密运行时间

	128位密钥	192位密钥	256位密钥
加密	220 cycles	260 cycles	300 cycles
解密	290 cycles	332 cycles	398 cycles

### 38.2.7 操作注意事项

- 在 AES 加解密过程中，数据寄存器会改变，如果下次运算操作的数据就是本次运算的结果，就无需重新写入数据。
- 支持 128、192 和 256 位密钥，128 位密钥写入偏移地址 0x20~0x2C，192 位密钥写入偏移地址 0x20~0x34，256 位密钥写入偏移地址 0x20~0x3C。
- 判断模块运算结束的方法：不断读取 AES\_CR.START，如果其值变为 0，则表示运算结束。

### 38.2.8 寄存器说明

表 38-5 寄存器列表

寄存器地址: 0x40008000

寄存器名	符号	偏移量	位宽	复位值
AES控制寄存器	AES_CR	0x0000	32	0x00000000
AES数据寄存器0~3	AES_DR0~3	0x0010~0x001C	32	0x00000000
AES密钥寄存器0~7	AES_KR0~7	0x0020~0x003C	32	0x00000000

#### 38.2.8.1 AES 控制寄存器 (AES\_CR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
<hr/>															
位	标记	位名	功能	读写											
b31~b5	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b4~b3	KEYSIZE	密钥长度选择	2'b00: 密钥长度为128位 2'b01: 密钥长度为192位 2'b10: 密钥长度为256位 2'b11: 密钥长度为128位	R/W											
b2	Reserved	-	读出时为“0”,写入时写“0”	-											
b1	MODE	加密解密模式选择	0: 加密运算 1: 解密运算	R/W											
b0	START	启动	0: 本模块运算结束或未被启动 1: 启动本模块进行运算	R/W											

说明:

1. AES\_CR.START 位的操作方法是：软件对本位写入 1 后，本模块将启动运行，本次运行结束后本模块硬件会自动将本位清 0，软件查询到本位为 0 即表示本次运行完成。
2. 对本寄存器的写入操作只能在本模块不处于运算状态时（即 AES\_CR.START = 0 时）才能进行，否则硬件将自动忽略写操作。读操作则不受此限制。

**38.2.8.2 AES 数据寄存器 (AES\_DRx) (x=0~3)**

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16										
Data[31:16]																									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0										
Data[15:0]																									
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">位</th><th style="text-align: left; padding: 2px;">标记</th><th style="text-align: left; padding: 2px;">位名</th><th style="text-align: left; padding: 2px;">功能</th><th style="text-align: right; padding: 2px;">读写</th></tr> </thead> <tbody> <tr> <td style="padding: 2px;">b31~b0</td><td style="padding: 2px;">Data</td><td style="padding: 2px;">加密数据</td><td style="padding: 2px;">存放AES算法的128比特明文/密文</td><td style="text-align: right; padding: 2px;">R/W</td></tr> </tbody> </table>																位	标记	位名	功能	读写	b31~b0	Data	加密数据	存放AES算法的128比特明文/密文	R/W
位	标记	位名	功能	读写																					
b31~b0	Data	加密数据	存放AES算法的128比特明文/密文	R/W																					

说明：

1. 数据寄存器由 4 个 32 位的寄存器组成 128 位数据，用于在模块运算前存放需要被加密的明文或者需要被解密的密文，并且运算完成后存放加密后的密文或者解密后的明文。

加密运算		解密运算	
运算前	运算后	运算前	运算后
128位明文	128位密文	128位密文	128位明文

四个 32 位寄存器连接在一起组成一个 128 位的数据，读写操作时需要分别对四个寄存器进行操作。数据寄存器对应的操作顺序如下：

数据举例：0xFFEEDDCCBAA99887766554433221100

偏移地址	寄存器名称	填入数据
0x10	AES_DR0	0x33221100
0x14	AES_DR1	0x77665544
0x18	AES_DR2	0xBBAA9988
0x1C	AES_DR3	0xFFEEDDCC

2. 对于本寄存器的写入只能在本模块没有处于运算状态时（即 AES\_CR.START = 0 时）才能进行，否则硬件将自动忽略对本寄存器的写操作。
3. 对于本寄存器的读取只能在本模块没有处于运算状态时（即 AES\_CR.START = 0 时）才能进行，否则对本寄存器的读取将得到全 0。

**38.2.8.3 AES 密钥寄存器 (AES\_KRx) (x=0~7)**

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16										
Key[31:16]																									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0										
Key[15:0]																									
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; width: 10%;">位</th><th style="text-align: left; width: 10%;">标记</th><th style="text-align: left; width: 10%;">位名</th><th style="text-align: left; width: 10%;">功能</th><th style="text-align: right; width: 10%;">读写</th></tr> </thead> <tbody> <tr> <td>b31~b0</td><td>Key</td><td>密钥</td><td>存放AES算法的128/192/256位密钥</td><td style="text-align: right;">R/W</td></tr> </tbody> </table>																位	标记	位名	功能	读写	b31~b0	Key	密钥	存放AES算法的128/192/256位密钥	R/W
位	标记	位名	功能	读写																					
b31~b0	Key	密钥	存放AES算法的128/192/256位密钥	R/W																					

说明：

- 密钥寄存器由 8 个 32 位的寄存器组成，存放输入的初始密钥。写操作时需要分别对 8 个 32 位的寄存器进行操作。

对应的操作顺序如下：

128 位密钥举例：0x0F0E0D0C\_0B0A0908\_07060504\_03020100

偏移地址	寄存器名称	填入数据
0x20	AES_KR0	0x03020100
0x24	AES_KR1	0x07060504
0x28	AES_KR2	0xB0A0908
0x2C	AES_KR3	0xF0E0D0C

192 位密钥举例：0x17161514\_13121110\_0F0E0D0C\_0B0A0908\_07060504\_03020100

偏移地址	寄存器名称	填入数据
0x20	AES_KR0	0x03020100
0x24	AES_KR1	0x07060504
0x28	AES_KR2	0xB0A0908
0x2C	AES_KR3	0xF0E0D0C
0x30	AES_KR4	0x13121110
0x34	AES_KR5	0x17161514

256 位密钥举例：

0x1F1E1D1C\_1B1A1918\_17161514\_13121110\_0F0E0D0C\_0B0A0908\_07060504\_03020100

偏移地址	寄存器名称	填入数据
0x20	AES_KR0	0x03020100
0x24	AES_KR1	0x07060504
0x28	AES_KR2	0x0B0A0908
0x2C	AES_KR3	0x0F0E0D0C
0x30	AES_KR4	0x13121110
0x34	AES_KR5	0x17161514
0x38	AES_KR6	0x1B1A1918
0x3C	AES_KR7	0x1F1E1D1C

2. 对于本寄存器的写入只能在本模块没有处于运算状态时（即 AES\_CR.START = 0 时）才能进行，否则硬件将自动忽略对本寄存器的写操作。
3. 对于本寄存器的读取只能在本模块没有处于运算状态时（即 AES\_CR.START = 0 时）才能进行，否则对本寄存器的读取将得到全 0。

## 38.3 安全散列算法 (HASH)

### 38.3.1 算法简介

安全散列算法的步骤如下：

首先填充消息使其长度恰好为一个比 512 的倍数仅小 64 位的数。填充方法是附一个 1 在消息后面，后接所要求的多个 0，然后在其后附上 64 位的消息长度(填充前)，使消息长度恰好是 512 位的整数倍。

其次将 A、B、C、D、E、F、G、H 8 个 32 位变量，用十六进制初始化。然后开始算法的主循环，一次处理 512 位消息，循环次数是消息中 512 位分组的数目。

主循环一共进行 64 次操作，此操作称为压缩函数。每次操作都包含移位、循环移位、逻辑运算、模 232 加等，运算的过程见下图 38-3。最后的输出由 A、B、C、D、E、F、G、H 级联而成。其中  $W_t$  为由 512 位消息得到的第  $t$  步所用的临时值， $K_t$  为第  $t$  步所用的常数值， $t$  ( $0 \leq t \leq 63$ ) 是 64 步循环中的一步。

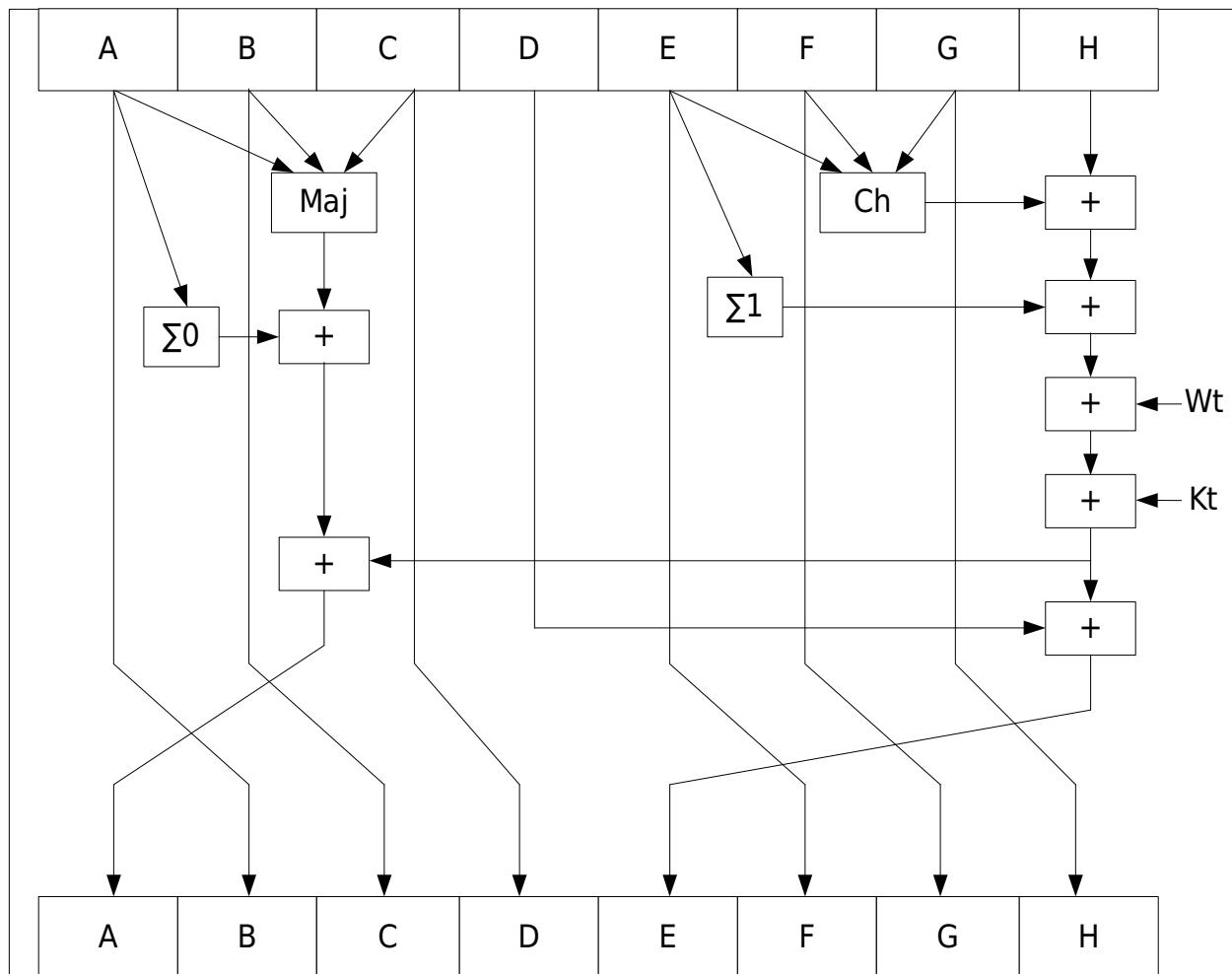


图 38-3 HASH 算法流程图

HASH 模块集成了硬件 HMAC 功能；也可以通过软件方式根据 HMAC 算法直接调用 HASH 运算实现 HMAC 功能。

### 38.3.2 操作流程

HSAH 模块的操作流程如下：

软件操作流程：

- 1) 软件将原始数据按照算法规则进行填充，并且将填充后的消息按 512 位进行分组。
- 2) 将数据写入数据寄存器 HASH\_DR 中。
- 3) 如果本次运算为消息分组的第一组数据，设置 HASH\_CR.FST\_GRP=1。
- 4) 设置 HASH\_CR.START=1，启动本模块进行运算。

注：以上的 3) 和 4) 可以同时进行

- 5) 判断本模块本次运算是否完成，通过以下方法：

不断读取 HASH\_CR.START 直到读到该位为 0 即表示运算完成；或者不断读取 HASH\_CR.BUSY 直到读到该位为 0 即表示运算完成；又或者设置 HASH 中断，通过中断方式判断运算结束。

- 6) 如果本次运算不是消息分组的最后一组数据，返回到 2)。
- 7) 如果本次运算为消息分组的最后一组数据，读取摘要寄存器 HASH\_HR 获得本次运算的结果。如需再次进行运算则返回到步骤 1)。

使用 DMA 操作流程：

1. 将需要处理的消息按照 512 比特进行分组，最后一组消息需要按照规则预先填充为 512 比特的块。
2. 设置内部触发事件寄存器的 HASH 运算启动方式为 DMA 传送完成和数据块传送完成；设置内部触发事件寄存器的 DMA 传送开始方式为 HASH 运算完成（或者其他内部触发事件）。
3. 设置 HASH\_CR.MODE=2'b00，HASH 运算方式；设置 HASH\_CR.FST\_GRP=1，并设置 HASH\_CR.START=1，启动 HASH 运算。如果选择其他内部触发事件启动 DMA，则需要等待触发事件发生后再开始 HASH 运算。
4. 等待 HASH\_CR.CYC\_END=1，HASH 运算完成，并清除该位。
5. 读取摘要寄存器 HASH\_HR 的结果。

**注意：**当只有一组运算数据时，请同时设置 HASH\_CR.FST\_GRP=1 和 HASH\_CR.KMSG\_END=1。

### 38.3.3 消息填充

SHA-256 的填充分组处理步骤如下：

#### 1. 原始消息分组

将原始消息以 512bit 的大小分为 L 组。令原始消息总的 bit 个数为 I。如果  $I \% 512 < 448$ ，那么分组数 L 为  $I / 512$ ；如果  $I \% 512 \geq 448$ ，那么分组数 L 为  $I / 512 + 1$ 。

#### 2. 添加长度

##### ■ 添加填充位：

在消息分组的第  $I / 512$  组的最后添加填充位：一个 1 和若干个 0，0 的个数可以为零。如果  $I \% 512 < 448$ ，填充使得数据位的长度满足长度为  $448 \bmod 512$ （后 64bit 留做原始消息长度的表示）；如果  $I \% 512 \geq 448$ ，用一个 1 和若干个 0 把第  $I / 512$  组的 512bit 的数据块填满，并将第 L ( $L = I / 512 + 1$ ) 组的前 448 位填 0。

##### ■ 添加原始消息长度：

一个 64bit 块，表示原始消息长度，为 64bit 无符号整数。将原始消息长度添加在第 L 个分组的最后 64bit 处。

举例说明填充分组的过程如下：

#### 1) Example1:

原始消息为字符串“abcde”，用二进制位串表示其 ASCII 码为：“01100001 01100010 01100011 01100100 01100101”，添加长度的步骤如下：

- 添加“1”。填充后的消息为“01100001 01100010 01100011 01100100 01100101 1”。
- 添加“0”。因为原始消息长度为 40bit，所以要添加的 0 的个数为  $512 - 64 - 40 - 1 = 407$ 。

填充后的消息变为（十六进制）：

61626364 65800000 00000000 00000000  
00000000 00000000 00000000 00000000  
00000000 00000000 00000000 00000000  
00000000 00000000

- 添加原始消息长度。原始消息长度 40 的用两个 32bit 字表达为（十六进制）：00000000 00000028。

填充后的消息变为（十六进制）：

61626364 65800000 00000000 00000000  
00000000 00000000 00000000 00000000  
00000000 00000000 00000000 00000000  
00000000 00000000 00000000 00000028

#### 2) Example2:

原始消息为字符串“abcdefghijklmnopqrstuvwxyz”。每个字符都可通过其 ASCII 码转换为 8bit，所以消息的长度为  $I = 56 * 8 = 448$ 。

- 添加“1”和“0”。填充后的消息（十六进制）为第一个消息块：

61626364 62636465 63646566 64656667  
65666768 66676869 6768696A 68696A6B  
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F  
6D6E6F70 6E6F7071 80000000 00000000.

B. 添加原始消息长度。原始消息长度 448 的用两个 32bit 字表达为(十六进制):00000000 000001C0。

填充后的消息 (十六进制) 为第二个消息块:

00000000 00000000 00000000 00000000  
00000000 00000000 00000000 00000000  
00000000 00000000 00000000 00000000  
00000000 00000000 00000000 000001C0

### 38.3.4 HMAC 运算

HMAC (Keyed-Hashing for Message Authentication) 以不可逆方式将正在处理的消息与用户所选的密钥进行绑定，从而用于消息验证。基本而言，HMAC 是通过加入共享密钥的方式由两个嵌套的底层 HASH 运算组成：

$\text{HMAC}(\text{message}) = \text{Hash}[(\text{key XOR opad}) \mid \text{Hash}((\text{key XOR ipad}) \mid \text{message})]$

其中：

ipad: 64 字节(SHA-256) 0x36。

opad:64 字节(SHA-256) 0x5C。

| :表示连接运算符。

#### 38.3.4.1 HMAC 运算过程

- 1) 设置 HASH\_CR.MODE=2'b01，选择 HMAC 工作模式；
- 2) 如果使用的密钥长度超过 64 字节，则需要设置 HASH\_CR.LKEY=1（在此情况下，HMAC 将使用规范规定的密钥替代给定密钥）；
- 3) 将密钥写入 HASH\_DR 寄存器，直到所有的密钥分组运算完成；
- 4) 将消息写入 HASH\_DR 寄存器，直到所有的消息分组运算完成；
- 5) 读取 HASH\_HR 寄存器的最终散列运算结果。

HMAC 运算过程中的数据加载，可以选择软件操作流程和 DMA 操作流程的方式，具体举例如下：

#### 38.3.4.2 HMAC 软件操作流程

- 1) 设置模式 HASH\_CR.MODE=2'b01，HMAC 工作模式；设置 HASH\_CR.LKEY=1,选择长密钥方式；
- 2) HASH\_DR 中写入密钥分组数据；
- 3) 如果为第一组数据，则设置 HASH\_CR.FST\_GRP=1。设置 HASH\_CR.START=1 启动密钥运算；
- 4) 等待 HASH\_CR.BUSY=0 运算完成；
- 5) 如果不是最后一组数据，则重复 2)~4)；
- 6) 如果是最后一组数据，则设置 HASH\_CR.KMSG\_END=1，并设置 HASH\_CR.START=1 启动最

后一组密钥运算。等待 HASH\_CR.CYC\_END=1 后，密钥运算完成，并将该位清除。

- 7) HASH\_DR 中写入消息分组数据；
- 8) 如果为第一组消息数据，则设置 HASH\_CR.FST\_GRP=1。设置 HASH\_CR.START=1 启动消息运算；
- 9) 等待 HASH\_CR.BUSY=0 运算完成；
- 10) 如果不是最后一组消息数据，则重复操作 7)~9)；
- 11) 如果是最后一组消息，则设置 HASH\_CR.KMSG\_END=1，并设置 HASH\_CR.START=1 启动最后一组消息运算。等待 HASH\_CR.CYC\_END=1 后，所有 HMAC 运算完成，并将该位清除；
- 12) 读取 HASH\_HR 中的最终消息认证结果。

#### 38.3.4.3 HMAC 使用 DMA 操作流程

- 1) 设置 DMA 为数据块传送方式，数据块的大小为 16 个字，不管是密钥还是消息都需要按照 16 个字的大小分组，未满 16 个字时，请按照 HASH 消息填充规则进行填充，配置 DMA 数据块源地址为密钥地址；
- 2) 设置内部触发事件寄存器的 HASH 运算启动方式为 DMA 传送完成和数据块传送完成；设置内部触发事件寄存器的 DMA 传送开始方式为 HASH 运算完成（或者其他内部触发事件）；
- 3) 设置 HASH\_CR.MODE=2'b01，HMAC 运算模式；根据密钥长度配置 HASH\_CR.LKEY 位；设置 HASH\_CR.FST\_GRP=1，HASH\_CR.START=1 启动密钥运算；如果选择其他内部触发事件，则需要等待触发事件后开始启动运算；
- 4) 等待 HASH\_CR.CYC\_END=1 密钥运算完成后，将该位清除；
- 5) 停止 DMA，将 DMA 数据块源地址指向消息地址；
- 6) 设置 HASH\_CR.FST\_GRP=1，HASH\_CR.START=1 启动消息运算；或等待其他触发事件后启动消息运算；
- 7) 等待 HASH\_CR.CYC\_END=1 消息运算完成后，将该位清除；
- 8) 读取 HASH\_HR 寄存器中最终的消息认证结果。

#### 38.3.4.4 HMAC 运算举例

密钥和消息三组 HMAC 运算结果如下：

#### 38.3.4.5 注意事项

1. 当填充后的密钥和消息只有一组数据，即 512 位时，启动运算前请同时设置 HASH\_CR.FST\_GRT=1 和 HASH\_CR.KMSG\_END=1；
  2. 关于密钥填充，当密钥小于 64 字节时，未满 64 字节部分请填充 0x00；当密钥大于 64 字节时，未满 64 字节部分请按照 HASH 填充规则进行填充。

### 38.3.5 中断说明

HASH 中断标记 HASH\_INT。不管是在 HASH 运算模式还是在 HMAC 运算模式下，在设置 HASH\_CR.HEIE=1 中断许可条件下，每组数据运算完成后将产生一次中断请求。HASH\_CR.HCIE=1 中断许可条件下，所有的密钥或消息运算完成后将产生一次中断请求。HEIE 和 HCIE 同时许可时，任何一个条件满足都将产生中断请求。另外，HEIE 使能后，在模块复位后或者运算完成后，直接设置 HASH\_CR.FST\_GRP 且 HASH\_CR.START=1 时，也将产生一次中断用于数据加载。

### 38.3.6 硬件触发事件选择

通过配置内部硬件触发事件选择寄存器 HASH\_TRGSEL0，HASH\_TRGSEL1 在数据传送完成后启动 HASH 运算，通常与 DMA 配合使用。例如，当选择 DMA\_2 传送数据和启动 HASH 运算时，首先，配置 DMA2\_TRGSELx(x=0~7) 选择 HASH\_INT 启动 DMA 传送；其次，配置 HASH\_TRGSEL0 选择 DMA\_BTCx(x=0~7)，表示完成一个数据块(512bit)传输并启动 HASH 运算，配置 HASH\_TRGSEL1 选择 DMA\_TCx(x=0~7)，表示完成所有数据块传输。当启动 HASH 后，DMA 和 HASH 相互触发，直到所有数据块完成传送和运算。

### 38.3.7 寄存器说明

表 38-6 寄存器一览

BASE ADDR: 0x40008400

寄存器名	符号	偏移量	位宽	复位值
HASH控制寄存器	HASH_CR	0x0000	32	0x00000000
HASH摘要寄存器7	HASH_HR7	0x0010	32	0x00000000
HASH摘要寄存器6	HASH_HR6	0x0014	32	0x00000000
HASH摘要寄存器5	HASH_HR5	0x0018	32	0x00000000
HASH摘要寄存器4	HASH_HR4	0x001C	32	0x00000000
HASH摘要寄存器3	HASH_HR3	0x0020	32	0x00000000
HASH摘要寄存器2	HASH_HR2	0x0024	32	0x00000000
HASH摘要寄存器1	HASH_HR1	0x0028	32	0x00000000
HASH摘要寄存器0	HASH_HR0	0x002C	32	0x00000000
HASH数据寄存器15	HASH_DR15	0x0040	32	0x00000000
HASH数据寄存器14	HASH_DR14	0x0044	32	0x00000000
HASH数据寄存器13	HASH_DR13	0x0048	32	0x00000000
HASH数据寄存器12	HASH_DR12	0x004C	32	0x00000000
HASH数据寄存器11	HASH_DR11	0x0050	32	0x00000000
HASH数据寄存器10	HASH_DR10	0x0054	32	0x00000000
HASH数据寄存器9	HASH_DR9	0x0058	32	0x00000000
HASH数据寄存器8	HASH_DR8	0x005C	32	0x00000000
HASH数据寄存器7	HASH_DR7	0x0060	32	0x00000000
HASH数据寄存器6	HASH_DR6	0x0064	32	0x00000000
HASH数据寄存器5	HASH_DR5	0x0068	32	0x00000000
HASH数据寄存器4	HASH_DR4	0x006C	32	0x00000000
HASH数据寄存器3	HASH_DR3	0x0070	32	0x00000000
HASH数据寄存器2	HASH_DR2	0x0074	32	0x00000000
HASH数据寄存器1	HASH_DR1	0x0078	32	0x00000000
HASH数据寄存器0	HASH_DR0	0x007C	32	0x00000000

### 38.3.7.1 HASH 控制寄存器 (HASH\_CR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>HEIE</b> <b>HCIE</b> -   -   - <b>HMAC_END</b> <b>CYC_END</b> <b>BUSY</b> - <b>LKEY</b> <b>MODE[1:0]</b> - <b>KMSG_END</b> <b>FST_GRP</b> <b>START</b>															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为“0”，写入时请写“0”	R/W											
b15	HEIE	每组数据运算完成中断许可	0: 每组运算完成中断禁止 1: 每组运算完成中断许可	R/W											
b14	HCIE	全部密钥或消息运算完成中断许可	0: 全部密钥或消息运算完成中断禁止 1: 全部密钥或消息运算完成中断许可	R/W											
b14~b11	Reserved	-	读出时为“0”，写入时请写“0”	R/W											
b10	HMAC_END	HMAC运算完成标志	0: HMAC未运算完成 1: HMAC运算完成 当标志位为“1”后，写“0”清除。注意，软件在确认标志位后请及时清“0”，否则标志位将保持为“1”。	R/W											
b9	CYC_END	密钥或消息运算完成标志	0: 密钥或消息运算未完成 1: 密钥或消息运算完成 当标志位为“1”后，写“0”清除。注意，软件在确认标志位后请及时清“0”，否则标志位将保持为“1”。	R/W											
b8	BUSY	转换标志	0: HASH运算处于空闲状态 1: HASH运算处理中	R											
b7	Reserved	-	读出时为“0”，写入时请写“0”	R/W											
b6	LKEY	长密钥选择	0: 短密钥 (<=64字节) 1: 长密钥 (>64字节) 在 HMAC 模式中，该位在短密钥 (<=64 字节) 或长密钥 (> 64 字节) 之间进行选择。在 MODE=2'b01, HMAC 模式时有效。	R/W											
b5~b4	MODE	工作模式	00: SHA-256工作模式 01: HMAC工作模式 1X: 设定禁止	R/W											
b3	Reserved	-	读出时为“0”，写入时请写“0”	R/W											
b2	KMSG_END	密钥或消息分组的最后一组	0: 密钥或消息非最后一组 1: 密钥或消息最后一组 当最后一组密钥或消息转换完成后，该位自动清“0”	R/W											
b1	FST_GRP	密钥或消息分组的第一组	0: 密钥或消息非第一组 1: 密钥或消息第一组 当第一组密钥或消息转换完成后，该位自动清“0”	R/W											
b0	START	启动	0: 本模块运算结束或未被启动 1: 启动本模块进行运算	R/W											

**注意：**

- START 位的操作方法是：软件对该位写入 1 后，本模块将启动运行；本次运行结束后硬件会自动将该位清 0；软件查询到该位为 0 即表示本次运行完成。
- 对该寄存器的写入操作只能在本模块非处于运算状态时（即 START 位为 0 时）才能进行，否则硬件将自动忽略写操作。读操作则不受此限制。

### 38.3.7.2 HASH 摘要寄存器 (HASH\_HR)

位数 : 256 位

偏移地址 : 10'h010 - hash[255:224]

10'h014 - hash[223:192]

10'h018 - hash[191:160]

10'h01C - hash[159:128]

10'h020 - hash[127:96]

10'h024 - hash[95:64]

10'h028 - hash[63:32]

10'h02C - hash[31:0]

复位值 : 0x00000000 (每个 32 位寄存器)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
HASH[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HASH[15:0]															
位	标记	位名	功能	读写											
b31~b0	HASH[31:0]	摘要值	在模块运算完成后通过读取本寄存器得到消息摘要	R/W											

#### 注意：

- 本寄存器由 8 个 32 位寄存器拼接而成。访问时依次对 8 个 32 位寄存器进行操作，低地址对应的 32 位寄存器存放消息摘要的高字。
- 硬件将自动忽略对本寄存器的写操作。
- 对于本寄存器的读取只能在本模块非处于运算状态时 (HASH\_CR.START=0) 才能进行，否则对本寄存器的读取将得到全 0。

### 38.3.7.3 HASH 数据寄存器 (HASH\_DR)

位数 : 512 位

偏移地址 : 10'h040 - data[511: 480]

10'h044 - data[479: 448]

10'h048 - data[447: 416]

10'h04C - data[415: 384]

.....

10'h070 - data[127: 96]

10'h074 - data[95: 64]

10'h078 - data[63: 32]

10'h07C - data[31: 0]

复位值 : 0x00000000 (每个 32 位寄存器)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DATA[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DATA[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b0	DATA[31:0]	数据寄存器	用于在模块运算前写入消息	R/W											

#### 注意:

- 本寄存器由 16 个 32 位寄存器拼接而成。访问时依次对 16 个 32 位寄存器进行操作，低地址对应的 32 位寄存器存放数据的高字。
- 对于本寄存器的写入只能在本模块非处于运算状态时 (HASH\_CR.START) 才能进行，否则硬件将自动忽略对本寄存器的写操作。
- 对本寄存器的读取将总是得到全 0。

## 38.4 真随机数发生器 (TRNG)

### 38.4.1 模块框图

TRNG 模块提供 1 个真随机数发生器，产生 1 个 64 位随机数。

TRNG 的系统框图如下图 38-4 所示。其中随机数发生器是一个模拟随机数发振器电路，用于得到随机性的噪声；算法模块捕捉随机噪声并把结果保存至数据模块、并通过总线输出；控制模块控制 TRNG 的模式和启动。

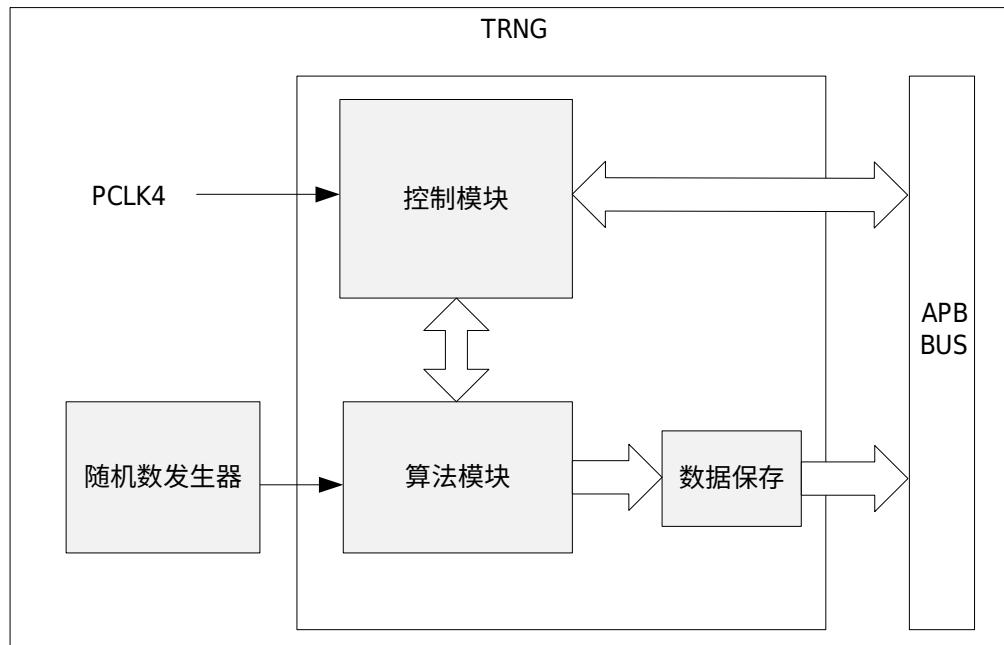


图 38-4 TRNG 系统框图

### 38.4.2 操作流程

真随机数生成流程如下：

1. 打开随机数发生器电路（设定 TRNG\_CR 的 EN 位为 1）。
2. 配置随机数生成模式（设定 TRNG\_MR）。
3. 启动随机数生成（设定 TRNG\_CR 的 RUN 位为 1）。
4. 读取随机数（读取 TRNG\_DR）。
5. 关闭随机数发生器电路（设定 TRNG\_CR 的 EN 位为 0）。

### 38.4.3 中断及事件输出

当随机数生成结束后，寄存器位 TRNG\_CR.RUN 硬件清零并产生随机数生成完了中断请求(TRNG\_END)。随机数生成结束同时也产生事件输出，可以触发其他模块联动。

### 38.4.4 寄存器说明

表 38-7 寄存器一览

BASE ADDR: 0x40042000

寄存器名	符号	偏移量	位宽	复位值
TRNG控制寄存器	TRNG_CR	0x0000	32	0x00000000
TRNG模式寄存器	TRNG_MR	0x0004	32	0x00000012
TRNG数据寄存器0	TRNG_DR0	0x000C	32	0x08000000
TRNG数据寄存器1	TRNG_DR1	0x0010	32	0x08000200

#### 38.4.4.1 TRNG 控制寄存器 (TRNG\_CR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	RUN	END

位	标记	位名	功能	读写
b31~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	RUN	随机数运算开始	0: 随机数运算停止 1: 随机数运算开始  软件写“1”, 产生新的64位随机数; 运行完毕后, 硬件清零。	R/W
b0	EN	模拟发振器使能	0: 关闭模拟随机数发生器电路 1: 打开模拟随机数发生器电路	R/W

**38.4.4.2 TRNG 模式寄存器 (TRNG\_MR)**

复位值：0x00000012

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	CNT[2:0]	-	LOAD		

位	标记	位名	功能	读写
b31~b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4~2	CNT[2:0]	移位次数控制位	捕捉随机噪声时, 移位次数控制位 011: 移位32次 100: 移位64次 101: 移位128次 110: 移位256次 000~010, 111: 功能保留位	R/W
b1	Reserved	-	读出时为“1”,写入时写“1”	R/W
b0	LOAD	装载控制位	随机数生成前, 数据寄存器是否从随机数发生器装载新的初始值 0: 不装载新的初始值 1: 装载新的初始值	R/W

**38.4.4.3 TRNG 数据寄存器 (TRNG\_DR)**

复位值：DR0: 0x08000000

DR1: 0x08000200

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DATA[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DATA[15:0]															

位	标记	位名	功能	读写
b31-b0	DATA[31:0]	随机数	64位随机数	R

## 39 CRC 运算 (CRC)

### 39.1 简介

在许多应用中，都需要 CRC 算法来校验数据的完整性和正确性。尤其是在数据传输中，CRC 校验更是被广泛应用。本模块可采用 CRC16 和 CRC32 两种算法对数据进行运算和校验。

### 39.2 功能框图

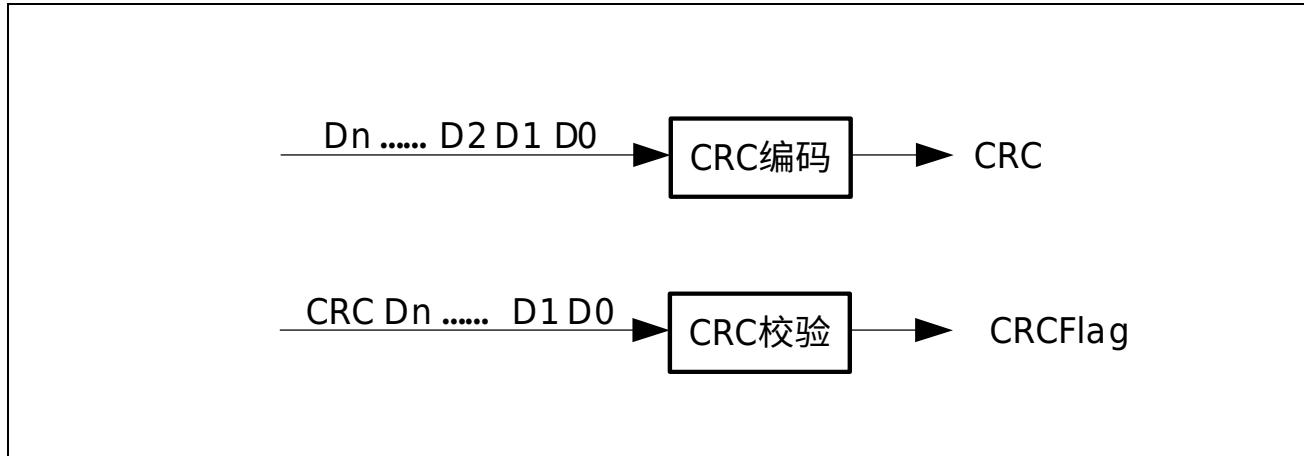


图 39-1 CRC 应用示意图

### 39.3 功能说明

本模块 CRC 算法遵从 ISO/IEC13239 的定义，分别采用 32 位和 16 位的 CRC。CRC32 的生成多项式为  $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$ ，32 位初值为 "0xFFFFFFFF"。CRC16 的生成多项式为  $X^{16}+X^{12}+X^5+1$ ，16 位初值为 "0xFFFF"。

本模块功能包括：

- CRC 编码和 CRC 校验；
- 3 种位宽访问方式 8 位，16 位，32 位：
  - 8 位位宽下输入数据示例为 0x00,0x11,0x22,0x33,0x44,0x55,0x66,0x77;
  - 16 位位宽下输入数据示例为 0x1100,0x3322,0x5544,0x7766;
  - 32 位位宽下输入数据示例为 0x33221100,0x77665544;

### 39.3.1 CRC16 编码模式

CRC 编码就是对原始数据编码以计算其 CRC 值，操作流程如下：

1. 向 CRC\_CR.CR 写入 1'b0,选择 CRC16。
2. 向 CRC\_RESLT[15:0]写入 0xFFFF，初始化 CRC 计算。
3. 将待编码的原始数据按 8 位/16 位/32 位组织方式，依次写入 CRC\_DAT 寄存器。  
**注：**按 8 位组织方式写 CRC 值，应先写低位，后写高位。
4. 读取 CRC\_RESLT[15:0]，即可获得 16 位 CRC 编码值。

### 39.3.2 CRC16 校验模式

校验模式可以校验已编码的数据是否被篡改，操作流程如下所示：

1. 向 CRC\_CR.CR 写入 1'b0,选择 CRC16。
2. 向 CRC\_RESLT[15:0]写入 0xFFFF，初始化 CRC 计算。
3. 将已编码的数据按 8 位/16 位/32 位组织方式，依次写入 CRC\_DAT 寄存器。  
**注：**按 8 位组织方式写 CRC 值，应先写低位，后写高位。
4. 校验码以 16 位位宽将数据写入 CRC\_DAT 寄存器。
5. 读取 CRC\_CR.FLG 寄存器，为 1 则表示校验成功，为 0 则表示校验失败。

### 39.3.3 CRC32 编码模式

CRC 编码就是对原始数据编码以计算其 CRC 值，操作流程如下：

1. 向 CRC\_CR.CR 写入 1'b1,选择 CRC32。
2. 向 CRC\_RESLT[31:0]写入 0xFFFF\_FFFF，初始化 CRC 计算。
3. 将待编码的原始数据按 8 位/16 位/32 位组织方式，依次写入 CRC\_DAT 寄存器。  
**注：**按 8 位组织方式写 CRC 值，应先写低位，后写高位。
4. 读取 CRC\_RESLT[31:0]，即可获得 32 位 CRC 编码值。

### 39.3.4 CRC32 校验模式

校验模式可以校验已编码的数据是否被篡改，操作流程如下所示：

1. 向 CRC\_CR.CR 写入 1'b1,选择 CRC32。
2. 向 CRC\_RESLT[31:0]写入 0xFFFF\_FFFF，初始化 CRC 计算。
3. 将已编码的数据按 8 位/16 位/32 位组织方式，依次写入 CRC\_DAT 寄存器。  
**注：**按 8 位组织方式写 CRC 值，应先写低位，后写高位。
4. 校验码按 8 位/16 位/32 位组织方式，写入 CRC\_DAT 寄存器。
5. 读取 CRC\_CR.FLG 寄存器，为 1 则表示校验成功，为 0 则表示校验失败。

## 39.4 寄存器说明

表 39-1 所示，为 CRC 模块的寄存器列表。

表 39-1 CRC 寄存器列表

CRC\_BASE\_ADDR: 0x40008C00

寄存器名	符号	偏移地址	位宽	复位值
CRC控制寄存器	CRC_CR	0x00	32	0x0000_0001
CRC结果寄存器	CRC_RESLT	0x04	32	0x0000_0000
CRC数据寄存器	CRC_DAT	0x80~0xFF	32	0x0000_0000

### 39.4.1 控制寄存器 (CRC\_CR)

复位值: 0x00000001

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	FLAG	CR

位	标记	位名	功能	读写
b30~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	FLAG	校验结果位	0: 当前校验错误 1: 当前校验正确	R
b0	CR	运算控制位	0: CRC16 1: CRC32	R/W

### 39.4.2 结果寄存器 (CRC\_RESLT)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RESULT[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESULT[15:0]															

位	标记	位名	功能	读写
b31~b0	RESULT[31:0]	结果位	选择CRC16时, 取RESULT[15:0]; 选择CRC32时, 取RESULT[31:0];	R/W

### 39.4.3 数据寄存器 (CRC\_DAT)

复位值: 0x00000000

b31 b30 b29 b28 b27 b26 b25 b24 b23 b22 b21 b20 b19 b18 b17 b16

CRC\_DAT[31:16]

b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0

CRC\_DAT[15:0]

位	标记	位名	功能	读写
b31~b0	CRC_DAT[31:0]	数据寄存器	本寄存器用于输入需要运算的数据；本寄存器的地址是一个范围(0x80~0xFF)，对该范围内任何一个地址进行操作，都会认为对本寄存器进行操作。该寄存器读出为全0。	R/W

## 40 滤波数学加速器 (FIR)

### 40.1 简介

滤波数学加速器 (FIR) 是一个 FIR 滤波计算的硬件加速模块。该模块可进行最大 16 阶，且阶数可配置的 FIR 数字滤波。内置 16x16 bit 乘法器、32+5bit 加法器，用户可自定义输出数据精度。本系列产品搭载 4 个 FIR 模块。

### 40.2 基本框图

该模块的基本特性如下：

- 最大 16 阶 FIR 滤波，阶数可配置
- 滤波系数可配置
- 16 位有符号数输入、输出数据，16 位有符号数滤波系数
- 16x16 bit 乘法器
- 37bit 加法器
- 提供中断信号和状态查询
- AHB slave 接口配置参数和输入输出数据

其基本框图如下：

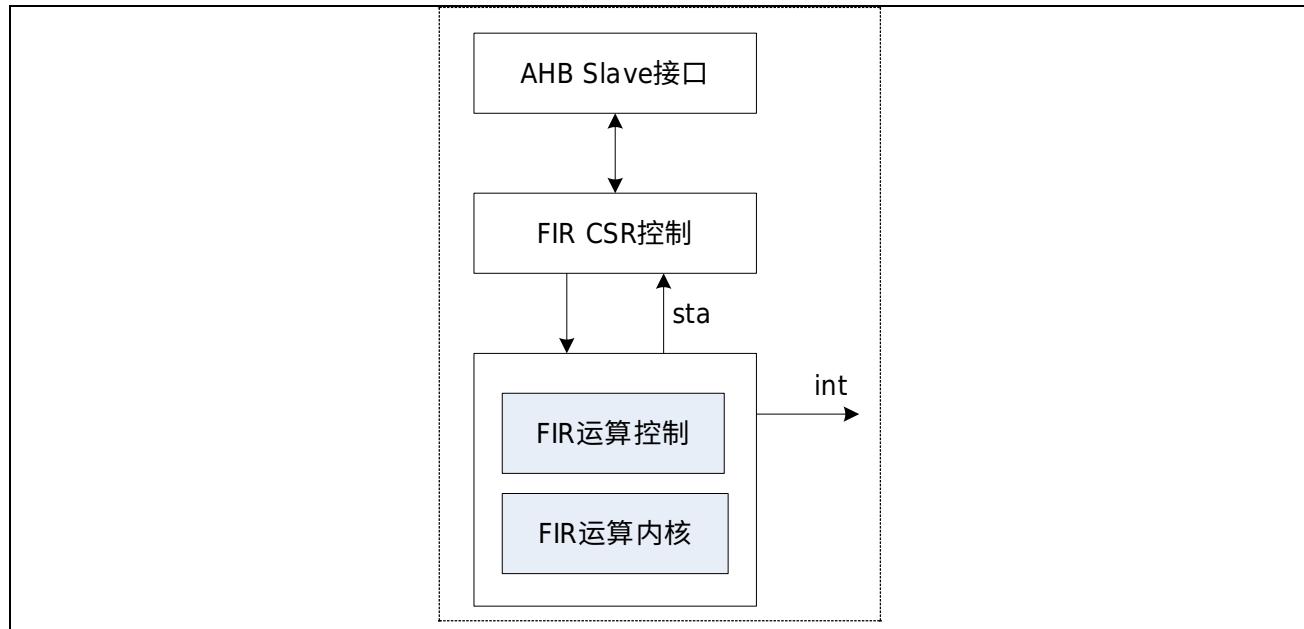


图 40-1 FIR 基本框图

在选择 16 阶 FIR 滤波时，FIR 运算控制和运算内核需要 17 个时钟周期处理一个数据，在 150MHz 的工作时钟频率下，可以处理最大 8.8MHz 的输入数据流。不同的阶数，处理所需的时间不同。阶数越小，处理速度越快。

## 40.3 操作流程

本模块需要按照以下步骤来使用：

- 1) 将使能位 (FMAC\_ENR.FIREN) 置为 1;
- 2) 配置滤波器阶数 (FMAC\_CTR.STAGE\_NUM[4:0]) 及输出有效结果移位数 (FMAC\_CTR.SHIFT[4:0]);
- 3) 如果需要使用中断，则配置中断使能位 (FMAC\_IER.INTEN) 为 1;
- 4) 写入滤波器系数 (有 N 阶，就只需配置 FMAC\_COR0~FMAC\_CORN);
- 5) 写入待处理数据 (FMAC\_DTR);
- 6) 等待完成中断或者查询完成状态 (FMAC\_STR.READY) 为 1;
- 7) 读取结果 (FAMC\_RTR0 或 FAMC\_RTR1);
- 8) 重复 5) ~7)，直到所有数据处理完。

## 40.4 模块使能

模块使能位 (FMAC\_ENR.FIREN) 的作用有两个，作用一是使能整个模块，只有其为 1 时，模块才允许工作；作用二是模块的软复位作用，当某种阶数和系数的配置使用完以后，需要换另一种阶数和系数的配置，那么需要将 FIREN 置为 0，将前面计算残留的信息清零。再按照操作流程来操作。

FIREN 清 0 时，不会将配置参数清零，只会将内部寄存器和中间结果清零。

## 40.5 系数归一化

如果输出结果只取 16 位，那么在计算过程中以及计算结果，总共截去的位数多达 20 位，所以如果输入的参与计算的数据太小，则输出结果有可能为 0，或者很小的数（损失了很大的精度）。所以滤波器系数有必要做归一化，即将其中的最大值（如果是正数）归一化成 32767（如果是负数，归一化成 -32768），其他系数乘以相同的比例。

例如，有 5 个系数为 10, 13, 35, 96, 42，要将其归一化。首先找出其中的最大值，为 96，然后用 32767 除以该最大值， $32767/96=341.3$ ，取值 341，即为归一化的倍数，其他系数都要乘以该倍数，即为归一化以后的数。

所以归一化以后的系数为：3410, 4433, 11935, 32736, 14322。

## 40.6 中断及事件说明

### 40.6.1 中断输出

在每一次运算完成之后，都会有相应的运算完成事件产生。若使能中断控制位 (FMAC\_IER.INTEN=1)，则对应的中断请求信号 (FMAC\_m\_FIR, m=1~4) 也会产生。

### 40.6.2 事件输出

在每一次运算完成之后，都会有相应的运算完成事件产生，相应的事件请求可以用于选择触发其它模块。

## 40.7 寄存器说明

表 40-1 所示，为 FMAC 模块的寄存器列表。

表 40-1 FMAC 寄存器列表

BASE ADDR: 0x40058000 (U1)、0x40058400 (U2)、

0x40058800 (U3)、0x40058C00 (U4)

寄存器名	符号	偏移量	位宽	复位值
模块使能寄存器	FMAC_ENR	0x0000	32	0x00000000
基本控制寄存器	FMAC_CTR	0x0004	32	0x00000010
中断控制寄存器	FMAC_IER	0x0008	32	0x00000000
数据输入寄存器	FMAC_DTR	0x000C	32	0x00000000
滤波系数寄存器0	FMAC_COR0	0x0020	32	0x00000000
滤波系数寄存器1	FMAC_COR1	0x0024	32	0x00000000
...	...	...	...	...
滤波系数寄存器16	FMAC_COR16	0x0060	32	0x00000000
结果输出寄存器0	FMAC_RTR0	0x0010	32	0x00000000
结果输出寄存器1	FMAC_RTR1	0x0014	32	0x00000000
运算状态寄存器1	FMAC_STR	0x0018	32	0x00000000

### 40.7.1 模块使能寄存器 (FMAC\_ENR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16														
Reserved																													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0														
Reserved															FMACEN														
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>位</th> <th>标记</th> <th>位名</th> <th>功能</th> <th>读写</th> </tr> </thead> <tbody> <tr> <td>b31~b1</td> <td>Reserved</td> <td>-</td> <td>读出时为“0”,写入时写“0”</td> <td>R/W</td> </tr> <tr> <td>b0</td> <td>FMACEN</td> <td>模块使能</td> <td>0: 不使能 1: 使能, 即允许工作  注: 该信号作为本模块的软复位信号使用, 即当需要改变滤波阶数和滤波系数的时候, 需要先将该位信号置为0, 再将该位置为1</td> <td>R/W</td> </tr> </tbody> </table>															位	标记	位名	功能	读写	b31~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W	b0	FMACEN	模块使能	0: 不使能 1: 使能, 即允许工作  注: 该信号作为本模块的软复位信号使用, 即当需要改变滤波阶数和滤波系数的时候, 需要先将该位信号置为0, 再将该位置为1	R/W
位	标记	位名	功能	读写																									
b31~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W																									
b0	FMACEN	模块使能	0: 不使能 1: 使能, 即允许工作  注: 该信号作为本模块的软复位信号使用, 即当需要改变滤波阶数和滤波系数的时候, 需要先将该位信号置为0, 再将该位置为1	R/W																									

## 40.7.2 基本控制寄存器 (FMAC\_CTR)

复位值: 0x000000010

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-		SHIFT[4:0]			-	-	-	-		STAGE_NUM[4:0]			
<hr/>															
位	标记	位名	功能	读写											
b31~b13	Reserved	-	读出时为“0”,写入时写“0”	R/W											
滤波结果右移位数, 控制FMAC_RTR1输出的结果															
00000: 不移位, 即选取运算结果的[31:0]给FMAC_RTR1															
00001: 右移1位, 即选取运算结果的[32:1]给FMAC_RTR1															
00010: 右移2位, 即选取运算结果的[33:2]给FMAC_RTR1															
b12~b8	SHIFT[4:0]	滤波结果移位	..... 10100: 右移20位, 即选取运算结果的[35:20]给FMAC_RTR1, 高位移入符号位 10101: 右移21位, 即选取运算结果的[36:21]给FMAC_RTR1, 高位移入符号位 其他值: 不移位	R/W											
b7~b5	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b4~b0	STAGE_NUM[4:0]	滤波阶数设定	设定滤波器的滤波阶数	R/W											

## 40.7.3 中断控制寄存器 (FMAC\_IER)

复位值: 0x000000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
<hr/>															
位	标记	位名	功能	读写											
b31~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b0	INTEN	中断使能	0: 运算完成后不产生中断 1: 运算完成后产生中断	R/W											

#### 40.7.4 数据输入寄存器 (FMAC\_DTR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FMAC_DIN[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b15~b0	FMAC_DIN[15:0]	数据输入	FIR数据输入	R/W											

#### 40.7.5 滤波系数寄存器 (FMAC\_CRO~16)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FMAC_CIN[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b15~b0	FMAC_CIN[15:0]	滤波系数	滤波系数输入	R/W											

#### 40.7.6 结果输出寄存器 0 (FMAC\_RTO)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
FMAC_DOUT0[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FMAC_DOUT0[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b0	FMAC_DOUT0[31:0]	结果输出	累加器结果的高32位结果输出	R/W											

#### 40.7.7 结果输出寄存器 1 (FMAC\_RTR1)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
FMAC_DOUT1[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FMAC_DOUT1[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b0	FMAC_DOUT1[31:0]	结果输出	累加器结果的低32位结果输出	R/W											

#### 40.7.8 运算状态寄存器 (FMAC\_STR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
READY	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
<hr/>															
位	标记	位名	功能	读写											
b31	READY	运算完成位	0: 计算未完成, 结果不可用 1: 计算完成, 结果可用 注: 该寄存器读之后自动清零	R/W											
b30~b0	Reserved	-	读出时为“0”, 写入时写“0”	R/W											

## 41 数据计算单元 (DCU)

### 41.1 概要

数据计算单元(Data Computing Unit)是一个不借助于CPU的简单处理数据的模块。每个DCU单元具有3个数据寄存器，能够进行2个数据的加减和比较大小以及窗口比较功能。本产品搭载3个DCU单元，每个单元均可独立完成自身功能。

#### 功能概要：

- 可进行4种数据处理：2个数据的加法，减法，比较以及3个数据窗口比较
- 加法和减法对DATA0和DATA1寄存器中的数据进行运算，结果存放在DATA0中
- 加法和减法可以选择写寄存器后计算或由其他外围电路事件触发进行运算
- 加法和减法运算可自动将结果进行一次减半操作，并将减半的结果和加减运算的结果分别放入2个数据寄存器内供其他模块使用
- 比较模式可以对DATA0和DATA1寄存器之间，以及DATA0和DATA2寄存器之间的2个数据进行比较，可以分别选择当满足大于，小于，等于时产生中断和标志位
- 比较模式可以用来进行窗口比较，即分别设置DATA1和DATA2为窗口的上下限，根据DATA0和DATA1及DATA0和DATA2的比较结果判断DATA0在窗口内或窗口外
- 能够由其他外围电路事件触发进行运算，并根据运算结果产生多种中断和事件信号。当其他具有硬件触发启动功能的外围电路选择DCU作为触发源时，由DCU产生事件信号来启动外围电路开始动作

## 41.2 功能描述

### 41.2.1 加法模式

加法模式计算 DATA0 和 DATA1 的和，其中 DATA0 作为被加数，DATA1 作为加数。每次写 DATA1 寄存器进行一次 $(\text{DATA0} + \text{DATA1})/2$  的运算，DATA0+DATA1 的结果存入 DATA0，而 $(\text{DATA0} + \text{DATA1})/2$  的结果存入 DATA2。当 DATA0 + DATA1 结果超过 0xFF(8bit 模式)或 0xFFFF(16bit 模式)或 0xFFFFFFFF(32bit 模式)时产生标志位并产生中断。

加法模式应用举例：

1. 控制寄存器 DCU\_CTL 选择加法模式，数据宽度为 16bit
2. 中断条件选择寄存器 DCU\_INTSEL 选择运算条件
3. 分别在 DATA0 和 DATA1 中写入 0xFF00 和 0x55，此时计算结果为 0xFF55，结果保存在 DATA0 中
4. DATA1 中继续写入 0xFF，此时计算结果溢出，产生结果标志。读取标志寄存器 DCU\_FLAG 的得到结果标志
5. 写标志复位寄存器 DCU\_FLAGCLR 清除标志位

### 41.2.2 减法模式

减法模式计算 DATA0 和 DATA1 的差，其中 DATA0 作为被减数，DATA1 作为减数。每次写 DATA1 寄存器进行一次 $(\text{DATA0} - \text{DATA1})/2$  的运算，DATA0-DATA1 的结果存入 DATA0，而 $(\text{DATA0}-\text{DATA1})/2$  的结果存入 DATA2。当 DATA0-DATA1 结果小于 0x0(8bit, 16bit, 32bit 模式)时产生标志位并产生中断。

### 41.2.3 硬件触发启动模式

DCU 能够根据外围电路产生的事件触发启动运算。使用硬件触发启动模式时，需要先将功能时钟控制 0 寄存器(FCG0)的外围电路触发功能使能位置为有效。每个 DCU 单元可独立选择由其它外围电路送来的触发启动信号，选择启动信号时，在触发源选择寄存器(DCU\_TRGSEL)中写入要选择的外围电路启动源的编号。当该外围电路事件发生时，会将事件信号输入 DCU 并触发 DCU 启动开始运算。硬件触发启动模式包括触发加模式和触发减模式。在触发加模式时，每发生一次事件触发，DCU 将启动并进行一次 $(\text{DATA0} + \text{DATA1})/2$  的运算，DATA0+DATA1 的结果存入 DATA0，而 $(\text{DATA0} + \text{DATA1})/2$  的结果存入 DATA2。当 DATA0+DATA1 的结果超过 0xFF(8bit 模式)或 0xFFFF(16bit 模式)或 0xFFFFFFFF(32bit 模式)时产生标志位并产生中断。在触发减模式时，每发生一次事件触发，DCU 将启动并进行一次 $(\text{DATA0} - \text{DATA1})/2$  的运算，DATA0-DATA1 的结果存入 DATA0，而 $(\text{DATA0}-\text{DATA1})/2$  的结果存入 DATA2。当 DATA0-DATA1 的结果小于 0x0(8bit, 16bit, 32bit 模式)时产生标志位并产生中断。

硬件触发启动模式应用举例：

1. 控制寄存器 DCU\_CTL 选择触发加模式，数据宽度为 8bit

2. 中断条件选择寄存器 DCU\_INTSEL 选择运算条件
3. 分别在 DATA0 和 DATA1 中写入 0x00 和 0x56
4. 在触发源选择寄存器 DCU\_TRGSEL 中写入事件编号
5. 使选中的外围电路动作并产生事件, DCU 由该事件触发并进行一次加法运算, 结果为 0x56 并保存在 DATA0 中
6. 在连续 3 次触发后, 计算结果溢出, 产生结果标志。读取标志寄存器 DCU\_FLAG 得到结果标志
7. 写标志复位寄存器 DCU\_FLAGCLR 清除标志位

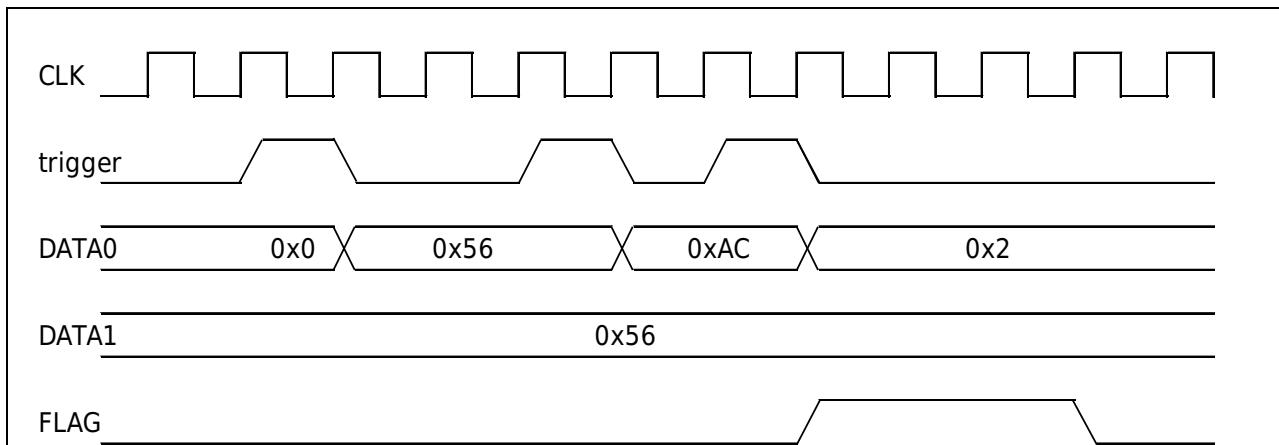


图 41-1 DCU 硬件触发启动

#### 41.2.4 比较模式

比较 DATA0 和 DATA1 以及 DATA0 和 DATA2 的大小, 可选择当 DATA0 大于 DATA1, DATA0 小于 DATA1, DATA0 等于 DATA1 时以及当 DATA0 大于 DATA2, DATA0 小于 DATA2, DATA0 等于 DATA2 时产生标志位并产生中断。比较模式下可以选择数据开始比较的条件, 写 DATA0 后比较或写任何数据寄存器后比较。

比较模式应用举例:

1. 控制寄存器 DCU\_CTL 选择比较模式, 数据宽度为 16bit。在写入 DATA0 后开始比较。
2. 中断条件选择寄存器 DCU\_INTSEL 选择 DATA0>DATA1 时产生标志。
3. 在 DATA1 和 DATA0 中分别写入 0xBBB 和 0xAAA, 此时由于不满足 DATA0>DATA1 因此不产生标志。
4. 在 DATA1 写入 0x8888, 此时虽然满足 DATA0>DATA1, 但由于设置为写入 DATA0 后开始比较, 因此也不产生标志。
5. DATA0 写入 0x9999 后满足标志产生条件, 产生标志。
6. 写标志复位寄存器 DCU\_FLAGCLR 清除标志位。

#### 41.2.5 中断和事件信号输出

DCU 具有多种中断和用于触发启动其他外围电路的事件输出供用户选择。中断和事件输出的控制由中断和事件寄存器(DCU\_INTEVT)控制，当需要输出事件信号时，用户需要将中断和事件寄存器(DCU\_INTEVT)的相应控制位置为有效。每个 DCU 单元各输出一个 DCU 事件信号，在事件列表中分别为 DCU1~DCU3。当需要对应事件在发生时产生中断，用户需要将中断和事件寄存器(DCU\_INTEVT)的相应控制位置为有效的同时，将控制寄存器(DCU\_CTL)的 INTEN 位置 1。每个 DCU 单元各输出一个 DCU 中断信号，在中断列表中分别为 DCU1~DCU3。

## 41.3 寄存器说明

表 41-1 寄存器一览

单元 1

名称	英文缩写	说明	地址
DCU1 控制寄存器	DCU1_CTL	配置 DCU 的动作模式	0x40056000
DCU1 标志寄存器	DCU1_FLAG	DCU 的结果标识	0x40056004
DCU1 数据寄存器 0	DCU1_DATA0	存放运算数据	0x40056008
DCU1 数据寄存器 1	DCU1_DATA1	存放运算数据	0x4005600C
DCU1 数据寄存器 2	DCU1_DATA2	存放运算数据	0x40056010
DCU1 标志复位寄存器	DCU1_FLAGCLR	清除 DCU 的结果标识	0x40056014
DCU1 中断和事件寄存器	DCU1_INTEVT	DCU 中断和事件的条件	0x40056018

单元 2

名称	英文缩写	说明	地址
DCU2 控制寄存器	DCU2_CTL	配置 DCU 的动作模式	0x40056400
DCU2 标志寄存器	DCU2_FLAG	DCU 的结果标识	0x40056404
DCU2 数据寄存器 0	DCU2_DATA0	存放运算数据	0x40056408
DCU2 数据寄存器 1	DCU2_DATA1	存放运算数据	0x4005640C
DCU2 数据寄存器 2	DCU2_DATA2	存放运算数据	0x40056410
DCU2 标志复位寄存器	DCU2_FLAGCLR	清除 DCU 的结果标识	0x40056414
DCU2 中断和事件寄存器	DCU2_INTEVT	DCU 中断和事件的条件	0x40056418

单元 3

名称	英文缩写	说明	地址
DCU3 控制寄存器	DCU3_CTL	配置 DCU 的动作模式	0x40056800
DCU3 标志寄存器	DCU3_FLAG	DCU 的结果标识	0x40056804
DCU3 数据寄存器 0	DCU3_DATA0	存放运算数据	0x40056808
DCU3 数据寄存器 1	DCU3_DATA1	存放运算数据	0x4005680C
DCU3 数据寄存器 2	DCU3_DATA2	存放运算数据	0x40056810
DCU3 标志复位寄存器	DCU3_FLAGCLR	清除 DCU 的结果标识	0x40056814
DCU3 中断和事件寄存器	DCU3_INTEVT	DCU 中断和事件的条件	0x40056818

### 41.3.1 DCU 控制寄存器(DCUx\_CTL)(x=1~3)

寄存器说明：该寄存器用来配置 DCU 的动作模式

复位值：0x80000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
INTEN	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					COMPTRG	Reserved	DATASIZE[1:0]	MODE[3:0]							

位	标记	位名	功能	读写
b31	INTEN	中断使能	0: 不允许产生中断 1: 允许产生中断	R/W
b30~b9	Reserved	-	读时读出0, 写时请写0	R/W
b8	COMPTRG	比较模式触发 比较的时机	0: 写入DATA0后比较 1: 写入DATA0或DATA1或DATA2后比较	R/W
b7~b6	Reserved	-	读时读出0, 写时请写0	R/W
b5~b4	DATASIZE[1:0]	加/减法模式，比 较模式数据大小	00: 8bit 01: 16bit 10: 32bit  0000: DCU无效 0001: 加法模式，运算在DATA1寄存器写入数据后进行 0010: 减法模式，运算在DATA1寄存器写入数据后进行 0011: 硬件触发加法模式，由其他外围电路触发启动进行加法运算 0100: 硬件触发减法模式，由其他外围电路触发启动进行减法运算 0101: 比较模式 其它: 设定禁止	R/W
b3~b0	MODE[2:0]	动作模式	0011: 硬件触发加法模式，由其他外围电路触发启动进行加法运算 0100: 硬件触发减法模式，由其他外围电路触发启动进行减法运算 0101: 比较模式 其它: 设定禁止	R/W

### 41.3.2 DCU 标志寄存器(DCUx\_FLAG)(x=1~3)

寄存器说明：该寄存器产生 DCU 的结果标识

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															

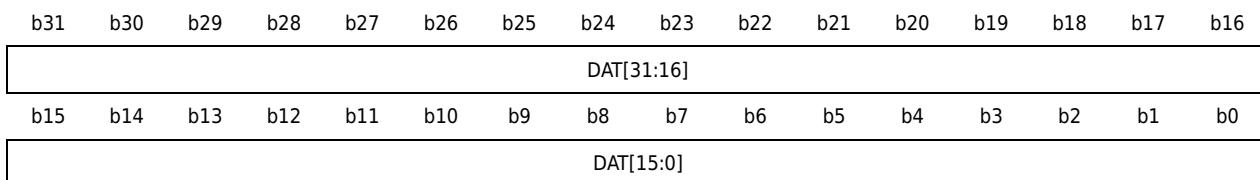
位	标记	位名	功能	读写
b31~b7	Reserved	-	读时读出0，写时请写0	R/W
b6	FLAG_GT1	大于标志位1	比较模式下，当 DATA0 > DATA1 时置位，DCU 标志复位寄存器 DCU_FLAGCLR 的 CLR_GT1 位写1时清除	R
b5	FLAG_EQ1	等于标志位1	比较模式下，当 DATA0 = DATA1 时置位，DCU 标志复位寄存器 DCU_FLAGCLR 的 CLR_EQ1 位写1时清除	R
b4	FLAG_LS1	小于标志位1	比较模式下，当 DATA0 < DATA1 时置位，DCU 标志复位寄存器 DCU_FLAGCLR 的 CLR_LS1 位写1时清除	R
b3	FLAG_GT2	大于标志位2	比较模式下，当 DATA0 > DATA2 时置位，DCU 标志复位寄存器 DCU_FLAGCLR 的 CLR_GT2 位写1时清除	R
b2	FLAG_EQ2	等于标志位2	比较模式下，当 DATA0 = DATA2 时置位，DCU 标志复位寄存器 DCU_FLAGCLR 的 CLR_EQ2 位写1时清除	R
b1	FLAG_LS2	小于标志位2	比较模式下，当 DATA0 < DATA2 时置位，DCU 标志复位寄存器 DCU_FLAGCLR 的 CLR_LS2 位写1时清除	R
b0	FLAG_OP	运算标志位	加法，减法和触发加，触发减模式时，当加法产生上溢或减法产生下溢时置位，DCU 标志复位寄存器 DCU_FLAGCLR 的 CLR_OP 位写1时清除	R

### 41.3.3 DCU 数据寄存器(DCUx\_DATAy)(x=1~3,y=0,1,2)

各个数据寄存器在各模式下的功能如下

	<b>DATA0</b>	<b>DATA1</b>	<b>DATA2</b>
加法模式	被加数/存放结果	加数	存放减半结果
触发加模式	被加数/存放结果	加数	存放减半结果
减法模式	被减数/存放结果	减数	存放减半结果
触发减模式	被减数/存放结果	减数	存放减半结果
比较模式	被比较对象	比较对象 1	比较对象 2
比较模式(窗口比较)	被比较对象	窗口上限	窗口下限

复位值：0x00000000



加/减法模式，触发加/减法模式，比较模式下 DCUx\_DATAy (x=1~3,y=0,1,2)用来存放运算数据。

位	标记	位名	功能	读写
b31~b0	DAT[31:0]	运算数据	存放运算数据，实际使用的位数根据DCU_CTL.DATASIZE设置， 当DCU_CTL.DATASIZE=00时DATA[7:0]为有效数据， 当DCU_CTL.DATASIZE=01时DATA[15:0]为有效数据， 当DCU_CTL.DATASIZE=10时DATA[31:0]为有效数据	R/W

#### 41.3.4 DCU 标志复位寄存器(DCUx\_FLAGCLR) (x=1~3)

寄存器说明：该寄存器用于清除 DCU 的结果标识

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															

位	标记	位名	功能	读写
b31~b7	Reserved	-	读时读出0，写时请写0	R/W
b6	CLR_GT1	清除大于标志位1	写1时清除DCU_FLAG的FLAG_GT1位	W
b5	CLR_EQ1	清除等于标志位1	写1时清除DCU_FLAG的FLAG_EQ1位	W
b4	CLR_LS1	清除小于标志位1	写1时清除DCU_FLAG的FLAG_LS1位	W
b3	CLR_GT2	清除大于标志位2	写1时清除DCU_FLAG的FLAG_GT2位	W
b2	CLR_EQ2	清除等于标志位2	写1时清除DCU_FLAG的FLAG_EQ2位	W
b1	CLR_LS2	清除小于标志位2	写1时清除DCU_FLAG的FLAG_LS2位	W
b0	CLR_OP	清除运算标志位	写1时清除DCU_FLAG的FLAG_OP位	W

### 41.3.5 DCU 中断和事件寄存器(DCUx\_INTEVTSEL) (x=1~3)

寄存器说明：该寄存器能够选择 DCU 在何种条件下产生中断和输出事件信号

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								SEL_WIN[1:0]	SEL_GT1	SEL_EQ1	SEL_LS1	SEL_GT2	SEL_EQ2	SEL_LS2	SEL_OP

位	标记	位名	功能	读写
b31~b9	Reserved	-	读时读出0，写时请写0	R/W
b8~b7	SEL_WIN[1:0]	窗口比较条件选择	比较模式下满足INT_WIN设定的窗口比较条件时产生中断和输出事件信号，在INT_WIN设定有效时，满足其他比较条件时不会产生中断和输出事件信号 00：不产生窗口比较中断和输出事件信号，此设定下由该寄存器的b1~b6选择其他中断和事件信号产生条件 01：当DATA0数据在窗口内时产生中断和输出事件信号，即DATA2≤DATA0≤DATA1 10：当DATA0数据在窗口外时产生中断和输出事件信号，即DATA0>DATA1或DATA0<DATA2 11：比较模式下不产生任何中断和事件信号	R/W
b6	SEL_GT1	大于条件选择1	0：比较模式下并且INT_WIN=00时，当DATA0>DATA1时不产生中断和输出事件信号 INT_WIN≠00时该位无效	R/W
b5	SEL_EQ1	等于条件选择1	0：比较模式下并且INT_WIN=00时，当DATA0=DATA1时不产生中断和输出事件信号 INT_WIN≠00时该位无效	R/W
b4	SEL_LS1	小于条件选择1	0：比较模式下并且INT_WIN=00时，当DATA0<DATA1时不产生中断和输出事件信号 INT_WIN≠00时该位无效	R/W
b3	SEL_GT2	大于条件选择2	0：比较模式下并且INT_WIN=00时，当DATA0>DATA2时不产生中断和输出事件信号 INT_WIN≠00时该位无效	R/W
b2	SEL_EQ2	等于条件选择2	0：比较模式下并且INT_WIN=00时，当DATA0=DATA2时不产生中断和输出事件信号 INT_WIN≠00时该位无效	R/W

---

			1: 比较模式下并且INT_WIN=00时, 当DATA0<DATA2时产生中断和输出事件信号 0: 比较模式下并且INT_WIN=00时, 当DATA0<DATA2时不产生中断 R/W 和输出事件信号 INT_WIN≠00时该位无效
b1	SEL_LS2	小于条件选择2	1: 加法和减法模式下当运算结果上溢或下溢时产生中断和输出事件信号 0: 加法和减法模式下当运算结果上溢或下溢时不产生中断和输出事件信号 R/W

---

## 42 数学运算单元 (MAU)

### 42.1 简介

数学运算单元 (MAU) 是一个内含开方运算和正弦运算两种运算类型的硬件加速运算模块，支持定点数的开方和正弦运算。正弦函数支持  $360^\circ/2^{12}$  运算精度。

### 42.2 功能说明

#### 42.2.1 开方运算

开方运算支持 32 位定点数输入，运算结果为 17 位定点数输出。根据输入数据的大小不同，运算执行周期会不同。

##### 42.2.1.1 操作流程

中断无效 (MAU\_CSR.INTEN=0) 时，开方运算的操作流程如下；中断有效时，用户根据该流程自定义加入中断处理过程：

- 1) 在数据输入寄存器 0 (MAU\_DTR0) 中写入被开方数据
- 2) 将控制状态寄存器的启动位置 1 (MAU\_CSR.START=1)
- 3) 运算开始后，控制状态寄存器的状态位 (MAU\_CSR.BUSY) 会置位，直到运算完成，该位自动清零。期待 BUSY 位清零，表明运算完成
- 4) 从结果输出寄存器 0 (MAU\_RTR0) 中读取运算结果
- 5) 重复以上流程 1) ~4)

关于流程 3)，从 MAU\_CSR.START 置位后，BUSY 位开始变高，在持续 8~16 个运算时钟周期后（根据输入数据的大小运算时间不等）自动变为 0。用户在使用开方运算模块时可根据此位的变化灵活判断当前的运算状态。

##### 42.2.1.2 结果处理

开放运算的运算结果是 17 位数据输出，其最低位结果采用四舍五入方式得到。例如，当输入数据为 0x0000000CH 时，其结果为 0x00003H；当输入数据为 0x0000000DH 时，其结果为 0x00004H。

##### 42.2.1.3 定点数处理

当输入数据为定点小数，且需要保持输入输出数据的格式统一时，用户可通过控制状态寄存器的移位控制位 (MAU\_CSR.SHIFT) 来对输出数据进行相应的处理。

例如，当采用 Q14.2 格式进行定点数运算时，可设定 MAU\_CSR.SHIFT=1，使输出结果也遵循 Q14.2 格式。定点数 6.25 (Q14.2 格式为 0x00000019H) 在移位之前的输出结果位 5 (0x00000005H)，移位 1 位之后的结果为 2.5 (Q14.2 格式为 0x0000000AH)。

## 42.2.2 正弦运算

正弦运算支持 12 位数据输入，运算结果为 16 位有符号数输出。其中 12 位数据输入表示要运算的直角坐标系角度；16 位数据输出中，最高位表示结果的正负。

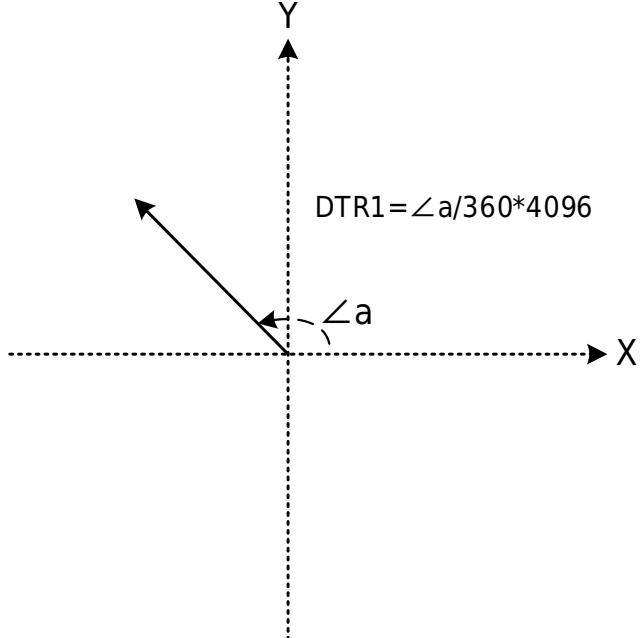


图 42-1 正弦运算角度值示意图

### 42.2.2.1 操作流程

正弦运算的操作流程如下：

- 1) 在数据输入寄存器 1 (MAU\_DTR1) 中写入角度值
- 2) 在 1 个周期之后从结果输出寄存器 1 (MAU\_RTR1) 中读取运算结果
- 3) 重复以上流程 1) ~2)

## 42.3 中断及事件说明

### 42.3.1 中断输出

在每一次开方运算完成之后，都会有相应的运算完成事件产生。若使能中断控制位 (MAU\_CSR.INTEN=1)，则对应的中断请求信号 (MAU\_SQRT) 也会产生。

### 42.3.2 事件输出

在每一次开方运算完成之后，都会有相应的运算完成事件产生，相应的事件请求信号 (MAU\_SQRT) 也会输出，可以用于选择触发其它模块。

## 42.4 寄存器说明

表 42-1 所示，为 MAU 模块的寄存器列表。

表 42-1 MAU 寄存器列表

BASE ADDR: 0x40055000

寄存器名	符号	偏移量	位宽	复位值
控制状态寄存器	MAU_CSR	0x0000	32	0x00000000
数据输入寄存器0	MAU_DTR0	0x0004	32	0x00000000
结果输出寄存器0	MAU_RTR0	0x000C	32	0x00000000
数据输入寄存器1	MAU_DTR1	0x0010	32	0x00000000
结果输出寄存器1	MAU_RTR1	0x0014	32	0x00000000

### 42.4.1 控制状态寄存器 (MAU\_CSR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16												
Reserved																											
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0												
-	-	-		SHIFT[4:0]			-	-	-	-	BUSY	-	INTEN	START													
<hr/>																											
位	标记	位名	功能	读写																							
b31~b13	Reserved	-	读出时为“0”,写入时写“0”	R/W																							
00000: 运算结果不移位输出																											
00001: 运算结果左移1位后输出																											
b12~b8	SHIFT[4:0]	开方结果移位	00010: 运算结果左移2位后输出 ... 10000: 运算结果左移16位后输出 请不要设定其它值	R/W																							
b7~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W																							
b3	BUSY	开方运算状态	0: 开方运算未开始或运算完成 1: 正在进行开方运算	R/W																							
b2	Reserved	-	读出时为“0”,写入时写“0”	R/W																							
b1	INTEN	开方运算中断使能	0: 开方运算完成后不产生中断 1: 开方运算完成后产生中断	R/W																							
b0	START	开方运算开始	0: 该位写零无效 1: 开方运算开始 注: 该位在写入1时内部产生一次START请求, 在读出时始终为0	R/W																							

#### 42.4.2 数据输入寄存器 0 (MAU\_DTR0)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SQRT_DIN[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SQRT_DIN[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b0	SQRT_DIN[31:0]	被开方数	被开方数输入	R/W											

#### 42.4.3 结果输出寄存器 0 (MAU\_RTR0)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SQRT_DOUT[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b17	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b16~b0	SQRT_DOUT[16:0]	开方结果	开方结果输出	R/W											

#### 42.4.4 数据输入寄存器 1 (MAU\_DTR1)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
SIN_DIN[11:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b11~b0	SIN_DIN[11:0]	角度设定	直角坐标系角度输入	R/W											

#### 42.4.5 结果输出寄存器 1 (MAU\_RTR1)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SIN_DOUT[15:0]															
<hr/>															
位	标记	位名	功能	读写											
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b15~b0	SIN_DOUT[15:0]	正弦运算结果	正弦运算结果输出	R/W											

## 43 调试控制器 (DBG)

本产品参考如下 ARM 技术文档：

- Cortex M4 Technical Reference Manual r0p1
- Arm Debug Interface Architecture Specification version 5
- CoreSight Components Technical Reference Manual version 1.0

### 43.1 简介

本 MCU 的内核是 Cortex-M4，该内核包含用于高级调试功能的硬件，支持嵌入式跟踪宏单元 (ETM)。利用这些调试功能，可以在取指（指令断点）或访问数据（数据断点）时停止内核。内核停止时，可以查询内核的内部状态和系统的外部状态。查询完成后，将恢复内核和系统并恢复程序执行。

提供两个调试接口：

- 串行调试跟踪接口 SWD
- 并行调试跟踪接口 JTAG

### 43.2 DBG 系统框图

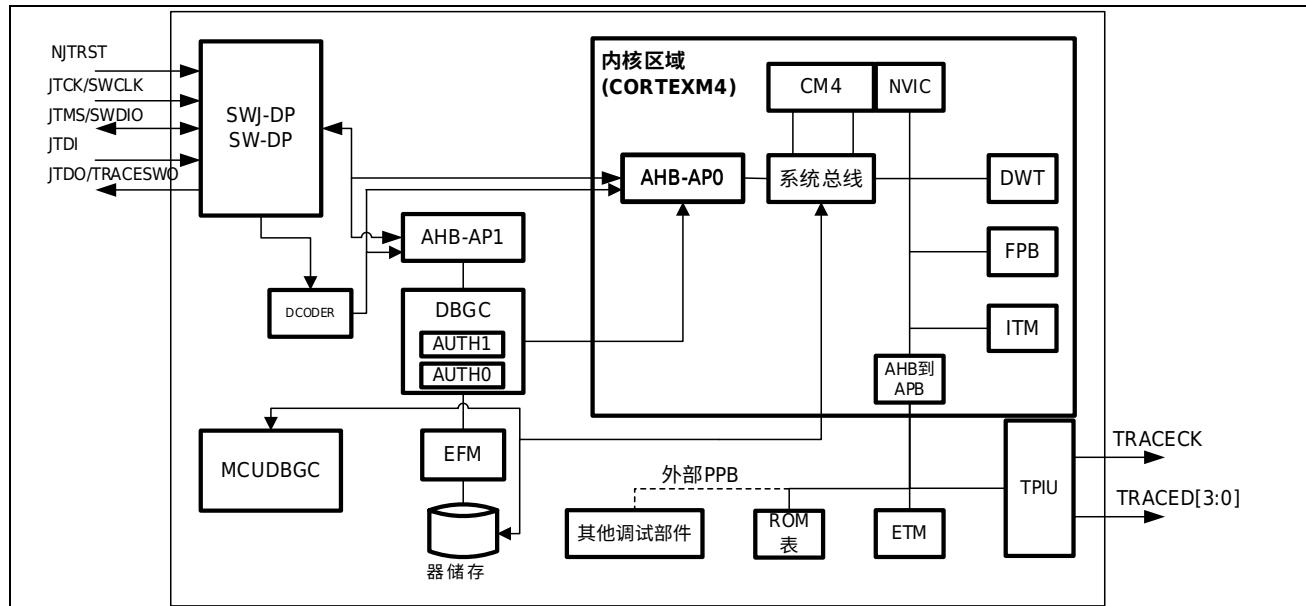


图 43-1 调试控制系统

ARM Cortex-M4 内核提供集成片上调试支持。它包括：

- SWJ-DP: SWD/JTAG 调试端口
- AHB-AP: AHB 访问端口
- ITM: 指令跟踪单元
- ETM: 嵌入式跟踪宏单元

- FPB: Flash 指令断点
- DWT: 数据断点触发
- TPIU: 跟踪端口单元接口（大封装上提供，其中会映射相应引脚）
- 灵活调试引脚分配

**注意：**

- 有关 ARM Cortex-M4 内核支持的调试功能的更多信息，请参见《Cortex-M4F-r0p1 技术参考手册》和《CoreSight 设计套件 r0p1 技术参考手册》。

### 43.3 SWJ-DP 调试端口 (SWD 和 JTAG)

该 MCU 内核集成了 SWD/JTAG 调试端口(SWJ-DP)。该端口是 ARM 标准 CoreSight 调试端口，具有 JTAG-DP (5 引脚) 接口和 SW-DP (2 引脚) 接口。

- JTAG 调试端口(JTAG-DP)提供用于连接到 AHB-AP 端口的 5 引脚标准 JTAG 接口。
  - 串行线调试端口(SW-DP)提供用于连接到 AHB-AP 端口的 2 引脚 (时钟+数据) 接口。
- 在 SWJ-DP 中，SW-DP 的 2 个 JTAG 引脚与 JTAG-DP 的 5 个 JTAG 引脚中的部分引脚复用。

下图中 JTDO 复用了 TRACESWO 与 TDO。这意味着异步跟踪只能在 SW-DP 上实现，不能在 JTAG-DP 上实现。

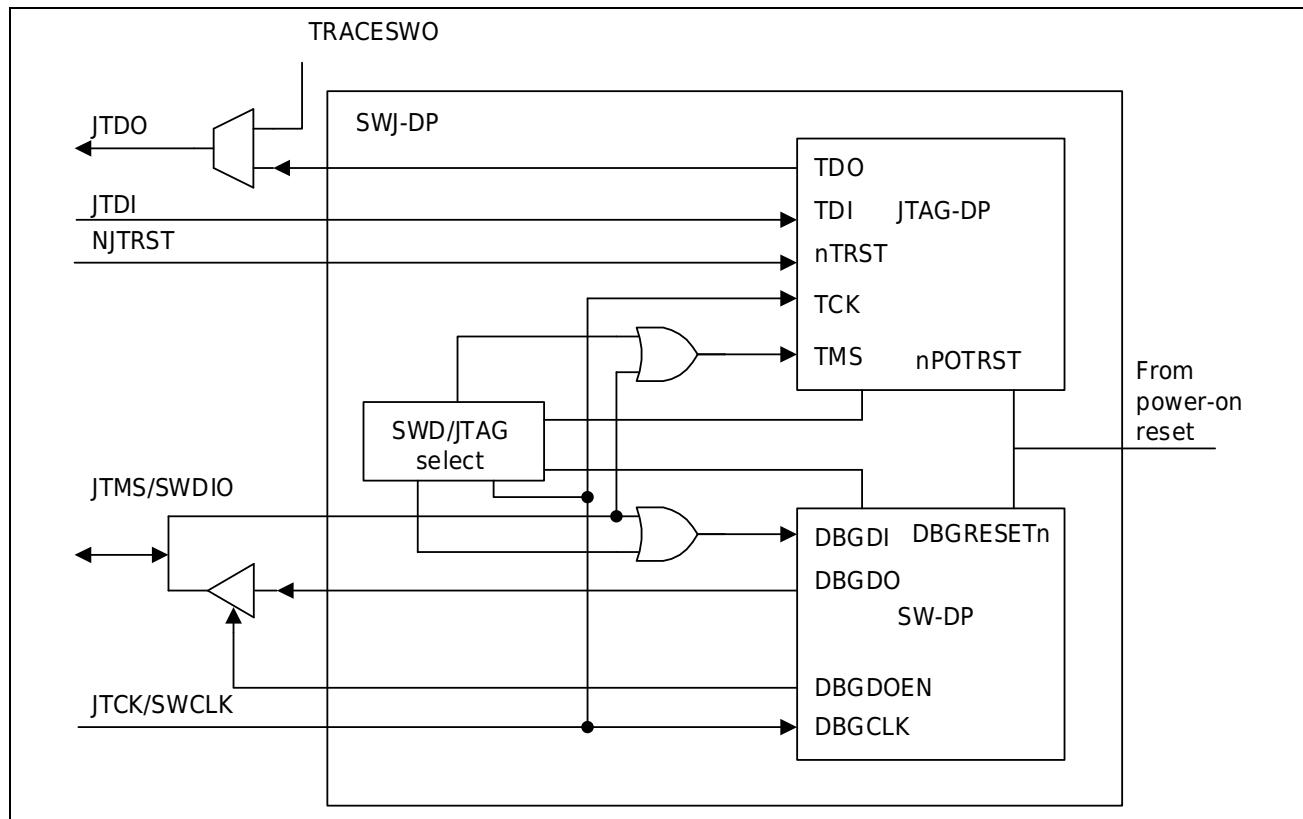


图 43-2 调试控制系统

#### 43.3.1 JTAG-DP 或 SW-DP 的切换机制

默认调试接口是 JTAG-DP 接口。

如果调试工具想要切换到 SW-DP，它必须在 JTMS(SWDIO)/JTCK(SWCLK)上提供专用的 JTAG 序列，用于禁止 JTAG-DP 并使能 SW-DP。这样便可仅使用 SWCLK 和 SWDIO 引脚来访问 SW-DP。

该序列为：

1. 输出超过 50 个 JTCK 周期的 JTMS (SWDIO) =1 信号
  2. 输出 16 个 JTMS (SWDIO) 信号 0111\_1001\_1110\_0111 (MSB)
  3. 输出超过 50 个 JTCK 周期的 JTMS (SWDIO) =1 信号

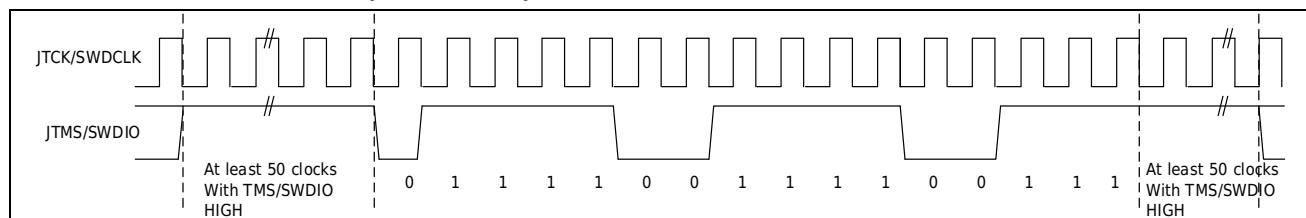


图 43-3 JTAG-DP 到 SW-DP 切换时序

#### 43.4 引脚排列和调试端口引脚

根据 MCU 的不同封装有不同的有效引脚数。因此，某些与引脚相关的功能可能随封装而不同。

#### 43.4.1 SWJ 调试端口引脚

MCU 的 5 个普通 I/O 口可用作 SWI-DP 接口引脚。

表 43-1 SWJ 调试端口引脚

SWJ-DP 引脚名称	JTAG调试端口		SW调试端口	
	类型	说明	类型	调试分配
JTMS/SWDIO	I	JTAG模式选择	I/O	串行线数据输入/输出
JTCK/SWCLK	I	JTAG时钟	I	串行时钟
JTDI	I	JTAG数据输入	-	-
JTDO/TRACESWO	O	JTAG数据输出	-	TRACESWO(如果使能异步跟踪)
NJTRST	I	JTAG复位	-	-

#### 43.4.2 灵活的 SWJ-DP 引脚分配

复位（上电或引脚复位）后，会将用于 SWJ-DP 的全部 5 个引脚指定为专用引脚，可供调试工具立即使用（请注意，除非明确编程，否则不分配跟踪输出）。但是，MCU 可以禁止部分或全部 SWJ-DP 端口，进而释放相关引脚以用作 GPIO。有关如何禁止 SWJ-DP 端口引脚的更多详细信息，请参见：通用 IO 特殊控制寄存器 PSPCR。

表 43-2 灵活的 SWJ-DP 引脚分配

可用的调试端口	分配的 SWJ IO 引脚				
	JTMS/ SWDIO	JTCK/ SWCLK	JTDI	JTDO	NJTRST
全部 SWJ (JTAG-DP+SW-DP) - 复位状态	✓	✓	✓	✓	✓
禁止 JTAG-DP 和使能 SW-DP	✓	✓	可释放	可释放	可释放
禁止 JTGA-DP 和禁止 SW-DP	可释放	可释放	可释放	可释放	可释放

### 43.4.3 JTAG 引脚上的内部上拉

根据 JTAG IEEE 标准，必须确保 JTAG 输入引脚不悬空，因为这些引脚直接连接到 MCU 内部用于控制调试功能。还必须特别注意 JTCK/SWCLK 引脚，该引脚直接用于调试控制时钟功能。为避免 IO 电平浮空，MCU 在 JTAG 引脚上内置有内部上拉电阻：

- NJTRST：内部上拉
- JTDI：内部上拉
- JTMS/SWDIO：内部上拉
- JTCK/SWCLK：内部上拉
- JTDO：高阻状态

在未连接调试器状态下，用户软件可以通过设定 GPIO 特殊控制寄存器释放 JTAG IO 作为普通的 I/O 口使用。由于芯片内部上拉为<100K 欧的弱上拉，建议采用外部 10K 欧的外部上拉。

### 43.4.4 使用串行接口以及释放未使用的调试引脚以作 GPIO

使用 SWD 时可以释放一些 GPIO，用户软件必须在 GPIO 控制寄存器中更改 GPIO 配置，这样便可释放相应的引脚用作 GPIO。

调试时，主机执行以下操作：

- 在系统复位状态下，分配所有 SWJ 引脚(JTAG-DP+SW-DP)。
- 在系统复位状态下，调试主机发送 JTAG 序列，以从 JTAG-DP 切换到 SW-DP。
- 仍然在系统复位状态下，调试主机在复位地址处设置断点。
- 释放复位信号，内核停止在复位地址处。
- 从此调试端口切换为 SW-DP。然后通过用户软件将其它 JTAG 引脚重新分配为 GPIO。

**注意：**

- 对于用户软件设计，需要释放调试引脚时，在复位后一直到用户软件释放引脚这段期间，这些引脚仍然处于输入上拉 (NJTRST、JTMS、JTDI 和 JTCK)

## 43.5 寄存器说明

表 43-3 寄存器列表

基址地址：0xE0042000

寄存器名	符号	偏移地址	位宽	初始值	访问主机
芯片专属标志寄存器	CHIPID	0x000C	32	0x48440682	调试IDE
DBG状态寄存器	MCUDBGSTAT	0x001C	32	0x00000000	CPU/调试IDE*
外设调试暂停寄存器	MCUSTPCTL	0x0020	32	0x0000003B	CPU/调试IDE*
调试组件配置寄存器	MCUTRACECTL	0x0024	32	0x00000000	CPU/调试IDE*
外设调试暂停寄存器	MCUSTPCTL2	0x0028	32	0x00000000	CPU/调试IDE*

注意：

- 寄存器位于 PPB 区域，只能在特权模式下 CPU 才能进行访问。

### 43.5.1 芯片专属标志寄存器 (CHIPID)

用于表示公司信息及产品型号信息。

复位值：0x4844 0682

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CHIPID[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CHIPID[15:0]															
<hr/>															
位	标记	位名			功能						读写				
b31~0	CHIPID[31:0]	芯片专属标志			CHIPID[31:16]:HD的对应ASCII码，0x4844						R				
CHIPID[15:0]:产品型号，0x0682															

### 43.5.2 DBG 状态寄存器 (MCUDBGSTAT)

DBG 调试上电状态确认寄存器。

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	CDB G PWR UPAC K	CDB G PWR UPRE Q

位	标记	位名	功能	读写
b31~2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	CDBGWPWRUPACK	调试器上电反馈	0: 无反馈 1: 调试上电反馈	R/W
b0	CDBGWPWRUPREQ	调试器上电请求	0: 无上电请求 1: 上电请求	R/W

### 43.5.3 外设调试暂停寄存器 (MCUSTPCTL)

当 CPU 处于调试状态时，周边模块暂停控制。

复位值：0x0000 003B

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	M24 STP	M24 STP	M22 STP	M21 STP	M20 STP	M19 STP	M18 STP	M17 STP	M16 STP
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
M15 STP	-	-	M12 STP	M11 STP	M10 STP	M09 STP	M08 STP	M07 STP	M06 STP	-	-	-	RTC STP	WDT STP	SWD TSTP

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“0”,写入时写“0”	R/W
b30	Reserved	-	读出时为“0”,写入时写“0”	R/W
b29	Reserved	-	读出时为“0”,写入时写“0”	R/W
b28	Reserved	-	读出时为“0”,写入时写“0”	R/W
b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26	Reserved	-	读出时为“0”,写入时写“0”	R/W
b25	Reserved	-	读出时为“0”,写入时写“0”	R/W
b24	M24STP	TMR6_10计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b23	M23STP	TMR6_9计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b22	M22STP	TMR6_8计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b21	M21STP	TMR6_7计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b20	M20STP	TMR6_6计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b19	M19STP	TMR6_5计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b18	M18STP	TMR6_4计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b17	M17STP	TMR6_3计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b16	M16STP	TMR6_2计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b15	M15STP	TMR6_1计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b14	Reserved	-	读出时为“0”,写入时写“0”	R/W
b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	M12STP	TMR4_1计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b11	M11STP	TMR2_4计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W

b10	M10STP	TMR2_3计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b9	M09STP	TMR2_2计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b8	M08STP	TMR2_1计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b7	M07STP	TMRO_2计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b6	M06STP	TMRO_1计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b5~3	Reserved	-	读出时为“1”,写入时写“1”	R/W
b2	RTCSTP	RTC计数暂停信号	0: 即使内核停止, RTC计数器仍然计数 1: 内核停止时, RTC计数器暂停计数	R/W
b1	WDTSTP	WDT计数暂停信号	0: 即使内核停止, WDT计数器仍然计数 1: 内核停止时, WDT计数器暂停计数	R/W
b0	SWDTSTP	SWDT计数暂停信号	0: 即使内核停止, SWDT计数器仍然计数 1: 内核停止时, SWDT计数器暂停计数	R/W

#### 43.5.4 调试组件配置寄存器 (MCUTRACECTL)

通过该寄存器配置 TRACE 输出引脚。

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	TRACE_IOEN	TRACE_MODE	

位	标记	位名	功能	读写
b31~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	TRACE_IOEN	TRACE引脚输出控制	0: 同步跟踪引脚输出禁止 1: 同步跟踪引脚输出许可	R/W
b1~b0	TRACE_MODE	TRACED输出引脚控制	00: 异步跟踪 01: 同步跟踪1位TRACED[0] 10: 同步跟踪2位TRACED[1:0] 11: 同步跟踪4位TRACED[3:0]	R/W

### 43.5.5 外设调试暂停寄存器 2 (MCUSTPCTL2)

当 CPU 处于调试状态时，周边模块暂停控制。

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	M37 STP	M36 STP	M35 STP	M34 STP	M33 STP	M32 STP

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“0”,写入时写“0”	R/W
b30	Reserved	-	读出时为“0”,写入时写“0”	R/W
b29	Reserved	-	读出时为“0”,写入时写“0”	R/W
b28	Reserved	-	读出时为“0”,写入时写“0”	R/W
b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26	Reserved	-	读出时为“0”,写入时写“0”	R/W
b25	Reserved	-	读出时为“0”,写入时写“0”	R/W
b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	Reserved	-	读出时为“0”,写入时写“0”	R/W
b22	Reserved	-	读出时为“0”,写入时写“0”	R/W
b21	Reserved	-	读出时为“0”,写入时写“0”	R/W
b20	Reserved	-	读出时为“0”,写入时写“0”	R/W
b19	Reserved	-	读出时为“0”,写入时写“0”	R/W
b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17	Reserved	-	读出时为“0”,写入时写“0”	R/W
b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14	Reserved	-	读出时为“0”,写入时写“0”	R/W
b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12~6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5	M37STP	TMRA_6计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b4	M36STP	TMRA_5计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b3	M35STP	TMRA_4计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b2	M34STP	TMRA_3计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b1	M33STP	TMRA_2计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W

---

b0	M32STP	TMRA_1计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
----	--------	--------------	-----------------------------------------	-----

---

## 43.6 SW 调试端口

### 43.6.1 SW 协议简介

同步串行协议使用两个引脚：

- SWCLK：从主机到从机的时钟
- SWDIO：双向

传输数据时，LSB 在前。

对于 SWCLK 和 SWDIO，需要在电路板上对线路进行上拉（建议采用 10 K 欧）。

## 43.7 TPIU (跟踪端口接口单元)

### 43.7.1 简介

TPIU 是 ITM 和 ETM 与片上跟踪数据之间的桥接器。

输出数据流封装成跟踪源 ID，然后被跟踪端口分析器 (TPA) 捕获。

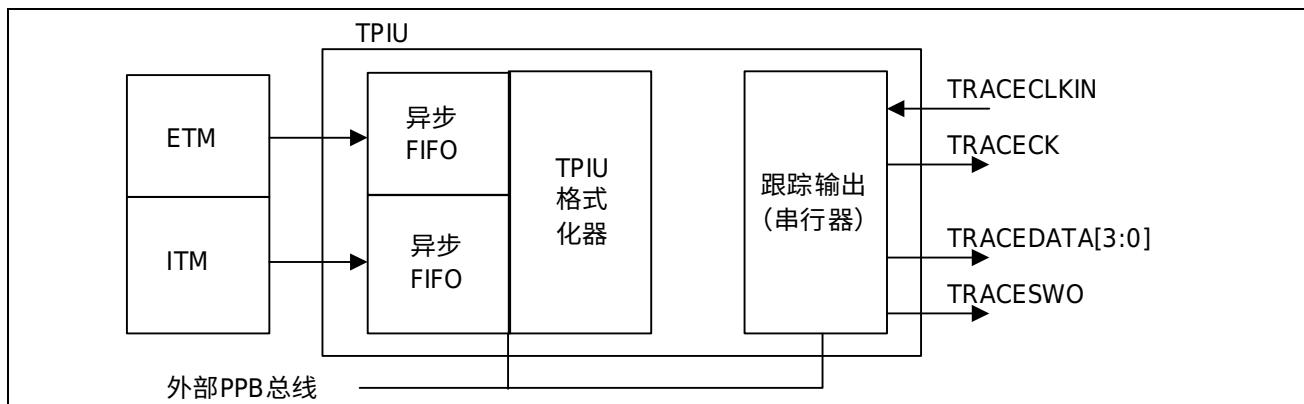


图 43-4 TPIU 框图

### 43.7.2 TRACE 引脚分配

#### ■ 异步模式

异步模式需要 1 个额外引脚，并且适用于所有封装。仅在使用串行模式时异步模式才可用（在 JTAG 模式下不可用）。

表 43-4 异步模式引脚

TPIU引脚名称	跟踪异步模式	
	类型	说明
TRACESWO	输出	TRACE异步数据输出

#### ■ 同步模式

同步模式需要 2 到 5 个额外引脚，具体取决于所跟踪数据的长度，并且仅适用于较大型的封装。此外，同步模式在 JTAG 模式和串行模式下均可用，并可提供比异步跟踪更高的带宽输出能力。

表 43-5 同步模式引脚

TPIU引脚名称	跟踪同步模式	
	类型	说明
TRACECK	输出	TRACE时钟
TRACED[3:0]	输出	TRACE同步数据输出，可以是1、2或4.

#### TPIU TRACE 引脚分配

默认情况下，不分配这些引脚。可通过将 MCU 调试组件配置寄存器 (MCUTRACECTL) 中的 TRACE\_IOEN 和 TRACE\_MODE 位来配置这些引脚。必须由调试主机或 CPU 来完成此配置。

此外，要分配的引脚数目取决于跟踪配置（异步跟踪或同步跟踪）。

- 异步模式：需要 1 个额外引脚
- 同步模式：需要 5 个额外引脚
  - TRACECK
  - TRACED[0]（如果端口数据长度配置为 1, 2 或 4）
  - TRACED[1]（如果端口数据长度配置为 2 或 4）
  - TRACED[2]（如果端口数据长度配置 4）
  - TRACED[3]（如果端口数据长度配置 4）

要分配 TRACE 引脚，调试主机必须对 MCU 调试组件配置寄存器 (MCUTRACECTL) 的位 TRACE\_IOEN 和 TRACE\_MODE[1:0] 进行编程。默认情况下不分配 TRACE 引脚。

此寄存器映射到外部 PPB 总线上，通过上电（而非引脚复位）复位。可在引脚复位状态下通过调试器写入该寄存器。

表 43-6 引脚释放

TPIU引脚用途	分配的TRACE IO引脚					
	JTDO/ TRACESWO	TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]
无跟踪 (默认状态) TRACE_IOEN = 0 TRACE_MODE=XX	释放*	释放	释放	释放	释放	释放
异步跟踪 TRACE_IOEN = 1 TRACE_MODE=00	TRACESWO	释放	释放	释放	释放	释放
同步跟踪1位 TRACE_IOEN = 1 TRACE_MODE=01	释放*	TRACECK	TRACED[0]	释放	释放	释放
同步跟踪2位 TRACE_IOEN = 1 TRACE_MODE=10	释放*	TRACECK	TRACED[0]	TRACED[1]	释放	释放
同步跟踪4位 TRACE_IOEN = 1 TRACE_MODE=11	释放*	TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]

注意：

- 使用串行模式时，释放此引脚。但使用 JTAG 时，此引脚分配给 TDO。

### 43.7.3 MCU 内部 TRACECLKIN 连接

在本 MCU 中，TPIU 的时钟 TRACECLKIN 连接到内部时钟。MCU 的默认时钟是内部 MRC 振荡器。复位状态下的频率与复位释放后的频率不同。原因是，由于系统复位状态下采用默认的 MRC 校准值，而在每次系统复位释放时会更新该 MRC 校准值。因此，跟踪端口分析器 (TPA) 在系统复位状态下不应使能跟踪（使用 TRACE\_IOEN 位），原因是，在复位状态下的同步帧包的比特宽度与复位后的包不同。

### 43.7.4 TPIU 寄存器

仅当调试异常和监视控制寄存器 (DEMCR) 的位 TRCENA 置 1 时才能对 TPIU APB 寄存器进行读写操作。否则，这些寄存器将读为零（此位的输出会使能 TPIU 的时钟）。

### 43.7.5 TPIU 配置示例

- 将调试异常和监视控制寄存器 (DEMCR) 中的位 TRCENA 置 1
- 将所需值写入 TPIU 当前端口大小寄存器（对于 1 位端口大小，默认值为 0x1）
- 将 0x102 写入 TPIU 格式化器和刷新控制寄存器（默认值）
- 写入 TPIU 选择引脚协议以选择同步模式或异步模式。示例：0x2 表示异步 NRZ 模式（类似于 USART）
- 将 0x00 写入 MCUTRACECTL 控制寄存器（位 TRACE\_IOEN），为异步模式分配 TRACE I/O
- 此时发送 TPIU 同步数据包 (FF\_FF\_FF\_7F)
- 配置 ITM 并对 ITM 激励寄存器进行写操作以输出值

## 版本修订记录

版本号	修订日期	修订内容
Rev1.0	2023/12/18	初版发布。