

HC32F334 系列

32 位 ARM[®] Cortex[®]-M4 微控制器

参考手册

Rev1.00 2024 年 06 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。

©2024 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.....	32
图索引.....	38
文档约定.....	48
寄存器相关缩略语.....	48
词汇表.....	48
关于本手册.....	48
简介 (Overview)	49
1 存储器映射 (Memory Mapping)	50
1.1 存储器映射.....	50
1.2 位段空间.....	56
1.3 地址重映射.....	56
1.4 寄存器描述.....	57
1.4.1 访问保护寄存器 (MMF_REMPRT)	58
1.4.2 重映射控制寄存器 (MMF_REMCrx, x=0, 1)	58
2 总线架构 (BUS)	59
2.1 概述.....	59
2.2 总线架构.....	60
2.3 总线功能.....	61
3 复位控制 (RMU)	62
3.1 简介.....	62
3.2 复位方式和复位标志位.....	63
3.3 复位时序.....	65
3.3.1 上电复位.....	65
3.3.2 NRST 引脚复位.....	66
3.3.3 欠压复位.....	66
3.3.4 可编程电压检测 1 复位、可编程电压检测 2 复位.....	67
3.3.5 看门狗复位、专用看门狗复位.....	69
3.3.6 掉电唤醒复位.....	69
3.3.7 软件复位.....	70
3.3.8 MPU 错误复位.....	70
3.3.9 RAM 奇偶校验复位.....	70

3.3.10	RAMECC 复位	71
3.3.11	时钟频率异常复位	71
3.3.12	外部高速振荡器异常停振复位	71
3.3.13	Cortex-M4 Lockup 复位	72
3.3.14	复位方式的判断	72
3.3.15	模块功能复位	73
3.3.16	各个模块的复位条件	73
3.4	寄存器描述	75
3.4.1	功能复位控制 0 (RMU_FRST0)	76
3.4.2	功能复位控制 1 (RMU_FRST1)	77
3.4.3	功能复位控制 2 (RMU_FRST2)	78
3.4.4	功能复位控制 3 (RMU_FRST3)	79
3.4.5	复位控制寄存器 (RMU_PRSTCR0)	80
3.4.6	复位标志寄存器 (RMU_RSTF0)	81
4	时钟控制器 (CMU)	83
4.1	概述	83
4.2	系统框图	84
4.2.1	系统框图	84
4.2.2	时钟频率测量框图	85
4.3	时钟源规格	86
4.4	工作时钟规格	87
4.5	晶振电路	89
4.5.1	外部高速振荡器	89
4.5.2	外部高速振荡器故障检测	91
4.5.3	外部高速振荡器小数分频	93
4.5.4	外部低速振荡器	94
4.6	内部 RC 时钟	95
4.6.1	HRC 时钟	95
4.6.2	MRC 时钟	95
4.6.3	LRC 时钟	95
4.6.4	SWDTLRC 时钟	96
4.7	PLL 时钟	97
4.8	时钟切换步骤	97
4.8.1	时钟源切换	98
4.8.2	时钟分频切换	99

4.9	时钟输出功能	100
4.10	时钟频率测量	101
4.10.1	时钟频率测量	101
4.10.2	数字滤波功能	102
4.10.3	中断/复位功能	102
4.11	寄存器描述	103
4.11.1	CMU XTAL 配置寄存器 (CMU_XTALCFGR)	105
4.11.2	CMU XTAL 稳定配置寄存器 (CMU_XTALSTBCR)	106
4.11.3	CMU XTAL 小数分频配置寄存器 (CMU_XTALDIVR)	106
4.11.4	CMU XTAL 小数分频控制寄存器 (CMU_XTALDIVCR)	107
4.11.5	CMU XTAL 控制寄存器 (CMU_XTALCR)	107
4.11.6	CMU XTAL 振荡故障控制寄存器 (CMU_XTALSTDCR)	108
4.11.7	CMU XTAL 振荡故障状态寄存器 (CMU_XTALSTDSR)	109
4.11.8	CMU XTAL32 配置寄存器 (CMU_XTAL32CFGR)	109
4.11.9	CMU XTAL32 滤波寄存器 (CMU_XTAL32NFR)	110
4.11.10	CMU XTAL32 控制寄存器 (CMU_XTAL32CR)	110
4.11.11	CMU HRC 校准寄存器 (CMU_HRCTRM)	111
4.11.12	CMU HRC 控制寄存器 (CMU_HRCCR)	111
4.11.13	CMU MRC 校准寄存器 (CMU_MRCTRM)	112
4.11.14	CMU MRC 控制寄存器 (CMU_MRCCR)	112
4.11.15	CMU LRC 校准寄存器 (CMU_LRCTRM)	113
4.11.16	CMU LRC 控制寄存器 (CMU_LRCCR)	113
4.11.17	CMU PLLH 配置寄存器 (CMU_PLLHCFGR)	114
4.11.18	CMU PLLH 控制寄存器 (CMU_PLLHCR)	116
4.11.19	CMU 时钟源稳定状态器 (CMU_OSCSTBSR)	116
4.11.20	CMU 系统时钟源切换寄存器 (CMU_CKSWR)	117
4.11.21	CMU 时钟分频配置寄存器 (CMU_SCFGR)	118
4.11.22	CMU CAN 时钟配置寄存器 (CMU_CANCKCFGR)	120
4.11.23	CMU ADC/DAC 时钟配置寄存器 (CMU_PERICKSEL)	121
4.11.24	CMU 调试时钟配置寄存器 (CMU_TPIUCKCFGR)	121
4.11.25	CMU MCO1 时钟输出配置寄存器 (CMU_MCO1CFGR)	122
4.11.26	CMU MCO2 时钟输出配置寄存器 (CMU_MCO2CFGR)	123
4.11.27	FCM 下限比较值寄存器 (FCM_LVR)	123
4.11.28	FCM 上限比较值寄存器 (FCM_UVR)	124
4.11.29	FCM 计数器值寄存器 (FCM_CNTR)	124

4.11.30	FCM 开始停止寄存器 (FCM_STR)	124
4.11.31	FCM 测量对象控制寄存器 (FCM_MCCR)	125
4.11.32	FCM 测量基准控制寄存器 (FCM_RCCR)	126
4.11.33	FCM 中断复位控制寄存器 (FCM_RIER)	127
4.11.34	FCM 标志寄存器 (FCM_SR)	128
4.11.35	FCM 标志位清除寄存器 (FCM_CLR)	128
5	电源控制 (PWC)	129
5.1	概述	129
5.2	电源分布	129
5.3	电源电压检测单元 (PVD) 说明	130
5.3.1	上电复位/掉电复位动作说明	130
5.3.2	欠压复位 (BOR) 说明	131
5.3.3	可编程电压检测 1 (PVD1)、可编程电压检测 2 (PVD2)	132
5.3.4	PVD1、PVD2 中断/复位框图	132
5.3.5	输入/输出引脚	133
5.3.6	PVD1 中断和复位	133
5.3.7	PVD2 中断和复位	134
5.3.8	内部电压采样和检测功能	135
5.4	唤醒定时器	136
5.5	动作模式与低功耗模式	136
5.5.1	睡眠模式	139
5.5.2	停止模式	140
5.5.3	掉电模式	141
5.5.4	降低系统时钟速度	144
5.5.5	关闭不使用的时钟源	144
5.5.6	功能时钟停止	144
5.5.7	关闭不使用的 RAM	145
5.6	寄存器保护功能	145
5.7	寄存器描述	146
5.7.1	电源模式控制寄存器 0 (PWC_PWRC0)	148
5.7.2	电源模式控制寄存器 1 (PWC_PWRC1)	148
5.7.3	电源模式控制寄存器 4 (PWC_PWRC4)	149
5.7.4	掉电唤醒使能寄存器 0 (PWC_PDWKE0)	149
5.7.5	掉电唤醒使能寄存器 1 (PWC_PDWKE1)	150
5.7.6	掉电唤醒使能寄存器 2 (PWC_PDWKE2)	150

5.7.7	掉电唤醒事件边沿选择寄存器 (PWC_PDWKES)	151
5.7.8	掉电唤醒标志寄存器 0 (PWC_PDWKF0)	152
5.7.9	掉电唤醒标志寄存器 1 (PWC_PDWKF1)	153
5.7.10	电源模式控制寄存器 5 (PWC_PWRC5)	153
5.7.11	电源模式控制寄存器 6 (PWC_PWRC6)	154
5.7.12	低功耗状态寄存器 (PWC_LPMCSR)	154
5.7.13	低功耗状态清除寄存器 (PWC_LPMCSCR)	154
5.7.14	功能时钟控制 0 (PWC_FCG0)	155
5.7.15	功能时钟控制 1 (PWC_FCG1)	156
5.7.16	功能时钟控制 2 (PWC_FCG2)	157
5.7.17	功能时钟控制 3 (PWC_FCG3)	159
5.7.18	PWC_FCG0 保护控制 (PWC_FCG0PC)	160
5.7.19	功能保护控制寄存器 (PWC_FPRC)	160
5.7.20	STOP 模式控制寄存器 (PWC_STPMCR)	161
5.7.21	RAM 功耗控制寄存器 0 (PWC_RAMPC0)	161
5.7.22	外设 RAM 低功耗控制寄存器 (PWC_PRAMLPC)	162
5.7.23	PVD 控制寄存器 0 (PWC_PVDCR0)	162
5.7.24	PVD 控制寄存器 1 (PWC_PVDCR1)	163
5.7.25	PVD 滤波控制寄存器 (PWC_PVDFCR)	164
5.7.26	PVD 电平控制寄存器 (PWC_PVDLCR)	165
5.7.27	PVD 中断控制寄存器 (PWC_PVDICR)	166
5.7.28	PVD 检测状态寄存器 (PWC_PVDDSR)	167
5.7.29	唤醒计时器控制寄存器 (PWC_WKTCR)	168
6	初始化配置 (ICG)	169
6.1	概述	169
6.2	寄存器描述	170
6.2.1	初始化配置寄存器 0 (ICG0)	171
6.2.2	初始化配置寄存器 1 (ICG1)	173
6.2.3	初始化配置寄存器 3 (ICG3)	174
7	嵌入式 Flash (EFM)	175
7.1	概述	175
7.2	主要特性	175
7.3	嵌入式 Flash	175
7.4	读接口	177
7.4.1	CPU 时钟和 Flash 读取时间之间的关系	177

7.5	Flash 读加速缓存	178
7.6	Flash 编程和擦除操作	179
7.6.1	解锁 EFM_KEY1 寄存器	179
7.6.2	写保护功能	179
7.6.3	单次编程无回读功能	179
7.6.4	单编程回读功能	180
7.6.5	连续编程功能	180
7.6.6	擦除功能	181
7.6.7	全擦除功能	181
7.6.8	数据安全保护	182
7.6.9	D-BUS 读保护功能	183
7.6.10	总线保持功能	183
7.6.11	中断	183
7.7	一次性可编程字节 (OTP)	184
7.8	引导交换	185
7.9	寄存器描述	187
7.9.1	访问写保护寄存器 (EFM_FAPRT)	188
7.9.2	Flash 密钥 1 寄存器 (EFM_KEY1)	188
7.9.3	Flash 密钥 2 寄存器 (EFM_KEY2)	189
7.9.4	Flash 停止寄存器 (EFM_FSTP)	189
7.9.5	读模式寄存器 (EFM_FRMC)	190
7.9.6	Flash 擦写模式寄存器 (EFM_FWMC)	191
7.9.7	Flash 状态寄存器 (EFM_FSR)	192
7.9.8	Flash 状态清除寄存器 (EFM_FSCLR)	193
7.9.9	Flash 中断许可寄存器 (EFM_FITE)	194
7.9.10	Flash 引导交换状态寄存器 (EFM_FSWP)	194
7.9.11	芯片专属标志寄存器 (EFM_CHIPID)	195
7.9.12	UNIQUE ID 寄存器 0 (EFM_UQID0)	195
7.9.13	UNIQUE ID 寄存器 1 (EFM_UQID1)	195
7.9.14	UNIQUE ID 寄存器 2 (EFM_UQID2)	196
7.9.15	Flash 写保护锁存寄存器 (EFM_WLOCK)	196
7.9.16	Flash 写保护寄存器 (EFM_FONWPRT)	197
7.10	注意事项	198
8	内置 SRAM (SRAM)	199
8.1	概述	199

8.2	寄存器描述.....	201
8.2.1	SRAM 校验控制寄存器 (SRAMC_CKCR)	202
8.2.2	SRAM 校验控制保护寄存器 (SRAMC_CKPR)	204
8.2.3	SRAM 校验状态寄存器 (SRAMC_CKSR)	205
8.2.4	SRAM0 错误注入使能寄存器 (SRAM0_EIEN)	205
8.2.5	SRAM0 错误注入位使能寄存器 0 (SRAM0_EIBIT0)	206
8.2.6	SRAM0 错误注入位使能寄存器 1 (SRAM0_EIBIT1)	206
8.2.7	SRAM0 校验错误地址寄存器 (SRAM0_ECCERRADDR)	207
8.2.8	SRAMH 错误注入使能寄存器 (SRAMH_EIEN)	207
8.2.9	SRAMH 错误注入位使能寄存器 0 (SRAMH_EIBIT0)	208
8.2.10	SRAMH 错误注入位使能寄存器 1 (SRAMH_EIBIT1)	208
8.2.11	SRAMH 校验错误地址寄存器 (SRAMH_ECCERRADDR)	209
8.2.12	Ret SRAM 错误注入使能寄存器 (SRAMB_EIEN)	209
8.2.13	Ret SRAM 错误注入位使能寄存器 0 (SRAMB_EIBIT0)	209
8.2.14	Ret SRAM 错误注入位使能寄存器 1 (SRAMB_EIBIT1)	210
8.2.15	Ret SRAM 校验错误地址寄存器 (SRAMB_ECCERRADDR)	210
8.3	注意事项.....	211
9	通用 IO (GPIO)	212
9.1	简介.....	212
9.2	功能说明.....	213
9.2.1	GPIO 端口基本结构.....	213
9.2.2	通用输入输出 GPIO 功能.....	214
9.2.3	周边功能	214
9.2.4	双周边功能.....	215
9.2.5	Event Port 输入输出功能	215
9.2.6	外部中断 EIRQ 输入功能	215
9.2.7	模拟功能	215
9.2.8	通用控制	216
9.3	寄存器描述.....	217
9.3.1	通用输入数据寄存器 (PIDRx)	218
9.3.2	通用输出数据寄存器 (PODRx)	218
9.3.3	通用输出许可寄存器 (POERx)	219
9.3.4	通用输出置位寄存器 (POSRx)	219
9.3.5	通用输出复位寄存器 (PORRx)	220
9.3.6	通用输出翻转寄存器 (POTRx)	220

9.3.7	HRPWM 控制寄存器 (HRPWM)	221
9.3.8	特殊控制寄存器 (PSPCR)	223
9.3.9	公共控制寄存器 (PCCR)	223
9.3.10	写保护寄存器 (PWPR)	224
9.3.11	通用控制寄存器 (PCRxy)	225
9.3.12	功能选择寄存器 (PFSRxy)	227
9.3.13	Event Port 方向选择寄存器 (PEVNTDIRRm)	227
9.3.14	Event Port 输入数据寄存器 (PEVNTIDRm)	228
9.3.15	Event Port 输出数据寄存器 (PEVNTODRm)	228
9.3.16	Event Port 输出数据复位寄存器 (PEVNTORRm)	229
9.3.17	Event Port 输出数据置位寄存器 (PEVNTOSRm)	229
9.3.18	Event Port 上升沿输入许可寄存器 (PEVNTRISRm)	230
9.3.19	Event Port 下降沿输入许可寄存器 (PEVNTFALRm)	230
9.3.20	Event Port 输入滤波控制寄存器 (PEVNTNFCR)	231
9.4	注意事项	232
10	中断控制器 (INTC)	233
10.1	简介	233
10.2	系统框图	234
10.3	向量表	235
10.3.1	中断向量表	235
10.3.2	中断事件表	240
10.4	功能说明	256
10.4.1	不可屏蔽中断	256
10.4.2	外部中断	256
10.4.3	中断事件多路器	257
10.4.4	软件中断	257
10.4.5	事件输入	257
10.4.6	WFE 事件管理	258
10.4.7	噪声滤波器	258
10.4.8	低功耗模式返回	259
10.5	寄存器描述	260
10.5.1	不可屏蔽中断使能寄存器 (INTC_NMIER)	262
10.5.2	不可屏蔽中断标志寄存器 (INTC_NMIFR)	263
10.5.3	不可屏蔽中断标志清除寄存器 (INTC_NMIFCR)	264
10.5.4	外部中断控制寄存器 (INTC_EIRQCRx, x=0~15)	265

10.5.5	外部中断标志寄存器 (INTC_EIFR)	266
10.5.6	外部中断标志清除寄存器 (INTC_EIFCR)	266
10.5.7	中断事件选择寄存器 (INTC_INTSEL0~15).....	266
10.5.8	中断事件使能寄存器 (INTC_INTEN0~15)	267
10.5.9	停止模式唤醒事件使能寄存器 (INTC_WKEN)	268
10.5.10	软件中断寄存器 (INTC_SWIER)	269
10.5.11	事件使能寄存器 (INTC_EVTER)	269
10.5.12	中断请求使能寄存器 (INTC_IER)	269
10.5.13	FPU 中断使能寄存器 (INTC_FPUIER)	270
10.6	注意事项	271
11	自动运行系统 (AOS)	272
11.1	概述	272
11.1.1	功能概览	272
11.2	功能说明	273
11.2.1	AOS 框图	273
11.2.2	AOS 源事件和目标	274
11.2.3	专用触发源	275
11.2.4	公共触发源	275
11.3	寄存器描述	276
11.3.1	外设触发事件寄存器 (INTSFTRG)	277
11.3.2	DMA 传输启动触发源选择寄存器 (DMA_TRGSELx, x=0~7)	277
11.3.3	DMA 通道重置触发源选择寄存器 (DMA_RC_TRGSEL)	278
11.3.4	Timer6 硬件触发事件选择寄存器 (TMR6_TRGSELx, x=0~3)	278
11.3.5	HRPWM 硬件触发事件选择寄存器 (HRPWM_TRGSELx, x=0~3)	279
11.3.6	Timer4 硬件触发事件选择寄存器 (TMR4_TRGSEL)	279
11.3.7	Event Port 触发源选择寄存器 (PEVNT_TRGSEL12, PEVNT_TRGSEL34)	280
11.3.8	Timer0 硬件触发事件选择寄存器 (TMR0_TRGSEL)	280
11.3.9	TimerA 硬件触发事件选择寄存器 (TMRA_TRGSELx, x=0~3)	281
11.3.10	ADC1 转换开始片内触发源选择寄存器 (ADC1_TRGSELx, x=0~1)	281
11.3.11	ADC2 转换开始片内触发源选择寄存器 (ADC2_TRGSELx, x=0~1)	282
11.3.12	ADC3 转换开始片内触发源选择寄存器 (ADC3_TRGSELx, x=0, 1)	282
11.3.13	公共触发源选择寄存器 (AOS_COMTRGSELx, x=1~2)	283
12	存储保护单元 (MPU)	284
12.1	概述	284
12.2	功能说明	284

12.2.1	区域范围设置	284
12.2.2	权限设置	284
12.2.3	MPU 动作选择.....	285
12.2.4	启动 MPU.....	285
12.3	应用举例	285
12.3.1	只允许部分空间访问.....	285
12.3.2	只禁止部分空间访问.....	286
12.4	寄存器描述.....	287
12.4.1	区域范围描述寄存器 (MPU_RGDn, n=0~15)	288
12.4.2	状态标志寄存器 (MPU_SR)	288
12.4.3	标志清除寄存器 (MPU_ECLR)	289
12.4.4	写保护寄存器 (MPU_WP)	289
12.4.5	IP 访问保护寄存器 (MPU_IPPR)	290
12.4.6	MSP 保护开始地址寄存器 (MPU_MSPPBA)	292
12.4.7	MSP 保护控制寄存器 (MPU_MSPPCTL)	292
12.4.8	PSP 保护开始地址寄存器 (MPU_PSPPBA)	293
12.4.9	PSP 保护控制寄存器 (MPU_PSPPCTL)	293
12.4.10	SMPU1 区域使能寄存器 (MPU_S1RGE)	294
12.4.11	SMPU1 区域写权限寄存器 (MPU_S1RGWP)	295
12.4.12	SMPU1 区域读权限寄存器 (MPU_S1RGRP)	296
12.4.13	SMPU1 控制寄存器 (MPU_S1CR)	297
13	内部时钟校准器 (CTC)	298
13.1	概述	298
13.2	主要特性	298
13.3	功能说明	298
13.3.1	CTC 框图	298
13.3.2	参考时钟	299
13.3.3	频率校准	303
13.4	中断	304
13.5	应用举例	304
13.6	寄存器描述.....	306
13.6.1	时钟校准控制寄存器 1 (CTC_CR1)	307
13.6.2	时钟校准控制寄存器 2 (CTC_CR2)	309
13.6.3	时钟校准状态寄存器 (CTC_STR)	310
13.6.4	时钟校准计数器 (CTC_CNT)	310

14 DMA 控制器 (DMA)	311
14.1 概述	311
14.2 主要特性	311
14.3 功能说明	312
14.3.1 DMA 框图	312
14.3.2 使能 DMA 控制器	312
14.3.3 通道选择和通道优先级	312
14.3.4 启动 DMA	313
14.3.5 数据块	313
14.3.6 传输地址控制	313
14.3.7 传输次数	313
14.3.8 连锁传输	314
14.3.9 不连续地址传输	316
14.3.10 通道重置	317
14.3.11 传输提前终止	318
14.4 中断	319
14.5 应用举例	319
14.5.1 存储器到存储器的传输	319
14.5.2 存储器到外围电路的传输	320
14.5.3 存储器到存储器的连锁传输	322
14.6 寄存器描述	324
14.6.1 DMA 使能寄存器 (DMA_EN)	325
14.6.2 中断状态寄存器 0 (DMA_INTSTAT0)	325
14.6.3 中断状态寄存器 1 (DMA_INTSTAT1)	326
14.6.4 中断屏蔽寄存器 0 (DMA_INTMASK0)	326
14.6.5 中断屏蔽寄存器 1 (DMA_INTMASK1)	327
14.6.6 中断复位寄存器 0 (DMA_INTCLR0)	327
14.6.7 中断复位寄存器 1 (DMA_INTCLR1)	328
14.6.8 通道使能寄存器 (DMA_CHEN)	328
14.6.9 通道使能复位寄存器 (DMA_CHENCLR)	329
14.6.10 通道重置控制寄存器 (DMA_RCFGCTL)	330
14.6.11 软件启动寄存器 (DMA_SWREQ)	332
14.6.12 传输请求状态寄存器 (DMA_REQSTAT)	333
14.6.13 通道状态观测寄存器 (DMA_CHSTAT)	333
14.6.14 传输源地址寄存器 (DMA_SARx, x=0~7)	334

14.6.15	传输目标地址寄存器 (DMA_DARx, x=0~7)	334
14.6.16	数据控制寄存器 (DMA_DTCTLx, x=0~7)	335
14.6.17	重复区域大小寄存器 (DMA_RPTx, x=0~7)	336
14.6.18	重复区域大小寄存器 B (DMA_RPTBx, x=0~7)	337
14.6.19	源地址不连续传输控制寄存器 (DMA_SNSEQCTLx, x=0~7)	338
14.6.20	源地址不连续传输控制寄存器 B (DMA_SNSEQCTLBx, x=0~7)	339
14.6.21	目标地址不连续传输控制寄存器 (DMA_DNSEQCTLx, x=0~7)	340
14.6.22	目标地址不连续传输控制寄存器 B (DMA_DNSEQCTLBx, x=0~7)	341
14.6.23	链指针寄存器 (DMA_LLPx, x=0~7)	342
14.6.24	通道控制寄存器 (DMA_CHCTLx, x=0~7)	343
14.6.25	传输源地址监视寄存器 (DMA_MONSARx, x=0~7)	344
14.6.26	传输目标地址监视寄存器 (DMA_MONDARx, x=0~7)	344
14.6.27	数据控制监视寄存器 (DMA_MONDTCTLx, x=0~7)	345
14.6.28	重复区域计数器监视寄存器 (DMA_MONRPTx, x=0~7)	345
14.6.29	源地址不连续传输计数器监视寄存器 (DMA_MONSSEQCTLx, x=0~7)	346
14.6.30	目标地址不连续传输计数器监视寄存器 (DMA_MONDNSEQCTLx, x=0~7)	346
14.7	注意事项	347
15	电压比较器 (CMP)	348
15.1	概述	348
15.2	主要特性	348
15.3	功能说明	349
15.3.1	CMP 功能框图	349
15.3.2	输入选择	350
15.3.3	普通比较	350
15.3.4	窗口比较	351
15.3.5	输入扫描	352
15.3.6	窗口输出	354
15.3.7	迟滞功能	356
15.3.8	噪声滤波器	357
15.3.9	外部管脚输出	357
15.4	中断和事件	357
15.5	寄存器描述	358
15.5.1	比较器工作模式寄存器 (CMPx_MDR, x=1~3)	359
15.5.2	比较器滤波和中断寄存器 (CMPx_FIR, x=1~3)	360
15.5.3	比较器输出控制寄存器 (CMPx_OCR, x=1~3)	361

15.5.4	比较器结果监视寄存器 (CMPx_OMR, x=1~3)	361
15.5.5	比较器正负端输入选择寄存器 (CMPx_PMSR, x=1~3)	362
15.5.6	比较器输出空白窗口设定寄存器 1 (CMPx_BWSR1, x=1~3)	363
15.5.7	比较器输出空白窗口设定寄存器 2 (CMPx_BWSR2, x=1~3)	364
15.5.8	比较器输入扫描控制寄存器 (CMPx_SCCR, x=1~3)	365
15.5.9	比较器输入扫描监视寄存器 (CMPx_SCMR, x=1~3)	366
15.6	注意事项	367
15.6.1	模块停止功能	367
15.6.2	模块停止时的功耗	367
15.6.3	停止低功耗模式时的功耗	367
16	模数转换模块 (ADC)	368
16.1	概述	368
16.2	主要特性	368
16.3	功能说明	369
16.3.1	ADC 系统框图	369
16.3.2	ADC 时钟	370
16.3.3	通道选择	370
16.3.4	触发源选择	372
16.3.5	序列 A 单次扫描模式	372
16.3.6	序列 A 连续扫描模式	373
16.3.7	序列 A 数据缓冲模式	374
16.3.8	双序列扫描模式	376
16.3.9	模拟看门狗功能	378
16.3.10	模拟输入的采样时间和转换时间	379
16.3.11	ADC 数据寄存器自动清除功能	380
16.3.12	转换数据平均计算功能	380
16.3.13	过采样	381
16.3.14	多 ADC 协同工作模式	381
16.4	中断和事件	387
16.5	寄存器描述	388
16.5.1	ADC 启动寄存器 (ADC_STR)	389
16.5.2	ADC 控制寄存器 0 (ADC_CR0)	390
16.5.3	ADC 控制寄存器 1 (ADC_CR1)	391
16.5.4	ADC 控制寄存器 2 (ADC_CR2)	392
16.5.5	ADC 转换开始触发寄存器 (ADC_TRGSR)	393

16.5.6	ADC 通道选择寄存器 A (ADC_CHSELRA)	394
16.5.7	ADC 通道选择寄存器 B (ADC_CHSELRB)	395
16.5.8	ADC 平均通道选择寄存器 (ADC_AVCHSELR)	396
16.5.9	ADC 扩展通道选择寄存器 (ADC_EXCHSELR)	396
16.5.10	ADC 采样状态寄存器 (ADC_SSTRx)	397
16.5.11	ADC 通道映射控制寄存器 (ADC_CHMUXR)	398
16.5.12	ADC 中断状态寄存器 (ADC_ISR)	399
16.5.13	ADC 中断许可寄存器 (ADC_ICR)	399
16.5.14	ADC 中断状态复位寄存器 (ADC_ISCLRR)	400
16.5.15	ADC 协同模式控制寄存器 (ADC_SYNCCR)	401
16.5.16	ADC 数据寄存器 (ADC_DRx)	402
16.5.17	模拟看门狗控制寄存器 (ADC_AWDCR)	403
16.5.18	模拟看门狗状态寄存器 (ADC_AWDSR)	404
16.5.19	模拟看门狗状态复位寄存器 (ADC_AWDSCLR)	404
16.5.20	模拟看门狗阈值寄存器 (ADC_AWD0DR0、ADC_AWD0DR1、ADC_AWD1DR0、ADC_AWD1DR1)	405
16.5.21	模拟看门狗比较通道选择寄存器 (ADC_AWD0CHSR、ADC_AWD1CHSR)	405
16.6	注意事项	406
16.6.1	数据寄存器读取时注意事项	406
16.6.2	扫描完成中断处理注意事项	406
16.6.3	模块停止与低功耗设定的注意事项	406
16.6.4	ADC 转换模拟通道输入的引脚设定	406
16.6.5	噪声控制	406
17	数模转换器 (DAC)	407
17.1	概述	407
17.2	主要特性	407
17.3	功能说明	407
17.3.1	DAC 功能框图	407
17.3.2	DAC 转换	408
17.3.3	同步转换	408
17.3.4	ADC 转换优先模式	409
17.3.5	数据传送模式	410
17.4	寄存器描述	411
17.4.1	DAC 数据寄存器 (DADRy, y=1~2)	412
17.4.2	DAC 控制寄存器 (DACR)	413

17.4.3	DAC 模拟输出控制寄存器 (DAOOCR)	414
17.4.4	DAC ADC 转换优先控制寄存器 (DAADPCR)	415
17.4.5	DAC 控制寄存器 2 (DACR2)	416
17.4.6	DAC 数据 ACTIVE 值寄存器 (DADACTR _y , y=1~2)	416
17.5	注意事项	417
17.5.1	模块停止功能的设置	417
17.5.2	模块停止状态下的 DAC 动作	417
17.5.3	停止模式下的 DAC 动作	417
17.5.4	掉电模式下的 DAC 动作	417
17.5.5	用输出放大器的注意事项	417
18	高级控制定时器 (Timer6)	418
18.1	概述	418
18.2	主要特性	418
18.3	功能说明	419
18.3.1	Timer6 框图	419
18.3.2	波形模式	420
18.3.3	时钟源选择	420
18.3.4	计数方向	420
18.3.5	比较输出	421
18.3.6	捕获输入	422
18.3.7	计数器刷新	423
18.3.8	软件同步	424
18.3.9	硬件同步	425
18.3.10	脉宽测量	427
18.3.11	周期测量	427
18.3.12	缓存功能	428
18.3.13	数字滤波	432
18.3.14	通用 PWM 输出	433
18.3.15	周期间隔响应	440
18.3.16	正交编码计数	441
18.3.17	EMB 控制	446
18.3.18	功能汇总表	446
18.4	中断和事件	448
18.4.1	中断	448
18.4.2	事件	448

18.5	应用举例	450
18.5.1	基本计数及中断动作	450
18.5.2	比较输出及中断动作	450
18.5.3	捕获输入及中断动作	451
18.5.4	缓存传送动作（周期基准值）	451
18.5.5	缓存传送动作（通用比较基准值）	452
18.5.6	缓存传送动作（专用比较基准值）	452
18.5.7	缓存传送动作（死区基准值）	453
18.5.8	同步启动动作（软件方式）	453
18.5.9	同步启动动作（硬件方式）	453
18.5.10	正交编码计数动作（2相）	454
18.5.11	正交编码计数动作（3相）	454
18.5.12	单路 PWM 输出	455
18.5.13	互补 PWM 输出（软件死区）	455
18.5.14	互补 PWM 输出（硬件死区）	455
18.5.15	EMB 监控及中断动作	456
18.6	寄存器描述	457
18.6.1	通用计数值寄存器（TMR6_CNTER）	459
18.6.2	通用刷新值寄存器（TMR6_UPDAR）	459
18.6.3	通用周期基准值寄存器（TMR6_PERmR, m=A~C）	459
18.6.4	通用比较基准值寄存器（TMR6_GCMmR, m=A~F）	460
18.6.5	专用比较基准值寄存器（TMR6_SCMmR, m=A~F）	460
18.6.6	死区时间基准值寄存器（TMR6_DTmnR, m=D、U, n=A、B）	460
18.6.7	通用控制寄存器（TMR6_GCONR）	461
18.6.8	中断控制寄存器（TMR6_ICONR）	462
18.6.9	缓存控制寄存器（TMR6_BCONR）	463
18.6.10	死区控制寄存器（TMR6_DCONR）	465
18.6.11	端口控制寄存器 A（TMR6_PCNAR）	466
18.6.12	端口控制寄存器 B（TMR6_PCNBR）	468
18.6.13	滤波控制寄存器（TMR6_FCNGR）	470
18.6.14	有效周期寄存器（TMR6_VPERR）	471
18.6.15	状态标志寄存器（TMR6_STFLR）	472
18.6.16	硬件启动事件选择寄存器（TMR6_HSTAR）	474
18.6.17	硬件停止事件选择寄存器（TMR6_HSTPR）	476
18.6.18	硬件清零事件选择寄存器（TMR6_HCLRR）	478

18.6.19	硬件刷新事件选择寄存器 (TMR6_HUPDR)	480
18.6.20	硬件捕获事件选择寄存器 A (TMR6_HCPAR)	482
18.6.21	硬件捕获事件选择寄存器 B (TMR6_HCPBR)	484
18.6.22	硬件递加事件选择寄存器 (TMR6_HCUPR)	486
18.6.23	硬件递减事件选择寄存器 (TMR6_HCDOR)	488
18.6.24	公共滤波控制寄存器 (TMR6_FCNTR)	490
18.6.25	公共软件同步启动控制寄存器 (TMR6_SSTAR)	491
18.6.26	公共软件同步停止控制寄存器 (TMR6_SSTPR)	491
18.6.27	公共软件同步清零控制寄存器 (TMR6_SCLRR)	492
18.6.28	公共软件同步刷新控制寄存器 (TMR6_SUPDR)	492
18.7	注意事项	493
19	高精度 PWM (HRPWM)	495
19.1	概述	495
19.2	基本框图	497
19.3	功能说明	498
19.3.1	HRWPM 框图	498
19.3.2	波形模式	500
19.3.3	计数器时钟	500
19.3.4	校准功能	501
19.3.5	计数方向	501
19.3.6	捕获功能	502
19.3.7	PWM 动作输出	503
19.3.8	死区模块	506
19.3.9	空闲控制	508
19.3.10	相位功能	516
19.3.11	外部事件	518
19.3.12	软件同步	524
19.3.13	硬件同步	525
19.3.14	同步输入端口数字滤波	526
19.3.15	同步输出	526
19.3.16	缓存功能	527
19.3.17	通用 PWM 输出	534
19.3.18	周期间隔响应	543
19.3.19	EMB 控制	544
19.3.20	输出管理	544

19.3.21	DAC 触发源	546
19.3.22	功能汇总表	548
19.4	中断	549
19.4.1	中断输出	549
19.4.2	事件输出	550
19.5	典型应用例	553
19.5.1	校准功能	553
19.5.2	基本计数及中断动作	553
19.5.3	比较输出及中断动作	553
19.5.4	间隔输出及中断动作	554
19.5.5	缓存传送动作 (周期基准值)	554
19.5.6	缓存传送动作 (通用比较基准值)	555
19.5.7	缓存传送动作 (专用比较基准值)	555
19.5.8	缓存传送动作 (外部事件滤波偏移值)	555
19.5.9	缓存传送动作 (外部事件滤波窗口)	556
19.5.10	缓存传送动作 (间隔输出周期基准值)	556
19.5.11	缓存传送动作 (间隔输出比较基准值)	557
19.5.12	缓存传送动作 (死区基准值)	557
19.5.13	同步启动动作 (软件方式)	557
19.5.14	同步启动动作 (硬件方式)	558
19.5.15	同步输出	558
19.5.16	单路 PWM 输出	558
19.5.17	互补 PWM 输出 (软件死区)	558
19.5.18	互补 PWM 输出 (硬件死区)	559
19.5.19	EMB 监控及中断动作	559
19.6	寄存器说明	561
19.6.1	HRPWM 单元 1~6 寄存器	565
19.6.2	HRPWM 公共寄存器	644
19.7	注意事项	672
20	通用控制定时器 (Timer4)	674
20.1	概述	674
20.2	主要特性	674
20.3	功能说明	675
20.3.1	Timer4 框图	675
20.3.2	基本动作	676

20.3.3	缓存功能	680
20.3.4	通用 PWM 输出	686
20.3.5	周期间隔响应	690
20.3.6	EMB 控制	692
20.3.7	监测输出	693
20.3.8	硬件启动	695
20.3.9	同步启动	695
20.4	中断和事件	695
20.4.1	通用计数比较匹配中断及事件	695
20.4.2	计数周期匹配中断及事件	695
20.4.3	重载计数匹配中断及事件	695
20.4.4	专用比较匹配中断及事件	696
20.5	寄存器描述	697
20.5.1	计数值寄存器 (TMR4_CNTR)	700
20.5.2	周期基准寄存器 (TMR4_CPSR)	700
20.5.3	控制状态寄存器 (TMR4_CCSR)	701
20.5.4	有效周期寄存器 (TMR4_CVPR)	703
20.5.5	通用比较基准寄存器 (TMR4_OCCRm)	703
20.5.6	通用控制状态寄存器 (TMR4_OCSRn)	704
20.5.7	通用扩展控制寄存器 (TMR4_OCERn)	705
20.5.8	通用模式控制寄存器 (TMR4_OCMRm)	707
20.5.9	专用比较基准寄存器 (TMR4_SCCRm)	712
20.5.10	专用控制状态寄存器 (TMR4_SCSRm)	713
20.5.11	专用扩展控制寄存器 (TMR4_SCER)	714
20.5.12	专用模式控制寄存器 (TMR4_SCMRm)	715
20.5.13	PWM 基本控制寄存器 (TMR4_POCRn)	716
20.5.14	PWM 状态控制寄存器 (TMR4_PSCR)	717
20.5.15	PWM 滤波控制寄存器 (TMR4_PFSRn)	720
20.5.16	PWM 死区控制寄存器 (TMR4_PDARn)	720
20.5.17	重载控制状态寄存器 (TMR4_RCSR)	721
20.5.18	专用中断控制寄存器 (TMR4_SCIR)	723
20.5.19	专用标志控制寄存器 (TMR4_SCFR)	724
21	紧急刹车模块 (EMB)	725
21.1	概述	725
21.2	功能说明	725

21.2.1	EMB 结构框图.....	727
21.2.2	外部端口输入电平变化时控制 PWM 信号输出	728
21.2.3	PWM 输出端口电平发生同相（同高或同低）时停止 PWM 信号输出	729
21.2.4	根据电压比较器比较结果停止 PWM 信号输出	730
21.2.5	系统错误发生时停止 PWM 信号输出	730
21.2.6	写寄存器软件控制 PWM 信号输出	731
21.3	寄存器描述.....	732
21.3.1	EMB 控制寄存器 1 (EMB_CTL1, Group0~5)	733
21.3.2	EMB 控制寄存器 1 (EMB_CTL1, Group6~7)	735
21.3.3	EMB 控制寄存器 1 (EMB_CTL1, Group8)	737
21.3.4	EMB 控制寄存器 2 (EMB_CTL2, Group0~5)	739
21.3.5	EMB 控制寄存器 2 (EMB_CTL2, Group6~7)	742
21.3.6	EMB 控制寄存器 2 (EMB_CTL2, Group8)	745
21.3.7	EMB 软件输出使能控制寄存器 (EMB_SOE)	748
21.3.8	EMB 状态寄存器 (EMB_STAT)	749
21.3.9	EMB 状态复位寄存器 (EMB_STATCLR)	751
21.3.10	EMB 中断许可寄存器 (EMB_INTEN)	753
21.3.11	EMB 控制 PWM 输出释放方式选择寄存器 (EMB_RLSSEL)	754
22	通用定时器 (TimerA)	755
22.1	概述	755
22.2	主要特性	755
22.3	功能说明	756
22.3.1	TimerA 框图	756
22.3.2	对称单元	757
22.3.3	波形模式	757
22.3.4	时钟源选择	757
22.3.5	比较输出	758
22.3.6	捕获输入	759
22.3.7	同步启动	760
22.3.8	数字滤波	761
22.3.9	缓存功能	762
22.3.10	级联计数	763
22.3.11	PWM 输出	764
22.3.12	正交编码计数	765
22.4	中断和事件	770

22.4.1	比较匹配中断及事件	770
22.4.2	周期匹配中断及事件	770
22.5	寄存器描述	771
22.5.1	通用计数值寄存器 (TMRA_CNTER)	773
22.5.2	周期基准值寄存器 (TMRA_PERAR)	774
22.5.3	比较基准值寄存器 (TMRA_CMPAR _m , m=1~4)	774
22.5.4	控制状态寄存器 L (TMRA_BCSTRL)	775
22.5.5	控制状态寄存器 H (TMRA_BCSTRH)	776
22.5.6	中断控制寄存器 (TMRA_ICONR)	776
22.5.7	事件控制寄存器 (TMRA_ECONR)	777
22.5.8	滤波控制寄存器 (TMRA_FCONR)	778
22.5.9	状态标志寄存器 (TMRA_STFLR)	779
22.5.10	缓存控制寄存器 (TMRA_BCONR _m , m=1~4)	780
22.5.11	捕获控制寄存器 (TMRA_CCONR _m , m=1~4)	781
22.5.12	端口控制寄存器 (TMRA_PCONR _m , m=1~4)	783
22.5.13	硬件触发事件选择寄存器 (TMRA_HCONR)	785
22.5.14	硬件递加事件选择寄存器 (TMRA_HCUPR)	787
22.5.15	硬件递减事件选择寄存器 (TMRA_HCDOR)	789
23	通用定时器 (Timer0)	791
23.1	概述	791
23.2	功能说明	791
23.2.1	Timer0 框图	791
23.2.2	时钟源选择	792
23.2.3	基本计数	792
23.2.4	硬件触发	793
23.2.5	软件同步启动	793
23.3	中断和事件	793
23.3.1	中断	793
23.3.2	事件	794
23.4	寄存器描述	795
23.4.1	计数值寄存器 (TMR0_CNTMR, m=A~B)	796
23.4.2	基准值寄存器 (TMR0_CMPmR, m=A~B)	796
23.4.3	基本控制寄存器 (TMR0_BCONR)	797
23.4.4	状态标志寄存器 (TMR0_STFLR)	800
23.4.5	计数同步使能寄存器 (TMR_SYNENR)	801

23.5	注意事项	803
24	实时时钟 (RTC)	804
24.1	简介	804
24.2	主要特性	804
24.3	功能说明	805
24.3.1	RTC 框图	805
24.3.2	上电设定	805
24.3.3	RTC 计数开始设定	805
24.3.4	系统低功耗模式切换	806
24.3.5	读出计数寄存器	806
24.3.6	写入计数寄存器	806
24.3.7	闹钟设定	807
24.3.8	时钟误差补偿	807
24.3.9	1Hz 输出	807
24.4	中断	809
24.4.1	闹钟中断	809
24.4.2	定周期中断	809
24.5	寄存器描述	810
24.5.1	控制寄存器 0 (RTC_CR0)	811
24.5.2	控制寄存器 1 (RTC_CR1)	812
24.5.3	控制寄存器 2 (RTC_CR2)	813
24.5.4	控制寄存器 3 (RTC_CR3)	814
24.5.5	秒计数寄存器 (RTC_SEC)	814
24.5.6	分计数寄存器 (RTC_MIN)	815
24.5.7	时计数寄存器 (RTC_HOUR)	816
24.5.8	日计数寄存器 (RTC_DAY)	818
24.5.9	周计数寄存器 (RTC_WEEK)	819
24.5.10	月计数寄存器 (RTC_MON)	820
24.5.11	年计数寄存器 (RTC_YEAR)	820
24.5.12	分闹钟寄存器 (RTC_ALMMIN)	820
24.5.13	时闹钟寄存器 (RTC_ALMHOUR)	821
24.5.14	周闹钟寄存器 (RTC_ALMWEK)	821
24.5.15	时钟误差补偿寄存器 (RTC_ERRCRH、RTC_ERRCRL)	822
25	看门狗计数器 (WDT/ SWDT)	824
25.1	简介	824

25.2	功能说明	825
25.2.1	启动看门狗	825
25.2.2	硬件启动方式	825
25.2.3	软件启动方式	826
25.2.4	刷新动作	827
25.2.5	标志位	827
25.2.6	中断复位	827
25.2.7	计数下溢	828
25.2.8	刷新错误	829
25.3	寄存器描述	830
25.3.1	控制寄存器 (SWDT_CR、WDT_CR)	831
25.3.2	状态寄存器 (SWDT_SR、WDT_SR)	833
25.3.3	刷新寄存器 (SWDT_RR、WDT_RR)	833
25.4	注意事项	834
26	通用同步异步收发器 (USART)	835
26.1	简介	835
26.2	主要特性	835
26.3	功能说明	837
26.3.1	USART 系统框图	837
26.3.2	UART 功能	838
26.3.3	多处理器通信	850
26.3.4	UART-LIN	854
26.3.5	智能卡	857
26.3.6	时钟同步模式	861
26.3.7	数字滤波功能	867
26.3.8	中断	868
26.4	寄存器描述	869
26.4.1	状态寄存器 (USART_SR)	870
26.4.2	数据寄存器 (USART_TDR)	874
26.4.3	数据寄存器 (USART_RDR)	874
26.4.4	波特率寄存器 (USART_BRR)	875
26.4.5	控制寄存器 1 (USART_CR1)	877
26.4.6	控制寄存器 2 (USART_CR2)	881
26.4.7	控制寄存器 3 (USART_CR3)	884
26.4.8	预分频寄存器 (USART_PR)	886

26.4.9	LIN 波特率测量计数寄存器 (USART_LBMC)	887
26.4.10	USART1 滤波控制寄存器 (USART1_NFC)	887
26.5	注意事项	888
26.5.1	UART 注意事项	888
26.5.2	时钟同步模式注意事项	888
26.5.3	其他注意事项	888
27	集成电路总线 (I2C)	889
27.1	概述	889
27.2	主要特性	889
27.3	功能说明	890
27.3.1	I2C 系统框图	890
27.3.2	I2C 协议	892
27.3.3	地址匹配	901
27.3.4	SMBus 动作	907
27.3.5	复位	908
27.3.6	可编程数字滤波	908
27.4	中断和事件	909
27.5	应用举例	910
27.5.1	I2C 初始化流程	910
27.6	寄存器描述	911
27.6.1	I2C 控制寄存器 1 (I2C_CR1)	912
27.6.2	I2C 控制寄存器 2 (I2C_CR2)	914
27.6.3	I2C 控制寄存器 3 (I2C_CR3)	916
27.6.4	I2C 控制寄存器 4 (I2C_CR4)	917
27.6.5	I2C 从机地址寄存器 0 (I2C_SLR0)	918
27.6.6	I2C 从机地址寄存器 1 (I2C_SLR1)	919
27.6.7	I2C SCL 电平超时控制寄存器 (I2C_SLTR)	920
27.6.8	I2C 状态寄存器 (I2C_SR)	921
27.6.9	I2C 状态清零寄存器 (I2C_CLR)	926
27.6.10	I2C 数据发送寄存器 (I2C_DTR)	927
27.6.11	I2C 数据接收寄存器 (I2C_DRR)	928
27.6.12	I2C 数据移位寄存器 (I2C_DSR)	928
27.6.13	I2C 时钟控制寄存器 (I2C_CCR)	929
27.6.14	I2C 滤波控制寄存器 (I2C_FLTR)	931
27.6.15	I2C FIFO 控制器寄存器 (I2C_FSTR)	932

27.6.16	I2C 从机地址接收器寄存器 (I2C_SLVADRR)	933
28	可编程逻辑阵列 (PLA)	934
28.1	概述	934
28.2	功能说明	934
28.2.1	PLA 框图	934
28.2.2	动作说明	935
28.2.3	PLA 输入多路选择器	936
28.2.4	查找表 (LUT) 控制	938
28.2.5	PLA 输出	939
28.3	寄存器说明	940
28.3.1	全局控制寄存器 (PLA_GCTL)	941
28.3.2	状态寄存器 (PLA_STAT)	942
28.3.3	多路选择器选择寄存器 (PLAx_MUXS, x=0~15)	943
28.3.4	查找表控制寄存器 (PLAx_LUTCTL, x=0~15)	944
28.3.5	控制寄存器 (PLAx_CTL, x=0~15)	944
28.4	注意事项	945
29	串行外设接口 (SPI)	946
29.1	简介	946
29.2	SPI 主要特性	946
29.3	SPI 功能说明	947
29.3.1	SPI 系统框图	947
29.3.2	SPI 动作系统说明	948
29.3.3	数据通信说明	950
29.3.4	运行说明	958
29.3.5	奇偶校验位自诊断	965
29.3.6	错误检测	966
29.3.7	SPI 的初始化	969
29.4	中断和事件	970
29.4.1	中断	970
29.4.2	事件	971
29.5	寄存器说明	972
29.5.1	SPI 数据寄存器 (SPI_DR)	973
29.5.2	SPI 控制寄存器 (SPI_CR)	974
29.5.3	SPI 通信配置寄存器 1 (SPI_CFG1)	975
29.5.4	SPI 状态寄存器 (SPI_SR)	977

29.5.5	SPI 通信配置寄存器 2 (SPI_CFG2)	978
30	控制器局域网 (MCAN)	980
30.1	概述	980
30.2	主要特性	980
30.3	功能说明	981
30.3.1	MCAN 系统框图	981
30.3.2	引脚说明	983
30.3.3	工作模式	983
30.3.4	时间戳生成	991
30.3.5	超时计数器	991
30.3.6	接收处理	992
30.3.7	发送处理	999
30.3.8	FIFO 确认处理	1004
30.3.9	消息 RAM	1005
30.4	寄存器描述	1014
30.4.1	MCAN 字节序寄存器 (MCAN_ENDN)	1016
30.4.2	MCAN 数据位时间和预分频寄存器 (MCAN_DBTP)	1017
30.4.3	MCAN 测试寄存器 (MCAN_TEST)	1018
30.4.4	MCAN RAM 看门狗寄存器 (MCAN_RWD)	1019
30.4.5	MCAN CC 控制寄存器 (MCAN_CCCR)	1020
30.4.6	MCAN 标称位时间和预分频寄存器 (MCAN_NBTP)	1022
30.4.7	MCAN 时间戳计数器配置寄存器 (MCAN_TSCC)	1023
30.4.8	MCAN 时间戳计数器值寄存器 (MCAN_TSCV)	1024
30.4.9	MCAN 超时计数器配置寄存器 (MCAN_TOCC)	1025
30.4.10	MCAN 超时计数器值寄存器 (MCAN_TOCV)	1026
30.4.11	MCAN 错误计数器寄存器 (MCAN_ECR)	1027
30.4.12	MCAN 协议状态寄存器 (MCAN_PSR)	1028
30.4.13	MCAN 发送延迟补偿寄存器 (MCAN_TDCR)	1030
30.4.14	MCAN 中断寄存器 (MCAN_IR)	1031
30.4.15	MCAN 中断使能寄存器 (MCAN_IE)	1034
30.4.16	MCAN 中断线选择寄存器 (MCAN_ILS)	1037
30.4.17	MCAN 中断线使能寄存器 (MCAN_ILE)	1040
30.4.18	MCAN 全局过滤器配置寄存器 (MCAN_GFC)	1041
30.4.19	MCAN 标准 ID 过滤器配置寄存器 (MCAN_SIDFC)	1042
30.4.20	MCAN 扩展 ID 过滤器配置寄存器 (MCAN_XIDFC)	1043

30.4.21	MCAN 扩展 ID 与掩码寄存器 (MCAN_XIDAM)	1043
30.4.22	MCAN 高优先级消息状态寄存器 (MCAN_HPMS)	1044
30.4.23	MCAN 新数据 1 寄存器 (MCAN_NDAT1)	1045
30.4.24	MCAN 新数据 2 寄存器 (MCAN_NDAT2)	1045
30.4.25	MCAN 接收 FIFO0 配置寄存器 (MCAN_RXF0C)	1046
30.4.26	MCAN 接收 FIFO0 状态寄存器 (MCAN_RXF0S)	1047
30.4.27	MCAN 接收 FIFO0 确认寄存器 (MCAN_RXF0A)	1048
30.4.28	MCAN 接收缓冲区配置寄存器 (MCAN_RXBC)	1048
30.4.29	MCAN 接收 FIFO1 配置寄存器 (MCAN_RXF1C)	1049
30.4.30	MCAN 接收 FIFO1 状态寄存器 (MCAN_RXF1S)	1050
30.4.31	MCAN 接收 FIFO1 确认寄存器 (MCAN_RXF1A)	1051
30.4.32	MCAN 接收缓冲区和 FIFO 元素大小配置寄存器 (MCAN_RXESC)	1052
30.4.33	MCAN 发送缓冲区配置寄存器 (MCAN_TXBC)	1053
30.4.34	MCAN 发送 FIFO/队列状态寄存器 (MCAN_TXFQS)	1054
30.4.35	MCAN 发送缓冲区元素大小配置寄存器 (MCAN_TXESC)	1055
30.4.36	MCAN 发送缓冲区请求挂起寄存器 (MCAN_TXBRP)	1056
30.4.37	MCAN 发送缓冲区添加请求寄存器 (MCAN_TXBAR)	1057
30.4.38	MCAN 发送缓冲区取消请求寄存器 (MCAN_TXBCR)	1058
30.4.39	MCAN 发送缓冲区发送已发生寄存器 (MCAN_TXBTO)	1059
30.4.40	MCAN 发送缓冲区取消已完成寄存器 (MCAN_TXBCF)	1059
30.4.41	MCAN 发送缓冲区发送中断使能寄存器 (MCAN_TXBTIE)	1060
30.4.42	MCAN 发送缓冲区取消已完成中断使能寄存器 (MCAN_TXBCIE)	1060
30.4.43	MCAN 发送事件 FIFO 配置寄存器 (MCAN_TXEFC)	1061
30.4.44	MCAN 发送事件 FIFO 状态寄存器 (MCAN_TXEFS)	1062
30.4.45	MCAN 发送事件 FIFO 确认寄存器 (MCAN_TXEFA)	1063
30.4.46	MCAN 消息 RAM 校验控制寄存器 (MCANRAM_CKCR)	1064
30.4.47	MCAN 消息 RAM 校验状态寄存器 (MCANRAM_CKSR)	1065
30.4.48	MCAN 消息 RAM 错误注入使能寄存器 (MCANRAM_EIEN)	1065
30.4.49	MCAN 消息 RAM 错误注入位使能寄存器 0 (MCANRA_EIBIT0)	1065
30.4.50	MCAN 消息 RAM 错误注入位使能寄存器 1 (MCANRAM_EIBIT1)	1066
30.4.51	MCAN 消息 RAM 校验错误地址寄存器 (MCANRAM_ECCERRADDR)	1066
30.5	注意事项	1067
30.5.1	CAN 使用注意事项	1067
30.5.2	CAN 总线抗干扰措施	1067
30.5.3	CAN 控制器噪声制约	1067

31 CRC 运算 (CRC)	1068
31.1 主要特性	1068
31.2 功能说明	1068
31.2.1 功能框图	1068
31.2.2 CRC16 编码模式	1069
31.2.3 CRC16 校验模式	1069
31.2.4 CRC32 编码模式	1069
31.2.5 CRC32 校验模式	1069
31.3 寄存器描述	1071
31.3.1 控制寄存器 (CRC_CR)	1072
31.3.2 结果寄存器 (CRC_RESLT)	1072
31.3.3 数据寄存器 (CRC_DAT)	1072
32 调试控制器 (DBGC)	1073
32.1 概述	1073
32.2 DBGC 系统框图	1074
32.3 SWJ-DP 调试端口 (SWD 和 JTAG)	1075
32.3.1 JTAG-DP 或 SW-DP 的切换机制	1076
32.4 引脚排列和调试端口引脚	1077
32.4.1 SWJ 调试端口引脚	1077
32.4.2 灵活的 SWJ-DP 引脚分配	1077
32.4.3 JTAG 引脚上的内部上拉	1078
32.4.4 使用串行接口以及释放未使用的调试引脚以作 GPIO	1078
32.5 寄存器说明	1079
32.5.1 芯片专属标志寄存器 (CHIPID)	1080
32.5.2 DBG 状态寄存器 (MCUDBGSTAT)	1080
32.5.3 外设调试暂停寄存器 (MCUSTPCTL)	1081
32.5.4 调试组件配置寄存器 (MCUTRACECTL)	1083
32.5.5 外设调试暂停寄存器 2 (MCUSTPCTL2)	1083
32.6 SW 调试端口	1084
32.6.1 SW 协议简介	1084
32.7 TPIU (跟踪端口接口单元)	1084
32.7.1 简介	1084
32.7.2 TRACE 引脚分配	1084
32.7.3 MCU 内部 TRACECLKIN 连接	1086
32.7.4 TPIU 寄存器	1086

32.7.5 TPIU 配置示例	1086
版本修订记录	1087

表索引

表 1-1	存储器映射	50
表 1-2	目标地址配置示例	56
表 1-3	MMF 基地址	57
表 1-4	寄存器列表	57
表 3-1	复位方式及产生条件	63
表 3-2	复位方式及复位标志	64
表 3-3	各模块复位条件	73
表 3-4	RMU 基地址	75
表 3-5	RMU 寄存器列表	75
表 4-1	时钟源主要特性	86
表 4-2	各个内部时钟的规格	87
表 4-3	CMU 寄存器列表	103
表 5-1	BOR 配置	131
表 5-2	PVD1/ PVD2 特性	132
表 5-3	运行模式	137
表 5-4	低功耗模式	137
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	137
表 5-6	掉电模式子模式	142
表 5-7	RAM 模块与 RAM 掉电控制位	145
表 5-8	寄存器保护列表	145
表 5-9	PWC 寄存器列表	146
表 6-1	ICG 基地址	170
表 6-2	ICG 寄存器列表	170
表 7-1	CPU 时钟频率和 Flash 读等待周期对照表	177
表 7-2	Flash 实际读周期数	178
表 7-3	OTP 地址分布表	184
表 7-4	EFM 基地址	187
表 7-5	EFM 寄存器列表	187
表 8-1	SRAM 空间分配	199
表 8-2	SRAM 基地址	201
表 8-3	SRAM 寄存器列表	201
表 9-1	GPIO 基地址	217
表 9-2	GPIO 寄存器	217

表 9-3	Event Port 基地址	217
表 9-4	Event Port 寄存器	217
表 9-5	32bit 访问时 PORT 寄存器一览	232
表 10-1	外部中断输入管脚	233
表 10-2	中断向量表	235
表 10-3	中断事件表	240
表 10-4	INTC 基地址	260
表 10-5	INTC 寄存器列表	260
表 11-1	AOS 目标列表	274
表 11-2	AOS 基地址	276
表 11-3	AOS 寄存器列表	276
表 12-1	MPU 基地址	287
表 12-2	MPU 寄存器列表	287
表 13-1	HRC 目标频率为 20MHz 时的测量误差	299
表 13-2	HRC 目标频率为 16MHz 时的测量误差	301
表 13-3	CTC 基地址	306
表 13-4	CTC 寄存器列表	306
表 14-1	通道重置说明	317
表 14-2	DMA 基地址	324
表 14-3	DMA 寄存器列表	324
表 15-1	CMP 引脚一览表	349
表 15-2	CMP 模拟输入一览表	350
表 15-3	CMP 基地址	358
表 15-4	CMP 寄存器列表	358
表 15-5	定时器窗口 PWM 一览表	363
表 16-1	各 ADC 单元规格	369
表 16-2	序列 A 和 B 的各种竞争	376
表 16-3	AD 转换时间	380
表 16-4	ADC 基地址	388
表 16-5	ADC 寄存器列表	388
表 17-1	DAC 引脚	408
表 17-2	DAC 基地址	411
表 17-3	DAC 寄存器列表	411
表 17-4	DAC 转换和模拟输出控制	414
表 18-1	Timer6 的基本功能及特性	418

表 18-2	Timer6 端口列表.....	419
表 18-3	不同模式下的功能对比表	446
表 18-4	Timer6 基地址	457
表 18-5	Timer6 寄存器列表	457
表 18-6	Timer6 公共寄存器列表	458
表 18-7	计数器 (CNTER) 控制优先级	493
表 18-8	PWMA 端口输出控制优先级	493
表 18-9	PWMB 端口输出控制优先级	494
表 19-1	HRPWM 的基本功能及特性	497
表 19-2	HRPWM 端口列表	499
表 19-3	HRPWM 分辨率	500
表 19-4	通道 B 跟随功能退出和进入空闲时刻	516
表 19-5	外部事件映射表	519
表 19-6	外部事件 EEysrc2 选则	520
表 19-7	HRPWM<t>_EEFLTCR1<2>.EEyFM 对应的消隐时间设定	522
表 19-8	窗口模式下滤波信号映射	524
表 19-9	HRPWM<t>_EEFLTCR1<2>.EEyFM 对应的窗口时间设定	524
表 19-10	支持缓存的寄存器.....	527
表 19-11	各单元支持的缓存传送点	529
表 19-12	各单元全局缓存状态标志置位条件	533
表 19-13	不同模式下的功能对比表	548
表 19-14	HRPWN 基地址	561
表 19-15	HRPWM 寄存器列表	561
表 19-16	HRPWM 公共寄存器列表	564
表 19-17	计数器 (CNTER) 控制优先级	672
表 19-18	PWMA、B 端口输出控制优先级.....	673
表 20-1	Timer4 的基本功能及特性.....	674
表 20-2	Timer4 端口列表.....	676
表 20-3	Timer4 基准地址.....	697
表 20-4	Timer4 寄存器列表	697
表 20-5	OCCR*h 计数匹配一览表.....	707
表 20-6	OCCR*h/l 计数匹配一览表.....	708
表 20-7	PWM 端口输出状态与寄存器设定值.....	719
表 21-1	端口分配表	728
表 21-2	端口分配表	728

表 21-3	EMB Group 对照表.....	729
表 21-4	EMB 基地址	732
表 21-5	EMB 寄存器列表	732
表 22-1	TimerA 的基本功能及特性.....	755
表 22-2	TimerA 端口列表.....	756
表 22-3	内部触发事件 TRGSEL 选择关系对应表	759
表 22-4	TimerA 基地址	771
表 22-5	TimerA 寄存器列表	771
表 23-1	Timer0/ TMR_SYNENR 基地址	795
表 23-2	Timer0 寄存器列表	795
表 23-3	TMR_SYNENR 寄存器列表	795
表 24-1	RTC 的基本规格	804
表 24-2	RTC 基地址	810
表 24-3	RTC 寄存器列表	810
表 25-1	看门狗计数器的基本特性	824
表 25-2	WDT/ SWDT 基地址	830
表 25-3	WDT 寄存器列表.....	830
表 25-4	SWDT 寄存器列表.....	830
表 26-1	USART 管脚说明	837
表 26-2	DIV_Fraction 为 0 时 UART 接收器的容差	843
表 26-3	DIV_Fraction 不为 0 时 UART 接收器的容差.....	843
表 26-4	UART 中断/事件表.....	849
表 26-5	多处理器模式中断/事件表	854
表 26-6	LIN 中断/事件表	856
表 26-7	智能卡模式中断/事件表.....	861
表 26-8	时钟同步模式中断/事件表	866
表 26-9	USART 整体中断一览表.....	868
表 26-10	USART 基地址.....	869
表 26-11	USART 寄存器列表	869
表 26-12	USART1_NFC 寄存器列表	869
表 26-13	波特率计算公式 (小数波特率无效 FBME=0)	875
表 26-14	波特率计算公式 (小数波特率有效 FBME=1)	876
表 27-1	输入/输出引脚.....	891
表 27-2	中断一览表	909
表 27-3	事件信号输出一览表	909

表 27-4	I2C 基地址	911
表 27-5	I2C 寄存器列表	911
表 28-1	PLAxMUX0 输入选择	936
表 28-2	PLAxMUX1 输入选择	937
表 28-3	LUT 控制	938
表 28-4	PLA 输出端口对应表	939
表 28-5	PLA 基地址	940
表 28-6	寄存器一览	940
表 29-1	SPI 的特性要点	946
表 29-2	管脚说明	947
表 29-3	主机模式时 SPI 管脚状态说明	948
表 29-4	从机模式时 SPI 管脚状态说明	948
表 29-5	部分设定值的位速	950
表 29-6	SPI 模式和寄存器设定关系	958
表 29-7	错误检测对应表	966
表 29-8	SPI 中断源说明	970
表 29-9	SPI 基地址	972
表 29-10	SPI 寄存器列表一览表	972
表 30-1	MCAN 引脚说明	983
表 30-2	MCAN 配置寄存器列表	983
表 30-3	MCAN 中的 DLC 编码	985
表 30-4	接收缓冲器和 FIFO 元素大小	997
表 30-5	接收缓冲区过滤器配置示例	998
表 30-6	调试消息过滤器配置示例	999
表 30-7	发送帧的可能配置	999
表 30-8	发送缓冲区、FIFO (队列) 元素大小	1001
表 30-9	接收缓冲区和 FIFO 元素	1006
表 30-10	接收缓冲区和 FIFO 元素说明	1006
表 30-11	发送缓冲区元素	1008
表 30-12	发送缓冲区元素说明	1008
表 30-13	发送事件 FIFO 元素	1010
表 30-14	发送事件 FIFO 元素说明	1010
表 30-15	标准消息 ID 过滤器元素	1011
表 30-16	标准消息 ID 过滤器元素说明	1011
表 30-17	扩展消息 ID 过滤器元素	1012

表 30-18	扩展消息 ID 过滤器元素说明	1013
表 30-19	MCAN/MCANRAM 基地址	1014
表 30-20	MCAN 寄存器列表	1014
表 30-21	MCANRAM 寄存器列表	1015
表 31-1	CRC 基地址	1071
表 31-2	CRC 寄存器列表	1071
表 32-1	SWJ 调试端口引脚	1077
表 32-2	灵活的 SWJ-DP 引脚分配	1077
表 32-3	DBGC 基地址	1079
表 32-4	DBGC 寄存器列表	1079

图索引

图 2-1	总线架构图	60
图 3-1	上电复位.....	65
图 3-2	NRST 复位时序	66
图 3-3	欠压复位.....	66
图 3-4	可编程电压检测 1 复位	67
图 3-5	可编程电压检测 2 复位	68
图 3-6	看门狗和专用看门狗复位	69
图 3-7	掉电唤醒复位.....	69
图 3-8	软件复位.....	70
图 3-9	MPU 错误复位.....	70
图 3-10	RAM 奇偶校验复位	70
图 3-11	RAMECC 复位	71
图 3-12	时钟频率异常复位.....	71
图 3-13	外部高速振荡异常复位.....	72
图 3-14	Cortex-M4 Lockup 复位	72
图 3-15	模块功能复位.....	73
图 4-1	时钟系统框图.....	84
图 4-2	时钟频率测量框图.....	85
图 4-3	外部高速振荡器连接事例	89
图 4-4	外部时钟输入的连接事例图.....	90
图 4-5	外部高速振荡器故障检测例.....	91
图 4-6	系统时钟选择 XTAL, 检测到 XTAL 振荡故障例	92
图 4-7	外部低速振荡器连接事例	94
图 4-8	时钟源切换	98
图 4-9	时钟分频切换.....	99
图 4-10	时钟频率测量时序图	101
图 5-1	电源构成图	130
图 5-2	上电复位、掉电复位波形	131
图 5-3	欠压复位波形.....	131
图 5-4	PVD1 中断/复位框图	132
图 5-5	PVD2 中断/复位框图	133
图 5-6	电压检测 1 中断时序图	133
图 5-7	电压检测 1 复位时序图	134

图 5-8 电压检测 2 中断运行时序图	135
图 5-9 电压检测 2 复位运行时序图	135
图 5-10 内部电压采样示意图	136
图 5-11 PTWKn 结构框图	144
图 7-1 Flash 地址分配	176
图 7-2 启动扇区交换功能 1	186
图 7-3 启动交换功能 2	186
图 8-1 错误注入功能示意图	200
图 9-1 端口基本结构示意图	213
图 10-1 中断系统框图	234
图 10-2 中断请求/事件输入选择	257
图 10-3 数字滤波器工作示意图	258
图 11-1 AOS 模块框图	273
图 12-1 只允许部分空间访问示意图	285
图 12-2 只禁止部分空间访问示意图	286
图 13-1 CTC 的基本框图	298
图 13-2 CTC 校准示意图	303
图 14-1 DMA 结构图	312
图 14-2 连锁传输示意图	315
图 14-3 不连续地址传输示意图 (以源地址为例)	316
图 14-4 不连续式重置示意图	318
图 14-5 应用例 1: 存储器到存储器传输	320
图 14-6 应用例 2: 存储器到外围电路的传输	321
图 15-1 CMP 功能框图	349
图 15-2 普通比较示意图	351
图 15-3 窗口比较示意图	352
图 15-4 输入扫描示意图	353
图 15-5 窗口输出示意图一	355
图 15-6 窗口输出示意图二	355
图 15-7 窗口输出示意图三	356
图 16-1 ADC 框图	369
图 16-2 通道映射示意图	371
图 16-3 内部模拟通道选择	372
图 16-4 序列 A 单次扫描模式	373
图 16-5 连续扫描	374

图 16-6	数据缓冲模式.....	375
图 16-7	双序列扫描模式（序列 A 从被中断通道重新启动）	377
图 16-8	双序列扫描模式（序列 A 从第一个通道重新启动）	377
图 16-9	模拟看门狗保护区（比较模式）	378
图 16-10	ADC 转换时间	379
图 16-11	平均功能有效时的转换动作.....	381
图 16-12	单次并行触发模式（三 ADC）	382
图 16-13	单次延迟触发模式（三 ADC）	384
图 16-14	循环并行触发模式（三 ADC）	385
图 16-15	循环延迟触发模式（两 ADC）	386
图 16-16	循环延迟触发模式（三 ADC）	386
图 16-17	ADC 中断和事件输出时序	387
图 17-1	DAC 转换通道框图	407
图 17-2	DAC 转换示意图.....	408
图 17-3	ADC 转换优先模式动作示意图	410
图 18-1	Timer6 基本框图.....	419
图 18-2	锯齿波波形（递加计数）	420
图 18-3	三角波波形	420
图 18-4	比较输出动作.....	422
图 18-5	捕获输入动作.....	423
图 18-6	硬件刷新动作.....	423
图 18-7	软件同步动作.....	425
图 18-8	硬件同步动作.....	426
图 18-9	脉宽测量.....	427
图 18-10	周期测量.....	428
图 18-11	单缓存方式比较输出时序	429
图 18-12	双缓存方式捕获输入时序	430
图 18-13	锯齿波模式时计数缓存动作.....	430
图 18-14	三角波模式时计数缓存动作 1.....	431
图 18-15	三角波模式时计数缓存动作 2.....	432
图 18-16	捕获输入端口的滤波功能	433
图 18-17	单边对齐独立 PWM	433
图 18-18	双边对称独立 PWM	434
图 18-19	软件设定 GCMBR 互补 PWM 波输出	435
图 18-20	硬件设定 GCMBR 互补 PWM 波输出	436

图 18-21	双边非对称 PWM 输出.....	437
图 18-22	8 相单边对齐独立 PWM.....	438
图 18-23	带死区时间 4 相双边对称互补 PWM	439
图 18-24	周期间隔有效请求信号动作.....	440
图 18-25	位置模式-基本计数.....	441
图 18-26	位置计数模式-相位差计数 (1 倍计数)	442
图 18-27	位置计数模式-相位差计数 (2 倍计数)	442
图 18-28	位置计数模式-相位差计数 (4 倍计数)	442
图 18-29	位置计数模式-方向计数.....	443
图 18-30	公转计数模式-Z 相计数	443
图 18-31	公转计数模式-位置溢出计数.....	444
图 18-32	公转计数模式-混合计数.....	444
图 18-33	公转计数模式-混合计数 Z 相屏蔽动作例 1.....	445
图 18-34	公转计数模式-混合计数 Z 相屏蔽动作例 2.....	445
图 18-35	锯齿波模式时中断&事件输出例.....	449
图 19-1	HRPWM 基本框图	498
图 19-2	HRPWM 各单元基本框图.....	499
图 19-3	锯齿波波形	500
图 19-4	三角波波形	500
图 19-5	三角波计数方向	501
图 19-6	捕获输入动作.....	502
图 19-7	PWM 动作输出模块.....	503
图 19-8	比较输出动作.....	504
图 19-9	半波模式有效 PWM 输出图例	505
图 19-10	锯齿波占空比限制 1.....	505
图 19-11	锯齿波占空比限制 2.....	505
图 19-12	三角波占空比限制 1.....	506
图 19-13	死区模块.....	506
图 19-14	硬件实现互补 PWM 波输出.....	507
图 19-15	脉宽异常时的死区模下的波形输出	507
图 19-16	空闲控制基本框图.....	508
图 19-17	立即空闲进入和退出	509
图 19-18	延迟空闲的进入和退出.....	510
图 19-19	间隔输出模式 1 时序图	512
图 19-20	间隔输出模式 2 时序图	512

图 19-21	间隔输出的常规模式	513
图 19-22	延迟模式有效间隔输出进入图例	514
图 19-23	延迟模式有效间隔输出期间状态切换图例	514
图 19-24	间隔输出的退出	515
图 19-25	通道 B 跟随功能退出和进入空闲图例	516
图 19-26	三角波模式多相 PWM 输出从单元启动、清零动作	517
图 19-27	锯齿波模式多相 PWM 输出从单元启动、清零动作	517
图 19-28	外部事件处理框图	518
图 19-29	事件消隐模式	521
图 19-30	消隐模式指定时间	521
图 19-31	消隐模式下触发计数器清零的外部事件延迟功能	522
图 19-32	事件窗口模式	523
图 19-33	窗口事件指定时间	523
图 19-34	软件同步动作	525
图 19-35	HRPWM_TRIGA 的滤波功能	526
图 19-36	三角波周期缓存传送点缓存周期值异常情况	531
图 19-37	锯齿波模式周期缓存传送点缓存动作	531
图 19-38	三角波模式时计数缓存动作 1	532
图 19-39	三角波模式时计数缓存动作 2	532
图 19-40	间隔输出缓存动作	533
图 19-41	单边对齐独立高分辨率 PWM	534
图 19-42	双边对称独立 PWM	535
图 19-43	软件实现互补 PWM 波输出	536
图 19-44	硬件实现互补 PWM 波输出	537
图 19-45	双边非对称 PWM 输出	538
图 19-46	6 相单边对齐独立 PWM	539
图 19-47	三相交错互补高分辨率 PWM	540
图 19-48	锯齿波模式多相交错 PWM 变频输出	541
图 19-49	三角波模式多相交错 PWM 变频输出	542
图 19-50	周期间隔有效请求信号动作	543
图 19-51	输出管理	545
图 19-52	立即交换输出波形	545
图 19-53	完整周期点交换输出波形	546
图 19-54	DAC 同步触发源	547
图 19-55	单元 1~3 专用计数匹配事件	551

图 19-56 锯齿波模式时中断&事件输出例.....	552
图 20-1 Timer4 基本框图.....	675
图 20-2 Timer4 锯齿波波形.....	676
图 20-3 Timer4 三角波波形.....	676
图 20-4 Timer4 锯齿波模式计数动作.....	677
图 20-5 Timer4 三角波模式计数动作.....	677
图 20-6 锯齿波模式波形输出例.....	678
图 20-7 三角波模式波形输出示例.....	679
图 20-8 缓存无效时修改锯齿波计数周期.....	680
图 20-9 缓存使能时修改锯齿波计数周期.....	681
图 20-10 缓存使能时修改三角波计数周期.....	681
图 20-11 OCCR 缓冲数据传输（周期间隔响应链接禁止时）.....	682
图 20-12 OCCR 缓冲数据传输（周期间隔响应链接使能）.....	683
图 20-13 输出比较缓冲数据传输（OCMR 缓冲使能）.....	684
图 20-14 SCCR 缓冲传输操作（周期间隔响应链接传输禁止时）.....	684
图 20-15 SCCR 缓冲传输操作（周期间隔响应链接传输使能时）.....	685
图 20-16 锯齿波独立 PWM 输出示例.....	686
图 20-17 三角波独立 PWM 输出示例.....	687
图 20-18 锯齿波扩展 PWM 输出.....	687
图 20-19 软件实现互补 PWM 输出.....	688
图 20-20 死区定时器模式下的互补 PWM 输出.....	689
图 20-21 脉宽异常时的死区定时器模式下的波形输出.....	689
图 20-22 死区定时器滤波模式下的互补 PWM 输出.....	690
图 20-23 周期间隔响应时序图.....	691
图 20-24 专用事件输出信号周期间隔响应输出.....	692
图 20-25 PWM 周期输出监测示例.....	694
图 20-26 比较启动模式下专用事件输出 TMR4_ADSTM 时序.....	694
图 20-27 延时启动模式下专用事件输出信号的输出时序.....	696
图 21-1 EMB 通道功能框图.....	726
图 21-2 EMB 结构框图.....	727
图 22-1 TimerA 基本框图.....	756
图 22-2 锯齿波波形（递加计数）.....	757
图 22-3 三角波波形.....	757
图 22-4 比较输出动作.....	758
图 22-5 捕获输入动作.....	760

图 22-6	软件同步动作.....	761
图 22-7	时钟输入端口的滤波功能	761
图 22-8	锯齿波模式时缓存动作.....	762
图 22-9	锯齿波模式时缓存与特殊缓存动作	763
图 22-10	32 位级联计数动作	763
图 22-11	单边对齐 PWM 输出例.....	764
图 22-12	双边对称 PWM 输出例.....	765
图 22-13	位置模式-基本计数.....	766
图 22-14	位置计数模式-相位差计数 (1 倍计数)	766
图 22-15	位置计数模式-相位差计数 (2 倍计数)	767
图 22-16	位置计数模式-相位差计数 (4 倍计数)	767
图 22-17	位置计数模式-方向计数.....	767
图 22-18	公转计数模式-Z 相计数	768
图 22-19	公转计数模式-位置溢出计数.....	768
图 22-20	公转计数模式-混合计数.....	769
图 23-1	Timer0 基本框图.....	791
图 23-2	Timer0 计数时序图	792
图 24-1	RTC 的基本框图	805
图 25-1	硬件启动例	825
图 25-2	软件启动例	826
图 25-3	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)	827
图 25-4	计数器下溢动作例.....	828
图 25-5	计数器刷新动作例.....	829
图 26-1	USART 系统框图	837
图 26-2	UART 数据格式	839
图 26-3	UART 发送数据图例 1.....	841
图 26-4	UART 发送数据图例 2.....	841
图 26-5	UART 内部同步和采样时序	842
图 26-6	UART 接收数据图例 1.....	844
图 26-7	UART 接收数据图例 2.....	844
图 26-8	半双工工作时序	847
图 26-9	over8=1 DE 断言	848
图 26-10	over8=0 DE 断言	848
图 26-11	over8=1 DE 去断言	848
图 26-12	over8=0 DE 去断言	849

图 26-13	多处理器通信图例.....	850
图 26-14	多处理器模式数据格式.....	850
图 26-15	多处理器模式发送数据图例.....	852
图 26-16	多处理器模式接收数据图例 1.....	853
图 26-17	多处理器模式接收数据图例 2.....	853
图 26-18	LIN 总线数据行为.....	854
图 26-19	智能卡连接示意图.....	857
图 26-20	智能卡模式同步时序和采样时序图.....	858
图 26-21	智能卡模式数据格式.....	858
图 26-22	智能卡模式发送数据图例.....	860
图 26-23	智能卡模式接收数据图例.....	860
图 26-24	时钟同步模式数据格式.....	862
图 26-25	时钟同步模式发送数据图例 1.....	863
图 26-26	时钟同步模式发送数据图例 2.....	864
图 26-27	时钟同步模式接收数据图例 1.....	865
图 26-28	时钟同步模式接收数据图例 2.....	865
图 27-1	I2C 系统框图.....	890
图 27-2	I2C 总线的结构例.....	891
图 27-3	I2C 总线的时序图.....	892
图 27-4	I2C 总线的数据格式.....	893
图 27-5	7 位地址格式的主机发送数据时序图 (例).....	894
图 27-6	7 位地址格式的主机接收数据的时序图 (例).....	895
图 27-7	7 位地址格式的从机发送模式时序图 (例).....	896
图 27-8	7 位地址格式从机接收模式时序图 (例).....	897
图 27-9	SCL 同步时序.....	898
图 27-10	从机发送时序图 (1).....	899
图 27-11	从机发送时序图 (2).....	899
图 27-12	从机接收时序图.....	900
图 27-13	快速 ACK/NACK 时序图.....	900
图 27-14	选择 7 位地址格式时的时序.....	901
图 27-15	选择 10 位地址格式时的时序.....	902
图 27-16	广播地址匹配时序图.....	903
图 27-17	SMBus 主机地址匹配时序图.....	904
图 27-18	SMBus 报警响应地址匹配时序图.....	905
图 27-19	SMBus 默认地址时序图.....	906

图 27-20 数字滤波电路框图.....	908
图 28-1 PLA 模块框图.....	934
图 28-2 PLA 单元框图.....	935
图 29-1 系统框图.....	947
图 29-2 主机模式结构.....	949
图 29-3 三线式时钟同步运行.....	950
图 29-4 数据格式.....	951
图 29-5 MSB 先传, 奇偶校验无效.....	951
图 29-6 LSB 先传, 奇偶校验无效.....	952
图 29-7 MSB 先传, 奇偶校验有效时.....	952
图 29-8 LSB 先传, 奇偶校验有效时.....	953
图 29-9 数据传送格式图 (CPHA=0).....	953
图 29-10 数据传送格式 (CPHA=1).....	954
图 29-11 全双工同步串行通信.....	955
图 29-12 只进行发送通信.....	956
图 29-13 常规通信.....	957
图 29-14 连续通信.....	957
图 29-15 奇偶校验流程.....	965
图 29-16 过载错误处理.....	967
图 29-17 启用时钟自动停止功能时的动作示意图 (CPHA=1).....	968
图 29-18 启用时钟自动停止功能时的动作示意图 (CPHA=0).....	968
图 29-19 奇偶校验错误.....	969
图 30-1 MCAN 系统框图.....	981
图 30-2 收发器延迟测量.....	987
图 30-3 总线监听模式下的引脚控制.....	988
图 30-4 回环模式的引脚控制.....	991
图 30-5 标准消息 ID 过滤器路径.....	994
图 30-6 扩展消息 ID 过滤器路径.....	995
图 30-7 接收 FIFO 状态.....	996
图 30-8 接收 FIFO 溢出处理.....	998
图 30-9 混合配置专用发送缓冲区和发送 FIFO 示例.....	1002
图 30-10 混合配置专用发送缓冲区和发送队列示例.....	1003
图 30-11 消息 RAM 配置.....	1005
图 31-1 CRC 应用示意图.....	1068
图 32-1 调试控制系统.....	1074

图 32-2 调试控制系统.....	1075
图 32-3 JTAG-DP 到 SW-DP 切换时序.....	1076
图 32-4 TPIU 框图.....	1084

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)

HC32F334 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。

HC32F334 系列集成了高速片上存储器，包括最大 128KB 的 Flash，最大 36KB 的 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 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。

HC32F334 系列集成了丰富的外设功能。包括 3 个独立的 12bit 2.5MSPS ADC，3 个 12 位 15MSPS 的 DAC。3 个高速电压比较器 (CMP)。4 个多功能 PWM Timer (Timer6)，支持正交编码输入及 8 路互补 PWM 输出；6 个 22bit 高精度 PWM (HRPWM)，支持 12 路 130ps 高分辨率 PWM 波形输出；1 个电机 PWM Timer (Timer4) 支持 8 路互补 PWM 输出；4 个 16bit 通用 Timer (TimerA) 及 1 个 32bit 通用 Timer (TimerA)，支持正交编码输入及 20 路占空比独立可设 PWM 输出，6 个串行通信接口 (I2C/UART/ SPI)，2 路 MCAN 控制器。

HC32F334 系列支持宽电压范围 (1.8~3.63V)，宽运行温度范围 (-40~105°C) 和各种低功耗模式。

典型应用

HC32F334 系列提供 64pin、48pin 的 LQFP 封装，48pin、32pin 的 QFN 封装，可用于 AC/DC、DC/DC 数字电源应用，如通信与服务器电源、砖块电源、微逆、充电桩 DC/DC、高性能变频控制等领域。

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*2		开始地址*1	结束地址*1	空间大小	模块*3	保护*4	说明
系统	-	0xE010 0000	0xFFFF FFFF	511MB	Reserved	-	自定义空间
	私有外设 外部总线	0xE00F F000	0xE00F FFFF	4KB	ROMTABLE	-	-
		0xE004 2400	0xE00F EFFF	755KB	BLANK	-	-
		0xE004 2000	0xE004 23FF	1KB	DBGC	-	调试控制寄存器区域
		0xE004 1000	0xE004 1FFF	4KB	BLANK	-	-
		0xE004 0000	0xE004 0FFF	4KB	TPIU	-	-
	私有外设 内部总线	0xE000 F000	0xE003 FFFF	196KB	BLANK	-	-
		0xE000 E000	0xE000 EFFF	4KB	SCS	-	系统控制空间 NVIC/MPU 等
		0xE000 3000	0xE000 DFFF	44KB	BLANK	-	-
		0xE000 2000	0xE000 2FFF	4KB	FPB	-	-
0xE000 1000		0xE000 1FFF	4KB	DWT	-	-	
	0xE000 0000	0xE000 0FFF	4KB	ITM	-	-	
外部设备	-	0xA000 0000	0xDFFF FFFF	1024MB	Reserved	-	-
外部存储	AHB5 时钟: HCLK	0x9800 0000	0x9FFF FFFF	128MB	Reserved	-	-
		0x8820 0000	0x97FF FFFF	254MB	Reserved	-	-
	AHB5 时钟: EXCLK	0x8810 0000	0x881F FFFF	1MB	Reserved	-	-
		0x8800 0800	0x880F FFFF	1022KB	Reserved	-	-
		0x8800 0400	0x8800 07FF	1KB	Reserved	-	-
		0x8800 0000	0x8800 03FF	1KB	Reserved	-	-
		0x8000 0000	0x87FF FFFF	128MB	Reserved	-	-
	0x6000 0000	0x7FFF FFFF	512MB	Reserved	-	-	
外设		0x4400 0000	0x5FFF FFFF	448MB	Reserved	-	-
		0x4200 0000	0x43FF FFFF	32MB	PeriBitBand	-	-
		0x4010 0000	0x41FF FFFF	31MB	Reserved	-	-
		0x400C 0000	0x400F FFFF	256KB	Reserved	-	-
		0x4008 0000	0x400B FFFF	256KB	Reserved	-	-
		0x4007 8800	0x4007 FFFF	30KB	Reserved	-	-

存储器分类*2	开始地址*1	结束地址*1	空间大小	模块*3	保护*4	说明	
外设	AHB2 时钟: PCLK0	0x4007 8400	0x4007 87FF	1KB	Reserved	-	-
		0x4007 8000	0x4007 83FF	1KB	Reserved	-	-
	AHB3 时钟: PCLK0	0x4007 3840	0x4007 7FFF	18KB	BLANK	-	-
		0x4007 3800	0x4007 383F	64B	CANRAM_ECC	-	-
		0x4007 3000	0x4007 37FF	2KB	BLANK	-	-
		0x4007 2800	0x4007 2FFF	2KB	CAN_RAM	-	-
		0x4007 1800	0x4007 27FF	4KB	BLANK	-	-
		0x4007 1600	0x4007 17FF	512B	BLANK	-	-
		0x4007 1400	0x4007 15FF	512B	MCAN_2	-	-
		0x4007 1200	0x4007 13FF	512B	BLANK	-	-
		0x4007 1000	0x4007 11FF	512B	MCAN_1	-	-
	0x4006 0000	0x4007 0FFF	68KB	BLANK	-	-	
	AHB1 时钟: HCLK	0x4005 9000	0x4005 FFFF	28KB	BLANK	-	-
		0x4005 8C00	0x4005 8FFF	1KB	BLANK	-	-
		0x4005 8800	0x4005 8BFF	1KB	BLANK	-	-
		0x4005 8400	0x4005 87FF	1KB	BLANK	-	-
		0x4005 8000	0x4005 83FF	1KB	BLANK	-	-
		0x4005 7C00	0x4005 7FFF	1KB	BLANK	-	-
		0x4005 7800	0x4005 7BFF	1KB	BLANK	-	-
		0x4005 7400	0x4005 77FF	1KB	BLANK	-	-
		0x4005 7000	0x4005 73FF	1KB	BLANK	-	-
		0x4005 6C00	0x4005 6FFF	1KB	BLANK	-	-
		0x4005 6800	0x4005 6BFF	1KB	BLANK	-	-
		0x4005 6400	0x4005 67FF	1KB	BLANK	-	-
		0x4005 6000	0x4005 63FF	1KB	BLANK	-	-
		0x4005 5C00	0x4005 5FFF	1KB	BLANK	-	-
		0x4005 5800	0x4005 5BFF	1KB	BLANK	-	-
		0x4005 5400	0x4005 57FF	1KB	PERIC	-	周边模块控制寄存器
		0x4005 5000	0x4005 53FF	1KB	BLANK	-	-
		0x4005 4400	0x4005 4FFF	3KB	BLANK	-	禁止访问
0x4005 4000		0x4005 43FF	1KB	CMU	带保护	-	
0x4005 3800		0x4005 3FFF	2KB	GPIO	-	-	
0x4005 3400		0x4005 37FF	1KB	BLANK	-	-	
0x4005 3000	0x4005 33FF	1KB	DMA	-	-		
0x4005 2000	0x4005 2FFF	4KB	BLANK	-	-		
0x4005 1000	0x4005 1FFF	4KB	INTC	带保护	-		
0x4005 0C00	0x4005 0FFF	1KB	BLANK	-	-		

存储器分类*2	开始地址*1	结束地址*1	空间大小	模块*3	保护*4	说明	
外设		0x4005 0800	0x4005 0BFF	1KB	RAMIF	带保护	-
		0x4005 0400	0x4005 07FF	1KB	BLANK	-	-
		0x4005 0000	0x4005 03FF	1KB	DMPU	带保护	-
	APB4 时钟: PCLK3	0x4004 FC00	0x4004 FFFF	1KB	PLA	-	-
		0x4004 F800	0x4004 FBFF	1KB	BLANK	-	-
		0x4004 F400	0x4004 F7FF	1KB	BLANK	-	-
		0x4004 F000	0x4004 F3FF	1KB	BLANK	-	-
		0x4004 EC00	0x4004 EFFF	1KB	BLANK	-	-
		0x4004 E800	0x4004 EBFF	1KB	BLANK	-	-
		0x4004 E400	0x4004 E7FF	1KB	BLANK	-	-
		0x4004 E000	0x4004 E3FF	1KB	BLANK	-	-
		0x4004 D000	0x4004 DFFF	1KB	BLANK	-	-
		0x4004 CC00	0x4004 CFFF	1KB	PWC, CMU	带保护	-
		0x4004 C800	0x4004 CBFF	1KB	BLANK	-	-
		0x4004 C400	0x4004 C7FF	1KB	WKTMR	带保护	-
		0x4004 C000	0x4004 C3FF	1KB	RTC	带保护	-
		0x4004 AC00	0x4004 BFFF	5KB	BLANK	-	-
		0x4004 A800	0x4004 ABFF	1KB	BLANK	-	-
		0x4004 A400	0x4004 A7FF	1KB	BLANK	-	-
		0x4004 A000	0x4004 A3FF	1KB	BLANK	-	-
		0x4004 9C00	0x4004 9FFF	1KB	CTC	-	-
		0x4004 9800	0x4004 9BFF	1KB	BLANK	-	-
		0x4004 9400	0x4004 97FF	1KB	SWDT	带保护	-
		0x4004 9000	0x4004 93FF	1KB	WDT	带保护	-
		0x4004 8800	0x4004 8FFF	2KB	BLANK	-	-
		0x4004 8400	0x4004 87FF	1KB	FCM	-	-
		0x4004 8000	0x4004 83FF	1KB	FCG, CMU	带保护	-
		外设	APB3 时钟: PCLK4	0x4004 2400	0x4004 7FFF	23KB	BLANK
0x4004 2000	0x4004 23FF			1KB	BLANK	-	-
0x4004 1C00	0x4004 1FFF			1KB	BLANK	-	-
0x4004 1800	0x4004 1BFF			1KB	BLANK	-	-
0x4004 1400	0x4004 17FF			1KB	DAC_2	-	-
0x4004 1000	0x4004 13FF			1KB	DAC_1	-	-
0x4004 0C00	0x4004 0FFF			1KB	BLANK	-	-
0x4004 0800	0x4004 0BFF			1KB	ADC_3	-	-
0x4004 0400	0x4004 07FF			1KB	ADC_2	-	-
0x4004 0000	0x4004 03FF			1KB	ADC_1	-	-

存储器分类*2	开始地址*1	结束地址*1	空间大小	模块*3	保护*4	说明	
外设	APB5 时钟: PCLK0	0x4003 C000	0x4003 FFFF	16KB	HRPWM	-	-
		0x4003 BC00	0x4003 BFFF	1KB	BLANK	-	-
		0x4003 B800	0x4003 BBFF	1KB	BLANK	-	-
		0x4003 B400	0x4003 B7FF	1KB	I2C	-	-
		0x4003 B000	0x4003 B3FF	1KB	CMP_3	-	-
		0x4003 AC00	0x4003 AFFF	1KB	TimerA_4	-	-
		0x4003 A800	0x4003 ABFF	1KB	TimerA_3	-	-
		0x4003 A400	0x4003 A7FF	1KB	TimerA_2	-	-
		0x4003 A000	0x4003 A3FF	1KB	TimerA_1	-	-
		0x4003 9000	0x4003 9FFF	4KB	Timer6	-	-
		0x4003 8800	0x4003 8EFF	2KB	BLANK	-	-
		0x4003 8600	0x4003 87FF	512B	EMB	-	-
		0x4003 8400	0x4003 85FF	1KB	CMP_12	-	-
		0x4003 8000	0x4003 83FF	1KB	Timer4	-	-
	APB2 时钟: PCLK1	0x4002 B800	0x4002 7FFF	50KB	BLANK	-	-
		0x4002 B000	0x4002 B7FF	2KB	BLANK	-	-
		0x4002 9800	0x4002 AFFF	6KB	BLANK	-	-
		0x4002 9400	0x4002 97FF	1KB	BLANK	-	-
		0x4002 9000	0x4002 93FF	1KB	BLANK	-	-
		0x4002 8800	0x4002 8FFF	2KB	BLANK	-	-
		0x4002 8400	0x4002 87FF	1KB	BLANK	-	-
		0x4002 8000	0x4002 83FF	1KB	BLANK	-	-
		0x4002 7C00	0x4002 7FFF	1KB	BLANK	-	-
		0x4002 7800	0x4002 7BFF	1KB	BLANK	-	-
		0x4002 7400	0x4002 77FF	1KB	BLANK	-	-
		0x4002 7000	0x4002 73FF	1KB	BLANK	-	-
		0x4002 6C00	0x4002 6FFF	1KB	BLANK	-	-
		0x4002 6800	0x4002 6BFF	1KB	BLANK	-	-
0x4002 6400	0x4002 67FF	1KB	BLANK	-	-		
0x4002 6000	0x4002 63FF	1KB	TimerA_5	-	-		
0x4002 5800	0x4002 5FFF	2KB	BLANK	-	-		
0x4002 5400	0x4002 57FF	1KB	BLANK	-	-		
0x4002 5000	0x4002 53FF	1KB	BLANK	-	-		
0x4002 4C00	0x4002 4FFF	1KB	BLANK	-	-		
0x4002 4800	0x4002 4BFF	1KB	BLANK	-	-		
0x4002 4400	0x4002 47FF	1KB	Timer0_2	-	-		
0x4002 4000	0x4002 43FF	1KB	Timer0_1	-	-		
0x4002 2800	0x4002 3FFF	6KB	BLANK	-	-		
0x4002 2400	0x4002 27FF	1KB	BLANK	-	-		

存储器分类*2	开始地址*1	结束地址*1	空间大小	模块*3	保护*4	说明	
外设		0x4002 2000	0x4002 23FF	1KB	BLANK	-	-
		0x4002 1C00	0x4002 1FFF	1KB	BLANK	-	-
		0x4002 1800	0x4002 1BFF	1KB	BLANK	-	-
		0x4002 1400	0x4002 17FF	1KB	BLANK	-	-
		0x4002 1000	0x4002 13FF	1KB	BLANK	-	-
		0x4002 0C00	0x4002 0FFF	1KB	BLANK	-	-
		0x4002 0800	0x4002 0BFF	1KB	BLANK	-	-
		0x4002 0400	0x4002 07FF	1KB	BLANK	-	-
		0x4002 0000	0x4002 03FF	1KB	BLANK	-	-
	APB1 时钟: PCLK1	0x4001 E800	0x4001 FFFF	6KB	BLANK	-	-
		0x4001 E400	0x4001 E7FF	1KB	BLANK	-	-
		0x4001 E000	0x4001 E3FF	1KB	BLANK	-	-
		0x4001 DC00	0x4001 DFFF	1KB	BLANK	-	-
		0x4001 D800	0x4001 DBFF	1KB	USART_4	-	-
		0x4001 D400	0x4001 D7FF	1KB	USART_3	-	-
		0x4001 D000	0x4001 D3FF	1KB	USART_2	-	-
		0x4001 CC00	0x4001 CFFF	1KB	USART_1	-	-
		0x4001 C800	0x4001 CBFF	1KB	BLANK	-	-
		0x4001 C400	0x4001 C7FF	1KB	BLANK	-	-
		0x4001 C000	0x4001 C3FF	1KB	SPI	-	-
		0x4001 8000	0x4001 BFFF	16KB	BLANK	-	-
		0x4001 7C00	0x4001 7FFF	1KB	EMB	-	-
		0x4001 0C00	0x4001 7BFF	28KB	BLANK	-	-
		0x4001 0800	0x4001 0BFF	1KB	AOS	-	内部触发事件寄存器区域
	0x4001 0400	0x4001 07FF	1KB	EFM	带保护	-	
	0x4001 0000	0x4001 03FF	1KB	BLANK	-	-	
	AHB4 时钟: PCLK1	0x4000 9400	0x4000 FFFF	27KB	BLANK	-	-
		0x4000 9000	0x4000 93FF	1KB	BLANK	-	-
		0x4000 8C00	0x4000 8FFF	1KB	CRC	-	-
		0x4000 8800	0x4000 8BFF	1KB	BLANK	-	-
		0x4000 8400	0x4000 87FF	1KB	BLANK	-	-
		0x4000 8000	0x4000 83FF	1KB	BLANK	-	-
		0x4000 0000	0x4000 7FFF	32KB	Reserved	-	-
SRAM 时钟: HCLK		0x2400 0000	0x3FFF FFFF	448MB	Reserved	-	-
	0x2200 0000	0x23FF FFFF	32MB	SRAMBitBand	-	-	
	0x2010 0000	0x21FF FFFF	31MB	Reserved	-	-	
	0x200F 1000	0x200F FFFF	60KB	Reserved	-	-	
	0x200F 0000	0x200F 0FFF	4KB	Ret SRAM	带保护	-	

存储器分类*2	开始地址*1	结束地址*1	空间大小	模块*3	保护*4	说明	
	0x2006 0000	0x200E FFFF	576KB	Reserved	-	-	
	0x2004 0000	0x2005 FFFF	128KB	Reserved	-	-	
	0x2000 3000	0x2003 FFFF	240KB	Reserved	-	-	
	0x2000 0000	0x2000 3FFF	16KB	SRAM0	-	ECC RAM	
CODE	SRAM 时钟: HCLK	0x1FFF C000	0x1FFF FFFF	16KB	SRAMH	-	ECC RAM
		0x0300 4000	0x1FFF BFFF	471.94MB	BLANK	-	-
	OTP, Flash 时钟: HCLK	0x0300 204C	0x0300 3FFF	8116B	BLANK	-	-
		0x0300 2040	0x0300 204B	12B	数据安全保护	-	用于配置数据安全保护
		0x0300 2004	0x0300 203F	60B	BLANK	-	-
		0x0300 2000	0x0300 2003	4B	引导交换	-	-
		0x0300 1000	0x0300 1FFF	4KB	BLANK	-	-
		0x0300 0A00	0x0300 0FFF	1.5KB	OTP	-	-
		0x0300 0000	0x0300 09FF	2.5KB	BLANK	-	-
	-	0x0210 0000	0x02FF FFFF	15M	BLANK	-	-
	REMAP 时钟: HCLK	0x020A 0000	0x020F FFFF	384KB	BLANK	-	-
		0x0208 0000	0x0209 FFFF	128KB	REMAP1	-	地址重映射区域 1
		0x0202 0000	0x0207 FFFF	384KB	BLANK	-	-
		0x0200 0000	0x0201 FFFF	512KB	REMAP0	-	地址重映射区域 0
	-	0x0008 0000	0x01FF FFFF	31.5M	BLANK	-	-
	Flash 时钟: HCLK	0x0002 0000	0x0007 FFFF	384KB	BLANK	-	-
0x0000 0000		0x0001 FFFF	128KB	Embedded Flash 0	-	-	

注:

1. 请参考 *ARM Cortex-M4 Processor Technical Reference Manual Revision:r0p1*。
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 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。

1.3 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_REMCRO/1.RMSIZE[4:0])

重映射地址 1:

0x0208 0000~0x0210 0000 (视重映射空间而定 MMF_REMCRO/1.RMSIZE[4:0])

重映射功能有效时，地址对应表如下:

表 1-2 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]=0b011110情形 (重映射空间: 16KB)	0x0200 0000~0x0200 3FFF	全0	RMTADDR[16:2]	CPUADDR[13:0]
RMSIZE[4:0]=0b011111情形 (重映射空间: 32KB)	0x0200 0000~0x0200 7FFF	全0	RMTADDR[16:3]	CPUADDR[14:0]
RMSIZE[4:0]=0b10000情形 (重映射空间: 64KB)	0x0208 0000~0x0208 FFFF	全0	RMTADDR[16:4]	CPUADDR[15:0]
RMSIZE[4:0]=0b10001情形 (重映射空间: 128KB)	0x0208 0000~0x0209 FFFF	全0	RMTADDR[16:5]	CPUADDR[16:0]

举例:

1. 使用重映射地址 0 功能，设定源地址为主闪存 FLASH 地址 0x0000 8000，重映射空间 32K，寄存器 MMF_REMCRO 需设定为 0x8000 800F。
2. 使用重映射地址 1 功能，设定源地址为高速 SRAM 地址 0x1FFF 8000，重映射空间为 16K，寄存器 MMF_REMCR1 需设定为 0x9FFF 800E。

注意:

源地址的起始地址应设为重映射空间的整数倍。

1.4 寄存器描述

表 1-3 MMF 基地址

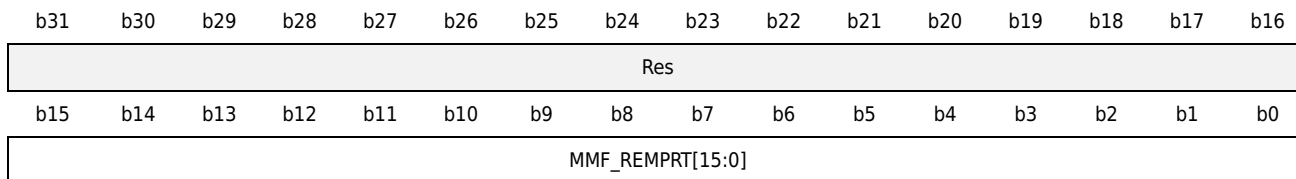
名称	基地址	描述
MMF	0x4001 0500	MMF 基地址

表 1-4 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
MMF_REMPRT	访问保护寄存器	0x0000	32	0x0000 0000
MMF_REMCRO	重映射控制寄存器0	0x0004	32	0x0000 0000
MMF_REMCR1	重映射控制寄存器1	0x0008	32	0x0000 0000

1.4.1 访问保护寄存器 (MMF_REMPRT)

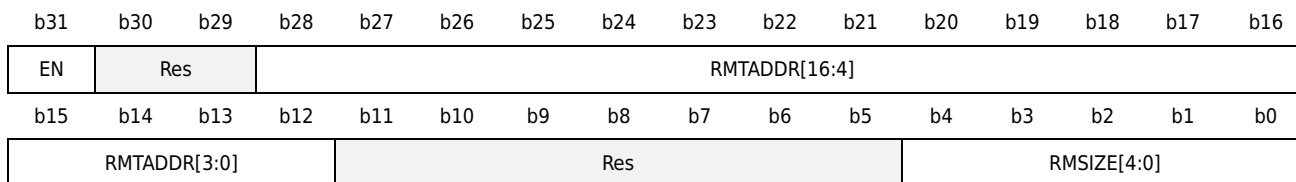
复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”，写入时写“0”	R
b15~b0	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	RW

1.4.2 重映射控制寄存器 (MMF_REMCRx, x=0, 1)

复位值: 0x0000 0000



位/位域	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	RW
b30~b29	Res	保留位	读出时为“0”，写入时写“0”	R
b28~b12	RMTADDR[16:0]	源地址	有效位数和RMSIZE[4:0]设定有关。 设定可参考表 1-2。	RW
b11~b5	Res	保留位	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，禁止设定 01100: 4KB 01101: 8KB 01110: 16KB 01111: 32KB 10000: 64KB 10001: 128KB (64KB 产品禁止设定) 10010~11111: 保留，禁止设定	RW

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 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 SRAMH (SRAMH 16KB) 总线
 - 系统 SRAM (SRAM0 16KB) 总线
 - 系统 SRAM (Ret SRAM 4KB) 总线
 - APB1 外设总线 (SPI/ USART/ EFM/ AOS)
 - APB2 外设总线 (TimerA/ Timer0/)
 - APB3 外设总线 (ADC/ DAC)
 - APB4 外设总线 (PLA/FCM/ WDT/ SWDT/ PWC/ CTC/ RTC/ WKTM)
 - APB5 外设总线 (TimerA/ Timer4/ Timer6/ CMP/ I2C/EMB/HRPWM)
 - AHB1 外设总线 (CMU/ GPIO/ DMA/ INTC / DMPU)
 - AHB3 外设总线(MCAN)
 - AHB4 外设总线 (CRC)

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

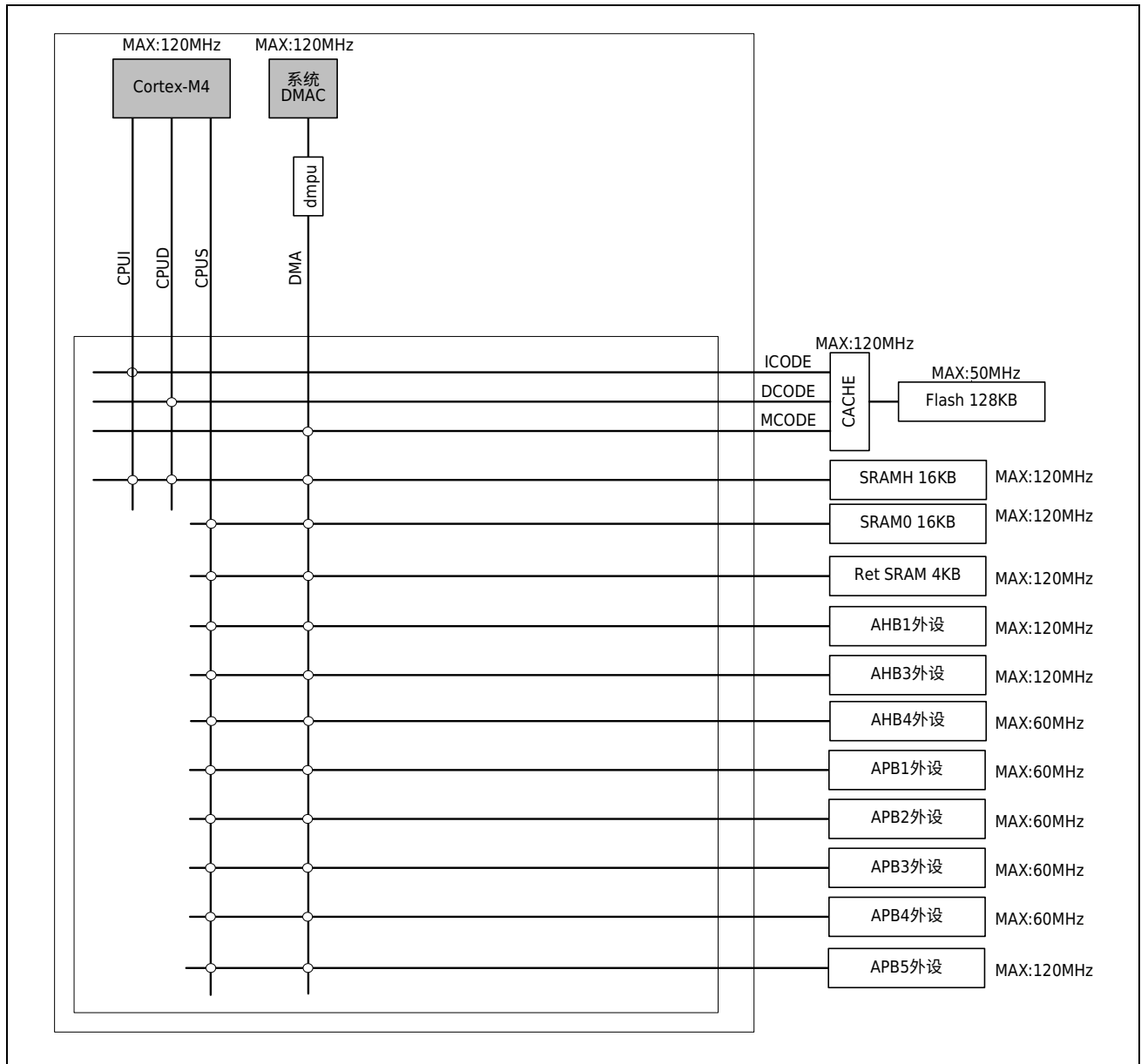


图 2-1 总线架构图

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 SRAM0/ Ret SRAM 所有外设以及外部扩展空间。

- DMA 总线

系统 DMA 专用总线, DMA 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。

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。

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 访问 SRAM0, 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														
	上电复位	引脚复位	欠压复位	电压检测 ↑ 复位	电压检测 ~ 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	Cortex-M4 Lockup 复位
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 ×: 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】。

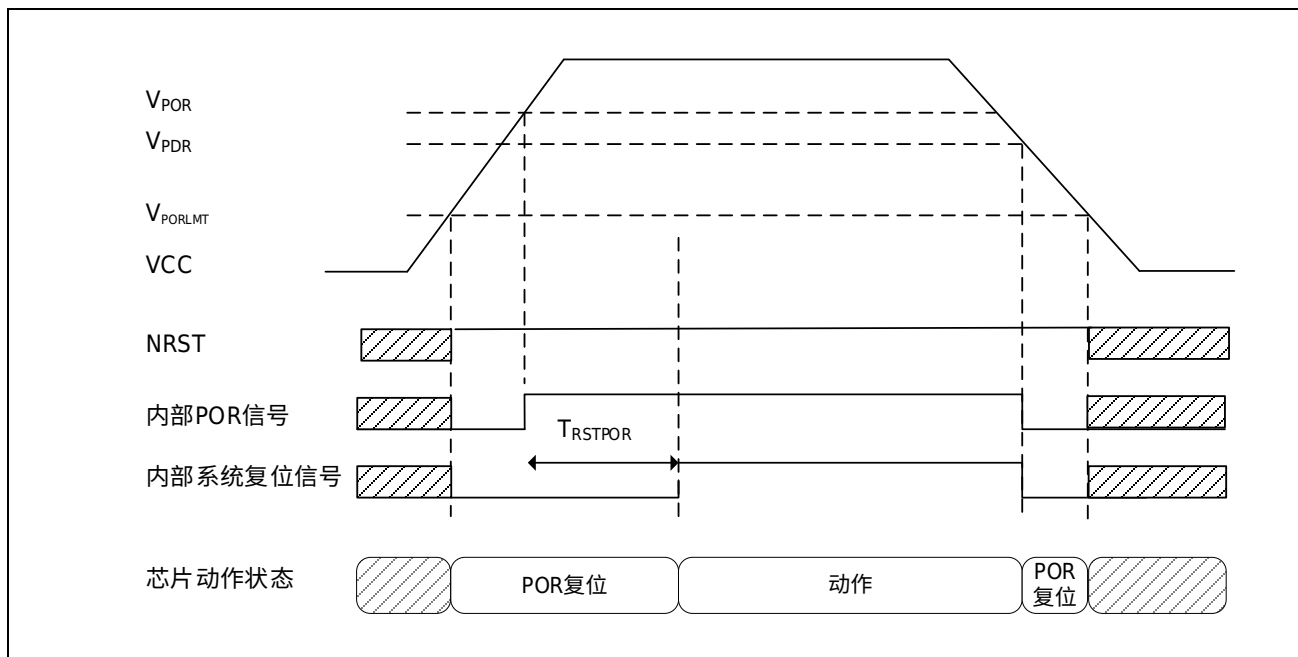


图 3-1 上电复位

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。

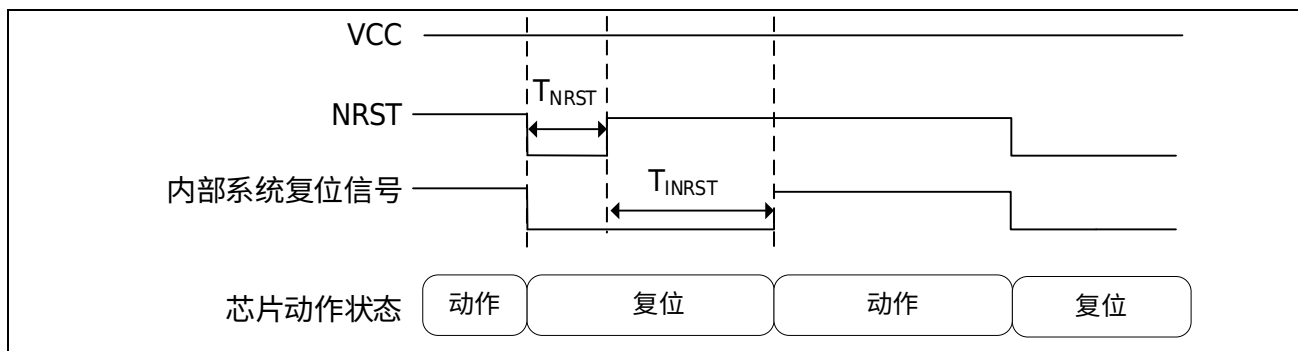


图 3-2 NRST 复位时序

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】。

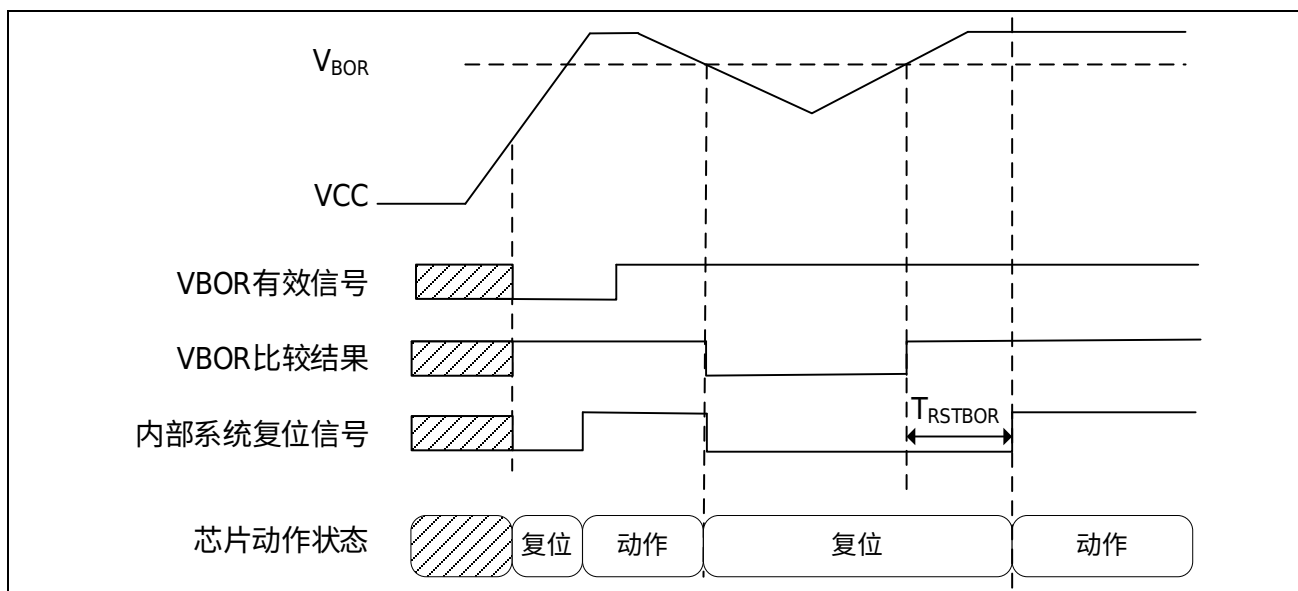


图 3-3 欠压复位

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 复位是电压监测电路引起的复位。

当 VCC 电压低于可编程电压检测 1 的监测电压时，根据设定，可产生可编程电压检测 1 复位，RMU_RSTF0.PVD1F 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。

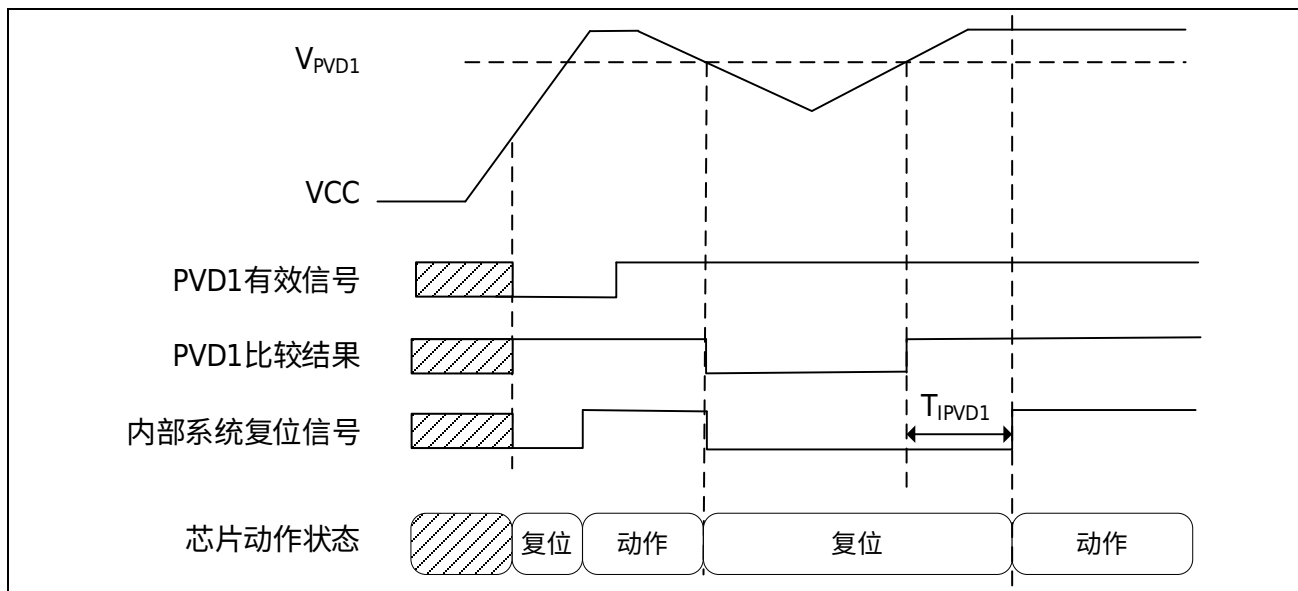


图 3-4 可编程电压检测 1 复位

当 VCC 电压低于可编程电压检测 2 的监测电压时，根据设定，可产生可编程电压检测 2 复位，RMU_RSTF0.PVD2F 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。

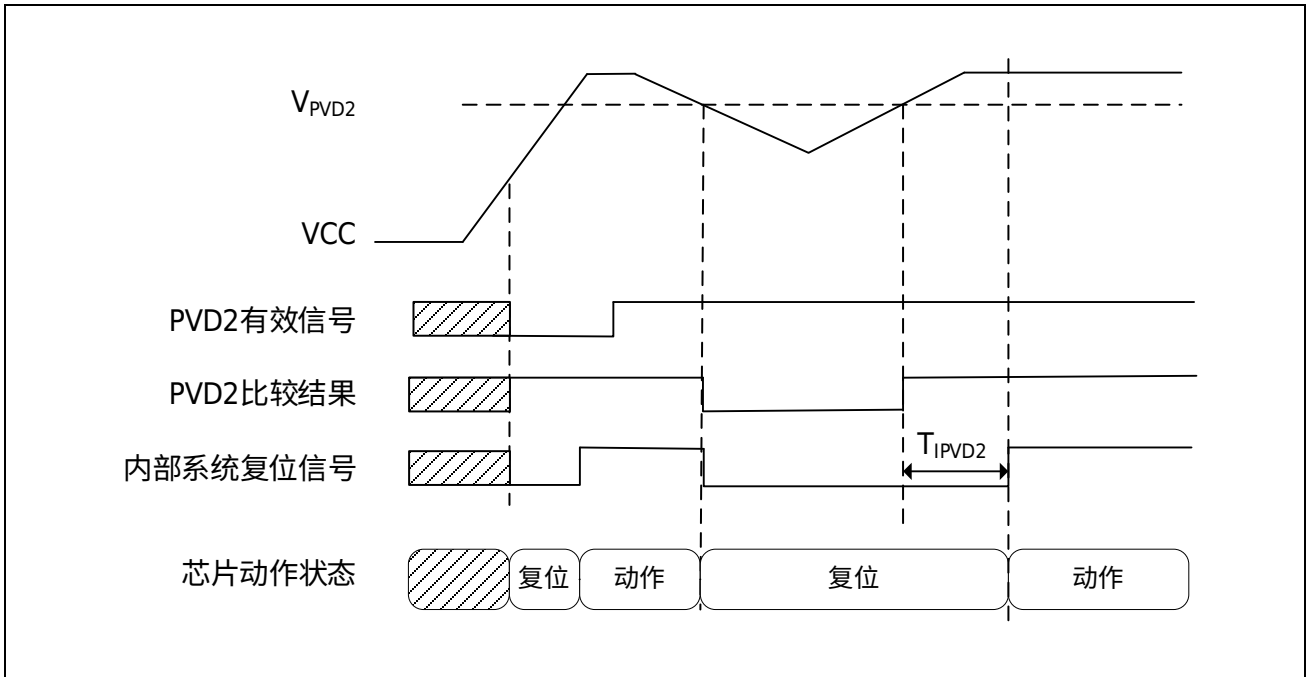


图 3-5 可编程电压检测 2 复位

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因为的内部复位，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。

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）】。

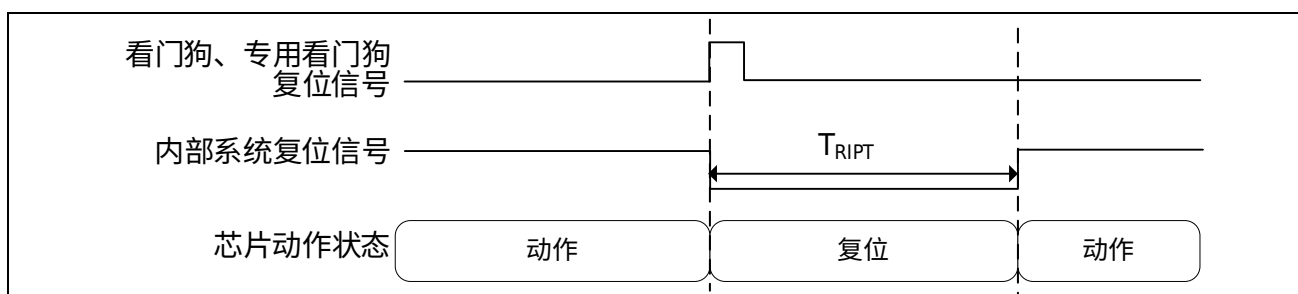


图 3-6 看门狗和专用看门狗复位

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】。

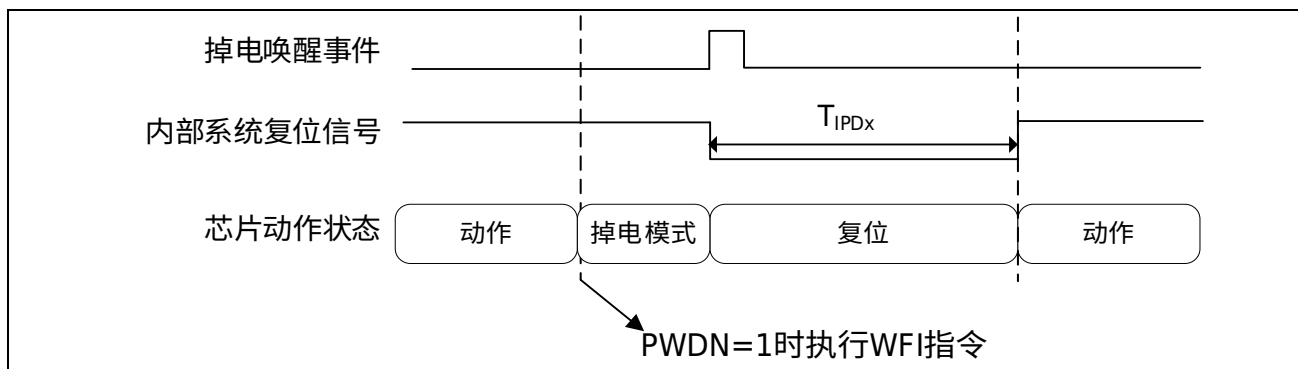


图 3-7 掉电唤醒复位

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。

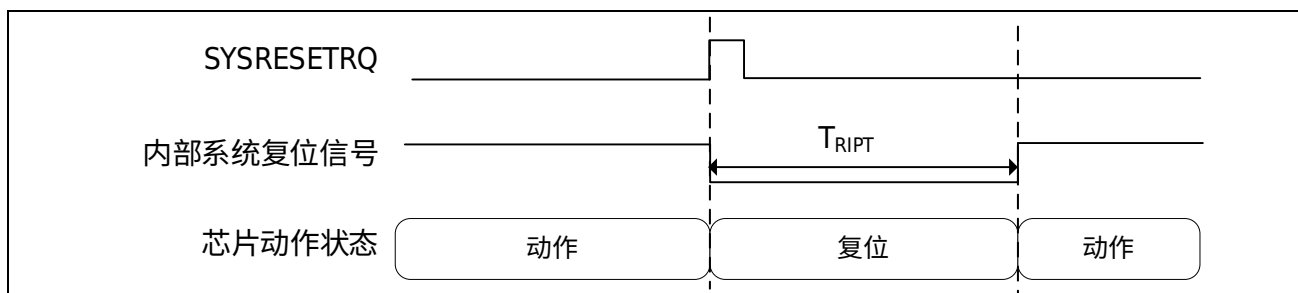


图 3-8 软件复位

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)】。

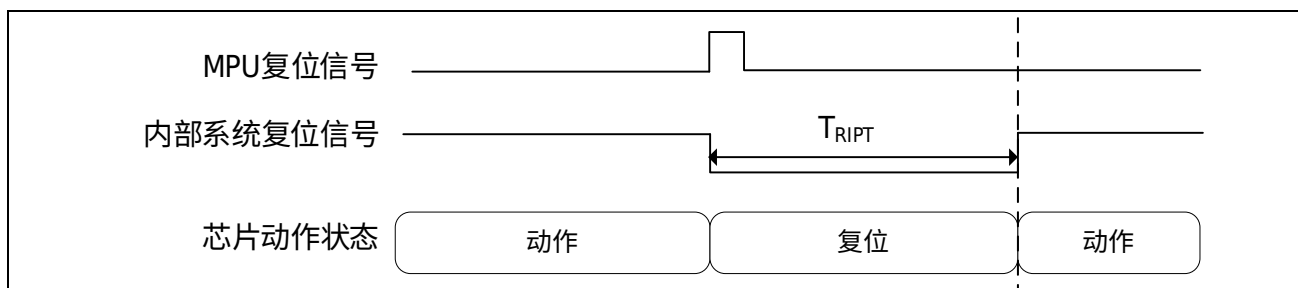


图 3-9 MPU 错误复位

3.3.9 RAM 奇偶校验复位

RAM 奇偶校验发生错误时，根据设定，可产生 RAM 奇偶校验复位，其复位时序如图 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)】。

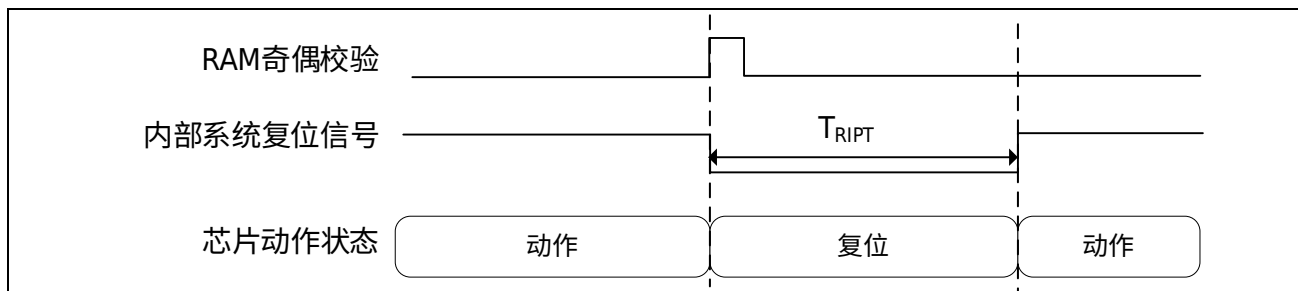


图 3-10 RAM 奇偶校验复位

3.3.10 RAMECC 复位

RAMECC 校验发生错误时，根据设定，可产生 RAMECC 复位，其复位时序如图 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)】。

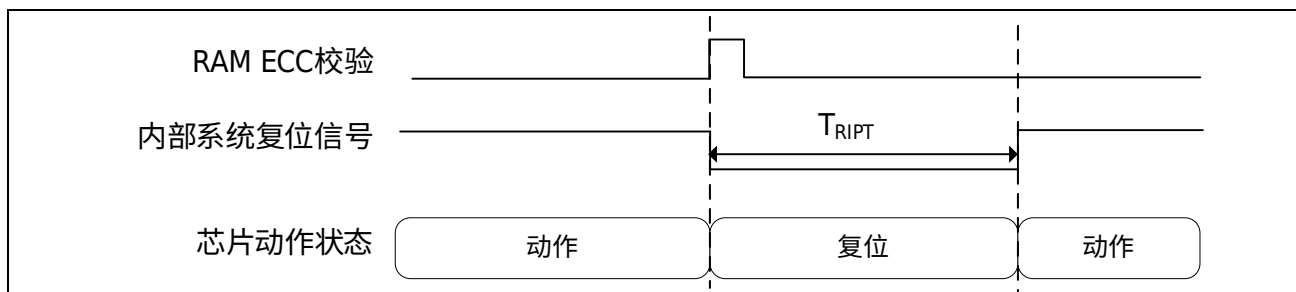


图 3-11 RAMECC 复位

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，根据设定，可产生时钟频率异常复位，其复位时序如图 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】。

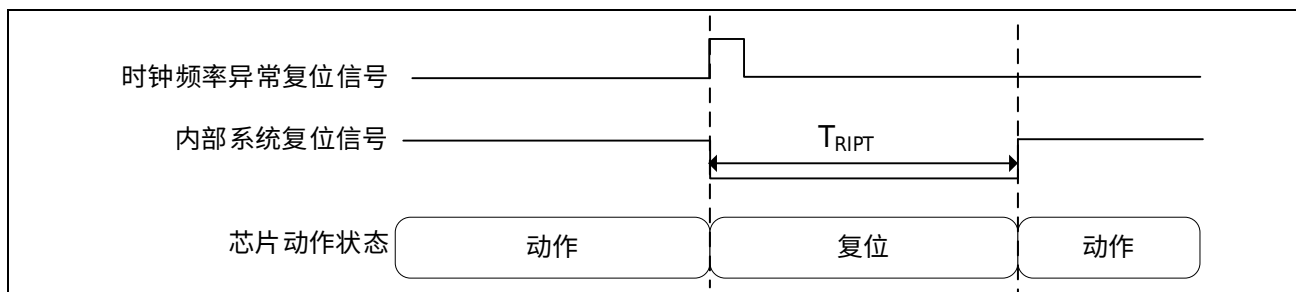


图 3-12 时钟频率异常复位

3.3.12 外部高速振荡器异常停振复位

当芯片的振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】。

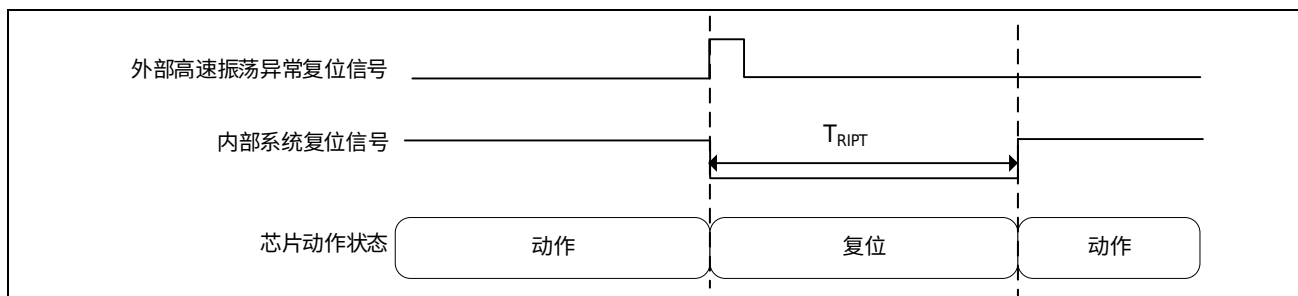


图 3-13 外部高速振荡异常复位

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。

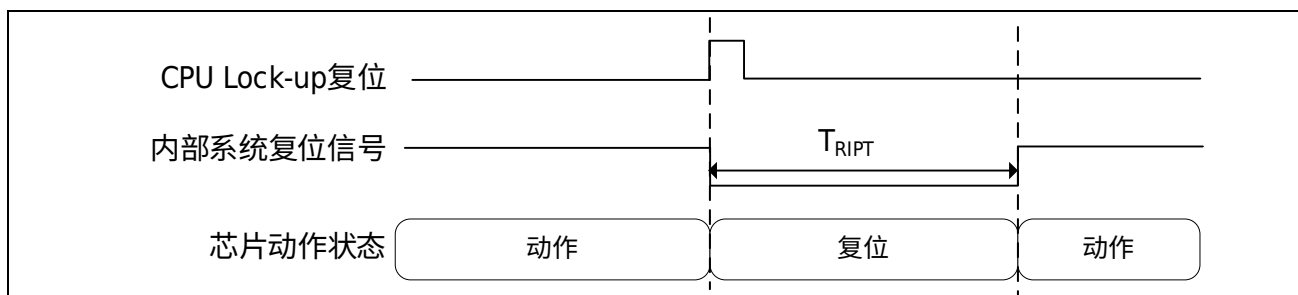


图 3-14 Cortex-M4 Lockup 复位

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。

3.3.15 模块功能复位

除上述 15 种芯片复位以外,产品还为各外设模块配置了功能复位寄存器 RMU_FRST0~3。对 RMU_FRST 的对应位写 0,可以让对应模块的功能复位。RMU_FRST 写 0 后经过 2 个外设模块的时钟上升沿后会自动置 1。

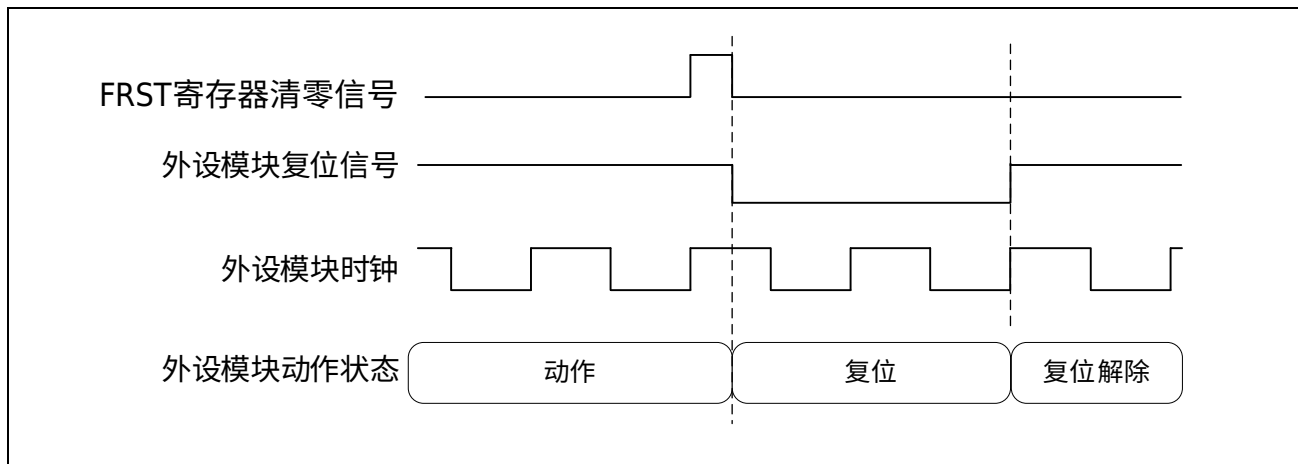


图 3-15 模块功能复位

3.3.16 各个模块的复位条件

表中所有复位源是指本章简介中描述的 15 种复位源。

表 3-3 各模块复位条件

模块	寄存器	复位源
调试控制器 (DBGC)	MCUSTPCTL、MCUSTPCTL2、 MCUTRACECTL、MCUDBGSTAT	1. 上电复位 2. 掉电唤醒复位
实时时钟 (RTC)	RTC 内部寄存器	模块软件复位控制位: RTC_CR0.RESET
内置 SRAM (SRAM)	SRAMC_CKSR	1. 上电复位 2. 掉电模式 3 唤醒复位
	SRAMC_CKSR 以外的其他寄存器	所有复位源
电源控制 (PWC) 时钟控制 (CMU)	PWC_PWRC0、PWC_PWRC1、 PWC_PDWKE0、PWC_PDWKE1、 PWC_PDWKE2、PWC_PDWKES、 CMU_XTALCFGR	下记复位以外的所有复位源: 1. 掉电模式 1 唤醒复位 2. 掉电模式 2 唤醒复位 3. 掉电模式 4 唤醒复位
	PWC_PDWKF0、PWC_PDWKF1	掉电唤醒复位以外的所有复位源
	PWC_PVDLCR、PWC_PVDCR1、 PWC_PVDFCR、PWC_PVDCCR0、 PWR_PWRC2、PWR_PWRC4	1. 上电复位 2. 引脚复位 3. 欠压复位 4. 看门狗复位 5. 专用看门狗复位 6. 掉电模式 3 唤醒复位

模块	寄存器	复位源
	PWC_PVDICR.PVD1EDGS[1:0]、 PWC_PVDDSR.PVD1DETFLG、 PWC_PVDICR.PVD2EDGS[1:0]、 PWC_PVDDSR.PVD2DETFLG	1. 上电复位 2. 引脚复位 3. 欠压复位 4. 看门狗复位 5. 专用看门狗复位 6. 掉电唤醒复位
	PWC_WKTC	上电复位
	上記以外	所有复位源
以上模块以外的寄存器		所有复位源

3.4 寄存器描述

表 3-4 RMU 基地址

名称	基地址	描述
RMU	0x4004 CCE0	RMU基地址

表 3-5 RMU 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
RMU_FRST0	功能复位控制寄存器0	0x00	32	0xFFFF FFFF
RMU_FRST1	功能复位控制寄存器1	0x04	32	0xFFFF FFFX
RMU_FRST2	功能复位控制寄存器2	0x08	32	0xFFFF FFFF
RMU_FRST3	功能复位控制寄存器3	0x0C	32	0xFFFF FFFF
RMU_PRSTCR0	复位控制寄存器	0x10	8	0x40
RMU_RSTF0	复位标志寄存器	0x14	32	根据不同的复位方式复位值不同

3.4.1 功能复位控制 0 (RMU_FRST0)

复位值: 0xFFFF FFFF

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

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时所有位均为“1”，写入时所有位请写“1”	RW
b23	CRC	CRC复位控制	0: CRC功能复位 1: CRC功能无效	RW
b22~b19	Res	保留位	读出时为“1”，写入时写“1”	RW
b18	CTC	CTC复位控制	0: 内部时钟校准器CTC功能复位有效 1: 内部时钟校准器CTC功能复位解除	RW
b17	AOS	AOS复位控制	0: 自动运行系统AOS功能复位有效 1: 自动运行系统AOS功能复位解除	RW
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功能复位有效 1: 时钟控制器CMU中的时钟频率测量模块FCM功能复位解除	RW
b15	Res	保留位	读出时为“1”，写入时写“1”	RW
b14	DMA	DMA复位控制	0: DMA功能复位有效 1: DMA功能复位解除	RW
b13~b12	Res	保留位	读出时为“1”，写入时写“1”	RW
b11	PLA	PLA复位控制	0: PLA功能复位有效 1: PLA功能复位解除	RW
b10~b0	Res	保留位	读出时所有位均为“1”，写入时所有位请写“1”	RW

3.4.2 功能复位控制 1 (RMU_FRST1)

复位值: 0xFFFF FFFX

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															SPI
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res											I2C	Res	MCAN2	MCAN1	

位/位域	标记	位名	功能	读写
b31~b17	Res	保留位	读出时所有位均为“1”，写入时所有位请写“1”	RW
b16	SPI	SPI复位控制	0: 串行外设接口SPI功能复位有效 1: 串行外设接口SPI功能复位解除	RW
b15~b5	Res	保留位	读出时所有位均为“1”，写入时所有位请写“1”	RW
b4	I2C	I2C复位控制	0: I2C功能复位有效 1: I2C功能复位解除	RW
b3	Res	保留位	读出时所有位均为“1”，写入时请写“1”	RW
b2	Res	保留位	读出时为不定，写入时请写“1”	RW
b1	MCAN2	MCAN2复位控制	0: MCAN2功能复位有效 1: MCAN2功能复位解除	RW
b0	MCAN1	MCAN1复位控制	0: MCAN1功能复位有效 1: MCAN1功能复位无效	RW

注:

寄存器 (CMU_CANCKCFGR) 默认选择 XTAL 作为 can 时钟, 如果 XTAL 没有配置发振, bit2-0 为 0; 如果 XTAL 配置好了发振, bit2-0 为 1。

3.4.3 功能复位控制 2 (RMU_FRST2)

复位值: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res							TMRA_5	TMRA_4	TMRA_3	TMRA_2	TMRA_1	Res			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EMB	Res	TMR0_2	TMR0_1	HRPWM	Res		TMR4	Res						TMR6	

位/位域	标记	位名	功能	读写
b31~b25	Res	保留位	读出时所有位均为“1”，写入时所有位请写“1”	RW
b24	TMRA_5	TimerA_5复位控制	0: TimerA单元5功能复位有效 1: TimerA单元5功能复位解除	RW
b23	TMRA_4	TimerA_4复位控制	0: TimerA单元4功能复位有效 1: TimerA单元4功能复位解除	RW
b22	TMRA_3	TimerA_3复位控制	0: TimerA单元3功能复位有效 1: TimerA单元3功能复位解除	RW
b21	TMRA_2	TimerA_2复位控制	0: TimerA单元2功能复位有效 1: TimerA单元2功能复位解除	RW
b20	TMRA_1	TimerA_1复位控制	0: TimerA单元1功能复位有效 1: TimerA单元1功能复位解除	RW
b19~b16	Res	保留位	读出时所有位均为“1”，写入时所有位请写“1”	RW
b15	EMB	EMB复位控制	0: 紧急刹车模块EMB功能复位有效 1: 紧急刹车模块EMB功能复位解除	RW
b14~b13	Res	保留位	读出时所有位均为“1”，写入时所有位请写“1”	RW
b13	TMR0_2	Timer0_2复位控制	0: Timer0单元2功能复位有效 1: Timer0单元2功能复位解除	RW
b12	TMR0_1	Timer0_1复位控制	0: Timer0单元1功能复位有效 1: Timer0单元1功能复位解除	RW
b11	HRPWM	HRPWM复位控制	0: HRPWM单元1功能复位有效 1: HRPWM单元1功能复位解除	RW
b10~b9	Res	保留位	读出时为“1”，写入时写“1”	RW
b8	TMR4	Timer4复位控制	0: Timer4功能复位有效 1: Timer4功能复位解除	RW
b7~b1	Res	保留位	读出时所有位均为“1”，写入时所有位请写“1”	RW
b0	TMR6	Timer6复位控制	0: Timer6单元1234功能复位有效 1: Timer6单元1234功能复位解除	RW

注:

TMR_SYNNR 不受 RMU_FRST 寄存器的控制。

3.4.4 功能复位控制 3 (RMU_FRST3)

复位值: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Res								USART4	USART3	USART2	USART1	Res				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Res						CMP3	CMP12	Res		DAC2	DAC1	Res	ADC3	ADC2	ADC1	

位/位域	标记	位名	功能	读写
b31-b24	Res	保留位	读出时所有位均为“1”，写入时所有位请写“1”	RW
b23	USART4	USART4复位控制	0: 通用同步异步收发器USART单元4功能复位有效 1: 通用同步异步收发器USART单元4功能复位解除	RW
b22	USART3	USART3复位控制	0: 通用同步异步收发器USART单元3功能复位有效 1: 通用同步异步收发器USART单元3功能复位解除	RW
b21	USART2	USART2复位控制	0: 通用同步异步收发器USART单元2功能复位有效 1: 通用同步异步收发器USART单元2功能复位解除	RW
b20	USART1	USART1复位控制	0: 通用同步异步收发器USART单元1功能复位有效 1: 通用同步异步收发器USART单元1功能复位解除	RW
b19-b10	Res	保留位	读出时所有位均为“1”，写入时所有位请写“1”	RW
b9	CMP3	CMP3复位控制	0: 电压比较器CMP单元3功能复位有效 1: 电压比较器CMP单元3功能复位解除	RW
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	RW
b7-b6	Res	保留位	读出时所有位均为“1”，写入时所有位请写“1”	RW
b5	DAC2	DAC2复位控制	0: 数模转换器DAC2功能复位有效 1: 数模转换器DAC2功能复位解除	RW
b4	DAC1	DAC1复位控制	0: 数模转换器DAC1功能复位有效 1: 数模转换器DAC1功能复位解除	RW
b3	Res	保留位	读出时为“1”，写入时写“1”	RW
b2	ADC3	ADC3复位控制	0: 模数转换模块ADC单元3功能复位有效 1: 模数转换模块ADC单元3功能复位解除	RW
b1	ADC2	ADC2复位控制	0: 模数转换模块ADC单元2功能复位有效 1: 模数转换模块ADC单元2功能复位解除	RW
b0	ADC1	ADC1复位控制	0: 模数转换模块ADC单元1功能复位有效 1: 模数转换模块ADC单元1功能复位解除	RW

3.4.5 复位控制寄存器 (RMU_PRSTCR0)

复位值: 0x40

b7	b6	b5	b4	b3	b2	b1	b0
Res		LKUPREN	Res				

位/位域	标记	位名	功能	读写
b7	Res	保留位	读出时为“0”，写入时写“0”	RW
b6	Res	保留位	读出为“1”，写入“1”	RW
b5	LKUPREN	Lockup复位使能	0: Lockup复位无效 1: Lockup复位使能	RW
b4-b0	Res	保留位	读出时为“0”，写入时写“0”	RW

3.4.6 复位标志寄存器 (RMU_RSTF0)

复位值: 0xXXXX XXXX (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	Res													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	LKUPRF	XTALRF	CKFERF	RAECRF	RAPERF	MPUERF	SWRF	PDRF	SWDRF	WDRF	PVD2RF	PVD1RF	BORF	PINRF	PORF

位/位域	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	RW
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	RW
b29~b15	Res	保留位	读出时为“0”, 写入时写“0”	RW
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	RW
b13	XTALRF	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	RW
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	RW
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	RW
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	RW
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	RW
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	RW
b7	PDRF	掉电模式复位	发生掉电模式复位时, 由硬件置位。	RW

			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	
b6	SWDRF	专用看门狗复位标志	发生专用看门狗复位时, 由硬件置位。 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	RW
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	RW
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	RW
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	RW
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	RW
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	RW
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	RW

4 时钟控制器 (CMU)

4.1 概述

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。系统时钟的最大运行时钟频率可以达到 120MHz，有 6 个可选择的时钟源：

- 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)

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。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、内部低速振荡器或者 XTAL 小数分频时钟作为时钟源。

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

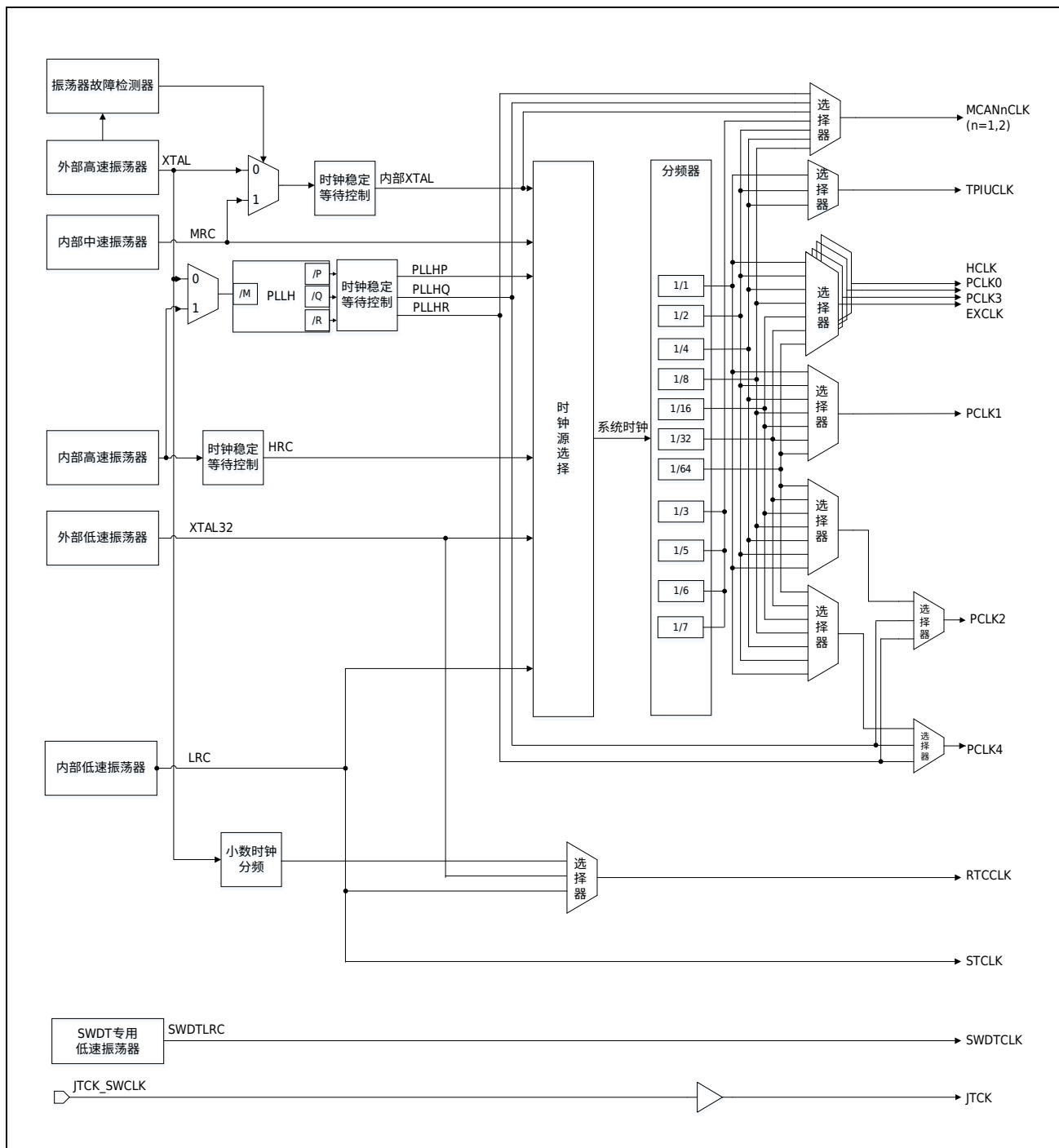


图 4-1 时钟系统框图

4.2.2 时钟频率测量框图

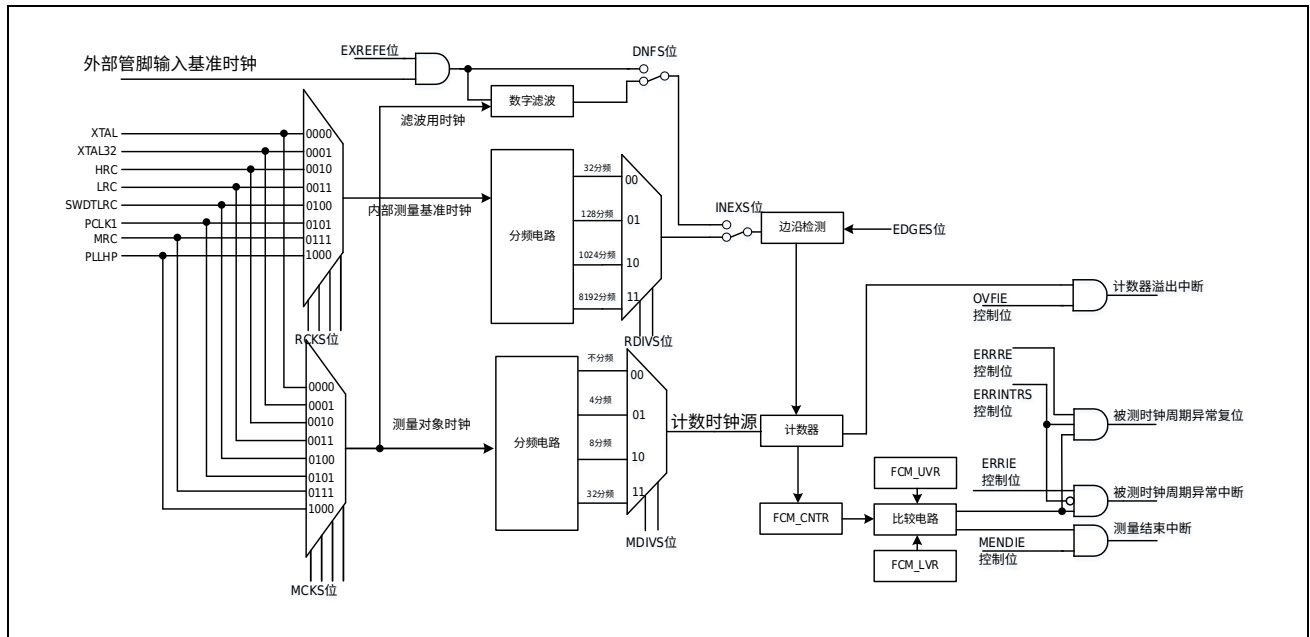


图 4-2 时钟频率测量框图

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、EFM、SRAM0、SRAMH、Ret SRAM、MPU、GPIO、INTC	最高频率 120MHz。 由 CMU_SCFGR 寄存器 HCLKS 位配置。 可选系统时钟的分频：1, 2, 4, 8, 16, 32, 64
PCLK0	HRPWM_n (n=1~6)、Timer6_n (n=1~4)、Timer4、TimerA_n (n=1~4)、EMB、I2C、CMP12、CMP3、MCANn(n=1~2)控制逻辑	最高频率 120MHz 由 CMU_SCFGR 寄存器 PCLK0S 位配置。 可选系统时钟的分频：1, 2, 4, 8, 16, 32, 64
PCLK1	USARTn (n=1~4)、SPI、Timer0_n (n=1~2)、TimerA_5、CRC	最高频率 60MHz 由 CMU_SCFGR 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_SCFGR 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、FCM、CTC、PLA	最高频率 60MHz 由 CMU_SCFGR 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)、DAC (控制逻辑)	最高频率 60MHz 由 CMU_SCFGR 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
MCANnCLK	MCAN1/MCAN2 通信时钟	最高频率 80MHz 由 CMU_CANCKCFGR 寄存器的 MCAN2CKS/MCAN1CKS 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, 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

时钟	作用范围	规格
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：PCLK4 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。

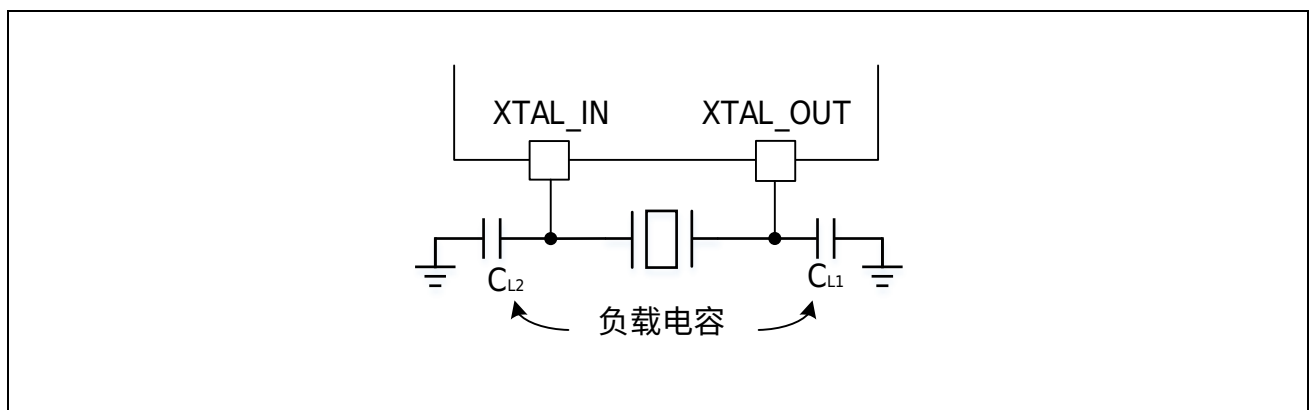


图 4-3 外部高速振荡器连接事例

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图 4-4 所示。

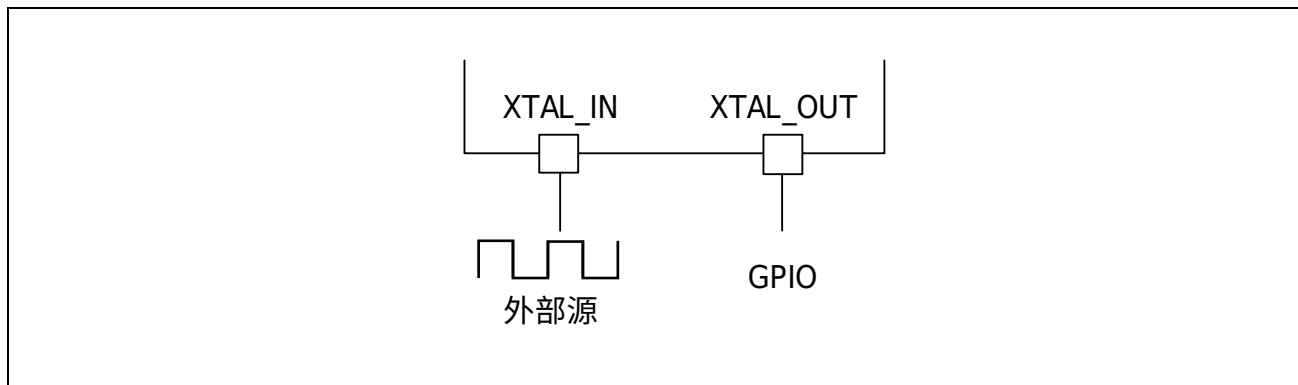


图 4-4 外部时钟输入的连接事例图

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，动作波形如图 4-5 所示。操作流程参照【检测到 XTAL 故障检测动作】。

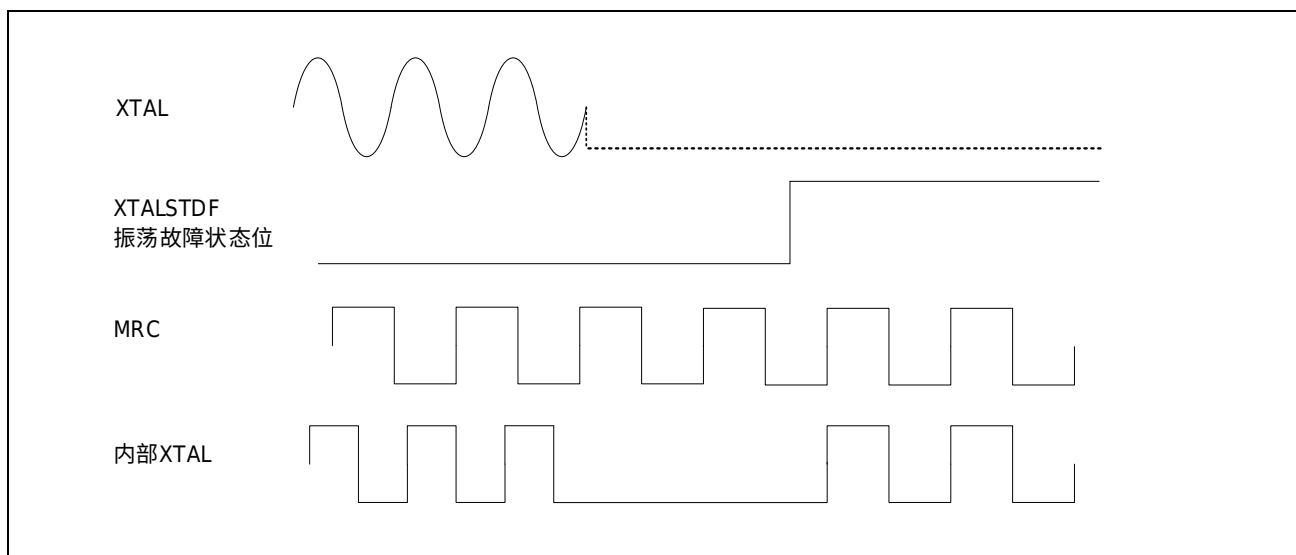


图 4-5 外部高速振荡器故障检测例

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 Hi-Z 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，动作例如图 4-6 所示。

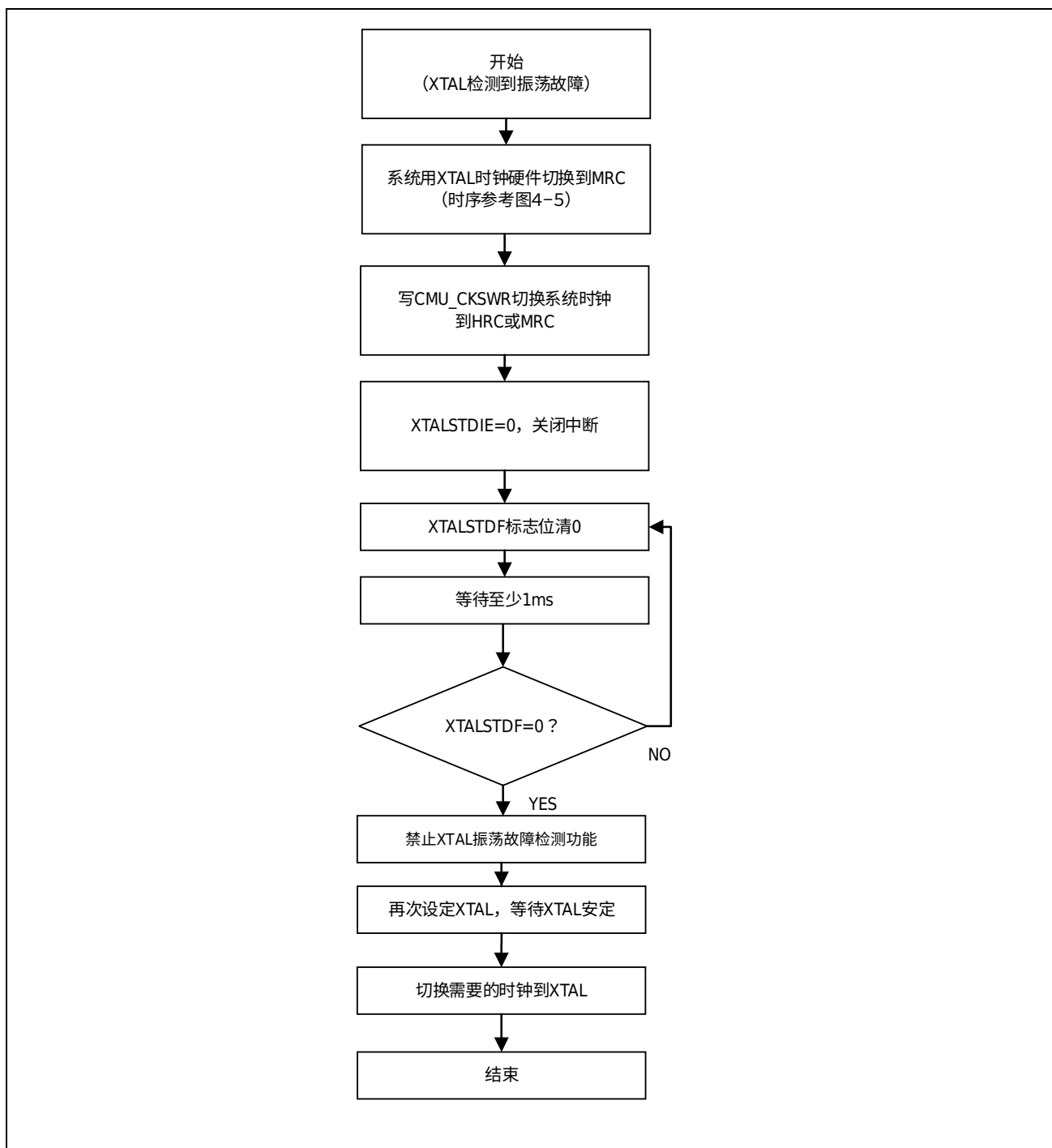


图 4-6 系统时钟选择 XTAL，检测到 XTAL 振荡故障例

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不可屏蔽中断，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：

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

NUMER[16:0]寄存器设定为 0x1312D，DENOM[10:0]寄存器设定为 0x100，即寄存器 CMU_XTALDIVR 设定为 0x1312 D100，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：

$$\text{分频数} = \frac{8000000\text{Hz}}{32768\text{Hz}} = \frac{31250 \text{ (10 进制)}}{128 \text{ (10 进制)}} = \frac{7\text{A}12 \text{ (16 进制)}}{80 \text{ (16 进制)}}$$

NUMER[16:0]寄存器设定为 0x07A12，DENOM[10:0]寄存器设定为 0x080，即寄存器 CMU_XTALDIVR 设定为 0x07A1 2080，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。

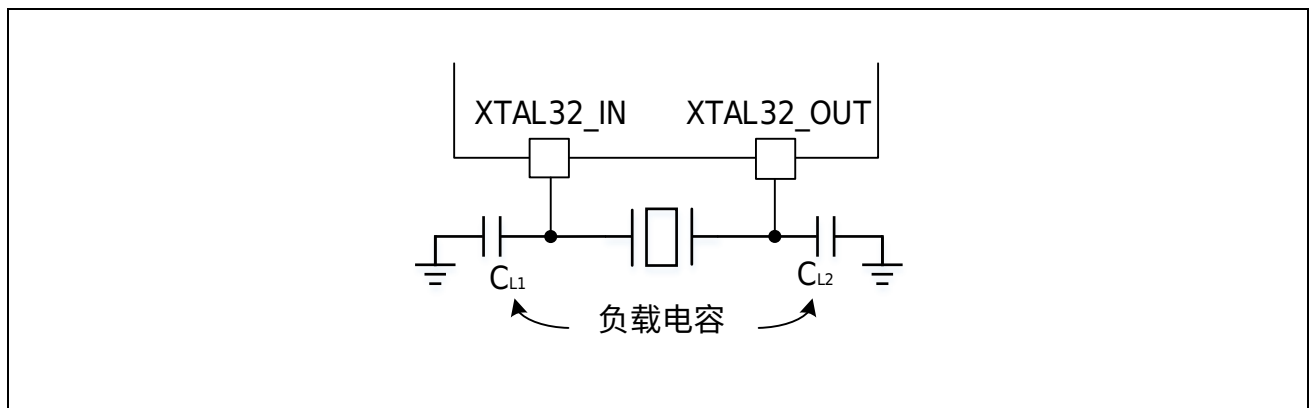


图 4-7 外部低速振荡器连接事例

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**对应数据手册电气特性**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，保证精度参照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，保证精度参照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 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，保证精度参照**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，保证精度参照**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

HC32F334 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 120MHz）
- 三个输出都可用于生成 ADC、MCAN 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/ SRAM 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

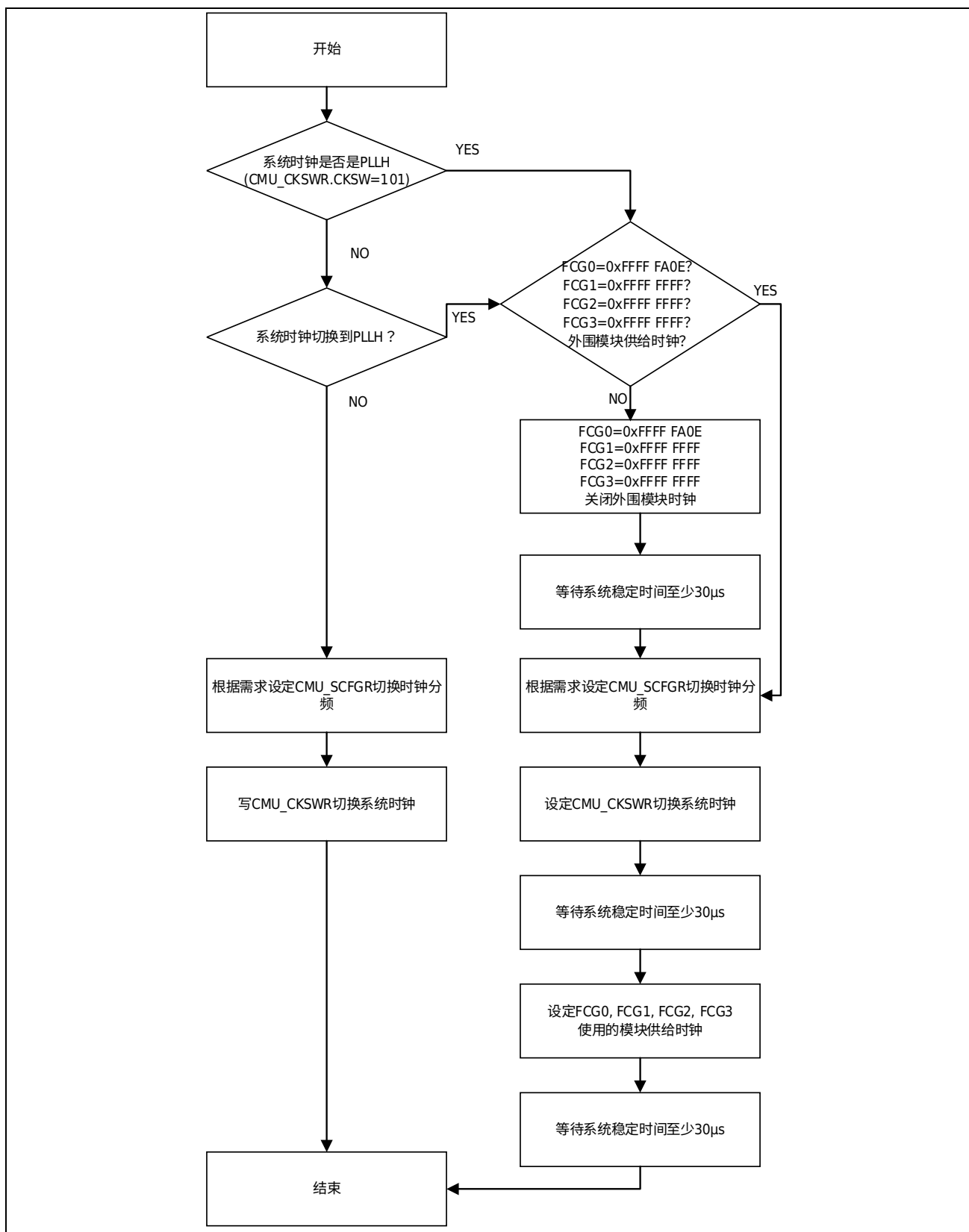


图 4-8 时钟源切换

4.8.2 时钟分频切换

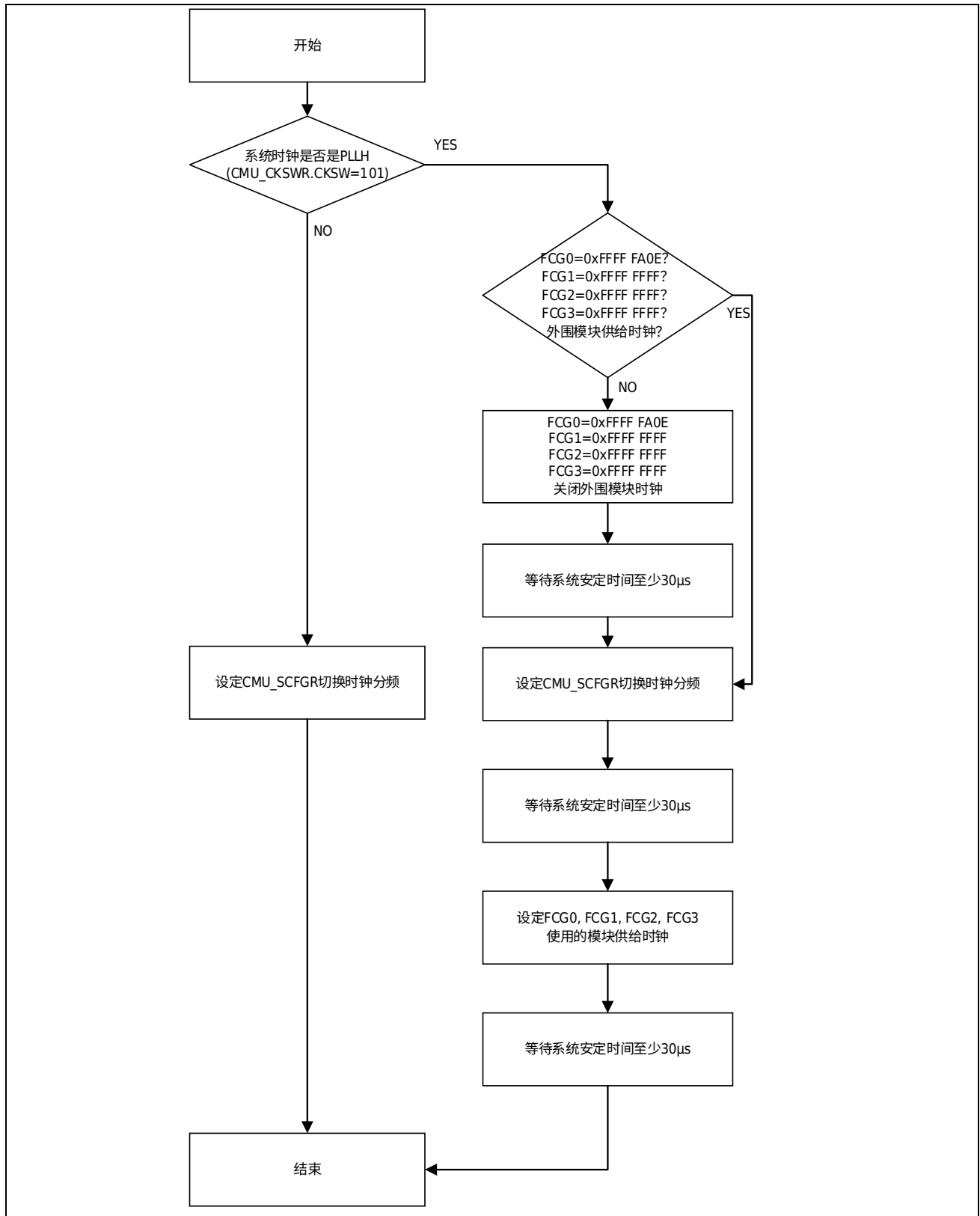


图 4-9 时钟分频切换

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（1、2、4、8、...、128 分频）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_MCO1CFGR.MCO1SEL 位选择。

■ MCO_2

用户可通过可配置的预分配器（1、2、4、8、...、128 分频）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_MCO2CFGR.MCO2SEL 位选择。

MCO_1/ MCO_2 输出时钟不得超过 100MHz（最大 I/O 速度）。

4.10 时钟频率测量

4.10.1 时钟频率测量

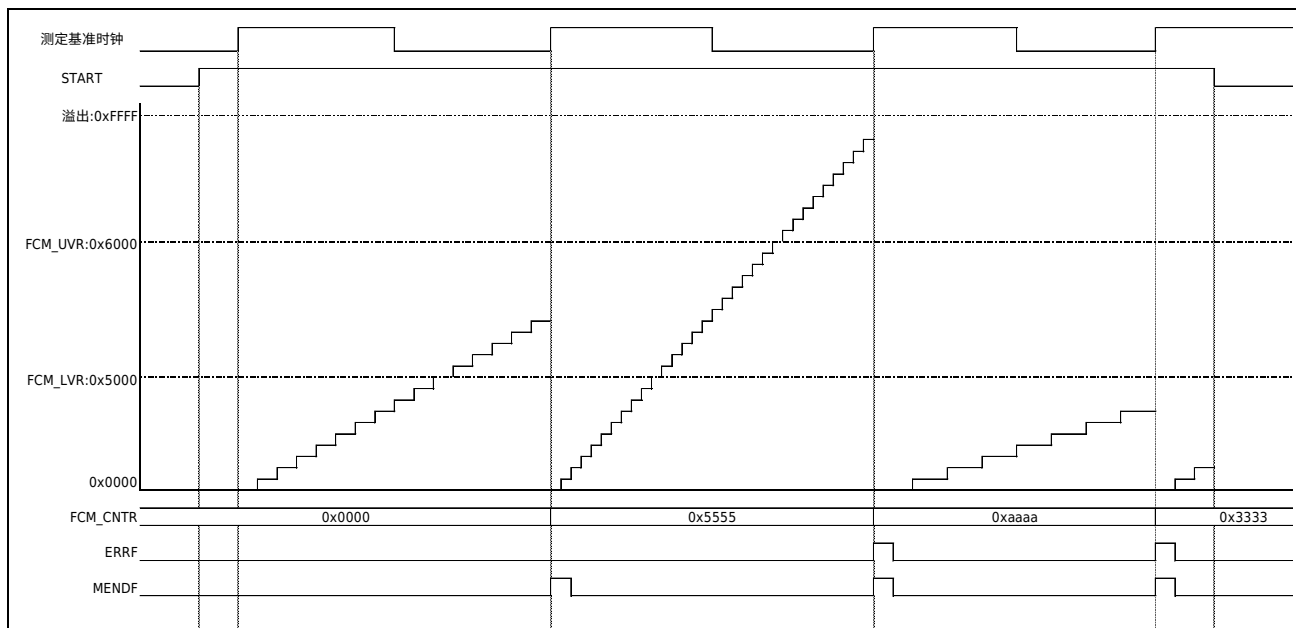


图 4-10 时钟频率测量时序图

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：

频率异常复位

4.11 寄存器描述

表 4-3 CMU 寄存器列表

基地址 1: 0x4004 8400

寄存器	描述	偏移地址	位宽	复位值
FCM_LVR	FCM下限比较值寄存器	0x00	32	0x0000 0000
FCM_UVR	FCM上限比较值寄存器	0x04	32	0x0000 0000
FCM_CNTR	FCM计数器值寄存器	0x08	32	0x0000 0000
FCM_STR	FCM开始停止寄存器	0x0C	32	0x0000 0000
FCM_MCCR	FCM测量对象控制寄存器	0x10	32	0x0000 0000
FCM_RCCR	FCM测量基准控制寄存器	0x14	32	0x0000 0000
FCM_RIER	FCM中断复位控制寄存器	0x18	32	0x0000 0000
FCM_SR	FCM标志寄存器	0x1C	32	0x0000 0000
FCM_CLR	FCM标志位清除寄存器	0x20	32	0x0000 0000

基地址 2: 0x4004 CC00

寄存器	描述	偏移地址	位宽	复位值
CMU_XTALCFGR	CMU XTAL配置寄存器	0x50	8	0x80
CMU_XTAL32CR	CMU XTAL32控制寄存器	0x54	8	0x00
CMU_XTAL32CFGR	CMU XTAL32配置寄存器	0x58	8	0x00
CMU_XTAL32NFR	CMU XTAL32滤波寄存器	0x68	8	0x00
CMU_LRCCR	CMU LRC控制寄存器	0x6C	8	0x00
CMU_LRCTRM	CMU LRC校准寄存器	0x74	8	0x00

基地址 3: 0x4004 8000

寄存器	描述	偏移地址	位宽	复位值
CMU_XTALDIVR	CMU XTAL小数分频配置寄存器	0x18	32	0x03D0 9040
CMU_XTALDIVCR	CMU XTAL小数分频控制寄存器	0x1C	32	0x0000 0000

基地址 4: 0x4005 4000

寄存器	描述	偏移地址	位宽	复位值
CMU_XTALSTBCR	CMU XTAL稳定配置寄存器	0x0A2	8	0x05
CMU_XTALCR	CMU XTAL控制寄存器	0x032	8	0x01
CMU_XTALSTDCR	CMU XTAL振荡故障控制寄存器	0x040	8	0x00
CMU_XTALSTDSR	CMU XTAL振荡故障状态寄存器	0x041	8	0x00
CMU_HRCCR	CMU HRC控制寄存器	0x036	8	由ICG1.HRCSTP值决定
CMU_HRCTRM	CMU HRC校准寄存器	0x062	8	0x00
CMU_MRCCR	CMU MRC控制寄存器	0x038	8	0x80
CMU_MRCTRM	CMU MRC校准寄存器	0x061	8	0x00
CMU_PLLHCFGR	CMU PLLH配置寄存器	0x100	32	0x1110 1300
CMU_PLLHCR	CMU PLLH控制寄存器	0x02A	8	0x01
CMU_OSCSTBSR	CMU 时钟源稳定状态寄存器	0x03C	8	0x00
CMU_CKSWR	CMU 系统时钟源切换寄存器	0x026	8	0x01
CMU_SCFGR	CMU 时钟分频配置寄存器	0x020	32	0x0000 0000
CMU_CANCKCFGR	CMU CAN时钟配置寄存器	0x018	8	0xDD
CMU_PERICKSEL	CMU AD/DA时钟配置寄存器	0x010	16	0x0000
CMU_TPIUCKCFGR	CMU 调试时钟配置寄存器	0x03F	8	0x00
CMU_MCO1CFGR	CMU MCO1时钟输出配置寄存器	0x03D	8	0x00
CMU_MCO2CFGR	CMU MCO2时钟输出配置寄存器	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
Res	XTALDRV[1:0]		XTALMS		Res		

位/位域	标记	位名	功能	读写
b7	Res	保留位	写入时写“1”	W
b6	Res	保留位	写入时写“0”	RW
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: 低驱动能力 (建议8~16MHz晶振) 11: 超低驱动能力 (建议4~8MHz晶振)	RW
b3	XTALMS	XTAL模式选择位	0: 振荡器模式 1: 外部时钟输入模式	RW
b2~b0	Res	保留位	读出时为“0”，写入时写“0”	RW

4.11.2 CMU XTAL 稳定配置寄存器 (CMU_XTALSTBCR)

复位值: 0x05

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

位/位域	标记	位名	功能	读写
b7~b4	Res	保留位	读出时为“0”，写入时写“0”	RW
b3~b0	XTALSTB[3:0]	XTAL稳定时间选择	0001: 稳定计数器35周期	RW
			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	
			CMU_XTALCR.XTALSTP位1且CMU_OSCSTBSR.XTALSTBF位为0的状态下配置此寄存器。	

4.11.3 CMU XTAL 小数分频配置寄存器 (CMU_XTALDIVR)

复位值: 0x03D0 9040

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

位/位域	标记	位名	功能	读写
b31~b29	Res	保留位	读出时为“0”，写入时写“0”	RW
b28~b12	NUMER[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	RW
b11	Res	保留位	读出时为“0”，写入时写“0”	RW
b10~b0	DENOM[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。	RW

4.11.4 CMU XTAL 小数分频控制寄存器 (CMU_XTALDIVCR)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

4.11.5 CMU XTAL 控制寄存器 (CMU_XTALCR)

复位值: 0x01

b7	b6	b5	b4	b3	b2	b1	b0
Res							XTALSTP

位/位域	标记	位名	功能	读写
b7~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

注意:

1. 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。
2. 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。
3. 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_XTALSTDCR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
XTALSTDE	Res			XTALSTDRIS	XTALSTDRE	XTALSTDIE	

位/位域	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。	RW
b6~b3	Res	保留位	读出时为“0”, 写入时写“0”	RW
b2	XTALSTDRIS	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。	RW
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	RW
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Hi-Z输出, XTALSTDIE位需要设置成1。	RW

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
Res							XTALSTDF

位/位域	标记	位名	功能	读写
b7~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
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。	RW

4.11.8 CMU XTAL32 配置寄存器 (CMU_XTAL32CFGR)

复位值: 0xXX

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

位/位域	标记	位名	功能	读写
b7~b3	Res	保留位	读出时为“0”，写入时写“0”	RW
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: 使用方法参考数据手册电气特性章节【晶振/陶瓷谐振器产生的低速外部时钟】	RW

4.11.9 CMU XTAL32 滤波寄存器 (CMU_XTAL32NFR)

复位值: 0xXX

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

位/位域	标记	位名	功能	读写
b7~b2	Res	保留位	读出时为“0”，写入时写“0”	RW
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3μs滤波有效 01: RUN模式XTAL32的3μs滤波有效，停止模式或掉电模式 XTAL32的3μs滤波无效 10: 禁止设定 11: RUN模式/停止模式/掉电模式，XTAL32的3μs滤波无效	RW

4.11.10 CMU XTAL32 控制寄存器 (CMU_XTAL32CR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res						XTAL32STP	

位/位域	标记	位名	功能	读写
b7~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

注意:

1. XTAL32 选作系统时钟源时，禁止 XTAL32STP 写 1 停止 XTAL32 振荡器。
2. 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。
3. 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/位域	标记	位名	功能	读写
			频率校准需在 HRC 频率保证范围内。	
			10000000: -128	
			10000001: -127	
			
b7~b0	HRCTRM[7:0]	HRC频率校准位	11111111: -1	RW
			00000000: 中心Code	
			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
Res							HRCSTP

位/位域	标记	位名	功能	读写
b31~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
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。	RW

注意:

1. 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。
2. 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。
3. 软件设定 HRC 停止, 通过 HRCSTBF 位确认 HRC 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_MRCTRM)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
MRCTRM[7:0]							

位/位域	标记	位名	功能	读写
			10000000: -128	
			10000001: -127	
			
			11111111: -1	
b7~b0	MRCTRM[7:0]	MRC频率校准位	00000000: 中心Code	RW
			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
Res							MRCSTP

位/位域	标记	位名	功能	读写
b7	Res	保留位	读出时为“1”，写入时写“1”	RW
b6~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
			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。	RW

注意:

1. MRC 选作系统时钟源时, 禁止 MRCSTP 写 1 停止 MRC 时钟。
2. 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。
3. 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。
4. 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, 无视 MRCSTP 位设定, MRC 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/位域	标记	位名	功能	读写
			10000000: -128	
			10000001: -127	
			
			11111111: -1	
b7~b0	LRCTRM[7:0]	LRC频率校准位	00000000: 中心Code	RW
			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
Res							LRCSTP

位/位域	标记	位名	功能	读写
b7~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
b0	LRCSTP	LRC振荡器开启停止位	0: LRC振荡器振荡 1: LRC振荡器停止	RW

注意:

1. LRC 选作系统时钟源时，禁止 LRCSTP 写 1 停止 LRC 时钟。
2. 软件设定 LRC 振荡，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。
3. 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。
4. 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: 0x1110 1300

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]				Res		PLLHN[8]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PLLHN[7:0]							PLLS RC	Res	PLLHM[1:0]						

位/位域	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	RW
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	RW
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	RW
b19~b17	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

			111011101: 478	
			111011110: 479	
			111011111: 480	
b7	PLLSRC	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	RW
b6~b5	Res	保留位	读出时为“0”，写入时写“0”	RW
b4~b0	PLLHM[4:0]	PLLH输入时钟分频系数	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。 00000: 1分频 00001: 2分频 00010: 3分频 10111: 24分频 其他: 禁止设定	RW

4.11.18 CMU PLLH 控制寄存器 (CMU_PLLHCR)

复位值: 0x01

b7	b6	b5	b4	b3	b2	b1	b0
Res							PLLHOFF

位/位域	标记	位名	功能	读写
b7~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

注意:

1. PLLH 选作系统时钟源时，禁止 PLLHOFF 写 1 停止 PLLH 时钟。
2. 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。
3. 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。
4. 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
Res		PLLHSTBF	Res	XTALSTBF	Res		HRCSTBF

位/位域	标记	位名	功能	读写
b7~b6	Res	保留位	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	Res	保留位	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	Res	保留位	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
Res					CKSW[2:0]		

位/位域	标记	位名	功能	读写
b31~b3	Res	保留位	读出时为“0”，写入时写“0”	RW
b2~b0	CKSW[2:0]	系统时钟源切换	000: 选择HRC时钟作为系统时钟 001: 选择MRC时钟作为系统时钟 010: 选择LRC时钟作为系统时钟 011: 选择XTAL时钟作为系统时钟 100: 选择XTAL32时钟作为系统时钟 101: 选择PLLH作为系统时钟 110: 禁止设定 111: 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。	RW

4.11.21 CMU 时钟分频配置寄存器 (CMU_SCFGR)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b27	Res	保留位	读出时为“0”，写入时写“0”	RW
b26~b24	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。	RW
b23	Res	保留位	读出时为“0”，写入时写“0”	RW
b22~b20	EXCKS[2:0]	EXCLK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。	RW
b19	Res	保留位	读出时为“0”，写入时写“0”	RW
b18~b16	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。	RW
b15	Res	保留位	读出时为“0”，写入时写“0”	RW

			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, PCLK3为系统时钟的1分频。	
b14~b12	PCLK3S[2:0]	PCLK3时钟分频选择位		RW
b11	Res	保留位	读出时为“0”, 写入时写“0”	RW
			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。	
b10~b8	PCLK2S[2:0]	PCLK2时钟分频选择位		RW
b7	Res	保留位	读出时为“0”, 写入时写“0”	RW
			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。	
b6~b4	PCLK1S[2:0]	PCLK1时钟分频选择位		RW
b3	Res	保留位	读出时为“0”, 写入时写“0”	RW
			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。	
b2~b0	PCLK0S[2:0]	PCLK0时钟分频选择位		RW

4.11.22 CMU CAN 时钟配置寄存器 (CMU_CANCKCFGR)

复位值: 0xDD

b7	b6	b5	b4	b3	b2	b1	b0
MCAN2CKS[3:0]				MCAN1CKS[3:0]			

位/位域	标记	位名	功能	读写
			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	
b7~b4	MCAN2CKS[3:0]	MCAN2通讯时钟选择	1001: PLLH/R 1101: XTAL 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. 系统时钟选择PLLH时, 需要关闭各外设模块时钟, 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, MCANCLK为XTAL时钟源。	RW
			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	
b3~b0	MCAN1CKS[3:0]	MCAN1通讯时钟选择	1001: PLLH/R 1101: XTAL 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. 系统时钟选择PLLH时, 需要关闭各外设模块时钟, 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30μs。 3. PWC_STPMCR.CKSMRC位为1时, 停止模式唤醒后, 此寄存器初始化, MCANCLK为XTAL时钟源。	RW

4.11.23 CMU ADC/DAC 时钟配置寄存器 (CMU_PERICKSEL)

复位值: 0x0000

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

位/位域	标记	位名	功能	读写
b15~b4	Res	保留位	读出时为“0”，写入时写“0”	RW
b3~b0	PERICKSEL[3:0]	ADC/ DAC时钟源选择	0000: CMU_SCFGR设定的PCLK2/ PCLK4 1000: PCLK2/PCLK4配置为PLLHQ 1001: PCLK2/PCLK4配置为PLLHR 其他: 禁止设定。	RW

注意:

- 切换的目标时钟源为 *PLLH* 时，需保证在 *PLLH* 时钟稳定状态。

4.11.24 CMU 调试时钟配置寄存器 (CMU_TPIUCKCFGR)

复位值: 0x00

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

位/位域	标记	位名	功能	读写
b7	TPIUCKOE	TPIU时钟供给允许位	0: 禁止 1: 允许	RW
b6~b2	Res	保留位	读出时为“0”，写入时写“0”	RW
b1~b0	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	RW

4.11.25 CMU MCO1 时钟输出配置寄存器 (CMU_MCO1CFGR)

复位值: 0x00

	b7	b6	b5	b4	b3	b2	b1	b0
	MCO1EN	MCO1DIV[2:0]			MCO1SEL[3:0]			

位/位域	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	RW
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	RW
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。	RW

4.11.26 CMU MCO2 时钟输出配置寄存器 (CMU_MCO2CFGR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
MCO2EN	MCO2DIV[2:0]			MCO2SEL[3:0]			

位/位域	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	RW
b6~b4	MCO2DIV[2:0]	MCO_2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	RW
b3~b0	MCO2SEL[3:0]	MCO_2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。	RW

4.11.27 FCM 下限比较值寄存器 (FCM_LVR)

复位值: 0x0000 0000

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

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

4.11.28 FCM 上限比较值寄存器 (FCM_UVR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UVR[15:0]															
位/位域	标记	位名	功能												读写
b31~b16	Res	保留位	读出时为“0”，写入时写“0”												RW
b15~b0	UVR[15:0]	上限比较值	START位为0时配置此寄存器。												RW

4.11.29 FCM 计数器值寄存器 (FCM_CNTR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNTR[15:0]															
位/位域	标记	位名	功能												读写
b31~b16	Res	保留位	读出时为“0”，写入时写“0”												RW
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

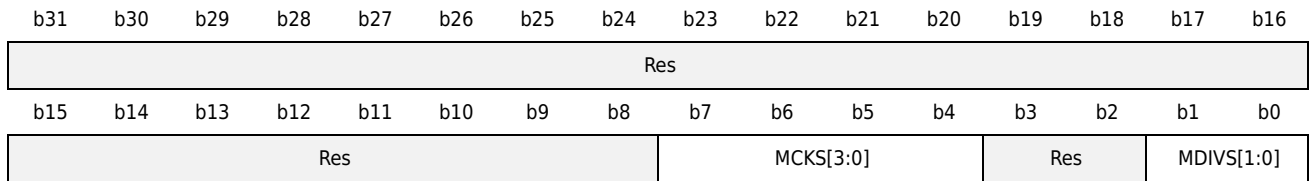
4.11.30 FCM 开始停止寄存器 (FCM_STR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															STAR T
位/位域	标记	位名	功能												读写
b31~b1	Res	保留位	读出时为“0”，写入时写“0”												RW
b0	START	频率测量开始位	0: 频率测量停止 1: 频率测量开始												RW

4.11.31 FCM 测量对象控制寄存器 (FCM_MCCR)

复位值: 0x0000 0000



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

4.11.32 FCM 测量基准控制寄存器 (FCM_RCCR)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b14	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b11~b10	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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: 禁止设定	RW
b2	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

4.11.33 FCM 中断复位控制寄存器 (FCM_RIER)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	读出时为“0”，写入时写“0”	RW
b7	ERRE	频率异常复位允许位	0: 禁止 1: 允许	RW
b6~b5	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b3	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW

4.11.34 FCM 标志寄存器 (FCM_SR)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	读出时为“0”，写入时写“0”	RW
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

4.11.35 FCM 标志位清除寄存器 (FCM_CLR)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b3	Res	保留位	读出时为“0”，写入时写“0”	RW
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 概述

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、切换和检测。电源控制器由低功耗控制逻辑 (PWCL) 和电源电压检测单元 (PVD) 构成。

芯片的工作电压 (VCC) 为 1.8V 到 3.63V。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) 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。

VDDR 区域在芯片进入掉电模式通过 RLDO 维持电源, 保持 4KB 的 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。

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_PWRC5.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)、模拟系的输入输出管脚构成, 由 AVCC/AVSS 引脚供电。

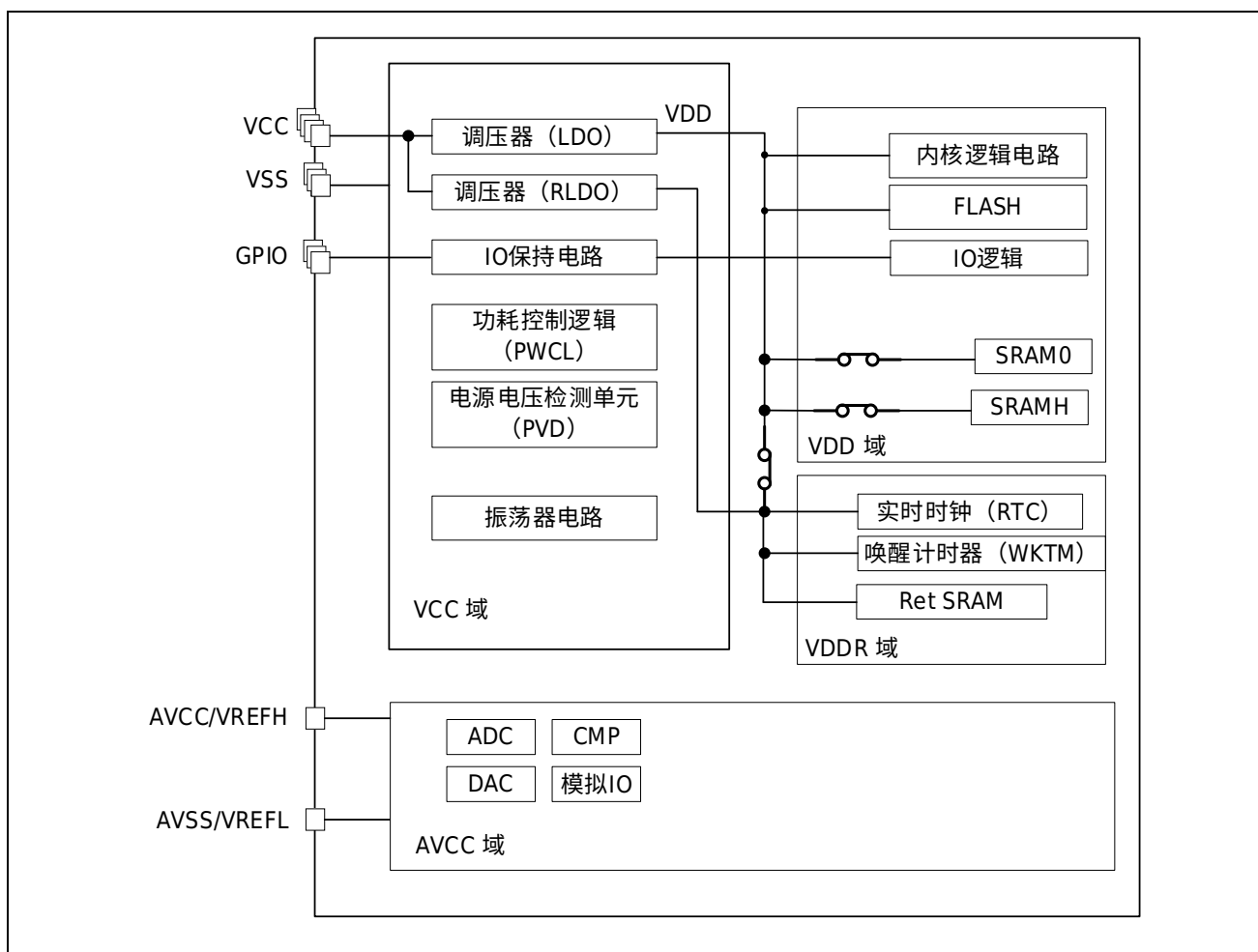


图 5-1 电源构成图

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。

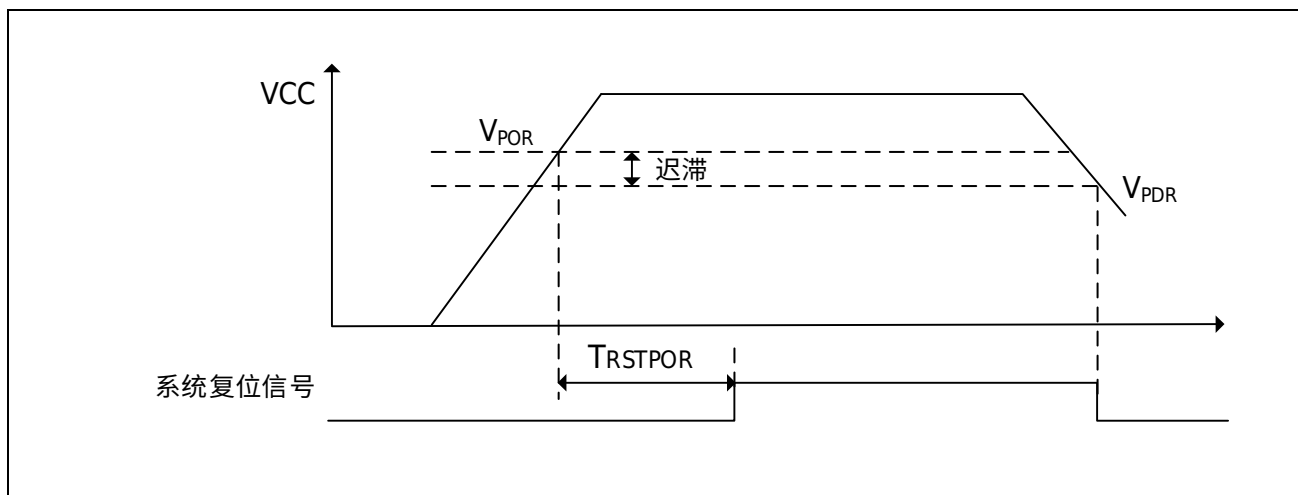


图 5-2 上电复位、掉电复位波形

5.3.2 欠压复位 (BOR) 说明

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

V_{BOR} 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，请参考数据手册电气特性章节。

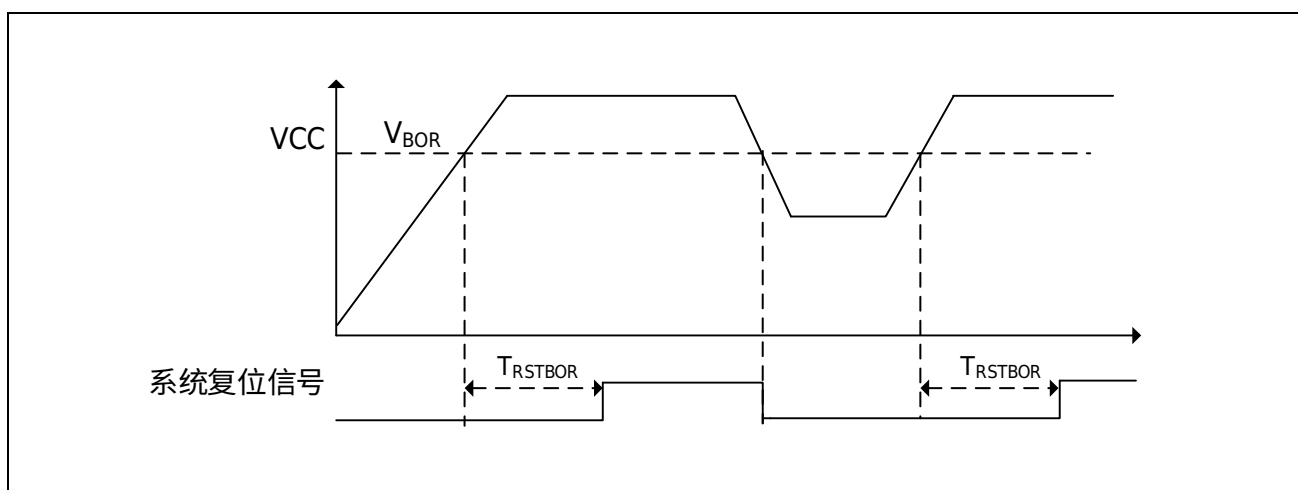


图 5-3 欠压复位波形

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/不可屏蔽中断/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]=0b111)
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中断或非可屏蔽中断	配置成电压检测2中断或非可屏蔽中断
	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)	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、PVD2 中断/复位框图

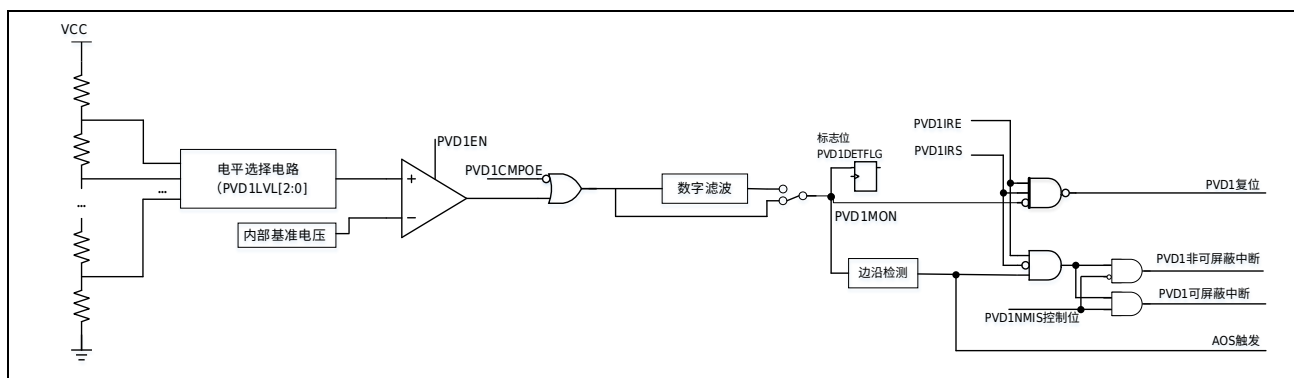


图 5-4 PVD1 中断/复位框图

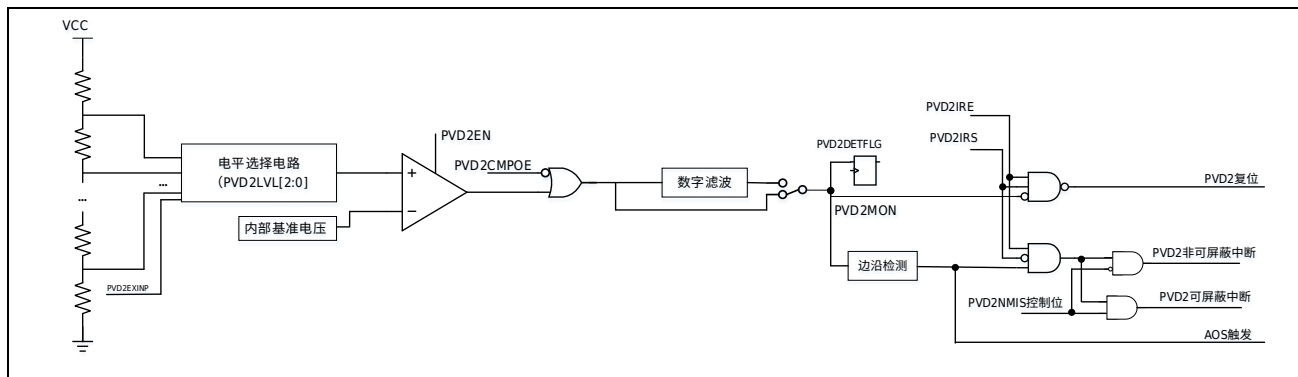


图 5-5 PVD2 中断/复位框图

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。

1. 停止模式

- 1) 必须将数字滤波器无效。

2. 掉电模式

- 1) 必须将数字滤波器无效。
- 2) 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。

图 5-6 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。

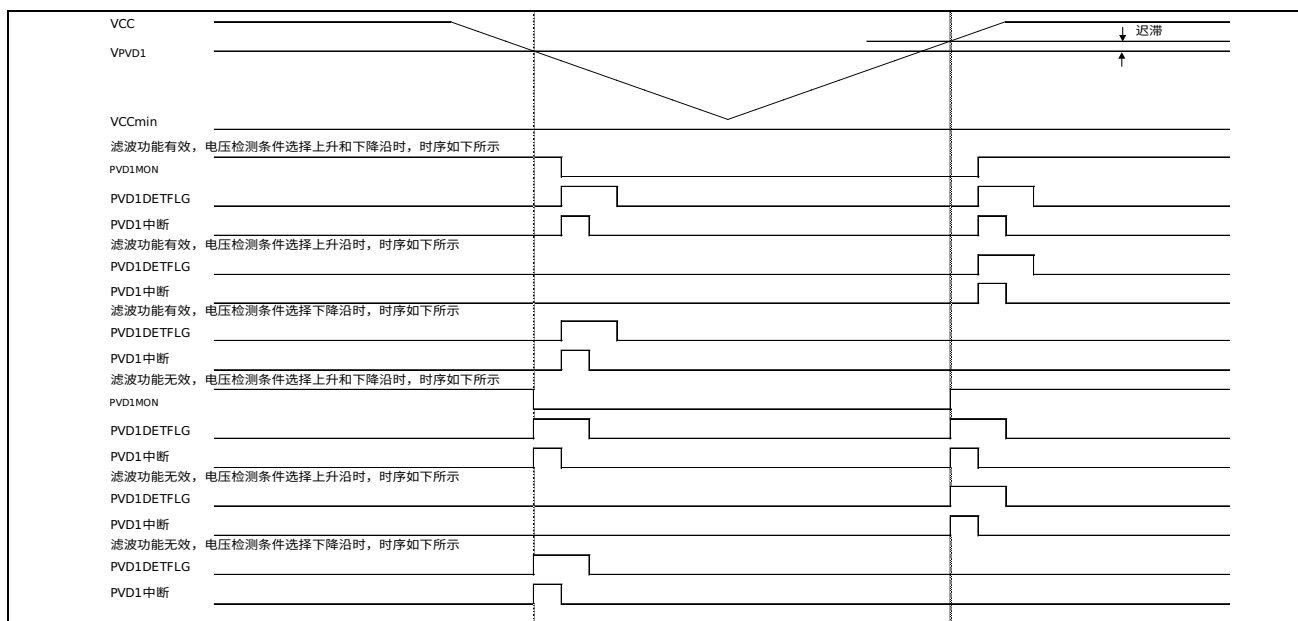


图 5-6 电压检测 1 中断时序图

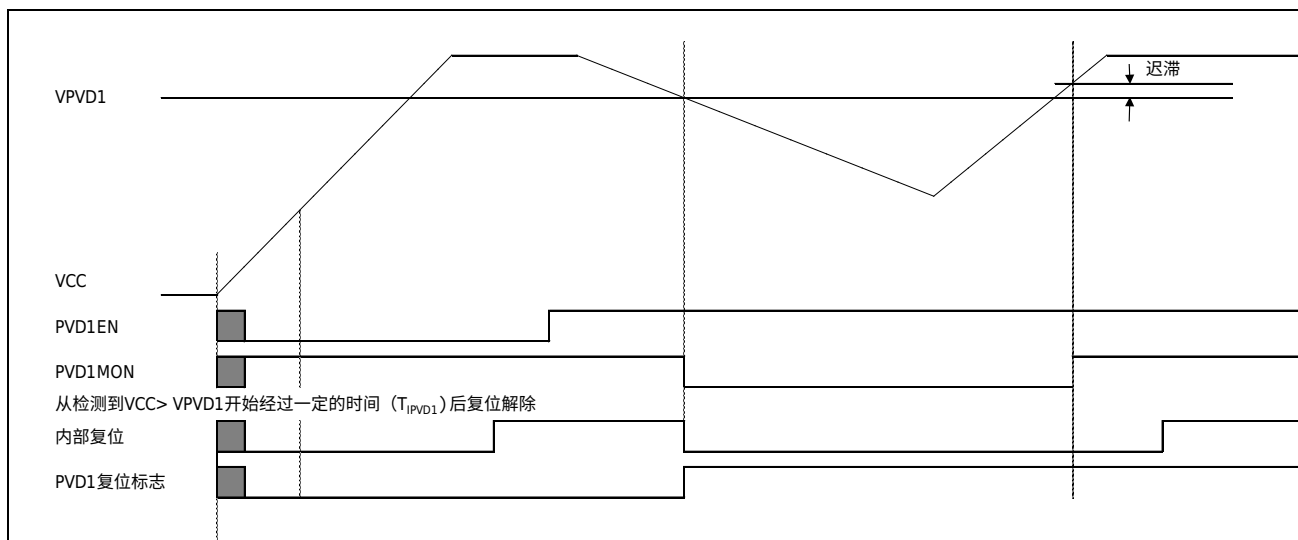


图 5-7 电压检测 1 复位时序图

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：

1. 停止模式
 - 1) 必须将数字滤波器无效。
2. 掉电模式
 - 1) 必须将数字滤波器无效。
 - 2) 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。

图 5-8 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。

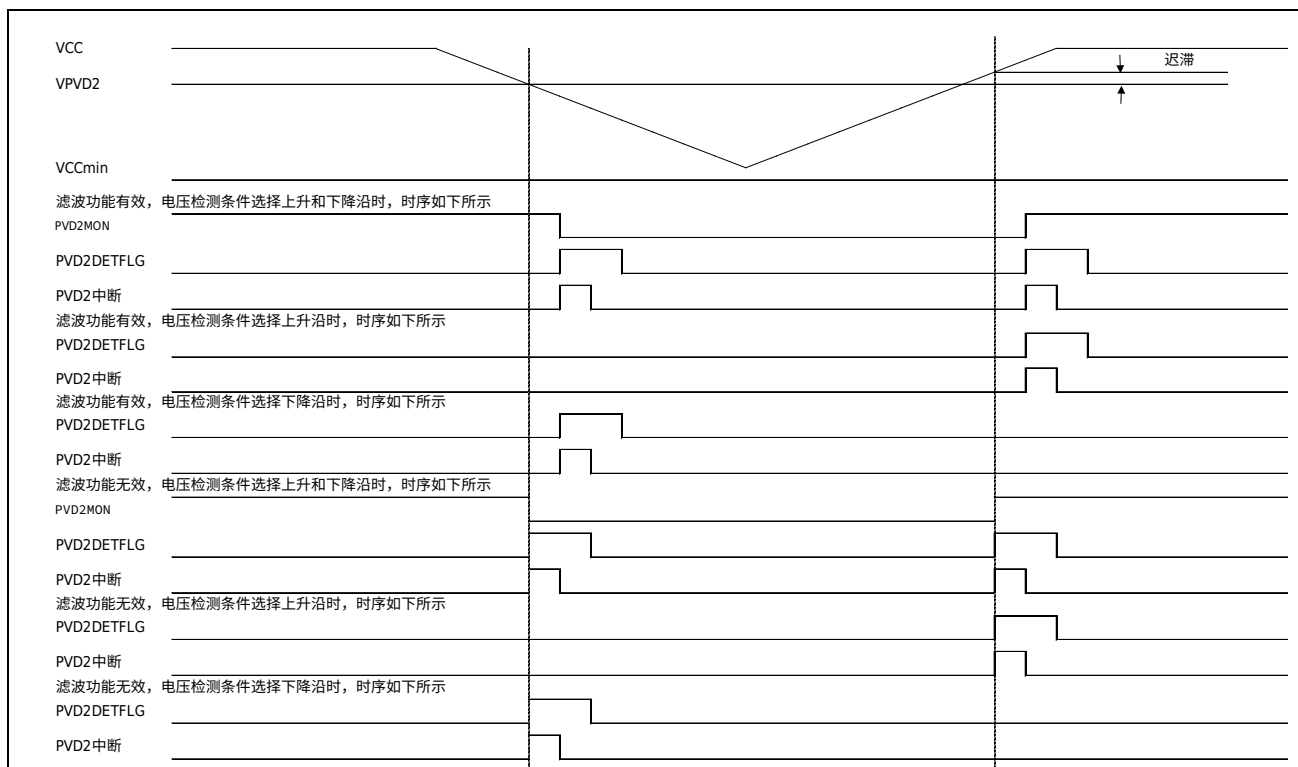


图 5-8 电压检测 2 中断运行时序图

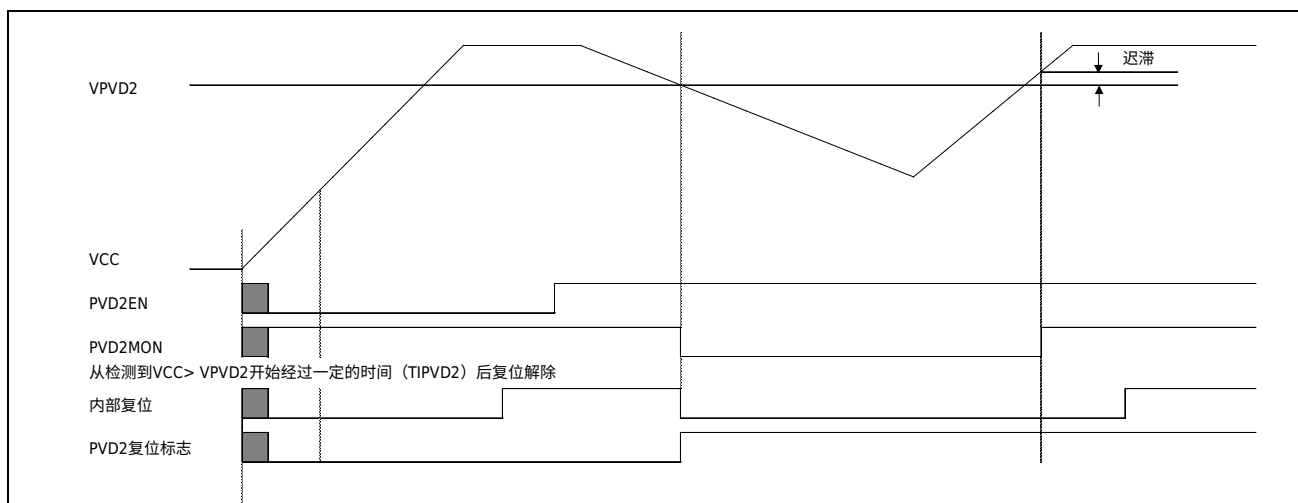


图 5-9 电压检测 2 复位运行时序图

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包括基准电压测量, AVCC 电压测量两个功能。基准电压测量通路, 是使用 ADC 测量基准电压的功能。内部基准电压约为 1.10V。

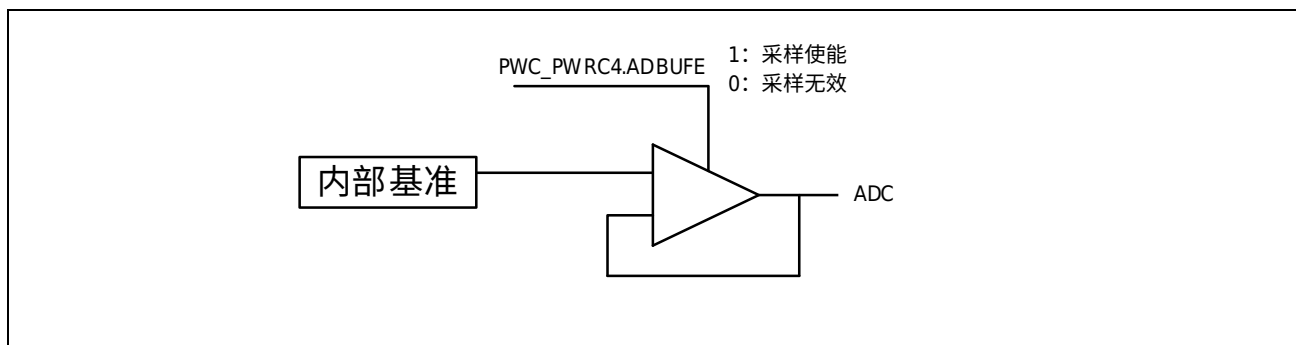


图 5-10 内部电压采样示意图

■ 基准电压测量通路

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. ADC_EXCHSELR.EXCHSEL=1，使 ADC 选通内部电压测量通道；
3. 等待 50 μ s 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 WKTM，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，WKTM 还可以选择 64Hz 的内部时钟信号作为时钟源。该计数器是累加计数器，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]设定值的一个周期后，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和掉电模式。

WKTM 寄存器值上电复位后不定，上电复位后，设置 PWC_WKTCR。在启动 WKTM 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。芯片可以配置的运行模式如表 5-3 所示。

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。

用户可以通过下列方法优化运行下的功耗。

- 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_FCGn (n=0~3) 关闭不需要使用的功能
- 4) 关闭 RAM 的电源

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 (High Speed Run Mode)	主频120MHz及以下

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)	VDD域掉电
	掉电模式2 (PDMD2)	VDD域掉电之外，电压检测单元无效
	掉电模式3 (PDMD3)	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，与掉电模式4相比，掉电唤醒时除 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

项目	睡眠模式	停止模式	掉电模式
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
PLLH	工作可设	停止	掉电
CPU	停止（保持）	停止（保持）	掉电
RAM (Ret SRAM以外)	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 掉电模式3/4下掉电
Timer0	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
ADC	工作可设	停止	掉电
DAC	工作可设	工作可设	掉电
CMP	工作可设	工作可设	掉电
PA0~PA12, PB0~PB2, PB5~PB15, PC0~PC13, PD2,	工作可设	保持	保持或者高阻

项目	睡眠模式	停止模式	掉电模式
PF2			
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，振荡器停止振荡，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，振荡器停止振荡，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
PF3/MD	工作可设	保持；	保持； 芯片外面通过电阻接到地
PA13~PA15, PB3, PB4	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 ICG0 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 ICG0 的 SWDTSLPOFF 位为 1，专用看门狗在睡眠模式下停止计数；如果 SWDTSLPOFF 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 ICG0 的 SWDTSLP0FF 位为 1，专用看门狗在停止模式下停止计数；如果 SWDTSLP0FF 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 也会产生功耗，除非在进入停止模式前将其禁止。要禁止 DAC，需要将 DACR.DAE、DACR.DA1E 和 DACR.DA2E 清零。要禁止 ADC，需要 ADC_STR.STRT 位清零，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。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。

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。

执行 WFI 指令进入停止模式之前，需要确保 HCLK 与 PCLK1/2/3/4、EXCLK 的分频比例控制在 1:1 或 2:1 或 4:1。若 HCLK 与任一外设时钟分频比例超过 4:1，需要配置 PCLK1=PCLK3 且 PCLK1 与 PCLK3 使用其最大时钟分频（CMU_SCFGR.PCLK1S=0b110，CMU_SCFGR.PCLK3S=0b110）。

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。

通过非可屏蔽中断解除停止模式时，需要设置 INT_NMIER 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。

■ 进入停止模式

1. 清除 PWC_LPMCSR.LPMF；
2. 在 PWC_STPMCR.STOP=1，PWC_PWRC0.PWDN=0 时执行 WFI 指令即可进入停止模式；
3. 读 PWC_LPMCSR.LPMF，如果该位为 1，芯片正常进入了 STOP 模式，如果该位为 0，返回步骤 2。

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。

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。

执行 WFI 指令进入停止模式的同时产生了不用于解除停止模式的中断事件，芯片会优先响应该中断而不进入 STOP 模式。如果需要优先进入 STOP 模式而不执行该中断，在执行 WFI 指令前需要关闭不用于解除停止模式的中断。

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 ICG0 的 SWDTSLPOFF 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 SWDTSLPOFF 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.PDMDS[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，请参考对应**数据手册电气特性**章节。

VDDR 域在掉电模式 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_PWRC5.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	PDMS[1:0]	功耗	唤醒时间	说明
掉电模式1	0b00	l _{PD1}	T _{PD1}	VCC域电源电压检测单元有效
掉电模式2	0b01	l _{PD2}	T _{PD2}	VCC域POR、PDR检测电路有效，BOR、PVD1、PVD2无效
掉电模式3	0b10	l _{PD3}	T _{PD3}	VCC域POR、PDR，BOR、PVD1、PVD2无效
掉电模式4	0b11	l _{PD4}	T _{PD4}	VCC域POR、PDR，BOR、PVD1、PVD2无效

功耗和唤醒时间的关系： $l_{pd1} > l_{pd2} > l_{pd3} = l_{pd4}$ ， $T_{pd1} < T_{pd2} < T_{pd4} < T_{pd3}$

■ 进入掉电模式

1. 清除 PWC_LPMCSR.LPMF；
2. 在 PWC_STPMCR.STOP=1, PWC_PWRC0.PWDN=1, PWC_PWCR4.PD1PW=1 时执行 WFI 指令即可进入掉电模式；
3. 读 PWC_LPMCSR.LPMF，如果该位为 0，返回步骤 2；如果正常产生掉电复位，步骤 3 不会被执行。

■ 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、上电复位和欠压复位。能够用于掉电模式唤醒的事件包括：

WKUPn_0/1/2/3 (n=0~3) 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、PVD1、PVD2 电路都处于无效状态，从掉电模式 3 唤醒后，除 PWC_PDWKFO/ 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、PVD1、PVD2 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_PDWKFO-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 掉电模式2 掉电模式4	PWC_PWRC0, PWC_PWRC1, PWC_PWRC3, PWC_PDWKE0, PWC_PDWKE1, PWC_PDWKE2 PWC_PDWKES, PWC_PDWKF0, PWC_PDWKF1, PWC_PVDCR0, PWC_PVDCR1, PWC_PVDFCR, PWC_PVDLCR, PWC_PVDICR, 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]=0b10 或者 0b11，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]=0b00，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]=0b01，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，WKTM 还可以选择 64Hz 的内部时钟信号作为时钟源。该计数器是累加计数器，在 WKTC0.WKTCE 置位后，计数器开始计数，当计数值等于 WKTCMP[11:0]设定值的一个周期后，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_n 的结构框图如图 5-11 所示。

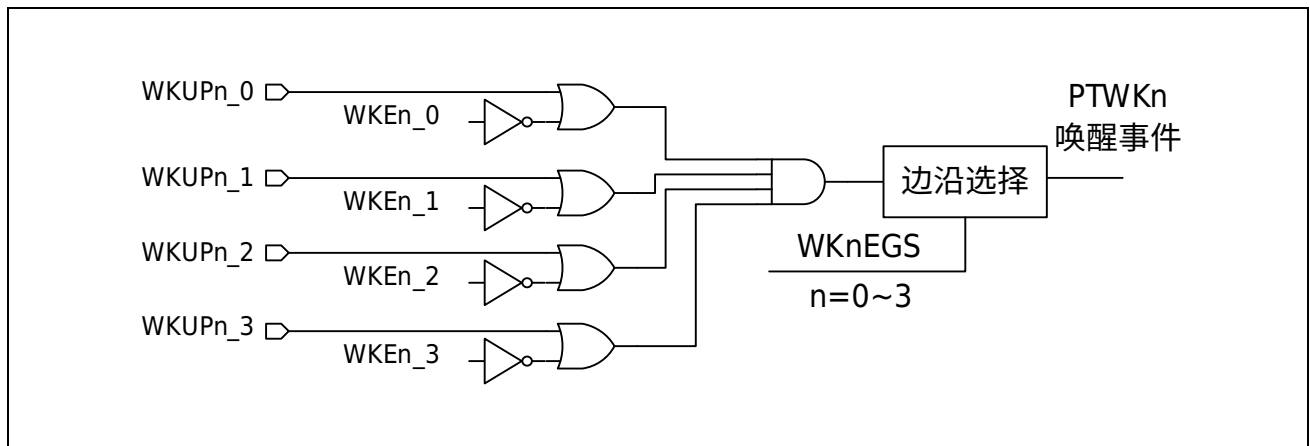


图 5-11 PTWKn 结构框图

5.5.4 降低系统时钟速度

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=0b11 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	系统RAM，地址范围： 0x2000 0000~0x2000 3FFF	PWC_RAMPC0.RAMPDC0
SRAMH	系统RAM，地址范围： 0x1FFF C000~0x1FFF FFFF	PWC_RAMPC0.RAMPDC10
CAN RAM	CAN用RAM	PWC_PRAMLPC.PRAMD0
CACHERAM	Cache用RAM	PWC_PRAMLPC.PRAMD2
Ret SRAM	掉电模式保持RAM，地址范围： 0x200F 0000~0x200F 0FFF	PWR_PWRC5.SRAMBSD

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_XTALCFGR, CMU_XTALSTBCR, CMU_XTALCR, CMU_XTALSTDCR, CMU_XTALSTDSR, CMU_HRCTRM, CMU_HRCCR, CMU_MRCTRM, CMU_MRCCR, CMU_PLLHCFGR, CMU_PLLHCR, CMU_OSCSTBSR, CMU_CKSWR, CMU_SCFGR, 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_PWRC2, PWC_PWRC3, PWC_PWRC4, PWC_PWRC5, PWC_PWRC6, PWC_PDWKE0, PWC_PDWKE1, PWC_PDWKE2, PWC_PDWKES, PWC_PDWKF0, PWC_PDWKF1, CMU_PERICKSEL, PWC_STPMCR, PWC_RAMPC0, PWC_PRAMLPC, RMU_RSTF0, PWC_WKTC, RMU_FRST0, RMU_FRST1, RMU_FRST2, RMU_FRST3, RMU_PRSTCR0
PWC_FPRC.FPRCB3	PWC_PVDCR0, PWC_PVDCR1, PWC_PVDFCR, PWC_PVDLDR, PWC_PVDICR, PWC_PVDDSR
PWC_FCG0PC.PRT0	PWC_FCG0

5.7 寄存器描述

表 5-9 PWC 寄存器列表

基地址 1: 0x4004 CC00

寄存器	描述	偏移地址	位宽	复位值
PWC_PWRC0	电源控制寄存器0	0x00	8	0x00
PWC_PWRC1	电源模式控制寄存器1	0x04	8	0xC0
PWC_PWRC4	电源模式控制寄存器4	0x10	8	0x00
PWC_PVDCR0	PVD控制寄存器0	0x14	8	0x00
PWC_PVDCR1	PVD控制寄存器1	0x18	8	0x00
PWC_PVDFCR	PVD滤波控制寄存器	0x1C	8	0x11
PWC_PVDLCR	PVD电平控制寄存器	0x20	8	0x00
PWC_PDWKE0	掉电唤醒使能寄存器0	0x28	8	0x00
PWC_PDWKE1	掉电唤醒使能寄存器1	0x2C	8	0x00
PWC_PDWKE2	掉电唤醒使能寄存器2	0x30	8	0x00
PWC_PDWKES	掉电唤醒事件边沿选择寄存器	0x34	8	0x00
PWC_PDWKF0	掉电唤醒标志寄存器0	0x38	8	0x00
PWC_PDWKF1	掉电唤醒标志寄存器1	0x3C	8	0x00
PWC_PWRC5	电源模式控制寄存器5	0x40	8	0x00
PWC_PWRC6	电源模式控制寄存器6	0x44	8	0x00
PWC_PVDICR	PVD中断控制寄存器	0xC0	8	0x00
PWC_PVDDSR	PVD检测状态寄存器	0xC4	8	0x11
PWC_RAMPC0	RAM功耗控制寄存器0	0xC8	32	0x0000 0000
PWC_PRAMLPC	外设RAM低功耗控制寄存器	0xD0	32	0x0000 0000
PWC_LPMCSR	低功耗状态寄存器	0xD8	16	0x0000
PWC_LPMCSRC	低功耗状态清除寄存器	0xDA	16	0x0000

基地址 2: 0x4005 4000

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

基地址 3: 0x4004 C400

寄存器	描述	偏移地址	位宽	复位值
PWC_WKTCR	唤醒计时器控制寄存器	0x50	16	0x0000

基地址 4: 0x4004 8000

寄存器	描述	偏移地址	位宽	复位值
PWC_FCG0	功能时钟控制0	0x00	32	0xFFFF FA0E
PWC_FCG1	功能时钟控制1	0x04	32	0xFFFF FFFF
PWC_FCG2	功能时钟控制2	0x08	32	0xFFFF FFFF
PWC_FCG3	功能时钟控制3	0x0C	32	0xFFFF FFFF
PWC_FCG0PC	PWC_FCG0保护控制	0x10	32	0x0000 0000

5.7.1 电源模式控制寄存器 0 (PWC_PWRC0)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
PWDN	Res	IORTN[1:0]		Res		PDMDS[1:0]	

位/位域	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	RW
b6	Res	保留位	读出时为“0”，写入时写“0”	RW
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0b00，解除IO保持状态 1x: 掉电模式中及掉电唤醒后IO为高阻	RW
b3~b2	Res	保留位	读出时为“0”，写入时写“0”	RW
b1~b0	PDMDS[1:0]	掉电模式选择控制	00: 掉电模式1 01: 掉电模式2 10: 掉电模式3 11: 掉电模式4	RW

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
Res					VHRCSD	VPL LSD[1:0]	

位/位域	标记	位名	功能	读写
b7~b3	Res	保留位	读出时为“0b11000”，写入时写“0b11000”	RW
b2	VHRCSD	HRC电源关闭	0: HRC电源使能 1: 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25μs后再启动HRC模块	RW
b1~b0	VPL LSD[1:0]	PLL电源关闭	00: PLL电源使能 11: PLL电源关闭 其他: 禁止设定 PLLH关闭并等待50μs后，设置VPL LSD=0b11后关闭PLL用电源，进一步降低功耗。设置VPL LSD=0b00后，需要等待25μs后再启动PLLH模块。	RW

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	Res	PD1PW	Res				

位/位域	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	RW
b6	Res	保留位	读出时为“0”，写入时写“0”	RW
b5	PD1PW	掉电模式1唤醒电源稳定控制	进入掉电模式1之前，PD1PW必须置位。	RW
b4~b0	Res	保留位	读出时为“0”，写入时写“0”	RW

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/位域	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	RW
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	RW
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	RW
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	RW
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	RW
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	RW
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	RW
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	RW

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/位域	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	RW
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	RW
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	RW
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	RW
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	RW
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	RW
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	RW
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	RW

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	Res	RTCALMWKE	RTCPRDWKE	Res	VD2WKE	VD1WKE	

位/位域	标记	位名	功能	读写
b7	WKTMWKE	WKT唤醒事件使能	0: WKT唤醒事件无效 1: WKT唤醒事件使能	RW
b6	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
b3~b2	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW

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
Res		VD2EGS	VD1EGS	WK3EGS	WK2EGS	WK1EGS	WK0EGS

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

5.7.8 掉电唤醒标志寄存器 0 (PWC_PDWKFO)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	VD2WKF	VD1WKF	PTWK3F	PTWK2F	PTWK1F	PTWK0F	

位/位域	标记	位名	功能	读写
b7~b6	Res	保留位	读出时为“0”，写入时写“0”	RW
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。	RW
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。	RW
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。	RW
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。	RW
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。	RW
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。	RW

5.7.9 掉电唤醒标志寄存器 1 (PWC_PDWKF1)

复位值: 0x00

位/位域	标记	位名	功能	读写
b7	WKTMWKF	WKT唤醒标志位	0: 未发生WKT唤醒事件 1: 发生WKT唤醒事件 掉电唤醒后, 需要写零清除本位。	RW
b6	Res	保留位	读出时为“0”, 写入时写“0”	RW
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。	RW
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。	RW
b3~b0	Res	保留位	读出时为“0”, 写入时写“0”	RW

5.7.10 电源模式控制寄存器 5 (PWC_PWRC5)

复位值: 0x00

位/位域	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, 在关闭XTAL32, VDDR域RLDO后, 可以置位CSDIS, 以降低功耗。	RW
b6~b2	Res	保留位	读出时为“0”, 写入时写“0”	RW
b1	SRAMBSD	Ret SRAM (保持RAM) 掉电控制	0: Ret SRAM不掉电 1: Ret SRAM掉电	RW
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RTC/ WTKM/ Ret SRAM的电源被关闭, 降低功耗。	RW

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
Res.						RTCKSEL[1:0]	

位/位域	标记	位名	功能	读写
b7~b2	Res	保留位	读出时为“0”，写入时写“0”	RW
b1~b0	RTCKSEL[1:0]	RTC时钟选择	00: LRC 01: XTAL小数分频 10: 端子输入 其他: 禁止设定。	RW

5.7.12 低功耗状态寄存器 (PWC_LPMCSR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															LPMF

位/位域	标记	位名	功能	读写
b15~b1	Res	保留位	读时读出“0”，写入时请写“0”	RW
b0	LPMF	低功耗模式标志位	0: 芯片未进入停止模式或者掉电模式 1: 芯片进入停止模式 注: 芯片进入掉电模式后，该位清理	R

5.7.13 低功耗状态清除寄存器 (PWC_LPMCSR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															LPMFC

位/位域	标记	位名	功能	读写
b15~b1	Res	保留位	读时读出“0”，写入时请写“0”	RW
b0	LPMFC	LPMF清除位	0: 不清除LPMF标志 1: 清除LPMF标志	RW

5.7.14 功能时钟控制 0 (PWC_FCG0)

复位值: 0xFFFF FA0E

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								CRC	Res				CTC	AOS	FCM
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	DMA	Res		PLA	SRAM B	Res				SRAM 0	Res		SRAM H		

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时所有位为“1”，写入时所有位写“1”	RW
b23	CRC	CRC功能控制	0: CRC功能使能 1: CRC功能无效	RW
b22~b19	Res	保留位	读出时为“1”，写入时写“1”	RW
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	RW
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	RW
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	RW
b15	Res	保留位	读出时为“1”，写入时写“1”	RW
b14	DMA	DMA功能控制	0: DMA功能使能 1: DMA功能无效	RW
b13~b12	Res	保留位	读出时为所有位“1”，写入时所有位写“1”	RW
b11	PLA	PLA功能控制	0: PLA功能使能 1: PLA功能无效	RW
b10	SRAMB	Ret SRAM功能控制	0: Ret SRAM功能使能 1: Ret SRAM功能无效	RW
b9-b5	Res	保留位	读出时为“10000”，写入时写“10000”	RW
b4	SRAM0	SRAM0功能控制	0: SRAM0功能使能 1: SRAM0功能无效	RW
b3~b1	Res	保留位	读出时为所有位“1”，写入时所有位写“1”	RW
b0	SRAMH	SRAMH功能控制	0: SRAMH功能使能 1: SRAMH功能无效	RW

5.7.15 功能时钟控制 1 (PWC_FCG1)

复位值: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							SPI
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res			I2C	Res		MCAN2	MCAN1

位/位域	标记	位名	功能	读写
b31~b17	Res	保留位	读出时所有位为“1”，写入时所有位写“1”	RW
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	RW
b15~b5	Res	保留位	读出时为所有位“1”，写入时所有位写“1”	RW
b4	I2C	I2C功能控制	0: 集成电路总线I2C功能使能 1: 集成电路总线I2C功能无效	RW
b3~b2	Res	保留位	读出时为“1”，写入时写“1”	RW
b1	MCAN2	MCAN2功能控制	0: MCAN单元2功能使能 1: MCAN单元2功能无效	RW
b0	MCAN1	MCAN1功能控制	0: MCAN单元1功能使能 1: MCAN单元1功能无效	RW

5.7.16 功能时钟控制 2 (PWC_FCG2)

复位值: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Res							TMRA_5	TMRA_4	TMRA_3	TMRA_2	TMRA_1	Res				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
EMB	Res	TMR0_2	TMR0_1	Res	TMR4	HRPWM_6	HRPWM_5	HRPWM_4	HRPWM_3	HRPWM_2	HRPWM_1	TMR6_4	TMR6_3	TMR6_2	TMR6_1	

位/位域	标记	位名	功能	读写
b31~b25	Res	保留位	读出时为所有位“1”，写入时所有位写“1”	RW
b24	TMRA_5	TimerA_5功能控制	0: TimerA单元5功能使能 1: TimerA单元5功能无效	RW
b23	TMRA_4	TimerA_4功能控制	0: TimerA单元4功能使能 1: TimerA单元4功能无效	RW
b22	TMRA_3	TimerA_3功能控制	0: TimerA单元3功能使能 1: TimerA单元3功能无效	RW
b21	TMRA_2	TimerA_2功能控制	0: TimerA单元2功能使能 1: TimerA单元2功能无效	RW
b20	TMRA_1	TimerA_1功能控制	0: TimerA单元1功能使能 1: TimerA单元1功能无效	RW
b19~b16	Res	保留位	读出时为所有位“1”，写入时写所有位“1”	RW
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	RW
b14	Res	保留位	读出时为“1”，写入时写“1”	RW
b13	TMR0_2	Timer0_2功能控制	0: Timer0单元2功能使能 1: Timer0单元2功能无效	RW
b12	TMR0_1	Timer0_1功能控制	0: Timer0单元1功能使能 1: Timer0单元1功能无效	RW
b11	Res	保留位	读出时为“1”，写入时写“1”	RW
b10	TMR4	TimerR4功能控制	0: Timer4功能单元1使能 1: Timer4功能单元1无效	RW
b9	HRPWM_6	HRPWM功能控制	0: HRPWM功能单元6使能 1: HRPWM功能单元6无效	RW
b8	HRPWM_5	HRPWM功能控制	0: HRPWM功能单元5使能 1: HRPWM功能单元5无效	RW
b7	HRPWM_4	HRPWM功能控制	0: HRPWM功能单元4使能 1: HRPWM功能单元4无效	RW
b6	HRPWM_3	HRPWM功能控制	0: HRPWM功能单元3使能 1: HRPWM功能单元3无效	RW
b5	HRPWM_2	HRPWM功能控制	0: HRPWM功能单元2使能 1: HRPWM功能单元2无效	RW
b4	HRPWM_1	HRPWM功能控制	0: HRPWM功能单元1使能 1: HRPWM功能单元1无效	RW
b3	TMR6_4	Timer6_4功能控制	0: Timer6单元4功能使能 1: Timer6单元4功能无效	RW

b2	TMR6_3	Timer6_3功能控制	0: Timer6单元3功能使能 1: Timer6单元3功能无效	RW
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	RW
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	RW

5.7.17 功能时钟控制 3 (PWC_FCG3)

复位值: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
USART4	USART3	USART2	USART1	Res			
b15	b14	b13	b12	b11	b10	b9	b8
Res						CMP3	CMP12
b7	b6	b5	b4	b3	b2	b1	b0
Res		DAC2	DAC1	Res	ADC3	ADC2	ADC1

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为所有位“1”，写入时所有位写“1”	RW
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	RW
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	RW
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	RW
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	RW
b19~b10	Res	保留位	读出时所有位为“1”，写入时所有位写“1”	RW
b9	CMP3	CMP3功能控制	0: 电压比较器CMP通道3功能使能 1: 电压比较器CMP通道3功能无效	RW
b8	CMP12	CMP1和CMP2功能控制	0: 电压比较器CMP通道1和通道2功能使能 1: 电压比较器CMP通道1和通道2功能无效	RW
b7~b6	Res	保留位	读出时所有位为“1”，写入时所有位写“1”	RW
b4	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	RW
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	RW
b3	Res	保留位	读出时所有位为“1”，写入时所有位写“1”	RW
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	RW
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	RW
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	RW

5.7.18 PWC_FCG0 保护控制 (PWC_FCG0PC)

复位值: 0x0000 0000

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
Res														PRT0	

位/位域	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	RW
b15~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
b0	PRT0	保护位	PWC_FCG0写使能控制位 0: PWC_FCG0写无效 1: PWC_FCG0写使能	RW

5.7.19 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	Res	FPRCB1	FPRCB0	

位/位域	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。	RW
b7	FPRCB7	FPRC位7	预留，读出时为“0”，写入时写“0”	RW
b6	FPRCB6	FPRC位6	预留，读出时为“0”，写入时写“0”	RW
b5	FPRCB5	FPRC位5	预留，读出时为“0”，写入时写“0”	RW
b4	FPRCB4	FPRC位4	预留，读出时为“0”，写入时写“0”	RW
b3	FPRCB3	FPRC位3	保护寄存器位，保护对象参考表 5-8 0: 写保护 1: 写使能	RW
b2	Res	保留位	预留，读出时为“0”，写入时写“0”	RW
b1	FPRCB1	FPRC位1	保护寄存器位，保护对象参考表 5-8 0: 写保护 1: 写使能	RW
b0	FPRCB0	FPRC位0	保护寄存器位，保护对象参考表 5-8 0: 写保护 1: 写使能	RW

5.7.20 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	Res													CKSMR C	FLNWT

位/位域	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	RW
b14~b2	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW

5.7.21 RAM 功耗控制寄存器 0 (PWC_RAMPC0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res					RAMDC10	Res	
b7	b6	b5	b4	b3	b2	b1	b0
Res							RAMPDC0

位/位域	标记	位名	功能	读写
b32~b11	Res	保留位	读出时为“0”，写入时写“0”	RW
b10	RAMPDC10	RAM掉电控制位10	0: SRAMH动作 1: SRAMH掉电	RW
b9~b1	Res	保留位	读出时为“0”，写入时写“0”	
b0	RAMPDC0	RAM掉电控制位0	0: SRAM0动作 1: SRAM0掉电	RW

5.7.22 外设 RAM 低功耗控制寄存器 (PWC_PRAMLPC)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res					PRAMPDC2	Res	PRAMPDC0

位/位域	标记	位名	功能	读写
b31~b3	Res	保留位	读出时为“0”，写入时写“0”	RW
b2	PRAMPDC2	外设RAM掉电控制位2	CACHE RAM掉电控制 0: 不掉电 1: 掉电	RW
b1	Res	保留位	读出时为“0”，写入时写“0”	RW
b0	PRAMPDC0	外设RAM掉电控制位0	MCAN RAM掉电控制 0: 不掉电 1: 掉电	RW

5.7.23 PVD 控制寄存器 0 (PWC_PVDCR0)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	PVD2EN	PVD1EN	Res				EXVCCINEN

位/位域	标记	位名	功能	读写
b7	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
b4~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

5.7.24 PVD 控制寄存器 1 (PWC_PVDCR1)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	PVD2CMPOE	PVD2IRS	PVD2IRE	Res	PVD1CMPOE	PVD1IRS	PVD1IRE

位/位域	标记	位名	功能	读写
b7	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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”	RW
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”	RW
b3	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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”	RW
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”	RW

5.7.25 PVD 滤波控制寄存器 (PWC_PVDFCR)

复位值: 0x11

b7	b6	b5	b4	b3	b2	b1	b0
Res	PVD2NFCKS[1:0]	PVD2NFDIS	Res	PVD1NFCKS[1:0]	PVD1NFDIS		

位/位域	标记	位名	功能	读写
b7	Res	保留位	读出时为“0”，写入时写“0”	RW
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 注: 只能在PVD2NFDIS位为“1”时改写该位	RW
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	RW
b3	Res	保留位	读出时为“0”，写入时写“0”	RW
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 注: 只能在PVD1NFDIS位为“1”时改写该位	RW
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	RW

5.7.26 PVD 电平控制寄存器 (PWC_PVDLCR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	PVD2LVL[2:0]			Res	PVD1LVL[2:0]		

位/位域	标记	位名	功能	读写
b7	Res	保留位	读出时为“0”，写入时写“0”	RW
b6~b4	PVD2LVL[2:0]	PVD2阈值电压选择	000: 2.1V	RW
			001: 2.3V	
			010: 2.5V	
			011: 2.6V	
			100: 2.7V	
b3	Res	保留位	101: 2.8V	RW
			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)	
			读出时为“0”，写入时写“0”	
b2~b0	PVD1LVL[2:0]	PVD1阈值电压选择	000: 2.0V	RW
			001: 2.1V	
			010: 2.3V	
			011: 2.5V	
			100: 2.6V	
b1	Res	保留位	101: 2.7V	RW
			110: 2.8V	
			111: 2.9V	

5.7.27 PVD 中断控制寄存器 (PWC_PVDICR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	PVD2EDGS[1:0]	PVD2NMIS	Res	PVD1EDGS[1:0]	PVD1NMIS		

位/位域	标记	位名	功能	读写
b7	Res	保留位	读出时为“0”，写入时写“0”	RW
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: 设定禁止	RW
b4	PVD2NMIS	PVD2中断类型	0: PVD2中断为非可屏蔽中断 1: PVD2中断为可屏蔽中断	RW
b3	Res	保留位	读出时为“0”，写入时写“0”	RW
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: 设定禁止	RW
b0	PVD1NMIS	PVD1中断类型	0: PVD1中断为非可屏蔽中断 1: PVD1中断为可屏蔽中断	RW

5.7.28 PVD 检测状态寄存器 (PWC_PVDDSR)

复位值: 0x11

b7	b6	b5	b4	b3	b2	b1	b0
Res	PVD2DETF LG	PVD2MON	Res	PVD1DETF LG	PVD1MON		
位/位域	标记	位名	功能				读写
b7~b6	Res	保留位	读出时为“0”，写入时写“0”				RW
b5	PVD2DETF LG	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				RW
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	Res	保留位	读出时为“0”，写入时写“0”				RW
b1	PVD1DETF LG	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				RW
b0	PVD1MON	PVD1监视位	0: VCC<VPVD1 1: PVD1无效时或者VCC>VPVD1				R

5.7.29 唤醒计时器控制寄存器 (PWC_WKTCR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
WKTCE	WKCKS[1:0]	WKOVF	WKTMCMP[11:0]												

位/位域	标记	位名	功能	读写
b15	WKTCE	WKTM使能	0: WKTM停止 1: WKTM计数 WKTM计数时清零WKTCE位会将计数器清零, 计数器停止动作。	RW
b14~b13	WKCKS[1:0]	WKTM时钟选择	00: 64Hz时钟 01: XTAL32 10: LRC 11: 预留	RW
b12	WKOVF	计时器比较结果标志	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。	RW
b11~b0	WKTMCMP[11:0]	WKTM比较位	WKTM计数器的比较值	RW

6 初始化配置 (ICG)

6.1 概述

芯片复位解除后，硬件电路会读取 Flash 地址 0x0000 0400~0x0000 045F 把数据加载到初始化配置寄存器。地址 0x0000 0408~0x0000 040B、0x0000 0410~0x0000 041F、0x0000 0438~0x0000 045F 为保留地址，请写入全 1 保证芯片正常动作。Flash 引导交换无效时，该区域位于 FLASH 扇区 0；Flash 引导交换有效，并且 Flash 扇区 0 的 OTP 没有锁存（0x0300 0A80~0x0300 0A83 数据全 1）时，该区域位于 Flash 扇区 1；否则该区域位于 Flash 扇区 0。用户可通过编程或擦除扇区 0 来修改初始化配置 (ICG) 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。

6.2 寄存器描述

表 6-1 ICG 基地址

名称	基地址	描述
ICG	0x0000 0400	ICG基地址

表 6-2 ICG 寄存器列表

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

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
Res			WDT SLP OFF	WDTWDPT[3:0]				WDTCK[3:0]			WDTPERI[1:0]		WDTI TS	WDT AUTS	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			SWD TSLP OFF	SWDTWDPT[3:0]				SWDTCK[3:0]			SWDTPERI[1: 0]		SWD TITS	SWD TAUT S	

位/位域	标记	位名	功能	读写
b31~b29	Res	保留位	功能预留位	R
b28	WDTSLPOFF	WDT在sleep模式下计数停止	0: WDT在睡眠模式下计数不停止 1: WDT在睡眠模式下计数停止	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	WDTCK[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	Res	保留位	功能预留位	R
b12	SWDTSLOFF	SWDT在Sleep, Stop模式下计数停止	0: SWDT在睡眠模式和停止模式下计数不停止 1: SWDT在睡眠模式和停止模式下计数停止	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%	
b11~b8	SWDWDPT[3:0]	刷新允许区域计数值百分比	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	R
b7~b4	SWDTCKS[3:0]	SWDT计数时钟	00: 256周期 01: 4096周期 10: 16384周期 11: 65536周期	R
b3~b2	SWDTPERI[1:0]	SWDT计数溢出周期	0: 中断请求 1: 复位请求	R
b1	SWDTITS	SWDT中断选择	0: 复位后, SWDT自动启动 (硬件启动) 1: 复位后, SWDT停止状态	R
b0	WDTAUTS	WDT自动启动		R

6.2.2 初始化配置寄存器 1 (ICG1)

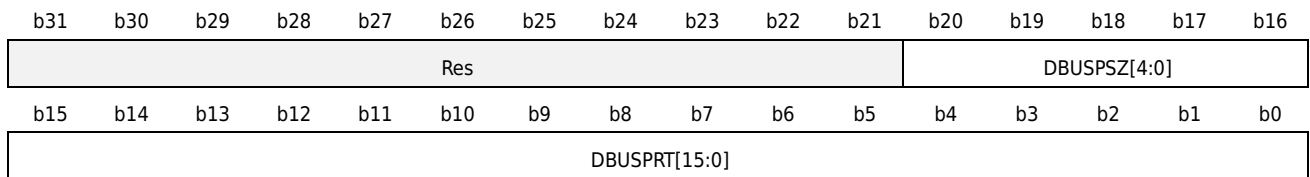
复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res													BOR DIS	BOR_LEV[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							HRC STOP	Res						HRCF REQS	

位/位域	标记	位名	功能	读写
b31~b19	Res	保留位	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	Res	保留位	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	Res	保留位	读出时为所有位“1”，写入时所有位写“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



位/位域	标记	位名	功能	读写
b31~b21	Res	保留位	预留，读出时为“1”，写入时写“1”	R
b20~b16	DBUSPSZ[4:0]	D-BUS读保护空间	设定D-BUS读保护空间。 当DBUSPRT[15:0]=0x4450时有效。 00000~01100：保留，禁止设定 01101：8KB 01110：16KB 01111：32KB 10000：64KB（64KB 产品禁止设定） 10001~11111：保留，禁止设定	R
b15~b0	DBUSPRT[15:0]	D-BUS读保护功能	当DBUSPRT[15:0]=0x4450时， 对0x0000 0000开始的空间进行D-BUS读保护使能；否则 D-BUS读保护无效。	R

7 嵌入式 Flash (EFM)

7.1 概述

Flash 接口通过 ICODE、DCODE 和 MCODE 总线对 Flash 进行访问, 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

- 最大 5KBytes 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 1KBytes; DCODE 总线缓存空间 128Bytes
- 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 128KBytes, 由一块 128KBytes 的 Flash 构成, 共 32 个扇区, 每个扇区为 4KBytes。扇区 0 为可配置为 OTP 区域。
- OTP (One Time Program) 区域共 5KBytes, 其中 4KBytes 配置在地址 0x0000 0000~0x0000 0FFF (扇区 0), 1KBytes 配置在地址 0x0300 0C00~0x0300 0FFF。地址 0x0300 0A80~0x0300 0AFF 为 OTP 数据锁存区。0x0300 0A00~0x0300 0A03 为 OTP 使能区。
- 64 位宽数据读取, 读缓存 64 位宽缓冲, 加速代码执行。
- 编程单位为 4Bytes, 擦除单位为 4KBytes。

0x0300 2040	数据安全配置区
0x0300 2000	引导交换区域

0x0300 0FFF 0x0300 0A00	OTP区域

0x0001 FFFF 0x0001 F000	扇区31

0X0000 0FFF 0x0000 0000	扇区0

图 7-1 Flash 地址分配

OTP 区域 0x0300 0A00~0x0300 0FFF 享有 1 个扇区，只能进行编程，禁止擦除/全擦除。

引导交换使能地址 0x0300 2000~0x0300 2003 和数据安全配置区 0x0300 2040~0x0300 204B 享有同 1 个扇区，可进行编程，擦除，但全擦除禁止。每次擦除以后需要重新对数据安全配置区 0x0300 2040~0x0300 204B 编程。

这 2 个扇区物理上从属于 FLASH，擦写标志位跟随 FLASH，但对 FLASH 全擦除这 2 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
100MHz<HCLK≤120MHz	FLWT[3:0]=0b0010
50MHz<HCLK≤100MHz	FLWT[3:0]=0b0001
HCLK≤50MHz	FLWT[3:0]=0b0000

7.5 Flash 读加速缓存

每次 Flash 读操作为 64 位读取，数据送给 CPU 同时也存入缓冲寄存器，这 64 位数据可以是 2 行 32 位指令，也可以是 4 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存储器中，从而提高了程序执行速度。

系统提供 1.125Kbytes 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间损耗。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（64 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 4KBytes。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.BUSHLCTL=0，则总线被占有，直至擦写结束；EFM_FWMC.BUSHLCTL=1，则总线被释放，需要程序在 RAM 上执行，实现程序在执行同时 Flash 在进行编程或擦除。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 0x0123 4567，再写 0xFEDC BA98）

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_FONWPRT 中。每个写保护寄存器存在 1 个写保护锁定位（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.PRTWERR=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.PRTWERR=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 0x0123 4567，再写 0xFEDC BA98）；
- 3) 设定单次编程模式（EFM_FWMC.PEMODE[2:0]=0b001）；

- 4) 解除写保护 (EFM_FONWPRT 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.RDY=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.OPTEND)。

对已锁存的 OTP 地址发送单编程不回读写操作, 编程不成功, 标志位 EFM_FSR.OTPWERR 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.MISMTCH 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 0x0123 4567, 再写 0xFEDC BA98);
- 3) 设定单次编程回读模式 (EFM_FWMC.PEMODE[2:0]=0b010);
- 4) 解除写保护 (EFM_FONWPRT 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.RDY=1);
- 7) 判断编程自读取结果标志位 (EFM_FSR.MISMTCH):
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.OPTEND)。

对已锁存的 OTP 地址发送单编程回读写操作, 编程不成功, 标志位 EFM_FSR.OTPWERR 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 16 μ s。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 0x0123 4567, 再写 0xFEDC BA98);
- 3) 设定连续编程模式 (EFM_FWMC.PEMOD[2:0]=0b011);

- 4) 解除写保护 (EFM_FONWPRT 对应位设定为 1);
- 5) 对编程地址写 32 位数据。(执行程序在 Flash 以外区域运行);
- 6) 等待操作结束标志位 (EFM_FSR.OPTEND) 置位;
- 7) 清除操作结束标志位, 直至读到 EFM_FSR.OPTEND 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]=0b000);
- 10) 等待 Flash 处于空闲状态 (EFM_FSR.RDY=1);
- 11) 读取编程地址并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。

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.OTPWERR 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 (4KBytes 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 0x0123 4567, 再写 0xFEDC BA98);
- 3) 设定擦除模式 (EFM_FWMC.PEMOD[2:0]=0b100);
- 4) 解除写保护 (EFM_FONWPRT 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.RDY=1);
- 7) 清除擦除结束标志位 (EFM_FSR.OPTEND)。

对已锁存的 OTP 地址发送擦除操作, 擦除不成功, OTP 区域数据保留, 标志位 EFM_FSR.OTPWERR 置位。

7.6.7 全擦除功能

EFM 提供了单块 Flash 全擦除方式。全擦除操作设定步骤如下:

- 1) 解除 Flash 的寄存器写保护 (EFM_FAPRT 先写 0x0123, 再写 0x3210);
- 2) 解除 EFM_KEY1 锁定 (EFM_KEY1 先写 0x0123 4567, 再写 0xFEDC BA98);
- 3) 单块 Flash 全擦除时, 设定 EFM_FWMC.PEMOD[2:0]=0b101;

- 4) 解除写保护 (EFM_FONWPRT 所有位都设定为 1);
- 5) 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;
- 6) 等待 Flash 处于空闲状态。(EFM_FSR.RDY=1);
- 7) 清除擦除结束标志位。(EFM_FSR.OPTEND)。

OTP 使能后, 对 Flash 地址发送单块全擦除写操作, 全擦除不成功, Flash 数据保留, 标志位 EFM_FSR.OTPWERR 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 0xAF18 0402, 同时地址 0x0300 2040~0x0300 204B 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 和扇区 1 进行编程和扇区擦除。
- 地址 0x0300 2040~0x0300 204B 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 0xA851 73AE, 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 和扇区 1 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 0x4254 5048 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 和扇区 1 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

D-BUS 保护功能对 ICG3 寄存器 DBUSPSZ[4:0] 设定的 D-BUS 读保护空间进行读保护。使能 D-BUS 保护，需设定 ICG3 寄存器 DBUSPRT[15:0] 为 0x4450，DBUSPSZ[4:0] 设定空间，复位重启后，功能使能。CPU-PC 指针在此 D-BUS 读保护空间时（即 CPU 在此区域执行程序），CPU 可以正常访问；当 CPU-PC 指针在此 D-BUS 读保护空间以外时（即 CPU 在 D-BUS 读保护以外空间执行程序），CPU 对 D-BUS 读保护空间内进行 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。

7.6.10 总线保持功能

通过设定寄存器 EFM_FWMC.BUSHLDCTL 位，可设定 FLASH 编程，擦除期间，总线处于保持状态还是释放状态。Flash 编程，擦除指令在 Flash 上执行时，必须把该控制位设定为 0；擦除指令在 Flash 以外空间（比如高速 RAM）执行时，可根据需要自由设定该控制位。

当设定 BUSHLDCTL 为 1（Flash 编程，擦写期间，总线释放状态）时，在编程（连续编程除外），擦写结束前（EFM_FSR.RDY=1）对 Flash 的读写访问将会被忽视，标志位 EFM_FSR.COLERR 位置位。

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（OTPWERR=1）。
- OTP 功能启用后，对 0x0300 0A00~0x0300 0FFF 空间进行擦除，全擦除操作（OTPWERR=1）。
- 对写保护扇区发出编程，擦除，全擦除操作（PRTWERR=1）。
- 编程地址非以 4 对齐或者数据大小非 32 位（PGSZERR=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（MISMTCH=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 (RDY=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 (OPTEND=1)。
- 擦除模式: 扇区擦除, 全擦除结束 (OPTEND=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 5KBytes 的 OTP 领域, 分为 1 个 4KBytes、16 个 64Bytes。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 0FFF	4KB	0x0300 0A80~0x0300 0A83
1	0x0300 0C00~0x0300 0C3F	64B	0x0300 0AC0~0x0300 0AC3
2	0x0300 0C40~0x0300 0C7F	64B	0x0300 0AC4~0x0300 0AC7
3	0x0300 0C80~0x0300 0CBF	64B	0x0300 0AC8~0x0300 0ACB
4	0x0300 0CC0~0x0300 0CFF	64B	0x0300 0ACC~0x0300 0ACF
5	0x0300 0D00~0x0300 0D3F	64B	0x0300 0AD0~0x0300 0AD3
6	0x0300 0D40~0x0300 0D7F	64B	0x0300 0AD4~0x0300 0AD7
7	0x0300 0D80~0x0300 0DBF	64B	0x0300 0AD8~0x0300 0ADB
8	0x0300 0DC0~0x0300 0DFE	64B	0x0300 0ADC~0x0300 0ADF
9	0x0300 0E00~0x0300 0E3F	64B	0x0300 0AE0~0x0300 0AE3
10	0x0300 0E40~0x0300 0E7F	64B	0x0300 0AE4~0x0300 0AE7
11	0x0300 0E80~0x0300 0EBF	64B	0x0300 0AE8~0x0300 0AEB
12	0x0300 0EC0~0x0300 0EFF	64B	0x0300 0AEC~0x0300 0AEF
13	0x0300 0F00~0x0300 0F3F	64B	0x0300 0AF0~0x0300 0AF3
14	0x0300 0F40~0x0300 0F7F	64B	0x0300 0AF4~0x0300 0AF7
15	0x0300 0F80~0x0300 0FBF	64B	0x0300 0AF8~0x0300 0AFB
16	0x0300 0FC0~0x0300 0FFF	64B	0x0300 0AFC~0x0300 0AFF

OTP 每块数据块对应 1 个 4Bytes 的锁存地址。锁存地址用于锁存对应的数据块。锁存地址数据为全 1 时，对应的 OTP 区域数据块可以编程；当锁存地址数据包含 0 时，对应的 OTP 区域数据不可编程。OTP 数据块 1~16 和锁存地址均无法擦除。

启用 OTP 功能，需要预先对地址 0x0300 0A00~0x0300 0A03 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 与 Flash 地址 0x0000 0000~0x0000 0FFF（扇区 0）共享 4KBytes 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，不可编程和擦除，Flash 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 0x1032 5476, 再写 0xEFCD AB89)。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 0x0123 4567, 再写 0xFEDC BA98)；
- 3) 设定编程模式 (EFM_FWMC.PEMODE[2:0]=0b001, 0b010, 0b011)；
- 4) 根据需要对 OTP 地址写入数据；
- 5) 解除 EFM_KEY2 锁定 (EFM_KEY2 先写 0x1032 5476, 再写 0xEFCD AB89)；
- 6) 对 OTP 使能地址 0x0300 0A00~0x0300 0A03 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 0A00~0x0300 0A03 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 OTPWERR 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

用户如要升级引导程序，需要对扇区 0 (0x0000 0000~0x0000 0FFF) 进行擦写，如擦写时遇到不可期的意外（如复位、掉电）时，有可能会整个芯片不能正常启动。在对扇区 0 进行擦除前，预先将新的引导程序写入扇区 1 (0x0000 1000~0x0000 1FFF)，然后对 EFM 地址 0x0300 2000 进行编程数

据 0x5A5A5A。MCU 复位后 CPU 就从扇区 1 启动新的引导程序，此时再对扇区 0 进行擦除，重新编程新的用户程序。

启动交换操作请参考图 7-2。

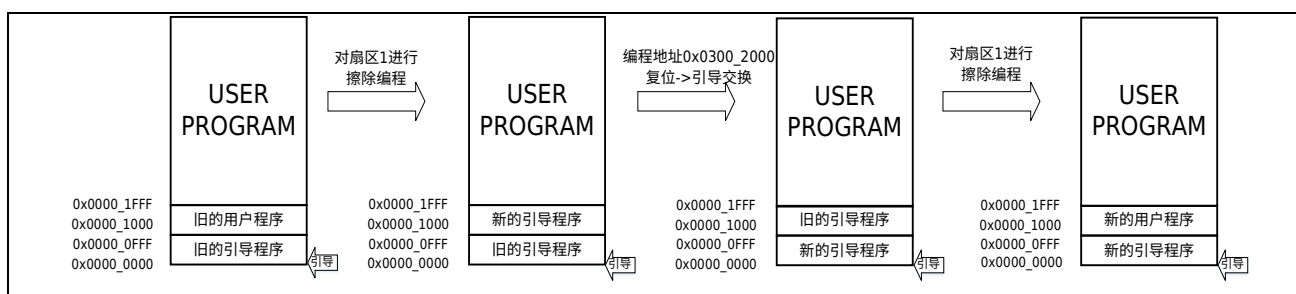


图 7-2 启动扇区交换功能 1

当用户需要再次升级启动程序，由于保存启动扇区交换信息的地址 0x0300 2000 已经被编程过了（用户可以通过读 FLASH 地址或者 EFM_FSWP 寄存器判断是否使用过启动交换功能），需要对 0x0300 2000 进行扇区擦除（注），再次进行启动程序的升级。

在对扇区 0 进行擦除前，预先将新的引导程序写入扇区 1，然后对 0x0300 2000 进行擦除（注）。MCU 复位后 CPU 就从扇区 1 启动新的引导程序，此时再对扇区 0 进行擦除，重新编程新的用户程序。

注：

1. 需要在擦除后对数据安全配置区 0x0300 2040~0x0300 204B 重新编程
2. 引导交换使能地址 0x0300 2000~0x0300 2003 和数据安全配置区 0x0300 2040~0x0300 204B 享有同 1 个扇区，可进行编程，擦除，但全擦除禁止。每次擦除以后需要重新对数据安全配置区 0x0300 2040~0x0300 204B 编程。

操作流程如图 7-3。

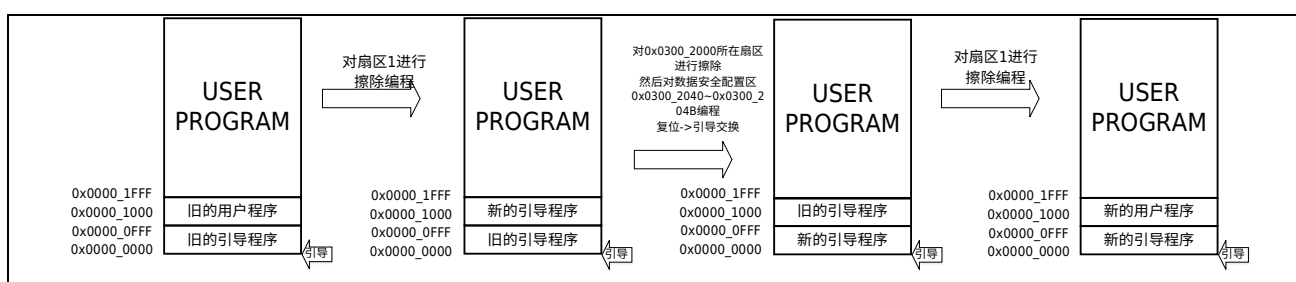


图 7-3 启动交换功能 2

若扇区 0 的 OTP 没有锁存（0x0300 0A80~0x0300 0A83 数据全 1），引导交换有效后，Flash 的扇区 0 和扇区 1 全体地址互换；且扇区 0 的 OTP 锁存地址（0x0300 0A80~0x0300 0A83）无法进行编程。

若扇区 0 的 OTP 已锁存（0x0300 0A80~0x0300 0A83 数据非全 1），当试图设定引导交换有效时，引导交换也会无效，Flash 的扇区 0 和扇区 1 地址不互换。

7.9 寄存器描述

表 7-4 EFM 基地址

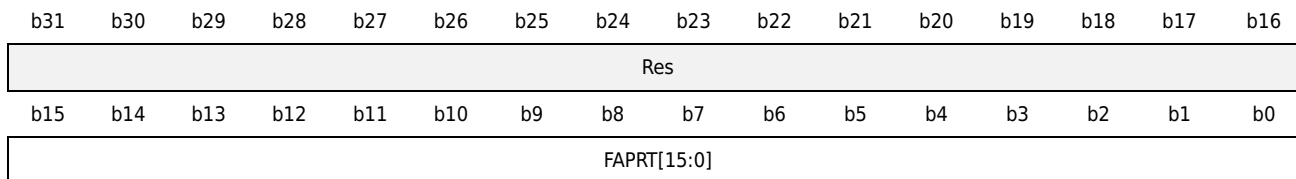
名称	基地址	描述
EFM	0x4001 0400	EFM基地址

表 7-5 EFM 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
EFM_FAPRT	Flash访问写保护寄存器	0x0000	32	0x0000 0000
EFM_KEY1	Flash密钥1寄存器	0x0004	32	0x0000 0000
EFM_KEY2	Flash密钥2寄存器	0x0008	32	0x0000 0000
EFM_FSTP	Flash停止寄存器	0x0014	32	0x0000 0000
EFM_FRMC	Flash读模式寄存器	0x0018	32	0x0000 0000
EFM_FWMC	Flash擦写模式寄存器	0x001C	32	0x0003 0000
EFM_FSR	Flash状态寄存器	0x0020	32	0x0000 0100
EFM_FSCLR	Flash状态清除寄存器	0x0024	32	0x0000 0000
EFM_FITE	Flash中断许可寄存器	0x0028	32	0x0000 0000
EFM_FSWP	Flash引导交换状态寄存器	0x002C	32	不定
EFM_CHIPID	芯片专属标志寄存器	0x0040	32	0x5848 0334
FEM_UQID0	UNIQUE ID寄存器0	0x0050	32	不定
EFM_UQID1	UNIQUE ID寄存器1	0x0054	32	不定
EFM_UQID2	UNIQUE ID寄存器2	0x0058	32	不定
EFM_WLOCK	Flash写保护锁存寄存器	0x0180	32	0x0000 0000
EFM_FONWPRT	Flash写保护寄存器	0x0190	32	0x0000 0000

7.9.1 访问写保护寄存器 (EFM_FAPRT)

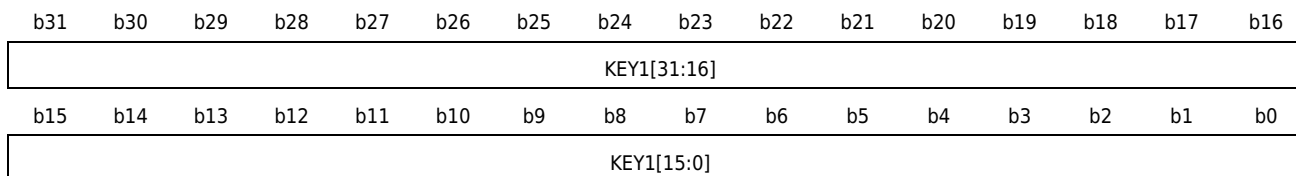
复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”，写入时写“0”	RW
b15~b0	FAPRT[15:0]	EFM寄存器写保护	EFM寄存器访问写保护寄存器。 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。 EFM寄存器访问保护有效时，该寄存器读出值为0x0000 0000。 EFM寄存器访问保护无效时，该寄存器读出值为0x0000 0001。	RW

7.9.2 Flash 密钥 1 寄存器 (EFM_KEY1)

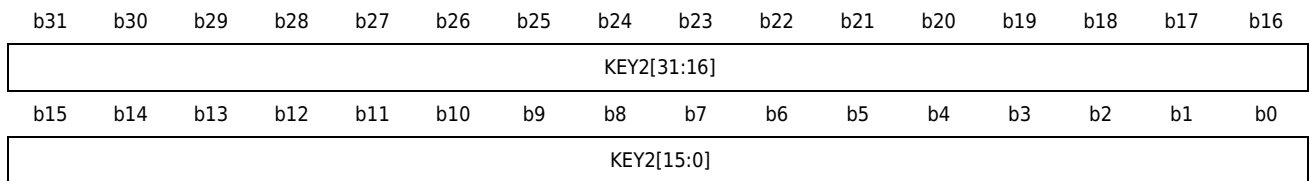
复位值: 0x0000 0000



位/位域	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0x0123 4567，再写0xFEDC BA98，解除对EFM_FWMC的写保护。 该寄存器读出值为0x0000 0000。	RW

7.9.3 Flash 密钥 2 寄存器 (EFM_KEY2)

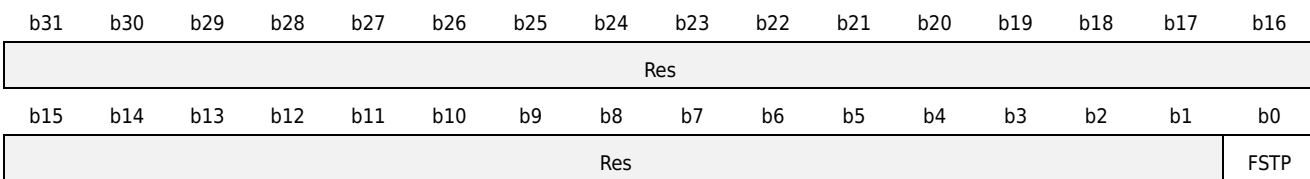
复位值: 0x0000 0000



位/位域	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EFM_KEY2先写0x1032 5476，再写0xEFCD AB89， 解除对OTP锁存地址的写保护。 该寄存器读出值为0x0000 0000。	RW

7.9.4 Flash 停止寄存器 (EFM_FSTP)

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
b0	FSTP	Flash停止模式控制	0: Flash活动状态 1: Flash处于停止模式 当寄存器位由1设为0后，请在确认FSR.RDY位为1后，进行Flash访问。	RW

7.9.5 读模式寄存器 (EFM_FRMC)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b20	Res	保留位	读出时为“0”，写入时写“0”	RW
b19	CRST	缓存复位位	0: 缓存数据解除复位 1: 缓存数据复位 (缓存、缓冲、预取值)	RW
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	RW
b17	DCACHE	DCODE缓存许可位	0: 关闭DCODE缓存功能 1: DCODE缓存功能使能	RW
b16	ICACHE	ICODE缓存许可位	0: 关闭ICODE缓存功能 1: ICODE缓存功能使能	RW
b15~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
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”	RW
b7~b4	Res	保留位	读出时为“0”，写入时写“0”	RW
b3~b0	FLWT[3:0]	Flash读插入的等待周期	0000: 不插入读等待周期 0001: 插入1个读等待周期 0010: 插入2个读等待周期 1110: 插入14个读等待周期 1111: 插入15个读等待周期	RW

7.9.6 Flash 擦写模式寄存器 (EFM_FWMC)

复位值: 0x0003 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res													KEY2 LOCK	KEY 1LO CK	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							BUSH LDCTL	Res					PEMOD[2:0]		

位/位域	标记	位名	功能	读写
b31~b18	Res	保留位	读出时为“0”，写入时写“0”	RW
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。	RW
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。	RW
b15~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
b8	BUSHLCTL	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。	RW
b7~b3	Res	保留位	读出时为“0”，写入时写“0”	RW
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: 保留值，禁止设定 111: 只读模式	RW

7.9.7 Flash 状态寄存器 (EFM_FSR)

复位值: 0x0000 0100

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							RDY	Res		COLERR	OPTEND	MISMATCH	PGSZERR	PRTWERR	OTPWERR

位/位域	标记	位名	功能	读写
b31~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
b8	RDY	Flash忙/空闲状态	0: Flash忙状态 1: Flash空闲状态	R
b7~b6	Res	保留位	读出时为“0”，写入时写“0”	RW
b5	COLERR	Flash读写冲突错误标志位	置位条件: RDY未置位情况下, 对Flash读写操作。 连续编程模式下, 对Flash写不置位。 清零条件: EFM_FSCLR对应清零位写1。	R
b4	OPTEND	Flash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
b3	MISMATCH	Flash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, Flash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
b2	PGSZERR	Flash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
b1	PRTWERR	Flash写保护地址错误标志位	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
b0	OTPWERR	Flash的OTP擦写错误标志位	置位条件: 对OTP锁存区域进行编程, 擦除, 全擦除操作。 清零条件: EFM_FSCLR对应清零位写1	R

7.9.8 Flash 状态清除寄存器 (EFM_FSCLR)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b6	Res	保留位	读出时为“0”，写入时写“0”	RW
b5	COLERRCLR	清除读写冲突错误标志位	0: 不发生清除动作 1: 清除FSR.COLERR 该位读时, 始终为0。	RW
b4	OPTENDCLR	清除操作结束标志	0: 不发生清除动作 1: 清除FSR.OPTEND位 该位读时, 始终为0。	RW
b3	MISMATCHCLR	清除编程回读不一致标志位	0: 不发生清除动作 1: 清除FSR.MISMATCH位 该位读时, 始终为0。	RW
b2	PGSZERRCLR	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.PGSZERR位 该位读时, 始终为0。	RW
b1	PEPRTERRCLR	清除写保护错误标志位	0: 不发生清除动作 1: 清除FSR.PRTWERR位 该位读时, 始终为0。	RW
b0	OTPWERRCLR	清除OTP擦写错误标志位	0: 不发生清除动作 1: 清除FSR.OTPWERR位 该位读时, 始终为0。	RW

7.9.9 Flash 中断许可寄存器 (EFM_FITE)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b3	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW

7.9.10 Flash 引导交换状态寄存器 (EFM_FSWP)

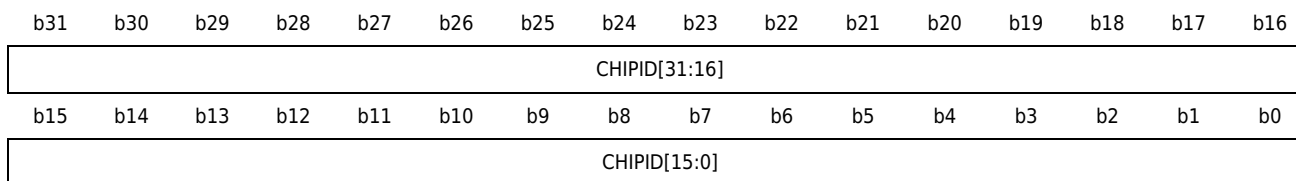
复位值: 不定

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

位/位域	标记	位名	功能	读写
b31-b1	Res	保留位	读出时为“0”，写入时写“0”	RW
b0	FSWP	扇区0和扇区1地址交换位	0: 复位后, CPU从扇区0启动。 1: 复位后, CPU从扇区1启动。 寄存器初始值由Flash地址 0x0300 2000~0x0300 2003 的值决定, 其数据为0x005A 5A5A时, 并且扇区0的OTP没有锁存 (0x0300 0A80~0x0300 0A83数据全1), 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)

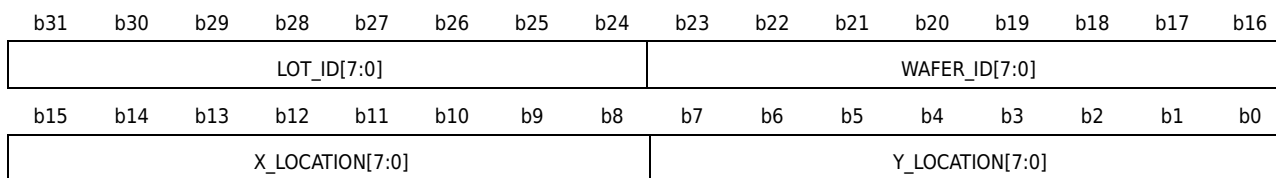
复位值: 0x5848 0334



位/位域	标记	位名	功能	读写
b31~b0	CHIPID[31:0]	芯片专属标志	CHIPID[31:16]: 0x5848 CHIPID[15:0]: 0x0334 该寄存器只读。	R

7.9.12 UNIQUE ID 寄存器 0 (EFM_UQID0)

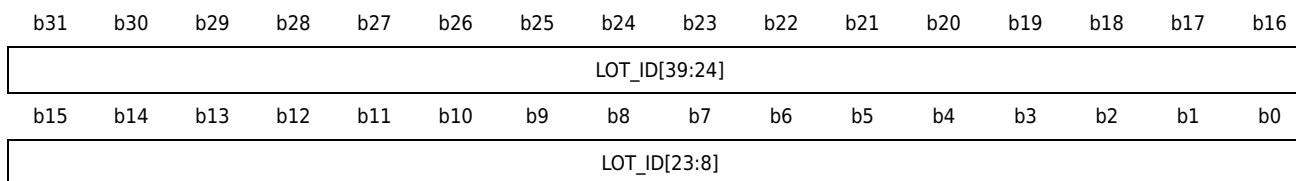
复位值: 不定



位/位域	标记	位名	功能	读写
b31~b24	LOT_ID[7:0]	LOT ID	芯片LOT ID[7:0]	R
b23~b16	WAFER_ID[7:0]	Wafer ID	Wafer ID	R
b15~b8	X_LOCATION[7:0]	X坐标	芯片在Wafer上的X坐标	R
b7~b0	Y_LOCATION[7:0]	Y坐标	芯片在Wafer上的Y坐标	R

7.9.13 UNIQUE ID 寄存器 1 (EFM_UQID1)

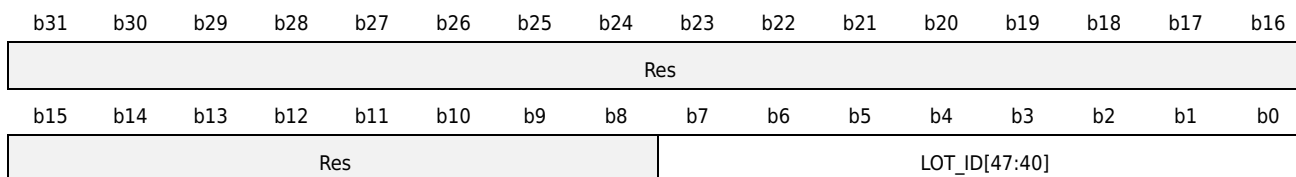
复位值: 不定



位/位域	标记	位名	功能	读写
b31~b0	LOT_ID[39:8]	LOT ID	芯片LOT ID[39:8]	R

7.9.14 UNIQUE ID 寄存器 2 (EFM_UQID2)

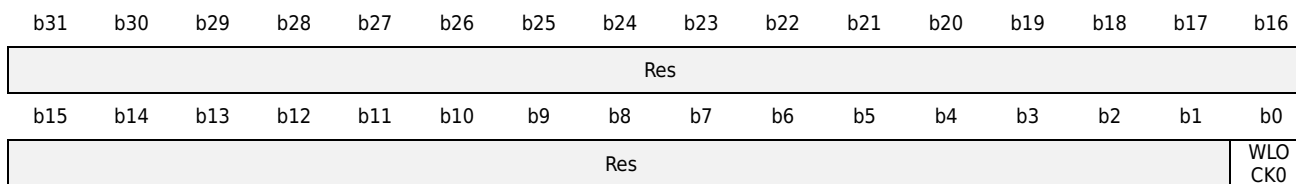
复位值：不定



位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	固定全为0	R
b7~b0	LOT_ID[47:40]	LOT ID	芯片LOT ID[47:40]	R

7.9.15 Flash 写保护锁存寄存器 (EFM_WLOCK)

复位值：0x0000 0000



位/位域	标记	位名	功能	读写
b31~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
b0	WLOCK0	FONWPRT寄存器锁定位	0: FONWPRT寄存器控制的扇区写禁止未Lockup 1: FONWPRT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。	RW

7.9.16 Flash 写保护寄存器 (EFM_FONWPRT)

复位值: 0x0000 0000

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	FON WPRT 28	FON WPRT 27	FON WPRT 26	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	FON WPRT 10	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/位域	标记	位名	功能	读写
b31	FONWPRT31	Flash扇区31的写许可位	0: Flash扇区31的写禁止 1: Flash扇区31的写许可 WLOCK[0]=0时, 该位写0, 写1许可。 WLOCK[0]=1时, 该位只能0, 写1无效。	RW
b30	FONWPRT30	Flash扇区30的写保护位	0: Flash扇区30的写禁止 1: Flash扇区30的写许可 WLOCK[0]=0时, 该位写0, 写1许可。 WLOCK[0]=1时, 该位只能0, 写1无效。	RW
b29~b2	RW
b1	FONWPRT1	Flash扇区1的写保护位	0: Flash扇区1的写禁止 1: Flash扇区1的写许可 WLOCK[0]=0时, 该位写0, 写1许可。 WLOCK[0]=1时, 该位只能0, 写1无效。	RW
b0	FONWPRT0	Flash扇区0的写保护位	0: Flash扇区0的写禁止 1: Flash扇区0的写许可 WLOCK[0]=0时, 该位写0, 写1许可。 WLOCK[0]=1时, 该位只能0, 写1无效。	RW

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）。
- 6) 编程、擦除时设定总线释放（FWMC.BUSHLDCTL=1）时，编程、擦除期间如需响应中断，请把向量和子程序设定到 RAM。

8 内置 SRAM (SRAM)

8.1 概述

本产品带有 32KB 系统 SRAM (SRAMH/ SRAM0) 和 4KB 掉电模式保持 SRAM (Ret SRAM)。

各 SRAM 可按照字节、半字 (16 位) 或全字 (32 位) 访问。所有 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 1/2 下提供 4KB 的数据保持空间。

SRAMH, 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。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	16KB	0x2000 0000~0x2000 3FFF	ECC
SRAMH	16KB	0x1FFF C000~0x1FFF FFFF	ECC
Ret SRAM	4KB	0x200F 0000~0x200F 0FFF	ECC

注:

- 在允许产生 RAM ECC 校验错误产生 NMI 中断或复位的情况下 (SRAMC_CKCR.BECCMOD 设置不为 0 时), 当存取数据时, 必须先设置 SRAMC_CKCR.BECCMOD, 然后对所用 RAM (SRAMH、SRAM0、Ret SRAM) 空间以字为单位进行初始化。

所有 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。

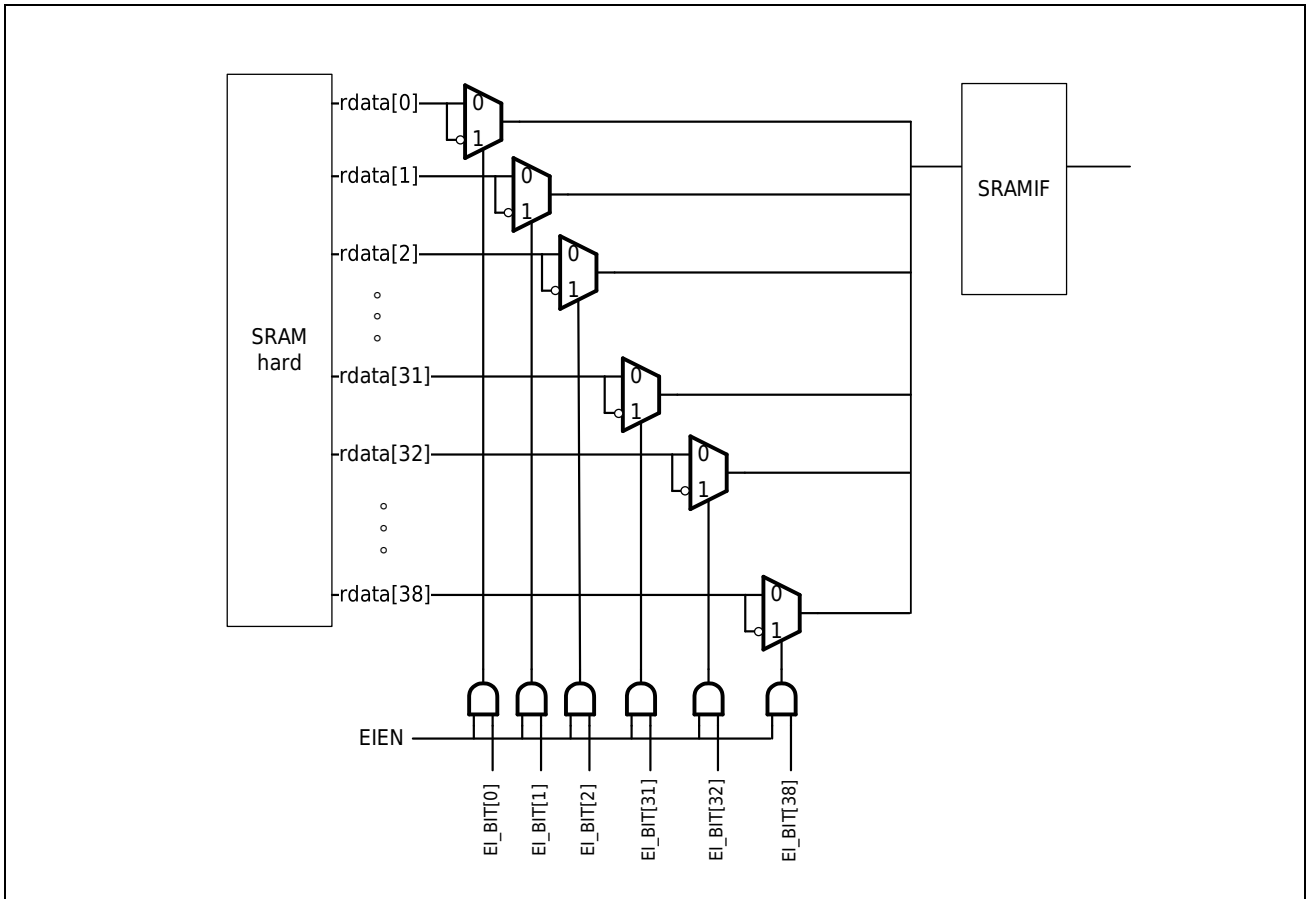


图 8-1 错误注入功能示意图

注:

0x2000 0000~0x2000 0002 不支持错误注入和 ECC 校验。

8.2 寄存器描述

表 8-2 SRAM 基地址

名称	基地址	描述
SRAMC	0x4005 0800	SRAM控制寄存器基地址

表 8-3 SRAM 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
SRAMC_CKCR	SRAM校验控制寄存器	0x08	32	0x0000 0000
SRAMC_CKPR	SRAM校验控制保护寄存器	0x0C	32	0x0000 0000
SRAMC_CKSR	SRAM校验状态寄存器	0x10	32	0x0000 0000
SRAM0_EIEN	SRAM0错误注入使能寄存器	0x14	32	0x0000 0000
SRAM0_EIBIT0	SRAM0错误注入位使能寄存器0	0x18	32	0x0000 0000
SRAM0_EIBIT1	SRAM0错误注入位使能寄存器1	0x1C	32	0x0000 0000
SRAM0_ECCERRADDR	SRAM0校验错误地址寄存器	0x20	32	0x0000 0000
SRAMH_EIEN	SRAMH错误注入使能寄存器	0x54	32	0x0000 0000
SRAMH_EIBIT0	SRAMH错误注入位使能寄存器0	0x58	32	0x0000 0000
SRAMH_EIBIT1	SRAMH错误注入位使能寄存器1	0x5C	32	0x0000 0000
SRAMH_ECCERRADDR	SRAMH校验错误地址寄存器	0x60	32	0x0000 0000
SRAMB_EIEN	Ret SRAM错误注入使能寄存器	0x64	32	0x0000 0000
SRAMB_EIBIT0	Ret SRAM错误注入位使能寄存器0	0x68	32	0x0000 0000
SRAMB_EIBIT1	Ret SRAM错误注入位使能寄存器1	0x6C	32	0x0000 0000
SRAMB_ECCERRADDR	Ret SRAM校验错误地址寄存器	0x70	32	0x0000 0000

8.2.1 SRAM 校验控制寄存器 (SRAMC_CKCR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res		HECCMOD [1:0]		BECCMOD [1:0]		ECCMOD [1:0]		Res				HECC OAD	BECC OAD	ECC OAD	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														PYOA D	

位/位域	标记	位名	功能	读写
b31~b30	Res	保留位	读出时为“0”，写入时写“0”	RW
b29~b28	HECCMOD[1:0]	SRAMH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。	RW
b27~b26	BECCMOD[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。	RW
b25~b24	ECCMOD[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,	RW

			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。	
b23~b19	Res	保留位	读出时为“0”，写入时写“0”	RW
b18	HECCOAD	SRAMH ECC校验出错后操作	0: Non-maskable interrupt 1: Reset	RW
b17	BECCOAD	Ret SRAM ECC校验出错后操作	0: Non-maskable interrupt 1: Reset	RW
b16	ECCOAD	SRAM0 ECC校验出错后操作	0: Non-maskable interrupt 1: Reset	RW
b15~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
b0	PYOAD	Cache RAM奇偶校验出错后操作	0: Non-maskable interrupt 1: Reset	RW

注：

在允许产生 RAM ECC 校验错误产生 NMI 中断或复位的情况下（SRAMC_CKCR.BECCMOD 设置不为 0 时），当存取数据时，必须先设置 SRAMC_CKCR.BECCMOD，然后对所用 RAM（SRAMH，SRAM0，Ret SRAM）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 (SRAMC_CKPR)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	读出时为“0”，写入时写“0”	RW
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“0x3B”使能当前寄存器的使能	RW
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	RW

注:

CKPRC 控制寄存器 SRAM_CKCR 的写操作。当 CKPRC 为 1 时，允许对 SRAM_CKCR 进行写操作；当 CKPRC 为 0 时，不能对 SRAM_CKCR 进行写操作。当写入此位时，必须同时对 CTPRKW[6:0] 写入 0x3B。

8.2.3 SRAM 校验状态寄存器 (SRAMC_CKSR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							CACHE_PYERR	SRAMB_2ERR	SRAMB_1ERR	SRAM0_2ERR	SRAM0_1ERR	SRAMH_2ERR	SRAMH_1ERR	Res	

位/位域	标记	位名	功能	读写
b31~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
b8	CACHE_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	RW
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	RW
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	RW
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	RW
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	RW
b3	SRAMH_2ERR	SRAMH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	RW
b2	SRAMH_1ERR	SRAMH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	RW
b1~b0	Res	保留位	读出时为“0”，写入时写“0”	RW

注:

本寄存器中的标志位写 1 清 0。

8.2.4 SRAM0 错误注入使能寄存器 (SRAM0_EIEN)

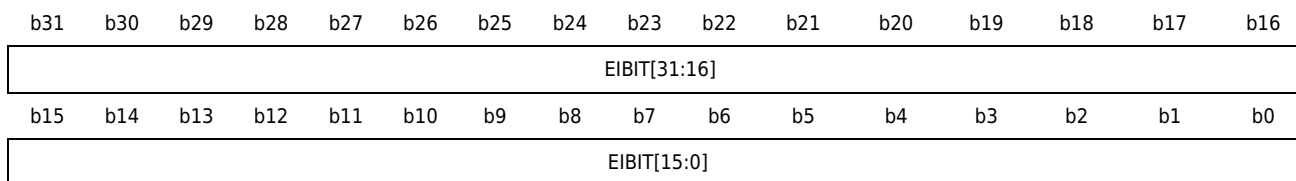
复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
b0	EIEN	错误注入使能	0: 错误注入禁止 1: 错误注入有效	RW

8.2.5 SRAM0 错误注入位使能寄存器 0 (SRAM0_EIBIT0)

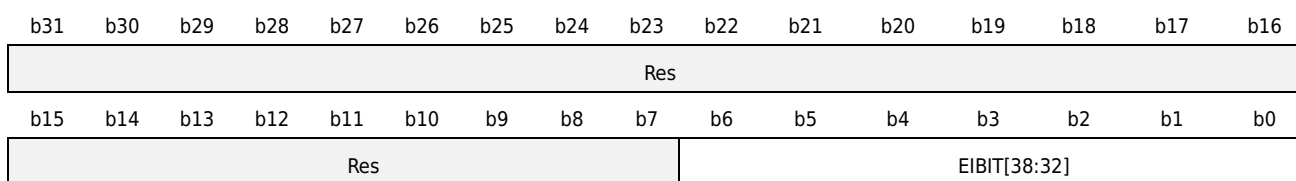
复位值: 0x0000 0000



位/位域	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	RW

8.2.6 SRAM0 错误注入位使能寄存器 1 (SRAM0_EIBIT1)

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b7	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

注:

SRAM0 输出数据的 32~38 位为 ECC 位。

8.2.7 SRAM0 校验错误地址寄存器 (SRAM0_ECCERRADDR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	ECCERRADDR [14:0]														

位/位域	标记	位名	功能	读写
b31~b15	Res	保留位	读出时为“0”，写入时写“0”	RW
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 (偏移地址)	R

8.2.8 SRAMH 错误注入使能寄存器 (SRAMH_EIEN)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
b0	EIEN	错误注入使能	0: 错误注入禁止 1: 错误注入有效	RW

8.2.9 SRAMH 错误注入位使能寄存器 0 (SRAMH_EIBIT0)

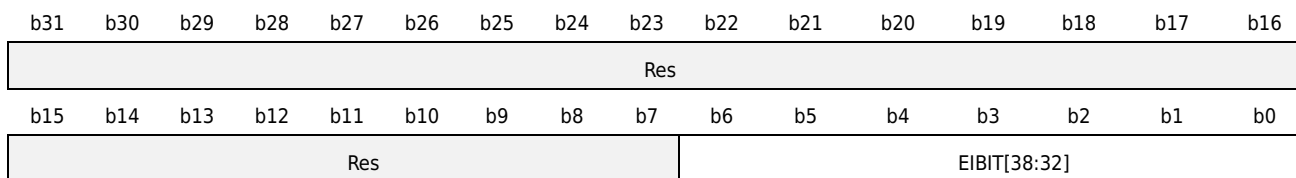
复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b0	EIBIT[31:0]	错误注入位使能	控制SRAMH输出数据的0~31位是否取反 0: 不取反 1: 取反	RW

8.2.10 SRAMH 错误注入位使能寄存器 1 (SRAMH_EIBIT1)

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b7	Res	保留位	读出时为“0”，写入时写“0”	RW
b6~b0	EIBIT[38:32]	错误注入位使能	控制SRAMH输出数据的32~38位是否取反 0: 不取反 1: 取反	RW

注:

SRAMH 输出数据的 32~38 位为 ECC 位。

8.2.11 SRAMH 校验错误地址寄存器 (SRAMH_ECCERRADDR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	ECCERRADDR [14:0]														

位/位域	标记	位名	功能	读写
b31~b15	Res	保留位	读出时为“0”，写入时写“0”	RW
b14~b0	ECCERRADDR [14:0]	校验错误地址	记录SRAMH发生1bit或2bit ECC错误时的访问地址 (偏移地址)	R

8.2.12 Ret SRAM 错误注入使能寄存器 (SRAMB_EIEN)

复位值: 0x0000 0000

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/位域	标记	位名	功能	读写
b31~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
b0	EIEN	错误注入使能	0: 错误注入禁止 1: 错误注入有效	RW

8.2.13 Ret SRAM 错误注入位使能寄存器 0 (SRAMB_EIBIT0)

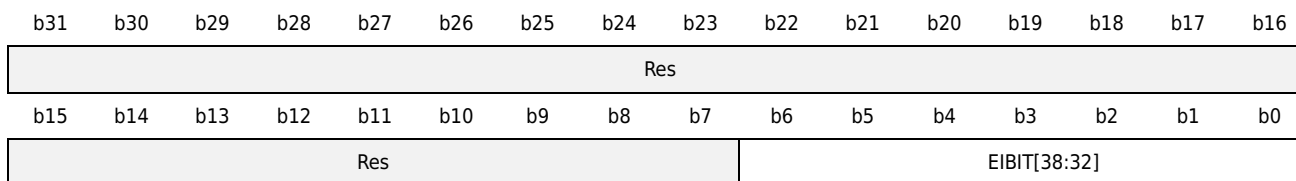
复位值: 0x0000 0000

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]															

位/位域	标记	位名	功能	读写
b31~b0	EIBIT[31:0]	错误注入位使能	控制Ret SRAM输出数据的0~31位是否取反 0: 不取反 1: 取反	RW

8.2.14 Ret SRAM 错误注入位使能寄存器 1 (SRAMB_EIBIT1)

复位值: 0x0000 0000



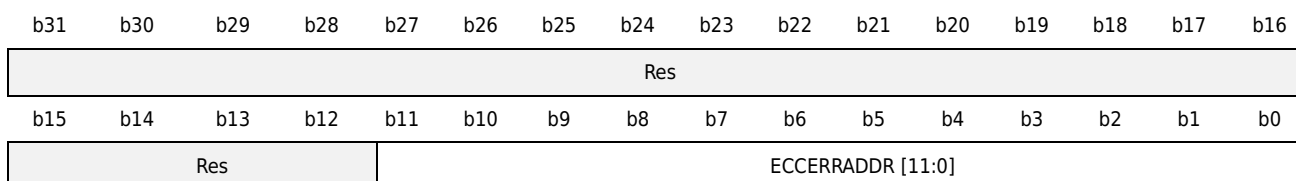
位/位域	标记	位名	功能	读写
b31~b7	Res	保留位	读出时为“0”，写入时写“0”	RW
b6~b0	EIBIT[38:32]	错误注入位使能	控制Ret SRAM输出数据的32~38位是否取反 0: 不取反 1: 取反	RW

注:

Ret SRAM 输出数据的 32~38 位为 ECC 位。

8.2.15 Ret SRAM 校验错误地址寄存器 (SRAMB_ECCERRADDR)

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
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 (偏移地址)	R

8.3 注意事项

CPU 或 DMA 同时访问 SRAM0 任意地址时, DMA 或 CPU 不要连续访问 SRAM0 的 0x2000 0000~0x2000 0002 地址。

9 通用 IO (GPIO)

本章中使用的一些略称：

- Px (x=A~D, 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~D, 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 (Nmos Open Drain) NMOS 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 51 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 功能说明

9.2.1 GPIO 端口基本结构

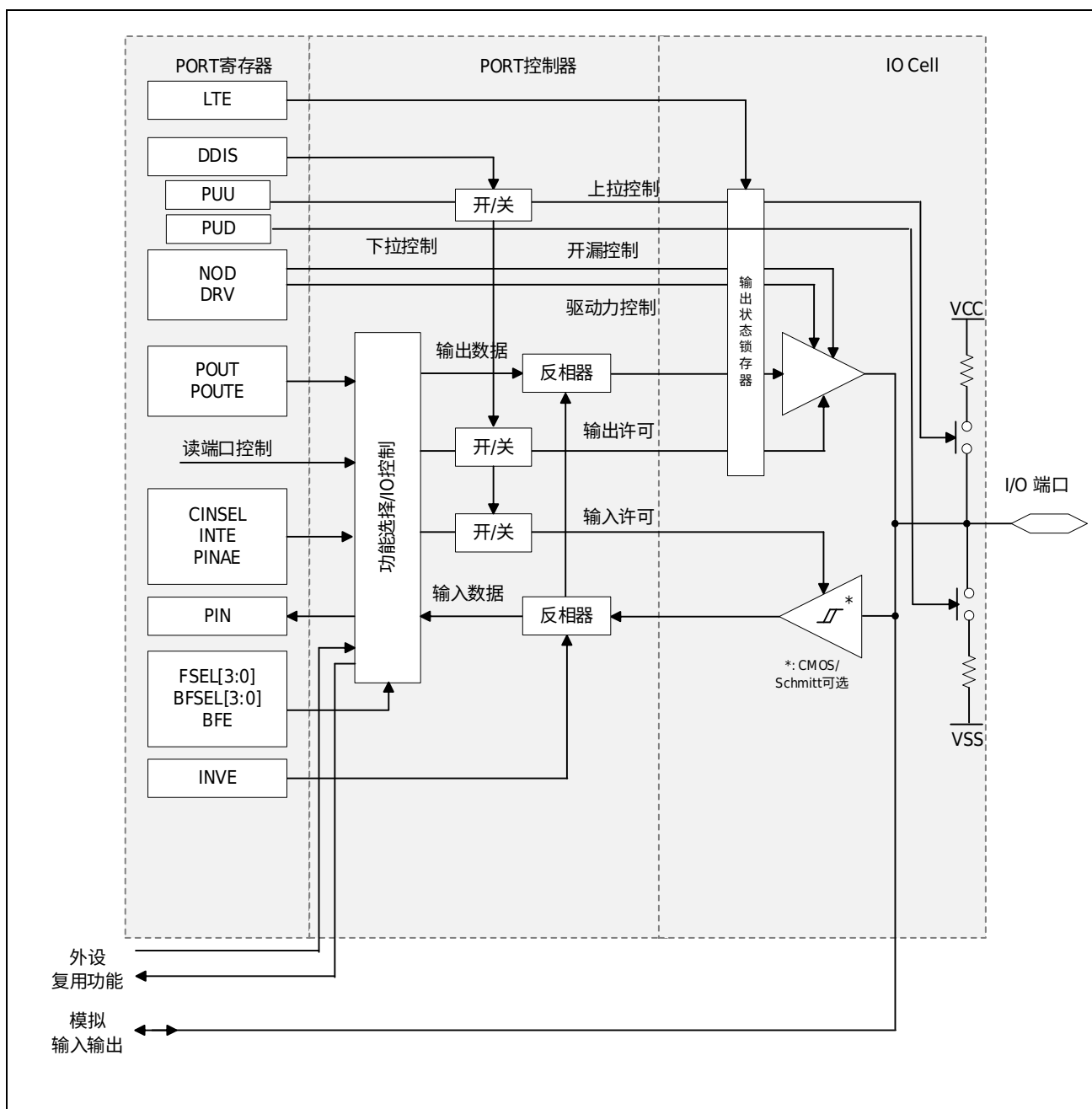


图 9-1 端口基本结构示意图

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。

9.2.2 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.PINy 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.RDWT[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.POUTEy 等价。可通过设置 PCRxy.POUT 来控制 I/O 的输出值，PCRxy.POUT 寄存器位与 PODRx.POUTy 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）。

9.2.3 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 51 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。

9.2.4 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。

9.2.5 Event Port 输入输出功能

支持 4 组 Event Port，其中 Event Port (0~3) 每组 16 个端口，Event Port4 只有一个端口。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他外设 (如 TIMER, ADC, DMA 等) 开始特定的动作。也可以作为被触发对象，接收事件，自动输入或输出。

作为触发源时，设置 PEVNTISRm、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，输出到其他周边设备以触发其开始动作。

作为被触发对象时，设置 PEVNT_TRGSELM 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 PEVNTIDRm。

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。

9.2.6 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)】。

9.2.7 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。

9.2.8 通用控制

1. 上拉电阻和下拉电阻

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。

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。

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。

9.3 寄存器描述

表 9-1 GPIO 基地址

名称	基地址	描述
GPIO	0x4005 3800	GPIO基地址

表 9-2 GPIO 寄存器

寄存器	描述	偏移地址	位宽	复位值
PIDRx	通用输入数据寄存器	0x00+0x10*n ⁽¹⁾	16/32	0xXXXX
PODRx	通用输出数据寄存器	0x04+0x10*n	16/32	0x0000
POERx	通用输出许可寄存器	0x06+0x10*n	16/32	0x0000
POSRx	通用输出置位寄存器	0x08+0x10*n	16/32	0x0000
PORRx	通用输出复位寄存器	0x0A+0x10*n	16/32	0x0000
POTRx	通用输出翻转寄存器	0x0C+0x10*n	16/32	0x0000
PHRPWM	PHRPWM控制寄存器	0x3F0	16/32	0x0000 0000
PSPCR	特殊控制寄存器	0x3F4	16/32	0x001F
PCCR	公共控制寄存器	0x3F8	16/32	0x1000
PWPR	写保护寄存器	0x3FC	16/32	0x0000
PCRxy	通用控制寄存器	0x400+0x40*n+0x4*y ⁽¹⁾	16/32	0x0X00 ⁽²⁾
PFSRxy	功能选择寄存器	0x402+0x40*n+0x4*y	16/32	0x0000

注：

- (1) 地址计算公式中 x=A~D、F 对应 n=0~3, 5; y=0~15。
 (2) 32KHz 副振荡器复用端口 PCRC14 和 PCRC15 的复位值为 0x8100。

表 9-3 Event Port 基地址

名称	基地址	描述
Event Port	0x4001 0800	Event Port基地址

表 9-4 Event Port 寄存器

寄存器	描述	偏移地址	位宽	复位值
PEVNTDIRRm	Event Port方向选择寄存器	0x100+0x1C*(m-1)	32	0x0000 0000
PEVNTIDRm	Event Port输入数据寄存器	0x104+0x1C*(m-1)	32	0x0000 0000
PEVNTODRm	Event Port输出数据寄存器	0x108+0x1C*(m-1)	32	0x0000 0000
PEVNTORRm	Event Port输出数据复位寄存器	0x10C+0x1C*(m-1)	32	0x0000 0000
PEVNTOSRm	Event Port输出数据置位寄存器	0x110+0x1C*(m-1)	32	0x0000 0000
PEVNTISRm	Event Port上升沿输入许可寄存器	0x114+0x1C*(m-1)	32	0x0000 0000
PEVNTFALRm	Event Port下降沿输入许可寄存器	0x118+0x1C*(m-1)	32	0x0000 0000
PEVNTNFCR	Event Port输入滤波控制寄存器	0x170	32	0x0000 0000

注：m=1~4。

9.3.1 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
PIN15	PIN14	PIN13	PIN12	PIN11	PIN10	PIN9	PIN8	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0

位/位域	标记	位名	功能	读写
b15~b0	PIN _y y=0~15	输入状态	每个I/O端口对应一个PIN位 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

注:

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 $PFSR_{xy}.FSEL[5:0]$ 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 $0x1$ 。

9.3.2 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
POUT15	POUT14	POUT13	POUT12	POUT11	POUT10	POUT9	POUT8	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0

位/位域	标记	位名	功能	读写
b15~b0	POUT _y y=0~15	输出数据	每个I/O端口对应一个POUT位 0: 输出低电平 1: 输出高电平	RW

注:

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。

9.3.3 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
POUTE15	POUTE14	POUTE13	POUTE12	POUTE11	POUTE10	POUTE9	POUTE8	POUTE7	POUTE6	POUTE5	POUTE4	POUTE3	POUTE2	POUTE1	POUTE0

位/位域	标记	位名	功能	读写
b15~b0	POUTE _y y=0~15	输出许可	每个I/O端口对应一个POUTE位 0: 输出禁止 1: 输出许可	RW

注:

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。

9.3.4 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
POS15	POS14	POS13	POS12	POS11	POS10	POS9	POS8	POS7	POS6	POS5	POS4	POS3	POS2	POS1	POS0

位/位域	标记	位名	功能	读写
b15~b0	POS _y y=0~15	输出高	每个I/O端口对应一个POS位 0: 对应PODRx.POUT无变化 1: 对应PODRx.POUT置1	W

注:

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。

9.3.5 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
POR15	POR14	POR13	POR12	POR11	POR10	POR9	POR8	POR7	POR6	POR5	POR4	POR3	POR2	POR1	POR0

位/位域	标记	位名	功能	读写
b15~b0	PORy y=0~15	输出低	每个I/O端口对应一个POR位 0: 对应PODRx.POUT无变化 1: 对应PODRx.POUT清零	W

注:

此寄存器的读出值始终为 0x0000。

9.3.6 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
POT15	POT14	POT13	POT12	POT11	POT10	POT9	POT8	POT7	POT6	POT5	POT4	POT3	POT2	POT1	POT0

位/位域	标记	位名	功能	读写
b15~b0	POTy y=0~15	输出翻转	每个I/O端口对应一个POT位 0: 对应PODRx.POUT无变化 1: 对应PODRx.POUT取反	W

注:

此寄存器的读出值始终为 0x0000。

9.3.7 HRPWM 控制寄存器 (HRPWM)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				HREX[11:0]											
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HRSEL[11:0]											

位	标记	位名	功能	读写
b31~28	Res	保留位	读出时为0, 写时请写0	RW
b27	HREX[11]	高精度PWM输出交换	0: PC9上的HRPWM_6_PWM输出不交换 1: PC9上的HRPWM_6_PWM输出交换为对称通道	RW
b26	HREX[10]	高精度PWM输出交换	0: PC8上的HRPWM_6_PWM输出不交换 1: PC8上的HRPWM_6_PWM输出交换为对称通道	RW
b25	HREX[9]	高精度PWM输出交换	0: PB15上的HRPWM_5_PWM输出不交换 1: PB15上的HRPWM_5_PWM输出交换为对称通道	RW
b24	HREX[8]	高精度PWM输出交换	0: PB14上的HRPWM_5_PWM输出不交换 1: PB14上的HRPWM_5_PWM输出交换为对称通道	RW
b23	HREX[7]	高精度PWM输出交换	0: PB13上的HRPWM_4_PWM输出不交换 1: PB13上的HRPWM_4_PWM输出交换为对称通道	RW
b22	HREX[6]	高精度PWM输出交换	0: PB12上的HRPWM_4_PWM输出不交换 1: PB12上的HRPWM_4_PWM输出交换为对称通道	RW
b21	HREX[5]	高精度PWM输出交换	0: PA11上的HRPWM_3_PWM输出不交换 1: PA11上的HRPWM_3_PWM输出交换为对称通道	RW
b20	HREX[4]	高精度PWM输出交换	0: PA10上的HRPWM_3_PWM输出不交换 1: PA10上的HRPWM_3_PWM输出交换为对称通道	RW
b19	HREX[3]	高精度PWM输出交换	0: PA9上的HRPWM_2_PWM输出不交换 1: PA9上的HRPWM_2_PWM输出交换为对称通道	RW
b18	HREX[2]	高精度PWM输出交换	0: PA8上的HRPWM_2_PWM输出不交换 1: PA8上的HRPWM_2_PWM输出交换为对称通道	RW
b17	HREX[1]	高精度PWM输出交换	0: PC7上的HRPWM_1_PWM输出不交换 1: PC7上的HRPWM_1_PWM输出交换为对称通道	RW
b16	HREX[0]	高精度PWM输出交换	0: PC6上的HRPWM_1_PWM输出不交换 1: PC6上的HRPWM_1_PWM输出交换为对称通道	RW
b15~12	Res	保留位	读出时为0, 写时请写0	RW
b11	HRSEL[11]	高精度PWM功能选择	0: PC9上的HRPWM_6_PWMB输出功能未选中 1: PC9上的HRPWM_6_PWMB输出功能已选中	RW
b10	HRSEL[10]	高精度PWM功能选择	0: PC8上的HRPWM_6_PWMA输出功能未选中 1: PC8上的HRPWM_6_PWMA输出功能已选中	RW
b9	HRSEL[9]	高精度PWM功能选择	0: PB15上的HRPWM_5_PWMB输出功能未选中 1: PB15上的HRPWM_5_PWMB输出功能已选中	RW
b8	HRSEL[8]	高精度PWM功能选择	0: PB14上的HRPWM_5_PWMA输出功能未选中 1: PB14上的HRPWM_5_PWMA输出功能已选中	RW
b7	HRSEL[7]	高精度PWM功能选择	0: PB13上的HRPWM_4_PWMB输出功能未选中 1: PB13上的HRPWM_4_PWMB输出功能已选中	RW
b6	HRSEL[6]	高精度PWM功能选择	0: PB12上的HRPWM_4_PWMA输出功能未选中	RW

			1: PB12上的HRPWM_4_PWMA输出功能已选中	
b5	HRSEL[5]	高精度PWM功能选择	0: PA11上的HRPWM_3_PWMB输出功能未选中 1: PA11上的HRPWM_3_PWMB输出功能已选中	RW
b4	HRSEL[4]	高精度PWM功能选择	0: PB10上的HRPWM_3_PWMA输出功能未选中 1: PB10上的HRPWM_3_PWMA输出功能已选中	RW
b3	HRSEL[3]	高精度PWM功能选择	0: PA9上的HRPWM_2_PWMB输出功能未选中 1: PA9上的HRPWM_2_PWMB输出功能已选中	RW
b2	HRSEL[2]	高精度PWM功能选择	0: PA8上的HRPWM_2_PWMA输出功能未选中 1: PA8上的HRPWM_2_PWMA输出功能已选中	RW
b1	HRSEL[1]	高精度PWM功能选择	0: PC7上的HRPWM_1_PWMB输出功能未选中 1: PC7上的HRPWM_1_PWMB输出功能已选中	RW
b0	HRSEL[0]	高精度PWM功能选择	0: PC6上的HRPWM_1_PWMA输出功能未选中 1: PC6上的HRPWM_1_PWMA输出功能已选中	RW

9.3.8 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
Res											SPFE[4:0]				

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

注:

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

9.3.9 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
Res	RDWT[2:0]		Res						BFSEL[5:0]						

位/位域	标记	位名	功能	读写		
b15	Res	保留位	读出时为0, 写时请写0	RW		
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, 默认值0b001	RW		
			工作频率		等待周期	设定值
			~50MHz		0周期	0~7
			50~100MHz		1周期	1~7
100~120MHz	2周期	2~7				
b11~b6	Res	保留位	读出时为0, 写时请写0	RW		
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	RW		

9.3.10 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]								Res							WE

位/位域	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	Res	保留位	读出时为0, 写时请写0	RW
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	RW

9.3.11 通用控制寄存器 (PCRxy)

复位值: 0x0X00

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

位/位域	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	RW
b14	LTE	输出状态锁存	0: 输出锁存无效 1: 输出锁存有效	RW
b13	PINAE	输入常开	0: 输入MOS常开无效 1: 输入MOS常开有效	RW
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	RW
b11	Res	保留位	读出时为0, 写时请写0	RW
b10	CINSEL	输入模式选择	0: Schmitt输入 1: CMOS输入	RW
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	RW
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POUTy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	RW
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	RW
b5~b4	DRV[1:0]	驱动模式选择	00: 低驱动力模式 01: 中驱动力模式 10, 11: 高驱动力模式	RW
b3	Res	保留位	读出时为0, 写时请写0	RW
b2	NOD	NMOS开漏	0: 正常CMOS输出模式 1: NMOS开漏输出	RW
b1	POUTE	输出许可	0: 输出禁止 1: 输出许可 与寄存器POERx中POUTEy功能一致	RW
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	RW

注:

1. 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。

2. *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不到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。
3. *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。
4. *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。
5. 以下端口通用控制寄存器 *PCR* 的复位值不是 0x0X00，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。

9.3.12 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
Res							BFE	Res			FSEL[5:0]				

位/位域	标记	位名	功能	读写
b15~b9	Res	保留位	读出时为0, 写时请写0	RW
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	RW
b7~b6	Res	保留位	读出时为0, 写时请写0	RW
b5~b0	FSEL[5:0]	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	RW

注:

1. 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 0 表示选择 Func0, 设为 1 表示选择 Func1, 以此类推, 设 15 表示选择 Func15。其中 Func0 对应的通用输出功能 GPO。
2. 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。

9.3.13 Event Port 方向选择寄存器 (PEVNTDIRRm)

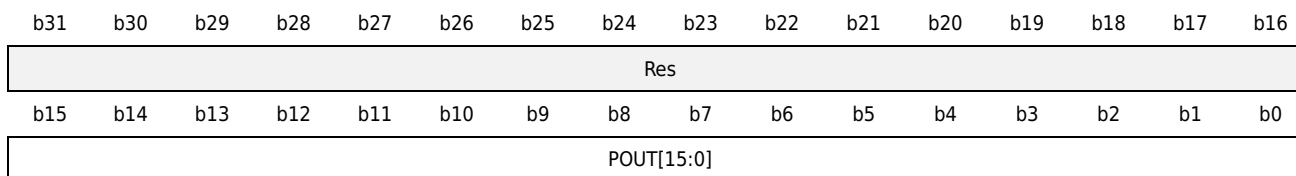
复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为0, 写时请写0	RW
b15~b0	PDIR15:0]	方向选择	0: Event Port为输入功能 1: Event Port为输出功能	RW

9.3.14 Event Port 输入数据寄存器 (PEVNTIDRm)

复位值: 0x0000 0000



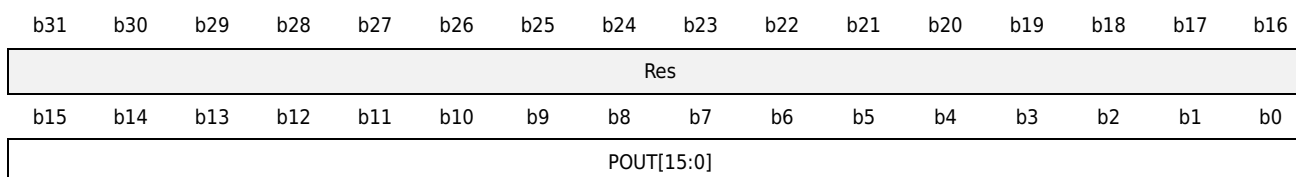
位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为0, 写时请写0	RW
b15~b0	POUT[15:0]	端口输入状态	0: 事件触发时Event Port输入状态为低电平 1: 事件触发时Event Port输入状态为高电平	R

注:

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。

9.3.15 Event Port 输出数据寄存器 (PEVNTODRm)

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为0, 写时请写0	RW
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	RW

注:

当 *Event Port* 的方向设置为输出时, 在设置的事件触发前, *Event Port* 输出值为对应 *POUT* 位的写入值。当设置的事件触发后, 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。

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

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为0, 写时请写0	RW
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	RW

注:

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

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

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为0, 写时请写0	RW
b15~b0	POS[15:0]	输出位置位	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	RW

注:

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

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

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为0, 写时请写0	RW
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	RW

注:

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, 用以触发其他外设。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。

9.3.19 Event Port 下降沿输入许可寄存器 (PEVNTRIFALRm)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为0, 写时请写0	RW
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 PEVNTRIFALRm.FAL[n]对应EVNTPmn	RW

注:

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, 用以触发其他外设。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。

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

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				DIVS4[1:0]			NFEN4	Res				DIVS3[1:0]		NFEN3	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				DIVS2[1:0]			NFEN2	Res				DIVS1[1:0]		NFEN1	

位/位域	标记	位名	功能	读写
b31~b27	Res	保留位	读出时为0, 写时请写0	RW
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	RW
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	RW
b23~b19	Res	保留位	读出时为0, 写时请写0	RW
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	RW
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	RW
b15~b11	Res	保留位	读出时为0, 写时请写0	RW
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	RW
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	RW
b7~b3	Res	保留位	读出时为0, 写时请写0	RW
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	RW
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	RW

9.4 注意事项

1. 请不要将同一功能设置到多个端口上。
2. 使用模拟功能时，请将相应端口的数字功能关闭（PCRxy.DDIS=1）。
3. 请在输出锁存有效时（PCRxy.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。
4. 9.3 小节中所述的寄存器，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 9-5 32bit 访问时 PORT 寄存器一览

地址	b31 ~ b16	b15 ~ b0
0x4005 3800+0x10*n	Res	PIDRx*1
0x4005 3804+0x10*n	POERx	PODRx
0x4005 3808+0x10*n	PORRx	POSRx
0x4005 380C+0x10*n	Res	POTRx
0x4005 3BF0	PHRPWM	PHRPWM
0x4005 3BF4	Res	PSPCR
0x4005 3BF8	Res	PCCR
0x4005 3BFC	Res	PWPR
0x4005 3C00+0x40*n+0x04*y	PFSRxy	PCRxy

注：

*1: 地址计算公式中 $x=A\sim D$, F 对应 $n=0\sim 3, 5$ 。

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 303 个中断事件, 处理后作为中断请求 (IRQ) 发送给 NVIC, 支持 145 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】。
- 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

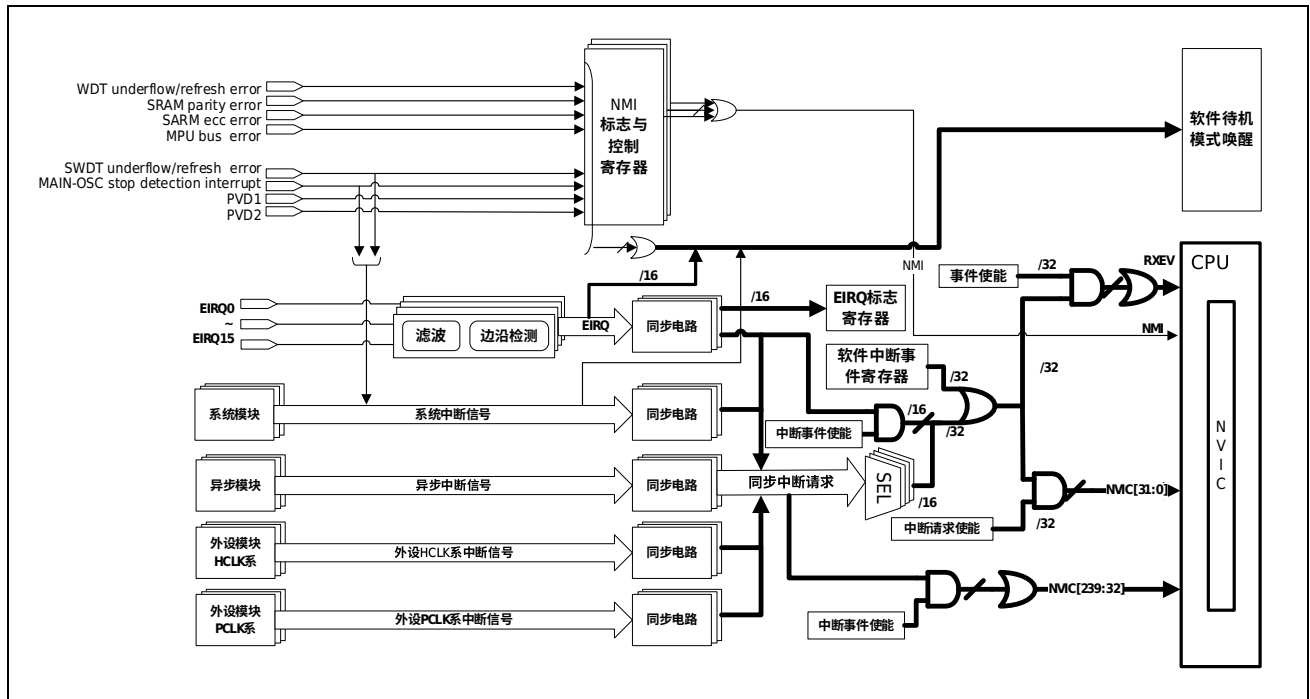


图 10-1 中断系统框图

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				
0x0000 0000	0	-	ARM core	Initial Stack Pointer
0x0000 0004	1	-	ARM core	Initial Program Counter
0x0000 0008	2	-	ARM core	Non-maskable Interrupt(NMI)
0x0000 000C	3	-	ARM core	Hard Fault
0x0000 0010	4	-	ARM core	MemManage Fault
0x0000 0014	5	-	ARM core	Bus Fault
0x0000 0018	6	-	ARM core	Usage Fault
0x0000 001C	7	-	ARM core	Reserved
0x0000 0020	8	-	ARM core	Reserved
0x0000 0024	9	-	ARM core	Reserved
0x0000 0028	10	-	ARM core	Reserved
0x0000 002C	11	-	ARM core	Supervisor Call(SVCall)
0x0000 0030	12	-	ARM core	Debug Monitor
0x0000 0034	13	-	ARM core	Reserved
0x0000 0038	14	-	ARM core	Pendable request for system service(PendSV)
0x0000 003C	15	-	ARM core	System tick timer(SysTick)
非ARM内核中断				
0x0000 0040	16	0	INT_INTSEL0	寄存器INTC_INTSEL0选择的 中断事件/软件中断0。
0x0000 0044	17	1	INT_INTSEL1	寄存器INTC_INTSEL1选择的 中断事件/软件中断1。
0x0000 0048	18	2	INT_INTSEL2	寄存器INTC_INTSEL2选择的 中断事件/软件中断2。
0x0000 004C	19	3	INT_INTSEL3	寄存器INTC_INTSEL3选择的 中断事件/软件中断3。
0x0000 0050	20	4	INT_INTSEL4	寄存器INTC_INTSEL4选择的 中断事件/软件中断4。
0x0000 0054	21	5	INT_INTSEL5	寄存器INTC_INTSEL5选择的 中断事件/软件中断5。
0x0000 0058	22	6	INT_INTSEL6	寄存器INTC_INTSEL6选择的 中断事件/软件中断6。
0x0000 005C	23	7	INT_INTSEL7	寄存器INTC_INTSEL7选择的 中断事件/软件中断7。
0x0000 0060	24	8	INT_INTSEL8	寄存器INTC_INTSEL8选择的 中断事件/软件中断8。
0x0000 0064	25	9	INT_INTSEL9	寄存器INTC_INTSEL9选择的 中断事件/软件中断9。
0x0000 0068	26	10	INT_INTSEL10	寄存器INTC_INTSEL10选择的 中断事件/软件中断10。
0x0000 006C	27	11	INT_INTSEL11	寄存器INTC_INTSEL11选择的 中断事件/软件中断11。
0x0000 0070	28	12	INT_INTSEL12	寄存器INTC_INTSEL12选择的 中断事件/软件中断12。
0x0000 0074	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
0x0000 0078	30	14	INT_INTSEL14	寄存器INTC_INTSEL14选择的 中断事件/软件中断14。
0x0000 007C	31	15	INT_INTSEL15	寄存器INTC_INTSEL15选择的 中断事件/软件中断15。
0x0000 0080	32	16	PORT	外部中断0/软件中断16
0x0000 0084	33	17	PORT	外部中断1/软件中断17
0x0000 0088	34	18	PORT	外部中断2/软件中断18
0x0000 008C	35	19	PORT	外部中断3/软件中断19
0x0000 0090	36	20	PORT	外部中断4/软件中断20
0x0000 0094	37	21	PORT	外部中断5/软件中断21
0x0000 0098	38	22	PORT	外部中断6/软件中断22
0x0000 009C	39	23	PORT	外部中断7/软件中断23
0x0000 00A0	40	24	PORT	外部中断8/软件中断24
0x0000 00A4	41	25	PORT	外部中断9/软件中断25
0x0000 00A8	42	26	PORT	外部中断10/软件中断26
0x0000 00AC	43	27	PORT	外部中断11/软件中断27
0x0000 00B0	44	28	PORT	外部中断12/软件中断28
0x0000 00B4	45	29	PORT	外部中断13/软件中断29
0x0000 00B8	46	30	PORT	外部中断14/软件中断30
0x0000 00BC	47	31	PORT	外部中断15/软件中断31
0x0000 00C0	48	32	DMA	传输错误中断
0x0000 00C4	49	33	DMA	通道0传输完成中断/通道0数据块传输完成中断
0x0000 00C8	50	34	DMA	通道1传输完成中断/通道1数据块传输完成中断
0x0000 00CC	51	35	DMA	通道2传输完成中断/通道2数据块传输完成中断
0x0000 00D0	52	36	DMA	通道3传输完成中断/通道3数据块传输完成中断
0x0000 00D4	53	37	DMA	通道4传输完成中断/通道4数据块传输完成中断
0x0000 00D8	54	38	DMA	通道5传输完成中断/通道5数据块传输完成中断
0x0000 00DC	55	39	DMA	通道6传输完成中断/通道5数据块传输完成中断
0x0000 00E0	56	40	DMA	通道7传输完成中断/通道5数据块传输完成中断
0x0000 00E4	57	41	EFM	编程（擦除）错误中断/读写冲突错误中断
0x0000 00E8	58	42	EFM	操作结束中断
0x0000 00EC	59	43	Timer0_1	通道A, B计数比较匹配中断/输入捕获中断/计数溢出中断
0x0000 00F0	60	44	Timer0_2	通道A, B计数比较匹配中断/输入捕获中断/计数溢出中断
0x0000 00F4	61	45	RTC	闹钟中断/定周期中断
0x0000 00F8	62	46	CMU	XTAL振荡故障中断
0x0000 00FC	63	47	PWC	WKTM比较匹配中断
0x0000 0100	64	48	SWDT	计数下溢/刷新错误中断
0x0000 0104	65	49	FPU	FPU中断
0x0000 0108	66	50	Timer6_1	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
0x0000 010C	67	51	Timer6_1	计数周期匹配中断
0x0000 0110	68	52	Timer6_1	死区时间错误中断
0x0000 0114	69	53	Timer6_1	专用计数比较匹配中断
0x0000 0118	70	54	TimerA_1	计数周期匹配中断
0x0000 011C	71	55	TimerA_1	计数比较匹配中断
0x0000 0120	72	56	Timer6_2	通用计数比较匹配中断/输入捕获中断
0x0000 0124	73	57	Timer6_2	计数周期匹配中断
0x0000 0128	74	58	Timer6_2	死区时间错误中断
0x0000 012C	75	59	Timer6_2	专用计数比较匹配中断
0x0000 0130	76	60	TimerA_2	计数周期匹配中断
0x0000 0134	77	61	TimerA_2	计数比较匹配中断
0x0000 0138	78	62	Timer6_3	通用计数比较匹配中断/输入捕获中断
0x0000 013C	79	63	Timer6_3	计数周期匹配中断
0x0000 0140	80	64	Timer6_3	死区时间错误中断
0x0000 0144	81	65	Timer6_3	专用计数比较匹配中断
0x0000 0148	82	66	TimerA_3	计数周期匹配中断
0x0000 014C	83	67	TimerA_3	计数比较匹配中断
0x0000 0150	84	68	Timer6_4	通用计数比较匹配中断/输入捕获中断
0x0000 0154	85	69	Timer6_4	计数周期匹配中断
0x0000 0158	86	70	Timer6_4	死区时间错误中断
0x0000 015C	87	71	Timer6_4	专用计数比较匹配中断
0x0000 0160	88	72	TimerA_4	计数周期匹配中断
0x0000 0164	89	73	TimerA_4	计数比较匹配中断
0x0000 0168	90	74	HRPWM_1	通用计数比较匹配中断
0x0000 016C	91	75	HRPWM_1	计数周期/零点匹配中断
0x0000 0170	92	76	HRPWM_1	专用计数比较匹配中断
0x0000 0174	93	77	HRPWM_1	捕获中断
0x0000 0178	94	78	HRPWM_2	通用计数比较匹配中断
0x0000 017C	95	79	HRPWM_2	计数周期/零点匹配中断
0x0000 0180	96	80	HRPWM_2	专用计数比较匹配中断
0x0000 0184	97	81	HRPWM_2	捕获中断
0x0000 0188	98	82	HRPWM	突发模式周期匹配中断/校准完成中断
0x0000 018C	99	83	EMB	Group0中断
0x0000 0190	100	84	EMB	Group1中断
0x0000 0194	101	85	EMB	Group2中断
0x0000 0198	102	86	EMB	Group3中断
0x0000 019C	103	87	HRPWM_3	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
0x0000 01A0	104	88	HRPWM_3	计数周期/零点匹配中断
0x0000 01A4	105	89	HRPWM_3	专用计数比较匹配中断
0x0000 01A8	106	90	HRPWM_3	捕获中断
0x0000 01AC	107	91	HRPWM_4	通用计数比较匹配中断
0x0000 01B0	108	92	HRPWM_4	计数周期/零点匹配中断
0x0000 01B4	109	93	HRPWM_4	专用计数比较匹配中断
0x0000 01B8	110	94	HRPWM_4	捕获中断
0x0000 01BC	111	95	EMB	Group4中断
0x0000 01C0	112	96	EMB	Group5中断
0x0000 01C4	113	97	EMB	Group6中断
0x0000 01C8	114	98	EMB	Group7中断
0x0000 01CC	115	99	EMB	Group8中断
0x0000 01D0	116	100	HRPWM_5	通用计数比较匹配中断
0x0000 01D4	117	101	HRPWM_5	计数周期/零点匹配中断
0x0000 01D8	118	102	HRPWM_5	专用计数比较匹配中断
0x0000 01DC	119	103	HRPWM_5	捕获中断
0x0000 01E0	120	104	HRPWM_6	通用计数比较匹配中断
0x0000 01E4	121	105	HRPWM_6	计数周期/零点匹配中断
0x0000 01E8	122	106	HRPWM_6	专用计数比较匹配中断
0x0000 01EC	123	107	HRPWM_6	捕获中断
0x0000 01F0	124	108	MCANRAM	MCAN RAM ECC错误中断
0x0000 01F4	125	109	MCAN_1	MCAN_1中断0
0x0000 01F8	126	110	MCAN_1	MCAN_1中断1
0x0000 01FC	127	111	MCAN_2	MCAN_2中断0
0x0000 0200	128	112	MCAN_2	MCAN_2中断1
0x0000 0204	129	113	Timer4_1	通用计数比较匹配中断
0x0000 0208	130	114	Timer4_1	计数周期匹配中断
0x0000 020C	131	115	Timer4_1	重载计数匹配中断
0x0000 0210	132	116	Timer4_1	专用计数比较匹配中断
0x0000 0214	133	117	CMP	CMP1中断
0x0000 0218	134	118	CMP	CMP2中断
0x0000 021C	135	119	CMP	CMP3中断
0x0000 0220	136	120	I2C	I2C中断
0x0000 0224	137	121	USART_1	USART_1中断 (TCI及RX线唤醒停止模式中断除外)
0x0000 0228	138	122	USART_1	USART_1_TCI中断
0x0000 022C	139	123	SPI	SPI中断
0x0000 0230	140	124	TimerA_5	计数周期匹配中断

向量地址	向量序号	IRQ 序号	中断源	说明
0x0000 0234	141	125	TimerA_5	计数比较匹配中断
0x0000 0238	142	126	PORT	Event Port1中断
0x0000 023C	143	127	PORT	Event Port2中断
0x0000 0240	144	128	PORT	Event Port3中断
0x0000 0244	145	129	PORT	Event Port4中断
0x0000 0248	146	130	USART_2	USART_2中断 (TCI中断除外)
0x0000 024C	147	131	USART_2	USART_2_TCI中断
0x0000 0250	148	132	USART_3	USART_3中断 (TCI中断除外)
0x0000 0254	149	133	USART_3	USART_3_TCI中断
0x0000 0258	150	134	USART_4	USART_4中断 (TCI中断除外)
0x0000 025C	151	135	USART_4	USART_4_TCI中断
0x0000 0264	153	137	USART_1	RX线唤醒停止模式中断
0x0000 0268	154	138	PWC	可编程电压检测1 (PVD1) 中断
0x0000 026C	155	139	PWC	可编程电压检测1 (PVD2) 中断
0x0000 0270	156	140	CMU	FCM时钟频率测量中断
0x0000 0274	157	141	WDT	计数下溢/刷新错误中断
0x0000 0278	158	142	CTC	HRC频率校准值溢出中断
0x0000 027C	159	143	ADC_1	ADC_1中断
0x0000 0280	160	144	ADC_2	ADC_2中断
0x0000 0284	161	145	ADC_3	ADC_3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 136 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	中断类型*1	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*2)	类型B (固定)
0	0x000	PORT	EIRQ0	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[16]
1	0x001		EIRQ1	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[17]
2	0x002		EIRQ2	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[18]
3	0x003		EIRQ3	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[19]
4	0x004		EIRQ4	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[20]
5	0x005		EIRQ5	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[21]
6	0x006		EIRQ6	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[22]
7	0x007		EIRQ7	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[23]
8	0x008		EIRQ8	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[24]
9	0x009		EIRQ9	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[25]
10	0x00A		EIRQ10	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[26]
11	0x00B		EIRQ11	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[27]
12	0x00C		EIRQ12	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[28]
13	0x00D		EIRQ13	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[29]
14	0x00E		EIRQ14	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[30]
15	0x00F		EIRQ15	P/L	✓	✓	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)	-

序号	中断事件编号	模块	中断名称	中断类型*1	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*2)	类型B (固定)
29	0x01D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
30	0x01E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
31	0x01F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
32	0x020	DMA	DMA_ERR	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[32]
33	0x021		DMA_TC0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[33]
34	0x022		DMA_BTC0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
35	0x023		DMA_TC1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[34]
36	0x024		DMA_BTC1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
37	0x025		DMA_TC2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[35]
38	0x026		DMA_BTC2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
39	0x027		DMA_TC3	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[36]
40	0x028		DMA_BTC3	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
41	0x029		DMA_TC4	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[37]
42	0x02A		DMA_BTC4	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
43	0x02B		DMA_TC5	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[38]
44	0x02C		DMA_BTC5	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
45	0x02D		DMA_TC6	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[39]
46	0x02E		DMA_BTC6	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
47	0x02F		DMA_TC7	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[40]
48	0x030		DMA_BTC7	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
49	0x031	EFM	EFM_PEERR	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[41]
50	0x032		EFM_RDCOL	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
51	0x033		EFM_OPTEND	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[42]
52	0x034	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
53	0x035	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
54	0x036	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
55	0x037	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
56	0x038	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
57	0x039	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
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	FPU	FPU_ERR-	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[43]

序号	中断事件编号	模块	中断名称	中断类型*1	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*2)	类型B (固定)
64	0x040	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
65	0x041	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
66	0x042	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
67	0x043	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
68	0x044	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
69	0x045	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
70	0x046	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
71	0x047	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
72	0x048	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
73	0x049	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
74	0x04A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
75	0x04B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
76	0x04C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
77	0x04D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
78	0x04E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
79	0x04F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
80	0x050	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
81	0x051	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
82	0x052	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
83	0x053	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
84	0x054	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
85	0x055	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
86	0x056	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
87	0x057	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
88	0x058	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
89	0x059	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
90	0x05A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
91	0x05B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
92	0x05C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
93	0x05D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
94	0x05E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
95	0x05F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
96	0x060	Timer0_1	TMRO_1_CMPA	P	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	IRQ[44]
97	0x061		TMRO_1_CMPB	P	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	↑
98	0x062		TMRO_1_OVFA	P	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	↑

序号	中断事件编号	模块	中断名称	中断类型*1	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*2)	类型B (固定)
99	0x063	Timer0_1	TMR0_1_OVFB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
100	0x064	Timer0_2	TMR0_2_CMPA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[45]
101	0x065		TMR0_2_CMPB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
102	0x066		TMR0_2_OVFA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
103	0x067		TMR0_2_OVFB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
104	0x068	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
105	0x069	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
106	0x06A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
107	0x06B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
108	0x06C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
109	0x06D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
110	0x06E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
111	0x06F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
112	0x070	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
113	0x071	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
114	0x072	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
115	0x073	-	-	-	-	-	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	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[46]
122	0x07A		RTC_PRD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
123	0x07B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
124	0x07C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
125	0x07D	XTAL	XTAL_STOP	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[47]
126	0x07E	WKTm	WKTm_PRD	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[48]
127	0x07F	SWDT	SWDT_REFUDF	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[49]
128	0x080	Timer6_1	TMR6_1_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[50]
129	0x081		TMR6_1_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
130	0x082		TMR6_1_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
131	0x083		TMR6_1_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
132	0x084		TMR6_1_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
133	0x085		TMR6_1_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑

序号	中断事件编号	模块	中断名称	中断类型*1	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*2)	类型B (固定)
134	0x086	Timer6_1	TMR6_1_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[51]
135	0x087		TMR6_1_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
136	0x088		TMR6_1_GDTE	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[52]
137	0x089		TMR6_1_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[53]
138	0x08A		TMR6_1_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
139	0x08B	TimerA_1	TMRA_1_OVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[54]
140	0x08C		TMRA_1_UDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
141	0x08D		TMRA_1_CMP	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[55]
142	0x08E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
143	0x08F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
144	0x090	Timer6_2	TMR6_2_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[56]
145	0x091		TMR6_2_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
146	0x092		TMR6_2_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
147	0x093		TMR6_2_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
148	0x094		TMR6_2_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
149	0x095		TMR6_2_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
150	0x096		TMR6_2_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[57]
151	0x097		TMR6_2_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
152	0x098		TMR6_2_GDTE	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[58]
153	0x099		TMR6_2_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[59]
154	0x09A	TMR6_2_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
155	0x09B	TimerA_2	TMRA_2_OVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[60]
156	0x09C		TMRA_2_UDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
157	0x09D		TMRA_2_CMP	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[61]
158	0x09E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
159	0x09F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
160	0x0A0	Timer6_3	TMR6_3_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[62]
161	0x0A1		TMR6_3_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
162	0x0A2		TMR6_3_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
163	0x0A3		TMR6_3_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
164	0x0A4		TMR6_3_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
165	0x0A5		TMR6_3_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
166	0x0A6		TMR6_3_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[63]
167	0x0A7		TMR6_3_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
168	0x0A8		TMR6_3_GDTE	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[64]

序号	中断事件编号	模块	中断名称	中断类型*1	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*2)	类型B (固定)
169	0x0A9	Timer6_3	TMR6_3_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[65]
170	0x0AA		TMR6_3_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
171	0x0AB	TimerA_3	TMRA_3_OVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[66]
172	0x0AC		TMRA_3_UDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
173	0x0AD		TMRA_3_CMP	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[67]
174	0x0AE	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
175	0x0AF	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
176	0x0B0	Timer6_4	TMR6_4_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[68]
177	0x0B1		TMR6_4_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
178	0x0B2		TMR6_4_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
179	0x0B3		TMR6_4_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
180	0x0B4		TMR6_4_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
181	0x0B5		TMR6_4_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
182	0x0B6		TMR6_4_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[69]
183	0x0B7		TMR6_4_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
184	0x0B8		TMR6_4_GDTE	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[70]
185	0x0B9		TMR6_4_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[71]
186	0x0BA		TMR6_4_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
187	0x0BB		TimerA_4	TMRA_4_OVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)
188	0x0BC	TMRA_4_UDF		P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
189	0x0BD	TMRA_4_CMP		P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[73]
190	0x0BE	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
191	0x0BF	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
192	0x0C0	HRPWM_1	HRPWM_1_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[74]
193	0x0C1		HRPWM_1_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
194	0x0C2		HRPWM_1_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
195	0x0C3		HRPWM_1_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
196	0x0C4		HRPWM_1_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
197	0x0C5		HRPWM_1_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
198	0x0C6		HRPWM_1_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[75]
199	0x0C7		HRPWM_1_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
200	0x0C8		HRPWM_1_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[76]
201	0x0C9		HRPWM_1_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
202	0x0CA		HRPWM_1_GCAPA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[77]
203	0x0CB		HRPWM_1_GCAPB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑

序号	中断事件编号	模块	中断名称	中断类型*1	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*2)	类型B (固定)	
204	0x0CC	HRPWM_2	HRPWM_2_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[78]	
205	0x0CD		HRPWM_2_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
206	0x0CE		HRPWM_2_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
207	0x0CF		HRPWM_2_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
208	0x0D0		HRPWM_2_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
209	0x0D1		HRPWM_2_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
210	0x0D2		HRPWM_2_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[79]	
211	0x0D3		HRPWM_2_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
212	0x0D4		HRPWM_2_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[80]	
213	0x0D5		HRPWM_2_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
214	0x0D6		HRPWM_2_GCAPA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[81]	
215	0x0D7		HRPWM_2_GCAPB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
216	0x0D8		HRPWM	HRPWM_BMOVF	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[82]
217	0x0D9			HRPWM_CALENF	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
218	0x0DA	HRPWM_DLYPRT		P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
219	0x0DB	HRPWM_EEDET		P	-	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	-	
220	0x0DC	EMB	EMB_GR0	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[83]	
221	0x0DD		EMB_GR1	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[84]	
222	0x0DE		EMB_GR2	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[85]	
223	0x0DF		EMB_GR3	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[86]	
224	0x0E0	HRPWM_3	HRPWM_3_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[87]	
225	0x0E1		HRPWM_3_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
226	0x0E2		HRPWM_3_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
227	0x0E3		HRPWM_3_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
228	0x0E4		HRPWM_3_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
229	0x0E5		HRPWM_3_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
230	0x0E6		HRPWM_3_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[88]	
231	0x0E7		HRPWM_3_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
232	0x0E8		HRPWM_3_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[89]	
233	0x0E9		HRPWM_3_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
234	0x0EA		HRPWM_3_GCAPA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[90]	
235	0x0EB		HRPWM_3_GCAPB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
236	0x0EC	HRPWM_4	HRPWM_4_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[91]	
237	0x0ED		HRPWM_4_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	
238	0x0EE		HRPWM_4_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	

序号	中断事件编号	模块	中断名称	中断类型*1	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*2)	类型B (固定)
239	0x0EF	HRPWM_4	HRPWM_4_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
240	0x0F0		HRPWM_4_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
241	0x0F1		HRPWM_4_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
242	0x0F2		HRPWM_4_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[92]
243	0x0F3		HRPWM_4_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
244	0x0F4		HRPWM_4_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[93]
245	0x0F5		HRPWM_4_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
246	0x0F6		HRPWM_4_GCAPA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[94]
247	0x0F7		HRPWM_4_GCAPB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
248	0x0F8		EMB	EMB_GR4	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)
249	0x0F9	EMB_GR5		L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[96]
250	0x0FA	EMB_GR6		L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[97]
251	0x0FB	EMB_GR7		L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[98]
252	0x0FC	EMB_GR8		L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[99]
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	HRPWM_5	HRPWM_5_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[100]
257	0x101		HRPWM_5_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
258	0x102		HRPWM_5_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
259	0x103		HRPWM_5_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
260	0x104		HRPWM_5_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
261	0x105		HRPWM_5_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
262	0x106		HRPWM_5_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[101]
263	0x107		HRPWM_5_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
264	0x108		HRPWM_5_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[102]
265	0x109		HRPWM_5_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
266	0x10A		HRPWM_5_GCAPA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[103]
267	0x10B		HRPWM_5_GCAPB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
268	0x10C	HRPWM_6	HRPWM_6_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[104]
269	0x10D		HRPWM_6_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
270	0x10E		HRPWM_6_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
271	0x10F		HRPWM_6_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
272	0x110		HRPWM_6_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
273	0x111		HRPWM_6_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑

序号	中断事件编号	模块	中断名称	中断类型*1	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*2)	类型B (固定)
274	0x112	HRPWM_6	HRPWM_6_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[105]
275	0x113		HRPWM_6_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
276	0x114		HRPWM_6_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[106]
277	0x115		HRPWM_6_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
278	0x116		HRPWM_6_GCAPA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[107]
279	0x117		HRPWM_6_GCAPB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
280	0x118	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
281	0x119	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
282	0x11A	MCANRAM	MCANRAM_ECCER	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[108]
283	0x11B	MCAN	MCAN_1_INT0	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[109]
284	0x11C		MCAN_1_INT1	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[110]
285	0x11D		MCAN_2_INT0	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[111]
286	0x11E		MCAN_2_INT1	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[112]
287	0x11F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
288	0x120	Timer4	TMR4_GCMUH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[113]
289	0x121		TMR4_GCMUL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
290	0x122		TMR4_GCMVH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
291	0x123		TMR4_GCMVL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
292	0x124		TMR4_GCMWH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
293	0x125		TMR4_GCMWL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
294	0x126		TMR4_GCMXH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
295	0x127		TMR4_GCMXL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
296	0x128		TMR4_GOVF	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[114]
297	0x129		TMR4_GUDF	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
298	0x12A		TMR4_GRLU	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[115]
299	0x12B		TMR4_GRLV	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
300	0x12C		TMR4_GRLW	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
301	0x12D		TMR4_GRLX	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
302	0x12E		TMR4_SCM0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[116]
303	0x12F		TMR4_SCM1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
304	0x130		TMR4_SCM2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
305	0x131		TMR4_SCM3	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
306	0x132		TMR4_SCM4	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
307	0x133		TMR4_SCM5	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
308	0x134	TMR4_SCM6	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑	

序号	中断事件编号	模块	中断名称	中断类型*1	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*2)	类型B (固定)
309	0x135	Timer4	TMR4_SCM7	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
310	0x136	CMP	CMP_1_IRQ	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[117]
311	0x137		CMP_2_IRQ	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[118]
312	0x138		CMP_3_IRQ	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[119]
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	I2C	I2C_RXI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[120]
317	0x13D		I2C_TXI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
318	0x13E		I2C_TEI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
319	0x13F		I2C_EEI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
320	0x140	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
321	0x141	USART_1	USART_1_EI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[121]
322	0x142		USART_1_RI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
323	0x143		USART_1_TI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
324	0x144		USART_1_RTO	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
325	0x145		USART_1_TENDI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
326	0x146		USART_1_TCI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[122]
327	0x147	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
328	0x148	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
329	0x149	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
330	0x14A	SPI	SPI_SPRI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[123]
331	0x14B		SPI_SPTI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
332	0x14C		SPI_SPII	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
333	0x14D		SPI_SPEI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
334	0x14E		SPI_SPEND	P	-	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
335	0x14F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
336	0x150	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
337	0x151	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
338	0x152	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
339	0x153	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
340	0x154	TimerA_5	TMRA_5_OVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[124]
341	0x155		TMRA_5_UDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
342	0x156		TMRA_5_CMP	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[125]
343	0x157	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-

序号	中断事件编号	模块	中断名称	中断类型*1	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*2)	类型B (固定)
344	0x158	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
345	0x159	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
346	0x15A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
347	0x15B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
348	0x15C	EVENT port	EVENT_PORT1	P	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	IRQ[126]
349	0x15D		EVENT_PORT2	P	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	IRQ[127]
350	0x15E		EVENT_PORT3	P	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	IRQ[128]
351	0x15F		EVENT_PORT4	P	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	IRQ[129]
352	0x160	USART_2	USART_2_BRKWKPI	L	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	IRQ[130]
353	0x161		USART_2_EI	L	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	↑
354	0x162		USART_2_RI	P	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	↑
355	0x163		USART_2_TI	P	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	↑
356	0x164		USART_2_RTO	L	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	↑
357	0x165		USART_2_TENDI	L	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	↑
358	0x166		USART_2_TCI	L	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	IRQ[131]
359	0x167	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
360	0x168	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
361	0x169	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
362	0x16A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
363	0x16B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
364	0x16C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
365	0x16D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
366	0x16E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
367	0x16F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
368	0x170	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
369	0x171	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
370	0x172	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
371	0x173	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
372	0x174	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
373	0x175	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
374	0x176	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
375	0x177	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
376	0x178	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
377	0x179	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-

序号	中断事件编号	模块	中断名称	中断类型*1	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*2)	类型B (固定)
378	0x17A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
379	0x17B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
380	0x17C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
381	0x17D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
382	0x17E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
383	0x17F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
384	0x180	USART_3	USART_3_BRKWKPI	L	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	IRQ[132]
385	0x181		USART_3_EI	L	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	↑
386	0x182		USART_3_RI	P	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	↑
387	0x183		USART_3_TI	P	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	↑
388	0x184		USART_3_RTO	L	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	↑
389	0x185		USART_3_TENDI	L	✓	-	IRQ[0]~[15] (INTC_INTSELO~15)	↑
390	0x186		USART_3_TCI	L	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	IRQ[133]
391	0x187	USART_4	USART_4_BRKWKPI	L	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	IRQ[134]
392	0x188		USART_4_EI	L	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	↑
393	0x189		USART_4_RI	P	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	↑
394	0x18A		USART_4_TI	P	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	↑
395	0x18B		USART_4_RTO	L	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	↑
396	0x18C		USART_4_TENDI	L	✓	-	IRQ[0]~[15] (INTC_INTSELO~15)	↑
397	0x18D		USART_4_TCI	L	✓	✓	IRQ[0]~[15] (INTC_INTSELO~15)	IRQ[135]
398	0x18E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
399	0x18F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
400	0x190	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
401	0x191	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
402	0x192	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
403	0x193	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
404	0x194	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
405	0x195	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
406	0x196	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
407	0x197	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
408	0x198	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
409	0x199	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
410	0x19A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-
411	0x19B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSELO~15)	-

序号	中断事件编号	模块	中断名称	中断类型*1	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*2)	类型B (固定)
412	0x19C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
413	0x19D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
414	0x19E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
415	0x19F	AOS	AOS_STRG*3	P	-	✓	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)	-
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)	-
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)	-

序号	中断事件编号	模块	中断名称	中断类型*1	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*2)	类型B (固定)
447	0x1BF	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
448	0x1C0	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
449	0x1C1	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
450	0x1C2	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
451	0x1C3	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
452	0x1C4	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
453	0x1C5	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
454	0x1C6	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
455	0x1C7	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
456	0x1C8	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
457	0x1C9	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
458	0x1CA	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
459	0x1CB	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
460	0x1CC	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
461	0x1CD	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
462	0x1CE	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
463	0x1CF	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
464	0x1D0	USART_1	USART_1_WUPI	边沿	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[137]
465	0x1D1	PVD	PVD_PVD1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[138]
466	0x1D2		PVD_PVD2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[139]
467	0x1D3	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
468	0x1D4	FCM	FCMFERRI	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[140]
469	0x1D5		FCMMENDI	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
470	0x1D6		FCMCOVFI	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
471	0x1D7	WDT	WDT_REFUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[141]
472	0x1D8	CTC	CTC_ERR	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[142]
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	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[143]

序号	中断事件编号	模块	中断名称	中断类型*1	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*2)	类型B (固定)
481	0x1E1	ADC_1	ADC_1_EOCB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
482	0x1E2		ADC_1_CMP0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
483	0x1E3		ADC_1_CMP1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
484	0x1E4	ADC_2	ADC_2_EOCA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[144]
485	0x1E5		ADC_2_EOCB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
486	0x1E6		ADC_2_CMP0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
487	0x1E7		ADC_2_CMP1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
488	0x1E8	ADC_3	ADC_3_EOCA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[145]
489	0x1E9		ADC_3_EOCB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
490	0x1EA		ADC_3_CMP0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
491	0x1EB		ADC_3_CMP1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
492	0x1EC	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
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)	-
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: “P”表示脉冲中断, “L”表示电平中断。电平中断必须在中断处理程序中清除中断标志位。

*2: 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。

*3: AOS_STRG 由软件设定寄存器 INTSFTRG 的 STRG 位产生。

*4: “↑”表示 IRQ 序号同上。

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的功能。
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=0~15），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 NVIC 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】章节。使用 NVIC 中断请求 0~15 时可将中断事件使能寄存器 INTC_INTEN 中对应的中断事件使能位置 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，由中断请求使能寄存器 INTC_IER 控制许可。

10.4.5 事件输入

编号为 0~31 的 NVIC 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：

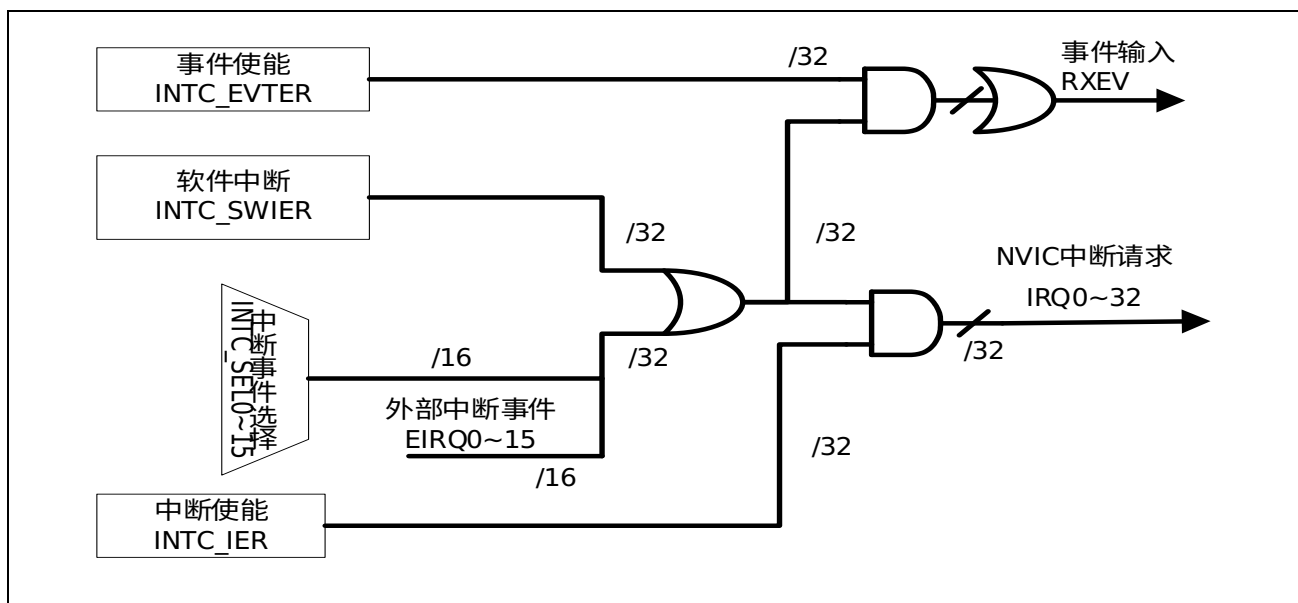


图 10-2 中断请求/事件输入选择

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=0~15）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。

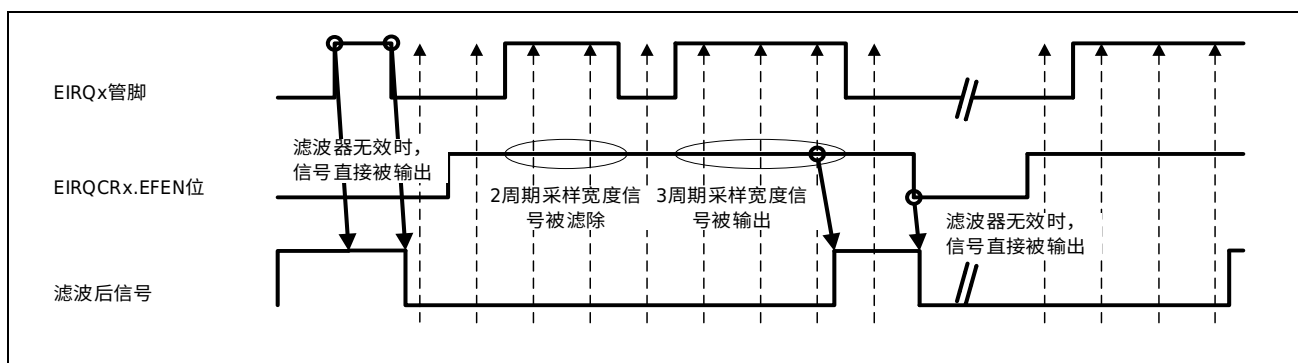


图 10-3 数字滤波器工作示意图

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 0~31 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 0~31 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：检测主发振器停止中断、SWDT 下溢/刷新中断。
- 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、上电复位以及欠压复位条件返回。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 寄存器描述

表 10-4 INTC 基地址

名称	基地址	描述
INTC	0x4005 1000	INTC基地址

表 10-5 INTC 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
INTC_NMIER	不可屏蔽中断使能寄存器	0x0004	32	0x0000 0000
INTC_NMIFR	不可屏蔽中断标志寄存器	0x0008	32	0x0000 0000
INTC_NMIFCR	不可屏蔽中断标志清除寄存器	0x000C	32	0x0000 0000
INTC_EIRQCR0	外部中断控制寄存器0	0x0010	32	0x0000 0000
INTC_EIRQCR1	外部中断控制寄存器1	0x0014	32	0x0000 0000
INTC_EIRQCR2	外部中断控制寄存器2	0x0018	32	0x0000 0000
INTC_EIRQCR3	外部中断控制寄存器3	0x001C	32	0x0000 0000
INTC_EIRQCR4	外部中断控制寄存器4	0x0020	32	0x0000 0000
INTC_EIRQCR5	外部中断控制寄存器5	0x0024	32	0x0000 0000
INTC_EIRQCR6	外部中断控制寄存器6	0x0028	32	0x0000 0000
INTC_EIRQCR7	外部中断控制寄存器7	0x002C	32	0x0000 0000
INTC_EIRQCR8	外部中断控制寄存器8	0x0030	32	0x0000 0000
INTC_EIRQCR9	外部中断控制寄存器9	0x0034	32	0x0000 0000
INTC_EIRQCR10	外部中断控制寄存器10	0x0038	32	0x0000 0000
INTC_EIRQCR11	外部中断控制寄存器11	0x003C	32	0x0000 0000
INTC_EIRQCR12	外部中断控制寄存器12	0x0040	32	0x0000 0000
INTC_EIRQCR13	外部中断控制寄存器13	0x0044	32	0x0000 0000
INTC_EIRQCR14	外部中断控制寄存器14	0x0048	32	0x0000 0000
INTC_EIRQCR15	外部中断控制寄存器15	0x004C	32	0x0000 0000
INTC_WKEN	停止模式唤醒事件使能寄存器	0x0050	32	0x0000 0000
INTC_EIFR	外部中断标志寄存器	0x0054	32	0x0000 0000
INTC_EIFCR	外部中断标志清除寄存器	0x0058	32	0x0000 0000
INTC_INTSEL0	中断事件选择寄存器0	0x005C	32	0x0000 01FF
INTC_INTSEL1	中断事件选择寄存器1	0x0060	32	0x0000 01FF
INTC_INTSEL2	中断事件选择寄存器2	0x0064	32	0x0000 01FF
INTC_INTSEL3	中断事件选择寄存器3	0x0068	32	0x0000 01FF
INTC_INTSEL4	中断事件选择寄存器4	0x006C	32	0x0000 01FF
INTC_INTSEL5	中断事件选择寄存器5	0x0070	32	0x0000 01FF
INTC_INTSEL6	中断事件选择寄存器6	0x0074	32	0x0000 01FF
INTC_INTSEL7	中断事件选择寄存器7	0x0078	32	0x0000 01FF
INTC_INTSEL8	中断事件选择寄存器8	0x007C	32	0x0000 01FF

寄存器	描述	偏移地址	位宽	复位值
INTC_INTSEL9	中断事件选择寄存器9	0x0080	32	0x0000 01FF
INTC_INTSEL10	中断事件选择寄存器10	0x0084	32	0x0000 01FF
INTC_INTSEL11	中断事件选择寄存器11	0x0088	32	0x0000 01FF
INTC_INTSEL12	中断事件选择寄存器12	0x008C	32	0x0000 01FF
INTC_INTSEL13	中断事件选择寄存器13	0x0090	32	0x0000 01FF
INTC_INTSEL14	中断事件选择寄存器14	0x0094	32	0x0000 01FF
INTC_INTSEL15	中断事件选择寄存器15	0x0098	32	0x0000 01FF
INTC_INTEN0	中断事件使能寄存器0	0x025C	32	0xFFFF FFFF
INTC_INTEN1	中断事件使能寄存器1	0x0260	32	0xFFFF FFFF
INTC_INTEN2	中断事件使能寄存器2	0x0264	32	0xFFFF FFFF
INTC_INTEN3	中断事件使能寄存器3	0x0268	32	0xFFFF FFFF
INTC_INTEN4	中断事件使能寄存器4	0x026C	32	0xFFFF FFFF
INTC_INTEN5	中断事件使能寄存器5	0x0270	32	0xFFFF FFFF
INTC_INTEN6	中断事件使能寄存器6	0x0274	32	0xFFFF FFFF
INTC_INTEN7	中断事件使能寄存器7	0x0278	32	0xFFFF FFFF
INTC_INTEN8	中断事件使能寄存器8	0x027C	32	0xFFFF FFFF
INTC_INTEN9	中断事件使能寄存器9	0x0280	32	0xFFFF FFFF
INTC_INTEN10	中断事件使能寄存器10	0x0284	32	0xFFFF FFFF
INTC_INTEN11	中断事件使能寄存器11	0x0288	32	0xFFFF FFFF
INTC_INTEN12	中断事件使能寄存器12	0x028C	32	0xFFFF FFFF
INTC_INTEN13	中断事件使能寄存器13	0x0290	32	0xFFFF FFFF
INTC_INTEN14	中断事件使能寄存器14	0x0294	32	0xFFFF FFFF
INTC_INTEN15	中断事件使能寄存器15	0x0298	32	0xFFFF FFFF
INTC_SWIER	软件中断寄存器	0x029C	32	0x0000 0000
INTC_EVTER	事件使能寄存器	0x02A0	32	0x0000 0000
INTC_IER	中断请求使能寄存器	0x02A4	32	0xFFFF FFFF
INTC_FPUER	中断请求使能寄存器	0x02A8	32	0x0000 0000

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	
Res																
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Res				WDT EN	BUS ERRE N	REC CERR EN	RPAR ERRE N	Res			XTAL STPE N	Res	PVD 2EN	PVD 1EN	SWD TEN	Res

位/位域	标记	位名	功能	读写
b31~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
b8	RPARERREN	Cache RAM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	RW
b7~b6	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b4	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
b0	Res	保留位	读出时为“0”，写入时写“0”	RW

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	
Res																
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Res				WDT F	BUS ERRF	REC CERR F	RPAR ERRF	Res			XTAL STPF	Res	PVD 2F	PVD 1F	SWD TF	Res

位/位域	标记	位名	功能	读写
b31~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
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	Res	保留位	读出时为“0”，写入时写“0”	RW
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	Res	保留位	读出时为“0”，写入时写“0”	RW
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	Res	保留位	读出时为“0”，写入时写“0”	RW

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	
Res																
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Res				WDT FCLR	BUSE RRFC LR	RECC ERRF CLR	RPAR ERRF CLR	Res			XTAL STPF CLR	Res	PVD2 FCLR	PVD1 FCLR	SWD TFCL R	Res

位/位域	标记	位名	功能	读写
b31~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW
b7~b6	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b4	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
b0	Res	保留位	读出时为“0”，写入时写“0”	RW

注:

本寄存器读出时为 0。

10.5.4 外部中断控制寄存器 (INTC_EIRQCRx, x=0~15)

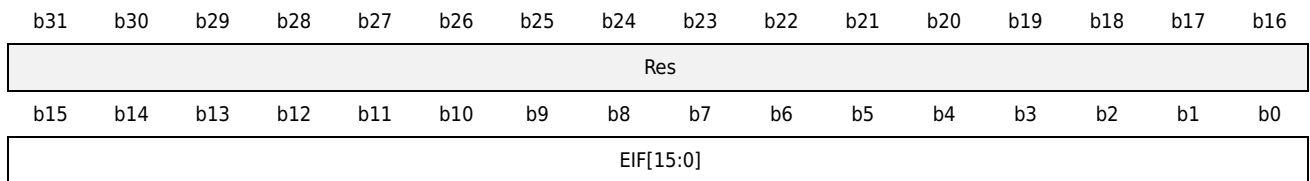
复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
NOC EN	Res	NOCSEL[1:0]		Res				EFEN	Res	EISMPCLK [1:0]		Res		EIRQTRG[1:0]	

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”，写入时写“0”	RW
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。	RW
b14	Res	保留位	读出时为“0”，写入时写“0”	RW
b13~b12	NOCSEL[1:0]	模拟滤波滤波宽度选择	00: 滤波宽度档位1 01: 滤波宽度档位2 10: 滤波宽度档位3 11: 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。	RW
b11~b8	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b6	Res	保留位	读出时为“0”，写入时写“0”	RW
b5~b4	EISMPCLK[1:0]	数字滤波采样周期选择	00: 1*PCLK3周期 01: 8*PCLK3周期 10: 32*PCLK3周期 11: 64*PCLK3周期	RW
b3~b2	Res	保留位	读出时为“0”，写入时写“0”	RW
b1~b0	EIRQTRG[1:0]	触发选择	00: 下降沿 01: 上升沿 10: 双边沿 11: 低电平	RW

10.5.5 外部中断标志寄存器 (INTC_EIFR)

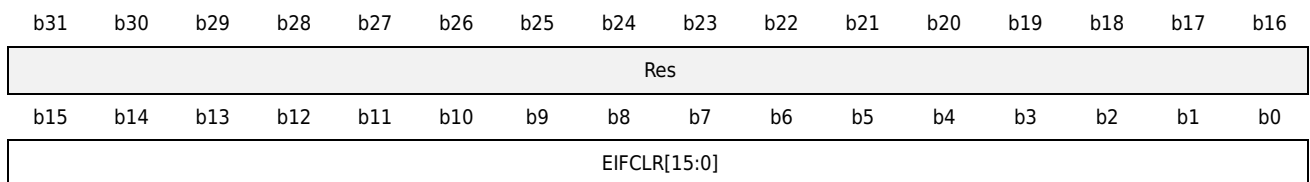
复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”，写入时写“0”	RW
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)

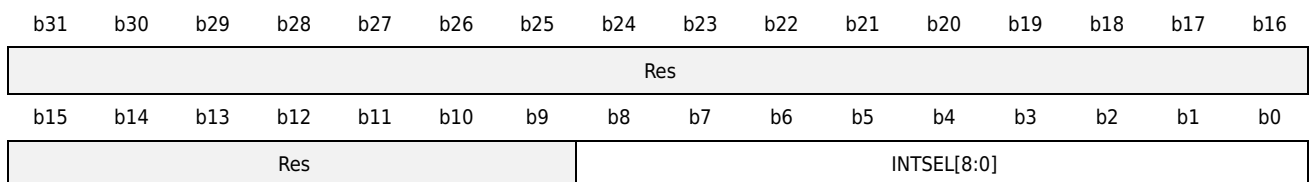
复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”，写入时写“0”	RW
b15~b0	EIFCLR	EIRQ标志清除位	0: 写“0”无效 1: 写“1”清除EIFR寄存器	RW

10.5.7 中断事件选择寄存器 (INTC_INTSEL0~15)

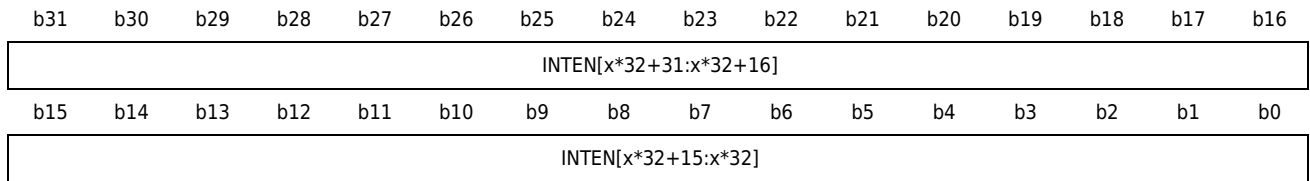
复位值: 0x0000 01FF



位/位域	标记	位名	功能	读写
b31~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
b8~b0	INTSEL[8:0]	中断事件请求选择	0x000~0x1FE: 【中断事件表】所对应的事件	RW

10.5.8 中断事件使能寄存器 (INTC_INTEN0~15)

复位值: 0xFFFF FFFF



位/位域	标记	位名	功能	读写
b31~b0	INTEN [x*32+31:x*32] x=0~15	中断事件使能	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”。	RW

10.5.9 停止模式唤醒事件使能寄存器 (INTC_WKEN)

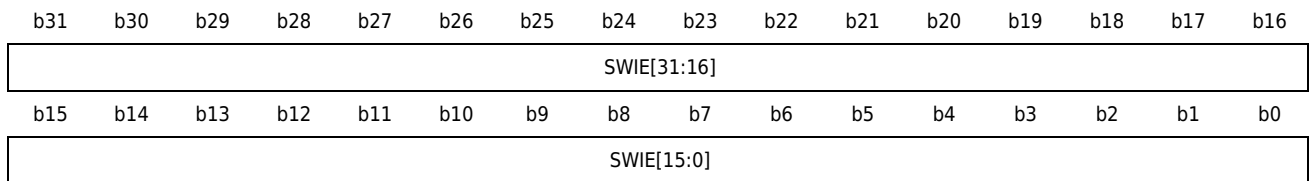
复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	CMP3WKEN	CMP2WKEN	Res	Res	RXWKEN	Res	Res	TMR0CMPWKEN	RTCPRDWKEN	RTCALMWKEN	WKTMWKEN	CMP1WKEN	Res	Res	SWDTWKEN
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIRQWKEN[15:0]															

位/位域	标记	位名	功能	读写
b31	Res	保留位	读出时为“0”，写入时写“0”	RW
b30	CMP3WKEN	CMP3停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	RW
b29	CMP2WKEN	CMP2停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	RW
b28~b27	Res	保留位	读出时为“0”，写入时写“0”	RW
b26	RXWKEN	USART_1 RX线停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	RW
b25~b24	Res	保留位	读出时为“0”，写入时写“0”	RW
b23	TMR0CMPWKEN	TMR0_1_CMPA停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	RW
b22	RTCPRDWKEN	RTC_PRD停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	RW
b21	RTCALMWKEN	RTC_ALM停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	RW
b20	WKTMWKEN	WKT_CMP停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	RW
b19	CMP1WKEN	CMP1停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	RW
b18~b17	Res	保留位	读出时为“0”，写入时写“0”	RW
b16	SWDTWKEN	SWDT_REFUDF停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	RW
b15~b0	EIRQWKEN[15:0]	EIRQ停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	RW

10.5.10 软件中断寄存器 (INTC_SWIER)

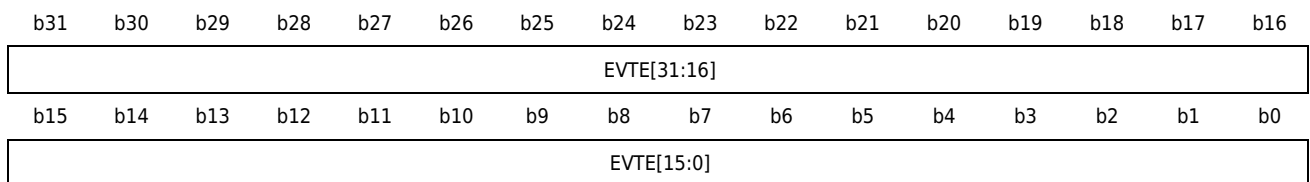
复位值: 0x0000 0000



位/位域	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 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。	RW

10.5.11 事件使能寄存器 (INTC_EVTER)

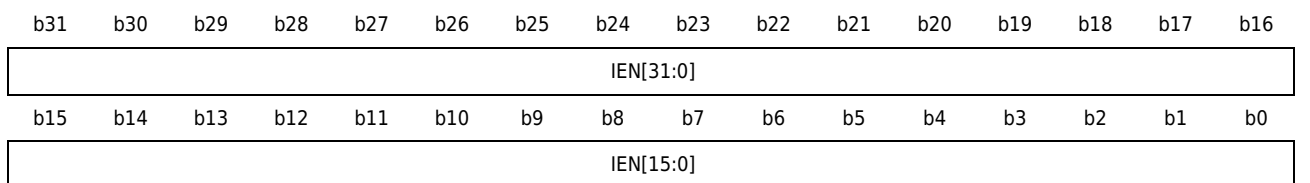
复位值: 0x0000 0000



位/位域	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	RW

10.5.12 中断请求使能寄存器 (INTC_IER)

复位值: 0xFFFF FFFF



位/位域	标记	位名	功能	读写
b31~b0	IEN	中断请求使能	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	RW

10.5.13 FPU 中断使能寄存器 (INTC_FPUIER)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										IDCI EN	IXCI EN	UFCI EN	OFCI EN	DZC IEN	IOCI EN

位/位域	标记	位名	功能	读写
b31~b6	Res	保留位	读出时为“0”，写入时写“0”	RW
b5	IDCIEN	FPU IDCI中断使能位	0: FPU IDCI中断禁止 1: FPU IDCI中断允许	RW
b4	IXCIEN	FPU IXCI中断使能位	0: FPU IXCI中断禁止 1: FPU IXCI中断允许	RW
b3	UFCIEN	FPU UFCI中断使能位	0: FPU UFCI中断禁止 1: FPU UFCI中断允许	RW
b2	OFCIEN	FPU OFCI中断使能位	0: FPU OFCI中断禁止 1: FPU OFCI中断允许	RW
b1	DZCIEN	FPU DZCI中断使能位	0: FPU DZCI中断禁止 1: FPU DZCI中断允许	RW
b0	IOCIEN	FPU IOCI中断使能位	0: FPU IOCI中断禁止 1: FPU IOCI中断允许	RW

10.6 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 概述

自动运行系统（Automatic Operation System）用于在不借助 CPU 的情况下实现外设之间的联动。利用外设产生的事件作为 AOS 源（AOS Source），如定时器的比较匹配、定时器的计数溢出、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 的转换结束等事件，来触发其他外设动作。被触发的外设动作称为 AOS 目标（AOS Target）。

11.1.1 功能概览

AOS 有多个触发源，详见【AOS 源事件】章节。除特殊限制外，每个 AOS 目标可选择其中一个或多个作为触发源。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，最多可以选择 3 个触发源，其中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：

- 9 个 DMA 触发目标
- 4 个高级控制定时器（Timer6）触发目标
- 4 个高精度 PWM（HRPWM）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
- 4 个通用定时器 A（TimerA）触发目标
- 2 个 Event Port 触发目标
- 3 组每组 2 个 ADC 触发目标，用于 ADC1~ADC3 序列触发

11.2 功能说明

11.2.1 AOS 框图

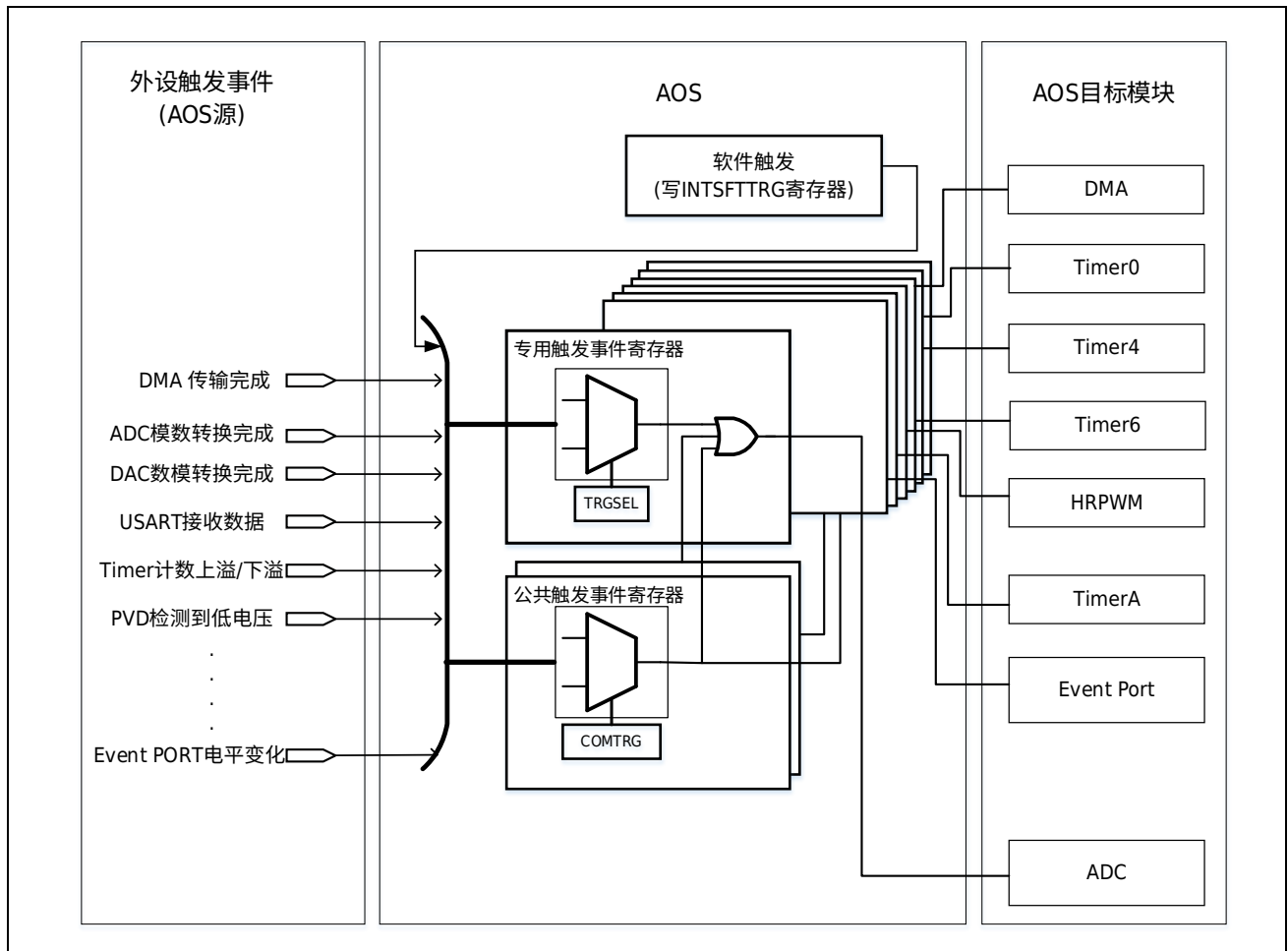


图 11-1 AOS 模块框图

11.2.2 AOS 源事件和目标

AOS 源事件编号见【中断事件表】，表中“触发源”列中打“√”的事件可以用作 AOS 源。AOS 目标如表 11-1 所示。

表 11-1 AOS 目标列表

模块	序号	动作
DMA	0	触发通道 0 开始传输
	1	触发通道 1 开始传输
	2	触发通道 2 开始传输
	3	触发通道 3 开始传输
	4	触发通道 4 开始传输
	5	触发通道 5 开始传输
	6	触发通道 6 开始传输
	7	触发通道 7 开始传输
DMA	8	触发事件触发通道重置
Timer6	9	触发计数器启动/停止/清零/刷新/捕获/递增/递减
	10	触发计数器启动/停止/清零/刷新/捕获/递增/递减
	11	触发计数器启动/停止/清零/刷新/捕获/递增/递减
	12	触发计数器启动/停止/清零/刷新/捕获/递增/递减
HRPWM	13	触发计数器启动/清零/捕获
	14	触发计数器启动/清零/捕获
	15	触发计数器启动/清零/捕获
	16	触发计数器启动/清零/捕获
Timer4	17	触发计数器启动/停止/清零/捕获
Event Port	18	触发 Event Port1/2 动作
	19	触发 Event Port3/4 动作
Timer0	20	触发计数器启动/停止/清零/捕获
TimerA	21	触发计数器单元 1/5 启动/停止/清零/递增/递减
		触发计数器单元 2 输入捕获
	22	触发计数器单元 2 启动/停止/清零/递增/递减
		触发计数器单元 1/5 输入捕获
	23	触发计数器单元 3 启动/停止/清零/递增/递减
		触发计数器单元 4 输入捕获
	24	触发计数器单元 4 启动/停止/清零/递增/递减
		触发计数器单元 3 输入捕获
ADC_1	25, 26	触发序列 A/B 开始模数转换
ADC_2	27, 28	触发序列 A/B 开始模数转换
ADC_3	29, 30	触发序列 A/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）。用于实现同一触发源触发一个或多个 AOS 目标的功能，也可以实现多个 AOS 触发源触发同一个 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 寄存器描述

表 11-2 AOS 基地址

名称	基地址	描述
AOS	0x4001 0800	AOS 基地址

表 11-3 AOS 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
INTSFTRG	外设触发事件寄存器	0x00	32	0x0000 0000
DMA_TRGSELx (x=0~7)	DMA 传输启动触发源选择寄存器	0x04+0x04*x	32	0x0000 01FF
DMA_RC_TRGSEL	DMA 通道重置触发源选择寄存器	0x24	32	0x0000 01FF
TMR6_TRGSELx (x=0~3)	Timer6 硬件触发事件选择寄存器	0x28+0x04*x	32	0x0000 01FF
HRPWM_TRGSELx (x=0~3)	HRPWM 硬件触发事件选择寄存器	0x38+0x04*x	32	0x0000 01FF
TMR4_TRGSEL	Timer4 硬件触发事件选择寄存器	0x48	32	0x0000 01FF
PEVNT_TRGSEL12	Event Port1/2 触发源选择寄存器	0x4C	32	0x0000 01FF
PEVNT_TRGSEL34	Event Port3/4 触发源选择寄存器	0x50	32	0x0000 01FF
TMR0_TRGSEL	Timer0 硬件触发事件选择寄存器	0x54	32	0x0000 01FF
TMRA_TRGSELx (x=0~3)	TimerA 硬件触发事件选择寄存器	0x58+0x04*x	32	0x0000 01FF
ADC1_TRGSELx (x=0~1)	ADC1 转换开始片内触发源选择寄存器	0x68+0x04*x	32	0x0000 01FF
ADC2_TRGSELx (x=0~1)	ADC2 转换开始片内触发源选择寄存器	0x70+0x04*x	32	0x0000 01FF
ADC3_TRGSELx (x=0~1)	ADC3 转换开始片内触发源选择寄存器	0x78+0x04*x	32	0x0000 01FF
AOS_COMTRGSELx (x=1~2)	公共触发源选择寄存器	0x80+0x04*x	32	0x0000 01FF

11.3.1 外设触发事件寄存器 (INTSFTRG)

该寄存器每次写 1 将产生一次触发事件。

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
b0	STRG	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

11.3.2 DMA 传输启动触发源选择寄存器 (DMA_TRGSELx, x=0~7)

在 DMA 使能后，请用该寄存器来选择触发事件源。当编号对应的外设事件发生时，DMA 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。

复位值: 0x0000 01FF

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

位/位域	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	RW
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	RW
b29~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

11.3.3 DMA 通道重置触发源选择寄存器 (DMA_RC_TRGSEL)

在 DMA 使能后，请用该寄存器来选择触发事件源，将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。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。

复位值：0x0000 01FF

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

位/位域	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	RW
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	RW
b29~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

11.3.4 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，请用该寄存器来选择触发事件源，将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。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。

复位值：0x0000 01FF

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

位/位域	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	RW
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	RW
b29~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

11.3.5 HRPWM 硬件触发事件选择寄存器 (HRPWM_TRGSELx, x=0~3)

在 HRPWM 选择内部触发硬件同步后，请用该寄存器来选择触发事件源，将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。当编号对应的外设电路事件发生时，HRPWM 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 HRPWM 单元共用这 4 个寄存器。

复位值：0x0000 01FF

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

位/位域	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	RW
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	RW
b29~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

11.3.6 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，请用该寄存器来选择触发事件源，将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。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。

复位值：0x0000 01FF

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

位/位域	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	RW
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	RW
b29~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

11.3.7 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，请用该寄存器来选择触发事件源。将触发事件的编号写入该寄存器的 TRGSEL[8:0] 位中。Event Port12 共用 PEVNT_TRGSEL12 寄存器，EventPort34 共用 PEVNT_TRGSEL34 寄存器。

复位值：0x0000 01FF

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

位/位域	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	RW
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	RW
b29~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

11.3.8 Timer0 硬件触发事件选择寄存器 (TMR0_TRGSEL)

在 Timer0 选择硬件触发后，请用该寄存器来选择触发事件源，将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。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。

复位值：0x0000 01FF

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

位/位域	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	RW
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	RW
b29~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

11.3.9 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，请用该寄存器来选择触发事件源，将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。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。

复位值：0x0000 01FF

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

位/位域	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	RW
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	RW
b29~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

11.3.10 ADC1 转换开始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，请用该寄存器来选择触发事件源，将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。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 的序列 A 和序列 B 共用这 2 个寄存器。

复位值：0x0000 01FF

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

位/位域	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	RW
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	RW
b29~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

11.3.11 ACD2 转换开始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，请用该寄存器来选择触发事件源，将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。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 序列 A 和序列 B 共用这 2 个寄存器。

复位值：0x0000 01FF

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

位/位域	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	RW
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	RW
b29~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

11.3.12 ADC3 转换开始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，请用该寄存器来选择触发事件源，将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。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 的序列 A 和序列 B 共用这 2 个寄存器。

复位值：0x0000 01FF

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

位/位域	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	RW
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	RW
b29~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

11.3.13 公共触发源选择寄存器 (AOS_COMTRGSELx, x=1~2)

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。

复位值：0x0000 01FF

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

位/位域	标记	位名	功能	读写
b31~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

12 存储保护单元 (MPU)

12.1 概述

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，1 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 MPU: SMPU1	系统 DMA 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
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。

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 提供系统 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 功能说明

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地址和区域大小，可设的范围为 32Bytes~4GBytes，大小必须为 2^n Bytes ($n=5\sim32$)，对应的基地址低 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 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 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。

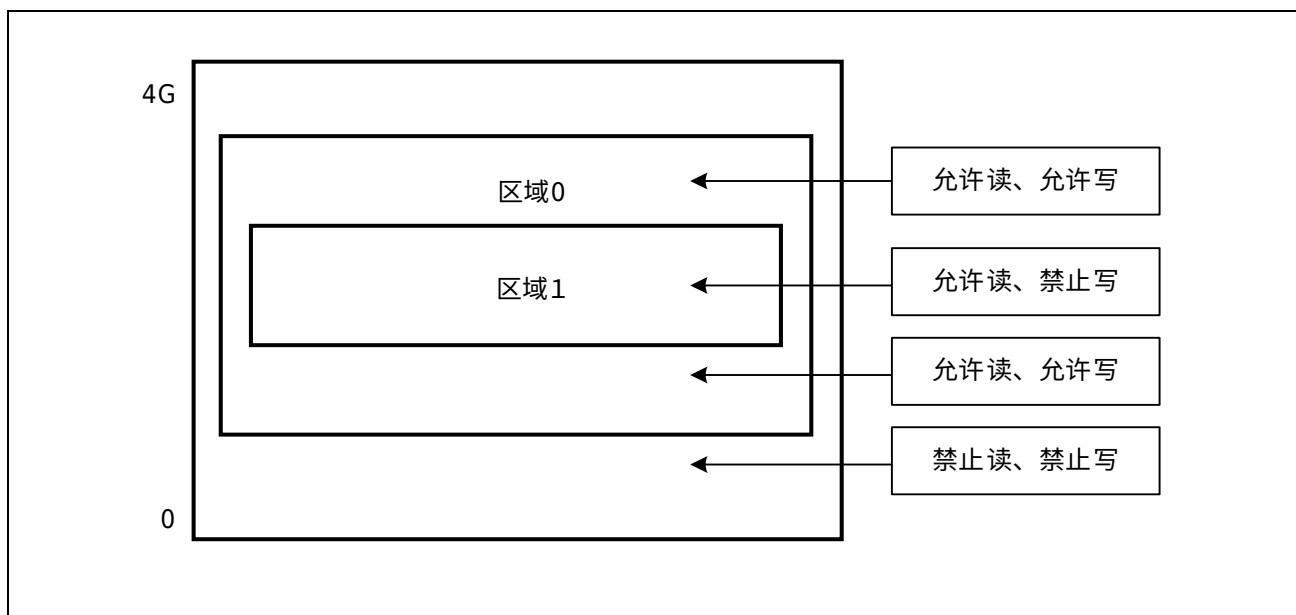


图 12-1 只允许部分空间访问示意图

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。

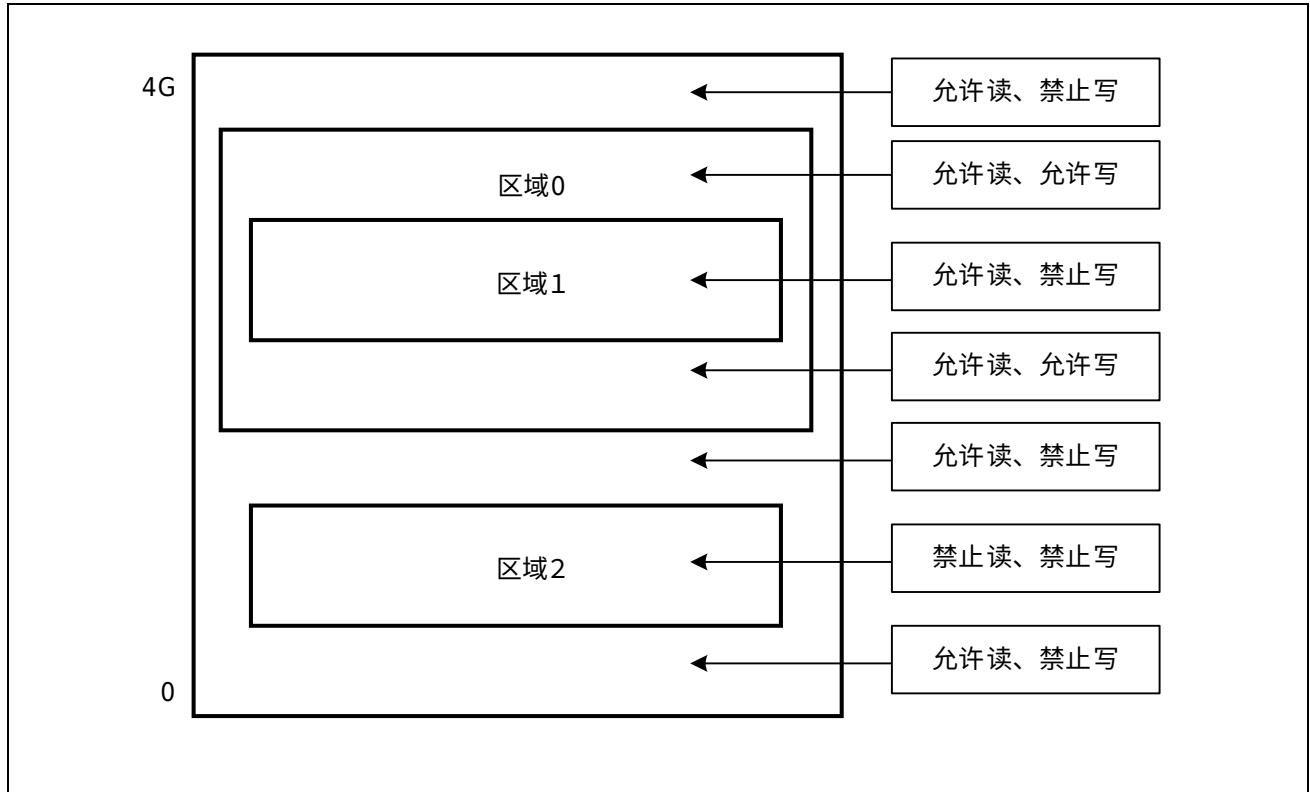


图 12-2 只禁止部分空间访问示意图

12.4 寄存器描述

本模块的寄存器只能通过 CPU 来设置。

表 12-1 MPU 基地址

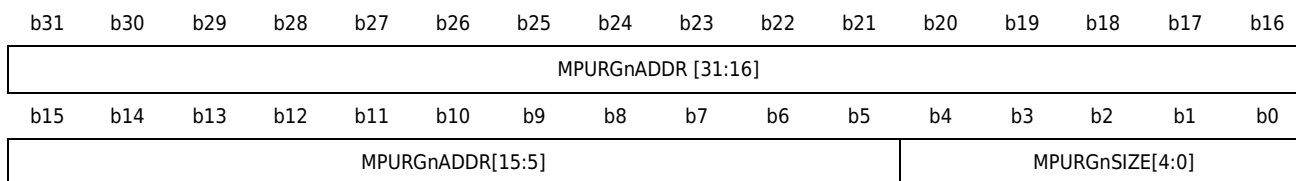
名称	基地址	描述
MPU	0x4005 0000	MPU 基地址

表 12-2 MPU 寄存器列表

寄存器	描述	偏移地址	位宽	复位值	写保护
MPU_RGDO~15	区域 0~15 范围描述寄存器	0x00~0x3C	32	0x0000 0000	MPUWE
MPU_SR	状态标志寄存器	0x40	32	0x0000 0000	无
MPU_ECLR	标志清除寄存器	0x44	32	0x0000 0000	无
MPU_WP	写保护寄存器	0x48	32	0x0000 0000	WKEY
MPU_IPPR	IP 访问保护寄存器	0x4C	32	0x0000 0000	MPUWE
MPU_MSPPBA	MSP 保护开始地址寄存器	0x50	32	0x0000 0000	MPUWE
MPU_MSPPCTL	MSP 保护控制寄存器	0x54	32	0x0000 0003	MPUWE
MPU_PSPPBA	PSP 保护开始地址寄存器	0x58	32	0x0000 0000	MPUWE
MPU_PSPPCTL	PSP 保护控制寄存器	0x5C	32	0x0000 0003	MPUWE
MPU_S1RGE	SMPU1 区域使能寄存器	0x60	32	0x0000 0000	MPUWE
MPU_S1RGWP	SMPU1 区域写权限寄存器	0x64	32	0x0000 0000	MPUWE
MPU_S1RGRP	SMPU1 区域读权限寄存器	0x68	32	0x0000 0000	MPUWE
MPU_S1CR	SMPU1 控制寄存器	0x6C	32	0x0000 0000	MPUWE

12.4.1 区域范围描述寄存器 (MPU_RGDn, n=0~15)

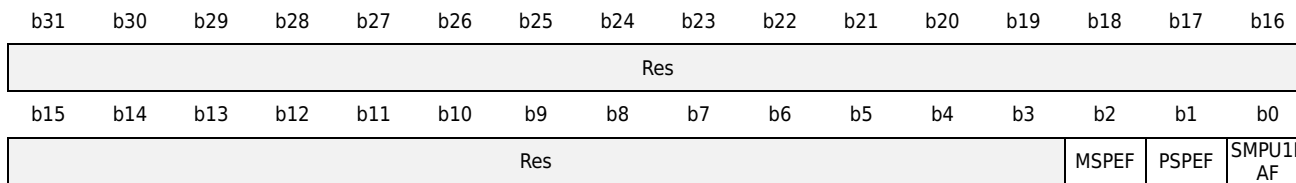
复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b5	MPURGnADDR[31:5]	区域基地址	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	RW
b4~b0	MPURGnSIZE[4:0]	区域大小	设定区域 n 的大小 00000~00011: 保留, 禁止设定 00100: 32Bytes 00101: 64Bytes ... 11110: 2GBytes 11111: 4GBytes	RW

12.4.2 状态标志寄存器 (MPU_SR)

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b3	Res	保留位	保留位, 读数为 0, 写入时写 0	RW
b2	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
b1	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
b0	SMPU1EAF	SMPU1 错误标志	0: 系统 DMA 未发生错误访问 1: 系统 DMA 发生了错误访问	R

注:

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: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													MSP ECLR	PSP ECLR	SMPU1E CLR

位/位域	标记	位名	功能	读写
b31~b3	Res	保留位	保留位, 读出为 0, 写入时写 0	RW
b2	MSPECLR	MSP 错误标志清除	写入 1 可以将 MSPEF 清 0	RW
b1	PSPECLR	PSP 错误标志清除	写入 1 可以将 PSPEF 清 0	RW
b0	SMPU1ECLR	SMPU1 错误标志清除	写入 1 可以将 SMPU1EAF 清 0	RW

注:

本寄存器读出值固定为 0x0000 0000。

12.4.4 写保护寄存器 (MPU_WP)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
WKEY[15:1]															MPUWE

位/位域	标记	位名	功能	读写
b31~b6	Res	保留位	保留位, 读出为 0, 写入时写 0	RW
b15~b1	WKEY[15:1]	写入码	在写入 MPUWE 时, 必须同时向 WKEY 写入 0x96A5, 读出为 0	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

注:

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: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
BUSERRE	Res	MSTPWRP	MSTPRDP	SYSCWRP	SYSCRD	INTCW	INTCRD	SRAMCWRP	SRAMCRDP	DMPUWRP	DMPURDP	RTCWRP	RTC	BKSRAMWRP	BKSRAMRDP
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SWDTWRP	SWDTRDP	WDTWRP	WDTRDP	Res	EFMWRP	EFMRDP	CRCWRP	CRCRDP	Res						

位/位域	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	RW
b30	Res	保留位	保留位, 读出为 0, 写入时写 0	RW
b29	MSTPWRP	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	RW
b28	MSTPRDP	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	RW
b27	SYSCWRP	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	RW
b26	SYSCRD	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	RW
b25	INTCW	INTC 写保护	0: 允许对 INTC 的写操作 1: 禁止对 INTC 的写操作	RW
b24	INTCRDP	INTC 读保护	0: 允许对 INTC 的读操作 1: 禁止对 INTC 的读操作	RW
b23	SRAMCWRP	SRAMC 写保护	0: 允许对 SRAMC 的写操作 1: 禁止对 SRAMC 的写操作	RW
b22	SRAMCRDP	SRAMC 读保护	0: 允许对 SRAMC 的读操作 1: 禁止对 SRAMC 的读操作	RW
b21	DMPUWRP	DMPU 写保护	0: 允许对 DMPU 的写操作 1: 禁止对 DMPU 的写操作	RW
b20	DMPURDP	DMPU 读保护	0: 允许对 DMPU 的读操作 1: 禁止对 DMPU 的读操作	RW
b19	RTCWRP	RTC 写保护	0: 允许对 RTC 的写操作 1: 禁止对 RTC 的写操作	RW
b18	RTC	RTC 读保护	0: 允许对 RTC 的读操作 1: 禁止对 RTC 的读操作	RW
b17	BKSRAMWRP	BKSRAM 写保护	0: 允许对 Ret SRAM 的写操作 1: 禁止对 Ret SRAM 的写操作	RW
b16	BKSRAMRDP	BKSRAM 读保护	0: 允许对 Ret SRAM 的读操作 1: 禁止对 Ret SRAM 的读操作	RW
b15	SWDTWRP	SWDT 写保护	0: 允许对 SWDT 的写操作 1: 禁止对 SWDT 的写操作	RW

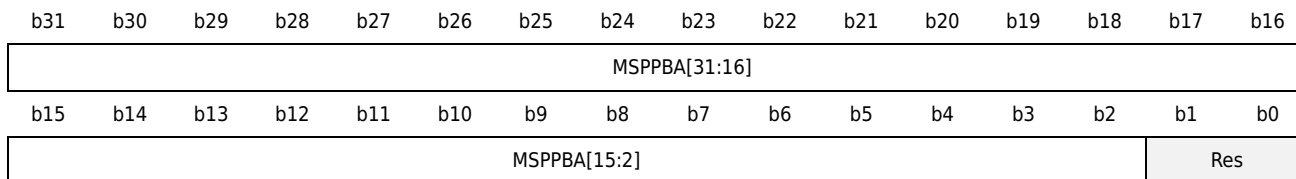
b14	SWDTRDP	SWDT 读保护	0: 允许对 SWDT 的读操作 1: 禁止对 SWDT 的读操作	RW
b13	WDTWRP	WDT 写保护	0: 允许对 WDT 的写操作 1: 禁止对 WDT 的写操作	RW
b12	WDTRDP	WDT 读保护	0: 允许对 WDT 的读操作 1: 禁止对 WDT 的读操作	RW
b11-b10	Res	保留位	保留位, 读出为 0, 写入时写 0	RW
b9	EFMWRP	EFM 写保护	0: 允许对 EFM 的写操作 1: 禁止对 EFM 的写操作	RW
b8	EFMRDP	EFM 读保护	0: 允许对 EFM 的读操作 1: 禁止对 EFM 的读操作	RW
b7	CRCWRP	CRC 写保护	0: 允许对 CRC 的写操作 1: 禁止对 CRC 的写操作	RW
b6	CRCRDP	CRC 读保护	0: 允许对 CRC 的读操作 1: 禁止对 CRC 的读操作	RW
b5~b0	Res	保留位	保留位, 读出为 0, 写入时写 0	RW

注:

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_MSPPPBA)

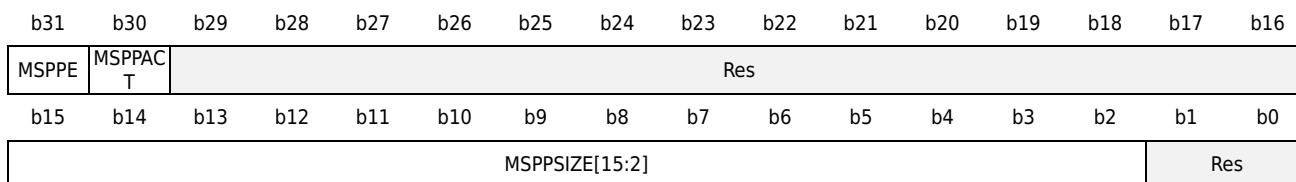
复位值: 0x0000 0000



位/位域	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	RW
b1~b0	Res	保留位	保留位, 读出为 0, 写入时写 0	RW

12.4.7 MSP 保护控制寄存器 (MPU_MSPPPCTL)

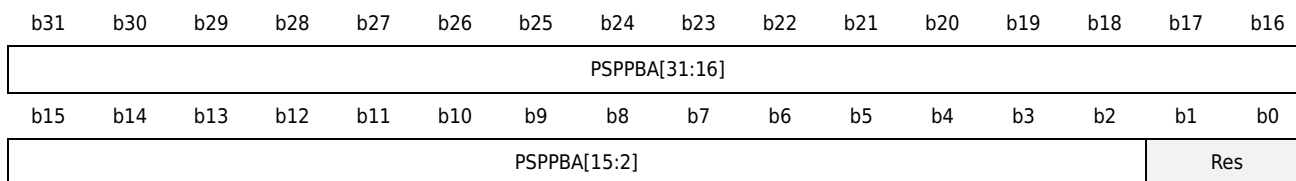
复位值: 0x0000 0003



位/位域	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	RW
b30	MSPPPACT	MSP 保护动作选择	0: 不可屏蔽中断 1: 复位	
b29~b16	Res	保留位	保留位, 读出为 0, 写入时写 0	RW
b15~b2	MSPPPSIZE[15:2]	允许范围大小	设定 MSP 指针允许范围, 设定值 4Bytes~64KBytes 当 MSPPPSIZE=0x0 时, MSP 指针允许范围为 4Bytes, 当 MSPPPSIZE=0x3FFF 时, MSP 指针允许范围为 64KBytes	RW
b1~b0	Res	保留位	保留位, 读出为 1, 写入时写 1	RW

12.4.8 PSP 保护开始地址寄存器 (MPU_PSPBA)

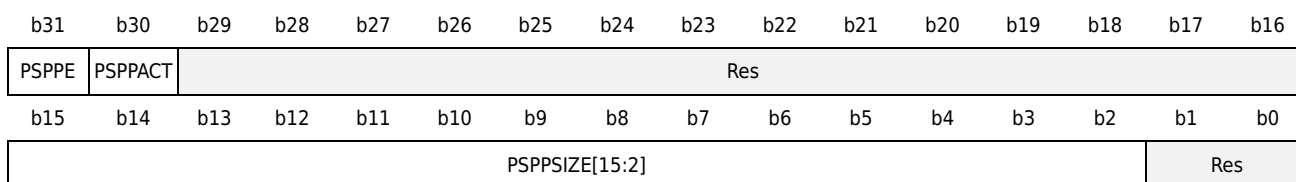
复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b2	PSPBA[31:2]	允许范围开始地址	设定 PSP 指针允许范围开始地址	RW
b1~b0	Res	保留位	保留位, 读出为 0, 写入时写 0	RW

12.4.9 PSP 保护控制寄存器 (MPU_PSPCTL)

复位值: 0x0000 0003



位/位域	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	RW
b30	PSPPACT	PSP 保护动作选择	0: 不可屏蔽中断 1: 复位	-
b29~b16	Res	保留位	保留位, 读出为 0, 写入时写 0	RW
b15~b2	PSPSIZE[15:2]	允许范围大小	设定 PSP 指针允许范围, 设定值 4Bytes~64KBytes 当 PSPSIZE=0x0 时, PSP 指针允许范围为 4Bytes, 当 PSPSIZE=0x3FFF 时, PSP 指针允许范围为 64KBytes	RW
b1~b0	Res	保留位	保留位, 读出为 1, 写入时写 1	RW

12.4.10 SMPU1 区域使能寄存器 (MPU_S1RGE)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
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

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	保留位, 读出为 0, 写入时写 0	RW
b15	S1RG15E	SMPU1 区域 15 使能	0: SMPU1 的区域 15 无效 1: SMPU1 的区域 15 有效	RW
b14	S1RG14E	SMPU1 区域 14 使能	0: SMPU1 的区域 14 无效 1: SMPU1 的区域 14 有效	RW
b13	S1RG13E	SMPU1 区域 13 使能	0: SMPU1 的区域 13 无效 1: SMPU1 的区域 13 有效	RW
b12	S1RG12E	SMPU1 区域 12 使能	0: SMPU1 的区域 12 无效 1: SMPU1 的区域 12 有效	RW
b11	S1RG11E	SMPU1 区域 11 使能	0: SMPU1 的区域 11 无效 1: SMPU1 的区域 11 有效	RW
b10	S1RG10E	SMPU1 区域 10 使能	0: SMPU1 的区域 10 无效 1: SMPU1 的区域 10 有效	RW
b9	S1RG9E	SMPU1 区域 9 使能	0: SMPU1 的区域 9 无效 1: SMPU1 的区域 9 有效	RW
b8	S1RG8E	SMPU1 区域 8 使能	0: SMPU1 的区域 8 无效 1: SMPU1 的区域 8 有效	RW
b7	S1RG7E	SMPU1 区域 7 使能	0: SMPU1 的区域 7 无效 1: SMPU1 的区域 7 有效	RW
b6	S1RG6E	SMPU1 区域 6 使能	0: SMPU1 的区域 6 无效 1: SMPU1 的区域 6 有效	RW
b5	S1RG5E	SMPU1 区域 5 使能	0: SMPU1 的区域 5 无效 1: SMPU1 的区域 5 有效	RW
b4	S1RG4E	SMPU1 区域 4 使能	0: SMPU1 的区域 4 无效 1: SMPU1 的区域 4 有效	RW
b3	S1RG3E	SMPU1 区域 3 使能	0: SMPU1 的区域 3 无效 1: SMPU1 的区域 3 有效	RW
b2	S1RG2E	SMPU1 区域 2 使能	0: SMPU1 的区域 2 无效 1: SMPU1 的区域 2 有效	RW
b1	S1RG1E	SMPU1 区域 1 使能	0: SMPU1 的区域 1 无效 1: SMPU1 的区域 1 有效	RW
b0	S1RG0E	SMPU1 区域 0 使能	0: SMPU1 的区域 0 无效 1: SMPU1 的区域 0 有效	RW

12.4.11 SMPU1 区域写权限寄存器 (MPU_S1RGWP)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
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/位域	标记	位名	功能	读写
b31~b16	Res	保留位	保留位, 读出为 0, 写入时写 0	RW
b15	S1RG15WP	SMPU1 区域 15 写权限	0: SMPU1 区域 15 允许系统 DMA 写 1: SMPU1 区域 15 禁止系统 DMA 写	RW
b14	S1RG14WP	SMPU1 区域 14 写权限	0: SMPU1 区域 14 允许系统 DMA 写 1: SMPU1 区域 14 禁止系统 DMA 写	RW
b13	S1RG13WP	SMPU1 区域 13 写权限	0: SMPU1 区域 13 允许系统 DMA 写 1: SMPU1 区域 13 禁止系统 DMA 写	RW
b12	S1RG12WP	SMPU1 区域 12 写权限	0: SMPU1 区域 12 允许系统 DMA 写 1: SMPU1 区域 12 禁止系统 DMA 写	RW
b11	S1RG11WP	SMPU1 区域 11 写权限	0: SMPU1 区域 11 允许系统 DMA 写 1: SMPU1 区域 11 禁止系统 DMA 写	RW
b10	S1RG10WP	SMPU1 区域 10 写权限	0: SMPU1 区域 10 允许系统 DMA 写 1: SMPU1 区域 10 禁止系统 DMA 写	RW
b9	S1RG9WP	SMPU1 区域 9 写权限	0: SMPU1 区域 9 允许系统 DMA 写 1: SMPU1 区域 9 禁止系统 DMA 写	RW
b8	S1RG8WP	SMPU1 区域 8 写权限	0: SMPU1 区域 8 允许系统 DMA 写 1: SMPU1 区域 8 禁止系统 DMA 写	RW
b7	S1RG7WP	SMPU1 区域 7 写权限	0: SMPU1 区域 7 允许系统 DMA 写 1: SMPU1 区域 7 禁止系统 DMA 写	RW
b6	S1RG6WP	SMPU1 区域 6 写权限	0: SMPU1 区域 6 允许系统 DMA 写 1: SMPU1 区域 6 禁止系统 DMA 写	RW
b5	S1RG5WP	SMPU1 区域 5 写权限	0: SMPU1 区域 5 允许系统 DMA 写 1: SMPU1 区域 5 禁止系统 DMA 写	RW
b4	S1RG4WP	SMPU1 区域 4 写权限	0: SMPU1 区域 4 允许系统 DMA 写 1: SMPU1 区域 4 禁止系统 DMA 写	RW
b3	S1RG3WP	SMPU1 区域 3 写权限	0: SMPU1 区域 3 允许系统 DMA 写 1: SMPU1 区域 3 禁止系统 DMA 写	RW
b2	S1RG2WP	SMPU1 区域 2 写权限	0: SMPU1 区域 2 允许系统 DMA 写 1: SMPU1 区域 2 禁止系统 DMA 写	RW
b1	S1RG1WP	SMPU1 区域 1 写权限	0: SMPU1 区域 1 允许系统 DMA 写 1: SMPU1 区域 1 禁止系统 DMA 写	RW
b0	S1RG0WP	SMPU1 区域 0 写权限	0: SMPU1 区域 0 允许系统 DMA 写 1: SMPU1 区域 0 禁止系统 DMA 写	RW

12.4.12 SMPU1 区域读权限寄存器 (MPU_S1RGRP)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
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/位域	标记	位名	功能	读写
b31~b16	Res	保留位	保留位, 读出为 0, 写入时写 0	RW
b15	S1RG15RP	SMPU1 区域 15 读权限	0: SMPU1 区域 15 允许系统 DMA 读 1: SMPU1 区域 15 禁止系统 DMA 读	RW
b14	S1RG14RP	SMPU1 区域 14 读权限	0: SMPU1 区域 14 允许系统 DMA 读 1: SMPU1 区域 14 禁止系统 DMA 读	RW
b13	S1RG13RP	SMPU1 区域 13 读权限	0: SMPU1 区域 13 允许系统 DMA 读 1: SMPU1 区域 13 禁止系统 DMA 读	RW
b12	S1RG12RP	SMPU1 区域 12 读权限	0: SMPU1 区域 12 允许系统 DMA 读 1: SMPU1 区域 12 禁止系统 DMA 读	RW
b11	S1RG11RP	SMPU1 区域 11 读权限	0: SMPU1 区域 11 允许系统 DMA 读 1: SMPU1 区域 11 禁止系统 DMA 读	RW
b10	S1RG10RP	SMPU1 区域 10 读权限	0: SMPU1 区域 10 允许系统 DMA 读 1: SMPU1 区域 10 禁止系统 DMA 读	RW
b9	S1RG9RP	SMPU1 区域 9 读权限	0: SMPU1 区域 9 允许系统 DMA 读 1: SMPU1 区域 9 禁止系统 DMA 读	RW
b8	S1RG8RP	SMPU1 区域 8 读权限	0: SMPU1 区域 8 允许系统 DMA 读 1: SMPU1 区域 8 禁止系统 DMA 读	RW
b7	S1RG7RP	SMPU1 区域 7 读权限	0: SMPU1 区域 7 允许系统 DMA 读 1: SMPU1 区域 7 禁止系统 DMA 读	RW
b6	S1RG6RP	SMPU1 区域 6 读权限	0: SMPU1 区域 6 允许系统 DMA 读 1: SMPU1 区域 6 禁止系统 DMA 读	RW
b5	S1RG5RP	SMPU1 区域 5 读权限	0: SMPU1 区域 5 允许系统 DMA 读 1: SMPU1 区域 5 禁止系统 DMA 读	RW
b4	S1RG4RP	SMPU1 区域 4 读权限	0: SMPU1 区域 4 允许系统 DMA 读 1: SMPU1 区域 4 禁止系统 DMA 读	RW
b3	S1RG3RP	SMPU1 区域 3 读权限	0: SMPU1 区域 3 允许系统 DMA 读 1: SMPU1 区域 3 禁止系统 DMA 读	RW
b2	S1RG2RP	SMPU1 区域 2 读权限	0: SMPU1 区域 2 允许系统 DMA 读 1: SMPU1 区域 2 禁止系统 DMA 读	RW
b1	S1RG1RP	SMPU1 区域 1 读权限	0: SMPU1 区域 1 允许系统 DMA 读 1: SMPU1 区域 1 禁止系统 DMA 读	RW
b0	S1RG0RP	SMPU1 区域 0 读权限	0: SMPU1 区域 0 允许系统 DMA 读 1: SMPU1 区域 0 禁止系统 DMA 读	RW

12.4.13 SMPU1 控制寄存器 (MPU_S1CR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								SMPU1 E	Res			SMPU1ACT[1:0]	SMPU1B WP	SMPU1 BRP	

位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	保留位, 读出为 0, 写入时写 0	RW
b7	SMPU1E	SMPU1 使能	0: SMPU1 无效 1: SMPU1 有效	RW
b6~b4	Res	保留位	保留位, 读出为 0, 写入时写 0	RW
b3~b2	SMPU1ACT[1:0]	SMPU1 动作选择	设定系统 DMA 发生被禁止的访问时的动作 00: 读访问读到 0, 写访问忽略 01: 读访问读到 0, 写访问忽略; 产生总线错误 10: 读访问读到 0, 写访问忽略; 产生不可屏蔽中断 11: 复位	RW
b1	SMPU1BWP	SMPU1 背景写权限设置	0: SMPU1 背景空间允许系统 DMA 写 1: SMPU1 背景空间禁止系统 DMA 写	RW
b0	SMPU1BRP	SMPU1 背景读权限设置	0: SMPU1 背景空间允许系统 DMA 读 1: SMPU1 背景空间禁止系统 DMA 读	RW

注:

多个区域设定重叠时, 优先级为: 设定的禁止>设定的允许

13 内部时钟校准器 (CTC)

13.1 概述

内部时钟校准器 (Clock Trimming Controller, 以下称 CTC) 可以自动校准内部高速振荡器 (HRC)。由于工作环境的影响 HRC 的频率可能会产生偏差, 用 CTC 基于外部高精度参考时钟, 采用硬件方式自动调整 HRC 的频率以得到一个精准的 HRC 时钟。

13.2 主要特性

- 硬件自动校准无需软件参与
- 三个外部参考时钟源: XTAL、XTAL32、CTCREF
- 连续校准和单次校准两种模式
- 用于频率测量并具有重载功能的 16 位校准计数器
- 用于频率校准的 8 位校准目标偏差值和 6 位校准值
- 用于提示校准失败的错误中断

13.3 功能说明

13.3.1 CTC 框图

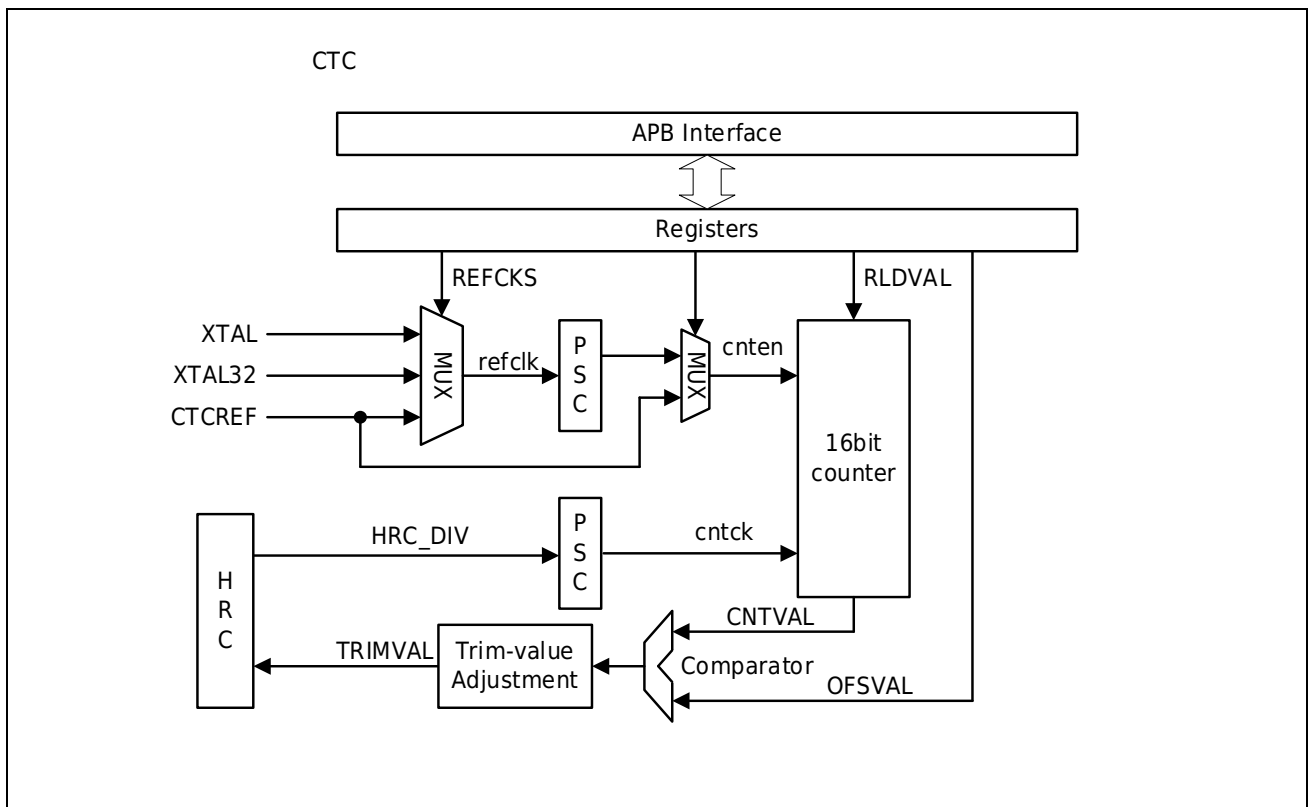


图 13-1 CTC 的基本框图

13.3.2 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 的抖动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 REFPSK[2:0]和 HRCPSK[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_{hrc} 表示 HRC 的目标频率，单位 Hz；

F_{ref} 表示参考时钟的频率，单位 Hz；

PSC_{hrc} 表示 HRC 分频数；

PSC_{ref} 表示参考时钟分频数。

表 13-1、表 13-2 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 13-1 HRC 目标频率为 20MHz 时的测量误差

Fref		1KHz							
PSCref		2	4	8	16	64	256	1024	4096
PSChrc	1	0.167%	不可设	不可设	不可设	不可设	不可设	不可设	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%
Fref		32.768KHz							
PSCref		2	4	8	16	64	256	1024	4096
PSChrc	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%
Fref		100KHz							
PSCref		2	4	8	16	64	256	1024	4096
PSC _{hrc}	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	不推荐	1.069%	0.285%
	2048	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.035%
Fref		1MHz							
PSCref		2	4	8	16	64	256	1024	4096
PSC _{hrc}	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	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	0.734%
	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
PSC _{hrc}	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	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.559%
	128	不推荐	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	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		20MHz							
PSCref		2	4	8	16	64	256	1024	4096
PSC _{hrc}	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	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	0.880%

	8	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.270%
	32	不推荐	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	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	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

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 13-2 HRC 目标频率为 16MHz 时的测量误差

Fref		1KHz							
PSCref		2	4	8	16	64	256	1024	4096
PSC _{hrc}	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
PSC _{hrc}	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
PSC _{hrc}	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
PSC _{hrc}	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
PSC _{hrc}	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
PSC _{hrc}	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	不推荐	0.737%
PSC _{hrc}	32	不推荐	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	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	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		20MHz							
PSCref		2	4	8	16	64	256	1024	4096
PSC _{hrc}	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	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	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	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	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

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 REF_PSC[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。

$$TE = \frac{0.099\%}{\sqrt{F_{hrc} \times T_{trim} \div 10^5}} + \frac{4}{F_{hrc} \times T_{trim} \div PSC_{hrc}}$$

注:

F_{hrc} 表示 HRC 的目标频率, 单位 Hz;

T_{trim} 表示校准周期, 单位 S;

PSC_{hrc} 表示 HRC 分频数。

13.3.3 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。

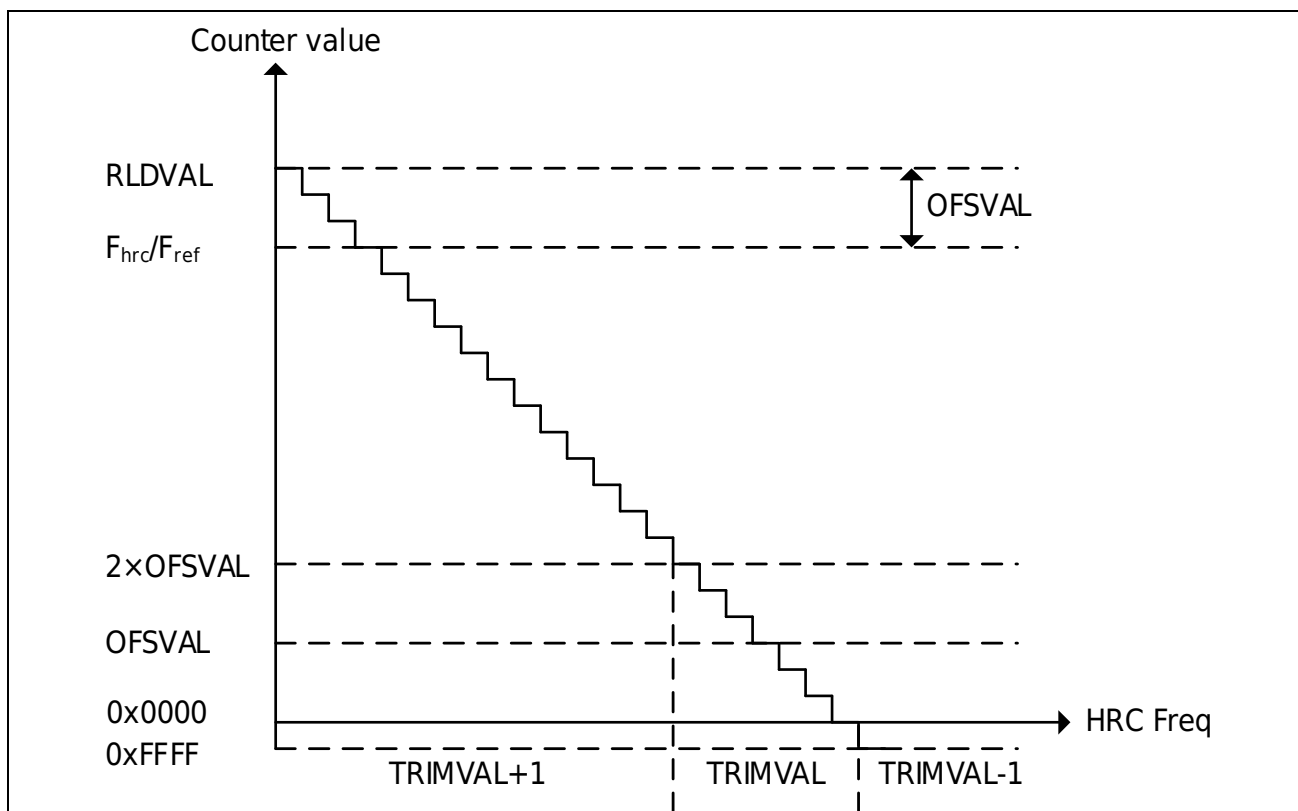


图 13-2 CTC 校准示意图

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 $2 \times OFSVAL$ 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 $2 \times OFSVAL$, 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 > (OFSVAL×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。

13.4 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【中断控制器（INTC）】章节。

13.5 应用举例

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 13-2 可知校准误差为 0.405%，小于 0.5% 的校准目标偏差，方案可行。

1) 设定寄存器。

CTC_CR2 = 0x1EAB 0027，其中

OFSVAL = $(16 \div 0.032768) \times 16 \times 0.5\% \approx 39$ (0x27)；

RLDVAL = $(16 \div 0.032768) \times 16 + 39 \approx 7851$ (0x1EAB)；

2) 设定寄存器。

CTC_CR1 = 0x0000 00E3，其中

TRMVAL = 0x00 (从中间值开始校准)

HRCPSC = 0b000 (选择 HRC 不分频)

CTCEN = 1 (启动 CTC)

ERRIE = 1 (允许错误中断)

REFCKS = 0b10 (选择 XTAL32 作为参考时钟进行连续校准)

REFPSC = 0b011 (选择参考时钟的 8 分频)

3) 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)；

4) 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。

13.6 寄存器描述

表 13-3 CTC 基地址

名称	基地址	描述
CTC	0x4004 9C00	CTC基地址

表 13-4 CTC 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
CTC_CR1	时钟校准控制寄存器1	0x00	32	0x0000 0000
CTC_CR2	时钟校准控制寄存器2	0x04	32	0x0000 0000
CTC_STR	时钟校准状态寄存器	0x08	32	0x0000 0000
CTC_CNT	时钟校准计数器	0x0C	16	0x0000

13.6.1 时钟校准控制寄存器 1 (CTC_CR1)

复位值: 0x0000 0000

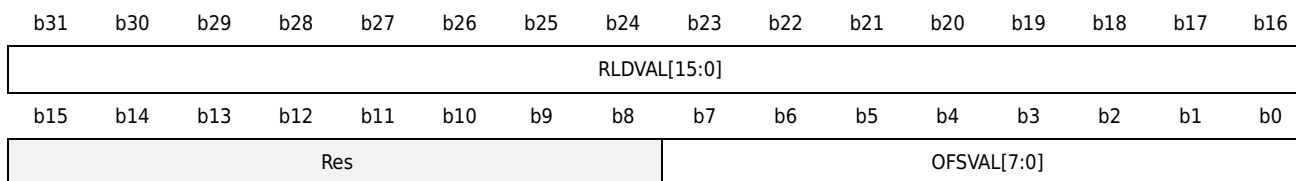
b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res										TRMVAL[5:0]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res		REFEDG[1:0]		Res	HRCPSC[2:0]			CTC EN	ERR IE	REFCKS[1:0]		Res	REFPSC[2:0]		

位/位域	标记	位名	功能	读写
b31~b22	Res	保留位	读出时为“0”，写入时写“0”	R
b21~b16	TRMVAL[5:0]	HRC校准值	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 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。	RW
b15~b14	Res	保留位	读出时为“0”，写入时写“0”	R
b13~b12	REFEDG[1:0]	参考信号边沿选择	用于REFCKS=0b01时选择CTCREF的有效边沿。 00: 选择CTCREF的上升沿到上升沿进行校准 01: 选择CTCREF的上升沿到下降沿进行校准 10: 选择CTCREF的下降沿到上升沿进行校准 11: 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。	RW
b11	Res	保留位	读出时为“0”，写入时写“0”	R
b10~b8	HRCPSC[2:0]	HRC分频数选择	000: 不分频 001: 2分频 010: 4分频 011: 8分频 100: 32分频 101: 128分频 110: 512分频 111: 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。	RW
b7	CTCEN	校准使能	用于启动频率校准。 0: 停止频率校准	RW

			<p>1: 启动频率校准</p> <p>置位条件:</p> <p> 软件置位;</p> <p>清零条件:</p> <p> (1) 软件清零;</p> <p> (2) 校准发生上溢或者下溢时自动清零;</p> <p> (3) 单次校准结束时自动清零;</p>	
b6	ERRIE	错误中断允许	<p>用于选择发生校准上溢或者下溢时是否产生错误中断。</p> <p>0: 禁止发生错误中断</p> <p>1: 允许发生错误中断</p>	RW
b5~b4	REFCKS[1:0]	参考时钟选择	<p>用于为频率校准选择参考时钟源。</p> <p>00: 选择CTCREF作为参考时钟 (连续校准)</p> <p>01: 选择CTCREF作为参考信号 (单次校准)</p> <p>10: 选择XTAL32作为参考时钟 (连续校准)</p> <p>11: 选择XTAL作为参考时钟 (连续校准)</p> <p>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。</p>	RW
b3	Res	保留位	<p>读出时为“0”，写入时写“0”</p>	R
b2~b0	REFPSC[2:0]	参考时钟分频数选择	<p>REFCKS=0b00/0b10/0b11时选择参考时钟的分频数。</p> <p>000: 2分频</p> <p>001: 4分频</p> <p>010: 8分频</p> <p>011: 16分频</p> <p>100: 64分频</p> <p>101: 256分频</p> <p>110: 1024分频</p> <p>111: 4096分频</p> <p>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。</p>	RW

13.6.2 时钟校准控制寄存器 2 (CTC_CR2)

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b16	RLDVAL[15:0]	校准计数重载值	RLDVAL设定的是16位校准计数器的重载值。 连续校准时 $RLDVAL = (F_{hrc} \div PSC_{hrc}) \div (F_{ref} \div PSC_{ref}) + OFSVAL$ 单次校准时 $RLDVAL = ((F_{hrc} \div PSC_{hrc}) \times T_{trim}) + OFSVAL$ F _{hrc} : HRC目标频率; F _{ref} : 参考时钟频率; T _{trim} : 单次校准周期;	RW
b15~b8	Res	保留位	读出时为“0”，写入时写“0”	R
b7~b0	OFSVAL[7:0]	校准目标偏差值	OFSVAL设定的是HRC在校准时允许的偏差值。 连续校准时 $OFSVAL = (F_{hrc} \div PSC_{hrc}) \div (F_{ref} \div PSC_{ref}) \times T_A$ 单次校准时 $OFSVAL = ((F_{hrc} \div PSC_{hrc}) \times T_{trim}) \times T_A$ F _{hrc} : HRC目标频率; F _{ref} : 参考时钟频率; T _{trim} : 单次校准周期; T _A : 校准精度, 例如0.5%; PSC _{hrc} : HRC分频数; PSC _{ref} : 参考时钟分频数;	RW

13.6.3 时钟校准状态寄存器 (CTC_STR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												CTC SY	TRM UDF	TRM OVF	TRIM OK

位/位域	标记	位名	功能	读写
b31~b4	Res	保留位	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

13.6.4 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/位域	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

14 DMA 控制器 (DMA)

14.1 概述

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。

14.2 主要特性

- DMA 总线独立于 CPU 总线，按照 AMBA AHB-Lite 总线协议传输
- 拥有 1 个 DMA 控制单元，共 8 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 1~65535 次传输或无限次传输
- 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，传输完成中断和传输错误中断。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
- 支持软件启动和软件触发通道重置功能

14.3 功能说明

14.3.1 DMA 框图

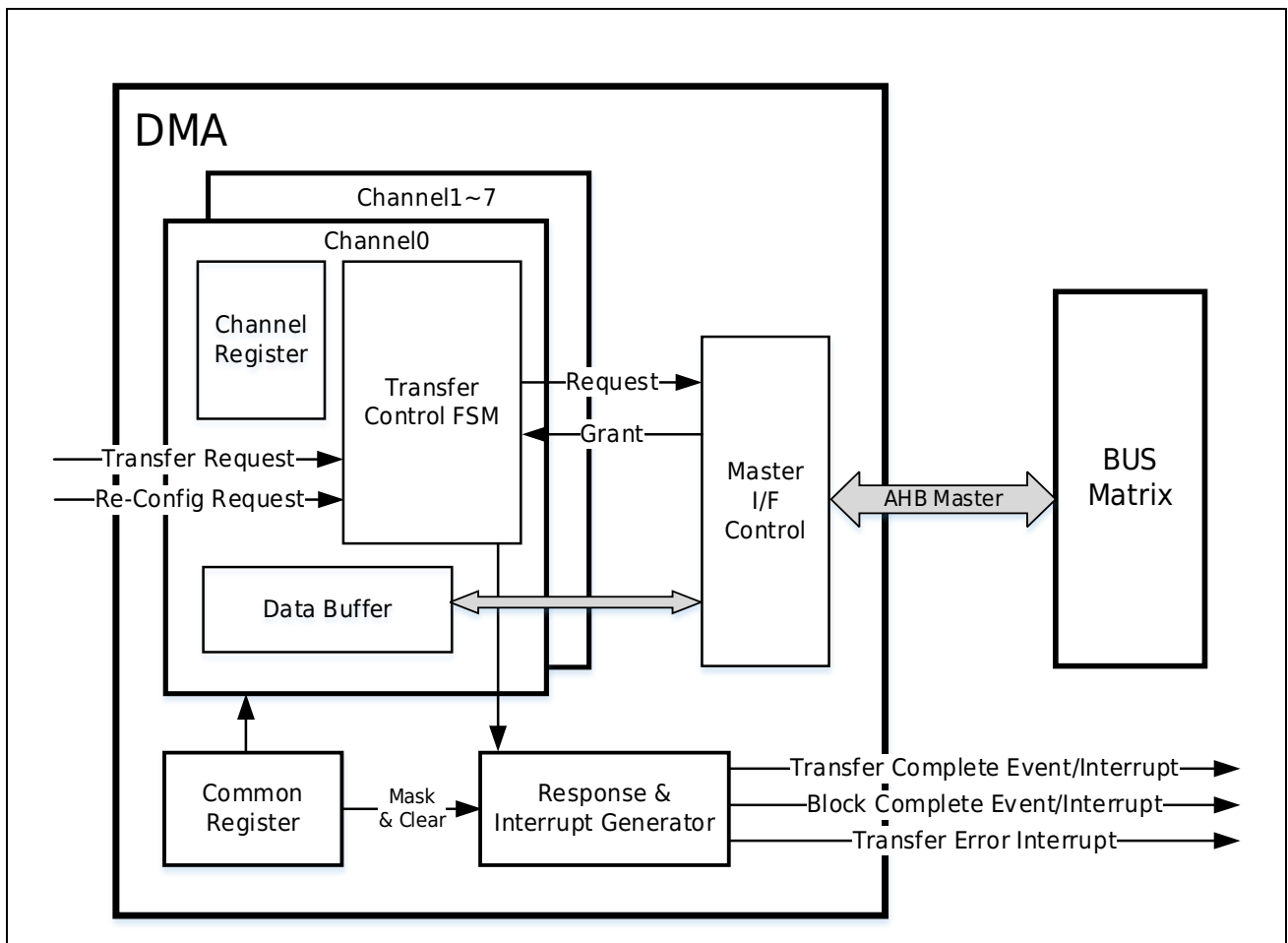


图 14-1 DMA 结构图

14.3.2 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。

14.3.3 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。

14.3.4 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。

14.3.5 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。

14.3.6 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。

14.3.7 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。

14.3.8 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_LL Px
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 LLPEN 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 LLPEN 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，LLP 指定的下一个描述符将被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。或者根据寄存器 DMA_CHCTLx.LLPRUN 的设置，在载入新描述符后直接开始第一次传输。

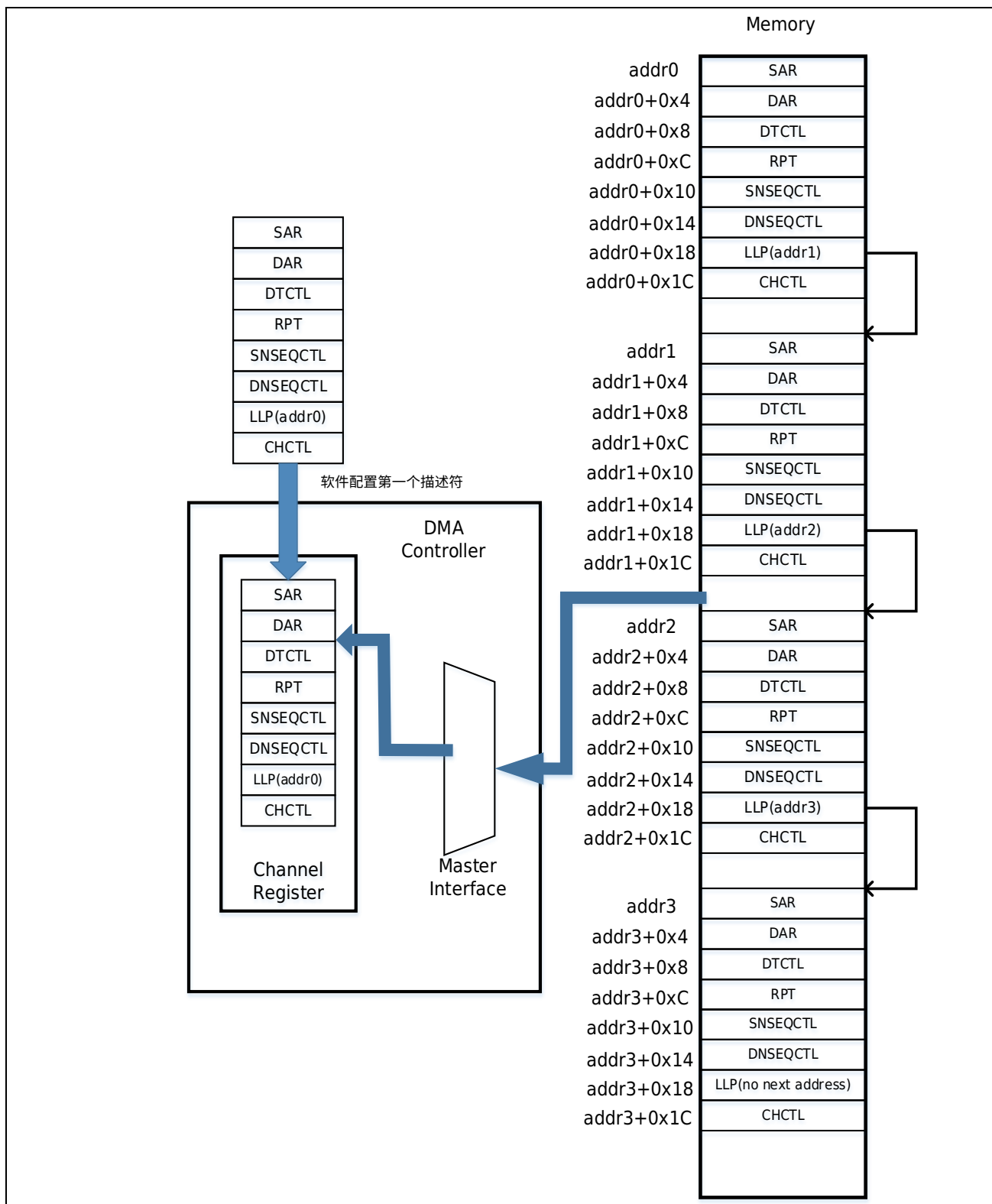


图 14-2 连锁传输示意图

14.3.9 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 will 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。

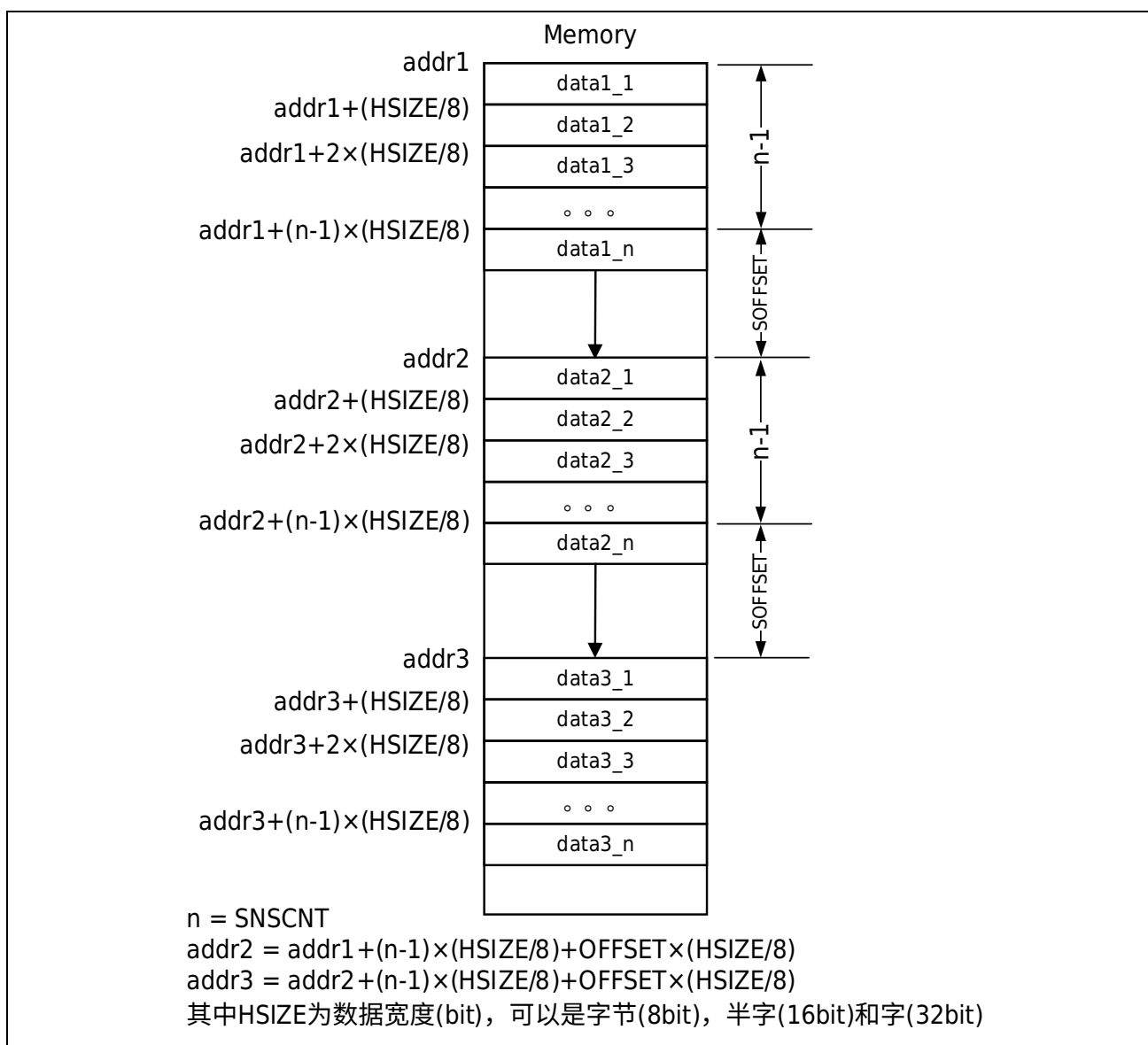


图 14-3 不连续地址传输示意图（以源地址为例）

14.3.10 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、重复式重置时，通道内部状态按下表所述进行更新。

表 14-1 通道重置说明

重置方式 (寄存器DMA_RCFGCTL)			通道内部状态		
CNTMD	SARMD	DARMD	剩余传输次数计数器	下次传输的源地址	下次传输的目标地址
0x0	任意	任意	保持不变	根据SARMD更新	根据DARMD更新
0x1	0x0	任意	保持不变	保持不变	根据DARMD更新
	0x1		更新为正常状态下，下一次源地址不连续跳转发生后的数值	更新为下一个源地址不连续传输区域的首地址	
	0x2, 0x3		更新为正常状态下，下一次源地址重载发生后的数值	更新为寄存器DMA_SARx的初始设定值	
0x2, 0x3	任意	0x0	保持不变	根据SARMD更新	保持不变
		0x1	更新为正常状态下，下一次目标地址不连续跳转发生后的数值		更新为下一个目标地址不连续传输区域的首地址
		0x2, 0x3	更新为正常状态下，下一次目标地址重载发生后的数值		更新为寄存器DMA_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。

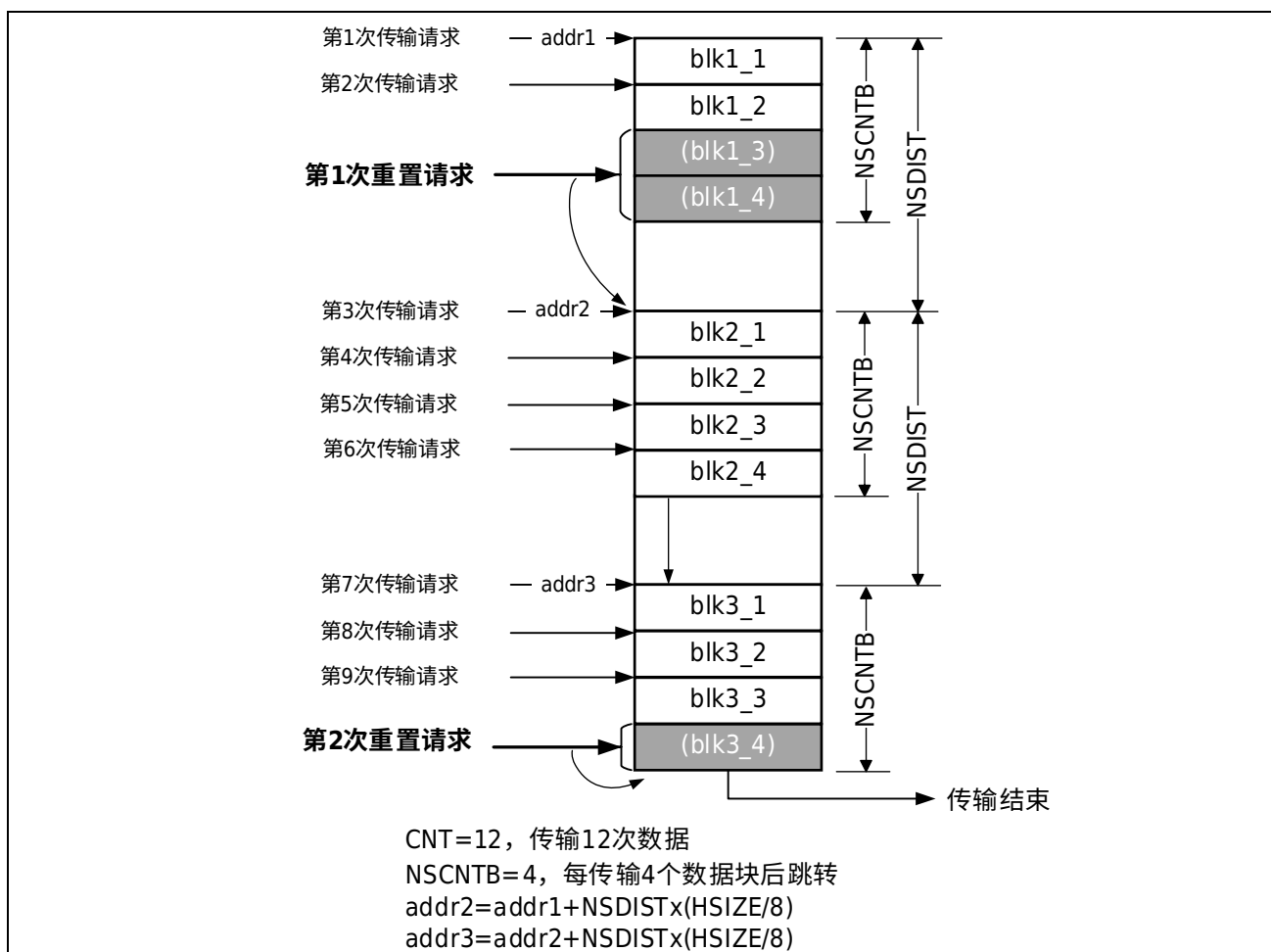


图 14-4 不连续式重置示意图

图 14-4 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。

14.3.11 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.CHENCLR x 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 CHENCLR x 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。

14.4 中断

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。

14.5 应用举例

14.5.1 存储器到存储器的传输

目标：将 12 个数据从 RAM 地址 0x2000 0000 传送到 0x2000 1000，数据宽度为 32bit。

1. 寄存器设定

- 1) DMA_EN.EN 写 1 使能 DMA 控制器
- 2) 选择一个通道，例如通道 0，配置通道寄存器以实现：
 - a. 写 DMA_SAR0 配置源地址为 SRAM 区域 0x2000 0000
 - b. 写 DMA_DAR0 配置目标地址为 SRAM 区域 0x2000 1000
 - c. 写 DMA_DTCTL0 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
 - d. 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
 - e. 配置通道控制寄存器 DMA_CHCTL0 以实现：
 - a) 源地址和目标地址连锁传输无效
 - b) 源地址重载有效，目标地址更新方式为自增
 - c) 数据宽度为 32bit
 - d) 中断使能有效
 - f. 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

- g. 通道使能位 DMA_CHEN.CHEN0 写 1, 使能通道 0
 - 3) 写外设事件软件触发寄存器 INTSFTRG.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 INTSFTRG.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_MONDTCTL0.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 INTSFTRG.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 0x2000 0000 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_MONDTCTL0.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 INTSFTRG.STRG 开始第三次传输。第三次传输完成后, 传输次数 DMA_MONDTCTL0.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.CHEN0 将被自动清零。

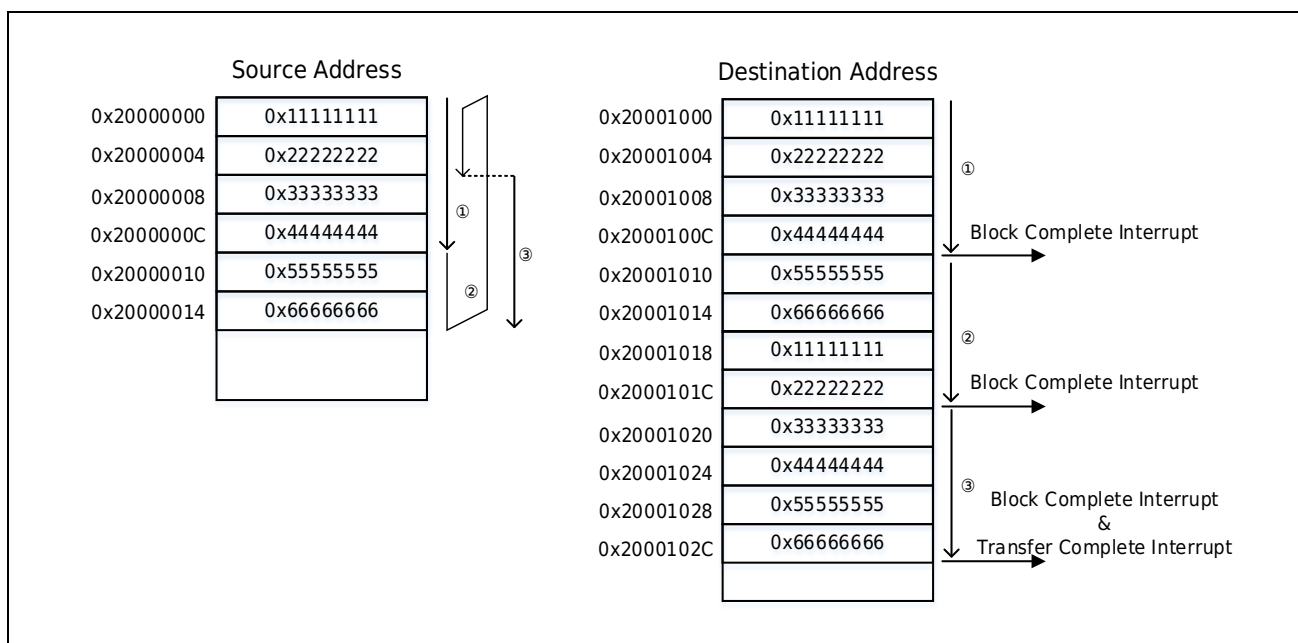


图 14-5 应用例 1: 存储器到存储器传输

14.5.2 存储器到外围电路的传输

目标:将 10 个宽度为 16bit 的数据从 RAM 地址 0x2000 0000 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
 - 1) DMA_EN.EN 写 1 使能 DMA 控制器
 - 2) 配置 DMA_INTMSK 寄存器, 屏蔽块传输完成中断, 使能传输完成中断
 - 3) 选择一个通道, 配置通道寄存器, 例如选择通道 0
 - a. 写 DMA_SAR0 配置源地址为 SRAM 区域 0x2000 0000

- b. 写 DMA_DAR0 配置目标地址为外围电路的寄存器地址 0x4000 0000
- c. 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。
- d. 配置通道控制寄存器 DMA_CHCTL0 以实现：
 - a) 源地址和目标地址连锁传输无效
 - b) 源地址更新方式为自增，目标地址为固定
 - c) 源地址和目标地址数据宽度为 16bit
 - d) 中断使能有效
- e. 配置触发源控制器 DMA_TRGSEL0，选择通信模块的发送寄存器空作为 DMA 通道 0 的启动请求
- f. 通道使能位 DMA_CHEN.CHEN0 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 中的数据传送到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.CHEN0 将被自动清零。

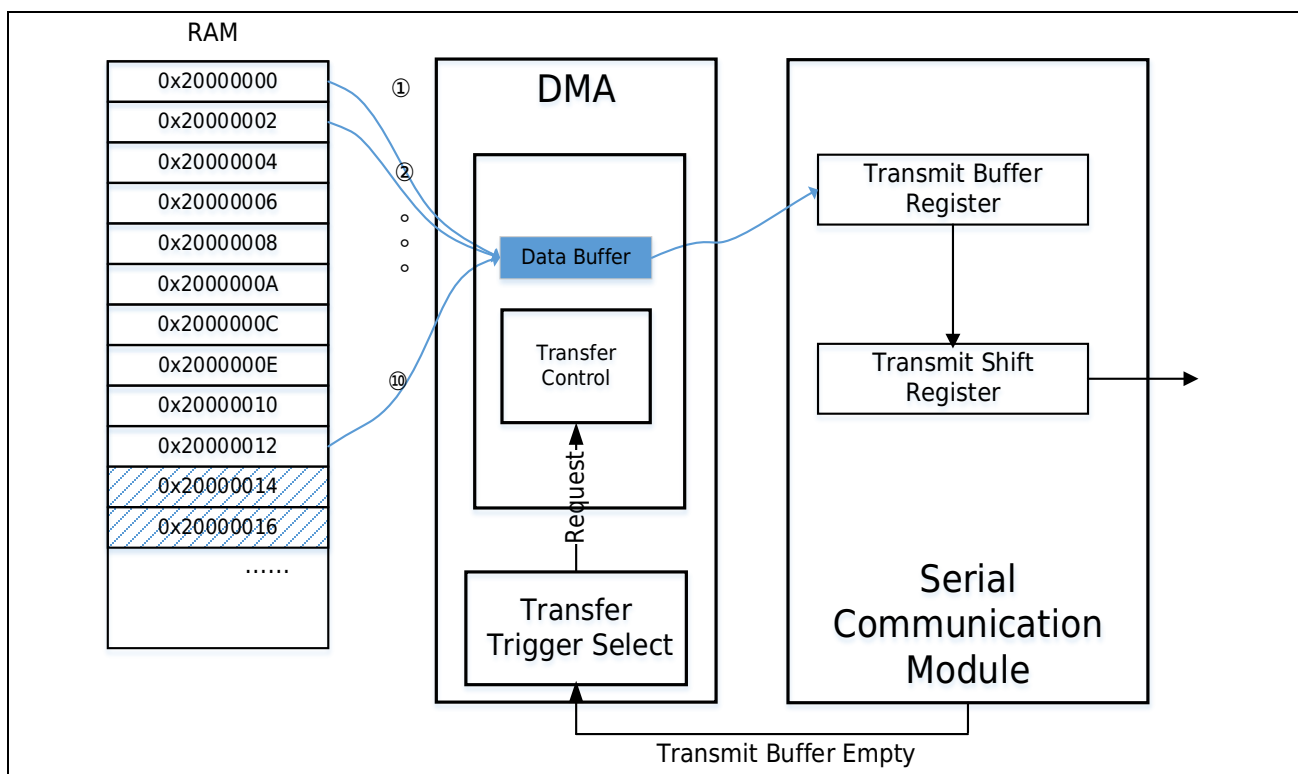


图 14-6 应用例 2：存储器到外围电路的传输

14.5.3 存储器到存储器的连锁传输

目标：通过连锁传输将 RAM 中 3 块数据传输到 3 个不同地址。其中，第一块数据为 10 个 32 位数据，从 0x2000 0000 传到 0x2000 1000。第二块数据为 20 个 16 位数据，从 0x2000 0100 传到 0x2000 1100。第三块数据为 40 个 8 位数据，从 0x2000 0200 传到 0x2000 1200。

1. 寄存器设定

1) DMA_EN.EN 写 1 使能 DMA 控制器

2) 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)

- a. 写 DMA_SAR0 配置源地址为 SRAM 区域 0x2000 0000
- b. 写 DMA_DAR0 配置目标地址为 SRAM 区域 0x2000 1000
- c. 写 DMA_DTCTL0 配置数据块的大小为 10
- d. 在链指针寄存器 DMA_LLPO 中写入第二个描述符 (descriptor1) 的地址 0x2000 2000
- e. 配置通道控制寄存器 DMA_CHCTL0，配置第一个数据块的传输参数以实现：
 - a) 连锁传输有效
 - b) 连锁传输模式为直接启动下一次传输
 - c) 源地址和目标地址更新方式为自增
 - d) 数据宽度为 32bit
 - e) 中断使能无效

3) 在 RAM 空间的 0x2000 2000 地址中配置第二次传输的描述符 (descriptor1)，包括：

- a. 0x2000 2000 中写入 32 位数据 0x2000 0100，此为第二次传输的源地址
- b. 0x2000 2004 中写入 32 位数据 0x2000 1100，此为第二次传输的目标地址
- c. 0x2000 2008 中配置数据块的大小为 20
- d. 0x2000 2018 中写入 32 位数据 0x2000 2020，此为第三次传输的描述符 (descriptor2) 所在地址
- e. 0x2000 201C 中写入第二次传输的控制数据，该控制数据实现：
 - a) 连锁传输有效
 - b) 连锁传输模式为直接启动下一次传输
 - c) 源地址和目标地址更新方式为自增
 - d) 数据宽度为 16bit
 - e) 中断使能无效

4) 在 RAM 空间的 0x2000 2020 地址中配置第三次传输的描述符 (descriptor2)，包括：

- a. 0x2000 2020 中写入 32 位数据 0x2000 0200，此为第三次传输的源地址
- b. 0x2000 2024 中写入 32 位数据 0x2000 1200，此为第三次传输的目标地址
- c. 0x2000 2028 配置数据块的大小为 40

- d. 0x2000 2038 中写入 32 位数据 0x0，代表该次传输为连锁传输的最后一次传输
- e. 0x2000 203C 中写入第三次传输的控制数据实现：
 - a) 连锁传输无效
 - b) 源地址和目标地址更新方式为自增
 - c) 数据宽度为 8bit
 - d) 中断使能有效
 - 5) 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
 - 6) 通道使能位 DMA_CHEN.CHEN0 写 1，使能通道 0
 - 7) 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，由于设置了连锁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锁传输的最后一次，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.CHEN0。

14.6 寄存器描述

表 14-2 DMA 基地址

名称	基地址	描述
DMA	0x4005 3000	DMA基地址

表 14-3 DMA 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
DMA_EN	DMA使能寄存器	0x00	32	0x0000 0000
DMA_INTSTAT0	中断状态寄存器0	0x04	32	0x0000 0000
DMA_INTSTAT1	中断状态寄存器1	0x08	32	0x0000 0000
DMA_INTMASK0	中断屏蔽寄存器0	0x0C	32	0x0000 0000
DMA_INTMASK1	中断屏蔽寄存器1	0x10	32	0x0000 0000
DMA_INTCLR0	中断复位寄存器0	0x14	32	0x0000 0000
DMA_INTCLR1	中断复位寄存器1	0x18	32	0x0000 0000
DMA_CHEN	通道使能寄存器	0x1c	32	0x0000 0000
DMA_CHENCLR	通道使能复位寄存器	0x34	32	0x0000 0000
DMA_REQSTAT	传输请求状态寄存器	0x20	32	0x0000 0000
DMA_CHSTAT	通道状态观测寄存器	0x24	32	0x0000 0000
DMA_RCFGCTL	通道重置控制寄存器	0x2c	32	0x0000 0000
DMA_SWREQ	软件启动寄存器	0x30	32	0x0000 0000
DMA_SARx ⁽¹⁾	传输源地址寄存器	0x40+0x40*x ⁽¹⁾	32	0x0000 0000
DMA_DARx	传输目标地址寄存器	0x44+0x40*x	32	0x0000 0000
DMA_DTCTLx	数据控制寄存器	0x48+0x40*x	32	0x0000 0001
DMA_RPTx	重复区域大小寄存器	0x4C+0x40*x	32	0x0000 0000
DMA_RPTBx	重复区域大小寄存器B			
DMA_SNSEQCTLx	源地址不连续传输控制寄存器	0x50+0x40*x	32	0x0000 0000
DMA_SNSEQCTLBx	源地址不连续传输控制寄存器B			
DMA_DNSEQCTLx	目标地址不连续传输控制寄存器	0x54+0x40*x	32	0x0000 0000
DMA_DNSEQCTLBx	目标地址不连续传输控制寄存器B			
DMA_LL Px	链指针寄存器	0x58+0x40*x	32	0x0000 0000
DMA_CHCTLx	通道控制寄存器	0x5C+0x40*x	32	0x0000 1000
DMA_MONSARx	传输源地址监视寄存器	0x60+0x40*x	32	0x0000 0000
DMA_MONDARx	传输目标地址监视寄存器	0x64+0x40*x	32	0x0000 0000
DMA_MONDTCTLx	数据控制监视寄存器	0x68+0x40*x	32	0x0000 0001
DMA_MONRPTx	重复区域计数器监视寄存器	0x6C+0x40*x	32	0x0000 0000
DMA_MONSNSEQCTLx	源地址不连续传输计数器监视寄存器	0x70+0x40*x	32	0x0000 0000
DMA_MONDNSEQCTLx	目标地址不连续传输计数器监视寄存器	0x74+0x40*x	32	0x0000 0000

注：(1) x=0~7。

14.6.1 DMA 使能寄存器 (DMA_EN)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
b0	EN	DMA使能位	0: DMA无效 1: DMA使能	RW

14.6.2 中断状态寄存器 0 (DMA_INTSTAT0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								REQERR[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								TRNERR[7:0]							

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”，写入时写“0”	RW
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	Res	保留位	读出时为“0”，写入时写“0”	RW
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

14.6.3 中断状态寄存器 1 (DMA_INTSTAT1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								BTC[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								TC[7:0]							

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”，写入时写“0”	RW
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	Res	保留位	读出时为“0”，写入时写“0”	RW
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

14.6.4 中断屏蔽寄存器 0 (DMA_INTMASK0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								MSKREQERR[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								MSKTRNERR[7:0]							

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b15~b6	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

14.6.5 中断屏蔽寄存器 1 (DMA_INTMASK1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								MSKBTC[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								MSKTC[7:0]							

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b15~b8	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

14.6.6 中断复位寄存器 0 (DMA_INTCLR0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								CLRREQERR[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								CLRTRNERR[7:0]							

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”，写入时写“0”	RW
b23~b16	CLRREQERR[7:0]	传输请求溢出错误中断复位	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	W
b15~b8	Res	保留位	读出时为“0”，写入时写“0”	RW
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	W

14.6.7 中断复位寄存器 1 (DMA_INTCLR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								CLRBTC[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								CLRTC[7:0]							

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”，写入时写“0”	RW
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	W
b15~b8	Res	保留位	读出时为“0”，写入时写“0”	RW
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	W

14.6.8 通道使能寄存器 (DMA_CHEN)

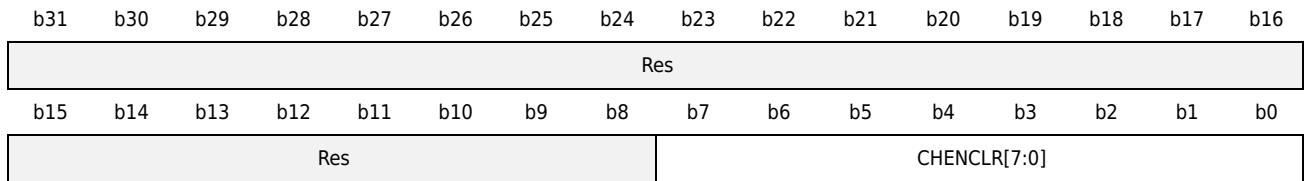
复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								CHEN[7:0]							

位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	读出时为“0”，写入时写“0”	RW
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。	RW

14.6.9 通道使能复位寄存器 (DMA_CHENCLR)

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	读出时为“0”，写入时写“0”	RW
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。	W

14.6.10 通道重置控制寄存器 (DMA_RCFGCTL)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res										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
Res				RCFGCHS[3:0]				Res					RCFGLP	RCFGEN	

位/位域	标记	位名	功能	读写
b31~b22	Res	保留位	读出时为“0”，写入时写“0”	RW
b21~b20	CNTMD[1:0]	剩余传输次数计数器重置方式	00: 保持不变 01: 按源地址方式 10, 11: 按目标地址方式 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。	RW
b19~b18	DARMD[1:0]	目标地址重置方式	00: 保持不变 01: 不连续式重置 下次传输的目标地址更新为 $addr_base + (DMA_DNSEQCTLBx.DNSDIST \times DMA_CHCTLx.HSIZE(bit)/8)$ 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。	RW
b17~b16	SARMD[1:0]	源地址重置方式	00: 保持不变 01: 不连续式重置 下次传输的源地址更新为 $addr_base + (DMA_SNSEQCTLBx.SNSDIST \times DMA_CHCTLx.HSIZE(bit)/8)$ 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。	RW
b15~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
b11~b8	RCFGCHS[3:0]	重置通道选择	0x0: 通道0 0x1: 通道1 依此类推 0x5: 通道5 其它: 禁止设定	RW
b7~b2	Res	保留位	读出时为“0”，写入时写“0”	RW
b1	RCFGLLP	链指针式通道重置	0: 链指针式重置无效 1: 链指针式重置有效	RW

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-bit25全部无效。

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	RW
----	--------	--------	--	----

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。

14.6.11 软件启动寄存器 (DMA_SWREQ)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SWRCFGWP[7:0]								SWREQWP[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SWR CFGR EQ	Res							SWREQ[7:0]							

位/位域	标记	位名	功能	读写
b31~b24	SWRCFGWP[7:0]	通道重置请求写保护	软件重置请求写保护, SWRCFGWP写入值为0xA2解除写保护。	W
b23~b16	SWREQWP[7:0]	通道重置请求写保护	软件触发请求写保护, SWREQWP写入值为0xA1解除写保护。	W
b15	SWRCFGREQ	软件通道重置请求	软件触发通道重置功能, 与AOS触发效果相同。 0b0: 写0无效 0b1: SWRCWP写0xA2的同时SWRCFGREQ写入值为1时, 产生一次软件通道重置请求	W
b14~b8	Res	保留位	读出时为“0”, 写入时请写“0”	RW
b7~b0	SWREQ[7:0]	软件通道传输请求	软件触发启动并进行传输, 每位对应一通道, 与AOS触发效果相同。 0b0: 写0无效 0b1: SWREQWP写0xA1的同时SWREQ[x]写入值为1时, 通道产生一次软件通道重置请求	W

14.6.12 传输请求状态寄存器 (DMA_REQSTAT)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															RCFG REQ
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								CHREQ[7:0]							

位/位域	标记	位名	功能	读写
b31~b17	Res	保留位	读出时为“0”，写入时写“0”	RW
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	Res	保留位	读出时为“0”，写入时写“0”	RW
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，但本标志位保持1，继续等待传输。 0: 该通道无传输请求 1: 该通道有传输请求	R

14.6.13 通道状态观测寄存器 (DMA_CHSTAT)

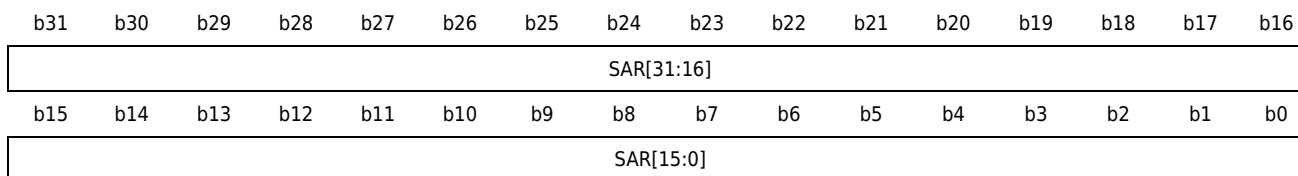
复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								CHACT[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													RCFG ACT	DMAACT	

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”，写入时写“0”	RW
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	Res	保留位	读出时为“0”，写入时写“0”	RW
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

14.6.14 传输源地址寄存器 (DMA_SARx, x=0~7)

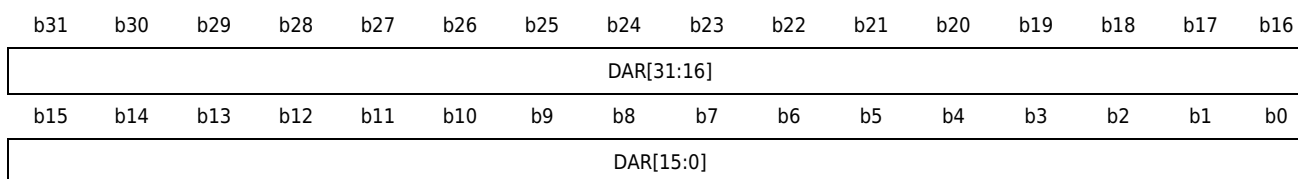
复位值: 0x0000 0000



位/位域	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=0b01, SAR[0]无效。传输数据宽度为32bit时, 即DMA_CHCTLx.HSIZE=0b1x, SAR[1:0]无效。	RW

14.6.15 传输目标地址寄存器 (DMA_DARx, x=0~7)

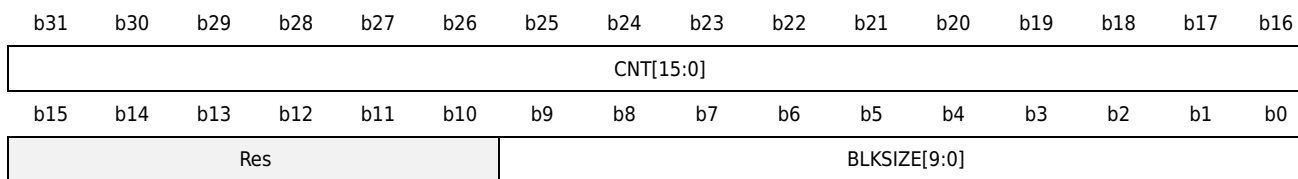
复位值: 0x0000 0000



位/位域	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=0b01, DAR[0]无效。传输数据宽度为32bit时, 即DMA_CHCTLx.HSIZE=0b1x, DAR[1:0]无效。	RW

14.6.16 数据控制寄存器 (DMA_DTCTLx, x=0~7)

复位值: 0x0000 0001



位/位域	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。	RW
b15~b10	Res	保留位	读出时为“0”，写入时写“0”	RW
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。	RW

14.6.17 重复区域大小寄存器 (DMA_RPTx, x=0~7)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res						DRPT[9:0]									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						SRPT[9:0]									

位/位域	标记	位名	功能	读写
b31~b26	Res	保留位	读出时为“0”，写入时写“0”	RW
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.HSIZE决定。	RW
b15~b10	Res	保留位	读出时为“0”，写入时写“0”	RW
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.HSIZE决定。	RW

注:

1. 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。
2. 该寄存器在本通道的重置功能打开时无效。

14.6.18 重复区域大小寄存器 B (DMA_RPTBx, x=0~7)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res						DRPTB[9:0]									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						SRPTB[9:0]									

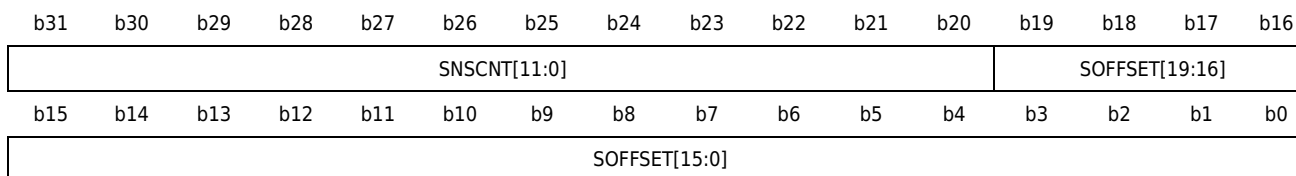
位/位域	标记	位名	功能	读写
b31~b26	Res	保留位	读出时为“0”，写入时写“0”	RW
b25~b16	DRPTB[9:0]	目标地址重复区域大小	设置目标地址重复区域大小 目标设备在每传输DRPTB个数据块后目标地址重载为DMA_DARx寄存器初始设定值。数据块大小由DMA_DTCTLx.BLKSIZE与DMA_CHCTLx.HSIZE决定。 注: 本通道重置功能有效 (DMA_RCFGCTL.RCFGEN=1) 时, DRPTB[15:0]的设定值必须大于0	RW
b16~b10	Res	保留位	读出时为“0”，写入时写“0”	RW
b9~b0	SRPTB[9:0]	源地址重复区域大小	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: 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	RW

注:

1. 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。
2. 该寄存器只在本通道的重置功能打开时有效，替代寄存器 DMA_RPTx。通道重置功能关闭时无效。

14.6.19 源地址不连续传输控制寄存器 (DMA_SNSEQCTLx, x=0~7)

复位值: 0x0000 0000



位/位域	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4096个数据后地址跳转。	RW
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 14-3。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×(DMA_CHCTLx.HSIZE(bit)/8) 例如，当SOFFSET设为10，数据宽度为字32bit时 (DMA_CHCTLx.HSIZE =0b1x) 时，地址偏移量为10×4=40，若数据宽度为半字16bit时 (DMA_CHCTLx.HSIZE=0b01) ，偏移量为10×2=20，若数据宽度为8bit (DMA_CHCTLx.HSIZE=0b00) 时，偏移量为10×1=10。 下次传输的源地址=当前传输的源地址±地址偏移量	RW

注:

1. 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。
2. 该寄存器在本通道的重置功能打开时无效。

14.6.20 源地址不连续传输控制寄存器 B (DMA_SNSEQCTLB_x, x=0~7)

复位值: 0x0000 0000

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]															

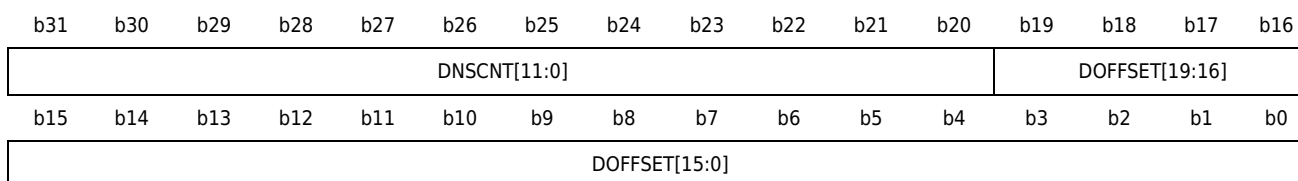
位/位域	标记	位名	功能	读写
b31~b20	SNSCNTB[11:0]	源地址跳转的数据量	<p>设置源地址跳转前传输的数据量大小。</p> <p>源设备在每传输SNSCNTB个数据块后源地址按SNSDIST指定地址间距跳转。数据块大小由DMA_DTCTLx.BLKSIZE与DMA_CHCTLx.HSIZE决定。</p> <p>注: 本通道重置功能有效 (DMA_RCFGCTL.RCFGEN=1) 时, SNSCNTB[11:0]的设定值必须大于0。</p>	RW
b19~b0	SNSDIST[19:0]	源不连续区域地址间距	<p>当不连续地址传输时, 设置源设备两个不连续区域的间距。</p> <p>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 14-4。当DMA_CHCTLx.SINC设为地址固定时, 不连续地址传输无效。</p> <p>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。</p> <p>地址间距= SNSDIST × (DMA_CHCTLx.HSIZE(bit) / 8)</p> <p>例如, 当SNSDIST设为10, 数据宽度为32bit (DMA_CHCTLx.HSIZE = 0b1x) 时, 地址间距为10 × 4 = 40, 若数据宽度为16bit (DMA_CHCTLx.HSIZE = 0b01) 时, 间距为10 × 2 = 20, 若数据宽度为8bit (DMA_CHCTLx.HSIZE = 0b00) 时, 间距为10 × 1 = 10。</p> <p>下次传输的源地址 = 当前源不连续区域首地址 ± 地址间距</p>	RW

注:

1. 使用源设备不连续传输需要配置 DMA_CHCTL_x 寄存器的 SNSEQEN 位有效, 并且配置 DMA_CHCTL_x 寄存器的 SINC 位使地址更新方式为自增或自减。
2. 该寄存器只在本通道的重置功能打开时有效, 替代寄存器 DMA_SNSEQCTL_x。通道重置功能关闭时无效。

14.6.21 目标地址不连续传输控制寄存器 (DMA_DNSEQCTLx, x=0~7)

复位值: 0x0000 0000



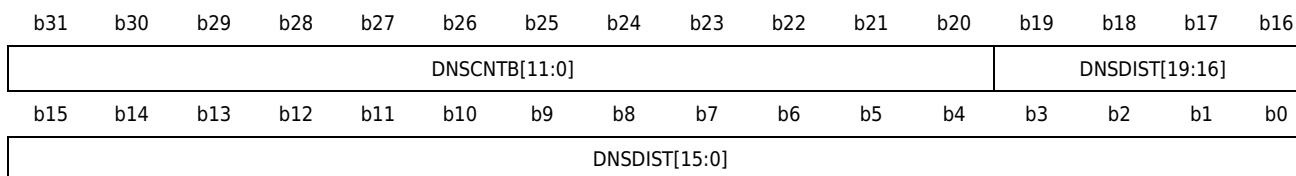
位/位域	标记	位名	功能	读写
b31~b20	DNSCNT[11:0]	目标地址跳转的数据量	<p>设置目标地址跳转前传输的数据量大小。</p> <p>目标设备在每传输DNSCNT个数据后目标地址按DOFFSET指定偏移量跳转。寄存器设为10则每传输10个数据后地址跳转，设为0则每传4096个数据后地址跳转。</p>	RW
b9~b0	DSOFFSET[19:0]	目标地址跳转的地址偏移量	<p>当不连续地址传输时，设置目标地址跳转的偏移量。偏移量是相对于当前传输地址，即跳转前最后一次传输地址而言的。跳转的方向根据通道控制寄存器DMA_CHCTLx.DINC的值向前或向后跳转。参考图 14-3。当DMA_CHCTLx.DINC设为地址固定时，不连续地址传输将无效。</p> <p>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。</p> <p>地址偏移量=DOFFSET×(DMA_CHCTLx.HSIZE(bit) / 8)</p> <p>例如，当DOFFSET设为10，若数据宽度为32bit (DMA_CHCTLx.HSIZE=0b1x) 时，地址偏移量为10×4=40，若数据宽度为16bit (DMA_CHCTLx.HSIZE=0b01) 时，偏移量为10×2=20，若数据宽度为8bit (DMA_CHCTLx.HSIZE=0b00) 时，偏移量为10×1=10。</p> <p>下次传输的目标地址=当前传输的目标地址±地址偏移量</p>	RW

注:

1. 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。
2. 该寄存器在本通道的重置功能打开时无效。

14.6.22 目标地址不连续传输控制寄存器 B (DMA_DNSEQCTLBx, x=0~7)

复位值: 0x0000 0000



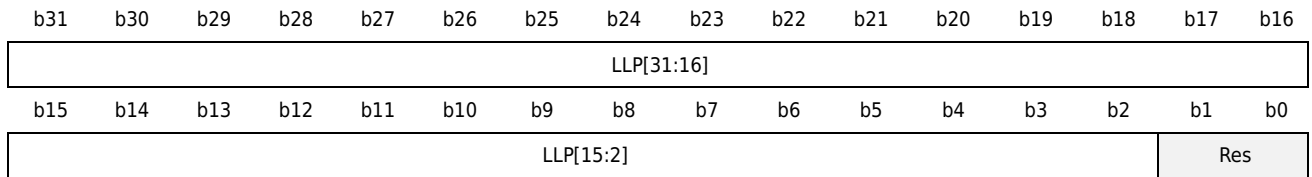
位/位域	标记	位名	功能	读写
b31~b20	DNSCNTB[11:0]	目标地址跳转的数据量	设置目标地址跳转前传输的数据量大小。 目标设备在每传输DNSCNTB个数据块后目标地址按DNSDIST指定地址间距跳转。数据块大小由DMA_DTCTLx.BLKSIZE与DMA_CHCTLx.HSIZE决定。 注: 本通道重置功能有效 (DMA_RCFGCTL.RCFGEN=1) 时, DNSCNTB[11:0]的设定值必须大于0。	RW
b19~b0	DNSDIST[19:0]	目标不连续区域地址间距	当不连续地址传输时, 设置目标设备两个不连续区域的间距。 跳转的方向根据通道控制寄存器DMA_CHCTLx.DINC的值向前或向后跳转。参考图 14-4。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× (DMA_CHCTLx.HSIZE(bit) / 8) 例如, 当DNSDIST设为10, 数据宽度为32bit (DMA_CHCTLx.HSIZE=0b1x) 时, 地址间距为10×4=40, 若数据宽度为16bit (DMA_CHCTLx.HSIZE=0b01) 时, 间距为10×2=20, 若数据宽度为8bit (DMA_CHCTLx.HSIZE=0b00) 时, 间距为10×1=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	RW

注:

1. 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。
2. 该寄存器只在本通道的重置功能打开时有效, 替代寄存器 DMA_DNSEQCTLx。通道重置功能关闭时无效。

14.6.23 链指针寄存器 (DMA_LL Px, x=0~7)

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b2	LLP[31:2]	链指针	连锁传输有效时, 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	RW
b1~b0	Res	保留位	读出时为“0”, 写入时写“0”	RW

14.6.24 通道控制寄存器 (DMA_CHCTLx, x=0~7)

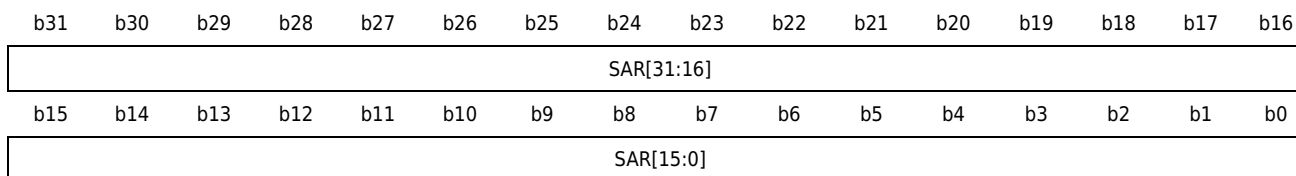
复位值: 0x0000 1000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			IE	LLPR UN	LLPEN	HSIZE[1:0]		DNSEQEN	SNSEQEN	DRPTEN	SRPTEN	DINC[1:0]		SINC[1:0]	

位/位域	标记	位名	功能	读写
b31~b13	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
b10	LLPEN	连锁传输使能	0: 连锁传输无效 1: 连锁传输有效	RW
b9~b8	HSIZE[1:0]	传输数据的宽度	00: 8bit 01: 16bit 10, 11: 32bit	RW
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	RW
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	RW
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	RW
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	RW
b3~b2	DINC[1:0]	目标地址的更新方式	00: 固定 01: 递增 10, 11: 递减	RW
b1~b0	SINC[1:0]	源地址的更新方式	00: 固定 01: 递增 10, 11: 递减	RW

14.6.25 传输源地址监视寄存器 (DMA_MONSARx, x=0~7)

复位值: 0x0000 0000



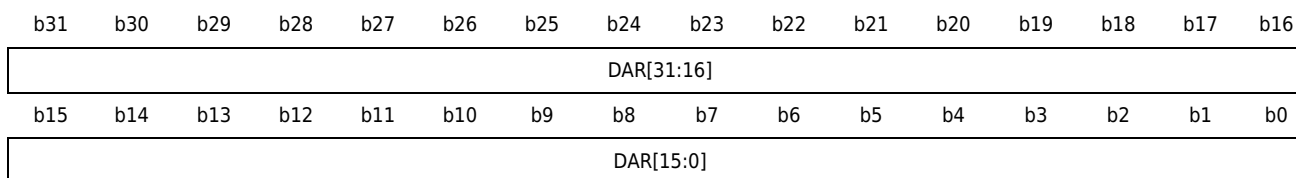
位/位域	标记	位名	功能	读写
			传输源地址。	
			注意:	
b31~b0	SAR[31:0]	传输源地址	传输数据宽度为16bit时 (DMA_CHCTLx.HSIZE=0b01), SAR[0]无 效。传输数据宽度为32bit时 (DMA_CHCTLx.HSIZE=0b10或 0b11), SAR[1:0]无效。	R

注:

传输源地址寄存器 (DMA_SARx) 在 DMA 传输前后保持不变, 而传输源地址监视寄存器 (DMA_MONSARx) 会在 DMA 每完成一个数据块传输后更新。更新内容和方式为: DMA_MONSARx.SAR[31:0]按寄存器设置的固定/递增/递减/重载/不连续跳转方式更新为下一次传输的地址。

14.6.26 传输目标地址监视寄存器 (DMA_MONDARx, x=0~7)

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
			传输目标地址。	
			注意:	
b31~b0	DAR[31:0]	传输目标地址	传输数据宽度为16bit时 (DMA_CHCTLx.HSIZE=0b01), DAR[0]无 效。传输数据宽度为32bit时 (DMA_CHCTLx.HSIZE=0b10或 0b11), DAR[1:0]无效。	R

注:

传输目标地址寄存器 (DMA_DARx) 在 DMA 传输前后保持不变, 而传输目标地址监视寄存器 (DMA_MONDARx) 会在 DMA 每完成一个数据块传输后更新。更新内容和方式为: DMA_MONDARx.DAR[31:0]按寄存器设置的固定/递增/递减/重载/不连续跳转方式更新为下一次传输的地址。

14.6.27 数据控制监视寄存器 (DMA_MONDTCTLx, x=0~7)

复位值: 0x0000 0000

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
Res							BLKSIZE[9:0]								

位/位域	标记	位名	功能	读写
b31~b16	CNT[15:0]	传输次数	剩余传输次数。 注意： 传输次数为0时 (DMA_DTCTLx.CNT=0x0000)，该位域保持为0。	R
b15~b10	Res	保留位	读出时为“0”，写入时写“0”	R
b9~b0	BLKSIZE[9:0]	数据块的大小	数据块的大小。 注意： 该位域为0时，表示每次传输1024个数据。	R

14.6.28 重复区域计数器监视寄存器 (DMA_MONRPTx, x=0~7)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res							DRPT[9:0]								
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							SRPT[9:0]								

位/位域	标记	位名	功能	读写
b31~b26	Res	保留位	读出时为“0”，写入时写“0”	RW
b25~b16	DRPT[9:0]	目标地址重复区域大小	目标地址重复区域大小。	R
b16~b10	Res	保留位	读出时为“0”，写入时写“0”	RW
b9~b0	SRPT[9:0]	源地址重复区域大小	源地址重复区域大小。	R

注:

重复区域大小寄存器 (DMA_RPTx) 在 DMA 传输前后保持不变，而重复区域计数控制监视寄存器 (DMA_MONRPTx) 会在 DMA 每完成一个数据块传输后更新。更新内容和方式为：DMA_MONRPTx.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_RPTx 设定值。重置有效时，减 1，减到 0 时重载原始设定值。

14.6.29 源地址不连续传输计数器监视寄存器 (DMA_MONSNSEQCTLx, x=0~7)

复位值: 0x0000 0000

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]															

b31~b20	SNSCNT[11:0]	源地址跳转的数据量	源地址跳转前传输的数据量大小。	R
b19~b0	SOFFSET[19:0]	源地址跳转的地址偏移量	源地址跳转的偏移量。	R

注:

源设备不连续地址传输控制寄存器 (DMA_SNSEQCTLx) 在 DMA 传输前后保持不变, 而源设备不连续传输计数器监视寄存器 (DMA_MONSNSEQCTLx) 会在 DMA 每完成一个数据块传输后更新。更新内容和方式为: DMA_MONSNSEQCTLx.SNSCNT[11:0]通道重置无效时, 减去块大小, 减到 0 时重载 DMA_SNSEQCTLx 原始设定值。重置有效时, 减 1, 减到 0 时重载原始设定值。

14.6.30 目标地址不连续传输计数器监视寄存器 (DMA_MONDNSEQCTLx, x=0~7)

复位值: 0x0000 0000

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/位域	标记	位名	功能	读写
b31~b20	DNSCNT[11:0]	目标地址跳转的数据量	目标地址跳转前传输的数据量大小。	R
b9~b0	DSOFFSET[19:0]	目标地址跳转的地址偏移量	目标地址跳转的偏移量。	R

注:

目标设备不连续地址传输控制寄存器 (DMA_DNSEQCTLx) 在 DMA 传输前后保持不变, 而目标设备不连续传输计数器监视寄存器 (DMA_MONDNSEQCTLx) 会在 DMA 每完成一个数据块传输后更新。更新内容和方式为: DMA_MONDNSEQCTLx.SNSCNT[11:0]通道重置无效时, 减去块大小, 减到 0 时重载 DMA_DNSEQCTLx 原始设定值。重置有效时, 减 1, 减到 0 时重载原始设定值。

14.7 注意事项

- DMA 的寄存器只支持 32bit 读写，8/16bit 读写操作无效。

15 电压比较器 (CMP)

15.1 概述

电压比较器 (Comparator, 以下简称 CMP) 是将两个模拟电压进行比较并且输出比较结果的外设模块。本产品搭载两组共 3 个比较通道: CMP1/ CMP2、CMP3。

15.2 主要特性

- 3 个可比较通道可独立进行普通比较
- CMP1/CMP2 两个比较通道组合使用可实现窗口比较
- 每个比较通道的正/负端电压均有多个输入源 (IO/DAC) 供选择
- 迟滞电压可配置
- 噪声滤波器可以对比较器输出滤波, 7 种采样时钟可选
- 可使用定时器 PWM 进行比较器空白窗口控制
- 可在比较结果的变化边沿产生中断、触发其他外设以及唤醒 STOP 模式
- 比较结果可通过寄存器监视, 也可输出到外部管脚 VCOUT
- 比较结果可用于紧急刹车 (EMB) 控制事件
- 比较结果可用于 HRPWM 的外部事件源
- 软件复位和看门狗复位发生时比较器输出保持

15.3 功能说明

15.3.1 CMP 功能框图

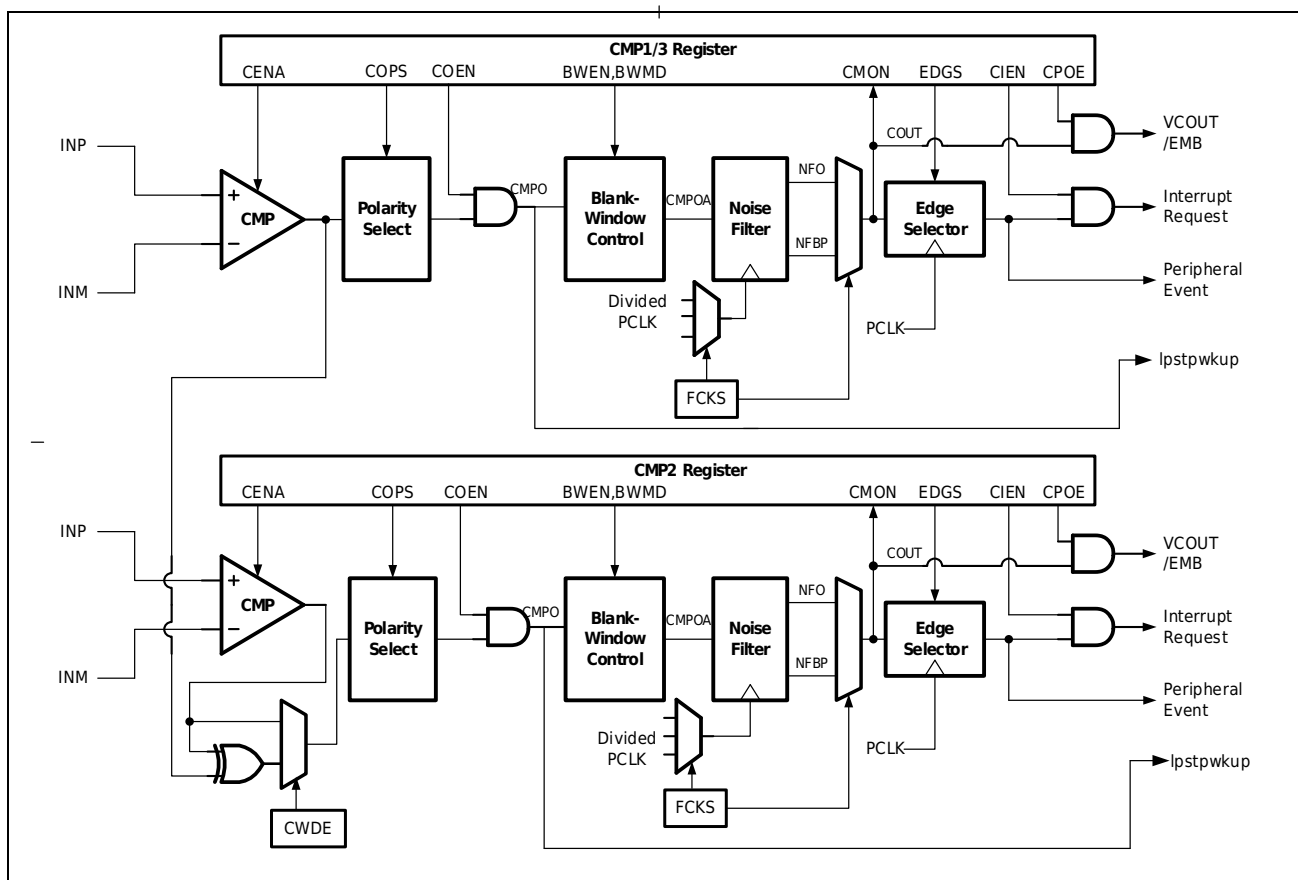


图 15-1 CMP 功能框图

表 15-1 CMP 引脚一览表

引脚名称	输入/输出	功能
AVCC	输入	模拟电源
AVSS	输入	模拟地
CMP1_INP1	输入	CMP1正端模拟输入
CMP123_INP2	输入	CMP1, 2, 3正端模拟输入
CMP123_INM1	输入	CMP1, 2, 3负端模拟输入
CMP1_INM2	输入	CMP1负端模拟输入
CMP2_INP2	输入	CMP2正端模拟输入
CMP2_INM2	输入	CMP2负端模拟输入
CMP3_INP2	输入	CMP3正端模拟输入
CMP3_INM2	输入	CMP3负端模拟输入
VCOUT1	输出	CMP1的外部端口输出
VCOUT2	输出	CMP2的外部端口输出

引脚名称	输入/输出	功能
VCOUT3	输出	CMP3的外部端口输出
VCOUT	输出	CMP1~3的外部端口输出

15.3.2 输入选择

CMP1~3 的模拟输入分别按照下表进行选择。

表 15-2 CMP 模拟输入一览表

CMP1				
序号	寄存器 PMSR.CVSL	正端输入 INP	寄存器 PMSR.RVSL	负端输入 INM
1	0b01	CMP1_INP1	0b00001	CMP123_INM1
2	0b10	CMP123_INP2	0b00010	CMP1_INM2
3	-	-	0b00100	DA101
4	-	-	0b01000	DA102
5	-	-	0b10000	DA201
CMP2				
序号	寄存器 PMSR.CVSL	正端输入 INP	寄存器 PMSR.RVSL	负端输入 INM
1	0b01	CMP2_INP1	0b00001	CMP123_INM1
2	0b10	CMP123_INP2	0b00010	CMP2_INM2
3	-	-	0b00100	DA101
4	-	-	0b01000	DA102
5	-	-	0b10000	DA201
CMP3				
序号	寄存器 PMSR.CVSL	正端输入 INP	寄存器 PMSR.RVSL	负端输入 INM
1	0b01	CMP3_INP1	0b00001	CMP123_INM1
2	0b10	CMP123_INP2	0b00010	CMP3_INM2
3	-	-	0b00100	DA101
4	-	-	0b01000	DA102
5	-	-	0b10000	DA201

15.3.3 普通比较

CMP1~3 可以独立进行普通比较。以 CMP1 为例，设定步骤如下：

1. 使能 CMP1 模块功能 (PWC_FCG3.CMP12 写 0)；
2. 设定 CMP_PMSR 寄存器的 CVSL 位，选择正端电压；
3. 设定 CMP_PMSR 寄存器的 RVSL 位，选择负端电压；

4. CMP_MDR 寄存器的 CENA 位设为 1，启动比较器；
5. 等待比较器的稳定时间 tCMP（具体数值请参考产品数据手册）；
6. 设定 CMP_FIR 寄存器的 FCKS 位，设置数字滤波的采样周期；
7. 设定 CMP_FIR 寄存器的 EDGS、CIEN 位，设置中断发生的边沿；
8. 设定 CMP_OCR 寄存器的 CPOE、COPS 位，设置 VCOUT 的输出；
9. CMP_OCR 寄存器的 COEN 位设为 1，允许比较器输出。

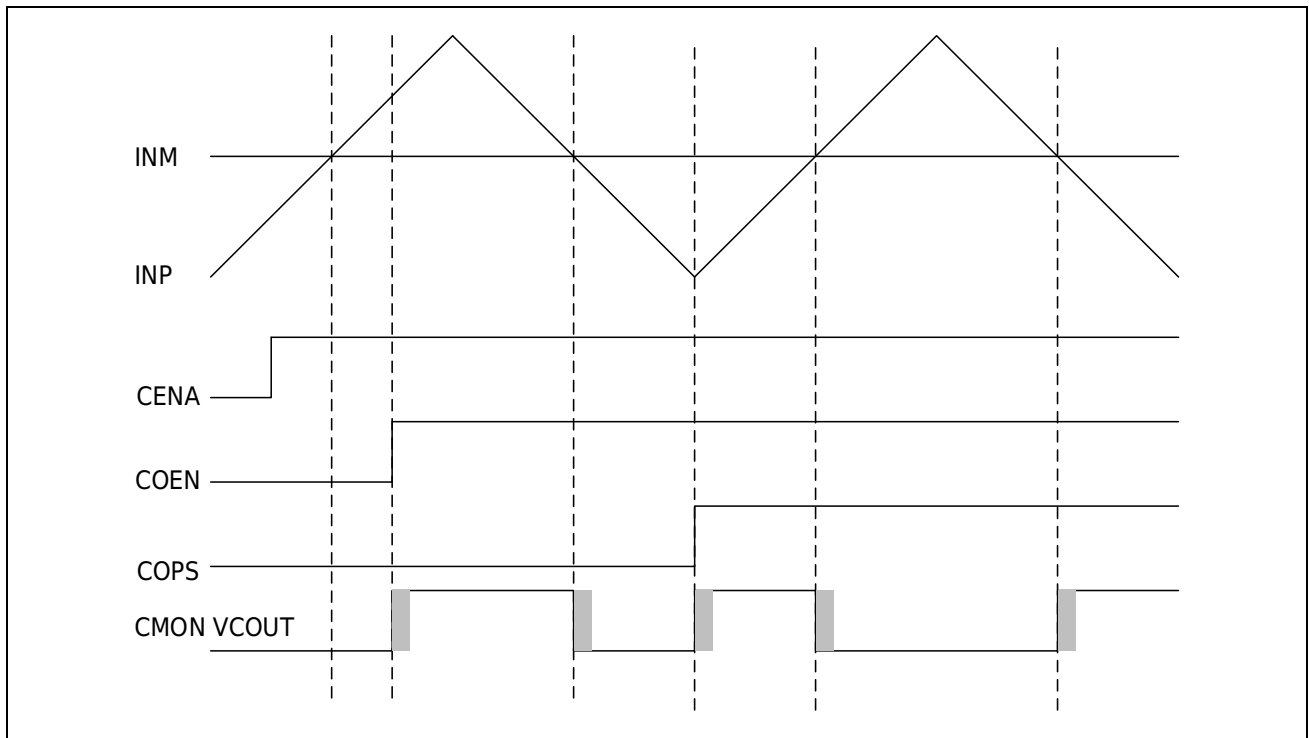


图 15-2 普通比较示意图

如图 15-2，在正相输出时，CMP 允许输出后如果正端电压高于负端电压，CMP_MDR 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。

15.3.4 窗口比较

CMP1 与 CMP2 同时运行时可实现窗口比较。设定步骤如下：

1. 使能 CMP1 与 CMP2 模块功能（PWC_FCG3.CMP12 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，选择的负端电压作为窗口下限；
3. 设定 CMP2_PMSR 寄存器的 CVSL 位，选择与 CMP1 相同的正端电压，

- 设定 CMP2_PMSR 寄存器的 RVSL 位，选择的负端电压作为窗口上限；
4. CMP2_MDR 寄存器的 CWDE 位设为 1，启动窗口比较；
 5. CMP1_MDR 和 CMP2_MDR 寄存器的 CENA 位设为 1，启动比较器；
 6. 等待稳定时间 t_{CMP}；（具体数值请参考产品数据手册）；
 7. 设定 CMP2_FIR 寄存器的 FCKS 位，设置数字滤波器；
 8. 设定 CMP2_FIR 寄存器的 EDGS、CIEN 位，设置中断和边沿检测条件；
 9. 设定 CMP2_OCR 寄存器的 CPOE、COPS 位，设置 V_{CO} 的输出；
 10. CMP1_OCR, CMP2_OCR 寄存器的 COEN 位设为 1，允许比较器输出。

窗口比较时，由 CMP2 完成对比较结果的监视、滤波、中断和输出等操作。

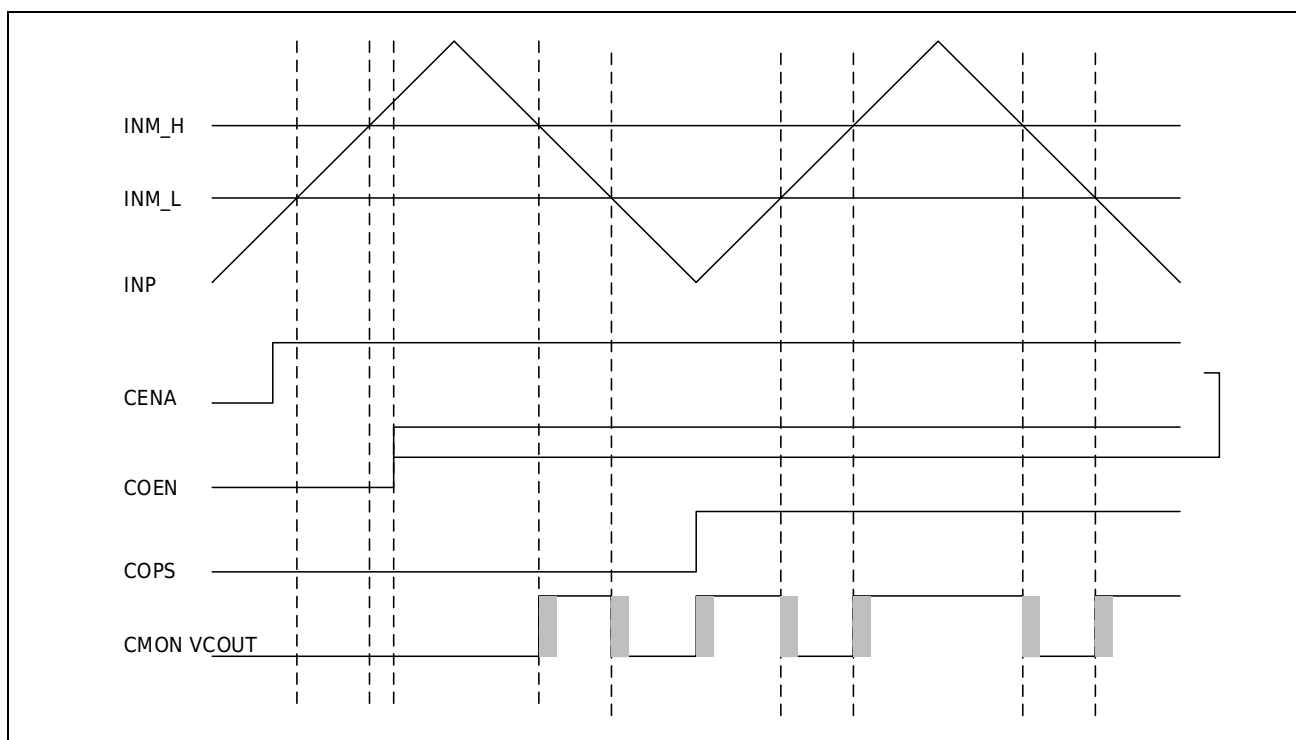


图 15-3 窗口比较示意图

如图 15-3, 在正相输出时, CMP 允许输出后如果正端电压处于两个负端电压组成的窗口内, CMP2_MDR 寄存器的 CMON 位为 1; 如果正端电压处于两个负端电压组成的窗口外, 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。

15.3.5 输入扫描

输入扫描可实现多个输入轮流与同一个输入进行比较。使用时先通过设置 CMP_MDR 寄存器的 CSMD 位选择扫描正端或者负端输入, 然后设置 CMP_SCCR 寄存器的 SISL 位选择具体参与扫描的正/负端输

入。CMP_SCCR 寄存器的 SPRD 位设定扫描周期，也就是扫描时切换输入的时间间隔。SSTB 位设定切换输入后关闭 CMP 输出的时间，这是为了等待 CMP 输出稳定从而避免系统产生误动作。

将 CMP_MDR 寄存器的 CSST 位设为 1，扫描开始。扫描时，非扫描端口保持不变，扫描端口根据设定的扫描周期循环切换。其间如果监测到比较结果有上升沿发生，CSST 自动清零并停止扫描。这时可以通过读取 CMP_SCMR 寄存器来确定当前正负端的输入状态。当 CSST 再次被写为 1 时，扫描动作重新开始。

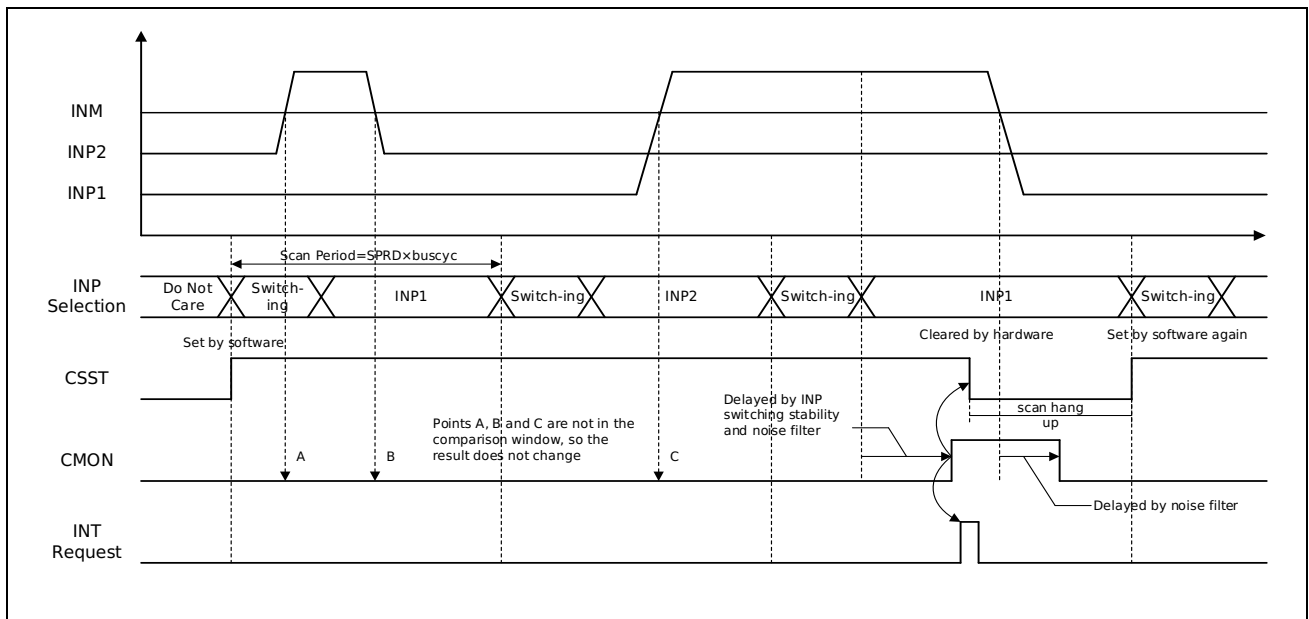


图 15-4 输入扫描示意图

如图 15-4 是扫描两个正端输入的动作示意图。在前两个扫描周期，由于 INP1 和 INP2 的变化（图中 A, B, C 三个变化点）都未发生在各自的扫描周期内，比较结果未变化。接下来再次扫描 INP1 时，INP1 高于 INM 引起比较器反转，此时比较器输出由于等待输入切换稳定而关闭。等到比较器输出稳定，比较结果经过噪声滤波后输出。同时，扫描动作自动停止并产生中断。接下来，CMP 会一直保持工作在 INP1 与 INM 比较的状态，直到软件重启扫描。

使用时请参照以下流程进行设定：

1. 设定 CMP_MDR 寄存器的 CSMD 为 0b10 或者 0b11；
2. CSMD 设为 0b10 时，设定 CMP_PMSR 寄存器的 RVSL 位，选择一个 INM 输入，CSMD 设为 0b11 时，设定 CMP_PMSR 寄存器的 CVSL 位，选择一个 INP 输入；
3. 设定 CMP_SCCR 寄存器的 SISL 位，选择扫描对象端口；
4. 设定 CMP_SCCR 寄存器的 SSTB 位，设置输出稳定时间；
5. 设定 CMP_SCCR 寄存器的 SPRD 位，设置扫描周期；
6. 设定 CMP_FIR 寄存器的 EDGS 位，选择检测上升沿；

7. 设定 CMP_FIR 寄存器的 CIEN 位，允许中断；
8. 设定 CMP_MDR 寄存器的 CENA 位，启动比较器；
9. 等待比较器的稳定时间 tCMP（具体数值请参考产品数据手册）；
10. 设定 CMP_OCR 寄存器的 COEN 位，允许比较器输出；
11. 设定 CMP_MDR 寄存器的 CSST 位，启动扫描。

注意：

为了保证动作正确必须使设定值满足“扫描周期 > 输出稳定时间 + 滤波采样周期 × 4 + CMPCLK 周期 × 8”。同时，要设定检测上升沿来自动停止扫描或者发生中断。如果要检测下降沿，可通过设定 CMP 反相输出实现。另外，由于 CMP1, 2 的输入扫描需要单独启动，很难做到步调一致。建议不要在窗口比较时使用输入扫描。

15.3.6 窗口输出

CMP 可将 TMR4, TMR6, HRPWM, TMRA 的输出作为空白窗口 (Blank-Window) 信号对比较结果 CMPOA 进行窗口输出控制，这为定时器 PWM 过零检测等应用场景提供了方便。窗口输出 (Blank-Window) 有电平和边沿两种模式。

1) 电平模式

启动窗口输出 (CMP_MDR 寄存器的 BWEN 位为 1) 并且选择电平模式 (CMP_OCR 寄存器的 BWMD 位为 0) 时，通过 CMP_BWSR1 寄存器的 CTWS 位选择一个或多个定时器 PWM 作为 Blank-Window 并通过 CTWP 位为各 Blank-Window 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 CMP_OCR 寄存器的 BWOL 位设定的状态。

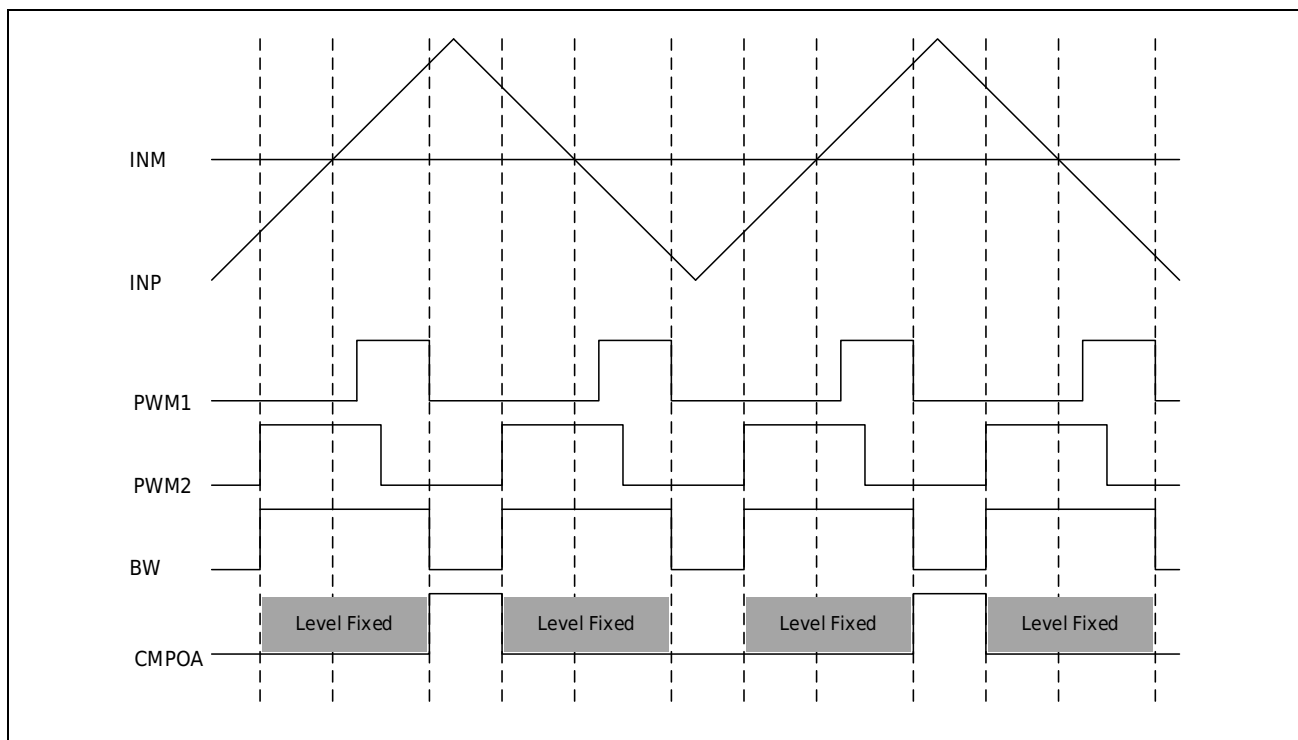


图 15-5 窗口输出示意图一

如图 15-5, 选择了两个 PWM 作为 Blank-Window 并且都是高电平有效, Blank-Window 有效时 CMPOA 的输出状态选择的是低电平。于是在 Blank-Window 信号为高电平时 CMPOA 被固定成了低电平, 而在 Blank-Window 信号为低电平时 CMPOA 则正常输出比较结果。

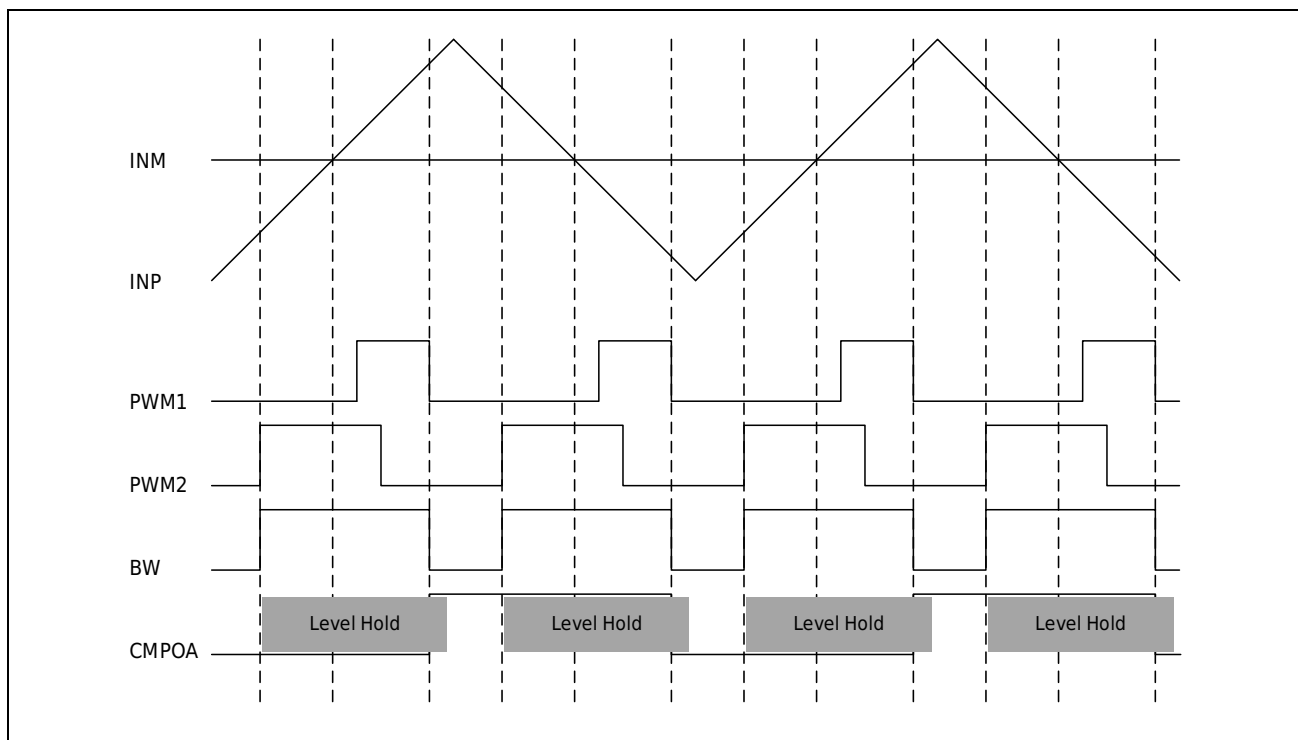


图 15-6 窗口输出示意图二

如图 15-6, 选择了两个 PWM 作为 Blank-Window 并且都是高电平有效, Blank-Window 有效时 CMPOA 的输出状态选择的是保持。于是 Blank-Window 信号为高电平时 CMPOA 则保持了前面的状态; 而在 Blank-Window 信号为低电平时 CMPOA 则正常输出比较结果, 但输出会晚 1~3 个 CMPCLK 周期, 这是由同步和采样造成的。

2) 边沿模式

启动窗口输出 (CMP_MDR 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 CMP_BWSR1 寄存器的 CTWS 位选择一个或多个定时器 PWM 作为 Blank-Window, 并通过 CMP_BWSR2 寄存器的 TWEG 位选择窗口信号的有效边沿。这样窗口输出 CMPOA 在 Blank-Window 信号的有效边沿处会被固定成由 CMP_OCR 寄存器的 BWOL 位设定的状态, 并维持由 CMP_BWSR2 寄存器的 MSKW 位设定的 N 个 CMPCLK 周期。

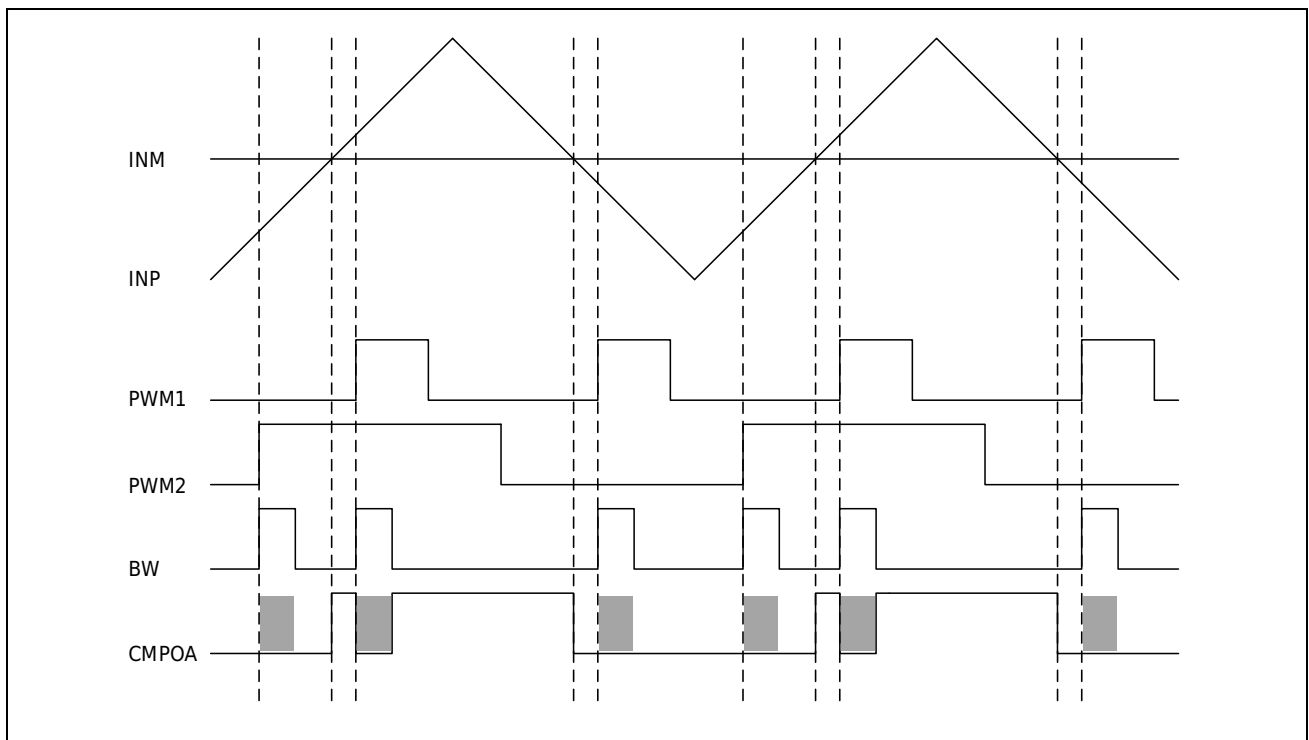


图 15-7 窗口输出示意图三

如图 15-7, 选择了两个 PWM 作为 Blank-Window 并且选择上升沿为有效边沿, Blank-Window 有效时 CMPOA 的输出状态选择的是低电平。于是在 PWM 信号的每个上升沿都会产生 Blank-Window 信号, 并且在 Blank-Window 信号为高电平时 CMPOA 被固定成了低电平, 而在 Blank-Window 信号为低电平时 CMPOA 则正常输出比较结果。

15.3.7 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 0b000 时迟滞功能关闭。

15.3.8 噪声滤波器

所有比较通道各包含一个噪声滤波器。噪声滤波器会在每个采样时钟周期对比较器输出 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 CMP_FIR 寄存器的 FCKS 位选择，当 FCKS 位设为 0b00 时噪声滤波器关闭。

15.3.9 外部管脚输出

可以将噪声滤波器输出的比较结果 COUT 输出到外部端口 VCOUT。通过 CMP_OCR 寄存器的 CPOE、COPS 位可以分别设定是否允许 VCOUT 输出以及输出极性（正相输出或者反相输出）。复位后 VCOUT 对应的端口为通用端口且为输入状态，因此还必须通过端口寄存器将其设定为 VCOUT 输出功能。

15.4 中断和事件

所有比较通道的结果均可以产生中断请求。使用时，必须将 CMP_FIR 寄存器的 CIEN 位设置成 1 并通过 CMP_FIR 寄存器的 EDGS 位（0b00 以外的值）设定可以产生中断的比较结果的边沿。如果使用了噪声滤波和反相输出，那么边沿检测的是经过噪声滤波滤波和反相输出后的信号。

INTC_WKEN.CMP*WKEN=1 的时候，CMPO 可以唤醒低功耗停止模式，系统将在 CMPO 由低变高时被唤醒并产生一次中断。

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。

15.5 寄存器描述

表 15-3 CMP 基地址

名称	基地址	描述
CMP1	0x4003 8400	CMP1基地址
CMP2	0x4003 8500	CMP2基地址
CMP3	0x4003 B000	CMP3基地址

表 15-4 CMP 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
CMPx_MDR	比较器工作模式寄存器	0x00	8	0x00
CMPx_FIR	比较器滤波和中断寄存器	0x01	8	0x00
CMPx_OCR	比较器输出控制寄存器	0x02	8	0x00
CMPx_OMR	比较器结果监视寄存器	0x03	8	0x00
CMPx_PMSR	比较器正负端输入选择寄存器	0x04	32	0x0000 0000
CMPx_BWSR1	比较器输出空白窗口设定寄存器1	0x10	32	0x0000 0000
CMPx_BWSR2	比较器输出空白窗口设定寄存器2	0x14	16	0x0000
CMPx_SCCR	比较器输入扫描控制寄存器	0x18	32	0x0000 0000
CMPx_SCMR	比较器输入扫描监视寄存器	0x1C	32	0x0000 0000

表中 x=1~3

所有寄存器在软件复位和看门狗复位发生时将保持。

15.5.1 比较器工作模式寄存器 (CMPx_MDR, x=1~3)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
CSST	CSMD[1:0]		WDEN	HYST[2:0]			CENA

位	标记	位名	功能	读写
b7	CSST	输入扫描启动	0: 停止输入扫描 1: 开始输入扫描	RW
b6~b5	CSMD[1:0]	输入扫描模式	0x: 关闭输入扫描 10: 扫描正端输入INP 11: 扫描负端输入INM	RW
b4	WDEN	窗口比较使能	WDEN只存在于CMP2中。设为窗口比较后，比较结果的滤波、边缘检测以及输出等控制均由CMP2完成。 0: 普通比较模式 1: 窗口比较模式	RW
b3~b1	HYST[2:0]	迟滞电压选择	以下列出的均为典型值，误差范围请参考产品数据手册。 000: 0mv (无迟滞) 001: 10mv 010: 20mv 011: 30mv 100: 40mv 101: 50mv 110: 60mv 111: 70mv	RW
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。	RW

15.5.2 比较器滤波和中断寄存器 (CMPx_FIR, x=1~3)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
CRF	CFF	EDGS[1:0]		CIEN	FCKS[2:0]		

位/位域	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。	RW
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。	RW
b5~b4	EDGS[1:0]	比较器输出边沿检测选择	00: 检测比较器输出COUT的高电平 01: 检测比较器输出COUT的上升沿 10: 检测比较器输出COUT的下降沿 11: 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。	RW
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	RW
b2~b0	FCKS[2:0]	比较器结果滤波采样选择	000: 不使用噪声滤波器 001: 使用噪声滤波器, 通过PCLK0进行采样 010: 使用噪声滤波器, 通过PCLK0/2进行采样 011: 使用噪声滤波器, 通过PCLK0/4进行采样 100: 使用噪声滤波器, 通过PCLK0/8进行采样 101: 使用噪声滤波器, 通过PCLK0/16进行采样 110: 使用噪声滤波器, 通过PCLK0/32进行采样 111: 禁止设定 请在比较器输出禁止 (即COEN位为“0”) 时改写FCKS[2:0]。 FCKS[2:0]从“0b00”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。	RW

15.5.3 比较器输出控制寄存器 (CMPx_OCR, x=1~3)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
BWOL[1:0]		BWMD	BWEN	Res	CPOE	COPS	COEN

位/位域	标记	位名	功能	读写
b7~b6	BWOL[1:0]	Blank-Window输出电平	0 0: Blank-Window有效 (输出关闭) 期间比较器输出低电平 0 1: Blank-Window有效 (输出关闭) 期间比较器输出高电平 1 x: Blank-Window有效 (输出关闭) 期间比较器输出保持	RW
b5	BWMD	Blank-Window模式	0: 电平模式 1: 边沿模式	RW
b4	BWEN	Blank-Window使能	0: 关闭Blank-window功能 1: 打开Blank-window功能	RW
b3	Res	保留位	读出时为“0”，写入时写“0”	RW
b2	CPOE	VCOUT输出允许	0: 禁止VCOUT输出 1: 允许VCOUT输出	RW
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。	RW
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	RW

15.5.4 比较器结果监视寄存器 (CMPx_OMR, x=1~3)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res							CMON

位	标记	位名	功能	读写
b7~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
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有效) 0: 正端电压在窗口外 1: 正端电压在窗口内	R

15.5.5 比较器正负端输入选择寄存器 (CMPx_PMSR, x=1~3)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res						CVSL[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res			RVSL[4:0]				

位/位域	标记	位名	功能	读写
b31~b18	Res	保留位	读出时为“0”，写入时写“0”	RW
b17~b16	CVSL[1:0]	正端输入选择	<p>选择比较器的正端输入电压，正端输入扫描时所设值无效。</p> <p>00: 无正端电压输入</p> <p>01: 选择INP1作为正端电压（所选输入见表 15-2）</p> <p>10: 选择INP2作为正端电压（所选输入见表 15-2）</p> <p>其他: 禁止设定</p> <p>当扫描正端输入有效（即CMP_MDR.CSMD[1:0]=“0b10”）时，本寄存器设定无效。</p> <p>请在比较器输出禁止（即COEN位为“0”）时改写CVSL。每次改写CVSL后请等待300ns再允许比较器输出（即COEN位设为“1”）。改变CVSL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。</p>	RW
b15~b5	Res	保留位	读出时为“0”，写入时写“0”	RW
b4~b0	RVSL[4:0]	负端输入选择	<p>选择电压比较器的负端输入电压，负端输入扫描时所设值无效。</p> <p>00000: 无负端电压输入</p> <p>00001: 选择INM1作为负端电压（所选输入见表 15-2）</p> <p>00010: 选择INM2作为负端电压（所选输入见表 15-2）</p> <p>00100: 选择INM3作为负端电压（所选输入见表 15-2）</p> <p>01000: 选择INM4作为负端电压（所选输入见表 15-2）</p> <p>10000: 选择INM5作为负端电压（所选输入见表 15-2）</p> <p>其他: 禁止设定</p> <p>当扫描负端输入有效（即CMP_MDR.CSMD[1:0]=“0b11”）时，本寄存器设定无效。</p> <p>请在比较器输出禁止（即COEN位为“0”）时改写RVSL。每次改写RVSL后请等待300ns再允许比较器输出（即COEN位设为“1”）。改变RVSL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。</p>	RW

15.5.6 比较器输出空白窗口设定寄存器 1 (CMPx_BWSR1, x=1~3)

复位值: 0x0000 0000

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	CTWS13	CTWS12	CTWS11	CTWS10	CTWS9	CTWS8
b7	b6	b5	b4	b3	b2	b1	b0
CTWS7	CTWS6	CTWS5	CTWS4	CTWS3	CTWS2	CTWS1	CTWS0

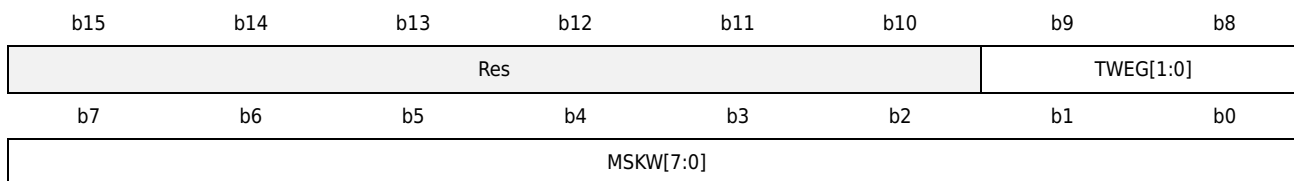
位/位域	标记	位名	功能	读写
b31~b16	CTWP15~0	窗口信号极性	Blank-Window为电平模式 (CMP_OCR.BWMD=0) 时设定用于MASK比较器输出的窗口信号的有效电平。(仅对应的CTWS为1时该位的设定有效) 0: 窗口信号为高电平时比较器输出关闭 1: 窗口信号为低电平时比较器输出关闭	RW
b15~b0	CTWS15~0	窗口信号使能	Blank-Window有效 (CMP_OCR.BWEN=1) 时选择用于MASK比较器输出的窗口信号。(各bit对应的定时器PWM信号见表 15-5) 0: 不选择定时器PWM输出作为窗口信号 1: 选择定时器PWM输出作为窗口信号	RW

表 15-5 定时器窗口 PWM 一览表

CTWS bit	CMP1	CMP2	CMP3
15	TMR4_OWL	TMR4_OWL	TMR4_OWL
14	TMR4_OWH	TMR4_OWH	TMR4_OWH
13	TMR4_OVL	TMR4_OVL	TMR4_OVL
12	TMR4_OVH	TMR4_OVH	TMR4_OVH
11	TMR4_OUL	TMR4_OUL	TMR4_OUL
10	TMR4_OUH	TMR4_OUH	TMR4_OUH
9	TMR6_1_PWMA	TMR6_2_PWMA	TMR6_1_PWMB
8	HRPWM_PWMA_5_PRE	HRPWM_PWMA_6_PRE	HRPWM_PWMB_5_PRE
7	HRPWM_PWMA_3_PRE	HRPWM_PWMA_4_PRE	HRPWM_PWMB_3_PRE
6	HRPWM_PWMA_1_PRE	HRPWM_PWMA_2_PRE	HRPWM_PWMB_1_PRE
5	TMRA_2_PWM3	TMRA_4_PWM3	TMRA_3_PWM3
4	TMRA_2_PWM2	TMRA_4_PWM2	TMRA_3_PWM2
3	TMRA_2_PWM1	TMRA_4_PWM1	TMRA_3_PWM1
2	TMRA_1_PWM3	TMRA_3_PWM3	TMRA_1_PWM3
1	TMRA_1_PWM2	TMRA_3_PWM2	TMRA_1_PWM2
0	TMRA_1_PWM1	TMRA_3_PWM1	TMRA_1_PWM1

15.5.7 比较器输出空白窗口设定寄存器 2 (CMPx_BWSR2, x=1~3)

复位值: 0x0000



位/位域	标记	位名	功能	读写
b15~b10	Res	保留位	读出时为“0”，写入时写“0”	RW
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MASK比较器输出的窗口信号的有效边沿。 00: 无有效边沿 01: 窗口信号的上升沿处比较器输出关闭 10: 窗口信号的下降沿处比较器输出关闭 11: 窗口信号的变化沿处比较器输出关闭	
b7~b0	MSKW[7:0]	Blank-Window有效宽度	Blank-Window为边沿模式 (CMP_OCR.BWMD=1) 时设定MASK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。	RW

15.5.8 比较器输入扫描控制寄存器 (CMPx_SCCR, x=1~3)

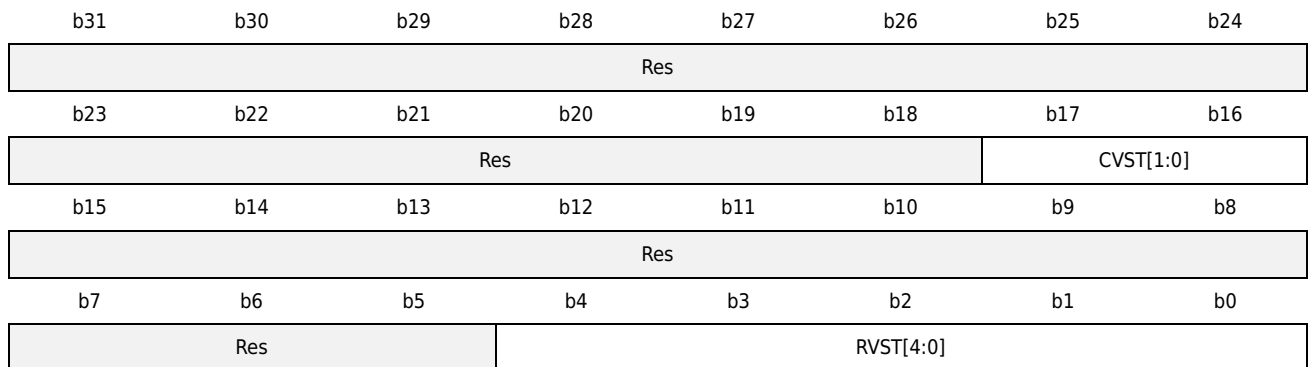
复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res		SSTB[5:0]					
b23	b22	b21	b20	b19	b18	b17	b16
SPRD[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res			SISL[4:0]				

位/位域	标记	位名	功能	读写
b31~b30	Res	保留位	读出时为“0”，写入时写“0”	RW
b29~b24	SSTB[5:0]	输出稳定时间	输入扫描时切换输入后CMP的输出稳定时间。 CMP输出稳定时间=CMPCLK周期×SSTB设定值 注意: 1.请结合CMPCLK的周期设定INP切换稳定时间大于电气特性中记载的“输入通道切换稳定时间”。 2.如需改写SSTB,请先停止扫描。	RW
b23~b16	SPRD[7:0]	输入切换周期	输入扫描时切换输入的时间间隔,即扫描周期。 扫描周期=CMPCLK周期×SPRD设定值 注意: 1.为了保证动作正确,设定SPRD时请确认: 扫描周期>输出稳定时间+滤波采样周期×4+CMPCLK周期×8 2.如需改写SPRD,请先停止扫描。	RW
b15~b5	Res	保留位	读出时为“0”，写入时写“0”	RW
b4~b0	SISL[4:0]	输入选择	正端输入扫描时, b1~b0分别对应CMP的正端输入INP2~INP1; 负端输入扫描时, b4~b0分别对应CMP的正端输入INM5~INM1; 0: 输入不作为扫描对象 1: 输入作为扫描对象 注意: 如需改写SISL,请先停止扫描。	RW

15.5.9 比较器输入扫描监视寄存器 (CMPx_SCMR, x=1~3)

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b18	Res	保留位	读出时为“0”，写入时写“0”	RW
b17~b16	CVST[1:0]	CMP正端输入状态	本寄存器为只读寄存器，用于在输入扫描时监视CMP的输入状态，b17~b16分别对应CMP的正端输入INP2~INP1。 0: 不在比较INPn (n=1~2)。 1: 正在比较INPn (n=1~2)。	R
b15~b5	Res	保留位	读出时为“0”，写入时写“0”	RW
b4~b0	RVST[4:0]	CMP负端输入状态	本寄存器为只读寄存器，用于在输入扫描时监视CMP的输入状态，b4~b0分别对应CMP的负端输入INM5~INM1。 0: 不在比较INMn (n=1~5)。 1: 正在比较INMn (n=1~5)。	R

15.6 注意事项

15.6.1 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, 2 共用一个模块停止控制位 PWC_FCG3.bit8, CMP3 模块停止控制位为 PWC_FCG3.bit9。CMP1~3 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【动作模式与低功耗模式】章节。

15.6.2 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 CMP_MDR 寄存器的 CENA 位清 0。

15.6.3 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 CMP_MDR 寄存器的 CENA 位清 0。

16 模数转换模块 (ADC)

16.1 概述

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 支持 16 个通道，单元 2 支持 12 个通道，单元 3 支持 10 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。

16.2 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，转换 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
 - 过采样功能
 - 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 22 个外部模拟输入 (3 个内部模拟通道)
 - 1 个内部模拟输入：内部基准电压
- 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
- 序列 A 多重数据缓冲模式
- 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

16.3 功能说明

16.3.1 ADC 系统框图

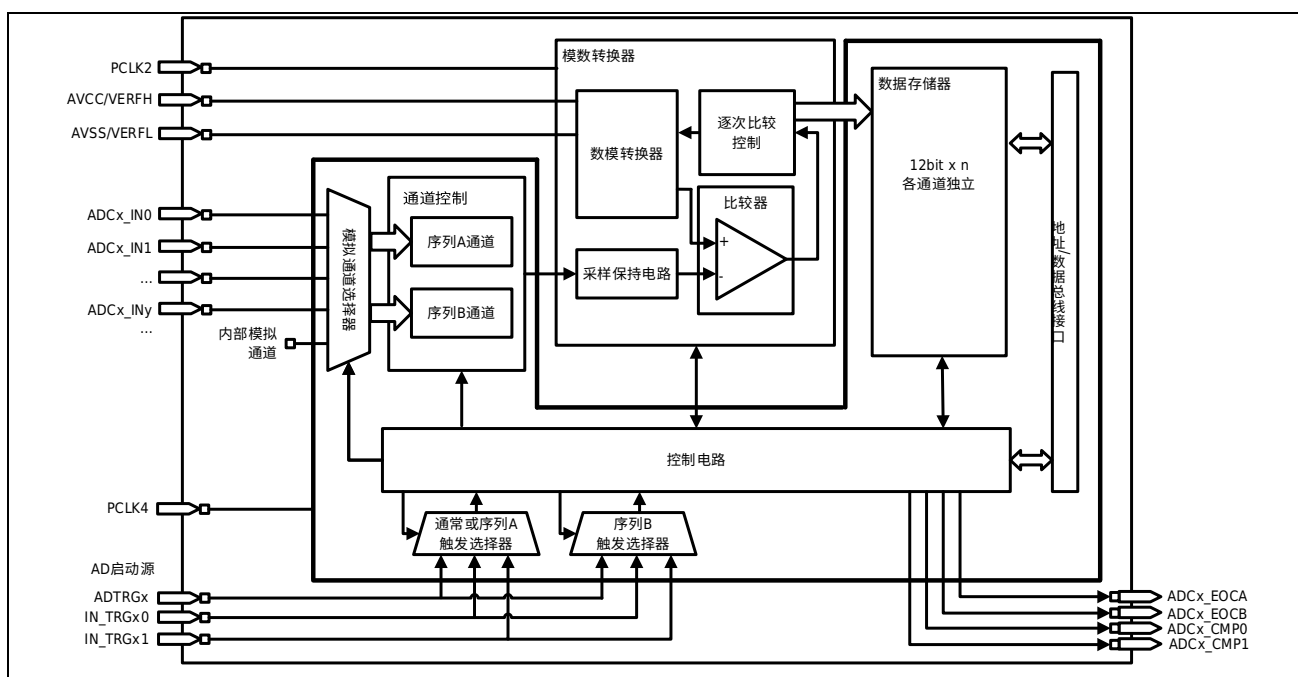


图 16-1 ADC 框图

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 16-1 各 ADC 单元规格

项目	单元1 (ADC1)	单元2 (ADC2)	单元3 (ADC3)
电源/基准电压	AVCC/VREFH		
	AVSS/VREFL		
模拟通道*1	CH0	ADC1_IN0	ADC3_IN0
	CH1	ADC1_IN1	ADC3_IN1
	CH2	ADC1_IN2	ADC3_IN2

项目		单元1 (ADC1)	单元2 (ADC2)	单元3 (ADC3)	
	CH3	ADC1_IN3	-	ADC3_IN3	
	CH4	ADC12_IN4	ADC12_IN4	-	
	CH5	ADC12_IN5	ADC12_IN5	-	
	CH6	ADC123_IN6	ADC123_IN6	ADC123_IN6	
	CH7	ADC123_IN7	ADC123_IN7	ADC123_IN7	
	CH8	ADC123_IN8	ADC123_IN8	ADC123_IN8	
	CH9	ADC123_IN9	ADC123_IN9	ADC123_IN9	
	CH10	ADC12_IN10	ADC12_IN10	ADC3_IN10	
	CH11	ADC12_IN11	ADC12_IN11	ADC3_IN11/ 内部模拟通道	
	CH12	ADC12_IN12	ADC12_IN12	-	
	CH13	ADC12_IN13	ADC12_IN13	-	
	CH14	ADC12_IN14	ADC12_IN14	-	
	CH15	ADC12_IN15/ 内部模拟通道	ADC12_IN15/ 内部模拟通道	-	
	硬件触发源	外部引脚	ADTRG1	ADTRG2	ADTRG3
		片内周边	IN_TRG10	IN_TRG20	IN_TRG30
IN_TRG11			IN_TRG21	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。

16.3.2 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。

16.3.3 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。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，具体参考寄存器描述。本章中若无特别说明，通道 n 或 CHn 均表示虚拟通道。

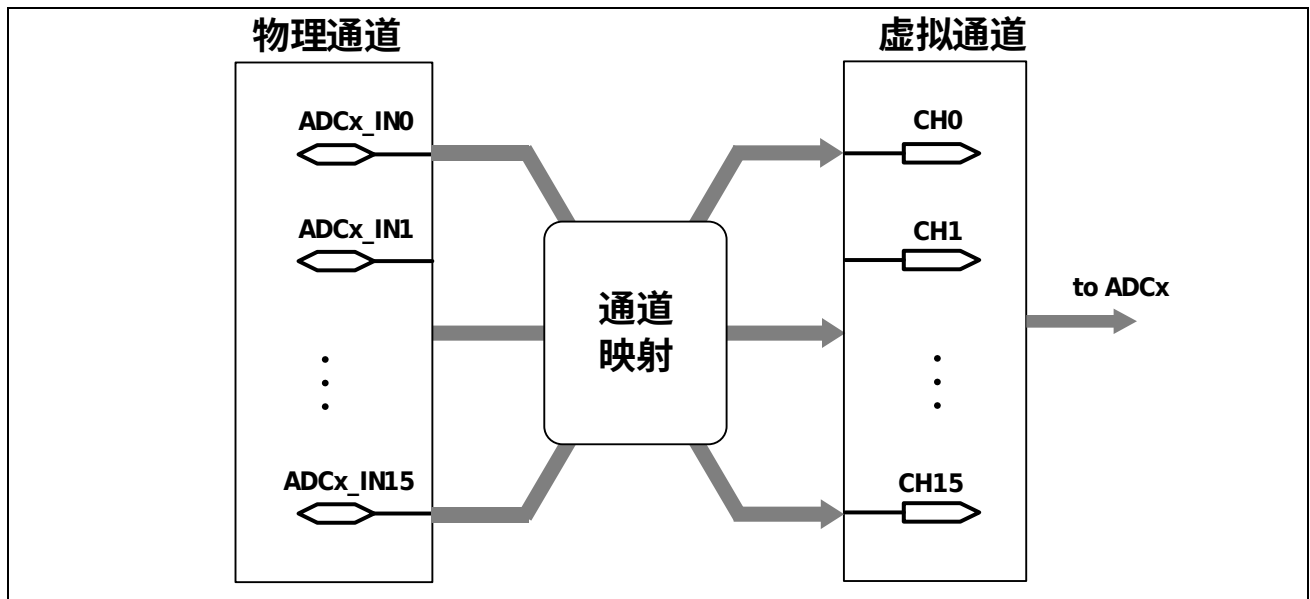


图 16-2 通道映射示意图

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，内部模拟输出与每个单元的最大外部输入通道 ADCx_INy (x=1 时 y=15, x=2 时 y=15, x=3 时 y=11) 共用一个 CH。设置 ADC_CHMUX 同时改变内部模拟输出和 ADCx_INy 与 CHn 的映射关系。当需要转换内部模拟通道或 ADCx_INy 时，需要先设置扩展通道寄存器 ADC_EXCHSELR，然后将 ADC_CHSELRA/ADC_CHSELB 中与之对应的虚拟通道位写 1。当 ADC_EXCHSELR 设置为 0x0 时，表示转换外部模拟输入，当 ADC_EXCHSELR 设置为 0x1 时，表示转换内部模拟通道。

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)】章节，设置相应的寄存器，允许内部模拟电压输出。

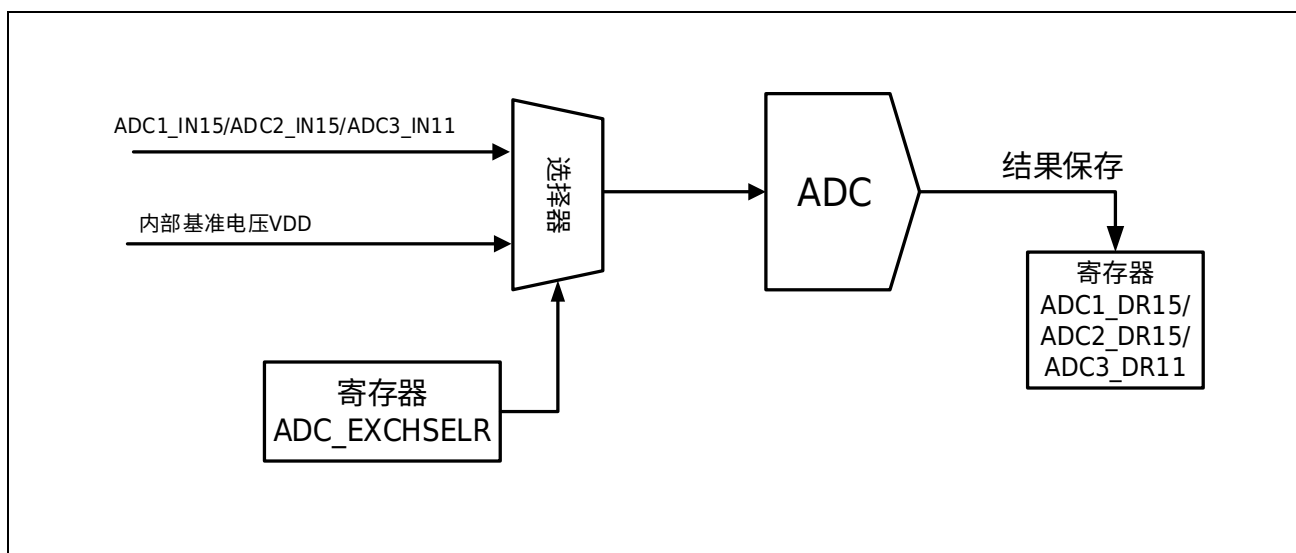


图 16-3 内部模拟通道选择

另外，大部分的物理通道可以输入到多个 ADC 模块单元，如 ADC123_IN6 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。

16.3.4 触发源选择

序列 A，序列 B 独立选择触发源。可选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。

16.3.5 序列 A 单次扫描模式

ADC 控制寄存器 ADC_CR0.MS[2:0] 设为 0b000 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.EOCAP 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。

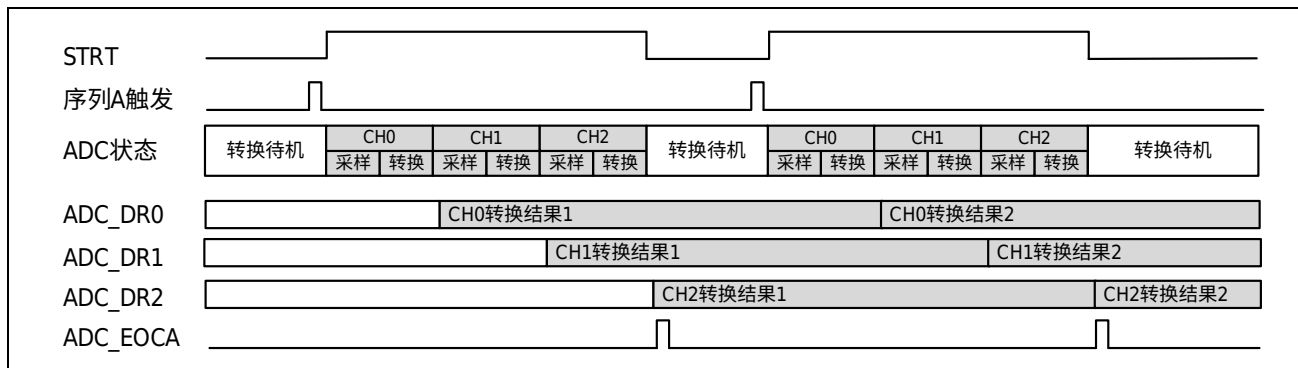


图 16-4 序列 A 单次扫描模式

序列 A 单次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[2:0] 设为 0b000 选择序列 A 单次扫描模式；
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。

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。

16.3.6 序列 A 连续扫描模式

ADC 控制寄存器 ADC_CR0.MS[2:0] 设为 0b001 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。

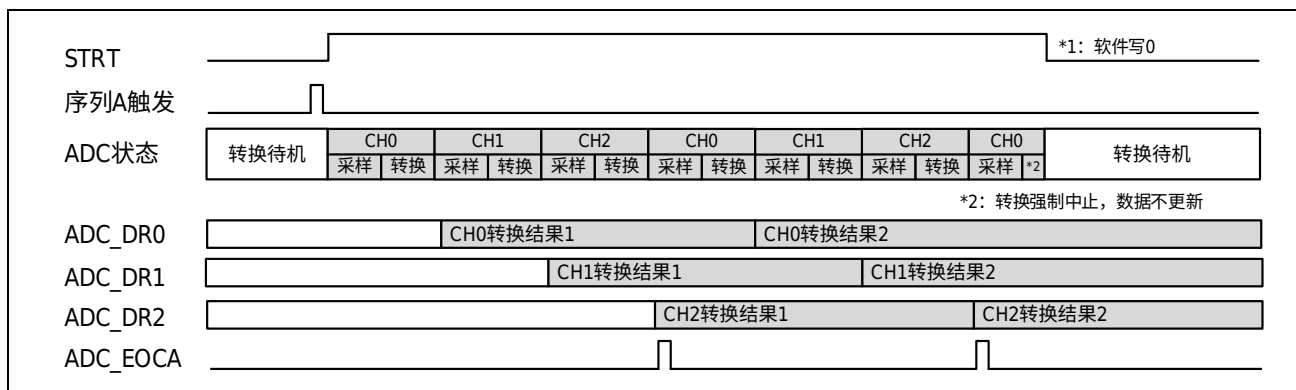


图 16-5 连续扫描

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[2:0] 设为 0b001 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 STRT 位写 0，并读 STRT 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。

16.3.7 序列 A 数据缓冲模式

ADC 控制寄存器 ADC_CR0.MS[2:0] 设为 0b100 选择序列 A 数据缓冲模式。

序列 A 数据缓冲模式与序列 A 单次扫描模式动作类似，不同点在于，数据缓冲模式在一个通道转换结束后不是直接开始下一个通道的转换，而是进入转换待机状态，直到启动条件再次发生才开始下一个通道的转换。

当所有通道转换结束时，序列 A 转换结束标志位 ADC_ISR.EOCALF 置 1，并产生序列 A 转换结束事件 ADC_EOACA，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。

该模式下，可以用寄存器 ADC_CHSELRA 选择多个虚拟通道，再用通道映射功能，将这些虚拟通道映射到同一个物理通道，这样当所有通道转换结束后，可以依次从各个虚拟通道对应的 ADC_DR 寄存器中读取转换数据。

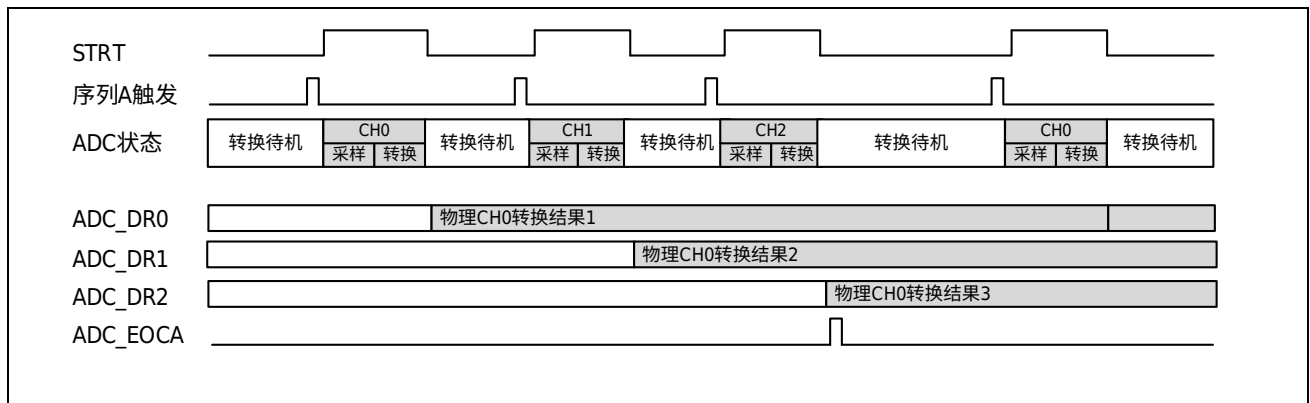


图 16-6 数据缓冲模式

序列 A 数据缓冲模式的软件流程：

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[2:0] 设为 0b100 选择序列 A 数据缓冲模式；
3. 设置序列 A 通道选择寄存器 ADC_CHSELRA；
4. 设置通道映射控制寄存器 ADC_CHMUXR；
5. 设置采样时间寄存器 ADC_SSTR；
6. 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；
7. 查询单次转换结束 ADC_STR.STRT=0；
8. 重复步骤 6~7 直到所有通道转换结束；
9. 查询序列 A 转换结束标志位 EOCALF；
10. 读取各通道数据寄存器 ADC_DR；
11. 写 0 清除 EOCALF 标志位，为下次转换准备。

上面的步骤 9~11 的查询方式也可以替换成中断方式，利用 ADC_EOACA 中断来处理转换数据。或者利用 ADC_EOACA 事件启动 DMA 读取数据。

16.3.8 双序列扫描模式

ADC 控制寄存器 ADC_CR0.MS[2:0] 设为 0b010 或 0b011 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[2:0]=0b010 时，序列 A 和 B 等效于两个独立的单次扫描序列。MS[2:0]=0b011 序列 A 为连续扫描模式，B 为单次扫描模式。MS[2:0]=110b 序列 A 为数据缓冲模式，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.EOCAL 置 1，并产生序列 A 转换结束事件 ADC_EOCA，若 ADC_ISR.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.EOCBL 置 1，并产生序列 B 转换结束事件 ADC_EOCB，若 ADC_ISR.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 16-2 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	

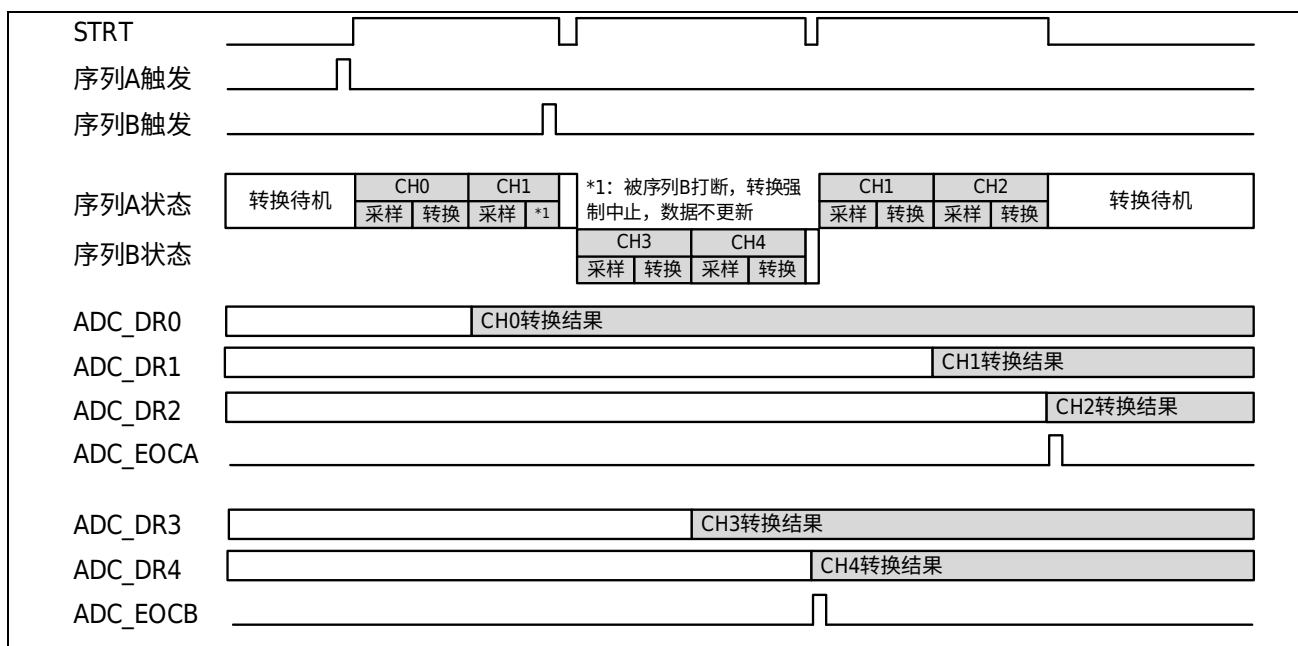


图 16-7 双序列扫描模式 (序列 A 从被中断通道重新启动)

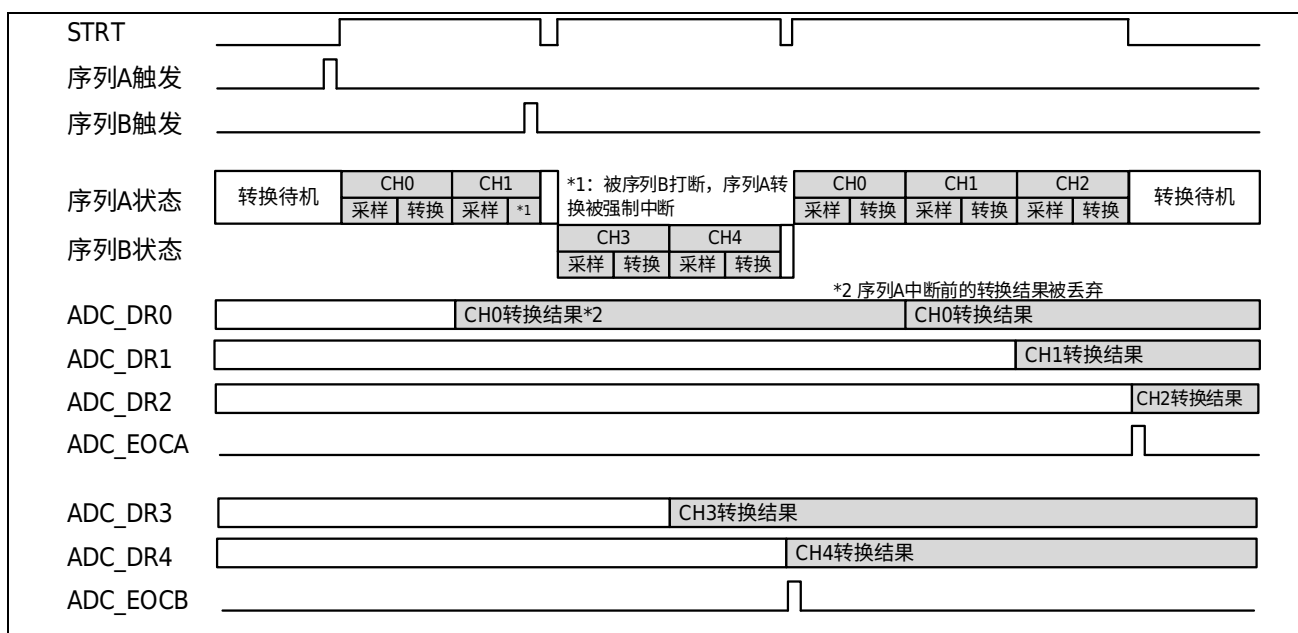


图 16-8 双序列扫描模式 (序列 A 从第一个通道重新启动)

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[2:0]设为 0b010 或 0b011 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。

16.3.9 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。

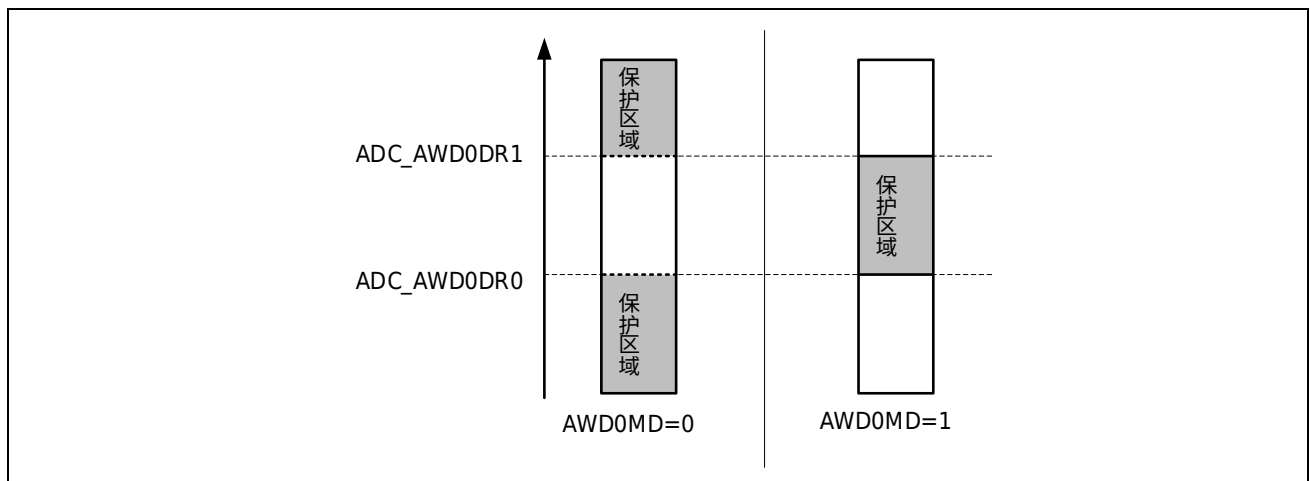


图 16-9 模拟看门狗保护区域 (比较模式)

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。

16.3.10 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。

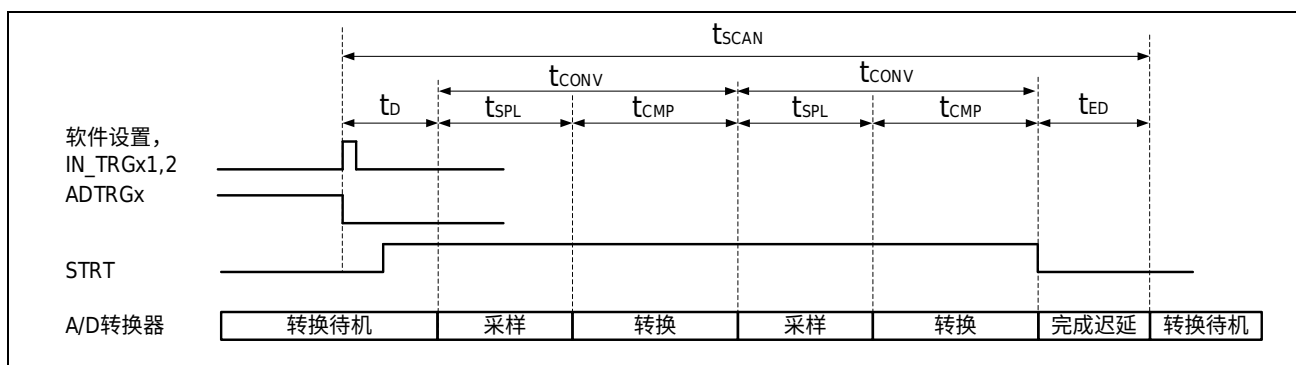


图 16-10 ADC 转换时间

表 16-3 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 处理时间	ADC空闲中, 启动转换	4 PCLK2	3 PCLK4 + 4 PCLK2 + 1 PCLK4_SYNC	2 PCLK4 + 4 PCLK2	3 PCLK2
		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}	逐次转换时间	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 _D	最小连续触发时间间隔		$\sum t_{CONV} + 6 PCLK2$			

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_SCFGR 设置的时钟), 此时的 PCLK4, PCLK2 相同, 均为异步的 PLL 时钟。

16.3.11 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。

16.3.12 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_AVCHSELR 选择任意一个或多个需要平均的通道。

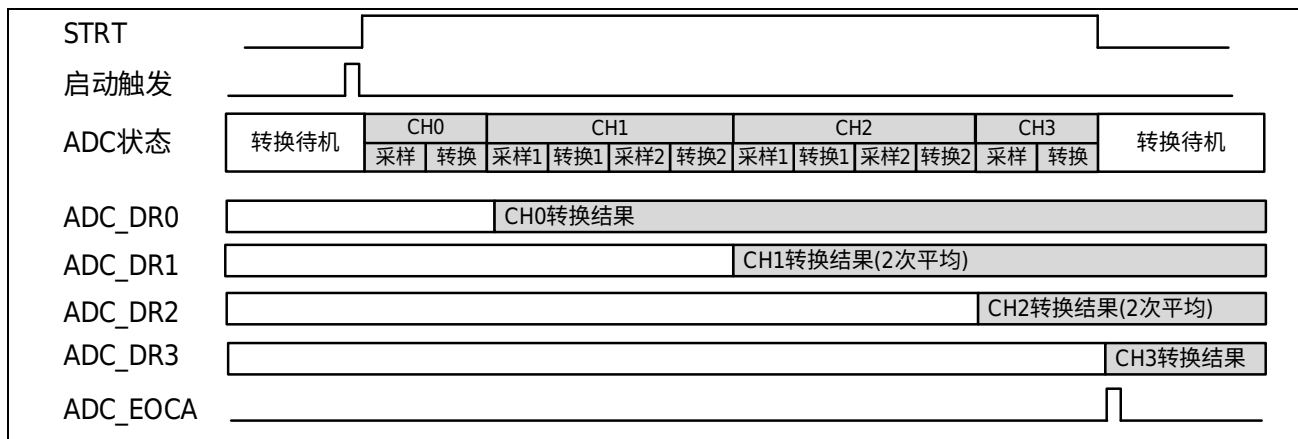


图 16-11 平均功能有效时的转换动作

图 16-11 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。

16.3.13 过采样

通过设置寄存器 ADC_CR2.OVSMOD 为 1 和设置寄存器 ADC_AVCHSELR 选择任意一个或多个需要过采样的通道，使 AD 进入过采样模式，在该模式下，通过连续多次采样转换和结果移位，得到 16bit 精度的转换结果。

通过以下公式计算得到最终的转换结果。

$$AD \text{ 过采样结果} = \frac{1}{2^M} \times \sum_{n=1}^{n=N} AD \text{ 第 } n \text{ 次转换结果}$$

采样率 N 通过寄存器 ADC_CR0.AVCNT[2:0]设置，可以设置 2 到 256 次采样。

除数系列 M 通过寄存器 ADC_CR2.OVSS[3:0]设置，最大可以设置向右移位 8 位。

16.3.14 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.TRGSELA[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.STRT 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。

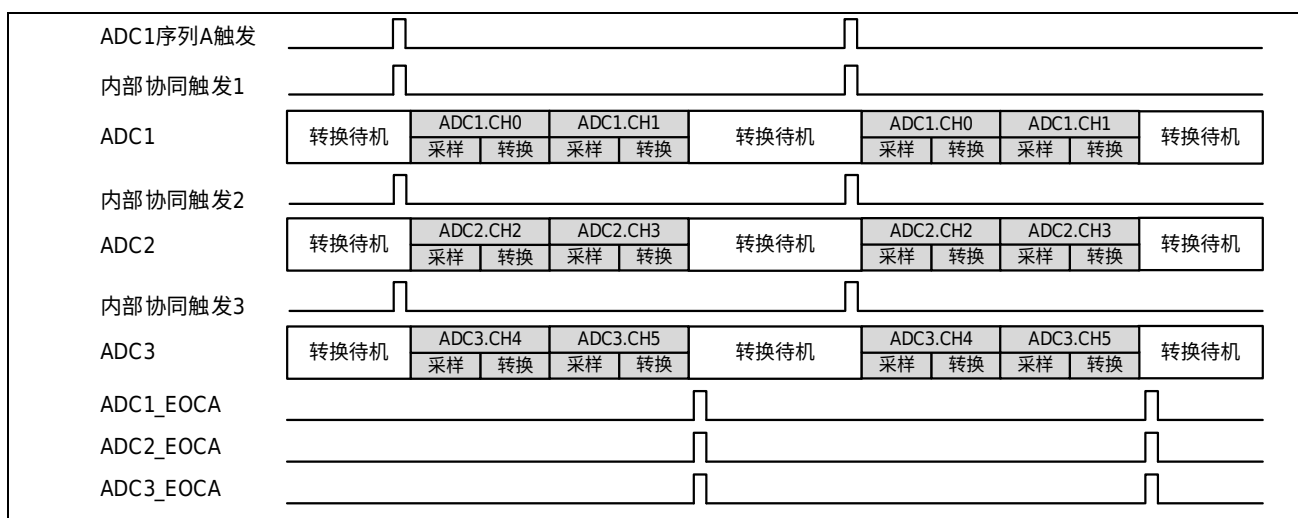


图 16-12 单次并行触发模式 (三 ADC)

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_SYNCRR.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[2:0]为 0b000: 序列 A 单次扫描模式, 或 0b001: 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[2:0], 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[2:0], 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_SYNCRR.SYNCMD[2:0], 写 0b010: ADC1, ADC2 两 ADC 协同工作。或者写 0b011: ADC1, ADC2, ADC3 三 ADC 协同工作;

6. 协同工作许可寄存器 ADC_SYNCRR.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。

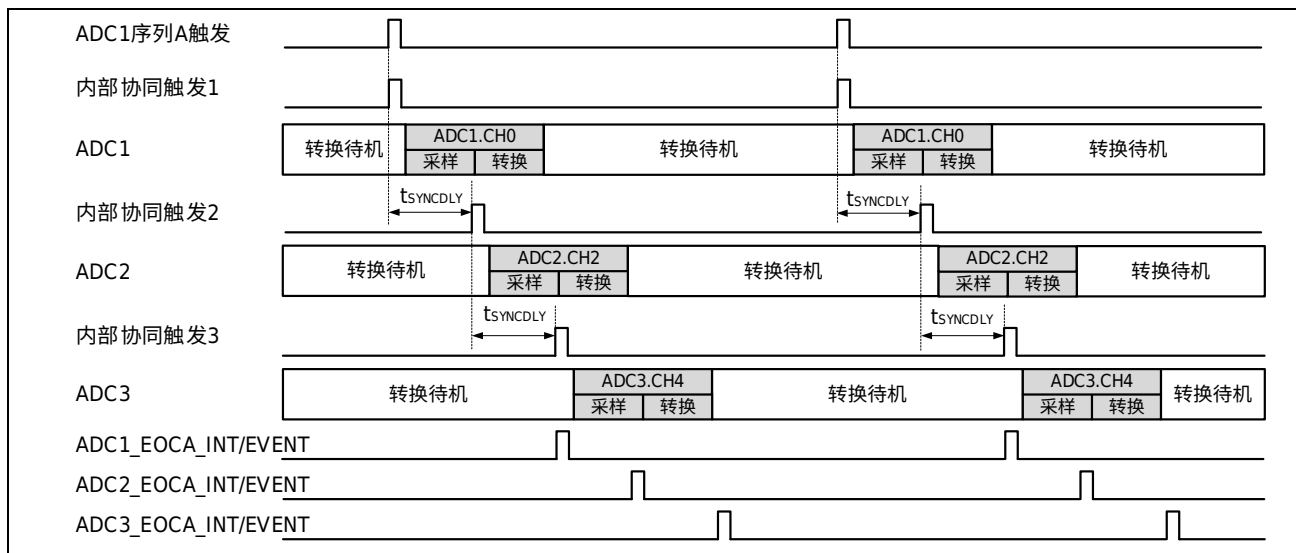


图 16-13 单次延迟触发模式 (三 ADC)

注意:

1. 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。
2. 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_{SYNC_DLY} 与通道开采样时间 t_{SPL} 需满足： $t_{SYNC_DLY} > t_{SPL}$ 。

本模式的软件设定流程如下：

1. 协同工作许可寄存器 ADC_SYNC_CR.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_SYNC_CR.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_SYNC_CR.SYNCMD[2:0]，写 0b000：ADC1，ADC2 两 ADC 协同工作。或者写 0b001：ADC1，ADC2，ADC3 三 ADC 协同工作；
5. 协同工作许可寄存器 ADC_SYNC_CR.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。

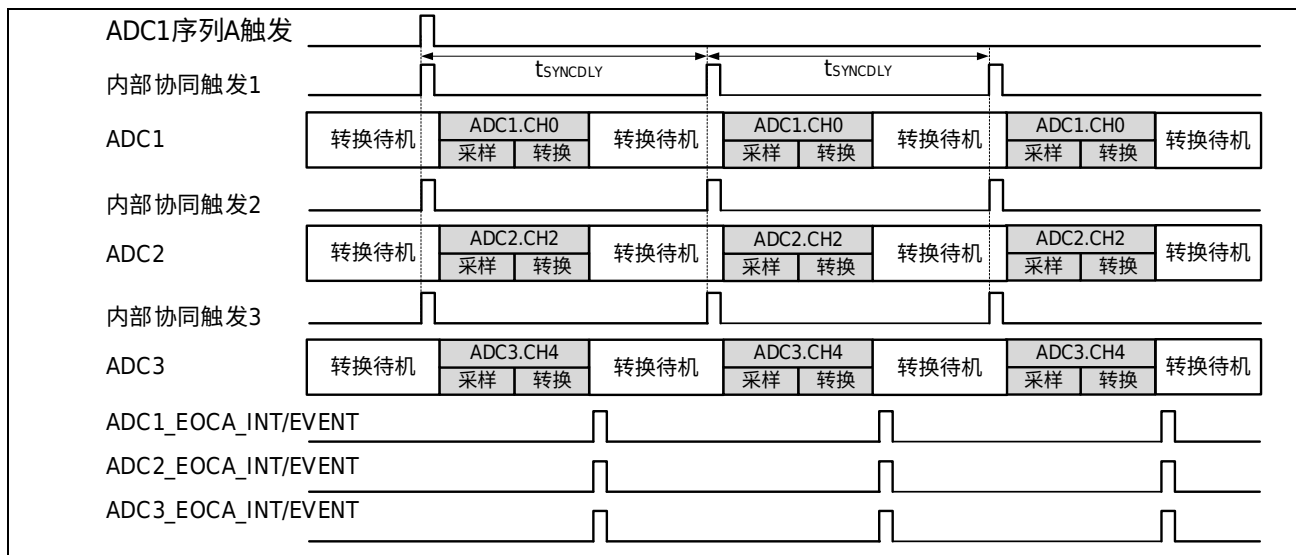


图 16-14 循环并行触发模式 (三 ADC)

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_SYNCRR.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[2:0]` 设置为 0b000：序列 A 单次扫描模式；
3. 设置协同模式控制寄存器 `ADC_SYNCRR.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_SYNCRR.SYNCMD[2:0]`，写 0b110：ADC1，ADC2 两 ADC 协同工作。或者写 0b111：ADC1，ADC2，ADC3 三 ADC 协同工作；
5. 协同工作许可寄存器 `ADC_SYNCRR.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。

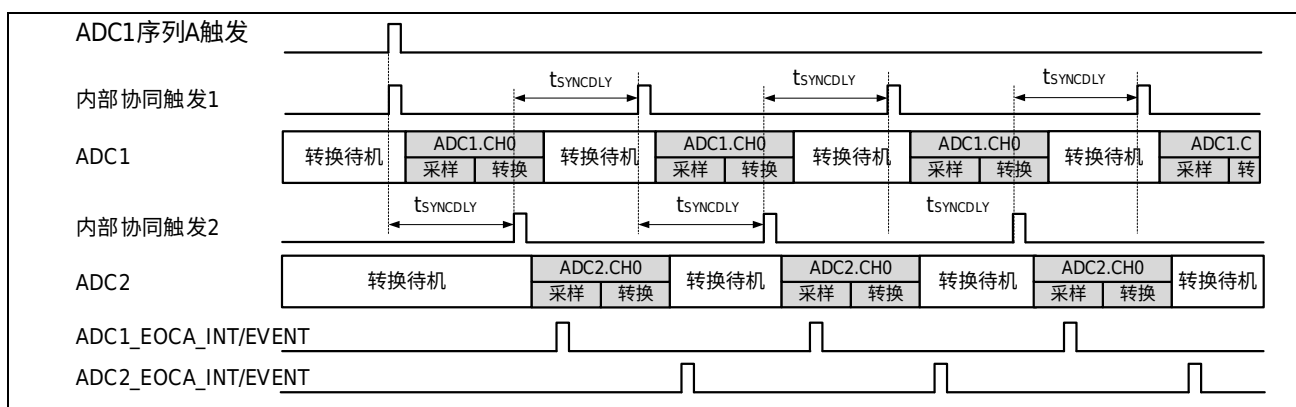


图 16-15 循环延迟触发模式 (两 ADC)

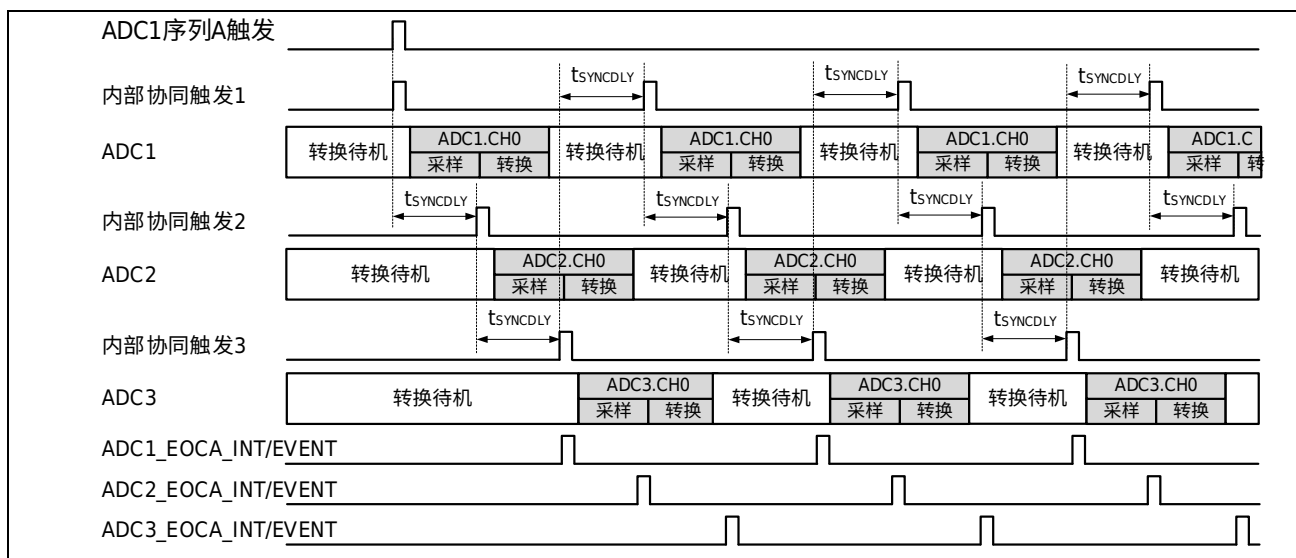


图 16-16 循环延迟触发模式 (三 ADC)

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.SYNCDELAY[7:0]，设置每次触发的延迟；

4. 设置协同模式控制寄存器 ADC_SYNCRR.SYNCMD[2:0]，写 0b100：ADC1，ADC2 两 ADC 协同工作。或者写 0b101：ADC1，ADC2，ADC3 三 ADC 协同工作；
5. 协同工作许可寄存器 ADC_SYNCRR.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。

16.4 中断和事件

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 控制器 (DMA)】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。

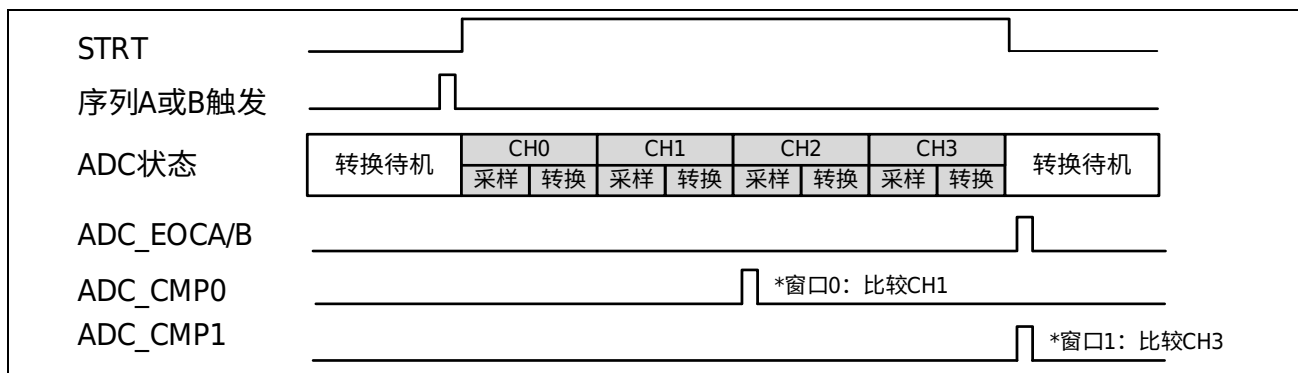


图 16-17 ADC 中断和事件输出时序

16.5 寄存器描述

表 16-4 ADC 基地址

名称	基地址	描述
ADC1	0x4004 0000	ADC1基地址
ADC2	0x4004 0400	ADC2基地址
ADC3	0x4004 0800	ADC3基地址

表 16-5 ADC 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
ADC_STR	ADC启动寄存器	0x00	8	0x00
ADC_CR0	ADC控制寄存器0	0x02	16	0x0000
ADC_CR1	ADC控制寄存器1	0x04	16	0x0000
ADC_CR2	ADC控制寄存器2	0x06	16	0x0000
ADC_TRGSR	ADC转换开始触发寄存器	0x0a	16	0x0000 0000
ADC_CHSELRA	ADC通道选择寄存器A	0x0c	32	0x0000 0000
ADC_CHSELRB	ADC通道选择寄存器B	0x10	32	0x0000 0000
ADC_AVCHSELR	ADC平均通道选择寄存器	0x14	32	0x0000 0000
ADC_EXCHSELR	ADC扩展通道选择寄存器	0x18	8	0x00
ADC_SSTRx ⁽¹⁾	ADC采样周期寄存器	0x20+x	8	0x0B
ADC_CHMUXR0	ADC通道映射控制寄存器0	0x38	16	0x3210
ADC_CHMUXR1	ADC通道映射控制寄存器1	0x3A	16	0x7654
ADC_CHMUXR2	ADC通道映射控制寄存器2	0x3C	16	0xBA98
ADC_CHMUXR3	ADC通道映射控制寄存器3	0x3E	16	0xFEDC
ADC_ISR	ADC中断状态寄存器	0x44	8	0x00
ADC_ICR	ADC中断许可寄存器	0x45	8	0x03
ADC_ISCLRR	ADC中断状态复位寄存器	0x46	8	0x00
ADC_SYNCCR	ADC协同模式控制寄存器	0x4C	16	0x0C00
ADC_DRx ⁽¹⁾	ADC数据寄存器	0x50+2*x	16	0x0000
ADC_AWDCR	模拟看门狗控制寄存器	0xA0	16	0x0000
ADC_AWDSR	模拟看门狗状态寄存器	0xA2	8	0x00
ADC_AWDSCLR	模拟看门狗状态复位寄存器	0xA3	8	0x00
ADC_AWD0DR0	模拟看门狗窗口0阈值寄存器0	0xA4	16	0x0000
ADC_AWD0DR1	模拟看门狗窗口0阈值寄存器1	0xA6	16	0xFFFF
ADC_AWD0CHSR	模拟看门狗窗口0比较通道选择寄存器	0xA8	8	0x00
ADC_AWD1DR0	模拟看门狗窗口1阈值寄存器0	0xAC	16	0x0000
ADC_AWD1DR1	模拟看门狗窗口1阈值寄存器1	0xAE	16	0xFFFF
ADC_AWD1CHSR	模拟看门狗窗口1比较通道选择寄存器	0xB0	8	0x00

注:ADC_SSTRx和ADC_DRx中的x表示通道,ADC1中x=0~15,ADC2中x=4~15,ADC3中x=0~3,6~11。

16.5.1 ADC 启动寄存器 (ADC_STR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res							STRT

位/位域	标记	位名	功能	RW
b7~b1	Res	保留位	读出时为0，写入时写0	RW
b0	STRT	AD转换开始	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: 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	RW

16.5.2 ADC 控制寄存器 0 (ADC_CR0)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
Res					AVCNT[2:0]		
b7	b6	b5	b4	b3	b2	b1	b0
DFMT	CLREN	ACCSEL[1:0]		Res	MS[2:0]		

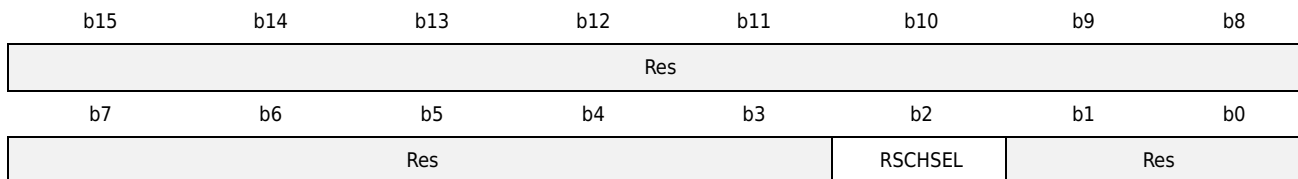
位/位域	标记	位名	功能	RW
b15~b11	Res	保留位	读出时为0, 写入时写0	RW
b10~b8	AVCNT[2:0]	次数选择	000: 连续转换2次平均 001: 连续转换4次平均 010: 连续转换8次平均 011: 连续转换16次平均 100: 连续转换32次平均 101: 连续转换64次平均 110: 连续转换128次平均 111: 连续转换256次平均	RW
b7	DFMT	数据格式	0: 转换数据右对齐 1: 转换数据左对齐 注: 过采样模式下该bit设置无效	RW
b6	CLREN	数据寄存器自动清除	0: 自动清除禁止 1: 自动清除许可 注意: CLREN位设定后, 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。	RW
b5~b4	ACCSEL[1:0]	分辨率选择	00: 12位分辨率 01: 10位分辨率 10: 8位分辨率 11: 禁止设定 注: 过采样模式下, 该bit请设置为00.	RW
b3	Res	保留位	读出时为0, 写入时写0	RW
b2~b0	MS[2:0]	模式选择	000: 序列A单次扫描模式, 序列B无效 001: 序列A连续扫描模式, 序列B无效 010: 序列A单次扫描模式, 序列B单次扫描模式 011: 序列A连续扫描模式, 序列B单次扫描模式 100: 序列A数据缓冲模式, 序列B无效 110: 序列A数据缓冲模式, 序列B单次扫描模式 其他设定值禁止	RW

注意:

请在 ADC_STR.STRT 为 0 时设置本寄存器。

16.5.3 ADC 控制寄存器 1 (ADC_CR1)

复位值: 0x0000



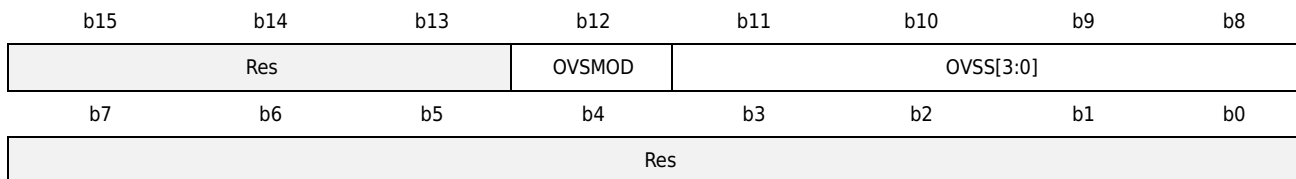
位/位域	标记	位名	功能	RW
b15~b3	Res	保留位	读出时为0, 写入时写0	RW
b2	RSCHSEL	序列A重启通道选择	0: 被序列B中断后, 序列A重启时从被中断通道开始继续扫描 1: 被序列B中断后, 序列A重启时从被第一个通道开始重新扫描	RW
b1~b0	Res	保留位	读出时为0, 写入时写0	RW

注意:

请在 `ADC_STR.STRT` 为 0 时设置本寄存器。

16.5.4 ADC 控制寄存器 2 (ADC_CR2)

复位值: 0x0000



位/位域	标记	位名	功能	RW
b15~b13	Res	保留位	读出时为0, 写入时写0	RW
b12	OVSMOD	过采样模式	0: 普通模式 1: 过采样模式	RW
b11~b8	OVSS[3:0]	移位选择	0000: 不移位	RW
			0001: 向右移1位	
			0010: 向右移2位	
			0011: 向右移3位	
			0100: 向右移4位	
			0101: 向右移5位	
			0110: 向右移6位	
			0111: 向右移7位	
			1000: 向右移8位	
b7~b0	Res	保留位	读出时为0, 写入时写0	RW

注意:

1. 请在 ADC_STR.STRT 为 0 时设置本寄存器。
2. 请 16bit 访问。

16.5.5 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	Res					TRGSELB[1:0]	
b7	b6	b5	b4	b3	b2	b1	b0
TRGENA	Res					TRGSELA[1:0]	

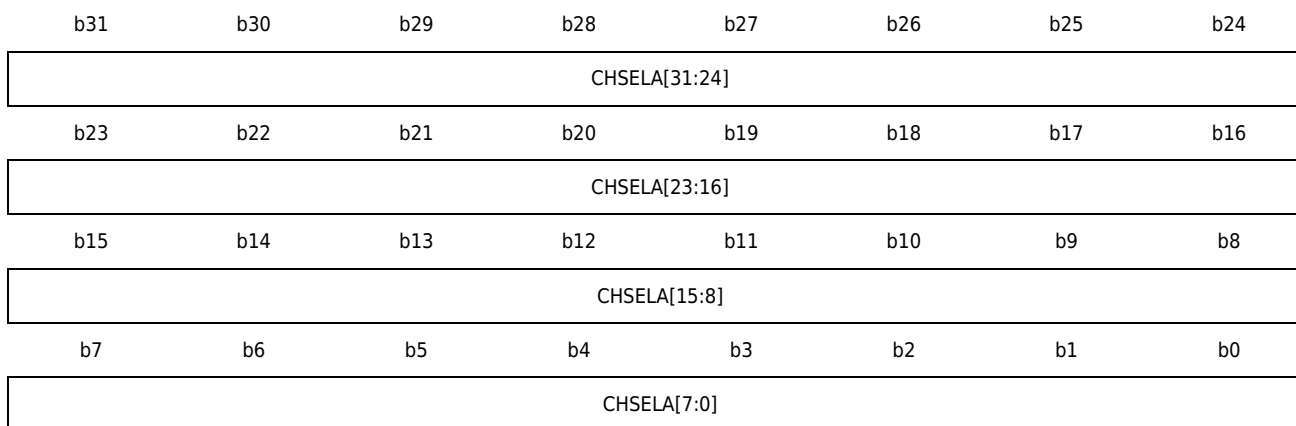
位/位域	标记	位名	功能	RW
b15	TRGENB	序列B触发使能	0: 序列B片内或外部引脚触发禁止 1: 序列B片内或外部引脚触发许可 注意: 选择外部引脚触发有效。如果ADTRGx由“High”变为“Low”，检测到下降沿，则扫描转换开始，请保持“Low”1.5*PCLK4周期以上。	RW
b14~b10	Res	保留位	读出时为0，写入时写0	RW
b9~b8	TRGSELB[1:0]	序列B触发条件选择	在序列B有效模式下 (ADC_CR0.MS[1]=1)，作为序列B的触发条件 00: ADTRGx 01: IN_TRGx0 10: IN_TRGx1 11: IN_TRGx0 + 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_{SCAN} ，若小于则触发无效。	RW
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。	RW
b6~b2	Res	保留位	读出时为0，写入时写0	RW
b1~b0	TRGSELA[1:0]	序列A触发条件选择	序列A的触发条件。 00: ADTRGx (x=1~3, 代表ADC单元编号) 01: IN_TRGx0 10: IN_TRGx1 11: IN_TRGx0 + IN_TRGx1 注意: ADC空闲中对ADC_STR.STRT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_{SCAN} ，若小于则触发无效。	RW

注意:

在 ADC_STR.STRT 为 0 时设置本寄存器。

16.5.6 ADC 通道选择寄存器 A (ADC_CHSELRA)

复位值: 0x0000 0000



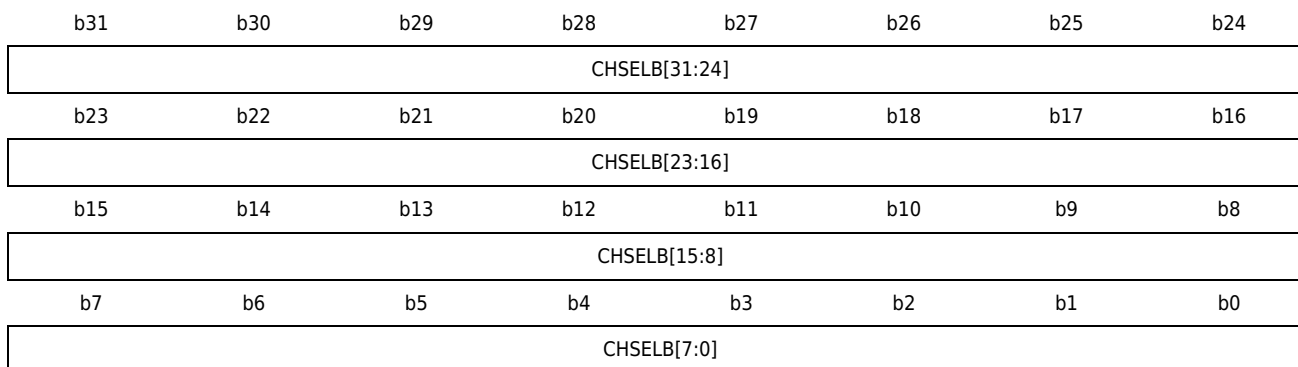
位/位域	标记	位名	功能	RW
			序列A的通道选择，每一位代表一个通道，CHSELA[x]代表通道CHx，可选择任意组合。	
b31~b0	CHSELA[31:0]	转换通道选择	0: 未选择对应通道 1: 选择对应通道 不存在通道的对应位为保留位，读出时为0，写入时写0。 注意： 请不要在序列A和序列B中选择相同的通道。	RW

注意：

请在 *ADC_STR.STRT* 为 0 时设置本寄存器。

16.5.7 ADC 通道选择寄存器 B (ADC_CHSELRB)

复位值: 0x0000 0000



位/位域	标记	位名	功能	RW
			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。 0: 未选择对应通道 1: 选择对应通道	RW
b31~b0	CHSELB[31:0]	转换通道选择	不存在通道的对应位为保留位，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。	RW

注意：

请在 *ADC_STR.STRT* 为 0 时设置本寄存器。

16.5.8 ADC 平均通道选择寄存器 (ADC_AVCHSELR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
AVCHSEL[31:24]							
b23	b22	b21	b20	b19	b18	b17	b16
AVCHSEL[23:16]							
b15	b14	b13	b12	b11	b10	b9	b8
AVCHSEL[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
AVCHSEL[7:0]							

位/位域	标记	位名	功能	RW
b31~b0	AVCHSEL[31:0]	平均通道选择	每一位代表一个通道，AVCHSEL[x]代表通道CHx，可选择任意组合。 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保留位，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。	RW

注意:

请在 ADC_STR.STRT 为 0 时设置本寄存器。

16.5.9 ADC 扩展通道选择寄存器 (ADC_EXCHSELR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res							EXCHSEL

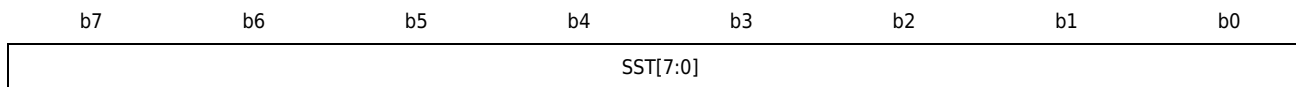
位/位域	标记	位名	功能	RW
b7~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
b0	EXCHSEL	扩展通道选择	ADCx_IN15与片内模拟通道的选择 0: 选择外部引脚ADC1_IN15或ADC2_IN7或ADC3_IN11 1: 选择内部模拟通道，内部模拟源的设置请参考【电源控制 (PWC)】说明	RW

注意:

请在 ADC_STR.STRT 为 0 时设置本寄存器。

16.5.10 ADC 采样状态寄存器 (ADC_SSTRx)

复位值: 0x0B



位/位域	标记	位名	功能	RW
b7~b0	SST[7:0]	采样周期数	<p>采样周期数可被设为5到255个周期。 通道CH0~15分别由ADC_SSTRx, x=0~15来设置。</p> <p>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_{AIN}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。</p> <p>$SST \geq (R_{AIN} + R_{ADC}) * C_{ADC} * \ln(2^{N+2}) * f_{ADC} + 1$ 其中: R_{AIN}表示外部输入阻抗 (Ω), R_{ADC}表示内部采样开关电阻 (Ω), C_{ADC}表示内部采样和保持电容 (F), N表示AD分辨率 (12/10/8), f_{ADC}表示PCLK2频率 (Hz)。具体参考对应数据手册电气特性相关说明。</p>	RW

注意:

1. 请在 ADC_STR.STRT 为 0 时设置本寄存器。
2. 内部基准电压的采样时间不要小于 $1\mu s$ 。
3. ADC_SSTRx 中的“x”表示通道, ADC1 中 x=0~15, ADC2 中 x=4~15, ADC3 中 x=0~3, 6~11。

16.5.11 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/位域	标记	位名	功能	RW																																																																				
			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:																																																																					
			<table border="1"> <thead> <tr> <th>设定值</th> <th>ADC1 映射对象</th> <th>ADC2 映射对象</th> <th>ADC3 映射对象</th> </tr> </thead> <tbody> <tr><td>0x0</td><td>ADC1_IN0</td><td>-</td><td>ADC3_IN0</td></tr> <tr><td>0x1</td><td>ADC1_IN1</td><td>-</td><td>ADC3_IN1</td></tr> <tr><td>0x2</td><td>ADC1_IN2</td><td>-</td><td>ADC3_IN2</td></tr> <tr><td>0x3</td><td>ADC1_IN3</td><td>-</td><td>ADC3_IN3</td></tr> <tr><td>0x4</td><td>ADC12_IN4</td><td>ADC12_IN4</td><td>-</td></tr> <tr><td>0x5</td><td>ADC12_IN5</td><td>ADC12_IN5</td><td>-</td></tr> <tr><td>0x6</td><td>ADC123_IN6</td><td>ADC123_IN6</td><td>ADC123_IN6</td></tr> <tr><td>0x7</td><td>ADC123_IN7</td><td>ADC123_IN7</td><td>ADC123_IN7</td></tr> <tr><td>0x8</td><td>ADC123_IN8</td><td>ADC123_IN8</td><td>ADC123_IN8</td></tr> <tr><td>0x9</td><td>ADC123_IN9</td><td>ADC123_IN9</td><td>ADC123_IN9</td></tr> <tr><td>0xA</td><td>ADC12_IN10</td><td>ADC12_IN10</td><td>ADC3_IN10</td></tr> <tr><td>0xB</td><td>ADC12_IN11</td><td>ADC12_IN11</td><td>ADC3_IN11/ 内部模拟通道</td></tr> <tr><td>0xC</td><td>ADC12_IN12</td><td>ADC12_IN12</td><td>-</td></tr> <tr><td>0xD</td><td>ADC12_IN13</td><td>ADC12_IN13</td><td>-</td></tr> <tr><td>0xE</td><td>ADC12_IN14</td><td>ADC12_IN14</td><td>-</td></tr> <tr><td>0xF</td><td>ADC12_IN15/ 内部模拟通道</td><td>ADC12_IN15/ 内部模拟通道</td><td>-</td></tr> </tbody> </table>	设定值	ADC1 映射对象	ADC2 映射对象	ADC3 映射对象	0x0	ADC1_IN0	-	ADC3_IN0	0x1	ADC1_IN1	-	ADC3_IN1	0x2	ADC1_IN2	-	ADC3_IN2	0x3	ADC1_IN3	-	ADC3_IN3	0x4	ADC12_IN4	ADC12_IN4	-	0x5	ADC12_IN5	ADC12_IN5	-	0x6	ADC123_IN6	ADC123_IN6	ADC123_IN6	0x7	ADC123_IN7	ADC123_IN7	ADC123_IN7	0x8	ADC123_IN8	ADC123_IN8	ADC123_IN8	0x9	ADC123_IN9	ADC123_IN9	ADC123_IN9	0xA	ADC12_IN10	ADC12_IN10	ADC3_IN10	0xB	ADC12_IN11	ADC12_IN11	ADC3_IN11/ 内部模拟通道	0xC	ADC12_IN12	ADC12_IN12	-	0xD	ADC12_IN13	ADC12_IN13	-	0xE	ADC12_IN14	ADC12_IN14	-	0xF	ADC12_IN15/ 内部模拟通道	ADC12_IN15/ 内部模拟通道	-	
设定值	ADC1 映射对象	ADC2 映射对象	ADC3 映射对象																																																																					
0x0	ADC1_IN0	-	ADC3_IN0																																																																					
0x1	ADC1_IN1	-	ADC3_IN1																																																																					
0x2	ADC1_IN2	-	ADC3_IN2																																																																					
0x3	ADC1_IN3	-	ADC3_IN3																																																																					
0x4	ADC12_IN4	ADC12_IN4	-																																																																					
0x5	ADC12_IN5	ADC12_IN5	-																																																																					
0x6	ADC123_IN6	ADC123_IN6	ADC123_IN6																																																																					
0x7	ADC123_IN7	ADC123_IN7	ADC123_IN7																																																																					
0x8	ADC123_IN8	ADC123_IN8	ADC123_IN8																																																																					
0x9	ADC123_IN9	ADC123_IN9	ADC123_IN9																																																																					
0xA	ADC12_IN10	ADC12_IN10	ADC3_IN10																																																																					
0xB	ADC12_IN11	ADC12_IN11	ADC3_IN11/ 内部模拟通道																																																																					
0xC	ADC12_IN12	ADC12_IN12	-																																																																					
0xD	ADC12_IN13	ADC12_IN13	-																																																																					
0xE	ADC12_IN14	ADC12_IN14	-																																																																					
0xF	ADC12_IN15/ 内部模拟通道	ADC12_IN15/ 内部模拟通道	-																																																																					
b15~b12 b11~b8 b7~b4 b3~b0	CHxMUX[3:0]	通道x映射选择 (x=0~15)		RW																																																																				
			注意: 请不要设置到不存在的模拟输入上。																																																																					

注意:

1. 请在 ADC_STR.STRT 为 0 时设置本寄存器。
2. 单元 2 仅配置了 ADC_CHMUXRn (n=1~3), 单元 3 仅配置了 ADC_CHMUXRn (n=0~2)。

16.5.12 ADC 中断状态寄存器 (ADC_ISR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res			SASTPDF	Res		EOCBF	EOCAF

位/位域	标记	位名	功能	RW
b7~b5	Res	保留位	读出时为0, 写入时写0	RW
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	Res	保留位	读出时为0, 写入时写0	RW
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

16.5.13 ADC 中断许可寄存器 (ADC_ICR)

复位值: 0x03

b7	b6	b5	b4	b3	b2	b1	b0
Res						EOCBIEN	EOCAIEN

位/位域	标记	位名	功能	RW
b7~b2	Res	保留位	读出时为0, 写入时写0	RW
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	RW
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	RW

16.5.14 ADC 中断状态复位寄存器 (ADC_ISCLRR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res			CLRSASTPDF	Res		CLREOCBF	CLREOCAF

位/位域	标记	位名	功能	RW
b7~b5	Res	保留位	读出时为0, 写入时写0	RW
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	W
b3~b2	Res	保留位	读出时为0, 写入时写0	RW
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	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。	W

16.5.15 ADC 协同模式控制寄存器 (ADC_SYNCRR)

复位值: 0x0c00

b15	b14	b13	b12	b11	b10	b9	b8
SYNCDLY[7:0]							
b7	b6	b5	b4	b3	b2	b1	b0
Res	SYNCMD[2]	SYNCMD[1]	SYNCMD[0]	Res			SYNCEN

位/位域	标记	位名	功能	RW
b15~b8	SYNCDLY[7:0]	同步延迟时间	<p>延迟触发模式时，两个ADC的启动延迟时间$t_{SYNCDLY}$。</p> <p>0x1表示 $t_{SYNCDLY} = 1 * PCLK2$，0xff表示$t_{SYNCDLY} = 255 * PCLK2$</p> <p>注意：</p> <p>在SYNCEN为“0”时设置本寄存器。请不要写入0x00。</p> <p>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：</p> <p>单次延迟触发模式：$t_{SYNCDLY} > t_{SPL}$</p> <p>两ADC循环延迟触发模式：$t_{SYNCDLY} > t_{SPL}$，且$t_{SYNCDLY} > t_{SCAN}/2$</p> <p>三ADC循环延迟触发模式：$t_{SYNCDLY} > t_{SPL}$，且$t_{SYNCDLY} > t_{SCAN}/3$</p> <p>单次并行触发模式：本寄存器设置无效。</p> <p>循环并行触发模式：$t_{SYNCDLY} > t_{SCAN}$</p>	RW
b7	Res	保留位	读出时为0，写入时写0	RW
b6~b4	SYNCMD[2:0]	同步模式选择	<p>SYNCMD[2]</p> <p>0: 单次触发</p> <p>1: 循环触发</p> <p>SYNCMD[1]</p> <p>0: 延迟触发模式</p> <p>1: 并行触发模式</p> <p>SYNCMD[0]</p> <p>0: ADC1和ADC2同步工作，ADC3独立工作</p> <p>1: ADC1，ADC2和ADC3同步工作</p> <p>注意：</p> <p>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。</p>	RW
b3~b1	Res	保留位	读出时为0，写入时写0	RW
b0	SYNCEN	同步模式许可	<p>0: 同步模式无效</p> <p>1: 同步模式有效</p> <p>注意：</p> <p>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。</p> <p>软件对ADC1_STR.STRT写0强制停止转换时，SYNCEN自动清0。</p>	RW

注意：

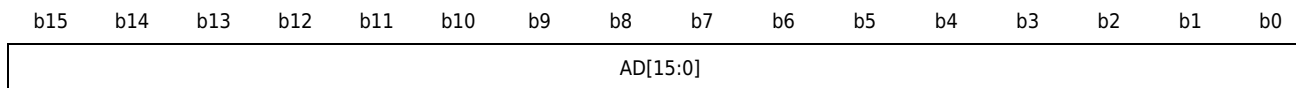
本寄存器只在主控 ADC (即 ADC1) 中搭载，其他 ADC 单元中无此寄存器。

16.5.16 ADC 数据寄存器 (ADC_DRx)

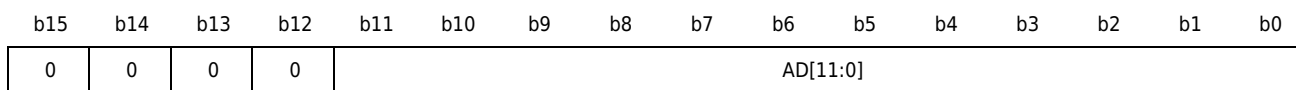
ADC_DR 寄存器是用于存储各通道 ADC 转换数据的只读寄存器。复位值为 0x0000。

根据采用模式，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。

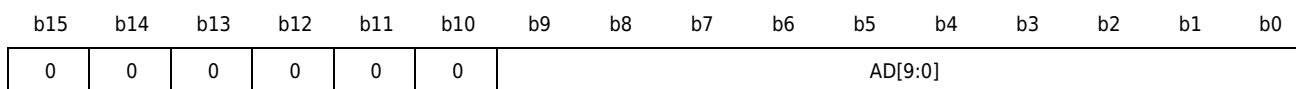
过采样模式



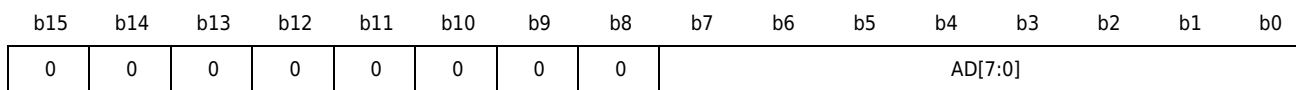
数据右对齐-12 位分辨率



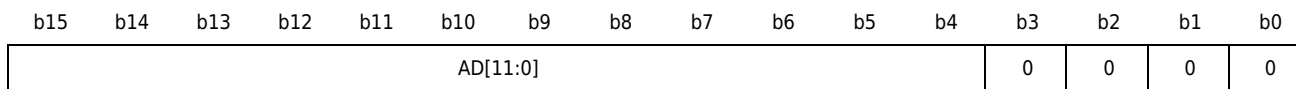
数据右对齐-10 位分辨率



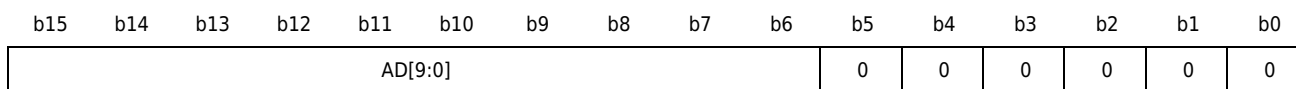
数据右对齐-8 位分辨率



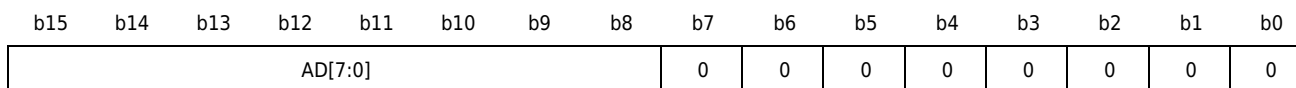
数据左对齐-12 位分辨率



数据左对齐-10 位分辨率



数据左对齐-8 位分辨率



注：

ADC_DRx 中的 x 表示通道，ADC1 中 x=0~15，ADC2 中 x=4~15，ADC3 中 x=0~3, 6~11。

16.5.17 模拟看门狗控制寄存器 (ADC_AWDCR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
Res						AWDCM[1:0]	
b7	b6	b5	b4	b3	b2	b1	b0
Res	AWD1MD	AWD1IEN	AWD1EN	Res	AWD0MD	AWD0IEN	AWD0EN

位/位域	标记	位名	功能	RW
b15~b10	Res	保留位	读出时为0, 写入时写0	RW
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。	RW
b7	Res	保留位	读出时为0, 写入时写0	RW
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。	RW
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	RW
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	RW
b3	Res	保留位	读出时为0, 写入时写0	RW
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。	RW
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	RW
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	RW

16.5.18 模拟看门狗状态寄存器 (ADC_AWDSR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res		AWDCMF		Res		AWD1F	AWD0F

位/位域	标记	位名	功能	RW
b7~b5	Res	保留位	读出时为0, 写入时写0	RW
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]=0b01/0b10/0b11) 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	Res	保留位	读出时为0, 写入时写0	RW
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

16.5.19 模拟看门狗状态复位寄存器 (ADC_AWDSCLRR)

复位值: 0x00

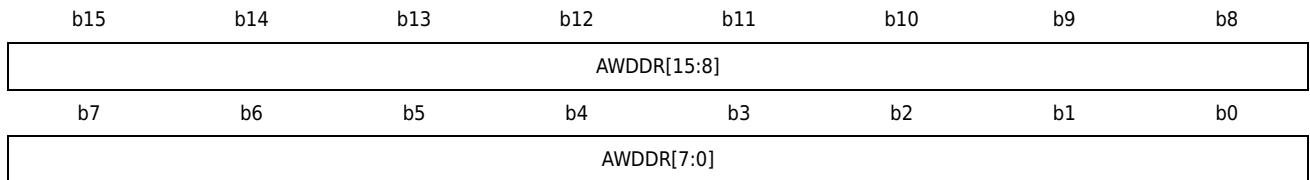
b7	b6	b5	b4	b3	b2	b1	b0
Res		CLRAWDCMF		Res		CLRAWD1F	CLRAWD0F

位/位域	标记	位名	功能	RW
b7~b5	Res	保留位	读出时为0, 写入时写0	RW
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	RW
b3~b2	Res	保留位	读出时为0, 写入时写0	RW
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	RW
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AWD0F状态位, 读出永远为0	RW

16.5.20 模拟看门狗阈值寄存器 (ADC_AWD0DR0、ADC_AWD0DR1、ADC_AWD1DR0、ADC_AWD1DR1)

复位值：ADC_AWD0DR0=0x0000，ADC_AWD0DR1=0xFFFF

ADC_AWD1DR0=0x0000，ADC_AWD1DR1=0xFFFF



位/位域	标记	位名	功能		RW
b15~b0	AWDDR[15:0]	比较数据	比较数据		RW

AWD0DR0 设置窗口 0 低阈值，AWD0DR1 设置窗口 0 高阈值。

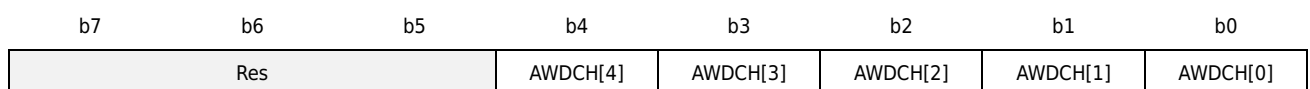
AWD1DR0 设置窗口 1 低阈值，AWD1DR1 设置窗口 1 高阈值。

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

16.5.21 模拟看门狗比较通道选择寄存器 (ADC_AWD0CHSR、ADC_AWD1CHSR)

复位值：0x00



位/位域	标记	位名	功能		RW
b7~b5	Res	保留位	读出时为0，写入时写0		RW
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： 请不要设置成不存在的通道。		RW

16.6 注意事项

16.6.1 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。

16.6.2 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。

16.6.3 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 1 μ s 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【动作模式与低功耗模式】章节。

16.6.4 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。

16.6.5 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。

17 数模转换器 (DAC)

17.1 概述

本 MCU 搭载了 2 个 12 位的数模转换器单元 DAC。DAC1 单元包含两个 DAC 转换通道，两个通道可以独立转换也可以同步转换，DAC2 单元包含 1 个 DAC 转换通道。模拟电压输出范围有两档可设。每个转换通道配有输出放大器，可以在没有外部运放时直接驱动外部负载。

17.2 主要特性

- 三个 DAC 转换通道
- 12 位转换数据可配置成左对齐或者右对齐格式
- 同一个 DAC 的两个转换通道可实现同步转换
- 输出可用于电压比较器 (CMP) 的负端电压
- 输出配有放大功能，可直接驱动外部负载
- ADC 转换优先模式可减少 ADC 转换时的干扰

17.3 功能说明

17.3.1 DAC 功能框图

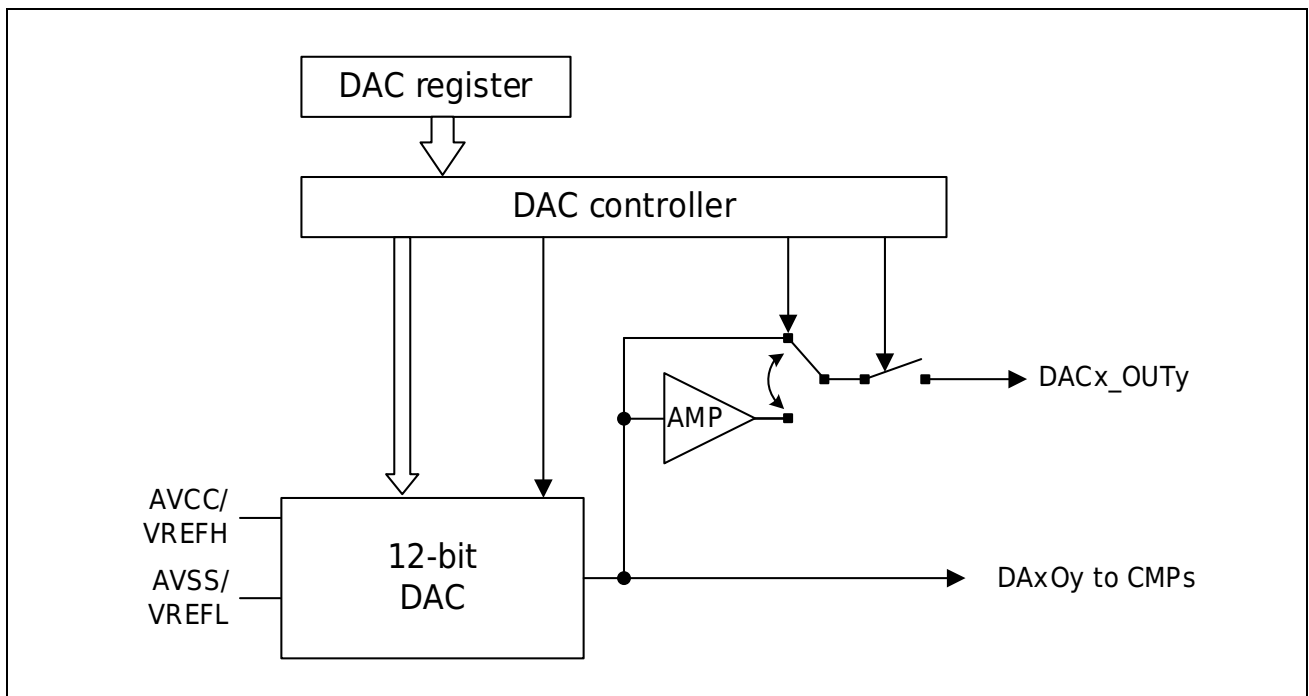


图 17-1 DAC 转换通道框图

表 17-1 DAC 引脚

引脚名称	输入/输出	功能
AVCC/ VREFH	输入	模拟电源/参考电压
AVSS/ VREFL	输入	模拟地/参考地
DACx_OUTy	输出	DAC转换模拟输出 (x=1~2, y=1~2)

17.3.2 DAC 转换

每个转换通道都可以独立进行 DAC 转换。DACx_DACR.DAyE (x=1~2, y=1, 2) 设置成 1 时，对应通道的 DAC 转换开始并从 DACx_OUTy 端口输出转换结果。如仅将转换结果作为电压比较器 CMP 的负端输入电压，可通过 DACx_DAOCR 寄存器将 DACx_OUTy 输出关闭。

下面是一个使用 DAC_ch1 进行单通道转换的例子，动作时序见图 17-2。

- 1) 设置 DAC 转换数据 (DADR1) 和数据格式 (DACR.DPSEL)。
- 2) 设置 DAC1_DACR.DA1E 为 1，DAC 转换开始。经过转换时间 tDCONV 后，转换结果从端口 DAC1_OUT1 输出并一直保持到 DAC1_DADR1 的值被改写或者 DAC1_DACR.DA1E 被设置成 0。输出的模拟电压值可由以下公式计算：

$$DAC_{output} = \frac{\text{Conversion}}{4096} \times VREFH$$

DAC1_DACR.DA1E 为 1 时，改写 DAC1_DADR1 的值将触发一次新的 DAC 转换。同样地，经过转换时间 tDCONV 后，从端口 DAC1_OUT1 输出新的转换结果。

- 3) 设置 DAC1_DACR.DA1E 位为 0，关闭 DAC1_ch1，DAC1_OUT1 输出高阻态。

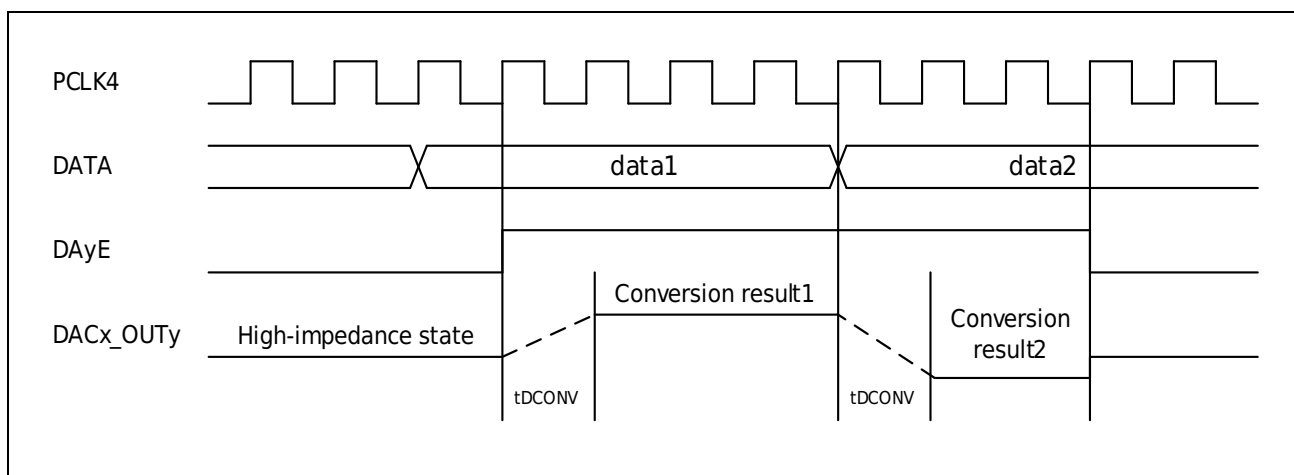


图 17-2 DAC 转换示意图

17.3.3 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, DAC1_DADR2) 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 和 DAC1_OUT2 输出并一直保持到 DADR 的值被改写, 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 和 DAC1_DADR2 的值将同步触发两个通道开始一次新的 DAC 转换。同样地, 经过转换时间 t_{DCONV} 后, 从 DAC1_OUT1 和 DAC1_OUT2 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 和 DAC1_OUT 输出高阻态。

17.3.4 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可以有效地避免这种干扰的发生。

设置 DACx_DAADPCR.ADPEN 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_DADR 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_DADR 时 ADC 处于停止状态 (ADC_STR.STRT 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 17-3。

1. 确认 ADC 处于停止状态, 设置 DAC1_DACR.ADPSL;
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;
4. 转换数据 A 被写入 DAC1_DADR1 时, ADC_ADCSR.ADST 为 0, ADC 处于停止状态, DAC 转换在 2 个 ADCLK 周期后开始;
5. 转换数据 B 被写入 DAC1_DADR1 时, ADC_ADCSR.ADST 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再改写 DADR 的值。

注:

$x=1\sim 2, y=1\sim 2$

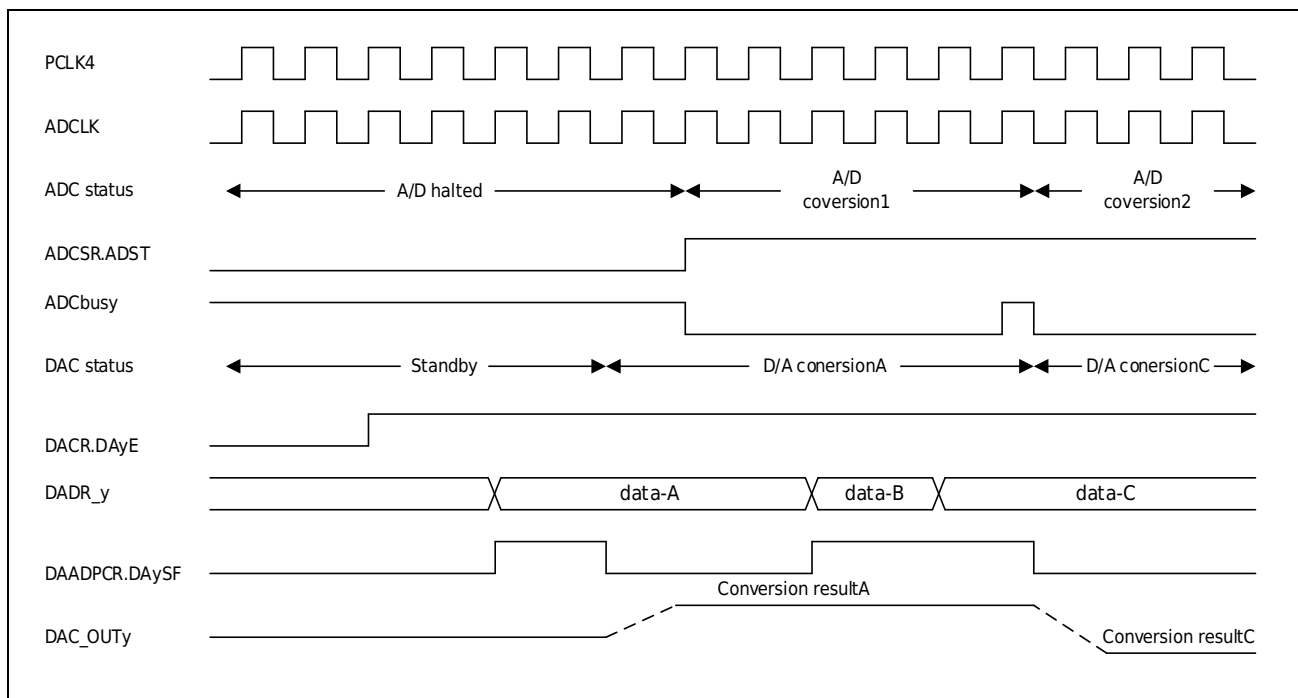


图 17-3 ADC 转换优先模式动作示意图

17.3.5 数据传送模式

通过设定 DACR2.LDMDy=1 可设定由 HRPWM 输出的触发事件来触发数据从 DADR 向 ACTIVE 寄存器 (DAC 转换的实时数据) 传送。

在通道使能的条件下:

- 如果没有 HRPWM 输出的触发事件, 即使改写 DADR 的数据也不会触发数据传输, 从而不发生 D/A 转换;
- 当 HRPWM 输出的触发事件发生时, DADR 的数据被传送至 ACTIVE 寄存器并实施 D/A 转换。DAC 数据转换的 ACTIVE 值可以通过寄存器 DADACTRy 读出。

17.4 寄存器描述

表 17-2 DAC 基地址

名称	基地址	描述
DAC1	0x4004 1000	DAC1基地址
DAC2	0x4004 1400	DAC2基地址

表 17-3 DAC 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
DADR1	DAC数据寄存器1	0x00	16	0x0000
DADR2	DAC数据寄存器2	0x02	16	0x0000
DACR	DAC控制寄存器	0x04	16	0x0000
DAADPCR	DAC ADC转换优先控制寄存器	0x06	16	0x0000
DAOCR	DAC模拟输出控制寄存器	0x1C	16	0x0000
DACR2	DAC控制寄存器2	0x20	16	0x0000
DDACTR1	DAC数据ACTIVE值寄存器1	0x2c	16	0x0000
DDACTR2	DAC数据ACTIVE值寄存器2	0x2e	16	0x0000

17.4.1 DAC 数据寄存器 (DADR_y, 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

Res	DADR[11:0]
-----	------------

位/位域	标记	位名	功能	读写
b15~b12	Res	保留位	读时读出0, 写时请写0	RW
b11~b0	DADR[11:0]	变换数据	变换数据设定值	RW

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]	Res
------------	-----

位/位域	标记	位名	功能	读写
b15~b4	DADR[11:0]	变换数据	变换数据设定值	RW
b3~b0	Res	保留位	读时读出0, 写时请写0	RW

DAC_x_DADR 用于存放 DAC 变换的数据, 每个通道对应有一个。只要允许 DAC 转换, DAC_x_DADR 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 DAC_x_DADR 进行 32 位操作可实现双通道同步转换。

17.4.2 DAC 控制寄存器 (DACR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					DAAMP2	DAAMP1	DPSEL	Res					DA2E	DA1E	DAE
位/位域	标记	位名	功能		读写										
b15~b11	Res	保留位	读出时为“0”，写入时写“0”		RW										
b10	DAAMP2	通道2输出放大器使能	0: 禁止通道2输出放大器 1: 启动通道2输出放大器		RW										
b9	DAAMP1	通道1输出放大器使能	0: 禁止通道1输出放大器 1: 启动通道1输出放大器		RW										
b8	DPSEL	数据寄存器格式选择	0: 右对齐格式 1: 左对齐格式		RW										
b7~b3	Res	保留位	读出时为“0”，写入时写“0”		RW										
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		RW										
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		RW										
b0	DAE	通道总使能	0: 通道1, 2禁止 1: 通道1, 2使能		RW										

注:

1. DACR 用于控制 DAC 转换和 DAC 输出。详见表 17-4。
2. DA1E 位为 0 并且 DAE 位也为 0 时，通道 1 停止 DAC 转换，这时不管 DAAMP1 是什么值，通道 1 的模拟输出被禁止，端口呈现高阻态。DA2E 位为 0 并且 DAE 位也为 0 时，通道 2 停止 DAC 转换，这时不管 DAAMP2 是什么值，通道 2 的模拟输出被禁止，端口呈现高阻态。
3. 如果选择 ADC 转换优先模式，请在 ADC 停止 (ADC_STR.STRT=0) 的状态下设定 DA2E, DA1E 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。

17.4.3 DAC 模拟输出控制寄存器 (DAO CR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DAO DIS2	DAO DIS1	Res													

位/位域	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	RW
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	RW
b13~b0	Res	保留位	读出时为“0”，写入时写“0”	RW

表 17-4 DAC 转换和模拟输出控制

DAE	DAyE	DAAMP	DAODIS	DAC转换	输出放大器	DAC_OUT 输出	DAO 输出	
0	0	0/1	0/1	停止	停止	高阻态	高阻态	
		0	0	启动	停止	普通输出	普通输出	
	1	0	1	启动	停止	高阻态	普通输出	
		1	0	启动	启动	放大输出	普通输出	
1	0	0	0	启动	停止	普通输出	普通输出	
		1	0	启动	停止	高阻态	普通输出	
		1	0	启动	启动	放大输出	普通输出	
		1	1	启动	启动	高阻态	普通输出	
	1	0	0	0	启动	停止	普通输出	普通输出
			1	0	启动	停止	高阻态	普通输出
		1	0	0	启动	启动	放大输出	普通输出
			1	0	启动	启动	高阻态	普通输出

17.4.4 DAC ADC 转换优先控制寄存器 (DAADPCR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADPEN	Res					DA2SF	DA1SF	Res					ADC SL3	ADC SL2	ADC SL1

位/位域	标记	位名	功能	读写
b15	ADPEN	ADC转换优先模式选择	0: ADC转换优先模式无效 1: ADC转换优先模式有效	RW
b14~b10	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
b7~b3	Res	保留位	读出时为“0”，写入时写“0”	RW
b2	ADC SL3	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	RW
b1	ADC SL2	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	RW
b0	ADC SL1	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	RW

注:

1. DAADPCR 用于 ADC 转换优先模式的控制。
2. 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。
3. DA1SF 为 DA 转换状态标志位,反映的是当前该通道数据寄存器中的数据是否已被转换。

17.4.5 DAC 控制寄存器 2 (DACR2)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LDMD2	Res							LDMD1	Res						

位	标记	位名	功能	读写
b15	LDMD2	通道2数据传送模式选择	仅转换数据选择DADR时有效(DACx_DACR.EXTDSL2=0) 0: 数据即时传送 1: HRPWM输出的触发事件触发数据传送	RW
b14~b8	Res	保留位	读出时为“0”，写入时写“0”	RW
b7	LDMD1	通道1数据传送模式选择	仅转换数据选择DADR时有效(DACx_DACR.EXTDSL1=0) 0: 数据即时传送 1: HRPWM输出的触发事件触发数据传送	RW
b6~b0	Res	保留位	读出时为“0”，写入时写“0”	RW

17.4.6 DAC 数据 ACTIVE 值寄存器 (DADACTy, y=1~2)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				DATAACT[11:0]											

位	标记	位名	功能	读写
b15~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
b11~b0	DATAACT	通道y变换数据ACTIVE值	变换数据的ACTIVE值	R

17.5 注意事项

17.5.1 模块停止功能的设置

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 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【动作模式与低功耗模式】。

17.5.2 模块停止状态下的 DAC 动作

如果 DAC 是在 DAC 转换时进入模块停止状态的，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模块停止状态下的功耗，请将 DACx_DACR 的 DAyE 和 DAE 位全部设为 0。

17.5.3 停止模式下的 DAC 动作

如果系统是在 DAC 转换时进入停止模式（Stop Mode）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。

17.5.4 掉电模式下的 DAC 动作

如果系统是在 DAC 转换时进入掉电模式（Power Down Mode）的，模拟输出将被置为高阻态。

17.5.5 用输出放大器的注意事项

使用输出放大器时请使用以下的初始化程序。

1. 给 DACx_DADRY 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 3 μ s 后将变换数据写入 DACx_DADRY。

关闭 DAC 可以使放大器停止工作。若要再次使用放大器，请重复步骤 1~4。

18 高级控制定时器 (Timer6)

18.1 概述

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 相正交编码计数和 3 相正交编码计数；支持 EMB 控制。本系列产品中搭载 4 个单元的 Timer6。

18.2 主要特性

Timer6 基本的功能及特性如表 18-1 所示。

表 18-1 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
中断输出	EMB控制
	计数比较匹配中断
	计数周期匹配中断
事件输出	死区时间错误中断
	计数比较匹配事件
	计数周期匹配事件

18.3 功能说明

18.3.1 Timer6 框图

Timer6 的基本框图如图 18-1 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。

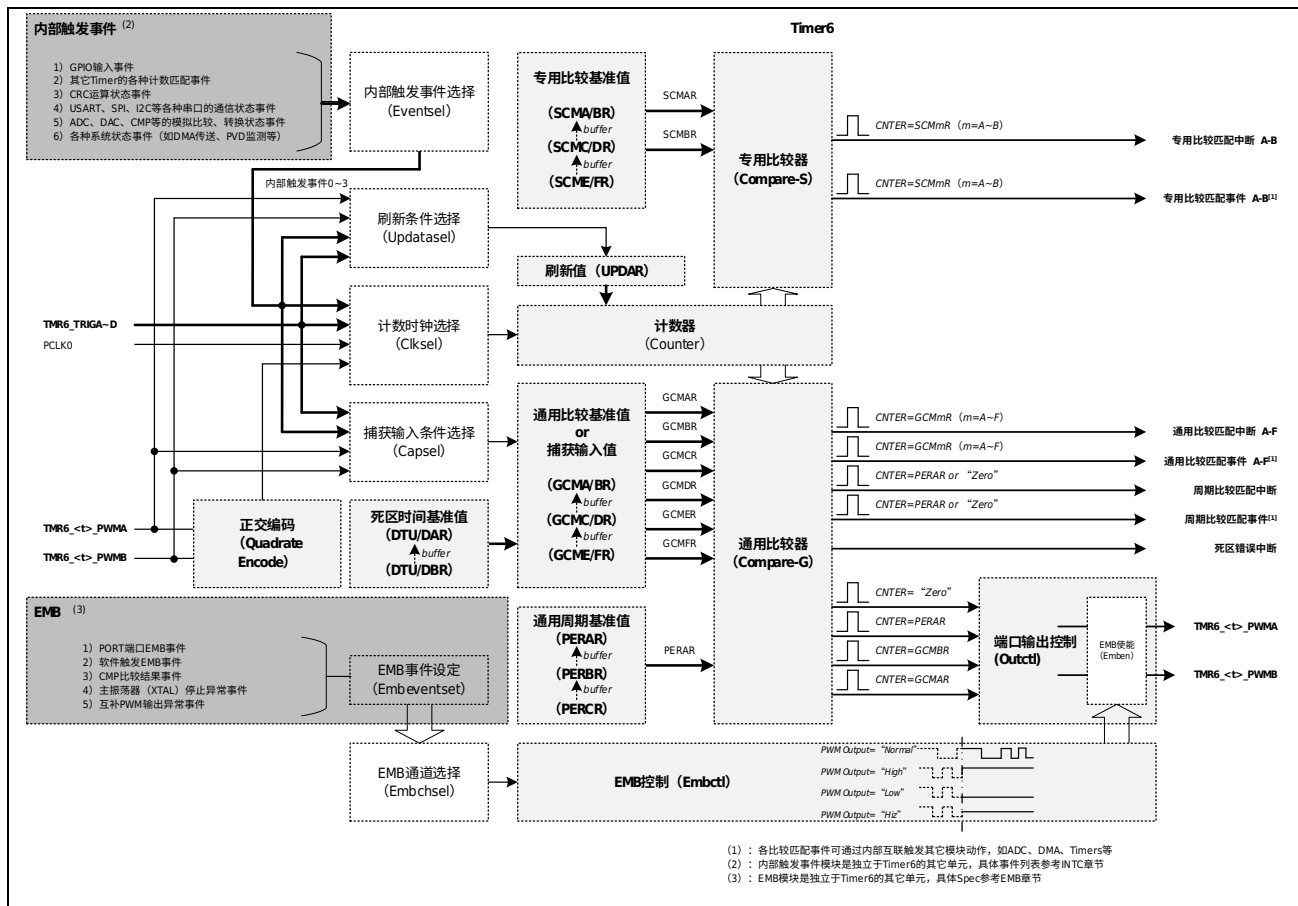


图 18-1 Timer6 基本框图

表 18-2 所示为 Timer6 的输入输出端口列表。

表 18-2 Timer6 端口列表

端口名	方向	功能
TMR6_<t>_PWMA	输入输出	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
TMR6_<t>_PWMB		
TMR6_TRIGA	输入	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
TMR6_TRIGB		
TMR6_TRIGC		
TMR6_TRIGD		

18.3.2 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 18-2、图 18-3 所示。

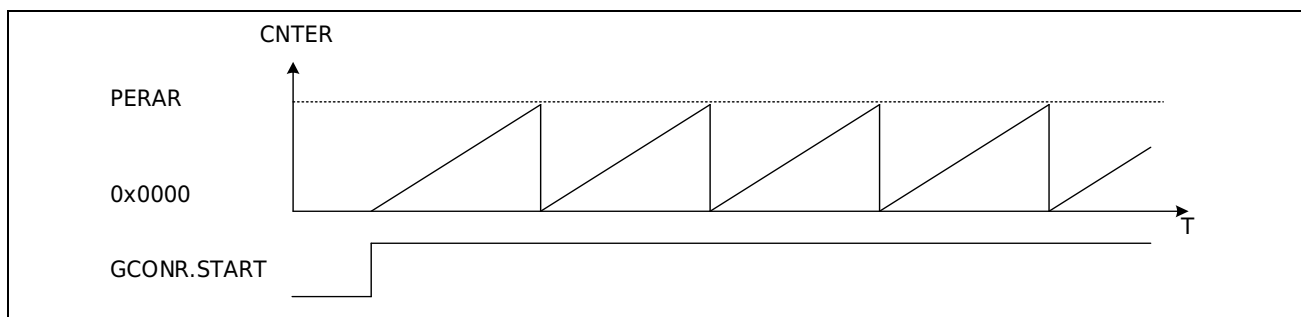


图 18-2 锯齿波波形 (递加计数)

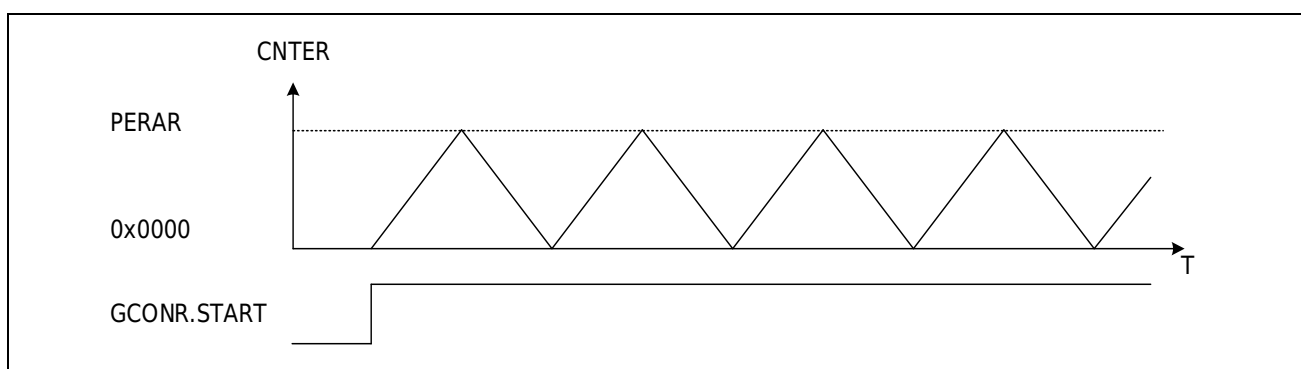


图 18-3 三角波波形

18.3.3 时钟源选择

Timer6 的计数时钟可以有以下几种选择：

- PCLK0 的 1、2、4、8、16、32、64、128、256、512、1024 分频 (GCONR.CKDIV[3:0]设定)
- 内部触发事件输入 0~3 (HCUPR.HCUP[11:8]或 HCDOR.HCDO[11:8]设定)
- TMR6_TRIGA~D 的端口输入 (HCUPR.HCUP[23:16]或 HCDOR.HCDO[23:16]设定)
- TMR6_<t>_PWMA 和 TMR6_<t>_PWMB 的正交编码关系输入 (HCUPR.HCUP[7:0] 或 HCDOR.HCDO[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。

18.3.4 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。

18.3.4.1 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。

18.3.4.2 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。

18.3.5 比较输出

每个单元的 Timer6 有 2 个比较输出端口（ $TMR6_{<t>_PWMA}$ 、 $TMR6_{<t>_PWMB}$ ），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 $TMR6_{<t>_PWMA}$ 、 $TMR6_{<t>_PWMB}$ 的计数比较基准值。当定时器的计数值和 $GCMAR$ 相等时， $TMR6_{<t>_PWMA}$ 或 $TMR6_{<t>_PWMB}$ 端口输出指定电平。

$TMR6_{<t>_PWMA}$ 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 $PCNAR.STACA$ 、 $PCNAR.STPCA$ 、 $PCNAR.OVFCA$ 、 $PCNAR.UDFCA$ 、 $PCNAR.CMAU<D>A$ 、 $PCNAR.FORCA$ 位设定。

$TMR6_{<t>_PWMB}$ 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 $PCNBR.STACB$ 、 $PCNBR.STPCB$ 、 $PCNBR.OVF CB$ 、 $PCNBR.UDFCB$ 、 $PCNBR.CMAU<D>B$ 、 $PCNBR.FORCB$ 位设定。图 18-4 为比较输出的动作例。

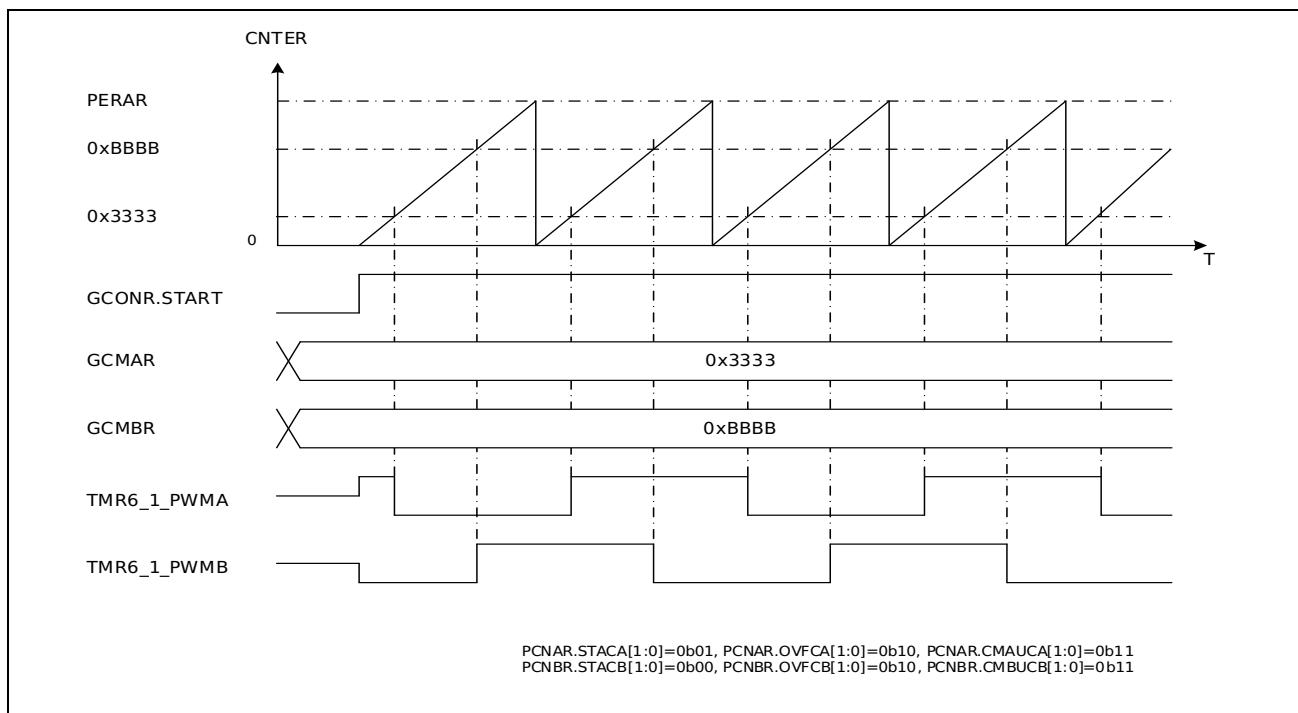


图 18-4 比较输出动作

18.3.6 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（PCNAR）的 CAPMDA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。

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、TMR6_TRIGA~D 的端口输入、TMR6_<t>_PWMA 或 TMR6_<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 18-5 为捕获输入的动作例。

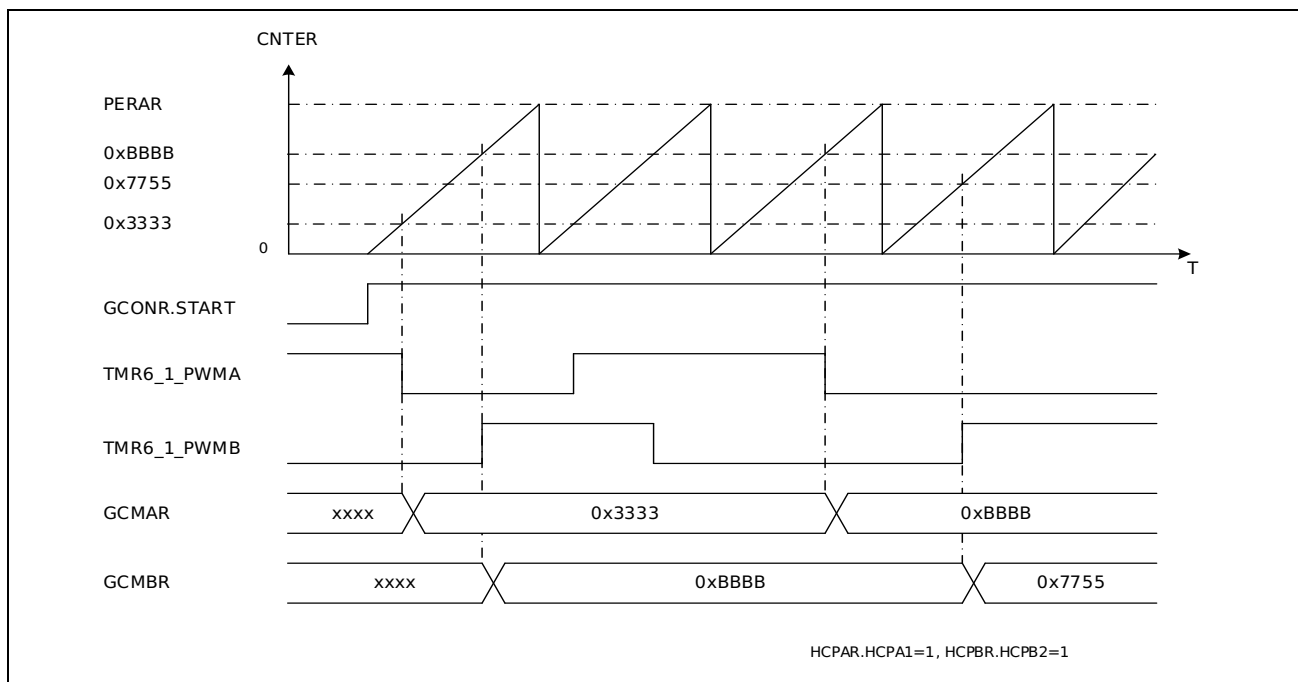


图 18-5 捕获输入动作

18.3.7 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 (CNTER) 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 (CNTER) 的值更新为通用刷新值寄存器 (UPDAR) 中指定的值。图 18-6 是锯齿波模式在计数中的硬件刷新动作例。

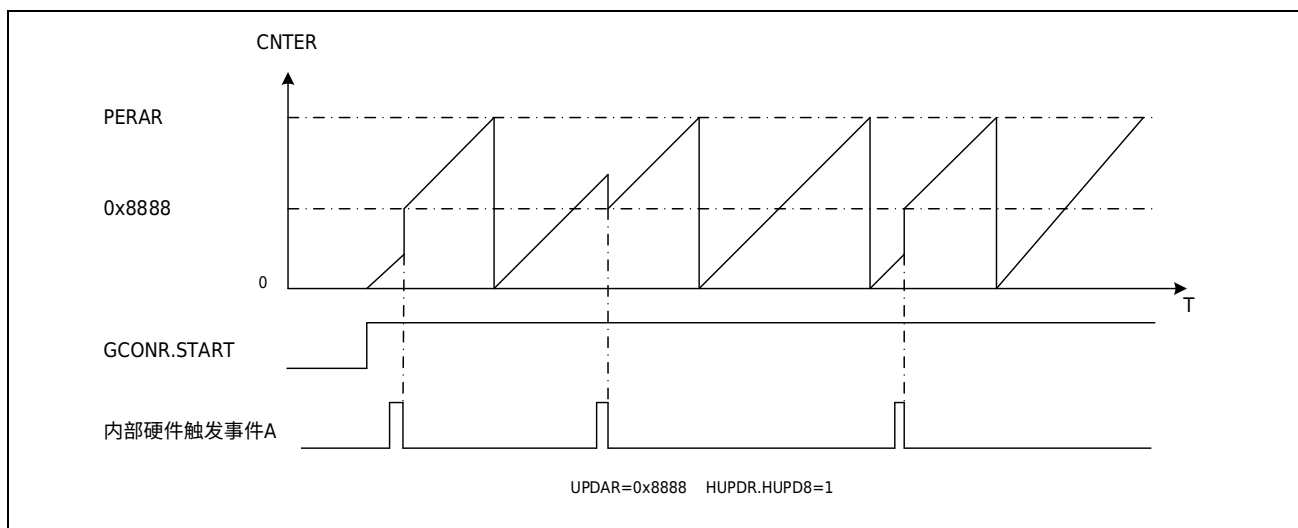


图 18-6 硬件刷新动作

18.3.8 软件同步

18.3.8.1 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 (CNTERR) 的同步启动。

另外, 产品还配置了一个计数同步使能寄存器 (TMR_SYNENR), 用于同步启动产品所搭载的所有 Timer (包括 Timer0、Timer4、Timer6、TimerA)。使用方法和寄存器说明参考【通用定时器 (Timer0)】章节。Timer6 设置通用控制寄存器里的计数时钟分频 (TMR6_GCONR.CKDIV[2:0]) 时, 不支持与其他 Timer 同步。

18.3.8.2 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 (CNTERR) 的同步停止。

18.3.8.3 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 (CNTERR) 的同步清零。

18.3.8.4 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 (CNTERR) 的同步刷新。

如图 18-7 所示、若设定 SSTAR.SSTA1=SSTAR.SSTA2=1, 即可实现单元 1~2 的软件同步启动、设定 SSTPR.SSTP1=SSTPR.SSTP2=1, 即可实现单元 1~2 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。

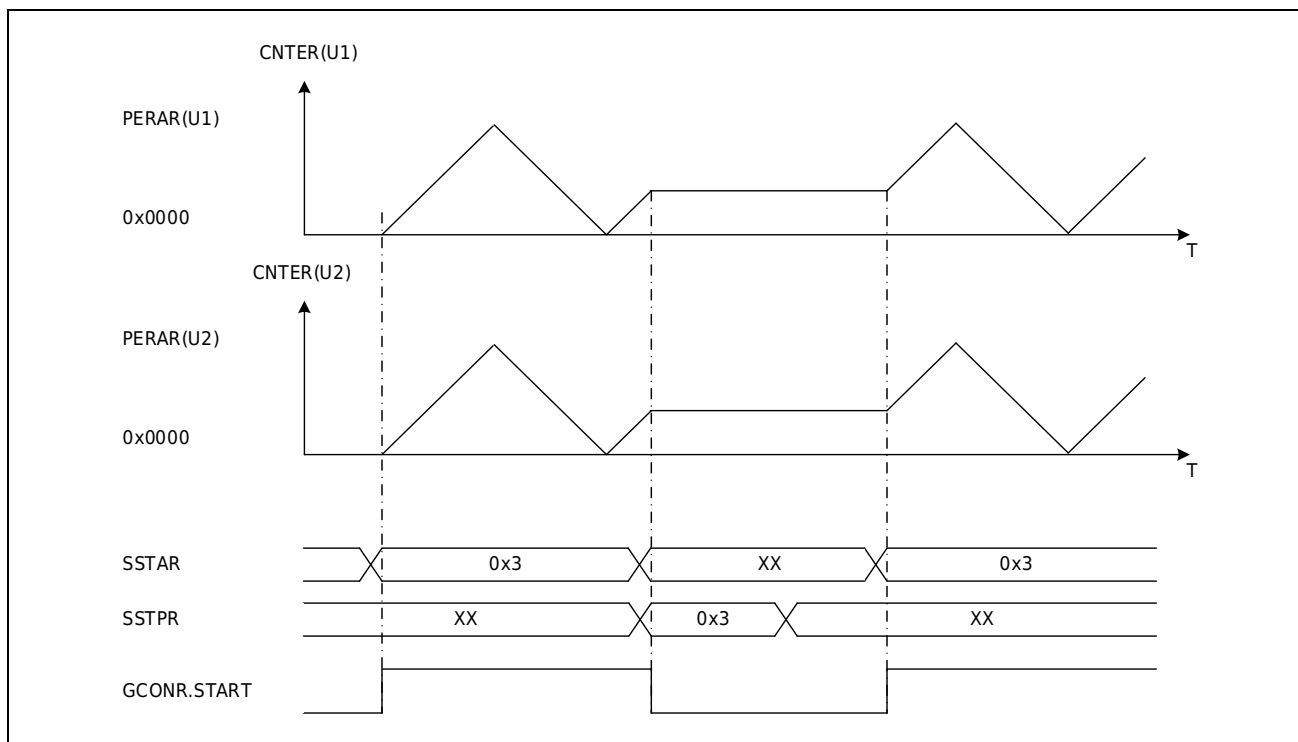


图 18-7 软件同步动作

18.3.9 硬件同步

每个单元除独立拥有 2 个通用输入端口 (TMR6_<t>_PWMA、TMR6_<t>_PWMB) 外, 还共同拥有 4 个通用外部触发输入端口 (TMR6_TRIGA、TMR6_TRIGB、TMR6_TRIGC、TMR6_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 (TMR6_TRGSELx, x=0~3) 中对应的编号设定来选择, 具体的寄存器内容详见【Timer6 硬件触发事件选择寄存器 (TMR6_TRGSELx, x=0~3)】章节, 具体的事件对应关系请参考【中断控制器 (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 1。

18.3.9.1 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。

18.3.9.2 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。

18.3.9.3 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。

18.3.9.4 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。

18.3.9.5 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。

18.3.9.6 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 18-8 所示是单元 1~2 的硬件同步动作例。

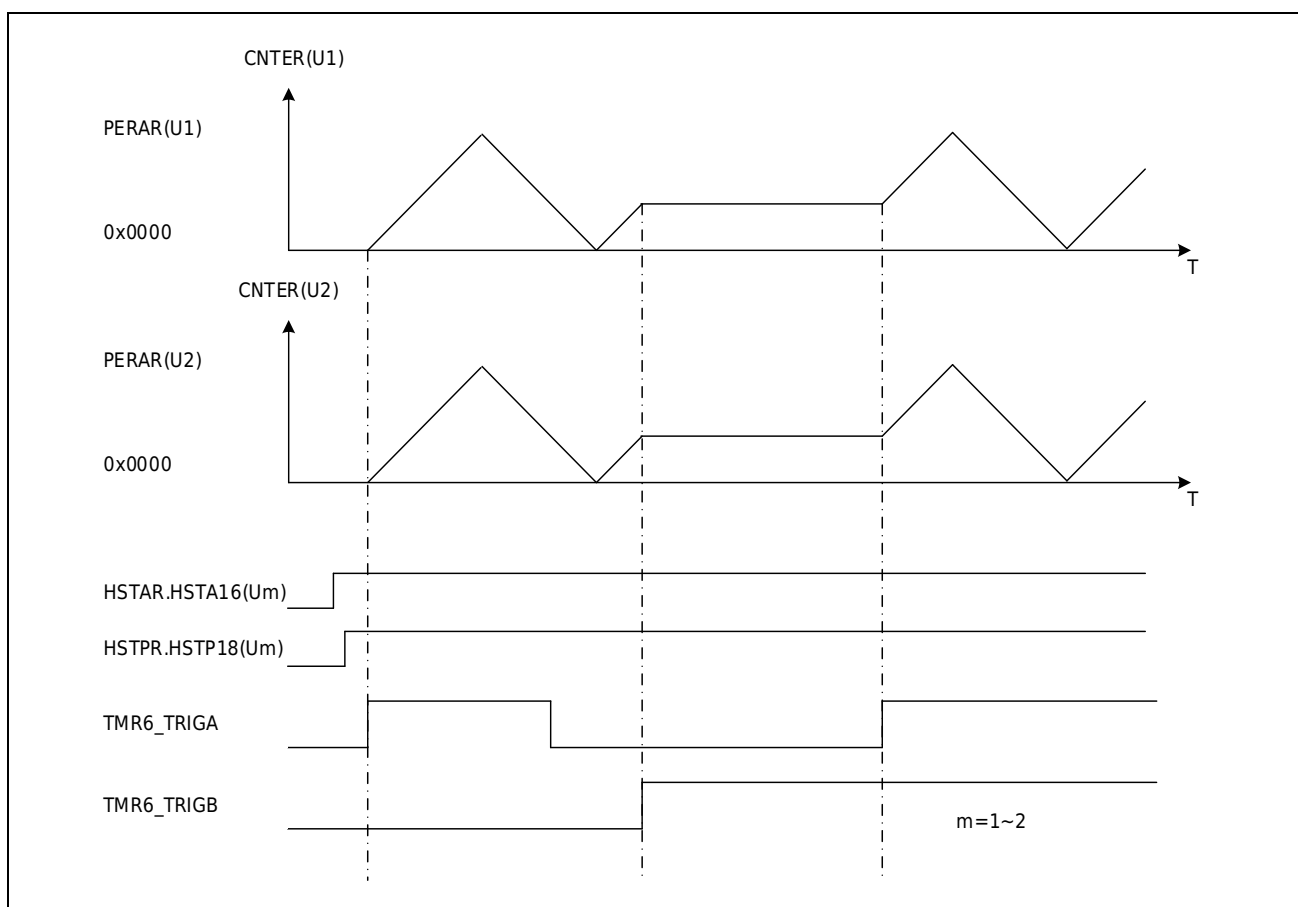


图 18-8 硬件同步动作

18.3.10 脉宽测量

在使用 TMR6_TRIGA~D 端口、TMR6_<t>_PWMA~B 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 TMR6_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 TMR6_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 18-9 所示。

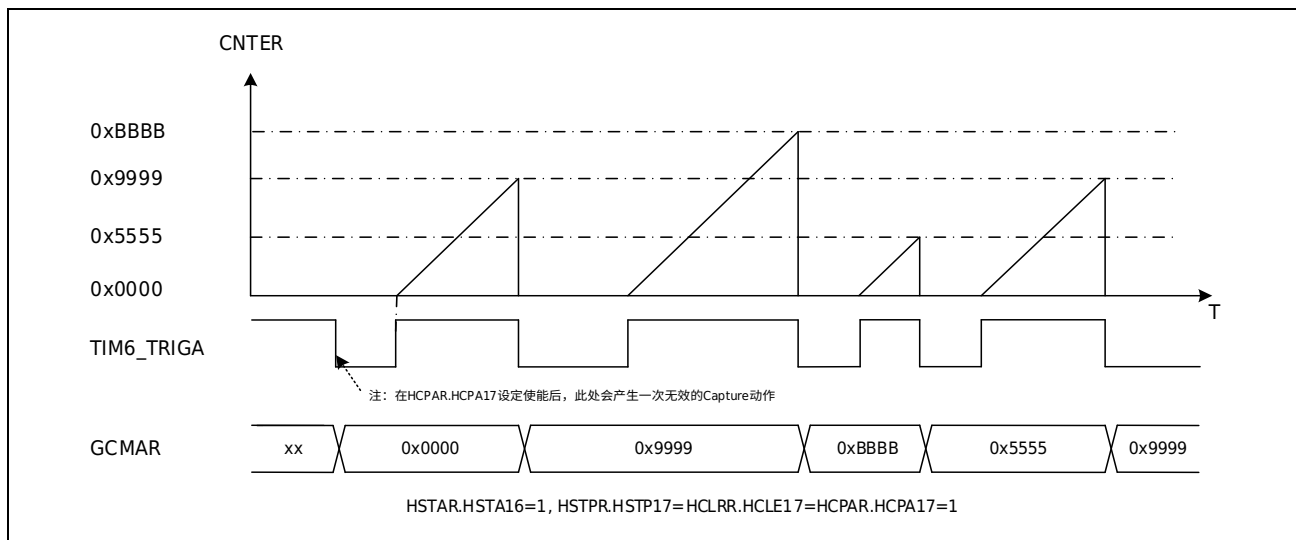


图 18-9 脉宽测量

18.3.11 周期测量

在使用 TMR6_TRIGA~D 端口、TMR6_<t>_PWMA~B 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、GCMAR 寄存器的捕获输入条件均设为 TMR6_TRIGB 的上升沿，就可以实现连续的周期测量。对应动作如图 18-10 所示。

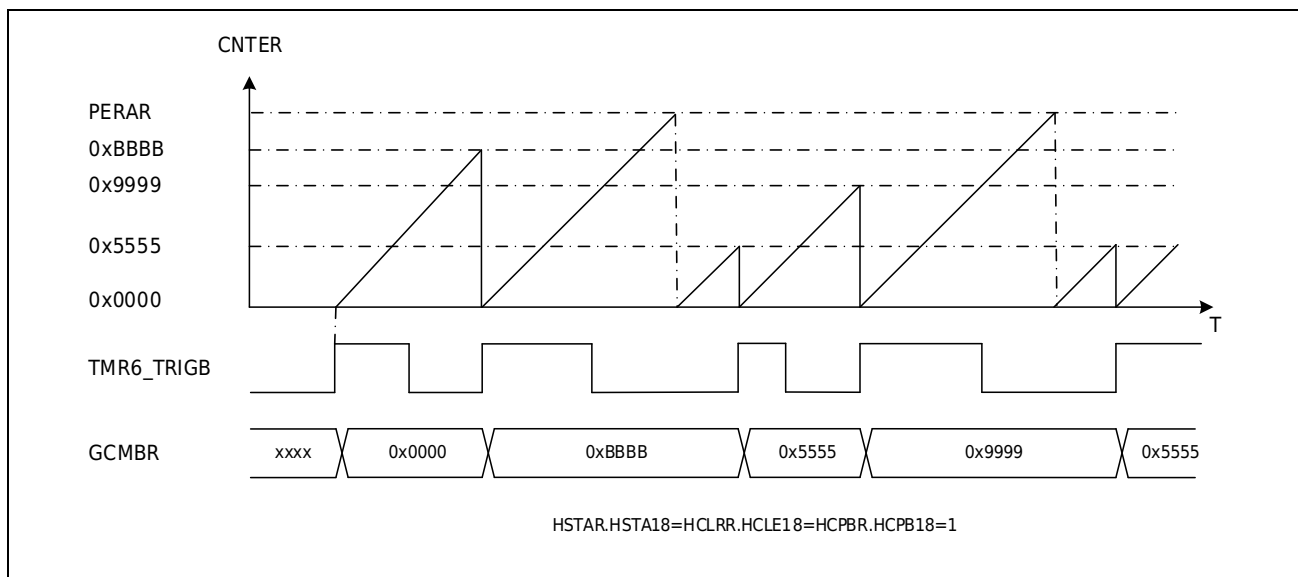


图 18-10 周期测量

18.3.12 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。

18.3.12.1 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<P><SPA><SPB>=1、BCONR.BSEA<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、GCMDBR）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、GCMDBR）中（捕获输入时）
- d) 专用比较基准值缓存寄存器（SCMCR、SCMDBR）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 18-11 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。

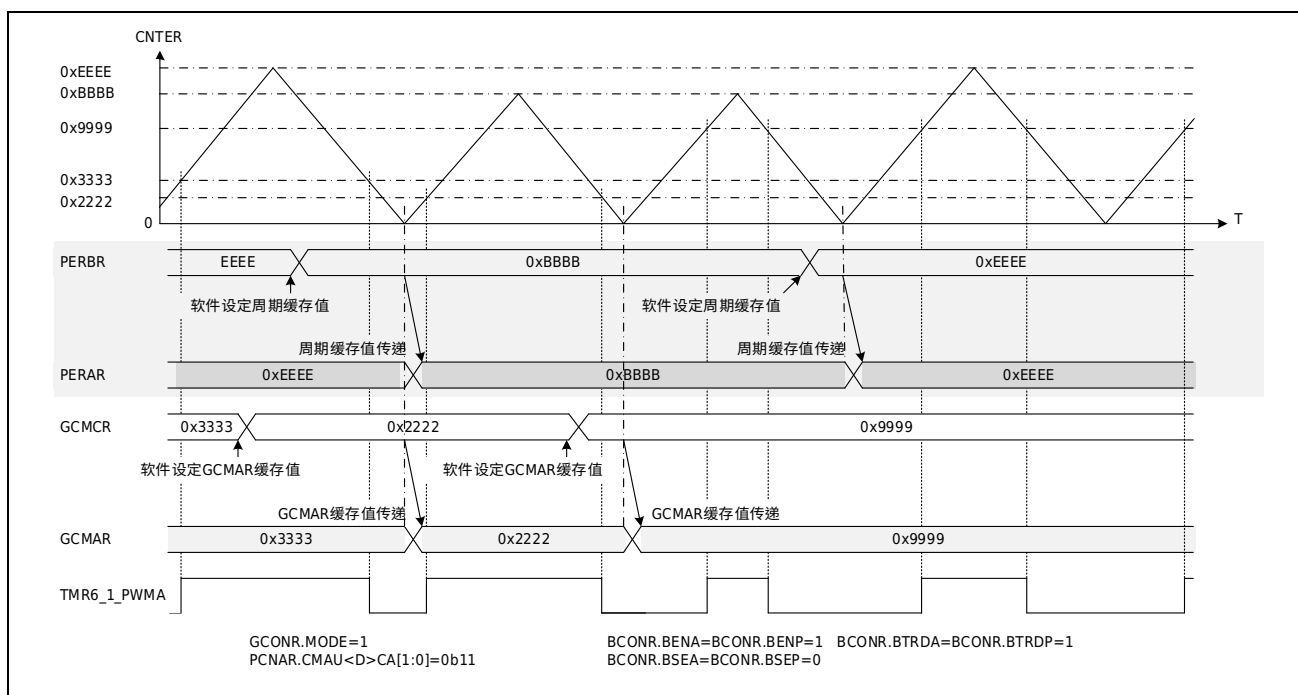


图 18-11 单缓存方式比较输出时序

18.3.12.2 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<P><SPA><SPB>=1、BCONR.BSEA<P><SPA><SPB>=1），在缓存传送时间点，选择发生以下事件：

- 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
 - 通用比较基准值缓存寄存器（GCMCR、GCMCR）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、GCMCR）中（比较输出时）
 - 通用比较基准值缓存寄存器（GCMCR、GCMCR）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、GCMCR）中（捕获输入时）
 - 专用比较基准值缓存寄存器（SCMCR、SCMCR）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、SCMCR）中
- 图 18-12 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。

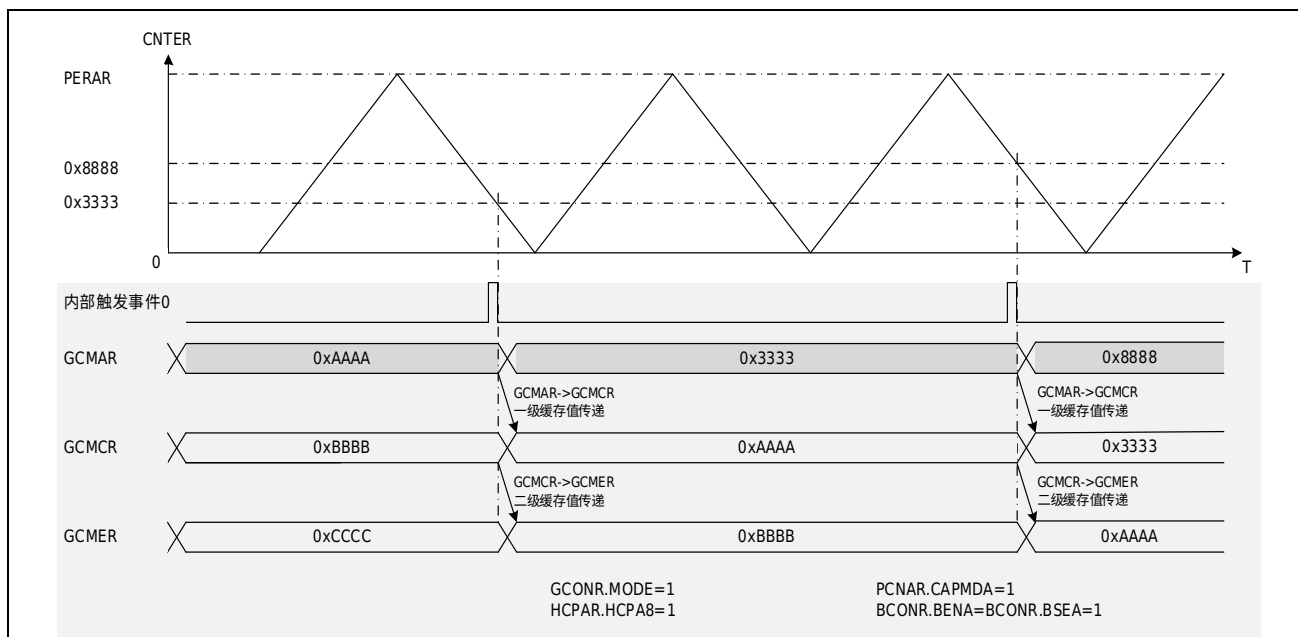


图 18-12 双缓存方式捕获输入时序

18.3.12.3 缓存传送节点

比较输出（锯齿波）缓存传送

在缓存功能有效（ $BCONR.BENA < B > < P > < SPA > < SPB > = DCONR.DTBENU < D > = 1$ ）&&比较输出计数（ $PCNA < B > .R.CAPMDA < B > = 0$ ）&&锯齿波计数模式（ $GCONR.MODE = 0$ ）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 18-13 所示，是锯齿波递加计数时的缓存动作。

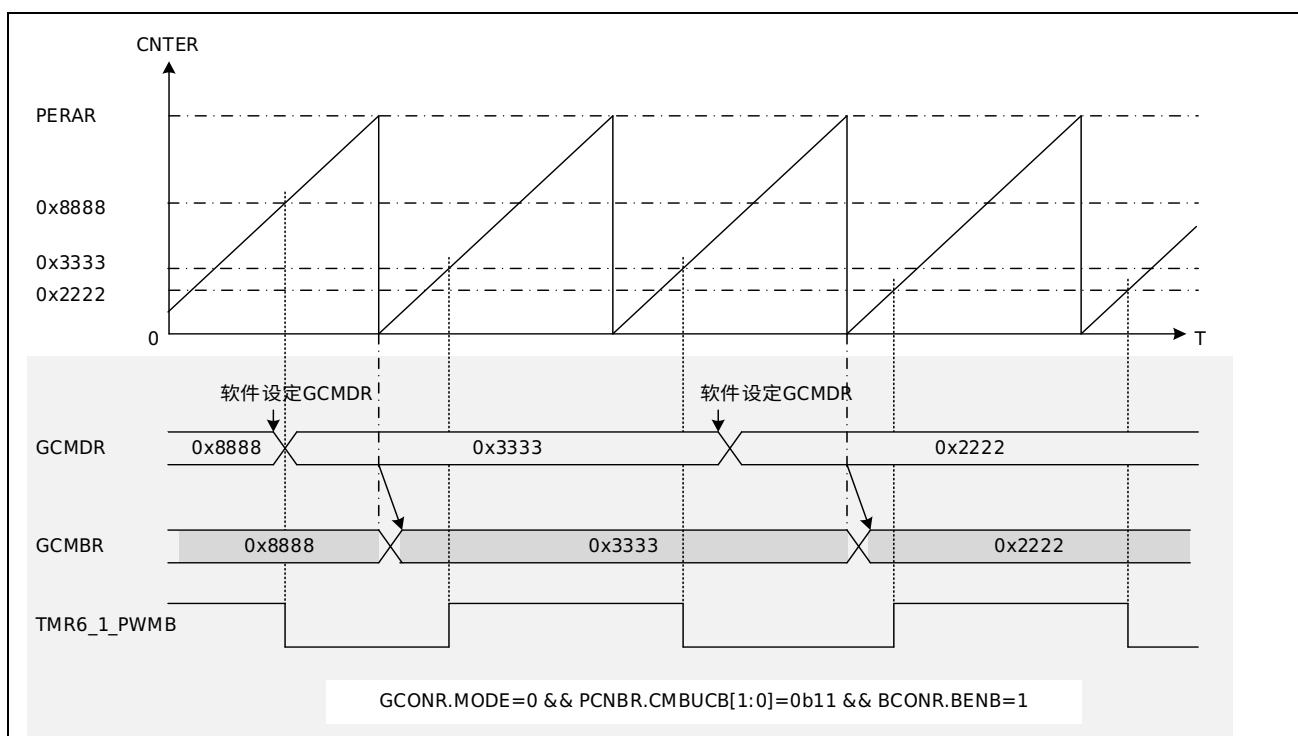


图 18-13 锯齿波模式时计数缓存动作

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（ $BCONR.BENA<P><SPA><SPB>=DCONR.DTBENU<D>=1$ ）&&比较输出计数（ $PCNAR.CAPMDA=0$ ）&&三角波计数模式（ $GCONR.MODE=1$ ）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 $BCONR.BTRUA<P><SPA><SPB>=1$ 或 $DCONR.DTBTRU=1$ 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 $BCONR.BTRDA<P><SPA><SPB>=1$ 或 $DCONR.DTBTRD=1$ 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 18-14 所示，是三角波计数到谷点时的缓存动作，图 18-15 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。

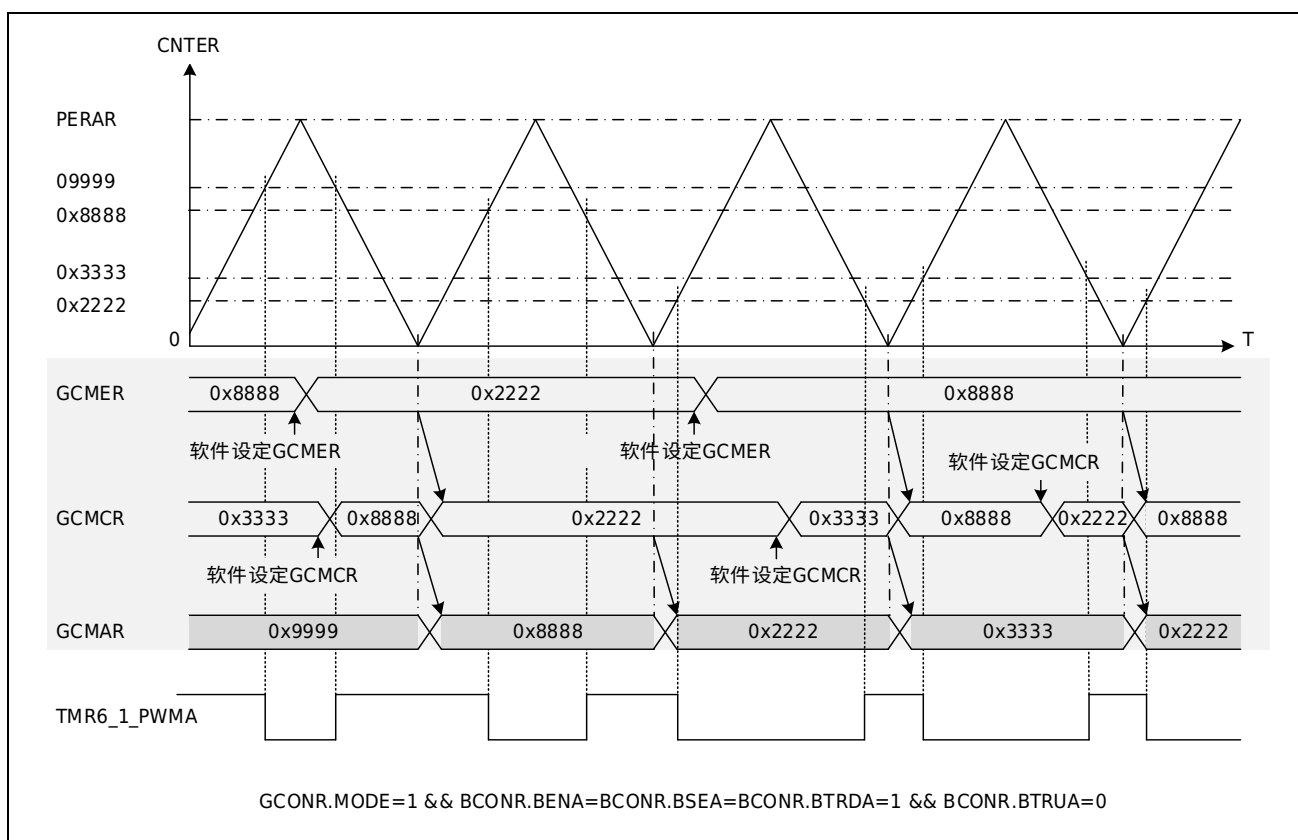


图 18-14 三角波模式时计数缓存动作 1

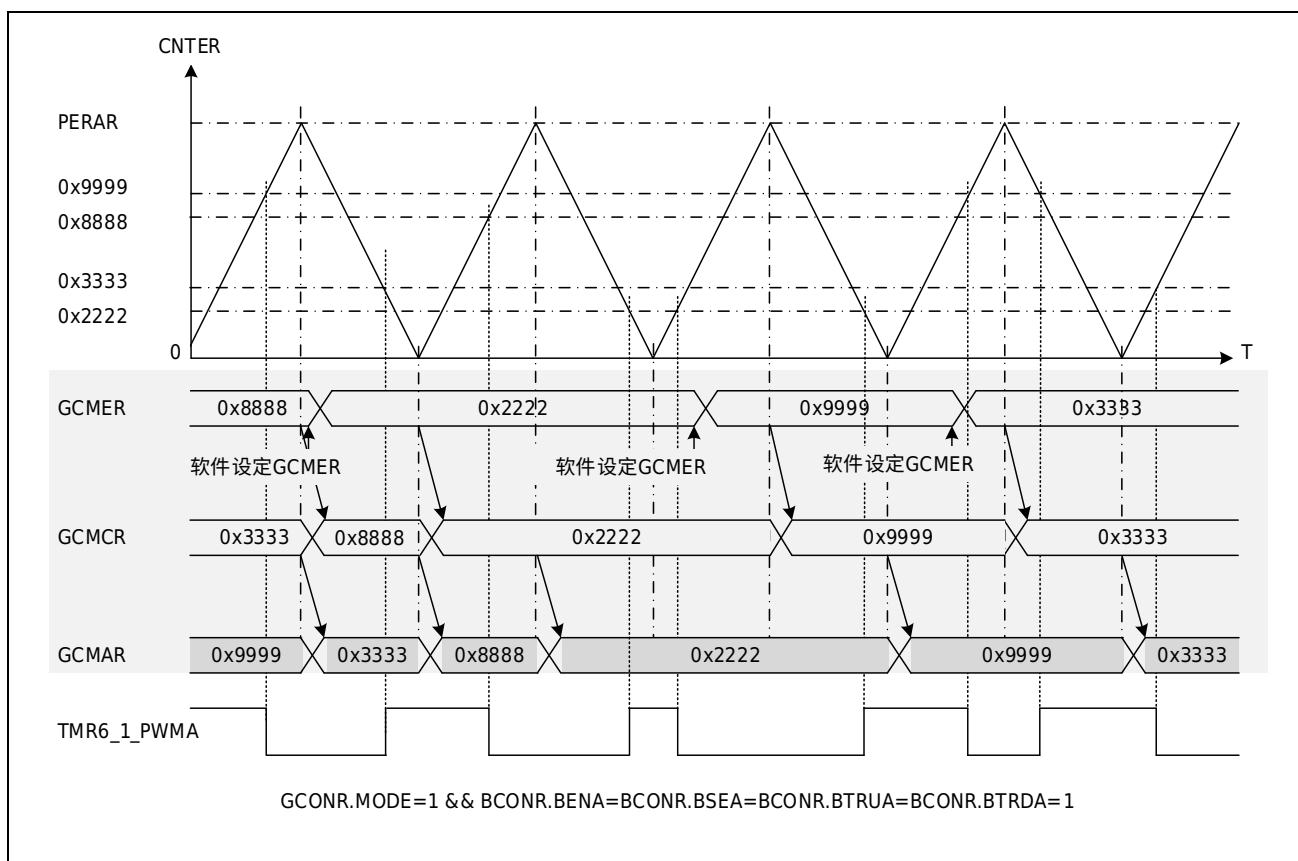


图 18-15 三角波模式时计数缓存动作 2

捕获输入缓存传送

在捕获输入动作 (PCNAR.CAPMDA=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=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设定)。

18.3.13 数字滤波

TMR6_<t>_PWMA、TMR6_<t>_PWMB、TMR6_TRIGA~D 端口输入都具有数字滤波功能。TMR6_<t>_PWMA、TMR6_<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; TMR6_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 18-16 所示。

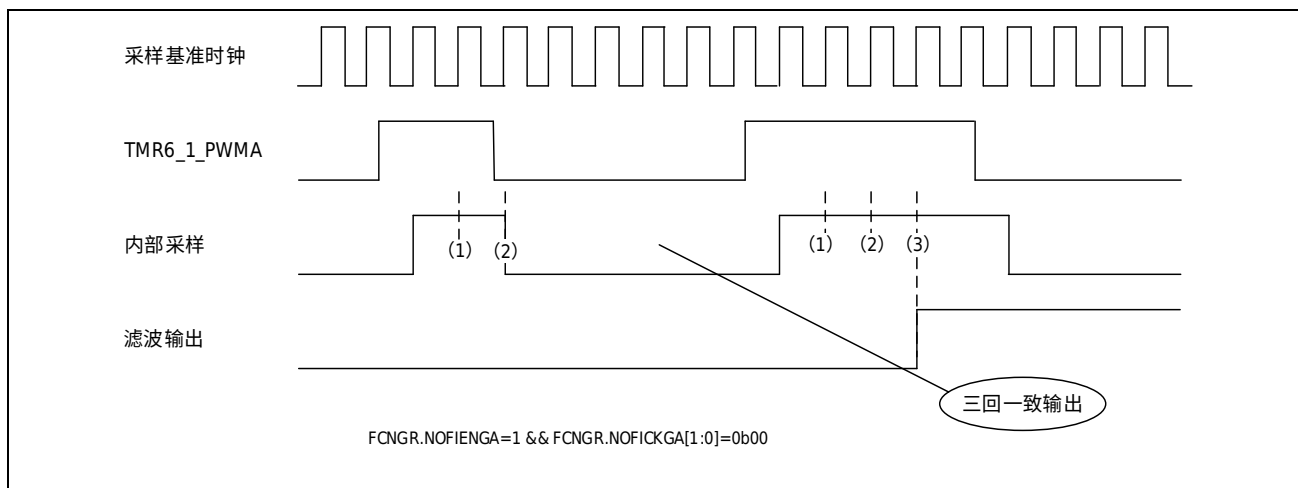


图 18-16 捕获输入端口的滤波功能

18.3.14 通用 PWM 输出

18.3.14.1 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 TMR6_<t>_PWMA、TMR6_<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 18-17 所示。

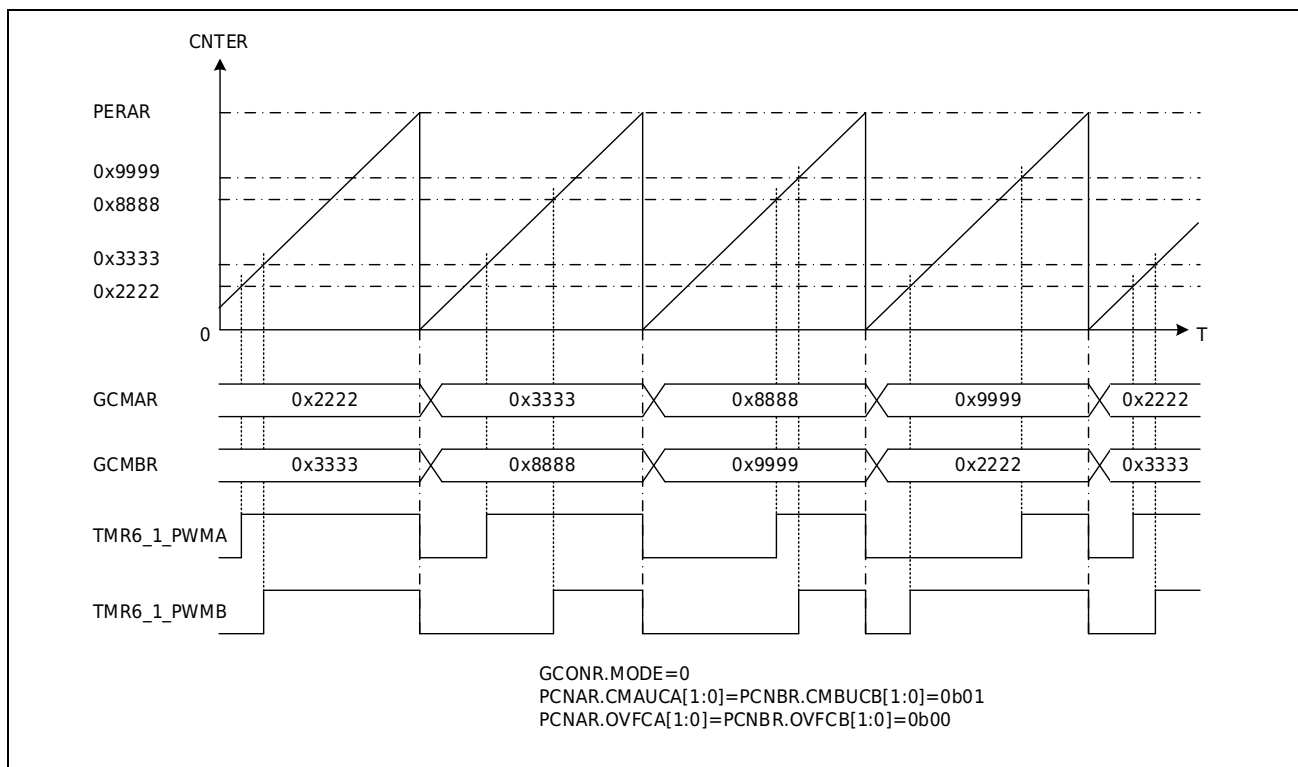


图 18-17 单边对齐独立 PWM

18.3.14.2 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 TMR6_<t>_PWMA、TMR6_<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 18-18 所示, 是 TMR6_<t>_PWMA 端口和 TMR6_<t>_PWMB 端口实现双边对称独立输出 PWM。

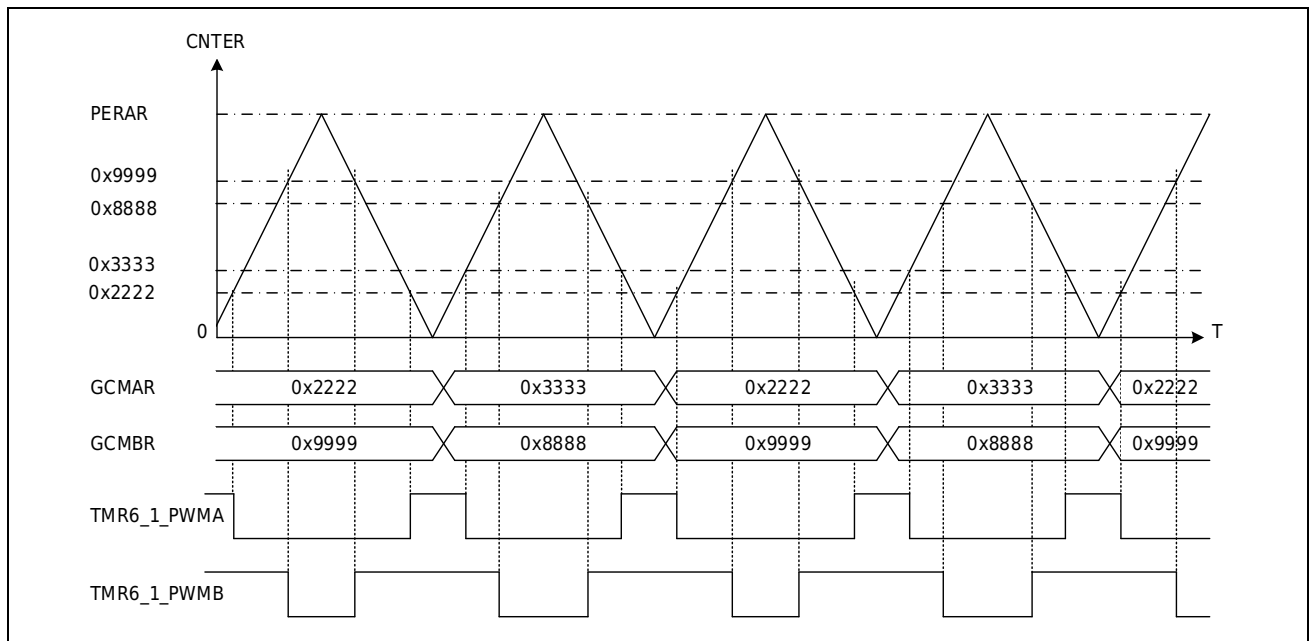


图 18-18 双边对称独立 PWM

18.3.14.3 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 TMR6_<t>_PWMA 和 TMR6_<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 TMR6_<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 18-19 为软件设定 GCMBR 互补 PWM 波的输出例。

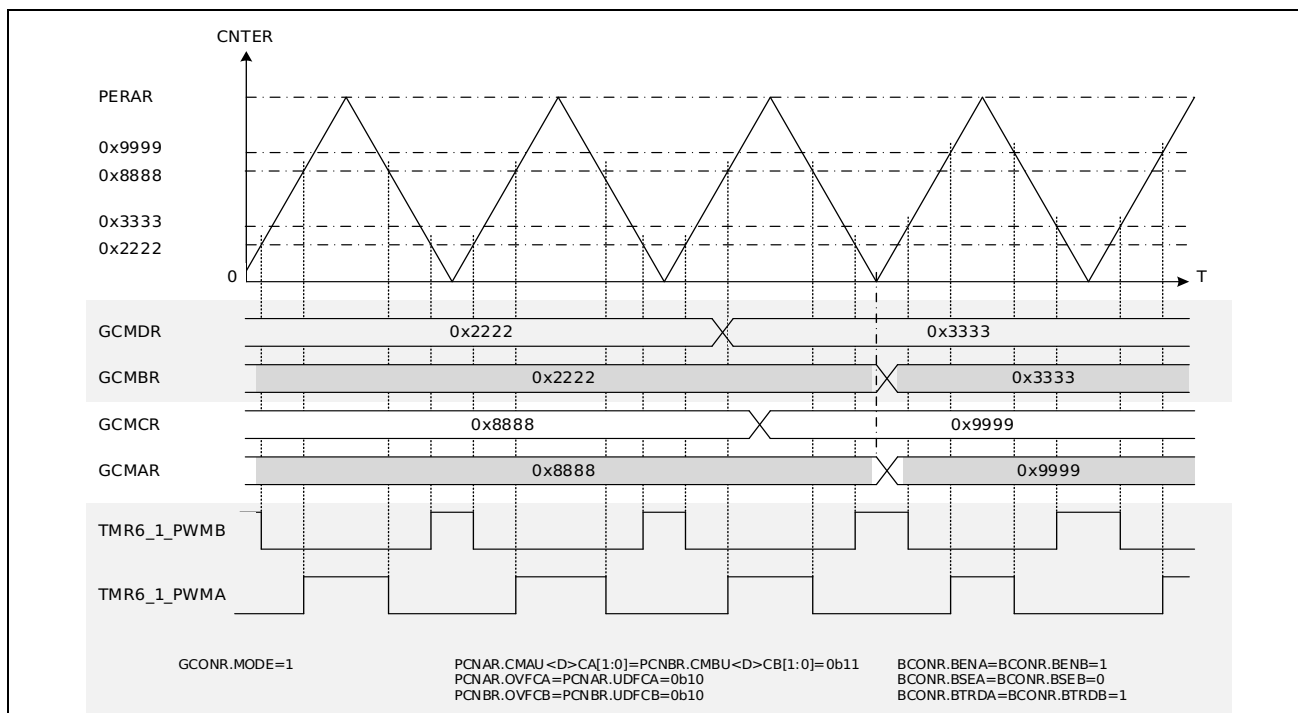


图 18-19 软件设定 GCMBR 互补 PWM 波输出

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 TMR6_<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 18-20 为硬件设定 GCMBR 互补 PWM 波输出例。

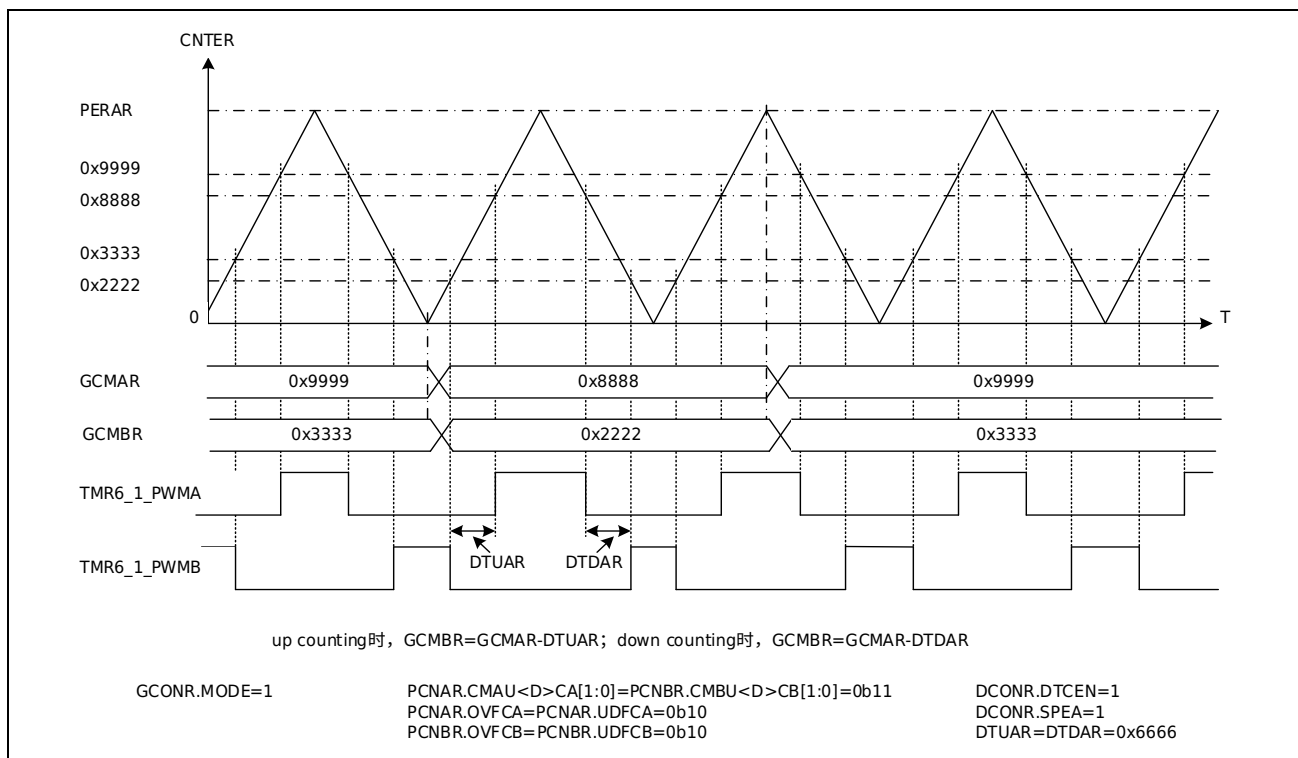


图 18-20 硬件设定 GCMBR 互补 PWM 波输出

18.3.14.4 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 TMR6_<t>_PWMA、TMR6_<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, TMR6_<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 TMR6_<t>_PWMA 端口上可实现双边非对称 PWM 输出。如图 18-21 所示, 是 TMR6_<t>_PWMA 端口的双边非对称 PWM 输出。

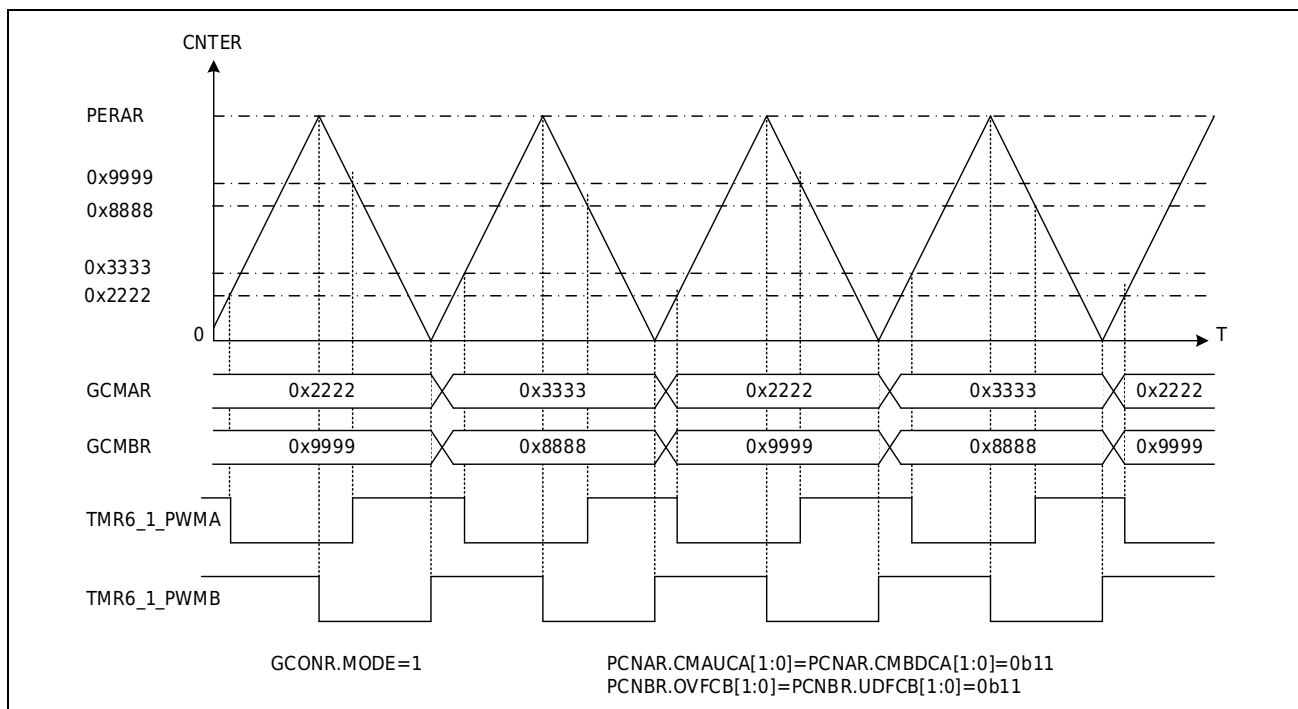


图 18-21 双边非对称 PWM 输出

18.3.14.5 单元间多相 PWM 输出

每个单元的 TMR6_<t>_PWMA、TMR6_<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 18-22，单元 1、单元 3、单元 1、单元 4 组合输出 8 相单边对齐独立 PWM；如图 18-23，单元 1、单元 3、单元 1、单元 4 组合输出 4 相双边对称互补 PWM。

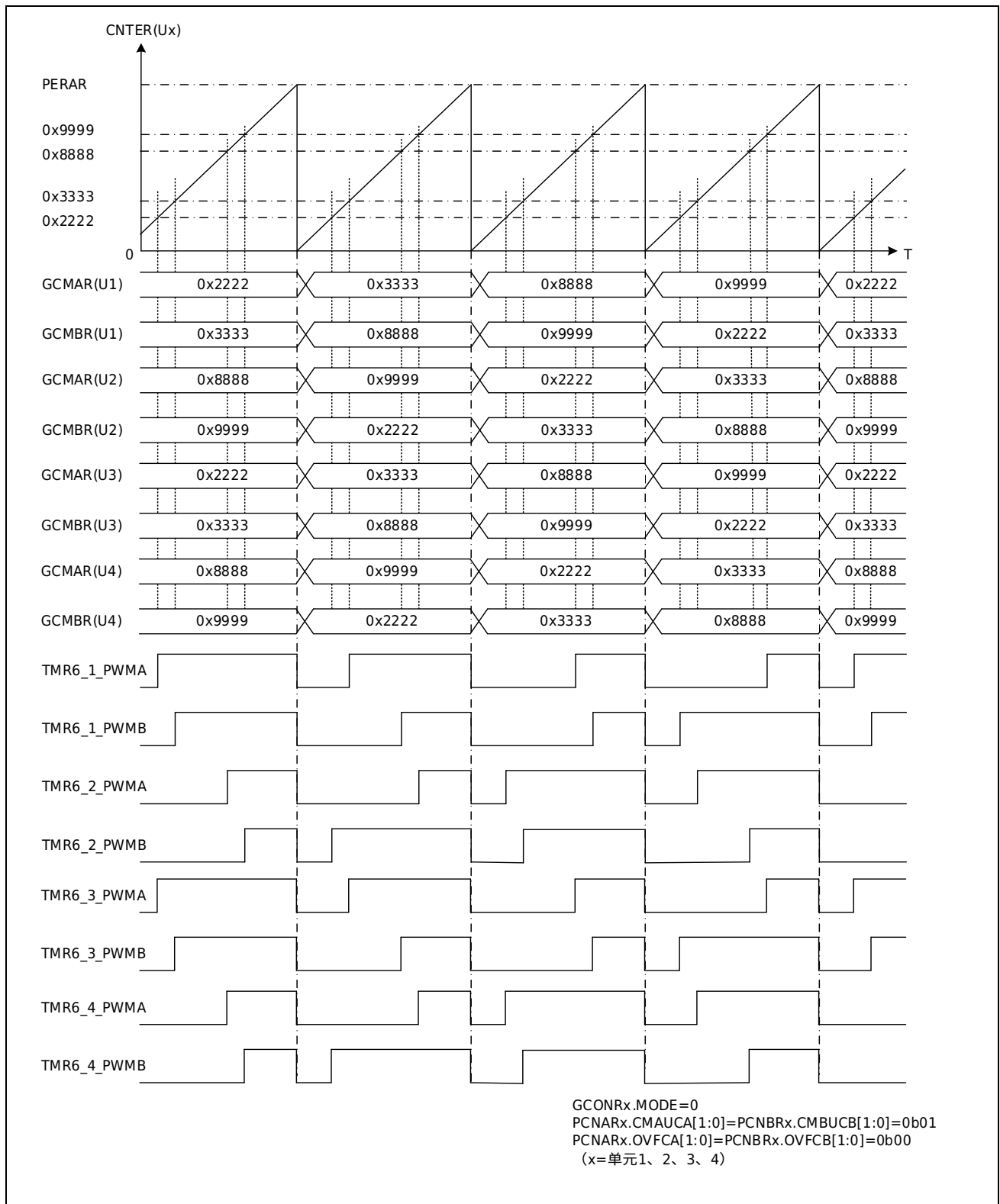


图 18-22 8相单边对齐独立 PWM

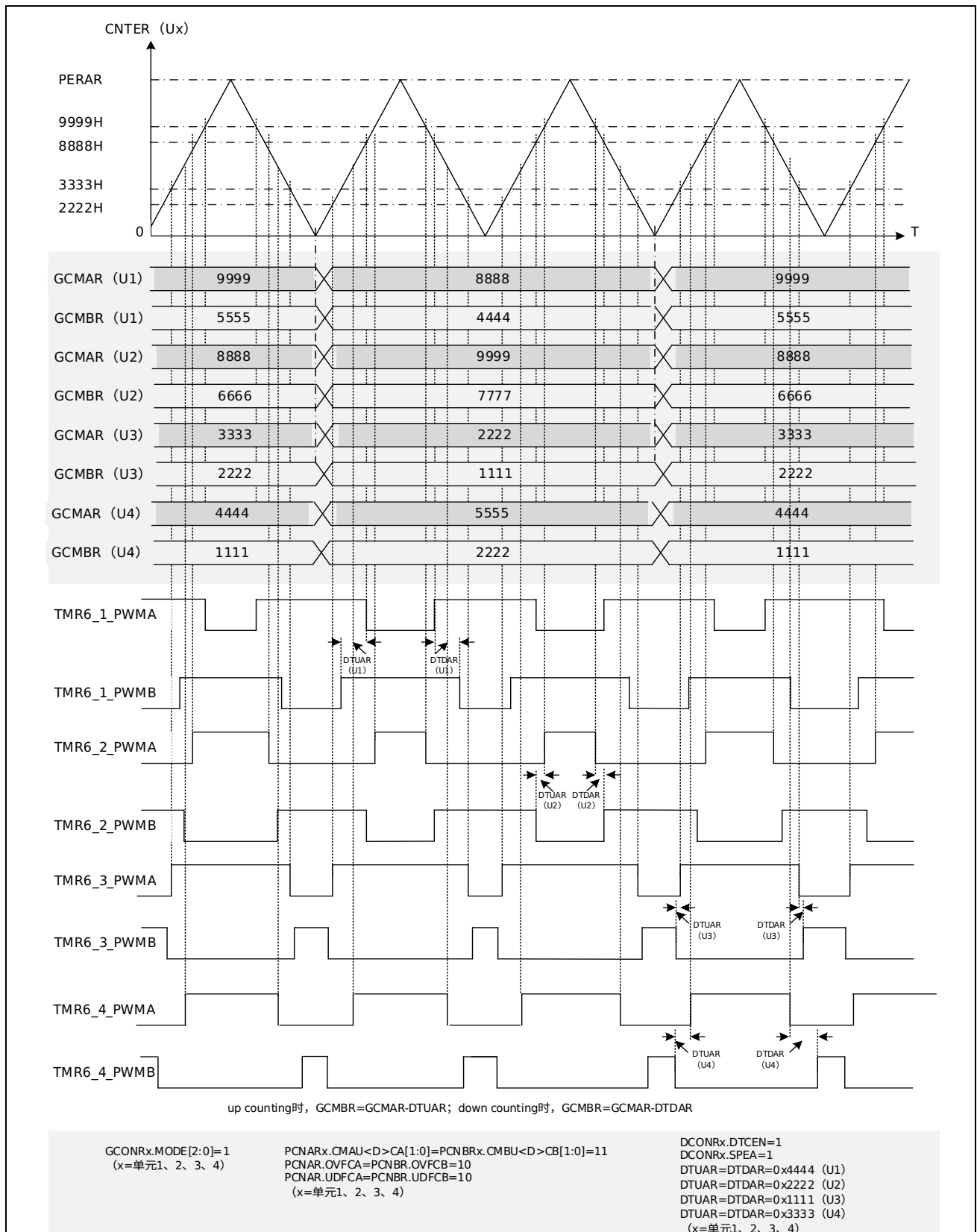


图 18-23 带死区时间 4 相双边对称互补 PWM

18.3.15 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.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 18-24 所示是周期间隔有效请求信号的动作用例。

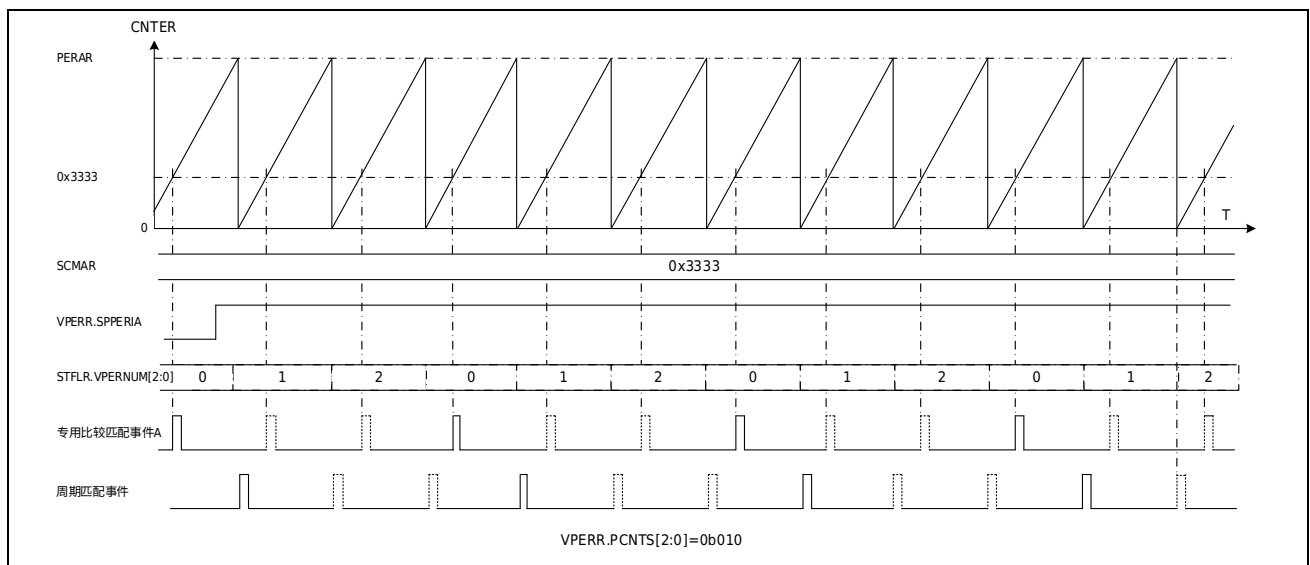


图 18-24 周期间隔有效请求信号动作

18.3.16 正交编码计数

将 TMR6_<t>_PWMA 输入看作 AIN 输入、TMR6_<t>_PWMB 输入看作 BIN 输入、TMR6_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 2 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。

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 TMR6_<t>_PWMA 和 TMR6_<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。

18.3.16.1 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 18-25 所示。

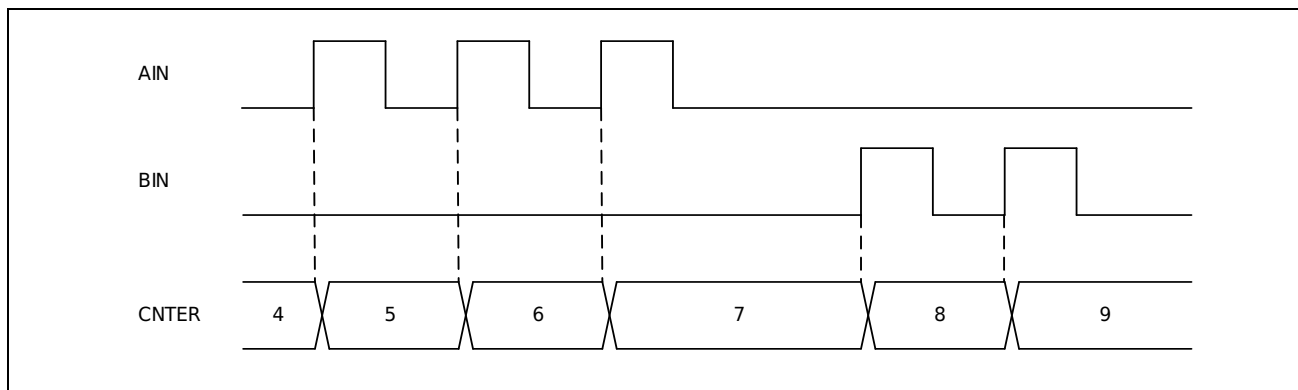


图 18-25 位置模式-基本计数

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 18-26~图 18-28 所示。

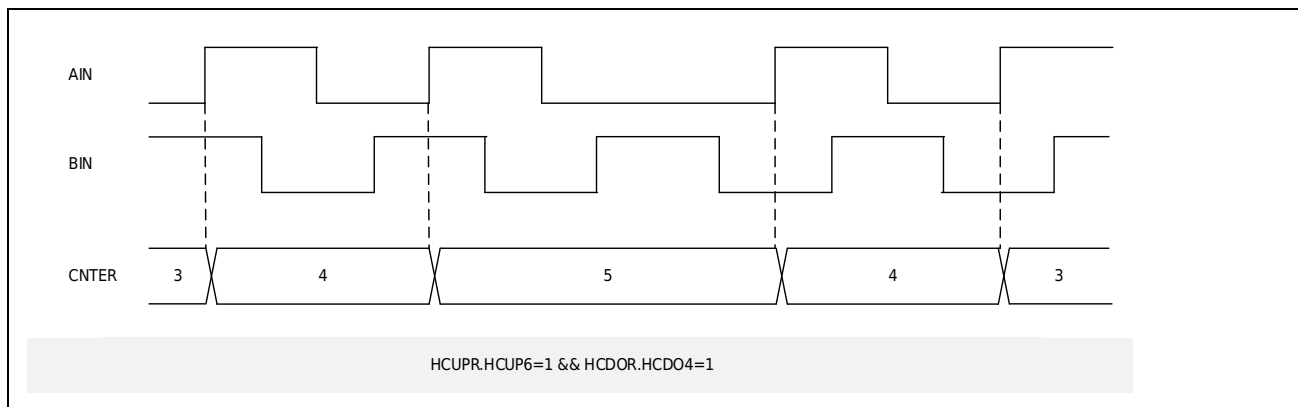


图 18-26 位置计数模式-相位差计数 (1 倍计数)

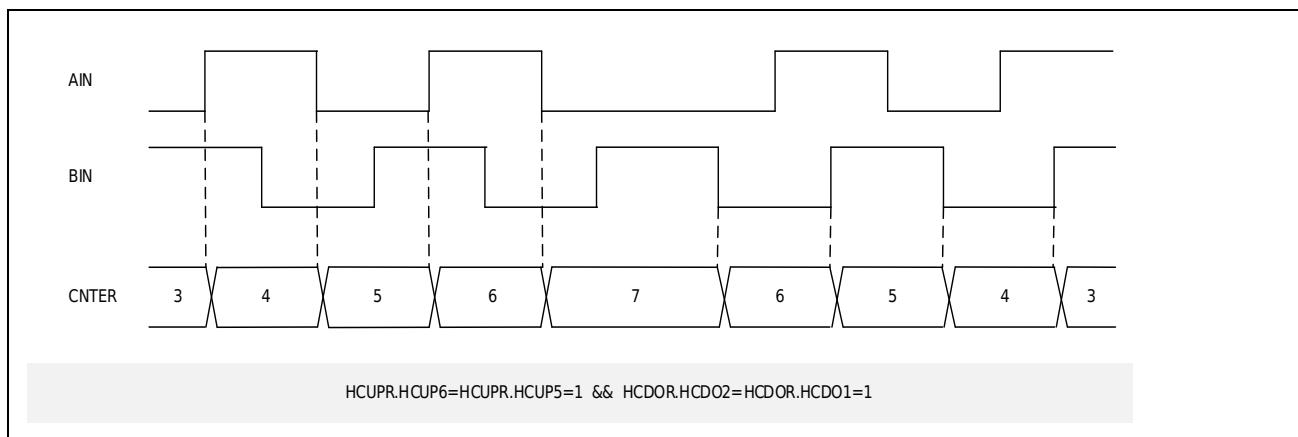


图 18-27 位置计数模式-相位差计数 (2 倍计数)

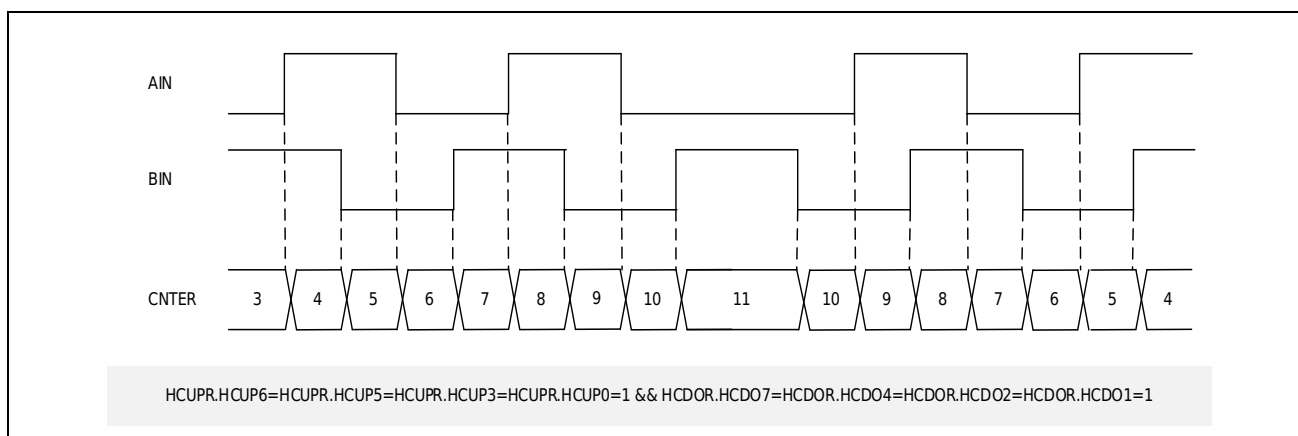


图 18-28 位置计数模式-相位差计数 (4 倍计数)

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 18-29 所示。

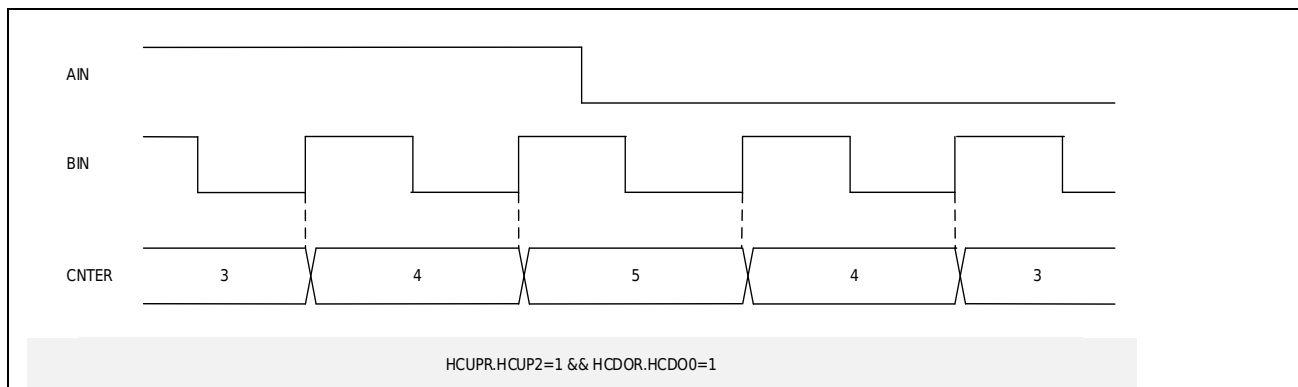


图 18-29 位置计数模式-方向计数

18.3.16.2 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 18-30 所示。

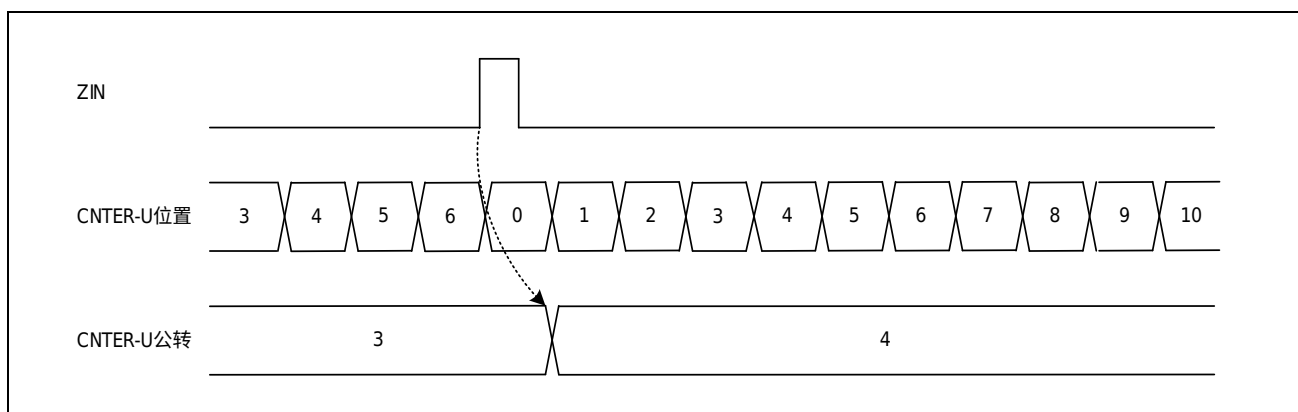


图 18-30 公转计数模式-Z 相计数

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（TMR6_TRGSELx, x=0~3）中的事件编号设定为位置计数单元的上溢或下溢事件。具体的寄存器内容详见【Timer6 硬件触发事件选择寄存器（TMR6_TRGSELx, x=0~3）】章节，具体事件编号参考【中断控制器（INTC）】章节。如下图 18-31 所示。

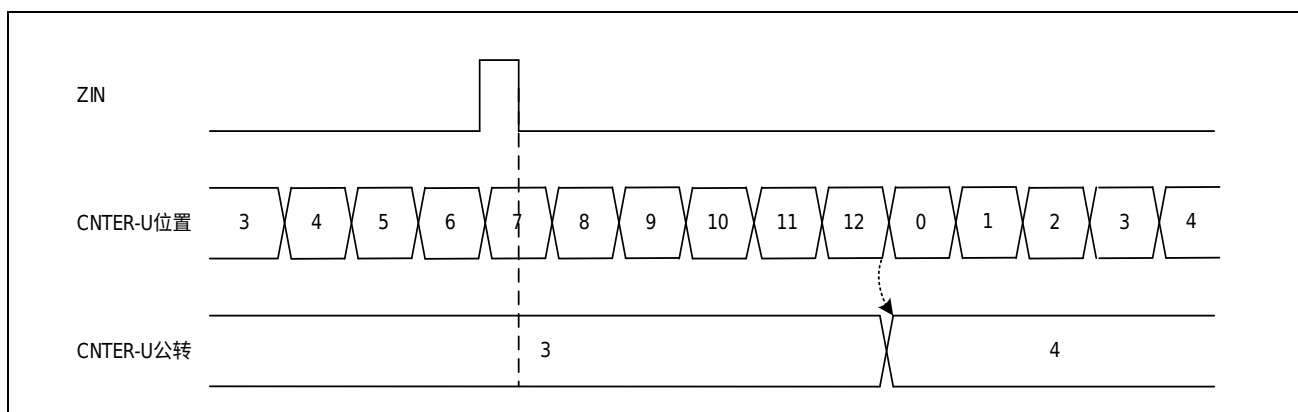


图 18-31 公转计数模式-位置溢出计数

混计数

混计数是指上述 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 18-32 所示。

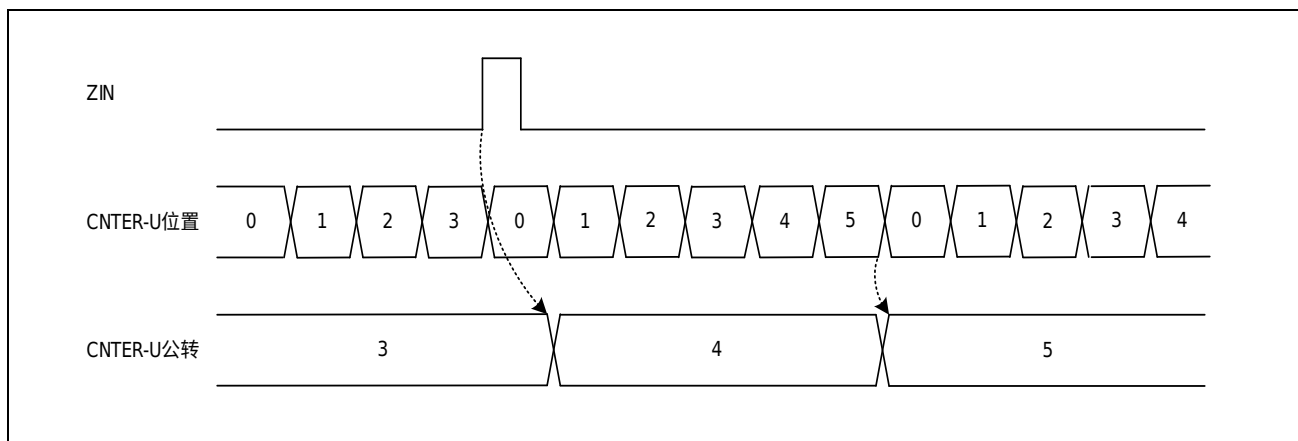


图 18-32 公转计数模式-混计数

18.3.16.3 Z 相动作屏蔽

在公转计数模式的 Z 相计数功能或混计数功能时，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 18-33 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。

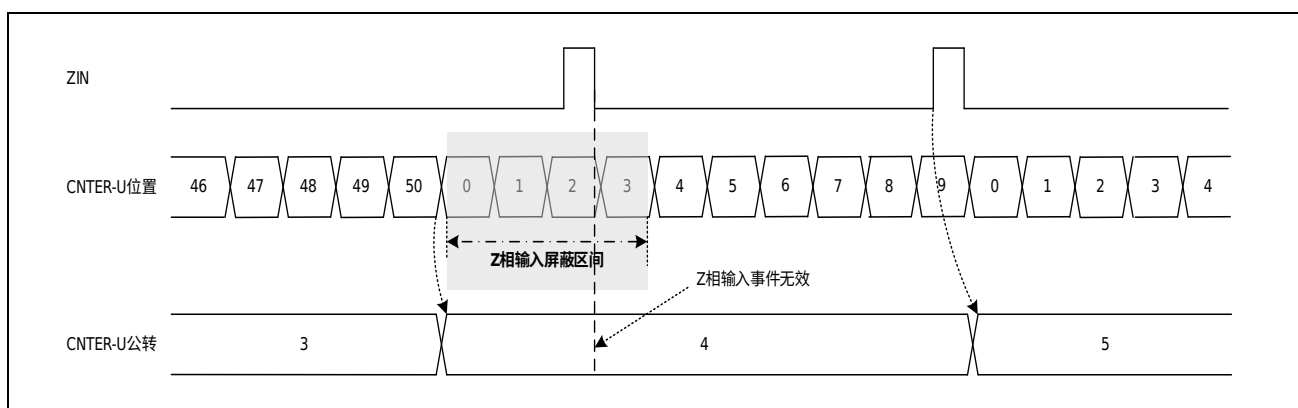


图 18-33 公转计数模式-混合计数 Z 相屏蔽动作例 1

图 18-34 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。

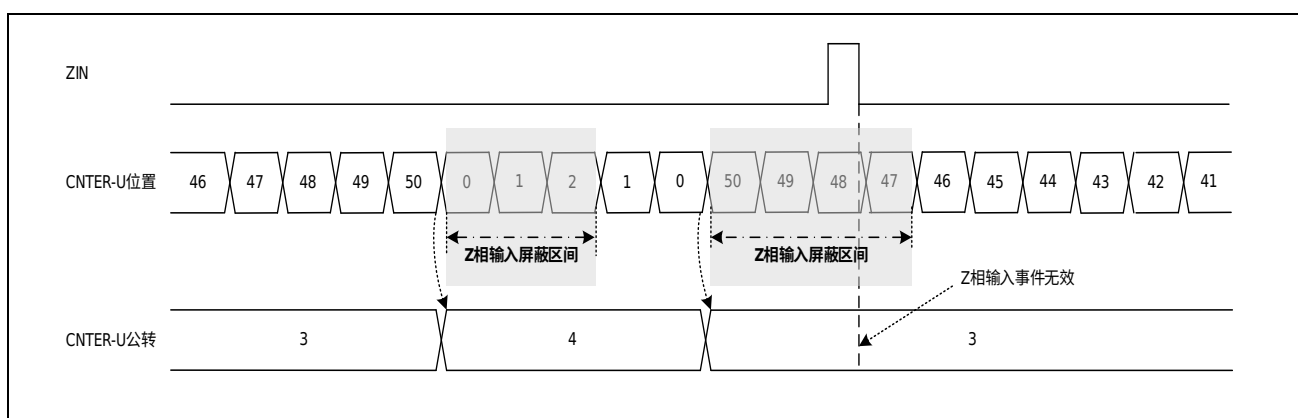


图 18-34 公转计数模式-混合计数 Z 相屏蔽动作例 2

18.3.17 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 1 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 (PCNAR.EMBSA) 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【紧急刹车模块 (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 PCNAR.EMBCA的设定决定)。例如, 若 PCNAR.EMBCA=01 设定时, 则在 TMR6_<t>_PWMA 端口正常输出期间, 若产生 EMB 事件, 则 TMR6_<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 (PCNAR.EMBRA=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 (PCNAR.EMBRA=0b01、0b10、0b11; 这种方式称之为 Cycle By Cycle 方式释放)。

注:

当 EMB 的事件释放方式配置为状态释放时 (通过 EMB_RLSSEL 寄存器配置), 且 EMB 的事件源配置外部端口或者内部电压比较器比较结果有效时, 必须保证外部端口和内部电压比较器比较结果的电平宽度大于 3 个计数时钟周期, 否则 PWM 输出上会出现最大 2 个计数时钟周期的毛刺。当 EMB 的事件释放方式配置为软件释放时 (通过 EMB_RLSSEL 寄存器配置), 且 EMB 的事件源配置 PWM 输出端口发生同相有效时, 在发生 PWM 同相输出到 PWM 进入安全状态之前最大会出现 2 个计数时钟周期的毛刺。

18.3.18 功能汇总表

Timer6 的锯齿波模式和三角波 A、B 模式下, 主要功能的功能汇总表如下表 18-3 所示。

表 18-3 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	PCNAR.STACA
	停止时	支持	支持	PCNAR.STPCA
	上溢时	支持	支持	PCNAR.OVFCA
	下溢时	支持	支持	PCNAR.UDFCA
	计数匹配时 (Up Counting)	支持	支持	PCNAR.CMAUCA
计数匹配时	支持	支持	PCNAR.CMADCA	

PWM输出功能			锯齿波	三角波	相关主要寄存器	
		(Down Counting)				
	缓存传送	周期基准值	单缓存	支持	支持	PERBR->PERAR
			双缓存	支持	支持	PERCR->PERBR PERBR->PERAR
		比较基准值	单缓存	支持	支持	GCMDR->GCMBR GCMCR->GCMAR
			双缓存	支持	支持	GCMFR->GCMDR GCMER->GCMCR GCMDR->GCMBR GCMCR->GCMAR
EMB			支持	支持	PCNA.EMBCA	
互补PWM 输出	端口状态控制	启动时		支持	支持	PCNA.R.STACA
		停止时		支持	支持	PCNA.R.STPCA
		上溢时		支持	支持	PCNA.R.OVFC A
		下溢时		支持	支持	PCNA.R.UDFCA
		计数匹配时 (Up Counting)		支持	支持	PCNA.R.CMA.UCA
		计数匹配时 (Down Counting)		支持	支持	PCNA.R.CMA.DCA
	缓存传送	周期基准值	单缓存	支持	支持	PERBR->PERAR
			双缓存	支持	支持	PERCR->PERBR PERBR->PERAR
		比较基准值	单缓存	支持	支持	GCMDR->GCMBR GCMCR->GCMAR
			双缓存	支持	支持	GCMFR->GCMDR GCMER->GCMCR GCMDR->GCMBR GCMCR->GCMAR
		死区基准值	单缓存	支持	支持	DTUBR->DTUAR DTDBR->DTDAR
	无死区PWM输出			支持	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=GCMA R-DTUA R GCNBR=GCMA R-DTDA R
	EMB			支持	支持	PCNA.R.EMBCA

18.4 中断和事件

18.4.1 中断

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。

18.4.1.1 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。

18.4.1.2 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）。

18.4.1.3 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）。

18.4.2 事件

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。

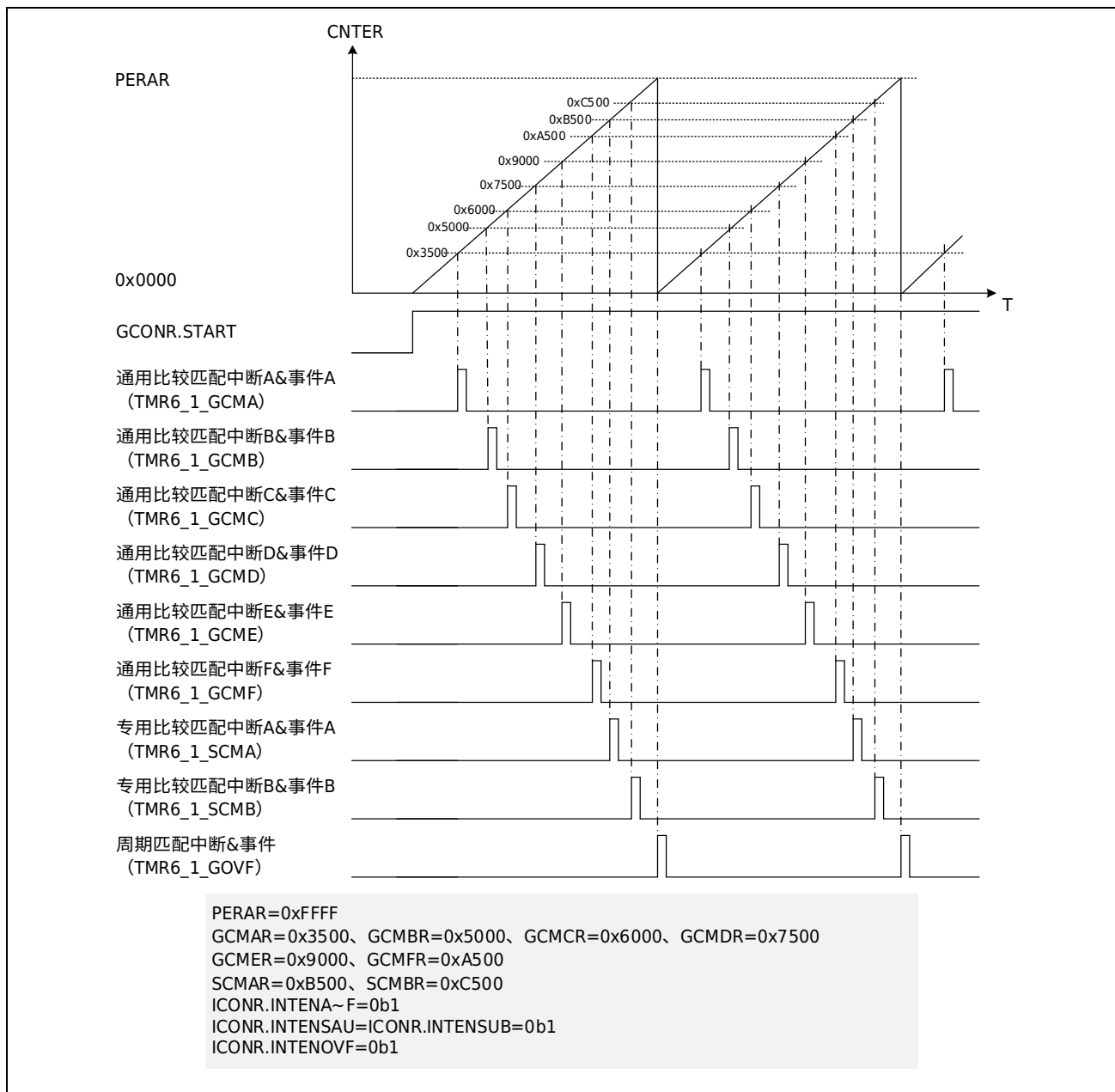


图 18-35 锯齿波模式时中断&事件输出例

18.5 应用举例

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。

18.5.1 基本计数及中断动作

1. 设定通用周期基准值 (PERAR)；
2. 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；
3. 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；
4. 设定内部计数时钟分频 (GCONR.CKDIV[3:0])；
5. 设定波形模式 (GCONR.MODE)；
6. 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定)；
7. 启动计数器 (GCONR.START=1)。

18.5.2 比较输出及中断动作

1. 设定通用周期基准值 (PERAR)；
2. 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)；
3. 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；
4. 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)；
5. 设定内部计数时钟分频 (GCONR.CKDIV[3:0])；
6. 设定波形模式 (GCONR.MODE)；
7. 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定)；
8. 设定各通道比较输出模式 (PCNAR.CAPMDA=0、PCNBR.CAPMDB=0)；
9. 设定各通道输出使能 (PCNAR.OUTENA=1、PCNBR.OUTENB=1)；
10. 启动计数器 (GCONR.START=1)。

18.5.3 捕获输入及中断动作

1. 设定通用周期基准值 (PERAR);
2. 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;
3. 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);
4. 设定内部计数时钟分频 (GCONR.CKDIV[3:0]);
5. 设定波形模式 (GCONR.MODE);
6. 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定);
7. 设定捕获输入模式 (PCNAR.CAPMDA=1、PCNBR.CAPMDB=1);
8. 启动计数器 (GCONR.START=1);
9. 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。

18.5.4 缓存传送动作 (周期基准值)

1. 设定需要的通用周期基准值 (PERAR、PERBR、PERCR);
2. 设定单、双缓存传送方式 (BCONR.BSEP);
3. 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);
4. 设定内部计数时钟分频 (GCONR.CKDIV[3:0]);
5. 设定波形模式 (GCONR.MODE);
6. 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定);
7. 设定缓存功能有效 (BCONR.BENP=1);
8. 启动计数器 (GCONR.START=1);
9. 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))。

18.5.5 缓存传送动作（通用比较基准值）

1. 设定需要的通用比较基准值（GCMAR、GCMCR、GCMER、GCMBR、GCMDR、GCMFR）；
2. 设定各通道单、双缓存传送方式（BCONR.BSEA、BCONR.BSEB）；
3. 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）；
4. 设定内部计数时钟分频（GCONR.CKDIV[3:0]）；
5. 设定波形模式（GCONR.MODE）；
6. 设定计数方向（仅在锯齿波模式 GCONR.MODE=0 时需要设定）；
7. 设定各通道缓存功能有效（BCONR.BENA=1、BCONR.BENB=1）；
8. 启动计数器（GCONR.START=1）；
9. 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））。

18.5.6 缓存传送动作（专用比较基准值）

1. 设定需要的专用比较基准值（SCMAR、SCMCR、SCMER、SCMBR、SCMDR、SCMFR）；
2. 设定各通道单、双缓存传送方式（BCONR.BESPA、BCONR.BESP B）；
3. 设定各通道缓存传送时间点（BCONR.BTRUSPA、BCONR.BTRDSP A、BCONR.BTRUSPB、BCONR.BTRDSP B，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）；
4. 设定内部计数时钟分频（GCONR.CKDIV[3:0]）；
5. 设定波形模式（GCONR.MODE）；
6. 设定计数方向（仅在锯齿波模式 GCONR.MODE=0 时需要设定）；
7. 设定各通道缓存功能有效（BCONR.BENSPA=1、BCONR.BENSP B=1）；
8. 启动计数器（GCONR.START=1）；
9. 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.BESPA=0 时）、SCMER->SCMCR->SCMAR（BCONR.BESPA=1 时）、SCMDR->SCMBR（BCONR.BESP B=0 时）、SCMFR->SCMDR->SCMBR（BCONR.BESP B=1 时））。

18.5.7 缓存传送动作（死区基准值）

1. 设定需要的死区时间基准值（DTUAR、DTUBR、DTDAR、DTDBR）；
2. 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）；
3. 设定内部计数时钟分频（GCONR.CKDIV[3:0]）；
4. 设定波形模式（GCONR.MODE）；
5. 设定计数方向（仅在锯齿波模式 GCONR.MODE=0 时需要设定）；
6. 设定缓存功能有效（DCONR.DTBENU=1、DCONR.DTBEND=1）；
7. 设定硬件死区功能有效（DCONR.DTCEN=1）；
8. 启动计数器（GCONR.START=1）；
9. 等待对应缓存传送时间点，发生缓存动作（DTUBR->DTUAR、DTDBR->DTDAR）。

18.5.8 同步启动动作（软件方式）

1. 参考【基本计数及中断动作】章节的步骤 1~6，对需要同步启动的各个单元做设定；
2. 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）。

18.5.9 同步启动动作（硬件方式）

1. 设定通用周期基准值（PERAR）；
2. 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；
3. 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；
4. 设定硬件启动条件（通过 HSTAR.HSTAx 选择，x=8~31）；
5. 设定硬件启动使能（HSTAR.STAS=1）；
6. 重复上述 1~5 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 4 的设定要一致）；
7. 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。

18.5.10 正交编码计数动作（2相）

1. 设定通用周期基准值（PERAR）；
2. 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；
3. 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；
4. 设定需要的硬件向上计数条件（通过 HCUPR.HCUPx 选择，x=0~7）；
5. 设定需要的硬件向下计数条件（通过 HCDOR.HCDOx 选择，x=0~7）；
6. 启动计数器（GCONR.START=1）；
7. 等待设定的正交编码计数事件产生，确认计数器正常计数。

18.5.11 正交编码计数动作（3相）

1. 参考【正交编码计数动作（2相）】章节的1~5步骤，对位置计数单元做设定；
2. 设定位置计数单元的硬件清零条件（通过 HCLRR.HCLR_x 选择，x=16~31）；
3. 设定位置计数单元硬件清零使能（HCLRR.CLES=1）；
4. 设定公转计数单元的通用周期基准值（PERAR）；
5. 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；
6. 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；
7. 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.HCUP_x 选择，x=16~31，此处的设定事件应和步骤 4 中位置计数单元设定的事件一致）；
8. 设定公转计数单元的硬件向上计数条件 2（位置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）；
9. 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）；
10. 设定 TMR6_TRGSEL0 中的触发源编号为位置计数单元的计数上溢事件（寄存器内容详见【Timer6 硬件触发事件选择寄存器（TMR6_TRGSEL_x, x=0~3）】章节，上溢事件编号参考【中断控制器（INTC）】

章节);

11. 设定 TMR6_TRGSEL1 中的触发源编号为位置计数单元的计数下溢事件 (寄存器内容详见【Timer6 硬件触发事件选择寄存器 (TMR6_TRGSELx, x=0~3)】章节, 下溢事件编号参考【中断控制器 (INTC)】章节);
12. 启动公转计数单元计数器 (GCONR.START=1);
13. 启动位置计数单元计数器 (GCONR.START=1);
14. 等待设定的 AIN、BIN、ZIN 相计数事件产生, 确认计数器正常计数。

18.5.12 单路 PWM 输出

1. 参考【比较输出及中断动作】章节步骤 1-10 的设定 (每个单元内部的 2 个 PWM 通道 TMR6_<t>_PWMA 和 TMR6_<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)。

18.5.13 互补 PWM 输出 (软件死区)

1. 设定通用周期基准值 (PERAR);
2. 设定通用比较基准值 A (GCMAR)、通用比较基准值 B (GCMBR);
3. 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;
4. 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);
5. 设定内部计数时钟分频 (GCONR.CKDIV[3:0]);
6. 设定波形模式为三角波模式 (GCONR.MODE=1);
7. 设定比较输出模式 (PCNAR.CAPMDA=0、PCNBR.CAPMDB=0);
8. 设定输出使能 (PCNAR.OUTENA=1、PCNBR.OUTENB=1);
9. 启动计数器 (GCONR.START=1)。

18.5.14 互补 PWM 输出 (硬件死区)

1. 设定通用周期基准值 (PERAR);
2. 设定通用比较基准值 A (GCMAR)、死区时间基准值 (DTUAR、DTDAR);
3. 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；

4. 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）；
5. 设定内部计数时钟分频（GCONR.CKDIV[3:0]）；
6. 设定波形模式为三角波模式（GCONR.MODE=1）；
7. 设定各通道比较输出模式（PCNAR.CAPMDA=0、PCNBR.CAPMDB=0）；
8. 设定各通道输出使能（PCNAR.OUTENA=1、PCNBR.OUTENB=1）；
9. 设定硬件死区功能有效（DCONR.DTCEN=1）；
10. 启动计数器（GCONR.START=1）。

18.5.15 EMB 监控及中断动作

1. 参考【互补 PWM 输出（软件死区）】章节的步骤 1-8 或【互补 PWM 输出（硬件死区）】章节步骤 1-9，对互补 PWM 输出动作进行设定；
2. 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）；
3. 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）；
4. 设定选择从 EMB 模块输入的 EMB 事件源（PCNAR.EMBSA、PCNBR.EMBSB）；
5. 设定 EMB 模块的相关寄存器（包括 EMB 中断许可寄存器（EMB_INTEN0~3）、EMB 控制寄存器 1/2（EMB_CTL1/2_0~3）、EMB 控制 PWM 输出释放方式选择寄存器（EMB_RLSSEL0~3）等）；
6. 启动计数器（GCONR.START=1），EMB 模块实时监控系统状态。

18.6 寄存器描述

表 18-4 Timer6 基地址

名称	基地址	描述
Timer6_1	0x4003 9000	Timer6_1基地址
Timer6_2	0x4003 9400	Timer6_2基地址
Timer6_3	0x4003 9800	Timer6_3基地址
Timer6_4	0x4003 9C00	Timer6_4基地址
Timer6_COMMON	0x4003 9300	Timer6公共寄存器基地址

表 18-5 Timer6 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
TMR6_CNTER	通用计数值寄存器	0x0000	32	0x0000 0000
TMR6_UPDAR	通用刷新值寄存器	0x0004	32	0x0000 0000
TMR6_PERAR	通用周期基准值寄存器A	0x0040	32	0x0000 FFFF
TMR6_PERBR	通用周期基准值寄存器B	0x0044	32	0x0000 FFFF
TMR6_PERCR	通用周期基准值寄存器C	0x0048	32	0x0000 FFFF
TMR6_GCMAR	通用比较基准值寄存器A	0x0080	32	0x0000 FFFF
TMR6_GCMBR	通用比较基准值寄存器B	0x0084	32	0x0000 FFFF
TMR6_GCMCR	通用比较基准值寄存器C	0x0088	32	0x0000 FFFF
TMR6_GCMDR	通用比较基准值寄存器D	0x008C	32	0x0000 FFFF
TMR6_GCMER	通用比较基准值寄存器E	0x0090	32	0x0000 FFFF
TMR6_GCMFR	通用比较基准值寄存器F	0x0094	32	0x0000 FFFF
TMR6_SCMAR	专用比较基准值寄存器A	0x00C0	32	0x0000 FFFF
TMR6_SCMBR	专用比较基准值寄存器B	0x00C4	32	0x0000 FFFF
TMR6_SCMCR	专用比较基准值寄存器C	0x00C8	32	0x0000 FFFF
TMR6_SCMDR	专用比较基准值寄存器D	0x00CC	32	0x0000 FFFF
TMR6_SCMER	专用比较基准值寄存器E	0x00D0	32	0x0000 FFFF
TMR6_SCMFR	专用比较基准值寄存器F	0x00D4	32	0x0000 FFFF
TMR6_DTUAR	死区时间基准值寄存器UA	0x0100	32	0x0000 FFFF
TMR6_DTDAR	死区时间基准值寄存器DA	0x0104	32	0x0000 FFFF
TMR6_DTUBR	死区时间基准值寄存器UB	0x0108	32	0x0000 FFFF
TMR6_DTDDBR	死区时间基准值寄存器DB	0x010C	32	0x0000 FFFF
TMR6_GCONR	通用控制寄存器	0x0140	32	0x0000 0002
TMR6_ICONR	中断控制寄存器	0x0144	32	0x0000 0000
TMR6_BCONR	缓存控制寄存器	0x0148	32	0x0000 0000
TMR6_DCONR	死区控制寄存器	0x014C	32	0x0000 0000
TMR6_PCNAR	端口控制寄存器A	0x0154	32	0x0000 0000
TMR6_PCNBR	端口控制寄存器B	0x0158	32	0x0000 0000

寄存器	描述	偏移地址	位宽	复位值
TMR6_FCNGR	滤波控制寄存器G	0x015C	32	0x0000 0000
TMR6_VPERR	有效周期寄存器	0x0160	32	0x0000 0000
TMR6_STFLR	状态标志寄存器	0x0164	32	0x8000 0000
TMR6_HSTAR	硬件启动事件选择寄存器	0x0180	32	0x0000 0000
TMR6_HSTPR	硬件停止事件选择寄存器	0x0184	32	0x0000 0000
TMR6_HCLRR	硬件清零事件选择寄存器	0x0188	32	0x0000 0000
TMR6_HUPDR	硬件刷新事件选择寄存器	0x018C	32	0x0000 0000
TMR6_HCPAR	硬件捕获事件选择寄存器A	0x0190	32	0x0000 0000
TMR6_HCPBR	硬件捕获事件选择寄存器B	0x0194	32	0x0000 0000
TMR6_HCUPR	硬件递加事件选择寄存器	0x0198	32	0x0000 0000
TMR6_HCDOR	硬件递减事件选择寄存器	0x019C	32	0x0000 0000

表 18-6 Timer6 公共寄存器列表

寄存器	描述	偏移地址	位宽	复位值
TMR6_FCNTNTR	公共滤波控制寄存器	0x00EC	32	0x0000 0000
TMR6_SSTAR	公共软件同步启动控制寄存器	0x00F0	32	0x0000 0000
TMR6_SSTPR	公共软件同步停止控制寄存器	0x00F4	32	0x0000 0000
TMR6_SCLLR	公共软件同步清零控制寄存器	0x00F8	32	0x0000 0000
TMR6_SUPDR	公共软件同步刷新控制寄存器	0x00FC	32	0x0000 0000

注意：

1. 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 4 个单元 Timer6 所共有。
2. 滤波控制寄存器（TMR6_FCNTNTR）是 1 个独立于单元的寄存器，为 4 个单元 Timer6 所共有。

18.6.1 通用计数值寄存器 (TMR6_CNTER)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”	R
b15~b0	CNT[15:0]	计数值	当前定时器的计数值 注: 1. 锯齿波向上计数模式, CNT初始值设置为0; 锯齿波向下计数模式, CNT初始值设置为pear; 2. 三角波模式计数CNT初始值设置为1;	RW

18.6.2 通用刷新值寄存器 (TMR6_UPDAR)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”	R
b15~b0	UPDA[15:0]	刷新值	设定要更新进时器的刷新值	RW

18.6.3 通用周期基准值寄存器 (TMR6_PERmR, m=A~C)

复位值: 0x0000 FFFF

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

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	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	RW

18.6.4 通用比较基准值寄存器 (TMR6_GCMmR, m=A~F)

复位值: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Res																
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
GCMA-F[15:0]																
位/位域	标记	位名	功能													读写
b31~b16	Res	保留位	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													RW

18.6.5 专用比较基准值寄存器 (TMR6_SCMmR, m=A~F)

复位值: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Res																
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
SCMA-F[15:0]																
位/位域	标记	位名	功能													读写
b31~b16	Res	保留位	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													RW

18.6.6 死区时间基准值寄存器 (TMR6_DTmnR, m=D、U, n=A、B)

复位值: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Res																
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
DTUA-B[15:0]/DTDA-B[15:0]																
位/位域	标记	位名	功能													读写
b31~b16	Res	保留位	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													RW

18.6.7 通用控制寄存器 (TMR6_GCONR)

复位值: 0x0000 0002

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res												ZMSK VAL[1:0]	ZMSK POS	ZMSK REV	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							OV STP	CKDIV[3:0]			Res	MODE	DIR	START	

位/位域	标记	位名	功能	读写
b31~b20	Res	保留位	读出时为“0”，写入时写“0”	RW
b19~b18	ZMSKVAL[1:0]	Z相输入屏蔽周期数	正交编码Z相输入被屏蔽的计数周期值 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	RW
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	RW
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	RW
b15~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b7~b4	CKDIV[2:0]	计数时钟选择	0000: PCLK0 0001: PCLK0/2 0010: PCLK0/4 0011: PCLK0/8 0100: PCLK0/16 0101: PCLK0/32 0110: PCLK0/64 0111: PCLK0/128 1000: PCLK0/256 1001: PCLK0/512 1010: PCLK0/1024	RW
b3	Res	保留位	读出时为“0”，写入时写“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: 递减计数 1: 递加计数	RW
b0	START	定时器启动	0: 定时器关闭 1: 定时器启动 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	RW

18.6.8 中断控制寄存器 (TMR6_ICONR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Res												INTEN SBD	INTEN SBU	INTEN SAD	INTEN SAU	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Res								INTEN DTE	INTEN UDF	INTEN OVF	INTEN F	INTEN E	INTEN D	INTEN C	INTEN B	INTEN A

位/位域	标记	位名	功能	读写
b31~b20	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW
b15~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW
b4	INTENE	计数匹配中断使能E	0: GCMEER寄存器与计数值相等时，该中断无效 1: GCMEER寄存器与计数值相等时，该中断使能	RW
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	RW
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	RW
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	RW
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	RW

18.6.9 缓存控制寄存器 (TMR6_BCONR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								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
Res				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/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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)	RW
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	RW
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	RW
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	RW
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)	RW
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	RW
b15~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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:	RW

			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	
b8	BENP	周期值缓存传送	0: 缓存传送无效 1: 缓存传送使能	RW
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	RW
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	RW
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)	RW
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	RW
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	RW
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	RW
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)	RW
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	RW

18.6.10 死区控制寄存器 (TMR6_DCONR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								DTB TRD	DTB TRU	DTB END	DTB ENU	Res	SEPA	DTC EN	

位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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)	RW
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)	RW
b3~b2	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b0	DTCEN	死区功能	0: 死区功能无效 1: 死区功能有效	RW

18.6.11 端口控制寄存器 A (TMR6_PCNAR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CAPMDA	Res		OUTENA	Res		EMBSA[1:0]		EMBRA[1:0]		EMBCA[1:0]		Res		FORCA[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMBDCA[1:0]		CMBUCA[1:0]		CMADCA[1:0]		CMAUCA[1:0]		UDFCA[1:0]		OVFCA[1:0]		STPCA[1:0]		STACA[1:0]	

位/位域	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	RW
b30~b29	Res	保留位	读出时为“0”，写入时写“0”	RW
b28	OUTENA	输出使能A	0: Timer6功能时的TMR6_<t>_PWMA端口输出无效 1: Timer6功能时的TMR6_<t>_PWMA端口输出有效	RW
b27~b26	Res	保留位	读出时为“0”，写入时写“0”	RW
b25~b24	EMBSA[1:0]	EMB事件时端口状态选择A	00: 选择EMB的Group6控制事件有效 01: 选择EMB的Group7控制事件有效 其他值禁止设定	RW
b23~b22	EMBRA[1:0]	EMB释放方式选择A	00: EMBSA[1:0]选择的EMB控制事件无效时，立即释放TMR6_<t>_PWMA端口 (One Shot) 01: EMBSA[1:0]选择的EMB控制事件无效时，等到计数器计数到上溢时释放TMR6_<t>_PWMA端口 (Cycle By Cycle 1) 10: EMBSA[1:0]选择的EMB控制事件无效时，等到计数器计数到下溢时释放TMR6_<t>_PWMA端口 (Cycle By Cycle 2) 11: EMBSA[1:0]选择的EMB控制事件无效时，等到计数器计数到上溢或下溢时释放TMR6_<t>_PWMA端口 (Cycle By Cycle 3)	RW
b21~b20	EMBCA[1:0]	EMB事件时端口状态设定A	00: 发生EMBSA[1:0]选择EMB控制事件时，TMR6_<t>_PWMA端口正常输出 01: 发生EMBSA[1:0]选择EMB控制事件时，TMR6_<t>_PWMA端口输出高阻态 10: 发生EMBSA[1:0]选择EMB控制事件时，TMR6_<t>_PWMA端口输出低电平 11: 发生EMBSA[1:0]选择EMB控制事件时，TMR6_<t>_PWMA端口输出高电平	RW
b19~b18	Res	保留位	读出时为“0”，写入时写“0”	RW
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，TMR6_<t>_PWMA端口输出设定为低电平 11: 下周期开始，TMR6_<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	RW
b15~b14	CMBDCA[1:0]	向下计数&&比较值B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GCMBR相等时，TMR6_<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，	RW

			<p>TMR6_<t>_PWMA端口输出设定为高电平</p> <p>10：在向下计数期间，定时器计数值与GCMBR相等时，TMR6_<t>_PWMA端口输出保持先前状态</p> <p>11：在向下计数期间，定时器计数值与GCMBR相等时，TMR6_<t>_PWMA端口输出设定为反转电平</p>	
b13~b12	CMBUCA[1:0]	向上计数&&比较值B匹配时端口状态设定A	<p>00：在向上计数期间，定时器计数值与GCMBR相等时，TMR6_<t>_PWMA端口输出设定为低电平</p> <p>01：在向上计数期间，定时器计数值与GCMBR相等时，TMR6_<t>_PWMA端口输出设定为高电平</p> <p>10：在向上计数期间，定时器计数值与GCMBR相等时，TMR6_<t>_PWMA端口输出保持先前状态</p> <p>11：在向上计数期间，定时器计数值与GCMBR相等时，TMR6_<t>_PWMA端口输出设定为反转电平</p>	RW
b11~b10	CMADCA[1:0]	向下计数&&比较值A匹配时端口状态设定A	<p>00：在向下计数期间，定时器计数值与GCMAR相等时，TMR6_<t>_PWMA端口输出设定为低电平</p> <p>01：在向下计数期间，定时器计数值与GCMAR相等时，TMR6_<t>_PWMA端口输出设定为高电平</p> <p>10：在向下计数期间，定时器计数值与GCMAR相等时，TMR6_<t>_PWMA端口输出保持先前状态</p> <p>11：在向下计数期间，定时器计数值与GCMAR相等时，TMR6_<t>_PWMA端口输出设定为反转电平</p>	RW
b9~b8	CMAUCA[1:0]	向上计数&&比较值A匹配时端口状态设定A	<p>00：在向上计数期间，定时器计数值与GCMAR相等时，TMR6_<t>_PWMA端口输出设定为低电平</p> <p>01：在向上计数期间，定时器计数值与GCMAR相等时，TMR6_<t>_PWMA端口输出设定为高电平</p> <p>10：在向上计数期间，定时器计数值与GCMAR相等时，TMR6_<t>_PWMA端口输出保持先前状态</p> <p>11：在向上计数期间，定时器计数值与GCMAR相等时，TMR6_<t>_PWMA端口输出设定为反转电平</p>	RW
b7~b6	UDFCA[1:0]	计数下溢端口状态设定A	<p>00：计数下溢时，TMR6_<t>_PWMA端口输出设定为低电平</p> <p>01：计数下溢时，TMR6_<t>_PWMA端口输出设定为高电平</p> <p>10：计数下溢时，TMR6_<t>_PWMA端口输出保持先前状态</p> <p>11：计数下溢时，TMR6_<t>_PWMA端口输出设定为反转电平</p>	RW
b5~b4	OVFCA[1:0]	计数上溢端口状态设定A	<p>00：计数上溢时，TMR6_<t>_PWMA端口输出设定为低电平</p> <p>01：计数上溢时，TMR6_<t>_PWMA端口输出设定为高电平</p> <p>10：计数上溢时，TMR6_<t>_PWMA端口输出保持先前状态</p> <p>11：计数上溢时，TMR6_<t>_PWMA端口输出设定为反转电平</p>	RW
b3~b2	STPCA[1:0]	计数停止端口状态设定A	<p>00：计数停止时，TMR6_<t>_PWMA端口输出设定为低电平</p> <p>01：计数停止时，TMR6_<t>_PWMA端口输出设定为高电平</p> <p>10：计数停止时，TMR6_<t>_PWMA端口输出保持先前状态</p> <p>11：计数停止时，TMR6_<t>_PWMA端口输出保持先前状态</p>	RW
b1~b0	STACA[1:0]	计数开始端口状态设定A	<p>00：计数开始时，TMR6_<t>_PWMA端口输出设定为低电平</p> <p>01：计数开始时，TMR6_<t>_PWMA端口输出设定为高电平</p> <p>10：计数开始时，TMR6_<t>_PWMA端口输出保持先前状态</p> <p>11：计数开始时，TMR6_<t>_PWMA端口输出保持先前状态</p>	RW

18.6.12 端口控制寄存器 B (TMR6_PCNBR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CAPMDB	Res		OUTENB	Res		EMBSB[1:0]		EMBRB[1:0]		EMBCB[1:0]		Res		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]		CMADCB[1:0]		CMAUCB[1:0]		UDFCB[1:0]		OVFCB[1:0]		STPCB[1:0]		STACB[1:0]	

位/位域	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	RW
b30~b29	Res	保留位	读出时为“0”，写入时写“0”	RW
b28	OUTENB	输出使能B	0: Timer6功能时的TMR6_<t>_PWMB端口输出无效 1: Timer6功能时的TMR6_<t>_PWMB端口输出有效	RW
b27~b26	Res	保留位	读出时为“0”，写入时写“0”	RW
b25~b24	EMBSB[1:0]	EMB事件时端口状态选择B	00: 选择EMB的Group6控制事件有效 01: 选择EMB的Group7控制事件有效 其他值禁止设定	RW
b23~b22	EMBRB[1:0]	EMB释放方式选择B	00: EMBSB[1:0]选择的EMB控制事件无效时，立即释放TMR6_<t>_PWMB端口 (One Shot) 01: EMBSB[1:0]选择的EMB控制事件无效时，等到计数器计数到上溢时释放TMR6_<t>_PWMB端口 (Cycle By Cycle 1) 10: EMBSB[1:0]选择的EMB控制事件无效时，等到计数器计数到下溢时释放TMR6_<t>_PWMB端口 (Cycle By Cycle 2) 11: EMBSB[1:0]选择的EMB控制事件无效时，等到计数器计数到上溢或下溢时释放TMR6_<t>_PWMB端口 (Cycle By Cycle 3)	RW
b21~b20	EMBCB[1:0]	EMB事件时端口状态设定B	00: 发生EMBSB[1:0]选择EMB控制事件时，TMR6_<t>_PWMB端口正常输出 01: 发生EMBSB[1:0]选择EMB控制事件时，TMR6_<t>_PWMB端口输出高阻态 10: 发生EMBSB[1:0]选择EMB控制事件时，TMR6_<t>_PWMB端口输出低电平 11: 发生EMBSB[1:0]选择EMB控制事件时，TMR6_<t>_PWMB端口输出高电平	RW
b19~b18	Res	保留位	读出时为“0”，写入时写“0”	RW
b17~b16	FORCB[1:0]	强制端口状态设定B	0x: 设定无效定 10: 下周期开始，TMR6_<t>_PWMB端口输出设定为低电平 11: 下周期开始，TMR6_<t>_PWMB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	RW
b15~b14	CMBDCB[1:0]	向下计数&&比较值B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GCMBR相等时，TMR6_<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，	RW

			<p>TMR6_<t>_PWMB端口输出设定为高电平</p> <p>10：在向下计数期间，定时器计数值与GCMBR相等时，TMR6_<t>_PWMB端口输出保持先前状态</p> <p>11：在向下计数期间，定时器计数值与GCMBR相等时，TMR6_<t>_PWMB端口输出设定为反转电平</p>	
b13~b12	CMBUCB[1:0]	向上计数&&比较值B匹配时端口状态设定B	<p>00：在向上计数期间，定时器计数值与GCMBR相等时，TMR6_<t>_PWMB端口输出设定为低电平</p> <p>01：在向上计数期间，定时器计数值与GCMBR相等时，TMR6_<t>_PWMB端口输出设定为高电平</p> <p>10：在向上计数期间，定时器计数值与GCMBR相等时，TMR6_<t>_PWMB端口输出保持先前状态</p> <p>11：在向上计数期间，定时器计数值与GCMBR相等时，TMR6_<t>_PWMB端口输出设定为反转电平</p>	RW
b11~b10	CMADCB[1:0]	向下计数&&比较值A匹配时端口状态设定B	<p>00：在向下计数期间，定时器计数值与GCMAR相等时，TMR6_<t>_PWMB端口输出设定为低电平</p> <p>01：在向下计数期间，定时器计数值与GCMAR相等时，TMR6_<t>_PWMB端口输出设定为高电平</p> <p>10：在向下计数期间，定时器计数值与GCMAR相等时，TMR6_<t>_PWMB端口输出保持先前状态</p> <p>11：在向下计数期间，定时器计数值与GCMAR相等时，TMR6_<t>_PWMB端口输出设定为反转电平</p>	RW
b9~b8	CMAUCB[1:0]	向上计数&&比较值A匹配时端口状态设定B	<p>00：在向上计数期间，定时器计数值与GCMAR相等时，TMR6_<t>_PWMB端口输出设定为低电平</p> <p>01：在向上计数期间，定时器计数值与GCMAR相等时，TMR6_<t>_PWMB端口输出设定为高电平</p> <p>10：在向上计数期间，定时器计数值与GCMAR相等时，TMR6_<t>_PWMB端口输出保持先前状态</p> <p>11：在向上计数期间，定时器计数值与GCMAR相等时，TMR6_<t>_PWMB端口输出设定为反转电平</p>	RW
b7~b6	UDFCB[1:0]	计数下溢端口状态设定B	<p>00：计数下溢时，TMR6_<t>_PWMB端口输出设定为低电平</p> <p>01：计数下溢时，TMR6_<t>_PWMB端口输出设定为高电平</p> <p>10：计数下溢时，TMR6_<t>_PWMB端口输出保持先前状态</p> <p>11：计数下溢时，TMR6_<t>_PWMB端口输出设定为反转电平</p>	RW
b5~b4	OVFCB[1:0]	计数上溢端口状态设定B	<p>00：计数上溢时，TMR6_<t>_PWMB端口输出设定为低电平</p> <p>01：计数上溢时，TMR6_<t>_PWMB端口输出设定为高电平</p> <p>10：计数上溢时，TMR6_<t>_PWMB端口输出保持先前状态</p> <p>11：计数上溢时，TMR6_<t>_PWMB端口输出设定为反转电平</p>	RW
b3~b2	STPCB[1:0]	计数停止端口状态设定B	<p>00：计数停止时，TMR6_<t>_PWMB端口输出设定为低电平</p> <p>01：计数停止时，TMR6_<t>_PWMB端口输出设定为高电平</p> <p>10：计数停止时，TMR6_<t>_PWMB端口输出保持先前状态</p> <p>11：计数停止时，TMR6_<t>_PWMB端口输出保持先前状态</p>	RW
b1~b0	STBCB[1:0]	计数开始端口状态设定B	<p>00：计数开始时，TMR6_<t>_PWMB端口输出设定为低电平</p> <p>01：计数开始时，TMR6_<t>_PWMB端口输出设定为高电平</p> <p>10：计数开始时，TMR6_<t>_PWMB端口输出保持先前状态</p> <p>11：计数开始时，TMR6_<t>_PWMB端口输出保持先前状态</p>	RW

18.6.13 滤波控制寄存器 (TMR6_FCNGR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16		
Res																	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		
Res								NOFI CKGB[1:0]		NOFI ENGB		Res		NOFI CKGA[1:0]		NOFI ENGA	

位/位域	标记	位名	功能	读写
b31~b7	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b4	NOFIENGB	捕获输入端口滤波GB	0: 该单元TMR6_<t>_PWMB输入端口滤波功能无效 1: 该单元TMR6_<t>_PWMB输入端口滤波功能使能	RW
b3	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b0	NOFIENGA	捕获输入端口滤波GA	0: 该单元TMR6_<t>_PWMA输入端口滤波功能无效 1: 该单元TMR6_<t>_PWMA输入端口滤波功能使能	RW

18.6.14 有效周期寄存器 (TMR6_VPERR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res											PCNTS[2:0]		PCNTE[1:0]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						SP PERIB	SP PERIA	Res							

位/位域	标记	位名	功能	读写
b31~b21	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
b15~b10	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
b7~b0	Res	保留位	读出时为“0”，写入时写“0”	RW

18.6.15 状态标志寄存器 (TMR6_STFLR)

复位值: 0x8000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DIRF		Res			CMBF2	CMAF2	Res		VPERNUM[2:0]			Res			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			CMSBDF	CMSBUF	CMSADF	CMSAUF	DTEF	UDFF	OVFF	CMFF	CMEF	CMDF	CMCF	CMBF	CMAF

位/位域	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	Res	保留位	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TMR6_<t>_PWMB捕获完成动作 1: 计数匹配B标志CMBF为1后再次发生TMR6_<t>_PWMB捕获完成动作	RW
b26	CMAF2	再次捕获A	0: 计数匹配A标志CMAF为1后未发生TMR6_<t>_PWMA捕获完成动作 1: 计数匹配A标志CMAF为1后再次发生TMR6_<t>_PWMA捕获完成动作	RW
b25-b24	Res	保留位	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	Res	保留位	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	RW
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	RW
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	RW
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	RW
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	RW
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	RW
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	RW
b4	CMEF	计数匹配E	0: GCMEF寄存器的值与计数值不相等 1: GCMEF寄存器的值与计数值相等	RW
b3	CMDF	计数匹配D	0: GCMDF寄存器的值与计数值不相等 1: GCMDF寄存器的值与计数值相等	RW
b2	CMCF	计数匹配C	0: GCMCF寄存器的值与计数值不相等 1: GCMCF寄存器的值与计数值相等	RW

b1	CMBF	计数匹配B	0：GCMBR寄存器的值与计数值不相等，且未发生TMR6_<t>_PWMB捕获完成动作 1：GCMBR寄存器的值与计数值相等，或发生TMR6_<t>_PWMB捕获完成动作	RW
b0	CMAF	计数匹配A	0：GCMAR寄存器的值与计数值不相等，且未发生TMR6_<t>_PWMA捕获完成动作 1：GCMAR寄存器的值与计数值相等，或发生TMR6_<t>_PWMA捕获完成动作	RW

18.6.16 硬件启动事件选择寄存器 (TMR6_HSTAR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								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
Res				HSTA 11	HSTA 10	HSTA 9	HSTA 8	STAS	Res			HSTA 3	HSTA 2	HSTA 1	HSTA 0

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”，写入时写“0”	RW
b23	HSTA23	硬件启动条件23	条件: TMR6_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	RW
b22	HSTA22	硬件启动条件22	条件: TMR6_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	RW
b21	HSTA21	硬件启动条件21	条件: TMR6_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	RW
b20	HSTA20	硬件启动条件20	条件: TMR6_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	RW
b19	HSTA19	硬件启动条件19	条件: TMR6_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	RW
b18	HSTA18	硬件启动条件18	条件: TMR6_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	RW
b17	HSTA17	硬件启动条件17	条件: TMR6_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	RW
b16	HSTA16	硬件启动条件16	条件: TMR6_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	RW
b15~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW

			1: 条件匹配时, 硬件启动有效	
			0: 硬件启动无效	
b7	STAS	硬件启动使能	1: 硬件启动有效 注: 硬件启动有效时, SSTAR的设置无效	RW
b6~b4	Res	保留位	读出时为“0”, 写入时写“0”	RW
			条件: TMR6_<t>_PWMB端口上采样到下降沿	
b3	HSTA3	硬件启动条件3	0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	RW
			条件: TMR6_<t>_PWMB端口上采样到上升沿	
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	RW
			条件: TMR6_<t>_PWMA端口上采样到下降沿	
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	RW
			条件: TMR6_<t>_PWMA端口上采样到上升沿	
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	RW

18.6.17 硬件停止事件选择寄存器 (TMR6_HSTPR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								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
Res				HSTP 11	HSTP 10	HSTP 9	HSTP 8	STPS	Res			HSTP 3	HSTP 2	HSTP 1	HSTP 0

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”，写入时写“0”	RW
b23	HSTP23	硬件停止条件23	条件: TMR6_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	RW
b22	HSTP22	硬件停止条件22	条件: TMR6_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	RW
b21	HSTP21	硬件停止条件21	条件: TMR6_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	RW
b20	HSTP20	硬件停止条件20	条件: TMR6_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	RW
b19	HSTP19	硬件停止条件19	条件: TMR6_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	RW
b18	HSTP18	硬件停止条件18	条件: TMR6_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	RW
b17	HSTP17	硬件停止条件17	条件: TMR6_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	RW
b16	HSTP16	硬件停止条件16	条件: TMR6_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	RW
b15~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW

			1: 条件匹配时, 硬件停止有效	
b7	STPS	硬件停止使能	0: 硬件停止无效 1: 硬件停止有效 注: 硬件停止有效时, SSTPR 的设置无效	RW
b6~b4	Res	保留位	读出时为“0”, 写入时写“0”	RW
b3	HSTP3	硬件停止条件3	条件: TMR6_<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	RW
b2	HSTP2	硬件停止条件2	条件: TMR6_<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	RW
b1	HSTP1	硬件停止条件1	条件: TMR6_<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	RW
b0	HSTP0	硬件停止条件0	条件: TMR6_<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	RW

18.6.18 硬件清零事件选择寄存器 (TMR6_HCLRR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								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
Res				HCLE 11	HCLE 10	HCLE 9	HCLE 8	CLES	Res			HCLE 3	HCLE 2	HCLE 1	HCLE 0

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”，写入时写“0”	RW
b23	HCLE23	硬件清零条件23	条件: TMR6_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	RW
b22	HCLE22	硬件清零条件22	条件: TMR6_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	RW
b21	HCLE21	硬件清零条件21	条件: TMR6_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	RW
b20	HCLE20	硬件清零条件20	条件: TMR6_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	RW
b19	HCLE19	硬件清零条件19	条件: TMR6_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	RW
b18	HCLE18	硬件清零条件18	条件: TMR6_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	RW
b17	HCLE17	硬件清零条件17	条件: TMR6_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	RW
b16	HCLE16	硬件清零条件16	条件: TMR6_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	RW
b15~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW

			1: 条件匹配时, 硬件清零有效	
			0: 硬件清零无效	
b7	CLES	硬件清零使能	1: 硬件清零有效 注: 硬件清零有效时, SCLRR的设置无效	RW
b6~b4	Res	保留位	读出时为“0”, 写入时写“0”	RW
			条件: TMR6_<t>_PWMB端口上采样到下降沿	
b3	HCLE3	硬件清零条件3	0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	RW
			条件: TMR6_<t>_PWMB端口上采样到上升沿	
b2	HCLE2	硬件清零条件2	0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	RW
			条件: TMR6_<t>_PWMA端口上采样到下降沿	
b1	HCLE1	硬件清零条件1	0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	RW
			条件: TMR6_<t>_PWMA端口上采样到上升沿	
b0	HCLE0	硬件清零条件0	0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	RW

18.6.19 硬件刷新事件选择寄存器 (TMR6_HUPDR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								HUP D23	HUP D22	HUP D21	HUP D20	HUPD 19	HUPD 18	HUPD 17	HUPD 16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HUP D11	HUP D10	HUPD 9	HUPD 8	UPD S	Res			HUPD 3	HUPD 2	HUPD 1	HUPD 0

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”，写入时写“0”	RW
b23	HUPD23	硬件刷新条件23	条件: TMR6_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	RW
b22	HUPD22	硬件刷新条件22	条件: TMR6_TRIGD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	RW
b21	HUPD21	硬件刷新条件21	条件: TMR6_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	RW
b20	HUPD20	硬件刷新条件20	条件: TMR6_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	RW
b19	HUPD19	硬件刷新条件19	条件: TMR6_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	RW
b18	HUPD18	硬件刷新条件18	条件: TMR6_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	RW
b17	HUPD17	硬件刷新条件17	条件: TMR6_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	RW
b16	HUPD16	硬件刷新条件16	条件: TMR6_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	RW
b15~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW

			1: 条件匹配时, 硬件刷新有效	
b7	UPDS	硬件刷新使能	0: 硬件刷新无效 1: 硬件刷新有效 注: 硬件刷新有效时, SUPDR的设置无效	RW
b6~b4	Res	保留位	读出时为“0”, 写入时写“0”	RW
b3	HUPD3	硬件刷新条件3	条件: TMR6_<t>_PWMB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	RW
b2	HUPD2	硬件刷新条件2	条件: TMR6_<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	RW
b1	HUPD1	硬件刷新条件1	条件: TMR6_<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	RW
b0	HUPD0	硬件刷新条件0	条件: TMR6_<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	RW

18.6.20 硬件捕获事件选择寄存器 A (TMR6_HCPAR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res						HCPA 25	HCPA 24	HCPA 19	HCPA 18	HCPA 17	HCPA 16	HCPA 19	HCPA 18	HCPA 17	HCPA 16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HCPA 11	HCPA 10	HCPA 9	HCPA 8	Res				HCPA 3	HCPA 2	HCPA 1	HCPA 0

位/位域	标记	位名	功能	读写
b31~b26	Res	保留位	读出时为“0”，写入时写“0”	RW
b25	HCPA25	硬件捕获A条件25	条件: TMR6_<t>_XOR信号采样到下降沿 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	RW
b24	HCPA24	硬件捕获A条件24	条件: TMR6_<t>_XOR信号采样到上升沿 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	RW
b23	HCPA23	硬件捕获A条件23	条件: TMR6_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	RW
b22	HCPA22	硬件捕获A条件22	条件: TMR6_TRIGD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	RW
b21	HCPA21	硬件捕获A条件21	条件: TMR6_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	RW
b20	HCPA20	硬件捕获A条件20	条件: TMR6_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	RW
b19	HCPA19	硬件捕获A条件19	条件: TMR6_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	RW
b18	HCPA18	硬件捕获A条件18	条件: TMR6_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	RW
b17	HCPA17	硬件捕获A条件17	条件: TMR6_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	RW
b16	HCPA16	硬件捕获A条件16	条件: TMR6_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	RW
b15~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW

			1: 条件匹配时, 硬件捕获A有效	
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	RW
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 1: 条件匹配时, 硬件捕获A有效	RW
b7~b4	Res	保留位	读出时为“0”, 写入时写“0”	RW
b3	HCPA3	硬件捕获A条件3	条件: TMR6_<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	RW
b2	HCPA2	硬件捕获A条件2	条件: TMR6_<t>_PWMB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	RW
b1	HCPA1	硬件捕获A条件1	条件: TMR6_<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	RW
b0	HCPA0	硬件捕获A条件0	条件: TMR6_<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	RW

注意:

TMR6_<t>_XOR 信号为各个单元的正交输入 AIN 与 BIN 异或后得到的信号。

18.6.21 硬件捕获事件选择寄存器 B (TMR6_HCPBR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res						HCPB 25	HCPB 24	HCP B23	HCP B22	HCP B21	HCP B20	HCPB 19	HCPB 18	HCPB 17	HCPB 16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HCP B11	HCP B10	HCPB 9	HCPB 8	Res				HCPB 3	HCPB 2	HCPB 1	HCPB 0

位/位域	标记	位名	功能	读写
b31~b26	Res	保留位	读出时为“0”，写入时写“0”	RW
b25	HCPB25	硬件捕获B条件25	条件: TMR6_<t>_XOR信号采样到下降沿 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	RW
b24	HCPB24	硬件捕获B条件24	条件: TMR6_<t>_XOR信号采样到上升沿 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	RW
b23	HCPB23	硬件捕获B条件23	条件: TMR6_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	RW
b22	HCPB22	硬件捕获B条件22	条件: TMR6_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	RW
b21	HCPB21	硬件捕获B条件21	条件: TMR6_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	RW
b20	HCPB20	硬件捕获B条件20	条件: TMR6_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	RW
b19	HCPB19	硬件捕获B条件19	条件: TMR6_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	RW
b18	HCPB18	硬件捕获B条件18	条件: TMR6_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	RW
b17	HCPB17	硬件捕获B条件17	条件: TMR6_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	RW
b16	HCPB16	硬件捕获B条件16	条件: TMR6_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	RW
b15~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW

			1: 条件匹配时, 硬件捕获B有效	
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	RW
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 1: 条件匹配时, 硬件捕获B有效	RW
b7~b4	Res	保留位	读出时为“0”, 写入时写“0”	RW
b3	HCPB3	硬件捕获B条件3	条件: TMR6_<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	RW
b2	HCPB2	硬件捕获B条件2	条件: TMR6_<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	RW
b1	HCPB1	硬件捕获B条件1	条件: TMR6_<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	RW
b0	HCPB0	硬件捕获B条件0	条件: TMR6_<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	RW

注意:

TMR6_<t>_XOR 信号为各个单元的正交输入 AIN 与 BIN 异或后得到的信号。

18.6.22 硬件递加事件选择寄存器 (TMR6_HCUPR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								HCU P23	HCU P22	HCU P21	HCU P20	HCU P19	HCU P18	HCU P17	HCU P16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HC UP11	HC UP10	HC UP9	HC UP8	HC UP7	HC UP6	HC UP5	HC UP4	HCUP 3	HCUP 2	HCUP 1	HCU P0

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”，写入时写“0”	RW
b23	HCUP23	硬件递加条件23	条件: TMR6_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	RW
b22	HCUP22	硬件递加条件22	条件: TMR6_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	RW
b21	HCUP21	硬件递加条件21	条件: TMR6_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	RW
b20	HCUP20	硬件递加条件20	条件: TMR6_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	RW
b19	HCUP19	硬件递加条件19	条件: TMR6_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	RW
b18	HCUP18	硬件递加条件18	条件: TMR6_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	RW
b17	HCUP17	硬件递加条件17	条件: TMR6_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	RW
b16	HCUP16	硬件递加条件16	条件: TMR6_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	RW
b15~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW

			1: 条件匹配时, 硬件递加有效	
b7	HCUP7	硬件递加条件7	条件: TMR6_<t>_PWMB 端口为高电平时, TMR6_<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	RW
b6	HCUP6	硬件递加条件6	条件: TMR6_<t>_PWMB 端口为高电平时, TMR6_<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	RW
b5	HCUP5	硬件递加条件5	条件: TMR6_<t>_PWMB 端口为低电平时, TMR6_<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	RW
b4	HCUP4	硬件递加条件4	条件: TMR6_<t>_PWMB 端口为低电平时, TMR6_<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	RW
b3	HCUP3	硬件递加条件3	条件: TMR6_<t>_PWMA 端口为高电平时, TMR6_<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	RW
b2	HCUP2	硬件递加条件2	条件: TMR6_<t>_PWMA 端口为高电平时, TMR6_<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	RW
b1	HCUP1	硬件递加条件1	条件: TMR6_<t>_PWMA 端口为低电平时, TMR6_<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	RW
b0	HCUP0	硬件递加条件0	条件: TMR6_<t>_PWMA 端口为低电平时, TMR6_<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	RW

18.6.23 硬件递减事件选择寄存器 (TMR6_HCDOR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								HCD O23	HCD O22	HCD O21	HCD O20	HCD O19	HCD O18	HCD O17	HCD O16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				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/位域	标记	位名	功能	读写
b31~b20	Res	保留位	读出时为“0”，写入时写“0”	RW
b23	HCDO23	硬件递减条件23	条件: TMR6_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	
b22	HCDO22	硬件递减条件22	条件: TMR6_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	
b21	HCDO21	硬件递减条件21	条件: TMR6_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	
b20	HCDO20	硬件递减条件20	条件: TMR6_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	
b19	HCDO19	硬件递减条件19	条件: TMR6_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	
b18	HCDO18	硬件递减条件18	条件: TMR6_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	
b17	HCDO17	硬件递减条件17	条件: TMR6_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	
b16	HCDO16	硬件递减条件16	条件: TMR6_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	
b15~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW

			1: 条件匹配时, 硬件递减有效	
b7	HCD07	硬件递减条件7	条件: TMR6_<t>_PWMB 端口为高电平时, TMR6_<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	RW
b6	HCD06	硬件递减条件6	条件: TMR6_<t>_PWMB 端口为高电平时, TMR6_<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	RW
b5	HCD05	硬件递减条件5	条件: TMR6_<t>_PWMB 端口为低电平时, TMR6_<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	RW
b4	HCD04	硬件递减条件4	条件: TMR6_<t>_PWMB 端口为低电平时, TMR6_<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	RW
b3	HCD03	硬件递减条件3	条件: TMR6_<t>_PWMA 端口为高电平时, TMR6_<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	RW
b2	HCD02	硬件递减条件2	条件: TMR6_<t>_PWMA 端口为高电平时, TMR6_<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	RW
b1	HCD01	硬件递减条件1	条件: TMR6_<t>_PWMA 端口为低电平时, TMR6_<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	RW
b0	HCD00	硬件递减条件0	条件: TMR6_<t>_PWMA 端口为低电平时, TMR6_<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	RW

18.6.24 公共滤波控制寄存器 (TMR6_FCNTN)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	NOFICKTD [1:0]	NOFI ENTD	Res	NOFICKTC [1:0]	NOFI ENTC	Res	NOFICKTB [1:0]	NOFI ENTB	Res	NOFICKTA [1:0]	NOFI ENTA				

位/位域	标记	位名	功能	读写
b31~b15	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b12	NOFIENTD	捕获输入端口滤波TD	0: TMR6_TRIGD端口输入滤波功能无效 1: TMR6_TRIGD端口输入滤波功能使能	RW
b11	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b8	NOFIENTC	捕获输入端口滤波TC	0: TMR6_TRIGC端口输入滤波功能无效 1: TMR6_TRIGC端口输入滤波功能使能	RW
b7	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b4	NOFIENTB	捕获输入端口滤波TB	0: TMR6_TRIGB端口输入滤波功能无效 1: TMR6_TRIGB端口输入滤波功能使能	RW
b3	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b0	NOFIENTA	捕获输入端口滤波TA	0: TMR6_TRIGA端口输入滤波功能无效 1: TMR6_TRIGA端口输入滤波功能使能	RW

18.6.25 公共软件同步启动控制寄存器 (TMR6_SSTAR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												SSTA4	SSTA3	SSTA2	SSTA1

位/位域	标记	位名	功能	读写
b31~b4	Res	保留位	读出时为“0”，写入时写“0”	RW
b3	SSTA4	单元4软件启动	0: 软件启动无效 1: 软件启动使能	RW
b2	SSTA3	单元3软件启动	0: 软件启动无效 1: 软件启动使能	RW
b1	SSTA2	单元2软件启动	0: 软件启动无效 1: 软件启动使能	RW
b0	SSTA1	单元1软件启动	0: 软件启动无效 1: 软件启动使能	RW

18.6.26 公共软件同步停止控制寄存器 (TMR6_SSTPR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												SSTP4	SSTP3	SSTP2	SSTP1

位/位域	标记	位名	功能	读写
b31~b4	Res	保留位	读出时为“0”，写入时写“0”	RW
b3	SSTP4	单元4软件停止	0: 软件停止无效 1: 软件停止使能	RW
b2	SSTP3	单元3软件停止	0: 软件停止无效 1: 软件停止使能	RW
b1	SSTP2	单元2软件停止	0: 软件停止无效 1: 软件停止使能	RW
b0	SSTP1	单元1软件停止	0: 软件停止无效 1: 软件停止使能	RW

18.6.27 公共软件同步清零控制寄存器 (TMR6_SCLRR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												SCLE4	SCLE3	SCLE2	SCLE1

位/位域	标记	位名	功能	读写
b31~b4	Res	保留位	读出时为“0”，写入时写“0”	RW
b3	SCLE4	单元4软件清零	0: 软件清零无效 1: 软件清零使能	RW
b2	SCLE3	单元3软件清零	0: 软件清零无效 1: 软件清零使能	RW
b1	SCLE2	单元2软件清零	0: 软件清零无效 1: 软件清零使能	RW
b0	SCLE1	单元1软件清零	0: 软件清零无效 1: 软件清零使能	RW

18.6.28 公共软件同步刷新控制寄存器 (TMR6_SUPDR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												SUPD4	SUPD3	SUPD2	SUPD1

位/位域	标记	位名	功能	读写
b31~b4	Res	保留位	读出时为“0”，写入时写“0”	RW
b3	SUPD4	单元4软件刷新	0: 软件刷新无效 1: 软件刷新使能	RW
b2	SUPD3	单元3软件刷新	0: 软件刷新无效 1: 软件刷新使能	RW
b1	SUPD2	单元2软件刷新	0: 软件刷新无效 1: 软件刷新使能	RW
b0	SUPD1	单元1软件刷新	0: 软件刷新无效 1: 软件刷新使能	RW

18.7 注意事项

- 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。
- GCMCR~GCMFR 在对应的缓存功能有效时，其中断输出、事件输出无效。
- 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。
- 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 18-7 所示。

表 18-7 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	

- 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（TMR6_<t>_PWMA 和 TMR6_<t>_PWMB）的输出变化控制优先级分别如下表 18-8 所示、表 18-9 所示。

表 18-8 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.EMBCA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	↓
	CNTER=0	PCNAR.UDFCA设定状态	↓

PWMA端口输出控制事件	端口状态	优先级
计数停止时	PCNAR.STPCA设定状态	低

表 18-9 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	↓
	CNTER=GCMBR	↓
	CNTER=GCMAR	↓
锯齿波向下计数时	CNTER=0	↓
	CNTER=GCMBR	↓
	CNTER=GCMAR	↓
三角波向上计数时	CNTER=GCMBR	↓
	CNTER=GCMAR	↓
	CNTER=PERAR	↓
三角波向下计数时	CNTER=GCMBR	↓
	CNTER=GCMAR	↓
	CNTER=0	↓
计数停止时	PCNBR.STPCB设定状态	低

19 高精度 PWM (HRPWM)

19.1 概述

高精度 PWM (HRPWM) 是一个 22 位定时器，最多能够产生 12 个高分辨率 PWM 波形。该定时器提供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、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、清零、捕获等）；支持同步输出以同步片外计数器；各基准值寄存器支持缓存功能；支持多种空闲控制：立即空闲、间隔输出和延迟空闲；外部事件支持消隐和窗口模式；支持 EMB 控制；支持同步 DAC；支持多相单元间交错移相控制。本系列产品 HRPWM 搭载 6 个计数单元，最多支持 6 组高精度 PWM 输出。

HRPWM 主要特性：

- 高分辨率定时器
 - 130ps 分辨率，支持周期校准
 - 所有输出支持高分辨率，支持占空比，周期以及相位调节等
 - 6 个 22bit 计数单元（每个单元独立计数器，支持 6 个通用比较匹配点和 2 个专用比较匹配点）
 - 支持锯齿波（向上）和三角波计数模式
 - 每个单元 1 对高精度 PWM 输出，支持本单元 4 个通用比较匹配事件、2 个专用匹配事件以及外部事件置位和复位 PWM
 - 支持 2 个捕获单元
 - 各基准值寄存器支持缓存功能
- 支持 10 个外部事件，可被任意单元使用
 - 支持电平和边沿敏感
 - 5 个外部事件支持快速异步模式
 - 5 个外部事件支持数字滤波
 - 单元内部可对外部事件进行消隐和窗口滤波
- 支持 EMB 功能
 - 每个单元有 6 个 EMB group 可进行选择
 - 释放支持立即释放以及逐周期释放
- 支持同步输出
 - 支持同步输出信号同步片外计数器
- 支持多种同步方式
 - 软件同步启动、清零、刷新和停止
 - 硬件同步启动、清零，硬件同步源可配置为片外和片上的事件
 - 单元 1 相位匹配事件同步清零和启动单元 2~6

- 灵活的 PWM 输出
 - 支持死区功能（分辨率为 130ps）
 - 支持输出极性选择
 - 支持 A/B 通道交换输出功能
- 多种空闲处理
 - 立即空闲
 - 间隔输出
 - 延迟空闲
- 支持中断
 - 每个单元 6 个通用比较匹配中断
 - 每个单元 2 个专用比较匹配中断
 - 每个单元 1 个周期匹配中断
 - 每个单元 1 个零点匹配中断
 - 每个单元 2 个捕获中断
 - 1 个校准完成中断
 - 1 个间隔输出周期匹配中断
 - 1 个延迟空闲触发中断
- 支持事件输出到 AOS 单元
 - 每个单元 6 个通用比较匹配事件
 - 每个单元 2 个专用比较匹配事件
 - 每个单元 1 个周期匹配事件
 - 每个单元 1 个零点匹配事件
 - 每个单元 2 个捕获事件
 - 1 个外部事件检出事件
- 支持 DAC 触发
 - 3 个 DAC 触发信号，可同步 DAC

19.2 基本框图

HRPWM 基本的功能及特性如表 19-1 所示。

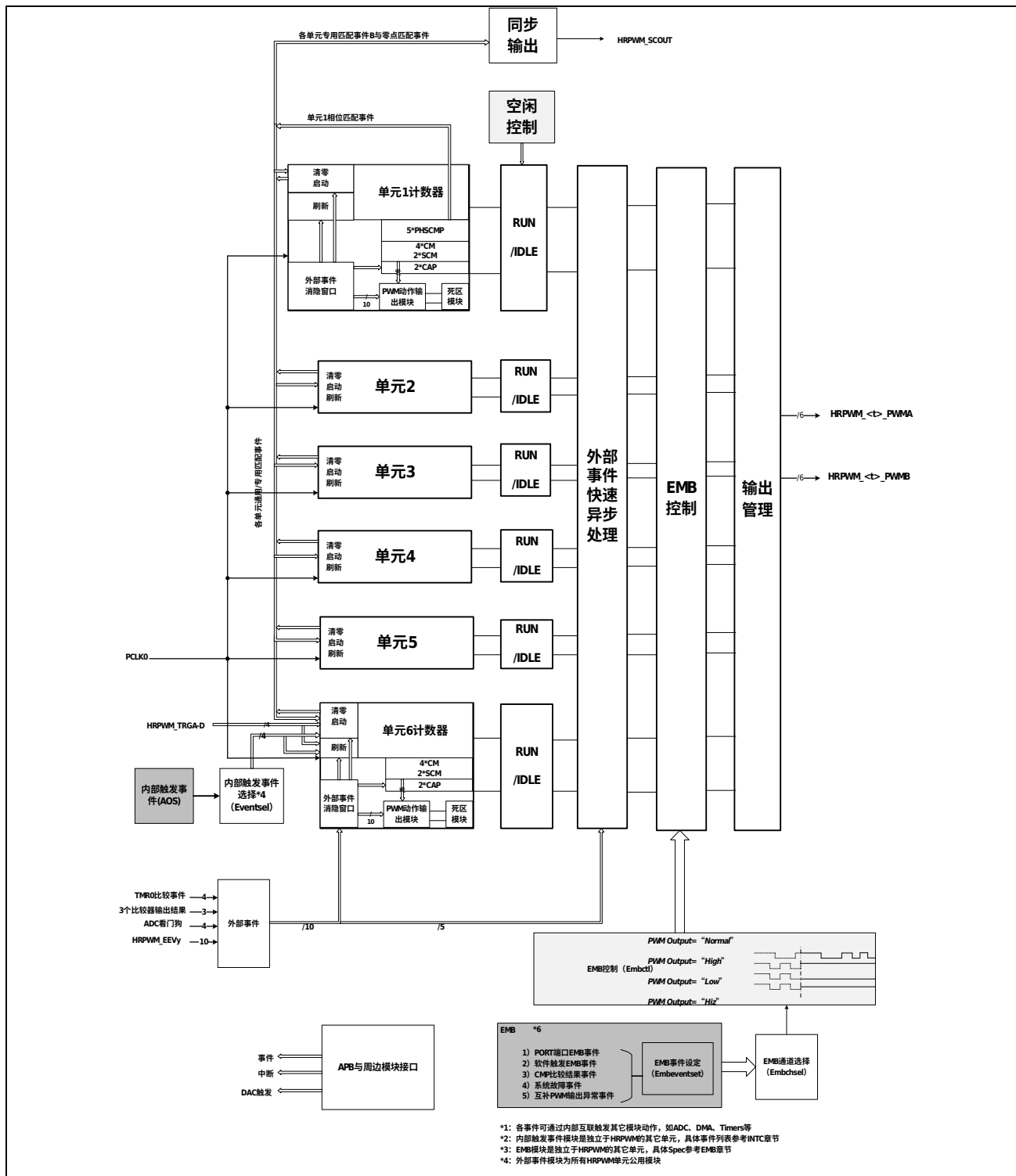
表 19-1 HRPWM 的基本功能及特性

波形模式	锯齿波（递加）、三角波（递加递减计数）
基本功能	捕获功能
	软件同步
	硬件同步
	同步输出
	缓存功能
	高分辨率通用PWM输出（提供130ps的分辨率）
	EMB控制
	间隔输出控制器
	支持延迟空闲
	外部事件支持消隐和窗口模式
中断输出	计数比较匹配中断
	计数周期匹配中断
	计数零点匹配中断
	捕获中断
	校准完成中断
	间隔输出周期匹配中断
	延迟空闲触发中断
事件输出	计数比较匹配事件
	外部事件检出事件
	计数周期匹配事件
	计数零点匹配事件
	捕获事件
DAC触发信号	DAC同步触发信号

19.3 功能说明

19.3.1 HRPWM 框图

HRPWM 的基本框如图 19-1 所示。框图中所示“<t>”表示单元编号，即“<t>”为 1~6，本章节后文提到“<t>”时均指单元编号，y 为外部事件编号，即 y 为 1~10。



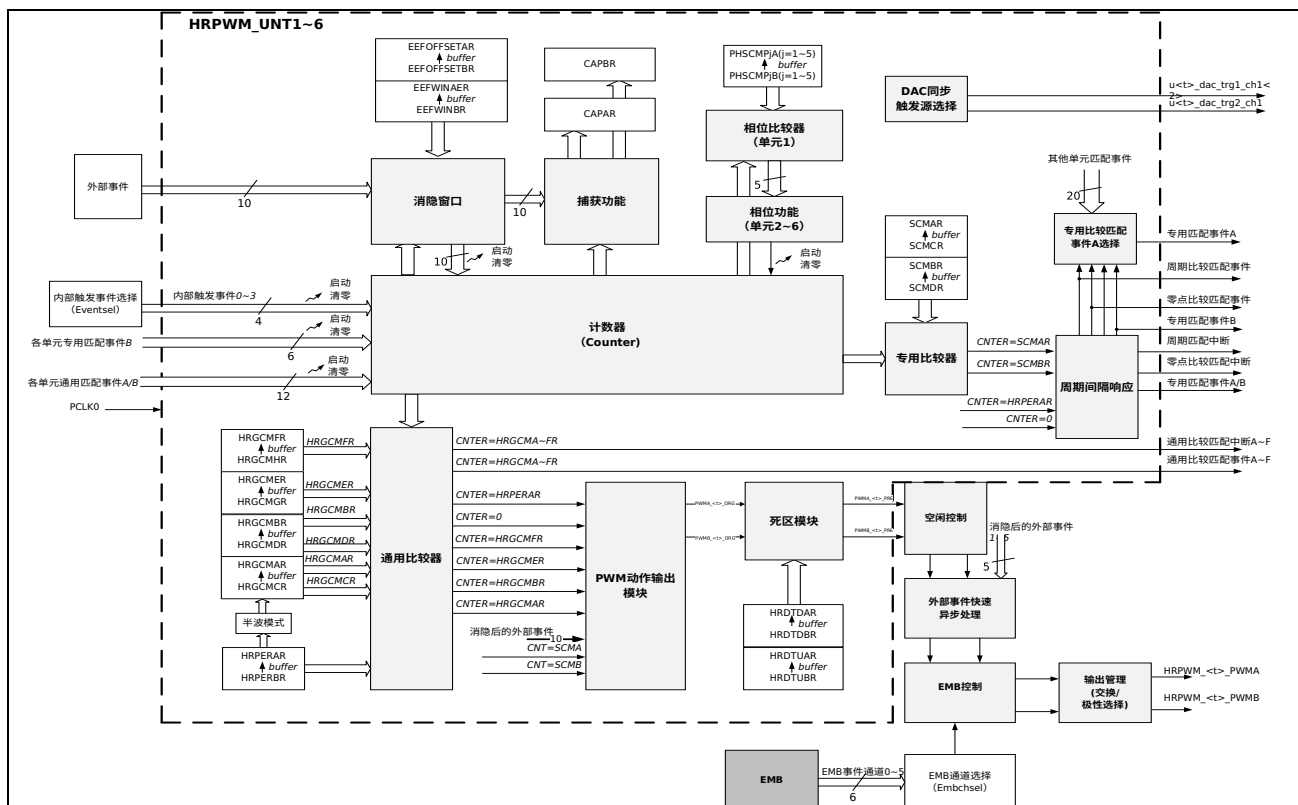


图 19-2 HRPWM 各单元基本框图

表 19-2 所示为 HRPWM 的输入输出端口列表。

表 19-2 HRPWM 端口列表

端口名	方向	功能
HRPWM_<t>_PWMA	out	PWM输出端口
HRPWM_<t>_PWMB		
HRPWM_TRIGA	in	硬件启动、清零条件同步输入端口
HRPWM_TRIGB		
HRPWM_TRIGC		
HRPWM_TRIGD		
HRPWM_EEvy (y=1~10)	in	外部事件输入端口
HRPWM_SCOUT	out	同步输出端口

19.3.2 波形模式

HRPWM 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 18-2、图 19-4 所示。

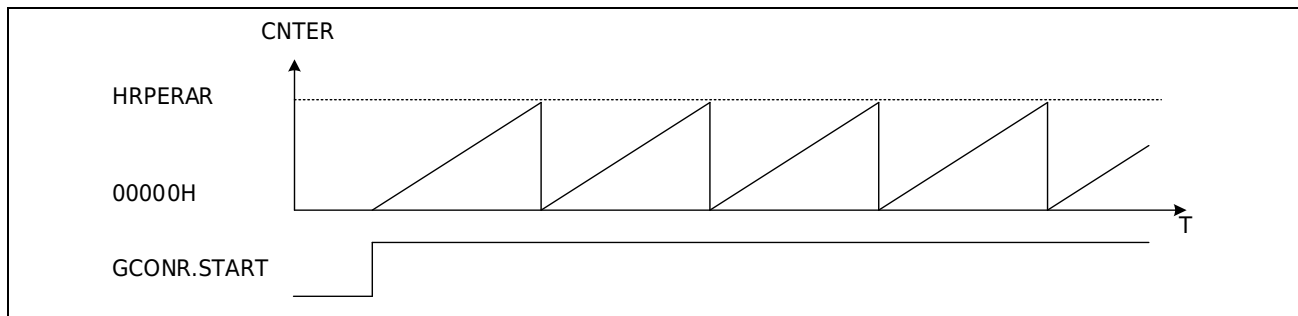


图 19-3 锯齿波波形

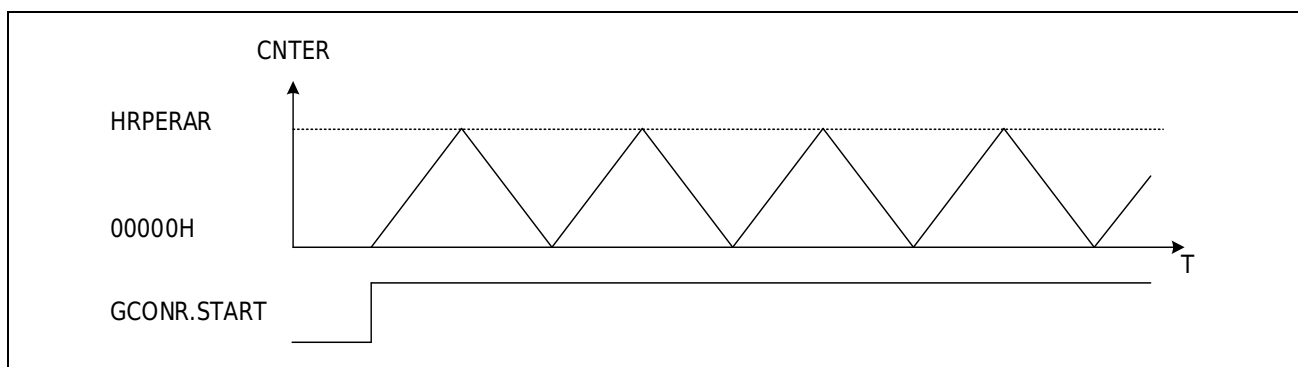


图 19-4 三角波波形

注：

在锯齿波模式下，当计数器的值大于周期值时，计数器将变为 0，并重新开始计数；在三角波模式下当计数值大于周期值时，计数器将跳变到周期值，并开始向下计数。

19.3.3 计数器时钟

HRPWM 校准模块通过将 PCLK0 均匀地分为 64 份来提高 PWM 波形的分辨率，即它可以为 PWM 波形提供 $t_{PCLK0}/64$ 的分辨率。HRPWM 的分辨率以及 PWM 波形的最小频率如表 19-3 所示。

表 19-3 HRPWM 分辨率

波形模式	PCLK0 频率	分辨率	HRPERAR	PWM 最小频率
锯齿波	120MHz	130ps	0x3FFFC0	1.83kHz
三角波			0x3FFFC0	0.916kHz

当高精度控制寄存器 HRPWM<t>_CR 的 EN 位为 1 时，高分辨率有效，可支持高精度周期，高精度占空比、高精度死区以及高精度相位的调整。

该模式下，以下功能不支持高分辨率：

- 计数器值的写入和读取
- 硬件同步清零事件
- 捕获功能
- 硬件刷新
- 外部事件消隐窗口
- 专用比较基准值

当高精度控制寄存器 HRPWM<t>_CR.EN 为 0 时，高分辨率无效，在该模式下，高分辨率周期基准值寄存器、高精度通用比较基准值寄存器、高分辨率死区基准值寄存器、高分辨率相位比较基准值寄存器的低六位设定的值无效。

19.3.4 校准功能

HRPWM 在使能 HRPWM (CR.EN=1) 之前，必须先利用校准单元进行校准，否则高分辨率功能无法正常使用。

通过配置校准控制寄存器 HRPWM_CALCR 的 CAL 和 CALEN 位，可以进行单次校准或者周期性校准。周期性校准的校准周期可以通过 HRPWM_CALCR.CALPRD[3:0]寄存器位配置。校准完成后 CALCR 的 CALENF 位被置 1。

本校准单元可对 120M 频率的 PCLK0 进行校准。当频率过低时，会发生校准错误，HRPWM_CALCR.ERRF 会被置 1。

19.3.5 计数方向

19.3.5.1 锯齿波计数方向

在锯齿波模式下，HRPWM 计数方向为递加计数。

19.3.5.2 三角波计数方向

三角波模式时，计数器向上计数计数到周期值-64 或向下计数计数到 64 时，自动改变计数方向。

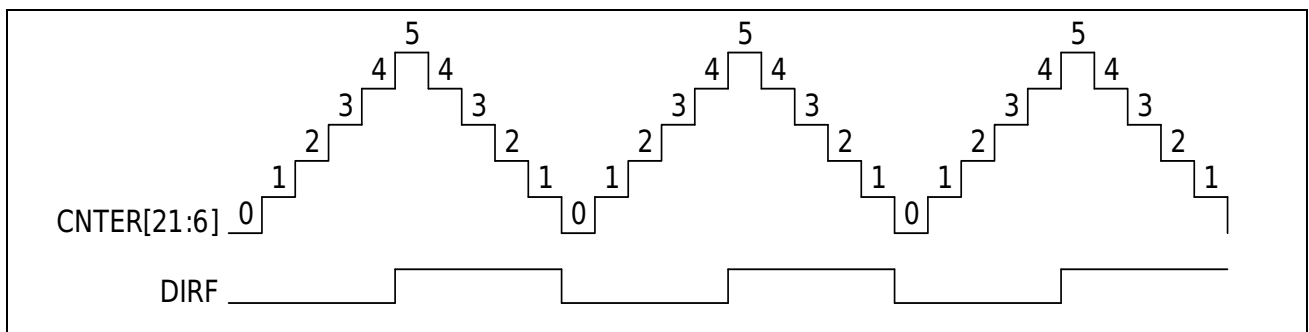


图 19-5 三角波计数方向

19.3.6 捕获功能

每个单元都具有捕获功能，具备 2 组捕获输入寄存器（HRPWM<t>_CAPAR、HRPWM<t>_CAPBR），用于保存捕获到的当前计数值。设定捕获事件选择寄存器（HRPWM<t>_HCPAR2）的 HCPAEN 位为 1，硬件捕获功能变为有效。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（HRPWM<t>_CAPAR、HRPWM<t>_CAPBR）中，其中寄存器中的 CAPDIRA/B 表示捕获发生时计数器的计数方向，为 1 表示捕获发生时计数方向为向下计数，为 0 表示捕获发生时计数方向为向上计数。捕获寄存器的捕获值不支持高精度，低六位读数为 0。同时，当捕获发生时状态标志寄存器（HRPWM<t>_STFLR1）的 CAPAF 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、HRPWM_TRIGA~D 的端口输入、外部事件 1~10 输入，具体的条件选择可通过硬件捕获事件选择寄存器（HRPWM<t>_HCPAR1<2>、HRPWM<t>_HCPBR1<2>）来设定。图 19-6 为捕获输入的动作例。

除上述硬件捕获条件外，通过对软件同步捕获控制寄存器 HRPWM_SCAPR.SCAP<t>A寄存器位写 1，可以触发一次软件捕获。

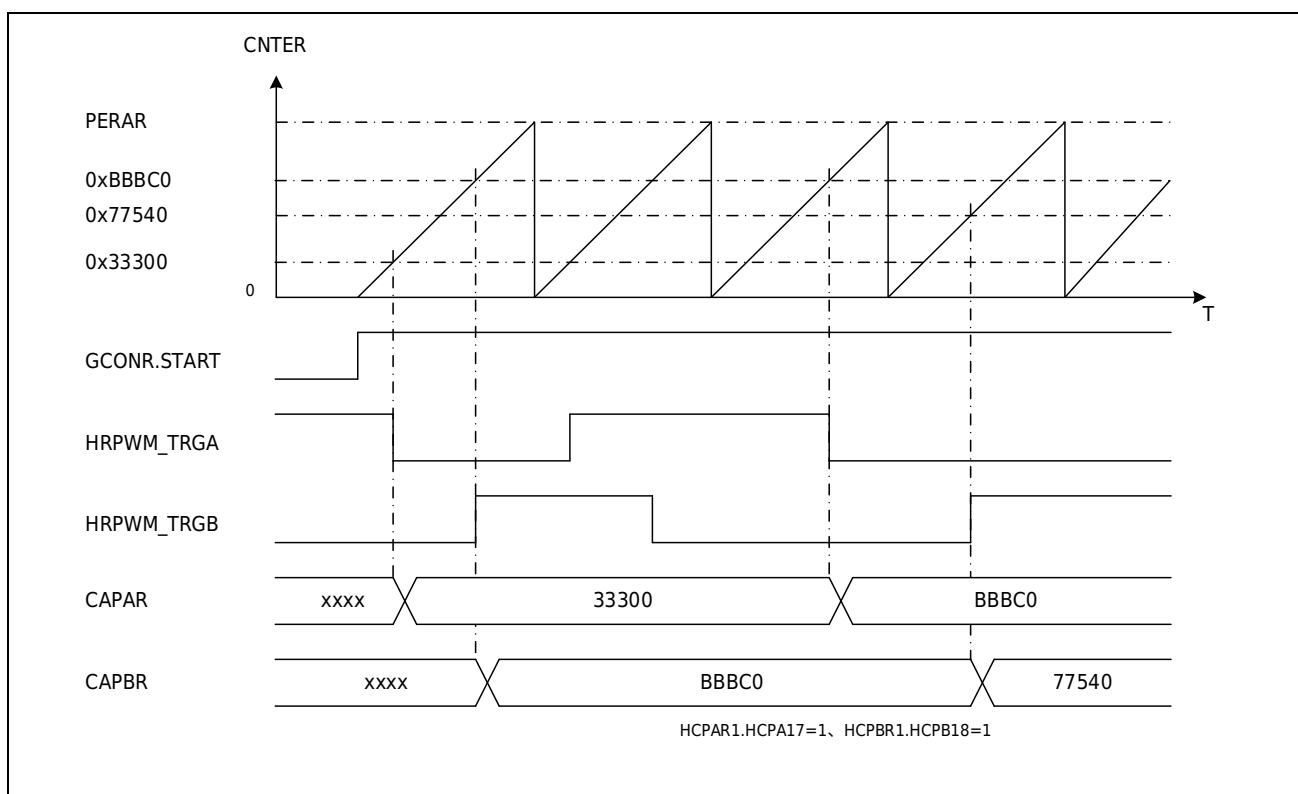


图 19-6 捕获输入动作

19.3.7 PWM 动作输出

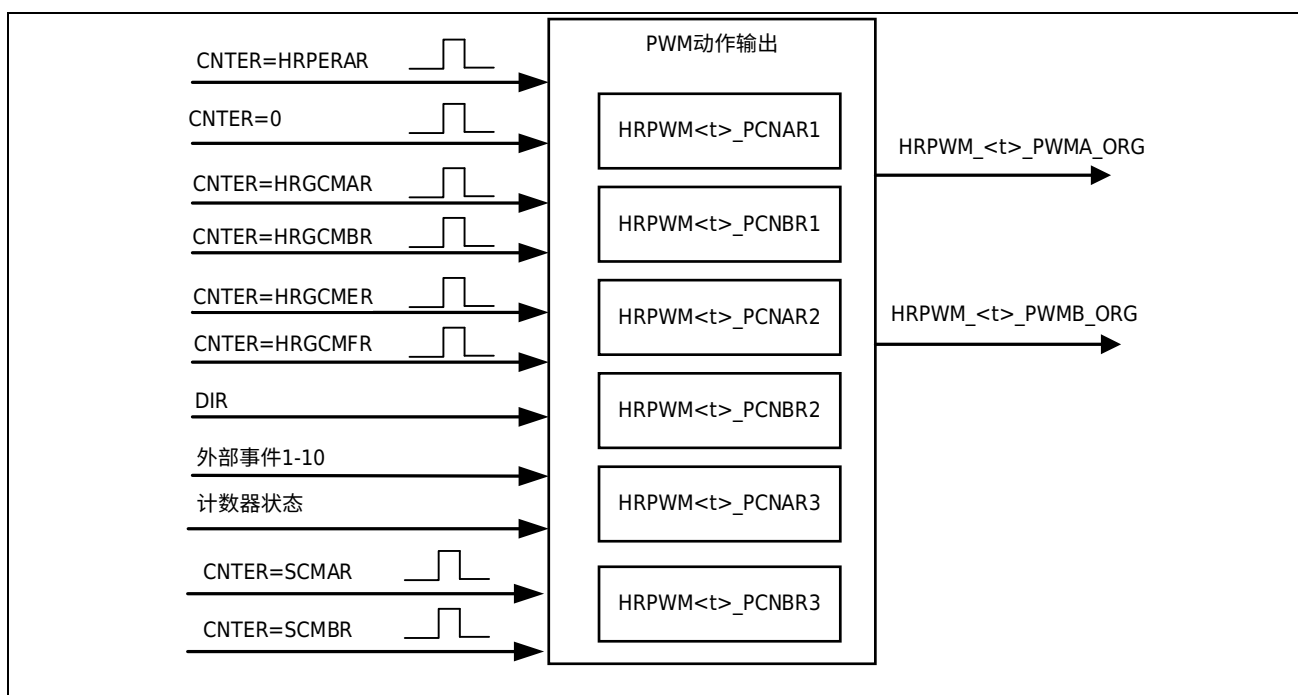


图 19-7 PWM 动作输出模块

注:

上图中 *CNTER* 表示计数器的值，*DIR* 指计数器计数方向

PWM 动作输出模块框图如图 19-8 所示，每个 HRPWM 单元有 2 个输出端口（HRPWM_<t>_PWMA、HRPWM_<t>_PWMB），可以设置在下述事件发生时输出指定电平。

- CNTER = HRGCMAR: 计数器计数值等于高分辨率通用比较基准值 HRGCMAR
- CNTER = HRGCMBR: 计数器计数值等于高分辨率通用比较基准值 HRGCMBR
- CNTER = HRGCMER: 计数器计数值等于高分辨率通用比较基准值 HRGCMER
- CNTER = HRGCMFR: 计数器计数值等于高分辨率通用比较基准值 HRGCMFR
- CNTER = SCMAR: 计数器计数值等于专用比较基准值 SCMAR
- CNTER = SCMBR: 计数器计数值等于专用比较基准值 SCMBR
- CNTER= HRPERAR: 计数器计数值等于周期值
- 硬件清零:硬件同步清零、主单元相位匹配时产生的清零事件，写 SCLRR 对应寄存器位软件清零；间隔输出控制计数器产生的清零不视为硬件清零；硬件清零仅在锯齿波模式下有效。
- CNTER= 0:计数器计数值等于 0
- 外部事件 1~10
- 开始计数
- 停止计数

比较匹配事件以及外部事件可以进一步由计数器的方向（向上或向下）限定，通过端口控制寄存器 HRPWM<t>_PCNAR1、HRPWM<t>_PCNAR2、HRPWM<t>_PCNAR3 进行配置。事件发生时，HRPWM_<t>_PWMA、HRPWM_<t>_PWMB 的状态可配置为置 0，置 1，翻转或者完全旁路，当所有事件同时发生时优先级请参考表 18-9。

注：

芯片复位之后，端口控制寄存器中仅外部事件对 PWM 的控制复位为旁路，其他事件均复位为置 0，如果需要完全旁路某个事件，请将对应的控制寄存器位配置为旁路状态；专用匹配事件不支持高精度，专用比较基准值寄存器（HRPWM_<t>_SCMAR）低 6 位读出永远为 0，当高分辨率使能时（HRPWM<t>_CR.EN=1），CNTER=SCMAR/SCMBR 会存在 1~2 个 PCLK0 的抖动。

图 19-8 为比较输出的动作例。

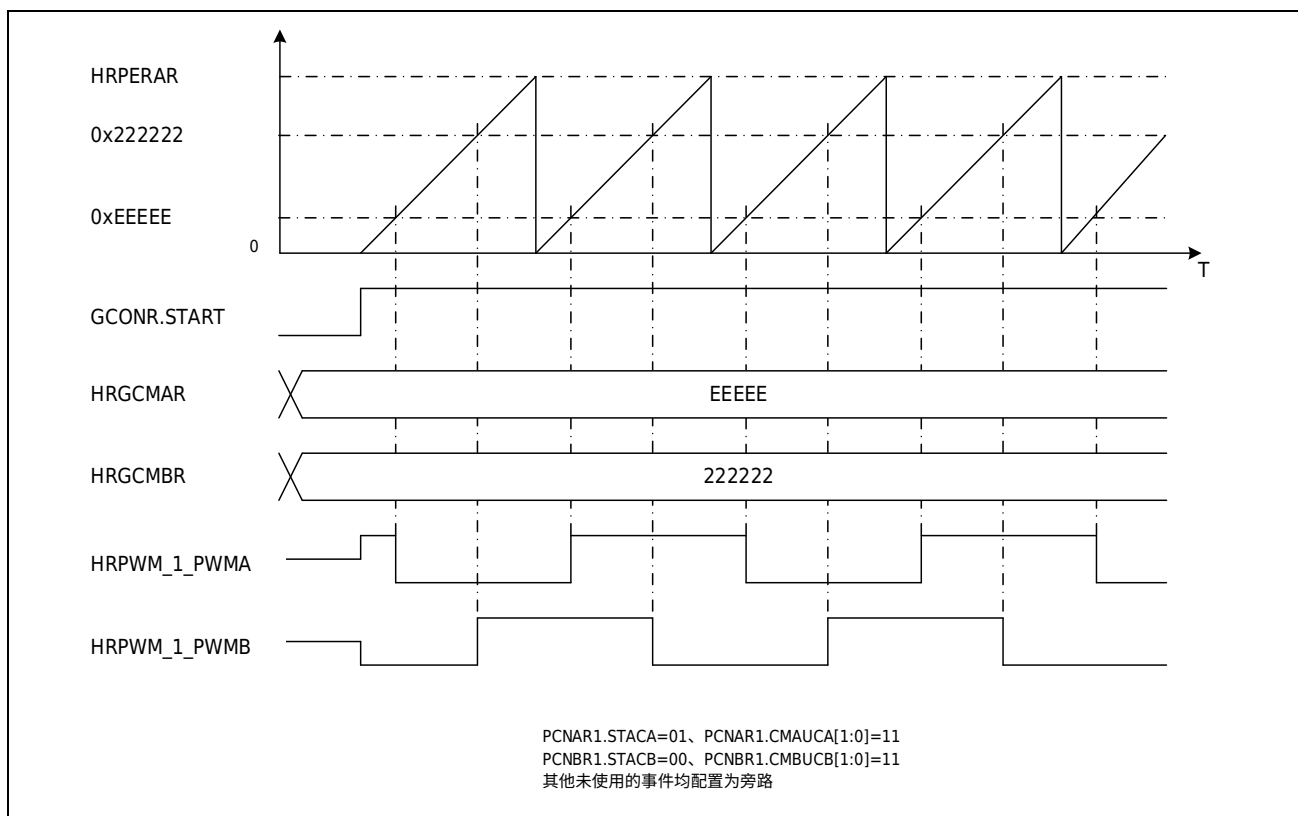


图 19-8 比较输出动作

半波模式

半波模式用于在不同频率下生成占空比 50% 的方波，通过配置通用控制寄存器 1（HRPWM<t>_GCONR1）的 HALF 为 1 使能半波功能。在该模式下比较值 HRGCMAR 的值自动变为（HRPERAR）/2，请勿在计数器启动后变化 HRPWM<t>_GCONR1.HALF 位，半波模式下 PWM 输出图例如图 19-9。

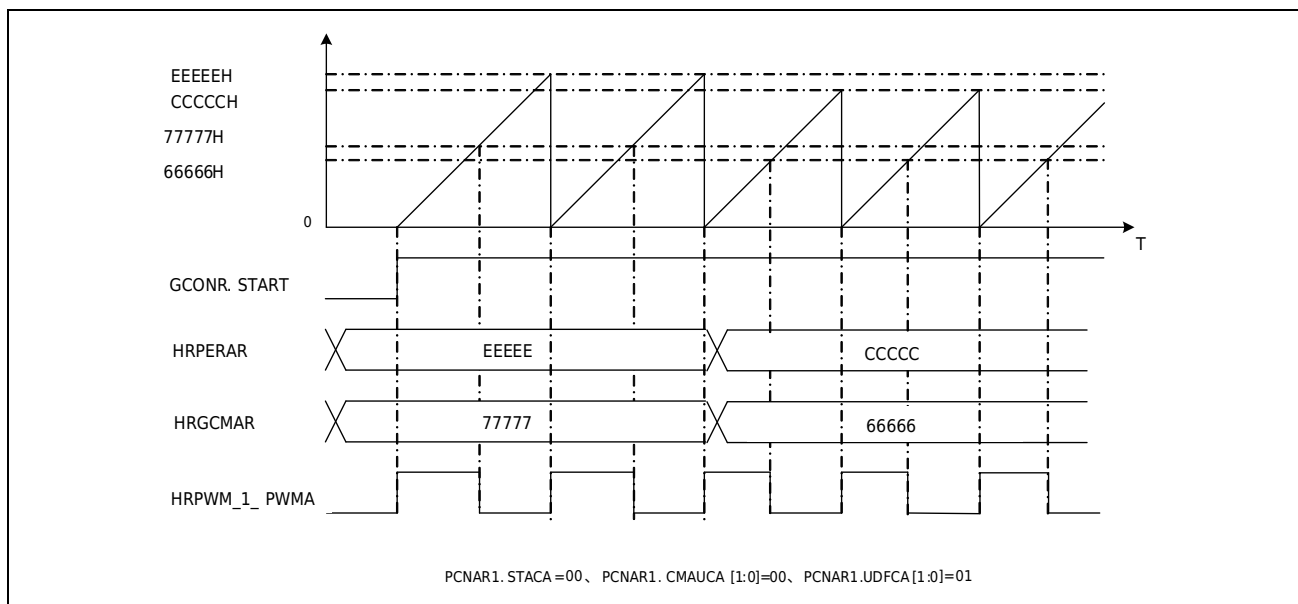


图 19-9 半波模式有效 PWM 输出图例

占空比限制

当高分辨率分辨率功能使能时($HRPWM<t>_CR.EN=1$), HRPWM 的占空比存在一定限制。

在锯齿波模式下, 比较值不得设置为大于周期值加减 1 个 PCLK0 周期的值 (不包括等于周期值), 否则将会出现不可预期的波形, 限制范围如图 19-10 所示。如果要设置比较值大于周期值, 必须将比较值设置为周期值+1 个 PCLK0, 此时才能保证比较匹配不发生。(比较值大于周期值时, 不产生匹配)

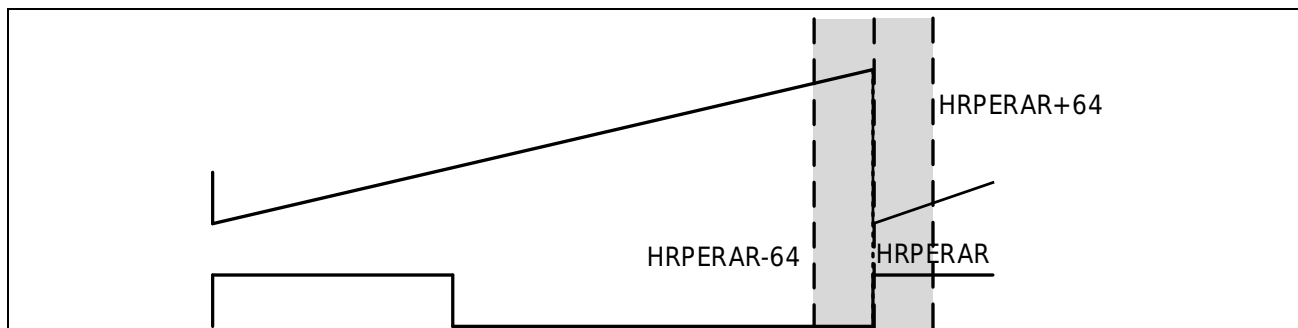


图 19-10 锯齿波占空比限制 1

当利用两个比较匹配事件控制同一个 PWM 通道时, 两个事件的之间的距离必须大于 1 个 PCLK0 周期, 否则将出现不可以预期的波形。

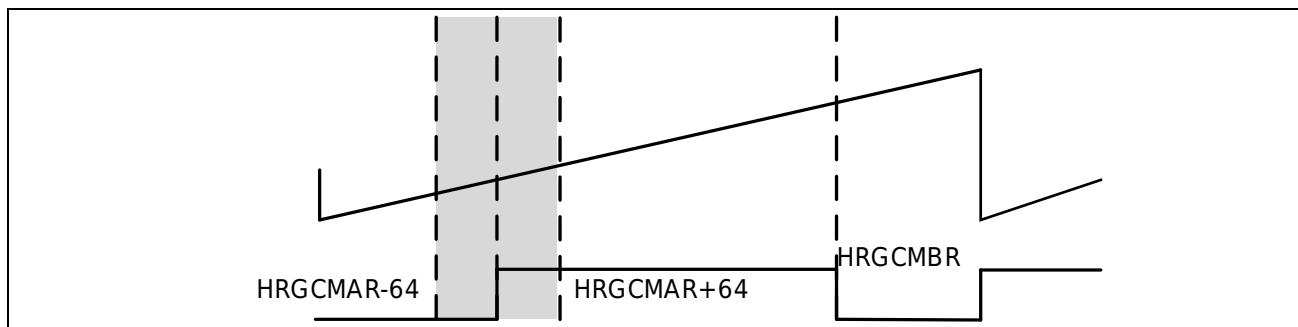


图 19-11 锯齿波占空比限制 2

在三角波模式下，比较值不得设置为周期值加减 2 个 PCLK0 周期的值，以及设置小于 2 个 PCLK0 周期的值，否则将会出现不可预期的波形，限制范围如图 19-10 所示。

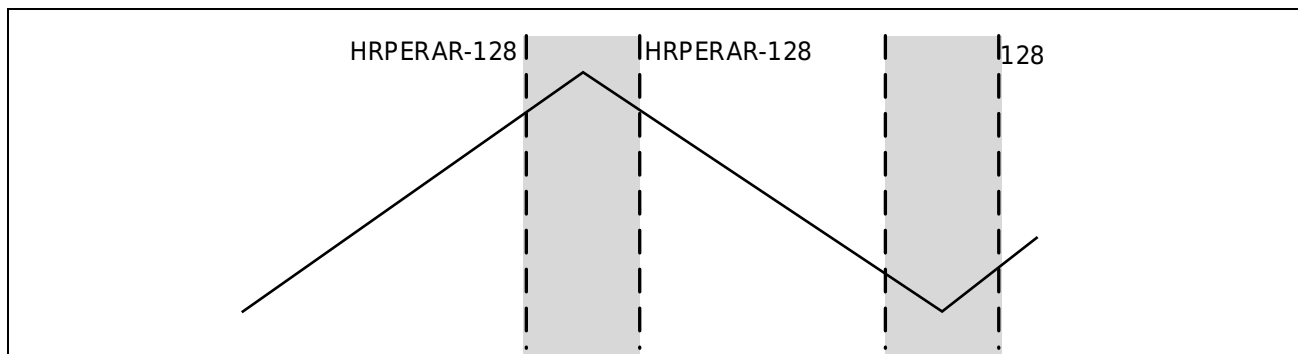


图 19-12 三角波占空比限制 1

与锯齿波相同的是两个比较匹配事件控制同一个 PWM 通道时，两个事件之间的距离必须大于 1 个 PCLK0 周期，否则将出现不可以预期的波形。

除上述比较值设定的限制,HRPWM 对 PWM 输出脉宽具有进一步限制,当脉冲宽度小于等于 1 个 PCLK0 周期时,窄脉宽被滤除,当脉冲宽度大于 1 个 PCLK0 小于 3 个 PCLK0 周期时,PWM 输出存在 1~2 个 PCLK0 周期的抖动。

19.3.8 死区模块

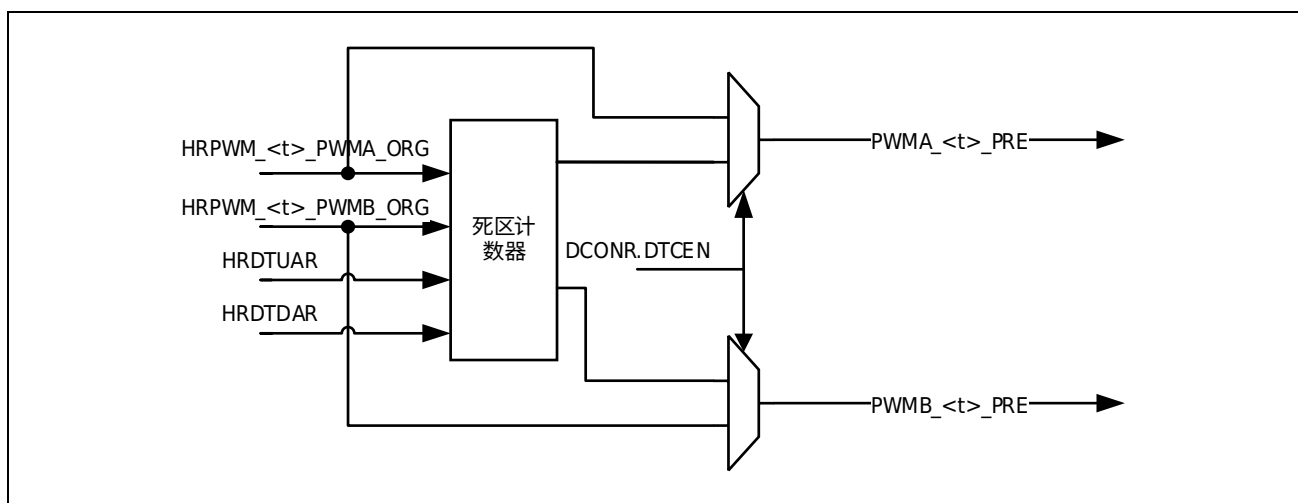


图 19-13 死区模块

死区模块如图 19-13 所示，在死区使能（HRPWM<t>_DCONR.DTCEN=1）时，设定好端口在计数启动、比较值等于比较值、计数值等于 0 和计数值等于周期值时的电平状态，由 PWM 动作输出模块产生通道 A 内部输出信号（HRPWM_<t>_PWMA_ORG），并和高精度死区时间基准值寄存器（HRPWM_DTU<D>AR）的设定值通过时序偏移，以硬件的方式实现在 PWMA_<t>_PRE 和 PWMB_<t>_PRE 端口上输出一对互补 PWM 波形。

在该模式下 PWMA_<t>_PRE 输出端口上的极性与 HRPWM_<t>_PWMA_ORG 相同，PWMB_<t>_PRE 端口上的极性与 HRPWM_<t>_PWMA_ORG 上的极性相反。

如果检测到 HRPWM_<t>_PWMA_ORG 上升沿, PWMB_<t>_PRE 输出变为低电平, 死区计数器加载 HRPWM<t>_HRDTUAR 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 PWMA_<t>_PRE 输出高电平; 如果检测到 HRPWM_<t>_PWMA_ORG 下降沿, PWMA_<t>_PRE 输出变为低电平, 死区计数器加载 HRPWM<t>_HRDTDAR 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 PWMB_<t>_PRE 输出高电平。

通过设置高精度死区时间基准值寄存器 HRPWM<t>_HRDTUAR 和 HRPWM<t>_HRDTDAR, 输出上升和下降变化的死区时间可以相应设定, 图 19-14 为硬件实现互补 PWM 输出波形。

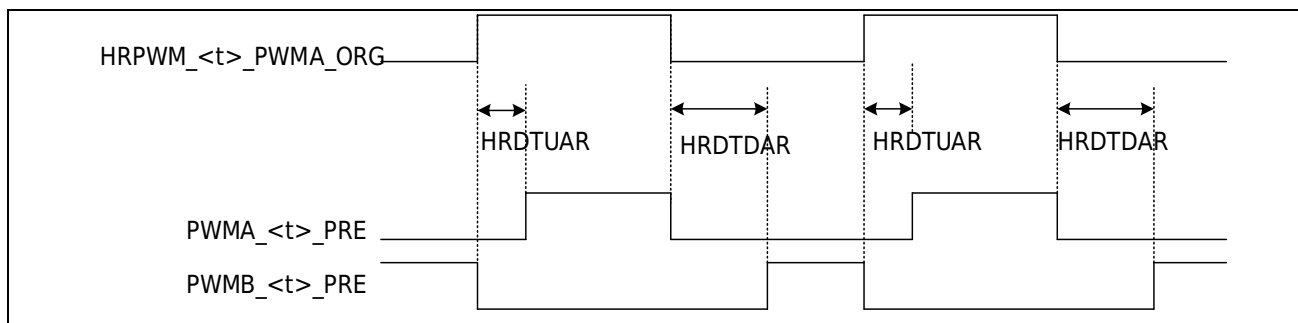


图 19-14 硬件实现互补 PWM 波输出

HRPWM_<t>_PWMA_ORG 的高电平脉冲宽度小于 HRPWM<t>_HRDTUAR 设定的死区时间时, 仅 PWMB_<t>_PRE 输出变为低电平。PWMB_<t>_PRE 输出电平变为高的条件是当 HRPWM_<t>_PWMA_ORG 下降沿之后经过 HRPWM<t>_HRDTDAR 寄存器设定的死区时间之后。在此情况下, PWMA_<t>_PRE 输出将持续保持为低电平。

HRPWM_<t>_PWMA_ORG 的低电平脉冲宽度小于 HRPWM<t>_HRDTDAR 设定的死区时间时, 仅 PWMA_<t>_PRE 输出变为低电平。PWMA_<t>_PRE 输出电平变为高的条件是当 HRPWM_<t>_PWMA_ORG 上升沿之后经过 HRPWM<t>_HRDTUAR 寄存器设定的死区时间之后。在此情况下, PWMB_<t>_PRE 输出将持续保持为低电平。如下图 19-15 所示。

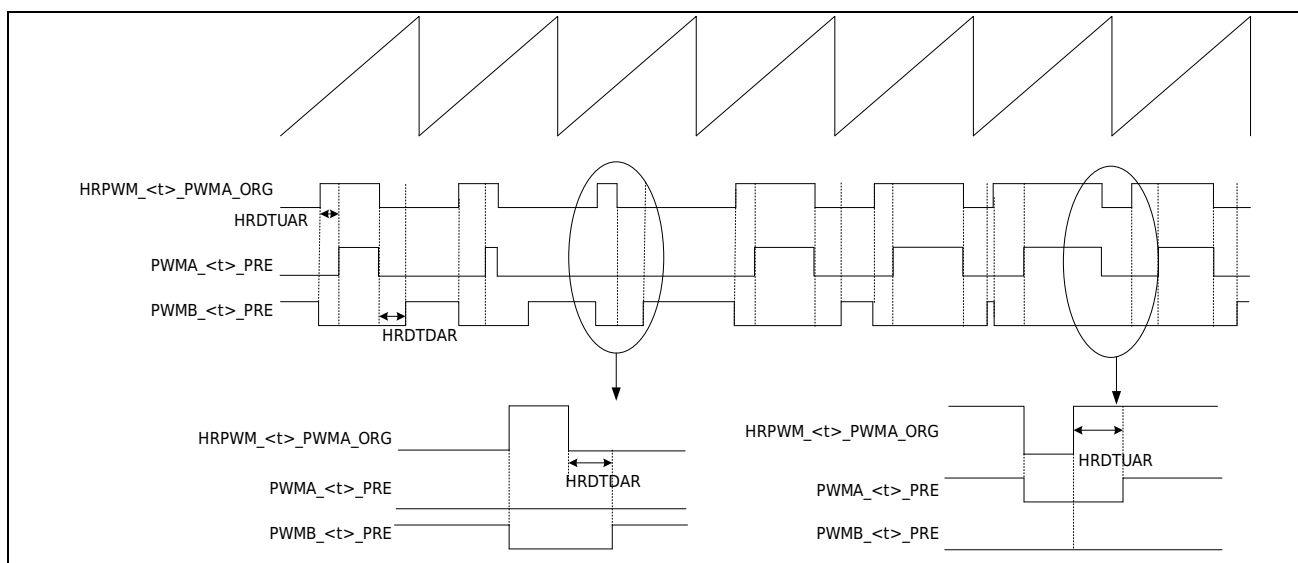


图 19-15 脉宽异常时的死区模式下的波形输出

19.3.9 空闲控制

空闲控制级有三种空闲状态方式：

- 立即空闲
- 延迟空闲
- 间隔输出空闲

输出在空闲状态下的电平可以通过配置 HRPWM<t>_IDLECR 的 IDLESA位选择为高电平或者低电平。

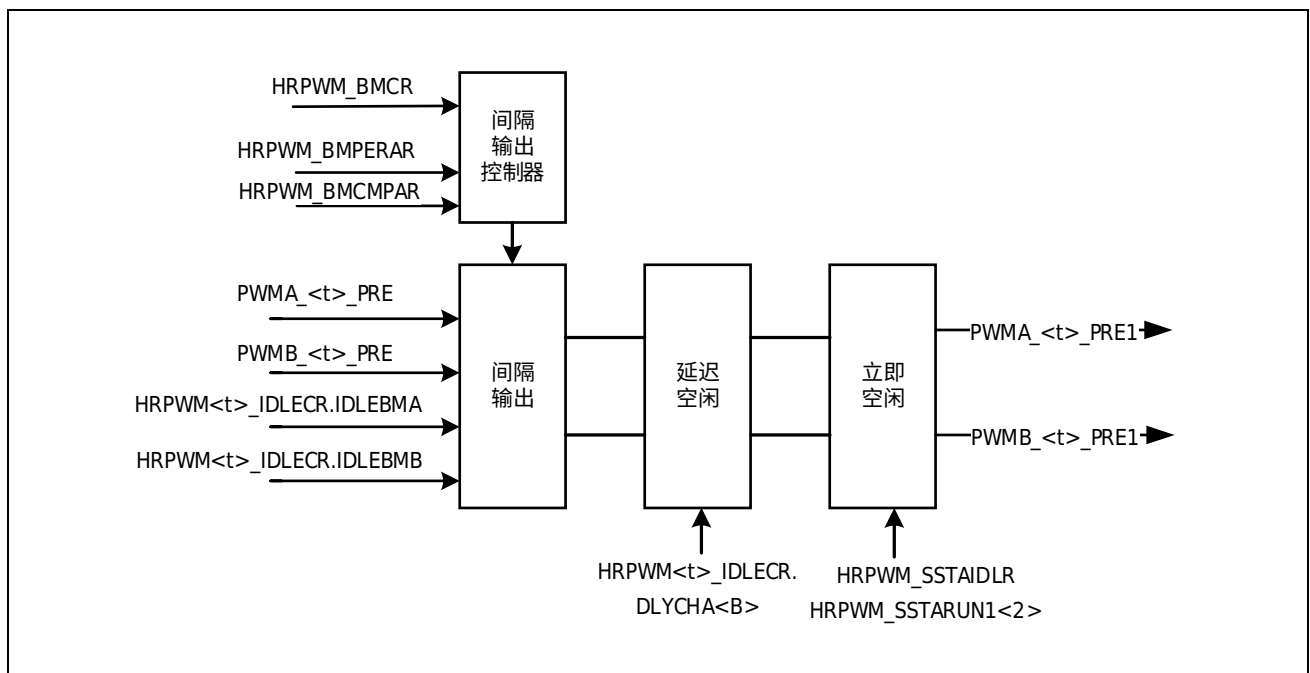


图 19-16 空闲控制基本框图

19.3.9.1 立即空闲

通过对 HRPWM_SSTAILDR 寄存器的 SSTAIDL<t>A位写 1，可以触发对应通道进入立刻空闲，对 HRPWM_SSTARUNR1 寄存器的 SSTARUN<t>A位写 1 后退出立即空闲进入运行状态。各个单元的 PWM 状态可以通过 HRPWM_SSTARUNR1.SSTARUN<t>A 位查询，当 HRPWM_SSTARUNR1.SSTARUN<t>A为 1 时，表示当前通道 PWM 处于运行状态，为 0 时处于空闲状态。该位在间隔输出模式期间可以表示当前通道 PWM 处于运行还是空闲状态。芯片复位之后，HRPWM_SSTARUNR1.SSTARUN<t>A默认为 0，即初始输出空闲状态。

退出立即空闲还可以对 HRPWM_SSTARUNR2.SSTARUN<t>A写 1 触发计数器启动时退出空闲，该触发信号仅在对单元计数器停止时有效，触发信号发生后 PWM 将在其后遇到的第一个计数器启动信号发生时进入运行状态，参考图 19-17。如果在对 HRPWM_SSTARUNR2.SSTARUN<t>A写 1 到计数器启动之间，发生触发立即空闲（对 HRPWM_SSTAILDR 寄存器的 SSTAIDL<t>A位写 1），在计数器启动时退出空闲的动作将被取消。

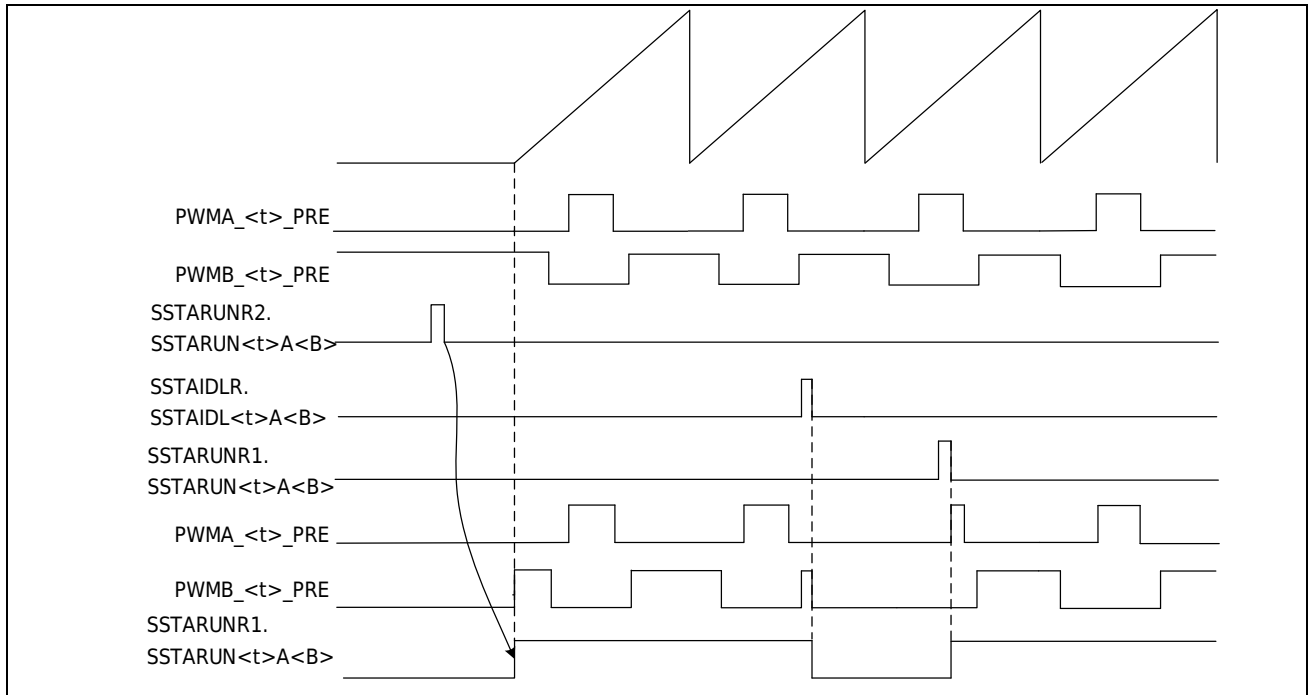


图 19-17 立即空闲进入和退出

19.3.9.2 延迟空闲

该模式用来在关闭 PWM 时延迟 PWM 进入空闲状态，在该模式下，会在完整周期点（HRPWM<t>_GCONR1.PRDSEL 定义）检测到设置的空闲状态与内部 PWM（PWMA_<t>_PRE）输出状态一致时进入预设的空闲状态，从而保证进入空闲状态波形的完整性。通过配置 HRPWM<t>_IDLECR.DLYPRTEN 位使能延迟空闲功能。

延迟空闲模式支持事件触发和软件触发，通过 HRPWM<t>_IDLECR.DLYEVSEL[2:0]寄存器位选择触发源：

1. EEV6、EEV7、EEV8 和 EEV9（各自单元滤波）
2. 软件触发

软件触发可以通过对每个单元的 HRPWM<t>_IDLECR.DLYSTRG 写 1 触发或者对 HRPWM_SSTADIDLR.DLYSTRGU<t>写 1 触发。

输出 A 和输出 B 具有单独的延迟空闲使能位，可以在延迟空闲触发源到来时，选择进入或者不进入延迟空闲状态。

一旦触发延迟空闲，当 HRPWM_<t>_PWMA 进入空闲状态后，HRPWM<t>_STFLR2 中的 DLYIDLEA 状态标志位被置 1，空闲模式将永久保持，直到对 HRPWM_SSTARUNR1.SSTARUN<t>A写 1 触发退出延迟空闲，在延迟空闲状态下计数器继续运行。延迟空闲的进入不会影响 HRPWM_SSTARUNR1.SSTARUN<t>A的状态。在退出延迟空闲时，需要先确认 HRPWM<t>_DLYIDLEA位为 1，再对 HRPWM_SSTARUNR1.SSTARUN<t>A写 1 退出延迟空闲，写 1 后不会立刻退出空闲状态，而是在完整周期点（由 HRPWM_<t>_GCONR1.PRDSEL 位定

义)后检测到设置的空闲状态与内部 PWM (PWMA_<t>_PRE) 输出状态一致后退出空闲状态后退出延迟空闲, 如图 19-18。

在进入延迟空闲模式时, 当 HRPWM<t>_IDLECR 的 INTENDLYPRT 位设置为 1, 一旦外部事件到达或者软件触发发生, 就会产生延迟空闲触发中断, HRPWM<t>_STFLR2 寄存器的 DLYPRT 标志位被置为 1。可以通过 HRPWM<t>_STFLR2 寄存器的 OASTAT 和 OBSTAT 位确认外部事件触发延迟空闲时 PWM 输出的状态, 当寄存器位为 1 时表示延迟空闲触发时 PWM 处于高电平状态, 为 0 时表示 PWM 处于低电平状态。

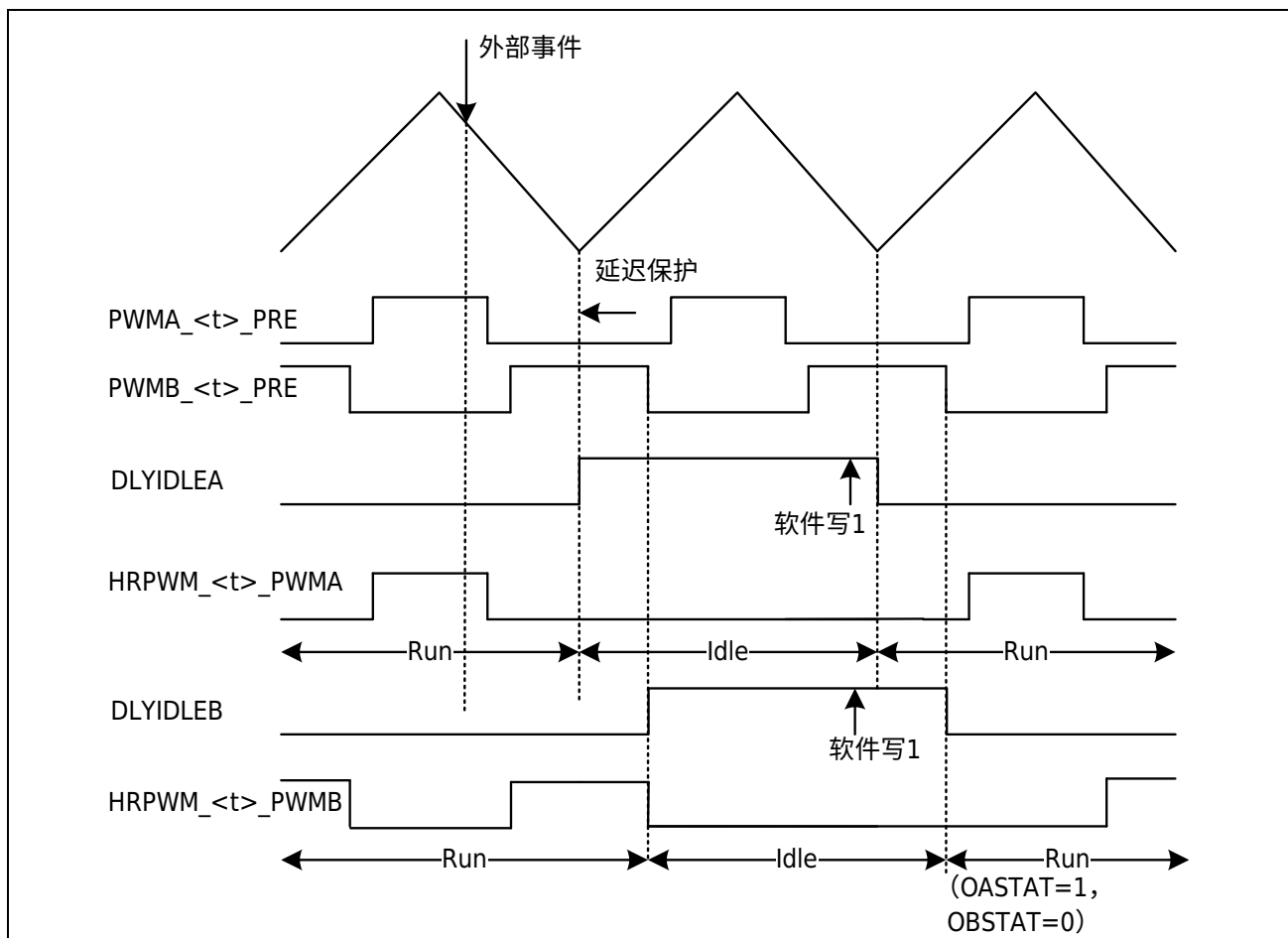


图 19-18 延迟空闲的进入和退出

延迟空闲模式的优先级高于间隔输出: 当延迟空闲被触发, PWM 进入空闲状态, 如果间隔输出仍在运行中, 即使间隔输出从空闲到运行也无法将 PWM 从空闲退出, PWM 将一直处于空闲状态直到退出延迟空闲。当退出延迟空闲, 间隔输出仍然在运行中, 间隔输出将恢复正常, PWM 根据间隔输出控制器进行运行空闲动作间隔输出

间隔输出控制器允许通过硬件控制 HRPWM<t>_PWMA交替输出空闲和运行状态。该模式由 HRPWM_BMCR 寄存器中的 BMEN 位使能。

在间隔输出空闲状态下，HRPWM_<t>_PWMA的端口状态可以设置为不受影响或者输出空闲状态的电平(配置 HRPWM<t>_IDLECR.IDLEBMA位)。

间隔输出控制器包括：

- 1 个计数器 (BM-counter)
- 1 个比较寄存器：HRPWM_BCMAR，用以定义空闲状态的持续时间
- 1 个周期寄存器：HRPWM_BMPERAR，用以定义空闲状态和运行状态的总和

注：

当 HRPWM_BMCR.BMMD 设置为 1 时，当 HRPWM_BCMAR 设置为 0 时空闲状态的持续时间为 0 个 BM-counter 的计数时钟周期；当 HRPWM_BMCR.BMMD 设置为 0 时，当 HRPWM_BCMAR 设置为 0 时空闲状态的持续时间为 1 个 BM-counter 的计数时钟周期。

BM-counter 是一个 16 位递加计数器，它支持连续模式和单次模式。当 HRPWM_BMCR.BMCTN=1 时，BM 计数器工作在连续模式，间隔输出过程一直持续到 HRPWM_BMCR.BMOPTF 位为 0。当 HRPWM_BMCR.BMCTN=0 时，BM 计数器工作在单次模式，当计数器达到周期值时停止计数。当 BM-counter 达到周期值时，HRPWM_BMCR 中的 BMOVFF 被置位。

在间隔输出工作期间（运行、空闲）通过设置 HRPWM_BMCR.BMTMR1~6 位，可以停止和复位每个单元的计数器，计数器仅在间隔输出空闲状态下停止计数和清零，在运行状态下正常运行。当间隔输出控制寄存器中的 BMTMR1~6=0 时，对应单元内的计数器不受影响，正常计数；当间隔输出控制寄存器中的 BMTMR1~6=1 时，对应单元内的计数器在间隔输出空闲状态停止，并且发生清零。

注：

请勿在 PWM 使能了间隔输出延迟模式时，将对应单元 BMTMR<t>设置为 1。

BM-counter 的计数时钟可选择多个时钟源，通过设置 HRPWM_BMCR.BMCLKS[3:0]进行选择。主要包括三种类型的时钟源：单元 1~6 的完整周期点、TMR0 的比较事件或者 PCLK0 的分频。当计数时钟选择为 PCLK0 的分频时，分频系数由 HRPWM_BMCR.BMPSC[3:0]位设置。间隔输出空闲状态的持续时间由 HRPWM_BMPCMAR 定义，间隔输出的周期由 HRPWM_BMPERAR 定义。

通过配置 HRPWM_BMCR.BMMD 位可以设置间隔输出的动作模式，当 HRPWM_BMCR.BMMD 为 0 时，间隔输出工作在模式 1，间隔输出的空闲时间的持续时间为 HRPWM_BMPCMAR+1 个间隔输出计数时钟周期，当触发信号发生时间间隔输出先进入空闲，当 BM-counter 等于 HRPWM_BCMAR 时进入运行；当 HRPWM_BMCR.BMMD 为 1 时，间隔输出工作在模式 2，间隔输出的空闲时间的持续时间为 HRPWM_BMPCMAR 个间隔输出计数时钟周期，当触发信号发生时间间隔输出先进入 1 个间隔输出计数时钟周期的运行状态，再进入空闲。两种模式下间隔输出运行状态如图 19-19 和图 19-20 所示。

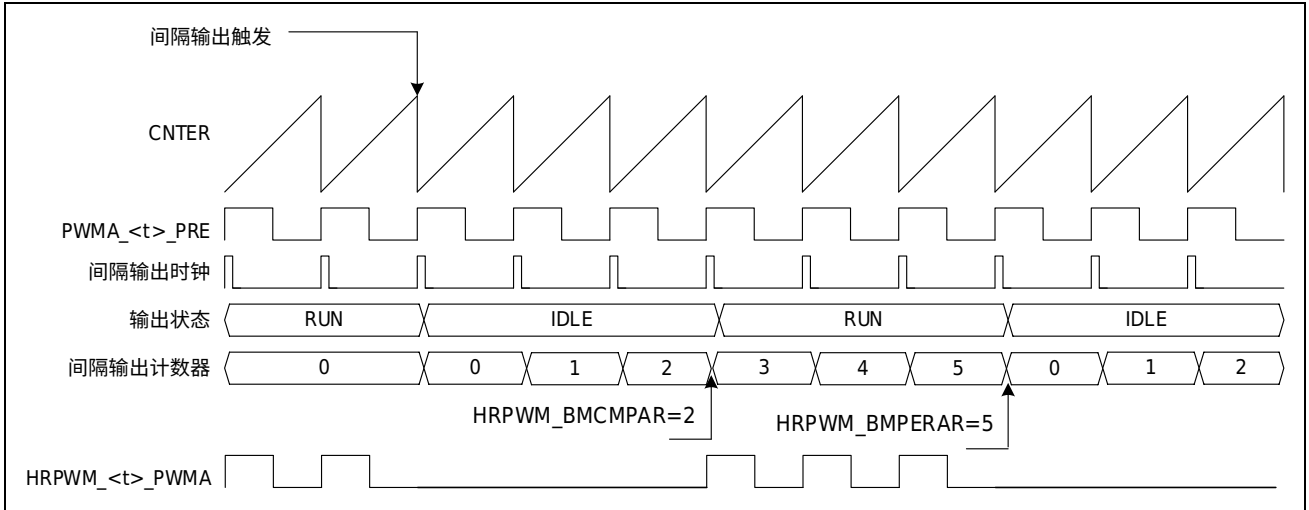


图 19-19 间隔输出模式 1 时序图

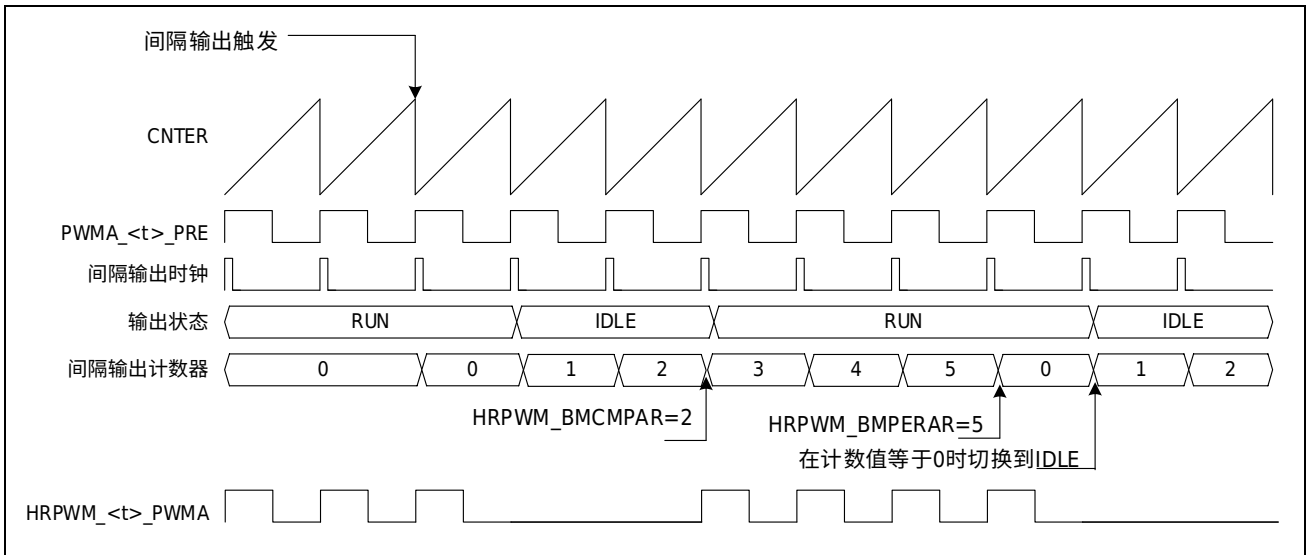


图 19-20 间隔输出模式 2 时序图

间隔输出触发源

间隔输出触发源一共有以下几种，通过寄存器 HRPWM_BMSTRG1<2>进行选择：

- 软件触发
- 6×4 个来自各个单元的事件：计数值等于 0、计数值等于周期值或者锯齿波硬件清零、计数值等于高分辨率通用比较基准值 HRGCMAR 和计数值等于高分辨率通用比较基准值 HRGCMBR
- TMR0 的比较事件
- 外部事件 7（由单元 1 滤波）和外部事件 8（由单元 4 滤波）
- 外部事件 7 之后的单元 1 的计数值等于周期值或者锯齿波硬件清零/计数值等于 0
- 外部事件 8 之后的单元 4 的计数值等于周期值或者锯齿波硬件清零/计数值等于 0

可以同时选择多个触发源有效。在间隔输出工作期间（空闲和运行），间隔输出不可被再次触发。

间隔输出进入及状态切换

间隔输出的进入、间隔模式期间由运行向空闲，或者空闲向运行转换时，分为两种模式，一种是常规模式，一种是延迟模式。

常规模式

当 $HRPWM\langle t \rangle_IDLECR.DIDLA\langle B \rangle$ 位为 0 时，为间隔输出的常规进入模式，当触发事件到来， $HRPWM\langle t \rangle_PWMA\langle B \rangle$ 的端口将立即进入预设的空闲状态；在间隔输出期间当由空闲切回运行或者运行切回空闲时， $HRPWM\langle t \rangle_PWMA\langle B \rangle$ 的端口根据间隔输出控制器的状态立刻进入空闲，或者立刻进入运行。下图是在外部事件或者外部事件后完整周期点(由 $HRPWM\langle t \rangle_GCONR1.PRDSEL$ 定义)触发下进入间隔输出以及在间隔输出期间状态切换的图例。

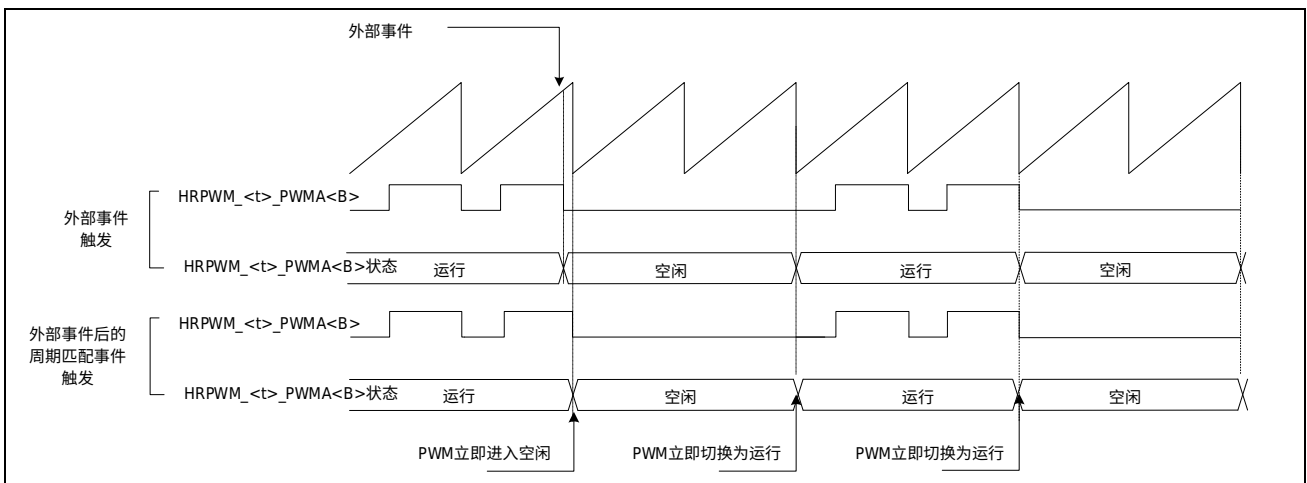


图 19-21 间隔输出的常规模式

延迟模式

当 $HRPWM\langle t \rangle_IDLECR.DIDLA\langle B \rangle$ 位为 1 时，为间隔输出的延迟模式。在进入间隔输出以及在间隔输出期间进行状态切换时，PWM 会在完整周期点发生后检测到设置的空闲状态与内部 PWM ($PWMA\langle B \rangle_PRE$) 输出状态一致时切换空闲或者运行。在延迟模式下，间隔输出空闲或者运行时间必须大于或者等于一个 PWM 的周期。

图 19-22 是延迟模式下进入间隔输出的图例，图 19-23 是延迟模式下间隔输出期间状态切换的图例。

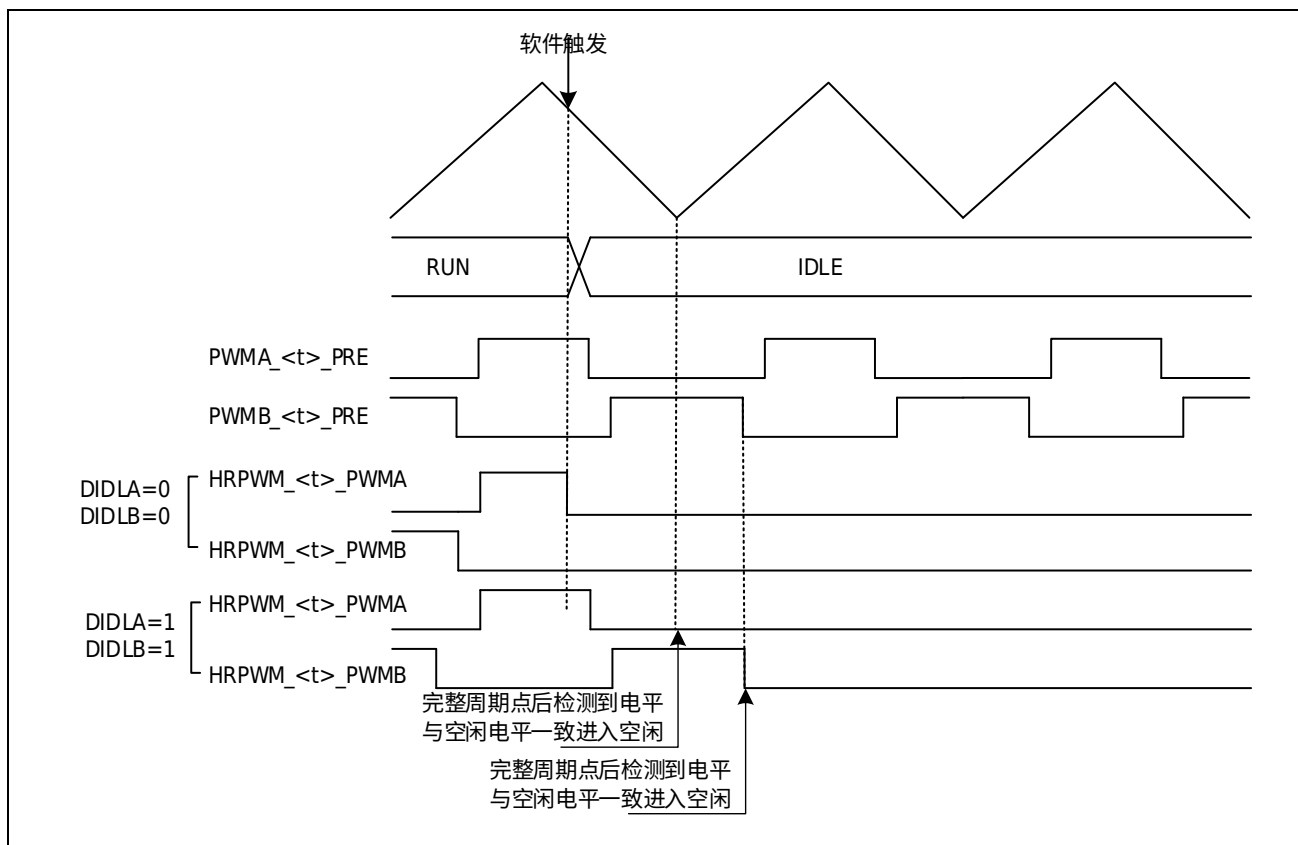


图 19-22 延迟模式有效间隔输出进入图例

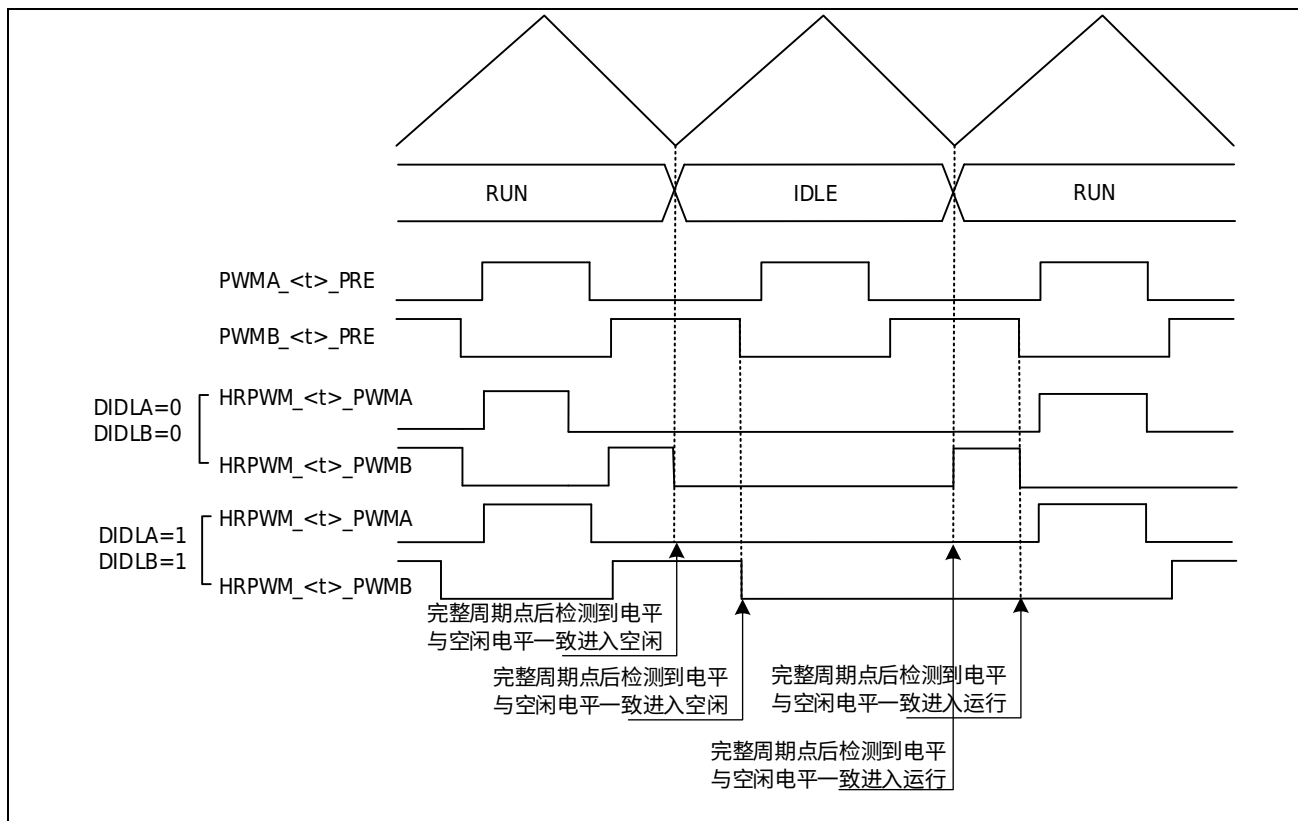


图 19-23 延迟模式有效间隔输出期间状态切换图例

间隔输出的退出

在连续模式下，间隔输出由软件强制退出；单次模式下，一旦经过空闲周期，就退出间隔输出。连续模式下，对 HRPWM_BMCR.BMOPTF 或者 BMEN 位写 0 退出间隔输出，退出时会在完整周期点发生后检测到设置的空闲状态与内部 PWM (PWMA<t>_PRE) 输出状态一致时退出间隔输出空闲状态，如图 19-24。

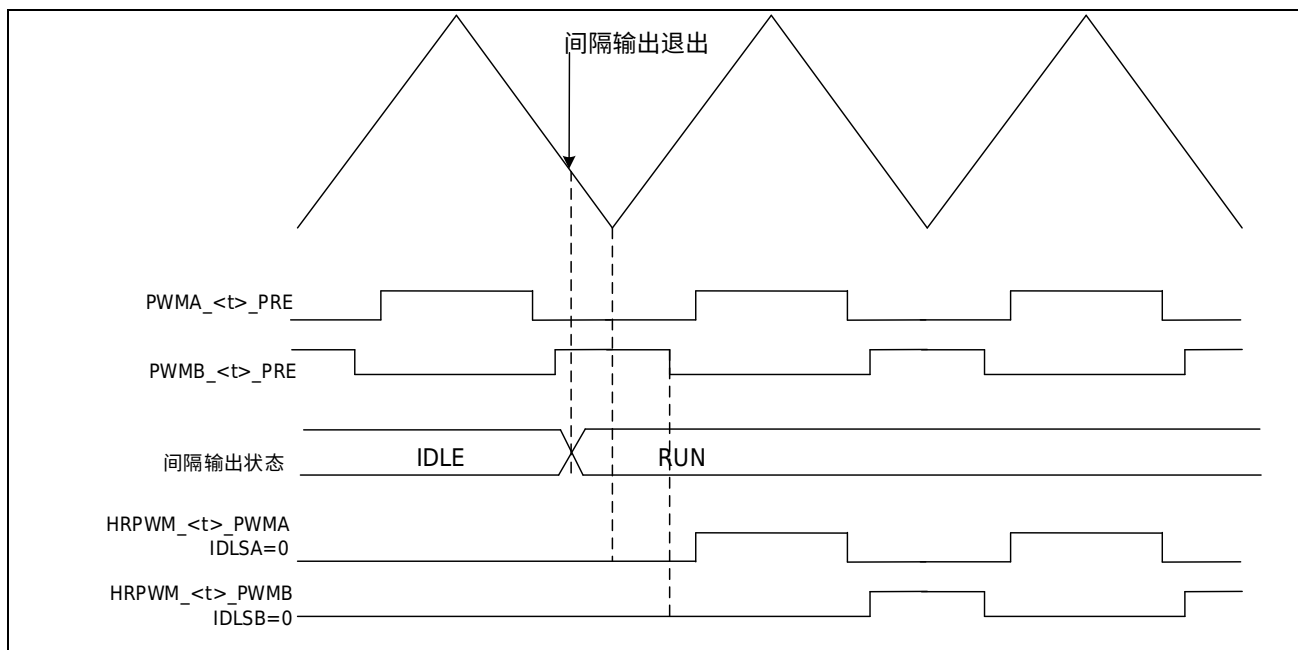


图 19-24 间隔输出的退出

注:

在对 HRPWM_BMCR.BMOPTF 或者 BMEN 位写 0 退出间隔输出后，在所有使能了间隔输出功能的通道均退出之前，间隔输出无法被再次触发，在所有通道退出间隔输出后，HRPWM_BMCR.BMOPTF 才会被硬件清零。

间隔输出跟随功能

HRPWM<t>_PWMB 的端口在间隔输出进入、退出以及间隔输出期间进行状态切换时，可跟随 PWM<A>_<t>_PRE 进出空闲。该功能仅在 PWMA 与 PWMB 空闲电平设置相同 (HRPWM<t>_IDLECR.IDLESA= HRPWM<t>_IDLECR.IDLESB)，且 A 通道与 B 通道均开启了延迟模式 (HRPWM<t>_IDLECR.DIDLA=1) 时有效。在该模式下，A 通道在间隔输出进入、退出或者在间隔输出期间在空闲和运行进行切换时，依然在完整周期点后检测到电平一致时进入空闲，或者退出空闲。B 通道进入空闲的时刻发生在 A 通道进入后检测到 PWM_<t>_PRE 发生翻转时进入，退出发生在 A 通道退出空闲后检测到 PWM_<t>_PRE 发生翻转时退出，具体退出时间参考表 19-4。图 19-25 为通道 A 和 B 空闲电平均设置为低电平，开启跟随功能时，通道 B 进入和退出空闲状态的图例。

表 19-4 通道 B 跟随功能退出和进入空闲时刻

HRPWM<t>_IDLECR.IDLESA	HRPWM<t>_IDLECR.IDLESB	通道 B 退出时刻
0	0	PWM_<t>_PRE 下降沿
1	1	PWM_<t>_PRE 上升沿
其他		在完整周期点后检测到电平一致进入和退出

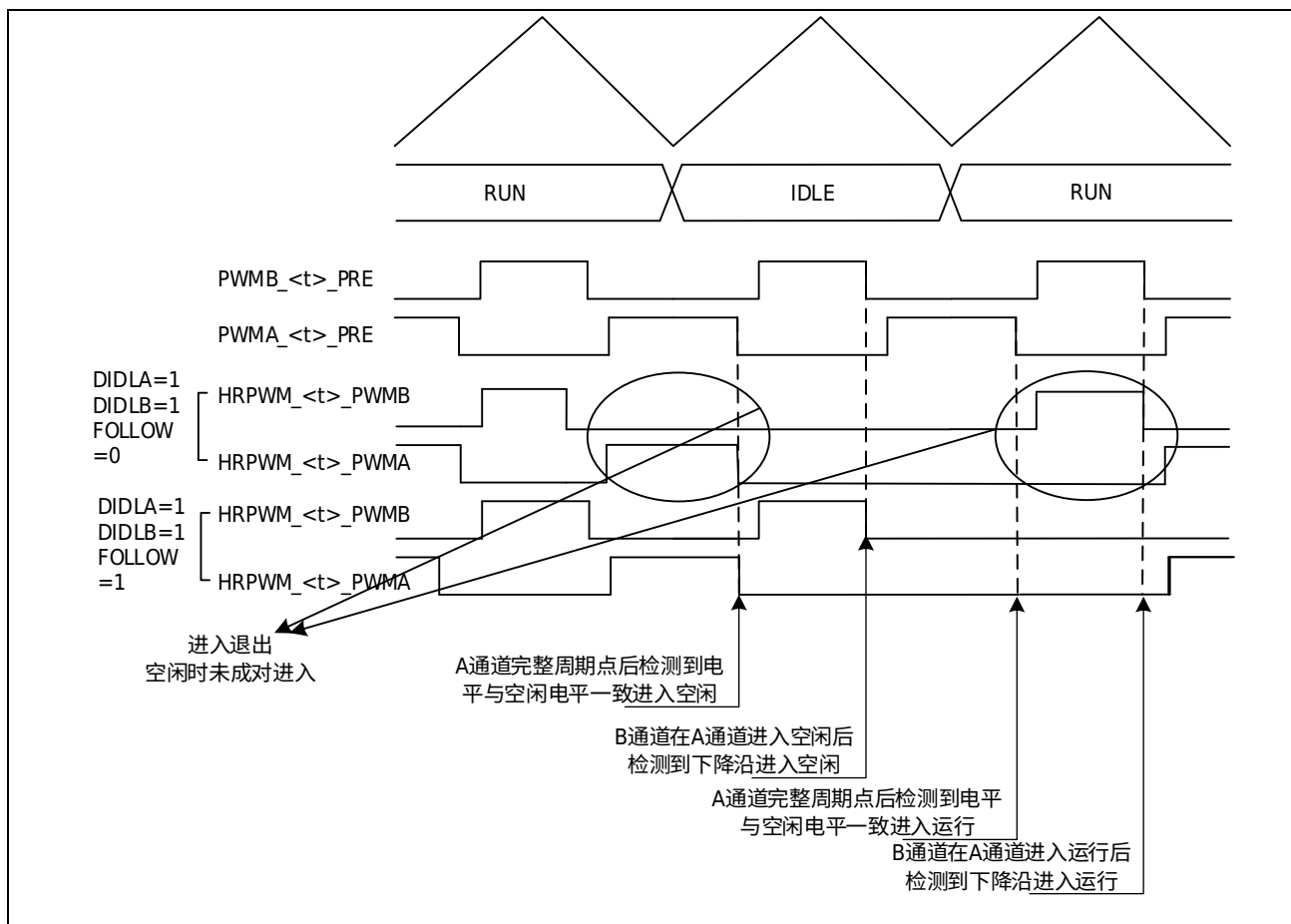


图 19-25 通道 B 跟随功能退出和进入空闲图例

19.3.10 相位功能

利用 HRPWM 的相位功能可以实现多相交错输出。HRPWM 包括 6 个计数单元，其中单元 1 可作为主单元同步单元 2~6。

单元 1 包括 5 个高分辨率相位比较基准值寄存器 HRPWM<t>_PHSCMP1~5A，可以在计数比较匹配时产生相位匹配事件。

5 个相位比较基准值寄存器都具有缓存功能，通过设置单元 1 相位控制寄存器(HRPWM<t>_PHSCTL)的 BENPHS 位使能缓存功能，缓存功能章节。

单元 2~6 作为从单元，在相位寄存器控制寄存器(HRPWM<t>_PHSCTL)的 PHSEN 使能时，通过设置 HRPWM<t>_PHSCTL.PHCMPSEL 位选择单元 1 的 5 个相位匹配事件中的一个作为相位匹配事件。从单

元在相位匹配事件发生时启动和清零。图 19-26 为三角波模式多相 PWM 主从单元计数器启动和清零的动作例，该示例中从单元配置为单次计数模式；图 19-27 为锯齿波模式多相 PWM 主从单元计数器启动和清零的动作例，该示例中从单元周期值设置为最大值。

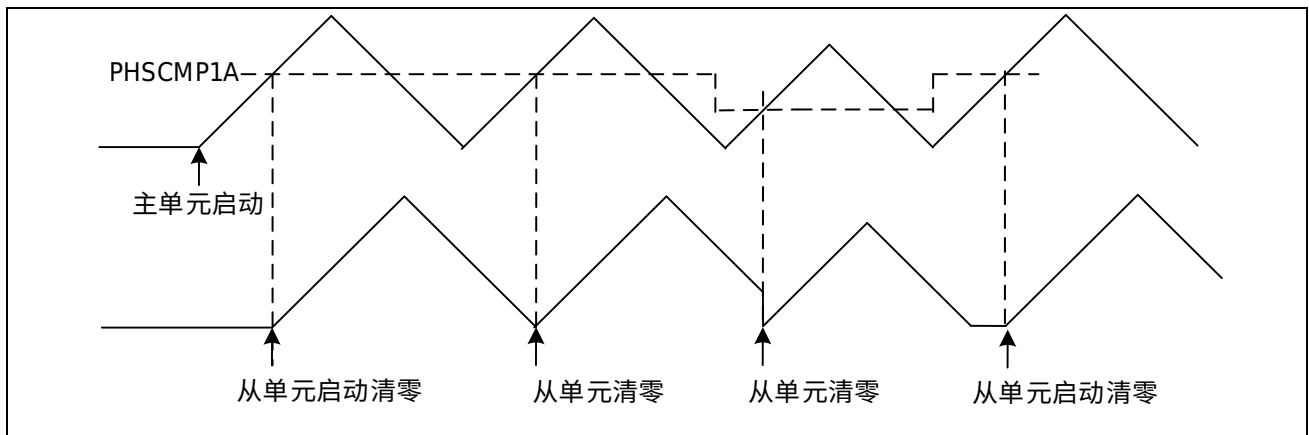


图 19-26 三角波模式多相 PWM 输出从单元启动、清零动作

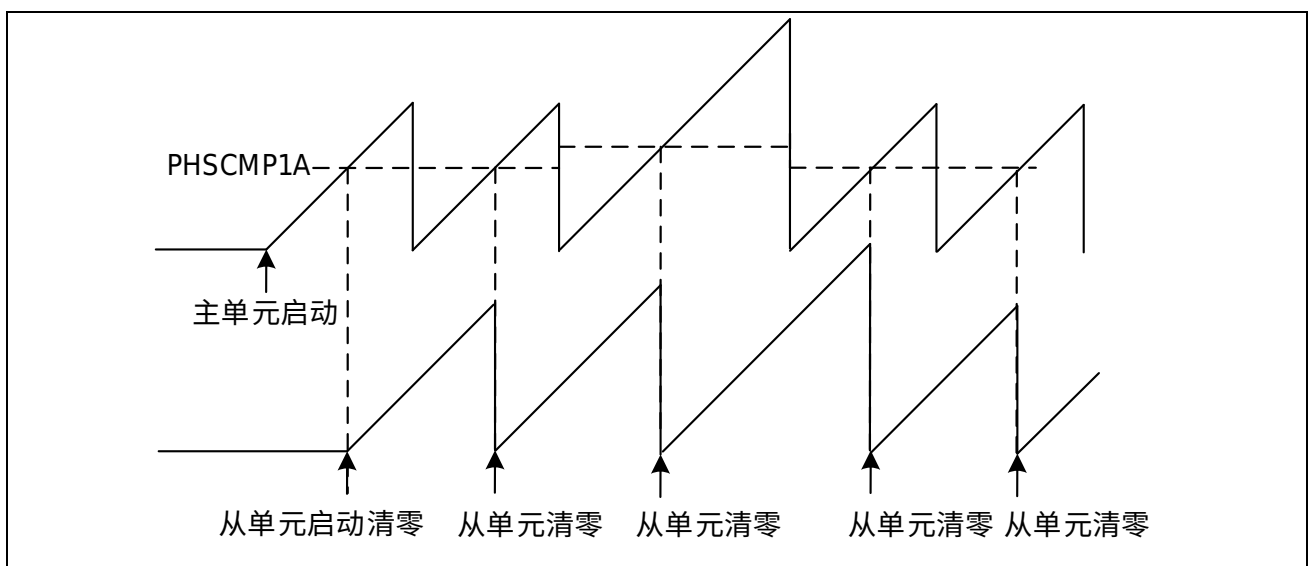


图 19-27 锯齿波模式多相 PWM 输出从单元启动、清零动作

在多相 PWM 输出期间，通过高分辨率周期基准值寄存器、高分辨比较基准值寄存器以及高分辨率相位比较基准值等的缓存功能，可以实现周期、占空比、死区等的动态调整，具体可参考【单元间多相交错 PWM 输出】。从单元通过设置 HRPWM<t>_PHSCTL 的 PHSFORCA位，可以配置在锯齿波和三角波不同模式下，对 PWM 进行强制设置为低，避免在多相 PWM 输出期间进行动态调整时出现连续波。在三角波模式下，使能对应位可以在相位匹配事件发生时将 PWM 配置为低电平；在锯齿波模式下，使能对应位可以在单元 1 单次缓存传送点检测到当前计数值大于等于缓存比较基准值寄存器中的值将 PWM 配置为低电平，上述功能仅在相位功能使能时有效 (HRPWM<t>_PHSCTL.PHSEN=1)，具体可参考【单元间多相交错 PWM 输出】章节。

主单元相位匹配事件发生到启动和清零从单元之间有 1 个 PCLK0 延迟。设定高分辨率相位比较基准值

寄存器 PHSCMP1~5A 按照如下公式设定：

- $PHSCMPxA = (M-1) * HRPERAR * 2 / N - 64$ (三角波模式)
- $PHSCMPxA = (M-1) * (HRPERAR + 64) / N - 64$ (锯齿波模式)
- N=相数
- M=PWM 单元编号 (M=2~6)

周期链接

配置 $HRPWM<t>_GCONR1.PRD LK$ 为 1 使能周期链接功能 (仅单元 2~6)，在该模式下本单元周期值寄存器 $HRPWM<t>_HRPERBR$ (t=2~6) 的值与单元 1 的周期值一致。

19.3.11 外部事件

HRPWM 有 10 组外部事件，可以用于所有的单元。通过 $HRPWM_EECR1$ 和 $HRPWM_EECR2$ 寄存器配置外部事件：

- 通过配置 $EEySRC(y=1\sim 10)$ 选择外部事件的源 (每个外部事件可以选择 4 个源)，详见表 19-5
- 通过配置 $EEySNS[1:0]$ 选择外部事件的有效边沿，可以设置为电平敏感或者边沿敏感 (上升、下降、双边沿)
- 当选择电平敏感时，通过 $EEyPOL$ 位选择外部事件电平极性

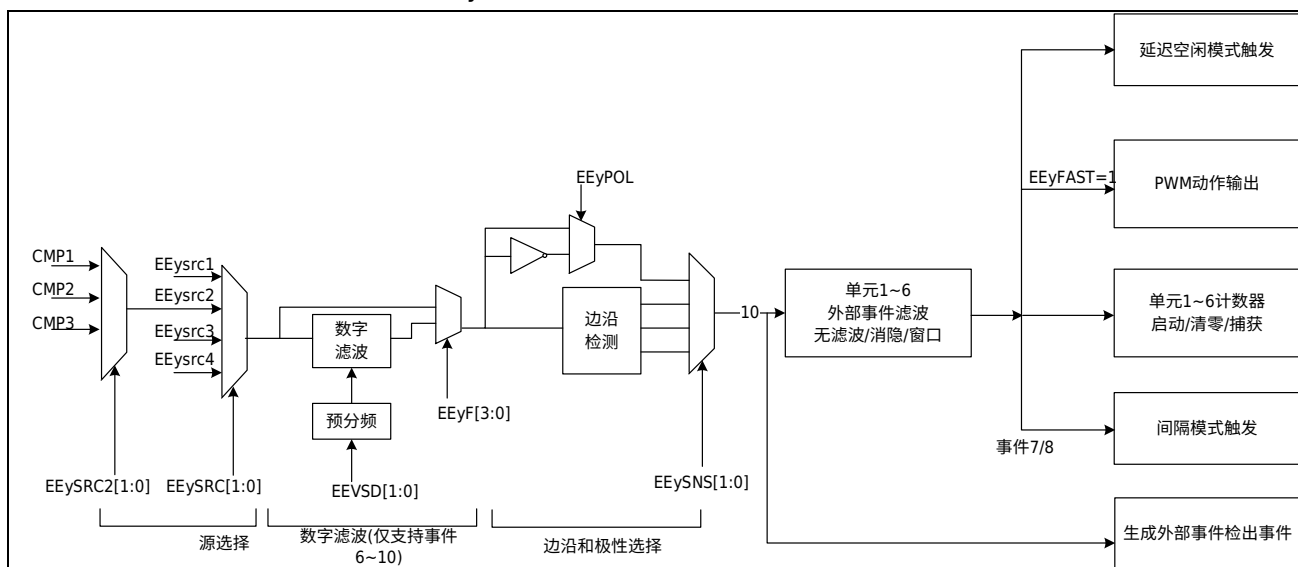


图 19-28 外部事件处理框图

这些外部事件可以用于触发硬件启动、清零、刷新计数器，间隔输出，延迟空闲以及设置端口状态，同时也可生成外部事件检出事件。

注：

外部事件用于触发硬件启动、清零、刷新计数器、捕获、间隔输出、延迟空闲以及生成外部事件检出事件时仅边沿敏感，即使 $EEySNS$ 位设置为 $0b00$ (电平敏感)，当 $EEyPOL=0$ 时，检测到上升沿时生成

触发信号，为 1 时检测到下降沿时生成触发信号。当 `HRPWM_GCTLR.BKDLY` 设置为 1 且电平有效时，将使能消隐模式外部事件延迟功能，当单元内的消隐窗口指定时间完成时，监测到外部事件为有效电平时，将生成触发信号。

外部事件 1~5 具有快速异步模式，通过 `HRPWM_EECCR1.EEVyFAST` 寄存器位配置，快速异步模式仅支持电平有效，快速异步使能时，如果将对应外部事件配置为边沿有效，外部事件将无快速异步功能。

外部事件快速异步模式仅支持控制 PWM 置 0 和置 1，不支持翻转功能，在快速异步模式使能时请勿将 `HRPWM<t>_PCNAR2.EXEVyCA` 设置为 11。三角波模式下使用快速异步模式时，外部事件向上计数和向下的动作必须设置为一致。

外部事件快速异步模式支持消隐窗口功能，当消隐后的外部事件宽度小于 6 个时钟周期时，PWM 最大存在 3 个时钟周期无法进入外部事件配置的电平状态。外部事件 6~10 具有数字滤波功能，通过 `HRPWM_EECCR3` 寄存器配置。

采样时钟可以配置为 `PCLK0` 的分频，或者从 `PCLK0` 分频来的 `fEEVS`，`fEEVS` 由 `HRPWM_EECCR3` 的 `EEVSD[1: 0]` 位设置。

`EEyFM[3:0]` 位可以选择滤波采样时钟和滤波长度 `N`，在滤波采样基准时钟采样到端口上 `N` 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 `N` 次一致的电平会被当作外部干扰滤掉，不传送到模块内部。

10 个外部事件的可选择源如下表：

表 19-5 外部事件映射表

外部事件	EEysrc1	EEysrc2	EEysrc3	EEysrc4
外部事件 1	HRPWM_EEV1	EE1src2	TMR0_1CMPA	ADC1_CMP0
外部事件 2	HRPWM_EEV2	EE2src2	TMR0_1CMPB	ADC1_CMP1
外部事件 3	HRPWM_EEV3	EE3src2	TMR0_1CMPA	-
外部事件 4	HRPWM_EEV4	EE4src2	-	ADC2_CMP0
外部事件 5	HRPWM_EEV5	EE5src2	-	ADC2_CMP1
外部事件 6	HRPWM_EEV6	EE6src2	TMR0_2CMPA	-
外部事件 7	HRPWM_EEV7	EE7src2	TMR0_2CMPB	ADC3_CMP0
外部事件 8	HRPWM_EEV8	EE8src2	-	ADC3_CMP1
外部事件 9	HRPWM_EEV9	EE9src2	TMR0_1CMPB	-
外部事件 10	HRPWM_EEV10	EE10src2	-	-

表 19-5 中 EEysrc2 的源可进一步由 HRPWM_GCTLR.EEySRC2[1:0]寄存器位进行选择，参考下表。

表 19-6 外部事件 EEysrc2 选则

EEySRC2[1:0]	EEysrc2
00	CMP1
01	CMP2
10	CMP3
11	无事件选择

19.3.11.1 单元内外部事件过滤功能

10 个外部事件可以用于所有单元，他们可以直接被各个单元使用，或者在限制的时间内被使用。在每个单元内，10 个外部事件可以通过外部事件滤波控制寄存器 HRPWM<t>_EEFLTCR1<2>.EEyFM (y=0~9) 选择以下两种事件滤波模式：

消隐模式：在指定时间发生的外部事件被忽略

窗口模式：在指定时间发生的外部事件有效

消隐模式下的消隐信号和窗口模式下窗口信号可以由本单元事件滤波偏移值寄存器 HRPWM<t>_EEOFFSETR，外部事件滤波窗口值寄存器 HRPWM<t>_EEFWINR 以及其对应的外部事件滤波初始电平寄存器 HRPWM<t>_EEFLTCR1.EEINTPOL 生成(定义计数器开始计数时消隐信号和窗口信号的初始电平)，也可以选择其它单元生成的消隐或者窗口信号，消隐模式时消隐信号还可以选择为其他单元的 PWMA_<t>_PRE1（空闲处理后的 PWM）。

偏移值 HRPWM<t>_EEOFFSETR.OFFSET 和窗口值 HRPWM<t>_EEFWINR.WIN 均定义为从计数器 0 开始的计数时间，计数值为 (OFFSET+64) 或 (WIN+64) 生效。HRPWM<t>_EEOFFSETR.OFFSETDIR 定义为偏移值计数方向， HRPWM<t>_EEFWINR.WINDIR 定义为窗口值计数方向。

在消隐模式下，当计数器计数方向为偏移值计数方向且计数值达到偏移值时，消隐信号产生复位；当计数器计数方向为窗口值计数方向且计数值达到窗口值时，消隐信号产生置位。

在窗口模式下，当计数器计数方向为偏移值计数方向且计数值达到偏移值时，窗口信号产生置位；当计数器计数方向为窗口值计数方向且计数值达到窗口值时，窗口信号产生复位。

当窗口值和偏移值同时发生时，偏移值有较高优先级。计数器开始计数时的初始电平由 HRPWM<t>_EEFLTCR1.EEINTPOL 设定，计数器停止后，不进行滤波。

注：

三角波计数时，当 OFFSET 或 WIN 设为周期值时，请将对应的 OFFSETDIR 或 WINDIR 设为 0，当 OFFSET 或 WIN 设为 0 时，请将对应的 OFFSETDIR 或 WINDIR 设为 1；锯齿波计数时，请将对应的 OFFSETDIR 和 WINDIR 都设为 1；外部事件滤波窗口值和偏移值不支持高精度。

消隐模式

在消隐模式中，在指定时间内发生的外部事件将被忽略，其他时间发生的外部事件有效。消隐模式的指定时间指计数器计数值从 OFFSET 到 WIN 的时间，如图 19-30 所示。

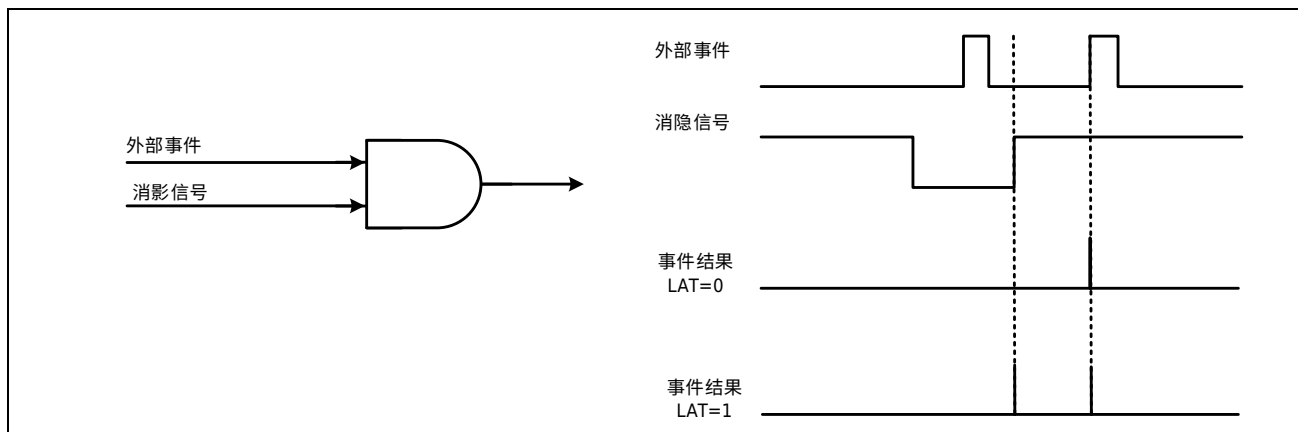


图 19-29 事件消隐模式

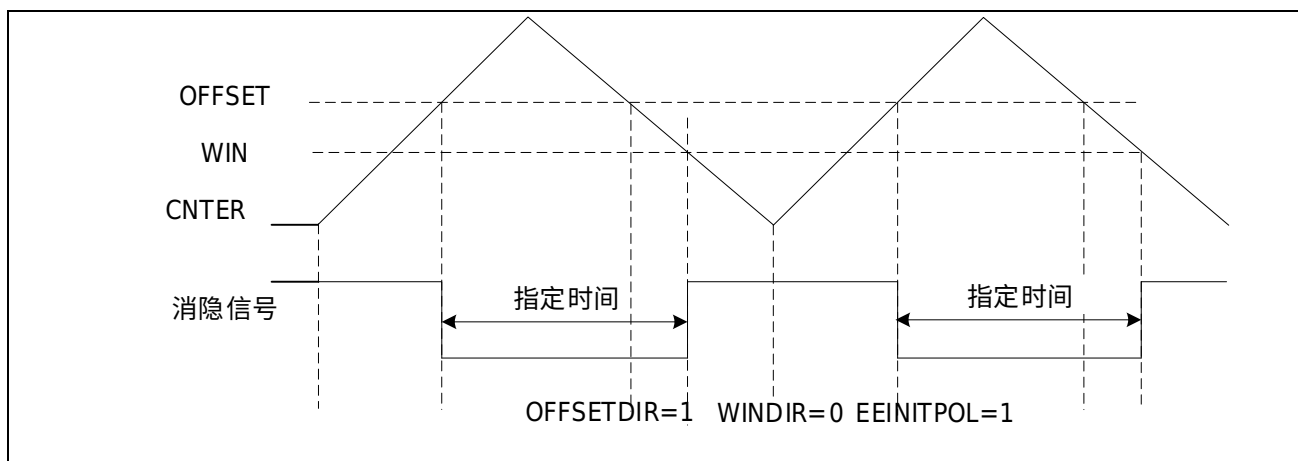


图 19-30 消隐模式指定时间

在消隐模式下，当 $HRPWM\langle t \rangle_EEFLTCR1\langle 2 \rangle.EVE1yLAT$ 设置为 1 时，将使能事件的外部存储功能，外部事件不会立即生效。一旦指定时间完成，该外部事件将被存储并生成，如图 19-29。

消隐模式下，用于触发硬件启动、清零、刷新计数器、捕获、间隔输出、延迟空闲外部事件具有消隐延迟功能。当 $HRPWM_GCTLR.BKDLY\langle t \rangle(t=1\sim 6)$ 设置为 1 时，将使能对应单元的外部事件消隐延迟功能，当外部事件选择电平有效时，指定时间完成时外部事件为有效电平时将生成触发信号，该功能仅在外部存储功能无效时有效。如图 19-31，外部事件有效电平设置为高电平，在消隐窗口结束时检测到外部事件为高电平，此时生成触发信号。

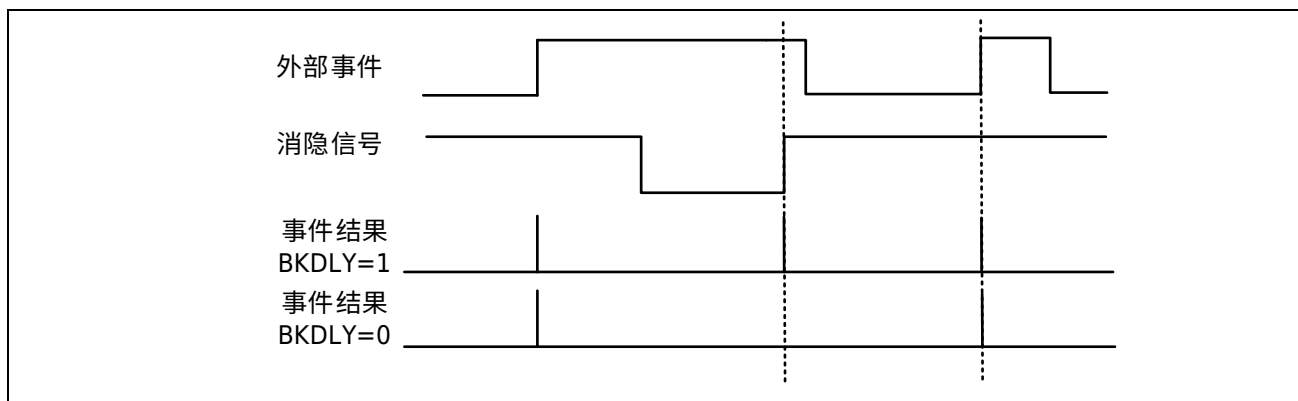


图 19-31 消隐模式下触发计数器清零的外部事件延迟功能

消隐信号包括以下信号源：

本单元：(HRPWM<t>_EEFLTCR1<2>.EEyFM[3:0]=00001) 指定时间为来自本单元偏移值到窗口值的时间，具体参照图 19-30。

来自其他单元：(HRPWM<t>_EEFLTCR1<2>.EEyFM[3:0]=0010~1101) 指定时间是指其他单元的偏移值到窗口值的时间；也可以选为其他单元的 PWMA_<t>_PRE1 信号(在这种情况下，只要该 PWM 信号为低电平，事件将被忽略)。

此外，通过配置 HRPWM<t>_GCONR1.EEFM 位可以对 HRPWM<t>_EEFLTCR1<2>.EEyFM[3:0] = 0010~0011 所选的消隐信号进行替换。替换后的消隐信号的消隐时间可设置为参考点到偏移值或者窗口值的时间；通过配置 HRPWM<t>_GCONR1.EEREF 位可以配置参考点为本单元计数值等于 0 以及计数值等于周期值或者锯齿波硬件清零，具体参考下表。当窗口值或者偏移值设定的与参考值重合时，消隐时间为整个计数区间。

表 19-7 HRPWM<t>_EEFLTCR1<2>.EEyFM 对应的消隐时间设定

EEFLTCR1<2>.EEyFM[3:0]	GCONR1.EEFM=0 GCONR1.EEREF=x	GCONR1.EEFM=1 GCONR1.EEREF=0	GCONR1.EEREF=1 GCONR1.EEFM=1
0b0010	消隐时间指单元 1 偏移值到窗口值的时间	消隐时间指本单元计数值等于周期值或者锯齿波硬件清零到偏移值的时间	消隐时间指本单元计数值等于 0 到偏移值的时间
0b0011	PWMA_1_PRE1	消隐时间指本单元计数值等于周期值或者锯齿波硬件清零到窗口值的时间	消隐时间指本单元计数值等于到窗口值的时间

窗口模式

在窗口模式中，在指定时间内发生的外部事件有效，其他时间发生的外部事件被忽略。窗口模式的指定时间为计数器计数从 OFF 到 WIN 的时间。在窗口模式下，当 HRPWM<t>_EEFLTCR1<2>.EEyLAT 设置为 0 且 HRPWM<t>_EEFLTCR1<2>.EEyTMO 设置为 0，指定时间内的事件有效，若将 HRPWM<t>_EEFLTCR1<2>.EEyTMO 设置为 1，如果在指定时间内未发生外部事件，则在指定时间结束时将产生超时事件；当 HRPWM<t>_EEFLTCR1<2>.EEyLAT 设置为 1 时，将使能事件的外部存储功

能，如果在指定时间内发生外部事件，外部事件不会立即生效，一旦指定时间完成，该外部事件将被存储并生成，如图 19-32。窗口模式通过配置 $HRPWM\langle t \rangle_EEFLTCR1\langle 2 \rangle.EEyFM[3:0]$ 位，范围从 1110~1111。

注： $HRPWM\langle t \rangle_EEFLTCR1\langle 2 \rangle.EEyLAT$ 设置为 1 时， $HRPWM\langle t \rangle_EEFLTCR1\langle 2 \rangle.EEyTMO$ 设置无效。

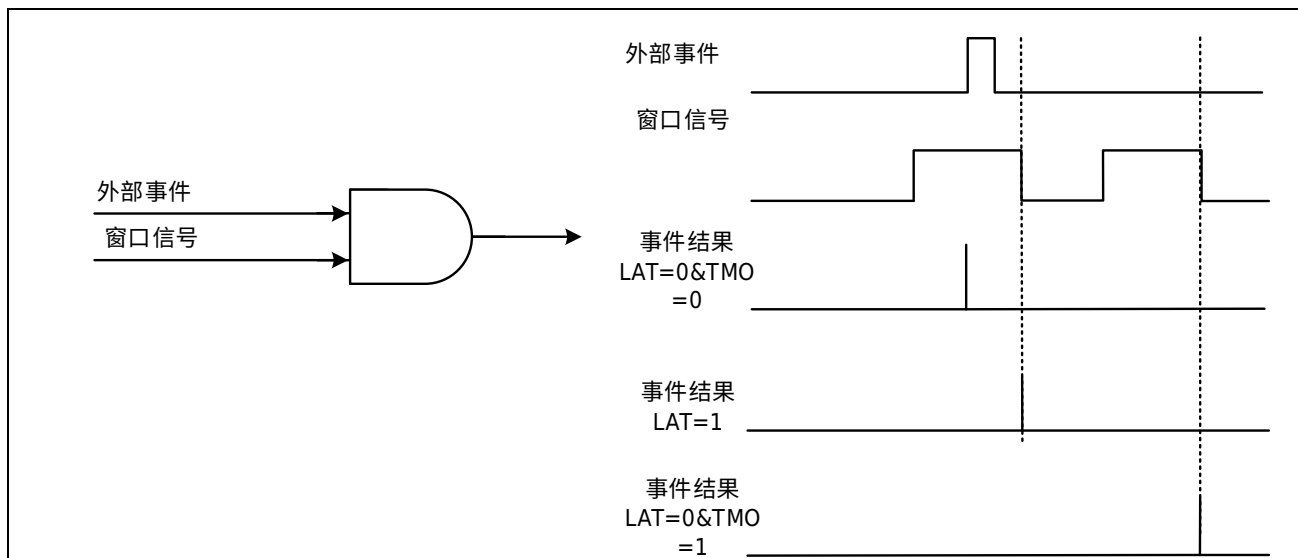


图 19-32 事件窗口模式

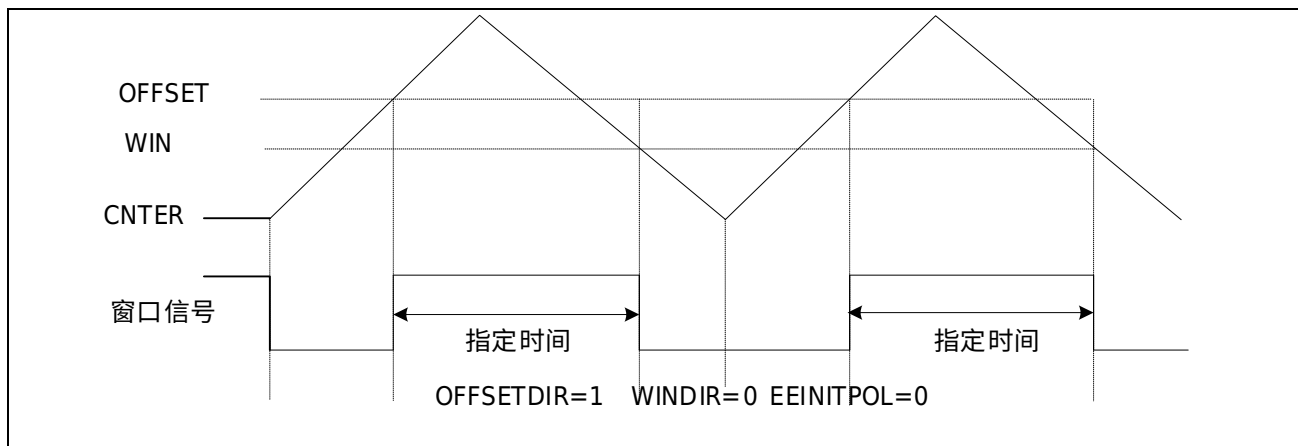


图 19-33 窗口事件指定时间

窗口信号源包括以下两种：

本单元： $(HRPWM\langle t \rangle_EEFLTCR1\langle 2 \rangle.EEyFM[3:0]=0b1110)$ 指定时间为本单元偏移值到窗口值的时间，具体参照图 19-33。

来自其他单元： $(HRPWM\langle t \rangle_EEFLTCR1\langle 2 \rangle.EEyFM[3:0]=0b1111)$ 指定时间为其他单元偏移值到窗口值的时间，详见表 19-8。

表 19-8 窗口模式下滤波信号映射

目标	单元 1	单元 2	单元 3	单元 4	单元 5	单元 6
TMRWIN	单元 2	单元 1	单元 4	单元 3	单元 6	单元 5

此外，通过配置 HRPWM<t>_GCONR1.EEFM 位可以将 HRPWM<t>_EEFLTCR1<2>.EEyFM[3:0]=1110~1111 对应的窗口信号进行替换。替换后的窗口信号的窗口时间为参考点到偏移值或者窗口值的时间；通过配置 HRPWM<t>_GCONR1.EEREF 位可以配置参考点为本单元计数值等于 0 以及计数值等于周期值或者锯齿波硬件清零，具体参考下表。当窗口值或者偏移值设定的与参考值重合时，窗口时间为整个计数区间。

表 19-9 HRPWM<t>_EEFLTCR1<2>.EEyFM 对应的窗口时间设定

EEFLTCR1<2>.EEyFM[3:0]	GCONR1.EEFM=0 GCONR1.EEREF=x	GCONR1.EEFM=1 GCONR1.EEREF=0	GCONR1.EEREF=1 GCONR1.EEFM=1
0b1110	窗口时间是指本单元偏移值到窗口值的时间	窗口时间指本单元计数值等于周期值或者锯齿波硬件清零到偏移值的时间	窗口时间指本单元计数值等于 0 到偏移值的时间
0b1111	窗口信号来自其他单元，TMRWIN 源	窗口时间指本单元计数值等于周期值或者锯齿波硬件清零到窗口值的时间	窗口时间指本单元计数值等于 0 到窗口值的时间

19.3.12 软件同步

19.3.12.1 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 (CNTERR) 的同步启动。

19.3.12.2 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 (CNTERR) 的同步停止。

19.3.12.3 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 (CNTERR) 的同步清零。

19.3.12.4 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 (CNTERR) 的同步刷新。软件同步刷新可再计数器计数过程中将计数器的值更新为任意值，这可能会导致比较匹配事件的丢失，在使用时需注意。

如图 19-34 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。

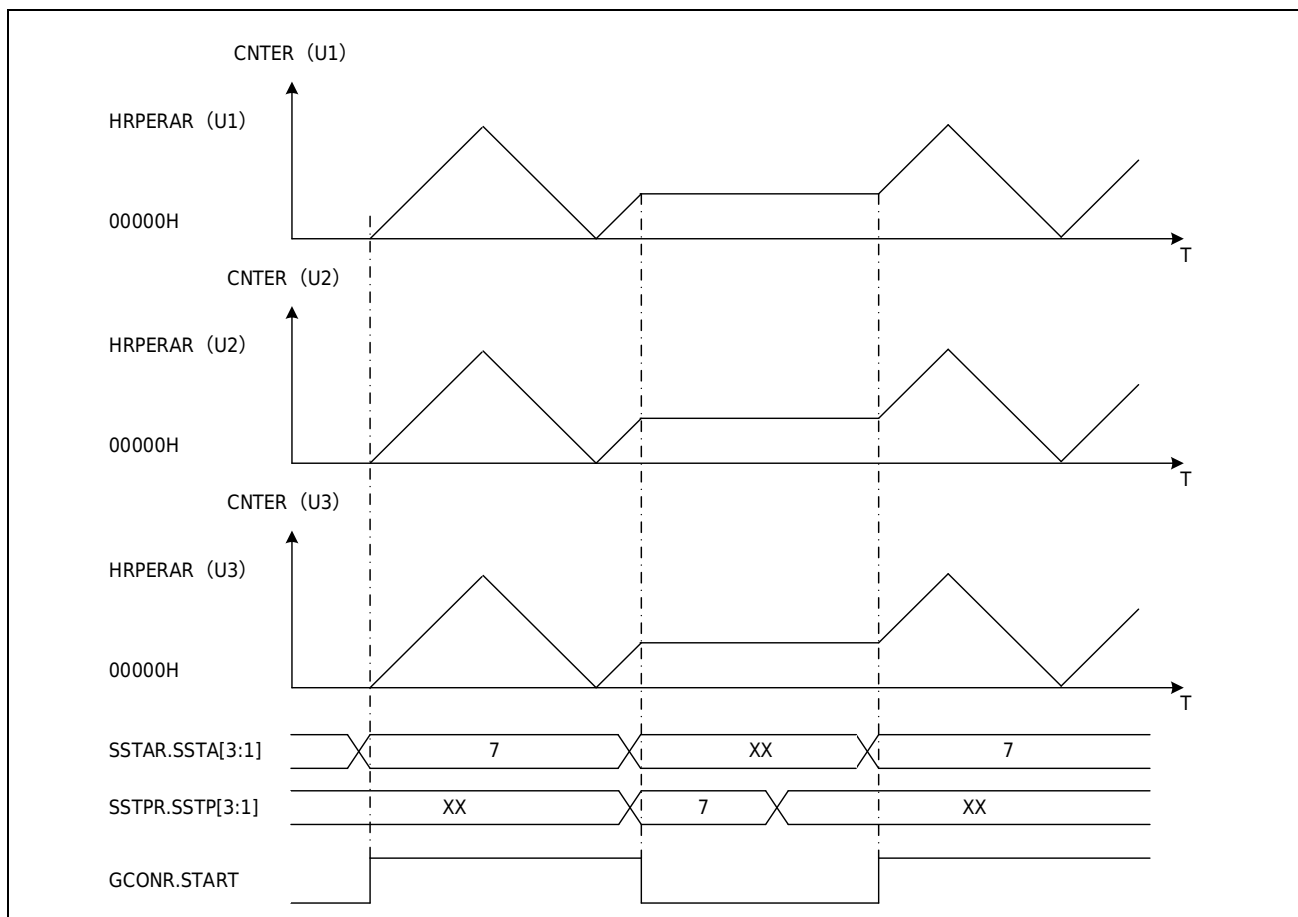


图 19-34 软件同步动作

19.3.13 硬件同步

每个单元拥有 4 个通用外部触发输入端口（HRPWM_TRIGA、HRPWM_TRIGB、HRPWM_TRIGC、HRPWM_TRIGD）、4 个内部触发事件输入条件、10 个外部事件以及各个单元的比较匹配事件，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 HRPWM 硬件触发事件选择寄存器（HRPWM_TRGSEL0~3）中对应的编号设定来选择，具体的事件对应关系请参考 AOS 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 1。

19.3.13.1 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（HSTAR1、HSTAR2）的设定来决定。

19.3.13.2 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（HCLRR1、HCLRR2）的设定来决定。

19.3.13.3 硬件同步捕获

各单元均可选择用硬件方式实现捕获功能，选择相同捕获输入功能条件的单元即可在捕获功能条件有效时实现同步捕获。具体的硬件捕获功能条件由硬件捕获事件选择寄存器（HRPWM<t>_HCPAR1<2>、HRPWM<t>_HCPBR1<2>）的设定来决定。

19.3.14 同步输入端口数字滤波

HRPWM_TRIGA~D 端口输入具有数字滤波功能。HRPWM_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 19-35 所示。

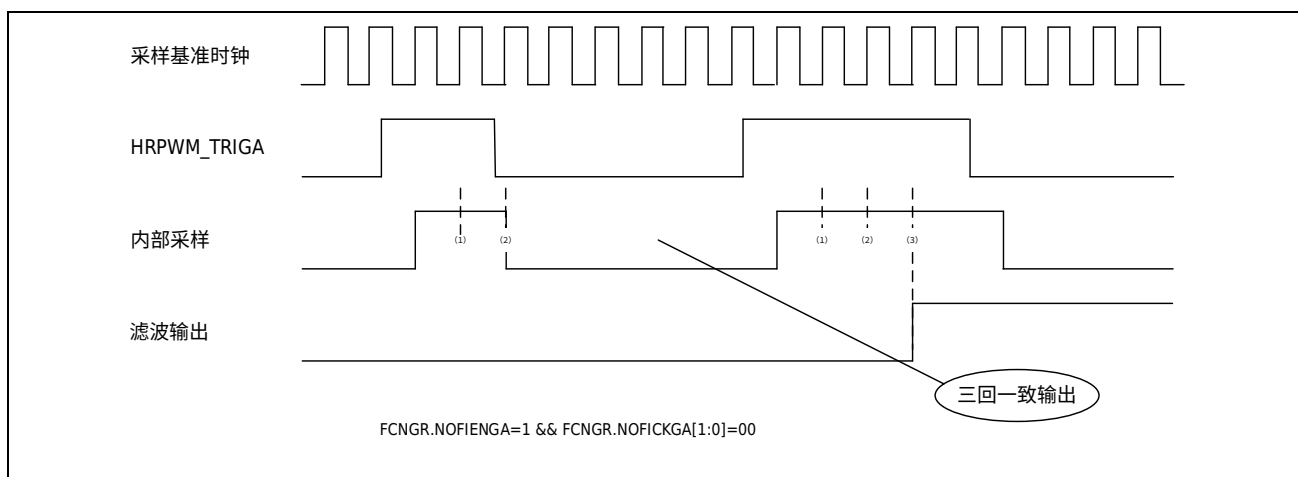


图 19-35 HRPWM_TRIGA 的滤波功能

19.3.15 同步输出

可以将 HRPWM 作为主机，同步外部资源。通过配置 HRPWM_SYNOCR 寄存器的 SYNC SR[3: 0]位，可以选择发送到同步输出上的源：

- 单元 1~6 计数器处于计数中且计数值等于 0 时
- 单元 1~6 计数器处于计数中且计数值等于 SCMBR 时

当同步源选为 SCMBR 时，通过配置 HRPWM_SYNOCR 寄存器的 SRCDIR 位为计数值等于 SCMBR 时计数器的有效方向，在三角波模式下可以设置为向上计数区间或者向下计数区间有效，在锯齿波模式下请将 SRCDIR 设置为 1。

通过 HRPWM_SYNOCR 寄存器的 SYNOPLS [1: 0]位可以配置同步输出信号的极性：

0b0x：脉冲产生禁止

0b10：在同步输出引脚 HRPWM_SCOUT 上产生正脉冲。正脉冲的宽度为 HRPWM_SYNOCR.SYNCCMP[7:0]个 PCLK0，跟随宽度为 HRPWM_SYNOCR.SYNCCMP[7:0]个 PCLK0 的无效低电平。

0b11：在同步输出引脚 HRPWM_SCOUT 上产生负脉冲。负脉冲的宽度为 HRPWM_SYNOCR.SYNCCMP[7:0]个 PCLK0，跟随宽度为 HRPWM_SYNOCR.SYNCCMP[7:0]个 PCLK0 的无效高电平。

在整个同步信号产生期间，（ HRPWM_SYNOCR.SYNCCMP[7:0] 个 PCLK0 的脉冲宽度 + HRPWM_SYNOCR.SYNCCMP[7:0]个 PCLK0 的无效电平），任何同步请求将被忽略。

19.3.16 缓存功能

对于单元 1~单元 6，HRPWM 高分辨率通用周期基准值、高分辨率通用比较基准值、专用比较基准值、高分辨率死区时间基准值、外部事件滤波偏移值、外部事件滤波窗口、中断控制寄存器、端口控制寄存器等都具有缓存功能，可在计数期间硬件实现周期变化、占空比变化、死区变化等。另外间隔输出周期基准值和间隔输出比较基准值同样具有缓存功能。

与其它单元不同的是单元 1 具有高分辨率相位基准值寄存器，同样具有缓存功能。

所有具有缓存功能的寄存器列表以及其对应的缓存传送使能位如表 19-10 所示。具有缓存功能的寄存器在相应的缓存使能位有效且缓存传送点发生时，缓存寄存器的值被传送到对应的实体寄存器中。需要注意的是，当支持缓存的寄存器，开启了缓存功能时，实体寄存器依然支持寄存器写，请勿在使能对使能了缓存功能的实体寄存器进行写，以免造成不可预期的波形。

表 19-10 支持缓存的寄存器

计数单元	支持缓存的实体寄存器	缓存寄存器	缓存使能位
单元 1	HRPWM1_HRPERAR	HRPWM1_HRPERBR	HRPWM1_BCONR1.BENP
	HRPWM1_HRGCMAR	HRPWM1_HRGCMCR	HRPWM1_BCONR1.BENAE
	HRPWM1_HRGCMBR	HRPWM1_HRGCMDR	HRPWM1_BCONR1.BENBF
	HRPWM1_HRGCMER	HRPWM1_HRGCMGR	HRPWM1_BCONR1.BENAE
	HRPWM1_HRGCMFR	HRPWM1_HRGCMHR	HRPWM1_BCONR1.BENBF
	HRPWM1_HRDTUAR	HRPWM1_HRDTUBR	HRPWM1_DCONR1.DTBENU
	HRPWM1_HRDTDAR	HRPWM1_HRDTDBR	HRPWM1_DCONR1.DTBEND

计数单元	支持缓存的实体寄存器	缓存寄存器	缓存使能位
	HRPWM1_PHSCMP1A	HRPWM1_PHSCMP1B	HRPWM1_PHSCCTL.BENPHS
	HRPWM1_PHSCMP2A	HRPWM1_PHSCMP2B	
	HRPWM1_PHSCMP3A	HRPWM1_PHSCMP3B	
	HRPWM1_PHSCMP4A	HRPWM1_PHSCMP4B	
	HRPWM1_PHSCMP5A	HRPWM1_PHSCMP5B	
	HRPWM1_SCMAR	HRPWM1_SCOCR	HRPWM1_BCONR1.BENSPA
	HRPWM1_SCOMBR	HRPWM1_SCOMDR	HRPWM1_BCONR1.BENSPB
	HRPWM1_ICONR	HRPWM1_BICONR	HRPWM1_BCONR2.BENCTL
	HRPWM1_PCNR1~3	HRPWM1_BPCNR1~3	
	HRPWM1_PCNR1~3	HRPWM1_BPCNR1~3	
	HRPWM1_GCONR1	HRPWM1_BGCONR1	HRPWM1_BCONR2.BENEEFFOFF
	HRPWM1_EEFOFFSETAR	HRPWM1_EEFOFFSETBR	
	HRPWM1_EEFWINAR	HRPWM1_EEFWINBR	
单元 2~6 (t=2~6)	HRPWM<t>_HRPERAR	HRPWM<t>_HRPERBR	HRPWM<t>_BCONR1.BENP
	HRPWM<t>_HRGCMAR	HRPWM<t>_HRGCMCR	HRPWM<t>_BCONR1.BENAE
	HRPWM<t>_HRGCOMBR	HRPWM<t>_HRGCOMDR	HRPWM<t>_BCONR1.BENBF
	HRPWM<t>_HRGCMER	HRPWM<t>_HRGCMGR	HRPWM<t>_BCONR1.BENAE
	HRPWM<t>_HRGCMFR	HRPWM<t>_HRGCMHR	HRPWM<t>_BCONR1.BENBF
	HRPWM<t>_HRDTUAR	HRPWM<t>_HRDTUBR	HRPWM<t>_DCONR1.DTBENU
	HRPWM<t>_HRDTDAR	HRPWM<t>_HRDTDABR	HRPWM<t>_DCONR1.DTBEND
	HRPWM<t>_SCMAR	HRPWM<t>_SCOCR	HRPWM<t>_BCONR1.BENSPA
	HRPWM<t>_SCOMBR	HRPWM<t>_SCOMDR	HRPWM1<t>_BCONR1.BENSPB
	HRPWM<t>_ICONR	HRPWM<t>_BICONR	HRPWM<t>_BCONR2.BENCTL
	HRPWM<t>_PCNR1~3	HRPWM<t>_BPCNR1~3	
	HRPWM<t>_PCNR1~3	HRPWM<t>_BPCNR1~3	
	HRPWM<t>_BGCONR1	HRPWM<t>_BGCONR1	HRPWM<t>_BCONR2.BENEEFFOFF
HRPWM<t>_EEFOFFSETAR	HRPWM<t>_EEFOFFSETBR		
HRPWM<t>_EEFWINAR	HRPWM<t>_EEFWINBR		
其他	HRPWM_BMPERAR	HRPWM_BMPERBR	HRPWM_BMCR.BENBMP
	HRPWM_BMCMAR	HRPWM_BMCOMBR	HRPWM_BMCR.BENBMCMP

19.3.16.1 缓存传送节点

单元内缓存传送节点

在三角波和锯齿波模式下，所有具有缓存功能的寄存器均可设置在在缓存功能使能时在不同的缓存传送点发生缓存传送，各单元支持的缓存的寄存器以及缓存传送点如表 19-11 所示。

表 19-11 各单元支持的缓存传送点

计数单元	支持缓存的寄存器	缓存传送点
单元 1	HRPWM1_HRPERAR	零点缓存传送点 周期缓存传送点 软件触发（计数器停止时） 单次缓存传送点（单元 1 单次缓存传送使能有效，其他缓存点失效）
	HRPWM1_HRGCMAR	
	HRPWM1_HRGCMBR	
	HRPWM1_HRGCMER	
	HRPWM1_HRGCMFR	
	HRPWM1_HRDTUAR	
	HRPWM1_HRDTDAR	
	HRPWM1_PHSCMP1A	
	HRPWM1_PHSCMP2A	
	HRPWM1_PHSCMP3A	
	HRPWM1_PHSCMP4A	
	HRPWM1_PHSCMP5A	
	HRPWM1_SCMAR	
	HRPWM1_SCMBR	
	HRPWM1_ICONR	
	HRPWM1_PCNAR1~3	
	HRPWM1_PCNBR1~3	
	HRPWM1_GCONR1	
	HRPWM1_EEFOFFSETAR	
HRPWM1_EEFWINAR		
单元 2~6 (t=2~6)	HRPWM<t>_HRPERAR	零点缓存传送点 周期缓存传送点 软件触发（计数器停止时） 单元 1 单次缓存传送点 单元 1 单次缓存传送点后的本单元锯齿波周期缓存传送点或者三角波零点缓存传送点（仅在单元 1 的单次缓存传送点使能时有效）
	HRPWM<t>_HRGCMAR	
	HRPWM<t>_HRGCMBR	
	HRPWM<t>_HRGCMER	
	HRPWM<t>_HRGCMFR	
	HRPWM<t>_HRDTUAR	
	HRPWM<t>_HRDTDAR	
	HRPWM<t>_SCMAR	
	HRPWM<t>_SCMBR	
	HRPWM<t>_ICONR	
	HRPWM<t>_PCNAR1~3	
	HRPWM<t>_PCNBR1~3	
	HRPWM<t>_BGCONR1	
	HRPWM<t>_EEFOFFSETAR	
	HRPWM<t>_EEFWINAR	
其他	HRPWM_BMPERAR	BM-counter 的上溢点
	HRPWM_BMCMAR	

注：

上表中,在三角波模式下,零点缓存传送点为向下计数计数值等于 64 的时刻或者硬件清零发生的时刻,周期缓存传送点为向上计数计数值等于周期值-64 的时刻;在锯齿波模式下,零点缓存传送点为计数值等于 0 的时刻,周期缓存传送点为计数值等于周期值或者硬件清零发生的时刻。

单元 1 在计数器计数时可将缓存传送点配置为零点缓存传送点或者周期缓存传送点,通过配置 HRPWM1_BCONR1、HRPWM1_BCONR2、HRPWM1_DCONR、HRPWM1_PHSCTL 使能对应缓存寄存器不同的缓存传送节点。在计数停止时,通过对 HRPWM_SBUFR.SBU1TRG 写 1 可以对使能了缓存功能的寄存器触发一次软件缓存。

单元 1 除上述缓存点外,在计数器计数过程中还具有一个单次缓存传送点,通过对 HRPWM_GBCONR 的 OSTENU1 位设置为 1 可以使能单元 1 的单次缓存传送点。

单次缓存传送点指:在三角波模式下,对 OSTBTRU1 位写 1 后,在其后遇到的第一个零点缓存传送点;在锯齿波模式下,对 OSTBTRU1 位写 1 后,在其后遇到的第一个周期缓存传送点。单次缓存点使能后(HRPWM_GBCONR.OSTENU1=1),HRPWM1_BCONR1 和 HRPWM1_BCONR2 中设定的其他缓存传送点无效。

单元 2~6 在计数器计数时可将缓存传送点配置为零点或者周期点缓存传送点,通过配置 HRPWM<t>_BCONR1、HRPWM<t>_BCONR2、HRPWM<t>_DCONR、HRPWM<t>_PHSCTL 使能对应寄存器不同的缓存传送节点。在计数停止时,通过对 SBUFR.SBU<t>TRG 写 1 可以对使能了缓存功能的寄存器触发一次软件缓存。单元 2~6 除上述缓存点外,在计数器计数过程中还可在单元 1 单次缓存传送点发生缓存,通过配置 HRPWM<t>_BCONR2、HRPWM<t>_PHSCTL 中的 BTRU0xx 位使能。通过配置 HRPWM<t>_BCONR2、HRPWM<t>_DCONR、HRPWM<t>_PHSCTL 中的 BTRU0Pxx 位,还可将缓存点配置为单元 1 单次缓存传送点后的本单元锯齿波周期缓存传送点或者三角波零点缓存传送点,该缓存点仅在单元 1 的单次缓存传送点使能时(HRPWM_GBCONR.OSTENU1=1)有效。

注：

在三角波模式下将周期值设置在周期缓存传送点时会出现以下异常情况,当周期值由小变大时,会出现丢失周期事件的情况,周期由大变小时,计数器立即更新为新的周期值开始向下计数。在三角波模式下避免在周期匹配传送点缓存周期值。图 19-36 为三角波周期缓存传送点缓存周期值的异常情况图例。

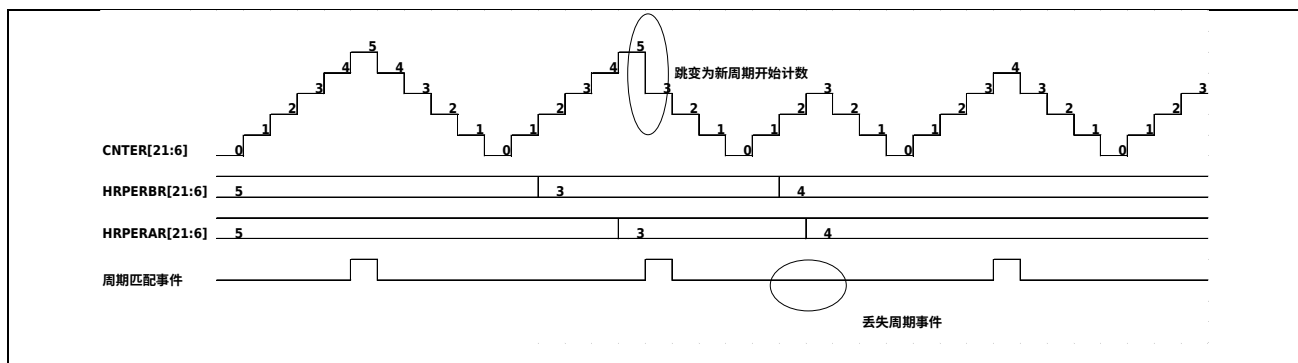


图 19-36 三角波周期缓存传送点缓存周期值异常情况

注:

1. 三角波模式下, 停止状态下将计数器的值设置为 0 时, 如果将缓存传送点设置为零点缓存传送点, 在计数启动后的第一个计数值等于 0 的时刻不发生缓存, 使用时需要注意。如果需要在计数器启动前缓存新的基准值, 可以在计数器停止状态下, 对 `HRPWM_SBUF.R.SBU<t>TRG` 写 1 触发一次软件缓存。
2. 三角波模式下, 相位值设置在周期缓存传送点发生缓存时, 向下计数区间的相位值为更新后的相位值, 而缓存之前的向上计数区间的相位值为更新前的相位值, 三角波模式下避免在周期缓存传送点缓存相位值。

图 19-37 为锯齿波周期缓存传送点缓存动作。

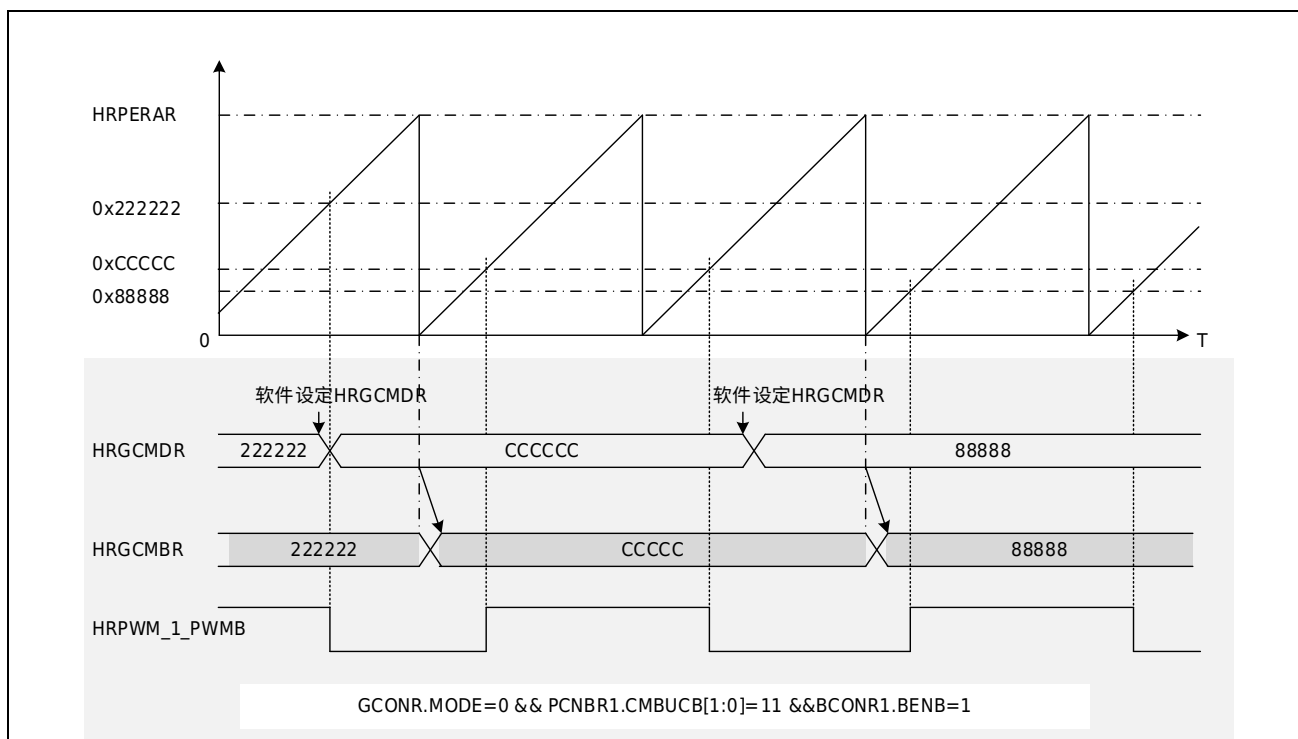


图 19-37 锯齿波模式周期缓存传送点缓存动作

图 19-38 所示, 是三角波在零点缓存传送点的缓存动作, 图 19-39 所示, 是三角波零点和周期缓存传送点均发生缓存动作的示意图。

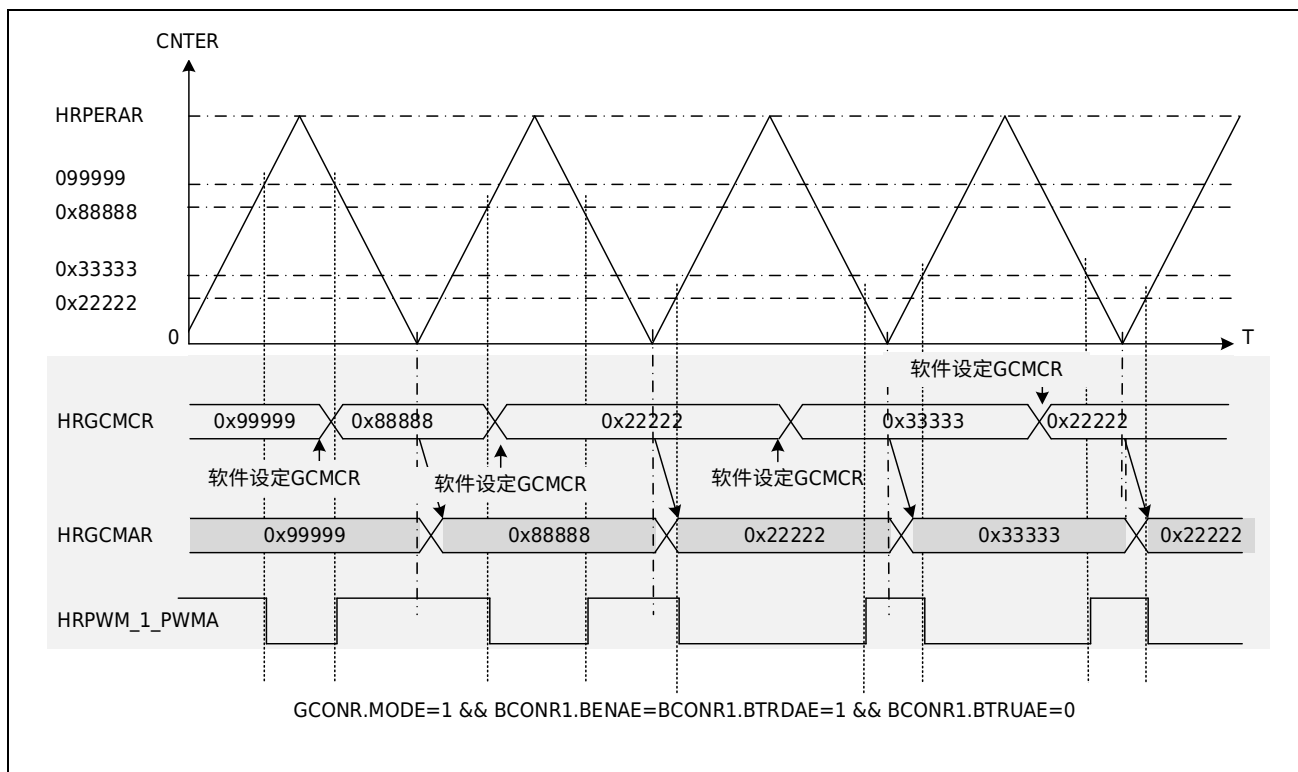


图 19-38 三角波模式时计数缓存动作 1

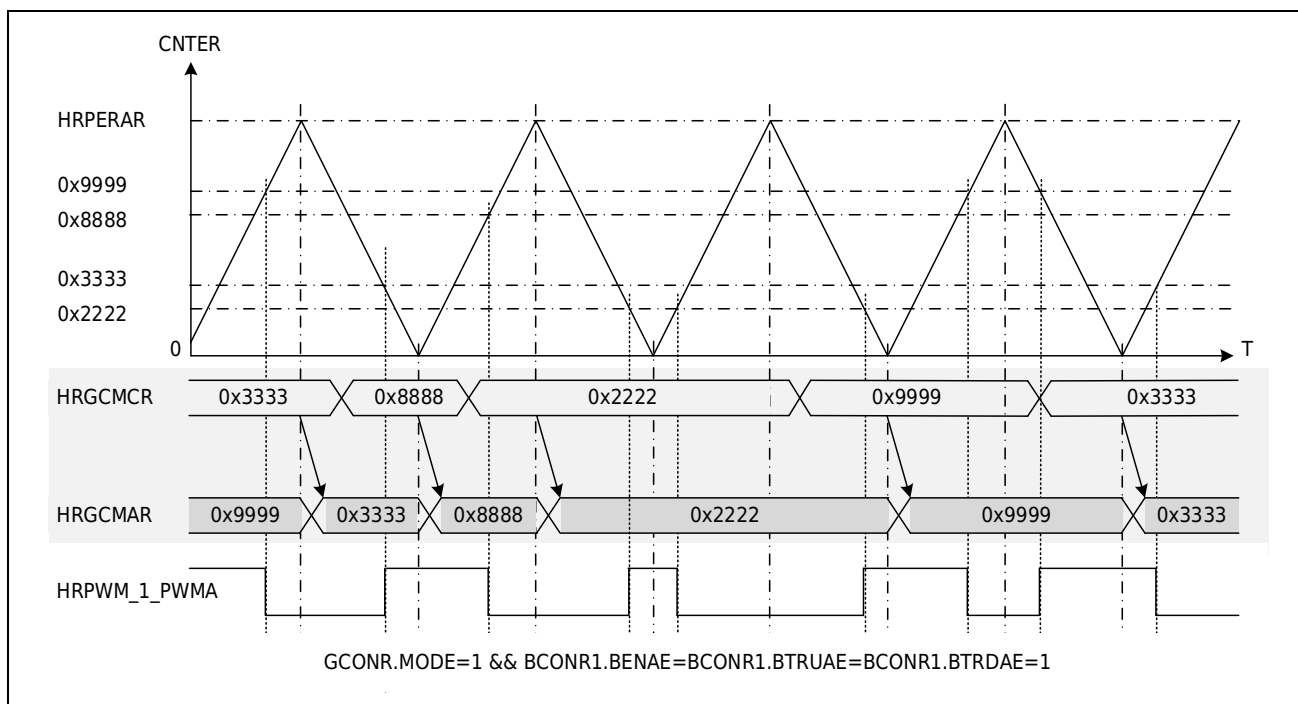


图 19-39 三角波模式时计数缓存动作 2

间隔输出缓存传送节点

间隔输出周期基准值和间隔输出比较基准值的缓存传送发生在 BM-counter 的上溢点，如图 19-40 为间隔输出缓存动作。

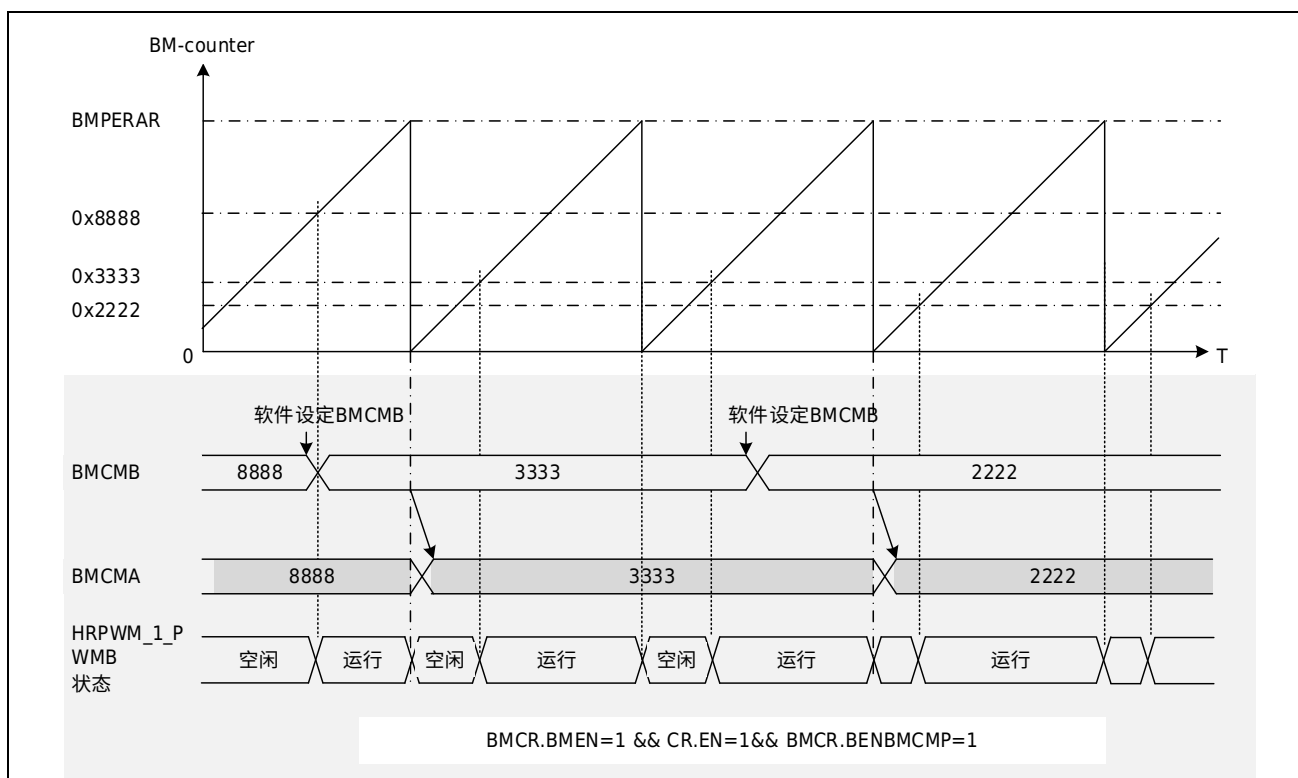


图 19-40 间隔输出缓存动作

注：

如果需要保证周期值和比较值同时缓存，请先写周期值再写比较值；如果只修改了周期值，写完周期值需要再写一次比较值；如果只修改比较值，写入比较值即可。

19.3.16.2 单元内全局缓存屏蔽功能

对 HRPWM_GBCONR 的对应单元控制位 BTRU<t>EN 设置为 0，可以同时表 19-10 中各个单元的缓存使能位屏蔽为 0。当对应单元控制位为 1 时，表 19-10 各个单元的缓存使能恢复各自寄存器设定的状态。

19.3.16.3 单元全局缓存完成标志

全局缓存状态标志寄存器位 HRPWM_GBSFLR.U<t>BTRENF 可以在不同条件下在各单元对应缓存传送点到来时被置高。单元 HRPWM_GBSFLR.U<t>BTRENF 置位时刻如下表，各单元 HRPWM_GBSFLR.U<t>BTRENF 在下述的置位时刻发生，并且对应的 HRPWM_GBCONR.BTRUxEN 为 1 时，发生置位。

表 19-12 各单元全局缓存状态标志置位条件

单元	计数器状态	HRPWM_GBSFLR.U<t>BTRENF 置位时刻	置位条件使能位
单元 1	停止	软件缓存触发，对 HRPWM_GBCONR.BTRU1SFT 写 1	-
	计数中	周期缓存传送点	HRPWM1_GCONR1.BFSEL[0]
		零点缓存传送点	HRPWM1_GCONR1.BFSEL[1]
		单次缓存传送点	HRPWM1_GCONR1.BFSEL[2]

单元	计数器状态	HRPWM_GBSFLR. U<t>BTRENF 置位时刻	置位条件使能位
单元<t> (t=2-6)	停止	软件触发, 对 HRPWM_GBCONR.BTRU<t>EN 写 1	-
	计数中	周期缓存传送点	HRPWM1_GCONR1.BFSEL[0]
		零点缓存传送点	HRPWM1_GCONR1.BFSEL[1]
		单元 1 单次缓存传送点	HRPWM1_GCONR1.BFSEL[2]
	单元 1 单次缓存传送点后的本单元锯齿波周期缓存传送点三角波零点缓存传送点	HRPWM1_GCONR1.BFSEL[3]	

请按照下述流程清除全局缓存完成状态标志位:

- 1) 读取到全局缓存完成状态标志位 1 后, 将对应单元的 HRPWM_GBCONR. BTRU<t>EN 设置为 0;
- 2) 对 HRPWM_GBSFLR. U<t>BTRENF 写 0 清除全局缓存完成状态标志。

19.3.17 通用 PWM 输出

HRPWM 输出 PWM 波形的分辨率为 $1/(fPCLK0*64)$, 主要支持以下几种 PWM 输出波形。

19.3.17.1 单边对齐独立 PWM 输出

在锯齿波计数模式 (HRPWM<t>_GCONR.MODE=0), 每个单元的 2 个端口 HRPWM_<t>_PWMA、HRPWM_<t>_PWMB 都能独立的输出 PWM 波。在计数值等于周期值若设定同样的电平变化, 则可实现单边对齐独立高分辨率 PWM 输出。如下图 19-41 所示。

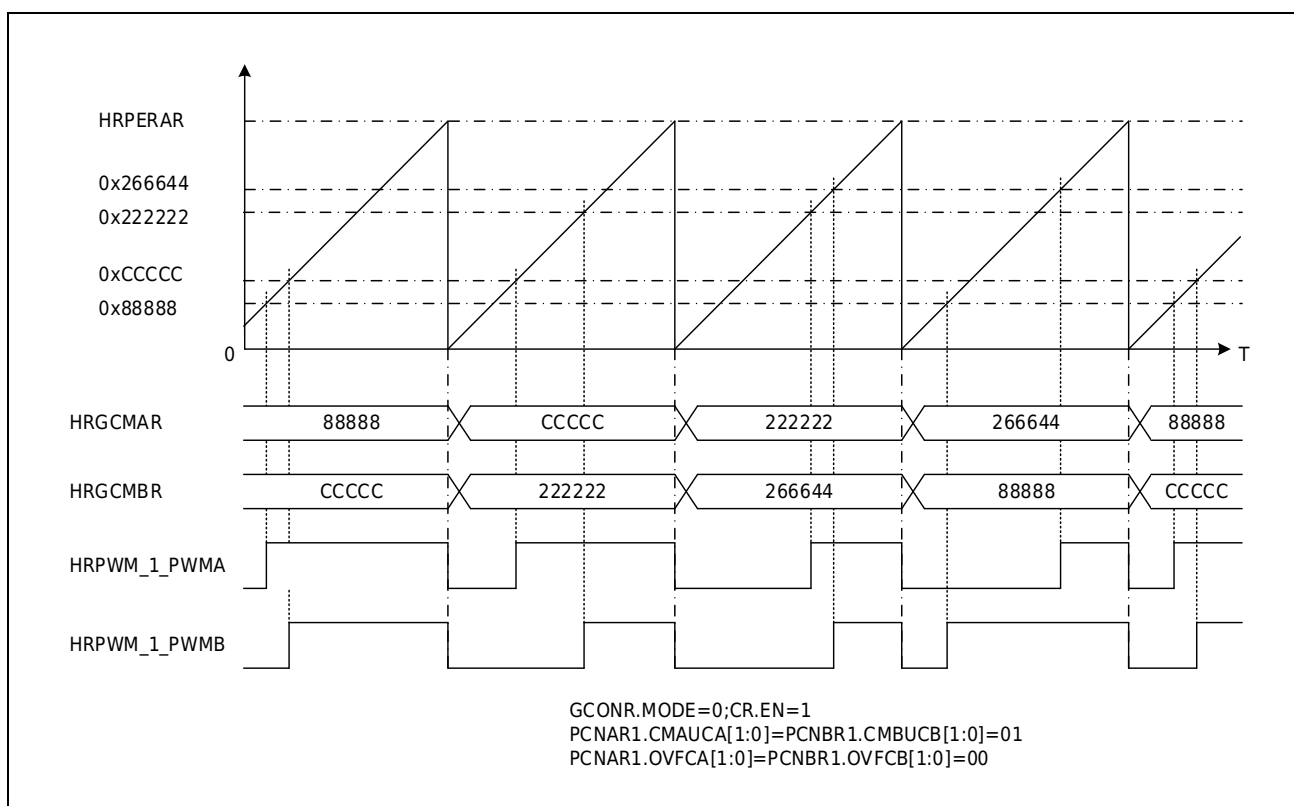


图 19-41 单边对齐独立高分辨率 PWM

19.3.17.2 双边对称独立 PWM 输出

在三角波计数模式 (HRPWM<t>_GCONR.MODE=1) 时, 每个单元的 2 个端口 HRPWM_<t>_PWMA、HRPWM_<t>_PWMB 都能独立的输出 PWM 波。在计数比较匹配若设定电平变化、在计数值等于 0 点或者计数值等于周期值设定电平不变化, 则可实现双边对称独立 PWM 输出。如图 19-42 所示, 是 HRPWM_<t>_PWMA 端口和 HRPWM_<t>_PWMB 端口实现双边对称独立输出 PWM。

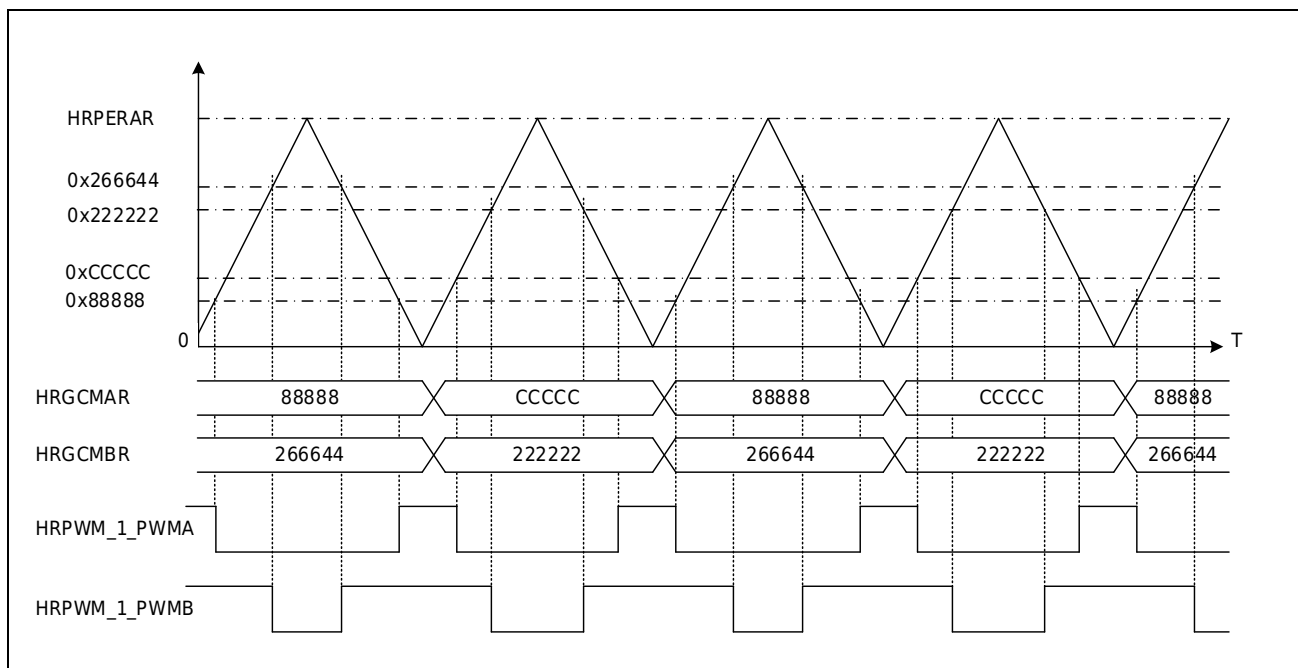


图 19-42 双边对称独立 PWM

19.3.17.3 互补 PWM 输出

软件设定实现互补 PWM 输出

在死区不使能 (HRPWM<t>_DCONR.DTCEN=0) 时, 设定好端口在计数启动、比较匹配、计数值等于周期值、计数值等于 0 发生时的电平状态, 可实现 HRPWM_<t>_PWMA 和 HRPWM_<t>_PWMB 端口上输出一对互补 PWM 波形。

图 19-43 为双边对称互补 PWM 波的输出例。

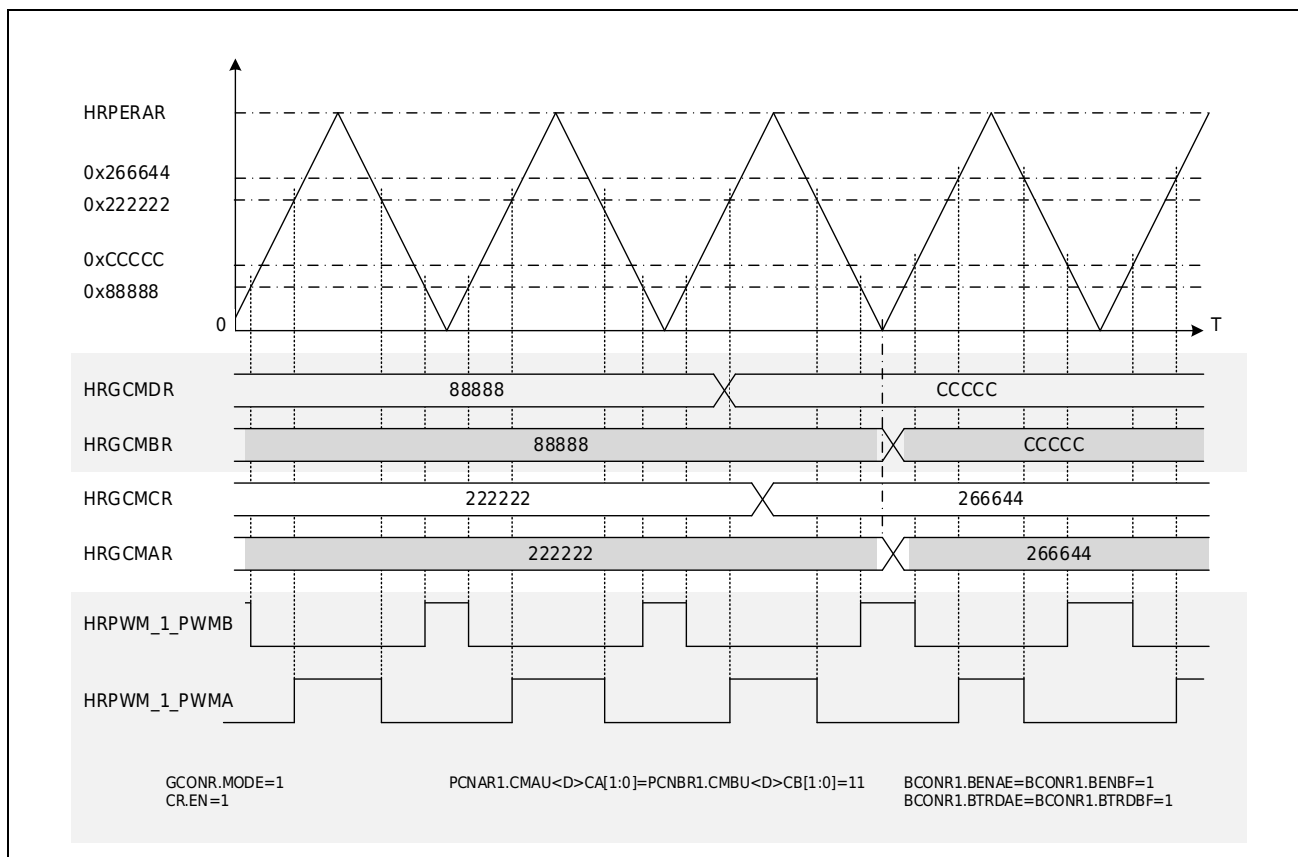


图 19-43 软件实现互补 PWM 波输出

硬件设定实现互补 PWM 输出

在死区使能 (HRPWM<t>_DCONR.DTCEN=1) 时, 设定好端口在计数启动、比较匹配、计数值等于周期值和计数值等于 0 发生时的电平状态, 产生通道 A 内部输出信号 (HRPWM_<t>_PWMA_ORG), 并和高精度死区时间基准值寄存器 (HRPWM<t>_DTU<D>AR) 的设定值通过时序偏移, 以硬件的方式实现在 HRPWM_<t>_PWMA 和 HRPWM_<t>_PWMB 端口上输出一对互补 PWM 波形, 具体参考【死区模块】说明。

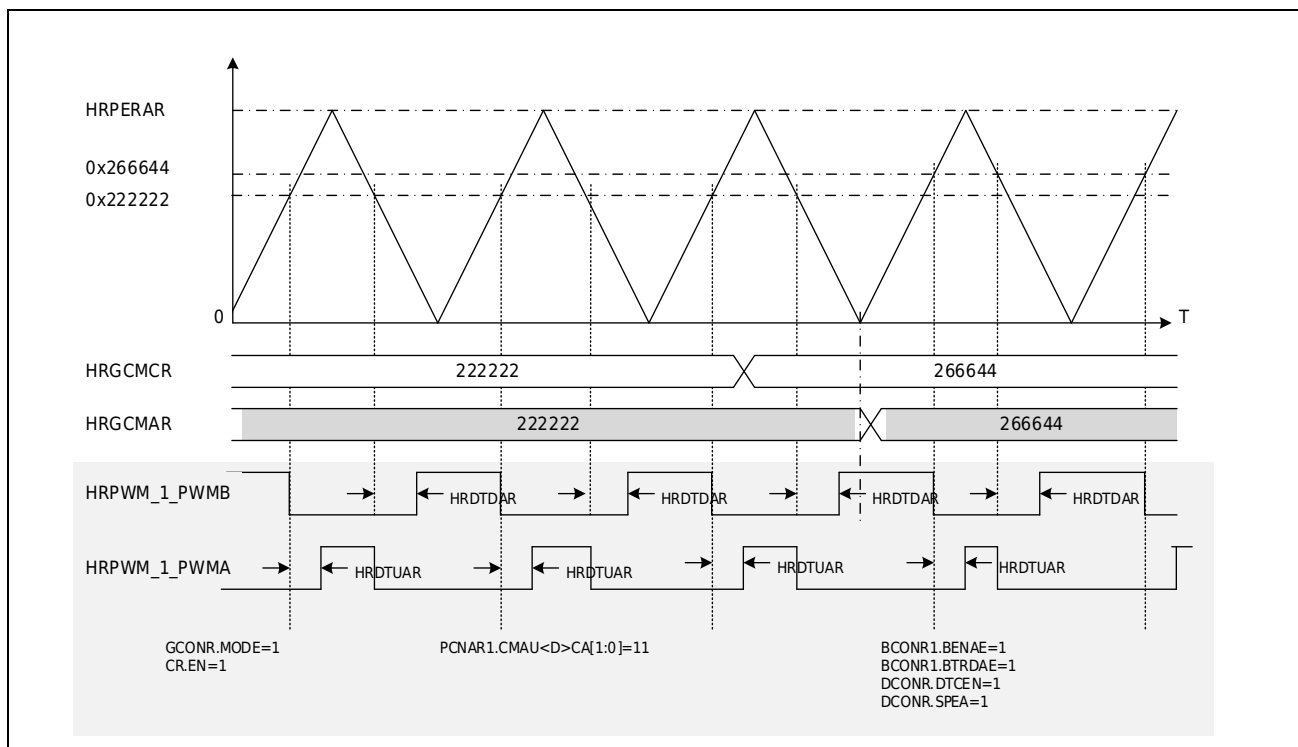


图 19-44 硬件实现互补 PWM 波输出

19.3.17.4 双边非对称 PWM 输出

在三角波计数模式 (HRPWM<t>_GCONR.MODE=1) 时, 每个单元的 2 个端口 HRPWM_<t>_PWMA、HRPWM_<t>_PWMB 都能独立的输出高分辨率 PWM 波, 且每个端口都能根据高分辨率通用比较基准值寄存器 (HRPWM<t>_HRGCMAR、HRPWM<t>_HRGCMBR) 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, HRPWM_<t>_PWMA 的端口电平变化分别由 HRPWM<t>_HRGCMAR、HRPWM<t>_HRGCMBR 基准值的比较结果控制, 则在 HRPWM_<t>_PWMA 端口上可实现双边非对称高分辨率 PWM 输出。如图 19-45 所示, 是 HRPWM_<t>_PWMA 端口的双边非对称高分辨率 PWM 输出。

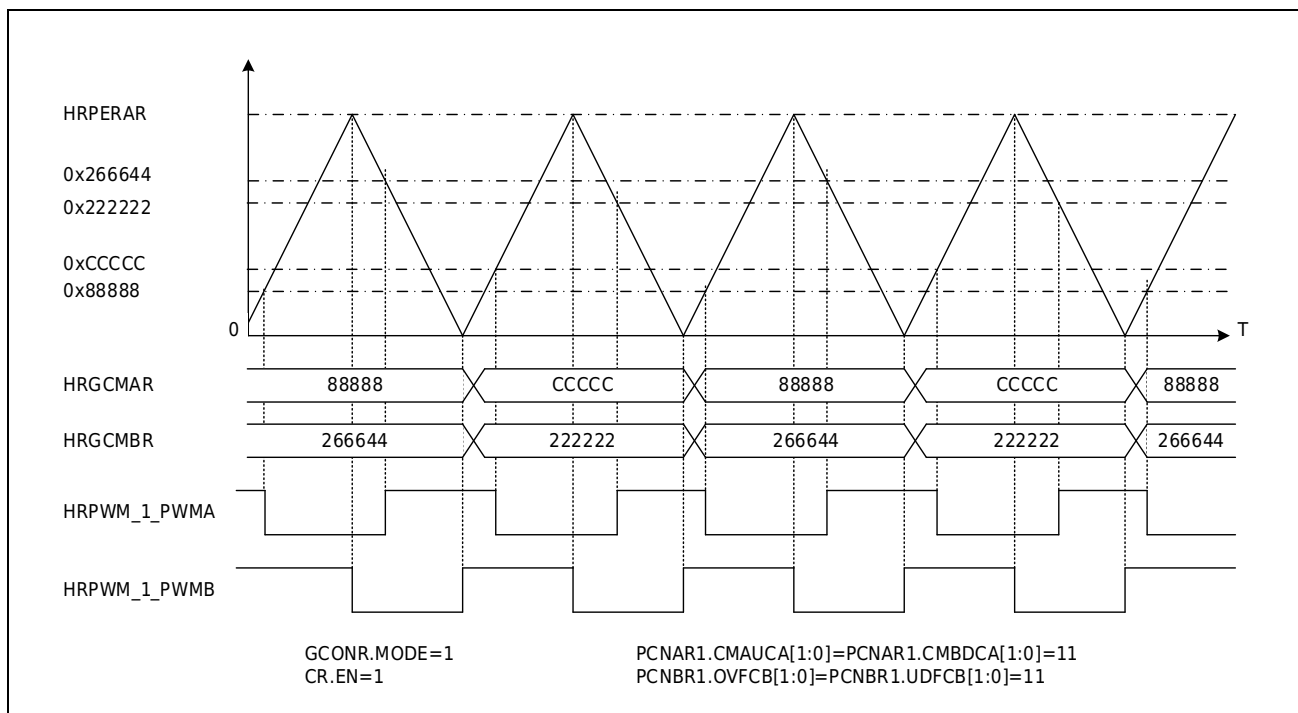


图 19-45 双边非对称 PWM 输出

19.3.17.5 单元间多相 PWM 输出

每个单元的 HRPWM_<t>_PWMA、HRPWM_<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高分辨率 PWM 波输出。如图 19-46,单元 1、单元 2、单元 3 组合输出 6 相单边对齐独立高分辨率 PWM。

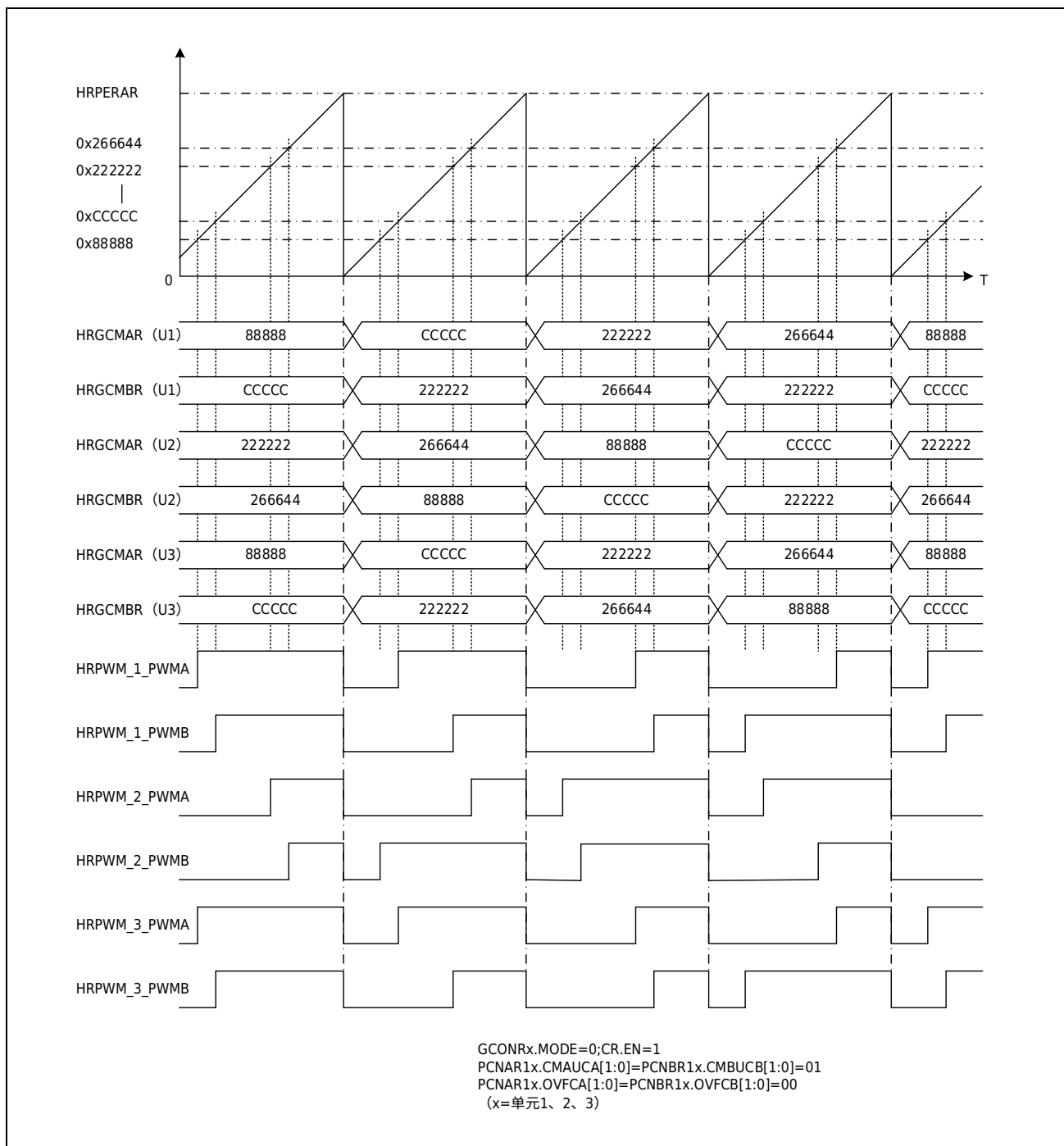


图 19-46 6相单边对齐独立 PWM

19.3.17.6 单元间多相交错 PWM 输出

多相交错 PWM 输出

每个单元的 HRPWM_<t>_PWMA、HRPWM_<t>_PWMB 端口都能输出 PWM，利用单元 1 相位基准值寄存器和单元 2~6 的相位功能，可以实现最多 6 个单元多相交错 PWM 输出。如图 19-47，单元 1、单元 2、单元 3 组合输出三相交错互补高分辨率 PWM。

该例中，单元 1 作为主单元分别在对应的相位比较点生成相位比较匹配事件，对于本例中计数方式为三角波模式，周期基准值为 4500，且需要实现三相，利用如下公式计算单元 2 和单元 3 的高分辨率相位比较基准值。

- $PHSCMP_x = (M-1) * HRPERAR * 2 / N - 64$ (三角波模式)
- $PHSCMP_{xA} = (M-1) * (HRPERAR + 1) / N - 64$ (锯齿波模式)
- $N = \text{相数}$
- $M = \text{PWM 模块编号}$

经过计算，单元 2 的相位匹配值设定为 2936，单元 3 的相位匹配值设定为 5936。

按照图示将各个单元寄存器设定完成后，启动单元 1，当单元 1 计数值到达对应的相位匹配值后相应的从单元启动计数。

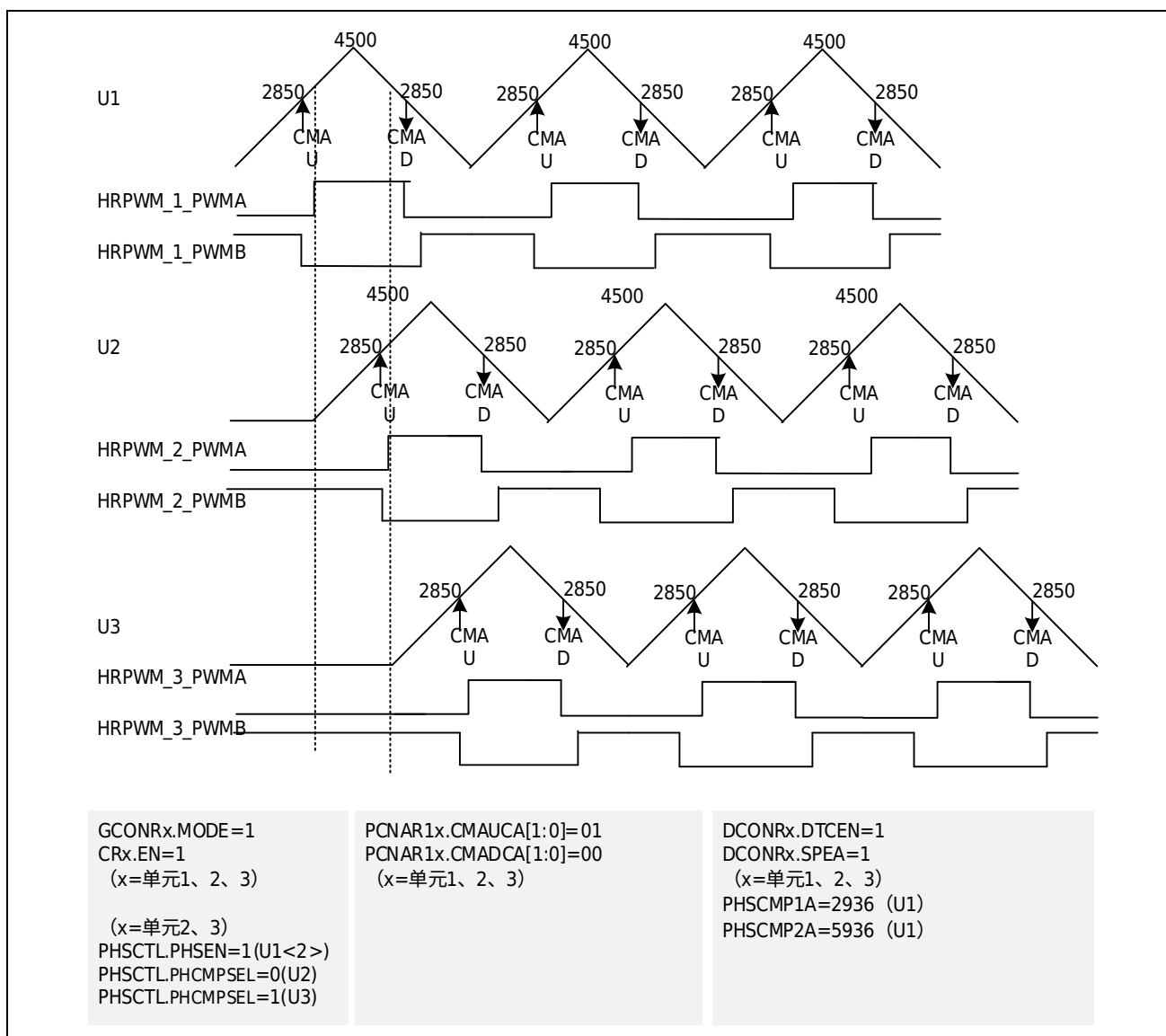


图 19-47 三相交错互补高分辨率 PWM

多相交错 PWM 动态调整

在多相 PWM 输出期间，结合各基准值寄存器的缓存功能可以硬件实现周期、占空比和死区等的调整。

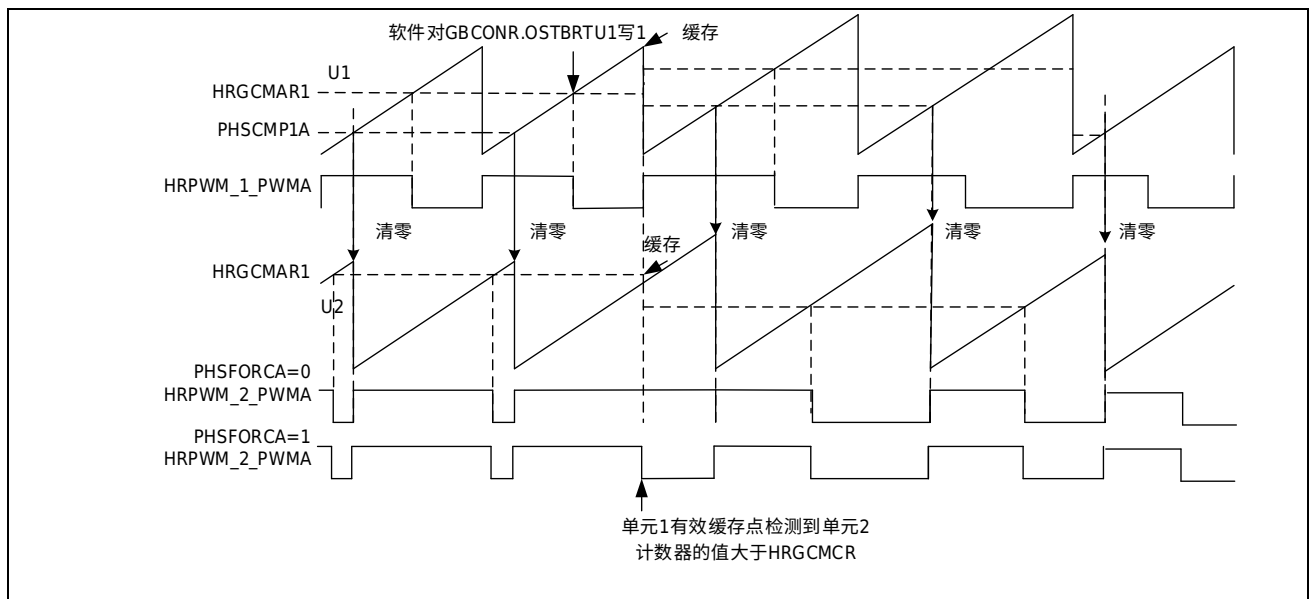


图 19-48 锯齿波模式多相交错 PWM 变频输出

图 19-48 为锯齿波模式多相交错 PWM 变频输出示例图，上图具体配置如下：

1. 主单元：

- 1) PWM 输出配置：在计数值等于 0 时置高，在计数值等于 HRGCMAR 置低；
- 2) 缓存配置：配置缓存控制寄存器 HRPWM1_BCONR1.BENP<AE><BF>=1 等分别使能高分辨率周期基准值寄存器、高分辨率比较基准值等寄存器的缓存功能；配置相位控制寄存器 HRPWM1_PHSCTL.BENPHS=1 使能单元 1 高分辨率相位比较基准值寄存器的缓存功能；
- 3) 配置全局缓存控制寄存器 HRPWM_GBCONR.OSTENU1 使能单元 1 单次缓存传送点；
- 4) 其他：配置比较值和周期值以及相位匹配值。

2. 从单元：

- 1) PWM 输出配置：在计数值等于 0 时置高，在计数值等于 HRGCMAR 置低；
- 2) 相位功能配置：配置相位控制寄存器 HRPWM<t>_PHSCTL.PHSEN=1 使能相位功能，HRPWM<t>_PHSCTL.PHCMPSEL=1 选择 PHSCMP1A 作为相位匹配点，HRPWM<t>_PHSCTL.PHSFORCA=1 使能在单元 1 单次缓存传送点检测到当前计数值大于等于比较值寄存器 HRPWM<t>_HRGCMCR 设定的值时，将输出 A 强制设置为低功能；
- 3) 缓存配置：配置缓存控制寄存器 HRPWM<t>_BCONR2.BENAE=1 分别使能高分辨率周期基准值寄存器，高分辨率比较基准值寄存器的缓存功能；配置缓存控制寄存器 HRPWM<t>_BCONR2.BTRU0PAE<BF>=1 将上述寄存器的缓存点设置为单元 1 单次缓存传送点；
- 4) 其他：配置比较值以及相位匹配值，比较值设置为最大周期值。

在进行动态调整时，通过读取 HRPWM_GBSFLR.U1BTRENF 确认各个单元的基准值已经缓存完成，清除标志位后，再将新的周期值和比较值写入缓存寄存器，对 OSTBTRU1 写 1 触发单元 1 单次缓存。

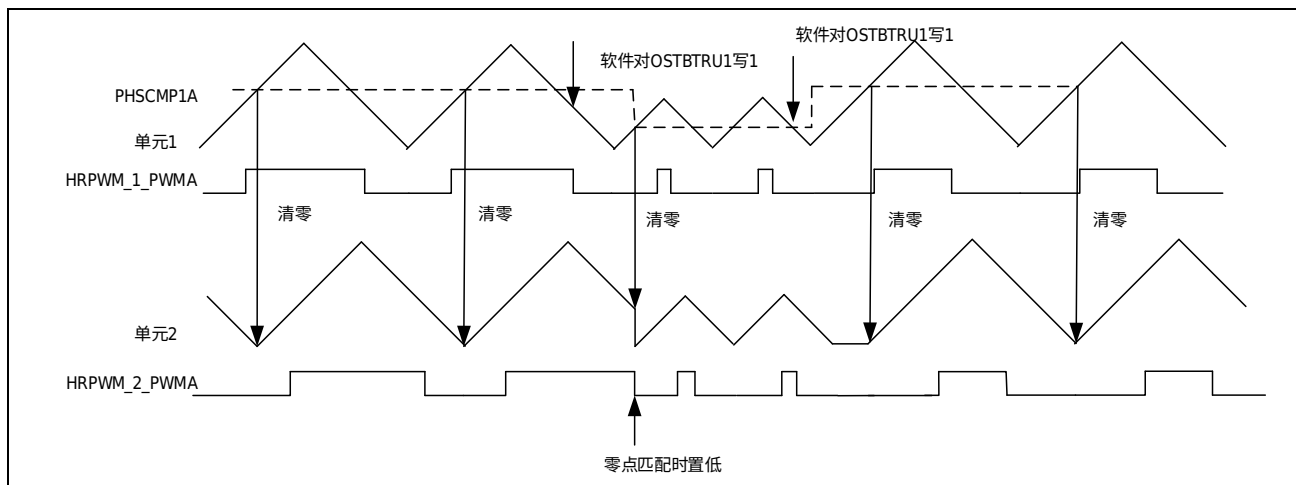


图 19-49 三角波模式多相交错 PWM 变频输出

图 19-49 为三角波模式多相交错 PWM 变频输出示例图，上图具体配置如下：

1. 主单元：

- 1) PWM 输出配置：在向上计数计数值等于 HRGCMAR 置高，在向下计数计数值等于 HRGCMAR 置低，在计数值等于 0 时置低，计数器停止置低；
- 2) 缓存配置：配置缓存控制寄存器 HRPWM1_BCONR1.BENP<AE>=1 等分别使能高分辨率周期基准值寄存器、高分辨率比较基准值等寄存器的缓存功能；配置相位控制寄存器 HRPWM1_PHSCTL.BENPHS=1 使能单元 1 高分辨率相位比较基准值寄存器的缓存功能；
- 3) 配置全局缓存控制寄存器 HRPWM_GBCONR. OSTENU1 使能单元 1 单次缓存传送点；
- 4) 其他：配置比较值和周期值以及相位匹配值。

2. 从单元

- 1) PWM 输出配置：在向上计数计数值等于 HRGCMAR 置高，在向下计数计数值等于 HRGCMAR 置低，在计数值等于 0 时置低，计数器停止置低；
- 2) 相位功能配置：配置相位控制寄存器 HRPWM<t>_PHSCTL.PHSEN=1 使能相位功能，HRPWM<t>_PHSCTL.PHCMPSEL =1 选择 PHSCMP1A 作为相位匹配点；
- 3) 缓存配置：配置缓存控制寄存器 HRPWM<t>_BCONR2.BENP<AE><BF>=1 分别使能高分辨率周期基准值寄存器，高分辨率比较基准值寄存器的缓存功能；配置缓存控制寄存器 HRPWM<t>_BCONR2. BTRU0PP<AE> =1 将上述寄存器的缓存点设置为单元 1 单次缓存点之后的锯齿波周期缓存传送点或者三角波零点缓存传送点；
- 4) 其他：配置比较值、以及周期值，配置 HRPWM<t>_GCONR.OVSTP 为 1 使能单次计数功能。

在进行动态调整时，通过读取 HRPWM_GBSFLR.U<t>BTRENF 确认各个单元的基准值已缓存完成，清除标志位后，在将新的周期值和比较值写入缓存寄存器，对 HRPWM_GBCONR.OSTBTRU1 写 1 触发单元 1 单次缓存。

19.3.18 周期间隔响应

HRPWM 的 2 个专用比较基准值寄存器 (HRPWM<t>_SCMAR、HRPWM<t>_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 (HRPWM<t>_VPERR) 的 HRPWM<t>_VPERR.PCNTE[1:0] 位和 HRPWM<t>_VPERR.SPPERIA/B 位使能。设定 HRPWM<t>_VPERR.PCNTS[4:0] 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 HRPWM<t>_SCMAR 或 HRPWM<t>_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/零点匹配中断和周期/零点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 HRPWM<t>_STFLR.VPERNUM=0 的周期）输出。图 18-24 所示是在周期间隔有效请求信号的动作例。

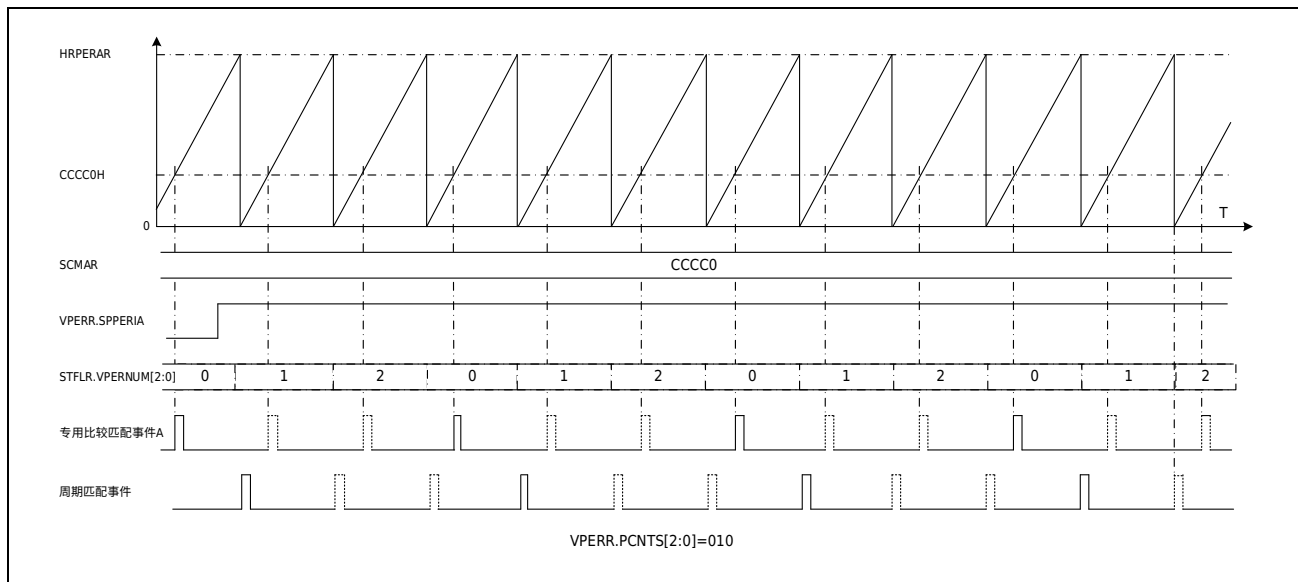


图 19-50 周期间隔有效请求信号动作

注：

当 HRPWM<t>_VPERR.PCNTE[4:0] 位不为 0，且 HRPWM<t>_VPERR.SPPERIB 位为 1，用作硬件清零、启动及各单元专用匹配事件 A 源的专用比较匹配事件 B 为经过周期间隔响应处理后的事件。当 HRPWM<t>_VPERR.PCNTE[4:0] 位不为 0，传送至周边模块以及用作单元专用匹配事件 A 源的周期/零点匹配事件为间隔响应处理后的事件。

19.3.19 EMB 控制

HRPWM 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 个端口输出控制接口，每个单元通过端口控制寄存器 (HRPWM<t>_PCNAR1.EMBSA) 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 HRPWM<t>_PCNAR1.EMBCA 的设定决定)。例如，若 HRPWM<t>_PCNAR1.EMBCA=01 设定时，则在 HRPWM_<t>_PWMA 端口正常输出期间，若产生 EMB 事件，则 HRPWM_<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 HRPWM 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 (HRPWM<t>_PCNAR1.EMBRA=00；这种方式称之为 One Shot 方式释放) 或在周期匹配或者零点匹配之后再恢复 PWM 的正常输出 (HRPWM<t>_PCNAR1.EMBRA=01、10、11；这种方式称之为 Cycle By Cycle 方式释放)。

注：

当 EMB 的事件释放方式配置为状态释放时 (通过 EMB_RLSSEL 寄存器配置)，且 EMB 的事件源配置外部端口或者内部电压比较器比较结果有效时，必须保证外部端口和内部电压比较器比较结果的电平宽度大于 3 个计数时钟周期，否则 PWM 输出上会出现最大 2 个计数时钟周期的毛刺。当 EMB 的事件释放方式配置为软件释放时 (通过 EMB_RLSSEL 寄存器配置)，且 EMB 的事件源配置 PWM 输出端口发生同相有效时，在发生 PWM 同相输出到 PWM 进入安全状态之前最大会出现 2 个计数时钟周期的毛刺。

19.3.20 输出管理

输出端口最后一级为极性选择以及交换输出选择，如图 19-51 所示。通过配置通用控制寄存器 1 (HRPWM<t>_GCONR1) 的 INVCAEN 位，可以将 HRPWM_<t>_PWMA 或 HRPWM_<t>_PWMB 端口电平极性取反。

配置 HRPWM<t>_GCONR1 的 SWAPEN 位，可以将 HRPWM_<t>_PWMA 和 HRPWM_<t>_PWMB 端口交换输出。交换支持两种模式，立即交换或在完整周期点 (由 HRPWM<t>_GCONR1.PRDSSEL 定义) 交换，通过 HRPWM<t>_GCONR1 的 SWAPMD 选择交换模式。

当 HRPWM<t>_GCONR1 的 SWAPMD 为 1 时，当检测到 HRPWM<t>_GCONR1 的 SWAPEN 位出现变化时，PWM 会在完整周期点 (由 HRPWM<t>_GCONR1.PRDSSEL 定义) 进入空闲周期，在下一个完整周期点交换输出或者正常输出，如图 19-53。空闲周期的电平由 HRPWM<t>_IDLECR.IDLESA设

定，如果由正常输出状态切换为交换输出状态，则空闲周期的电平由各自的 HRPWM<t>_IDLECR.IDLESA设定；如果由交换输出状态切换为正常输出状态，则空闲周期的电平由对方的 HRPWM<t>_IDLECR.IDLESA设定。

当 HRPWM<t>_GCONR1 的 SWAPMD 为 0 时, PWMA 和 PWMB 根据 HRPWM<t>_GCONR1 的 SWAPEN 位立即进行交换或者不交换，参照图 19-52。

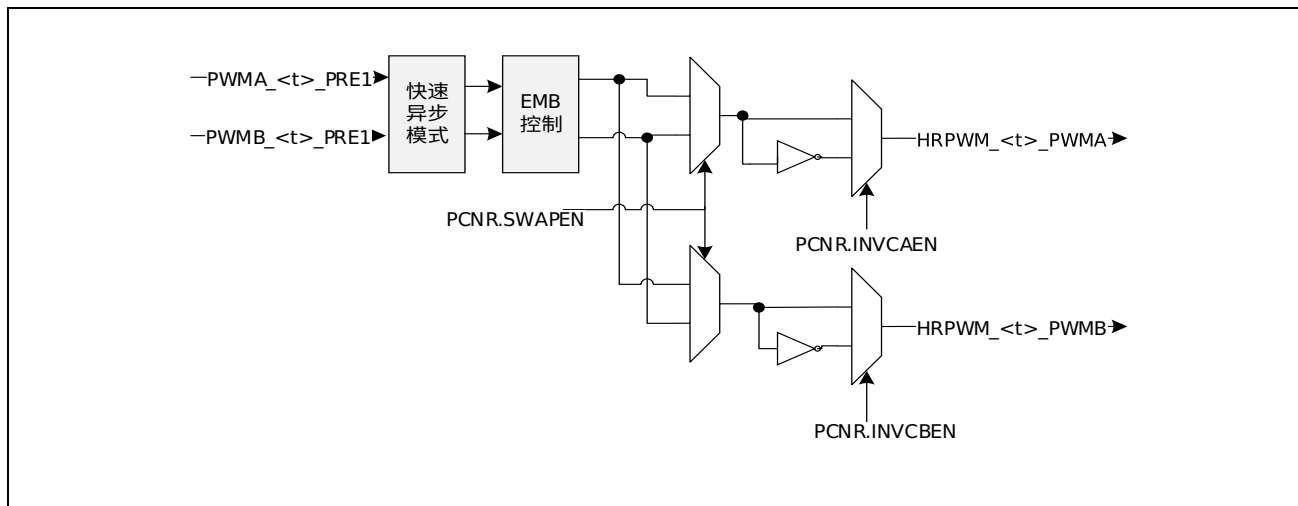


图 19-51 输出管理

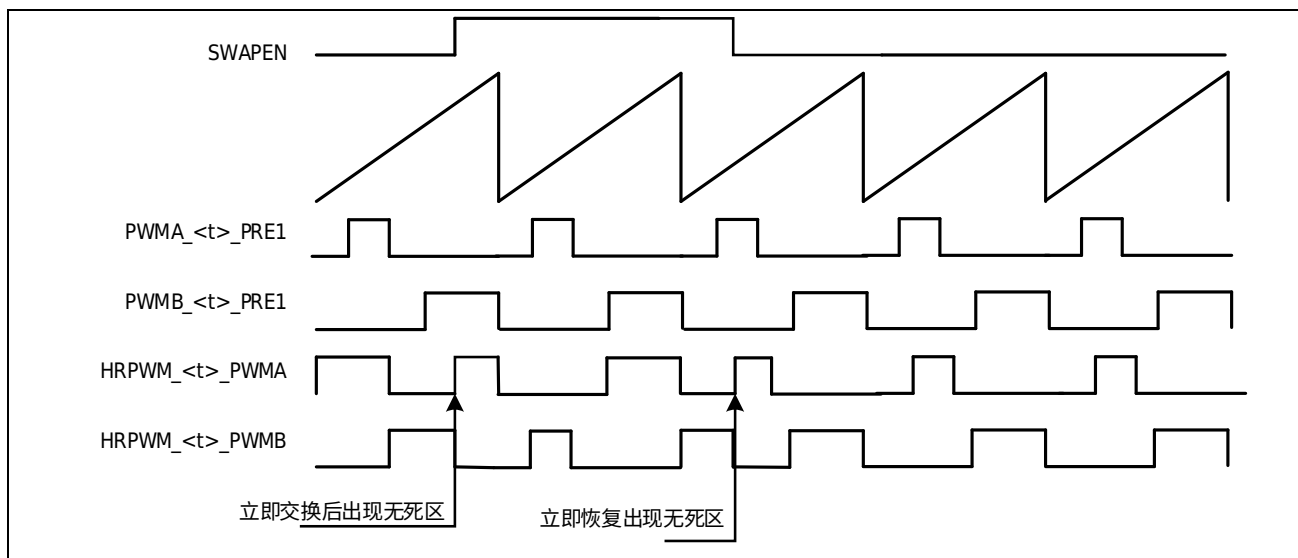


图 19-52 立即交换输出波形

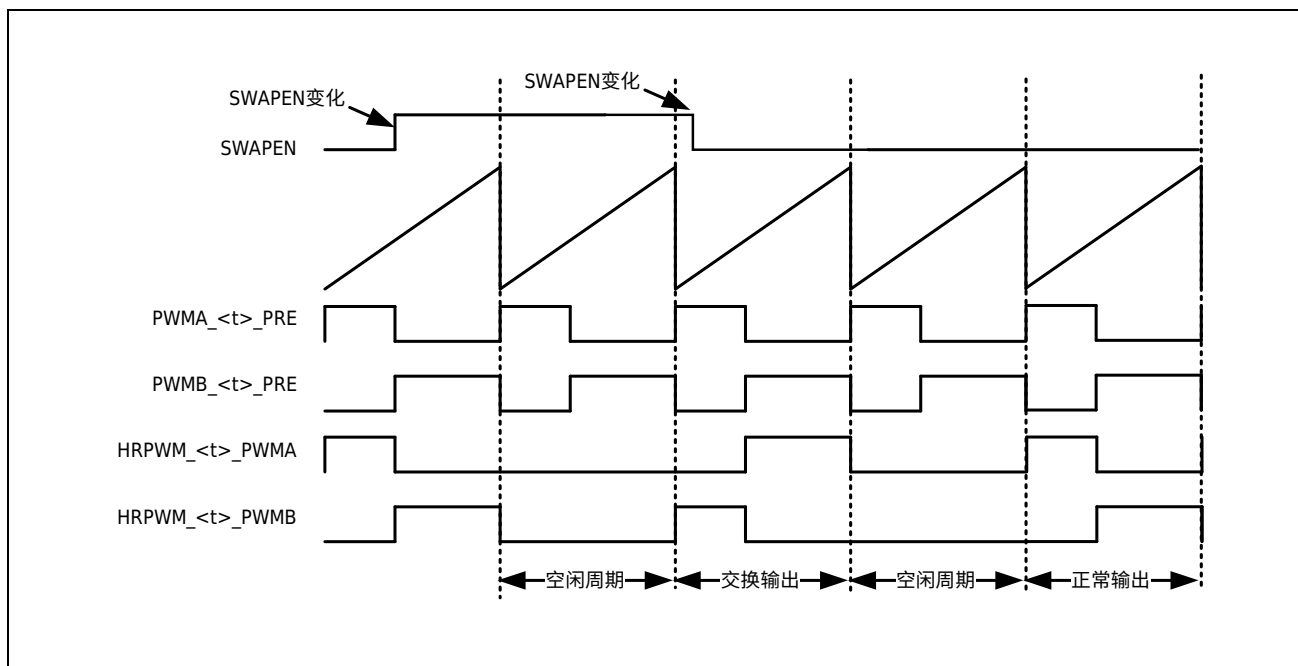


图 19-53 完整周期点交换输出波形

19.3.21 DAC 触发源

HRPWM 包括 3 个 DAC 同步触发信号，可以分别触发片上的 2 个 DAC 的数据与 HRPWM 同步更新。

通过配置每个单元内部的 HRPWM<t>_CR.DACSRC1<2>选择输出到 DAC 同步触发信号上的源，可以配置为计数值等于 0、计数值等于周期值或者锯齿波硬件清零、计数值等于 SCMAR、计数值等于 SCMBR。

通过配置单元内部 HRPWM<t>_CR.DACSYNC1<2>选择将上述源配置到 2 个 DAC 通道 1 和通道 2 同步触发信号上的一个。所有单元配置到同一个 DAC 通道的同步触发信号的源相或后送出，参考下图。

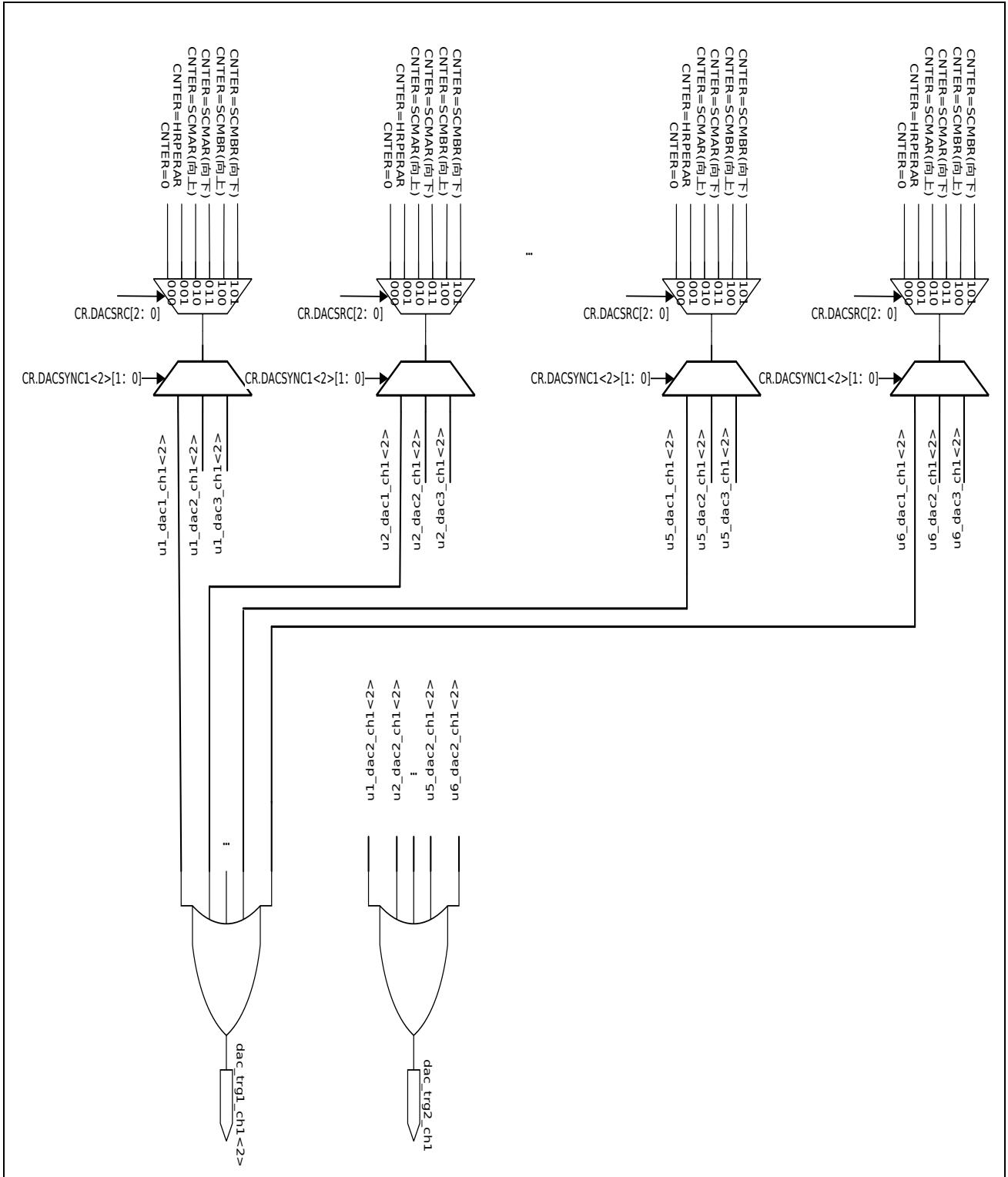


图 19-54 DAC 同步触发源

19.3.22 功能汇总表

HRPWM 的锯齿波模式和三角波模式下，主要功能的汇总表如下表 19-13 所示。

表 19-13 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	
端口状态控制	启动时	支持	支持	HRPWM<t>_PCNAR1.STACA	
	停止时	支持	支持	HRPWM<t>_PCNAR1.STPCA	
	周期匹配时	支持	支持	HRPWM<t>_PCNAR1.OVFCA	
	零点匹配时	支持	支持	HRPWM<t>_PCNAR1.UDFCA	
	外部事件发生 (Up Counting)	支持	支持	HRPWM<t>_PCNAR2.EXEVyUCA	
	外部事件发生 (Down Counting)	支持	支持	HRPWM<t>_PCNAR3.EXEVyDCA	
	计数匹配时 (Up Counting)	支持	支持	HRPWM<t>_PCNAR1.CMAUCA HRPWM<t>_PCNAR2.CME<F>UCA HRPWM<t>_PCNAR2.SCMAUCA	
	计数匹配时 (Down Counting)	支持	支持	HRPWM<t>_PCNAR1.CMADCA HRPWM<t>_PCNAR3.CME<F>DCA HRPWM<t>_PCNAR3.SCMAUCA	
缓存传送	间隔输出周 期基准值	单缓存	支持	支持	HRPWM_BMPERBR-> HRPWM_BMPERAR
	间隔输出比 较基准值	单缓存	支持	支持	HRPWM_BMCMBR-> HRPWM_BMCMAR
	外部事件滤 波偏移值	单缓存	支持	支持	HRPWM<t>_EEFFOFFSETBR-> HRPWM<t>_EEFFOFFSETAR
	外部事件滤 波窗口	单缓存	支持	支持	HRPWM<t>_EEFWINBR-> HRPWM<t>_EEFWINAR
	高分辨率周 期基准值	单缓存	支持	支持	HRPWM<t>_HRPERBR-> HRPWM<t>_HRPERAR
	高分辨率比 较基准值	单缓存	支持	支持	HRPWM<t>_HRGCMR-> HRPWM<t>_HRGCMBR HRPWM<t>_HRGCMCR-> HRPWM<t>_HRGCMAR HRPWM<t>_HRGCMGR-> HRPWM<t>_HRGCMER HRPWM<t>_HRGCMHR-> HRPWM<t>_HRGCMFR
	端口控制寄 存器	单缓存	支持	支持	HRPWM<t>_BPCNAR1-> HRPWM<t>_PCNAR1 HRPWM<t>_BPCNAR2-> HRPWM<t>_PCNAR2 HRPWM<t>_BPCNAR3-> HRPWM<t>_PCNAR3
	高分辨率死 区基准值	单缓存	支持	支持	HRDTUBR->HRDTUAR HRDTDBR->HRDTDAR
	中断控制寄 存器	单缓存	支持	支持	HRPWM<t>_BICONR-> HRPWM<t>_ICONR

PWM输出功能		锯齿波	三角波	相关主要寄存器
通用控制寄存器	单缓存	支持	支持	HRPWM<t>_BGCONR1-> HRPWM<t>_GCONR1
无死区PWM输出		支持	支持	HRPWM<t>_DCONR.DTCEN=0
带死区PWM输出		支持	支持	HRPWM<t>_DCONR.DTCEN=1
间隔输出		支持	支持	HRPWM<t>_IDLECR.IDLEBMA
延迟空闲模式		支持	支持	HRPWM<t>_IDLECR.DLYPRT
立即空闲		支持	支持	HRPWM_SSTAILDR. SSTAILD<t>A
EMB		支持	支持	HRPWM<t>PCNAR1.EMBCA

19.4 中断

19.4.1 中断输出

HRPWM 含有 6 个通用计数比较匹配中断、2 个专用计数比较匹配中断、1 个周期匹配中断、1 个零点匹配中断、2 个捕获中断、1 个间隔输出周期匹配中断、1 个校准完成中断和 1 个延迟空闲触发中断。

19.4.1.1 计数比较匹配中断

通用比较基准值寄存器 (HRPWM<t>_HRGCMAR~ HRPWM<t>_HRGCMFR) 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 (HRPWM<t>_STFLR1) 中的 HRPWM<t>_STFLR1.CMAF~STFLR1.CMFF 位分别会被置为 1。此时若设定中断控制寄存器 (HRPWM<t>_ICONR) 的 INTENA~INTENF 中相应位为 1 使能中断, 则对应的中断请求 (HRPWM_<t>_GCMA~F) 也会被触发。

2 个专用比较基准值寄存器 (HRPWM<t>_SCMAR~ HRPWM<t>_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 (HRPWM<t>_STFLR1) 中的 HRPWM<t>_STFLR1.CMSPAF~CMSPBF 位分别会被置为 1。此时若设定中断控制寄存器 (HRPWM<t>_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 (HRPWM_<t>_SCMA~B) 也会被触发。

19.4.1.2 周期/零点匹配中断

当计数器的计数值等于周期值或者锯齿波硬件清零发生时, 状态标志寄存器 (HRPWM<t>_STFLR1) 的 HRPWM<t>_STFLR1.OVFF 会被置为 1, 此时若设定中断控制寄存器 (HRPWM<t>_ICONR) 的 HRPWM<t>_ICONR.INTENOVF 位使能中断, 则会产生周期匹配中断 (HRPWM_<t>_GOVF)。

当计数器的计数值等于 0 时, HRPWM<t>_STFLR1.UDFF 位会被置为 1。此时若设定中断控制寄存器 (HRPWM<t>_ICONR) HRPWM<t>_ICONR.INTENUDF 位使能中断, 则在对应的时间点可触发计数零点匹配中断 (HRPWM_<t>_GUDF)。

19.4.1.3 捕获中断

在硬件捕获使能（HRPWM<t>_HCPAR2. HCPAEN），且在硬件捕获事件选择寄存器（HRPWM<t>_HCPAR1<2>、HRPWM<t>_HCPBR1<2>）选择的捕获输入有效条件产生时或者软件同步捕获触发时，捕获动作发生，状态标志寄存器（HRPWM<t>_STFLR1）中的 HRPWM<t>_STFLR1.CAPAF 位分别会被置为 1。此时若设置中断控制寄存器（HRPWM<t>_ICONR）的 INTENCAPA 或 INTENCAPB 位为 1 使能中断，则对应的中断请求（HRPWM_<t>_GCAPA）被触发。

19.4.1.4 间隔输出周期匹配中断

BM-counter 在递加计数至周期值时，间隔输出控制寄存器（BMCR）的 BMOVFF 会被置为 1。此时若设定间隔输出控制寄存器（BMCR）的 INTENBMOVFF 使能中断，则在对应的时间点可触发间隔输出周期匹配中断（HRPWM_BMOVFF）。

19.4.1.5 校准完成中断

当使用校准功能，且校准完成时，高分辨率校准控制寄存器（CALCR）的 CALENF 会被置为 1。高分辨率校准控制寄存器（CALCR）的 CALIE 使能中断，则在校准完成时可发生校准完成中断（HRPWM_CALENF）。

19.4.1.6 延迟空闲触发中断

当延迟空闲使能，一旦检测到触发信号，状态标志寄存器（HRPWM<t>_STFLR2）的 DLYPRT 会被置为 1。此时若设定输出控制寄存器（HRPWM<t>_IDLECR）的 INTENDLYPRT 使能中断，则可产生延迟空闲触发中断（HRPWM_DLYPRT）。所有单元共用一个中断，通过查询各个单元的 HRPWM<t>_STFLR2 寄存器的 DLYPRT 位确认发生延迟空闲的单元。

19.4.2 事件输出

在时钟计数过程中，若产生周期匹配事件、零点匹配事件（、通用计数比较匹配事件、专用计数比较匹配事件，捕获事件、外部事件 1~10 检出事件时，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。

外部事件 1~10 检出事件通过设定外部事件检出事件选择寄存器（HRPWM_EEDSELR）的 EEVy 寄存器位选择检出的外部事件，当选定的外部事件发生时，会生成外部事件检出事件（HRPWM_EEDET）。

各个单元专用匹配事件 A 和专用匹配事件 B 的可以在计数器向上计数器区间或者向下计数区间产生，通过配置 HRPWM<t>_GCONR1.CMSCAU<D>EN 使能不同的有效作用方向。

各单元专用计数匹配事件 A 不仅可以使能本单元的专用比较匹配事件 A 作为事件源，还可以通过配置专用匹配事件选择寄存器 A（HRPWM<t>_SCMASELR）选择多个事件源作为专用匹配事件 A 的源，包括单元 1~单元 6 的专用匹配事件 A~B、单元 1~6 的周期匹配事件、单元 1~6 的零点匹配事件、外

部事件检出事件。当寄存器 HRPWM_<t>_SCMASELR 值为 0 时，默认选择本单元的专用匹配事件 A 作为事件源。

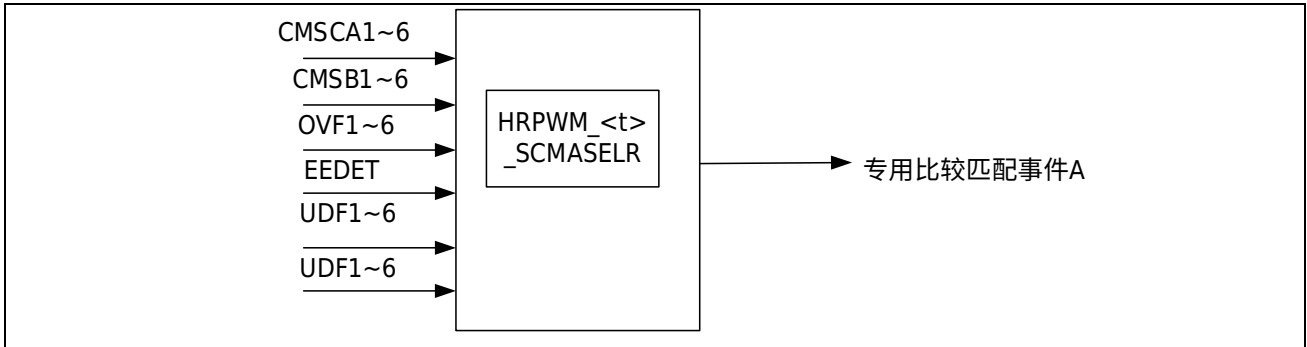


图 19-55 单元 1~3 专用计数匹配事件

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。

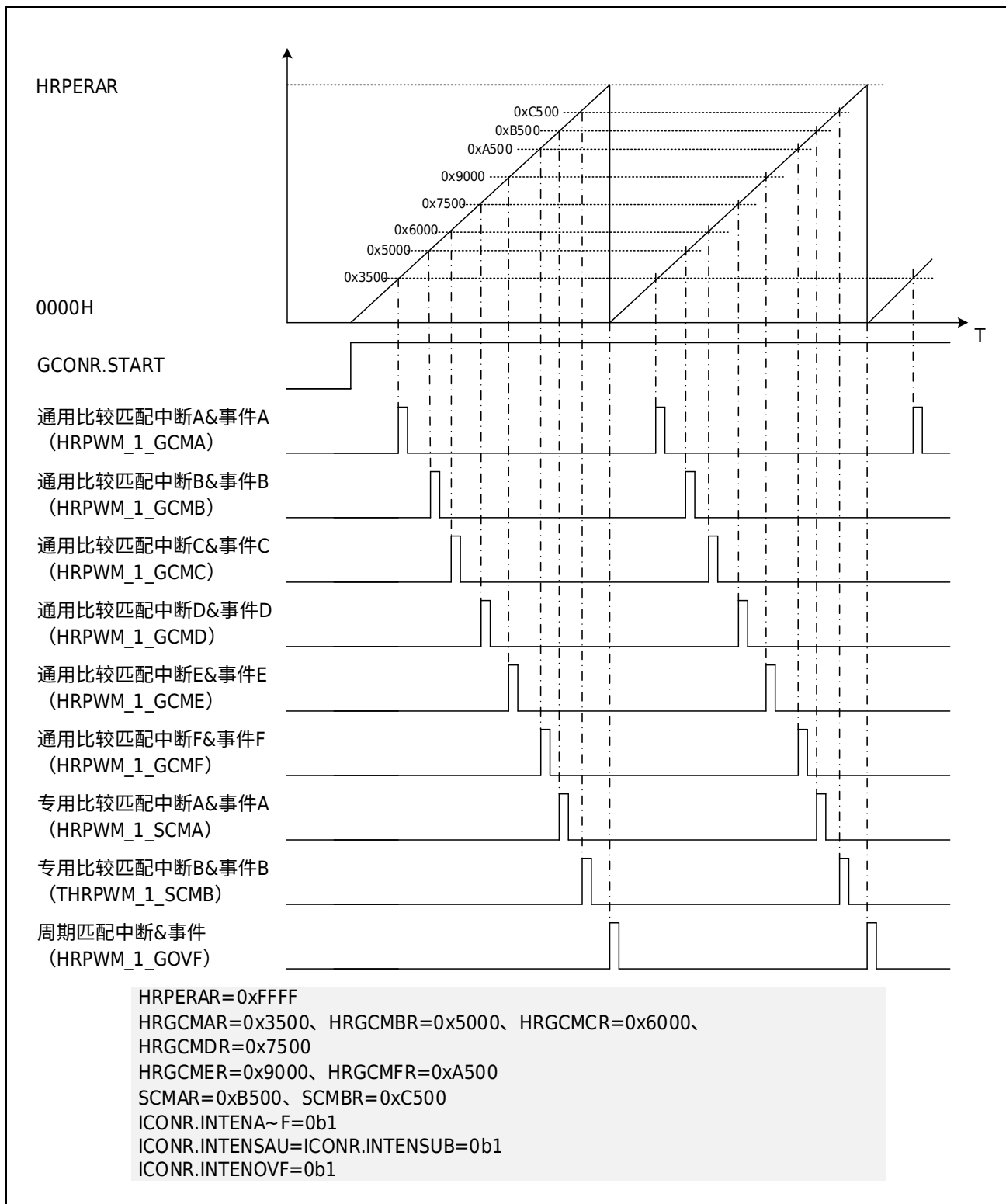


图 19-56 锯齿波模式时中断&事件输出例

19.5 典型应用例

下面描述几种典型应用情况下，HRPWM 相关寄存器的基本设定，供用户参考。

19.5.1 校准功能

在使能 HRPWM 前，需要按照以下步骤进行一次校准。

单次校准

- a) 配置使能时钟 PCLK0
- b) 配置 HRPWM_CALCR.CAL=1，使能单次校准
- c) 等待 HRPWM_CALCR.CALENF=1
- d) 确认 HRPWM_CALCR.ERRF=0

周期性校准

- a) 配置使能时钟 PCLK0
- b) 配置 HRPWM_CALCR.CALEN=1，使能周期校准
- c) 等待 HRPWM_CALCR.CALENF=1
- d) 确认 HRPWM_CALCR.ERRF=0

19.5.2 基本计数及中断动作

- a) 按照 19.5.1.1 设置步骤进行一次校准
- b) 使能 HRPWM (HRPWM<t>_CR.EN=1)
- c) 设定高分辨率通用周期基准值 (HRPWM<t>_HRPERAR)
- d) 设定需要的比较基准值，包括高分辨率通用比较基准值 (HRPWM<t>_HRGCMAR~HRPWM<t>_HRGCMFR)、专用比较基准值 (HRPWM<t>_SCMAR~HRPWM<t>_SCMBR) 等
- e) 设定需要的中断使能位，包括计数周期中断 (HRPWM<t>_ICONR.INTENOVF)、计数零点中断 (HRPWM<t>_ICONR.INTENUDF)、计数匹配中断 (HRPWM<t>_ICONR.INTENA~F、HRPWM<t>_ICONR.INTENSAU、HRPWM<t>_ICONR.INTENSAD、HRPWM<t>_ICONR.INTENSBU、HRPWM<t>_ICONR.INTENSBD) 等
- f) 设定波形模式 (HRPWM<t>_GCONR.MODE)
- g) 启动计数器 (HRPWM<t>_GCONR.START=1)

19.5.3 比较输出及中断动作

- a) 按照校准功能设置步骤进行一次校准
- b) 使能 HRPWM (HRPWM<t>_CR.EN=1)
- c) 设定高分辨率通用周期基准值 (HRPWM<t>_HRPERAR)
- d) 设定各通道的比较基准值，包括高分辨率通用比较基准值 A、E (HRPWM<t>_HRGCMA<E>R)、

高分辨率通用比较基准值 B、F (HRPWM<t>_HRGCMB<F>R)

- e) 设定需要的中断使能位，包括计数周期中断 (HRPWM<t>_ICONR.INTENOVF)、计数零点中断 (HRPWM<t>_ICONR.INTENUDF)、计数匹配中断 (HRPWM<t>_ICONR.INTENA~B) 等
- f) 设定各通道在不同计数状态时的端口输出状态 (参考 HRPWM<t>_PCNAR1<2><3>、HRPWM<t>_PCNBR1<2><3>、的相关控制)
- g) 设定波形模式 (HRPWM<t>_GCONR.MODE)
- h) 设定各通道输出使能 (HRPWM<t>_PCNAR1.OUTENA=1、HRPWM<t>_PCNBR1.OUTENB=1)
- i) 设定退出立即空闲 (HRPWM_SSTARUNR.SSTARUN<t>A=1)
- j) 启动计数器 (HRPWM<t>_GCONR.START=1)

19.5.4 间隔输出及中断动作

- a) 参考 19.5.1.3 章节的 a~i 步骤，对需要启动的各个单元做设定
- b) 设定需要的中断使能位，BM-counter 计数上溢中断 (HRPWM_BMCR.INTENBMOVF=1)
- c) 使能 PWM 端口间隔输出 (HRPWM<t>_IDLECR.IDLEBMA、HRPWM<t>_IDLECR.IDLEBMB)
- d) 配置 PWM 空闲状态下的状态 (HRPWM<t>_IDLECR.IDLESA、HRPWM<t>_IDLECR.IDLESB)
- e) 设定间隔输出周期基准值 (HRPWM_BMPERAR)
- f) 设定间隔输出比较基准值 (HRPWM_BMCMAR)
- g) 设定间隔输出运行模式 (HRPWM_BMCR.BMCTN)
- h) 设定间隔输出计数时钟 (HRPWM_BMCR.BMCLKS[1:0]，HRPWM_BMCR.BMPSC[3:0])
- i) 设定间隔输出下单元内计数器运行状态 (HRPWM_BMCR.BMTMR)
- j) 使能间隔输出 (HRPWM_BMCR.BMEN=1)
- k) 启动计数器 (HRPWM<t>_GCONR.START=1)
- l) 启动间隔输出 (HRPWM_BMSTRG.SSTRG=1)

19.5.5 缓存传送动作 (周期基准值)

- a) 按照校准功能设置步骤进行一次校准
- b) 使能 HRPWM (HRPWM<t>_CR.EN=1)
- c) 设定需要的高分辨率通用周期基准值 (HRPWM<t>_HRPERAR、HRPWM<t>_HRPERBR)
- d) 设定缓存传送时间点 (HRPWM<t>_BCONR1.BTRUP、HRPWM<t>_BCONR1.BTRDP、HRPWM<t>_BCONR2.BTRU0P、HRPWM<t>_BCONR2.BTRU0PP (仅相位功能使能时有效))
- e) 设定波形模式 (HRPWM<t>_GCONR.MODE)
- f) 设定缓存功能有效 (HRPWM<t>_BCONR1.BENP=1)
- g) 启动计数器 (HRPWM<t>_GCONR.START=1)
- h) 等待对应的缓存传送时间点，发生缓存动作 (HRPWM<t>_HRPERBR->H HRPWM<t>_RPERAR)

19.5.6 缓存传送动作（通用比较基准值）

- a) 按照校准功能设置步骤进行一次校准
- b) 使能 HRPWM (HRPWM<t>_CR.EN=1)
- c) 设定需要的高分辨率通用比较基准值 (HRPWM<t>_HRGCMAR~HR)
- d) 设定各通道缓存传送时间点 (HRPWM<t>_BCONR1.BTRUAE、HRPWM<t>_BCONR1.BTRDAE、
HRPWM<t>_BCONR1.BTRUBF 、 HRPWM<t>_BCONR1.BTRDBF 、
HRPWM<t>_BCONR2.BTRUP0AE 、 HRPWM<t>_BCONR2.BTRU0PBF 、
HRPWM<t>_BCONR2.BTRU0AE、HRPWM<t>_BCONR2.BTRU0BF (仅相位功能使能时有效))
- e) 设定波形模式 (HRPWM<t>_GCONR.MODE)
- f) 设定各通道缓存功能有效 (HRPWM<t>_BCONR1.BENAE=1、HRPWM<t>_BCONR1.BENBF=1)
- g) 启动计数器 (HRPWM<t>_GCONR.START=1)
- h) 等待各通道所设定的对应缓存传送时间点，发生缓存动作 (HRPWM<t>_HRGCMCR->
HRPWM<t>_HRGCMAR、HRPWM<t>_HRGCMCR->HRGCMBR、HRPWM<t>_HRGCMGR->
HRPWM<t>_HRGCMER、HRPWM<t>_HRGCMHR->HRPWM<t>_HRGCMFR)

19.5.7 缓存传送动作（专用比较基准值）

- a) 按照校准功能设置步骤进行一次校准
- b) 使能 HRPWM (HRPWM<t>_CR.EN=1)
- c) 设定需要的专用比较基准值 (HRPWM<t>_SCMAR、HRPWM<t>_SCMCR、HRPWM<t>_SCMBR、
HRPWM<t>_SCMDR)
- d) 设定各通道缓存传送时间点 (HRPWM<t>_BCONR.BTRUSPA、HRPWM<t>_BCONR1.BTRDSPA、
HRPWM<t>_BCONR1.BTRUSPB 、 HRPWM<t>_BCONR1.BTRDSPB 、
HRPWM<t>_BCONR2.BTRU0PSA 、 HRPWM<t>_BCONR2.BTRU0PSB 、
HRPWM<t>_BCONR2.BTRU0SPA、HRPWM<t>_BCONR2.BTRU0SPB (仅相位功能使能时有效))
- e) 设定波形模式 (HRPWM<t>_GCONR.MODE)
- f) 设定各通道缓存功能有效 (HRPWM<t>_BCONR1.BENSPA=1 、
HRPWM<t>_BCONR1.BENSPB=1)
- g) 启动计数器 (HRPWM<t>_GCONR.START=1)
- h) 等待各通道所设定的对应缓存传送时间点，发生缓存动作 (HRPWM<t>_SCMCR->
HRPWM<t>_SCMAR、HRPWM<t>_SCMDR->HRPWM<t>_SCMBR)

19.5.8 缓存传送动作（外部事件滤波偏移值）

- a) 按照校准功能设置步骤进行一次校准
- b) 使能 HRPWM (HRPWM<t>_CR.EN=1)
- c) 设定需要的外部事件滤波偏移值 (HRPWM<t>_EEFFOFFSETAR、HRPWM<t>_EEFFOFFSETBR)

- d) 设定各通道缓存传送时间点 (HRPWM<t>_BCONR2.BTRUEEFOFF、BCONR2.BTRDEEFOFF、HRPWM<t>_BCONR2.BTRU0PEEFOFF、HRPWM<t>_BCONR2.BTRU0EEFOFF (仅相位功能使能时有效))
- e) 设定波形模式 (HRPWM<t>_GCONR.MODE)
- f) 设定各通道缓存功能有效 (HRPWM<t>_BCONR2.BENEEOFF=1)
- g) 启动计数器 (HRPWM<t>_GCONR.START=1)
- h) 等待各通道所设定的对应缓存传送时间点, 发生缓存动作 (HRPWM<t>_EEOFFSETBR->HRPWM<t>_EEOFFSETAR)

19.5.9 缓存传送动作 (外部事件滤波窗口)

- a) 按照校准功能设置步骤进行一次校准
- b) 使能 HRPWM (HRPWM<t>_CR.EN=1)
- c) 设定需要的外部事件滤波窗口 (HRPWM<t>_EEFWINAR、HRPWM<t>_EEFWINBRR)
- d) 设定各通道缓存传送时间点 (HRPWM<t>_BCONR2.BTRUEEFWIN、HRPWM<t>_BCONR2.BTRDEEFWIN、HRPWM<t>_BCONR2.BTRU0PEEFWIN、HRPWM<t>_BCONR2.BTRU0EEFWIN (仅相位功能使能时有效))
- e) 设定波形模式 (HRPWM<t>_GCONR.MODE)
- f) 设定各通道缓存功能有效 (HRPWM<t>_BCONR2.BENEEFWIN=1)
- g) 启动计数器 (HRPWM<t>_GCONR.START=1)
- h) 等待各通道所设定的对应缓存传送时间点, 发生缓存动作 (HRPWM<t>_EEFWINBR->HRPWM<t>_EEFWINAR)

19.5.10 缓存传送动作 (间隔输出周期基准值)

- a) 参考比较输出及中断动作章节的 a~f 步骤, 对需要启动的各个单元做设定
- b) 设定需要的间隔输出周期基准值 (HRPWM_BMPERAR、HRPWM_BMPERBR)
- c) 设定间隔输出比较基准值 (HRPWM_BMCMAR, HRPWM_BMCMBR) (使用间隔输出周期基准值缓存功能时, 在写完周期值需要再写一次比较值)
- d) 设定缓存功能有效 (HRPWM_BMCR.BENBMP=1)
- e) 设定间隔输出空闲时 PWM 端口的状态 (HRPWM<t>_IDLECR.IDLES<A>B、HRPWM<t>_IDLECR.IDLEBM<A>B)
- f) 设定间隔输出运行模式 (HRPWM_BMCR.BMCTN)
- g) 设定间隔输出计数时钟 (HRPWM_BMCR.BMCLKS[1:0], HRPWM_BMCR.BMPSC[3:0])
- h) 设定间隔输出下单元内计数器运行状态 (HRPWM_BMCR.BMTMR<t>)
- i) 使能间隔输出 (HRPWM_BMCR.BMEN=1)
- j) 启动计数器 (HRPWM<t>_GCONR.START=1)

- k) 启动间隔输出 (HRPWM_BMSTRG.SSTRG=1)
- l) 等待设定的对应缓存传送时间点, 发生缓存动作 (HRPWM_BMPERBR-> HRPWM_BMPERAR)

19.5.11 缓存传送动作 (间隔输出比较基准值)

- a) 参考比较输出及中断动作章节的 a~f 步骤, 对需要启动的各个单元做设定
- b) 设定间隔输出周期基准值 (HRPWM_BMPERAR)
- c) 设定需要的间隔输出比较基准值 (HRPWM_BCMAR, HRPWM_BCMBR)
- d) 设定缓存功能有效 (HRPWM_BMCR.BENBMCMP=1)
- e) 设定间隔输出空闲时 PWM 端口的状态 (HRPWM<t>_IDLECR.IDLES<A>B、HRPWM<t>_IDLECR.IDLEBM<A>B)
- f) 设定间隔输出运行模式 (HRPWM_BMCR.BMCTN)
- g) 设定间隔输出计数时钟 (HRPWM_BMCR.BMCLKS[1:0], HRPWM_BMCR.BMPSC[3:0])
- h) 设定间隔输出下单元内计数器运行状态 (HRPWM_BMCR.BMTMR<t>)
- i) 使能间隔输出 (HRPWM_BMCR.BMEN=1)
- j) 启动计数器 (HRPWM<t>_GCONR.START=1)
- k) 启动间隔输出 (HRPWM_BMSTRG.SSTRG=1)
- l) 等待设定的对应缓存传送时间点, 发生缓存动作 (HRPWM_BMCMBR-> HRPWM_BCMAR)

19.5.12 缓存传送动作 (死区基准值)

- a) 按照校准功能设置步骤进行一次校准
- b) 使能 HRPWM (HRPWM<t>_CR.EN=1)
- c) 设定需要的高分辨率死区时间基准值 (HRPWM<t>_HRDTUAR、HRPWM<t>_HRDTUBR、HRPWM<t>_HRDTDAR、HRPWM<t>_HRDDBR)
- d) 设定缓存传送时间点 (HRPWM<t>_DCONR.DTBTRU、HRPWM<t>_DCONR.DTBTRD、HRPWM<t>_BCONR2.BTRU0PDT、HRPWM<t>_BCONR2.BTRU0DT (仅相位功能使能时有效))
- e) 设定波形模式 (HRPWM<t>_GCONR.MODE)
- f) 设定高分辨率缓存功能有效 (HRPWM<t>_DCONR.DTBENU=1、HRPWM<t>_DCONR.DTBEND=1)
- g) 设定硬件死区功能有效 (HRPWM<t>_DCONR.DTCEN=1)
- h) 启动计数器 (HRPWM<t>_GCONR.START=1)
- i) 等待对应缓存传送时间点, 发生缓存动作 (HRPWM<t>_HRDTUBR-> HRPWM<t>_HRDTUAR、HRPWM<t>_HRDDBR-> HRPWM<t>_HRDTDAR)

19.5.13 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 HRPWM_SSTAR 寄存器的对应位为 1，每个单元对应一个寄存器位）

19.5.14 同步启动动作（硬件方式）

- a) 按照校准功能设置步骤进行一次校准
- b) 使能 HRPWM (HRPWM<t>_CR.EN=1)
- c) 设定高分辨率通用周期基准值 (HRPWM<t>_HRPERAR)
- d) 设定需要的比较基准值, 包括高分辨率通用比较基准值 (HRPWM<t>_HRGCMAR~HRGCMFR)、专用比较基准值 (HRPWM<t>_SCMAR~SCMBR) 等
- e) 设定需要的中断使能位, 包括计数周期中断 (HRPWM<t>_ICONR.INTENOVF)、计数零点中断 (HRPWM<t>_ICONR.INTENUDF)、计数匹配中断 (HRPWM<t>_ICONR.INTENA~F、HRPWM<t>_ICONR.INTENSAU、HRPWM<t>_ICONR.INTENSAD、HRPWM<t>_ICONR.INTENSBU、HRPWM<t>_ICONR.INTENSB) 等
- f) 设定硬件启动条件 (通过 HRPWM<t>_HSTAR1, HRPWM<t>_HSTAR2 选择)
- g) 设定硬件启动使能 (HRPWM<t>_HSTAR1.STAS=1)
- h) 重复上述 a~g 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 f 的设定要一致)
- i) 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

19.5.15 同步输出

- a) 参考[比较输出及中断动作]章节 a~f 步骤的设定需要的比较基准值, 包括高分辨率通用比较基准值 (HRPWM<t>_HRGCMAR~HRPWM<t>_HRGCMFR)、专用比较基准值 (HRPWM<t>_SCMAR~HRPWM<t>_SCMBR) 等
- b) 设定同步输出信号的输出源 (HRPWM_SYNOCR.SYNCSRC[3:0])
- c) 设定同步输出信号源的计数方向 (HRPWM_SYNOCR.SRCDIR)
- d) 设定输出的脉冲类型 (HRPWM_SYNOCR.SYNOPLS[1:0])
- e) 设定同步输出信号脉冲长度 (HRPWM_SYNOCR.SYNCCMP[7:0])
- f) 启动计数器 (HRPWM<t>_GCONR.START=1)
- g) 等待设定的输出源发生, 确认 HRPWM_SCOUT 输出所需的脉冲

19.5.16 单路 PWM 输出

参考[比较输出及中断动作]章节 a~f 步骤的设定 (每个单元内部的 2 个 PWM 通道 HRPWM_<t>_PWMA 和 HRPWM_<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)

19.5.17 互补 PWM 输出（软件死区）

- a) 按照校准功能设置步骤进行一次校准
- b) 使能高分辨率 (HRPWM<t>_CR.EN=1)

- c) 设定通用高分辨率周期基准值 (HRPWM<t>_HRPERAR)
- d) 设定高分辨率通用比较基准值 A、E (HRPWM<t>_HRGCMA<E>R)、高分辨率通用比较基准值 B、F (HRPWM<t>_HRGCMB<F>R)。
- e) 设定需要的中断使能位，包括计数周期中断 (HRPWM<t>_ICONR.INTENOVF)、计数零点中断 (HRPWM<t>_ICONR.INTENUDF)、计数匹配中断 (HRPWM<t>_ICONR.INTENA~B) 等
- f) 设定在不同计数状态时的端口输出状态 (参考 HRPWM<t>_PCNAR1<2><3>、HRPWM<t>_PCNBR1<2><3>的相关控制，设置合适的高分辨率通用比较基准值，保证 2 个 PWM 输出之间形成互补死区)
- g) 设定波形模式 (HRPWM<t>_GCONR.MODE)
- h) 设定输出使能 (HRPWM<t>_PCNAR1.OUTENA=1、HRPWM<t>_PCNBR1.OUTENB=1)
- i) 设定退出立即空闲 (HRPWM_SSTARUNR<t>.SSTARUN1A=1)
- j) 启动计数器 (GCONR.START=1)

19.5.18 互补 PWM 输出 (硬件死区)

- a) 按照校准功能设置步骤进行一次校准
- b) 使能 HRPWM (CR.EN=1)
- c) 设定高分辨率通用周期基准值 (HRPERAR)
- d) 设定高分辨率通用比较基准值 A、E (HRGCMA<E>R)、高分辨率通用比较基准值 B、F (HRGCMB<F>R)。
- e) 设定需要的中断使能位，包括计数周期中断 (ICONR.INTENOVF)、计数零点中断 (ICONR.INTENUDF)、计数匹配中断 (ICONR.INTENA~B) 等
- f) 设定在不同计数状态时的通道 A 内部输出信号 (HRPWM_<t>_PWMA_ORG) 输出状态 (参考 PCNAR1<2><3>的相关控制，结合 HRGCMAR、HRDTUAR 和 HRDTDAR 的设定值，需要保证 2 个 PWM 输出之间形成互补死区)
- g) 设定波形模式 (GCONR.MODE)
- h) 设定各通道输出使能 (PCNAR.OUTENA=1、PCNBR.OUTENB=1)
- i) 设定退出立即空闲 (HRPWM_SSTARUNR.SSTARUN<t>A=1)
- j) 设定硬件死区功能有效 (HRPWM<t>_DCONR.DTCEN=1)
- k) 启动计数器 (HRPWM<t>_GCONR.START=1)

19.5.19 EMB 监控及中断动作

- a) 参考[互补 PWM 输出 (软件死区)]章节的 a~i 步骤或[互补 PWM 输出 (硬件死区)]章节 a~j 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 (HRPWM<t>_PCNAR1.EMBCA、HRPWM<t>_PCNBR1.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 (HRPWM<t>_PCNAR1.EMBRA、HRPWM<t>_PCNBR1.EMBRB)
- d) 设定选择从 EMB 模块输入的 EMB 事件源 (HRPWM<t>_PCNAR1.EMBSA 、 HRPWM<t>_PCNBR1.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 (HRPWM<t>_GCONR.START=1), EMB 模块实时监控状态

19.6 寄存器说明

表 19-14 HRPWN 基地址

名称	基地址	描述
HRPWM1	0x4003 C000	HRPWM1 基地址
HRPWM2	0x4003 C400	HRPWM2 基地址
HRPWM3	0x4003 C800	HRPWM3 基地址
HRPWM4	0x4003 CC00	HRPWM4 基地址
HRPWM5	0x4003 D000	HRPWM5 基地址
HRPWM6	0x4003 D400	HRPWM6 基地址
HRPWM_COMMON	0x4003 C2F0	HRPWM 公共寄存器基地址

表 19-15 HRPWM 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
HRPWM<t>_CNTER	通用计数值寄存器	0x0000	32	0x0000 0000
HRPWM<t>_UPDAR	刷新值寄存器	0x0004	32	0x0000 0000
HRPWM<t>_HRPERAR	高分辨率通用周期基准值寄存器A	0x0040	32	0x003F FFC0
HRPWM<t>_HRPERBR	高分辨率通用周期基准值寄存器B	0x0044	32	0x003F FFC0
HRPWM<t>_HRGCMAR	高分辨率通用比较基准值寄存器A	0x0080	32	0x003F FFC0
HRPWM<t>_HRGCMBR	高分辨率通用比较基准值寄存器B	0x0084	32	0x003F FFC0
HRPWM<t>_HRGCMCR	高分辨率通用比较基准值寄存器C	0x0088	32	0x003F FFC0
HRPWM<t>_HRGCMDR	高分辨率通用比较基准值寄存器D	0x008C	32	0x003F FFC0
HRPWM<t>_HRGCMER	高分辨率通用比较基准值寄存器E	0x0090	32	0x003F FFC0
HRPWM<t>_HRGCMFR	高分辨率通用比较基准值寄存器F	0x0094	32	0x003F FFC0
HRPWM<t>_HRGCMGR	高分辨率通用比较基准值寄存器G	0x0098	32	0x003F FFC0
HRPWM<t>_HRGCMHR	高分辨率通用比较基准值寄存器H	0x009C	32	0x003F FFC0
HRPWM<t>_SCMAR	专用比较基准值寄存器A	0x00C0	32	0x003F FFC0
HRPWM<t>_SCMBR	专用比较基准值寄存器B	0x00C4	32	0x003F FFC0
HRPWM<t>_SCMCR	专用比较基准值寄存器C	0x00C8	32	0x003F FFC0
HRPWM<t>_SCMDR	专用比较基准值寄存器D	0x00CC	32	0x003FFFC0
HRPWM<t>_CAPAR	捕获值寄存器A	0x00D0	32	0x003FFFC0
HRPWM<t>_CAPBR	捕获值寄存器B	0x00D4	32	0x003FFFC0
HRPWM<t>_HRDTUAR	高分辨率死区时间基准寄存器UA	0x0100	32	0x003FFFC0
HRPWM<t>_HRDTDAR	高分辨率死区时间基准寄存器DA	0x0104	32	0x003FFFC0
HRPWM<t>_HRDTUBR	高分辨率死区时间基准寄存器UB	0x0108	32	0x003FFFC0
HRPWM<t>_HRDTDBR	高分辨率死区时间基准寄存器DB	0x010C	32	0x003FFFC0
HRPWM<t>_GCONR	通用控制寄存器	0x0140	32	0x00000000
HRPWM<t>_ICONR	中断控制寄存器	0x0144	32	0x00000000
HRPWM<t>_BCONR1	缓存控制寄存器1	0x0148	32	0x00000000

寄存器	描述	偏移地址	位宽	复位值
HRPWM<t>_DCONR	死区控制寄存器	0x014C	32	0x00000000
HRPWM<t>_PCNAR1	端口控制寄存器A1	0x0154	32	0x00000000
HRPWM<t>_PCNBR1	端口控制寄存器B1	0x0158	32	0x00000000
HRPWM<t>_VPERR	有效周期寄存器	0x0160	32	0x00000000
HRPWM<t>_STFLR1	状态标志寄存器1	0x0164	32	0x80000000
HRPWM<t>_STFLR2	状态标志寄存器2	0x0168	32	0x00000000
HRPWM<t>_HSTAR1	硬件启动事件选择寄存器1	0x0180	32	0x00000000
HRPWM<t>_HCLRR1	硬件清零事件选择寄存器1	0x0188	32	0x00000000
HRPWM<t>_HCPAR1	硬件捕获事件选择寄存器A1	0x0190	32	0x00000000
HRPWM<t>_HCPBR1	硬件捕获事件选择寄存器B1	0x0194	32	0x00000000
HRPWM<t>_HCPAR2	硬件捕获事件选择寄存器A2	0x0198	32	0x00000000
HRPWM<t>_HCPBR2	硬件捕获事件选择寄存器B2	0x019C	32	0x00000000
HRPWM<t>_BCONR2	缓存控制寄存器2	0x0200	32	0x00000000
HRPWM<t>_PCNAR2	端口控制寄存器A2	0x0204	32	0x000AAAAA
HRPWM<t>_PCNBR2	端口控制寄存器B2	0x0208	32	0x000AAAAA
HRPWM<t>_HSTAR2	硬件启动事件选择寄存器2	0x0210	32	0x00000000
HRPWM<t>_HCLRR2	硬件清零事件选择寄存器2	0x0218	32	0x00000000
HRPWM<t>_EEFOFFSETAR	外部事件滤波偏移值寄存器A	0x0220	32	0x0040 0000
HRPWM<t>_EEFOFFSETBR	外部事件滤波偏移值寄存器B	0x0224	32	0x0040 0000
HRPWM<t>_EEFWINAR	外部事件滤波窗口寄存器A	0x0228	32	0x0040 0000
HRPWM<t>_EEFWINBR	外部事件滤波窗口寄存器B	0x022C	32	0x0040 0000
HRPWM<t>_EEFLTCR1	外部事件滤波控制器1	0x0230	32	0x0000 0000
HRPWM<t>_EEFLTCR2	外部事件滤波控制器2	0x0234	32	0x0000 0000
HRPWM<t>_PCNAR3	端口控制寄存器A3	0x0238	32	0x000A AAAA
HRPWM<t>_PCNBR3	端口控制寄存器B3	0x023C	32	0x000A AAAA
HRPWM<t>_IDLECR	空闲控制寄存器	0x0240	32	0x0000 0000
HRPWM<t>_GCONR1	通用控制寄存器1	0x0244	32	0x0000 0000
HRPWM<t>_BICONR	中断控制缓存寄存器	0x0248	32	0x0000 0000
HRPWM<t>_BPCNAR1	端口控制缓存寄存器A1	0x024C	32	0x0000 0000
HRPWM<t>_BPCNBR1	端口控制缓存寄存器B1	0x0250	32	0x0000 0000
HRPWM<t>_BPCNAR2	端口控制缓存寄存器A2	0x0254	32	0x000A AAAA
HRPWM<t>_BPCNBR2	端口控制缓存寄存器B2	0x0258	32	0x000A AAAA
HRPWM<t>_BPCNAR3	端口控制缓存寄存器A3	0x025C	32	0x000A AAAA
HRPWM<t>_BPCNBR3	端口控制缓存寄存器B3	0x0260	32	0x000A AAAA
HRPWM<t>_BGCONR1	通用控制缓存寄存器1	0x0264	32	0x0000 0000
HRPWM<t>_SCMASELR	专用匹配事件A选择寄存器	0x0268	32	0x0000 0000
HRPWM<t>_CR	高分辨率控制寄存器	0x02A0	32	0x0000 0000
HRPWM<t>_PHSCTL	相位控制寄存器	0x02A4	32	0x0000 0000

寄存器	描述	偏移地址	位宽	复位值
HRPWM1_PHSCMP1A	高分辨率相位比较基准值寄存器1A	0x02B4	32	0x007F FFC0
HRPWM1_PHSCMP1B	高分辨率相位比较基准值寄存器1B	0x02B8	32	0x007F FFC0
HRPWM1_PHSCMP2A	高分辨率相位比较基准值寄存器2A	0x02BC	32	0x007F FFC0
HRPWM1_PHSCMP2B	高分辨率相位比较基准值寄存器2B	0x02C0	32	0x007F FFC0
HRPWM1_PHSCMP3A	高分辨率相位比较基准值寄存器3A	0x02C4	32	0x007F FFC0
HRPWM1_PHSCMP3B	高分辨率相位比较基准值寄存器3B	0x02C8	32	0x007F FFC0
HRPWM1_PHSCMP4A	高分辨率相位比较基准值寄存器4A	0x02CC	32	0x007F FFC0
HRPWM1_PHSCMP4B	高分辨率相位比较基准值寄存器4B	0x02D0	32	0x007F FFC0
HRPWM1_PHSCMP5A	高分辨率相位比较基准值寄存器5A	0x02D4	32	0x007F FFC0
HRPWM1_PHSCMP5B	高分辨率相位比较基准值寄存器5B	0x02D8	32	0x007F FFC0

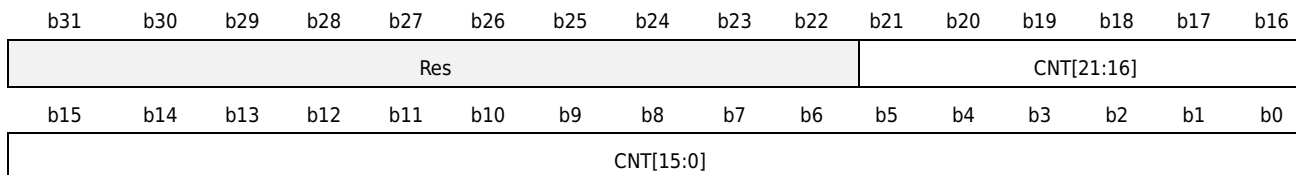
表 19-16 HRPWM 公共寄存器列表

寄存器	描述	偏移地址	位宽	复位值
HRPWM_CALCR	高分辨率校准控制寄存器	0x000C	32	0x0000 0000
HRPWM_SCAPR	软件同步捕获控制寄存器	0x00AC	32	0x0000 0000
HRPWM_SSTAIDL	软件同步空闲控制寄存器	0x00B0	32	0x0000 0000
HRPWM_SSTARUNR1	软件同步运行控制寄存器1	0x00B4	32	0x0000 0000
HRPWM_SSTARUNR2	软件同步运行控制寄存器2	0x00B8	32	0x0000 0000
HRPWM_SSTADIDL	软件同步延迟空闲控制寄存器	0x00BC	32	0x0000 0000
HRPWM_GCTRL	全局控制寄存器	0x00C0	32	0x0000 0000
HRPWM_GBCONR	全局缓存控制寄存器	0x00C4	32	0x0000 003F
HRPWM_GBSFLR	全局缓存状态标志寄存器	0x00C8	32	0x0000 0000
HRPWM_BMCR	间隔输出控制寄存器	0x00CC	32	0x0000 0000
HRPWM_BMSTRG1	间隔输出启动触发寄存器1	0x00D0	32	0x0000 0000
HRPWM_BMSTRG2	间隔输出启动触发寄存器2	0x00D4	32	0x0000 0000
HRPWM_BMPERAR	间隔输出周期基准值寄存器A	0x00D8	32	0x0000 0000
HRPWM_BMPERBR	间隔输出周期基准值寄存器B	0x00DC	32	0x0000 0000
HRPWM_BCMAR	间隔输出比较基准值寄存器A	0x00E0	32	0x0000 0000
HRPWM_BCMBR	间隔输出比较基准值寄存器B	0x00E4	32	0x0000 0000
HRPWM_EECCR1	外部事件控制器1	0x00E8	32	0x0000 0000
HRPWM_EECCR2	外部事件控制器2	0x00EC	32	0x0000 0000
HRPWM_EECCR3	外部事件控制器3	0x00F0	32	0x0000 0000
HRPWM_SYNOCR	同步输出控制寄存器	0x00F4	32	0x0000 0000
HRPWM_EEDSELR	外部事件检出事件选择寄存器	0x00F8	32	0x0000 0000
HRPWM_FCNT	滤波控制寄存器	0x00FC	32	0x0000 0000
HRPWM_SSTAR	软件同步启动控制寄存器	0x0100	32	0x0000 0000
HRPWM_SSTPR	软件同步停止控制寄存器	0x0104	32	0x0000 0000
HRPWM_SCLRR	软件同步清零控制寄存器	0x0108	32	0x0000 0000
HRPWM_SUPDR	软件同步刷新控制寄存器	0x010C	32	0x0000 0000

19.6.1 HRPWM 单元 1~6 寄存器

19.6.1.1 通用计数值寄存器 (HRPWM<t>_CNTER)

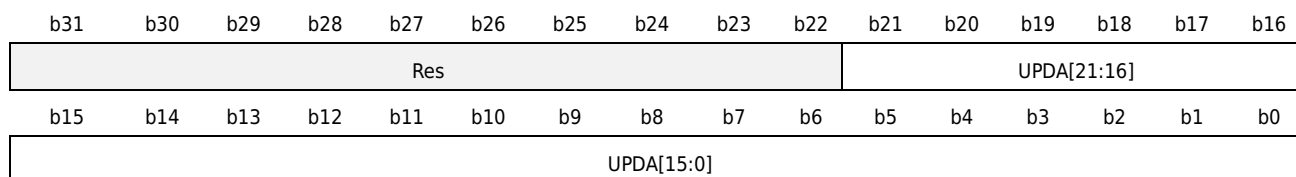
复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b22	Res	保留位	读出时为“0”	R
b21~b6	CNT[21:6]	计数值	当前定时器的计数值	RW
b5~b0	CNT[5:0]	计数值	读出时为“0”	R

19.6.1.2 刷新值寄存器 (HRPWM<t>_UPDAR)

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b22	Res	保留位	读出时为“0”	R
b21~b6	UPDA[21:6]	刷新值	设定要更新进时器的刷新值	RW
b5~b0	UPDA[5:0]	刷新值	读出时为“0”	R

19.6.1.3 高分辨率通用周期基准值寄存器 (HRPWM<t>_HRPERmR, m=A~B)

复位值: 0x003F FFC0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res										HRPERA-B[21:16]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HRPERA-B[15:0]															
位/位域	标记	位名	功能	读写											
b31~b22	Res	保留位	读时读出0, 写时请写0	RW											
b21~b0	HRPERA-B	高分辨率计数周期	设定每轮计数的计数周期值以及对应缓存值 HRPWM_HRPmR的值必须大于0xC0, 最大值不得大于0x3FFFC0	RW											

高分辨率通用周期寄存器的值按照下式进行计算:

$$\begin{cases} \text{HRPERAR} = (\text{fhigh_res}) / \text{fPWM} - 64 & (\text{锯齿波}) \\ \text{HRPERAR} = (\text{fhigh_res}) / (2 * \text{fPWM}) & (\text{三角波}) \end{cases}$$

其中, fPWM 为输出 PWM 波形的频率, fhigh_res 为计数器的分辨率 (具体参考【计数器时钟】章节)

19.6.1.4 高分辨率通用比较基准值寄存器 (HRPWM<t>_HRGCMmR, m=A~H)

复位值: 0x003F FFC0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res										HRGCMa-H [21:16]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HRGCMa-H [15:0]															
位/位域	标记	位名	功能	读写											
b31~b22	Res	保留位	读时读出0, 写时请写0	RW											
b21~b0	HRGCMa-H	高分辨率计数比较基准值	比较基准值设定, 与计数值相匹配信号有效 HRPWM_HRGCMmR的值必须大于0xC0, 最大值不得大于0x3FFFC0	RW											

19.6.1.5 专用比较基准值寄存器 (HRPWM<t>_SCMmR, m=A~D)

复位值: 0x003F FFC0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Res										SCMA~D[21:16]						
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
SCMA~D[15:0]																
位/位域	标记	位名	功能													读写
b31~b22	Res	保留位	读出时为“0”													R
b21~b6	SCMA~D[21:6]	专用比较基准值	设定比较基准值及缓存值													RW
b5~b0	SCMA~D[5:0]	专用比较基准值	读出时为“0” 该位域不支持写													R

19.6.1.6 捕获值寄存器 (HRPWM<t>_CAPmR, m=A~B)

复位值: 0x003F FFC0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Res										CAPDIRA-B	CAPA-B [21:16]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
CAPA-B[15:6]										CAPA-B[5:0]						
位/位域	标记	位名	功能													读写
b31~b23	Res	保留位	读出时为“0”													RW
b22	CAPDIRA-B	捕获方向	0: 发生捕获时处于向上计数区间 1: 发生捕获时处于向下计数区间													R
b21~b6	CAPA-B [21:6]	捕获基准值	捕获值													RW
b5~b0	CAPA-B [5:0]	捕获基准值	写入时写“0”，读出时为“0”													RW

19.6.1.7 高分辨率死区时间基准值寄存器 (HRPWM<t>_HRDTU/DmR, m=A~B)

复位值: 0x003F FFC0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Res										HRDTU/DA-B [21:16]						
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
HRDTU/DA-B [15:0]																
位/位域	标记	位名	功能													读写
b31~b22	Res	保留位	读时读出0, 写时请写0													RW
b21~b0	HRDTU/DA~B	死区时间值	死区时间设定以及缓存值 HRPWM_HRDTUmR的值必须大于0xC0, 最大值不得大于0x3FFFC0													RW

19.6.1.8 中断控制寄存器 (HRPWM<t>_ICONR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res				INTENSBD	INTENSBU	INTENSAD	INTENSAU
b15	b14	b13	b12	b11	b10	b9	b8
INTENCAPB	INTENCAPA	Res					
b7	b6	b5	b4	b3	b2	b1	b0
INTENUDF	INTENOVF	INTENF	INTENE	INTEND	INTENC	INTENB	INTENA

位/位域	标记	位名	功能	读写
b31~b20	Res	保留位	读出时为“0”,写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW
b15	INTENCAPB	捕获中断使能B	0: 捕获事件B发生时, 该中断无效 1: 捕获事件B发生时, 该中断使能	RW
b14	INTENCAPA	捕获中断使能A	0: 捕获事件A发生时, 该中断无效 1: 捕获事件A发生时, 该中断使能	RW
b13~b8	Res	保留位	读出时为“0”,写入时写“0”	RW
b7	INTENUDF	零点匹配中断使能	0: 计数值等于0时, 该中断无效 1: 计数值等于0时, 该中断使能	RW
b6	INTENOVF	周期匹配中断使能	0: 计数值等于周期时, 该中断无效 1: 计数值等于周期时, 该中断使能	RW
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	RW
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	RW
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	RW
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	RW
b1	INTENB	计数匹配中断使能B	0: GCMBR寄存器与计数值相等时, 该中断无效 1: GCMBR寄存器与计数值相等时, 该中断使能	RW
b0	INTENA	计数匹配中断使能A	0: GCMAR寄存器与计数值相等时, 该中断无效 1: GCMAR寄存器与计数值相等时, 该中断使能	RW

19.6.1.9 中断控制缓存寄存器 (HRPWM<t>_BICONR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved							
b23	b22	b21	b20	b19	b18	b17	b16
Reserved				INTENSBD	INTENSBU	INTENSAD	INTENSAU
b15	b14	b13	b12	b11	b10	b9	b8
INTENCAPB	INTENCAPA	Reserved					
b7	b6	b5	b4	b3	b2	b1	b0
INTENUDF	INTENOVF	INTENF	INTENE	INTEND	INTENC	INTENB	INTENA

位/位域	标记	位名	功能	读写
b31~b20	Res	保留位	读出时为“0”,写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW
b15	INTENCAPB	捕获中断使能B	0: 捕获事件B发生时, 该中断无效 1: 捕获事件B发生时, 该中断使能	RW
b14	INTENCAPA	捕获中断使能A	0: 捕获事件A发生时, 该中断无效 1: 捕获事件A发生时, 该中断使能	RW
b13~b8	Res	保留位	读出时为“0”,写入时写“0”	RW
b7	INTENUDF	零点匹配中断使能	0: 计数器计数零点匹配时, 该中断无效 1: 计数器计数零点匹配时, 该中断使能	RW
b6	INTENOVF	周期匹配中断使能	0: 计数器计数周期匹配时, 该中断无效 1: 计数器计数周期匹配时, 该中断使能	RW
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	RW
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	RW
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	RW
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	RW
b1	INTENB	计数匹配中断使能B	0: GCMBR寄存器与计数值相等时, 该中断无效 1: GCMBR寄存器与计数值相等时, 该中断使能	RW
b0	INTENA	计数匹配中断使能A	0: GCMAR寄存器与计数值相等时, 该中断无效 1: GCMAR寄存器与计数值相等时, 该中断使能	RW

19.6.1.10 死区控制寄存器 (HRPWM<t>_DCONR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								DTB TRD	DTB TRU	DTB END	DTB ENU	Res		SEPA	DTC EN

位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	读出时为“0”,写入时写“0”	RW
b7	DTBTRD	死区时间值缓存传送时间设定D	0: 零点缓存传送点时, 缓存值不传送 1: 零点缓存传送点时, 发生一次缓存值传送	RW
b6	DTBTRU	死区时间值缓存传送时间设定U	0: 周期缓存传送点时, 缓存值不传送 1: 周期缓存传送点时, 发生一次缓存值传送	RW
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 (HRDTDBR->HRDTDAR)	RW
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 (HRDTUBR->HRDTUAR)	RW
b3~b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1	SEPA	分离设定	0: HRDTUAR和HRDTDAR分别设定 1: HRDTUAR的值和HRDTDAR的值自动相等	RW
b0	DTCEN	死区功能	0: 死区功能无效 1: 死区功能有效	RW

19.6.1.11 有效周期寄存器 (HRPWM<t>_VPERR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								PCNTS[4:0]				PCNTE[1:0]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						SP PERIB	SP PERIA	Res							

位/位域	标记	位名	功能	读写
b31~b23	Res	保留位	读出时为“0”,写入时写“0”	RW
			00000: 有效周期选择功能无效 00001: 每隔1个周期有效一次 00010: 每隔2个周期有效一次 00011: 每隔3个周期有效一次 00100: 每隔4个周期有效一次 00101: 每隔5个周期有效一次 00110: 每隔6个周期有效一次 00111: 每隔7个周期有效一次 01000: 每隔8个周期有效一次 01001: 每隔9个周期有效一次 01010: 每隔10个周期有效一次 01011: 每隔11个周期有效一次 01100: 每隔12个周期有效一次 01101: 每隔13个周期有效一次 01110: 每隔14个周期有效一次 01111: 每隔15个周期有效一次 11110: 每隔30个周期有效一次 11111: 每隔31个周期有效一次	
b22~b18	PCNTS[4:0]	有效周期选择		RW
			00: 有效周期选择功能无效 01: 锯齿波计数值等于周期值或锯齿波硬件清零或三角波计数值等于0做为计数条件 10: 锯齿波计数值等于周期值或锯齿波硬件清零或三角波计数值等于周期值做为计数条件 11: 锯齿波计数值等于周期值或锯齿波硬件清零或三角波计数值等于周期值、计数值等于0做为计数条件	
b17~b16	PCNTE[1:0]	有效周期计数条件选择		RW
b15~b10	Res	保留位	读出时为“0”,写入时写“0”	RW
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	RW
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	RW
b7~b0	Res	保留位	读出时为“0”,写入时写“0”	RW

19.6.1.12 状态标志寄存器 (HRPWM<t>_STFLR1)

复位值: 0x8000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DIRF	Res			CAPBF	CAPAF	VPERNUM[4:0]				Res					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			CMSBDF	CMSBUF	CMSADF	CMSAUF	Res	UDFF	OVFF	CMFF	CMEF	CMDF	CMCF	CMBF	CMAF

位/位域	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	Res	保留位	读出时为“0”,写入时写“0”	R
b27	CAPBF	捕获事件B	0: 捕获事件B未发生 1: 捕获事件B发生 对该位写0清0, 写1无效	RW
b26	CAPAF	捕获事件A	0: 捕获事件A未发生 1: 捕获事件A发生 对该位写0清0, 写1无效	RW
b25~b21	VPERNUM[4:0]	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	Res	保留位	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 对该位写0清0, 写1无效	RW
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 对该位写0清0, 写1无效	RW
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 对该位写0清0, 写1无效	RW
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 对该位写0清0, 写1无效	RW
b8	Res	保留位	读出时为“0”,写入时写“0”	R
b7	UDFF	零点匹配	0: 未发生零点匹配 1: 发生零点匹配 对该位写0清0, 写1无效	RW
b6	OVFF	周期匹配	0: 未发生周期匹配 1: 发生周期匹配 对该位写0清0, 写1无效	RW
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 对该位写0清0, 写1无效	RW
b4	CMEF	计数匹配E	0: GCMEF寄存器的值与计数值不相等 1: GCMEF寄存器的值与计数值相等 对该位写0清0, 写1无效	RW

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 对该位写0清0, 写1无效	RW
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 对该位写0清0, 写1无效	RW
b1	CMBF	计数匹配B	0: GCMBR寄存器的值与计数值不相等 1: GCMBR寄存器的值与计数值相等 对该位写0清0, 写1无效	RW
b0	CMAF	计数匹配A	0: GCMAR寄存器的值与计数值不相等 1: GCMAR寄存器的值与计数值相等 对该位写0清0, 写1无效	RW

19.6.1.13 状态标志寄存器 (HRPWM<t>_STFLR2)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															OSTO VFF
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PHSC MP5F	PHSC MP4F	PHSC MP3F	PHSC MP2F	PHSC MP1F	Res					OBST AT	OAST AT	Res	DLYP RT	DLYID LEB	DLYID LEA

位/位域	标记	位名	功能	读写
b31~b17	Res	保留位	读出时为“0”,写入时写“0”	R
b16	OSTOVFF	单次模式周期点匹配	0: 单次计数模式下, 单次计数完成点未发生 1: 单次计数模式下, 单次计数完成点发生 该单次计数完成点指锯齿波计数值等于周期值三角波向下计数且计数值等于1 对该位写0清0, 写1无效	RW
b15	PHSCMP5F	相位匹配5	0: 相位匹配事件5未发生 1: 相位匹配事件5发生 仅支持单元1 对该位写0清0, 写1无效	RW
b14	PHSCMP4F	相位匹配4	0: 相位匹配事件4未发生 1: 相位匹配事件4发生 仅支持单元1 对该位写0清0, 写1无效	RW
b13	PHSCMP3F	相位匹配3	0: 相位匹配事件3未发生 1: 相位匹配事件3发生 仅支持单元1 对该位写0清0, 写1无效	RW
b12	PHSCMP2F	相位匹配2	0: 相位匹配事件2未发生 1: 相位匹配事件2发生 仅支持单元1 对该位写0清0, 写1无效	RW
b11	PHSCMP1F	相位匹配1	0: 相位匹配事件1未发生 1: 相位匹配事件1发生 仅支持单元1 对该位写0清0, 写1无效	RW
b10~b6	Res	保留位	读出时为“0”,写入时写“0”	R
b5	OBSTAT	进入延迟空闲时B的状态	0: 进入时为低电平 1: 进入时为高电平	R
b4	OASTAT	进入延迟空闲时A的状态	0: 进入时为低电平 1: 进入时为高电平	R
b3	Res	保留位	读出时为“0”,写入时写“0”	R
b2	DLYPRT	延迟空闲状态	0: 延迟空闲进入未发生 1: 延迟空闲进入发生 读出1后对该位写0清0,写1无效	RW
b1	DLYIDLEB	输出B处于延迟空闲状态	0: 输出B不处于延迟空闲状态 1: 输出B处于延迟空闲状态	R

b0	DLYIDLEA	输出A处于延迟空闲状态	0: 输出A不处于延迟空闲状态 1: 输出A处于延迟空闲状态	R
----	----------	-------------	-----------------------------------	---

19.6.1.14 硬件启动事件选择寄存器 1 (HRPWM<t>_HSTAR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								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
Res				HSTA 11	HSTA 10	HSTA 9	HSTA 8	STAS	Res						

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”,写入时写“0”	RW
b23	HSTA23	硬件启动条件23	条件: HRPWM_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	RW
b22	HSTA22	硬件启动条件22	条件: HRPWM_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	RW
b21	HSTA21	硬件启动条件21	条件: HRPWM_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	RW
b20	HSTA20	硬件启动条件20	条件: HRPWM_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	RW
b19	HSTA19	硬件启动条件19	条件: HRPWM_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	RW
b18	HSTA18	硬件启动条件18	条件: HRPWM_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	RW
b17	HSTA17	硬件启动条件17	条件: HRPWM_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	RW
b16	HSTA16	硬件启动条件16	条件: HRPWM_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	RW
b15~b12	Res	保留位	读出时为“0”,写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW

			1: 条件匹配时, 硬件启动有效	
			0: 硬件启动无效	
b7	STAS	硬件启动使能	1: 硬件启动有效	RW
			注: 硬件启动有效时, SSTAR的设置无效	
b6~b0	Res	保留位	读出时为“0”, 写入时写“0”	RW

19.6.1.15 硬件清零事件选择寄存器 1 (HRPWM<t>_HCLRR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								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
Res				HCLE 11	HCLE 10	HCLE 9	HCLE 8	CLES	Res						

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”,写入时写“0”	RW
b23	HCLE23	硬件清零条件23	条件: HRPWM_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	RW
b22	HCLE22	硬件清零条件22	条件: HRPWM_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	RW
b21	HCLE21	硬件清零条件21	条件: HRPWM_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	RW
b20	HCLE20	硬件清零条件20	条件: HRPWM_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	RW
b19	HCLE19	硬件清零条件19	条件: HRPWM_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	RW
b18	HCLE18	硬件清零条件18	条件: HRPWM_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	RW
b17	HCLE17	硬件清零条件17	条件: HRPWM_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	RW
b16	HCLE16	硬件清零条件16	条件: HRPWM_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	RW
b15~b12	Res	-	读出时为“0”,写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW

			1: 条件匹配时, 硬件清零有效	
			0: 硬件清零无效	
b7	CLES	硬件清零使能	1: 硬件清零有效 注: 硬件清零有效时, SCLRR的设置无效	RW
b6~b0	Res	-	读出时为“0”, 写入时写“0”	RW

19.6.1.16 硬件捕获事件选择寄存器 1 (HRPWM<t>_HCPAR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								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
Res				HCP A11	HCP A10	HCP A9	HCP A8	Res							

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”,写入时写“0”	RW
b23	HCPA23	硬件捕获A条件23	条件: HRPWM_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	RW
b22	HCPA22	硬件捕获A条件22	条件: HRPWM_TRIGD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	RW
b21	HCPA21	硬件捕获A条件21	条件: HRPWM_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	RW
b20	HCPA20	硬件捕获A条件20	条件: HRPWM_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	RW
b19	HCPA19	硬件捕获A条件19	条件: HRPWM_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	RW
b18	HCPA18	硬件捕获A条件18	条件: HRPWM_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	RW
b17	HCPA17	硬件捕获A条件17	条件: HRPWM_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	RW
b16	HCPA16	硬件捕获A条件16	条件: HRPWM_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	RW
b15~b12	Res	保留位	读出时为“0”,写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW

1: 条件匹配时, 硬件捕获A有效

b7~b0

Res

保留位

读出时为“0”, 写入时写“0”

RW

19.6.1.17 硬件捕获事件选择寄存器 1 (HRPWM<t>_HCPBR1)

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								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
Res				HCP B11	HCP B10	HCP B9	HCP B8	Res							

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”,写入时写“0”	RW
b23	HCPB23	硬件捕获B条件23	条件: HRPWM_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	RW
b22	HCPB22	硬件捕获B条件22	条件: HRPWM_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	RW
b21	HCPB21	硬件捕获B条件21	条件: HRPWM_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	RW
b20	HCPB20	硬件捕获B条件20	条件: HRPWM_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	RW
b19	HCPB19	硬件捕获B条件19	条件: HRPWM_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	RW
b18	HCPB18	硬件捕获B条件18	条件: HRPWM_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	RW
b17	HCPB17	硬件捕获B条件17	条件: HRPWM_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	RW
b16	HCPB16	硬件捕获B条件16	条件: HRPWM_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	RW
b15~b12	Res	保留位	读出时为“0”,写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW

1: 条件匹配时, 硬件捕获B有效				
b7~b0	Res	保留位	读出时为“0”,写入时写“0”	RW

19.6.1.18 硬件捕获事件选择寄存器 2 (HRPWM<t>_HCPAR2)

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					HCP AEN	HCPA EEV1 0	HCPA EEV9	HCPA EEV8	HCPA EEV7	HCPA EEV6	HCPA EEV5	HCPA EEV4	HCPA EEV3	HCPA EEV2	HCPA EEV1

位/位域	标记	位名	功能	读写
b31~b11	Res	保留位	读出时为“0”,写入时写“0”	RW
b10	HCPAEN	硬件捕获A使能	0: 硬件捕获A无效 1: 硬件捕获A有效	RW
b9	HCPAEEV10	外部事件10硬件捕获A使能	条件: 外部事件10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	RW
b8	HCPAEEV9	外部事件9硬件捕获A使能	条件: 外部事件9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	RW
b7	HCPAEEV8	外部事件8硬件捕获A使能	条件: 外部事件8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	RW
b6	HCPAEEV7	外部事件7硬件捕获A使能	条件: 外部事件7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	RW
b5	HCPAEEV6	外部事件6硬件捕获A使能	条件: 外部事件6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	RW
b4	HCPAEEV5	外部事件4硬件捕获A使能	条件: 外部事件5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	RW
b3	HCPAEEV4	外部事件4硬件捕获A使能	条件: 外部事件4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	RW
b2	HCPAEEV3	外部事件3硬件捕获A使能	条件: 外部事件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	RW
b1	HCPAEEV2	外部事件2硬件捕获A使能	条件: 外部事件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	RW
b0	HCPAEEV1	外部事件1硬件捕获A使能	条件: 外部事件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	RW

19.6.1.19 硬件捕获事件选择寄存器 2 (HRPWM<t>_HCPBR2)

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					HCP BEN	HCPB EEV1 0	HCPB EEV9	HCPB EEV8	HCPB EEV7	HCPB EEV6	HCPB EEV5	HCPB EEV4	HCPB EEV3	HCPB EEV2	HCPB EEV1

位/位域	标记	位名	功能	读写
b31~b11	Res	保留位	读出时为“0”,写入时写“0”	RW
b10	HCPBEN	硬件捕获B使能	0: 硬件捕获B无效 1: 硬件捕获B有效	RW
b9	HCPBEEV10	外部事件10硬件捕获B使能	条件: 外部事件10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	RW
b8	HCPBEEV9	外部事件9硬件捕获B使能	条件: 外部事件9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	RW
b7	HCPBEEV8	外部事件8硬件捕获B使能	条件: 外部事件8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	RW
b6	HCPBEEV7	外部事件7硬件捕获B使能	条件: 外部事件7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	RW
b5	HCPBEEV6	外部事件6硬件捕获B使能	条件: 外部事件6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	RW
b4	HCPBEEV5	外部事件4硬件捕获B使能	条件: 外部事件5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	RW
b3	HCPBEEV4	外部事件4硬件捕获B使能	条件: 外部事件4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	RW
b2	HCPBEEV3	外部事件3硬件捕获B使能	条件: 外部事件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	RW
b1	HCPBEEV2	外部事件2硬件捕获B使能	条件: 外部事件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	RW
b0	HCPBEEV1	外部事件1硬件捕获B使能	条件: 外部事件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	RW

19.6.1.20 硬件启动事件选择寄存器 2 (HRPWM<t>_HSTAR2)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				HSTA SCMB 6	HSTA SCMB 5	HSTA SCMB 4	HSTA SCMB 3	HSTA SCMB 2	HSTA SCMA 1	HSTA GCM B6	HSTA GCM A6	HSTA GCM B5	HSTA GCM A5	HSTA GCM B4	HSTA GCM A4
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HSTA GCM B3	HSTA GCM A3	HSTA GCM B2	HSTA GCM A2	HSTA GCM B1	HSTA GCM A1	HSTA EXEV 10	HSTA EXEV 9	HSTA EXEV 8	HSTA EXEV 7	HSTA EXEV 6	HSTA EXEV 5	HSTA EXEV 4	HSTA EXEV 3	HSTA EXEV 2	HSTA EXEV 1

位/位域	标记	位名	功能	读写
b31~b28	Res	保留位	读出时为“0”,写入时写“0”	RW
b27	HSTASCMB6	单元6专用比较匹配事件B启动使能	条件: 单元6专用比较匹配事件B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	RW
b26	HSTASCMB5	单元5专用比较匹配事件B启动使能	条件: 单元5专用比较匹配事件B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	RW
b25	HSTASCMB4	单元4专用比较匹配事件B启动使能	条件: 单元4专用比较匹配事件B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	RW
b24	HSTASCMB3	单元3专用比较匹配事件B启动使能	条件: 单元3专用比较匹配事件B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	RW
b23	HSTASCMB2	单元2专用比较匹配事件B启动使能	条件: 单元2专用比较匹配事件B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	RW
b22	HSTASCMB1	单元1专用比较匹配事件B启动使能	条件: 单元1比较匹配事件B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	RW
b21	HSTAGCMB6	单元6比较匹配事件B启动使能	条件: 单元6比较匹配事件B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	RW
b20	HSTAGCMA6	单元6比较匹配事件A启动使能	条件: 单元6比较匹配事件A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	RW
b19	HSTAGCMB5	单元5比较匹配事件B启动使能	条件: 单元5比较匹配事件B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	RW
b18	HSTAGCMA5	单元5比较匹配事件A启动使能	条件: 单元5比较匹配事件A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	RW
b17	HSTAGCMB4	单元4比较匹配事件B启动使能	条件: 单元4比较匹配事件B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	RW
b16	HSTAGCMA4	单元4比较匹配事件A启动使能	条件: 单元4比较匹配事件A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	RW

b15	HSTAGCMB3	单元3比较匹配事件B启动使能	条件：单元3比较匹配事件B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	RW
b14	HSTAGCMA3	单元3比较匹配事件A启动使能	条件：单元3比较匹配事件A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	RW
b13	HSTAGCMB2	单元2比较匹配事件B启动使能	条件：单元2比较匹配事件B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	RW
b12	HSTAGCMA2	单元2比较匹配事件A启动使能	条件：单元2比较匹配事件A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	RW
b11	HSTAGCMB1	单元1比较匹配事件B启动使能	条件：单元1比较匹配事件B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	RW
b10	HSTAGCMA1	单元1比较匹配事件A启动使能	条件：单元1比较匹配事件A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	RW
b9	HSTAEXEV10	外部事件10启动使能	条件：外部事件10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	RW
b8	HSTAEXEV9	外部事件9启动使能	条件：外部事件9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	RW
b7	HSTAEXEV8	外部事件8启动使能	条件：外部事件8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	RW
b6	HSTAEXEV7	外部事件7启动使能	条件：外部事件7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	RW
b5	HSTAEXEV6	外部事件6启动使能	条件：外部事件6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	RW
b4	HSTAEXEV5	外部事件5启动使能	条件：外部事件5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	RW
b3	HSTAEXEV4	外部事件4启动使能	条件：外部事件4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	RW
b2	HSTAEXEV3	外部事件3启动使能	条件：外部事件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	RW
b1	HSTAEXEV2	外部事件2启动使能	条件：外部事件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	RW
b0	HSTAEXEV1	外部事件1启动使能	条件：外部事件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	RW

19.6.1.21 硬件清零事件选择寄存器 2 (HRPWM<t>_HCLRR2)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				HCLR SCMB 6	HCLR SCMB 5	HCLR SCMB 4	HCLR SCMB 3	HCLR SCMB 2	HCLR SCMB 1	HCLR GCM B6	HCLR GCM A6	HCLR GCM B5	HCLR GCM A5	HCLR GCM B4	HCLR GCM A4
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HCLR GCM B3	HCLR GCM A3	HCLR GCM B2	HCLR GCM A2	HCLR GCM B1	HCLR GCM A1	HCLR EXEV 10	HCLR EXEV 9	HCLR EXEV 8	HCLR EXEV 7	HCLR EXEV 6	HCLR EXEV 5	HCLR EXEV 4	HCLR EXEV 3	HCLR EXEV 2	HCLR EXEV 1

位/位域	标记	位名	功能	读写
b31~b28	Res	保留位	读出时为“0”,写入时写“0”	RW
b27	HCLRSCMB6	单元6专用比较匹配事件B 清零使能	条件: 单元6专用比较匹配事件B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	RW
b26	HCLRSCMB5	单元5专用比较匹配事件B 清零使能	条件: 单元5专用比较匹配事件B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	RW
b25	HCLRSCMB4	单元4专用比较匹配事件B 清零使能	条件: 单元4专用比较匹配事件B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	RW
b24	HCLRSCMB3	单元3专用比较匹配事件B 清零使能	条件: 单元3专用比较匹配事件B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	RW
b23	HCLRSCMB2	单元2专用比较匹配事件B 清零使能	条件: 单元2专用比较匹配事件B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	RW
b22	HCLRSCMB1	单元1专用比较匹配事件B 清零使能	条件: 单元1专用比较匹配事件B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	RW
b21	HCLRGCM B6	单元6比较匹配事件B清零 使能	条件: 单元6比较匹配事件B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	RW
b20	HCLRGCM A6	单元6比较匹配事件A清零 使能	条件: 单元6比较匹配事件A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	RW
b19	HCLRGCM B5	单元5比较匹配事件B清零 使能	条件: 单元5比较匹配事件B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	RW
b18	HCLRGCM A5	单元5比较匹配事件A清零 使能	条件: 单元5比较匹配事件A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	RW
b17	HCLRGCM B4	单元4比较匹配事件B清零 使能	条件: 单元4比较匹配事件B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	RW
b16	HCLRGCM A4	单元4比较匹配事件A清零 使能	条件: 单元4比较匹配事件A有效 0: 条件匹配时, 硬件清零无效	RW

			1: 条件匹配时, 硬件清零有效	
b15	HCLRGCMB3	单元3比较匹配事件B启动使能	条件: 单元3比较匹配事件B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	RW
b14	HCLRGCM A3	单元3比较匹配事件A清零使能	条件: 单元3比较匹配事件A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	RW
b13	HCLRGCM B2	单元2比较匹配事件B清零使能	条件: 单元2比较匹配事件B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	RW
b12	HCLRGCM A2	单元2比较匹配事件A清零使能	条件: 单元2比较匹配事件A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	RW
b11	HCLRGCM B1	单元1比较匹配事件B清零使能	条件: 单元1比较匹配事件B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	RW
b10	HCLRGCM A1	单元1比较匹配事件A清零使能	条件: 单元1比较匹配事件A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	RW
b9	HCLREXEV10	外部事件10清零使能	条件: 外部事件10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	RW
b8	HCLREXEV9	外部事件9清零使能	条件: 外部事件9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	RW
b7	HCLREXEV8	外部事件8清零使能	条件: 外部事件8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	RW
b6	HCLREXEV7	外部事件7清零使能	条件: 外部事件7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	RW
b5	HCLREXEV6	外部事件6清零使能	条件: 外部事件6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	RW
b4	HCLREXEV5	外部事件5清零使能	条件: 外部事件5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	RW
b3	HCLREXEV4	外部事件4清零使能	条件: 外部事件4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	RW
b2	HCLREXEV3	外部事件3清零使能	条件: 外部事件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	RW
b1	HCLREXEV2	外部事件2清零使能	条件: 外部事件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	RW
b0	HCLREXEV1	外部事件1清零使能	条件: 外部事件1有效 0: 条件匹配时, 硬件清零无效	RW

1: 条件匹配时, 硬件清零有效

19.6.1.22 外部事件滤波偏移值寄存器 (HRPWM<t>_EEFOFFSETmR, m=A~B)

复位值: 0x0040 0000

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
	Res										OFFSET DIR	OFFSET [21:16]					
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
	OFFSET [15:6]										OFFSET[5:0]						
位/位域	标记		位名		功能		读写										
b31~b23	Res		保留位		读出时为“0”		RW										
b22	OFFSETDIR		偏移值计数方向		0: 偏移值处于向下计数区间 1: 偏移值处于向上计数区间		RW										
b21~b6	OFFSET [21:6]		偏移值		消隐/窗口偏移值		RW										
b5~b0	OFFSET[5:0]		偏移值		读出时为“0”		R										

19.6.1.23 外部事件滤波窗口寄存器 (HRPWM<t>_EEFWINmR, m=A~B)

复位值: 0x0040 0000

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
	Res										WINDIR	WIN[21:16]					
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
	WIN[15:6]										WIN[5:0]						
位/位域	标记		位名		功能		读写										
b31~b23	Res		保留位		读出时为“0”		R										
b22	WINDIR		窗口值计数方向		0: 窗口值处于向下计数区间 1: 窗口值处于向上计数区间		RW										
b21~b6	WIN[21:6]		窗口值		消隐/窗口信号窗口值		RW										
b5~b0	WIN[5:0]		窗口值		读出时为“0”		R										

19.6.1.24 外部事件滤波控制寄存器 (HRPWM<t>_EEFLTCR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	EEINTPOL	EE5TMO	EE5FM[3:0]			EE5LAT	EE4TMO	EE4FM[3:0]			EE4LAT	EE3TMO	EE3FM[3]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EE3FM[2:0]			EE3LAT	EE2TM	EE2FM[3:0]			EE2LAT	EE1TMO	EE1FM[3:0]			EE1LAT		

位/位域	标记	位名	功能	读写
b31~b30	Res	保留位	读出时为“0”,写入时写“0”	RW
b30	EEINTPOL	外部事件滤波初始电平	0: 初始电平为低电平 1: 初始电平为高电平	RW
b29	EE5TMO	外部事件超时功能使能	参考EE1TMO说明	RW
b28~b25	EE5FM[3:0]	硬件触发事件5滤波模式	参考EE1FM[3:0]说明	RW
b24	EE5LAT	外部事件5存储功能使能	参考EE1LAT说明	RW
b23	EE4TMO	外部事件超时功能使能	参考EE1TMO说明	RW
b22~b19	EE4FM[3:0]	硬件触发事件4滤波模式	参考EE1FM[3:0]说明	RW
b18	EE4LAT	外部事件4存储功能使能	参考EE1LAT说明	RW
b17	EE3TMO	外部事件超时功能使能	参考EE1TMO说明	RW
b16~b13	EE3FM[3:0]	硬件触发事件3滤波模式	参考EE1FM[3:0]说明	RW
b12	EE3LAT	外部事件3存储功能使能	参考EE1LAT说明	RW
b11	EE2TMO	外部事件超时功能使能	参考EE1TMO说明	RW
b10~b7	EE2FM[3:0]	硬件触发事件2滤波模式	参考EE1FM[3:0]说明	RW
b6	EE2LAT	外部事件2存储功能使能	参考EE1LAT说明	RW
b5	EE1TMO	外部事件超时功能使能	0: 超时功能不使能 1: 超时功能使能	RW
			0000: 滤波模式禁止	
			0001: 消隐模式。消隐时间是指本单元消隐信号低电平持续的时间。	
			0010: 消隐模式。消隐时间是指单元1消隐信号低电平持续的时间。	
			0011: 消隐模式。消隐信号为单元1的HRPWM_1_B。	
			0100: 消隐模式。消隐时间是指单元2消隐信号低电平持续的时间	
			0101: 消隐模式。消隐信号为单元2的HRPWM_2_B。	
			0110: 消隐模式。消隐时间是指单元3消隐信号低电平持续的时间。	
			0111: 消隐模式。消隐信号为单元3的HRPWM_3_B。	
			1000: 消隐模式。消隐时间是指单元4消隐信号低电平持续的时间。	
			1001: 消隐模式。消隐信号为单元4的HRPWM_4_B。	
			1010: 消隐模式。消隐时间是指单元5消隐信号低电平持续的时间。	
			1011: 消隐模式。消隐信号为单元5的HRPWM_5_B。	
b4~b1	EE1FM[3:0]	外部事件1滤波模式	0101: 消隐模式。消隐信号为单元2的HRPWM_2_B。 0110: 消隐模式。消隐时间是指单元3消隐信号低电平持续的时间。 0111: 消隐模式。消隐信号为单元3的HRPWM_3_B。 1000: 消隐模式。消隐时间是指单元4消隐信号低电平持续的时间。 1001: 消隐模式。消隐信号为单元4的HRPWM_4_B。 1010: 消隐模式。消隐时间是指单元5消隐信号低电平持续的时间。 1011: 消隐模式。消隐信号为单元5的HRPWM_5_B。	RW

			1100: 消隐模式。消隐时间是指单元6消隐信号低电平持续的时间。	
			1101: 消隐模式。消隐信号为单元6的HRPWM_6_B。	
			1110: 窗口模式。窗口时间是指本单元窗口信号高电平持续的时间。	
			1111: 窗口模式。窗口信号来自其他单元, TMRWIN源 (参考表 19-8)。	
			通过设定HRPWM<t>_GCONR1的EEFM和EEFREF位可以设定滤波模式2, 具体参考表 19-7和表 19-9。	
<hr/>				
b0	EE1LAT	外部事件1存储功能使能	0: 事件存储功能禁止 1: 事件存储功能使能	RW

19.6.1.25 外部事件滤波控制寄存器 (HRPWM<t>_EEFLTCR2)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res		EE10TMO	EE10FM[3:0]			EE10LAT	EE9TMO	EE9FM[3:0]			EE9LAT	EE8TMO	EE8FM[3:0]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EE8FM[2:0]		EE8LAT	EE7TMO	EE7FM[3:0]			EE7LAT	EE6TMO	EE6FM[3:0]			EE6LAT			

位/位域	标记	位名	功能	读写
b31~b30	Res	保留位	读出时为“0”,写入时写“0”	RW
b29	EE10TMO	外部事件超时功能使能	参考EE1TMO说明	RW
b28~b25	EE10FM[3:0]	硬件触发事件10滤波模式	参考EE1FM[3:0]说明	RW
b24	EE10LAT	外部事件10存储功能使能	参考EE1LAT说明	RW
b23	EE9TMO	外部事件超时功能使能	参考EE1TMO说明	RW
b22~b19	EE9FM[3:0]	硬件触发事件9滤波模式	参考EE1FM[3:0]说明	RW
b18	EE9LAT	外部事件9存储功能使能	参考EE1LAT说明	RW
b17	EE8TMO	外部事件超时功能使能	参考EE1TMO说明	RW
b16~b13	EE8FM[3:0]	硬件触发事件8滤波模式	参考EE1FM[3:0]说明	RW
b12	EE8LAT	外部事件8存储功能使能	参考EE1LAT说明	RW
b11	EE7TMO	外部事件超时功能使能	参考EE1TMO说明	RW
b10~b7	EE7FM[3:0]	硬件触发事件7滤波模式	参考EE1FM[3:0]说明	RW
b6	EE7LAT	外部事件7存储功能使能	参考EE1LAT说明	RW
b5	EE6TMO	外部事件超时功能使能	参考EE1TMO说明	RW
b4~b1	EE6FM[3:0]	外部事件6滤波模式	参考EE1FM[3:0]说明	RW
b0	EE6LAT	外部事件6存储功能使能	参考EE1LAT说明	RW

19.6.1.26 空闲控制寄存器 (HRPWM<t>_IDLECR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res						FOLLOW	INTENDLYPRT	Res				DLYCHB	DIDL B	IDLES B	IDLE BMB
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res		DLYSTRG	DLYEVSEL[2:0]			DLYPRTEN	Res				DLYCHA	DIDLA	IDLES A	IDLE BMA	

位/位域	标记	位名	功能	读写
b31~b26	Res	保留位	读出时为“0”,写入时写“0”	RW
b25	FOLLOW	跟随使能	0: 跟随不使能 1: 跟随使能	RW
b24	INTENDLYPRT	延迟空闲进入中断使能	0: 延迟空闲进入中断不使能 1: 延迟空闲进入中断使能	RW
b23~b20	Res	保留位	读出时为“0”,写入时写“0”	RW
b19	DLYCHB	延迟空闲输出B使能	0: 输出B不进入延迟空闲 1: 输出B由DLYEVSEL[3:0]选择的触发源触发进入延迟空闲	RW
b18	DIDL B	间隔输出延迟进入使能B	0: 输出立即为空闲状态 1: 输出延迟进入间隔输出空闲状态	RW
b17	IDLES B	空闲时端口状态设定B	0: 空闲时设定为低电平 1: 空闲时设定为高电平	RW
b16	IDLEBMB	间隔输出B使能	0: 输出不受间隔输出影响 1: 间隔输出空闲状态由IDLESB设定	RW
b15~b13	Res	保留位	读出时为“0”,写入时写“0”	RW
b12	DLYSTRG	延迟空闲软件触发	0: 无影响 1: 软件触发延迟空闲 使用中需要先将DLYEVSEL[2: 0]设置为100, 再对该位写1	RW
b11~b9	DLYEVSEL[2:0]	延迟空闲触发源选择	000: 触发源为外部事件6 001: 触发源为外部事件7 010: 触发源为外部事件8 011: 触发源为外部事件9 100: 触发源为软件触发 (对DLYSTRG写1产生一次软件触发)	RW
b8	DLYPRTEN	延迟空闲使能	0: 延迟空闲不使能 1: 延迟空闲使能	RW
b7~b4	Res	保留位	读出时为“0”,写入时写“0”	RW
b3	DLYCHA	延迟空闲输出A使能	0: 输出A不进入延迟空闲 1: 输出A由DLYEVSEL[2:0]选择的触发源触发进入延迟空闲	RW
b2	DIDLA	间隔输出延迟进入使能A	0: 输出立即为空闲状态 1: 输出延迟进入间隔输出空闲状态	RW
b1	IDLES A	空闲时端口状态设定A	0: 空闲时设定为低电平 1: 空闲时设定为高电平	RW
b0	IDLEBMA	间隔输出A使能	0: 输出不受间隔输出影响 1: 间隔输出空闲状态由IDLES A设定	RW

19.6.1.27 通用控制寄存器 (HRPWM<t>_GCONR)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b10	Res	保留位	读出时为“0”,写入时写“0”	RW
b8	OVSTP	计数溢出停止控制	0: 三角波模式在向下计数计数值等于1或锯齿波模式计数值等于周期值, 继续计数 1: 三角波模式下, 向下计数计数值等于1, 停止计数 锯齿波模式下, 计数器在计数值等于周期值后, 停止计数	RW
b7~b3	Res	保留位	读出时为“0”,写入时写“0”	RW
b2	MODE	计数模式	0: 锯齿波模式 1: 三角波模式	RW
b1	Res	保留位	读出时为“0”,写入时写“0”	RW
b0	START	定时器启动	0: 定时器关闭 1: 定时器启动 注: 该位在软件停止条件有效时, 会自动变为0	RW

19.6.1.28 通用控制寄存器 (HRPWM<t>_GCONR1, t=1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								BFSEL[2:0]			CMSCBDEN	CMSCBUEN	CMSCADEN	CMSCAUEN	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				PRDSEL[1:0]		Res	EEFREF	EEFM	Res	HALF	SWAPMD	SWAPEN	INVCBEN	INVCANEN	

位/位域	标记	位名	功能	读写
b31~b23	Res	保留位	读出时为“0”,写入时写“0”	RW
b22-b20	BFSEL[2:0]	计数中全局缓存完成标志位 置位使能	BFSEL[0] : 0: 周期缓存传送点发生时不置位 1: 周期缓存传送点发生时置位 BFSEL[1] : 0: 零点缓存传送点发生时不置位 1: 零点缓存传送点发生时置位 BFSEL[2] : 0: 单次缓存传送点发生时不置位 1: 单次缓存传送点发生时置位 点	RW
b19	CMSCBDEN	专用向下计数输出事件使能B	0: 向下计数期间, SCMBR寄存器和计数值相等时, 事件输出无效 1: 向下计数期间, SCMBR寄存器和计数值相等时, 事件输出有效	RW
b18	CMSCBUEN	专用向上计数输出事件使能B	0: 向上计数期间, SCMBR寄存器和计数值相等时, 事件输出无效 1: 向上计数期间, SCMBR寄存器和计数值相等时, 事件输出有效	RW
b17	CMSCADEN	专用向下计数输出事件使能A	0: 向下计数期间, SCMAR寄存器和计数值相等时, 事件输出无效 1: 向下计数期间, SCMAR寄存器和计数值相等时, 事件输出有效	RW
b16	CMSCAUEN	专用向上计数输出事件使能A	0: 向上计数期间, SCMAR寄存器和计数值相等时, 事件输出无效 1: 向上计数期间, SCMAR寄存器和计数值相等时, 事件输出有效	RW
b15~b11	Res	保留位	读出时为“0”,写入时写“0”	RW
b10~b9	PRDSEL[1:0]	完整周期点选择	00: 锯齿波模式计数值等于周期值或者硬件清零, 三角波模式计数值等于0 01: 计数值等于周期值或者锯齿波硬件清零 10: 计数值等于0 11: 计数值等于周期值或者锯齿波硬件清零或者计数值等于0	RW
b8	Res	保留位	读出时为“0”,写入时写“0”	RW
b7	EEFREF	外部事件过滤模式参考点	0: 外部事件过滤模式参考点为周期匹配点	RW

			1: 外部事件过滤模式参考点为零点匹配点	
b6	EEFM	外部事件过滤模式选择	0: 外部事件过滤模式1 1: 外部事件过滤模式2	RW
b5	Res	保留位	读出时为“0”,写入时写“0”	RW
b4	HALF	半波模式使能	0: 半波模式不使能 1: 半波模式使能	RW
b3	SWAPMD	交换模式	0: 完整周期点交换或不交换 1: 立即交换或不交换	RW
b2	SWAPEN	输出A和B交换使能	0: 输出A和B不交换 1: 输出A和输出B交换	RW
b1	INVCBEN	输出B极性取反使能	0: 输出B极性不取反 1: 输出B极性取反	RW
b0	INVCAEN	输出A极性取反使能	0: 输出A极性不取反 1: 输出A极性取反	RW

19.6.1.29 通用控制缓存寄存器 (HRPWM<t>_BGCONR1, t=1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								BFSEL[2:0]		CMSCBDEN	CMSCBUEN	CMSCADEN	CMSCAUEN		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				PRDSEL[1:0]		Res	EEFREF	EEFM	Res	HALF	SWAPMD	SWAPEN	INVCBEN	INVCANEN	

位/位域	标记	位名	功能	读写
b31~b22	Res	保留位	读出时为“0”,写入时写“0”	RW
b22-b20	BFSEL[2:0]	计数中全局缓存完成标志位 置位使能	BFSEL[0] : 0: 周期缓存传送点发生时不置位 1: 周期缓存传送点发生时置位 BFSEL[1] : 0: 零点缓存传送点发生时不置位 1: 零点缓存传送点发生时置位 BFSEL[2] : 0: 单次缓存传送点发生时不置位 1: 单次缓存传送点发生时置位 点	RW
b19	CMSCBDEN	专用向下计数输出事件使能B	0: 向下计数期间, SCMBR寄存器和计数值相等时, 事件输出无效 1: 向下计数期间, SCMBR寄存器和计数值相等时, 事件输出有效	RW
b18	CMSCBUEN	专用向上计数输出事件使能B	0: 向上计数期间, SCMBR寄存器和计数值相等时, 事件输出无效 1: 向上计数期间, SCMBR寄存器和计数值相等时, 事件输出有效	RW
b17	CMSCADEN	专用向下计数输出事件使能A	0: 向下计数期间, SCMAR寄存器和计数值相等时, 事件输出无效 1: 向下计数期间, SCMAR寄存器和计数值相等时, 事件输出有效	RW
b16	CMSCAUEN	专用向上计数输出事件使能A	0: 向上计数期间, SCMAR寄存器和计数值相等时, 事件输出无效 1: 向上计数期间, SCMAR寄存器和计数值相等时, 事件输出有效	RW
b15~b11	Res	保留位	读出时为“0”,写入时写“0”	RW
b10~b9	PRDSEL[1:0]	完整周期点选择	00: 锯齿波模式计数值等于周期值或者硬件清零, 三角波模式计数值等于0 01: 计数值等于周期值或者锯齿波硬件清零 10: 计数值等于0 11: 计数值等于周期值或者锯齿波硬件清零或者计数值等于0	RW
b8	Res	保留位	读出时为“0”,写入时写“0”	RW
b7	EEFREF	外部事件过滤模式参考点	0: 外部事件过滤模式参考点为周期匹配点	RW

			1: 外部事件过滤模式参考点为零点匹配点	
b6	EEFM	外部事件过滤模式选择	0: 外部事件过滤模式1 1: 外部事件过滤模式2	RW
b5	Res	保留位	读出时为“0”,写入时写“0”	RW
b4	HALF	半波模式使能	0: 半波模式不使能 1: 半波模式使能	RW
b3	SWAPMD	交换模式	0: 完整周期点交换或不交换 1: 立即交换或不交换	RW
b2	SWAPEN	输出A和B交换使能	0: 输出A和B不交换 1: 输出A和输出B交换	RW
b1	INVCBEN	输出B极性取反使能	0: 输出B极性不取反 1: 输出B极性取反	RW
b0	INVCAEN	输出A极性取反使能	0: 输出A极性不取反 1: 输出A极性取反	RW

19.6.1.30 通用控制寄存器 (HRPWM<t>_GCONR1, t=2~6)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								BFSEL[3:0]				CMSCBDEN	CMSCBUEN	CMSCADEN	CMSCAUEN
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				PRDSEL[1:0]		Res		EEFREF	EEFM	PRDLK	HALF	SWAPMD	SWAPEN	INVCBEN	INVCANEN

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”,写入时写“0”	RW
b23~b20	BFSEL[3:0]	计数中全局缓存完成标志位置位使能	BFSEL[0] : 0: 周期缓存传送点发生时不置位 1: 周期缓存传送点发生时置位 BFSEL[1] : 0: 零点缓存传送点发生时不置位 1: 零点缓存传送点发生时置位 BFSEL[2] : 0: 单元1单次缓存传送点发生时不置位 1: 单元1单次缓存传送点发生时置位 点 BFSEL[3] : 0: 单元1单次缓存传送点后的本单元锯齿波周期缓存传送点或者三角波两点缓存传送点发生时不置位 1: 单元1单次缓存传送点后的本单元锯齿波周期缓存传送点或者三角波两点缓存传送点发生时置位	RW
b19	CMSCBDEN	专用向下计数输出事件使能B	0: 向下计数期间, SCMBR寄存器和计数值相等时, 事件输出无效 1: 向下计数期间, SCMBR寄存器和计数值相等时, 事件输出有效	RW
b18	CMSCBUEN	专用向上计数输出事件使能B	0: 向上计数期间, SCMBR寄存器和计数值相等时, 事件输出无效 1: 向上计数期间, SCMBR寄存器和计数值相等时, 事件输出有效	RW
b17	CMSCADEN	专用向下计数输出事件使能A	0: 向下计数期间, SCMAR寄存器和计数值相等时, 事件输出无效 1: 向下计数期间, SCMAR寄存器和计数值相等时, 事件输出有效	RW
b16	CMSCAUEN	专用向上计数输出事件使能A	0: 向上计数期间, SCMAR寄存器和计数值相等时, 事件输出无效 1: 向上计数期间, SCMAR寄存器和计数值相等时, 事件输出有效	RW
b15~b11	Res	保留位	读出时为“0”,写入时写“0”	RW
b10~b9	PRDSEL[1:0]	完整周期点选择	00: 锯齿波模式计数值等于周期值或者硬件清零, 三角波模式计数值等于0 01: 计数值等于周期值或者锯齿波硬件清零	RW

			10: 计数值等于0 11: 计数值等于周期值或者锯齿波硬件清零或者计数值等于0	
b8	Res	保留位	读出时为“0”,写入时写“0”	RW
b7	EEFREF	外部事件过滤模式参考点	0: 外部事件过滤模式参考点为周期匹配点 1: 外部事件过滤模式参考点为零点匹配点	RW
b6	EEFM	外部事件过滤模式选择	0: 外部事件过滤模式1 1: 外部事件过滤模式2	RW
b5	PRDLK	周期链接使能	0: 周期链接不使能 1: 周期链接使能 仅支持单元2~6	RW
b4	HALF	半波模式使能	0: 半波模式不使能 1: 半波模式使能	RW
b3	SWAPMD	交换模式	0: 完整周期点交换或不交换 1: 立即交换或不交换	RW
b2	SWAPEN	输出A和B交换使能	0: 输出A和B不交换 1: 输出A和输出B交换	RW
b1	INVCBEN	输出B极性取反使能	0: 输出B极性不取反 1: 输出B极性取反	RW
b0	INVCAEN	输出A极性取反使能	0: 输出A极性不取反 1: 输出A极性取反	RW

19.6.1.31 通用控制缓存寄存器 (HRPWM<t>_BGCONR1, t=2~6)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								BFSEL[3:0]				CMSCBDEN	CMSCBUEN	CMSCADEN	CMSCAUEN
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				PRDSEL[1:0]		Res	EEFR EF	EEFM	PRDLK	HALF	SWAPMD	SWAPEN	INVCBEN	INVCANEN	

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”,写入时写“0”	RW
b23~b20	BFSEL[3:0]	计数中全局缓存完成标志位置位使能	BFSEL[0] :	RW
			0: 周期缓存传送点发生时不置位	
			1: 周期缓存传送点发生时置位	
			BFSEL[1] :	
0: 零点缓存传送点发生时不置位				
1: 零点缓存传送点发生时置位				
BFSEL[2] :				
0: 单元1单次缓存传送点发生时不置位				
1: 单元1单次缓存传送点发生时置位				
BFSEL[3] :				
0: 单元1单次缓存传送点后的本单元锯齿波周期缓存传送点或者三角波两点缓存传送点发生时不置位				
1: 单元1单次缓存传送点后的本单元锯齿波周期缓存传送点或者三角波两点缓存传送点发生时置位				
b19	CMSCBDEN	专用向下计数输出事件使能B	0: 向下计数期间, SCMBR寄存器和计数值相等时, 事件输出无效 1: 向下计数期间, SCMBR寄存器和计数值相等时, 事件输出有效	RW
b18	CMSCBUEN	专用向上计数输出事件使能B	0: 向上计数期间, SCMBR寄存器和计数值相等时, 事件输出无效 1: 向上计数期间, SCMBR寄存器和计数值相等时, 事件输出有效	RW
b17	CMSCADEN	专用向下计数输出事件使能A	0: 向下计数期间, SCMAR寄存器和计数值相等时, 事件输出无效 1: 向下计数期间, SCMAR寄存器和计数值相等时, 事件输出有效	RW
b16	CMSCAUEN	专用向上计数输出事件使能A	0: 向上计数期间, SCMAR寄存器和计数值相等时, 事件输出无效 1: 向上计数期间, SCMAR寄存器和计数值相等时, 事件输出有效	RW
b15~b11	Res	保留位	读出时为“0”,写入时写“0”	RW
b10~b9	PRDSEL[1:0]	完整周期点选择	00: 锯齿波模式计数值等于周期值或者硬件清零, 三角波模式计数值等于0	RW
			01: 计数值等于周期值或者锯齿波硬件清零	

			10: 计数值等于0 11: 计数值等于周期值或者锯齿波硬件清零或者计数值等于0	
b8	Res	保留位	读出时为“0”,写入时写“0”	RW
b7	EEFREF	外部事件过滤模式参考点	0: 外部事件过滤模式参考点为周期匹配点 1: 外部事件过滤模式参考点为零点匹配点	RW
b6	EEFM	外部事件过滤模式选择	0: 外部事件过滤模式1 1: 外部事件过滤模式2	RW
b5	PRDLK	周期链接使能	0: 周期链接不使能 1: 周期链接使能 仅支持单元2~6	RW
b4	HALF	半波模式使能	0: 半波模式不使能 1: 半波模式使能	RW
b3	SWAPMD	交换模式	0: 完整周期点交换或不交换 1: 立即交换或不交换	RW
b2	SWAPEN	输出A和B交换使能	0: 输出A和B不交换 1: 输出A和输出B交换	RW
b1	INVCBEN	输出B极性取反使能	0: 输出B极性不取反 1: 输出B极性取反	RW
b0	INVCAEN	输出A极性取反使能	0: 输出A极性不取反 1: 输出A极性取反	RW

19.6.1.32 缓存控制寄存器 1 (HRPWM<t>_BCONR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								BTR DSPB	BTR USPB	Res	BEN SPB	BTR DSPA	BTR USPA	Res	BEN SPA
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				BTR DP	BTR UP	Res	BEN P	BTR DBF	BTR UBF	Res	BEN BF	BTR DAE	BTR UAE	Res	BEN AE

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”,写入时写“0”	RW
b23	BTRDSPB	专用比较基准值缓存传送时间设定DB	参考BTRDAE说明	RW
b22	BTRUSPB	专用比较基准值缓存传送时间设定UB	参考BTRUAE说明	RW
b21	Res	保留位	读出时为“0”,写入时写“0”	RW
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	RW
b19	BTRDSPA	专用比较基准值缓存传送时间设定DA	参考BTRDAE说明	RW
b18	BTRUSPA	专用比较基准值缓存传送时间设定UA	参考BTRUAE说明	RW
b17	Res	保留位	读出时为“0”,写入时写“0”	RW
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	RW
b15~b12	Res	保留位	读出时为“0”,写入时写“0”	RW
b11	BTRDP	周期值缓存传送时间设定D	参考BTRDAE说明	RW
b10	BTRUP	周期值缓存传送时间设定U	参考BTRUAE说明	RW
b9	Res	保留位	读出时为“0”,写入时写“0”	RW
b8	BENP	周期值缓存传送	0: 缓存传送无效 1: 缓存传送使能	RW
b7	BTRDBF	通用比较值缓存传送时间设定DBF	参考BTRDAE说明	RW
b6	BTRUBF	通用比较值缓存传送时间设定UBF	参考BTRUAE说明	RW
b5	Res	保留位	读出时为“0”,写入时写“0”	RW
b4	BENBF	通用比较值缓存传送BF	0: 缓存传送无效 1: 缓存传送使能	RW
b3	BTRDAE	通用比较值缓存传送时间设定DAE	0: 零点缓存传送点, 缓存值不传送 1: 零点缓存传送点, 发生一次缓存值传送 锯齿波模式下, 该缓存点为在计数值等于0的时刻; 三角波模式下, 该缓存点为向下计数且计数值等于64或者硬件清零的时刻	RW
b2	BTRUAE	通用比较值缓存传送时间设定UAE	0: 周期缓存传送点, 缓存值不传送 1: 周期缓存传送点, 发生一次缓存值传送 锯齿波模式下, 该缓存点为在计数值等于周期值或者硬件清零发生的时刻; 三角波模式下, 该缓存点为向上计	RW

			数且计数值等于周期-64的时刻	
b1	Res	保留位	读出时为“0”,写入时写“0”	RW
b0	BENAE	通用比较值缓存传送AE	0: 缓存传送无效 1: 缓存传送使能	RW

19.6.1.33 缓存控制寄存器 2 (HRPWM<t>_BCONR2, t=1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	BTR DCTL	BTR UCTL	BEN CTL	Res						BTR DEEF WIN	BTR UE EF WIN	BEN EE FW IN	BTR DEEF OFF	BTR UE EF OFF	BEN EE FO FF

位/位域	标记	位名	功能	读写
b31~b15	Res	保留位	读出时为“0”,写入时写“0”	RW
b14	BTRDCTL	控制寄存器缓存传送时间设定DCTL	参考BTRDAE说明	RW
b13	BTRUCTL	控制寄存器缓存传送时间设定UCTL	参考BTRUAE说明	RW
b12	BENCTL	控制寄存器缓存传送	0: 缓存传送无效 1: 缓存传送使能 (BICONR-> ICONR BPCNAR1->PCNAR1 BPCNBR1->PCNBR1 BPCNAR2->PCNAR2 BPCNBR2->PCNBR2 BPCNAR3->PCNAR3 BPCNBR3->PCNBR3 BGCONR1->GCONR1)	RW
b11~b6	Res	保留位	读出时为“0”,写入时写“0”	RW
b5	BTRDEEFWIN	外部事件滤波窗口缓存传送时间设定DWIN	参考BTRDAE说明	RW
b4	BTRUEEFWIN	外部事件滤波窗口缓存传送时间设定UWIN	参考BTRUAE说明	RW
b3	BENE EFWIN	外部事件滤波窗口缓存传送	0: 缓存传送无效 1: 缓存传送使能 (EEFWINBR-> EEFWINAR)	RW
b2	BTRDEEF OFF	外部事件滤波偏移值缓存传送时间设定DOFF	参考BTRDAE说明	RW
b1	BTRUEEF OFF	外部事件滤波偏移值缓存传送时间设定UOFF	参考BTRUAE说明	RW
b0	BENE EF OFF	外部事件滤波偏移值缓存传送	0: 缓存传送无效 1: 缓存传送使能 (EEFOFFSETBR-> EEFOFFSETAR)	RW

19.6.1.34 缓存控制寄存器 2 (HRPWM<t>_BCONR2, t=2~6)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	BTRU OPCT L	BTRU OPDT	BTRU OP SPB	BTRU OP SPA	BTR UOPP	BTR UOP BF	BTR UOP AE	BTRU OCTL	BTRU ODT	BTRU O SPB	BTRU O SPA	BTRU OP	BTRU O BF	BTR UO AE	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	BTRD CTL	BTRU CTL	BENC TL	Res	BTR UOPE EFWI N	BTR UOPE EFOF F	BTRU OEEF WIN	BTRU OEEF OFF	BTRD EEFW IN	BTRU EEFW IN	BENE EFWI N	BTRD EEFO FF	BTRU EEFO FF	BEN EEFO FF	

位/位域	标记	位名	功能	读写
b31~b30	Res	保留位	读出时为“0”,写入时写“0”	RW
b29	BTRUOPCTL	控制寄存器缓存传送UOPCTL	0: 在单元1单次缓存传送点, 缓存值不传送 1: 在单元1单次缓存传送点, 发生一次缓存值传送	RW
b28	BTRUOPDT	死区时间值缓存传送UOPDT	0: 在单元1单次缓存传送点, 缓存值不传送 1: 在单元1单次缓存传送点, 发生一次缓存值传送	RW
b27	BTRUOPSPB	专用比较基准值缓存传送时间设定UOPSPB	0: 在单元1单次缓存传送点, 缓存值不传送 1: 在单元1单次缓存传送点, 发生一次缓存值传送	RW
b26	BTRUOPSPA	专用比较基准值缓存传送时间设定UOPSPA	0: 在单元1单次缓存传送点, 缓存值不传送 1: 在单元1单次缓存传送点, 发生一次缓存值传送	RW
b25	BTRUOPP	周期值缓存传送时间设定UOPP	0: 在单元1单次缓存传送点, 缓存值不传送 1: 在单元1单次缓存传送点, 发生一次缓存值传送	RW
b24	BTRUOPBF	通用比较值缓存传送时间设定UOPBF	0: 在单元1单次缓存传送点, 缓存值不传送 1: 在单元1单次缓存传送点, 发生一次缓存值传送	RW
b23	BTRUOPAE	通用比较值缓存传送时间设定UOPAE	0: 在单元1单次缓存传送点, 缓存值不传送 1: 在单元1单次缓存传送点, 发生一次缓存值传送	RW
b22	BTRUOCTL	控制寄存器缓存传送UOCTL	0: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者三角波零点缓存传送点不传送 1: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者三角波零点缓存传送点发生一次缓存值传送 注: 仅GBCONR.OSTENU1=1时设定有效	RW
b21	BTRUODT	死区时间值缓存传送UODT	0: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者三角波零点缓存传送点缓存值不传送 1: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者三角波零点缓存传送点发生一次缓存值传送 注: 仅GBCONR.OSTENU1=1时设定有效	RW
b20	BTRUOSPB	专用比较基准值缓存传送时间设定UOSPB	0: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者三角波零点缓存传送点缓存值不传送 1: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者三角波零点缓存传送点发生一次缓存值传送 注: 仅GBCONR.OSTENU1=1时设定有效	RW
b19	BTRUOSPA	专用比较基准值缓存传送时间设定UOSPA	0: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者三角波零点缓存传送点缓存值不传送 1: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者	RW

			三角波零点缓存传送点发生一次缓存值传送 注: 仅GBCONR.OSTENU1=1时设定有效	
b18	BTRU0P	周期值缓存传送时间设定U0P	0: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者三角波零点缓存传送点缓存值不传送 1: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者三角波零点缓存传送点发生一次缓存值传送 注: 仅GBCONR.OSTENU1=1时设定有效	RW
b17	BTRU0BF	通用比较值缓存传送时间设定U0BF	0: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者三角波零点缓存传送点缓存值不传送 1: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者三角波零点缓存传送点发生一次缓存值传送 注: 仅GBCONR.OSTENU1=1时设定有效	RW
b16	BTRU0AE	通用比较值缓存传送时间设定U0AE	0: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者三角波零点缓存传送点缓存值不传送 1: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者三角波零点缓存传送点发生一次缓存值传送 注: 仅GBCONR.OSTENU1=1时设定有效	RW
b15	Res	保留位	读出时为“0”,写入时写“0”	RW
b14	BTRDCTL	控制寄存器缓存传送时间设定DCTL	参考BTRDAE说明	RW
b13	BTRUCTL	控制寄存器缓存传送时间设定UCTL	参考BTRUAE说明	RW
b12	BENCTL	控制寄存器缓存传送	0: 缓存传送无效 1: 缓存传送使能 (BICONR-> ICONR BPCNAR1->PCNAR1 BPCNBR1->PCNBR1 BPCNAR2->PCNAR2 BPCNBR2->PCNBR2 BPCNAR3->PCNAR3 BPCNBR3->PCNBR3 BGCONR1->GCONR1)	RW
b11~b10	Res	保留位	读出时为“0”,写入时写“0”	RW
b9	BTRU0PEEFWIN	外部事件滤波窗口缓存传送时间设定U0PWIN	0: 在单元1单次缓存传送点, 缓存值不传送 1: 在单元1单次缓存传送点, 发生一次缓存值传送	RW
b8	BTRU0PEEFOFF	外部事件滤波窗口缓存传送时间设定U0POFF	0: 在单元1单次缓存传送点, 缓存值不传送 1: 在单元1单次缓存传送点, 发生一次缓存值传送	RW
b7	BTRU0EEFWIN	外部事件滤波窗口缓存传送时间设定U0WIN	0: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者三角波零点缓存传送点, 缓存值不传送 1: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者三角波零点缓存传送点, 发生一次缓存值传送 注: 仅PHSCTL.PHSEN=1且GBCONR.OSTENU1=1时设定有效	RW
b6	BTRU0EEFOFF	外部事件滤波窗口缓存传送时间设定U0OFF	0: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者三角波零点缓存传送点, 缓存值不传送 1: 在单元1单次缓存传送点后本单元的锯齿波周期缓存传送点或者三角波零点缓存传送点, 发生一次缓存值传送	RW

注：仅PHSCTL.PHSEN=1且GBCONR.OSTENU1=1时设定有效

b5	BTRDEEFWIN	外部事件滤波窗口缓存 传送时间设定DWIN	参考BTRDAE说明	RW
b4	BTRUEEFWIN	外部事件滤波窗口缓存 传送时间设定UWIN	参考BTRUAE说明	RW
b3	BENEEFWIN	外部事件滤波窗口缓存 传送	0: 缓存传送无效 1: 缓存传送使能 (EEFWINBR-> EEFWINAR)	RW
b2	BTRDEEFOFF	外部事件滤波偏移值缓 存传送时间设定DOFF	参考BTRDAE说明	RW
b1	BTRUEEFOFF	外部事件滤波偏移值缓 存传送时间设定UOFF	参考BTRUAE说明	RW
b0	BENEEOFF	外部事件滤波偏移值缓 存传送	0: 缓存传送无效 1: 缓存传送使能 (EEFOFFSETBR-> EEFOFFSETAR)	RW

19.6.1.35 端口控制寄存器 A1 (HRPWM<t>_PCNAR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res			OUT ENA	Res	EMBS A[2:0]		EMBR A[1:0]		EMBC A[1:0]		Res		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/位域	标记	位名	功能	读写
b31~b29	Res	保留位	读出时为“0”,写入时写“0”	RW
b28	OUTENA	输出使能A	0: HRPWM功能时的HRPWM_<t>_PWMA端口输出无效 1: HRPWM功能时的HRPWM_<t>_PWMA端口输出有效	RW
b27	Res	保留位	读出时为“0”,写入时写“0”	RW
b26~b24	EMBSA[2:0]	EMB事件时端口状态选择A	000: 选择EMB事件通道0有效 001: 选择EMB事件通道1有效 010: 选择EMB事件通道2有效 011: 选择EMB事件通道3有效 100: 选择EMB事件通道4有效 101: 选择EMB事件通道5有效	RW
b23~b22	EMBRA[1:0]	EMB释放方式选择A	00: 被选择的通道EMB事件无效时, 立即释放 HRPWM_<t>_PWMA端口 (One Shot) 01: 被选择的通道EMB事件无效时, 等到周期匹配时释放 HRPWM_<t>_PWMA端口 (Cycle By Cycle 1) 10: 被选择的通道EMB事件无效时, 等到零点匹配时释放 HRPWM_<t>_PWMA端口 (Cycle By Cycle 2) 11: 被选择的通道EMB事件无效时, 等到周期匹配或零点匹配时 释放HRPWM_<t>_PWMA端口 (Cycle By Cycle 3)	RW
b21~b20	EMBCA[1:0]	EMB事件时端口状态设定A	00: 被选择的通道发生EMB事件时, HRPWM_<t>_PWMA端口正 常输出 01: 被选择的通道发生EMB事件时, HRPWM_<t>_PWMA端口输 出高阻态 10: 被选择的通道发生EMB事件时, HRPWM_<t>_PWMA端口输 出低电平 11: 被选择的通道发生EMB事件时, HRPWM_<t>_PWMA端口输 出高电平	RW
b19~b18	Res	保留位	读出时为“0”,写入时写“0”	RW
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, HRPWM_<t>_PWMA端口输出设定为低电平 11: 下周期开始, HRPWM_<t>_PWMA端口输出设定为高电平 注1: 下周期是指由寄存器HRPWM_<t>_GCONR1.PRDSSEL配置 的完整周期点 注2: 该寄存器位可用于实现PWM输出占空比0%或100%的控制	RW
b15~b14	CMBDCA[1:0]	向下计数&&比较值B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GCMBR相等时, HRPWM_<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, HRPWM_<t>_PWMA端口输出设定为高电平	RW

			10: 在向下计数期间, 定时器计数值与GCMBR相等时, HRPWM_<t>_PWMA端口输出不受影响 11: 在向下计数期间, 定时器计数值与GCMBR相等时, HRPWM_<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, HRPWM_<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, HRPWM_<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, HRPWM_<t>_PWMA端口输出不受影响 11: 在向上计数期间, 定时器计数值与GCMBR相等时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
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, HRPWM_<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, HRPWM_<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, HRPWM_<t>_PWMA端口输出不受影响 11: 在向下计数期间, 定时器计数值与GCMAR相等时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
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, HRPWM_<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, HRPWM_<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, HRPWM_<t>_PWMA端口输出不受影响 11: 在向上计数期间, 定时器计数值与GCMAR相等时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b7~b6	UDFCA[1:0]	计数值等有0时端口状态设定A	00: 计数值等于0时, HRPWM_<t>_PWMA端口输出设定为低电平 01: 计数值等于0时, HRPWM_<t>_PWMA端口输出设定为高电平 10: 计数值等于0时, HRPWM_<t>_PWMA端口输出不受影响 11: 计数值等于0时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b5~b4	OVFCA[1:0]	计数值等于周期值或者锯齿波硬件清零时端口状态设定A	00: 计数值等于周期值或者锯齿波硬件清零时, HRPWM_<t>_PWMA端口输出设定为低电平 01: 计数值等于周期值或者锯齿波硬件清零时, HRPWM_<t>_PWMA端口输出设定为高电平 10: 计数值等于周期值或者锯齿波硬件清零时, HRPWM_<t>_PWMA端口输出不受影响 11: 计数值等于周期值或者锯齿波硬件清零时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b3~b2	STPCA[1:0]	计数停止端口状态设定A	00: 计数停止时, HRPWM_<t>_PWMA端口输出设定为低电平 01: 计数停止时, HRPWM_<t>_PWMA端口输出设定为高电平 10: 计数停止时, HRPWM_<t>_PWMA端口输出保持先前状态 11: 计数停止时, HRPWM_<t>_PWMA端口输出保持先前状态	RW
b1~b0	STACA[1:0]	计数开始端口状态设定A	00: 计数开始时, HRPWM_<t>_PWMA端口输出设定为低电平 01: 计数开始时, HRPWM_<t>_PWMA端口输出设定为高电平 10: 计数开始时, HRPWM_<t>_PWMA端口输出保持先前状态	RW

11: 计数开始时, HRPWM_<t>_PWMA端口输出保持先前状态-

19.6.1.36 端口控制寄存器 B1 (HRPWM<t>_PCNBR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res			OUT ENB	Res	EMBSB[1:0]		EMBRB[1:0]		EMBCB[1:0]		Res		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]		CMADCB[1:0]		CMAUCB[1:0]		UDFCB[1:0]		OVFCB[1:0]		STPCB[1:0]		STACB[1:0]	

位/位域	标记	位名	功能	读写
b31~b29	Res	保留位	读出时为“0”,写入时写“0”	RW
b28	OUTENB	输出使能B	0: HRPWM功能时的HRPWM_<t>_PWMB端口输出无效 1: HRPWM功能时的HRPWM_<t>_PWMB端口输出有效	RW
b27	Res	保留位	读出时为“0”,写入时写“0”	RW
b26~b24	EMBSB[2:0]	EMB事件时端口状态选择B	000: 选择EMB事件通道0有效 001: 选择EMB事件通道1有效 010: 选择EMB事件通道2有效 011: 选择EMB事件通道3有效 100: 选择EMB事件通道4有效 101: 选择EMB事件通道5有效	RW
b23~b22	EMBRB[1:0]	EMB释放方式选择B	00: 被选择的通道EMB事件无效时, 立即释放HRPWM_<t>_PWMB端口 (One Shot) 01: 被选择的通道EMB事件无效时, 等到周期匹配时释放HRPWM_<t>_PWMB端口 (Cycle By Cycle 1) 10: 被选择的通道EMB事件无效时, 等到零点匹配时释放HRPWM_<t>_PWMB端口 (Cycle By Cycle 2) 11: 被选择的通道EMB事件无效时, 等到周期匹配或者零点匹配时释放HRPWM_<t>_PWMB端口 (Cycle By Cycle 3)	RW
b21~b20	EMBCB[1:0]	EMB事件时端口状态设定B	00: 被选择的通道发生EMB事件时, HRPWM_<t>_PWMB端口正常输出 01: 被选择的通道发生EMB事件时, HRPWM_<t>_PWMB端口输出高阻态 10: 被选择的通道发生EMB事件时, HRPWM_<t>_PWMB端口输出低电平 11: 被选择的通道发生EMB事件时, HRPWM_<t>_PWMB端口输出高电平	RW
b19~b18	Res	保留位	读出时为“0”,写入时写“0”	RW
b17~b16	FORCB[1:0]	强制端口状态设定B	0x: 设定无效定 10: 下周期开始, HRPWM_<t>_PWMB端口输出设定为低电平 11: 下周期开始, HRPWM_<t>_PWMB端口输出设定为高电平 注1: 下周期是指由寄存器HRPWM_<t>_GCONR1.PRDESEL配置的完整周期点 注2: 该寄存器位可用于实现PWM输出占空比0%或100%的控制	RW
b15~b14	CMBDCB[1:0]	向下计数&&比较值B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GCMR相等时, HRPWM_<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GCMR相等时,	RW

			HRPWM_<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, HRPWM_<t>_PWMB端口输出不受影响 11: 在向下计数期间, 定时器计数值与GCMBR相等时, HRPWM_<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, HRPWM_<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, HRPWM_<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, HRPWM_<t>_PWMB端口输出不受影响 11: 在向上计数期间, 定时器计数值与GCMBR相等时, HRPWM_<t>_PWMB端口输出设定为反转电平	RW
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, HRPWM_<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, HRPWM_<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, HRPWM_<t>_PWMB端口输出不受影响 11: 在向下计数期间, 定时器计数值与GCMAR相等时, HRPWM_<t>_PWMB端口输出设定为反转电平	RW
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, HRPWM_<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, HRPWM_<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, HRPWM_<t>_PWMB端口输出不受影响 11: 在向上计数期间, 定时器计数值与GCMAR相等时, HRPWM_<t>_PWMB端口输出设定为反转电平	RW
b7~b6	UDFCB[1:0]	计数值等有0时端口状态设定B	00: 计数值等于0时, HRPWM_<t>_PWMB端口输出设定为低电平 01: 计数值等于0时, HRPWM_<t>_PWMB端口输出设定为高电平 10: 计数值等于0时, HRPWM_<t>_PWMB端口输出不受影响 11: 计数值等于0时, HRPWM_<t>_PWMB端口输出设定为反转电平	RW
b5~b4	OVFCB[1:0]	计数值等于周期值或者锯齿波硬件清零时端口状态设定B	00: 计数值等于周期值或者锯齿波硬件清零时, HRPWM_<t>_PWMB端口输出设定为低电平 01: 计数值等于周期值或者锯齿波硬件清零时, HRPWM_<t>_PWMB端口输出设定为高电平 10: 计数值等于周期值或者锯齿波硬件清零时, HRPWM_<t>_PWMB端口输出不受影响 11: 计数值等于周期值或者锯齿波硬件清零时, HRPWM_<t>_PWMB端口输出设定为反转电平	RW
b3~b2	STPCB[1:0]	计数停止端口状态设定B	00: 计数停止时, HRPWM_<t>_PWMB端口输出设定为低电平 01: 计数停止时, HRPWM_<t>_PWMB端口输出设定为高电平 10: 计数停止时, HRPWM_<t>_PWMB端口输出保持先前状态 11: 计数停止时, HRPWM_<t>_PWMB端口输出保持先前状态	RW

b1~b0	STBCB[1:0]	计数开始端口状态设定B	00: 计数开始时, HRPWM_<t>_PWMB端口输出设定为低电平 01: 计数开始时, HRPWM_<t>_PWMB端口输出设定为高电平 10: 计数开始时, HRPWM_<t>_PWMB端口输出保持先前状态 11: 计数开始时, HRPWM_<t>_PWMB端口输出保持先前状态	RW
-------	------------	-------------	--	----

19.6.1.37 端口控制缓存寄存器 A1 (HRPWM<t>_BPCNAR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res			OUT	Res	EMBS			EMBR		EMBC		Res		FORC	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMBDC		CMBUC		CMADC		CMAUC		UDFC		OVFC		STPC		STAC	

位/位域	标记	位名	功能	读写
b31~b29	Res	保留位	读出时为“0”,写入时写“0”	RW
b28	OUTENA	输出使能A	0: HRPWM功能时的HRPWM_<t>_PWMA端口输出无效 1: HRPWM功能时的HRPWM_<t>_PWMA端口输出有效	RW
b27	Res	保留位	读出时为“0”,写入时写“0”	RW
b26~b24	EMBSA[2:0]	EMBS事件时端口状态选择A	000: 选择EMBS事件通道0有效 001: 选择EMBS事件通道1有效 010: 选择EMBS事件通道2有效 011: 选择EMBS事件通道3有效 100: 选择EMBS事件通道4有效 101: 选择EMBS事件通道5有效	RW
b23~b22	EMBRA[1:0]	EMBS释放方式选择A	00: 被选择的通道EMBS事件无效时, 立即释放 HRPWM_<t>_PWMA端口 (One Shot) 01: 被选择的通道EMBS事件无效时, 等到周期匹配时释放 HRPWM_<t>_PWMA端口 (Cycle By Cycle 1) 10: 被选择的通道EMBS事件无效时, 等到零点匹配时释放 HRPWM_<t>_PWMA端口 (Cycle By Cycle 2) 11: 被选择的通道EMBS事件无效时, 等到周期匹配或零点匹配时释放HRPWM_<t>_PWMA端口 (Cycle By Cycle 3)	RW
b21~b20	EMBCA[1:0]	EMBS事件时端口状态设定A	00: 被选择的通道发生EMBS事件时, HRPWM_<t>_PWMA端口正常输出 01: 被选择的通道发生EMBS事件时, HRPWM_<t>_PWMA端口输出高阻态 10: 被选择的通道发生EMBS事件时, HRPWM_<t>_PWMA端口输出低电平 11: 被选择的通道发生EMBS事件时, HRPWM_<t>_PWMA端口输出高电平	RW
b19~b18	Res	保留位	读出时为“0”,写入时写“0”	RW
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, HRPWM_<t>_PWMA端口输出设定为低电平 11: 下周期开始, HRPWM_<t>_PWMA端口输出设定为高电平 注1: 下周期是指由寄存器HRPWM_<t>_GCONR1.PRDSEL配置的完整周期点 注2: 该寄存器位可用于实现PWM输出占空比0%或100%的控制	RW
b15~b14	CMBDCA[1:0]	向下计数&&比较值B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GCMBR相等时, HRPWM_<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,	RW

			HRPWM_<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, HRPWM_<t>_PWMA端口输出不受影响 11: 在向下计数期间, 定时器计数值与GCMBR相等时, HRPWM_<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, HRPWM_<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, HRPWM_<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, HRPWM_<t>_PWMA端口输出不受影响 11: 在向上计数期间, 定时器计数值与GCMBR相等时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
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, HRPWM_<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, HRPWM_<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, HRPWM_<t>_PWMA端口输出不受影响 11: 在向下计数期间, 定时器计数值与GCMAR相等时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
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, HRPWM_<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, HRPWM_<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, HRPWM_<t>_PWMA端口输出不受影响 11: 在向上计数期间, 定时器计数值与GCMAR相等时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b7~b6	UDFCA[1:0]	计数值等有0时端口状态设定A	00: 计数值等于0时, HRPWM_<t>_PWMA端口输出设定为低电平 01: 计数值等于0时, HRPWM_<t>_PWMA端口输出设定为高电平 10: 计数值等于0时, HRPWM_<t>_PWMA端口输出不受影响 11: 计数值等于0时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b5~b4	OVFCA[1:0]	计数值等于周期值或者锯齿波硬件清零时端口状态设定A	00: 计数值等于周期值或者锯齿波硬件清零时, HRPWM_<t>_PWMA端口输出设定为低电平 01: 计数值等于周期值或者锯齿波硬件清零时, HRPWM_<t>_PWMA端口输出设定为高电平 10: 计数值等于周期值或者锯齿波硬件清零时, HRPWM_<t>_PWMA端口输出不受影响 11: 计数值等于周期值或者锯齿波硬件清零时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b3~b2	STPCA[1:0]	计数停止端口状态设定A	00: 计数停止时, HRPWM_<t>_PWMA端口输出设定为低电平 01: 计数停止时, HRPWM_<t>_PWMA端口输出设定为高电平 10: 计数停止时, HRPWM_<t>_PWMA端口输出保持先前状态 11: 计数停止时, HRPWM_<t>_PWMA端口输出保持先前状态	RW

b1~b0	STACA[1:0]	计数开始端口状态设定A	00: 计数开始时, HRPWM_<t>_PWMA端口输出设定为低电平 01: 计数开始时, HRPWM_<t>_PWMA端口输出设定为高电平 10: 计数开始时, HRPWM_<t>_PWMA端口输出保持先前状态 11: 计数开始时, HRPWM_<t>_PWMA端口输出保持先前状态 -	RW
-------	------------	-------------	---	----

19.6.1.38 端口控制缓存寄存器 B1 (HRPWM<t>_BPCNBR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res			OUTENB	Res	EMBSB[1:0]		EMBRB[1:0]		EMBCB[1:0]		Res		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]		CMADCB[1:0]		CMAUCB[1:0]		UDFCB[1:0]		OVFCB[1:0]		STPCB[1:0]		STACB[1:0]

位/位域	标记	位名	功能	读写
b31~b29	Res	保留位	读出时为“0”,写入时写“0”	RW
b28	OUTENB	输出使能B	0: HRPWM功能时的HRPWM_<t>_PWMB端口输出无效 1: HRPWM功能时的HRPWM_<t>_PWMB端口输出有效	RW
b27	Res	保留位	读出时为“0”,写入时写“0”	RW
b26~b24	EMBSB[2:0]	EMB事件时端口状态选择B	000: 选择EMB事件通道0有效 001: 选择EMB事件通道1有效 010: 选择EMB事件通道2有效 011: 选择EMB事件通道3有效 100: 选择EMB事件通道4有效 101: 选择EMB事件通道5有效	RW
b23~b22	EMBRB[1:0]	EMB释放方式选择B	00: 被选择的通道EMB事件无效时, 立即释放 HRPWM_<t>_PWMB端口 (One Shot) 01: 被选择的通道EMB事件无效时, 周期匹配时释放 HRPWM_<t>_PWMB端口 (Cycle By Cycle 1) 10: 被选择的通道EMB事件无效时, 零点匹配时释放 HRPWM_<t>_PWMB端口 (Cycle By Cycle 2) 11: 被选择的通道EMB事件无效时, 周期匹配或者零点匹配时释放 HRPWM_<t>_PWMB端口 (Cycle By Cycle 3)	RW
b21~b20	EMBCB[1:0]	EMB事件时端口状态设定B	00: 被选择的通道发生EMB事件时, HRPWM_<t>_PWMB端口正常输出 01: 被选择的通道发生EMB事件时, HRPWM_<t>_PWMB端口输出高阻态 10: 被选择的通道发生EMB事件时, HRPWM_<t>_PWMB端口输出低电平 11: 被选择的通道发生EMB事件时, HRPWM_<t>_PWMB端口输出高电平	RW
b19~b18	Res	保留位	读出时为“0”,写入时写“0”	RW
b17~b16	FORCB[1:0]	强制端口状态设定B	0x: 设定无效定 10: 下周期开始, HRPWM_<t>_PWMB端口输出设定为低电平 11: 下周期开始, HRPWM_<t>_PWMB端口输出设定为高电平 注1: 下周期是指由寄存器HRPWM_<t>_GCONR1.PRDSEL配置的完整周期点 注2: 该寄存器位可用于实现PWM输出占空比0%或100%的控制	RW
b15~b14	CMBDCB[1:0]	向下计数&&比较值B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GCMBR相等时, HRPWM_<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, HRPWM_<t>_PWMB端口输出设定为高电平	RW

			10: 在向下计数期间, 定时器计数值与GCMBR相等时, HRPWM_<t>_PWMB端口输出不受影响 11: 在向下计数期间, 定时器计数值与GCMBR相等时, HRPWM_<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, HRPWM_<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, HRPWM_<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, HRPWM_<t>_PWMB端口输出不受影响 11: 在向上计数期间, 定时器计数值与GCMBR相等时, HRPWM_<t>_PWMB端口输出设定为反转电平	RW
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, HRPWM_<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, HRPWM_<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, HRPWM_<t>_PWMB端口输出不受影响 11: 在向下计数期间, 定时器计数值与GCMAR相等时, HRPWM_<t>_PWMB端口输出设定为反转电平	RW
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, HRPWM_<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, HRPWM_<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, HRPWM_<t>_PWMB端口输出不受影响 11: 在向上计数期间, 定时器计数值与GCMAR相等时, HRPWM_<t>_PWMB端口输出设定为反转电平	RW
b7~b6	UDFCB[1:0]	计数值等有0时端口状态设定B	00: 计数值等于0时, HRPWM_<t>_PWMB端口输出设定为低电平 01: 计数值等于0时, HRPWM_<t>_PWMB端口输出设定为高电平 10: 计数值等于0时, HRPWM_<t>_PWMB端口输出不受影响 11: 计数值等于0时, HRPWM_<t>_PWMB端口输出设定为反转电平	RW
b5~b4	OVFCB[1:0]	计数值等于周期值或者锯齿波硬件清零时端口状态设定B	00: 计数值等于周期值或者锯齿波硬件清零时, HRPWM_<t>_PWMB端口输出设定为低电平 01: 计数值等于周期值或者锯齿波硬件清零时, HRPWM_<t>_PWMB端口输出设定为高电平 10: 计数值等于周期值或者锯齿波硬件清零时, HRPWM_<t>_PWMB端口输出不受影响 11: 计数值等于周期值或者锯齿波硬件清零时, HRPWM_<t>_PWMB端口输出设定为反转电平	RW
b3~b2	STPCB[1:0]	计数停止端口状态设定B	00: 计数停止时, HRPWM_<t>_PWMB端口输出设定为低电平 01: 计数停止时, HRPWM_<t>_PWMB端口输出设定为高电平 10: 计数停止时, HRPWM_<t>_PWMB端口输出保持先前状态 11: 计数停止时, HRPWM_<t>_PWMB端口输出保持先前状态	RW
b1~b0	STBCB[1:0]	计数开始端口状态设定B	00: 计数开始时, HRPWM_<t>_PWMB端口输出设定为低电平 01: 计数开始时, HRPWM_<t>_PWMB端口输出设定为高电平 10: 计数开始时, HRPWM_<t>_PWMB端口输出保持先前状态	RW

11: 计数开始时, HRPWM_<t>_PWMB端口输出保持先前状态

19.6.1.39 端口控制寄存器 A2 (HRPWM<t>_PCNAR2)

复位值: 0x000A AAAA

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				SCMBUCA[1:0]		SCMAUCA[1:0]		CMFUCA[1:0]		CMEUCA[1:0]		EXEV10UCA[1:0]		EXEV9UCA[1:0]	
b15		b14		b13		b12		b11		b10		b9		b8	
EXEV8UCA[1:0]		EXEV7UCA[1:0]		EXEV6UCA[1:0]		EXEV5UCA[1:0]		EXEV4UCA[1:0]		EXEV3UCA[1:0]		EXEV2UCA[1:0]		EXEV1UCA[1:0]	
位/位域	标记			位名			功能			读写					
b31~b28	Rese			保留位			读出时为“0”,写入时写“0”			RW					
b27~b26	SCMBUCA[1:0]			向上计数&&专用比较值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SCMBR相等时, HRPWM_<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SCMBR相等时, HRPWM_<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SCMBR相等时, HRPWM_<t>_PWMA端口输出不受影响 11: 在向上计数期间, 定时器计数值与SCMBR相等时, HRPWM_<t>_PWMA端口输出设定为反转电平			RW					
b25~b24	SCMAUCA[1:0]			向上计数&&专用比较值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SCMAR相等时, HRPWM_<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SCMAR相等时, HRPWM_<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SCMAR相等时, HRPWM_<t>_PWMA端口输出不受影响 11: 在向上计数期间, 定时器计数值与SCMAR相等时, HRPWM_<t>_PWMA端口输出设定为反转电平			RW					
b23~b22	CMFUCA[1:0]			向上计数&&比较值F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HRGCMFR相等时, HRPWM_<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HRGCMFR相等时, HRPWM_<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HRGCMFR相等时, HRPWM_<t>_PWMA端口输出不受影响 11: 在向上计数期间, 定时器计数值与HRGCMFR相等时, HRPWM_<t>_PWMA端口输出设定为反转电平			RW					
b21~b20	CMEUCA[1:0]			向上计数&&比较值E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HRGCMER相等时, HRPWM_<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HRGCMER相等时, HRPWM_<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HRGCMER相等时, HRPWM_<t>_PWMA端口输出不受影响 11: 在向上计数期间, 定时器计数值与HRGCMER相等时, HRPWM_<t>_PWMA端口输出设定为反转电平			RW					
b19~b18	EXEV10UCA[1:0]			向上计数&&外部事件10发生端口状态设定A			参考EXEV1CA[1:0]说明			RW					
b17~b16	EXEV9UCA[1:0]			向上计数&&外部事件9发生端口状态设定A			参考EXEV1UCA[1:0]说明			RW					
b15~b14	EXEV8UCA[1:0]			向上计数&&外部事件8			参考EXEV1UCA[1:0]说明			RW					

		发生端口状态设定A		
b13~b12	EXEV7UCA[1:0]	向上计数&&外部事件7 发生端口状态设定A	参考EXEV1UCA[1:0]说明	RW
b11~b10	EXEV6UCA[1:0]	向上计数&&外部事件6 发生端口状态设定A	参考EXEV1UCA[1:0]说明	RW
b9~b8	EXEV5UCA[1:0]	向上计数&&外部事件5 发生端口状态设定A	参考EXEV1UCA[1:0]说明	RW
b7~b6	EXEV4UCA[1:0]	向上计数&&外部事件4 发生端口状态设定A	参考EXEV1UCA[1:0]说明	RW
b5~b4	EXEV3UCA[1:0]	向上计数&&外部事件3 发生端口状态设定A	参考EXEV1UCA[1:0]说明	RW
b3~b2	EXEV2UCA[1:0]	向上计数&&外部事件2 发生端口状态设定A	参考EXEV1UCA[1:0]说明	RW
b1~b0	EXEV1UCA[1:0]	向上计数&&外部事件1 发生端口状态设定A	00: 在向上计数期间, 外部事件1发生时, HRPWM_<t>_PWMA端口输出设定为低电平 01: 在向上计数期间, 外部事件1发生时, HRPWM_<t>_PWMA端口输出设定为高电平 10: 在向上计数期间, 外部事件1发生时, HRPWM_<t>_PWMA端口输出不受影响 11: 在向上计数期间, 外部事件1发生时, HRPWM_<t>_PWMA端口输出反转电平	RW

19.6.1.40 端口控制寄存器 B2 (HRPWM<t>_PCNBR2)

复位值: 0x000A AAAA

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				SCMBUCB[1:0]		SCMAUCB[1:0]		CMFUCB[1:0]		CMEUCB[1:0]		EXEV10UCB[1:0]		EXEV9UCB[1:0]	
b15		b14		b13		b12		b11		b10		b9		b8	
EXEV8UCB[1:0]		EXEV7UCB[1:0]		EXEV6UCB[1:0]		EXEV5UCB[1:0]		EXEV4UCB[1:0]		EXEV3UCB[1:0]		EXEV2UCB[1:0]		EXEV1UCB[1:0]	
位/位域	标记			位名			功能			读写					
b31~b28	Res			保留位			读出时为“0”,写入时写“0”			RW					
b27~b26	SCMBUCB[1:0]			向上计数&&专用比较值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SCMBR相等时,HRPWM_<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SCMBR相等时,HRPWM_<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SCMBR相等时,HRPWM_<t>_PWMB端口输出不受影响 11: 在向上计数期间,定时器计数值与SCMBR相等时,HRPWM_<t>_PWMB端口输出设定为反转电平			RW					
b25~b24	SCMAUCB[1:0]			向上计数&&专用比较值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SCMAR相等时,HRPWM_<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SCMAR相等时,HRPWM_<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SCMAR相等时,HRPWM_<t>_PWMB端口输出不受影响 11: 在向上计数期间,定时器计数值与SCMAR相等时,HRPWM_<t>_PWMB端口输出设定为反转电平			RW					
b23~b22	CMFUCB[1:0]			向上计数&&比较值F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HRGCMFR相等时,HRPWM_<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HRGCMFR相等时,HRPWM_<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HRGCMFR相等时,HRPWM_<t>_PWMB端口输出不受影响 11: 在向上计数期间,定时器计数值与HRGCMFR相等时,HRPWM_<t>_PWMB端口输出设定为反转电平			RW					
b21~b20	CMEUCB[1:0]			向上计数&&比较值E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HRGCMER相等时,HRPWM_<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HRGCMER相等时,HRPWM_<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HRGCMER相等时,HRPWM_<t>_PWMB端口输出不受影响 11: 在向上计数期间,定时器计数值与HRGCMER相等时,HRPWM_<t>_PWMB端口输出设定为反转电平			RW					
b19~b18	EXEV10UCB[1:0]			向上计数&&外部事件10发生端口状态设定B			参考EXEV1CB[1:0]说明			RW					
b17~b16	EXEV9UCB[1:0]			向上计数&&外部事件9发生端口状态设定B			参考EXEV1CB[1:0]说明			RW					

		定B		
b15~b14	EXEV8UCB[1:0]	向上计数&&外部事件8发生端口状态设定B	参考EXEV1CB[1:0]说明	RW
b13~b12	EXEV7UCB[1:0]	向上计数&&外部事件7发生端口状态设定B	参考EXEV1CB[1:0]说明	RW
b11~b10	EXEV6UCB[1:0]	向上计数&&外部事件6发生端口状态设定A	参考EXEV1CB[1:0]说明	RW
b9~b8	EXEV5UCB[1:0]	向上计数&&外部事件5发生端口状态设定B	参考EXEV1CB[1:0]说明	RW
b7~b6	EXEV4UCB[1:0]	向上计数&&外部事件4发生端口状态设定B	参考EXEV1CB[1:0]说明	RW
b5~b4	EXEV3UCB[1:0]	向上计数&&外部事件3发生端口状态设定B	参考EXEV1CB[1:0]说明	RW
b3~b2	EXEV2UCB[1:0]	向上计数&&外部事件2发生端口状态设定B	参考EXEV1CB[1:0]说明	RW
b1~b0	EXEV1UCB[1:0]	向上计数&&外部事件1发生端口状态设定B	00: 在向上计数期间, 外部事件1发生时, HRPWM_<t>_PWMB端口输出设定为低电平 01: 在向上计数期间, 外部事件1发生时, HRPWM_<t>_PWMB端口输出设定为高电平 10: 在向上计数期间, 外部事件1发生时, HRPWM_<t>_PWMB端口输出不受影响 11: 在向上计数期间, 外部事件1发生时, HRPWM_<t>_PWMB端口输出反转电平	RW

19.6.1.41 端口控制缓存寄存器 A2 (HRPWM<t>_BPCNAR2)

复位值: 0x000A AAAA

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				SCMBUCA[1:0]		SCMAUCA[1:0]		CMFUCA[1:0]		CMEUCA[1:0]		EXEV10UCA[1:0]		EXEV9UCA[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EXEV8UCA[1:0]		EXEV7UCA[1:0]		EXEV6UCA[1:0]		EXEV5UCA[1:0]		EXEV4UCA[1:0]		EXEV3UCA[1:0]		EXEV2UCA[1:0]		EXEV1UCA[1:0]	

位/位域	标记	位名	功能	读写
b31~b28	Res	保留位	读出时为“0”,写入时写“0”	RW
b27~b26	SCMBUCA[1:0]	向上计数&&专用比较值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SCMBR相等时, HRPWM_<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SCMBR相等时, HRPWM_<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SCMBR相等时, HRPWM_<t>_PWMA端口输出不受影响 11: 在向上计数期间, 定时器计数值与SCMBR相等时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b25~b24	SCMAUCA[1:0]	向上计数&&专用比较值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SCMAR相等时, HRPWM_<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SCMAR相等时, HRPWM_<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SCMAR相等时, HRPWM_<t>_PWMA端口输出不受影响 11: 在向上计数期间, 定时器计数值与SCMAR相等时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b23~b22	CMFUCA[1:0]	向上计数&&比较值F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HRGCMFR相等时, HRPWM_<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HRGCMFR相等时, HRPWM_<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HRGCMFR相等时, HRPWM_<t>_PWMA端口输出不受影响 11: 在向上计数期间, 定时器计数值与HRGCMFR相等时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b21~b20	CMEUCA[1:0]	向上计数&&比较值E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HRGCMER相等时, HRPWM_<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HRGCMER相等时, HRPWM_<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HRGCMER相等时, HRPWM_<t>_PWMA端口输出不受影响 11: 在向上计数期间, 定时器计数值与HRGCMER相等时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b19~b18	EXEV10UCA[1:0]	向上计数&&外部事件10发 生端口状态设定A	参考EXEV1CA[1:0]说明	RW
b17~b16	EXEV9UCA[1:0]	向上计数&&外部事件9发 生端口状态设定A	参考EXEV1UCA[1:0]说明	RW

b15~b14	EXEV8UCA[1:0]	向上计数&&外部事件8发生端口状态设定A	参考EXEV1UCA[1:0]说明	RW
b13~b12	EXEV7UCA[1:0]	向上计数&&外部事件7发生端口状态设定A	参考EXEV1UCA[1:0]说明	RW
b11~b10	EXEV6UCA[1:0]	向上计数&&外部事件6发生端口状态设定A	参考EXEV1UCA[1:0]说明	RW
b9~b8	EXEV5UCA[1:0]	向上计数&&外部事件5发生端口状态设定A	参考EXEV1UCA[1:0]说明	RW
b7~b6	EXEV4UCA[1:0]	向上计数&&外部事件4发生端口状态设定A	参考EXEV1UCA[1:0]说明	RW
b5~b4	EXEV3UCA[1:0]	向上计数&&外部事件3发生端口状态设定A	参考EXEV1UCA[1:0]说明	RW
b3~b2	EXEV2UCA[1:0]	向上计数&&外部事件2发生端口状态设定A	参考EXEV1UCA[1:0]说明	RW
b1~b0	EXEV1UCA[1:0]	向上计数&&外部事件1发生端口状态设定A	<p>00: 在向上计数期间, 外部事件1发生时, HRPWM_<t>_PWMA端口输出设定为低电平</p> <p>01: 在向上计数期间, 外部事件1发生时, HRPWM_<t>_PWMA端口输出设定为高电平</p> <p>10: 在向上计数期间, 外部事件1发生时, HRPWM_<t>_PWMA端口输出不受影响</p> <p>11: 在向上计数期间, 外部事件1发生时, HRPWM_<t>_PWMA端口输出反转电平</p>	RW

19.6.1.42 端口控制缓存寄存器 B2 (HRPWM<t>_BPCNBR2)

复位值: 0x000A AAAA

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				SCMBDCB[1:0]		SCMADCB[1:0]		CMFUCB[1:0]		CMEUCB[1:0]		EXEV10UCB[1:0]		EXEV9UCB[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EXEV8UCB[1:0]		EXEV7UCB[1:0]		EXEV6UCB[1:0]		EXEV5UCB[1:0]		EXEV4UCB[1:0]		EXEV3UCB[1:0]		EXEV2UCB[1:0]		EXEV1UCB[1:0]	

位/位域	标记	位名	功能	读写
b31~b28	Res	保留位	读出时为“0”,写入时写“0”	RW
b27~b26	SCMBUCB[1:0]	向上计数&&专用比较值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SCMBR相等时,HRPWM_<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SCMBR相等时,HRPWM_<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SCMBR相等时,HRPWM_<t>_PWMB端口输出不受影响 11: 在向上计数期间,定时器计数值与SCMBR相等时,HRPWM_<t>_PWMB端口输出设定为反转电平	RW
b25~b24	SCMAUCB[1:0]	向上计数&&专用比较值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SCMAR相等时,HRPWM_<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SCMAR相等时,HRPWM_<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SCMAR相等时,HRPWM_<t>_PWMB端口输出不受影响 11: 在向上计数期间,定时器计数值与SCMAR相等时,HRPWM_<t>_PWMB端口输出设定为反转电平	RW
b23~b22	CMFUCB[1:0]	向上计数&&比较值F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HRGCMFR相等时,HRPWM_<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HRGCMFR相等时,HRPWM_<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HRGCMFR相等时,HRPWM_<t>_PWMB端口输出不受影响 11: 在向上计数期间,定时器计数值与HRGCMFR相等时,HRPWM_<t>_PWMB端口输出设定为反转电平	RW
b21~b20	CMEUCB[1:0]	向上计数&&比较值E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HRGCMER相等时,HRPWM_<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HRGCMER相等时,HRPWM_<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HRGCMER相等时,HRPWM_<t>_PWMB端口输出不受影响 11: 在向上计数期间,定时器计数值与HRGCMER相等时,HRPWM_<t>_PWMB端口输出设定为反转电平	RW
b19~b18	EXEV10UCB[1:0]	向上计数&&外部事件10发生端口状态设定B	参考EXEV1CB[1:0]说明	RW
b17~b16	EXEV9UCB[1:0]	向上计数&&外部事件9发生端口状态设定B	参考EXEV1CB[1:0]说明	RW

		定B		
b15~b14	EXEV8UCB[1:0]	向上计数&&外部事件8发生端口状态设定B	参考EXEV1CB[1:0]说明	RW
b13~b12	EXEV7UCB[1:0]	向上计数&&外部事件7发生端口状态设定B	参考EXEV1CB[1:0]说明	RW
b11~b10	EXEV6UCB[1:0]	向上计数&&外部事件6发生端口状态设定A	参考EXEV1CB[1:0]说明	RW
b9~b8	EXEV5UCB[1:0]	向上计数&&外部事件5发生端口状态设定B	参考EXEV1CB[1:0]说明	RW
b7~b6	EXEV4UCB[1:0]	向上计数&&外部事件4发生端口状态设定B	参考EXEV1CB[1:0]说明	RW
b5~b4	EXEV3UCB[1:0]	向上计数&&外部事件3发生端口状态设定B	参考EXEV1CB[1:0]说明	RW
b3~b2	EXEV2UCB[1:0]	向上计数&&外部事件2发生端口状态设定B	参考EXEV1CB[1:0]说明	RW
b1~b0	EXEV1UCB[1:0]	向上计数&&外部事件1发生端口状态设定B	00: 在向上计数期间, 外部事件1发生时, HRPWM_<t>_PWMB端口输出设定为低电平 01: 在向上计数期间, 外部事件1发生时, HRPWM_<t>_PWMB端口输出设定为高电平 10: 在向上计数期间, 外部事件1发生时, HRPWM_<t>_PWMB端口输出不受影响 11: 在向上计数期间, 外部事件1发生时, HRPWM_<t>_PWMB端口输出反转电平	RW

19.6.1.43 端口控制寄存器 A3 (HRPWM<t>_PCNAR3)

复位值: 0x000A AAAA

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				SCMBDCA[1:0]		SCMADCA[1:0]		CMFDCA[1:0]		CMEDCA[1:0]		EXEV10DCA[1:0]		EXEV9DCA[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EXEV8DCA[1:0]		EXEV7DCA[1:0]		EXEV6DCA[1:0]		EXEV5DCA[1:0]		EXEV4DCA[1:0]		EXEV3DCA[1:0]		EXEV2DCA[1:0]		EXEV1DCA[1:0]	

位/位域	标记	位名	功能	读写
b31~b28	Res	保留位	读出时为“0”,写入时写“0”	RW
b27~b26	SCMBDCA[1:0]	向下计数&&专用比较值B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SCMBR相等时,HRPWM_<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SCMBR相等时,HRPWM_<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SCMBR相等时,HRPWM_<t>_PWMA端口输出不受影响 11: 在向下计数期间,定时器计数值与SCMBR相等时,HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b25~b24	SCMADCA[1:0]	向下计数&&专用比较值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SCMAR相等时,HRPWM_<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SCMAR相等时,HRPWM_<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SCMAR相等时,HRPWM_<t>_PWMA端口输出不受影响 11: 在向下计数期间,定时器计数值与SCMAR相等时,HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b23~b22	CMFDCA[1:0]	向下计数&&比较值F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HRGCMFR相等时,HRPWM_<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HRGCMFR相等时,HRPWM_<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HRGCMFR相等时,HRPWM_<t>_PWMA端口输出不受影响 11: 在向下计数期间,定时器计数值与HRGCMFR相等时,HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b21~b20	CMEDCA[1:0]	向下计数&&比较值E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HRGCMER相等时,HRPWM_<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HRGCMER相等时,HRPWM_<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HRGCMER相等时,HRPWM_<t>_PWMA端口输出不受影响 11: 在向下计数期间,定时器计数值与HRGCMER相等时,HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b19~b18	EXEV10DCA[1:0]	向下计数&&外部事件10发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW
b17~b16	EXEV9DCA[1:0]	向下计数&&外部事件9发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW

		定A		
b15~b14	EXEV8DCA[1:0]	向下计数&&外部事件8发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW
b13~b12	EXEV7DCA[1:0]	向下计数&&外部事件7发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW
b11~b10	EXEV6DCA[1:0]	向下计数&&外部事件6发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW
b9~b8	EXEV5DCA[1:0]	向下计数&&外部事件5发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW
b7~b6	EXEV4DCA[1:0]	向下计数&&外部事件4发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW
b5~b4	EXEV3DCA[1:0]	向下计数&&外部事件3发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW
b3~b2	EXEV2DCA[1:0]	向下计数&&外部事件2发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW
b1~b0	EXEV1DCA[1:0]	向下计数&&外部事件1发生端口状态设定A	<p>00: 在向下计数期间, 外部事件1发生时, HRPWM_<t>_PWMA端口输出设定为低电平</p> <p>01: 在向下计数期间, 外部事件1发生时, HRPWM_<t>_PWMA端口输出设定为高电平</p> <p>10: 在向下计数期间, 外部事件1发生时, HRPWM_<t>_PWMA端口输出不受影响</p> <p>11: 在向下计数期间, 外部事件1发生时, HRPWM_<t>_PWMA端口输出反转电平</p>	RW

19.6.1.44 端口控制寄存器 B3 (HRPWM<t>_PCNBR3)

复位值: 0x000A AAAA

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved				SCMBDCB[1:0]		SCMADCB[1:0]		CMFDCB[1:0]		CMEDCB[1:0]		EXEV10D CA[1:0]		EXEV9D CA[1:0]	
b15		b14		b13		b12		b11		b10		b9		b8	
EXEV8DCB [1:0]		EXEV7DCB [1:0]		EXEV6DCB [1:0]		EXEV5DCB [1:0]		EXEV4DCB [1:0]		EXEV3DCB [1:0]		EXEV2DCB [1:0]		EXEV1DCB [1:0]	
位/位域	标记			位名			功能			读写					
b31~b28	Rese			保留位			读出时为“0”,写入时写“0”			RW					
b27~b26	SCMBDCB[1:0]			向下计数&&专用比较值B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SCMBR相等时,HRPWM_<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SCMBR相等时,HRPWM_<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SCMBR相等时,HRPWM_<t>_PWMB端口输出不受影响 11: 在向下计数期间,定时器计数值与SCMBR相等时,HRPWM_<t>_PWMB端口输出设定为反转电平			RW					
b25~b24	SCMADCB[1:0]			向下计数&&专用比较值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SCMAR相等时,HRPWM_<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SCMAR相等时,HRPWM_<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SCMAR相等时,HRPWM_<t>_PWMB端口输出不受影响 11: 在向下计数期间,定时器计数值与SCMAR相等时,HRPWM_<t>_PWMB端口输出设定为反转电平			RW					
b23~b22	CMFDCB[1:0]			向下计数&&比较值F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HRGCMFR相等时,HRPWM_<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HRGCMFR相等时,HRPWM_<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HRGCMFR相等时,HRPWM_<t>_PWMB端口输出不受影响 11: 在向下计数期间,定时器计数值与HRGCMFR相等时,HRPWM_<t>_PWMB端口输出设定为反转电平			RW					
b21~b20	CMEDCB[1:0]			向下计数&&比较值E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HRGCMER相等时,HRPWM_<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HRGCMER相等时,HRPWM_<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HRGCMER相等时,HRPWM_<t>_PWMB端口输出不受影响 11: 在向下计数期间,定时器计数值与HRGCMER相等时,HRPWM_<t>_PWMB端口输出设定为反转电平			RW					
b19~b18	EXEV10DCB[1:0]			向下计数&&外部事件10发生端口状态设定B			参考EXEV1DCB[1:0]说明			RW					
b17~b16	EXEV9DCB[1:0]			向下计数&&外部事件9发生端口状态设定B			参考EXEV1DCB[1:0]说明			RW					

		定B		
b15~b14	EXEV8DCB[1:0]	向下计数&&外部事件8发生端口状态设定B	参考EXEV1DCB[1:0]说明	RW
b13~b12	EXEV7DCB[1:0]	向下计数&&外部事件7发生端口状态设定B	参考EXEV1DCB[1:0]说明	RW
b11~b10	EXEV6DCB[1:0]	向下计数&&外部事件6发生端口状态设定B	参考EXEV1DCB[1:0]说明	RW
b9~b8	EXEV5DCB[1:0]	向下计数&&外部事件5发生端口状态设定B	参考EXEV1DCB[1:0]说明	RW
b7~b6	EXEV4DCB[1:0]	向下计数&&外部事件4发生端口状态设定B	参考EXEV1DCB[1:0]说明	RW
b5~b4	EXEV3DCB[1:0]	向下计数&&外部事件3发生端口状态设定B	参考EXEV1DCB[1:0]说明	RW
b3~b2	EXEV2DCB[1:0]	向下计数&&外部事件2发生端口状态设定B	参考EXEV1DCB[1:0]说明	RW
b1~b0	EXEV1DCB[1:0]	向下计数&&外部事件1发生端口状态设定B	00: 在向下计数期间, 外部事件1发生时, HRPWM_<t>_PWMB端口输出设定为低电平 01: 在向下计数期间, 外部事件1发生时, HRPWM_<t>_PWMB端口输出设定为高电平 10: 在向下计数期间, 外部事件1发生时, HRPWM_<t>_PWMB端口输出不受影响 11: 在向下计数期间, 外部事件1发生时, HRPWM_<t>_PWMB端口输出反转电平	RW

19.6.1.45 端口控制缓存寄存器 A3 (HRPWM<t>_BPCNAR3)

复位值: 0x000A AAAA

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				SCMBDCA[1:0]		SCMADCA[1:0]		CMFDCA[1:0]		CMEDCA[1:0]		EXEV10DCA[1:0]		EXEV9DCA[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EXEV8DCA[1:0]		EXEV7DCA[1:0]		EXEV6DCA[1:0]		EXEV5DCA[1:0]		EXEV4DCA[1:0]		EXEV3DCA[1:0]		EXEV2DCA[1:0]		EXEV1DCA[1:0]	

位/位域	标记	位名	功能	读写
b31~b28	Res	保留位	读出时为“0”,写入时写“0”	RW
b27~b26	SCMBDCA[1:0]	向下计数&&专用比较值B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SCMBR相等时, HRPWM_<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SCMBR相等时, HRPWM_<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SCMBR相等时, HRPWM_<t>_PWMA端口输出不受影响 11: 在向下计数期间, 定时器计数值与SCMBR相等时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b25~b24	SCMADCA[1:0]	向下计数&&专用比较值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SCMAR相等时, HRPWM_<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SCMAR相等时, HRPWM_<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SCMAR相等时, HRPWM_<t>_PWMA端口输出不受影响 11: 在向下计数期间, 定时器计数值与SCMAR相等时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b23~b22	CMFDCA[1:0]	向下计数&&比较值F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HRGCMFR相等时, HRPWM_<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HRGCMFR相等时, HRPWM_<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HRGCMFR相等时, HRPWM_<t>_PWMA端口输出不受影响 11: 在向下计数期间, 定时器计数值与HRGCMFR相等时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b21~b20	CMEDCA[1:0]	向下计数&&比较值E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HRGCMER相等时, HRPWM_<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HRGCMER相等时, HRPWM_<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HRGCMER相等时, HRPWM_<t>_PWMA端口输出不受影响 11: 在向下计数期间, 定时器计数值与HRGCMER相等时, HRPWM_<t>_PWMA端口输出设定为反转电平	RW
b19~b18	EXEV10DCA[1:0]	向下计数&&外部事件10发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW
b17~b16	EXEV9DCA[1:0]	向下计数&&外部事件9发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW

		定A		
b15~b14	EXEV8DCA[1:0]	向下计数&&外部事件8发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW
b13~b12	EXEV7DCA[1:0]	向下计数&&外部事件7发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW
b11~b10	EXEV6DCA[1:0]	向下计数&&外部事件6发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW
b9~b8	EXEV5DCA[1:0]	向下计数&&外部事件5发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW
b7~b6	EXEV4DCA[1:0]	向下计数&&外部事件4发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW
b5~b4	EXEV3DCA[1:0]	向下计数&&外部事件3发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW
b3~b2	EXEV2DCA[1:0]	向下计数&&外部事件2发生端口状态设定A	参考EXEV1DCA[1:0]说明	RW
b1~b0	EXEV1DCA[1:0]	向下计数&&外部事件1发生端口状态设定A	<p>00: 在向下计数期间, 外部事件1发生时, HRPWM_<t>_PWMA端口输出设定为低电平</p> <p>01: 在向下计数期间, 外部事件1发生时, HRPWM_<t>_PWMA端口输出设定为高电平</p> <p>10: 在向下计数期间, 外部事件1发生时, HRPWM_<t>_PWMA端口输出不受影响</p> <p>11: 在向下计数期间, 外部事件1发生时, HRPWM_<t>_PWMA端口输出反转电平</p>	RW

19.6.1.46 端口控制缓存寄存器 B3 (HRPWM<t>_BPCNBR3)

复位值: 0x000A AAAA

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				SCMBDCB[1:0]		SCMADCB[1:0]		CMFDCB[1:0]		CMEDCB[1:0]		EXEV10DCA[1:0]		EXEV9DCA[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EXEV8DCB[1:0]		EXEV7DCB[1:0]		EXEV6DCB[1:0]		EXEV5DCB[1:0]		EXEV4DCB[1:0]		EXEV3DCB[1:0]		EXEV2DCB[1:0]		EXEV1DCB[1:0]	
位/位域	标记	位名	功能	读写											
b31~b28	Res	保留位	读出时为“0”,写入时写“0”	RW											
b27~b26	SCMBDCB[1:0]	向下计数&&专用比较值B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SCMBR相等时, HRPWM_<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SCMBR相等时, HRPWM_<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SCMBR相等时, HRPWM_<t>_PWMB端口输出不受影响 11: 在向下计数期间, 定时器计数值与SCMBR相等时, HRPWM_<t>_PWMB端口输出设定为反转电平	RW											
b25~b24	SCMADCB[1:0]	向下计数&&专用比较值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SCMAR相等时, HRPWM_<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SCMAR相等时, HRPWM_<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SCMAR相等时, HRPWM_<t>_PWMB端口输出不受影响 11: 在向下计数期间, 定时器计数值与SCMAR相等时, HRPWM_<t>_PWMB端口输出设定为反转电平	RW											
b23~b22	CMFDCB[1:0]	向下计数&&比较值F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HRGCMFR相等时, HRPWM_<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HRGCMFR相等时, HRPWM_<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HRGCMFR相等时, HRPWM_<t>_PWMB端口输出不受影响 11: 在向下计数期间, 定时器计数值与HRGCMFR相等时, HRPWM_<t>_PWMB端口输出设定为反转电平	RW											
b21~b20	CMEDCB[1:0]	向下计数&&比较值E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HRGCMER相等时, HRPWM_<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HRGCMER相等时, HRPWM_<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HRGCMER相等时, HRPWM_<t>_PWMB端口输出不受影响 11: 在向下计数期间, 定时器计数值与HRGCMER相等时, HRPWM_<t>_PWMB端口输出设定为反转电平	RW											
b19~b18	EXEV10DCB[1:0]	向下计数&&外部事件10发生端口状态设定B	参考EXEV1DCB[1:0]说明	RW											
b17~b16	EXEV9DCB[1:0]	向下计数&&外部事件9发生端口状态设定B	参考EXEV1DCB[1:0]说明	RW											

		定B		
b15~b14	EXEV8DCB[1:0]	向下计数&&外部事件8发生端口状态设定B	参考EXEV1DCB[1:0]说明	RW
b13~b12	EXEV7DCB[1:0]	向下计数&&外部事件7发生端口状态设定B	参考EXEV1DCB[1:0]说明	RW
b11~b10	EXEV6DCB[1:0]	向下计数&&外部事件6发生端口状态设定B	参考EXEV1DCB[1:0]说明	RW
b9~b8	EXEV5DCB[1:0]	向下计数&&外部事件5发生端口状态设定B	参考EXEV1DCB[1:0]说明	RW
b7~b6	EXEV4DCB[1:0]	向下计数&&外部事件4发生端口状态设定B	参考EXEV1DCB[1:0]说明	RW
b5~b4	EXEV3DCB[1:0]	向下计数&&外部事件3发生端口状态设定B	参考EXEV1DCB[1:0]说明	RW
b3~b2	EXEV2DCB[1:0]	向下计数&&外部事件2发生端口状态设定B	参考EXEV1DCB[1:0]说明	RW
b1~b0	EXEV1DCB[1:0]	向下计数&&外部事件1发生端口状态设定B	00: 在向下计数期间, 外部事件1发生时, HRPWM_<t>_PWMB端口输出设定为低电平 01: 在向下计数期间, 外部事件1发生时, HRPWM_<t>_PWMB端口输出设定为高电平 10: 在向下计数期间, 外部事件1发生时, HRPWM_<t>_PWMB端口输出不受影响 11: 在向下计数期间, 外部事件1发生时, HRPWM_<t>_PWMB端口输出反转电平	RW

19.6.1.47 专用匹配事件 A 选择寄存器 (HRPWM<t>_SCMASELR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res							EEVDET	UDF6	OVF6	CMSB6	CMSA6	UDF5	OVF5	CMSB5	CMSA5
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UDF4	OVF4	CMSB4	CMSA4	UDF3	OVF3	CMSB3	CMSA3	UDF2	OVF2	CMSB2	CMSA2	UDF1	OVF1	CMSB1	CMSA1

位/位域	标记	位名	功能	读写
b31~b25	Res	保留位	读出时为“0”,写入时写“0”	RW
b24	EEVDET	外部事件检出事件使能	参考CMSA1说明	RW
b23	UDF6	单元6零点匹配事件使能	参考CMSA1说明	RW
b22	OVF6	单元6周期匹配事件使能	参考CMSA1说明	RW
b21	CMSB6	单元6专用匹配事件B使能	参考CMSA1说明	RW
b20	CMSA6	单元6专用匹配事件A使能	参考CMSA1说明	RW
b19	UDF5	单元5零点匹配事件使能	参考CMSA1说明	RW
b18	OVF5	单元5周期匹配事件使能	参考CMSA1说明	RW
b17	CMSB5	单元5专用匹配事件B使能	参考CMSA1说明	RW
b16	CMSA5	单元5专用匹配事件A使能	参考CMSA1说明	RW
b15	UDF4	单元4零点匹配事件使能	参考CMSA1说明	RW
b14	OVF4	单元4周期匹配事件使能	参考CMSA1说明	RW
b13	CMSB4	单元4专用匹配事件B使能	参考CMSA1说明	RW
b12	CMSA4	单元4专用匹配事件A使能	参考CMSA1说明	RW
b11	UDF3	单元3零点匹配事件使能	参考CMSA1说明	RW
b10	OVF3	单元3周期匹配事件使能	参考CMSA1说明	RW
b9	CMSB3	单元3专用匹配事件B使能	参考CMSA1说明	RW
b8	CMSA3	单元3专用匹配事件A使能	参考CMSA1说明	RW
b7	UDF2	单元2零点匹配事件使能	参考CMSA1说明	RW
b6	OVF2	单元2周期匹配事件使能	参考CMSA1说明	RW
b5	CMSB2	单元2专用匹配事件B使能	参考CMSA1说明	RW
b4	CMSA2	单元2专用匹配事件A使能	参考CMSA1说明	RW
b3	UDF1	单元1零点匹配事件源使能	参考CMSA1说明	RW
b2	OVF1	单元1周期匹配事件源使能	参考CMSA1说明	RW
b1	CMSB1	单元1专用匹配事件B源使能	参考CMSA1说明	RW
b0	CMSA1	单元1专用匹配事件A源使能	0: 单元1专用匹配事件SCMA不生成专用匹配事件A 1: 单元1专用匹配事件SCMA生成专用匹配事件A	RW

19.6.1.48 高分辨率控制寄存器 (HRPWM<t>_CR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	DACSRC[2: 0]		Res	DACSYN2[1:0]		Res	DACSINC1[1:0]		EN	Res					

位/位域	标记	位名	功能	读写
b31~b15	Res	保留位	读时读出0, 写时请写0	RW
b14~b12	DACSRC	DAC触发源	000: 计时器值等于0 001: 计数器值等于周期值 010: 向上计数计数值等于SCMAR 011: 向下计数计数值等于SCMAR 100: 向上计数计数值等于SCMBR 101: 向下计数计数值等于SCMBR 其他: 禁止设置	RW
b11~b10	Res	保留位	读时读出0, 写时请写0	RW
b9~b8	DACSINC2[1:0]	DAC 通道2同步源	00: 不产生DAC同步触发源 01: 在dac_trg1_ch2上产生同步触发源 10: 禁止设定 11: 禁止设定	RW
b7~b6	Res	保留位	读时读出0, 写时请写0	RW
b5~b4	DACSINC1[1:0]	DAC 通道1同步源	00: 不产生DAC同步触发源 01: 在dac_trg1_ch1上产生同步触发源 10: 在dac_trg2_ch1上产生同步触发源 11: 禁止设定	RW
b3	EN	高分辨率模式使能位	0: 不使能高精度HRPWM 1: 使能高精度HRPWM	RW
b2~b0	Res	保留位	读时读出0, 写时请写0	RW

19.6.1.49 相位控制寄存器 (HRPWM<t>_PHSCTL, t=1)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b11	Res	保留位	读出时为“0”	R
b10	BTRDPHS	相位比较基准值缓存 传送时间设定DPHS	0: 零点缓存传送点, 缓存值不传送 1: 零点缓存传送点, 发生一次缓存值传送	RW
b9	BTRUPHS	相位比较基准值缓存 传送时间设定UPHS	0: 周期缓存传送点时, 缓存值不传送 1: 周期缓存传送点, 发生一次缓存值传送	RW
b8	BENPHS	相位准值缓存传送	0: 缓存传送无效 1: 缓存传送使能	RW
b7~b0	Res	保留位	读出时为“0”, 写入时写“0”	RW

19.6.1.50 相位控制寄存器 (HRPWM<t>_PHSCTL, t=2-6)

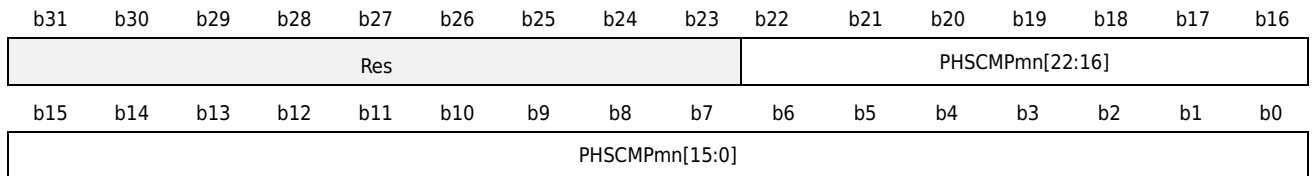
复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								PHSFORCB	PHSFORCA	Res			PHCMPSEL[2:0]	PHSEN	

位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	读出时为“0”	RW
b7	PHSFORCB	相位匹配事件发生输出B强制低使能	在锯齿波模式下: 0: 单元1单次缓存传送点检测到当前计数值大于等于比较值寄存器HRGCMDR设定的值时输出B无影响 1: 单元1单次缓存传送点检测到当前计数值大于等于比较值寄存器HRGCMDR设定的值时, 将输出B强制设置为低 三角波模式下: 0: 相位匹配事件发生时输出B无影响 1: 相位匹配事件发生时输出B强制设置为低	RW
b6	PHSFORCA	相位匹配事件发生输出A强制低使能	在锯齿波模式下: 0: 单元1单次缓存传送点检测到当前计数值大于等于比较值寄存器HRGCMCR设定的值时输出A无影响 1: 单元1单次缓存传送点检测到当前计数值大于等于比较值寄存器HRGCMCR设定的值时, 将输出A强制设置为低 三角波模式下: 0: 相位匹配事件发生时输出A无影响 1: 相位匹配事件发生时输出A强制设置为低	RW
b5~b4	Res	保留位	读出时为“0”	RW
b3~b1	PHCMPSEL[2:0]	相位匹配事件选择	000: 单元1相位匹配事件1 001: 单元1相位匹配事件2 010: 单元1相位匹配事件3 011: 单元1相位匹配事件4 100: 单元1相位匹配事件5	RW
b0	PHSEN	相位使能位	0: 不使能相位功能 1: 使能相位功能	RW

19.6.1.51 高分辨率相位比较基准值寄存器 (HRPWM<t>_PHSCMPmn, m=1~5, n=A~B)

复位值: 0x007F FFC0



位/位域	标记	位名	功能	读写
b31~b23	Res	保留位	读时读出0, 写时请写0	RW
b22~b0	PHSCMPmn	相位比较值	相位比较值设定以及缓存值 HRPWM_PHSCMPmn的值三角波模式下最大值不得大于0x7FFFC0, 锯齿波模式下最大值不得大于0x3FFFC0	RW

19.6.2 HRPWM 公共寄存器

19.6.2.1 高分辨率校准控制寄存器 (HRPWM_CALCR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				CALIE	ERRF	CALENF	Res			CALPRD[3: 0]			CALEN	CAL	

位/位域	标记	位名	功能	读写
b31~b11	Res	保留位	读时读出0, 写时请写0	RW
b10	CALIE	校准完成中断使能	0: 校准完成中断禁止 1: 校准完成中断使能	RW
b9	ERRF	校准错误标志	0: 未发生校准错误 1: 校准错误 对该位写0清0, 写1无效	RW
b8	CALENF	校准完成标志	0: 校准中或校准未使能 1: 校准完成 对该位写0清0, 写1无效	RW
b7~b6	标记	位名	读时读出0, 写时请写0	RW
b5~b2	CALPRD[3: 0]	周期性校准周期	0000: 周期为0xFFFF, 8.7ms	RW
			0001: 周期为0xFFFF*2, 17.4ms	
			0010: 周期为0xFFFF*4, 34.8ms	
			0011: 周期为0xFFFF*8, 69.6ms	
			0000: 周期为0xFFFF*16, 139.2ms	
			0001: 周期为0xFFFF*32, 278.4ms	
			0010: 周期为0xFFFF*64, 556.8ms	
			0011: 周期为0xFFFF*128, 1.11s	
			0100: 周期为0xFFFF*256, 2.22s	
			0101: 周期为0xFFFF*512, 4.45s	
0110: 周期为0xFFFF*1024, 8.91s				
0111: 周期为0xFFFF*2048, 17.81s				
1100: 周期为0xFFFF*4096, 35.63s				
1101: 周期为0xFFFF*8192, 71.27s				
1110: 周期为0xFFFF*16384, 142.54s				
1111: 周期为0xFFFF*32768, 285.08s				
b1	CALEN	周期性校准使能	0: 周期性校准关闭 1: 周期进行校准	RW
b0	CAL	单次校准使能	0: 单次校准关闭 1: 单次校准使能 该位不得与CALEN位同时设置为1	W

19.6.2.2 软件同步捕获控制寄存器 (HRPWM_SCAPR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				SCAP 6B	SCAP 6A	SCAP 5B	SCAP 5A	SCAP 4B	SCAP 4A	SCAP 3B	SCAP 3A	SCAP 2B	SCAP 2A	SCAP 1B	SCAP 1A

位/位域	标记	位名	功能	读写
b31~b12	Res	保留位	读时读出0, 写时请写0	RW
b11	SCAP6B	软件触发单元6 捕获B	0: 无影响 1: 软件触发单元6捕获B	W
b10	SCAP6A	软件触发单元6 捕获A	0: 无影响 1: 软件触发单元6捕获A	W
b9	SCAP5B	软件触发单元5 捕获B	0: 无影响 1: 软件触发单元5捕获B	W
b8	SCAP5A	软件触发单元5 捕获A	0: 无影响 1: 软件触发单元5捕获A	W
b7	SCAP4B	软件触发单元4 捕获B	0: 无影响 1: 软件触发单元4捕获B	W
b6	SCAP4A	软件触发单元4 捕获A	0: 无影响 1: 软件触发单元4捕获A	W
b5	SCAP3B	软件触发单元3 捕获B	0: 无影响 1: 软件触发单元3捕获B	W
b4	SCAP3A	软件触发单元3 捕获A	0: 无影响 1: 软件触发单元3捕获A	W
b3	SCAP2B	软件触发单元2 捕获B	0: 无影响 1: 软件触发单元2捕获B	W
b2	SCAP2A	软件触发单元2 捕获A	0: 无影响 1: 软件触发单元2捕获A	W
b1	SCAP1B	软件触发单元1 捕获B	0: 无影响 1: 软件触发单元1捕获B	W
b0	SCAP1A	软件触发单元1 捕获A	0: 无影响 1: 软件触发单元1捕获A	W

19.6.2.3 软件同步空闲控制寄存器 (HRPWM_SSTAILDR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				SSTAI DL6B	SSTAI DL6A	SSTAI DL5B	SSTAI DL5A	SSTAI DL4B	SSTAI DL4A	SSTAI DL3B	SSTAI DL3A	SSTAI DL2B	SSTAI DL2A	SSTAI DL1B	SSTAI DL1A

位/位域	标记	位名	功能	读写
b31~b12	Res	保留位	读时读出0, 写时请写0	RW
b11	SSTAIL6B	单元6B通道立 即空闲触发	参考SSTAIL1A功能描述	W
b10	SSTAIL6A	单元6 A通道立 即空闲触发	参考SSTAIL1A功能描述	W
b9	SSTAIL5B	单元5 B通道立 即空闲触发	参考SSTAIL1A功能描述	W
b8	SSTAIL5A	单元5 A通道立 即空闲触发	参考SSTAIL1A功能描述	W
b7	SSTAIL4B	单元4 B通道立 即空闲触发	参考SSTAIL1A功能描述	W
b6	SSTAIL4A	单元4 A通道立 即空闲触发	参考SSTAIL1A功能描述	W
b5	SSTAIL3B	单元3 B通道立 即空闲触发	参考SSTAIL1A功能描述	W
b4	SSTAIL3A	单元3 A通道立 即空闲触发	参考SSTAIL1A功能描述	W
b3	SSTAIL2B	单元2 B通道立 即空闲触发	参考SSTAIL1A功能描述	W
b2	SSTAIL2A	单元2 A通道立 即空闲触发	参考SSTAIL1A功能描述	W
b1	SSTAIL1B	单元1 B通道立 即空闲触发	参考SSTAIL1A功能描述	W
b0	SSTAIL1A	单元1 A通道立 即空闲触发	0: 无影响 1: 单元1A立即空闲触发	W

19.6.2.4 软件同步运行控制寄存器 1 (HRPWM_SSTARUNR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				SSTARUN6B	SSTARUN6A	SSTARUN5B	SSTARUN5A	SSTARUN4B	SSTARUN4A	SSTARUN3B	SSTARUN3A	SSTARUN2B	SSTARUN2A	SSTARUN1B	SSTARUN1A

位/位域	标记	位名	功能	读写
b31~b12	Res	保留位	读时读出0, 写时请写0	RW
b11	SSTARUN6B	单元6B通道立 即空闲退出	参考SSTARUN1A功能描述	RW
b10	SSTARUN6A	单元6 A通道立 即空闲退出	参考SSTARUN1A功能描述	RW
b9	SSTARUN5B	单元5 B通道立 即空闲退出	参考SSTARUN1A功能描述	RW
b8	SSTARUN5A	单元5 A通道立 即空闲退出	参考SSTARUN1A功能描述	RW
b7	SSTARUN4B	单元4 B通道立 即空闲退出	参考SSTARUN1A功能描述	RW
b6	SSTARUN4A	单元4 A通道立 即空闲退出	参考SSTARUN1A功能描述	RW
b5	SSTARUN3B	单元3 B通道立 即空闲退出	参考SSTARUN1A功能描述	RW
b4	SSTARUN3A	单元3 A通道立 即空闲退出	参考SSTARUN1A功能描述	RW
b3	SSTARUN2B	单元2 B通道立 即空闲退出	参考SSTARUN1A功能描述	RW
b2	SSTARUN2A	单元2 A通道立 即空闲退出	参考SSTARUN1A功能描述	RW
b1	SSTARUN1B	单元1 B通道立 即空闲退出	参考SSTARUN1A功能描述	RW
b0	SSTARUN1A	单元1 A通道立 即空闲退出	0: 单元1PWMA退出立 即空闲无效 1: 单元1PWMA退出立 即空闲有效 该位可以表示当前PWM的状态, 当该位为1时表示PWM已退出立 即空闲状态, 为0时表示PWM处于立 即空闲状态, 复位默认为0。该位在 间隔输出模式期间可以表示当前通道PWM处于运行还是空闲状态。	RW

19.6.2.5 软件同步运行控制寄存器 2 (HRPWM_SSTARUNR2)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				SSTARUN6 B2	SSTARUN6 A2	SSTARUN5 B2	SSTARUN5 A2	SSTARUN4 B2	SSTARUN4 A2	SSTARUN3 B2	SSTARUN3 A2	SSTARUN2 B2	SSTARUN2 A2	SSTARUN1 B2	SSTARUN1 A2

位/位域	标记	位名	功能	读写
b31~b12	Res	保留位	读时读出0, 写时请写0	RW
b11	SSTARUN6B2	单元6B通道立即空闲退出	参考SSTARUN1A2功能描述	W
b10	SSTARUN6A2	单元6 A通道立即空闲退出	参考SSTARUN1A2功能描述	W
b9	SSTARUN5B2	单元5 B通道立即空闲退出	参考SSTARUN1A2功能描述	W
b8	SSTARUN5A2	单元5 A通道立即空闲退出	参考SSTARUN1A2功能描述	W
b7	SSTARUN4B2	单元4 B通道立即空闲退出	参考SSTARUN1A2功能描述	W
b6	SSTARUN4A2	单元4 A通道立即空闲退出	参考SSTARUN1A2功能描述	W
b5	SSTARUN3B2	单元3 B通道立即空闲退出	参考SSTARUN1A2功能描述	W
b4	SSTARUN3A2	单元3 A通道立即空闲退出	参考SSTARUN1A2功能描述	W
b3	SSTARUN2B2	单元2 B通道立即空闲退出	参考SSTARUN1A2功能描述	W
b2	SSTARUN2A2	单元2 A通道立即空闲退出	参考SSTARUN1A2功能描述	W
b1	SSTARUN1B2	单元1 B通道立即空闲退出	参考SSTARUN1A2功能描述	W
b0	SSTARUN1A2	单元1 A通道立即空闲退出	0: 无影响 1: 退出立即空闲 在计数器停止时对该位写1后, 在其后遇到的第一个计数器启动事件PWMA进入运行状态, 在计数器计数中对该位写1无效	W

19.6.2.6 软件同步延迟空闲控制寄存器 (HRPWM_SSTADIDLR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										DLYS TRGU 6	DLYS TRGU 5	DLYS TRGU 4	DLYS TRGU 3	DLYS TRGU 2	DLYS TRGU 1

位/位域	标记	位名	功能	读写
b31~b6	Res	-	读时读出0, 写时请写0	RW
b5	DLYSTRGU6	单元6延迟空闲 软件触发	0: 无影响 1: 软件触发单元6延迟空闲 使用中需要先将对应单元DLYEVSEL[2: 0]设置为100, 再对该位写1	W
b4	DLYSTRGU5	单元5延迟空闲 软件触发	0: 无影响 1: 软件触发单元5延迟空闲 使用中需要先将对应单元DLYEVSEL[2: 0]设置为100, 再对该位写1	W
b3	DLYSTRGU4	单元4延迟空闲 软件触发	0: 无影响 1: 软件触发单元4延迟空闲 使用中需要先将对应单元DLYEVSEL[2: 0]设置为100, 再对该位写1	W
b2	DLYSTRGU3	单元3延迟空闲 软件触发	0: 无影响 1: 软件触发单元3延迟空闲 使用中需要先将对应单元DLYEVSEL[2: 0]设置为100, 再对该位写1	W
b1	DLYSTRGU2	单元2延迟空闲 软件触发	0: 无影响 1: 软件触发单元2延迟空闲 使用中需要先将对应单元DLYEVSEL[2: 0]设置为100, 再对该位写1	W
b0	DLYSTRGU1	单元1延迟空闲 软件触发	0: 无影响 1: 软件触发单元1延迟空闲 使用中需要先将对应单元DLYEVSEL[2: 0]设置为100, 再对该位写1	W

19.6.2.7 全局控制寄存器 (HRPWM_GCTLR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EE10SRC2[1:0]		EE9SRC2[1:0]		EE8SRC2[1:0]		EE7SRC2[1:0]		EE6SRC2[1:0]		EE5SRC2[1:0]		EE4SRC2[1:0]		EE3SRC2[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						BKDL Y6	BKDL Y5	BKDL Y4	BKDL Y3	BKDL Y2	BKDL Y1	EE2SRC2[1:0]		EE1SRC2[1:0]	

位/位域	标记	位名	功能	读写
b31~b30	EE2SRC2[1:0]	外部事件10源2选择	00: CMP1 01: CMP2 10: CMP3 11: 无事件源	RW
b29~b28	EE2SRC2[1:0]	外部事件9源2选择	00: CMP1 01: CMP2 10: CMP3 11: 无事件源	RW
b27~b26	EE2SRC2[1:0]	外部事件8源2选择	00: CMP1 01: CMP2 10: CMP3 11: 无事件源	RW
b25~b24	EE2SRC2[1:0]	外部事件7源2选择	00: CMP1 01: CMP2 10: CMP3 11: 无事件源	RW
b23~b22	EE2SRC2[1:0]	外部事件6源2选择	00: CMP1 01: CMP2 10: CMP3 11: 无事件源	RW
b21~b20	EE2SRC2[1:0]	外部事件5源2选择	00: CMP1 01: CMP2 10: CMP3 11: 无事件源	RW
b19~b18	EE2SRC2[1:0]	外部事件4源2选择	00: CMP1 01: CMP2 10: CMP3 11: 无事件源	RW
b17~b16	EE2SRC2[1:0]	外部事件3源2选择	00: CMP1 01: CMP2 10: CMP3 11: 无事件源	RW
b15	Res	保留位	读时读出0, 写时请写0	RW
b14~b10	Res	保留位	读时读出0, 写时请写0	RW
b9	BKDLY6	单元6消隐延迟模式	0: 消隐延迟模式不使能 1: 消隐延迟模式使能	RW
b8	BKDLY5	单元5消隐延迟模式	0: 消隐延迟模式不使能	RW

			1: 消隐延迟模式使能	
b7	BKDLY4	单元4消隐延迟模式	0: 消隐延迟模式不使能 1: 消隐延迟模式使能	RW
b6	BKDLY3	单元3消隐延迟模式	0: 消隐延迟模式不使能 1: 消隐延迟模式使能	RW
b5	BKDLY2	单元2消隐延迟模式	0: 消隐延迟模式不使能 1: 消隐延迟模式使能	RW
b4	BKDLY1	单元1消隐延迟模式	0: 消隐延迟模式不使能 1: 消隐延迟模式使能	RW
b3~b2	EE2SRC2[1:0]	外部事件2源2选择	00: CMP1 01: CMP2 10: CMP3 11: 无事件源	RW
b1~b0	EE1SRC2[1:0]	外部事件1源2选择	00: CMP1 01: CMP2 10: CMP3 11: 无事件源	RW

19.6.2.8 全局缓存控制寄存器 (HRPWM_GBCONR)

复位值: 0x0000 003F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
OSTE NU1	OSTB TRU1	Res													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res		BTRU 6SFT	BTRU 5SFT	BTRU 4SFT	BTRU 3SFT	BTRU 2SFT	BTRU 1SFT	Res		BTR U6E N	BTR U5EN	BTR U4EN	BTR U3EN	BTR U2EN	BTR U1EN

位/位域	标记	位名	功能	读写
b31	OSTENU1	单元1单次缓存使能	0: 单元1单次缓存不使能 1: 单元1单次缓存使能 注: 单元1的缓存传送点为OSTBTRU1写1后发生的第一个锯齿波计数值等于周期值或者锯齿波硬件清零或者三角波计数值等于0的点	RW
b30	OSTBTRU1	软件启动单元1单次缓存触发	0: 无影响 1: 单元1单次缓存触发	W
b29~b14	Res	保留位	读出时为“0”,写入时写“0”	RW
b13	BTRU6SFT	单元6软件缓存触发	0: 无影响 1: 软件触发单元6发生缓存 该位仅在计数器停止时有效	RW
b12	BTRU5SFT	单元5软件缓存触发	0: 无影响 1: 软件触发单元5发生缓存 该位仅在计数器停止时有效	RW
b11	BTRU4SFT	单元4软件缓存触发	0: 无影响 1: 软件触发单元4发生缓存 该位仅在计数器停止时有效	RW
b10	BTRU3SFT	单元3软件缓存触发	0: 无影响 1: 软件触发单元3发生缓存 该位仅在计数器停止时有效	RW
b9	BTRU2SFT	单元2软件缓存触发	0: 无影响 1: 软件触发单元2发生缓存 该位仅在计数器停止时有效	RW
b8	BTRU1SFT	单元1软件缓存触发	0: 无影响 1: 软件触发单元1发生缓存 该位仅在计数器停止时有效	RW
b7~b6	Res	保留位	读出时为“0”,写入时写“0”	RW
b5	BTRU6EN	单元6全局缓存使能	参考单元1全局缓存使能描述	RW
b4	BTRU5EN	单元5全局缓存使能	参考单元1全局缓存使能描述	RW
b3	BTRU4EN	单元4全局缓存使能	参考单元1全局缓存使能描述	RW
b2	BTRU3EN	单元3全局缓存使能	参考单元1全局缓存使能描述	RW
b1	BTRU2EN	单元2全局缓存使能	参考单元1全局缓存使能描述	RW
b0	BTRU1EN	单元1全局缓存使能	0: 单元1缓存使能被屏蔽 1: 无影响 该位为0时单元1的HRPWM<t>_BCONR1. BENAE<BF><P><SPA><SPB>、	RW

HRPWM<t>_DCONR.DTBENU<D> 以及
HRPWM<t>_BCONR2.BENEEFWIN <BENEEFOFF><CTL>被
屏蔽为0
同时HRPWM1_PHCTRL. BENPHS位被设置为0

19.6.2.9 全局缓存状态标志寄存器 (HRPWM_GBSFLR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										U6BT RENF	U5BT RENF	U4BT RENF	U3BT RENF	U2BT RENF	U1BT RENF

位/位域	标记	位名	功能	读写
b31~b6	Res	保留位	读出时为“0”,写入时写“0”	RW
b5	U6BTRENF	单元6全局缓存完成标志位	0: 单元6全局缓存未发生 1: 单元6全局缓存发生 全局缓存完成标志置位条件参考表 19-12。 对该位写0清0, 写1无效	RW
b4	U5BTRENF	单元5全局缓存完成标志位	0: 单元5全局缓存未发生 1: 单元5全局缓存发生 全局缓存完成标志置位条件参考表 19-12。 对该位写0清0, 写1无效	RW
b3	U4BTRENF	单元4全局缓存完成标志位	0: 单元4全局缓存未发生 1: 单元4全局缓存发生 全局缓存完成标志置位条件参考表 19-12。 对该位写0清0, 写1无效	RW
b2	U3BTRENF	单元3全局缓存完成标志位	0: 单元3全局缓存未发生 1: 单元3全局缓存发生 全局缓存完成标志置位条件参考表 19-12。 对该位写0清0, 写1无效	RW
b1	U2BTRENF	单元2全局缓存完成标志位	0: 单元2全局缓存未发生 1: 单元2全局缓存发生 全局缓存完成标志置位条件参考表 19-12。 对该位写0清0, 写1无效	RW
b0	U1BTRENF	单元1全局缓存完成标志位	0: 单元1全局缓存未发生 1: 单元1全局缓存发生 全局缓存完成标志置位条件参考表 19-12。 对该位写0清0, 写1无效	RW

19.6.2.10 间隔输出控制寄存器 (HRPWM_BMCR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res										BMM D	BENB MP	BENB MCM P	INTE NBM OVF	BMO VFF	BM OPT F
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BMT MR6	BMT MR5	BMT MR4	BMT MR3	BMT MR2	BMT MR1	BMPSC[3:0]			BMCLKS[3:0]			BMCT N	BME N		

位/位域	标记	位名	功能	读写
b31~b21	Res	保留位	读出时为“0”,写入时写“0”	RW
b21	BMMD	间隔输出模式	0: 模式1 1: 模式2	RW
b20	BENBMP	间隔输出周期基准值 缓存传送	0: 缓存传送无效 1: 缓存传送使能 (BMPERBR->BMPERAR) 在间隔输出计数器上溢出时, 发生一次缓存传送	RW
b19	BENBMCMP	间隔输出比较基准值 缓存传送	0: 缓存传送无效 1: 缓存传送使能 (BMCMBR->BMCMAR) 在间隔输出计数器上溢出时, 发生一次缓存传送	RW
b18	INTENBMOVFF	BM-counter上溢中断 使能	0: BM-counter计数上溢时, 该中断无效 1: BM-counter计数上溢时, 该中断使能	RW
b17	BMOVFF	BM-counter上溢匹配	0: BM-counter未发生上溢 1: 发生BM-counter上溢 对该位写0清0, 写1无效	RW
b16	BMOPTF	间隔输出运行标志	0: 常规运行 1: 间隔输出正在运行, 该位写0间隔输出提前结束 该标志位由硬件置位, 硬件清零, 当间隔输出使能 (BMEN=1) 且HRPWM_BMSTRG1/2所选择的触发源发生 时, 该位发生置位, 对该位写0或者将BMEN设置位0后, 再所 有使能了间隔输出的通道退出间隔输出后, 该位自动清零	RW
b15	BMTMR6	间隔输出单元6计数器 控制	0: 间隔输出下单元6计数器正常运行 1: 间隔输出下单元6计数器停止计数, 且计数器清零	RW
b14	BMTMR5	间隔输出单元5计数器 控制	0: 间隔输出下单元5计数器正常运行 1: 间隔输出下单元5计数器停止计数, 且计数器清零	RW
b13	BMTMR4	间隔输出单元4计数器 控制	0: 间隔输出下单元4计数器正常运行 1: 间隔输出下单元4计数器停止计数, 且计数器清零	RW
b12	BMTMR3	间隔输出单元3计数器 控制	0: 间隔输出下单元3计数器正常运行 1: 间隔输出下单元3计数器停止计数, 且计数器清零	RW
b11	BMTMR2	间隔输出单元2计数器 控制	0: 间隔输出下单元2计数器正常运行 1: 间隔输出下单元2计数器停止计数, 且计数器清零	RW
b10	BMTMR1	间隔输出单元1计数器 控制	0: 间隔输出下单元1计数器正常运行 1: 间隔输出下单元1计数器停止计数, 且计数器清零	RW
b9~b6	BMPSC[3:0]	间隔输出时钟分频	BMPSC[3:0] 计数时钟选择 0000: PCLK0 0001: PCLK0/2 0010: PCLK0/4 0011: PCLK0/8	RW

			0100: PCLK0/16 0101: PCLK0/32 0110: PCLK0/64 0111: PCLK0/128 1000: PCLK0/256 1001: PCLK0/512 1010: PCLK0/1024 1011: PCLK0/2048 1100: PCLK0/4096 1101: PCLK0/8192 1110: PCLK0/16384 1111: PCLK0/32768	
b5~b2	BMCLKS[3:0]	间隔输出计数时钟选择	0000: 单元1完整周期点做为计数条件 0001: 单元2完整周期点做为计数条件 0010: 单元3完整周期点做为计数条件 0011: 单元4锯完整周期点做为计数条件 0100: 单元5锯完整周期点做为计数条件 0101: 单元6锯完整周期点做为计数条件 0110: TMR0_1_CMPA 0111: TMR0_1_CMPB 1000: TMR0_2_CMPA 1001: TMR0_2_CMPB 1010: PCLK0的分频 (根据BMPS[3:0]设置) 注: 完整周期点由HRPWM<t>_GCONR1.PRDSEL定义	RW
b1	BMCTN	间隔输出下的连续模式	0: 单次模式, BM计数器达到HRPWM_BMCAR的值时, 计数停止。 1: 连续模式, BM计数器达到HRPWM_BMCAR的值时, 翻转到0并继续计数。	RW
b0	BMEN	间隔输出使能	0: 间隔输出禁用 1: 间隔输出使能 该位为1时, 间隔输出控制器已准备好接收间隔输出启动触发。该位写0时将终止间隔输出	RW

19.6.2.11 间隔输出启动触发寄存器 1 (HRPWM_BMSTRG1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	EEV8	EEV7	T4UD FEEV 8	T4OV FEEV 8	T1UD FEEV 7	T1OV FEEV 7	GCM B6	GCM A6	UDF6	OVF6	GCM B5	GCM A5	UDF5	OVF5	GCM B4
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GCM A4	UDF4	OVF4	GCM B3	GCM A3	UDF3	OVF3	GCM B2	GCM A2	UDF2	OVF2	GCM B1	GCM A1	UDF1	OVF1	SSTR G

位/位域	标记	位名	功能	读写
b31	Res	保留位	读出时为“0”,写入时写“0”	RW
b30	EEV8	外部事件8触发间隔输出	条件: 外部事件8间隔输出有效 (由单元4滤波) 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b29	EEV7	外部事件7触发间隔输出	条件: 外部事件7间隔输出有效 (由单元1滤波) 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b28	T4UDFEEV8	外部事件8之后的单元4下溢事件触发间隔输出	条件: 外部事件8之后的单元4计数值等于0触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b27	T4OVFEEV8	外部事件8之后的单元4周期匹配事件触发间隔输出	条件: 外部事件8之后的单元4计数值等于周期值或者锯齿波硬件清零触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b26	T1UDFEEV7	外部事件7之后的单元1零点匹配事件触发间隔输出	条件: 外部事件7之后的单元1计数值等于0触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b25	T1OVFEEV7	外部事件7之后的单元1周期匹配事件触发间隔输出	条件: 外部事件7之后的单元1计数值等于周期值或者锯齿波应将清零触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b24	GCM B6	单元6比较匹配事件B触发间隔输出	条件: 单元6比较匹配事件B触发间隔输出 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b23	GCM A6	单元6比较匹配事件A触发间隔输出	条件: 单元6比较匹配事件A触发间隔输出 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b22	UDF6	单元6零点匹配事件触发间隔输出	条件: 单元6零点匹配事件触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b21	OVF6	单元6周期匹配事件触发间隔输出	条件: 单元6周期匹配事件触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b20	GCM B5	单元5比较匹配事件B触发间隔输出	条件: 单元3比较匹配事件B触发间隔输出 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b19	GCM A5	单元5比较匹配事件A	条件: 单元3比较匹配事件A触发间隔输出	RW

		触发间隔输出	0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	
b18	UDF5	单元5零点匹配事件触发间隔输出	条件: 单元4零点匹配事件触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b17	OVF5	单元5周期匹配事件触发间隔输出	条件: 单元4周期匹配事件触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b16	GCMB4	单元4比较匹配事件B触发间隔输出	条件: 单元3比较匹配事件B触发间隔输出 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b15	GCMA4	单元4比较匹配事件A触发间隔输出	条件: 单元3比较匹配事件A触发间隔输出 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b14	UDF4	单元4零点匹配事件触发间隔输出	条件: 单元4零点匹配事件触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b13	OVF4	单元4周期匹配事件触发间隔输出	条件: 单元4周期匹配事件触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b12	GCMB3	单元3比较匹配事件B触发间隔输出	条件: 单元3比较匹配事件B触发间隔输出 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b11	GCMA3	单元3比较匹配事件A触发间隔输出	条件: 单元3比较匹配事件A触发间隔输出 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b10	UDF3	单元3零点匹配事件触发间隔输出	条件: 单元3零点匹配事件触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b9	OVF3	单元3周期匹配事件触发间隔输出	条件: 单元3周期匹配事件触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b8	GCMB2	单元2比较匹配事件B触发间隔输出	条件: 单元2比较匹配事件B触发间隔输出 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b7	GCMA2	单元2比较匹配事件A触发间隔输出	条件: 单元2比较匹配事件A触发间隔输出 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b6	UDF2	单元2零点匹配事件触发间隔输出	条件: 单元2零点匹配事件触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b5	OVF2	单元2周期匹配事件触发间隔输出	条件: 单元2周期匹配事件触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b4	GCMB1	单元1比较匹配事件B触发间隔输出	条件: 单元1比较匹配事件B触发间隔输出 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b3	GCMA1	单元1比较匹配事件A触发间隔输出	条件: 单元1比较匹配事件A触发间隔输出	RW

		触发间隔输出	0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	
b2	UDF1	单元1零点匹配事件触发间隔输出	条件: 单元1零点匹配事件触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b1	OVF1	单元1周期匹配事件触发间隔输出	条件: 单元1周期匹配事件触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出	RW
b0	SSTRG	软件触发间隔输出	对该位写1触发间隔输出。如果间隔输出未使能 (HRPWM_BMCR.BMEN=0), 则该位无效。 0: 对间隔输出无影响 1: 软件触发启动间隔输出	W

19.6.2.12 间隔输出启动触发寄存器 2 (HRPWM_BMSTRG2)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												IEV3	IEV2	IEV1	IEV0
位/位域	标记	位名	功能										读写		
b31~b4	Res	保留位	读出时为“0”,写入时写“0”										RW		
b3	IEV3	TMRO_2_CMPB触发间隔输出	条件: TMRO_2_CMPB触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出										RW		
b2	IEV2	TMRO_2_CMPA触发间隔输出	条件: TMRO_2_CMPA触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出										RW		
b1	IEV1	TMRO_1_CMPB触发间隔输出	条件: TMRO_1_CMPB触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出										RW		
b0	IEV0	TMRO_1_CMPA触发间隔输出	条件: TMRO_1_CMPA触发间隔输出有效 0: 条件匹配时, 对间隔输出无影响 1: 条件匹配时, 启动间隔输出										RW		

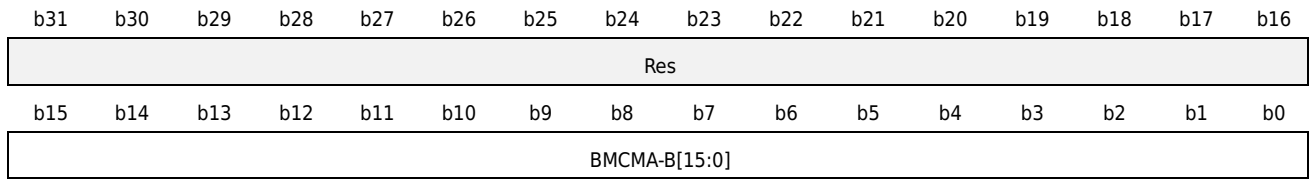
19.6.2.13 间隔输出周期基准值寄存器 (HRPWM_BMPERM, m=A~B)

复位值: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BMPERA-C[15:0]															
位/位域	标记	位名	功能										读写		
b31~b16	Res	保留位	读出时为“0”										R		
b15~b0	BMPERA-B[15:0]	间隔输出计数器计数周期值	设定每轮计数的计数周期值及对应缓存值, 该周期为空闲状态和运行状态持续时间之和。间隔输出使能时, 该寄存器不能是0。										RW		

19.6.2.14 间隔输出比较基准值寄存器 (HRPWM_BCMmR, m=A~B)

复位值: 0x0000 FFFF



位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”	R
b15~b0	BMCMA-B[15:0]	间隔输出比较基准值	该位包含了与BM计数器进行比较的值及其缓存值，并定义了间隔输出下空闲的持续时间。	RW

19.6.2.15 外部事件控制寄存器 1 (HRPWM_EECR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res		EE5FAST	EE5SNS[1:0]		EE5POL	EE5SRC[1:0]		EE4FAST	EE4SNS[1:0]		EE4POL	EE4SRC[1:0]		EE3FAST	EE3SNS[1]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EE3SNS[0]	EE3POL	EE3SRC[1:0]		EE2FAST	EE2SNS[1:0]		EE2POL	EE2SRC[1:0]		EE1FAST	EE1SNS[1:0]		EE1POL	EE1SRC[1:0]	

位/位域	标记	位名	功能	读写
b31~b30	Res	保留位	读出时为“0”,写入时写“0”	RW
b29	EE5FAST	外部事件5快速异步模式	参考EE1FAST的描述	RW
b28~b27	EE5SNS[1:0]	外部事件5有效边沿	参考EE1SNS的描述	RW
b26	EE5POL	外部事件5极性	参考EE1POL的描述	RW
b25~b24	EE5SRC[1:0]	外部事件5源	参考EE1SRC[1:0]的描述	RW
b23	EE4FAST	外部事件4快速异步模式	参考EE1FAST的描述	RW
b22~b21	EE4SNS[1:0]	外部事件4有效边沿	参考EE1SNS的描述	RW
b20	EE4POL	外部事件4极性	参考EE1POL的描述	RW
b19~b18	EE4SRC[1:0]	外部事件4源	参考EE1SRC[1:0]的描述	RW
b17	EE3FAST	外部事件3快速异步模式	参考EE1FAST的描述	RW
b16~b15	EE3SNS[1:0]	外部事件3有效边沿	参考EE1SNS的描述	RW
b14	EE3POL	外部事件3极性	参考EE1POL的描述	RW
b13~b12	EE3SRC[1:0]	外部事件3源	参考EE1SRC[1:0]的描述	RW
b11	EE2FAST	外部事件2快速异步模式	参考EE1FAST的描述	RW
b10~b9	EE2SNS[1:0]	外部事件2有效边沿	参考EE1SNS的描述	RW
b8	EE2POL	外部事件2极性	参考EE1POL的描述	RW
b7~b6	EE2SRC[1:0]	外部事件2源	参考EE1SRC[1:0]的描述	RW
b5	EE1FAST	外部事件1快速异步模式	0: 外部事件1快速异步模式不使能 1: 外部事件1快速异步模式使能	RW
b4~b3	EE1SNS[1:0]	外部事件1有效边沿	00: 电平有效,有效电平由EE1POL配置 01: 上升沿有效 10: 下降沿有效 11: 双边沿有效	RW
b2	EE1POL	外部事件1极性	该位仅在EE1SNS[1:0]=00时有效 0: 外部事件1高电平有效 1: 外部事件1低电平有效	
b1~b0	EE1SRC[1:0]	外部事件1源	00: 外部事件1的源为EE1src1 01: 外部事件1的源为EE1src2 10: 外部事件1的源为EE1src3 11: 外部事件1的源为EE1src4 当寄存器设置为01时, EE1src2的源根据 HRPWM_GCTLR.EE1SRC2[1:0]进行配置, 每个事件对应 的事件源参考表 19-5	RW

19.6.2.16 外部事件控制寄存器 2 (HRPWM_EECCR2)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res		EE10SNS[1:0]		EE10POL	EE10SRC[1:0]		Res	EE9SNS[1:0]		EE9POL	EE9SRC[1:0]		Res	EE9SNS[1]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EE8SNS[0]	EE8POL	EE8SRC[1:0]		Res	EE7SNS[1:0]		EE7POL	EE7SRC[1:0]		Res	EE6SNS[1:0]		EE6POL	EE6SRC[1:0]	

位/位域	标记	位名	功能	读写
b31~b29	Res	保留位	读出时为“0”,写入时写“0”	RW
b28~b27	EE10SNS[1:0]	外部事件10有效边沿	参考EE1SNS的描述	RW
b26	EE10POL	外部事件10极性	参考EE1POL的描述	RW
b25~b24	EE10SRC[1:0]	外部事件10源	参考EE1SRC[1:0]的描述	RW
b23	Res	保留位	读出时为“0”,写入时写“0”	RW
b22~b21	EE9SNS[1:0]	外部事件9有效边沿	参考EE1SNS的描述	RW
b20	EE9POL	外部事件9极性	参考EE1POL的描述	RW
b19~b18	EE9SRC[1:0]	外部事件9源	参考EE1SRC[1:0]的描述	RW
b17	Res	保留位	读出时为“0”,写入时写“0”	RW
b16~b15	EE8SNS[1:0]	外部事件8有效边沿	参考EE1SNS的描述	RW
b14	EE8POL	外部事件8极性	参考EE1POL的描述	RW
b13~b12	EE8SRC[1:0]	外部事件8源	参考EE1SRC[1:0]的描述	RW
b11	Res	保留位	读出时为“0”,写入时写“0”	RW
b10~b9	EE7SNS[1:0]	外部事件7有效边沿	参考EE1SNS的描述	RW
b8	EE7POL	外部事件7极性	参考EE1POL的描述	RW
b7~b6	EE7SRC[1:0]	外部事件7源	参考EE1SRC[1:0]的描述	RW
b5	Res	保留位	读出时为“0”,写入时写“0”	RW
b4~b3	EE6SNS[1:0]	外部事件6有效边沿	参考EE1SNS的描述	RW
b2	EE6POL	外部事件6极性	参考EE1POL的描述	RW
b1~b0	EE6SRC[1:0]	外部事件6源	参考EE1SRC[1:0]的描述	RW

19.6.2.17 外部事件控制寄存器 3 (HRPWM_EECR3)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EEVSD[1:0]		Res		EE10F[3:0]				Res		EE9F[3:0]				Res	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EE8F[3:0]				Res		EE7F[3:0]				Res		EE6F[3:0]			

位/位域	标记	位名	功能	读写
b31~b30	EEVSD[1:0]	外部事件采样时钟分频	00: $f_{EEVS}=f_{PCLK0}$ 01: $f_{EEVS}=f_{PCLK0}/2$ 10: $f_{EEVS}=f_{PCLK0}/4$ 11: $f_{EEVS}=f_{PCLK0}/8$	RW
b29~b28	Res	保留位	读出时为“0”,写入时写“0”	RW
b27~b24	EE10F[3:0]	外部事件10滤波	参考EE6F的描述	RW
b23~b22	Res	保留位	读出时为“0”,写入时写“0”	RW
b21~b18	EE9F[3:0]	外部事件9滤波	参考EE6F的描述	RW
b17~b16	Res	保留位	读出时为“0”,写入时写“0”	RW
b15~b12	EE8F[3:0]	外部事件8滤波	参考EE6F的描述	RW
b11~b10	Res	保留位	读出时为“0”,写入时写“0”	RW
b9~b6	EE7F[3:0]	外部事件7滤波	参考EE6F的描述	RW
b5~b4	Res	保留位	读出时为“0”,写入时写“0”	RW
该位定义了采样外部事件的频率, 以及应用于外部事件的滤波器长度。 0000: 无滤波 0001: 采样频率为 f_{PCLK0} , $N=2$ 0010: 采样频率为 f_{PCLK0} , $N=4$ 0011: 采样频率为 f_{PCLK0} , $N=8$ 0100: 采样频率为 $f_{EEVS}/2$, $N=6$ 0101: 采样频率为 $f_{EEVS}/2$, $N=8$ 0110: 采样频率为 $f_{EEVS}/4$, $N=6$ 0111: 采样频率为 $f_{EEVS}/4$, $N=8$ 1000: 采样频率为 $f_{EEVS}/8$, $N=6$ 1001: 采样频率为 $f_{EEVS}/8$, $N=8$ 1010: 采样频率为 $f_{EEVS}/16$, $N=5$ 1011: 采样频率为 $f_{EEVS}/16$, $N=6$ 1100: 采样频率为 $f_{EEVS}/16$, $N=8$ 1101: 采样频率为 $f_{EEVS}/32$, $N=5$ 1110: 采样频率为 $f_{EEVS}/32$, $N=6$ 1111: 采样频率为 $f_{EEVS}/32$, $N=8$				
b3~b0	EE6F[3:0]	外部事件6滤波		RW

19.6.2.18 同步输出控制寄存器 (HRPWM_SYNOCR)

复位值: 0x0000 1000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SYNCMP[7:0]								Res	SRCD IR	SYNOPLS[1:0]	SYNCSRC[3:0]				

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”,写入时写“0”	RW
b15~b8	SYNCMP[7:0]	同步输出脉冲宽度	在HRPWM_SCOUT上产生脉冲长度为SYNCMP[7:0]个PCLK0 该位必须设置为大于等于16	RW
b7	Res	保留位	读出时为“0”,写入时写“0”	RW
b6	SRCDIR	同步源计数方向	0: 同步源在向下计数区间有效 1: 同步源在向上计数区间有效	RW
b5~b4	SYNOPLS[1:0]	同步输出脉冲	0x: 脉冲产生禁能 10: 在HRPWM_SCOUT上产生正脉冲, 长度为SYNCMP[7:0] 个PCLK0 11: 在HRPWM_SCOUT上产生负脉冲, 长度为SYNCMP[7:0] 个PCLK0	RW
b3~b0	SYNCSRC[3:0]	同步源	0000: 单元1的定时器计数值为0 0001: 单元2的定时器计数值为0 0010: 单元3的定时器计数值为0 0011: 单元4的定时器计数值为0 0100: 单元5的定时器计数值为0 0101: 单元6的定时器计数值为0 0110: 单元1的定时器计数值为SCMB 0111: 单元2的定时器计数值为SCMB 1000: 单元3的定时器计数值为SCMB 1001: 单元4的定时器计数值为SCMB 1010: 单元5的定时器计数值为SCMB 1011: 单元6的定时器计数值为SCMB 当设定值为0110~1011, 需要配置SRCDIR位设定比较匹配事件的 有效计数区间	RW

19.6.2.19 外部事件检出事件选择寄存器 (HRPWM_EEDSELR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						EEV10	EEV9	EEV8	EEV7	EEV6	EEV5	EEV4	EEV3	EEV2	EEV1

位/位域	标记	位名	功能	读写
b31~b10	Res	保留位	读出时为“0”,写入时写“0”	RW
b9	EEV10	外部事件10选择	参考EEV1说明	RW
b8	EEV9	外部事件9选择	参考EEV1说明	RW
b7	EEV8	外部事件8选择	参考EEV1说明	RW
b6	EEV7	外部事件7选择	参考EEV1说明	RW
b5	EEV6	外部事件6选择	参考EEV1说明	RW
b4	EEV5	外部事件5选择	参考EEV1说明	RW
b3	EEV4	外部事件4选择	参考EEV1说明	RW
b2	EEV3	外部事件3选择	参考EEV1说明	RW
b1	EEV2	外部事件2选择	参考EEV1说明	RW
b0	EEV1	外部事件1选择	0: 外部事件1发生时不成外部事件检出事件 1: 外部事件1发生时生成外部事件检出事件	RW

19.6.2.20 滤波控制寄存器 (HRPWM_FCNTD)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	NOFI CKTD[1:0]	NOFI ENTD	Res	NOFI CKTC[1:0]	NOFI ENTC	Res	NOFI CKTB[1:0]	NOFI ENTB	Res	NOFI CKTA[1:0]	NOFI ENTA				

位/位域	标记	位名	功能	读写
b31~b15	Res	保留位	读出时为“0”,写入时写“0”	RW
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	RW
b12	NOFIENDT	捕获输入端口滤 波TD	0: HRPWM_TRIGD端口输入滤波功能无效 1: HRPWM_TRIGD端口输入滤波功能使能	RW
b11	Res	保留位	读出时为“0”,写入时写“0”	RW
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	RW
b8	NOFIENTC	捕获输入端口滤 波TC	0: HRPWM_TRIGC端口输入滤波功能无效 1: HRPWM_TRIGC端口输入滤波功能使能	RW
b7	Res	保留位	读出时为“0”,写入时写“0”	RW
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	RW
b4	NOFIENBT	捕获输入端口滤 波TB	0: HRPWM_TRIGB端口输入滤波功能无效 1: HRPWM_TRIGB端口输入滤波功能使能	RW
b3	Res	保留位	读出时为“0”,写入时写“0”	RW
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	RW
b0	NOFIENTA	捕获输入端口滤 波TA	0: HRPWM_TRIGA端口输入滤波功能无效 1: HRPWM_TRIGA端口输入滤波功能使能	RW

19.6.2.21 软件同步启动控制寄存器 (HRPWM_SSTAR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										SSTA 6	SSTA 5	SSTA 4	SSTA 3	SSTA 2	SSTA 1

位/位域	标记	位名	功能	读写
b31~b6	Res	保留位	读出时为“0”,写入时写“0”	RW
b5	SSTA6	单元6软件启动	0: 软件启动无效 1: 软件启动使能	RW
b4	SSTA5	单元5软件启动	0: 软件启动无效 1: 软件启动使能	RW
b3	SSTA4	单元4软件启动	0: 软件启动无效 1: 软件启动使能	RW
b2	SSTA3	单元3软件启动	0: 软件启动无效 1: 软件启动使能	RW
b1	SSTA2	单元2软件启动	0: 软件启动无效 1: 软件启动使能	RW
b0	SSTA1	单元1软件启动	0: 软件启动无效 1: 软件启动使能	RW

19.6.2.22 软件同步停止控制寄存器 (HRPWM_SSTPR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										SSTP 6	SSTP 5	SSTP 4	SSTP 3	SSTP 2	SSTP 1

位/位域	标记	位名	功能	读写
b31~b6	Res	保留位	读出时为“0”,写入时写“0”	RW
b5	SSTP6	单元6软件停止	0: 软件停止无效 1: 软件停止使能	W
b4	SSTP5	单元5软件停止	0: 软件停止无效 1: 软件停止使能	W
b3	SSTP4	单元4软件停止	0: 软件停止无效 1: 软件停止使能	W
b2	SSTP3	单元3软件停止	0: 软件停止无效 1: 软件停止使能	W
b1	SSTP2	单元2软件停止	0: 软件停止无效 1: 软件停止使能	W
b0	SSTP1	单元1软件停止	0: 软件停止无效 1: 软件停止使能	W

19.6.2.23 软件同步清零控制寄存器 (HRPWM_SCLRR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										SACLE6	SACLE5	SACLE4	SACLE3	SACLE2	SACLE1

位/位域	标记	位名	功能	读写
b31~b6	Res	保留位	读出时为“0”,写入时写“0”	RW
b5	SACLE6	单元6软件清零	0: 软件清零无效 1: 软件清零使能	W
b4	SACLE5	单元5软件清零	0: 软件清零无效 1: 软件清零使能	W
b3	SACLE4	单元4软件清零	0: 软件清零无效 1: 软件清零使能	W
b2	SACLE3	单元3软件清零	0: 软件清零无效 1: 软件清零使能	W
b1	SACLE2	单元2软件清零	0: 软件清零无效 1: 软件清零使能	W
b0	SACLE1	单元1软件清零	0: 软件清零无效 1: 软件清零使能	W

19.6.2.24 软件同步刷新控制寄存器 (HRPWM_SUPDR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										SUPD 6	SUPD 5	SUPD 4	SUPD 3	SUPD 2	SUPD 1

位/位域	标记	位名	功能	读写
b31~b6	Res	保留位	读出时为“0”,写入时写“0”	RW
b5	SUPD6	单元6软件刷新	0: 软件刷新无效 1: 软件刷新使能	W
b4	SUPD5	单元5软件刷新	0: 软件刷新无效 1: 软件刷新使能	W
b3	SUPD4	单元4软件刷新	0: 软件刷新无效 1: 软件刷新使能	W
b2	SUPD3	单元3软件刷新	0: 软件刷新无效 1: 软件刷新使能	W
b1	SUPD2	单元2软件刷新	0: 软件刷新无效 1: 软件刷新使能	W
b0	SUPD1	单元1软件刷新	0: 软件刷新无效 1: 软件刷新使能	W

19.7 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。软件同步刷新可再计数器计数过程中将计数器的值更新为任意值，这可能会导致比较匹配事件的丢失，在使用时需注意。
- 2) HRGCMCR、HRGCMR 在对应的缓存功能有效时，其中断输出、事件输出无效。
- 3) 高精度 PWM 输出的脉冲宽度存在一定限制，最小脉冲为 3 个 PCLK0 周期，最大脉宽为 PWM 周期值-3 个 PCLK0 的周期，在设定基准值时，确保脉冲宽度符合该限制，否则 PWM 将无法保证高精度。
- 4) 在使用互补 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。

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 19-17 所示。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 19-17 所示。

表 19-17 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
清零事件（软件清零或硬件清零）	锯齿波模式时	0	↓
	三角波模式时	0	↓
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

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 (HRPWM_<t>_PWMA 和 HRPWM_<t>_PWMB) 的输出变化控制优先级分别如下表 19-18 所示。

表 19-18 PWMA、B 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		PCNAR1.EMBCA设定状态	高
空闲状态		IDLECR.IDLESA	↓
设定强制端口输出		PCNAR1.FORCA设定状态	↓
计数开始时		PCNAR1.STACA设定状态	↓
锯齿波向上计数时	外部事件发生时	PCNAR2.EVEyUCA设定状态	↓
	相位强制输出	PHSCTL.PHSFORCA设定	↓
	计数值为HRPERAR	PCNAR1.OVFCA设定状态	↓
	计数值为HRGCMAR	PCNAR1.CMAUCA设定状态	↓
	计数值为HRGCMBR	PCNAR1.CMBUCA设定状态	↓
	计数值为HRGCMER	PCNAR2.CMEUCA设定状态	↓
	计数值为HRGCMFR	PCNAR2.CMFUCA设定状态	↓
	计数值为SCMAR	PCNAR2.SCMAUCA设定状态	↓
	计数值为SCMBR	PCNAR2.SCMBUCA设定状态	↓
计数值为0	PCNAR1.UDFCA设定状态	↓	
三角波向上计数时	外部事件发生时	PCNAR2.EVEyUCA设定状态	↓
	相位强制输出	PHSCTL.PHSFORCA设定	↓
	计数值为HRGCMAR	PCNAR1.CMAUCA设定状态	↓
	计数值为HRGCMBR	PCNAR1.CMBUCA设定状态	↓
	计数值为HRGCMER	PCNAR2.CMEUCA设定状态	↓
	计数值为HRGCMFR	PCNAR2.CMFUCA设定状态	↓
	计数值为SCMAR	PCNAR2.SCMAUCA设定状态	↓
	计数值为SCMBR	PCNAR2.SCMBUCA设定状态	↓
计数值为0	PCNAR1.UDFCA设定状态	↓	
三角波向下计数时	外部事件发生时	PCNAR3.EVEyDCA设定状态	↓
	相位强制输出	PHSCTL.PHSFORCA设定	↓
	计数值为HRGCMAR	PCNAR1.CMADCA设定状态	↓
	计数值为HRGCMBR	PCNAR1.CMBDCA设定状态	↓
	计数值为HRGCMER	PCNAR3.CMEDCA设定状态	↓
	计数值为HRGCMFR	PCNAR3.CMFDCA设定状态	↓
	计数值为SCMAR	PCNAR3.SCMAADCA设定状态	↓
	计数值为SCMBR	PCNAR3.SCMBDCA设定状态	↓
计数值为HRPERAR	PCNAR1.OVFCA设定状态	↓	
计数停止时		PCNAR1.STPCA设定状态	低

注：外部事件同时发生时,置 0 优先级大于置 1 大于翻转。

20 通用控制定时器（Timer4）

20.1 概述

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。

20.2 主要特性

Timer4 基本的功能及特性如表 20-1 所示。

表 20-1 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
	上溢、下溢、重载计数匹配事件
	专用比较匹配事件
	硬件启动计数
	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

20.3 功能说明

20.3.1 Timer4 框图

图 20-1 所示，描述了通用控制定时器 Timer4 的基本架构。

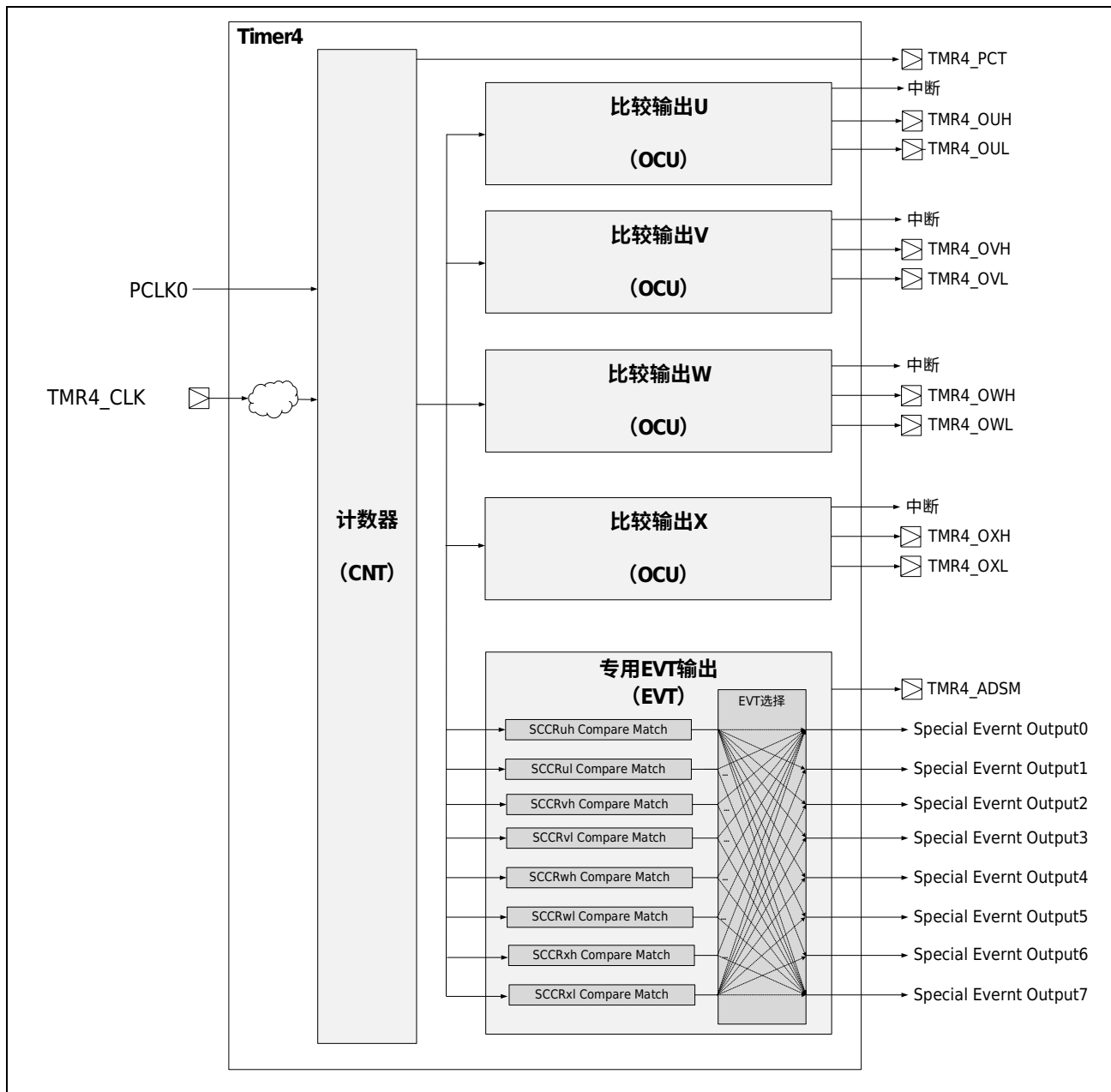


图 20-1 Timer4 基本框图

表 20-2 所示为 Timer4 的输入输出端口列表。

表 20-2 Timer4 端口列表

端口名	方向	功能
TMR4_CLK	输入	计数时钟输入端口
TMR4_PCT	输出	PWM周期输出监测
TMR4_ADSM	输出	专用事件输出监测
TMR4_OUH	输出	PWM输出端口
TMR4_OUL		
TMR4_OVH		
TMR4_OVL		
TMR4_OWH		
TMR4_OWL		
TMR4_OXH		
TMR4_OXL		

20.3.2 基本动作

20.3.2.1 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 20-2 和图 20-3 分别是锯齿波和三角波的波形图。

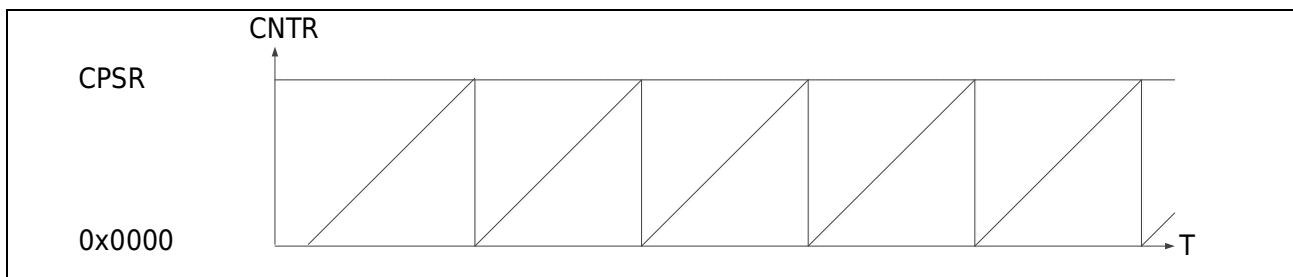


图 20-2 Timer4 锯齿波波形

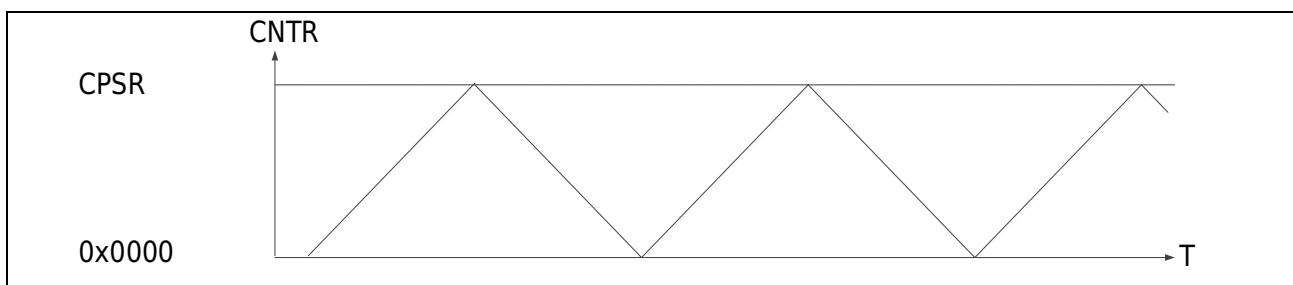


图 20-3 Timer4 三角波波形

20.3.2.2 计数动作

1) 锯齿波计数操作与控制流程如图 20-4 所示。

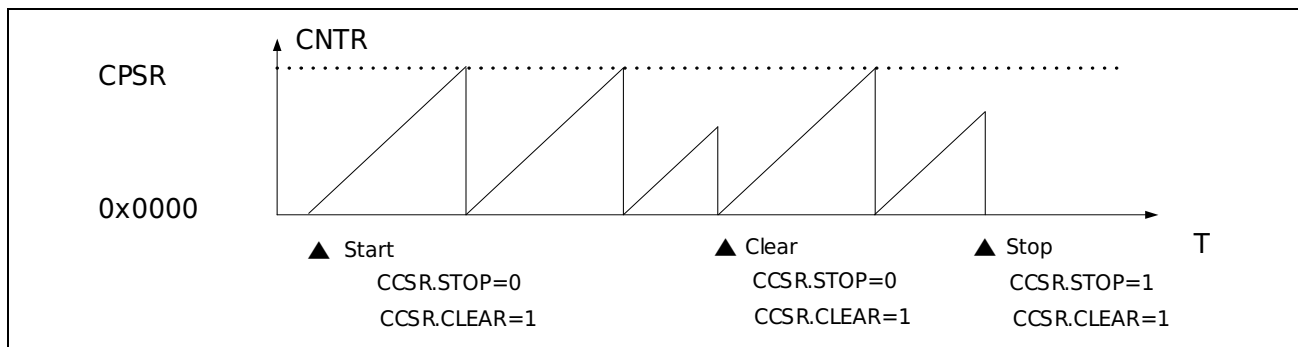


图 20-4 Timer4 锯齿波模式计数动作

1. 设置模式 $CCSR.MODE=0$;
2. 设置计数峰值 CPSR 寄存器;
3. 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;
4. 计数周期 = $(CPSR+1) \times$ 计数时钟周期;
5. 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;
6. 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 20-5 所示。

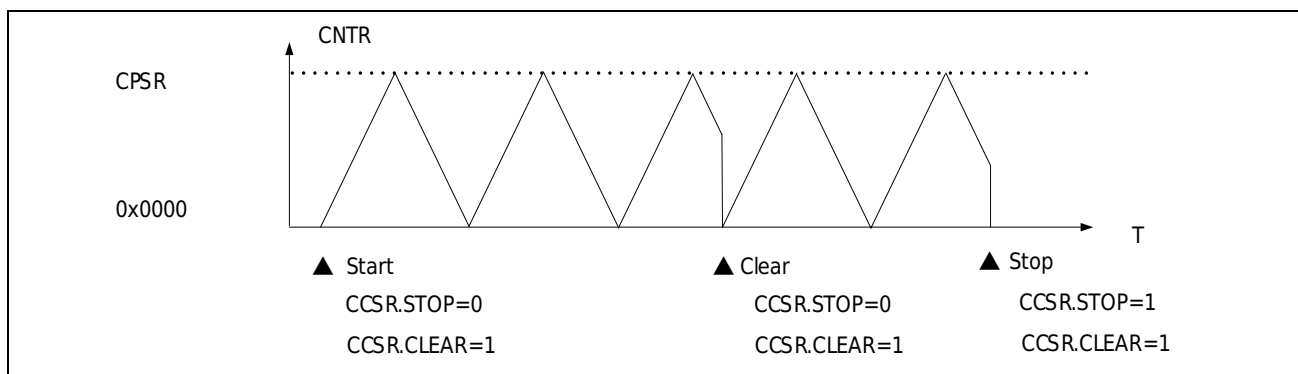


图 20-5 Timer4 三角波模式计数动作

1. 设置模式 $CCSR.MODE=1$;
2. 设置计数峰值 CPSR 寄存器;

3. 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;
4. 计数周期 = (CPSR) × 2 × 计数时钟周期;
5. 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;
6. 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。

20.3.2.3 比较输出

1) 下图 20-6 是锯齿波计数模式下的比较输出模块的波形输出示例。

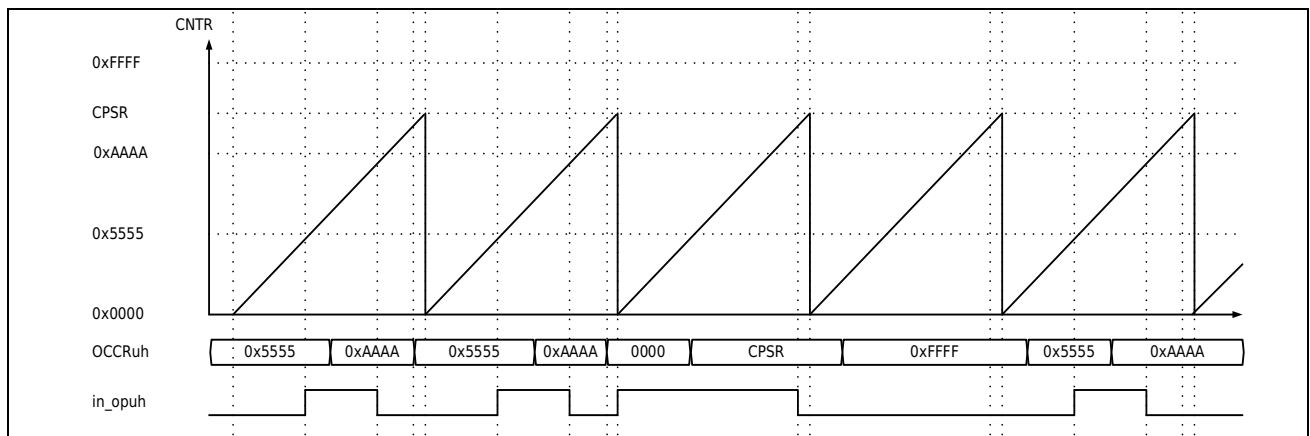


图 20-6 锯齿波模式波形输出例

2) 下图 20-7 是三角波计数模式下的比较输出模块的波形输出示例。

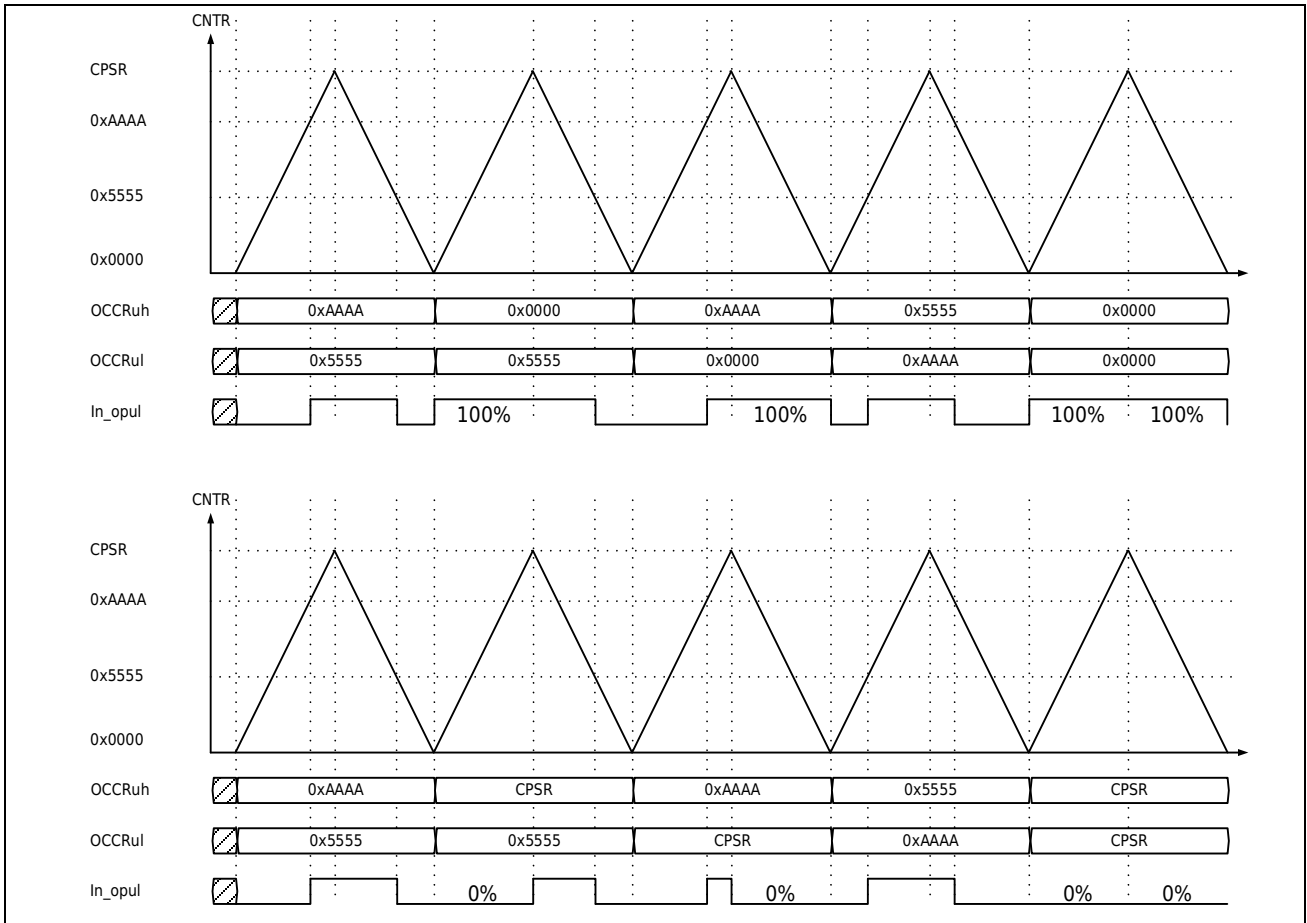


图 20-7 三角波模式波形输出示例

20.3.3 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。

20.3.3.1 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，数据从缓冲寄存器传输到 CPSR 寄存器中。

- a) 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。
- b) 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 20-8 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。

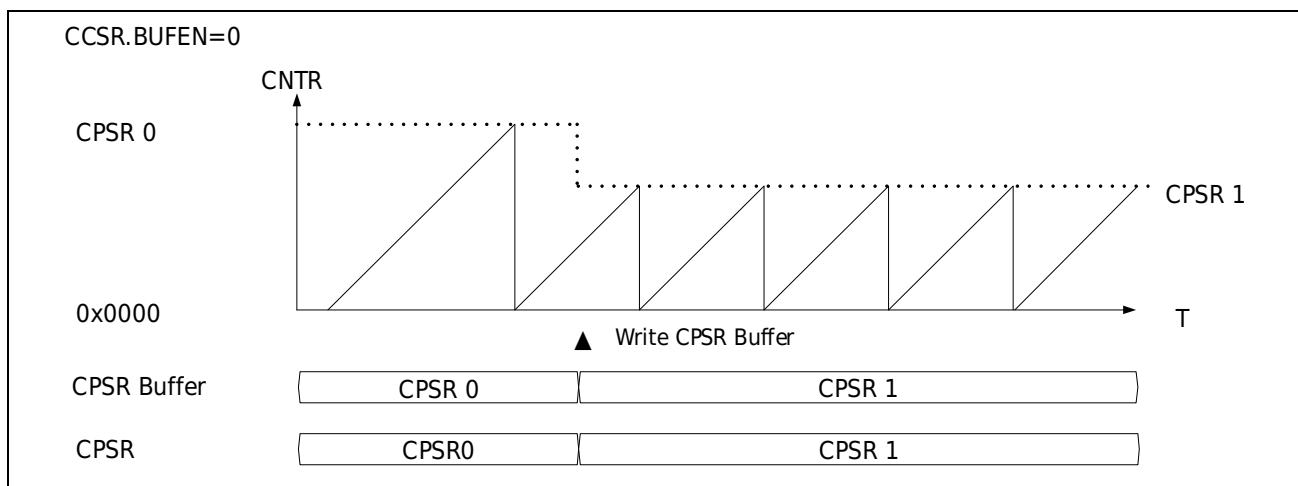


图 20-8 缓存无效时修改锯齿波计数周期

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 20-9 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。

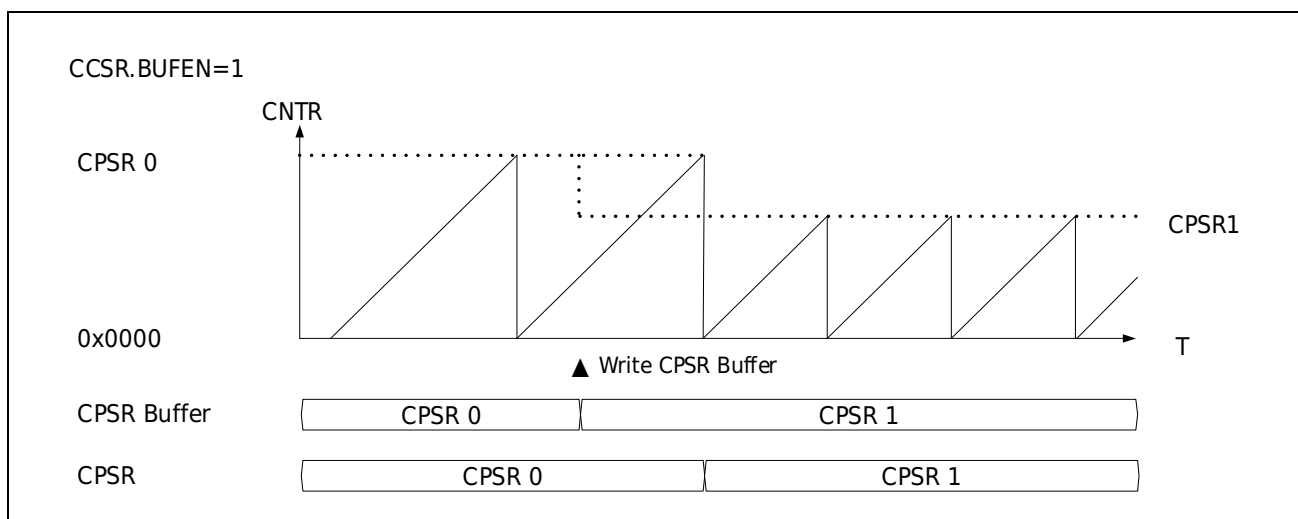


图 20-9 缓存使能时修改锯齿波计数周期

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 20-10 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。

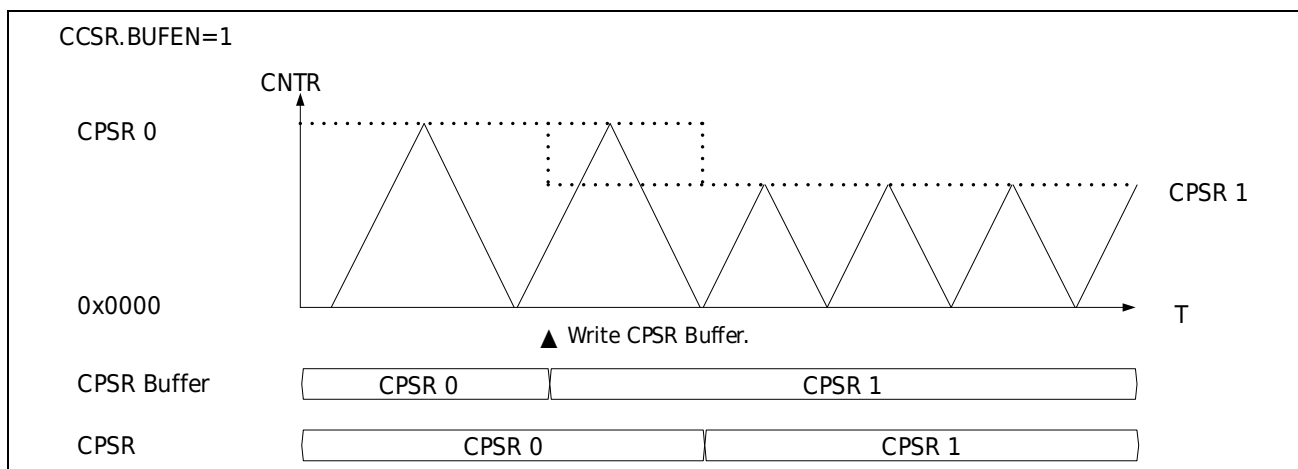


图 20-10 缓存使能时修改三角波计数周期

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。

20.3.3.2 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。

a) 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 20-11 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=0b01), 计数器周期间隔响应链接禁止 (OCER.LMCx=0) 时的波形 (x=L 或 H)。

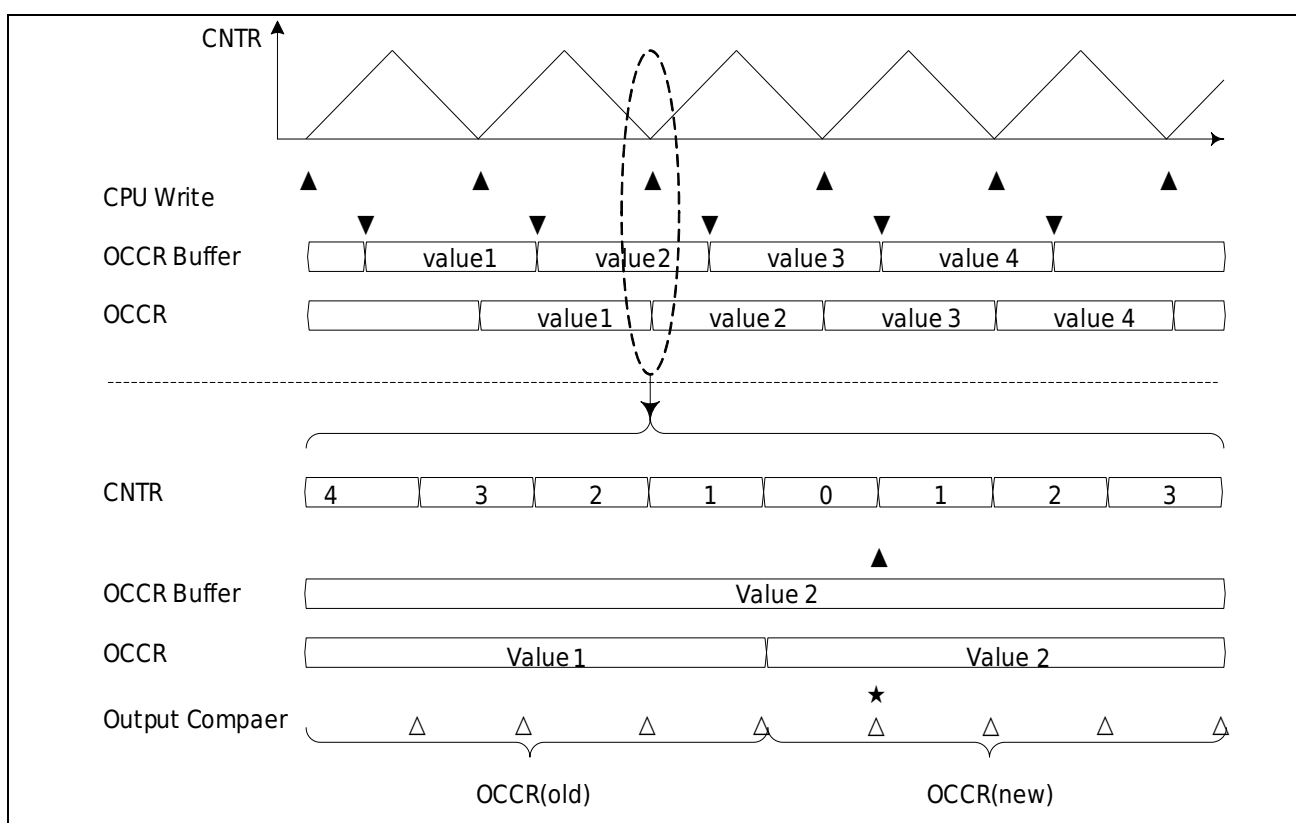


图 20-11 OCCR 缓冲数据传输 (周期间隔响应链接禁止时)

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 20-12 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=0b01),计数器周期间隔响应链接使能 (OCER.LMCx=1) 时的波形 (x=L 或 H)。

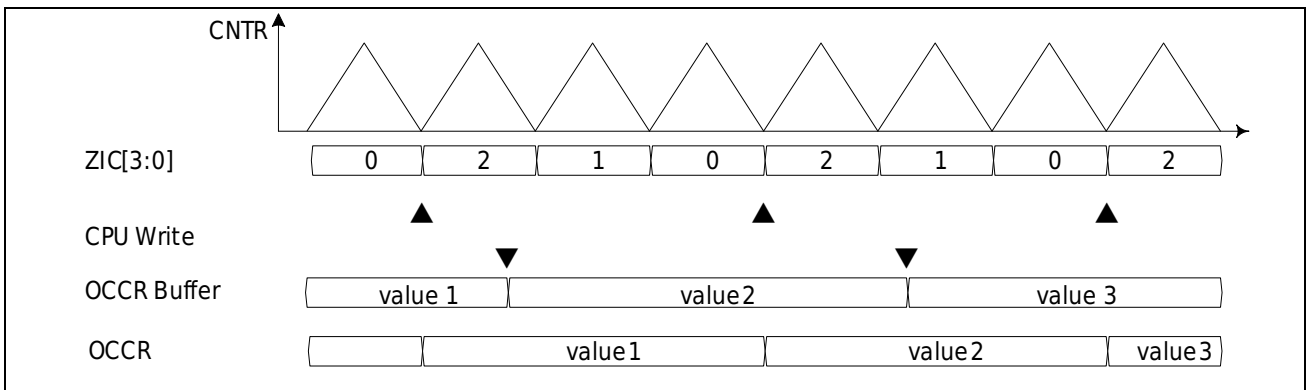


图 20-12 OCCR 缓冲数据传输 (周期间隔响应链接使能)

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 OCCRuh 和 OCCRul 缓冲功能可以产生各种不同的 PWM 输出波形。图 20-13 展示了在输出比较寄存器 OCCRuh 和 OCCRul 保持不变的情况下,改变 OCMR 寄存器的值,使 in_opul 输出不同的波形信号。

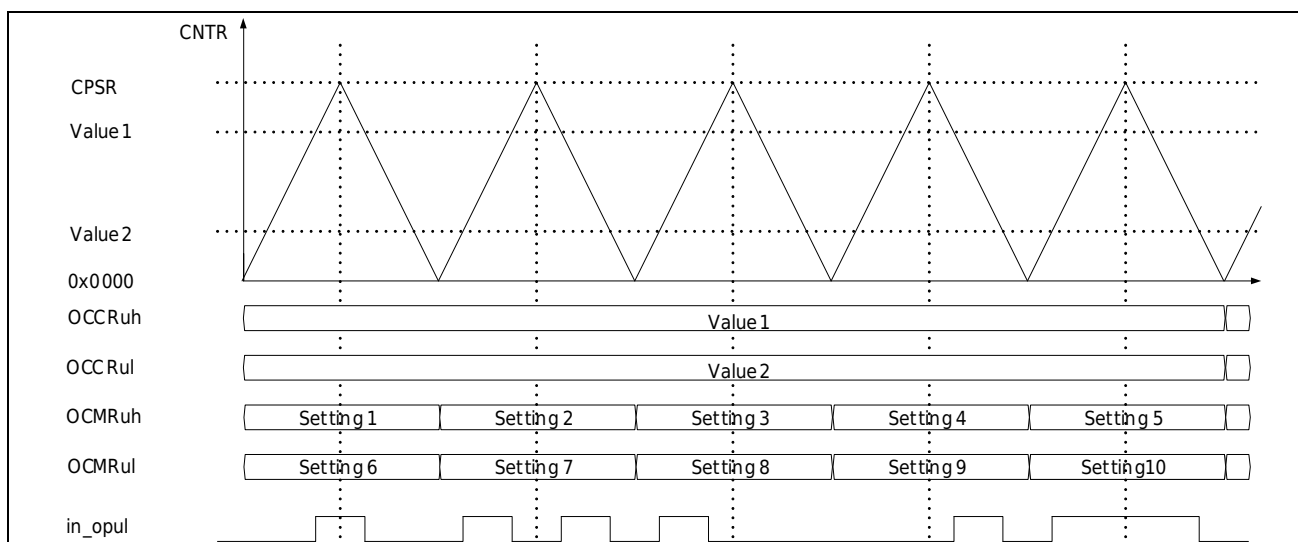


图 20-13 输出比较缓冲数据传输 (OCMR 缓冲使能)

20.3.3.3 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 20-14 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=0b01) 时传送到 SCCR 和 SCMR 寄存器。

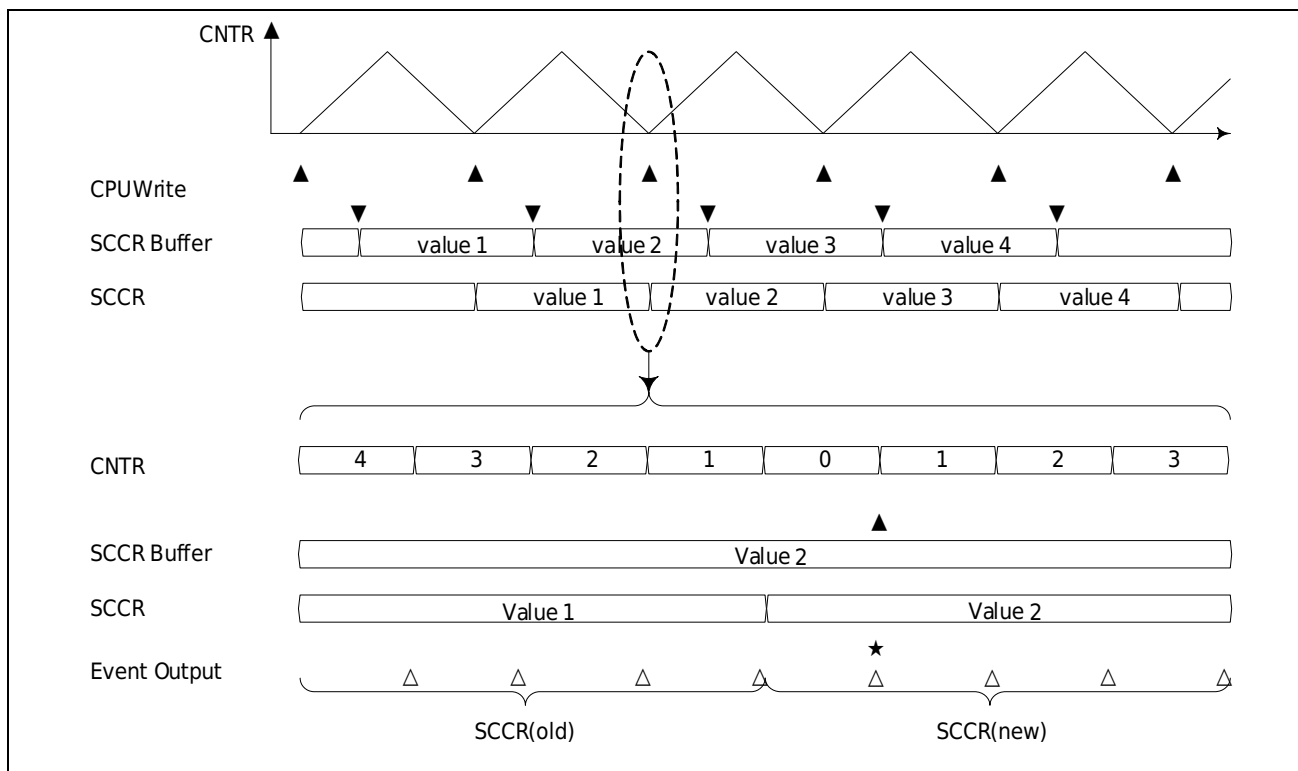


图 20-14 SCCR 缓冲传输操作 (周期间隔响应链接传输禁止时)

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 20-15 所示，是使能 SCCR 缓冲功能、计数器零值时加载 (SCSR.BUFEN=0b01)，计数器周期间隔响应链接传输使能时 (SCSR.LMC=1) 的示意图。

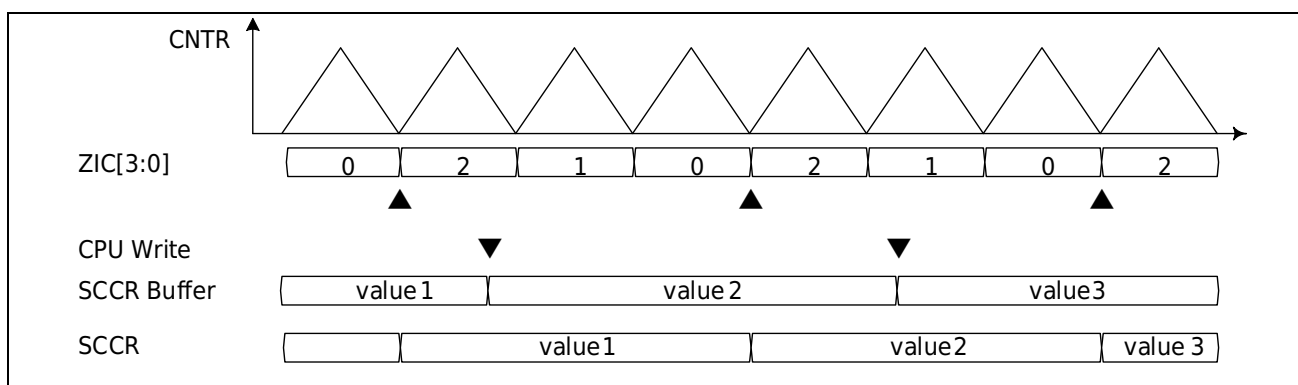


图 20-15 SCCR 缓冲传输操作（周期间隔响应链接传输使能时）

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。

20.3.4 通用 PWM 输出

本节后文提到 OCCR*h、OCCR*l、OCMR*h、OCMR*l、PSCR.OE*H、PSCR.OE*L、PDAR*、PDBR*、PSCR.OS*H、PSCR.OS*L、OCER*时，通配符“*”代表 u, v, w, x；提到内部输出信号 in_op*h、in_op*l 时，通配符“*”代表 u, v, w, x；提到端口 TMR4_O*H、TMR4_O*L 时，通配符“*”代表 U, V, W, X。

20.3.4.1 独立 PWM 输出

在直通模式 (POCR.PWMMD=00) 下，分别设定 OCCR*h、OCCR*l 的基准值和 OCMR*h、OCMR*l 的端口输出状态，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、OE*H、OE*L 位 (寄存器 PSCR) 以使能 Timer4 的输出，详细描述请参考【PWM 状态控制寄存器 (TMR4_PSCR)】。图 20-16 和图 20-17 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 (OCCR*h、OCCR*l) 的值发生比较匹配所产生的内部输出信号 (in_op*h、in_op*l) 直接输出至对应的端口 (TMR4_O*H、TMR4_O*L) 上。

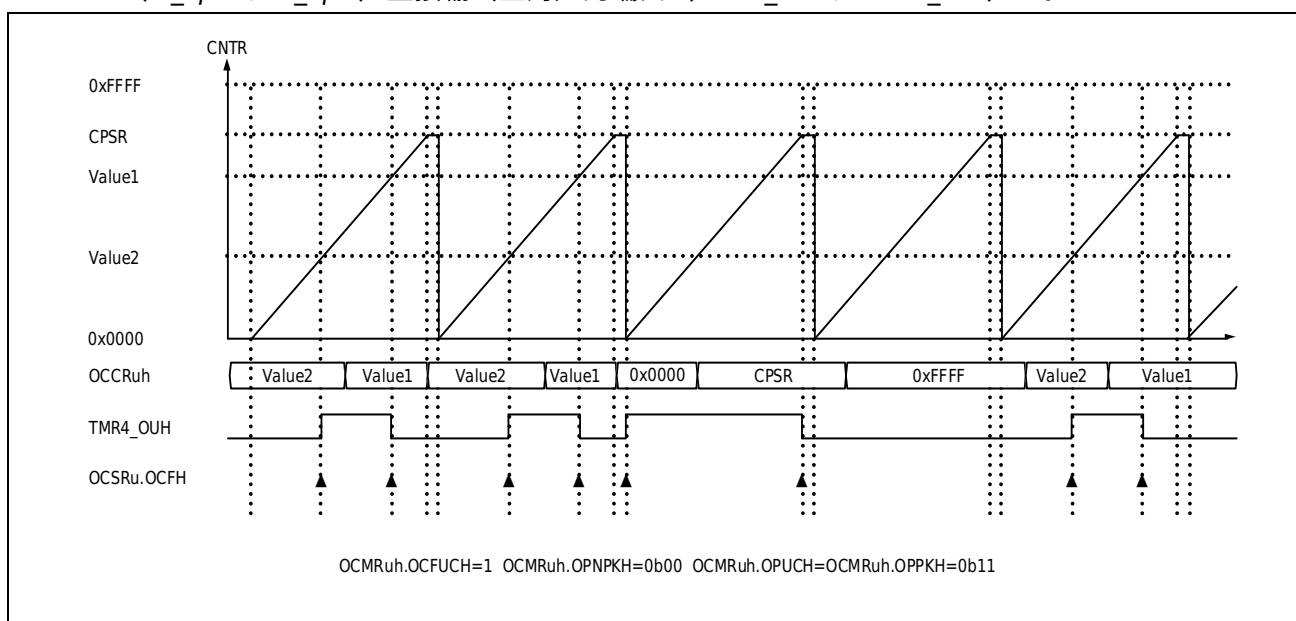


图 20-16 锯齿波独立 PWM 输出示例

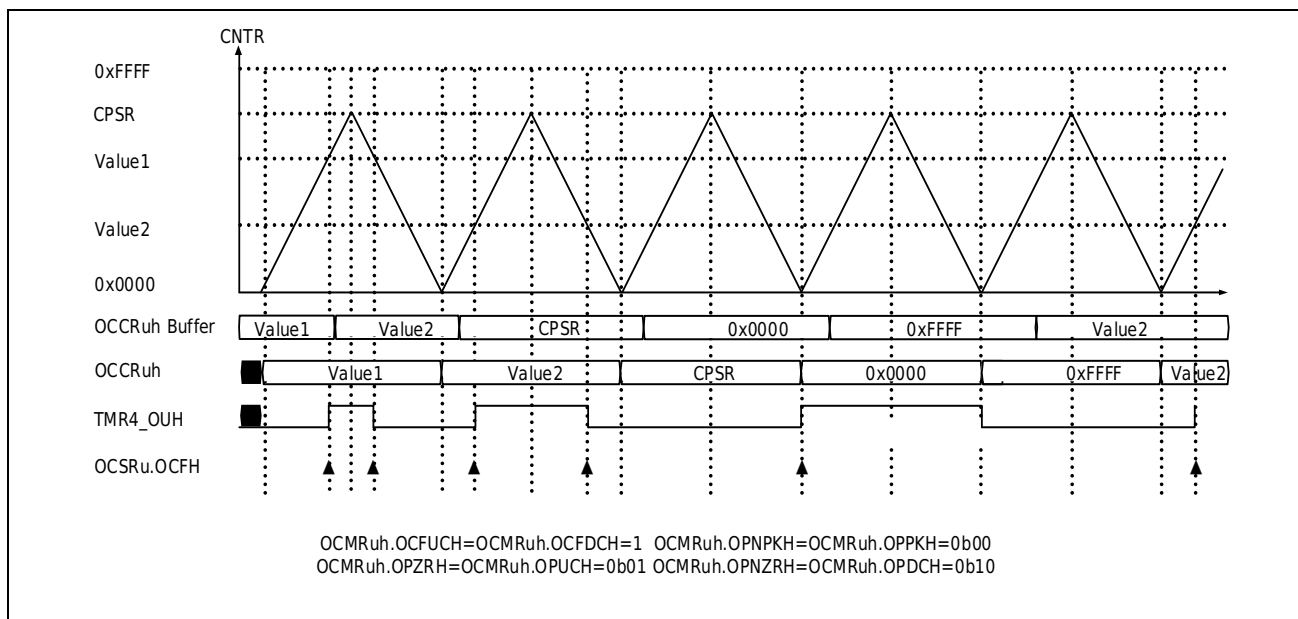


图 20-17 三角波独立 PWM 输出示例

20.3.4.2 扩展 PWM 输出

在直通模式 (POCR.PWMMD=00) 下, TMR4_O*L 的端口输出状态也可由 OCMR*I 寄存器中的扩展位 (bit32~16) 决定, 该扩展位的设定与 OCCR*h 基准值相关, 从而在 TMR4_O*L 端口上实现扩展 PWM 输出。使用扩展 PWM 输出模式前, 需要设定 MOE、OE*H、OE*L 位 (寄存器 PSCR) 以使能 Timer4 的输出, 详细描述请参考【PWM 状态控制寄存器 (TMR4_PSCR)】。图 20-18 所示是该方式下 TMR4_OUH 和 TMR4_OUL 端口的 PWM 输出。

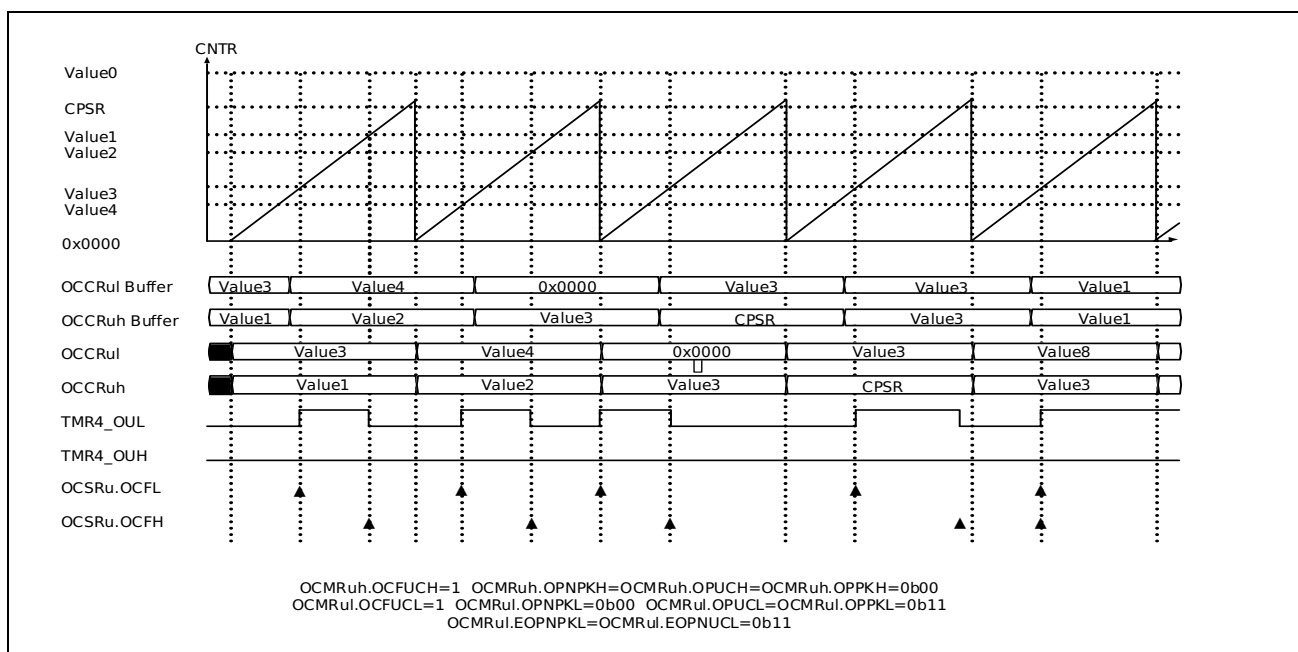


图 20-18 锯齿波扩展 PWM 输出

注意:

- 独立 PWM 输出方式下, TMR4_O*L 的端口状态由 OCMR*I 寄存器的 bit15~bit4 位决定, 仅与

OCCR**l* 基准值有关。

20.3.4.3 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 OCCR**h* 和 OCCR**l* 基准值实现一对互补 PWM 波形输出至端口，4 组端口同样方式设定即可实现 4 相互补 PWM 输出。如图 20-19 所示。

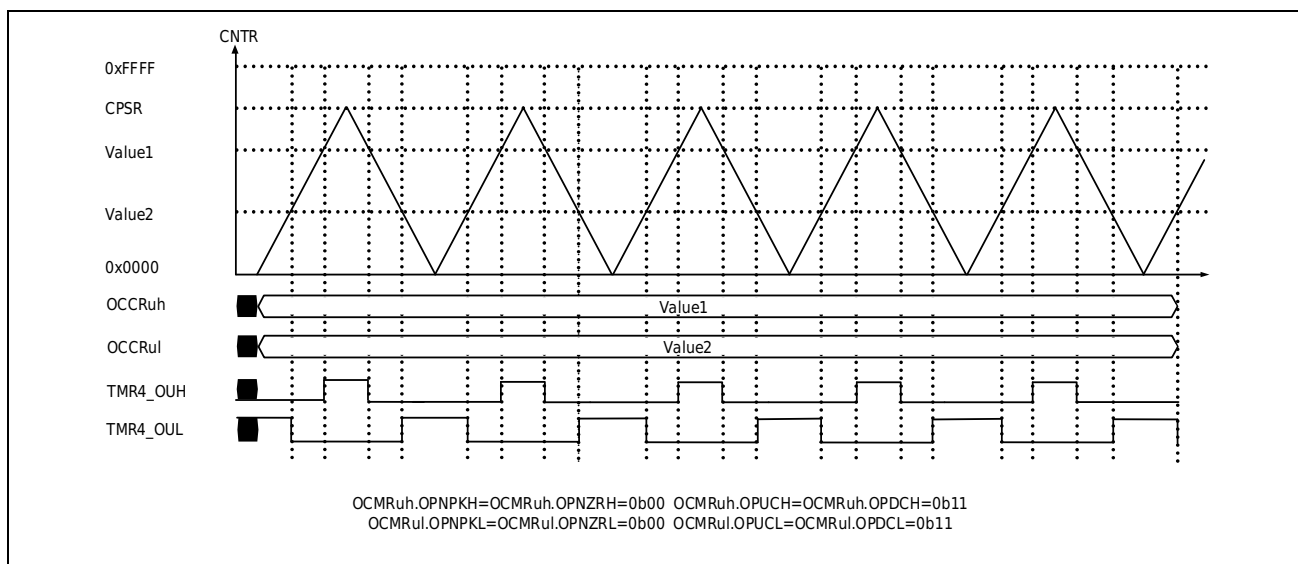


图 20-19 软件实现互补 PWM 输出

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**l*) 的值发生比较匹配产生的内部输出信号 (in_op**l*) 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，TMR4_O**H* 端口输出的极性与 in_op**l* 相同，TMR4_O**L* 端口输出的极性与 in_op**l* 相反。

如果检测到 in_op**l* 上升沿，TMR4_O**L* 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 TMR4_O**H* 输出高电平；如果检测到 in_op**l* 下降沿，TMR4_O**H* 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 TMR4_O**L* 输出高电平。

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。

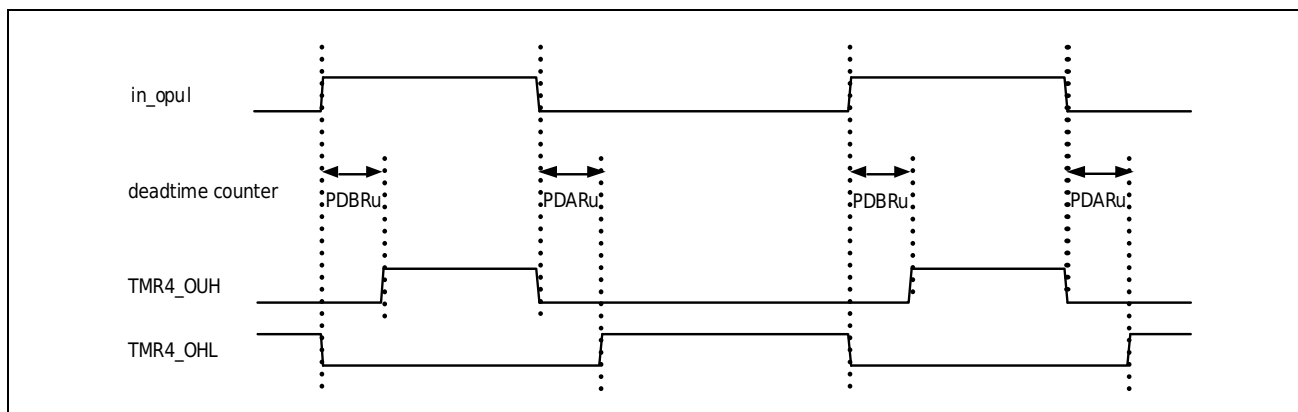


图 20-20 死区定时器模式下的互补 PWM 输出

in_op**l* 的高电平脉冲宽度小于 PDBR 设定的死区时间时，仅 TMR4_O*L 输出变为低电平。TMR4_O*L 输出电平变为高的条件是当 in_op**l* 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，TMR4_O**H* 输出将持续保持为低电平。

in_op**l* 的低电平脉冲宽度小于 PDAR 设定的死区时间时，仅 TMR4_O**H* 输出变为低电平。TMR4_O**H* 输出电平变为高的条件是当 in_op**l* 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，TMR4_O*L 输出将持续保持为低电平。如下图 20-21 所示。

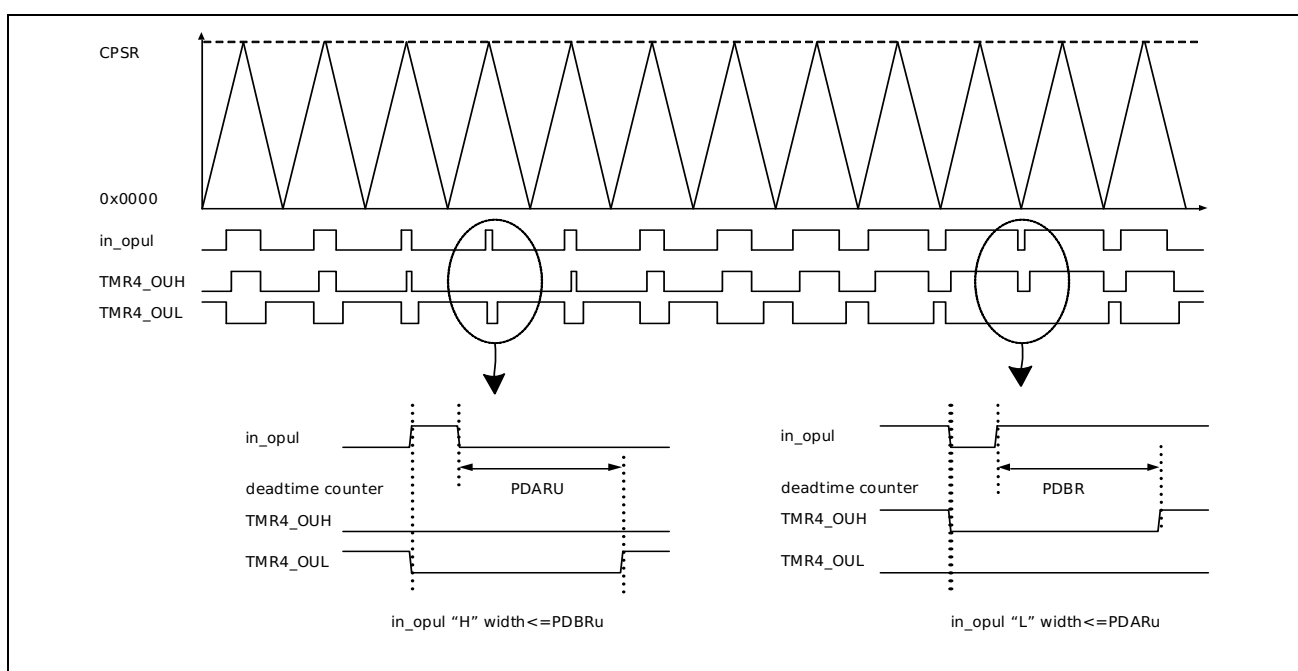


图 20-21 脉宽异常时的死区定时器模式下的波形输出

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_op**l*) 进行脉冲宽度的监测，从而实现对 in_op**l* 信号的滤波控制。这种 in_op**l* 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)。

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_op*I 的脉冲宽度大于寄存器 PFSRn 设定的时间时，滤波计数器将 in_op*I 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。

如果检测到信号 in_op*I 上升沿，滤波计数器加载 PFSR 寄存器值并启动计量 in_op*I 的高电平宽度，当 in_op*I 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，TMR4_O*L 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 TMR4_O*H 输出高电平；如果检测到信号 in_op*I 下降沿，滤波计数器加载 PFSR 寄存器值并启动计量 in_op*I 的低电平宽度，当 in_op*I 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，TMR4_O*H 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 TMR4_O*L 输出高电平。当 in_op*I 的电平脉冲宽度小于寄存器 PFSR 设定的时间时，输出 TMR4_O*H 和 TMR4_O*L 将保持不变。

图 20-22 所示，是死区计数器滤波模式下的互补 PWM 输出例。

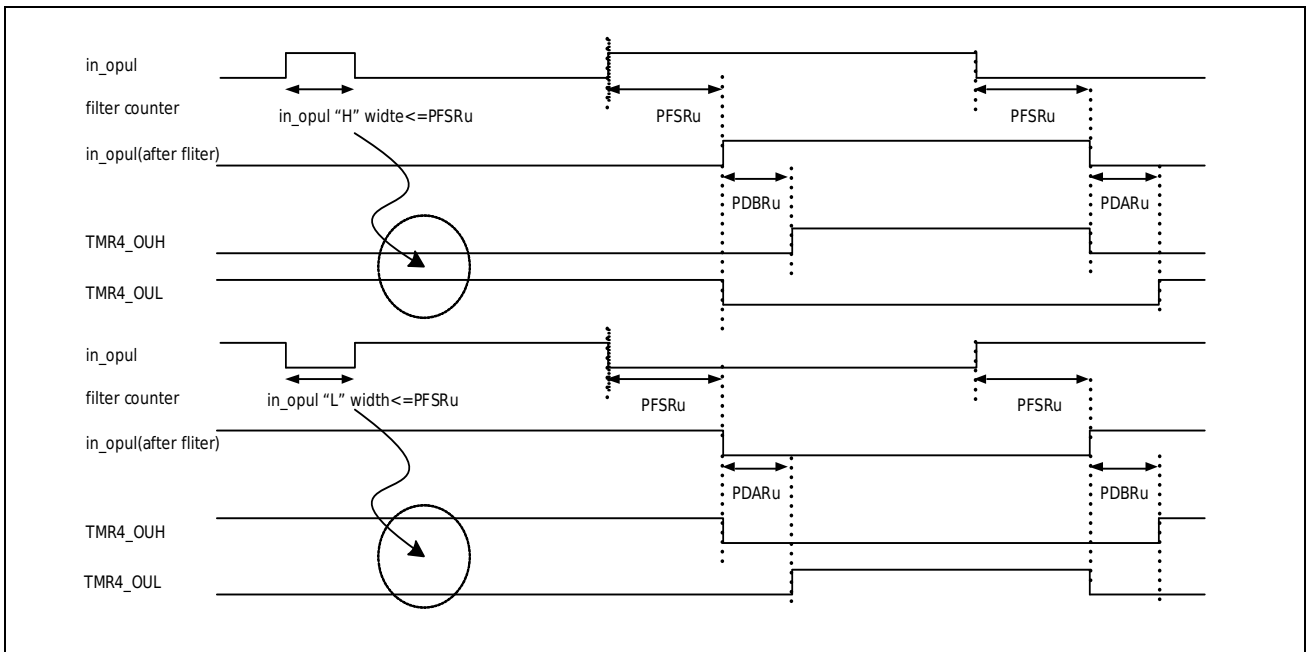


图 20-22 死区定时器滤波模式下的互补 PWM 输出

注意：

- 使用软件或硬件互补 PWM 输出模式前，需要设定 MOE、OE*H、OE*L 位（寄存器 PSCR）以使能 Timer4 的输出，详细描述请参考【PWM 状态控制寄存器 (TMR4_PSCR)】。

20.3.5 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 20-23 所示，是周期间隔响应时的 IRQZF、IRQPF 的置位时序图。

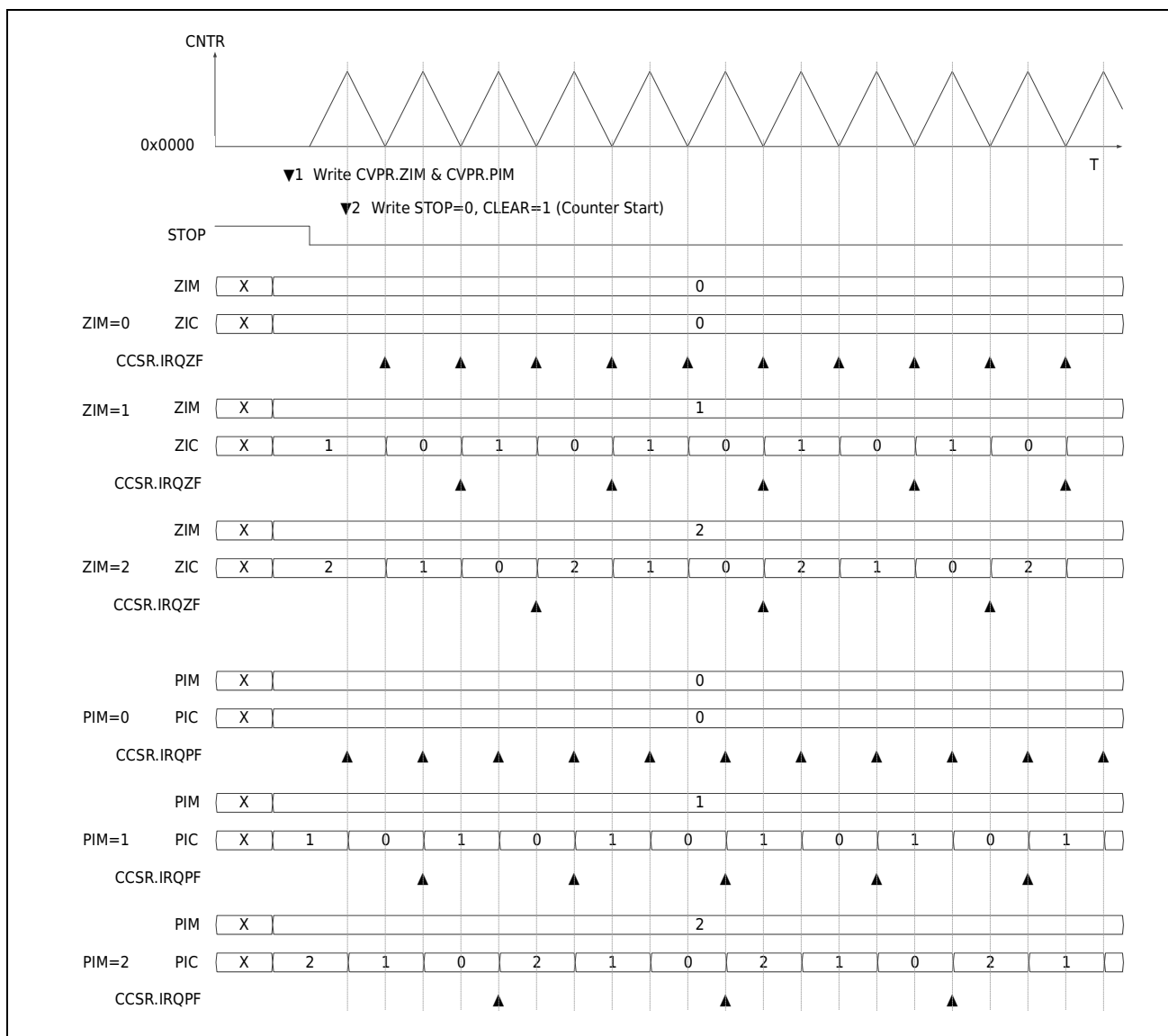


图 20-23 周期间隔响应时序图

- 计数器停止时，写入 CVPR.ZIM 和 CVPR.PIM 初始化值，初始值立即反映在内部计数器（CVPR.ZIC、CVPR.PIC）。
- 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 20-24 所示，是专用事件输出信号的周期间隔响应输出图。

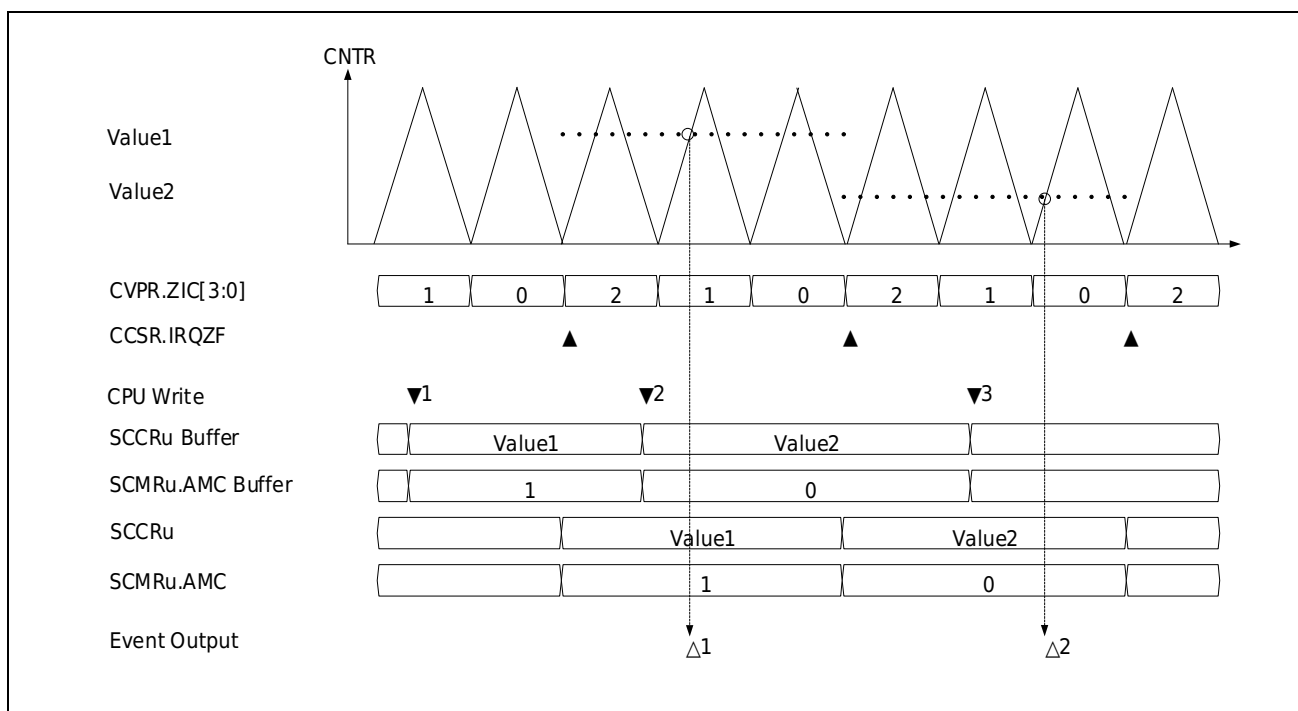


图 20-24 专用事件输出信号周期间隔响应输出

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 SCCRu 缓冲寄存器写入 value 1。同时，MZCE=1、MPCE=0、AMC=0001 写入 SCMRu 缓冲寄存器。之后，缓冲寄存器 SCCRu 和 SCMRu 传送操作在▲时刻执行。因为 MZCE=1 且 AMC=1 且计数值=SCCRu=Value 1，在时刻Δ1，专用事件输出信号置位。

在时刻▼2，向 SCCRu 缓冲寄存器写入 value 2。同时，MZCE=1、MPCE=0、AMC=0000 写入 SCMRu 缓冲寄存器。之后，缓冲寄存器 SCCRu 和 SCMRu 传送操作在▲时刻执行。因为 MZCE=1 且 AMC=0 且计数值=SCCRu=Value 2，在时刻Δ2，专用事件输出信号置位。

20.3.6 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 4 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.OS*H、PSCR.OS*L 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。

例如，若设定 PSCR.OSUH=0b01 时，在 Timer4 的 TMR4_OUH 端口正常输出期间产生 EMB 事件，则 TMR4_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。（EMB 异常事件发生后需要等待 4 个 Timer4 总线时钟周期，EMB 异常事件解除后需要等待 6 个 Timer4 总线时钟周期）。

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。

注：

当 EMB 的事件释放方式配置为状态释放时（通过 EMB_RLSSEL 寄存器配置），且 EMB 的事件源配置外部端口或者内部电压比较器比较结果有效时，必须保证外部端口和内部电压比较器比较结果的电平宽度大于 3 个计数时钟周期，否则 PWM 输出上会出现最大 2 个计数时钟周期的毛刺。当 EMB 的事件释放方式配置为软件释放时（通过 EMB_RLSSEL 寄存器配置），且 EMB 的事件源配置 PWM 输出端口发生同相有效时，在发生 PWM 同相输出到 PWM 进入安全状态之前最大会出现 2 个计数时钟周期的毛刺。

20.3.7 监测输出

每个 Timer4 单元都有一个 PWM 周期输出监测端口 TMR4_PCT 和专用事件输出监测端口 TMR4_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 TMR4_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 TMR4_PCT 上就发生一次翻转。图 18-25 是计数方向信号的示例。

TMR4_PCT 端口的周期输出监测功能只在计数时钟不分频 (CCSR.CKDIV=0x0) 的情况下使用。

专用事件输出监测端口 TMR4_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 TMR4_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 TMR4_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 20-26 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 TMR4_ADSTM 置位，计数器下溢后 TMR4_ADSTM 清零。

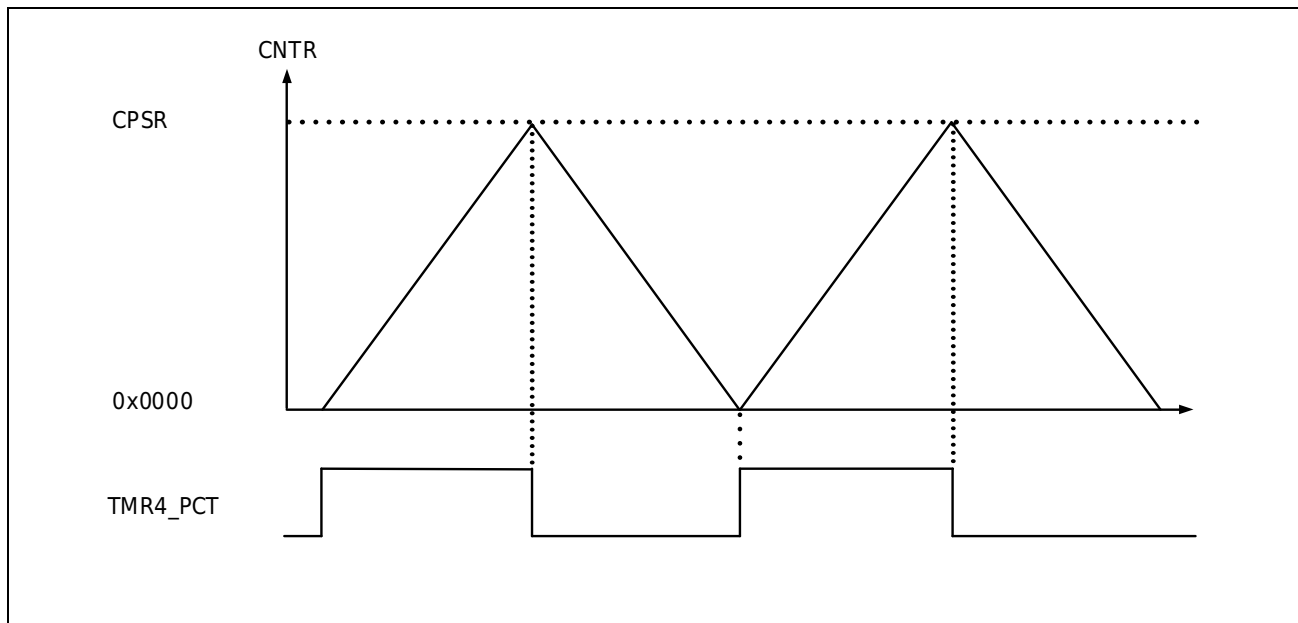


图 20-25 PWM 周期输出监测示例

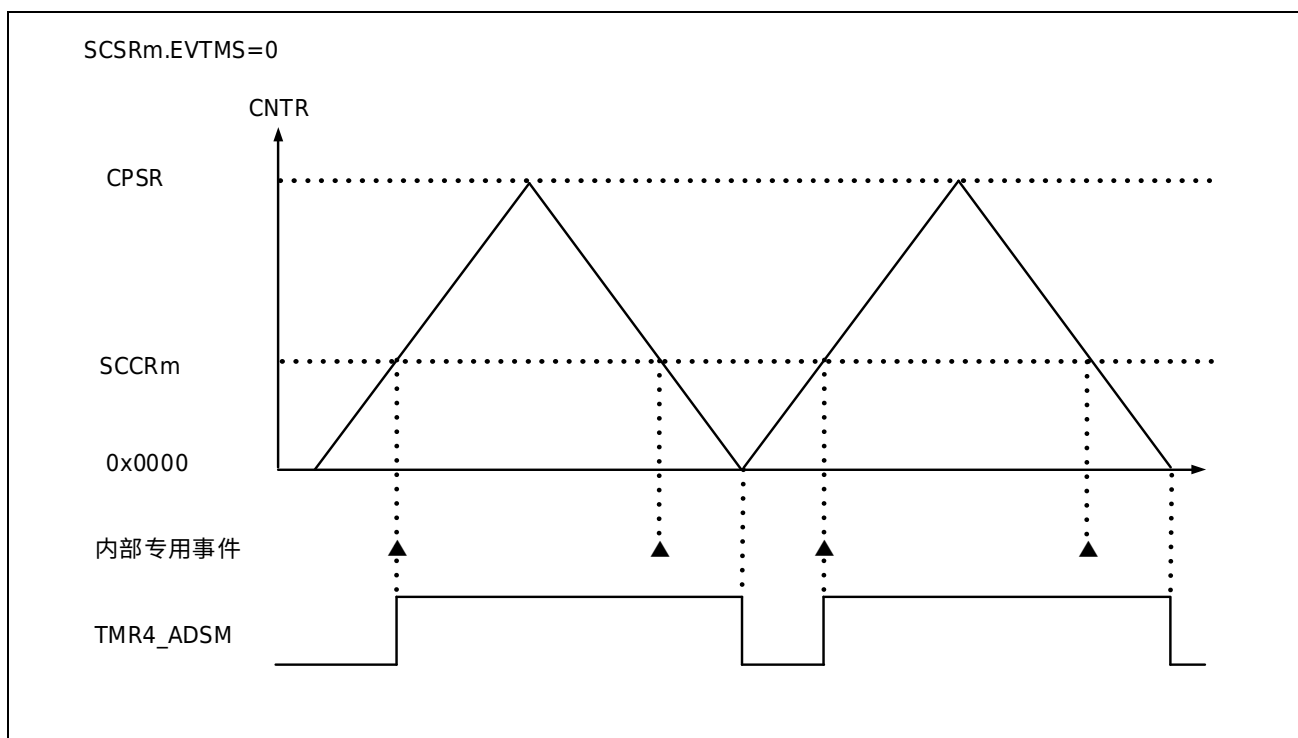


图 20-26 比较启动模式下专用事件输出 TMR4_ADSTM 时序

20.3.8 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 (TMR4_TRGSEL) 中对应的编号设定来选择，具体寄存器内容详见【Timer4 硬件触发事件选择寄存器 (TMR4_TRGSEL)】章节，具体的事件对应关系请参考【中断控制器 (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。

20.3.9 同步启动

产品配置了一个计数同步使能寄存器 (TMR_SYNENR)，用于同步启动产品所搭载的所有定时器 (包括 Timer0、Timer4、Timer6、TimerA)。使用方法和寄存器说明参考【通用定时器 (Timer0)】章节。

20.4 中断和事件

20.4.1 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 8 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_GCMmn, m=U、V、W、X; 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【中断事件表】章节。

20.4.2 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_GOVF 和 TMR4_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【中断事件表】章节。

20.4.3 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、RCSR.RTIFX 位分别会被置为 1。此时若设定 RCSR.RTIDU、RCSR.RTIDV、RCSR.RTIDW、RCSR.RTIFX 中断屏蔽无效，则对应的重载计数匹配中断请求 (TMR4_RLOm, m=U、V、W、X) 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【中断事件表】章节。

20.4.4 专用比较匹配中断及事件

Timer4 的 8 个专用比较基准寄存器 (SCCRm) 对应产生 8 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, 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.ITEN0~7 使能中断, 则对应的专用比较匹配中断请求 (TMR4_SCM0~7) 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 OCCR*h 或 OCCR*i (由 SCSR.EVTDS 位选择) 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 20-27 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。

专用比较匹配事件可以用于触发别的模块, 事件编号请参考【中断事件表】章节。

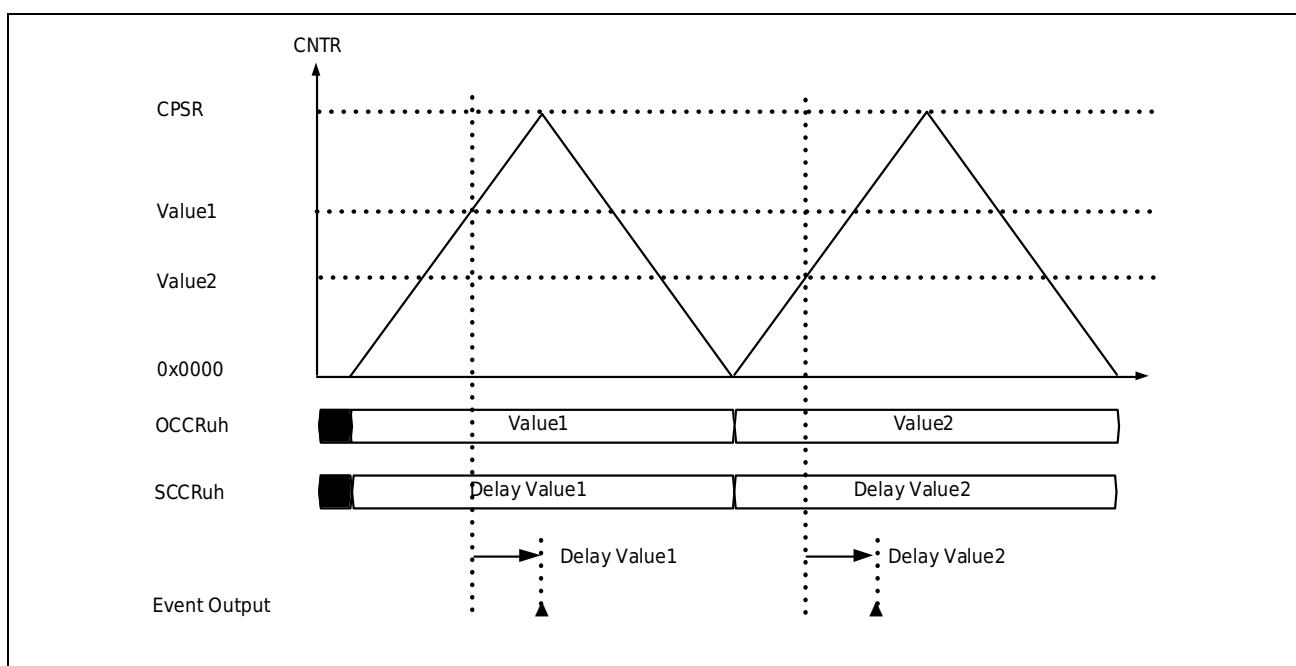


图 20-27 延时启动模式下专用事件输出信号的输出时序

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。

20.5 寄存器描述

表 20-3 Timer4 基准地址

名称	基地址	描述
Timer4	0x4003 8000	Timer4_1基地址

表 20-4 Timer4 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
TMR4_CNTR	计数值寄存器	0x0054	16	0x0000
TMR4_CPSR	周期基准寄存器	0x0050	16	0xFFFF
TMR4_CCSR	控制状态寄存器	0x0058	16	0x0040
TMR4_CVPR	有效周期寄存器	0x005A	16	0x0000
TMR4_PSCR	PWM状态控制寄存器	0x005C	32	0x5555 0000
TMR4_OCCRuh	通用比较基准寄存器UH	0x0000	16	0x0000
TMR4_OCCRul	通用比较基准寄存器UL	0x0004	16	0x0000
TMR4_OCCRvh	通用比较基准寄存器VH	0x0008	16	0x0000
TMR4_OCCRvl	通用比较基准寄存器VL	0x000C	16	0x0000
TMR4_OCCRwh	通用比较基准寄存器WH	0x0010	16	0x0000
TMR4_OCCRwl	通用比较基准寄存器WL	0x0014	16	0x0000
TMR4_OCCRxh	通用比较基准寄存器XH	0x0018	16	0x0000
TMR4_OCCRxl	通用比较基准寄存器XL	0x001C	16	0x0000
TMR4_OCSRu	通用控制状态寄存器U	0x0020	16	0xFF00
TMR4_OCSRv	通用控制状态寄存器V	0x0024	16	0xFF00
TMR4_OCSRw	通用控制状态寄存器W	0x0028	16	0xFF00
TMR4_OCSRx	通用控制状态寄存器X	0x002C	16	0xFF00
TMR4_OCERu	通用扩展控制寄存器U	0x0022	16	0x0000
TMR4_OCERv	通用扩展控制寄存器V	0x0026	16	0x0000
TMR4_OCERw	通用扩展控制寄存器W	0x002A	16	0x0000
TMR4_OCERx	通用扩展控制寄存器X	0x002E	16	0x0000
TMR4_OCMRuh	通用模式控制寄存器UH	0x0030	16	0x0000
TMR4_OCMRul	通用模式控制寄存器UL	0x0034	32	0x0000 0000
TMR4_OCMRvh	通用模式控制寄存器VH	0x0038	16	0x0000
TMR4_OCMRvl	通用模式控制寄存器VL	0x003C	32	0x0000 0000
TMR4_OCMRwh	通用模式控制寄存器WH	0x0040	16	0x0000
TMR4_OCMRwl	通用模式控制寄存器WL	0x0044	32	0x0000 0000
TMR4_OCMRxh	通用模式控制寄存器XH	0x0048	16	0x0000
TMR4_OCMRxl	通用模式控制寄存器XL	0x004C	32	0x0000 0000
TMR4_SCCRuh	专用比较基准寄存器UH	0x00B0	16	0x0000

寄存器	描述	偏移地址	位宽	复位值
TMR4_SCCRul	专用比较基准寄存器UL	0x00B4	16	0x0000
TMR4_SCCRvh	专用比较基准寄存器VH	0x00B8	16	0x0000
TMR4_SCCRvl	专用比较基准寄存器VL	0x00BC	16	0x0000
TMR4_SCCRwh	专用比较基准寄存器WH	0x00C0	16	0x0000
TMR4_SCCRwl	专用比较基准寄存器WL	0x00C4	16	0x0000
TMR4_SCCRxh	专用比较基准寄存器XH	0x00C8	16	0x0000
TMR4_SCCRxl	专用比较基准寄存器XL	0x00CC	16	0x0000
TMR4_SCSRuh	专用控制状态寄存器UH	0x00D0	16	0x0000
TMR4_SCSRul	专用控制状态寄存器UL	0x00D4	16	0x0000
TMR4_SCSRvh	专用控制状态寄存器VH	0x00D8	16	0x0000
TMR4_SCSRvl	专用控制状态寄存器VL	0x00DC	16	0x0000
TMR4_SCSRwh	专用控制状态寄存器WH	0x00E0	16	0x0000
TMR4_SCSRwl	专用控制状态寄存器WL	0x00E4	16	0x0000
TMR4_SCSRxh	专用控制状态寄存器XH	0x00E8	16	0x0000
TMR4_SCSRxl	专用控制状态寄存器XL	0x00EC	16	0x0000
TMR4_SCMRuh	专用模式控制寄存器UH	0x00D2	16	0xFF00
TMR4_SCMRul	专用模式控制寄存器UL	0x00D6	16	0xFF00
TMR4_SCMRvh	专用模式控制寄存器VH	0x00DA	16	0xFF00
TMR4_SCMRvl	专用模式控制寄存器VL	0x00DE	16	0xFF00
TMR4_SCMRwh	专用模式控制寄存器WH	0x00E2	16	0xFF00
TMR4_SCMRwl	专用模式控制寄存器WL	0x00E6	16	0xFF00
TMR4_SCMRxh	专用模式控制寄存器XH	0x00EA	16	0xFF00
TMR4_SCMRxl	专用模式控制寄存器XL	0x00EE	16	0xFF00
TMR4_SCER	专用扩展控制寄存器	0x00F0	16	0x0000
TMR4_POCRu	PWM基本控制寄存器U	0x00A0	16	0xFF00
TMR4_POCRv	PWM基本控制寄存器V	0x00A4	16	0xFF00
TMR4_POCRw	PWM基本控制寄存器W	0x00A8	16	0xFF00
TMR4_POCRx	PWM基本控制寄存器X	0x00AC	16	0xFF00
TMR4_PFSRu	PWM滤波控制寄存器U	0x0082	16	0x0000
TMR4_PFSRv	PWM滤波控制寄存器V	0x008A	16	0x0000
TMR4_PFSRw	PWM滤波控制寄存器W	0x0092	16	0x0000
TMR4_PFSRx	PWM滤波控制寄存器X	0x009A	16	0x0000
TMR4_PDARu	PWM死区控制寄存器AU	0x0084	16	0x0000
TMR4_PDBRu	PWM死区控制寄存器BU	0x0086	16	0x0000
TMR4_PDARv	PWM死区控制寄存器AV	0x008C	16	0x0000
TMR4_PDBRv	PWM死区控制寄存器BV	0x008E	16	0x0000
TMR4_PDARw	PWM死区控制寄存器AW	0x0094	16	0x0000
TMR4_PDBRw	PWM死区控制寄存器BW	0x0096	16	0x0000

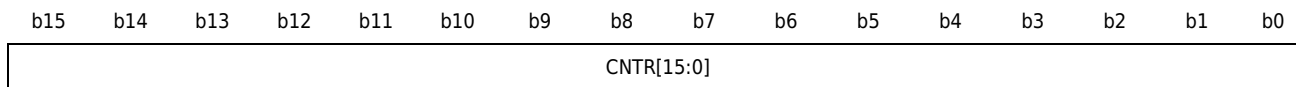
寄存器	描述	偏移地址	位宽	复位值
TMR4_PDARx	PWM死区控制寄存器AX	0x009C	16	0x0000
TMR4_PDBRx	PWM死区控制寄存器BX	0x009E	16	0x0000
TMR4_RCSR	重载控制状态寄存器	0x00F4	32	0x0000 0000
TMR4_SCIR	专用中断控制寄存器	0x00F8	16	0xFF00
TMR4_SCFR	专用标志控制寄存器	0x00FC	16	0xFF00

注意：

- 下述寄存器详细描述中， $m=uh、ul、vh、vl、wh、wl、xh、xl$ ， $n=u、v、w、x$ 。 m 所指寄存器分别在功能实现时对应端口 $TMR4_OUH$ 、 $TMR4_OUL$ 、 $TMR4_OVH$ 、 $TMR4_OVL$ 、 $TMR4_OWH$ 、 $TMR4_OWL$ 、 $TMR4_OXH$ 、 $TMR4_OXL$ 的输出控制等； n 所指寄存器分别在功能实现时对应端口 $TMR4_OUx$ 、 $TMR4_OVx$ 、 $TMR4_OWx$ 、 $TMR4_OXx$ 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。

20.5.1 计数值寄存器 (TMR4_CNTR)

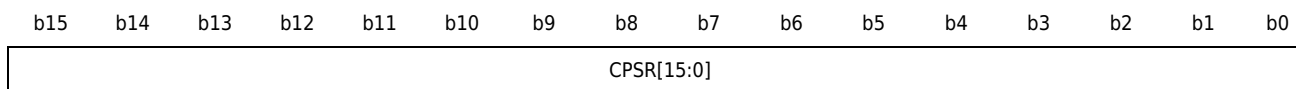
复位值: 0x0000



位/位域	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	RW

20.5.2 周期基准寄存器 (TMR4_CPSR)

复位值: 0xFFFF



位/位域	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	RW

20.5.3 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
ECK EN	Res	HST	Res	IRQ ZF	IRQ ZEN	IRQ PF	IRQ PEN	BUF EN	STOP	MODE	CLEAR	CKDIV[3:0]			
位/位域	标记	位名	功能										读写		
0: 内部PCLK0时钟 1: 外部TMR4_CLK端口输入时钟															
注:															
b15	ECKEN	时钟源选择	1. 该位在计数器停止时设定 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										RW		
b14	Res	保留位	读出时为“0”, 写入时写“0”										RW		
条件: TMR4_TRGSEL寄存器中指定的事件发生															
b13	HST	硬件启动使能	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												
b14	Res	保留位	读出时为“0”, 写入时写“0”										RW		
0: 未发生计数下溢 1: 发生计数下溢															
b11	IRQZF	下溢状态	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 2. 当计数器被总线复位或者写入CLEAR=1, IRQZF位将不会被置位										RW		
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										RW		
0: 未发生计数上溢 1: 发生计数上溢															
b9	IRQPF	上溢状态	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。 2. 当计数器被总线复位或者写入CLEAR=1, IRQZF位将不会被置位										RW		
b8	IRQPEN	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										RW		
b7	BUFEN	缓存使能	0: 禁止CPSR缓存功能 1: 使能CPSR缓存功能										RW		
b6	STOP	计数器使能	0: 计数器启动 1: 计数器停止										RW		
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										RW		
0: 无操作 1: 计数器清零															
b4	CLEAR	计数器清零	注: 该位读出时始终为0										RW		
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										RW		

0000: 计数时钟为PCLK0
0001: 计数时钟为PCLK0/2
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: 禁止设定
注:
时钟源为内部PCLK0时, 分频有效

20.5.4 有效周期寄存器 (TMR4_CVPR)

复位值: 0x0000

位/位域	标记	位名	功能	读写
b15~b12	PIC[3:0]	上溢中断屏蔽状态	<p>PIC是一个固定向下计数的计数器, 当发生上溢中断时PIC的值减1, 读取PIC可以获取当前剩余需要被屏蔽的上溢中断个数。</p> <p>写入条件: 计数器停止状态 (CCSR.STOP=1) 满足以下之一, PIM的值载入PIC寄存器:</p> <ol style="list-style-type: none"> 1. 计数器清零动作发生 (CCSR.CLEAR写1) 2. PIC=0且CNTR计数器发生上溢中断 	R
b11~b8	ZIC[3:0]	下溢中断屏蔽状态	<p>ZIC是一个固定向下计数的计数器, 当发生下溢中断时ZIC的值减1, 读取ZIC可以获取当前剩余需要被屏蔽的下溢中断个数。</p> <p>写入条件: 计数器停止状态 (CCSR.STOP=1) 满足以下之一, ZIM的值载入ZIC寄存器:</p> <ol style="list-style-type: none"> 1. 计数器清零动作发生 (CCSR.CLEAR写1) 2. ZIC=0且CNTR计数器发生下溢中断 	R
b7~b4	PIM[3:0]	上溢中断屏蔽设定	<p>设置需要被屏蔽的上溢中断个数</p> <p>满足以下之一, PIM的值载入PIC寄存器:</p> <ol style="list-style-type: none"> 1. 计数器清零动作发生 (CCSR.CLEAR写1) 2. PIC=0且CNTR计数器发生上溢中断 	RW
b3~b0	ZIM[3:0]	下溢中断屏蔽设定	<p>设置需要被屏蔽的下溢中断个数</p> <p>满足以下之一, ZIM的值载入ZIC寄存器:</p> <ol style="list-style-type: none"> 1. 计数器清零动作发生 (CCSR.CLEAR写1) 2. ZIC=0且CNTR计数器发生下溢中断 	RW

20.5.5 通用比较基准寄存器 (TMR4_OCCRM)

复位值: 0x0000

位/位域	标记	位名	功能	读写
b15~b0	OCCR[15:0]	通用比较基准值	<p>通用比较基准值</p> <p>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</p>	RW

20.5.6 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
Res								OCFL	OCFH	OCIEL	OCIEH	OCPL	OCPH	OCEL	OCEH
位/位域	标记	位名	功能										读写		
b15~b8	Res	保留位	读出时所有位为“1”，写入时所有位写“1”										RW		
b7	OCFL	计数匹配标志OCFL	0: 计数器计数值与OCCR*I设定值不相等 1: 计数器计数值与OCCR*I设定值相等 注: 该位必须在OCEL=1时有效										RW		
b6	OCFH	计数匹配标志OCFH	0: 计数器计数值与OCCR*H设定值不相等 1: 计数器计数值与OCCR*H设定值相等 注: 该位必须在OCEH=1时有效										RW		
b5	OCIEL	计数匹配中断使能L	0: OCFL置位时, 不发生中断 1: OCFL置位时, 发生中断										RW		
b4	OCIEH	计数匹配中断使能H	0: OCFH置位时, 不发生中断 1: OCFH置位时, 发生中断										RW		
b3	OCPL	比较输出无效时端口状态L	0: OCEL=0时, in_op*I输出低电平 1: OCEL=0时, in_op*I输出高电平										RW		
b2	OCPH	比较输出无效时端口状态H	0: OCEH=0时, in_op*H输出低电平 1: OCEH=0时, in_op*H输出高电平										RW		
b1	OCEL	端口输出选择L	0: 比较输出无效, in_op*I状态由OCPL决定 1: 比较输出有效, in_op*I状态由OCMR*I的设定和OCFL状态决定										RW		
b0	OCEH	端口输出选择H	0: 比较输出无效, in_op*H状态由OCPH决定 1: 比较输出有效, in_op*H状态由OCMR*H的设定和OCFH状态决定										RW		

20.5.7 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
Res		MCECL	MCECH	LMML	LMMH	LMCL	LMCH	MLBUF EN[1:0]		MHBUF EN[1:0]		CLBUF EN[1:0]		CHBUF EN[1:0]	
位/位域	标记	位名	功能			读写									
b15~b14	Res	保留位	读出时为“0”，写入时写“0”			RW									
b13	MCECL	计数匹配增加条件使能L	增加条件: 计数上溢 (CNTR=CPSR) 且OCCR*I > CPSR, 作为计数匹配条件 0: 禁止该条件 1: 使能该条件, 条件满足时, OCCR*I计数匹配 注: 1. 只能在输出比较功能无效 (OCSR.OCEL=0) 时修改该位			RW									
b12	MCECH	计数匹配增加条件使能H	增加条件: 计数上溢 (CNTR=CPSR) 且OCCR*h > CPSR, 作为计数匹配的条件 0: 禁止该条件 1: 使能该条件, 条件满足时, OCCR*h计数匹配 注: 1. 只能在输出比较功能无效 (OCSR.OCEH=0) 时修改该位			RW									
b11	LMML	OCCR*I周期间隔响应功能链接L	0: 周期间隔响应功能链接无效, OCMR*I的缓存传送由MLBUFEN设定决定 1: 周期间隔响应功能链接有效, OCMR*I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)			RW									
b10	LMMH	OCCR*h周期间隔响应功能链接H	0: 周期间隔响应功能链接无效, OCMR*h的缓存传送由MHBUFEN设定决定 1: 周期间隔响应功能链接有效, OCMR*h的缓存传送在MHBUFEN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)			RW									
b9	LMCL	OCCR*I周期间隔响应功能链接L	0: 周期间隔响应功能链接无效, OCCR*I的缓存传送由CLBUFEN设定决定 1: 周期间隔响应功能链接有效, OCCR*I的缓存传送在CLBUFEN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)			RW									
b8	LMCH	OCCR*h周期间隔响应功能链接H	0: 周期间隔响应功能链接无效, OCCR*h的缓存传送由CHBUFEN设定决定 1: 周期间隔响应功能链接有效, OCCR*h的缓存传送在CHBUFEN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)			RW									
b7~b6	MLBUFEN[1:0]	OCMR*I缓存传送	00: OCMR*I缓存寄存器的值直接写入OCMR*I 01: OCMR*I缓存寄存器的值在计数下溢时写入OCMR*I 10: OCMR*I缓存寄存器的值在计数上溢时写入OCMR*I 11: OCMR*I缓存寄存器的值在计数下溢或上溢时写入OCMR*I			RW									
b5~b4	MHBUFEN[1:0]	OCMR*h缓存传送	00: OCMR*h缓存寄存器的值直接写入OCMR*h 01: OCMR*h缓存寄存器的值在计数下溢时写入OCMR*h 10: OCMR*h缓存寄存器的值在计数上溢时写入OCMR*h			RW									

			11: OCMR*h缓存寄存器的值在计数下溢或上溢时写入OCMR*h	
b3~b2	CLBUFEN[1:0]	OCCRxI缓存传送	00: OCCR*I缓存寄存器的值直接写入OCCR*I 01: OCCR*I缓存寄存器的值在计数下溢时写入OCCR*I 10: OCCR*I缓存寄存器的值在计数上溢时写入OCCR*I 11: OCCR*I缓存寄存器的值在计数下溢或上溢时写入OCCR*I	RW
b1~b0	CHBUFEN[1:0]	OCCRxh缓存传送	00: OCCR*h缓存寄存器的值直接写入OCCR*h 01: OCCR*h缓存寄存器的值在计数下溢时写入OCCR*h 10: OCCR*h缓存寄存器的值在计数上溢时写入OCCR*h 11: OCCR*h缓存寄存器的值在计数下溢或上溢时写入OCCR*h	RW

20.5.8 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]		OCF ZRH	OCF UCH	OCF PKH	OCF DCH

表 20-5 OCCR*h 计数匹配一览表

OCCR*h 计数匹配	条件编号
0	1
1	2

注: 0: 不匹配, 1: 匹配

注: 该寄存器位描述用于 OCMRuh、OCMRvh、OCMRwh、OCMRxh 时

位/位域	标记	位名	功能	读写
b15~b14	OPNZRH[1:0]	下溢条件1端口状态H	下溢条件1 (同时满足): 计数下溢, OCCR*h计数不匹配 00: 满足条件时, in_op*h保持不变 01: 满足条件时, in_op*h输出高电平 10: 满足条件时, in_op*h输出低电平 11: 满足条件时, in_op*h输出反转	RW
b13~b12	OPNPKH[1:0]	上溢条件1端口状态H	上溢条件1 (同时满足): 计数上溢, OCCR*h计数不匹配 00: 满足条件时, in_op*h保持不变 01: 满足条件时, in_op*h输出高电平 10: 满足条件时, in_op*h输出低电平 11: 满足条件时, in_op*h输出反转	RW
b11~b10	OPZRH[1:0]	下溢条件2端口状态H	下溢条件2 (同时满足): 计数下溢, OCCR*h计数匹配 00: 满足条件时, in_op*h保持不变 01: 满足条件时, in_op*h输出高电平 10: 满足条件时, in_op*h输出低电平 11: 满足条件时, in_op*h输出反转	RW
b9~b8	OPUCH[1:0]	向上计数条件2端口状态H	向上计数条件2 (同时满足): 计数器向上计数, OCCR*h计数匹配 00: 满足条件时, in_op*h保持不变 01: 满足条件时, in_op*h输出高电平 10: 满足条件时, in_op*h输出低电平 11: 满足条件时, in_op*h输出反转	RW
b7~b6	OPPKH[1:0]	上溢条件2端口状态H	上溢条件2 (同时满足): 计数上溢, OCCR*h计数匹配 00: 满足条件时, in_op*h保持不变 01: 满足条件时, in_op*h输出高电平 10: 满足条件时, in_op*h输出低电平 11: 满足条件时, in_op*h输出反转	RW
b5~b4	OPDCH[1:0]	向下计数条件2端口状态H	向下计数条件2 (同时满足): 计数器向下计数, OCCR*h计数匹配 00: 满足条件时, in_op*h保持不变 01: 满足条件时, in_op*h输出高电平 10: 满足条件时, in_op*h输出低电平 11: 满足条件时, in_op*h输出反转	RW

b3	OCFZRH	下溢点OCFH状态H	条件 (同时满足): 计数下溢, OCCR*h计数匹配 0: 满足条件时, OCSR.OCFH位保持不变 1: 满足条件时, OCSR.OCFH位置位	RW
b2	OCFUCH	向上计数OCFH状态H	条件 (同时满足): 计数器向上计数, OCCR*h计数匹配 0: 满足条件时, OCSR.OCFH位保持不变 1: 满足条件时, OCSR.OCFH位置位	RW
b1	OCFPKH	上溢点OCFH状态H	条件 (同时满足): 计数上溢, OCCR*h计数匹配 0: 满足条件时, OCSR.OCFH位保持不变 1: 满足条件时, OCSR.OCFH位置位	RW
b0	OCFDCH	向下计数OCFH状态H	条件 (同时满足): 计数器向下计数, OCCR*h计数匹配 0: 满足条件时, OCSR.OCFH位保持不变 1: 满足条件时, OCSR.OCFH位置位	RW

复位值: 0x0000 0000

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	OCF PKL	OCF DCL

表 20-6 OCCR*h/I 计数匹配一览表

OCCR*h 计数匹配	OCCR*I 计数匹配	条件编号
0	0	1
0	1	2
1	0	3
1	1	4

注: 0: 不匹配, 1: 匹配

注: 该寄存器位描述用于 OCMRul、OCMRvl、OCMRwl、OCMRxl 时

位/位域	标记	位名	功能	读写
b31~b30	EOPNZRL[1:0]	下溢点条件3端口状态L	下溢点条件3 (同时满足): 计数下溢, OCCR*I计数不匹配, OCCR*h计数匹配 00: 满足条件时, in_op*I保持不变 01: 满足条件时, in_op*I输出高电平 10: 满足条件时, in_op*I输出低电平 11: 满足条件时, in_op*I输出反转	RW
b29~b28	EOPNPKL[1:0]	上溢条件3端口状态L	上溢条件3 (同时满足): 计数上溢, OCCR*I计数不匹配, OCCR*h计数匹配 00: 满足条件时, in_op*I保持不变 01: 满足条件时, in_op*I输出高电平 10: 满足条件时, in_op*I输出低电平 11: 满足条件时, in_op*I输出反转	RW
b27~b26	EOPZRL[1:0]	下溢条件4端口状态L	下溢条件4 (同时满足): 计数下溢, OCCR*I计数匹配, OCCR*h计数匹配 00: 满足条件时, in_op*I保持不变	RW

			01: 满足条件时, in_op*输出高电平 10: 满足条件时, in_op*输出低电平 11: 满足条件时, in_op*输出反转	
b25~b24	EOPUCL[1:0]	向上计数条件4端口状态L	向上计数条件4 (同时满足): 计数器向上计数, OCCR*i计数匹配, OCCR*h计数匹配 00: 满足条件时, in_op*保持不变 01: 满足条件时, in_op*输出高电平 10: 满足条件时, in_op*输出低电平 11: 满足条件时, in_op*输出反转	RW
b23~b22	EOPPKL[1:0]	上溢条件4端口状态L	上溢条件4 (同时满足): 计数上溢, OCCR*i计数匹配, OCCR*h计数匹配 00: 满足条件时, in_op*保持不变 01: 满足条件时, in_op*输出高电平 10: 满足条件时, in_op*输出低电平 11: 满足条件时, in_op*输出反转	RW
b21~b20	EOPDCL[1:0]	向下计数条件4端口状态L	向下计数条件4 (同时满足): 计数器向下计数, OCCR*i计数匹配, OCCR*h计数匹配 00: 满足条件时, in_op*保持不变 01: 满足条件时, in_op*输出高电平 10: 满足条件时, in_op*输出低电平 11: 满足条件时, in_op*输出反转	RW
b19~b18	EOPNUCL[1:0]	向上计数条件3端口状态L	向上计数条件3 (同时满足): 计数器向上计数, OCCR*i计数不匹配, OCCR*h计数匹配 00: 满足条件时, in_op*保持不变 01: 满足条件时, in_op*输出高电平 10: 满足条件时, in_op*输出低电平 11: 满足条件时, in_op*输出反转	RW
b17~b16	EOPNDCL[1:0]	向下计数条件3端口状态L	向下计数条件3 (同时满足): 计数器向下计数, OCCR*i计数不匹配, OCCR*h计数匹配 00: 满足条件时, in_op*保持不变 01: 满足条件时, in_op*输出高电平 10: 满足条件时, in_op*输出低电平 11: 满足条件时, in_op*输出反转	RW
b15~b14	OPNZRL[1:0]	下溢条件1端口状态L	下溢条件1 (同时满足): 计数下溢, OCCR*i计数不匹配, OCCR*h计数不匹配 00: 满足条件时, in_op*保持不变 01: 满足条件时, in_op*输出高电平 10: 满足条件时, in_op*输出低电平 11: 满足条件时, in_op*输出反转	RW
b13~b12	OPNPKL[1:0]	上溢条件1端口状态L	上溢条件1 (同时满足): 计数上溢, OCCR*i计数不匹配, OCCR*h计数不匹配 00: 满足条件时, in_op*保持不变 01: 满足条件时, in_op*输出高电平 10: 满足条件时, in_op*输出低电平 11: 满足条件时, in_op*输出反转	RW
b11~b10	OPZRL[1:0]	下溢条件2端口状态L	下溢条件2 (同时满足): 计数下溢, OCCR*i计数匹配, OCCR*h计数不匹配 00: 满足条件时, in_op*保持不变	RW

			01: 满足条件时, in_op*I输出高电平 10: 满足条件时, in_op*I输出低电平 11: 满足条件时, in_op*I输出反转	
b9~b8	OPUCL[1:0]	向上计数条件2端口状态L	向上计数条件2 (同时满足): 计数器向上计数, OCCR*I计数匹配, OCCR*h计数不匹配 00: 满足条件时, in_op*I保持不变 01: 满足条件时, in_op*I输出高电平 10: 满足条件时, in_op*I输出低电平 11: 满足条件时, in_op*I输出反转	RW
b7~b6	OPPKL[1:0]	上溢条件2端口状态L	上溢条件2 (同时满足): 计数上溢, OCCR*I计数匹配, OCCR*h计数不匹配 00: 满足条件时, in_op*I保持不变 01: 满足条件时, in_op*I输出高电平 10: 满足条件时, in_op*I输出低电平 11: 满足条件时, in_op*I输出反转	RW
b5~b4	OPDCL[1:0]	向下计数端口状态L	条件 (同时满足): 计数器向下计数, OCCR*I计数匹配, OCCR*h计数不匹配 00: 满足条件时, in_op*I保持不变 01: 满足条件时, in_op*I输出高电平 10: 满足条件时, in_op*I输出低电平 11: 满足条件时, in_op*I输出反转	RW
b3	OCFZRL	下溢点OCFL状态L	条件 (同时满足): 计数下溢, OCCR*I计数匹配 0: 满足条件时, OCSR.OCFL位保持不变 1: 满足条件时, OCSR.OCFL位置位	RW
b2	OCFUCL	向上计数OCFL状态L	条件 (同时满足): 计数器向上计数, OCCR*I计数匹配 0: 满足条件时, OCSR.OCFL位保持不变 1: 满足条件时, OCSR.OCFL位置位	RW
b1	OCFPKL	上溢点OCFL状态L	条件 (同时满足): 计数上溢, OCCR*I计数匹配 0: 满足条件时, OCSR.OCFL位保持不变 1: 满足条件时, OCSR.OCFL位置位	RW
b0	OCFDCL	向下计数OCFL状态L	条件 (同时满足): 计数器向下计数, OCCR*I计数匹配 0: 满足条件时, OCSR.OCFL位保持不变 1: 满足条件时, OCSR.OCFL位置位	RW

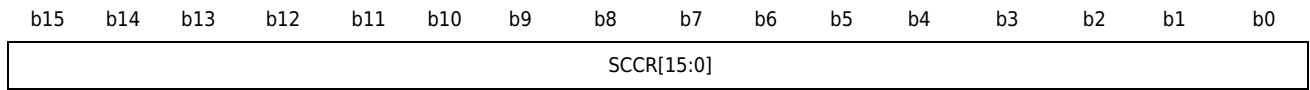
注意:

- 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。
- TMR4_O*L 可以由 OCCR*I 与计数器的计数值 (独立操作模式), 或者 OCCR*h 与计数器的计数值且 OCCR*I 与计数器的计数值 (链接操作模式) 来决定。向寄存器 OCMR*I 的 bit[31:20]和 bit[15:4] 写入相同的 12 位值, 同时将 OCMR*I[19:16]写入“0b0000”, 此时, in_op*I 输出将不受 OCCR*h 的影响, 而仅由 OCCR*I 决定。独立操作模式: 通道*h 由 OCCR*h 决定, 通道*L 由 OCCR*I 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*L 输出同时受 OCCR*h 和 OCCR*I 影响。
- 下面 2 种情况满足时, 也认为是符合 OCMR*h.OPPKH[7:6]设定的条件:
 - 1) CCSR.MODE=1 && 计数上溢 && OCCR*h=0xFFFF;

- 2) $OCER*.MCECH=1 \ \&\& \text{ 计数上溢} \ \&\& OCCR*h \geq CNTR$ 。
- 下面 2 种情况满足时, 也认为是符合 $OCMR*.OPPKL[7:6]$ 设定的条件:
 - 1) $CCSR.MODE=1 \ \&\& \text{ 计数上溢} \ \&\& OCCR*l=0xFFFF$;
 - 2) $OCER*.MCECL=1 \ \&\& \text{ 计数上溢} \ \&\& OCCR*l \geq CNTR$ 。

20.5.9 专用比较基准寄存器 (TMR4_SCCRm)

复位值: 0x0000



位/位域	标记	位名	功能	读写
b15~b0	SCCR[15:0]	专用比较基准值	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	RW

20.5.10 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	Res	EVTDS	EVTMS	Res	LMC	EVTOS[2:0]	BUFEN[1:0]					
位/位域	标记	位名	功能		读写										
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		RW										
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		RW										
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		RW										
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		RW										
b11~b10	Res	保留位	读出时为“0”, 写入时写“0”		RW										
b9	EVTDS	EVT延时对象选择	0: 在延时启动模式时, OCCR*h作为延时比较匹配对象 1: 在延时启动模式时, OCCR*l作为延时比较匹配对象 注: 该位在EVTMS=0时无效		RW										
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)		RW										
b7~b6	Res	保留位	读出时为“0”, 写入时写“0”		RW										
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)		RW										
b4~b2	EVTOS[2:0]	EVT输出选择	000: Special Event 0 (TMR4_SCM0) 的EVT输出有效 001: Special Event 1 (TMR4_SCM1) 的EVT输出有效 010: Special Event 2 (TMR4_SCM2) 的EVT输出有效 011: Special Event 3 (TMR4_SCM3) 的EVT输出有效 100: Special Event 4 (TMR4_SCM4) 的EVT输出有效 101: Special Event 5 (TMR4_SCM5) 的EVT输出有效 110: Special Event 6 (TMR4_SCM6) 的EVT输出有效 111: Special Event 7 (TMR4_SCM7) 的EVT输出有效		RW										
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 11: SCCR、SCMR缓存寄存器的值在计数下溢或上溢时写入SCCR、SCMR		RW										

20.5.11 专用扩展控制寄存器 (TMR4_SCER)

复位值: 0x0000

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

位/位域	标记	位名	功能	读写
b15~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
b8	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TMR4_PCT端口 1: 允许计数方向信号输出到TMR4_PCT端口	RW
b7~b4	Res	保留位	读出时为“0”，写入时写“0”	RW
b3~b0	EVTRS[3:0]	专用事件输出选择	0000: 禁止专用事件输出到端口 0001: Special Event 0 (TMR4_SCM0) 的EVT输出至TMR4_ADASM端口 0010: Special Event 1 (TMR4_SCM1) 的EVT输出至TMR4_ADASM端口 0011: Special Event 2 (TMR4_SCM2) 的EVT输出至TMR4_ADASM端口 0100: Special Event 3 (TMR4_SCM3) 的EVT输出至TMR4_ADASM端口 0101: Special Event 4 (TMR4_SCM4) 的EVT输出至TMR4_ADASM端口 0110: Special Event 5 (TMR4_SCM5) 的EVT输出至TMR4_ADASM端口 1000: Special Event 6 (TMR4_SCM6) 的EVT输出至TMR4_ADASM端口 1001: Special Event 7 (TMR4_SCM7) 的EVT输出至TMR4_ADASM端口 其他值: 禁止设定	RW

20.5.12 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	
Res								MPCE	MZCE	Res			AMC[3:0]			

位/位域	标记	位名	功能	读写
b15~b8	Res	保留位	读出时所有位为“1”，写入时所有位写“1”	RW
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	RW
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	RW
b5~b4	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

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。

20.5.13 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
Res								LVLS[1:0]		PWMMMD[1:0]		-	DIVCK[2:0]		

位/位域	标记	位名	功能	读写
b15~b8	Res	保留位	读出时所有位为“1”，写入时所有位写“1”	RW
b7~b6	LVLS[1:0]	PWM输出极性控制	00: TMR4_O*H和TMR4_O*L的输出都不反转 01: TMR4_O*H和TMR4_O*L的输出都反转 10: TMR4_O*H的输出反转, TMR4_O*L的输出不反转 11: TMR4_O*H的输出不反转, TMR4_O*L的输出反转	RW
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: 禁止设定	RW
b3	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

20.5.14 PWM 状态控制寄存器 (TMR4_PSCR)

复位值: 0x5555 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
OSXL[1:0]		OSXH[1:0]		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
Res				ODT[1:0]		AOE	MOE	OEXL	OEXH	OEWL	OEWH	OEVL	OEVH	OEUL	OEUH

位/位域	标记	位名	功能	读写
b31~b30	OSXL[1:0]	TMR4_OXL输出禁止状态	条件: 1. 发生EMB事件; 2. 软件写MOE=0; 3. MOE=1&OEXL=0 00: 满足以上条件之一时, TMR4_OXL端口正常输出 01: 满足以上条件之一时, TMR4_OXL端口输出为Hi-z 10: 满足以上条件之一时, TMR4_OXL端口输出固定为低电平 11: 满足以上条件之一时, TMR4_OXL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	RW
b29~b28	OSXH[1:0]	TMR4_OXH输出禁止状态	条件: 1. 发生EMB事件; 2. 软件写MOE=0; 3. MOE=1&OEXH=0 00: 满足以上条件之一时, TMR4_OXH端口正常输出 01: 满足以上条件之一时, TMR4_OXH端口输出为Hi-z 10: 满足以上条件之一时, TMR4_OXH端口输出固定为低电平 11: 满足以上条件之一时, TMR4_OXH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	RW
b27~b26	OSWL[1:0]	TMR4_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, TMR4_OWL端口正常输出 01: 满足以上条件之一时, TMR4_OWL端口输出为Hi-z 10: 满足以上条件之一时, TMR4_OWL端口输出固定为低电平 11: 满足以上条件之一时, TMR4_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	RW
b25~b24	OSWH[1:0]	TMR4_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, TMR4_OWH端口正常输出 01: 满足以上条件之一时, TMR4_OWH端口输出为Hi-z 10: 满足以上条件之一时, TMR4_OWH端口输出固定为低电平 11: 满足以上条件之一时, TMR4_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	RW
b23~b22	OSVL[1:0]	TMR4_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, TMR4_OVL端口正常输出 01: 满足以上条件之一时, TMR4_OVL端口输出为Hi-z 10: 满足以上条件之一时, TMR4_OVL端口输出固定为低电平 11: 满足以上条件之一时, TMR4_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	RW
b21~b20	OSVH[1:0]	TMR4_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, TMR4_OVH端口正常输出	RW

			01: 满足以上条件之一时, TMR4_OVH端口输出为Hi-z 10: 满足以上条件之一时, TMR4_OVH端口输出固定为低电平 11: 满足以上条件之一时, TMR4_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	
b19~b18	OSUL[1:0]	TMR4_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, TMR4_OUL端口正常输出 01: 满足以上条件之一时, TMR4_OUL端口输出为Hi-z 10: 满足以上条件之一时, TMR4_OUL端口输出固定为低电平 11: 满足以上条件之一时, TMR4_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	RW
b17~b16	OSUH[1:0]	TMR4_OUH输出禁止状态	条件: 1. 发生EMB事件; 2. 软件写MOE=0; 3. MOE=1且OEUH=0 00: 满足以上条件之一时, TMR4_OUH端口正常输出 01: 满足以上条件之一时, TMR4_OUH端口输出为Hi-z 10: 满足以上条件之一时, TMR4_OUH端口输出固定为低电平 11: 满足以上条件之一时, TMR4_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	RW
b15~b12	Res	保留位	读出时为“0”, 写入时写“0”	RW
b11~b10	ODT[1:0]	端口使能位生效时间	条件: 端口使能位OE*L、OE*H的值改变后 0X: 满足条件时, 立即生效 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	RW
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	RW
b8	MOE	主输出使能	0: TMR4_O*H/L端口输出由OS*H/L位指定的状态 1: OE*H、OE*L位设定为使能时TMR4_O*H、TMR4_O*L端口输出使能 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。	RW
b7	OEXL	TMR4_OXL输出使能	0: TMR4_OXL端口输出状态由OSXL位设定 1: MOE=1时, TMR4_OWL端口输出正常PWM波形	RW
b6	OEXH	TMR4_OXH输出使能	0: TMR4_OXH端口输出状态由OSXH位设定 1: MOE=1时, TMR4_OXH端口输出正常PWM波形	RW
b5	OEWL	TMR4_OWL输出使能	0: TMR4_OWL端口输出状态由OSWL位设定 1: MOE=1时, TMR4_OWL端口输出正常PWM波形	RW
b4	OEWL	TMR4_OWL输出使能	0: TMR4_OWH端口输出状态由OSWH位设定 1: MOE=1时, TMR4_OWH端口输出正常PWM波形	RW
b3	OEVL	TMR4_OVL输出使能	0: TMR4_OVL端口输出状态由OSVL位设定 1: MOE=1时, TMR4_OVL端口输出正常PWM波形	RW
b2	OEVL	TMR4_OVL输出使能	0: TMR4_OVH端口输出状态由OSVH位设定 1: MOE=1时, TMR4_OVH端口输出正常PWM波形	RW
b1	OEUL	TMR4_OUL输出使能	0: TMR4_OUL端口输出状态由OSUL位设定 1: MOE=1时, TMR4_OUL端口输出正常PWM波形	RW

b0	OEUH	TMR4_OUH输出使能	0: TMR4_OUH端口输出状态由OSUH位设定 1: MOE=1时, TMR4_OUH端口输出正常PWM波形	RW
----	------	--------------	---	----

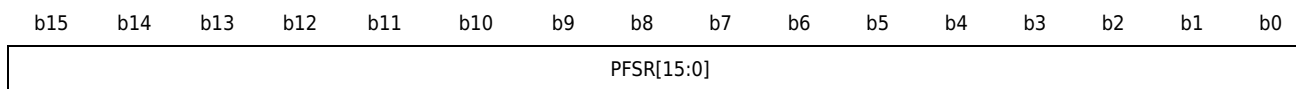
下表为 PWM 端口输出状态与寄存器设定值的关系说明。

表 20-7 PWM 端口输出状态与寄存器设定值

MOE位	OE*L/H位	OS*L/H位	TMR4_O*L/H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	XX	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

20.5.15 PWM 滤波控制寄存器 (TMR4_PFSRn)

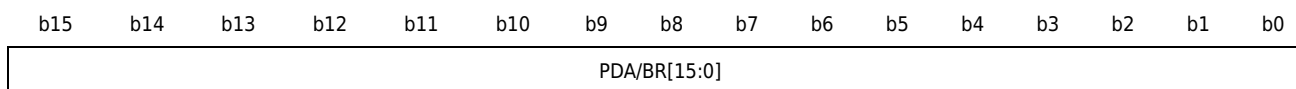
复位值: 0x0000



位/位域	标记	位名	功能	读写
			滤波计数初始值	
			注:	
b15~b0	PFSR[15:0]	滤波初始值	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。	RW

20.5.16 PWM 死区控制寄存器 (TMR4_PDARn)

复位值: 0x0000



位/位域	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	RW

20.5.17 重载控制状态寄存器 (TMR4_RCSR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res												RTSX	RTEX	RTICX	RTIFX
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 X	RTID W	RTID V	RTID U

位/位域	标记	位名	功能	读写
b31~b20	Res	保留位	读出时为“0”，写入时写“0”	RW
b19	RTSX	重载计数器停止X	0: 无操作 1: 停止重载计数器W并清除RTIFX 注: 该位在读出时始终为0	RW
b18	RTEX	重载计数器启动X	0: 写0无效，读出为0表示重载计数器X已停止 1: 写1启动重载计数器X，读出为1表示重载计数器X已启动	RW
b17	RTICX	清零计数匹配状态X	0: 无操作 1: 清零RTIFX标志位 注: 该位在读出时始终为0	RW
b16	RTIFX	计数匹配状态X	0: 重载计数器计数值与PFSRx发生未比较匹配 1: 重载计数器计数值与PFSRx发生比较匹配	R
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	RW
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	RW
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	RW
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	R
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	RW
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	RW
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	RW
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	R

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	RW
b6	RTEU	重载计数器启动U	0: 写0无效, 读数为0表示重载计数器U已停止 1: 写1启动重载计数器U, 读数为1表示重载计数器U已启动	RW
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	RW
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	R
b3	RTIDX	重载中断屏蔽X	0: 重载功能有效时, 重载中断X输出有效 1: 重载功能有效时, 重载中断X输出无效	RW
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	RW
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	RW
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	RW

20.5.18 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
Res								ITEN7	ITEN6	ITEN5	ITEN4	ITEN3	ITEN2	ITEN1	ITEN0
位/位域	标记	位名	功能		读写										
b15~b8	Res	保留位	读出时所有位为“1”，写入时所有位写“1”		RW										
b7	ITEN7	专用匹配中断7使能	0: 专用计数比较匹配事件7不使能 1: 专用计数比较匹配事件7使能		RW										
b6	ITEN6	专用匹配中断6使能	0: 专用计数比较匹配事件6不使能 1: 专用计数比较匹配事件6使能		RW										
b5	ITEN5	专用匹配中断5使能	0: 专用计数比较匹配事件5不使能 1: 专用计数比较匹配事件5使能		RW										
b4	ITEN4	专用匹配中断4使能	0: 专用计数比较匹配事件4不使能 1: 专用计数比较匹配事件4使能		RW										
b3	ITEN3	专用匹配中断3使能	0: 专用计数比较匹配事件3不使能 1: 专用计数比较匹配事件3使能		RW										
b2	ITEN2	专用匹配中断2使能	0: 专用计数比较匹配事件2不使能 1: 专用计数比较匹配事件2使能		RW										
b1	ITEN1	专用匹配中断1使能	0: 专用计数比较匹配事件1不使能 1: 专用计数比较匹配事件1使能		RW										
b0	ITEN0	专用匹配中断0使能	0: 专用计数比较匹配事件0不使能 1: 专用计数比较匹配事件0使能		RW										

20.5.19 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
Res								SF7	SF6	SF5	SF4	SF3	SF2	SF1	SF0

位/位域	标记	位名	功能	读写
b15~b8	Res	保留位	读出时为“0”，写入时写“0”	RW
b7	SF7	专用计数匹配标志位7	0: 未发生专用计数比较匹配事件7 1: 发生专用计数比较匹配事件7	RW
b6	SF6	专用计数匹配标志位6	0: 未发生专用计数比较匹配事件6 1: 发生专用计数比较匹配事件6	RW
b5	SF5	专用计数匹配标志位5	0: 未发生专用计数比较匹配事件5 1: 发生专用计数比较匹配事件5	RW
b4	SF4	专用计数匹配标志位4	0: 未发生专用计数比较匹配事件4 1: 发生专用计数比较匹配事件4	RW
b3	SF3	专用计数匹配标志位3	0: 未发生专用计数比较匹配事件3 1: 发生专用计数比较匹配事件3	RW
b2	SF2	专用计数匹配标志位2	0: 未发生专用计数比较匹配事件2 1: 发生专用计数比较匹配事件2	RW
b1	SF1	专用计数匹配标志位1	0: 未发生专用计数比较匹配事件1 1: 发生专用计数比较匹配事件1	RW
b0	SF0	专用计数匹配标志位0	0: 未发生专用计数比较匹配事件0 1: 发生专用计数比较匹配事件0	RW

21 紧急刹车模块（EMB）

21.1 概述

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，以控制定时器停止或更改向外部电机输出 PWM 信号的功能模块，下列源可用于产生控制事件：

- 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

21.2 功能说明

EMB 用于向具有 PWM 功能的定时器模块(Timer4, Timer6, HRPWM)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 9 个组群(group),其中 group0~group5 用于控制 HRPWM, group6~group7 用于控制 Timer6, group8 用于控制 Timer4, 分别对应 Timer4 的 1 个单元。EMB 结构框图如图 21-1 所示。

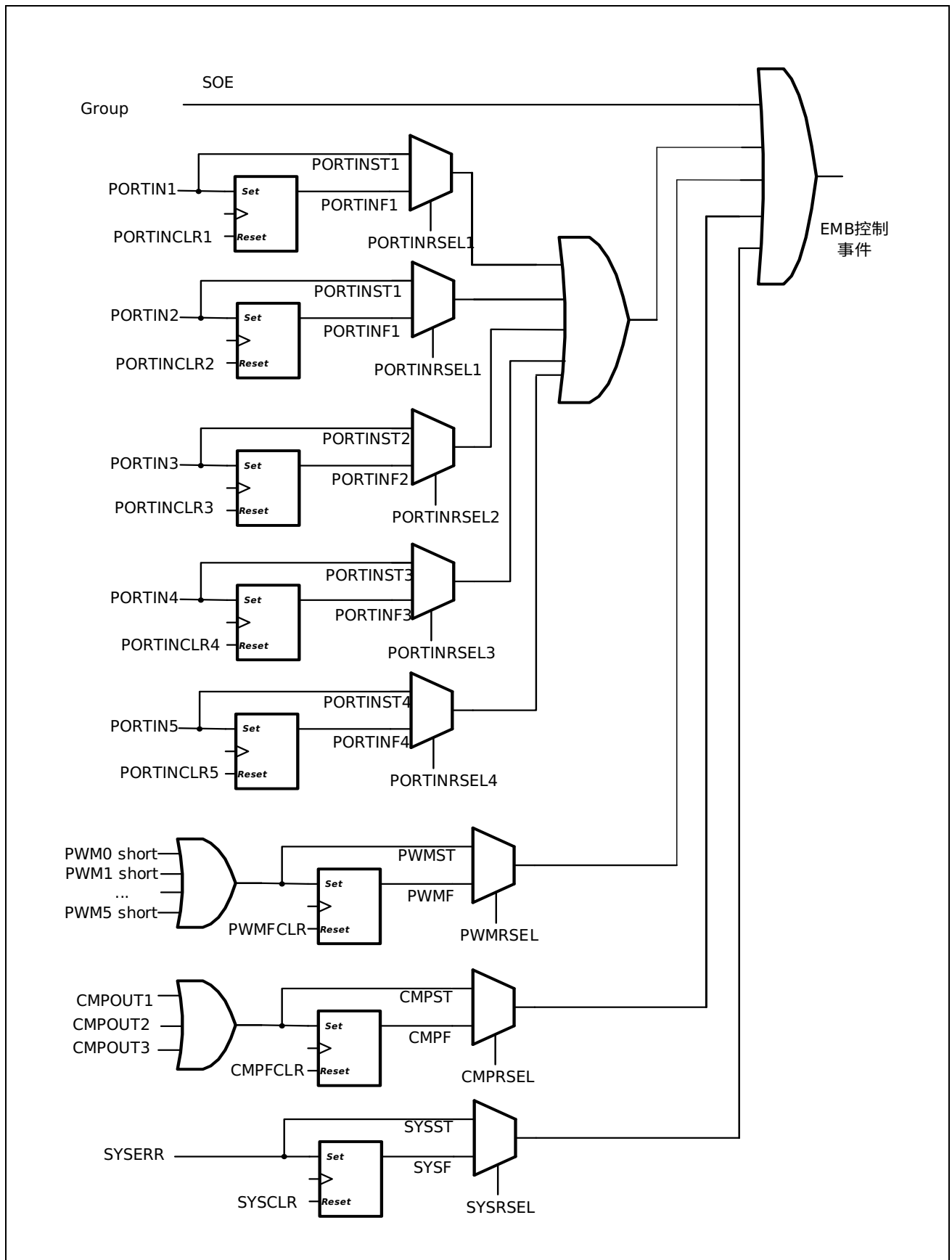


图 21-1 EMB 通道功能框图

21.2.1 EMB 结构框图

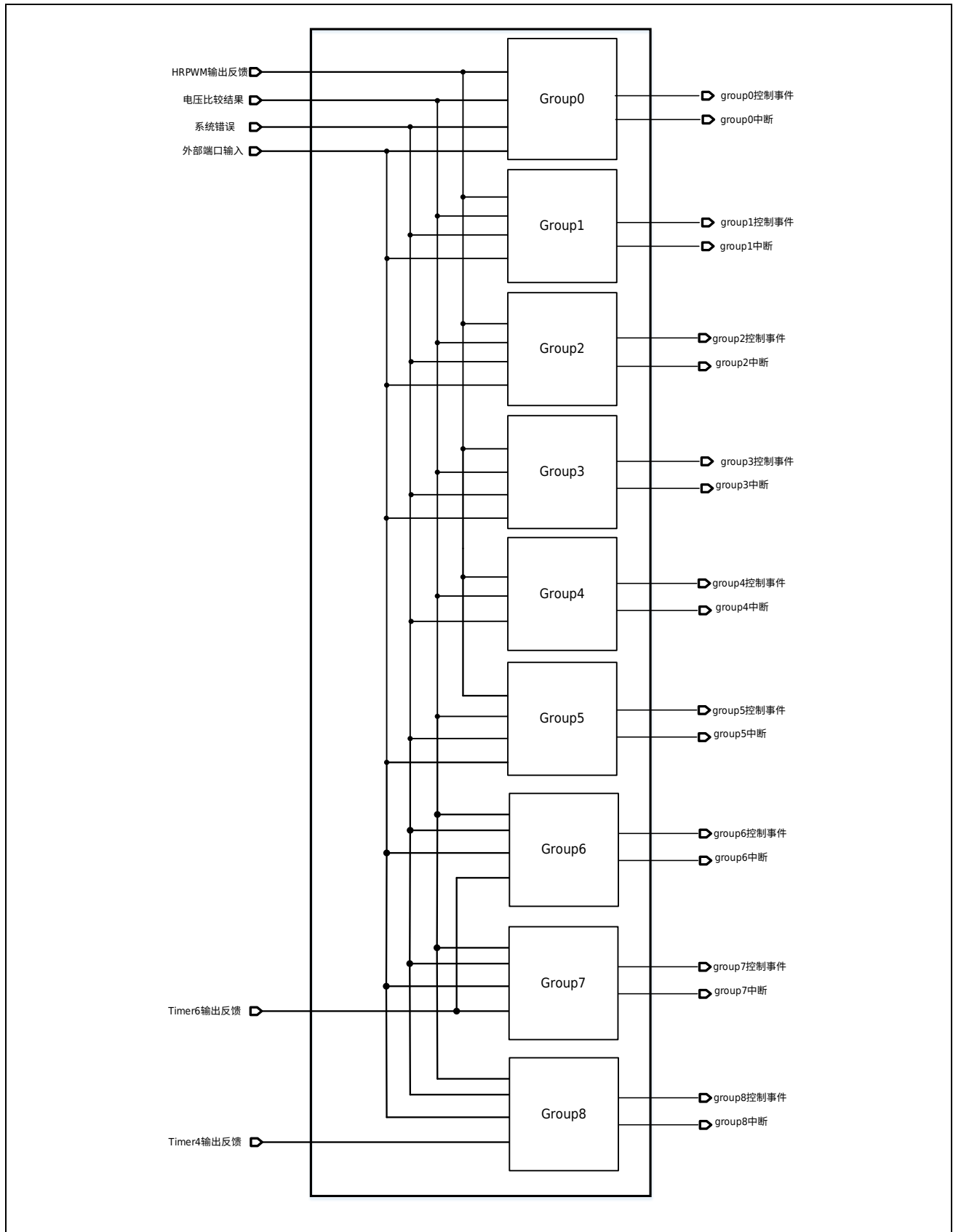


图 21-2 EMB 结构框图

21.2.2 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，group0~5 的端口分配如表 21-1 所示，group6~8 的端口分配如表 21-2 所示。

表 21-1 端口分配表

端口	PIN 名
EMB_PORT1_IN1	PA6、PB12、PA12、PB7
EMB_PORT1_IN2	PA15、PA11、PA6、PC3、PF3
EMB_PORT1_IN3	PA0、PA9、PB10、PC5、PB8
EMB_PORT1_IN4	PB11、PB9、PA1、PF2
EMB_PORT1_IN5	PC7、PA10、PA13

表 21-2 端口分配表

端口	PIN 名
EMB_PORT2_IN1	PA6、PB12、PA14、PA15、PB8
EMB_PORT2_IN2	PA6、PA11、PC3
EMB_PORT2_IN3	PA0、PA9、PC5
EMB_PORT2_IN4	PB5、PA1、PF2
EMB_PORT2_IN5	PD2、PA10、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=0~8, y=1~5) 置为有效，同时通过 EMB_CTL1_x.INVSEL y (x=0~8, y=1~5) 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=0~8, y=1~5)、EMB_CTL2_x.NFSELY (x=0~8, y=1~5) 以及 EMB_CTL2_x.NFSELY_2 (x=0~8, y=1~5) 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 N 次比较一致的方式进行滤波，比较一致的次数通过 EMB_CTL2_x.NFMD 配置。当滤波时钟采样到端口上 N 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 N 次一致的电平会被当作外部干扰滤掉，不传送到模块内部。

在配置完 EMB_CTL2_x 和 EMB_CTL1_x 寄存器后，对 EMB_STATCLR.x.PORTINFCLRy 写 1，确认 EMB_STATx.PORTINFy 为 0 后，再配置 EMB_INTENx.PORTINENy 位使能中断。

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=0~8, y=1~5) 被置位，同时端口输入控制标志位 EMB_STATx.PORTINFy (x=0~8, y=1~5) 被置位。当中断许可寄存器 EMB_INTENx.PORTINEN(x=0~8)=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=0~8, y=1~5) 选择由 EMB_STATx.PORTINSTy (x=0~8, y=1~5) 或者 EMB_STATx.PORTINFy (x=0~8, y=1~5) 向 HRPWM、

Timer6 和 Timer4 输出。HRPWM、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=0~8, y=1~5)=1，当端口输入电平转为无效时，EMB_STATx.PORTINSTy (x=0~8, y=1~5) 将自动清零，同时控制事件将立即释放。如果 EMB_RLSSELx.PORTINRSEly (x=0~8, y=1~5)=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=0~8)使 EMB_STATx.PORTINF (x=0~8)清 0，将控制事件释放。

21.2.3 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=0~6 或 x=6~7, y=0~3 或 x=8, y=0~3)置为有效，同时通过 EMB 控制寄存器 EMB_CTL2_x.PWMLVy (x=0~5, y=0~5 或 x=6~7, y=0~3 或 x=8, y=0~3)选择需要监控的有效电平。

完成设置后，EMB 监控 Timer6/HRPWM 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=0~8)被置位，同时 PWM 输出同相位控制标志位 EMB_STATx.PWMSF(x=0~8)被置位。当中断许可寄存器 EMB_INTENx.PWMEN (x=0~8)=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=0~8)选择由 EMB_STATx.PWMST (x=0~8)或者 EMB_STATx.PWMSF (x=0~8)向 Timer6 和 Timer4 输出。Timer6/HRPWM 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=0~8)=1，当 PWM 输出同高或同低状态解除，EMB_STATx.PWMST(x=0~8)将自动清零，同时控制事件将立即释放。如果 EMB_RLSSELx.PWMRSEL(x=0~8)=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 .PWMSFCLR (x=0~8)使 EMB_STATx.PWMSF (x=0~8)清 0 后，将控制事件释放。

表 21-3 EMB Group 对照表

端口名	功能	对应Group	EMB_CTL1 控制位	EMB_CTL2 控制位
HRPWM_m_PWMA(m=1~6)	HRPWM的互补PWM输出信号	Group0~5	PWMSEN[5:0]	PWMLV[5:0]
HRPWM_m_PWMB(m=1~6)				
TMR6_m_PWMA (m=1~4)	Timer6的互补PWM输出信号	Group6~7	PWMSEN[3:0]	PWMLV[3:0]
TMR6_m_PWMB (m=1~4)				
TMR4_OXH	Timer4_1的互补PWM输出信号	Group8	PWMSEN[3]	PWMLV[3]
TMR4_OXL			PWMSEN[2]	PWMLV[2]
TMR4_OUH				
TMR4_OUL				
TMR4_OVH			PWMSEN[1]	PWMLV[1]

端口名	功能	对应Group	EMB_CTL1 控制位	EMB_CTL2 控制位
TMR4_OVL			PWMSSEN[0]	PWMLV[0]
TMR4_OWH				
TMR4_OWL				

21.2.4 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 3 组比较结果向 Timer6/HRPWM 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.CMPEN_y(x=0~8)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=0~8)被置位，同时 EMB 电压比较器控制标志位 EMB_STATx.CMPF(x=0~8)被置位。当中断许可寄存器 EMB_INTENx.CMPINTEN(x=0~8)=1 时将产生中断。当多个电压比较器有效且 EMB 电压比较器控制标志位 EMB_STATx.CMPF(x=0~8)被置位时，通过 EMB 电压比较器标志位 EMB_STATx.CMP<y>F(x=0~8)确认比较器的源。

EMB 的控制事件通过 EMB 释放方式选择寄存器 EMB_RLSSELx.CMPRESEL_y(x=0~8)选择由 EMB_STATx.CMPST(x=0~8)或者 EMB_STATx.CMPF(x=0~8)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.CMPRESEL(x=0~8)=1，当电压比较器结果转为无效时，EMB_STATx.CMPST(x=0~8)将自动清零，同时控制事件将立即释放。如果 EMB_RLSSELx.CMPRESEL_y(x=0~8)=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_STATCLR_x.CMPFCLR(x=0~8)使 EMB_STATx.CMPF(x=0~8)清 0 后，或者将 EMB 电压比较器标志位 EMB_STATx.CMP<y>F 全部清除后，将控制事件释放。

在配置完 EMB_CTL2_x 和 EMB_CTL1_x 寄存器后，对 EMB_STATCLR_x.CMPFCLR 写 1，确认 EMB_STATx.CMPF 为 0 后，再配置 EMB_INTENx.CMPINTEN 位使能中断。

21.2.5 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、RAM (CACHERAM) 奇偶校验错误以及 SRAM 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=0~8)置为有效，设置 EMB_CTL1_x.PVDEN、LOCKUPEN、SRAMPYERREN、SRAMECCERREN、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=0~8)被置位，同时 EMB 系统错误控制标志位 EMB_STATx.SYSF(x=0~8)被置位。当中断许可寄存器 EMB_INTENx.

SYSINTEN(x=0~8)=1 时还将产生中断。当多个系统错误源有效且 EMB 系统错误控制标志位 EMB_STATx.SYSF (x=0~8)被置位时,通过 EMB_STATx(x=0~8)的 PVD 检测标志位、Cortex®-M4 LOCKUP 标志位、RAM (CACHERAM) 奇偶校验错误标志位、SRAM ECC 校验错误标志位以及振荡器停止标志位确认系统错误发生的原因。

EMB 的控制事件通过 EMB 释放方式选择寄存器 EMB_RLSSELx.SYSRSELy(x=0~8) 选择由 EMB_STATx.SYSST(x=0~8)或者 EMB_STATx.SYSF (x=0~8)向 Timer6、HRPWM 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=0~8)=1,当使能的全部系统错误源标志位转为无效时(清除对应标志位可参照相应模块),EMB_STATx.SYSST(x=0~8)将自动清零,同时控制事件将立即释放。如果 EMB_RLSSELx.SYSRSELy(x=0~8)=0,当使能的全部系统错误源标志位转为无效时(清除对应标志位可参照相应的模块),需要写 EMB 状态复位寄存器 EMB_STATCLRx.SYSFCLR(x=0~8)使 EMB_STATx.SYSF (x=0~8)清 0 后,或者通过写 EMB 状态复位寄存器 EMB_STATCLRx 的 b19 到 b23 位将各个系统错误标志位清零后,将控制事件释放。

注:

当 RAM (CACHERAM) 奇偶校验错误以及 SRAM ECC 校验错误和 PVD 检测使能时 (EMB_CTL1_x.PVDEN、RAMPYERREN、SRAMECCERREN 位为 1),无论 EMB_RLSSELx.SYSRSEL(x=0~8)设置为 0 还是 1,都需要清除相关模块内的对应的标志位,EMB_STATx.SYSST(x=0~8)才能转为无效。当 RAM (CACHERAM) 奇偶校验错误发生时,需要清除 SRAMC_CKSR 的 CACHE_PYERR 位;当 SRAM ECC 校验错误发生时,需要清除 SRAMC_CKSR 的 SRAMB_2ERR、SRAMB_1ERR、SRAM0_2ERR、SRAM0_1ERR 和 SRAMH_PYERR 位;当 PVD 检测发生时,需要清除 PWC_PVDDSR 的 PVD1DETFLG 和 PVD2DETFLG 位。

在 RAM (CACHERAM) 奇偶校验错误以及 SRAM ECC 校验错误和 PVD 检测使能之前,需要对上述相应模块内的标志位进行清除,避免 EMB 生成错误的控制事件信号并传送给 HRPWM、Timer6 和 Timer4。

21.2.6 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_SOE) 能够允许用户通过软件直接置位和复位的方式向 HRPWM、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。

21.3 寄存器描述

表 21-4 EMB 基地址

名称	基地址	描述
EMB0	0x4003 8600	EMB0基地址
EMB1	0x4003 8620	EMB1基地址
EMB2	0x4003 8640	EMB2基地址
EMB3	0x4003 8660	EMB3基地址
EMB4	0x4003 8680	EMB4基地址
EMB5	0x4003 86A0	EMB5基地址
EMB6	0x4003 86C0	EMB6基地址
EMB7	0x4003 86E0	EMB7基地址
EMB8	0x4003 8700	EMB8基地址

表 21-5 EMB 寄存器列表

寄存器	描述	偏移地址	位宽	复位值	说明
EMB_CTL1	EMB控制寄存器1	0x00	32	0x0000 0000	各个PWM输出控制事件的使能
EMB_CTL2	EMB控制寄存器2	0x04	32	0x0000 0000	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
EMB_SOE	EMB软件输出使能控制寄存器	0x08	32	0x0000 0000	软件产生PWM输出控制事件
EMB_STAT	EMB状态寄存器	0x0C	32	0x0000 0000	表示PWM输出控制的状态
EMB_STATCLR	EMB状态复位寄存器	0x10	32	0x0000 0000	清除PWM输出控制的状态
EMB_INTEN	EMB中断许可寄存器	0x14	32	0x0000 0000	中断使能
EMB_RLSSEL	EMB控制PWM输出释放方式选择寄存器	0x18	32	0x0000 0000	选择各个PWM输出控制事件的释放方式

21.3.1 EMB 控制寄存器 1 (EMB_CTL1, Group0~5)

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：0x4003 8600, 0x4003 8620, 0x4003 8640, 0x4003 8660, 0x4003 8680, 0x4003 86A0

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PVDEN	LOCKUPEN	RAMPYERREN	SRAMECCERREN	OSCSTPEN	INVSEL5	INVSEL4	INVSEL3	INVSEL2	INVSEL1	Res	PORTINEN5	PORTINEN4	PORTINEN3	PORTINEN2	PORTINEN1
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					PWMSEN5	PWMSEN4	PWMSEN3	PWMSEN2	PWMSEN1	PWMSEN0	SYSEN	Res	CMPE N3	CMPE N2	CMPE N1

位/位域	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	RW
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	RW
b29	RAMPYERREN	RAM 奇偶校验错误控制使能	0: RAM (CACHERAM) 奇偶校验错误发生时输出控制无效 1: RAM (CACHERAM) 奇偶校验错误发生时输出控制有效	RW
b28	SRAMECCERREN	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	RW
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	RW
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	RW
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	RW
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	RW
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	RW
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	RW
b21	Res	保留位	读时读出“0”，写入时请写“0”	RW
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	RW
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	RW
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	RW
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	RW
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	RW
b15~b11	Res	保留位	读时读出“0”，写入时请写“0”	RW

b10	PWMSEN5	HRPWM_6_PWMA/B短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	RW
b9	PWMSEN4	HRPWM_5_PWMA/B短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	RW
b8	PWMSEN3	HRPWM_4_PWMA/B短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	RW
b7	PWMSEN2	HRPWM_3_PWMA/B短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	RW
b6	PWMSEN1	HRPWM_2_PWMA/B短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	RW
b5	PWMSEN0	HRPWM_1_PWMA/B短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	RW
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	RW
b3	Res	保留位	读时读出“0”，写入时请写“0”	RW
b2	COMPEN3	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	RW
b1	COMPEN2	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	RW
b0	COMPEN1	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	RW

21.3.2 EMB 控制寄存器 1 (EMB_CTL1, Group6~7)

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：0x4003 86C0, 0x4003 86E0

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PVDEN	LOCKUPEN	RAMPYERREN	SRAMECCERREN	OSCSSTPEN	INVSEL5	INVSEL4	INVSEL3	INVSEL2	INVSEL1	Res	PORTINEN5	PORTINEN4	PORTINEN3	PORTINEN2	PORTINEN1
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							PWMSEN3	PWMSEN2	PWMSEN1	PWMSEN0	SYSEN	Res	CMPEN3	CMPEN2	CMPEN1

位/位域	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	RW
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	RW
b29	RAMPYERREN	RAM 奇偶校验错误控制使能	0: RAM (CACHERAM) 奇偶校验错误发生时输出控制无效 1: RAM (CACHERAM) 奇偶校验错误发生时输出控制有效	RW
b28	SRAMECCERREN	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	RW
b27	OSCSSTPEN	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	RW
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	RW
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	RW
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	RW
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	RW
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	RW
b21	Res	保留位	读时读出“0”，写入时请写“0”	RW
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	RW
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	RW
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	RW
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	RW
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	RW
b15~b9	Res	保留位	读时读出“0”，写入时请写“0”	RW

b8	PWMSEN3	TMR6_4_PWMA/B 短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	RW
b7	PWMSEN2	TMR6_3_PWMA/B 短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	RW
b6	PWMSEN1	TMR6_2_PWMA/B 短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	RW
b5	PWMSEN0	TMR6_1_PWMA/B 短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	RW
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	RW
b3	Res	保留位	读时读出“0”，写入时请写“0”	RW
b2	COMPEN3	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	RW
b1	COMPEN2	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	RW
b0	COMPEN1	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	RW

21.3.3 EMB 控制寄存器 1 (EMB_CTL1, Group8)

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：0x4003 8700

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PVDEN	LOCKUPEN	RAMPYERREN	SRAMECCERREN	OSCSTPEN	INVSEL5	INVSEL4	INVSEL3	INVSEL2	INVSEL1	Res	PORTINEN5	PORTINEN4	PORTINEN3	PORTINEN2	PORTINEN1
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							PWMSEN3	PWMSEN2	PWMSEN1	PWMSEN0	SYSEN	Res	CMPE N3	CMPE N2	CMPE N1

位/位域	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	RW
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	RW
b29	RAMPYERREN	RAM 奇偶校验错误控制使能	0: RAM (CACHERAM) 奇偶校验错误发生时输出控制无效 1: RAM (CACHERAM) 奇偶校验错误发生时输出控制有效	RW
b28	SRAMECCERREN	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	RW
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	RW
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	RW
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	RW
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	RW
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	RW
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	RW
b21	Res	保留位	读时读出“0”，写入时请写“0”	RW
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	RW
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	RW
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	RW
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	RW
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	RW
b15~b9	Res	保留位	读时读出“0”，写入时请写“0”	RW

b8	PWMSEN3	TMR4_OXH/L 短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	RW
b7	PWMSEN2	TMR4_OUH/L 短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	RW
b6	PWMSEN1	TMR4_OVH/L 短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	RW
b5	PWMSEN0	TMR4_OWH/L 短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	RW
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	RW
b3	Res	保留位	读时读出“0”，写入时请写“0”	RW
b2	COMPEN2	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	RW
b1	COMPEN1	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	RW
b0	COMPEN0	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	RW

21.3.4 EMB 控制寄存器 2 (EMB_CTL2, Group0~5)

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：0x4003 8604, 0x4003 8624, 0x4003 8644, 0x4003 8664, 0x4003 8684, 0x4003 86A4

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	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
NFMD	NFSEL 5_2	NFSEL 4_2	NFSEL 3_2	NFSEL 2_2	NFSEL 1_2	Res				PWML V5	PWML V4	PWML V3	PWML V2	PWML V1	PWML V0

位/位域	标记	位名	功能	读写
b31	Res	保留位	读时读出“0”，写入时请写“0”	RW
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	
b29~b28	NFSEL5[1:0]	端口5数字滤波器滤波时钟选择	该位与NFSEL5_2组合使用 当NFSEL5_2=0时 00: 使用PCLK0滤波 01: 使用PCLK0的8分频滤波 10: 使用PCLK0的32分频滤波 11: 使用PCLK0的128分频滤波 当NFSEL5_2=1时 00: 使用PCLK0的2分频滤波 01: 使用PCLK0的4分频滤波 10: 使用PCLK0的16分频滤波 11: 使用PCLK0的64分频滤波	
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	RW
b26~b25	NFSEL4[1:0]	端口4数字滤波器滤波时钟选择	该位与NFSEL4_2组合使用 当NFSEL4_2=0时 00: 使用PCLK0滤波 01: 使用PCLK0的8分频滤波 10: 使用PCLK0的32分频滤波 11: 使用PCLK0的128分频滤波 当NFSEL4_2=1时 00: 使用PCLK0的2分频滤波 01: 使用PCLK0的4分频滤波 10: 使用PCLK0的16分频滤波 11: 使用PCLK0的64分频滤波	RW
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	RW
b23~b22	NFSEL3[1:0]	端口3数字滤波器滤波时钟选择	该位与NFSEL3_2组合使用 当NFSEL3_2=0时 00: 使用PCLK0滤波 01: 使用PCLK0的8分频滤波	RW

			10: 使用PCLK0的32分频滤波 11: 使用PCLK0的128分频滤波 当NFSEL3_2=1时 00: 使用PCLK0的2分频滤波 01: 使用PCLK0的4分频滤波 10: 使用PCLK0的16分频滤波 11: 使用PCLK0的64分频滤波	
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	RW
b20~b19	NFSEL2[1:0]	端口2数字滤波器滤波时钟选择	该位与NFSEL2_2组合使用 当NFSEL2_2=0时 00: 使用PCLK0滤波 01: 使用PCLK0的8分频滤波 10: 使用PCLK0的32分频滤波 11: 使用PCLK0的128分频滤波 当NFSEL2_2=1时 00: 使用PCLK0的2分频滤波 01: 使用PCLK0的4分频滤波 10: 使用PCLK0的16分频滤波 11: 使用PCLK0的64分频滤波	RW
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	RW
b17~b16	NFSEL1[1:0]	端口1数字滤波器滤波时钟选择	该位与NFSEL1_2组合使用 当NFSEL1_2=0时 00: 使用PCLK0滤波 01: 使用PCLK0的8分频滤波 10: 使用PCLK0的32分频滤波 11: 使用PCLK0的128分频滤波 当NFSEL1_2=1时 00: 使用PCLK0的2分频滤波 01: 使用PCLK0的4分频滤波 10: 使用PCLK0的16分频滤波 11: 使用PCLK0的64分频滤波	RW
b15	NFMD	滤波次数选择	0: 滤波次数为3次 1: 滤波次数为2次	RW
b14	NFSEL5_2	端口5数字滤波器滤波时钟选择2	该位与NFSEL5[1:0]组合使用	RW
b13	NFSEL4_2	端口4数字滤波器滤波时钟选择2	该位与NFSEL4[1:0]组合使用	RW
b12	NFSEL3_2	端口3数字滤波器滤波时钟选择2	该位与NFSEL3[1:0]组合使用	RW
b11	NFSEL2_2	端口2数字滤波器滤波时钟选择2	该位与NFSEL2[1:0]组合使用	RW
b10	NFSEL1_2	端口1数字滤波器滤波时钟选择2	该位与NFSEL1[1:0]组合使用	RW
b9~b6	Res	保留位	读时读出“0”，写入时请写“0”	RW
b5	PWMLV5	HRPWM_6_PWMA/B 输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	RW
b4	PWMLV4	HRPWM_5_PWMA/B 输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	RW
b3	PWMLV3	HRPWM_4_PWMA/B 输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	RW

b2	PWMLV2	HRPWM_3_PWMA/B 输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	RW
b1	PWMLV1	HRPWM_2_PWMA/B 输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	RW
b0	PWMLV0	HRPWM_1_PWMA/B 输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	RW

21.3.5 EMB 控制寄存器 2 (EMB_CTL2, Group6~7)

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：0x4003 86C4, 0x4003 86E4

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	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
NFMD	NFSEL 5_2	NFSEL 4_2	NFSEL 3_2	NFSEL 2_2	NFSEL 1_2	Res					PWML V3	PWML V2	PWML V1	PWML V0	

位/位域	标记	位名	功能	读写
b31	Res	保留位	读时读出“0”，写入时请写“0”	RW
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	
b29~b28	NFSEL5[1:0]	端口5数字滤波器滤波时钟选择	该位与NFSEL5_2组合使用 当NFSEL5_2=0时 00: 使用PCLK0滤波 01: 使用PCLK0的8分频滤波 10: 使用PCLK0的2分频滤波 11: 使用PCLK0的128分频滤波 当NFSEL5_2=1时 00: 使用PCLK0的2分频滤波 01: 使用PCLK0的4分频滤波 10: 使用PCLK0的16分频滤波 11: 使用PCLK0的64分频滤波	
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	RW
b26~b25	NFSEL4[1:0]	端口4数字滤波器滤波时钟选择	该位与NFSEL4_2组合使用 当NFSEL4_2=0时 00: 使用PCLK0滤波 01: 使用PCLK0的8分频滤波 10: 使用PCLK0的32分频滤波 11: 使用PCLK0的128分频滤波 当NFSEL4_2=1时 00: 使用PCLK0的2分频滤波 01: 使用PCLK0的4分频滤波 10: 使用PCLK0的16分频滤波 11: 使用PCLK0的64分频滤波	RW
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	RW
b23~b22	NFSEL3[1:0]	端口3数字滤波器滤波时钟选择	该位与NFSEL3_2组合使用 当NFSEL3_2=0时 00: 使用PCLK0滤波 01: 使用PCLK0的8分频滤波	RW

			10: 使用PCLK0的32分频滤波 11: 使用PCLK0的128分频滤波 当NFSEL3_2=1时 00: 使用PCLK0的2分频滤波 01: 使用PCLK0的4分频滤波 10: 使用PCLK0的16分频滤波 11: 使用PCLK0的64分频滤波	
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	RW
b20~b19	NFSEL2[1:0]	端口2数字滤波器滤波时钟选择	该位与NFSEL2_2组合使用 当NFSEL2_2=0时 00: 使用PCLK0滤波 01: 使用PCLK0的8分频滤波 10: 使用PCLK0的32分频滤波 11: 使用PCLK0的128分频滤波 当NFSEL2_2=1时 00: 使用PCLK0的2分频滤波 01: 使用PCLK0的4分频滤波 10: 使用PCLK0的16分频滤波 11: 使用PCLK0的64分频滤波	RW
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	RW
b17~b16	NFSEL1[1:0]	端口1数字滤波器滤波时钟选择	该位与NFSEL1_2组合使用 当NFSEL1_2=0时 00: 使用PCLK0滤波 01: 使用PCLK0的8分频滤波 10: 使用PCLK0的32分频滤波 11: 使用PCLK0的128分频滤波 当NFSEL1_2=1时 00: 使用PCLK0的2分频滤波 01: 使用PCLK0的4分频滤波 10: 使用PCLK0的16分频滤波 11: 使用PCLK0的64分频滤波	RW
b15	NFMD	滤波次数选择	0: 滤波次数为3次 1: 滤波次数为2次	RW
b14	NFSEL5_2	端口5数字滤波器滤波时钟选择2	该位与NFSEL5[1:0]组合使用	RW
b13	NFSEL4_2	端口4数字滤波器滤波时钟选择2	该位与NFSEL4[1:0]组合使用	RW
b12	NFSEL3_2	端口3数字滤波器滤波时钟选择2	该位与NFSEL3[1:0]组合使用	RW
b11	NFSEL2_2	端口2数字滤波器滤波时钟选择2	该位与NFSEL2[1:0]组合使用	RW
b10	NFSEL1_2	端口1数字滤波器滤波时钟选择2	该位与NFSEL1[1:0]组合使用	RW
b9~b4	Res	保留位	读时读出“0”，写入时请写“0”	RW
b3	PWMLV3	TMR6_4_PWM/A/B输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	RW
b2	PWMLV2	TMR6_3_PWM/A/B输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	RW
b1	PWMLV1	TMR6_2_PWM/A/B输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	RW

b0	PWMLV0	TMR6_1_PWMA/B输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	RW
----	--------	-----------------------	----------------------------	----

21.3.6 EMB 控制寄存器 2 (EMB_CTL2, Group8)

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：0x4003 8704

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	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
NFMD	NFSEL5_2	NFSEL4_2	NFSEL3_2	NFSEL2_2	NFSEL1_2	Res					PWMLV3	PWMLV2	PWMLV1	PWMLV0	

位/位域	标记	位名	功能	读写
b31	Res	保留位	读时读出“0”，写入时请写“0”	RW
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	-
b29~b28	NFSEL5[1:0]	端口5数字滤波器滤波时钟选择	该位与NFSEL5_2组合使用 当NFSEL5_2=0时 00：使用PCLK0滤波 01：使用PCLK0的8分频滤波 10：使用PCLK0的32分频滤波 11：使用PCLK0的128分频滤波 当NFSEL5_2=1时 00：使用PCLK0的2分频滤波 01：使用PCLK0的4分频滤波 10：使用PCLK0的16分频滤波 11：使用PCLK0的64分频滤波	-
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	RW
b26~b25	NFSEL4[1:0]	端口4数字滤波器滤波时钟选择	该位与NFSEL4_2组合使用 当NFSEL4_2=0时 00：使用PCLK0滤波 01：使用PCLK0的8分频滤波 10：使用PCLK0的32分频滤波 11：使用PCLK0的128分频滤波 当NFSEL4_2=1时 00：使用PCLK0的2分频滤波 01：使用PCLK0的4分频滤波 10：使用PCLK0的16分频滤波 11：使用PCLK0的64分频滤波	RW
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	RW
b23~b22	NFSEL3[1:0]	端口3数字滤波器滤波时钟选择	该位与NFSEL3_2组合使用 当NFSEL3_2=0时 00：使用PCLK0滤波 01：使用PCLK0的8分频滤波	RW

			10: 使用PCLK0的32分频滤波 11: 使用PCLK0的128分频滤波 当NFSEL3_2=1时 00: 使用PCLK0的2分频滤波 01: 使用PCLK0的4分频滤波 10: 使用PCLK0的16分频滤波 11: 使用PCLK0的64分频滤波	
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	RW
b20~b19	NFSEL2[1:0]	端口2数字滤波器滤波时钟选择	该位与NFSEL2_2组合使用 当NFSEL2_2=0时 00: 使用PCLK0滤波 01: 使用PCLK0的8分频滤波 10: 使用PCLK0的32分频滤波 11: 使用PCLK0的128分频滤波 当NFSEL2_2=1时 00: 使用PCLK0的2分频滤波 01: 使用PCLK0的4分频滤波 10: 使用PCLK0的16分频滤波 11: 使用PCLK0的64分频滤波	RW
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	RW
b17~b16	NFSEL1[1:0]	端口1数字滤波器滤波时钟选择	该位与NFSEL1_2组合使用 当NFSEL1_2=0时 00: 使用总线时钟滤波 01: 使用PCLK0的8分频滤波 10: 使用PCLK0的32分频滤波 11: 使用PCLK0的128分频滤波 当NFSEL1_2=1时 00: 使用PCLK0的2分频滤波 01: 使用PCLK0的4分频滤波 10: 使用PCLK0的16分频滤波 11: 使用PCLK0的64分频滤波	RW
b15	NFMD	滤波次数选择	0: 滤波次数为3次 1: 滤波次数为2次	RW
b14	NFSEL5_2	端口5数字滤波器滤波时钟选择2	该位与NFSEL5[1:0]组合使用	RW
b13	NFSEL4_2	端口4数字滤波器滤波时钟选择2	该位与NFSEL4[1:0]组合使用	RW
b12	NFSEL3_2	端口3数字滤波器滤波时钟选择2	该位与NFSEL3[1:0]组合使用	RW
b11	NFSEL2_2	端口2数字滤波器滤波时钟选择2	该位与NFSEL2[1:0]组合使用	RW
b10	NFSEL1_2	端口1数字滤波器滤波时钟选择2	该位与NFSEL1[1:0]组合使用	RW
b9~b4	Res	保留位	读时读出“0”，写入时请写“0”	RW
b3	PWMLV3	TMR4_OXH/L输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	RW
b2	PWMLV2	TMR4_OUH/L输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	RW
b1	PWMLV1	TMR4_OVH/L输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	RW

b0	PWMLV0	TMR4_OWH/L输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	RW
----	--------	--------------------	----------------------------	----

21.3.7 EMB 软件输出使能控制寄存器 (EMB_SOE)

地址: 0x4003 8608, 0x4003 8628, 0x4003 8648, 0x4003 8668, 0x4003 8688, 0x4003 86A8,
0x4003 86C8, 0x4003 86E8, 0x4003 8708

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b1	Res	保留位	读时读出“0”，写入时请写“0”	RW
b0	SOE	软件控制输出	0: PWM正常输出 1: PWM停止输出	RW

21.3.8 EMB 状态寄存器 (EMB_STAT)

地址：0x4003 860C, 0x4003 862C, 0x4003 864C, 0x4003 866C, 0x4003 868C, 0x4003 86AC,
0x4003 86CC, 0x4003 86EC, 0x4003 870C

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				CMP3F	CMP2F	CMP1F	PVDF	LOCKUPF	RAMPYERRF	SRAMECCERRF	OSCF	PORTINST5	PORTINST4	PORTINST3	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PORTINST2	PORTINST1	Res	PORTINF5	PORTINF4	PORTINF3	PORTINF2	PORTINF1	SYSST	CMPST	PWMST	Res	SYSF	CMPF	PWMSF	Res

位/位域	标记	位名	功能	读写
b31~b27	Res	保留位	读时读出“0”，写入时请写“0”	RW
b26	CMP3F	电压比较器3标志位	电压比较器3结果有效时置位 EMB_STATCLR.CMP1FCLR写1时清0 或者 EMB_STATCLR.PWMSCLR写1时清0	R
b25	CMP2F	电压比较器2标志位	电压比较器2结果有效时置位 EMB_STATCLR.CMP1FCLR写1时清0 或者 EMB_STATCLR.PWMSCLR写1时清0	R
b24	CMP1F	电压比较器1标志位	电压比较器1结果有效时置位 EMB_STATCLR.CMP1FCLR写1时清0 或者 EMB_STATCLR.PWMSCLR写1时清0	R
b23	PVDF	PVD检测标志位	PVD检测结果有效时置位 EMB_STATCLR.PVDFCLR写1时清0 或者 EMB_STATCLR.SYSFCLR写1时清0	R
b22	LOCKUPF	Cortex®-M4 LOCKUP标志位	Cortex®-M4 LOCKUP发生时置位 EMB_STATCLR.LOCKUPFCLR写1时清0 或者 EMB_STATCLR.SYSFCLR写1时清0	R
b21	RAMPYERRF	Cache RAM奇偶校验错误标志位	Cache RAM奇偶校验错误发生时置位 EMB_STATCLR.RAMPYERRFCLR写1时清0 或者 EMB_STATCLR.SYSFCLR写1时清0	R
b20	SRAMECCERRF	SRAM ECC校验错误标志位	SRAM ECC校验错误发生时置位 EMB_STATCLR.SRAMECCERRFCLR写1时清0 或者 EMB_STATCLR.SYSFCLR写1时清0	R
b19	OSCF	振荡器停止标志位	振荡器停止时置位 EMB_STATCLR.OSCFCLR写1时清0 或者 EMB_STATCLR.SYSFCLR写1时清0	R
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	Res	保留位	读时读出“0”，写入时请写“0”	RW
b12	PORTINF5	端口5输入控制标志位	EMB_STAT.PORTINST5为1时置位 EMB_STATCLR.PORTINFCLR5写1时清0	R
b11	PORTINF4	端口4输入控制标志位	EMB_STAT.PORTINST4为1时置位 EMB_STATCLR.PORTINFCLR4写1时清0	R
b10	PORTINF3	端口3输入控制标志位	EMB_STAT.PORTINST3为1时置位 EMB_STATCLR.PORTINFCLR3写1时清0	R
b9	PORTINF2	端口2输入控制标志位	EMB_STAT.PORTINST2为1时置位 EMB_STATCLR.PORTINFCLR2写1时清0	R
b8	PORTINF1	端口1输入控制标志位	EMB_STAT.PORTINST1为1时置位 EMB_STATCLR.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	Res	保留位	读时读出“0”，写入时请写“0”	RW
b3	SYSF	系统错误标志位	EMB_STAT.SYSST为1时置位 EMB_STATCLR.OSFCLR写1时清0 当b23-b19位全部清除为0时，该位自动清除	R
b2	CMPF	电压比较器控制标志位	EMB_STAT.CMPST为1时置位 EMB_STATCLR.CMPFCLR写1时清0 当b26-b24位全部清除为0时，该位自动清除	R
b1	PWMSF	PWM输出同相位控制标志位	EMB_STAT.PWMST为1时置位 EMB_STATCLR.PWMSCLR写1时清0	R
b0	Res	保留位	读时读出“0”，写入时请写“0”	RW

21.3.9 EMB 状态复位寄存器 (EMB_STATCLR)

地址：0x4003 8610, 0x4003 8630, 0x4003 8650 0x4003 8670, 0x4003 8690, 0x4003 86B0,
0x4003 86D0, 0x4003 86F0, 0x4003 8710

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res					CMP3FCLR	CMP2FCLR	CMP1FCLR	PVDFCLR	LOCKUPFCLR	RAMPYERRFCLR	SRAMECCERRFCLR	OSCFCLR	Res		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			PORTINFCLR5	PORTINFCLR4	PORTINFCLR3	PORTINFCLR2	PORTINFCLR1	Res				SYSFCLR	CMPFCLR	PWMSFCLR	Res

位/位域	标记	位名	功能	读写
b31~b27	Res	保留位	读时读出“0”，写入时请写“0”	W
b26	CMP3FCLR	复位 EMB_STAT.CMP3F	0: 无任何效果 1: 将EMB_STAT.CMP3F清0 读该寄存器位无任何效果	W
b25	CMP2FCLR	复位 EMB_STAT.CMP2F	0: 无任何效果 1: 将EMB_STAT.CMP2F清0 读该寄存器位无任何效果	W
b24	CMP1FCLR	复位 EMB_STAT.CMP1F	0: 无任何效果 1: 将EMB_STAT.CMP1F清0 读该寄存器位无任何效果	W
b23	PVDFCLR	复位 EMB_STAT.PVDF	0: 无任何效果 1: 将EMB_STAT.PVDF清0 读该寄存器位无任何效果	W
b22	LOCKUPFCLR	复位 EMB_STAT.LOCKUPFCLR	0: 无任何效果 1: 将EMB_STAT.LOCKUPFCLR清0 读该寄存器位无任何效果	W
b21	RAMPYERRFCLR	复位 EMB_STAT.RAMPYERRF	0: 无任何效果 1: 将EMB_STAT.RAMPYERRF清0 读该寄存器位无任何效果	W
b20	SRAMECCERRFCLR	复位 EMB_STAT.SRAMECCERRF	0: 无任何效果 1: 将EMB_STAT.SRAMECCERRF清0 读该寄存器位无任何效果	W
b19	OSCFCLR	复位 EMB_STAT.OSCF	0: 无任何效果 1: 将EMB_STAT.OSCF清0 读该寄存器位无任何效果	W
b18~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	Res	保留位	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	Res	保留位	读时读出“0”, 写入时请写“0”	W

21.3.10 EMB 中断许可寄存器 (EMB_INTEN)

地址: 0x4003 8614, 0x4003 8634, 0x4003 8654 0x4003 8674, 0x4003 8694, 0x4003 86B04
0x4003 86D4, 0x4003 86F4, 0x4003 8714

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			PORTIN TEN5	PORTIN TEN4	PORTIN TEN3	PORTIN TEN2	PORTIN TEN1	Res				SYSIN TEN	CMPIN TEN	PWMS INTEN	Res

位/位域	标记	位名	功能	读写
b31~b13	Res	保留位	读时读出“0”，写入时请写“0”	RW
b12	PORTININTEN5	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	RW
b11	PORTININTEN4	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	RW
b10	PORTININTEN3	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	RW
b9	PORTININTEN2	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	RW
b8	PORTININTEN1	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	RW
b7~b4	Res	保留位	读时读出“0”，写入时请写“0”	RW
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	RW
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	RW
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	RW
b0	Res	保留位	读时读出“0”，写入时请写“0”	RW

21.3.11 EMB 控制 PWM 输出释放方式选择寄存器 (EMB_RLSSEL)

地址: 0x4003 8618, 0x4003 8638, 0x4003 8658 0x4003 8678, 0x4003 8698, 0x4003 86B8,
0x4003 86D8, 0x4003 86F8, 0x4003 8718

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			PORTIN RSEL5	PORTIN RSEL4	PORTIN RSEL3	PORTIN RSEL2	PORTIN RSEL1	Res				SYSR SEL	CMPR SEL	PWMR SEL	Res

位/位域	标记	位名	功能	读写
b31~b13	Res	保留位	读时读出“0”，写入时请写“0”	RW
b12	PORTINRSEL5	端口输入5控制释放方式选择	0: EMB_STAT.PORTINF5=0时释放PWM输出控制 1: EMB_STAT.PORTINST5=0时释放PWM输出控制	RW
b11	PORTINRSEL4	端口输入4控制释放方式选择	0: EMB_STAT.PORTINF4=0时释放PWM输出控制 1: EMB_STAT.PORTINST4=0时释放PWM输出控制	RW
b10	PORTINRSEL3	端口输入3控制释放方式选择	0: EMB_STAT.PORTINF3=0时释放PWM输出控制 1: EMB_STAT.PORTINST3=0时释放PWM输出控制	RW
b9	PORTINRSEL2	端口输入2控制释放方式选择	0: EMB_STAT.PORTINF2=0时释放PWM输出控制 1: EMB_STAT.PORTINST2=0时释放PWM输出控制	RW
b8	PORTINRSEL1	端口输入1控制释放方式选择	0: EMB_STAT.PORTINF1=0时释放PWM输出控制 1: EMB_STAT.PORTINST1=0时释放PWM输出控制	RW
b7~b4	Res	保留位	读时读出“0”，写入时请写“0”	RW
b3	SYSRSEL	系统错误控制释放方式选择	0: EMB_STAT.SYSF=0时释放PWM输出控制 1: EMB_STAT.SYSYST=0时释放PWM输出控制	RW
b2	CMRSEL	比较器比较结果控制释放方式选择	0: EMB_STAT.CMPF=0时释放PWM输出控制 1: EMB_STAT.CMPST=0时释放PWM输出控制	RW
b1	PWMRSEL	PWM输出控制释放方式选择	0: EMB_STAT.PWMSF=0时释放PWM输出控制 1: EMB_STAT.PWMST=0时释放PWM输出控制	RW
b0	Res	保留位	读时读出“0”，写入时请写“0”	RW

22 通用定时器 (TimerA)

22.1 概述

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 5 个单元 TimerA，其中单元 1(TimerA1) 为 32 位定时器；单元 2 (TimerA2) ~单元 5 (TimerA5) 为 16 位定时器；5 个单元最多可实现 20 路 PWM 输出。

22.2 主要特性

TimerA 基本的功能及特性如表 22-1 所示。

表 22-1 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

22.3 功能说明

22.3.1 TimerA 框图

TimerA 的基本框图如图 22-1 所示。图中“<t>”为单元编号，即“<t>”为 1~5，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 5 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 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。

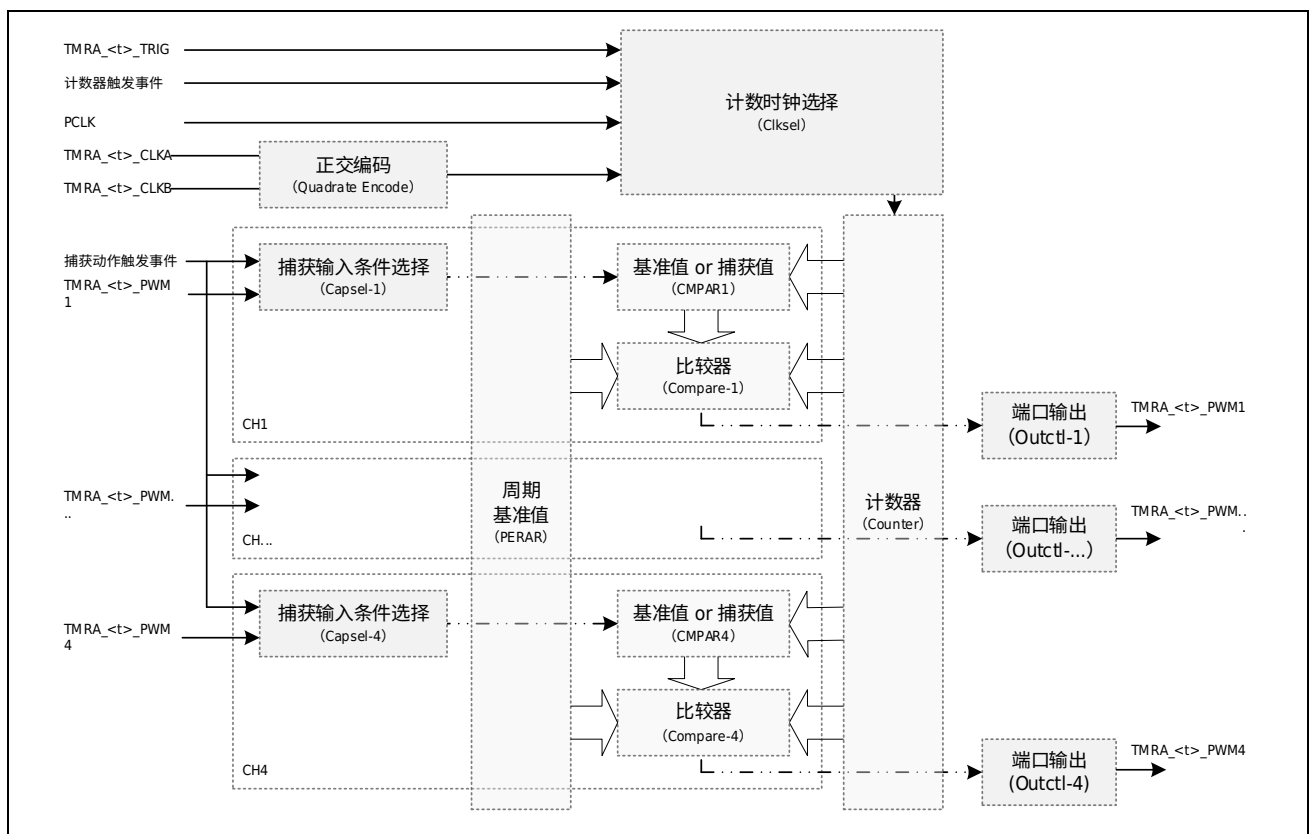


图 22-1 TimerA 基本框图

表 22-2 所示为 TimerA 的输入输出端口列表。

表 22-2 TimerA 端口列表

端口名	方向	功能
TMRA_<t>_PWMm	输入输出	捕获输入事件端口或PWM输出端口 (m=1~4)
TMRA_<t>_CLKA	输入	正交编码计数事件输入端口
TMRA_<t>_CLKB		
TMRA_<t>_TRIG	输入	硬件触发启动、停止、清零事件输入端口

22.3.2 对称单元

TimerA 的单元 1 和单元 2 互为对称单元，单元 3 和单元 4 互为对称单元，以此类推。对称单元间可联动，实现级联计数、同步启动等，详见以下各章节描述。

22.3.3 波形模式

TimerA 有两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 22-2、图 22-3 所示。

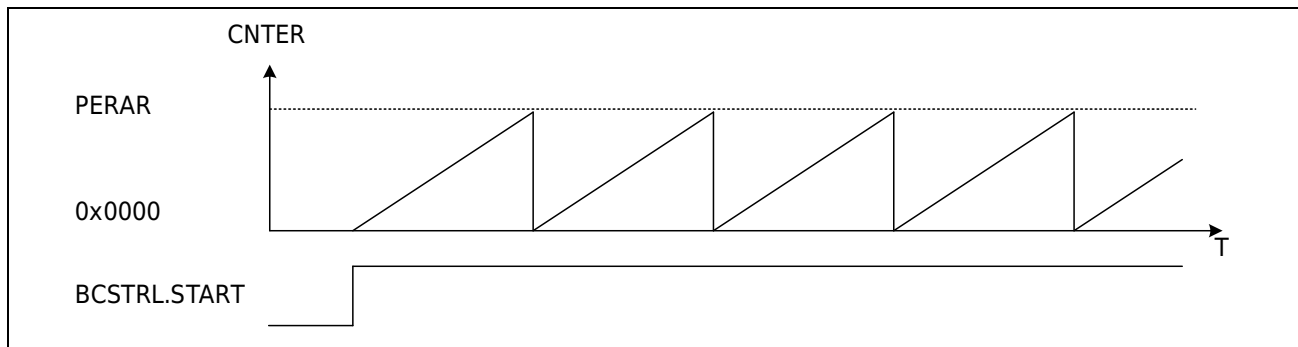


图 22-2 锯齿波波形（递加计数）

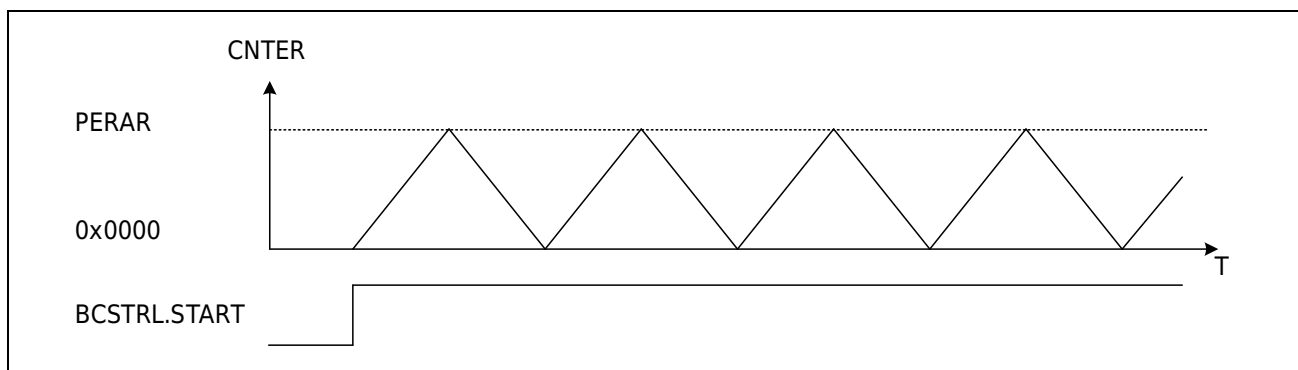


图 22-3 三角波波形

22.3.4 时钟源选择

TimerA 的计数时钟可以有以下几种选择：

- a) PCLK 的 1、2、4、8、16、32、64、128、256、512、1024 分频（BCSTRL.CKDIV[3:0]设定）
- b) TMRA_<t>_TRIG 端口事件输入（HCUPR[9:8]或 HCDOR[9:8]设定）
- c) 内部计数器触发事件输入（HCUPR[10]或 HCDOR[10]设定）
- d) 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）
- e) TMRA_<t>_CLKA、TMRA_<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]部分），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。

22.3.5 比较输出

各个 TimerA 单元内部均含有 4 个通道的比较输出（TMRA_<t>_PWMm），可在计数值与比较基准值比较匹配时输出指定的电平。TMRA_CMPARm 寄存器分别对应了 TMRA_<t>_PWMm 输出端口的计数比较基准值。当定时器的计数值和 TMRA_CMPARm 相等时，TMRA_<t>_PWMm 端口输出指定的电平（m=1~4）。

TMRA_<t>_PWMm 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（PCONRm）的 STAC、STPC、CMPC、PERC、FORC 位设定控制（m=1~4）。图 22-4 为单元 1 的比较输出动作例。

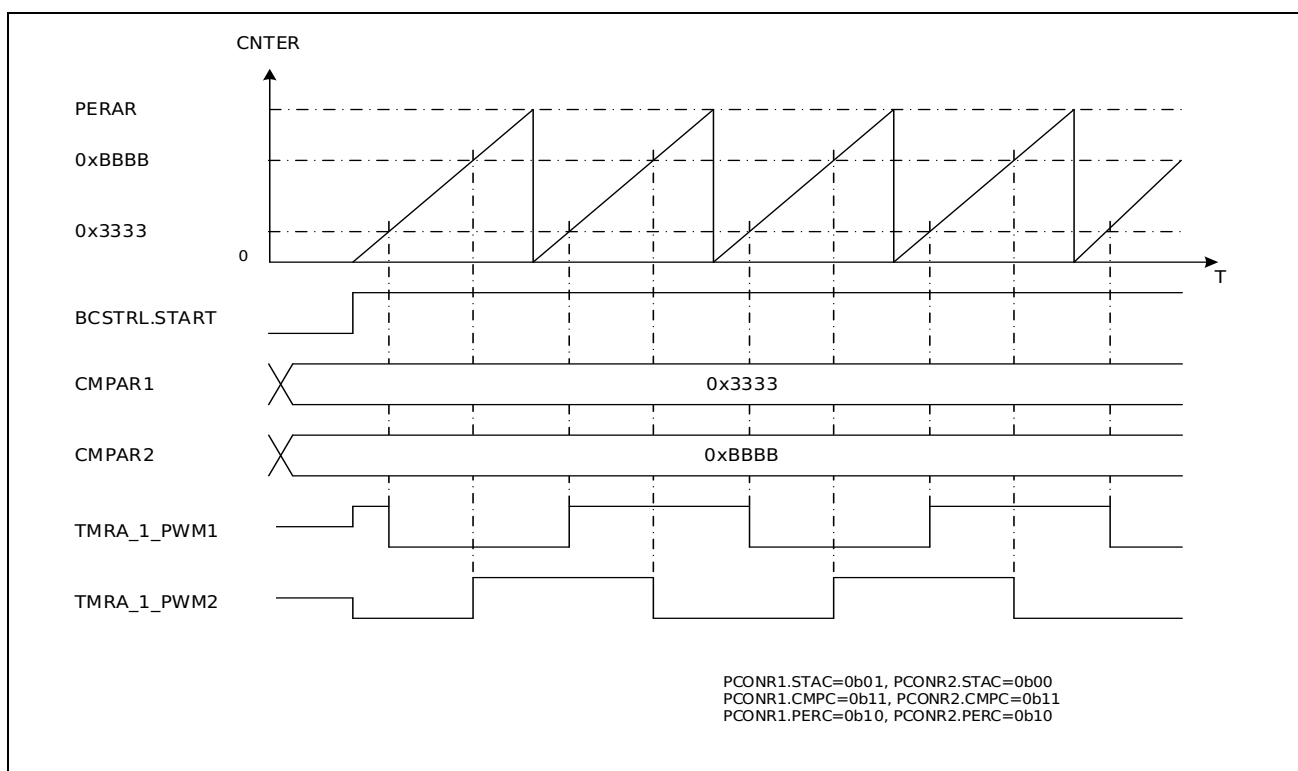


图 22-4 比较输出动作

22.3.6 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 (CCONRm) 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 (CMPARm) 中 (m=1~4)。

捕获输入条件可以选择内部捕获动作触发事件 (通过 TMRA_TRGSELx, x=0~3) 寄存器选择，具体参考【TimerA 硬件触发事件选择寄存器 (TMRA_TRGSELx, x=0~3)】章节)、TMRA_<t>_PWMm 端口输入等，具体的条件选择可通过捕获控制寄存器 (CCONRm) 的 HICP 位来设定 (m=1~4)。

内部触发事件选择寄存器为 5 个单元的 TimerA 所共有。具体分配如下表所示。

表 22-3 内部触发事件 TRGSEL 选择关系对应表

TimerA 单元	事件分类	所对应的选择寄存器
TimerA1	计数器触发事件： HCONR、HCUPR、HCDOR 所指的 TRGSEL 选择事件	TMRA_TRGSEL0
TimerA2		TMRA_TRGSEL1
TimerA3		TMRA_TRGSEL2
TimerA4		TMRA_TRGSEL3
TimerA5		TMRA_TRGSEL0
TimerA1	捕获动作触发事件： CCONR 中所指的 TRGSEL 选择事件	TMRA_TRGSEL1
TimerA2		TMRA_TRGSEL0
TimerA3		TMRA_TRGSEL3
TimerA4		TMRA_TRGSEL2
TimerA5		TMRA_TRGSEL1

图 22-5 为捕获输入动作例。

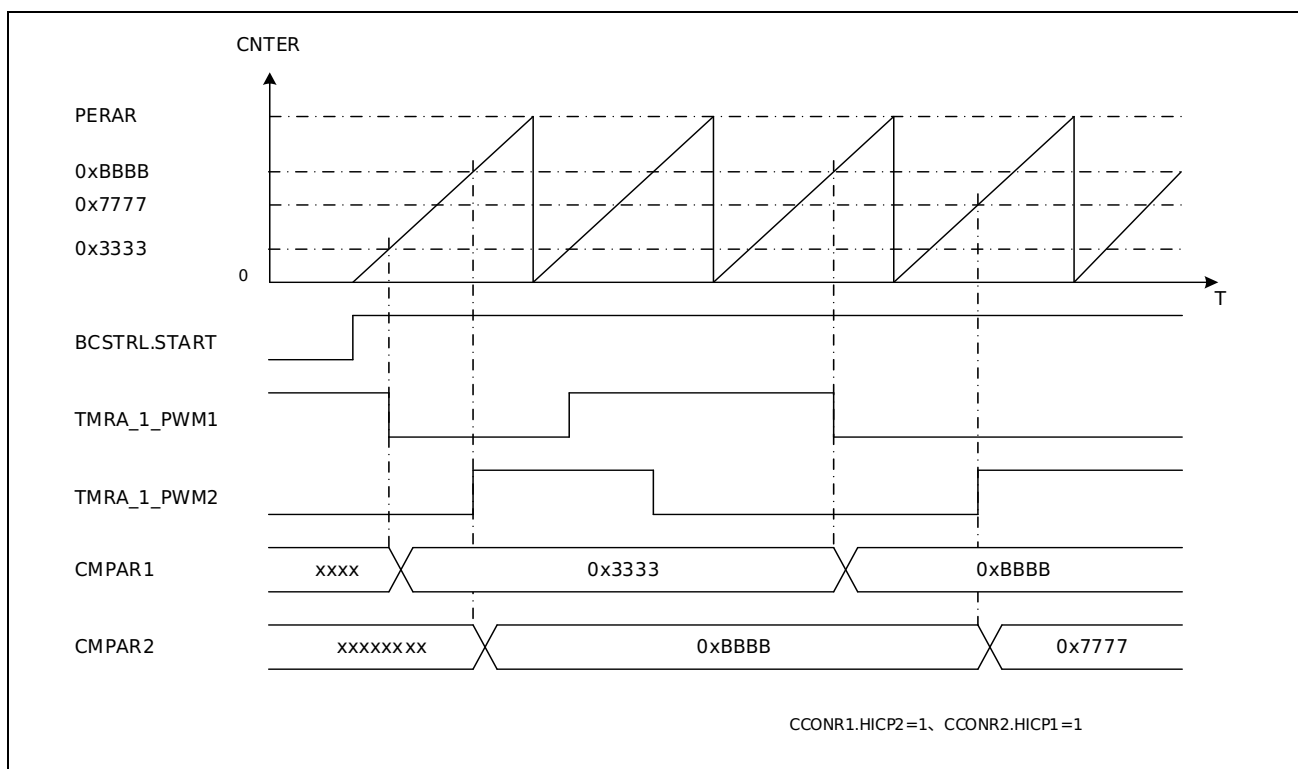


图 22-5 捕获输入动作

22.3.7 同步启动

TimerA 的对称单元，可以实现软件同步启动或硬件同步启动，由奇数单元同步启动偶数单元。

当对称单元中偶数单元的 BCSTR1.SYNST 位设定为 1 时，偶数单元与奇数单元的同步启动功能有效。此时，若软件设定奇数单元的 BCSTR1.START 位为 1，被同步的偶数单元的计数器开始软件同步计数；若硬件设定奇数单元的 HCONR.HSTA1~0 中任意位为 1，且其对应硬件事件发生时，被同步的偶数单元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，被同步的偶数单元的 HCONR.HSTA1~0 的对应位也必须设定为有效。

图 22-6 为设定单元 2 的 BCSTR1.SYNST=1 时的软件同步启动例。

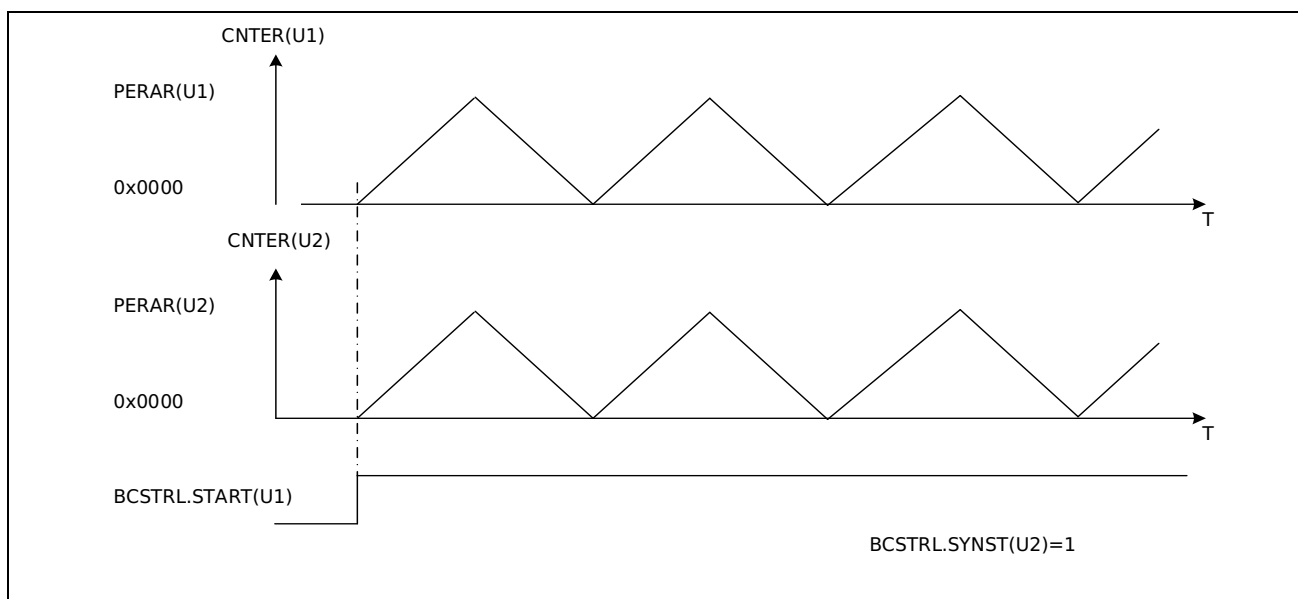


图 22-6 软件同步动作

另外，产品还配置了一个计数同步使能寄存器 (TMR_SYNENR)，用于同步启动产品所搭载的所有 Timer（包括 Timer0、Timer4、Timer6、TimerA）。使用方法和寄存器说明参考【通用定时器 (Timer0)】章节。

22.3.8 数字滤波

各个单元的 TMRA_<t>_CLKA、TMRA_<t>_CLKB、TMRA_<t>_TRIG、TMRA_<t>_PWMm（捕获输入功能时）端口输入都有数字滤波功能。各端口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 (CCONRm) 的对应位来实现 (m=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 22-7 所示为 TMRA_1_CLKA 端口滤波动作例。

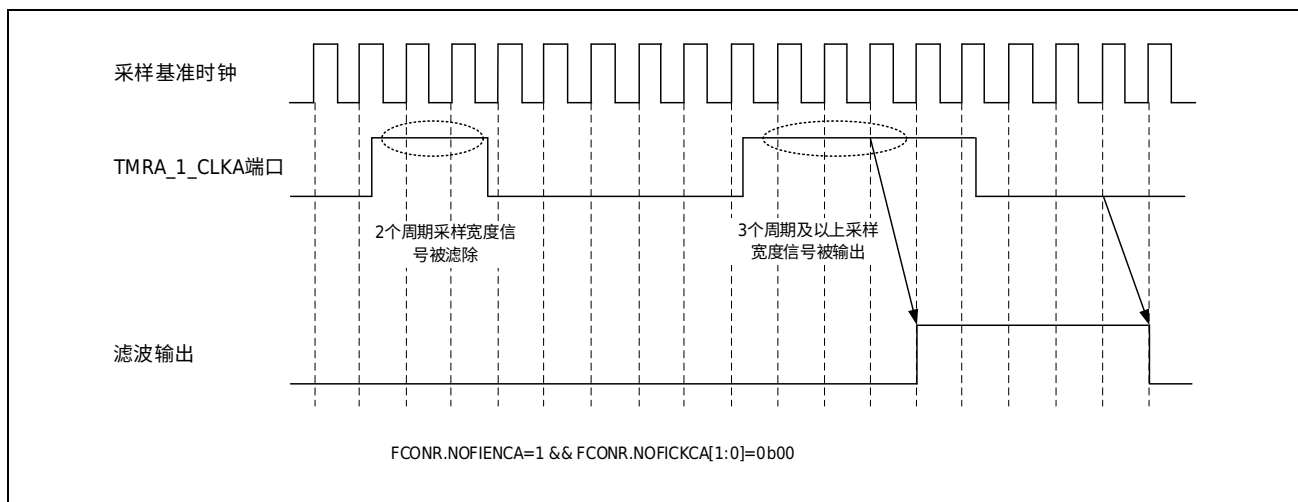


图 22-7 时钟输入端口的滤波功能

22.3.9 缓存功能

TimerA 的共计 4 个比较值寄存器 (CMPARm) 可以成对实现缓存功能 (m=1~4)。即 CMPAR2 作为 CMPAR1 的缓存比较值、CMPAR4 作为 CMPAR3 的缓存比较值。缓存控制寄存器 (BCONRm) 分别实现对四组缓存功能的控制 (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 (CMPAR2/4->CMPAR1/3)。该“特定时间点”有以下几种情况:

- a) 硬件计数模式时, 计数到上溢点 (HCUPR[12:0]! = 0 时) 或下溢点 (HCDOR[12:0]! = 0 时) 或发生硬件清零计数器动作
- b) 锯齿波计数模式 (BCSTRM.MODE=0) 时, 计数器计数到上溢点 (BCSTRM.DIR=1 时) 或下溢点 (BCSTRM.DIR=0 时) 或发生硬件清零计数器动作
- c) 三角波计数模式 (BCSTRM.MODE=1) 时, 计数到峰点 (BCSTRM.DIR=1 && BCONRm.BSE0=1 时) (m=1~2)
- d) 三角波计数模式 (BCSTRM.MODE=1) 时, 计数到谷点 (BCSTRM.DIR=0 && BCONRm.BSE1=1 时) (m=1~2)

当缓存控制寄存器 (BCONRm) 的 BSEN 位被置位时, 缓存特殊功能变为有效 (m=1~2)。对称单元的 CNTER 被读时, 本单元的 CMPAR 发生一次缓存传送 (CMPAR2/4->CMPAR1/3)。

下图 22-9 所示, 是锯齿波模式时的缓存传送与特殊缓存传送示意图。

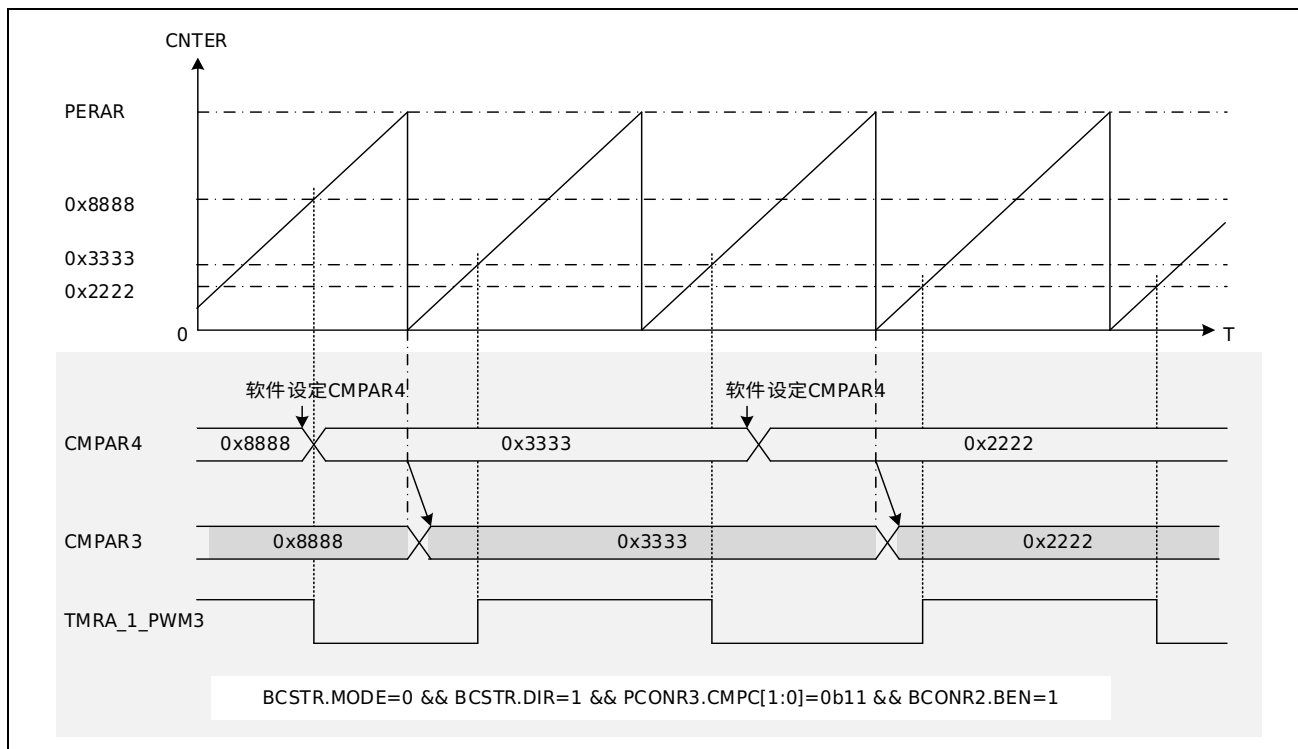


图 22-8 锯齿波模式时缓存动作

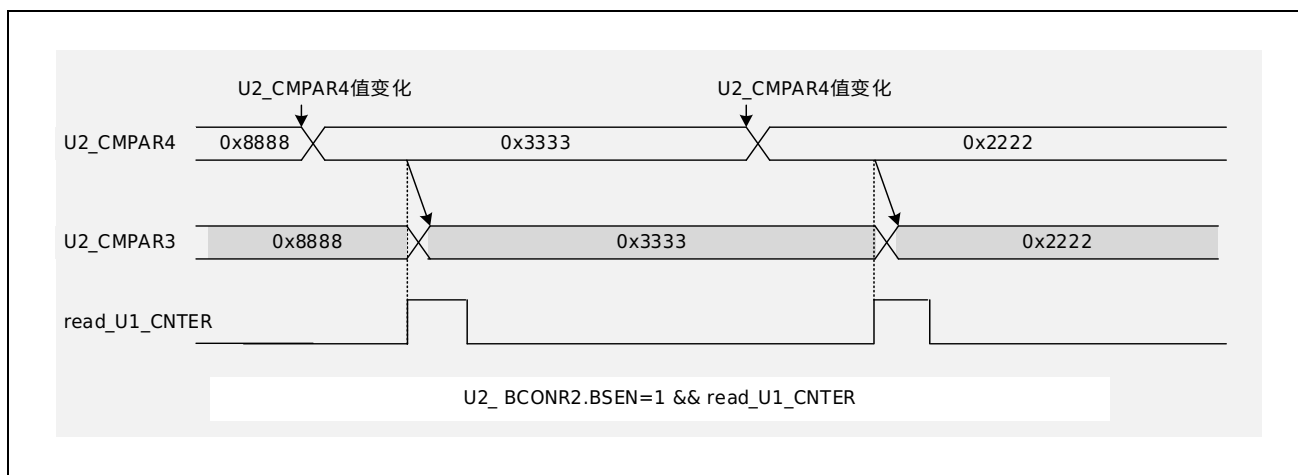


图 22-9 锯齿波模式时缓存与特殊缓存动作

22.3.10 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 (BCSTRL.MODE=0、BCSTRL.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 2、1 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 22-10 所示是单元 1、2 级联计数的示意图。

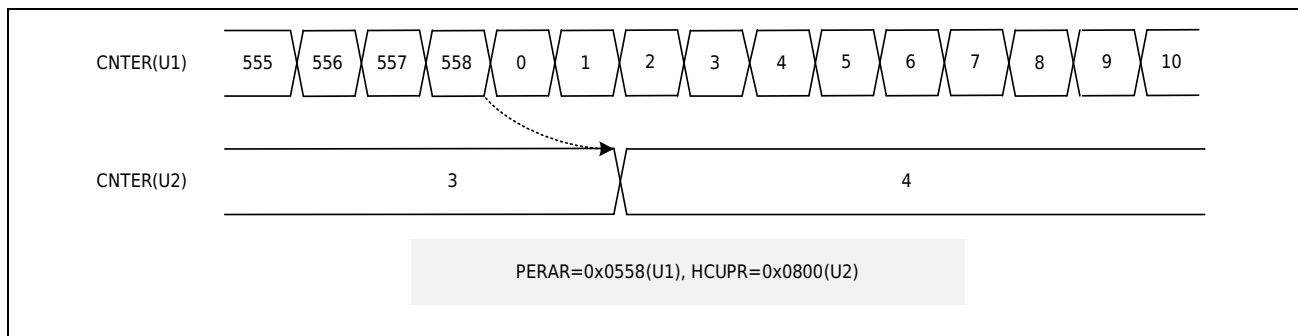


图 22-10 32 位级联计数动作

22.3.11 PWM 输出

22.3.11.1 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=0b11），就可实现在一个周期内产生单边对齐的 PWM 输出。

图 22-11 所示，是锯齿波模式下，单边对齐 PWM 输出波形例。

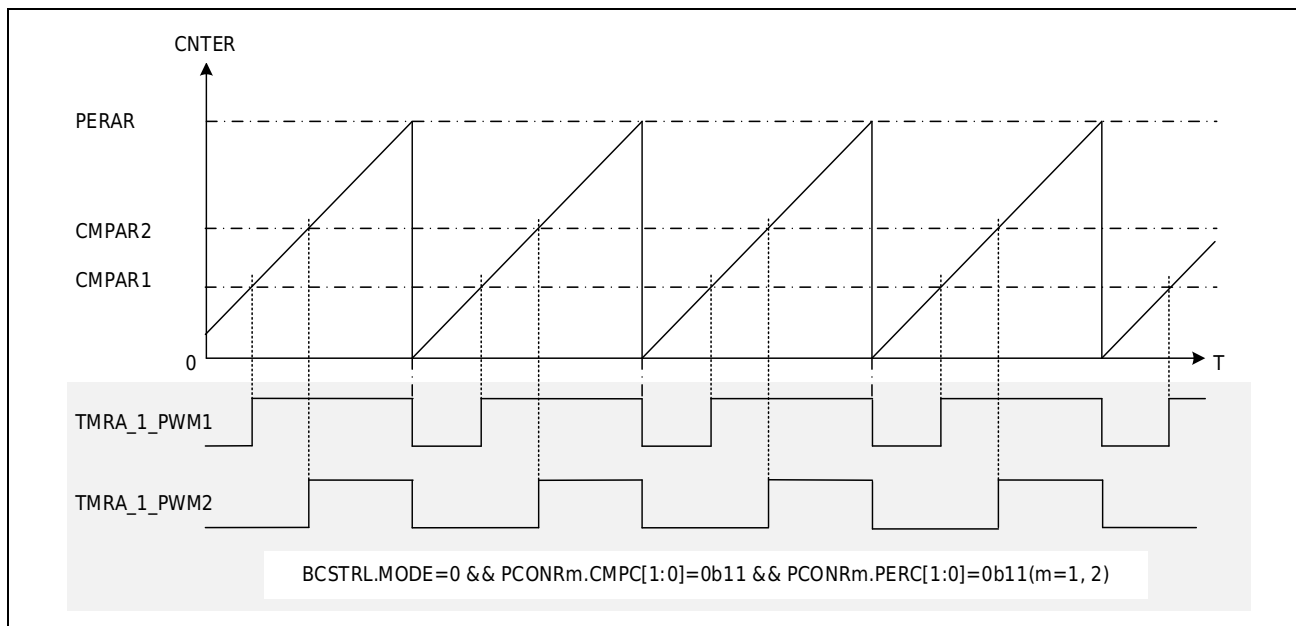


图 22-11 单边对齐 PWM 输出例

22.3.11.2 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 22-12 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。

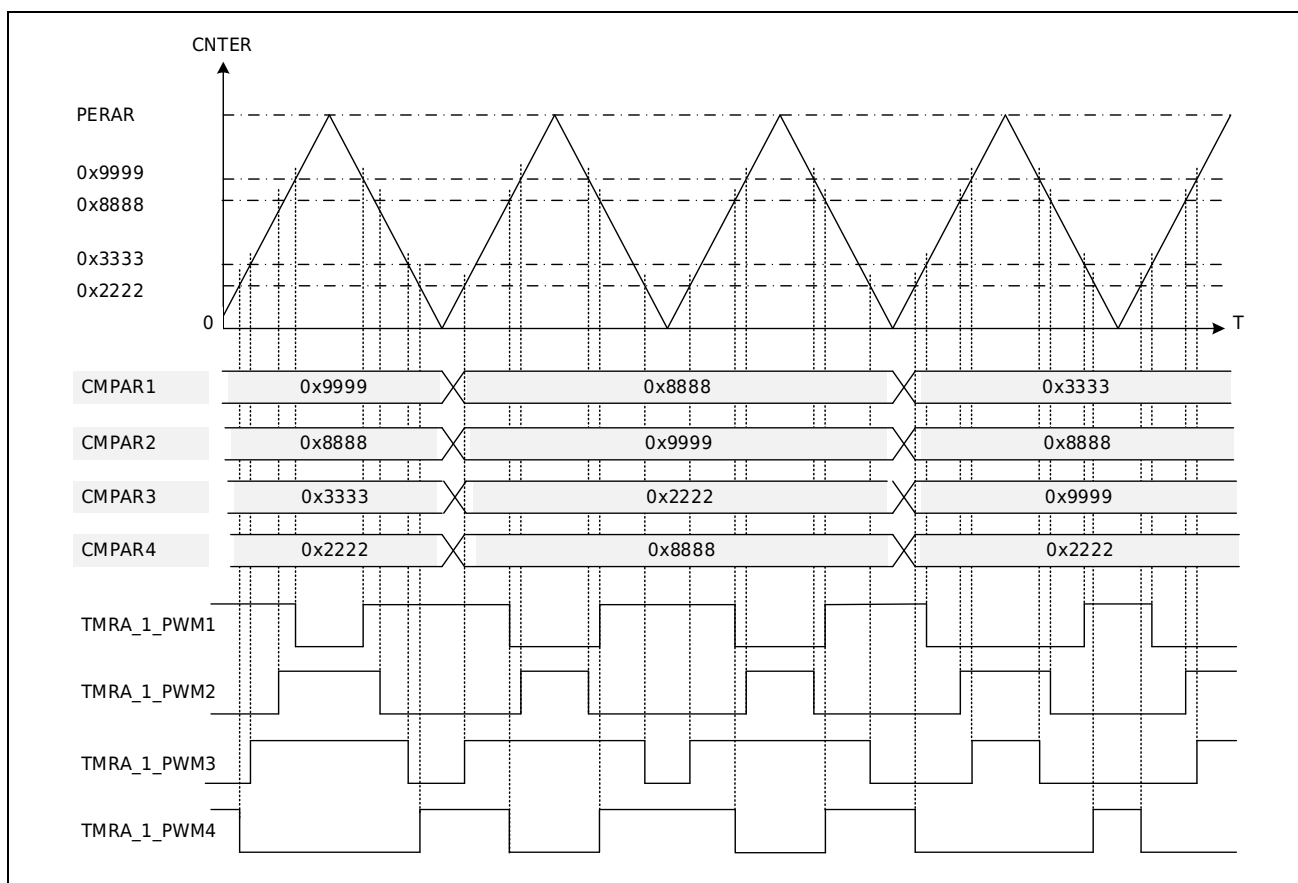


图 22-12 双边对称 PWM 输出例

22.3.12 正交编码计数

将 TMRA_<t>_CLKA 输入看作 AIN 输入、TMRA_<t>_CLKB 输入看作 BIN 输入、TMRA_<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），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 TMRA_<t>_CLKA 和 TMRA_<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、通过设定公转计数单元的硬件递加事件选择寄存器（HCUPR）实现公转定时器计数。

22.3.12.1 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 22-13 所示。

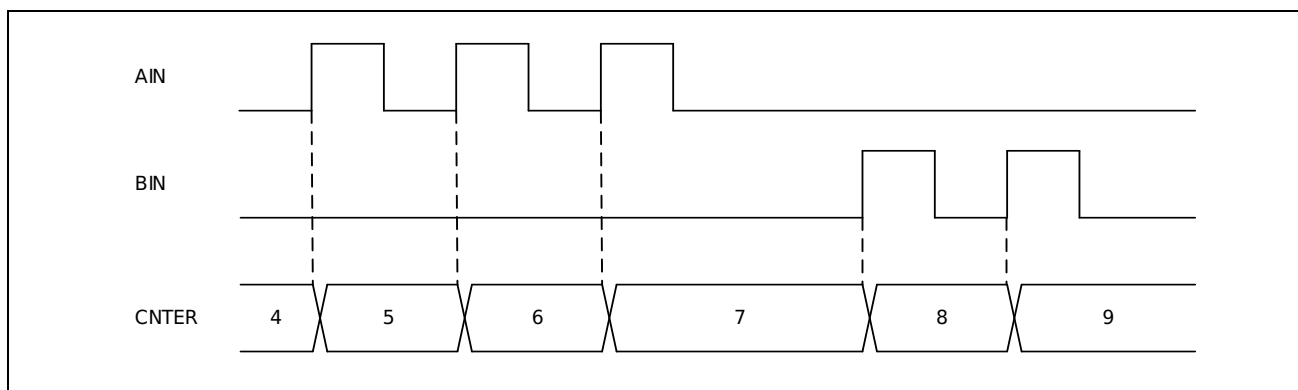


图 22-13 位置模式-基本计数

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 22-14~图 22-16 所示。

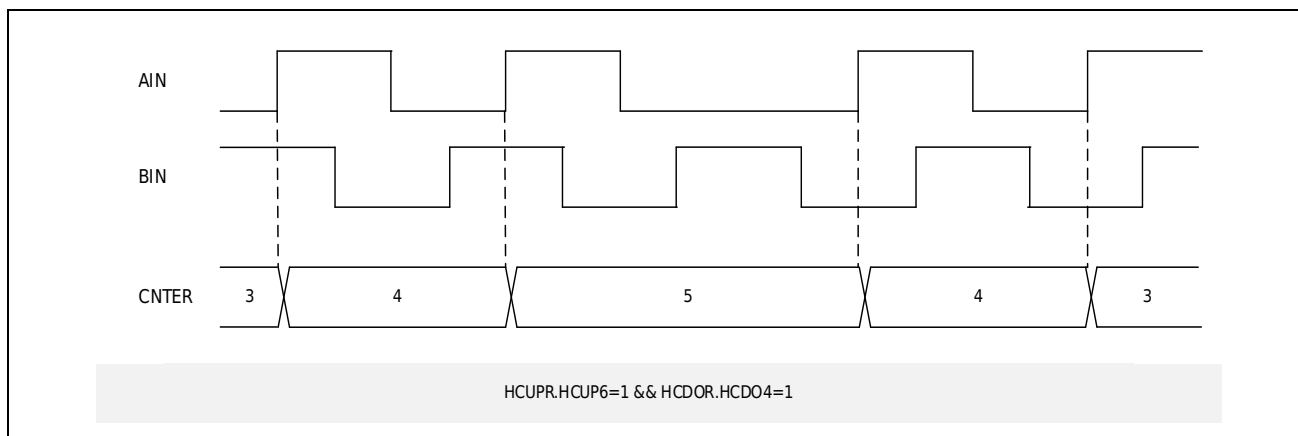


图 22-14 位置计数模式-相位差计数（1 倍计数）

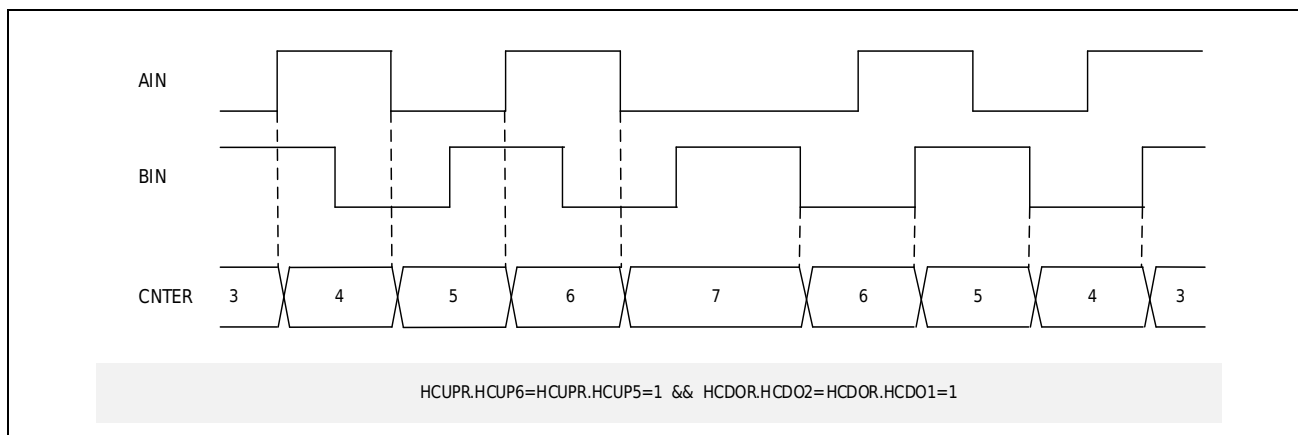


图 22-15 位置计数模式-相位差计数 (2 倍计数)

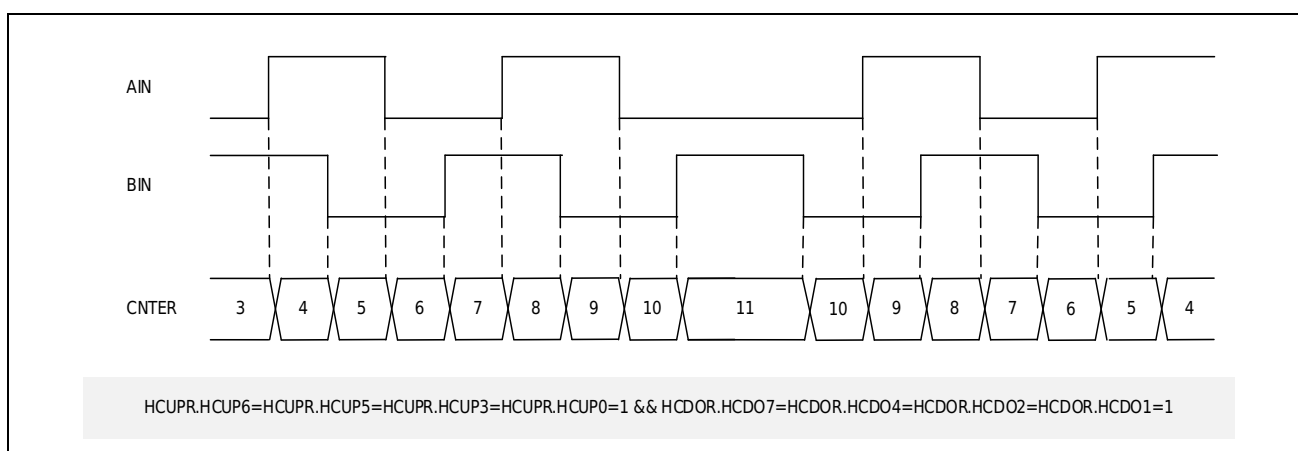


图 22-16 位置计数模式-相位差计数 (4 倍计数)

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 22-17 所示。

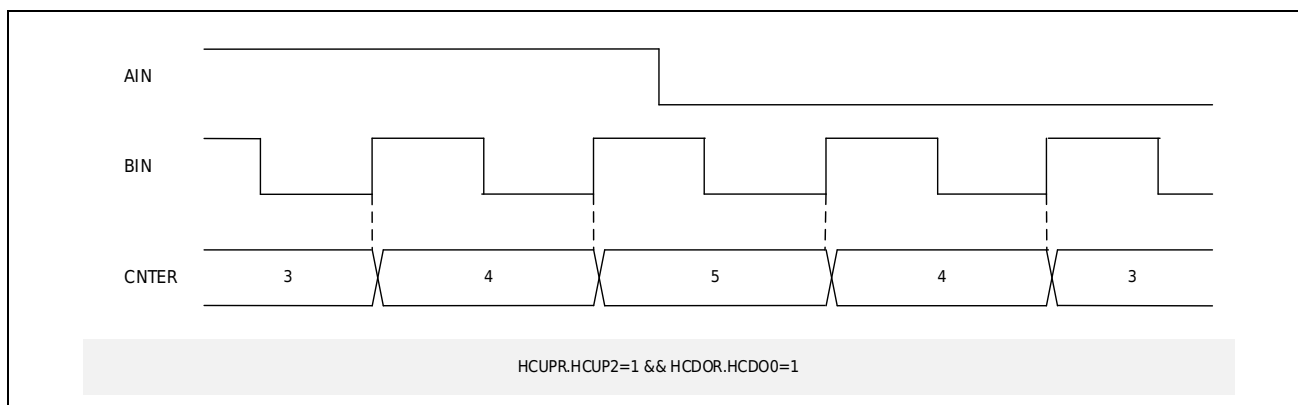


图 22-17 位置计数模式-方向计数

22.3.12.2 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 22-18 所示。

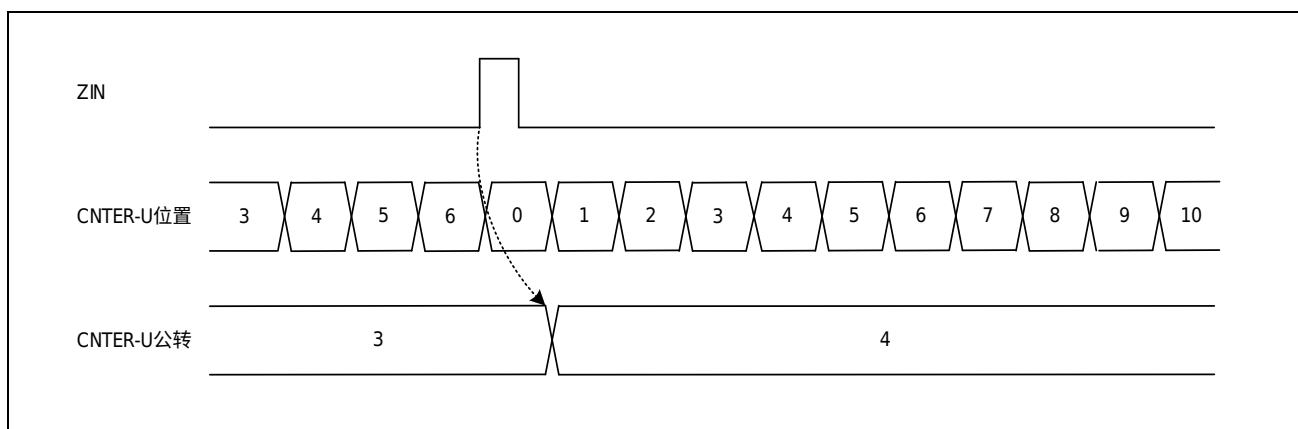


图 22-18 公转计数模式-Z 相计数

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 22-19 所示。

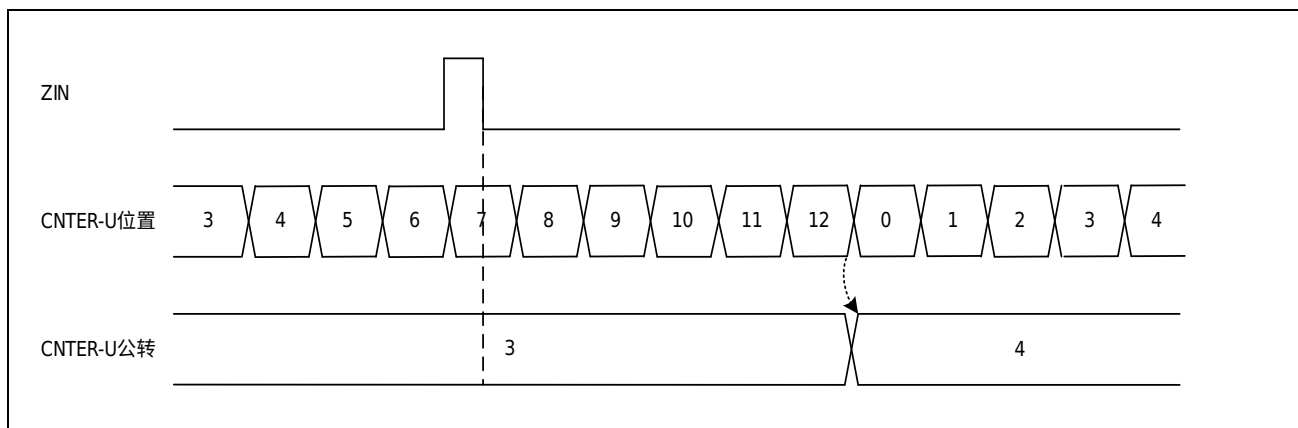


图 22-19 公转计数模式-位置溢出计数

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 22-20 所示。

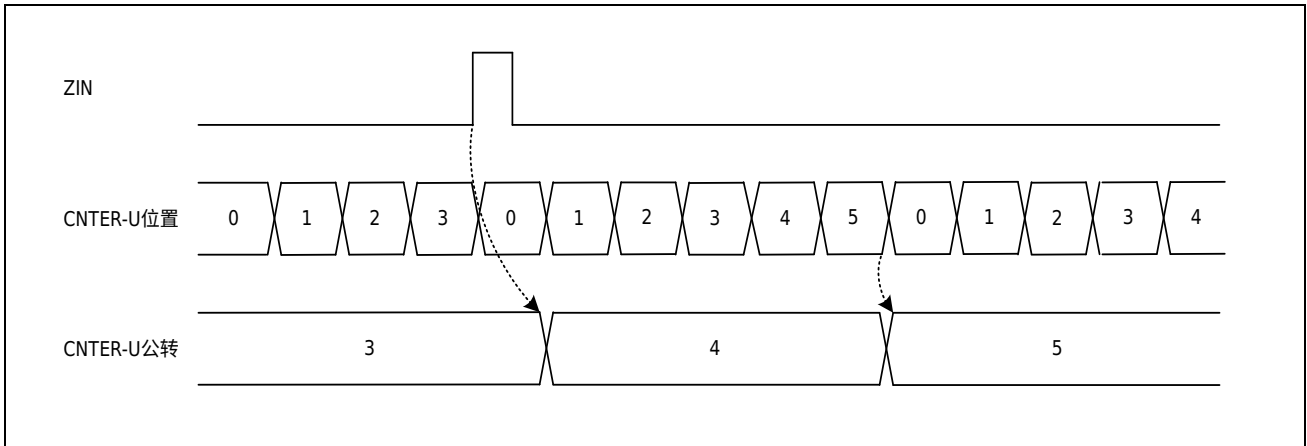


图 22-20 公转计数模式-混合计数

22.4 中断和事件

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。

22.4.1 比较匹配中断及事件

比较基准值寄存器 (CMPARm) 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.CMPFm) 会被置为 1。此时，若中断控制寄存器 (ICONR) 的对应位 (ICONR.ITENm) 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.ETENm) 设定为 1，则对应的事件请求 (TMRA_<t>_CMP) 会被触发 (m=1~4)。

捕获控制寄存器 (CCONRm) 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.ITENm) 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.ETENm) 设定为 1，则对应的事件请求 (TMRA_<t>_CMP) 会被触发 (m=1~4)。

每个单元内部的 8 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【中断控制器 (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。

22.4.2 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 UDF 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。

22.5 寄存器描述

表 22-4 TimerA 基地址

名称	基地址	描述
TimerA1	0x4003 A000	TimerA1基地址
TimerA2	0x4003 A400	TimerA2基地址
TimerA3	0x4003 A800	TimerA3基地址
TimerA4	0x4003 AC00	TimerA4基地址
TimerA5	0x4002 6000	TimerA5基地址

表 22-5 TimerA 寄存器列表

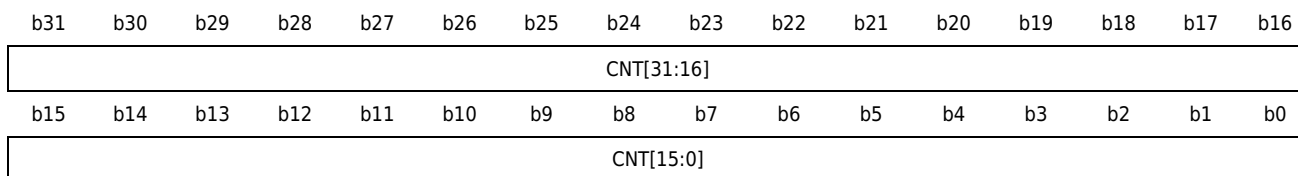
寄存器	描述	偏移地址	位宽	复位值
TMRA_CNTER	通用计数值寄存器	0x0000	32/16	0x0000 0000/ 0x0000
TMRA_PERAR	周期基准值寄存器	0x0004	32/16	0xFFFF FFFF/ 0xFFFF
TMRA_CMPAR1	比较基准值寄存器1	0x0040	32/16	0xFFFF FFFF/ 0xFFFF
TMRA_CMPAR2	比较基准值寄存器2	0x0044	32/16	0xFFFF FFFF/ 0xFFFF
TMRA_CMPAR3	比较基准值寄存器3	0x0048	32/16	0xFFFF FFFF/ 0xFFFF
TMRA_CMPAR4	比较基准值寄存器4	0x004C	32/16	0xFFFF FFFF/ 0xFFFF
TMRA_BCSTRL	控制状态寄存器L	0x0080	8	0x02
TMRA_BCSTRH	控制状态寄存器H	0x0081	8	0x00
TMRA_ICONR	中断控制寄存器	0x0090	16	0x0000
TMRA_ECONR	事件控制寄存器	0x0094	16	0x0000
TMRA_FCONR	滤波控制寄存器	0x0098	16	0x0000
TMRA_STFLR	状态标志寄存器	0x009C	16	0x0000
TMRA_BCONR1	缓存控制寄存器1	0x00C0	16	0x0000
TMRA_BCONR2	缓存控制寄存器2	0x00C8	16	0x0000
TMRA_BCONR3	缓存控制寄存器3	0x00D0	16	0x0000
TMRA_BCONR4	缓存控制寄存器4	0x00D8	16	0x0000
TMRA_CCONR1	捕获控制寄存器1	0x0100	16	0x0000
TMRA_CCONR2	捕获控制寄存器2	0x0104	16	0x0000
TMRA_CCONR3	捕获控制寄存器3	0x0108	16	0x0000
TMRA_CCONR4	捕获控制寄存器4	0x010C	16	0x0000
TMRA_PCONR1	端口控制寄存器1	0x0140	16	0x0000
TMRA_PCONR2	端口控制寄存器2	0x0144	16	0x0000
TMRA_PCONR3	端口控制寄存器3	0x0148	16	0x0000
TMRA_PCONR4	端口控制寄存器4	0x014C	16	0x0000
TMRA_HCONR	硬件触发事件选择寄存器	0x0084	16	0x0000
TMRA_HCUPR	硬件递加事件选择寄存器	0x0088	16	0x0000
TMRA_HCDOR	硬件递减事件选择寄存器	0x008C	16	0x0000

注意：

- 单元 1 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，各复位值与单元 2~单元 5 不一样。其中通用计数值寄存器的复位值为 $0x0000\ 0000$ ，周期基准值寄存器、比较基准值寄存器的复位值为 $0xFFFF\ FFFF$ 。

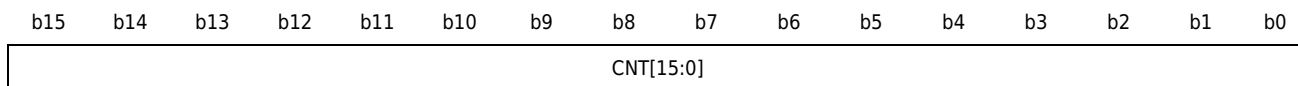
22.5.1 通用计数值寄存器 (TMRA_CNTER)

单元 1: 复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
			当前定时器的计数值	
			注:	
			1. 只能再定时器停止时写入;	
			2. 锯齿波向上计数模式, CNT初始值设置为0; 锯齿波向下计数模式, CNT初始值设置为pear;	
			3. 三角波模式计数CNT初始值设置为1;	
b31~b0	CNT[31:0]	计数值		RW

单元 2~5: 复位值: 0x0000



位/位域	标记	位名	功能	读写
			当前定时器的计数值	
			注:	
			1. 只能再定时器停止时写入;	
			2. 锯齿波向上计数模式, CNT初始值设置为0; 锯齿波向下计数模式, CNT初始值设置为pear;	
			3. 三角波模式计数CNT初始值设置为1;	
b15~b0	CNT[15:0]	计数值		RW

22.5.2 周期基准值寄存器 (TMRA_PERAR)

单元 1: 复位值: 0xFFFF FFFF

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/位域	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										RW		

单元 2~5: 复位值: 0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PER[15:0]															
位/位域	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										RW		

22.5.3 比较基准值寄存器 (TMRA_CMPARm, m=1~4)

单元 1: 复位值: 0xFFFF FFFF

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/位域	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										RW		

单元 2~5: 复位值: 0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMP[15:0]															
位/位域	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										RW		

22.5.4 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

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 注: 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 注: 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 注: 1. 该位在硬件停止条件有效时, 会自动变为0 2. 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:

1. 对于 TimerA 单元 1~单元 4, PCLK 为 PCLK0; 对于 TimerA 单元 5, PCLK 为 PCLK1。
2. 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。

22.5.5 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

22.5.6 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
Res												IT EN4	IT EN3	IT EN2	IT EN1

位/位域	标记	位名	功能	读写
b15~b8	Res	保留位	读出时为“0”, 写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW

22.5.7 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
Res												ET EN4	ET EN3	ET EN2	ET EN1
位/位域	标记	位名	功能											读写	
b15~b4	Res	保留位	读出时为“0”，写入时写“0”											RW	
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											RW	
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											RW	
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											RW	
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											RW	

22.5.8 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
Res	NOFI CKCB[1:0]	NOFI ENCB	Res	NOFI CKCA[1:0]	NOFI ENCA	Res					NOFI CKTG[1:0]	NOFI ENTG			

位/位域	标记	位名	功能	读写
b15	Res	保留位	读出时为“0”，写入时写“0”	RW
b14~b13	NOFICKCB[1:0]	滤波采样基准时钟选择CB	00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64	RW
b12	NOFIENCB	捕获输入端口滤波CB	0: TMRA_<t>_CLKB端口输入滤波功能无效 1: TMRA_<t>_CLKB端口输入滤波功能使能	RW
b11	Res	保留位	读出时为“0”，写入时写“0”	RW
b10~b9	NOFICKCA[1:0]	滤波采样基准时钟选择CA	00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64	RW
b8	NOFIENCA	捕获输入端口滤波CA	0: TMRA_<t>_CLKA端口输入滤波功能无效 1: TMRA_<t>_CLKA端口输入滤波功能使能	RW
b7~b3	Res	保留位	读出时为“0”，写入时写“0”	RW
b2~b1	NOFICKTG[1:0]	滤波采样基准时钟选择TG	00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64	RW
b0	NOFIENTG	捕获输入端口滤波TG	0: TMRA_<t>_TRIG输入端口滤波功能无效 1: TMRA_<t>_TRIG输入端口滤波功能使能	RW

注意:

对于 TimerA 单元 1~单元 4, PCLK 为 PCLK0; 对于 TimerA 单元 5, PCLK 为 PCLK1。

22.5.9 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
Res				ICPF4	ICPF3	ICPF2	ICPF1	Res				CMPF4	CMPF3	CMPF2	CMPF1

位/位域	标记	位名	功能	读写
b15~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW
b7~b4	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW

注意:

- 该寄存器中的标志位只能写入 0，无法写入 1。

22.5.10 缓存控制寄存器 (TMRA_BCONRm, m=1~4)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												BSEN	BSE1	BSE0	BEN

位/位域	标记	位名	功能	读写
b15~b4	Res	保留位	读出时为“0”，写入时写“0”	RW
b3	BSEN	缓存特殊使能	0: 当对称单元的COUNTER寄存器进行读操作时，本单元不进行缓存值传送 1: 当对称单元的COUNTER寄存器进行读操作时，本单元进行缓存值传送，即： CMPARm -> CMPARn (m=2、4、6、8, n=1、3、5、7)	RW
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、6、8, n=1、3、5、7) 注： 仅当BEN为1时，该位的配置有效。	RW
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、6、8, n=1、3、5、7) 注： 仅当BEN为1时，该位的配置有效。	RW
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、5、7)	RW

注意：

- 请不要将 BEN 与 BSEN 同时设定为 1。
- BEN 为 1 且 BSE1 和 BSE0 都为 0 时，硬件计数模式或软件计数的锯齿波模式，计数到上溢点或下溢点或发生硬件清零计数器动作时，缓存值传送。

22.5.11 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
Res	NOFI CKCP[1:0]	NOFI ENCP	HICP 6	HICP 5	HICP 4	HICP 3	Res	HICP 2	HICP 1	HICP 0	Res			CAPM D	

位/位域	标记	位名	功能	读写
b15	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b12	NOFIENCP	捕获输入端口滤波 CP	0: TMRA_<t>_PWMm端口输入滤波功能无效 1: TMRA_<t>_PWMm端口输入滤波功能使能 (m=1~4)	RW
b11	HICP6	捕获输入条件使能6	0: TMRA_<t>_XOR信号采样到下降沿时，不发生捕获输入动作 1: TMRA_<t>_XOR信号采样到下降沿时，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	RW
b10	HICP5	捕获输入条件使能5	0: TMRA_<t>_XOR信号采样到上升沿时，不发生捕获输入动作 1: TMRA_<t>_XOR信号采样到上升沿时，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	RW
b9	HICP4	捕获输入条件使能4	0: TMRA_<t>_TRIG端口输入采样到下降沿时，不发生捕获输入动作 1: TMRA_<t>_TRIG端口输入采样到下降沿时，产生捕获输入动作 注: 该位只有CCONR3寄存器有效。即，该位有效后且对应事件发生时，在CCONR3.CAPMD=1的条件下、当前计数器值被捕获保存到CMPAR3，且STFLR.CMPF3置位	RW
b8	HICP3	捕获输入条件使能3	0: TMRA_<t>_TRIG端口输入采样到上升沿时，不发生捕获输入动作 1: TMRA_<t>_TRIG端口输入采样到上升沿时，产生捕获输入动作 注: 该位只有CCONR3寄存器有效。即，该位有效后且对应事件发生时，在CCONR3.CAPMD=1的条件下、当前计数器值被捕获保存到CMPAR3，且STFLR.CMPF3置位	RW
b7	Res	保留位	读出时为“0”，写入时写“0”	RW
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: 具体请参考寄存器描述章节的注意事项说明	RW
b5	HICP1	捕获输入条件使能1	0: TMRA_<t>_PWMm端口输入采样到下降沿时，不发生捕获输入动作 1: TMRA_<t>_PWMm端口输入采样到下降沿时，产生捕获输入动作	RW

			(m=1~4)	
b4	HICP0	捕获输入条件使能	0: TMRA_<t>_PWWm端口输入采样到上升沿时, 不发生捕获输入动作 1: TMRA_<t>_PWWm端口输入采样到上升沿时, 产生捕获输入动作 (m=1~4)	RW
b3~b1	Res	保留位	读出时为“0”, 写入时写“0”	RW
b0	CAPMD	功能模式选择	0: 比较输出功能 1: 捕获输入功能	RW

注意:

1. 对于 *TimerA* 单元 1~单元 4, *PCLK* 为 *PCLK0*; 对于 *TimerA* 单元 5, *PCLK* 为 *PCLK1*。
2. *TMRA_<t>_XOR* 信号为各个单元的正交输入 *AIN* 与 *BIN* 异或后得到的信号。

22.5.12 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
Res		OUTEN		Res		FORC[1:0]		PERC[1:0]		CMPC[1:0]		STPC[1:0]		STAC[1:0]	
位/位域	标记	位名	功能		读写										
b15~b13	Res	保留位	读出时为“0”，写入时写“0”		RW										
b12	OUTEN	输出使能	0: PWM输出功能时的TMRA_<t>_PWMm端口输出无效 1: PWM输出功能时的TMRA_<t>_PWMm端口输出有效 (m=1~4)		RW										
b11~b10	Res	保留位	读出时为“0”，写入时写“0”		RW										
b9~b8	FORC[1:0]	强制端口状态设定	0x: 设定无效 10: 下周期开始, TMRA_<t>_PWMm端口输出设定为低电平 11: 下周期开始, TMRA_<t>_PWMm端口输出设定为高电平 (m=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 2. 该寄存器位可用于实现PWM输出占空比0%或100%的控制		RW										
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, TMRA_<t>_PWMm端口输出设定为低电平 01: 计数值与PERAR相等时, TMRA_<t>_PWMm端口输出设定为高电平 10: 计数值与PERAR相等时, TMRA_<t>_PWMm端口输出保持先前状态 11: 计数值与PERAR相等时, TMRA_<t>_PWMm端口输出设定为反转电平 (n=1~4)		RW										
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, TMRA_<t>_PWMm端口输出设定为低电平 01: 计数值与CMPARn相等时, TMRA_<t>_PWMm端口输出设定为高电平 10: 计数值与CMPARn相等时, TMRA_<t>_PWMm端口输出保持先前状态 11: 计数值与CMPARn相等时, TMRA_<t>_PWMm端口输出设定为反转电平 (m=1~4)		RW										
b3~b2	STPC[1:0]	计数停止时端口状态设定	00: 计数停止时, TMRA_<t>_PWMm端口输出设定为低电平 01: 计数停止时, TMRA_<t>_PWMm端口输出设定为高电平 10: 计数停止时, TMRA_<t>_PWMm端口输出保持先前状态 11: 计数停止时, TMRA_<t>_PWMm端口输出保持先前状态 (m=1~4)		RW										
b1~b0	STAC[1:0]	计数开始时端口状态设定	00: 计数开始时, TMRA_<t>_PWMm端口输出设定为低电平 01: 计数开始时, TMRA_<t>_PWMm端口输出设定为高电平 10: 计数开始时, TMRA_<t>_PWMm端口输出保持先前状态		RW										

11: 计数开始时, TMRA_<t>_PWWm端口输出保持先前状态
(m=1~4)

注:

该位设定只在不分频 (BCSTR.LCKDIV=0b0000) 的情况下有效, 其它分频请设定为0b10或0b11

22.5.13 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
HCLE 6	HCLE 5	HCLE 4	HCLE 3	Res	HCLE 2	HCLE 1	HCLE 0	Res	HSTP 2	HSTP 1	HSTP 0	Res	HSTA 2	HSTA 1	HSTA 0

位/位域	标记	位名	功能	读写
b15	HCLE6	硬件清零条件6	条件: TMRA_<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	RW
b14	HCLE5	硬件清零条件5	条件: TMRA_<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	RW
b13	HCLE4	硬件清零条件4	条件: 对称单元的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	RW
b12	HCLE3	硬件清零条件3	条件: 对称单元的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	RW
b11	Res	保留位	读出时为“0”, 写入时写“0”	RW
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: 具体请参考寄存器描述章节的注意事项说明	RW
b9	HCLE1	硬件清零条件1	条件: TMRA_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	RW
b8	HCLE0	硬件清零条件0	条件: TMRA_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	RW
b7	Res	保留位	读出时为“0”, 写入时写“0”	RW
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: 具体请参考寄存器描述章节的注意事项说明	RW
b5	HSTP1	硬件停止条件1	条件: TMRA_<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	RW
b4	HSTP0	硬件停止条件0	条件: TMRA_<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	RW
b3	Res	保留位	读出时为“0”, 写入时写“0”	RW
b2	HSTA2	硬件启动条件2	条件: TMRA_TRGSEL寄存器中指定的事件发生 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效 注:	RW

具体请参考寄存器描述章节的注意事项说明			
b1	HSTA1	硬件启动条件1	RW 条件： 1) 同步启动功能无效时， TMRA_<t>_TRIG端口输入采样到下降沿 2) 同步启动功能有效时，对称单元的TMRA_<t>_TRIG端口输入采样到下降沿（仅通道2和通道4支持该功能） 0: 条件匹配时，硬件启动无效 1: 条件匹配时，硬件启动有效
b0	HSTA0	硬件启动条件0	RW 条件： 1) 同步启动功能无效时， TMRA_<t>_TRIG端口输入采样到上升沿 2) 同步启动功能有效时，对称单元的TMRA_<t>_TRIG端口输入采样到上升沿（仅通道2和通道4支持该功能） 0: 条件匹配时，硬件启动无效 1: 条件匹配时，硬件启动有效

22.5.14 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
Res			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/位域	标记	位名	功能										读写		
b15~b13	Res	保留位	读出时为“0”，写入时写“0”										RW		
b12	HCUP12	硬件递加条件12	条件: 对称单元发生计数下溢0: 条件匹配时, 本单元硬件递加无效 1: 条件匹配时, 本单元硬件递加有效										RW		
b11	HCUP11	硬件递加条件11	条件: 对称单元发生计数上溢0: 条件匹配时, 本单元硬件递加无效 1: 条件匹配时, 本单元硬件递加有效										RW		
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: 具体请参考寄存器描述章节的注意事项说明										RW		
b9	HCUP9	硬件递加条件9	条件: TMRA_<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										RW		
b8	HCUP8	硬件递加条件8	条件: TMRA_<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										RW		
b7	HCUP7	硬件递加条件7	条件: TMRA_<t>_CLKB端口为高电平时, TMRA_<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										RW		
b6	HCUP6	硬件递加条件6	条件: TMRA_<t>_CLKB端口为高电平时, TMRA_<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										RW		
b5	HCUP5	硬件递加条件5	条件: TMRA_<t>_CLKB端口为低电平时, TMRA_<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										RW		
b4	HCUP4	硬件递加条件4	条件: TMRA_<t>_CLKB端口为低电平时, TMRA_<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										RW		
b3	HCUP3	硬件递加条件3	条件: TMRA_<t>_CLKA端口为高电平时, TMRA_<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										RW		
b2	HCUP2	硬件递加条件2	条件: TMRA_<t>_CLKA端口为高电平时, TMRA_<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										RW		

b1	HCUP1	硬件递加条件1	<p>条件：TMRA_<t>_CLKA端口为低电平时，TMRA_<t>_CLKB端口上采样到下降沿</p> <p>0：条件匹配时，硬件递加无效</p> <p>1：条件匹配时，硬件递加有效</p>	RW
b0	HCUP0	硬件递加条件0	<p>条件：TMRA_<t>_CLKA端口为低电平时，TMRA_<t>_CLKB端口上采样到上升沿</p> <p>0：条件匹配时，硬件递加无效</p> <p>1：条件匹配时，硬件递加有效</p>	RW

22.5.15 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
Res			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/位域	标记	位名	功能	读写
b15~b13	Res	保留位	读出时为“0”，写入时写“0”	RW
b12	HCD012	硬件递减条件12	条件：对称单元发生计数下溢0：条件匹配时，本单元硬件递减无效 1：条件匹配时，本单元硬件递减有效	RW
b11	HCD011	硬件递减条件11	条件：对称单元发生计数上溢0：条件匹配时，本单元硬件递减无效 1：条件匹配时，本单元硬件递减有效	RW
b10	HCD010	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：具体请参考寄存器描述章节的注意事项说明	RW
b9	HCD09	硬件递减条件9	条件：TMRA_<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	RW
b8	HCD08	硬件递减条件8	条件：TMRA_<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	RW
b7	HCD07	硬件递减条件7	条件：TMRA_<t>_CLKB端口为高电平时，TMRA_<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	RW
b6	HCD06	硬件递减条件6	条件：TMRA_<t>_CLKB端口为高电平时，TMRA_<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	RW
b5	HCD05	硬件递减条件5	条件：TMRA_<t>_CLKB端口为低电平时，TMRA_<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	RW
b4	HCD04	硬件递减条件4	条件：TMRA_<t>_CLKB端口为低电平时，TMRA_<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	RW
b3	HCD03	硬件递减条件3	条件：TMRA_<t>_CLKA端口为高电平时，TMRA_<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	RW
b2	HCD02	硬件递减条件2	条件：TMRA_<t>_CLKA端口为高电平时，TMRA_<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	RW

b1	HCDO1	硬件递减条件1	<p>条件：TMRA_<t>_CLKA端口为低电平时，TMRA_<t>_CLKB端口上采样到下降沿</p> <p>0：条件匹配时，硬件递减无效</p> <p>1：条件匹配时，硬件递减有效</p>	RW
b0	HCDO0	硬件递减条件0	<p>条件：TMRA_<t>_CLKA端口为低电平时，TMRA_<t>_CLKB端口上采样到上升沿</p> <p>0：条件匹配时，硬件递减无效</p> <p>1：条件匹配时，硬件递减有效</p>	RW

23 通用定时器 (Timer0)

23.1 概述

通用定时器 0 (Timer0) 是一个可以实现同步计数和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与计数溢出事件。该事件可以触发中断，也可以用于触发其他模块动作。本系列产品中搭载 2 个单元的 Timer0。

23.2 功能说明

23.2.1 Timer0 框图

Timer0 的基本框图如图 23-1 所示。

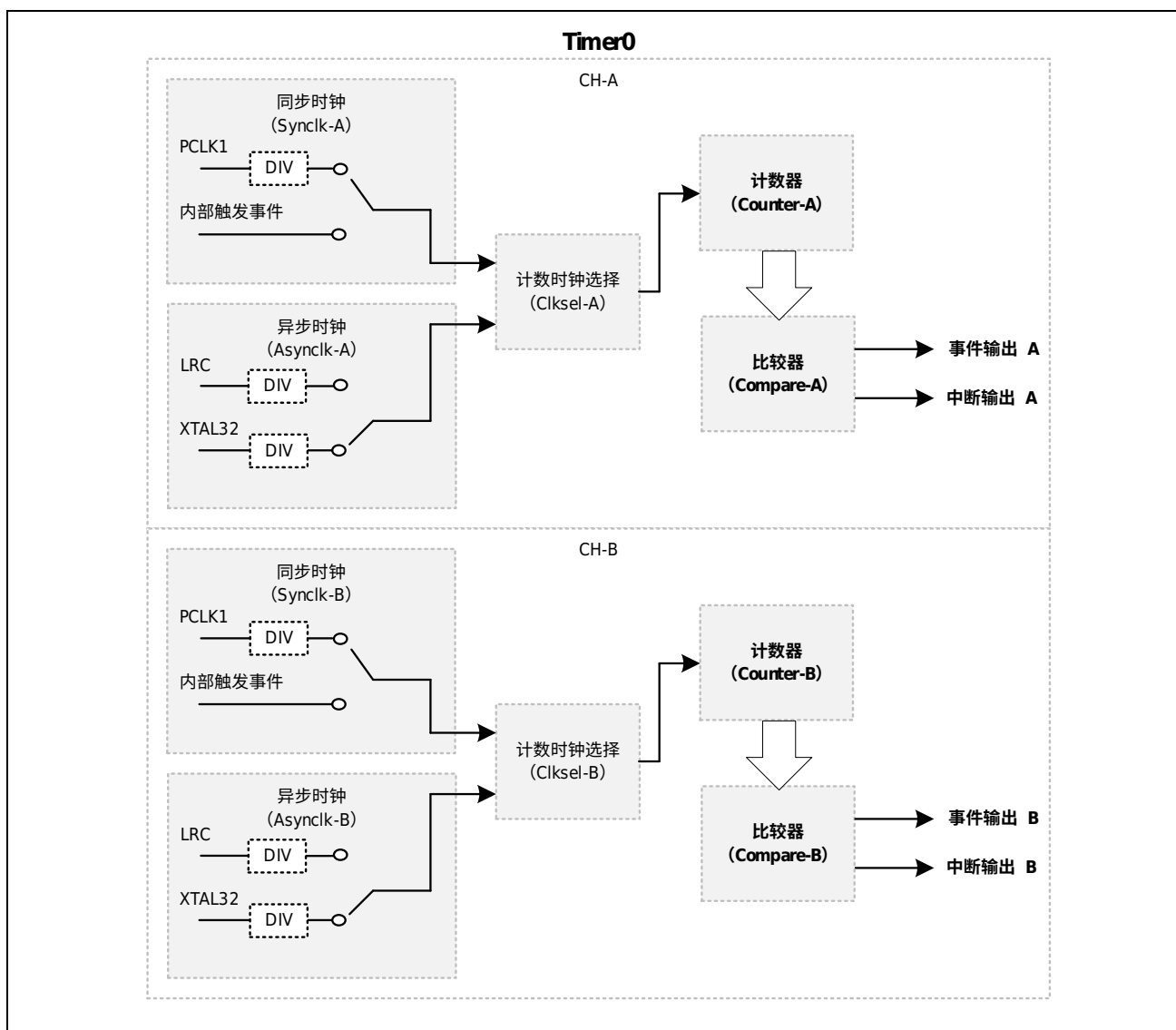


图 23-1 Timer0 基本框图

23.2.2 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。

23.2.2.1 同步计数时钟源

同步计数方式时 (BCONR.SYNSA=0)，时钟源可以有以下几种选择 (BCONR.SYNCLKA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=0 & BCONR.CKDIV A [3:0]设定)
- b) 内部硬件触发事件输入作为同步计数时钟 (BCONR.SYNCLK A=1)

23.2.2.2 异步计数时钟源

异步计数方式时 (BCONR.SYNS A=1)，时钟源可以有以下几种选择 (BCONR.ASYNCLK A设定选择)：

- 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=0 & BCONR.CKDIV A [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=1 & BCONR.CKDIV A [3:0]设定)

23.2.3 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 23-2 所示。

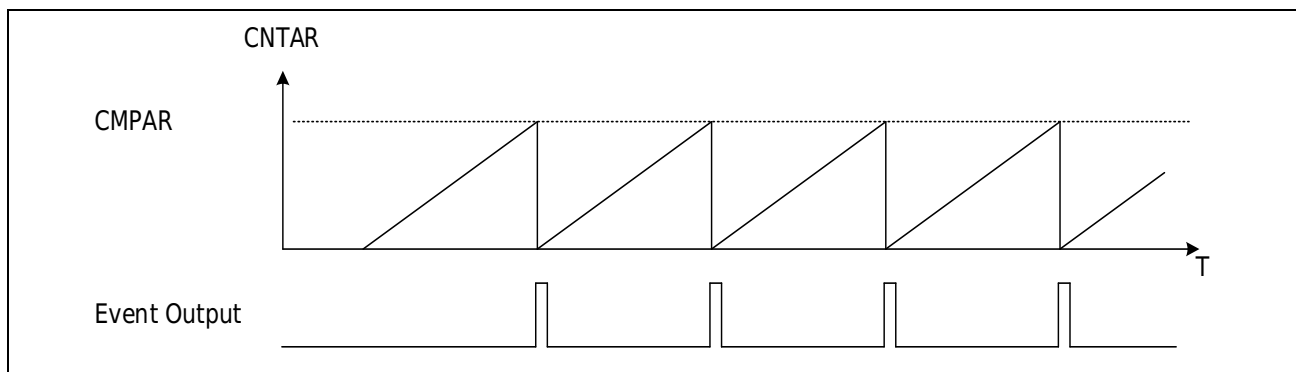


图 23-2 Timer0 计数时序图

23.2.4 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（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（TMR0_TRGSEL）中输入对应的编号来实现，寄存器内容请参考【自动运行系统（AOS）】章节，具体的事件对应关系请参考【中断控制器（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 AOS 功能控制位使能。

23.2.5 软件同步启动

产品配置了一个计数同步使能寄存器（TMR_SYNENR），用于同步启动产品所搭载的所有定时器（包括 Timer0, Timer4, Timer6, TimerA）。对 TMR_SYNENR 寄存器的多个启动位进行置 1，可以实现多个定时器同时开始计数。启动位的具体分配请参考计数同步使能寄存器（TMR_SYNENR）。

注意：

1. 使用同步启动功能时，请确保所有同步启动定时器的 PCLK 时钟频率相同。
2. 使用同步启动功能时，请确保所有同步启动定时器均是同步计数方式。
3. 当计数时钟选择 PCLK 的分频时钟时，Timer6 的启动时间和其他 Timer 不同步，请勿混用。

23.3 中断和事件

23.3.1 中断

一个 Timer0 含有 4 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、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位分别会被置为 1。此时若设定基本控制寄存器（BCONR）的 BCONR.CMENA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.CMENA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。

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位会被置为 1。若设定基本控制寄存器（BCONR）的 BCONR.OVENA位使能中断，则对应的中断请求（TMR0_m_OVFn, m=1、2；n=A、B）也会被触发。

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停止模式时对系统进行唤醒，具体请参考【中断控制器（INTC）】章节。

23.3.2 事件

一个 Timer0 含有 4 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、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，会分别产生相应的事件请求（TMR0_m_CMPn 或 TMR0_m_OVFn，m=1、2；n=A、B）输出信号，可以用于选择触发其它模块。

23.4 寄存器描述

表 23-1 Timer0/ TMR_SYENR 基地址

名称	基地址	描述
Timer0_1	0x4002 4000	Timer0_1基地址
Timer0_2	0x4002 4400	Timer0_2基地址
TMR_SYENR	0x4005 5400	TMR_SYENR基地址

表 23-2 Timer0 寄存器列表

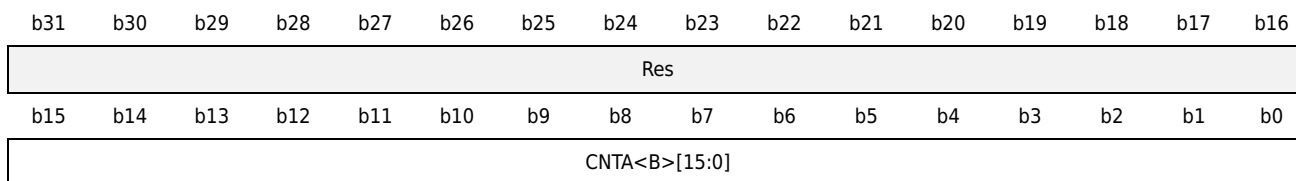
寄存器	描述	偏移地址	位宽	复位值
TMR0_CNTAR	计数值寄存器	0x0000	32	0x0000 0000
TMR0_CNTBR	计数值寄存器	0x0004	32	0x0000 0000
TMR0_CMPAR	基准值寄存器	0x0008	32	0x0000 FFFF
TMR0_CMPBR	基准值寄存器	0x000C	32	0x0000 FFFF
TMR0_BCONR	基本控制寄存器	0x0010	32	0x0000 0000
TMR0_STFLR	状态标志寄存器	0x0014	32	0x0100 0100

表 23-3 TMR_SYENR 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
TMR_SYENR	计数同步使能寄存器	0x14	32	0x0000 0000

23.4.1 计数值寄存器 (TMRO_CNTMR, m=A~B)

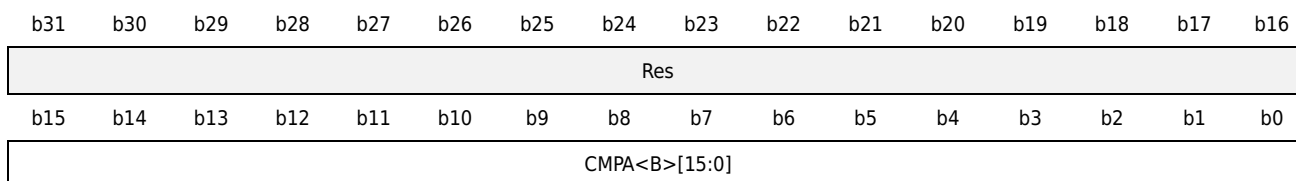
复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”，写入时写“0”	R
b15~b0	CNTA[15:0]	计数值	当前定时器的计数值 注: 在计数中时，不能向该寄存器中写入值	RW

23.4.2 基准值寄存器 (TMRO_CMPmR, m=A~B)

复位值: 0x0000 FFFF



位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”，写入时写“0”	R
b15~b0	CMPA[15:0]	基准值	设定计数基准值，产生比较匹配事件	RW

23.4.3 基本控制寄存器 (TMRO_BCONR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
HICP B	HCLE B	HSTP B	HSTA B	Res	ASYN CLKB	SYN CLKB	SYN SB	CKDIV B[3:0]			OV ENB	CM ENB	CAP MDB	CST B	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HICP A	HCLE A	HSTP A	HSTA A	Res	ASYN CLKA	SYN CLKA	SYN SA	CKDIV A[3:0]			OV ENA	CM ENA	CAP MDA	CST A	

位/位域	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	RW
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	RW
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	RW
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	RW
b27	Res	保留位	读出时为“0”, 写入时写“0”	RW
b26	ASYNCLKB	通道B异步计数时钟源选择	0: LRC 1: XTAL32 注: 当使能UART对应通道的超时功能时, 时钟源不再是XTAL32 (硬件上可以不存在XTAL32晶振), 而是UART波特率生成器产生的时钟tick。	RW
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	RW
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	RW
b23~b20	CKDIVB[3:0]	通道B计数时钟分频选择	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 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 其他: 禁止设定 注:	RW

			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	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	RW
b18	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	RW
b17	CAPMDB	功能模式选择B	0: 比较输出功能 1: 捕获输入功能	RW
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	RW
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	RW
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	RW
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	RW
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	RW
b11	Res	保留位	读出时为“0”, 写入时写“0”	RW
b10	ASYNCLKA	通道A异步计数时钟源选择	0: LRC 1: XTAL32 注: 当使能UART对应通道的超时功能时, 时钟源不再是XTAL32 (硬件上可以不存在XTAL32晶振), 而是UART波特率生成器产生的时钟tick。	RW
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	RW
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	RW
b7~b4	CKDIVA[3:0]	通道A计数时钟分频选择	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	RW

			1010: 时钟源/1024 其他: 禁止设定 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	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	RW
b2	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	RW
b1	CAPMDA	功能模式选择A	0: 比较输出功能 1: 捕获输入功能	RW
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 注: 该位在硬件触发停止条件有效时, 会自动变为0	RW

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。

23.4.4 状态标志寄存器 (TMRO_STFLR)

复位值: 0x0100 0100

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res							SYDB	Res					ICPB	OVFB	CMFB
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							SYDA	Res					ICPA	OVFA	CMFA

位/位域	标记	位名	功能	读写
b31~b25	Res	保留位	读出时为“0”，写入时写“0”	RW
b24	SYDB	同步完成标志B	0: 异步计数下B通道寄存器正在同步 1: 异步计数下B通道寄存器同步完成	R
b23~b19	Res	保留位	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	RW
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	RW
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	RW
b15~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
b8	SYDA	同步完成标志A	0: 异步计数下A通道寄存器正在同步 1: 异步计数下A通道寄存器同步完成	R
b7~b3	Res	保留位	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	RW
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	RW
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	RW

注意:

- 该寄存器中的标志位只能写入 0，无法写入 1。

23.4.5 计数同步使能寄存器 (TMR_SYNENR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res		HRP WMU 6	HRP WMU 5	HRP WMU 4	HRP WMU 3	HRP WMU 2	HRP WMU 1	Res			TMRA U5	TMRA U4	TMRA U3	TMRA U2	TMRA U1
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				TMR6 U4	TMR6 U3	TMR6 U2	TMR6 U1	Res			TMR4 U1	TMR0 U2B	TMR0 U2A	TMR0 U1B	TMR0 U1A

位/位域	标记	位名	功能	读写
b31~b30	Res	保留位	读出时为“0”，写入时写“0”	RW
b29	HRPWMU6	HRPWM单元6软件启动使能位	0: HRPWM 单元 6 软件启动无效 1: HRPWM 单元6软件启动有效且定时器开始计数	RW
b28	HRPWMU5	HRPWM单元5软件启动使能位	0: HRPWM 单元 5 软件启动无效 1: HRWPM单元5软件启动有效且定时器开始计数	RW
b27	HRPWMU4	HRPWM单元4软件启动使能位	0: HRPWM 单元 4 软件启动无效 1: HRWPM 单元4软件启动有效且定时器开始计数	RW
b26	HRPWMU3	HRPWM单元3软件启动使能位	0: HRPWM 单元 3 软件启动无效 1: HRWPM 单元3软件启动有效且定时器开始计数	RW
b25	HRPWMU2	HRPWM单元2软件启动使能位	0: HRPWM 单元 2 软件启动无效 1: HRWPM 单元2软件启动有效且定时器开始计数	RW
b24	HRPWMU1	HRPWM单元1软件启动使能位	0: HRPWM 单元 1 软件启动无效 1: HRWPM 单元1软件启动有效且定时器开始计数	RW
b23~b21	Res	保留位	读出时为“0”，写入时写“0”	RW
b20	TMRAU5	TMRA单元5软件启动使能位	0: TMRA 单元 5 软件启动无效 1: TMRA 单元 5 软件启动有效且定时器开始计数	RW
b19	TMRAU4	TMRA单元4软件启动使能位	0: TMRA 单元 4 软件启动无效 1: TMRA 单元 4 软件启动有效且定时器开始计数	RW
b18	TMRAU3	TMRA单元3软件启动使能位	0: TMRA 单元 3 软件启动无效 1: TMRA 单元 3 软件启动有效且定时器开始计数	RW
b17	TMRAU2	TMRA单元2软件启动使能位	0: TMRA 单元 2 软件启动无效 1: TMRA 单元 2 软件启动有效且定时器开始计数	RW
b16	TMRAU1	TMRA单元1软件启动使能位	0: TMRA 单元 1 软件启动无效 1: TMRA 单元 1 软件启动有效且定时器开始计数	RW
b15~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
b11	TMR6U4	TMR6单元2软件启动使能位	0: TMR6 单元 4 软件启动无效 1: TMR6 单元 4 软件启动有效且定时器开始计数	RW
b10	TMR6U3	TMR6单元3软件启动使能位	0: TMRA 单元 3 软件启动无效 1: TMRA 单元 3 软件启动有效且定时器开始计数	RW
b9	TMR6U2	TMR6单元2软件启动使能位	0: TMR6 单元 2 软件启动无效 1: TMR6 单元 2 软件启动有效且定时器开始计数	RW
b8	TMR6U1	TMR6单元1软件启动使能位	0: TMR6 单元 1 软件启动无效 1: TMR6 单元 1 软件启动有效且定时器开始计数	RW
b7~b5	Res	保留位	读出时为“0”，写入时写“0”	RW
b4	TMR4U1	TMR4单元1软件启动使能位	0: TMR4 单元 1 软件启动无效 1: TMR4 单元 1 软件启动有效且定时器开始计数	RW

b3	TMR0U2B	TMR0单元2B通道软件启动使能位	0: TMR0 单元 2 通道 B 软件启动无效 1: TMR0 单元 2 通道 B 软件启动有效且定时器开始计数 注: 请在同步计数的情况下使用该功能	RW
b2	TMR0U2A	TMR0单元2A通道软件启动使能位	0: TMR0 单元 2 通道 A 软件启动无效 1: TMR0 单元 2 通道 A 软件启动有效且定时器开始计数 注: 请在同步计数的情况下使用该功能	RW
b1	TMR0U1B	TMR0单元1B通道软件启动使能位	0: TMR0 单元 1 通道 B 软件启动无效 1: TMR0 单元 1 通道 B 软件启动有效且定时器开始计数 注: 请在同步计数的情况下使用该功能	RW
b0	TMR0U1A	TMR0单元1A通道软件启动使能位	0: TMR0 单元 1 通道 A 软件启动无效 1: TMR0 单元 1 通道 A 软件启动有效且定时器开始计数 注: 请在同步计数的情况下使用该功能	RW

注意:

- 请不要在本定时器异步计数的情况下使用计数同步功能。
- *TMR_SYNENR* 不受 *RMU_FRST* 寄存器的控制。

23.5 注意事项

- 1) 在异步计数动作时，需先设定 BCONR.ASYNCLKA 位选择异步时钟源，再设定 BCONR.SYNSA位选择异步计数方式，然后再启动 Timer0。
- 2) 在选择异步计数的情况下，修改计数值 (CNTAR)、基准值 (CMPAR)、启动位 (BCONR.CSTA)、状态位 (STFLR.CMFA)、重复捕获状态位 (STFLR.ICPA) 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。
- 3) 在选择异步计数的情况下，连续对计数值 (CNTAR)、基准值 (CMPAR)、启动位 (BCONR.CSTA)、状态位 (STFLR.CMFA)、重复捕获状态位 (STFLR.ICPA) 进行写动作时，需等待状态位 (STFLR.SYDA) 为 1 后再进行上述任一寄存器读写动作。
- 4) 在选择异步计数的情况下，请将 BCONR.SYNCLKA 设定为 0。
- 5) TMR0 作为事件输出时基准值设置需要大于 5。

24 实时时钟（RTC）

24.1 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。

24.2 主要特性

表 24-1 所示是其基本特性。

表 24-1 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	BCD码表示秒、分、时、日、周、月、年时间
	软件启动或停止
	12/24时制可选、闰年自动识别
	可编程闹钟
	分布式/均匀式补偿1Hz时钟输出
	时钟误差补偿功能
中断	时间戳功能
	周期中断 闹钟中断

24.3 功能说明

24.3.1 RTC 框图

RTC 的基本框图如图 24-1 所示。

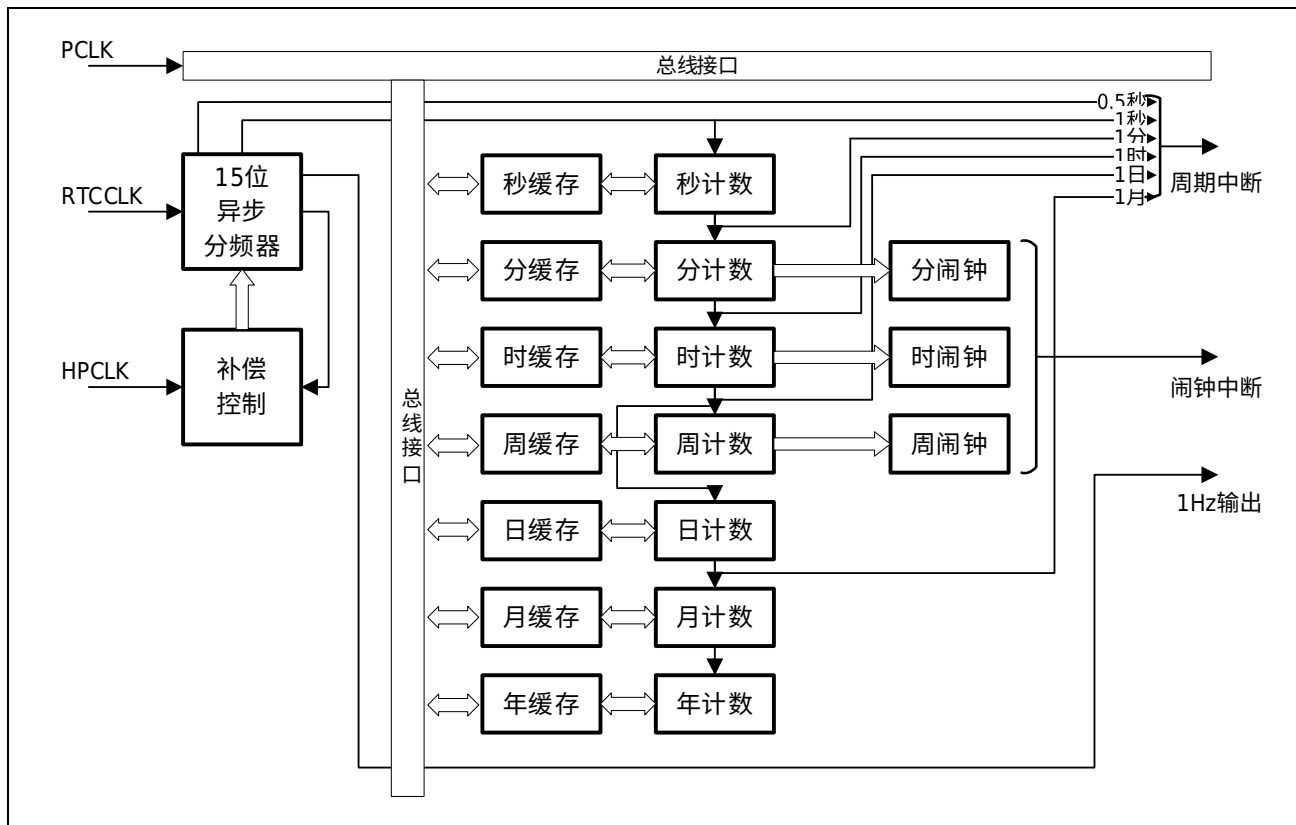


图 24-1 RTC 的基本框图

24.3.2 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。

24.3.3 RTC 计数开始设定

1. 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；
2. 设定 RTC_CR1.START=0，停止计数；
3. 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；
4. 设定 RTC_CR1，设定时制、周期、1Hz 时钟输出；

5. 设定秒，分，时，周，日，月，年的日历计数寄存器；
6. 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；
7. 清除寄存器 RTC_CR2 中的标志寄存器位，并使能中断；
8. 设定 RTC_CR1.START=1，计数开始。

24.3.4 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，请执行下列任何一种确认后再进行模式切换。

- 1) 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。
- 2) 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。

24.3.5 读出计数寄存器

1. 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；
2. 查询直到 RTC_CR2.RWEN=1；
3. 读出全部或者部分秒，分，时，周，日，月，年计数寄存器值；
4. 设定 RTC_CR2.RWREQ=0；
5. 查询直到 RTC_CR2.RWEN=0。

24.3.6 写入计数寄存器

1. 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；
2. 查询直到 RTC_CR2.RWEN=1；
3. 写入全部或者部分秒，分，时，周，日，月，年计数寄存器值；
4. 设定 RTC_CR2.RWREQ=0。注意，须在 1 秒内完成所有写操作；
5. 查询直到 RTC_CR2.RWEN=0。

24.3.7 闹钟设定

1. 设定 RTC_CR2.ALME=0，闹钟禁止；
2. 设定 RTC_CR2.ALMIE=1，闹钟中断许可；
3. 分闹钟 RTC_ALMMIN，时闹钟 RTC_ALMHOUR，周闹钟 RTC_ALMWEEK 设定；
4. 设定 RTC_CR2.ALME=1，闹钟许可；
5. 等待闹钟中断；
6. 闹钟发生，RTC_CR2.ALMF=1，进入闹钟中断处理。

24.3.8 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）。

24.3.9 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：

1. 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；
2. 设定 RTC_CR1.START=0，计数停止；
3. 1Hz 输出引脚设定；
4. RTC_CR1.ONEHZOE=1，时钟输出许可；
5. 设定 RTC_CR1.START=1，计数开始；
6. 等待 2 个计数周期以上；
7. 1Hz 输出开始。

分布式补偿 1Hz 输出设定如下：

1. 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；
2. 设定 RTC_CR1.START=0，计数停止；

3. 1Hz 输出引脚设定;
4. RTC_CR1.ONEHZOE=1, 时钟输出许可;
5. 时钟误差补偿寄存器 RTC_ERRCRL.COMP[7:0] 与 RTC_ERRCRH.COMP[8] 补偿数设定;
6. 时钟误差补偿寄存器 RTC_ERRCRH.COMPEN=1, 误差补偿有效;
7. 设定 RTC_CR1.START=1, 计数开始;
8. 等待 2 个计数周期以上;
9. 1Hz 输出开始。

均匀式补偿 1Hz 输出设定如下:

1. 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;
2. 设定 RTC_CR1.START=0, 计数停止;
3. RTC 输出引脚设定;
4. RTC_CR1.ONEHZOE=1, 时钟输出许可;
5. RTC_CR1.ONEHZSEL=1, 选择输出均匀式补偿 1Hz 时钟;
6. 时钟误差补偿寄存器 RTC_ERRCRL.COMP[7:0] 与 RTC_ERRCRH.COMP[8] 补偿数设定;
7. 时钟误差补偿寄存器 RTC_ERRCRH.COMPEN=1, 精度补偿有效;
8. 设定 RTC_CR1.START=1, 计数开始;
9. 等待 2 个计数周期以上;
10. 1Hz 输出开始。

24.4 中断

RTC 支持 2 种中断类型：计时闹钟中断、定周期中断。

24.4.1 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。

24.4.2 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。

24.5 寄存器描述

表 24-2 RTC 基地址

名称	基地址	描述
RTC	0x400 4C000	RTC 基地址

表 24-3 RTC 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
RTC_CR0	控制寄存器0	0x0000	8	0x00
RTC_CR1	控制寄存器1	0x0004	8	0x00
RTC_CR2	控制寄存器2	0x0008	8	0x00
RTC_CR3	控制寄存器3	0x000C	8	0x00
RTC_SEC	秒计数寄存器	0x0010	8	0x00
RTC_MIN	分计数寄存器	0x0014	8	0x00
RTC_HOUR	时计数寄存器	0x0018	8	0x12
RTC_WEEK	周计数寄存器	0x001C	8	0x00
RTC_DAY	日计数寄存器	0x0020	8	0x00
RTC_MON	月计数寄存器	0x0024	8	0x00
RTC_YEAR	年计数寄存器	0x0028	8	0x00
RTC_ALMMIN	分闹钟寄存器	0x002C	8	0x00
RTC_ALMHOUR	时闹钟寄存器	0x0030	8	0x12
RTC_ALMWEEK	周闹钟寄存器	0x0034	8	0x00
RTC_ERRCRH	时钟误差补偿寄存器	0x0038	8	0x00
RTC_ERRCRL	时钟误差补偿寄存器	0x003C	8	0x20

24.5.1 控制寄存器 0 (RTC_CR0)

复位值: 0x00

位/位域	标记	位名	功能	读写
b7				
b6				
b5				
b4				
b3				
b2				
b1				
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 初始化所有的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	RW
b7~b1	Res	保留位	读出时为“0”，写入时写“0”	RW

24.5.2 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/位域	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	RW																																
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。	RW																																
b5	ONEHZOE	1Hz输出许可	0: 1Hz输出禁止 1: 1Hz输出许可	RW																																
b4	Res	保留位	读出时为“0”, 写入时写“0”	RW																																
b3	AMPM	时制选择	0: 12小时时制 1: 24小时时制	RW																																
b2~0	PRDS[2:0]	周期中断选择	周期选择设定: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>PRDS[2]</th> <th>PRDS[1]</th> <th>PRDS[0]</th> <th>周期选择</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>不选择</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>每0.5秒周期</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>每1秒周期</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>每1分周期</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>每1时周期</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>每1日周期 (每日00时00分00秒)</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">X</td> <td>每1月周期 (每月1日00时00分00秒)</td> </tr> </tbody> </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)	RW
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: 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。																																	

24.5.3 控制寄存器 2 (RTC_CR2)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
ALME	ALMIE	PRDIE	Res	ALMF	PRDF	RWEN	RWREQ

位/位域	标记	位名	功能	读写
b7	ALME	闹钟功能使能	0: 闹钟功能禁止 1: 闹钟功能许可	RW
b6	ALMIE	闹钟中断使能	0: 闹钟中断禁止 1: 闹钟中断许可	RW
b5	PRDIE	周期中断使能	0: 周期中断禁止 1: 周期中断许可	RW
b4	Res	保留位	读出时为“0”，写入时写“0”	RW
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。	RW
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	RW
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。	RW
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”。 如果RWREQ置位后，没有在1秒内及时清零，将会导致计数器停止计数，使实时时钟RTC不能精确计时；如果RWREQ位一直维持置位状态，则实时时钟RTC会停止计时。	RW

24.5.4 控制寄存器 3 (RTC_CR3)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
RCKSEL	Res		LRCEN	Res			

位/位域	标记	位名	功能	读写
b7	RCKSEL	RTC计数时钟选择	0: 选择外部低速发振器XTAL32时钟 1: 选择内部低速发振器LRC时钟或XTAL小数分频时钟 注意: 在RCKSEL=1时, RTC计数时钟由PWC_PWRC6寄存器的RTCCKSEL[1:0]决定。	RW
b6-b5	Res	保留位	读出时为“0”, 写入时写“0”	RW
b4	LRCEN	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LRCEN位使能。	RW
b3-b0	Res	保留位	读出时为“0”, 写入时写“0”	RW

24.5.5 秒计数寄存器 (RTC_SEC)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	SECD[2:0]			SECU[3:0]			

位/位域	标记	位名	功能	读写
b7	Res	保留位	读出时为“0”, 写入时写“0”	RW
b6~b4	SECD[2:0]	秒十位	秒十位计数值	RW
b3~b0	SECU[3:0]	秒个位	秒个位计数值	RW

注:

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。

24.5.6 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/位域	标记	位名	功能	读写
b7	Res	保留位	读出时为“0”，写入时写“0”	RW
b6~b4	MIND[2:0]	分十位	分十位计数值	RW
b3~b0	MINU[3:0]	分个位	分个位计数值	RW

注:

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。

24.5.7 时计数寄存器 (RTC_HOUR)

复位值: 0x12

b7	b6	b5	b4	b3	b2	b1	b0
Res		HOURL[1:0]		HOURU[3:0]			

位/位域	标记	位名	功能	读写
b7~b6	Res	保留位	读出时为“0”，写入时写“0”	RW
b5~b4	HOURL[1:0]	时十位	时十位计数值	RW
b3~b0	HOURU[3:0]	时个位	时个位计数值	RW

注意:

- 24 小时制时, 表示 0-23 小时。12 小时制时, b5=0 表示 AM, 则 01~12 表示上午; b5=1 表示 PM, 则 13~23 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, 13~23 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小时制	AMPM=1	12小时制	AMPM=0
时间	寄存器表示	时间	寄存器表示
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

24小时时制	AMPM=1	12小时时制	AMPM=0
时间	寄存器表示	时间	寄存器表示
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

24.5.8 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/位域	标记	位名	功能	读写
b7~b6	Res	保留位	读出时为“0”，写入时写“0”	RW
b5~b4	HDAYD[1:0]	日十位	日十位计数值	RW
b3~b0	HDAYU[3:0]	日个位	日个位计数值	RW

注:

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

24.5.9 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/位域	标记	位名	功能	读写
b7~b3	Res	保留位	读出时为“0”，写入时写“0”	RW
b2~b0	WEEK[2:0]	周	周计数值	RW

注:

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	0x00
周一	0x01
周二	0x02
周三	0x03
周四	0x04
周五	0x05
周六	0x06

24.5.10 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/位域	标记	位名	功能	读写
b7~b5	Res	保留位	读出时为“0”，写入时写“0”	RW
b4~b0	MON[4:0]	月	月计数值	RW

注:

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。

24.5.11 年计数寄存器 (RTC_YEAR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
YEAR[3:0]				YEARU[3:0]			

位/位域	标记	位名	功能	读写
b7~b4	YEAR[3:0]	年十位	年十位计数值	RW
b3~b0	YEARU[3:0]	年个位	年个位计数值	RW

注:

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。

24.5.12 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/位域	标记	位名	功能	读写
b7	Res	保留位	读出时为“0”，写入时写“0”	RW
b6~b4	ALMMIND[3:0]	分闹钟十位	分闹钟十位匹配值	RW
b3~b0	ALMMINU[3:0]	分闹钟个位	分闹钟个位匹配值	RW

注:

请设定十进制 0~59 的 BCD 码。写入其他值，不会发生闹钟匹配。

24.5.13 时闹钟寄存器 (RTC_ALMHOUR)

复位值: 0x12

b7	b6	b5	b4	b3	b2	b1	b0
Res		ALMHOURD[1:0]		ALMHOURU[3:0]			

位/位域	标记	位名	功能	读写
b7~b6	Res	保留位	读出时为“0”，写入时写“0”	RW
b5~b4	ALMHOURD[1:0]	时闹钟十位	时闹钟十位匹配值	RW
b3~b0	ALMHOURU[3:0]	时闹钟个位	时闹钟个位匹配值	RW

注:

请根据时制设定正确的闹钟匹配值，否则不会发生时闹钟匹配。

24.5.14 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/位域	标记	位名	功能	读写
b7	Res	保留位	读出时为“0”，写入时写“0”	RW
b6~b0	ALMWEEK[6:0]	周闹钟	周闹钟匹配值。 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。	RW

注:

请根据时制设定正确的闹钟匹配值，否则不会发生时闹钟匹配。

24.5.15 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]

位/位域	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	RW
b6~b1	Res	保留位	读出时为“0”，写入时写“0”	RW
b0	COMP[8]	补偿值	与COMP[7:0]共同设定补偿值	RW

复位值: 0x20

RTC_ERRCRL

b7	b6	b5	b4	b3	b2	b1	b0
COMP[7:0]							

位/位域	标记	位名	功能	读写																																																
b7~b0	COMP[7:0]	补偿值	<p>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：</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2">补偿值设定</th> <th>补偿数</th> </tr> <tr> <th>COMPEN</th> <th>COMP[8:0]</th> <th></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1 0 0 0 0 0 0 0 0</td> <td>-275.5ppm</td> </tr> <tr> <td></td> <td>1 0 0 0 0 0 0 0 1</td> <td>-274.6ppm</td> </tr> <tr> <td></td> <td>~ ~ ~ ~ ~ ~ ~ ~ ~</td> <td>~</td> </tr> <tr> <td></td> <td>0 0 0 0 0 0 0 0 0</td> <td>-30.5ppm</td> </tr> <tr> <td></td> <td>~ ~ ~ ~ ~ ~ ~ ~ ~</td> <td>~</td> </tr> <tr> <td></td> <td>0 0 0 0 1 1 1 1 1</td> <td>-0.96ppm</td> </tr> <tr> <td></td> <td>0 0 0 1 0 0 0 0 0</td> <td>0ppm</td> </tr> <tr> <td></td> <td>0 0 0 1 0 0 0 0 1</td> <td>+0.96ppm</td> </tr> <tr> <td></td> <td>~ ~ ~ ~ ~ ~ ~ ~ ~</td> <td>~</td> </tr> <tr> <td></td> <td>0 0 1 0 0 0 0 0 0</td> <td>+30.5ppm</td> </tr> <tr> <td></td> <td>~ ~ ~ ~ ~ ~ ~ ~ ~</td> <td>~</td> </tr> <tr> <td></td> <td>0 1 1 1 1 1 1 1 0</td> <td>+212.0ppm</td> </tr> <tr> <td></td> <td>0 1 1 1 1 1 1 1 1</td> <td>+212.9ppm</td> </tr> <tr> <td>0</td> <td>X X X X X X X X 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 X	无效	RW
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 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：

实际发振频率 = 32768 × 0.9999888 ≈ 32767.63

补偿目标值 = (实际发振频率-目标频率) / 目标频率 × 10⁶

$$= (32767.63 - 32768) / 32768 \times 10^6$$

$$= -11.29 \text{ ppm}$$

设定值计算：

$$\text{COMP}[8:0] = \left(\frac{\text{补偿目标值}[\text{ppm}] \times 2^{15}}{10^6} \right)_{\text{取2的补码}} + 0\text{b}0001.00000$$

如果补偿目标值为+ 20.3ppm，计算相应的寄存器值如下：

$$\begin{aligned} \text{COMP}[8:0] &= (20.3 \times 2^{15} / 10^6)_{\text{取2的补码}} + 0\text{b}0001.00000 \\ &= (0.6651904)_{\text{取2的补码}} + 0\text{b}0001.00000 \\ &= 0\text{b}0000.10101 + 0\text{b}0001.00000 \\ &= 0\text{b}0001.10101 \end{aligned}$$

如果补偿目标值为- 20.3ppm，计算相应的寄存器值如下：

$$\begin{aligned} \text{COMP}[8:0] &= (-20.3 \times 2^{15} / 10^6)_{\text{取2的补码}} + 0\text{b}0001.00000 \\ &= (-0.6651904)_{\text{取2的补码}} + 0\text{b}0001.00000 \\ &= 0\text{b}1111.01011 + 0\text{b}0001.00000 \\ &= 0\text{b}0000.01011 \end{aligned}$$

25 看门狗计数器 (WDT/ SWDT)

25.1 简介

本产品有两个看门狗计数器，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的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 25-1。

表 25-1 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: 3.72小时 (SWDTLRC: 10KHz) WDT: 10.7秒 (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

25.2 功能说明

25.2.1 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。

25.2.2 硬件启动方式

ICG0 寄存器的位 16（WDTAUTC）、位 0（SWDTAUTC）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 25-1 为硬件启动方式的动作例。

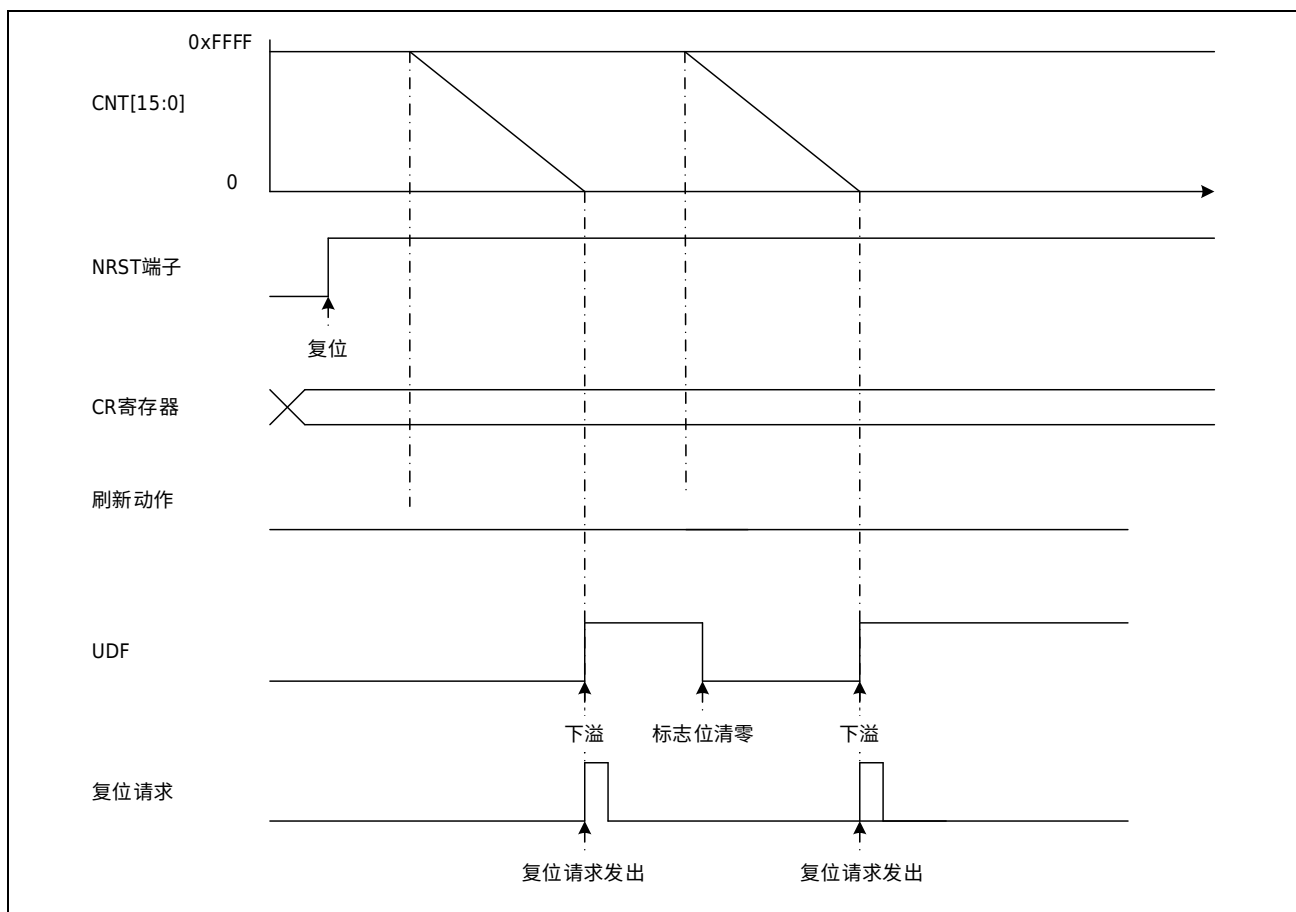


图 25-1 硬件启动例

25.2.3 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 25-2 为软件启动方式的动作例。

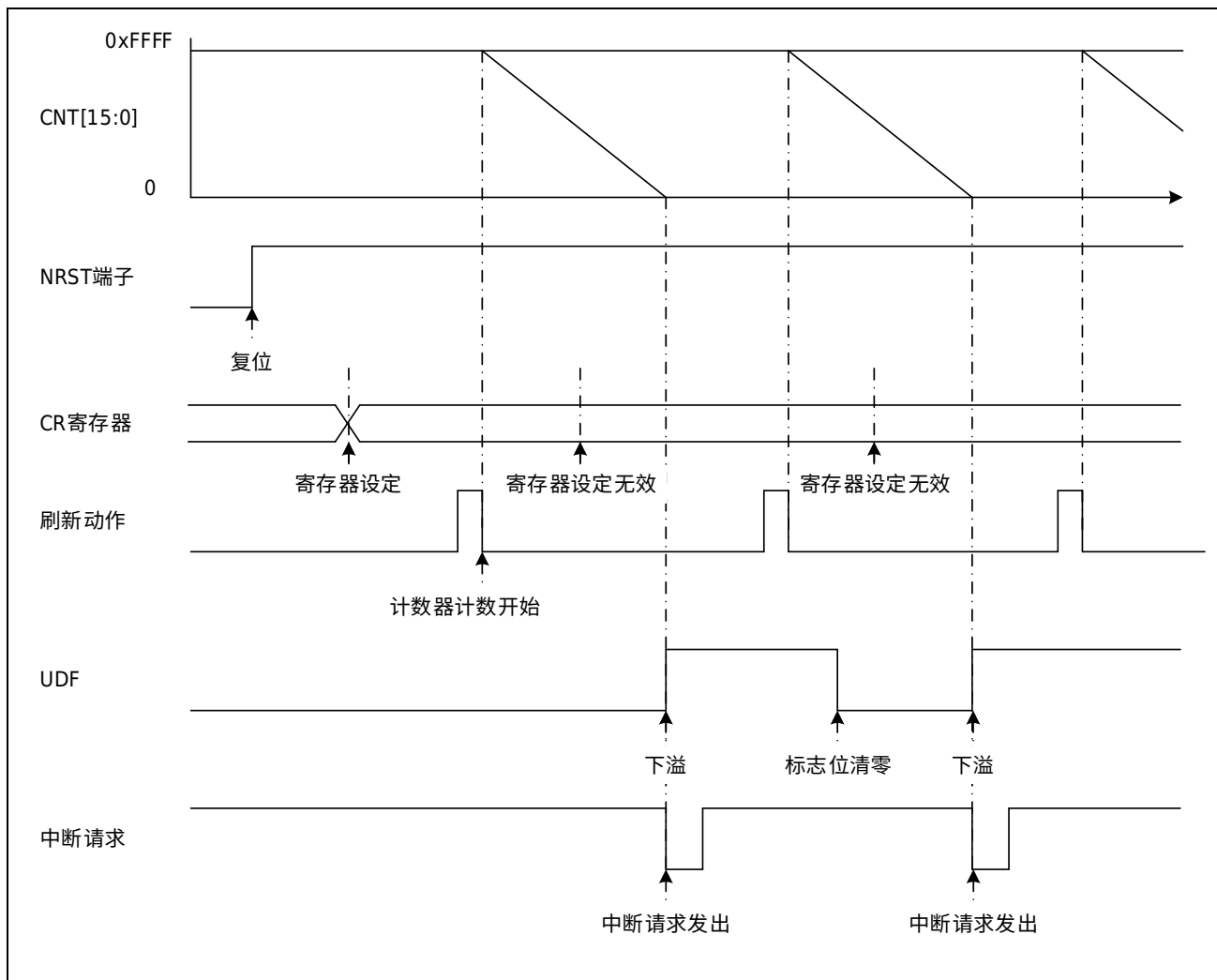


图 25-2 软件启动例

25.2.4 刷新动作

WDT/ SWDT_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。

WDT/ SWDT_RR 寄存器在写 0x0123、0x3210 之间，若发生对其他寄存器访问或读取 WDT/ SWDT_RR 寄存器等，不影响正常的刷新动作。

如图 25-3 所示动作示例。

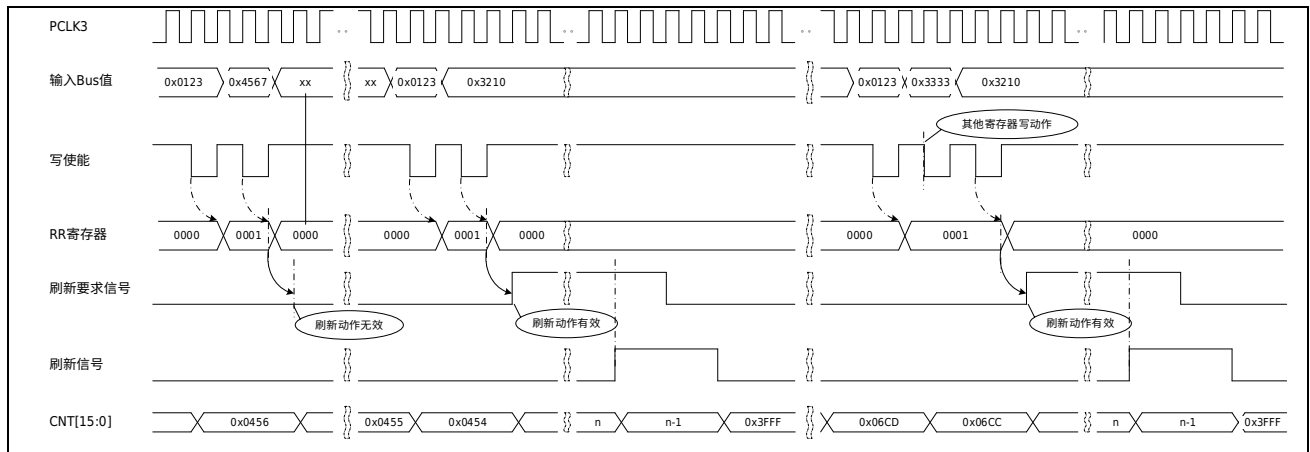


图 25-3 各种刷新动作时序示例（动作确认，刷新要求信号的下降沿等）

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。

25.2.5 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）。

25.2.6 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。

25.2.7 计数下溢

如图 25-4 例，在递减计数到零时产生下溢。

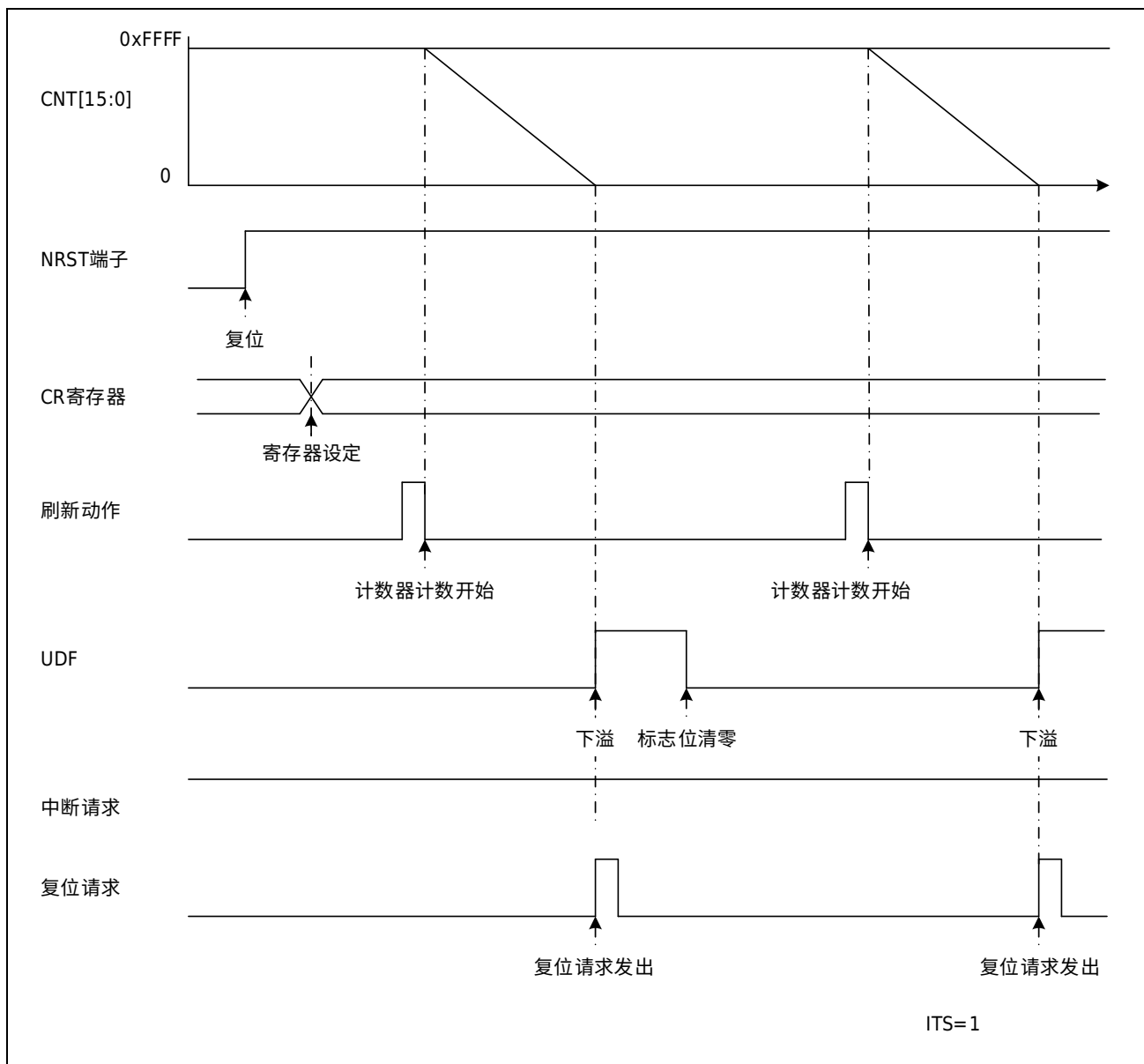


图 25-4 计数器下溢动作例

25.2.8 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 25-5 为刷新动作例。

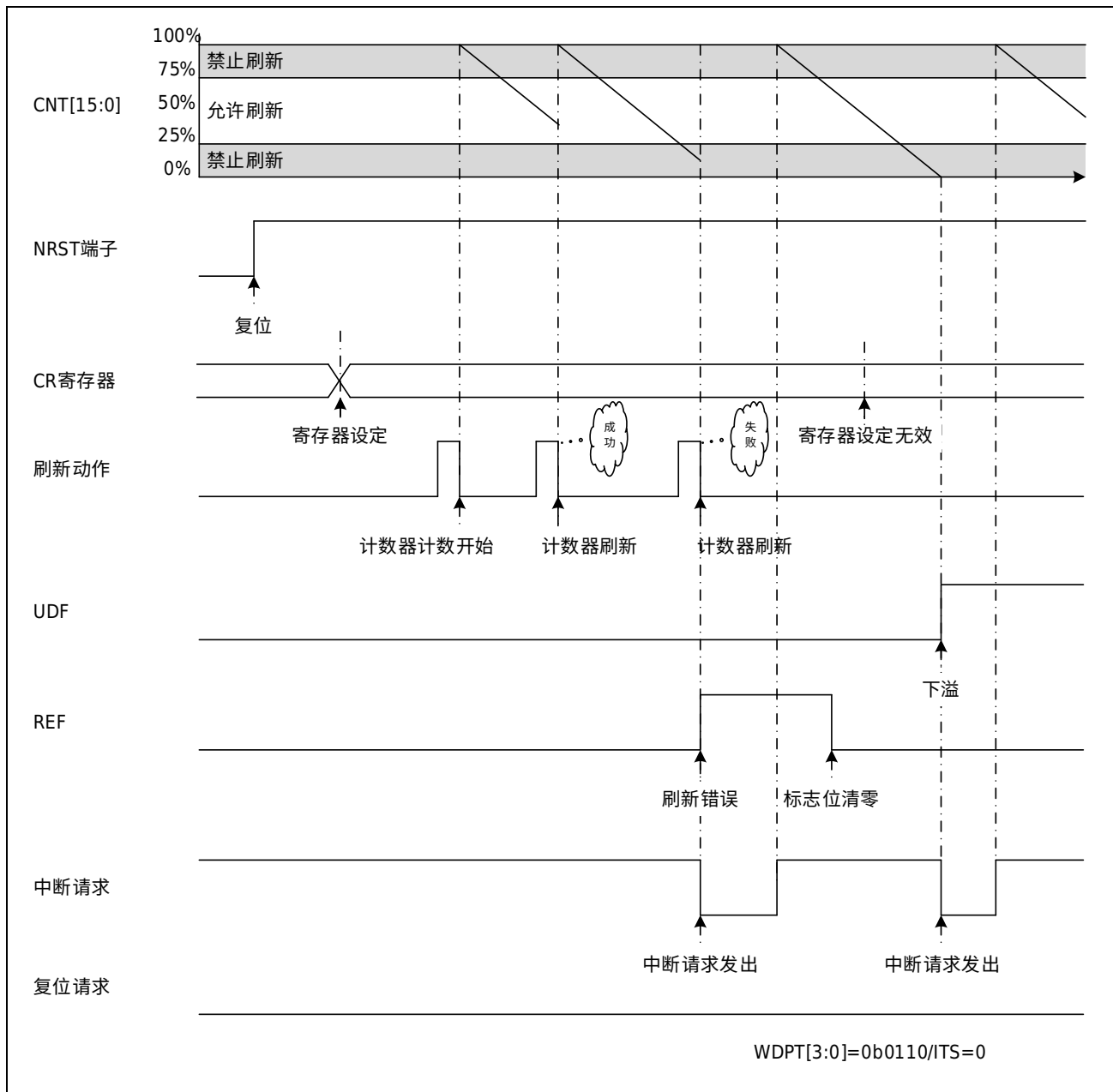


图 25-5 计数器刷新动作例

25.3 寄存器描述

表 25-2 WDT/ SWDT 基地址

名称	基地址	描述
WDT	0x4004 9000	WDT 基地址
SWDT	0x4004 9400	SWDT 基地址

表 25-3 WDT 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
WDT_CR	WDT控制寄存器	0x00	32	0x8001 0FF3
WDT_SR	WDT状态寄存器	0x04	32	0x0000 0000
WDT_RR	WDT刷新寄存器	0x08	32	0x0000 0000

表 25-4 SWDT 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
SWDT_CR	SWDT控制寄存器	0x00	32	0x8001 0FF3
SWDT_SR	SWDT状态寄存器	0x04	32	0x0000 0000
SWDT_RR	SWDT刷新寄存器	0x08	32	0x0000 0000

25.3.1 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	Res														SLPOFF
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				WDPT				CKS[3:0]				Res		PERI[1:0]	

位/位域	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	RW
b30~b17	Res	保留位	读出时为“0”，写入时写“0”	RW
b16	SLPOFF	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	RW
b15~b12	Res	保留位	读出时为“0”，写入时写“0”	RW
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%	RW
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:	RW

			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	Res	保留位	读出时为“0”，写入时写“0”	RW
			00: 256 cycle	
			01: 4096 cycle	
b1~b0	PERI[1:0]	计数周期	10: 16384 cycle	RW
			11: 65536 cycle	

25.3.2 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
Res													REF	UDF	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT[15:0]															

位/位域	标记	位名	功能	读写
b31~b18	Res	保留位	读出时为“0”，写入时写“0”	RW
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。	RW
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。	RW
b15~b0	CNT[15:0]	计数值	计数器当前计数值	R

25.3.3 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
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RF[15:0]															

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”，写入时写“0”	RW
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0x0123后，读出寄存器为0x0000 0001;其余情况读 出值都是0x0000 0000。	RW

25.4 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 $PCLK3 \text{ 频率} \geq \text{计数时钟频率} \times 4$ 。

26 通用同步异步收发器 (USART)

26.1 简介

本产品搭载通用同步异步收发器 (USART) 模块 4 个单元。能够灵活地与外部设备进行全双工数据交换；本产品搭载的 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，Driver Enable (DE) 功能。与 Timer0 模块配合支持 UART 接收超时功能。USART_1 支持通过 RX 线唤醒 STOP 模式功能。

26.2 主要特性

- 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_2, USART_3, USART_4 支持
- 智能卡：USART_1 支持
- UART 接收超时功能：USART_1, USART_2, USART_3, USART_4 支持
- RS485 Driver Enable：USART_1, USART_2, USART_3, USART_4 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
- 支持 RS485 Driver Enable 功能

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-Break Field）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

26.3 功能说明

26.3.1 USART 系统框图

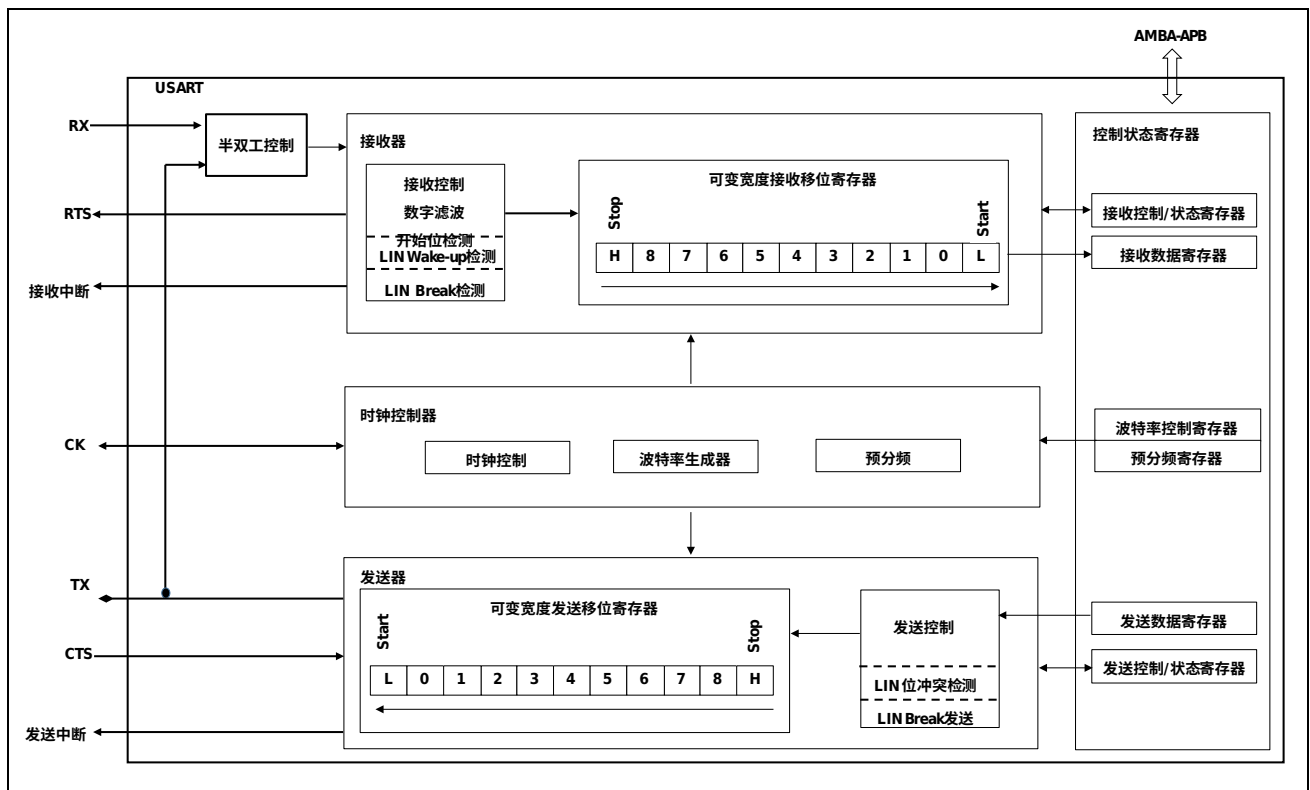


图 26-1 USART 系统框图

表 26-1 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/DE	输出	调制解调器操作管脚 请求发送管脚/driver enable管脚

n: 1~4

26.3.2 UART 功能

26.3.2.1 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 0b00 或者 0b01 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]=0b00 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]=0b01 时从 USARTn_CK 管脚输出与通信波特率相同频率的时钟。

当 USARTn_PR.ULBREN 为 0 时，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 PCLK1, PCLK1/4, PCLK1/16, PCLK1/64，当 USARTn_PR.ULBREN 为 1 时，超低波特率功能使能，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 PCLK1/128, PCLK1/256, PCLK1/512, PCLK1/1024。

外部时钟源

USARTn_CR2.CLKC[1:0]位设定为 0b10 或者 0b11 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：

$$B = \frac{C}{8 \times (2 - \text{OVER8}) \times (\text{DIV_Integer} + 1)}$$

B: 波特率，单位: Mbps

C: 当 USARTn_PR.ULBREN 为 0 时，USARTn_PR.PSC[1:0]位设定的时钟 (PCLK1, PCLK1/4, PCLK1/16, PCLK1/64) 单位: MHz

当 USARTn_PR.ULBREN 为 1 时，USARTn_PR.PSC[1:0]位设定的时钟 (PCLK1/128, PCLK1/256, PCLK1/512, PCLK1/1024) 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 PCLK1/8 (Mbps)。

外部时钟源时，外部输入 UART 时钟的最高频率要求为 PCLK1/4，所以时钟源为外部输入时钟时：当 USARTn_CR1.OVER8=0 时，最高波特率为 PCLK1/64 (Mbps)；当 USARTn_CR1.OVER8=1 时，最高波特率为 PCLK1/32 (Mbps)。

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**数据手册电气特性**章节规定的最高通信波特率。

26.3.2.2 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。

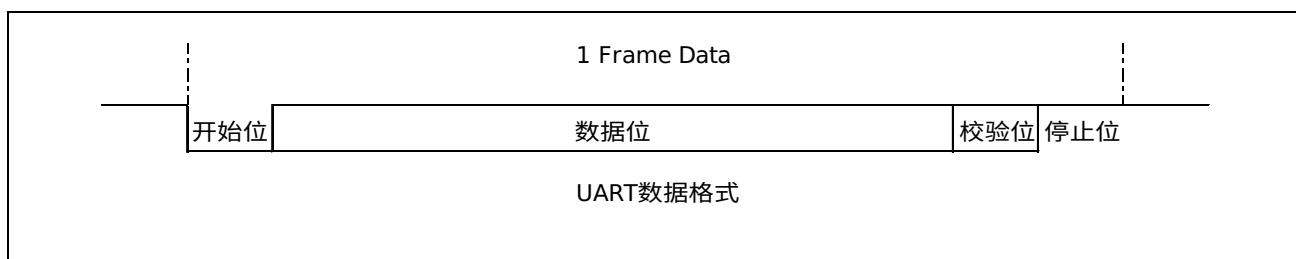


图 26-2 UART 数据格式

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

停止为固定为高电平，可以配置成 1 位或者 2 位。

26.3.2.3 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)，且不在接收数据
- USARTn_RDR.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

26.3.2.4 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。

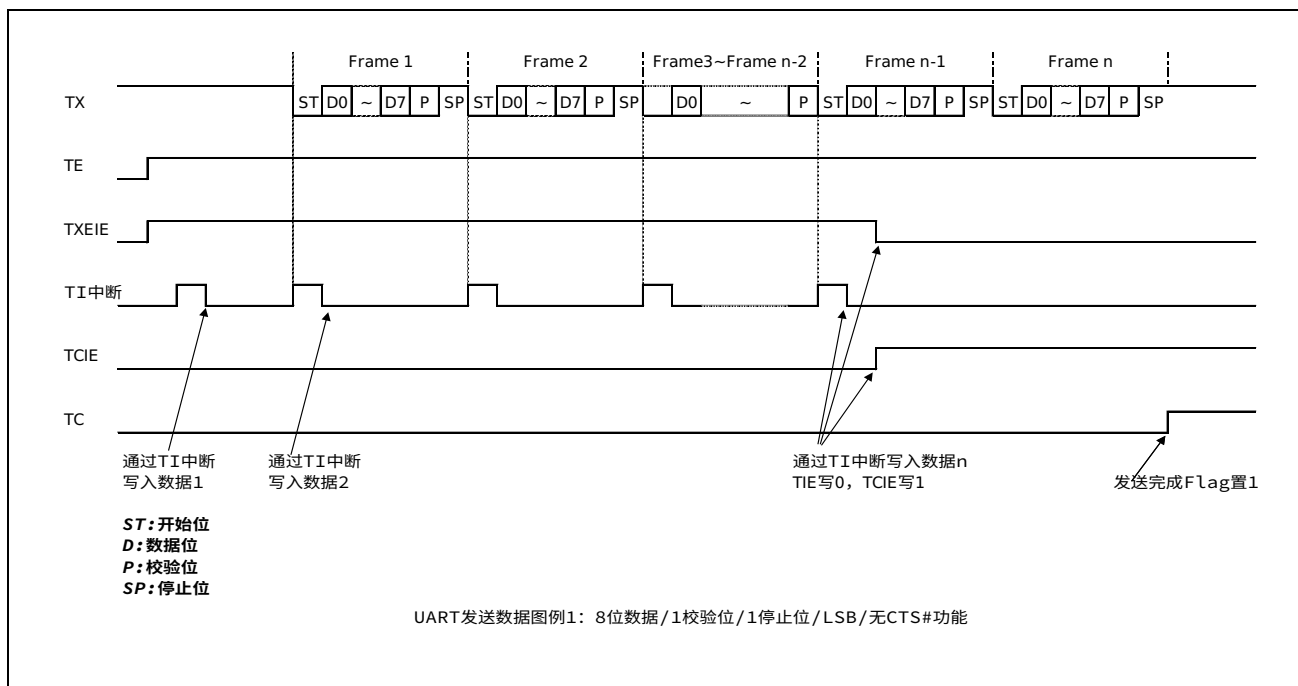


图 26-3 UART 发送数据图例 1

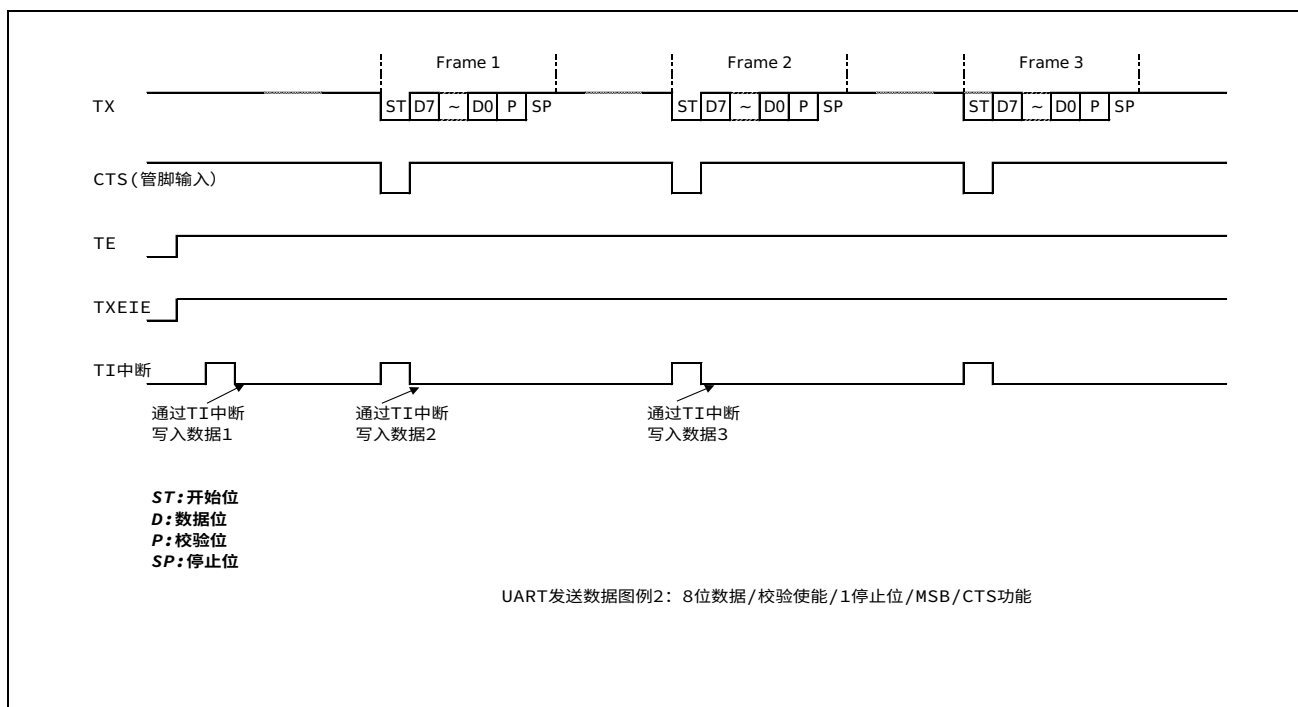


图 26-4 UART 发送数据图例 2

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。

26.3.2.5 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。

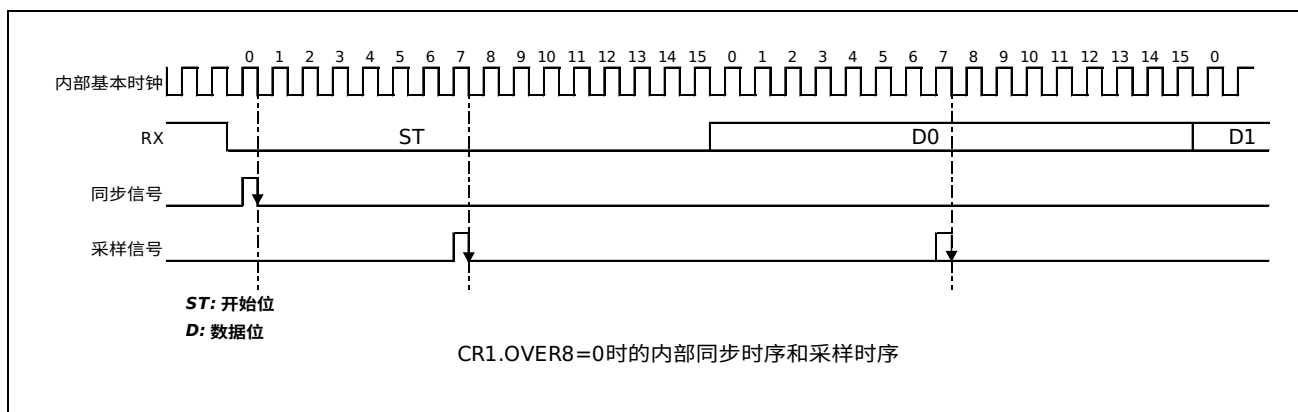


图 26-5 UART 内部同步和采样时序

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 (Frame Length)。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 26-2 DIV_Fraction 为 0 时 UART 接收器的容差

FL	OVER8位	OVER8位=1
10	4.375%	3.75%
11	3.97%	3.41%
12	3.646%	3.125%

表 26-3 DIV_Fraction 不为 0 时 UART 接收器的容差

FL	OVER8位	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 26-2 和表 26-3 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

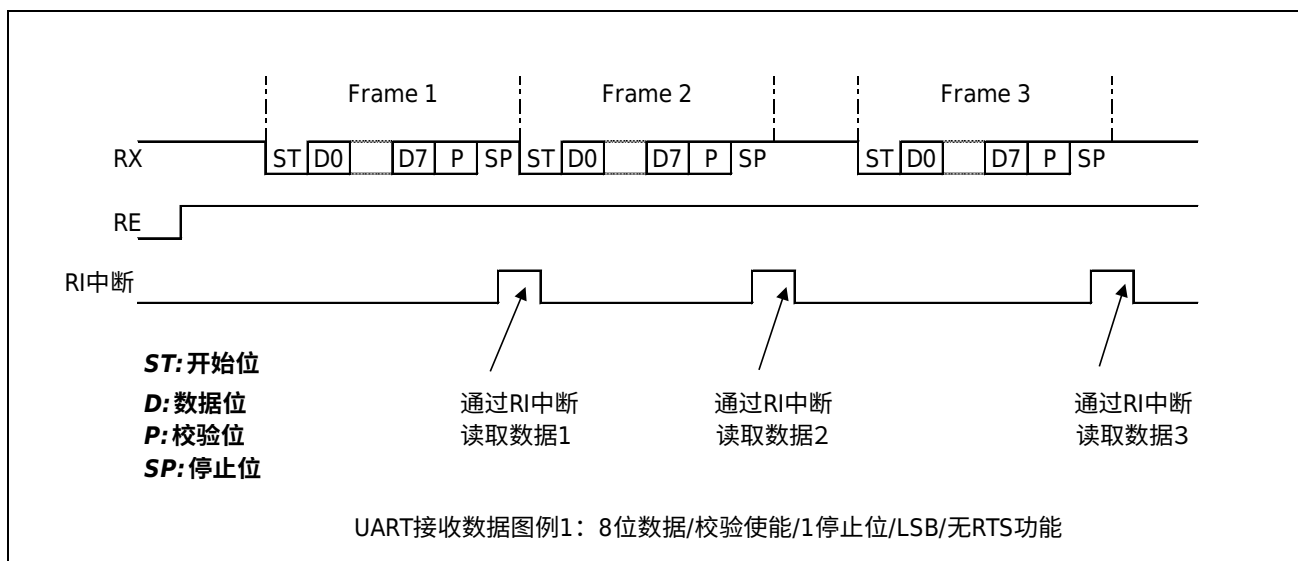


图 26-6 UART 接收数据图例 1

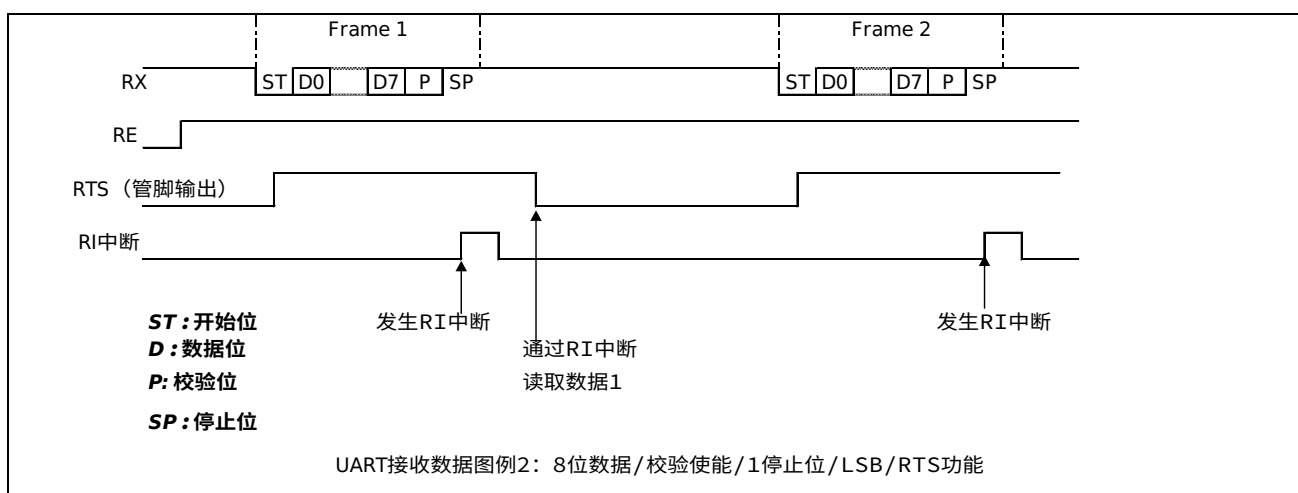


图 26-7 UART 接收数据图例 2

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。

26.3.2.6 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：

USART1: Timer0_1 A 通道

USART2: Timer0_1 B 通道

USART3: Timer0_2 A 通道

USART4: Timer0_2 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_CMPAR 值设定计算公式如下：

$$CMPAR = \frac{RTB}{2^{CKDIVA}} - \alpha$$

CMPAR: TMR0_CMPAR 寄存器值，计算结果请向上进位取整。

α : Timer0 异步计数同步电路带来的延迟

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 + $\alpha \times 2^{CKDIVA}$

实际发生 TIMEOUT 的时间与 RTB 值之间有一定误差，误差 $\leq 2^{CKDIVA}$ 。

CKDIRA: TMR0.BCONR.CKDIVA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.CLKC[1:0]位选择时钟源(如果选择内部时钟源时需设置 CR2.CLKC[1:0]=0b01);
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=0;
8. 设置 TMR0_CNTAR 为 0, 设置 TMR0_CMPAR 寄存器和 TMR0_BCONR.CKDIVA寄存器决定 TIMEOUT 时间;
9. 设置 TMR0_BCONR.HCLEA=1, TMR0_BCONR.HSTAA=1, TMR0_BCONR.ASYNCLKA=1, TMR0_BCONR.SYNCLKA=0, TMR0_BCONR.SYNSA=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=1 和 TMR0_BCONR.SYNA=0, 再同时设置 TMR0_BCONR.CSTA=0, TMR0_BCONR.SYNCLKA=0 和 TMR0_BCONRSYNSA=1;
通过写 USARTn_CR1.CRTOF 清除 USARTn_SR.RTOF 状态位。

26.3.2.7 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 INTC_WKEN.RXWKEN 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 USART_NFC 寄存器。

26.3.2.8 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。

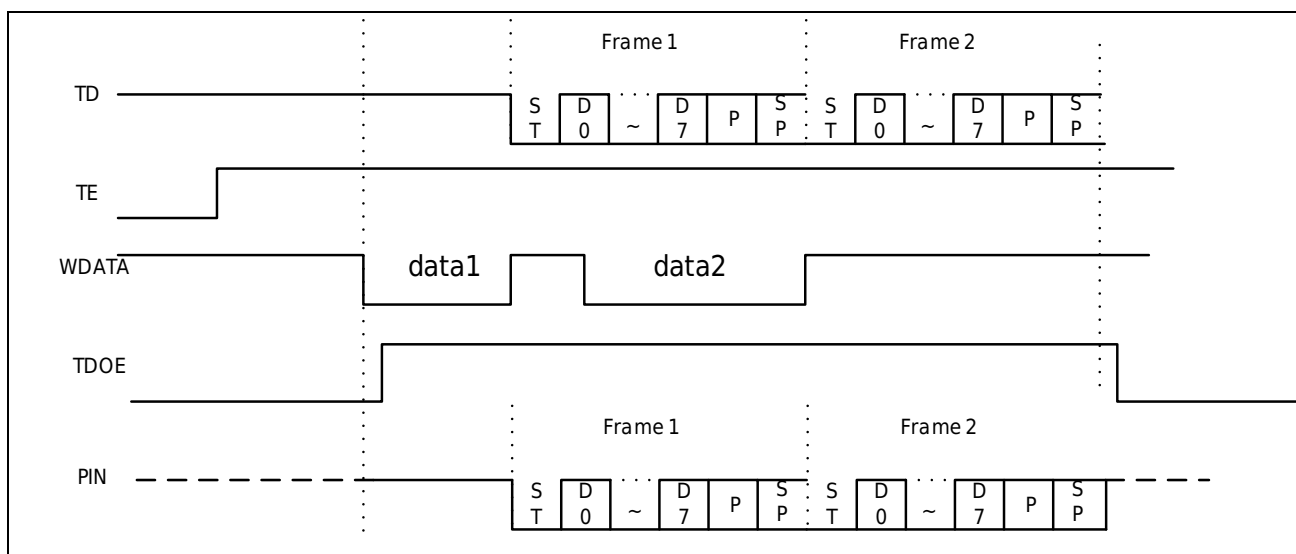


图 26-8 半双工工作时序

26.3.2.9 UART driver enable 功能

UART 模式支持 driver enable 模式，与 RTS 共用输出信号，需要先将 USART_CR3.DEM 配置为高，再把 USART_CR3.RTSE 配置为高。

通过在 USART_CTL3 控制寄存器中设置位 DEM 开启驱动器启用功能，允许用户通过 DE (驱动器启用) 信号激活外部收发器控制。使用 USART_CTL3 控制寄存器中的 DEA[3:0]位字段编程的断言时间是 DE 信号激活和 START 位开始之间的时间。

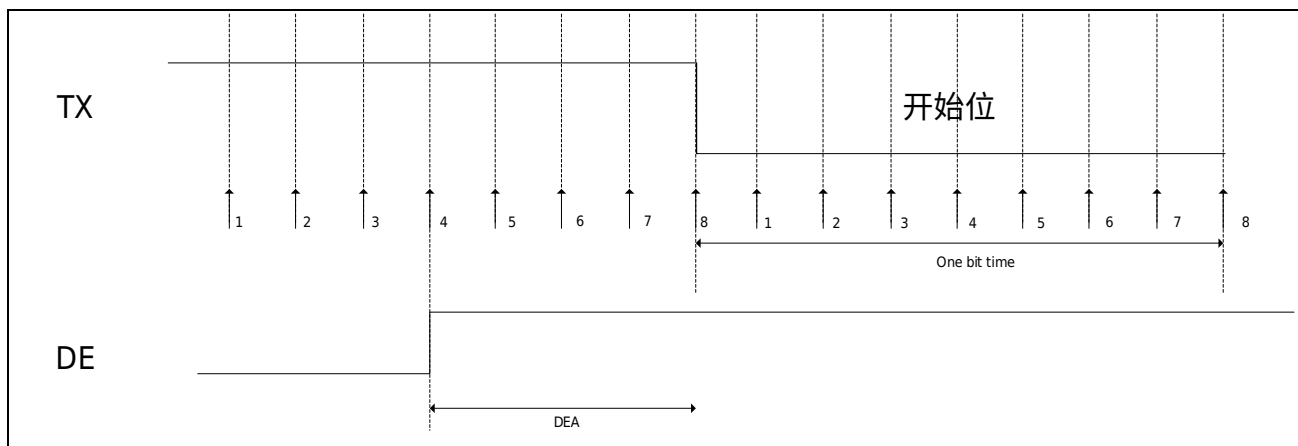


图 26-9 over8=1 DE 断言

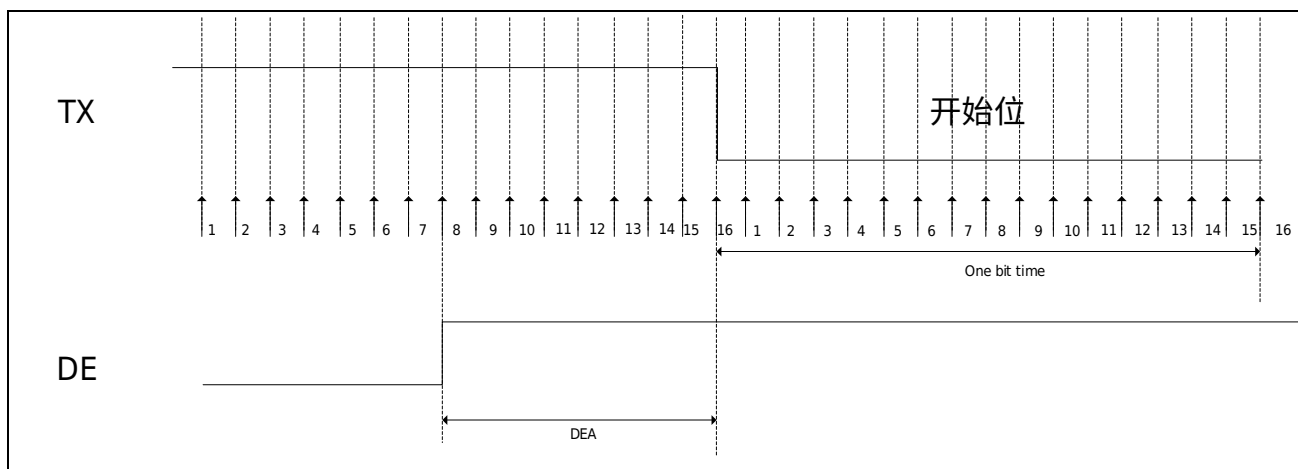


图 26-10 over8=0 DE 断言

使用 USART_CTL3 控制寄存器中的 DED[3:0]位字段编程的去断言时间是最后一个停止位结束与 DE 信号失效之间的时间。DE 信号的极性可以使用 USART_CTL3 控制寄存器中的 DEP 位来配置。

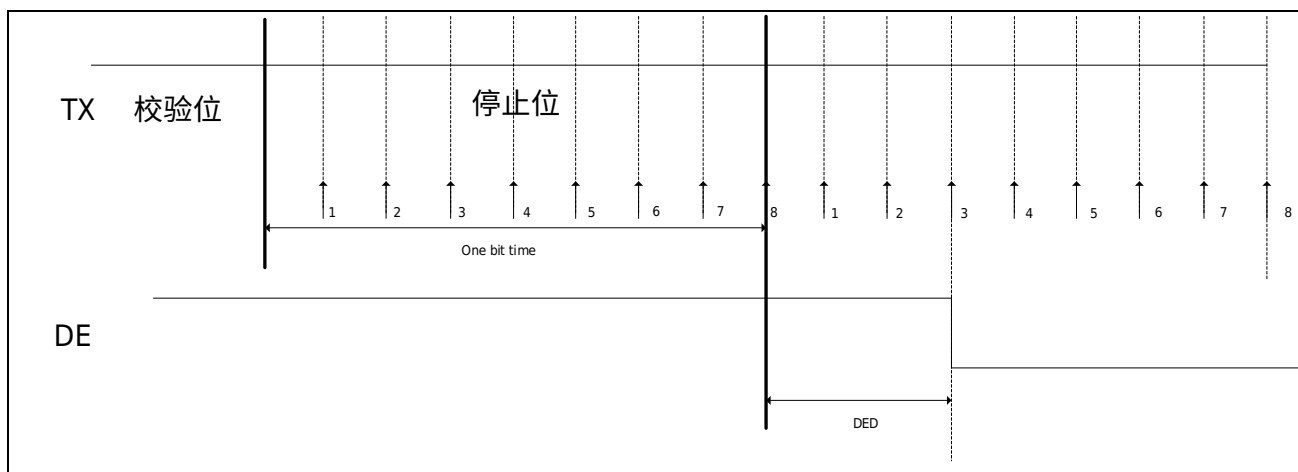


图 26-11 over8=1 DE 去断言

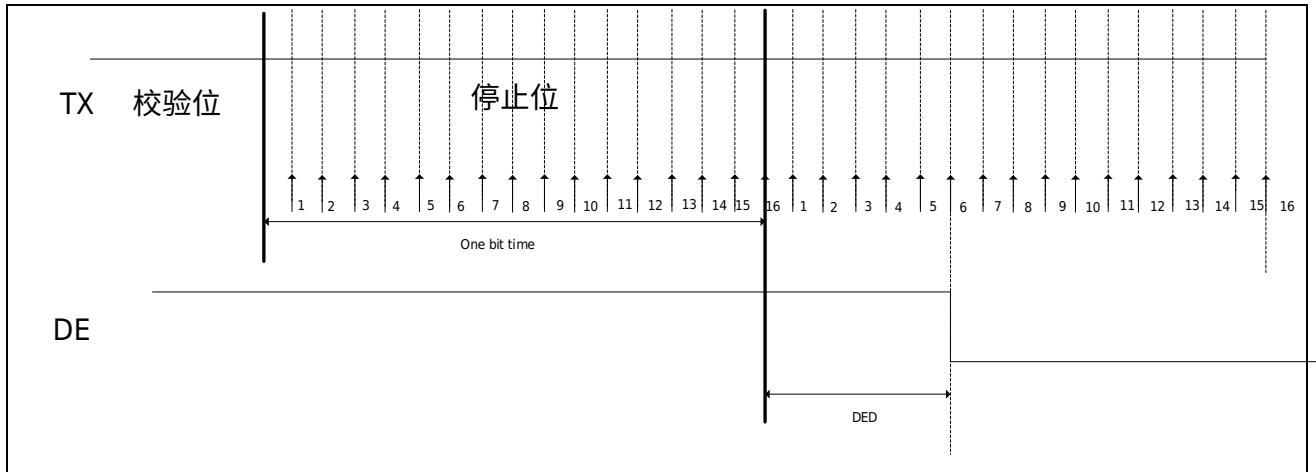


图 26-12 over8=0 DE 去断言

26.3.2.10 UART 中断和事件

表 26-4 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	可
TIMEOUT中断	RTOIE	RTOF	可
RX线唤醒停止模式中断	INT_WUPEN.RXWUEN	-	不可

26.3.3 多处理器通信

26.3.3.1 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。

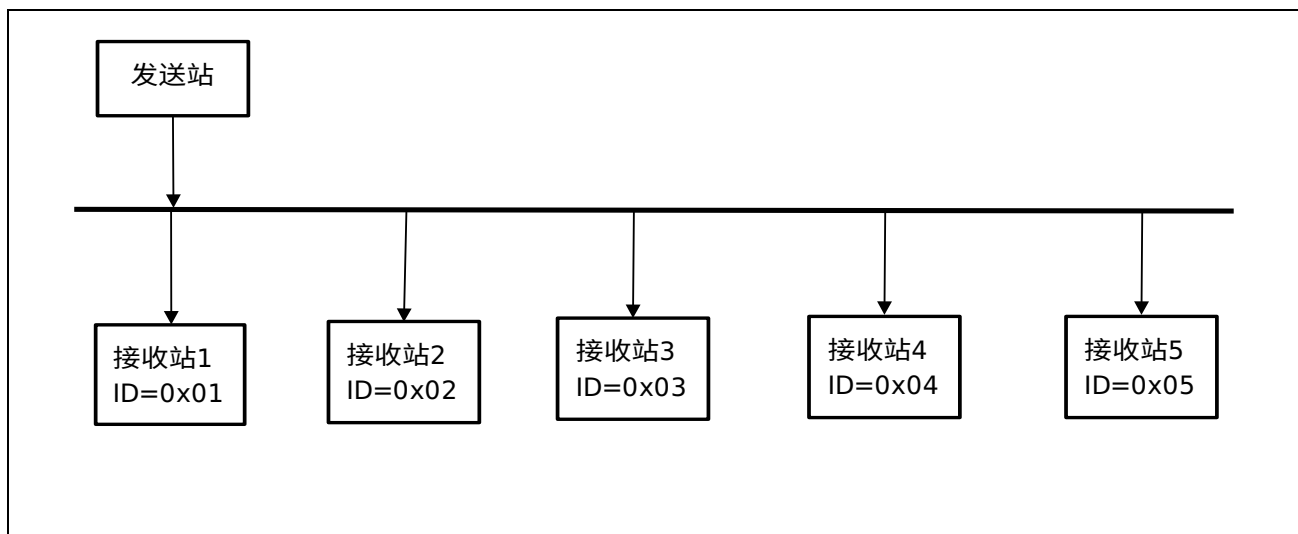


图 26-13 多处理器通信图例

26.3.3.2 数据格式

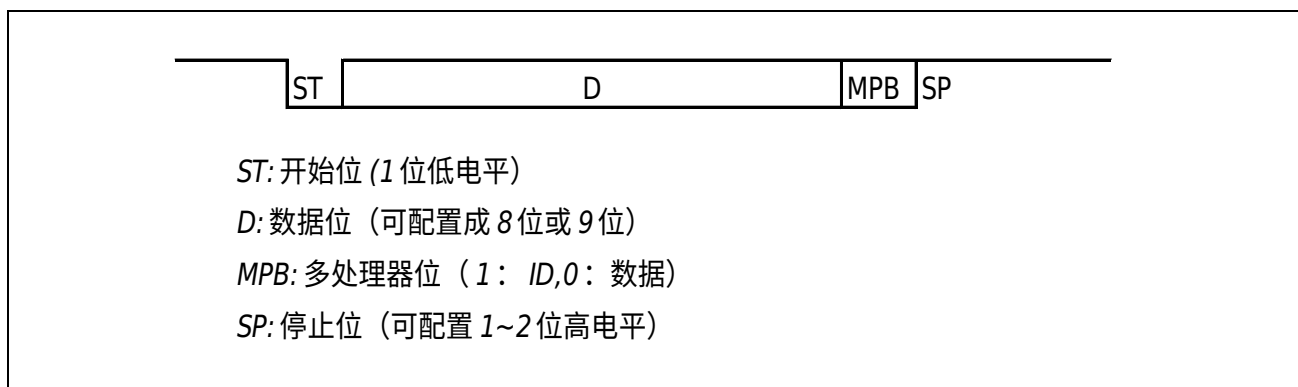


图 26-14 多处理器模式数据格式

26.3.3.3 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.CLK[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。

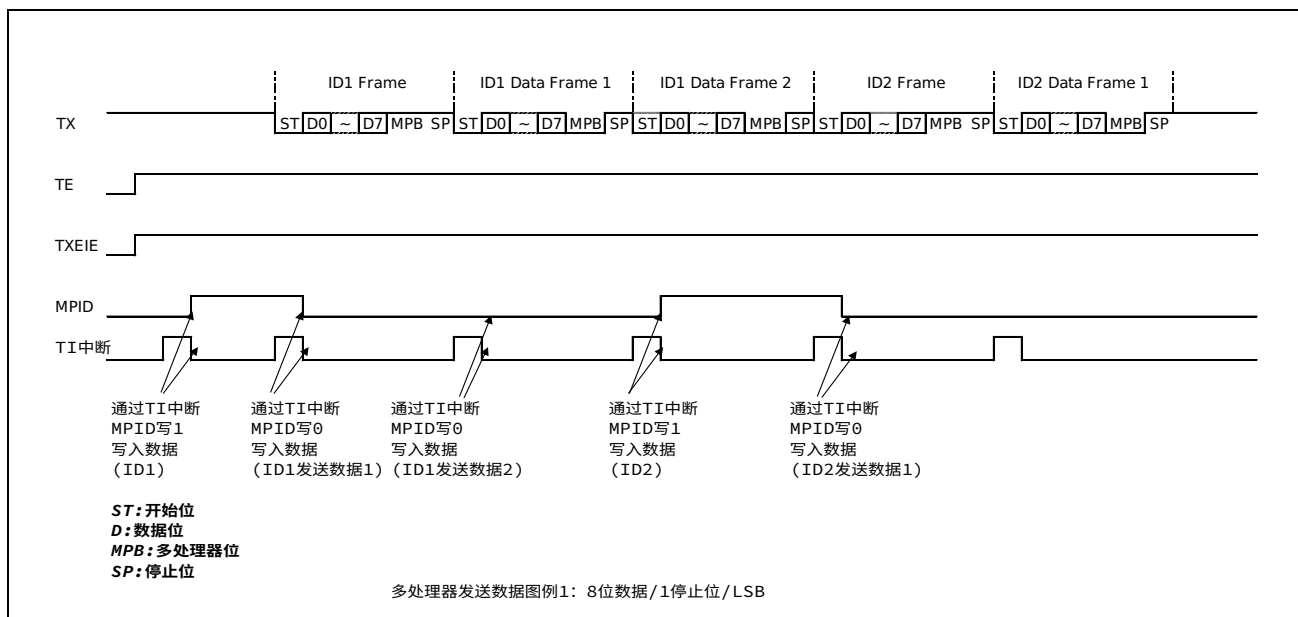


图 26-15 多处理器模式发送数据图例

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。
 - 1) 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
 - 2) 如果 ID 不一致, 软件再次将 USARTn_CR1.SLME 位写 1, 重复 8 的动作

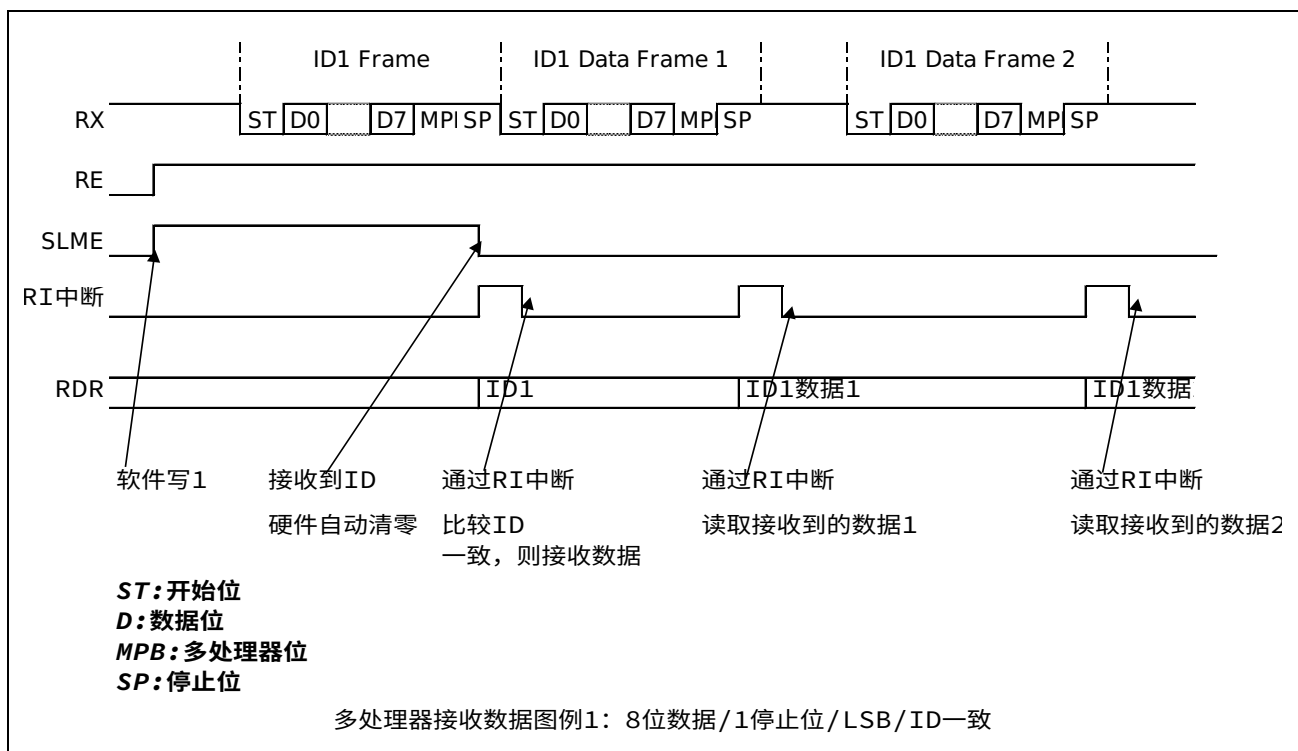


图 26-16 多处理器模式接收数据图例 1

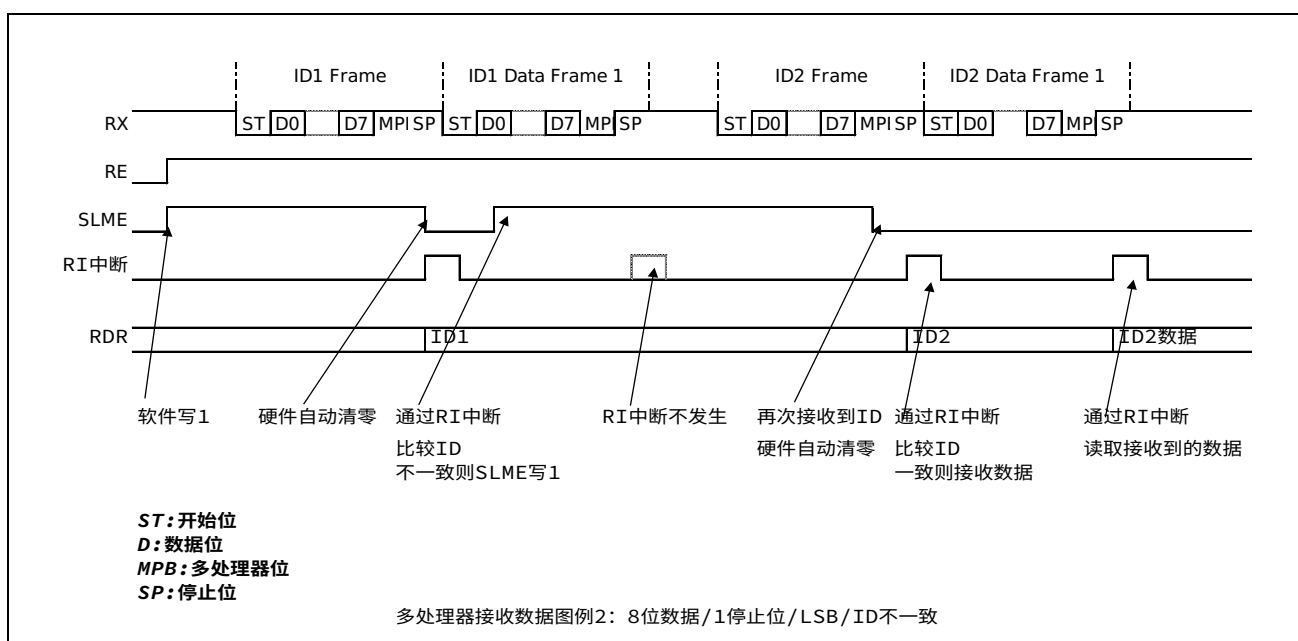


图 26-17 多处理器模式接收数据图例 2

26.3.3.4 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 26-5 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

26.3.4 UART-LIN

26.3.4.1 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。

26.3.4.2 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。

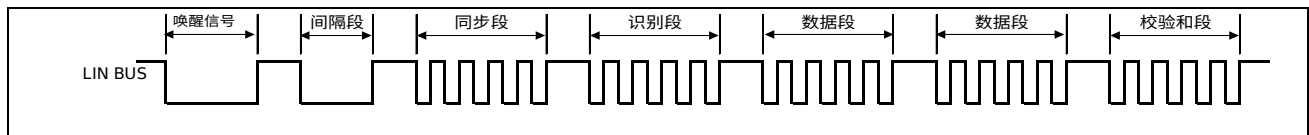


图 26-18 LIN 总线数据行为

26.3.4.3 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.SBK 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。

26.3.4.4 LIN 接收

唤醒信号帧的检测

USARTn_CR2.WKUP=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 130 μ s) 时,置位 USARTn_SR.WKUP 标志, USARTn_CR2.WKUPIE=1 时,产生对应的中断。

USARTn_CR2.WKUP 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.CLK[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.WKUPE=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。

26.3.4.5 LIN 中断和事件

表 26-6 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	可

26.3.5 智能卡

26.3.5.1 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。

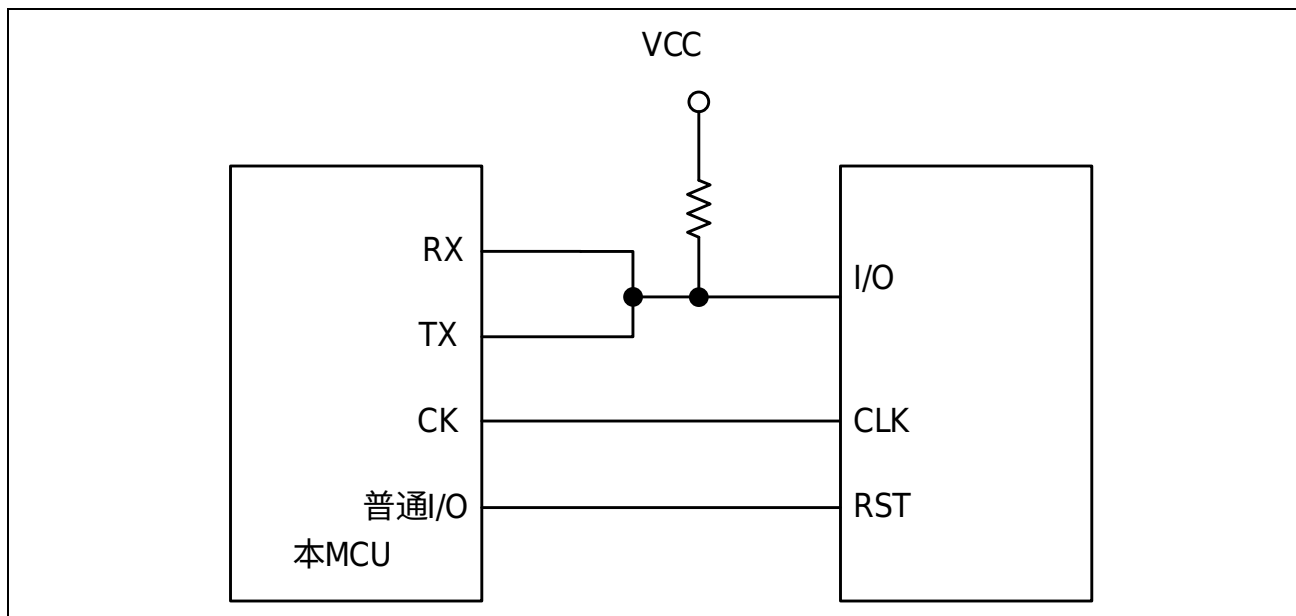


图 26-19 智能卡连接示意图

26.3.5.2 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: 当 `USARTn_PR.ULBREN` 为 0 时, `USARTn_PR.PSC[1:0]` 位设定的时钟 (`PCLK1`, `PCLK1/4`, `PCLK1/16`, `PCLK1/64`) 单位: *MHz*

当 `USARTn_PR.ULBREN` 为 1 时, `USARTn_PR.PSC[1:0]` 位设定的时钟 (`PCLK/128`, `PCLK/256`, `PCLK/512`, `PCLK/1024`), 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 PCLK1, DIV_Integer=0, BCN=0 时, 波特率为最高波特率为 PCLK1/64 (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。

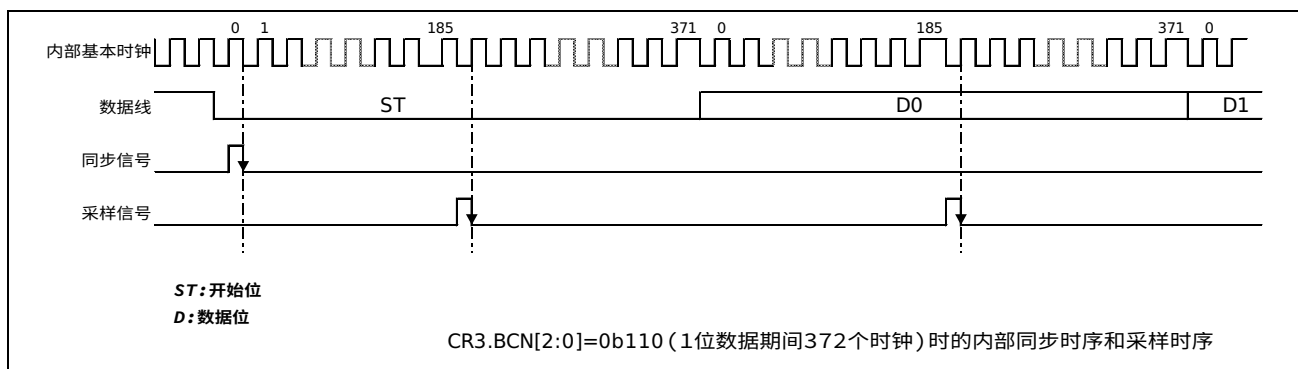


图 26-20 智能卡模式同步时序和采样时序图

接收容差的计算公式如下:

$$RM[\%] = \left\{ 0.5 \times \left(1 - \frac{1}{BCN} \right) - 9.5CFD \right\} \times 100$$

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

26.3.5.3 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。

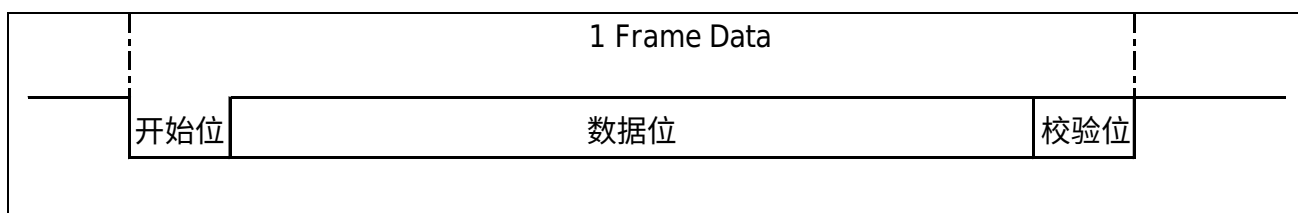


图 26-21 智能卡模式数据格式

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。

26.3.5.4 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.CLK[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。

26.3.5.5 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

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 位必须在接收到下一帧校验位之前清零。

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。

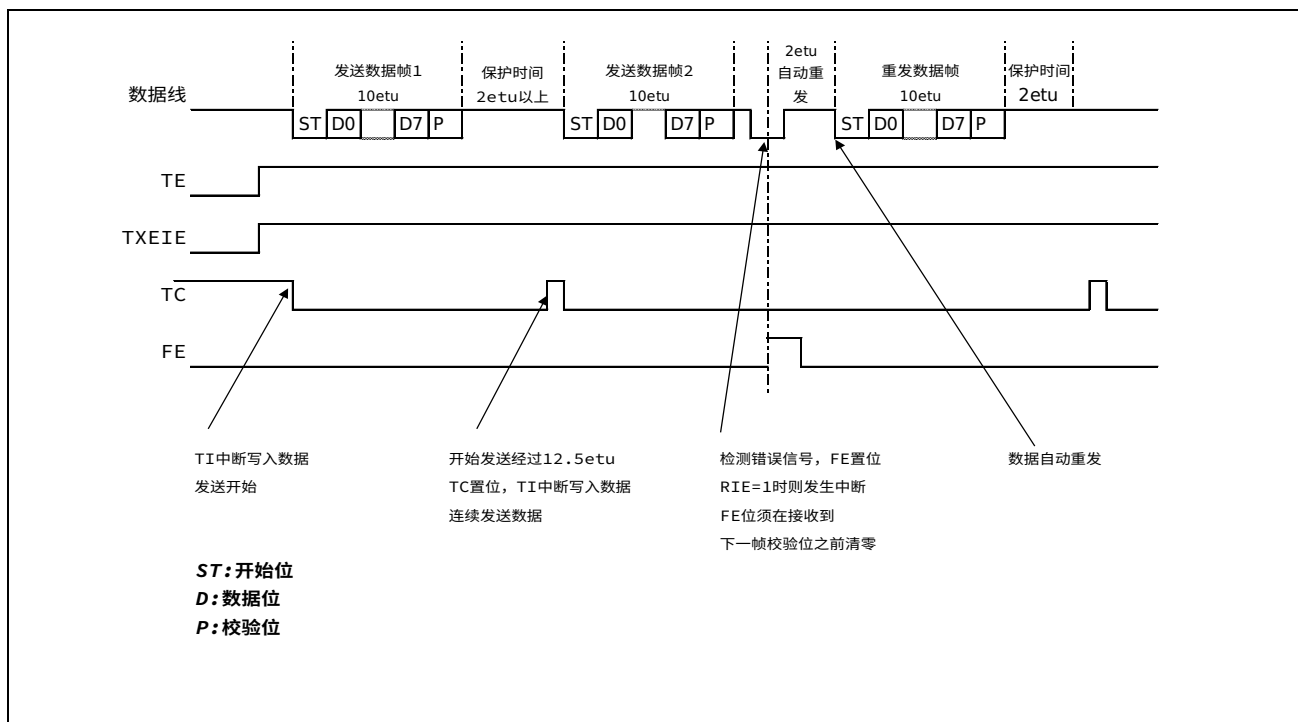


图 26-22 智能卡模式发送数据图例

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。

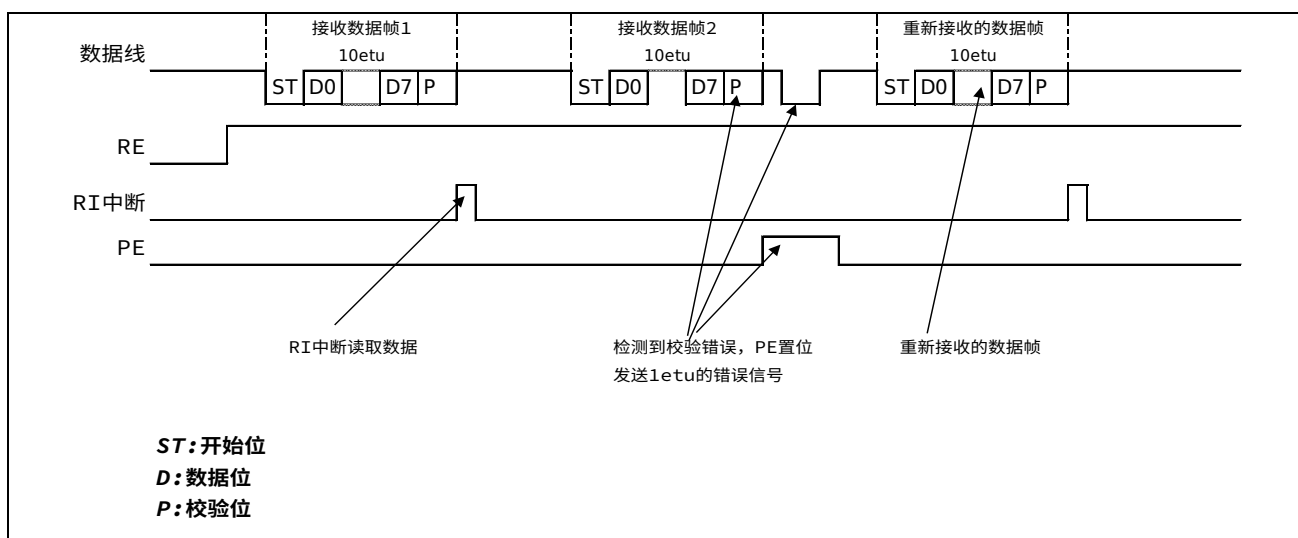


图 26-23 智能卡模式接收数据图例

26.3.5.6 中断和事件

表 26-7 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

26.3.6 时钟同步模式

26.3.6.1 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: 当 USARTn_PR.ULBREN 为 0 时, USARTn_PR.PSC[1:0]位设定的时钟 (PCLK1, PCLK1/4, PCLK1/16, PCLK1/64) 单位: MHz

当 USARTn_PR.ULBREN 为 1 时, USARTn_PR.PSC[1:0]位设定的时钟 (PCLK1/128, PCLK1/256, PCLK1/512, PCLK1/1024) 单位: MHz

DIV_Integer: USARTn_BRR.DIV_Integer 设定值

内部时钟源时, 当 C 为 PCLK1, DIV_Integer=1 时, 最高波特率为 PCLK1/8 (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 PCLK1/6, 所以最高波特率为 PCLK1/6 (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 PCLK1 的计算方法外, 还需要参考**数据手册电气特性**章节规定的最高通信波特率。

26.3.6.2 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。

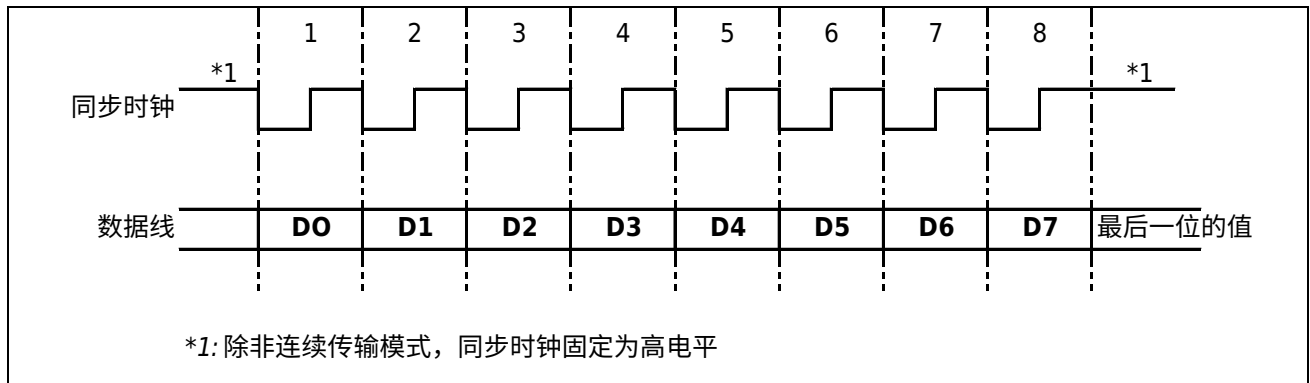


图 26-24 时钟同步模式数据格式

26.3.6.3 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。

26.3.6.4 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.CLK[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。

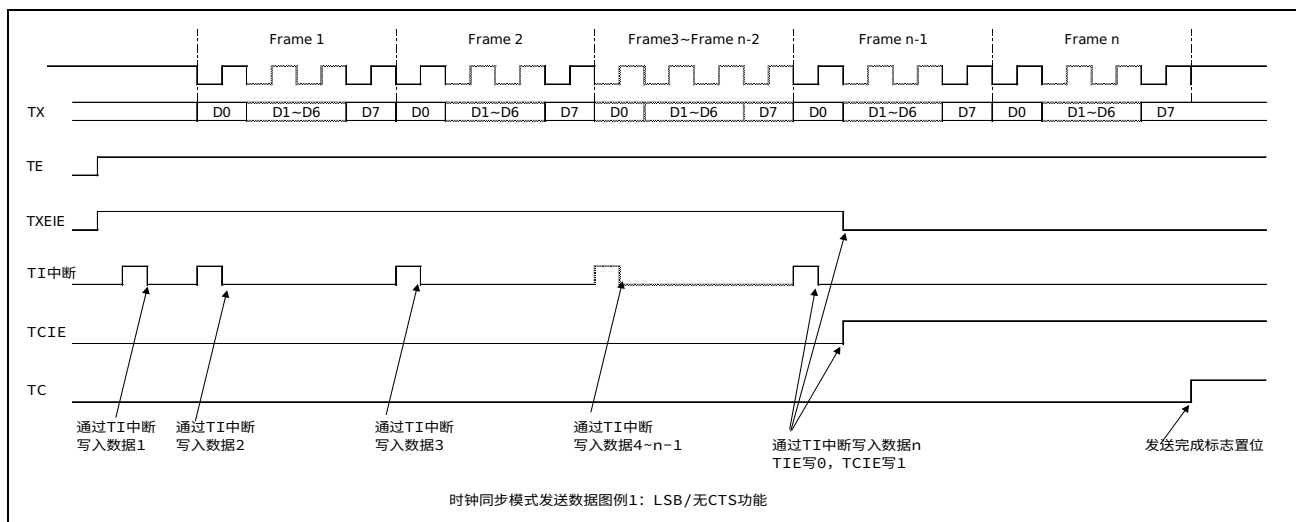


图 26-25 时钟同步模式发送数据图例 1

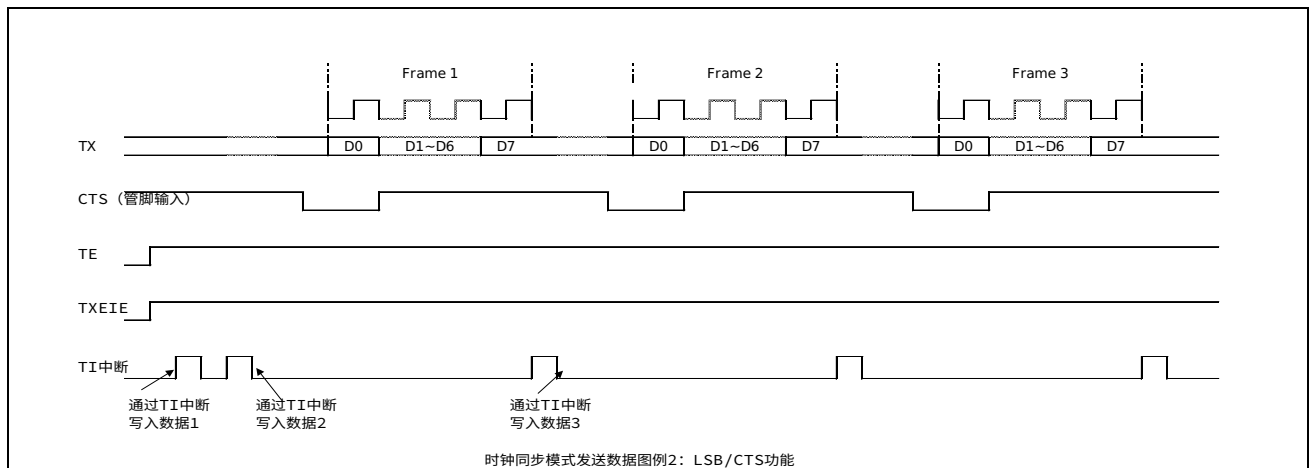


图 26-26 时钟同步模式发送数据图例 2

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。

26.3.6.5 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。
 - 1) 发生上溢错误时，数据丢失并置位 USARTn_SR.ORE 标志
 - 2) 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）

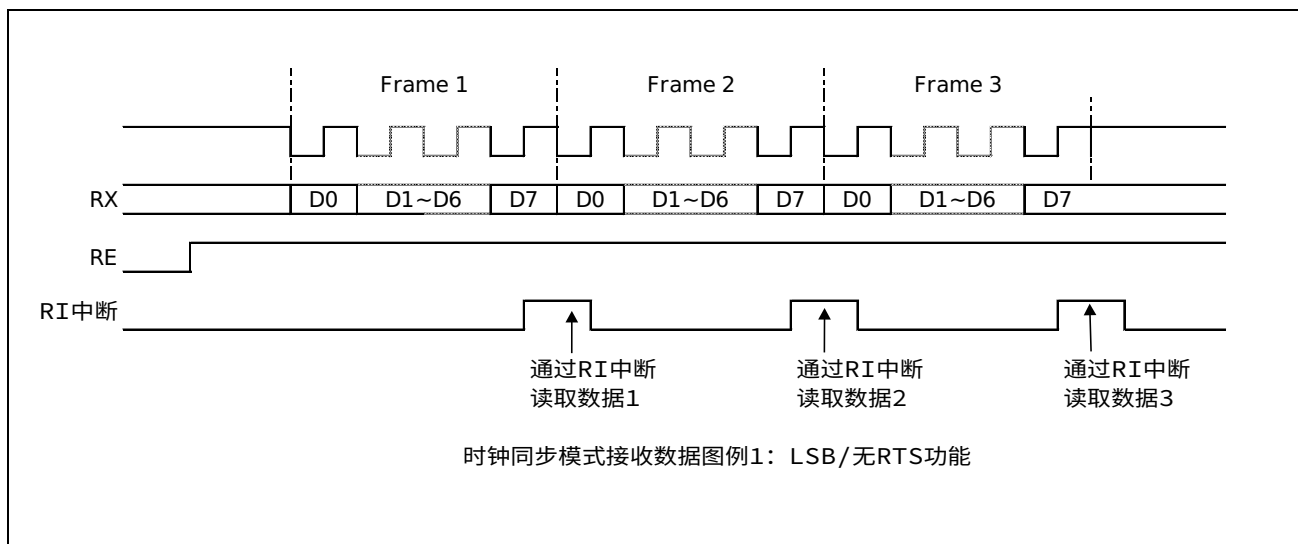


图 26-27 时钟同步模式接收数据图例 1

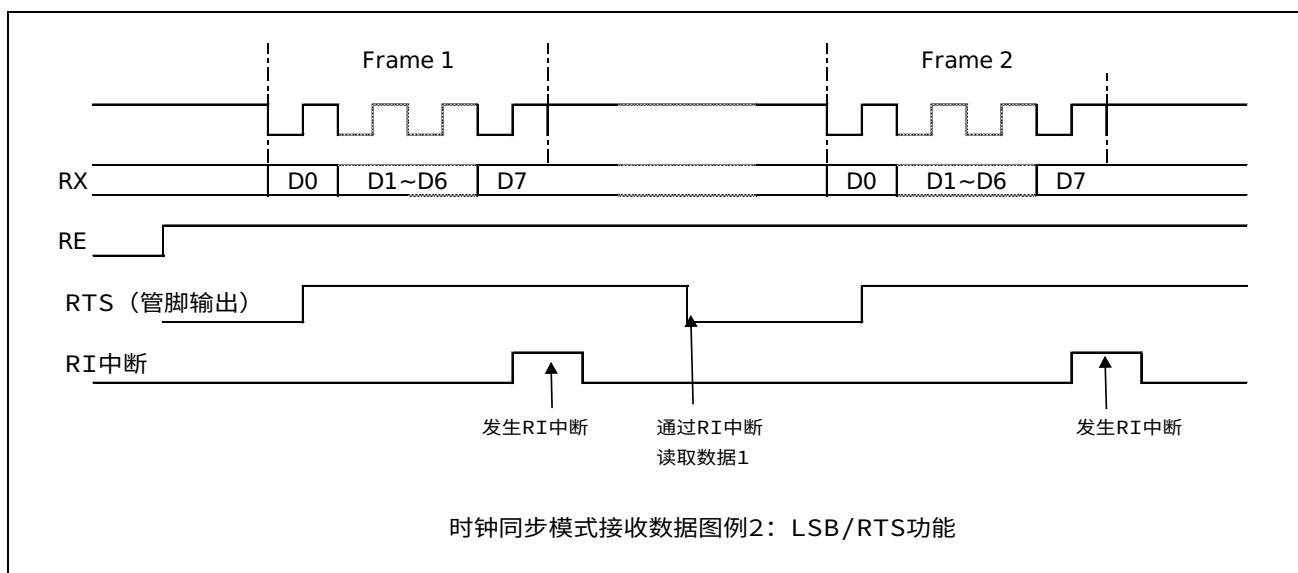


图 26-28 时钟同步模式接收数据图例 2

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。

26.3.6.6 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。

26.3.6.7 时钟同步模式中断和事件

表 26-8 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	可

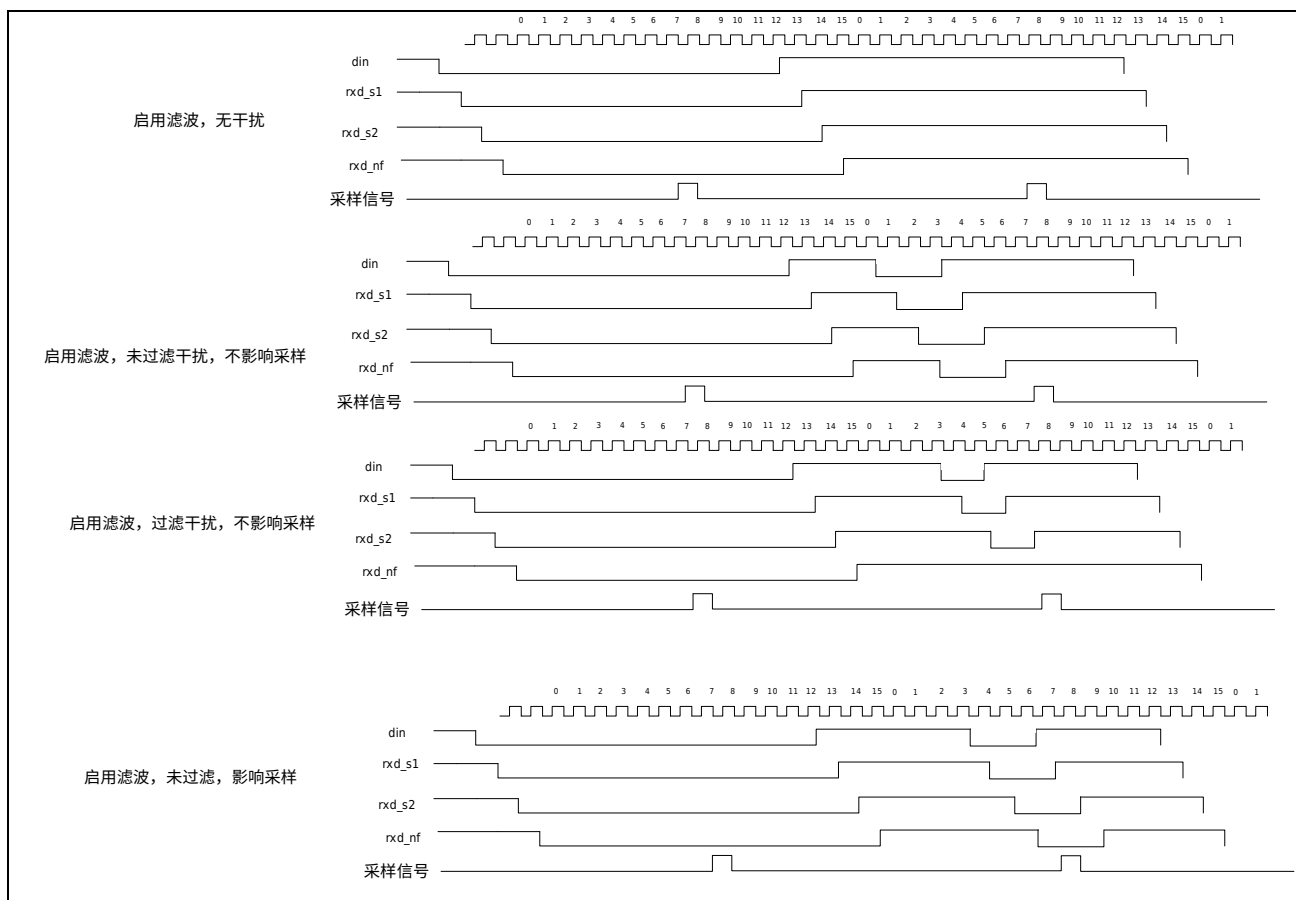
26.3.7 数字滤波功能

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。



打开 NF 功能，实际上是对输入 rd 信号进行了两次延迟操作，得到三个信号。通过这两个延迟，将输入信号与两个延迟信号必须保持一致后，才能在模块内部进行翻转。

26.3.8 中断

下表给出 USART 整体中断:

表 26-9 USART 整体中断一览表

中断名称	符号	说明
错误中断	USART_n_EI n=1~4	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=1~4	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=1~4	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=1~4	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=1~4	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_BRKWKPI n=2~4	LIN 唤醒信号/间隔段检测中断
UART接收TIMEOUT中断	USART_n_RTO n=1~4	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

26.4 寄存器描述

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。

表 26-10 USART 基地址

名称	基地址	描述
USART1	0x4001 CC00	USART1 基地址
USART2	0x4001 D000	USART2 基地址
USART3	0x4001 D400	USART3 基地址
USART4	0x4001 D800	USART4 基地址
USART1_NFC	0x4005 5400	USART1_NFC 基地址

表 26-11 USART 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
USART_SR	状态寄存器	0x00	32	0x0000 00C0
USART_TDR	发送数据寄存器	0x04	16	0x01FF
USART_RDR	接收数据寄存器	0x06	16	0x0000
USART_BRR	波特率寄存器	0x08	32	0x0000 FFFF
USART_CR1	控制寄存器1	0x0C	32	0x8000 0000
USART_CR2	控制寄存器2	0x10	32	0x0000 0600
USART_CR3	控制寄存器3	0x14	32	0x0000 0000
USART_PR	预分频寄存器	0x18	32	0x0000 0000
USART_LBMC	LIN波特率测量计数寄存器	0x1C	32	0x0000 0000

表 26-12 USART1_NFC 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
USART1_NFC	USART1滤波控制寄存器	0x1C	32	0x0000 0000

26.4.1 状态寄存器 (USART_SR)

复位值: 0x0000 00C0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															MPB
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				TENDF	LBD	WKUP	RTOF	TXE	TC	RXNE	BE	ORE	Res	FE	PE

位/位域	标记	位名	功能	读写
b31~b17	Res	保留位	读出时为“0”，写入时写“0”	RW
b16	MPB	多处理器位	多处理器位标志 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	Res	保留位	读出时为“0”，写入时写“0”	RW
b11	TENDF	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 TEND置位条件 • 送出一帧数据的最后一位时, 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	R
b10	LBD	LIN 间隔段检出标志	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.CLBD位写入1	R
b9	WKUP	LIN Wakeup信号标志	LIN 间隔段检出标志 (仅USART_2/USART_3/USART_4有效) 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寄存器设置的宽度后, 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_2/USART_3/USART_4有效) 0: 未检测到Wakeup信号 1: 检测到Wakeup信号	R

			<p>WKUP置位条件</p> <p>CR2.WKUP=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）</p> <p>清零条件</p> <p>清零寄存器CR1.CWKUP位写入1</p> <p>注意：</p> <p>RE=0并不能复位WKUP位</p> <p>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.WKUP=0后，进行后续间隔场检测等操作。</p>	
b8	RTOF	UART接收TIMEOUT标志位	<p>UART接收TIMEOUT标志位（全通道）</p> <p>0：无UART接收TIMEOUT</p> <p>1：发生UART接收TIMEOUT</p> <p>RTOF置位条件</p> <ul style="list-style-type: none"> 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 <p>RTOF清零条件</p> <ul style="list-style-type: none"> 清零寄存器CR1.CRTOF位写入 <p>注意：</p> <p>RTOF为由硬件置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。</p>	R
b7	TXE	发送数据寄存器空	<p>发送数据寄存器空标志</p> <p>TXE位在UART/时钟同步模式/LIN模式时有效。</p> <p>0：数据未传输到移位寄存器，发送数据寄存器非空</p> <p>1：数据传输到移位寄存器，发送数据寄存器空</p> <p>注意：</p> <p>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</p>	R
b6	TC	发送完成标志	<p>发送完成标志位</p> <p>0：发送数据中</p> <p>1：发送数据完成</p> <p>UART模式，时钟同步模式</p> <p>TC置位条件</p> <ul style="list-style-type: none"> TE=0发送禁止时 送出一帧数据的最后一位时，发送数据寄存器的值没有被更新 <p>TC清零条件</p> <ul style="list-style-type: none"> TE=1时，向发送数据寄存器写入发送数据 <p>智能卡模式</p> <p>TC置位条件</p> <ul style="list-style-type: none"> 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。 <p>TC置位的具体时序为：校验位送出后经过2.5位时间</p> <p>TC清零条件</p> <ul style="list-style-type: none"> TE=1时，向发送数据寄存器写入发送数据 <p>注意：</p> <p>TE位由0变化为1时TC为保持为1</p>	R
b5	RXNE	接收数据寄存器不为空	<p>接收数据寄存器不为空标志</p> <p>0：未接收到数据</p> <p>1：准备好读取接收到的数据</p>	R

			<p>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</p>	
b4	BE	LIN总线检错误标志	<p>LIN总线检错误标志 (仅USART_2/USART_3/USART_4有效) 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"> 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) <p>BE清零条件</p> <ul style="list-style-type: none"> 清零寄存器CR1.CBE位写入1 <p>注意: TE=0并不能复位BE位 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>	R
b3	ORE	接收上溢错误	<p>接收上溢错误标志位 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"> 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 <p>ORE清零条件</p> <ul style="list-style-type: none"> 清零寄存器CR1.CORE位写入1 <p>注意: RE=0并不能复位ORE位 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 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	Res	保留位	读出时为“0”, 写入时写“0”	RW
b1	FE	接收帧错误	<p>接收帧错误标志位 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"> 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 <p>FE清零条件</p> <ul style="list-style-type: none"> 清零寄存器CR1.CFE位写入1 <p>注意: UART模式时, RE=0并不能复位FE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"> 采样到低电平错误信号标志 <p>FE清零条件</p> <ul style="list-style-type: none"> 清零寄存器CR1.CFE位写入1 <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 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>	R

- 接收数据发生奇偶校验错误时

PE清零条件

- 清零寄存器CR1.CPE位写入1

注意:

RE=0并不能复位PE位

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

26.4.2 数据寄存器 (USART_TDR)

复位值: 0x01FF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						MPID	TDR[8:0]								

位/位域	标记	位名	功能	读写
b15~b10	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW

26.4.3 数据寄存器 (USART_RDR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						RDR[8:0]									

位/位域	标记	位名	功能	读写
b15~b9	Res	保留位	读出时为“0”，写入时写“0”	RW
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

26.4.4 波特率寄存器 (USART_BRR)

复位值: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DIV_Integer [7:0]								Res	DIV_Fraction[6:0]						

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”，写入时写“0”	RW
b15~b8	DIV_Integer[7:0]	整数分频寄存器	整数分频寄存器 注意: DIV_Integer[7:0]只能在TE=0&RE=0（发送/接收禁止）时设定	RW
b7	Res	保留位	写入时需写入复位值	RW
b6~b0	DIV_Fraction[6:0]	小数分频寄存器	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，而且仅在FBME=1时设定值有效	RW

表 26-13 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	$B = \frac{C}{8 \times (2 - \text{OVER8}) \times (\text{DIV_Integer} + 1)}$	$E(\%) = \left\{ \frac{C}{8 \times (2 - \text{OVER8}) \times (\text{DIV_Integer} + 1) \times B} - 1 \right\} \times 100$
时钟同步模式	$B = \frac{C}{4 \times (\text{DIV_Integer} + 1)}$	-
智能卡模式	$B = \frac{C}{2 \times \text{BCN} \times (\text{DIV_Integer} + 1)}$	$E(\%) = \left\{ \frac{C}{2 \times \text{BCN} \times (\text{DIV_Integer} + 1) \times B} - 1 \right\} \times 100$

B: 波特率 单位: Mbps

C: PR.ULBREN 与 PR.PSC[1:0]位设定的时钟 单位: MHz

BCN: CR3.BCN 寄存器设定值

表 26-14 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	$B = \frac{C \times (128 + \text{DIV_Fraction})}{8 \times (2 - \text{OVER8}) \times (\text{DIV_Integer} + 1) \times 256}$	$E(\%) = \left\{ \frac{C \times (128 + \text{DIV_Fraction})}{8 \times (2 - \text{OVER8}) \times (\text{DIV_Integer} + 1) \times 256 \times B} - 1 \right\} \times 100$
时钟同 步模式	$B = \frac{C \times (128 + \text{DIV_Fraction})}{4 \times (\text{DIV_Integer} + 1) \times 256}$	-
智能卡 模式	$B = \frac{C \times (128 + \text{DIV_Fraction})}{2 \times \text{BCN} \times (\text{DIV_Integer} + 1) \times 256}$	$E(\%) = \left\{ \frac{C \times (128 + \text{DIV_Fraction})}{2 \times \text{BCN} \times (\text{DIV_Integer} + 1) \times 256 \times B} - 1 \right\} \times 100$

B: 波特率 单位: Mbps

C: PR.ULBREN 与 PR.PSC[1:0]位设定的时钟 单位: MHz

BCN: CR3.BCN 寄存器设定值

26.4.5 控制寄存器 1 (USART_CR1)

复位值: 0x8000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SBS	NFE	FBME	ML	Res	CTEND	MS	CLBD	CWKUP	CBE	CRTOF	CORE	Res	CFE	CPE	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OVER8	Res	M	Res	PCE	PS	TENDIE	TXEIE	TCIE	RIE	SLME	TE	RE	RTOIE	RTOE	

位/位域	标记	位名	功能	读写
b31	SBS	UART模式接收数据开始位检测方式设定位	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	RW
b30	NFE	数字滤波使能位	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	RW
b29	FBME	小数波特率功能使能	小数波特率功能使能位 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	RW
b28	ML	MSB/LSB选择位	UART模式/时钟同步模式/智能卡模式时, MSB/LSB方式选择位 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	RW
b27~b26	Res	保留位	读出时为“0”, 写入时写“0”	RW
b25	CTEND	TEND清零位	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, 读取时返回0	
b24	MS	通信模式选择位	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	RW
b23	CLBD	LBD清零位	LBD清零位 (仅USART_2/USART_3/USART_4有效) 0:不清零LBD标志 1:清零LBD标志 注意:	RW

			CLBD位写1清除LBD标志，读取时返回0	
b22	CWKUP	WKUP清零位	<p>WKUP清零位（仅USART_2/USART_3/USART_4有效）</p> <p>0:不清零WKUP标志</p> <p>1:清零WKUP标志</p> <p>注意：</p> <p>CWKUP位写1清除WKUP标志，读取时返回0</p>	RW
b21	CBE	BE清零位	<p>BE清零位（仅USART_2/USART_3/USART_4有效）</p> <p>0:不清零BE标志</p> <p>1:清零BE标志</p> <p>注意：</p> <p>CBE位写1清除BE标志，读取时返回0</p>	RW
b20	CRTOF	RTOF清零位	<p>RTOF清零位</p> <p>0:不清零RTOF标志</p> <p>1:清零RTOF标志</p> <p>注意：</p> <p>CRTOF位写1清除RTOF标志，读取时返回0</p>	RW
b19	CORE	ORE标志清零位	<p>ORE标志清零位</p> <p>0:不清零ORE标志</p> <p>1:清零ORE标志</p> <p>注意：</p> <p>CORE位写1清除ORE标志，读取时返回0</p>	RW
b18	Res	保留位	读出时为“0”，写入时写“0”	RW
b17	CFE	FE标志清零位	<p>FE标志清零位</p> <p>0:不清零FE标志</p> <p>1:清零FE标志</p> <p>注意：</p> <p>CFE位写1清除FE标志，读取时返回0</p>	RW
b16	CPE	PE标志清零位	<p>PE标志清零位</p> <p>0:不清零PE标志</p> <p>1:清零PE标志</p> <p>注意：</p> <p>CPE位写1清除FE标志，读取时返回0</p>	RW
b15	OVER8	UART过采样模式	<p>UART过采样模式设定，即一位数据传输期间的基本时钟数</p> <p>0: 16位</p> <p>1: 8位</p> <p>注意：</p> <p>非UART模式时OVER8位必须保持复位值</p> <p>OVER8位只能在TE=0&RE=0（发送/接收禁止）时设定</p>	RW
b14~b13	Res	保留位	读出时为“0”，写入时写“0”	RW
b12	M	数据长度设定位	<p>UART模式时，发送/接收数据长度设定位</p> <p>0: 8位</p> <p>1: 9位</p> <p>注意：</p> <p>非UART模式时M位必须保持复位值</p> <p>M位只能在TE=0&RE=0（发送/接收禁止）时设定</p>	RW
b11	Res	保留位	读出时为“0”，写入时写“0”	RW
b10	PCE	校验使能位	UART模式时，奇偶校验使能位	RW

			<p>0: 无校验 1: 校验</p> <p>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</p>	
b9	PS	校验位	<p>UART模式时, 奇偶校验选择位</p> <p>0: 偶校验 1: 奇校验</p> <p>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</p>	RW
b8	TENDIE	发送完成中断使能位	<p>发送完成中断使能位</p> <p>0: TENDI中断请求无效, TENDI中断不发生 1: TENDI中断请求使能, TENDI中断发生</p>	RW
b7	TXEIE	发送数据寄存器空中断使能位	<p>发送数据寄存器空中断使能位</p> <p>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</p> <p>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</p>	RW
b6	TCIE	发送完成中断使能位	<p>发送完成中断使能位</p> <p>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</p> <p>注意: 非智能卡模式读取有效, 智能卡模式读取为0</p>	RW
b5	RIE	接收中断使能位	<p>接收中断使能位</p> <p>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</p>	RW
b4	SLME	静默模式使能位	<p>多处理器操作时, 静默模式使能位</p> <p>0: 正常模式 1: 静默模式</p> <p>SLME=1时, MPB位为0的通信数据不会从接收移位寄存器读取到接收数据寄存器, 同时错误标志ORE和FE位也不置位。接收到MPB为1的ID数据时, SLME自动清零, 开始正常的数据接收动作。</p> <p>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。</p>	RW
b3	TE	发送器使能位	<p>发送器使能位</p> <p>0: 发送器禁止 1: 发送器使能</p> <p>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。</p>	RW
b2	RE	接收器使能位	<p>接收器使能位</p> <p>0: 接收器禁止 1: 接收器使能</p> <p>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</p>	RW
b1	RTOIE	UART TIMEOUT中断使能位	<p>UART TIMEOUT中断使能位</p> <p>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</p>	RW

b0	RTOE	UART TIMEOUT功 能使能位	UART TIMEOUT功能使能位 0: UART TIMEOUT功能禁止 1: UART TIMEOUT功能使能	RW
----	------	-----------------------	---	----

26.4.6 控制寄存器 2 (USART_CR2)

复位值: 0x0000 0600

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res													SBKM	SBK	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	LINEN	STOP	CLKC[1:0]	Res	WKUPE	SBKL[1:0]	LBDL	LBDIE	BEE	BEIE	WKUPE	MPE			

位/位域	标记	位名	功能	读写
b31~b18	Res	保留位	读出时为“0”，写入时写“0”	RW
b17	SBKM	间隔段发送模式	间隔段发送模式选择（仅USART_2/USART_3/USART_4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	RW
b16	SBK	间隔段发送使能位	间隔段发送使能位（仅USART_2/USART_3/USART_4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	RW
b15	Res	保留位	读出时为“0”，写入时写“0”	RW
b14	LINEN	LIN功能使能	UART模式时，LIN功能使能位（仅USART_2/USART_3/USART_4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	RW
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	RW
b12~b11	CLKC[1:0]	时钟控制位	UART模式 0b00: 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 01: 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 10, 11: 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） 时钟同步模式 0b00 or 0b01: 时钟源为内部波特率生成器生成的时钟，输出到USARTn_CK管脚 0b10 or 0b11: 时钟源为外部输入时钟，输入时钟的频率和波特率相同	RW

智能卡模式			
00: 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			
01: 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			
其他: 禁止设定			
注意: CLKC[1:0]位只能在TE=0&RE=0 (发送/接收禁止) 时设定			
b10~b9	Res	保留位	需保持复位值0b11, 写入时写0b11 RW
b8	WKUPE	LIN唤醒信号检测使能位	LIN唤醒信号检测使能位 (仅USART_2/USART_3/USART_4有效) 0: 禁止唤醒信号检测 1: 使能唤醒信号检测 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 RW
b7~b6	SBKL[1:0]	LIN间隔段低电平宽度设定位	LIN间隔段低电平宽度设定位 (仅USART_2/USART_3/USART_4有效) 00: 10位 01: 11位 10: 13位 11: 14位 注意: 非UART-LIN模式时SBKL位必须保持复位值 SBKL位只能在TE=0&RE=0 (发送/接收禁止) 时设定 RW
b5	LBDL	LIN间隔段低电平宽度检测阈值设定位	LIN间隔段低电平宽度检测阈值设定位 (仅USART_2/USART_3/USART_4有效) 0: ≥10位 1: ≥11位 注意: 非UART-LIN模式时LBDL位必须保持复位值 LBDL位只能在TE=0&RE=0 (发送/接收禁止) 时设定 RW
b4	LBDIE	LIN间隔段检测中断使能位	LIN间隔段检测中断使能位 (仅USART_2/USART_3/USART_4有效) 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 注意: 非UART-LIN模式时LBDIE位必须保持复位值 RW
b3	BEE	LIN总线错误检测功能使能位	LIN总线错误检测功能使能位 (仅USART_2/USART_3/USART_4有效) 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 注意: 非UART-LIN模式时BEE位必须保持复位值 BEE位只能在TE=0&RE=0 (发送/接收禁止) 时设定 RW
b2	BEIE	LIN总线错误中断使能位	LIN总线错误中断使能位 (仅USART_2/USART_3/USART_4有效) 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 注意: 非UART-LIN模式时BEIE位必须保持复位值 RW
b1	WKUPIE	LIN唤醒信号检测中断使能位	LIN唤醒信号检测中断使能位 (仅USART_2/USART_3/USART_4有效) 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 注意: RW

			非UART-LIN模式WKUPIE位必须保持复位值	
			UART模式时，多处理器功能使能位	
			0: 禁止	
b0	MPE	多处理器功能使能位	1: 使能	RW
			注意:	
			非UART模式时MPE位必须保持复位值	
			MP位只能在TE=0&RE=0（发送/接收禁止）时设定	

26.4.7 控制寄存器 3 (USART_CR3)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DEP	DEM	Res	DEA[3:0]				Res	BCN[2:0]			Res				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b15	b14
Res		DED[3:0]				CTSE	RTSE	Res		SCEN	LOOP	HDSEL	Res		

位/位域	标记	位名	功能	读写
b31	DEP	DE功能极性	Driver enable极性模式 0: usart在IDLE时DE信号为高 1: usart在IDLE时DE信号为低 启动USART TE或者RE时, 无法写入此位字段	RW
b30	DEM	DE功能使能	启用driver enable模式 该位用于通过DE信号激活外部收发器控制 0: DE功能被禁用 1: DE功能启用 启用USART TE或者RE时, 禁止写入此位字段	RW
b29	Res	保留位	读出时为“0”, 写入时写“0”	RW
b28~b25	DEA[3:0]	DE断言启用时间	驱动程序启用断言时间 这些位用于定义DE (驱动器启用) 信号激活与起始位开始之间的时间, 它以采样时间单位 (1/8或者1/16位时间) 表示, 由OVER8位配置, 最大可配置0x7或者0xF。 启用USART TE或者RE时, 禁止写入此位字段。 最低配置为0, 配置为0时, DE信号与开始位同时发生; 配置为0x7或者0xF时, DE信号发生在开始位前7或者15个时间单位	RW
b24	Res	保留位	读出时为“0”, 写入时写“0”	RW
b23~b21	BCN[2:0]	基本时钟数	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 000 32 001 64 010 禁止设定 011 128 100 禁止设定 101 256 110 372 111 禁止设定 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_2/USART_3/USART_4不支持该功能, 读出为0	RW
b20~b14	Res	保留位	读出时为“0”, 写入时写“0”	RW
b13~b10	DED	DE断言取消时间	驱动程序启用取消断言时间 这些位用于定义传输消息中最后一个停止位的结束与DE (驱动器启用) 信号的停用之间。它以采样时间单位 (1/8或1/16位时间) 表示, 由OVER8位配置, 最大可配置0x7或者0xF 启用USART TE或者RE时, 禁止写入此位字段 最小可配置为0, 配置为0时, DE信号变换发生在结束位完成后的2个时间单位	RW

			后；配置为0x7或者0xF时，DE信号发生在结束位完成后的9或者17个时间单位	
b9	CTSE	CTS功能使能位	<p>CTS功能使能位</p> <p>0: CTS功能无效</p> <p>1: CTS功能有效</p> <p>注意:</p> <p>CTSE位只能在TE=0&RE=0（发送/接收禁止）时设定</p>	RW
b8	RTSE	RTS功能使能位	<p>RTS功能使能位</p> <p>0: RTS功能无效</p> <p>1: RTS功能有效</p> <p>注意:</p> <p>RTSE位只能在TE=0&RE=0（发送/接收禁止）时设定</p>	RW
b7~b6	Res	保留位	读出时为“0”，写入时写“0”	RW
b5	SCEN	智能卡模式使能位	<p>智能卡模式使能位</p> <p>0: 禁止智能卡模式</p> <p>1: 使能智能卡模式</p> <p>注意:</p> <p>非智能卡模式时SCEN位必须保持复位值</p> <p>SCEN位只能在TE=0&RE=0（发送/接收禁止）时设定</p> <p>USART_2/USART_3/USART_4不支持该功能，读出为0</p>	RW
b4	LOOP	LIN 回环模式使能位	<p>LIN 回环模式使能位（仅USART_2/USART_3/USART_4有效）</p> <p>0: 正常模式</p> <p>1: 回环模式</p> <p>注意:</p> <p>LOOP位只能在TE=0&RE=0（发送/接收禁止）时设定</p>	RW
b3	HDSEL	UART单线半双工模式使能位	<p>UART单线半双工模式使能位</p> <p>0: UART全双工模式</p> <p>1: UART半双工模式</p> <p>注意:</p> <p>HDSEL位只能在TE=0&RE=0（发送/接收禁止）时设定</p>	RW
b2~b0	Res	保留位	读出时为“0”，写入时写“0”	RW

26.4.8 预分频寄存器 (USART_PR)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b5	Res	保留位	读出时为“0”，写入时写“0”	RW
b4	ULBREN	超低波特率使能位	内部时钟源时，超低波特率功能使能位 0: 超低波特率功能无效 1: 超低波特率功能有效 且根据PSC[1:0]的值预分频器分频值为： 00: PCLK1/128 01: PCLK1/256 10: PCLK1/512 11: PCLK1/1024 注意： ULBREN位只能在TE=0&RE=0（发送/接收禁止）时设定	
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_2/3/4有效） 00: PCLK1 01: PCLK1/2 10: PCLK1/4 11: PCLK1/8 注意： LBMPSC[1:0]位只能在TE=0&RE=0（发送/接收禁止）时设定	RW
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: PCLK1 01: PCLK1/4 10: PCLK1/16 11: PCLK1/64 注意： PSC[1:0]位只能在TE=0&RE=0（发送/接收禁止）时设定	RW

26.4.9 LIN 波特率测量计数寄存器 (USART_LBMC)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”，写入时写“0”	RW
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_2/USART_3/USART_4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

26.4.10 USART1 滤波控制寄存器 (USART1_NFC)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													USART1_NFE	USASRT1_NFS[1:0]	

位/位域	标记	位名	功能	读写
b31~b3	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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 00: 滤波宽度档位1 01: 滤波宽度档位2 10: 滤波宽度档位3 11: 滤波宽度档位4 各档位具体值请参考数据手册电气特性USART1 STOP模式下RX滤波特性章节。	RW

26.5 注意事项

26.5.1 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。

26.5.2 时钟同步模式注意事项

- 1) 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。
- 2) 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。

26.5.3 其他注意事项

- 1) 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

27 集成电路总线（I2C）

27.1 概述

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。

27.2 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。支持 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
- 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
- 2 级发送 FIFO 和 2 级接收 FIFO

27.3 功能说明

27.3.1 I2C 系统框图

27.3.1.1 系统框图

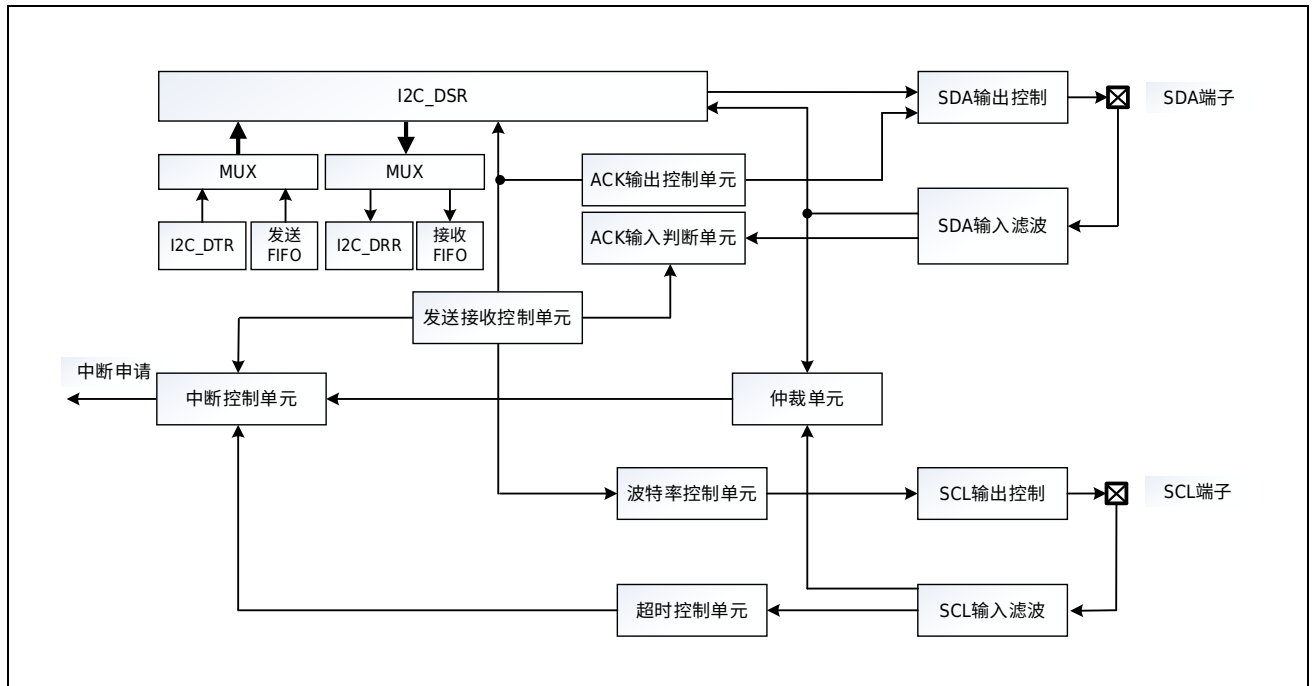


图 27-1 I2C 系统框图

27.3.1.2 结构图

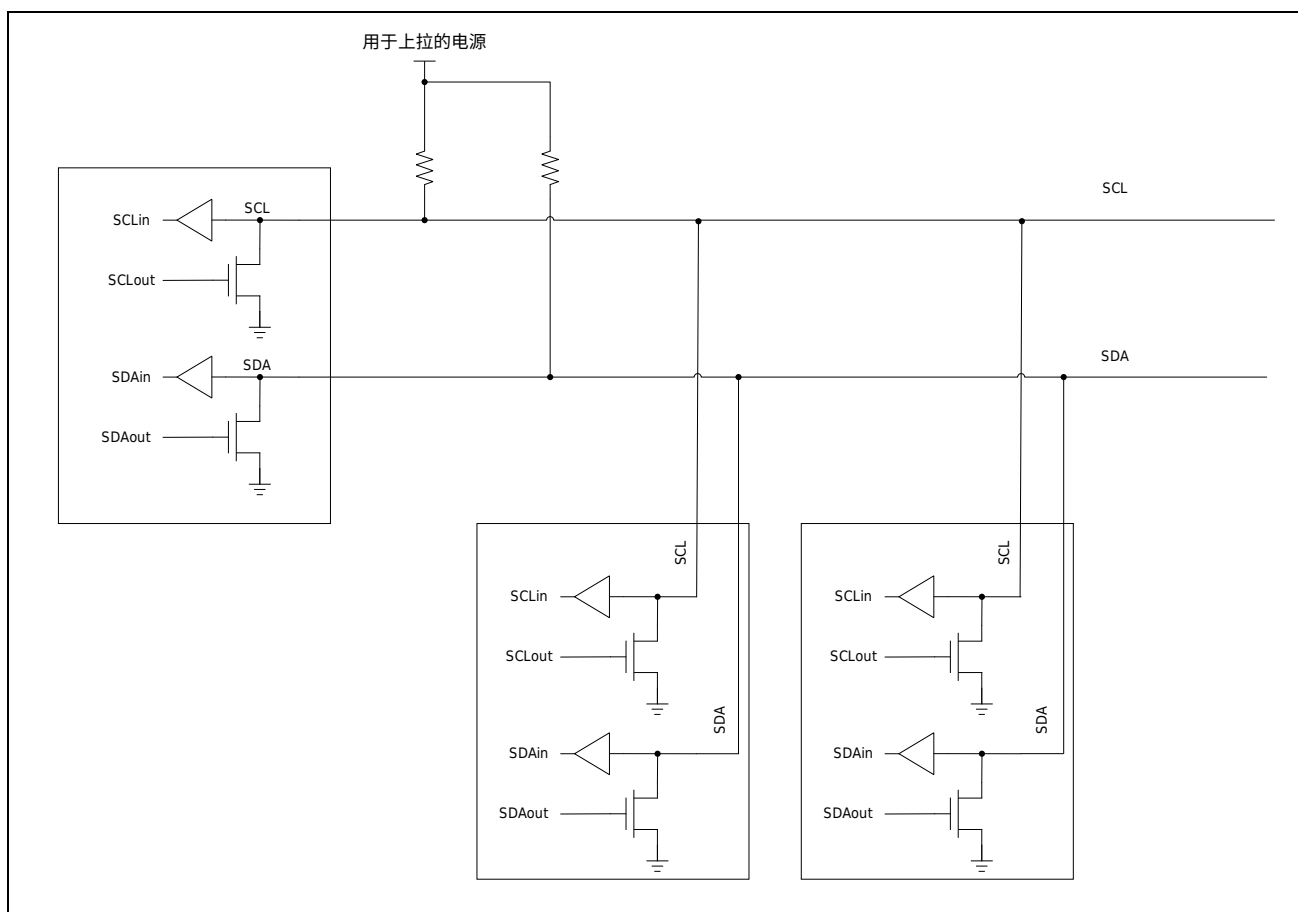


图 27-2 I2C 总线的结构例

表 27-1 输入/输出引脚

引脚名	输入/输出	功能
SCL	输入/输出	串行时钟的输入/输出引脚
SDA	输入/输出	串行数据的输入/输出引脚

当选择 I2C 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。

27.3.2 I2C 协议

I2C 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 I2C 总线的时序图。

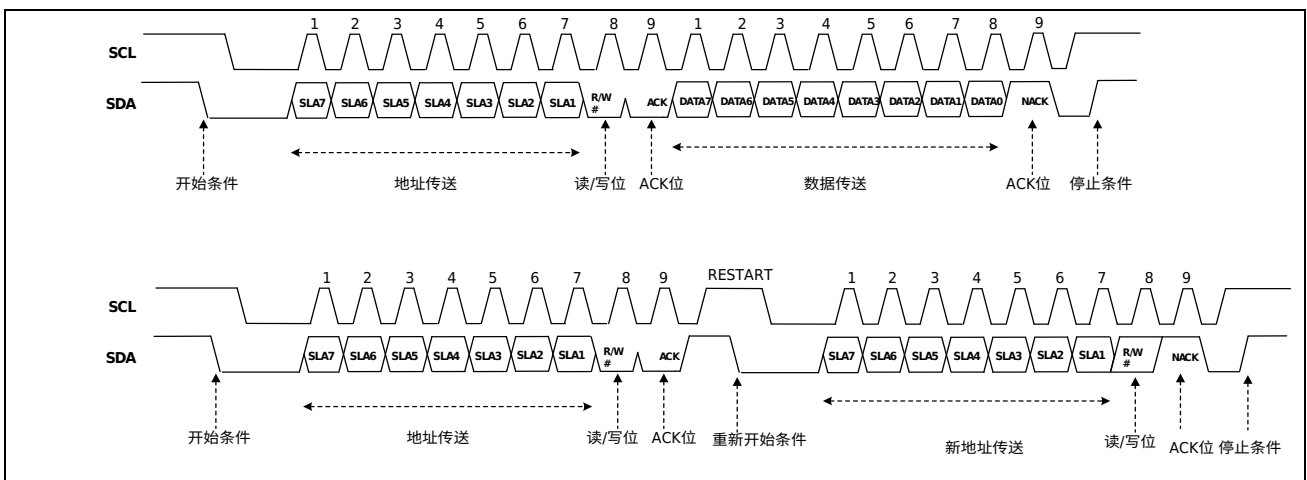


图 27-3 I2C 总线的时序图

27.3.2.1 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 I2C_SR.BUSY 标志为 0 (总线空闲) 的状态下如果将 START 位置 1，就发送开始条件。如果检测到开始条件，就自动将 I2C_SR.BUSY 标志和 I2C_SR.STARTF 标志置 1，并且自动将 START 位清 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 位正确地发送了开始条件，再将 I2C_SR.MSL 位和 I2C_SR.TRA 位自动置 1 后变为主控发送模式。另外，I2C_SR.TEMPTYF 因 TRA 位为 1 而自动变为 1。接下来把从机地址写入 I2C_DTR 寄存器，发送地址。

27.3.2.2 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。

1) 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

2) 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 (0b11110XX0，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。

主机接收模式，主机第一帧发送头序列 (0b11110XX0，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 (0b11110XX1，其中 XX 表示 10 位地址的高两位)。

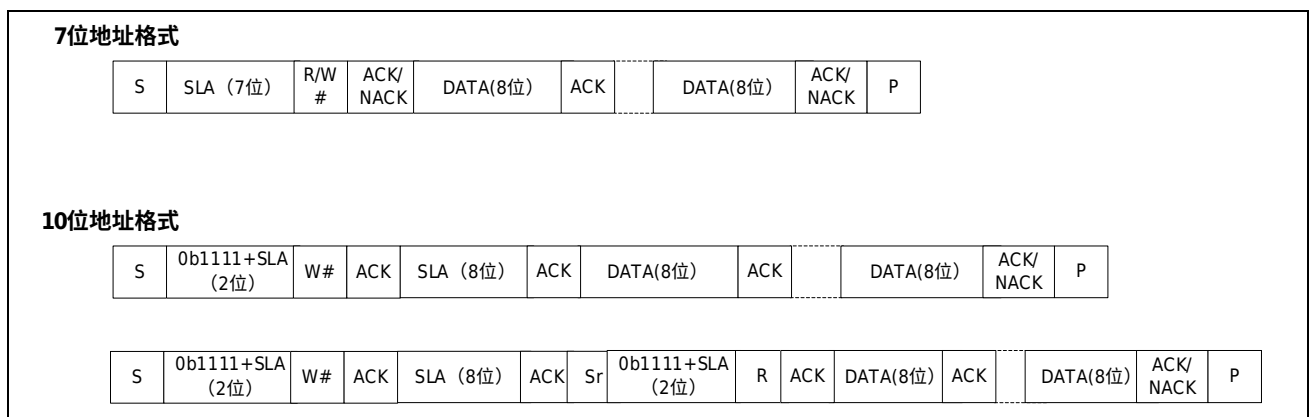


图 27-4 I2C 总线的数据格式

S: 表示开始条件

SLA: 表示从机地址

RW#: 表示发送和接收的方向。当 RW#为 1 时，将数据从从机发送到主机；当 RW#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

27.3.2.3 数据传送

地址匹配一致后，总线上的主机根据 RW 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：

- 1) 发送停止条件释放总线
- 2) 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如下图所示。

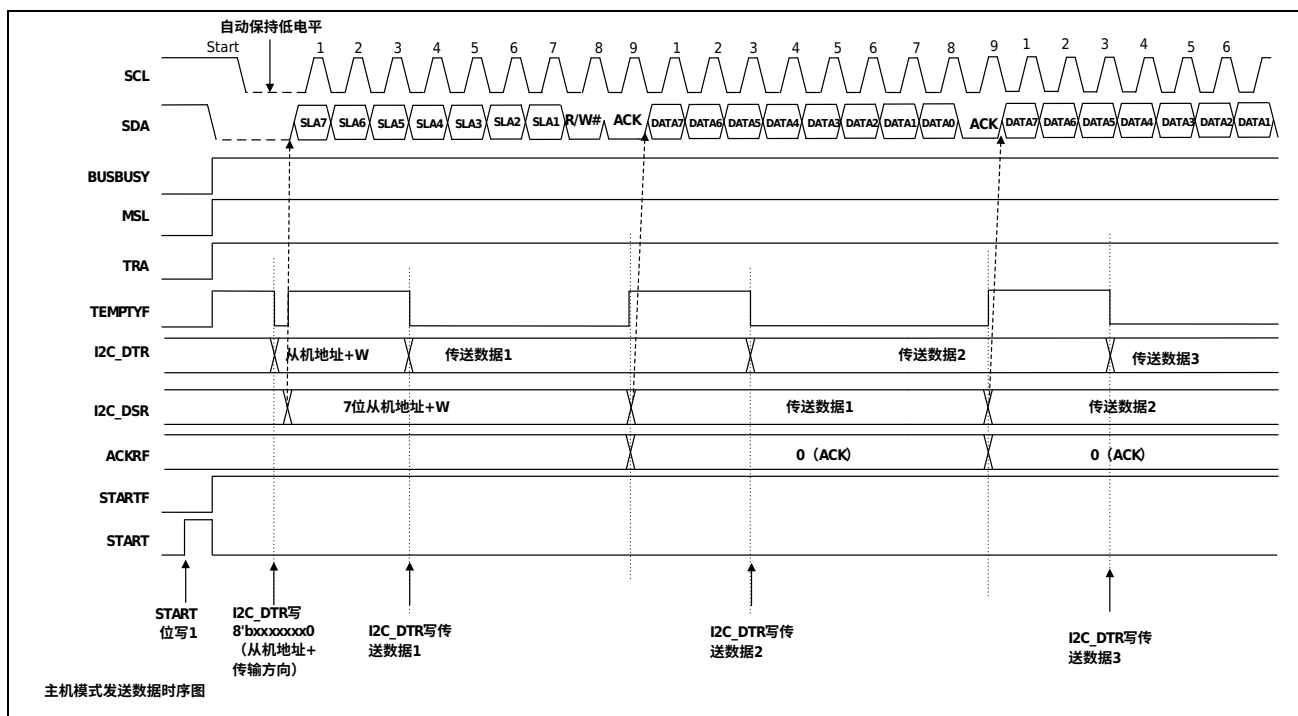


图 27-5 7 位地址格式的主机发送数据时序图 (例)

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。

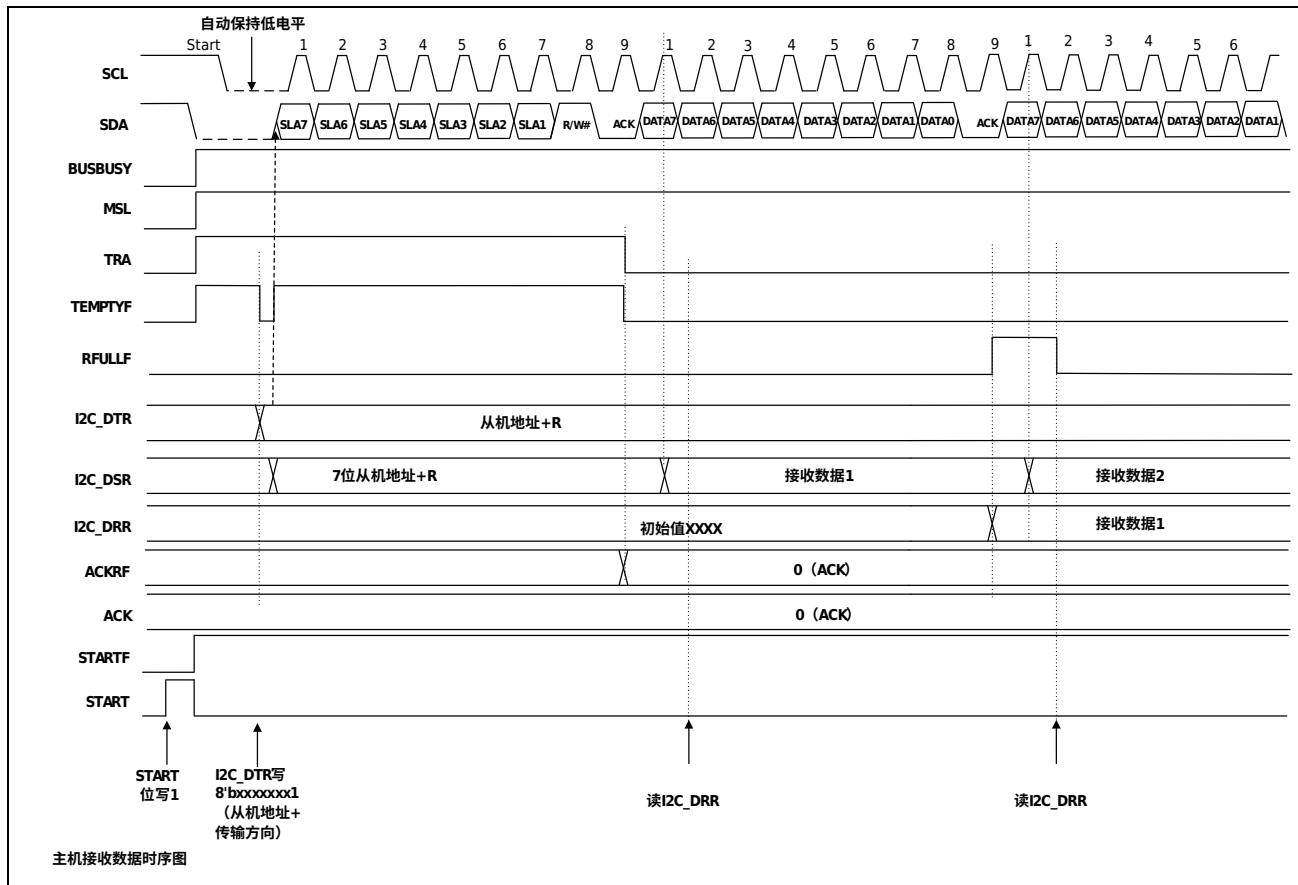


图 27-6 7 位地址格式的主机接收数据的时序图 (例)

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。

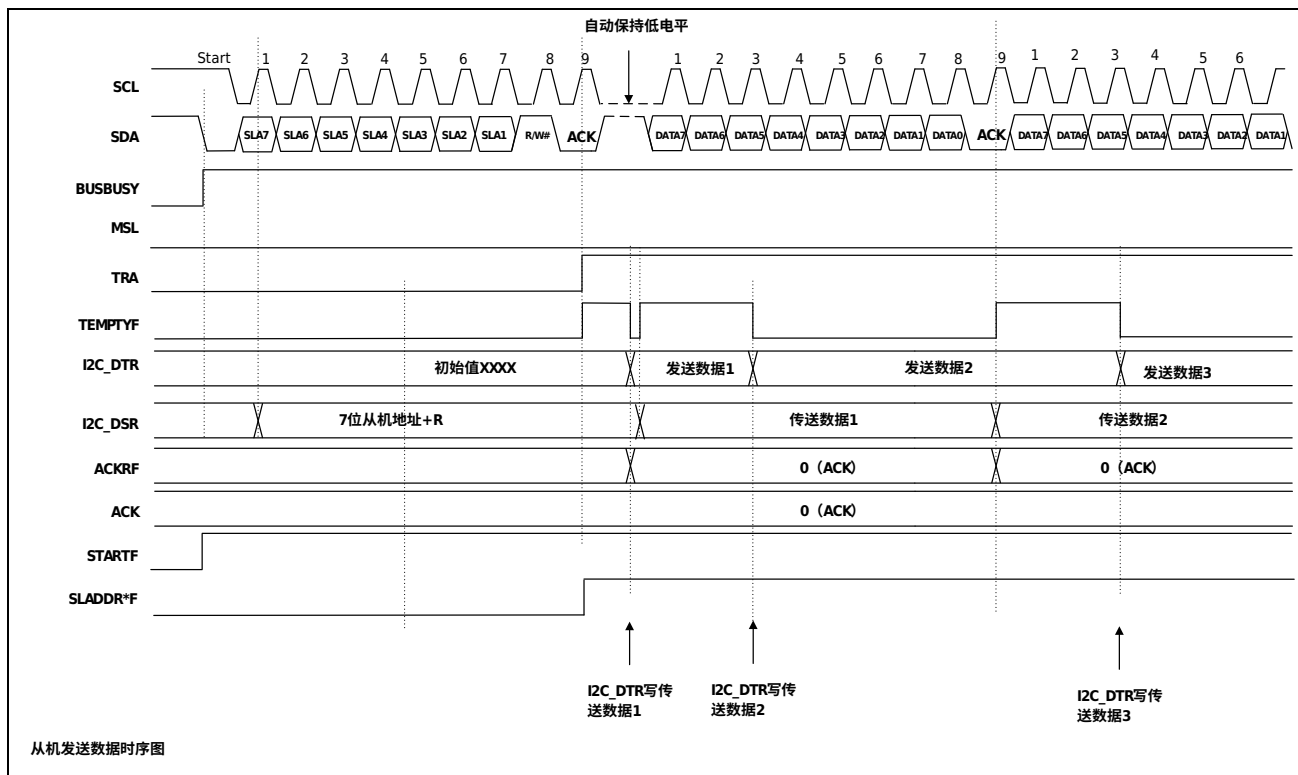


图 27-7 7 位地址格式的从机发送模式时序图（例）

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。

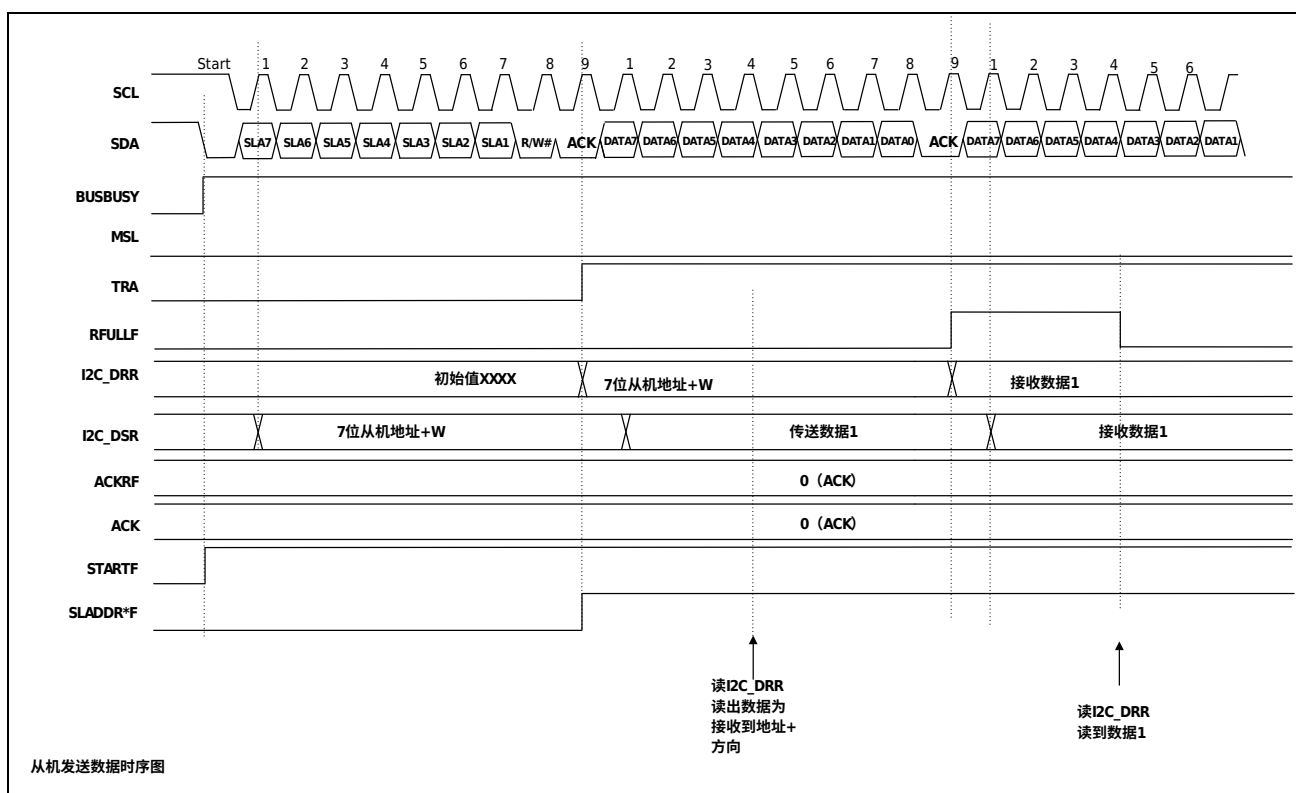


图 27-8 7 位地址格式从机接收模式时序图 (例)

27.3.2.4 停止条件

通过 I2C_CR1.STOP 位发送停止条件。

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, 就发送停止条件。

27.3.2.5 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。

27.3.2.6 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。

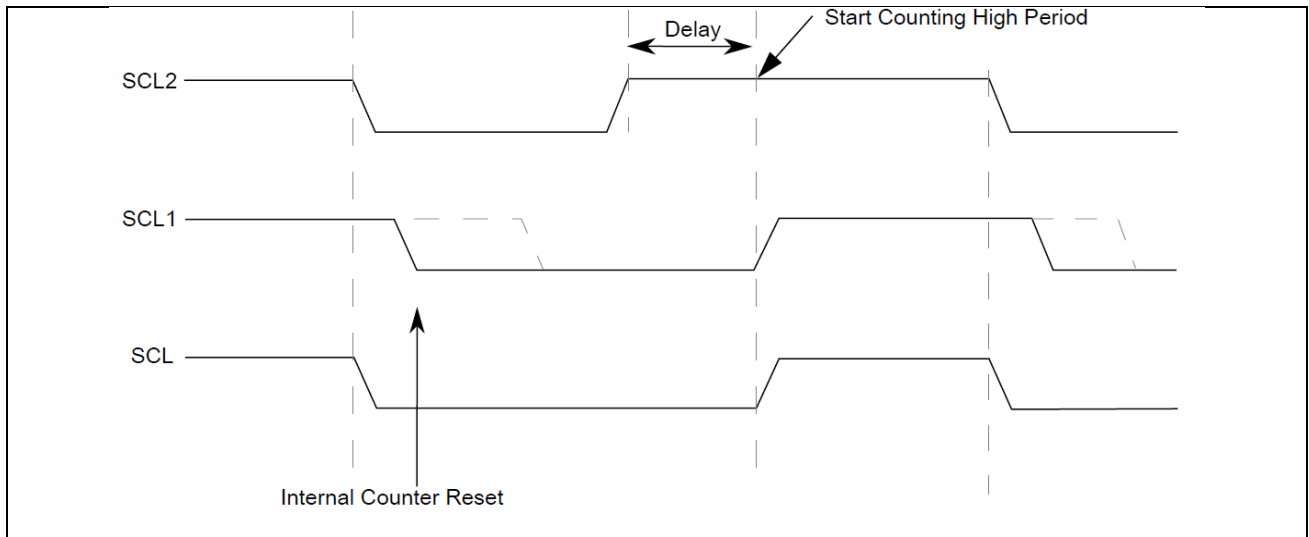


图 27-9 SCL 同步时序

27.3.2.7 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。

27.3.2.8 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】

- 1) 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（I2CDT 寄存器），就在第 9 个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。

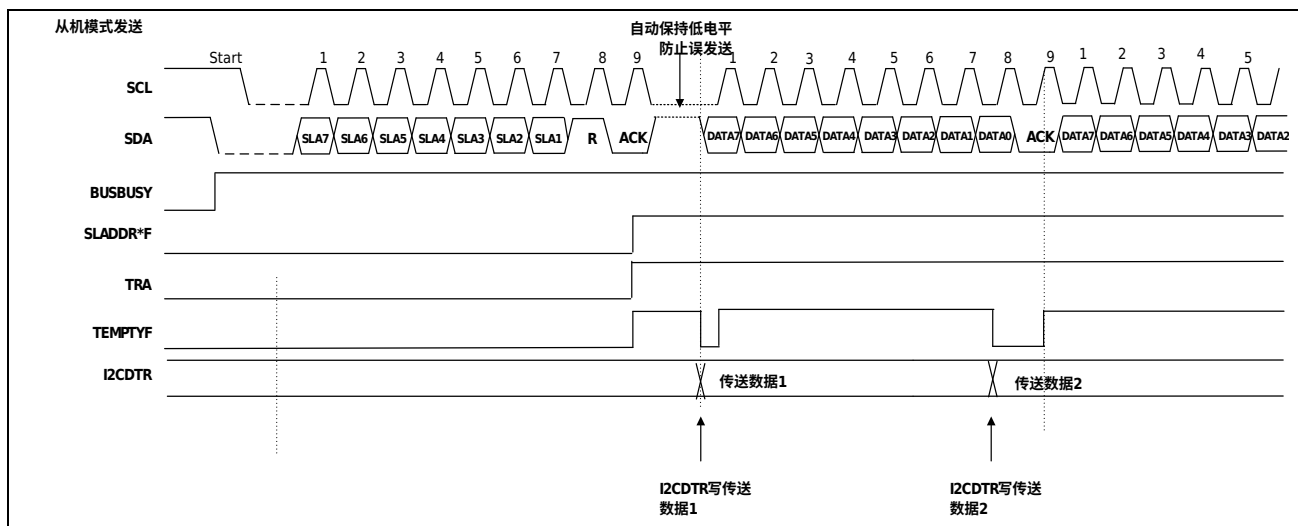


图 27-10 从机发送时序图 (1)

- 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 IC_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。

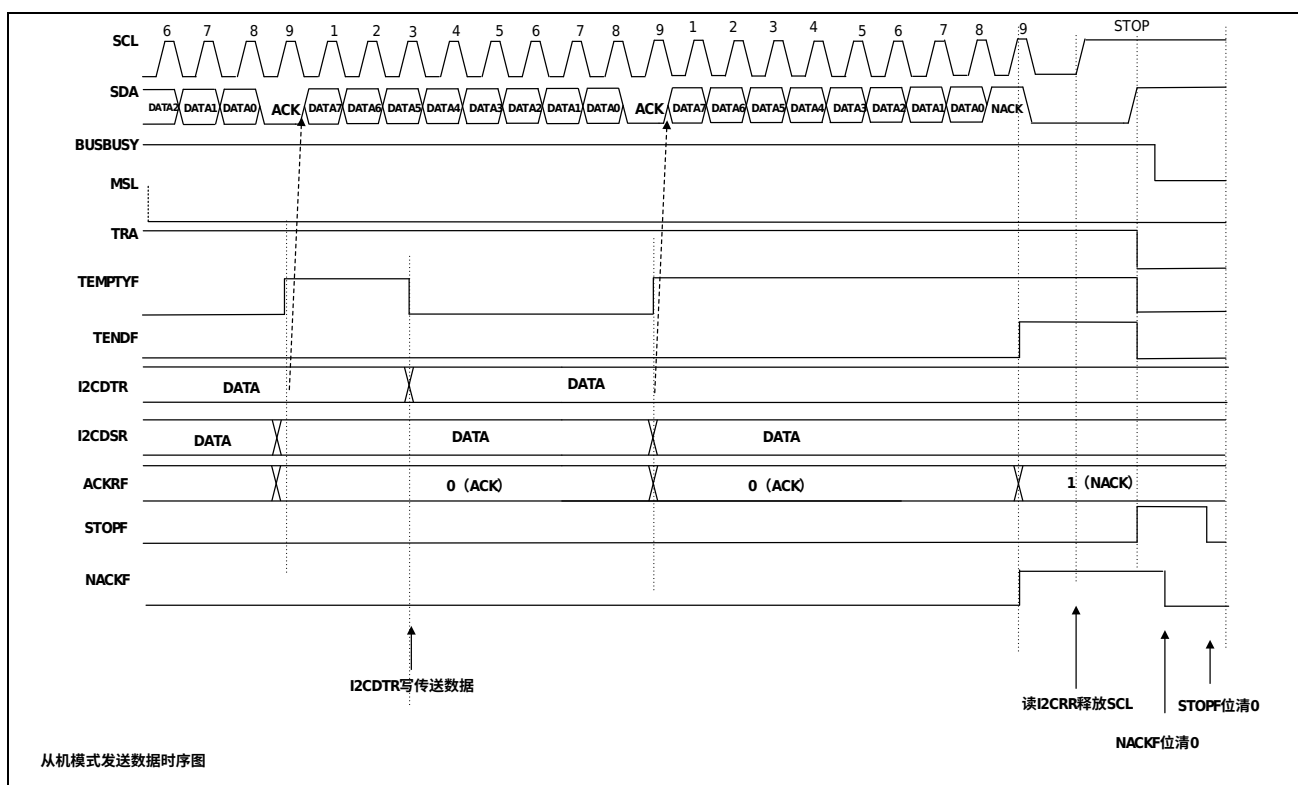


图 27-11 从机发送时序图 (2)

【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。

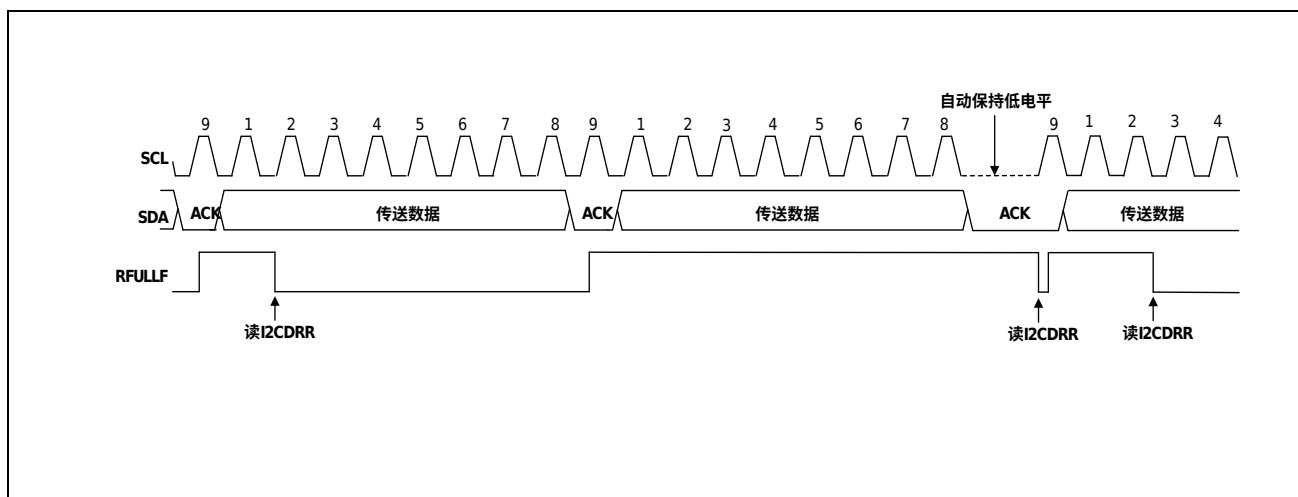


图 27-12 从机接收时序图

【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。

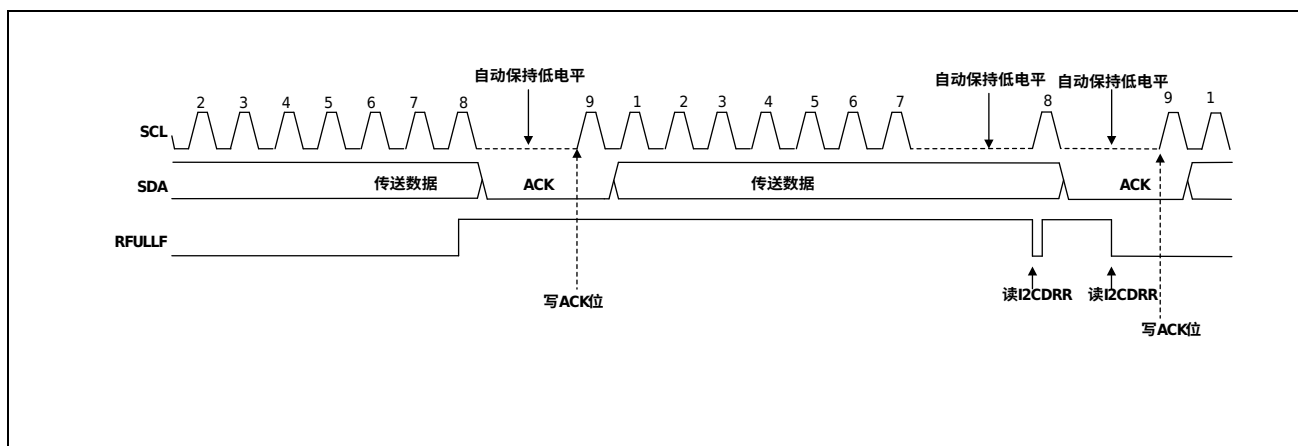


图 27-13 快速 ACK/NACK 时序图

27.3.3 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。

27.3.3.1 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，然后根据后续的数据位将 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。

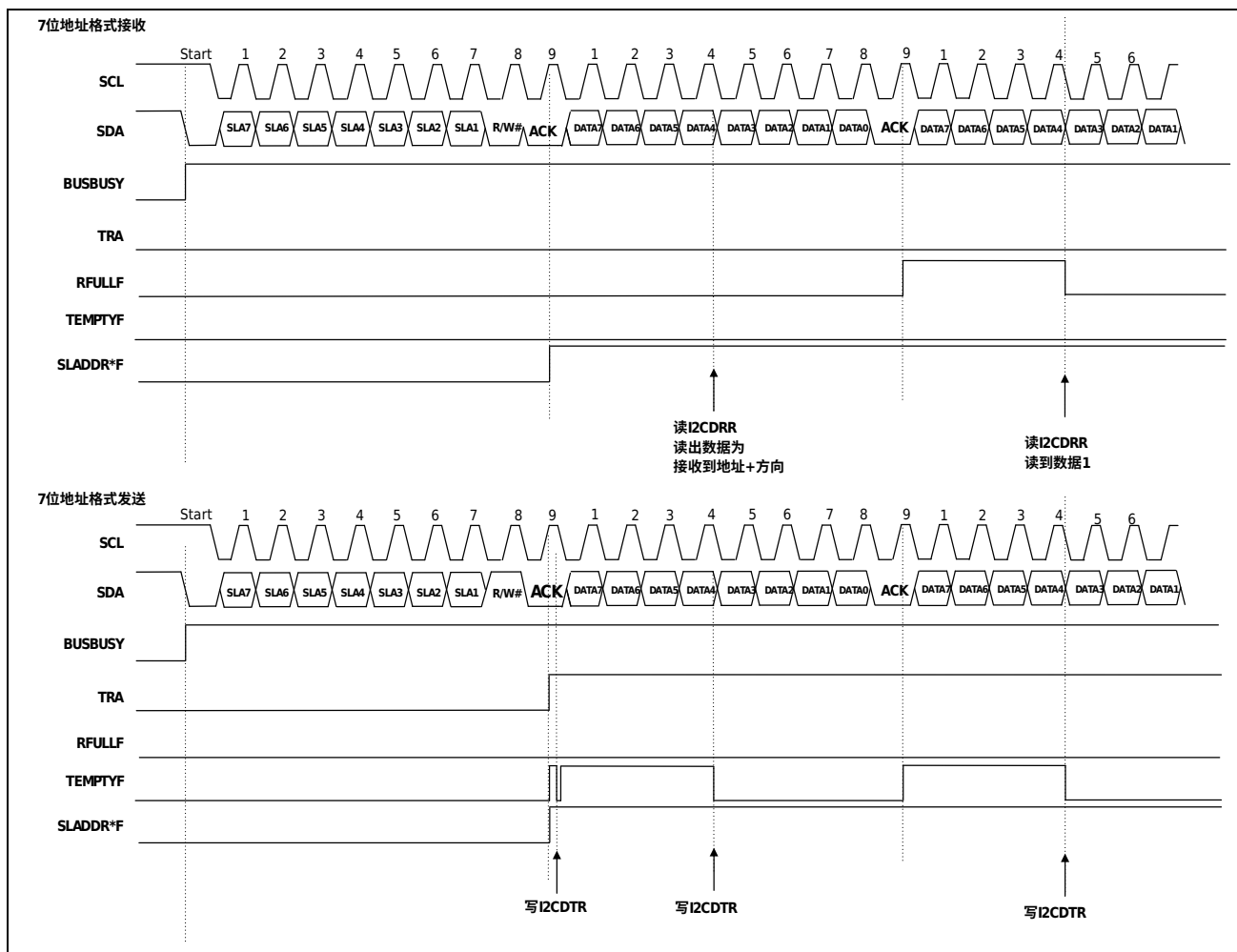


图 27-14 选择 7 位地址格式时的时序

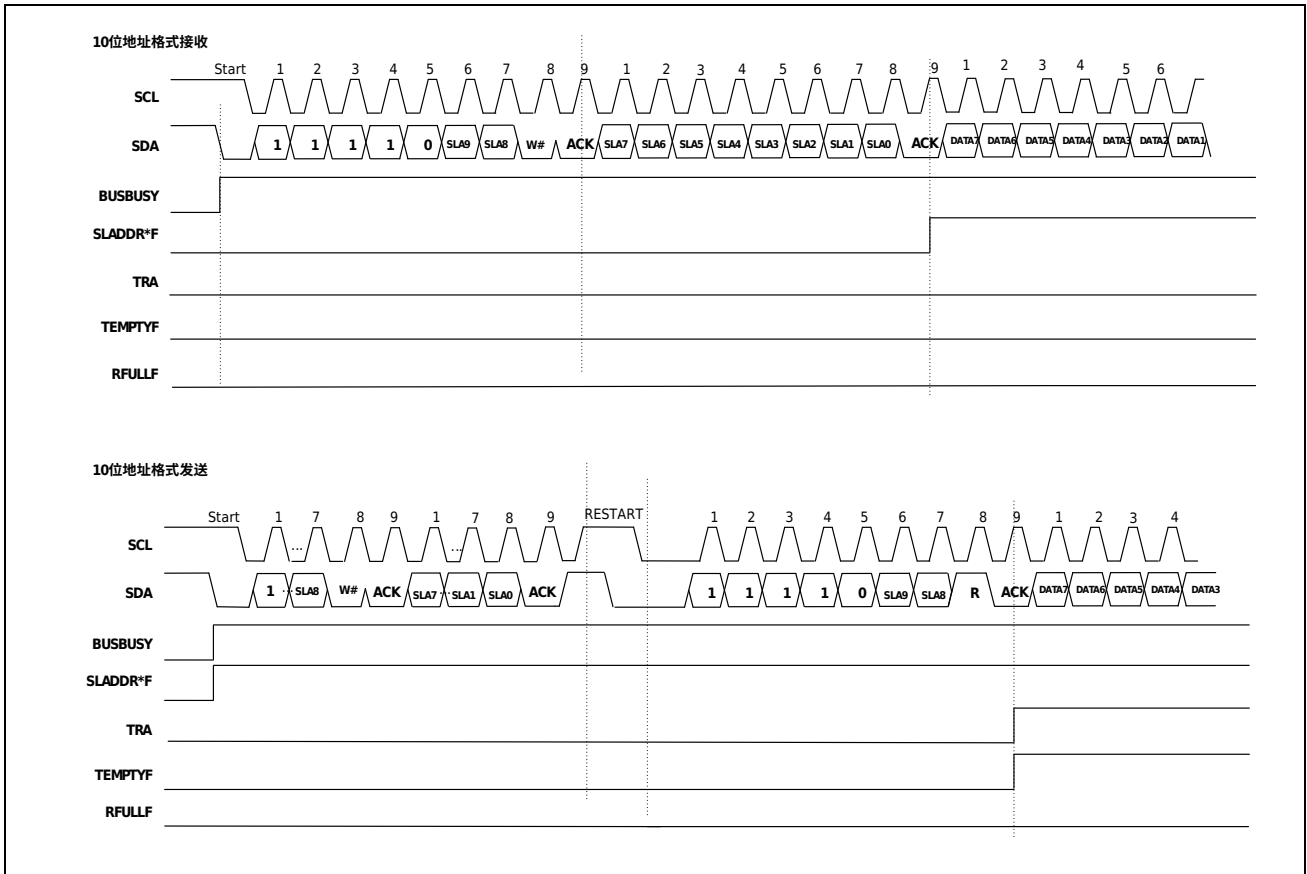


图 27-15 选择 10 位地址格式时的时序

27.3.3.2 广播地址匹配

当 I2C_CR1.GCEN 位为 1 时，能检测广播地址 (0b0000 000+0[W])。

但是开始条件或者重新开始条件后的地址为 0b0000 000+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.GENCALLF 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。

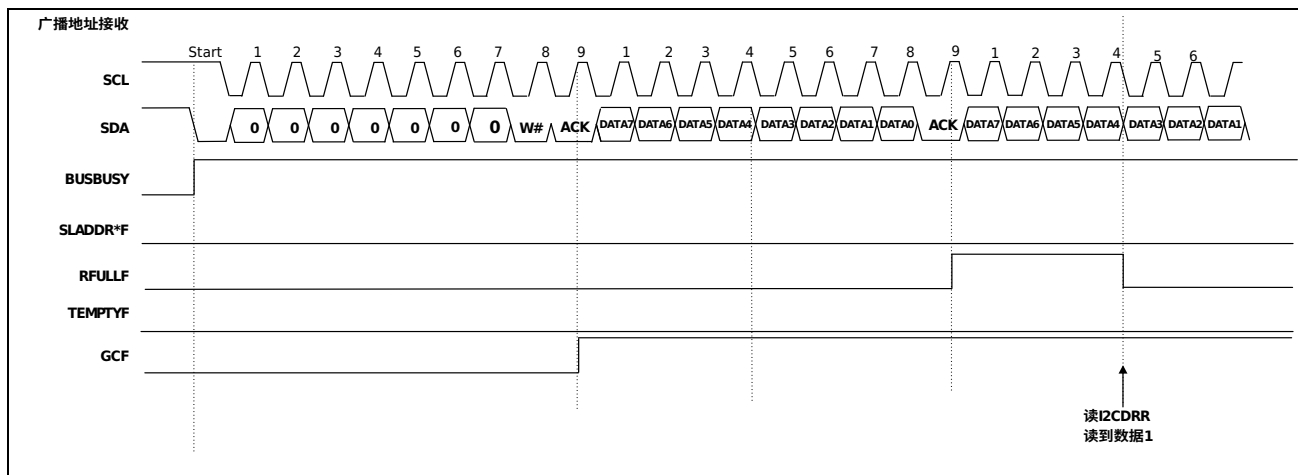


图 27-16 广播地址匹配时序图

27.3.3.3 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_SR.MSL 位 TRA 位为 0b00) 中检测主机地址 (0b0001 000)。

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 (0b0001 000) 后面的位是 Rd 位 (RW# 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。

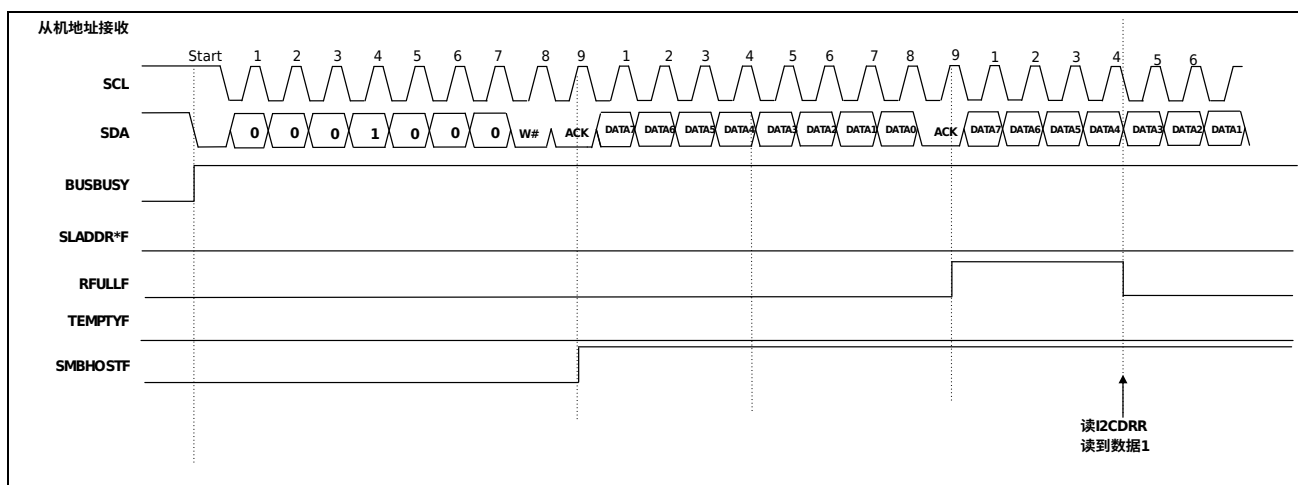


图 27-17 SMBus 主机地址匹配时序图

27.3.3.4 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_SR.MSL 位 TRA 位为 0b00) 中检测 SMBUS 报警响应地址 (0b0001 100)。

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.SMBALRTF 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。

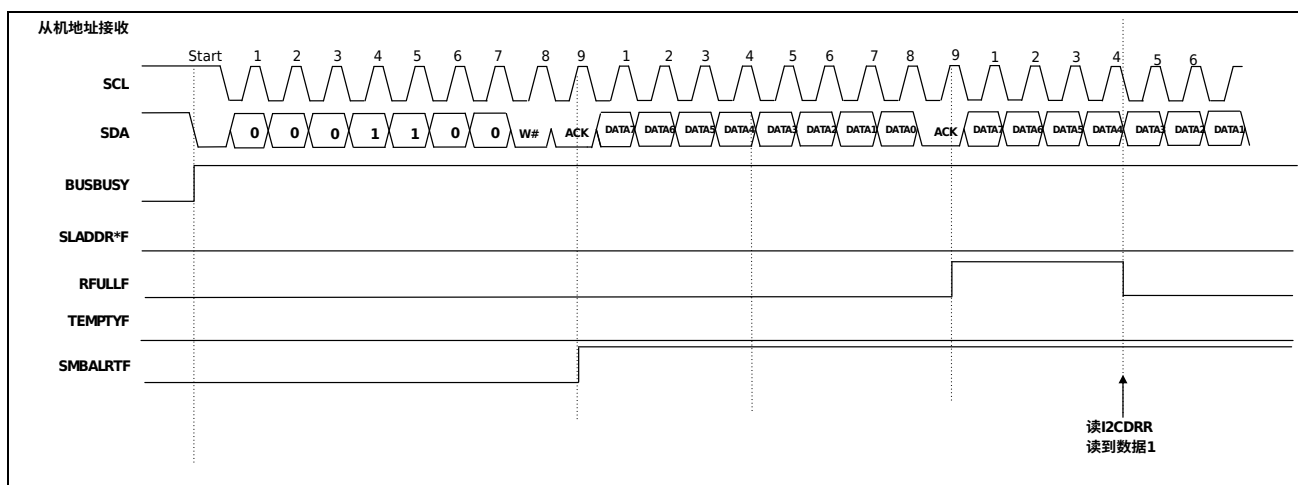


图 27-18 SMBus 报警响应地址匹配时序图

27.3.3.5 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_SR.MSL 位 TRA 位为 0b00) 中检测 SMBUS 默认地址 (0b1100 001)。

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。

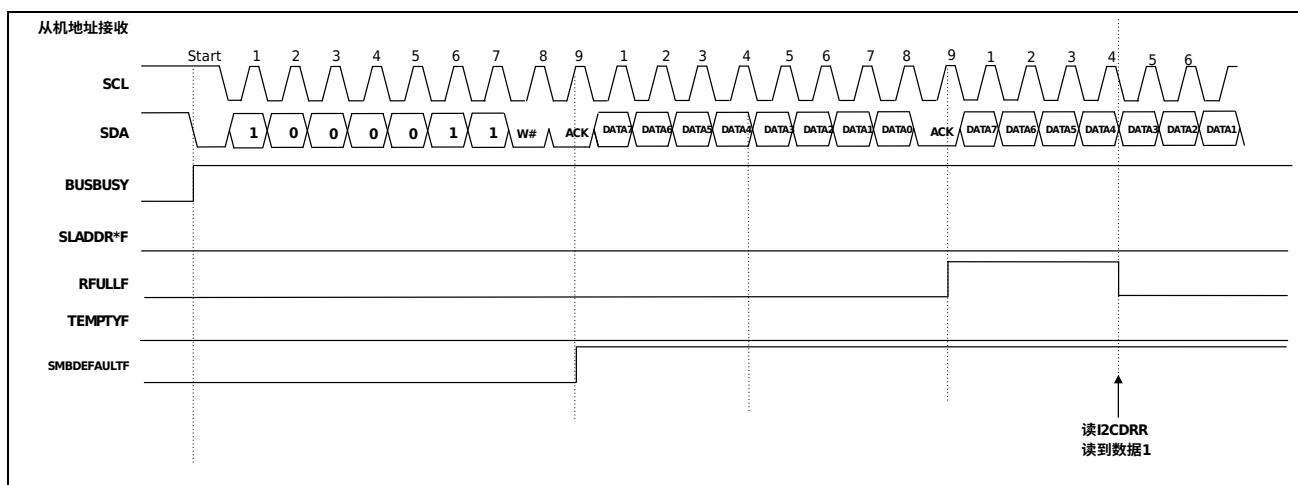


图 27-19 SMBus 默认地址时序图

27.3.4 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。

27.3.4.1 SMBus 超时测量

1) SCL 电平超时测量

工作在从机模式下时, 地址匹配一致后检测到 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。

工作在主机模式下时, 起始条件生成后检测到 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 TIMEOUT: 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 TIMEOUT: 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)。通过发送停止条件中止主机的处理。

27.3.4.2 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。

27.3.5 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 0b01) 过程中不能监视开始条件等的总线状态。

27.3.6 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。

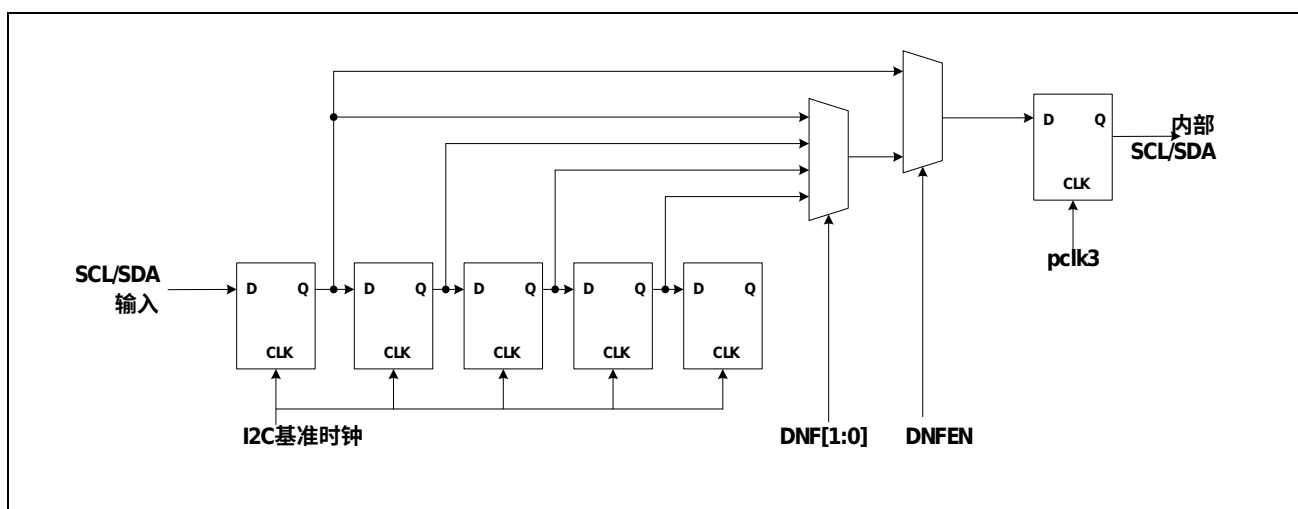


图 27-20 数字滤波电路框图

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。

27.4 中断和事件

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。

中断一览表如下所示。

表 27-2 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
STOPF	STOPF=1&STOPIE=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	EMPTYF	EMPTYF=1&EMPTYIE=1
I2C_TEI	发送结束	TENDF	TENDF=1&TENDIE=1

事件信号输出一览表如下所示。

表 27-3 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

名称	事件源	事件条件
		SMBALRTF=1
		SMBHOSTF=1
		SMBDEFAULTF=1
		GENCALLF=1
		NACKF=1
		TMOUTF=1
		STARTF=1
		STOPF=1
I2C_RXI	接收数据满	RFULLF=1
I2C_TXI	发送数据空	EMPTYF=1
I2C_TEI	发送结束	TENDF=1

27.5 应用举例

27.5.1 I2C 初始化流程

在开始发送或者接收数据时，必须按以下步骤进行初始化。

PE 位设定为 0。

1. SWRST 设定为 1，通信复位；
2. PE 位设定为 1，内部状态复位；
3. 设定从机地址格式和地址；
4. 设定波特率；
5. 根据需要设定控制寄存器功能及中断；
6. SWRST 位设定为 0，解除内部状态复位；
7. 初始化结束。可发送接收数据。

27.6 寄存器描述

表 27-4 I2C 基地址

名称	基地址	描述
I2C	0x4003 B400	I2C 基地址

表 27-5 I2C 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
I2C_CR1	I2C控制寄存器1	0x00	32	0x0000 0040
I2C_CR2	I2C控制寄存器2	0x04	32	0x0000 0000
I2C_CR3	I2C控制寄存器3	0x08	32	0x0000 0006
I2C_CR4	I2C控制寄存器4	0x0C	32	0x0030 1B07
I2C_SLR0	I2C从机地址寄存器0	0x10	32	0x0000 1000
I2C_SLR1	I2C从机地址寄存器1	0x14	32	0x0000 0000
I2C_SLTR	I2C SCL电平超时控制寄存器	0x18	32	0xFFFF FFFF
I2C_SR	I2C状态寄存器	0x1C	32	0x0500 0000
I2C_CLR	I2C状态清零寄存器	0x20	32	0x0000 0000
I2C_DTR	I2C数据发送寄存器	0x24	8	0xFF
I2C_DRR	I2C数据接收寄存器	0x28	8	0x00
I2C_CCR	I2C时钟控制寄存器	0x2C	32	0x0000 1F1F
I2C_FLTR	I2C滤波控制寄存器	0x30	32	0x0000 0010
I2C_FSTR	I2C FIFO状态寄存器	0x34	32	0x0000 0000
I2C_SLVADDR	I2C从机地址接收寄存器	0x38	32	0x0000 0000

27.6.1 I2C 控制寄存器 1 (I2C_CR1)

复位值: 0x0000 0040

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SWRST	Res				ACK	STOP	START	RESTART	GCEN	Res	SMBH OSTE N	SMBD EFAUL TEN	SMBA LRTEN	SMB US	PE

位/位域	标记	位名	功能	读写									
b31-16	Res	保留位	读出时为“0”，写入时写“0”	RW									
b15	SWRST	软件复位	0: 解除复位	RW									
			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"> <thead> <tr> <th>SWRST</th><th>PE</th><th>复位内容</th></tr> </thead> <tbody> <tr> <td>1</td><td>0</td><td>通信复位: I2C内部的全部寄存器 (I2C_FSTR[3:0]除外) 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 (除BUSY位), I2C_DSR寄存器和内部状态机进行复位</td></tr> </tbody> </table>	SWRST	PE	复位内容	1	0	通信复位: I2C内部的全部寄存器 (I2C_FSTR[3:0]除外) 及内部状态复位。	1	1	内部状态复位: I2C_SR (除BUSY位), I2C_DSR寄存器和内部状态机进行复位	
SWRST	PE	复位内容											
1	0	通信复位: I2C内部的全部寄存器 (I2C_FSTR[3:0]除外) 及内部状态复位。											
1	1	内部状态复位: I2C_SR (除BUSY位), I2C_DSR寄存器和内部状态机进行复位											
b14-11	Res	保留位	读出时为“0”，写入时写“0”	RW									
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)	RW									
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	RW									
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	RW									
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	RW									

b6	GCEN	广播呼叫使能	0: 广播地址检测无效 1: 广播地址检测有效	RW
b5	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
b1	SMBUS	SMBUS/I2C总线模式选择位	0: I2C总线模式 1: SMBUS总线模式	RW
b0	PE	I2C功能使能	0: I2C功能禁止 1: I2C功能允许 本位与SWRST位组合，选择内部状态复位或者通信复位	RW

27.6.2 I2C 控制寄存器 2 (I2C_CR2)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Res								SMBALRTIE	SMBHOSTIE	SMBDEFAULTIE	GENCALLIE	Res				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Res	TMOUTIE	Res	NACKIE	RFREQIE	Res	ARLOIE	Res	EMPTYIE	RFULLIE	Res	STOPIE	TENDIE	SLADDR1IE	SLADDR0IE	STARTIE	

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW
b19~b15	Res	保留位	读出时为“0”，写入时写“0”	RW
b14	TMOUTIE	超时中断允许	0: 超时中断禁止 1: 超时中断允许	RW
b13	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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中断请求允许	RW
b10	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b8	Res	保留位	读出时为“0”，写入时写“0”	RW
b7	EMPTYIE	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	RW
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	RW
b5	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW

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	RW
----	---------	-----------------	--------------------------------------	----

27.6.3 I2C 控制寄存器 1 (I2C_CR3)

复位值: 0x0000 0006

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								FACK EN	Res				HTM OUT	LTMO UT	TMOU TEN

位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	读出时为“0”，写入时写“0”	RW
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, RFULL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。	RW
b6~b3	Res	保留位	读出时为“0”，写入时写“0”	RW
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。	RW
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。	RW
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	RW

27.6.4 I2C 控制寄存器 4 (I2C_CR4)

复位值: 0x0030 1B07

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

位/位域	标记	位名	功能	读写
b31~b22	Res	保留位	读出时为“0”，写入时写“0”	RW
b21~b20	Res	保留位	读出时为所有位“1”，写入时所有位写“1”	RW
b19~b13	Res	保留位	读出时为“0”，写入时写“0”	RW
b12	BUSFREE_CLREN	总线结状态清除位	0: 禁止I2C总线STOP标志对I2C_SLVADRR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_SLVADRR从机接收地址寄存器以及I2C_SR状态寄存器中SMBALRTF, SMBHOSTF, SMBDEFAULTF, GENCALLF, MSL, TENDF, SLADDR1F, SLADDR0F, STARTF位进行清除	RW
b11	Res	保留位	读出时为“1”，写入时写“1”	RW
b10	BUSWAIT	总线等待位	0: I2C_DRR接收满、I2C_DSR为时空时，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为时空时，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。	RW
b9~b8	Res	保留位	读出时所有位为“1”，写入时所有位写“1”	RW
b7~b3	Res	保留位	读出时为“0”，写入时写“0”	RW
b2~b0	Res	保留位	读出时所有位为“1”，写入时所有位写“1”	RW

27.6.5 I2C 从机地址寄存器 0 (I2C_SLR0)

复位值: 0x0000 1000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res					MASK0EN	MSLADDR0[9:0]									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRMOD0	Res	SLADDR0EN	Res	SLADDR0[9:0]											

位/位域	标记	位名	功能	读写
b31~b27	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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。	RW
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	RW
b14~b13	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b11~b10	Res	保留位	读出时为“0”，写入时写“0”	RW
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。	RW
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。	RW

27.6.6 I2C 从机地址寄存器 1 (I2C_SLR1)

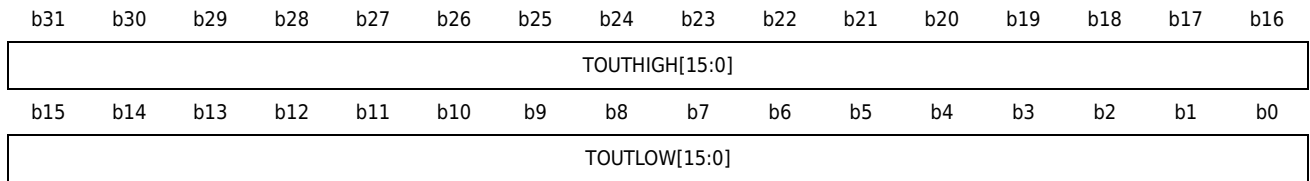
复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res					MASK1EN	MSLADDR1[9:0]									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRMOD1	Res		SLADDR1EN	Res		SLADDR1[9:0]									

位/位域	标记	位名	功能	读写
b31~b27	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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。	RW
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	RW
b14~b13	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b11~b10	Res	保留位	读出时为“0”，写入时写“0”	RW
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。	RW
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。	RW

27.6.7 I2C SCL 电平超时控制寄存器 (I2C_SLTR)

复位值: 0xFFFF FFFF



位/位域	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	RW
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	RW

27.6.8 I2C 状态寄存器 (I2C_SR)

复位值: 0x0500 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RF REQ	Res	TFST[1:0]		RF FULL	RF EMPTY	TF FULL	TF EMPTY	SMBALRTF	SMBHOSTF	SMBDEFAULTF	GENCALLF	Res	TRA	BUSY	MSL
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	TMOU TF	Res	NACK F	Res	ACKR F	ARLO F	Res	TEMP TYF	RFUL LF	Res	STOP F	TEND F	SLAD DR1F	SLAD DR0F	STAR TF

位/位域	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	Res	保留位	读出时为“0”，写入时写“0”	R
b29~b28	TFST	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 bytes有效数据 11: 保留位	
b27	RFFULL	接收FIFO满标志位	0: 当前接收FIFO未滿 1: 当前接收FIFO已滿	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未滿 1: 当前发送FIFO已滿	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0b0001 100匹配一致 清“0”条件: SMBALRTFCLR写“1” 检测到停止条件 通信复位 内部状态复位	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0b0001 000匹配一致 清“1”条件: SMBHOSTFCLR写“1” 检测到停止条件 通信复位 内部状态复位	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	R

			<p>置“1”条件： 接收到的地址和0b1100 001匹配一致</p> <p>清“0”条件： SMBDEFAULTFCLR写“1”</p> <p>检测到停止条件</p> <p>通信复位</p> <p>内部状态复位</p>																
b20	GENCALLF	广播呼叫地址匹配一致标志	<p>0: 未匹配到广播呼叫地址</p> <p>1: 匹配到广播呼叫地址</p> <p>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</p> <p>清“0”条件： GENCALLFCLR写“1”</p> <p>检测到停止条件</p> <p>通信复位</p> <p>内部状态复位</p>	R															
b19	Res	保留位	读出时为“0”，写入时写“0”	R															
b18	TRA	发送接收选择位	<p>此位表示选择发送数据还是接收数据。</p> <p>0: 接收数据</p> <p>1: 发送数据</p> <p>此位可软件置1和清0。</p> <p>硬件置“1”条件 检测到开始条件</p> <p>主机模式下，发送的RW位为0</p> <p>从机模式下，地址匹配且接收的RW位为1</p> <p>硬件清“0”条件 检测到停止条件</p> <p>主机模式下，发送的RW位为1</p> <p>从机模式下，地址匹配且接收的RW位为0</p> <p>通信复位</p> <p>内部状态复位</p>	RW															
b17	BUSY	总线忙标志位	<p>0: 空闲状态，总线上无通信</p> <p>1: 占有状态，总线正在通信</p> <p>置“1”条件： 检测到总线上开始条件</p> <p>清“0”条件： 检测到总线停止条件</p> <p>通信复位</p>	R															
b16	MSL	主从机选择位	<p>此位表示主机还是从机。</p> <p>0: 从机模式</p> <p>1: 主机模式</p> <p>通过和TRA位的组合，表示I2C的运行模式。</p> <table border="1"> <thead> <tr> <th>MSL</th> <th>TRA</th> <th>I2C运行模式</th> </tr> </thead> <tbody> <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> </tbody> </table>	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	RW
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																	

			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 3) 通信复位 内部状态复位	
b15	Res	保留位	读出时为“0”，写入时写“0”	R
b14	TMOUTF	超时标志位	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 内部状态复位	R
b13	Res	保留位	读出时为“0”，写入时写“0”	R
b12	NACKF	NACK标志位	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 内部状态复位	R
b11	Res	保留位	读出时为“0”，写入时写“0”	R
b10	ACKRF	接收应答位	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 内部状态复位	R
b9	ARLOF	仲裁失败标志位	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 内部状态复位	R
b8	Res	保留位	读出时为“0”，写入时写“0”	R
b7	EMPTYF	发送数据空标志位	0: I2C_DTR寄存器满 1: I2C_DTR寄存器空 置“1”条件：	R

			<p>I2C_DTR数据传送到I2C_DSR</p> <p>TRA位置1</p> <p>清“0”条件： 写I2C_DTR</p> <p>TRA位清0</p> <p>通信复位</p> <p>内部状态复位</p>	
b6	RFULLF	接收数据满标志位	<p>0: I2C_DRR寄存器空</p> <p>1: I2C_DRR寄存器满</p> <p>置“1”条件： 接收的数据从I2C_DSR传送到I2C_DRR</p> <p>清“0”条件： 读I2C_DRR RFULLFCLR写“1”</p> <p>通信复位</p> <p>内部状态复位</p>	R
b5	Res	保留位	<p>读出时为“0”，写入时写“0”</p>	R
b4	STOPF	停止条件标志位	<p>0: 总线未检测到停止条件</p> <p>1: 总线检测到停止条件</p> <p>置“1”条件： 检测到停止条件</p> <p>清“0”条件： STOPFCLR写“1”</p> <p>通信复位</p> <p>内部状态复位</p>	R
b3	TENDF	发送数据结束标志位	<p>0: I2C_DSR寄存器发送中</p> <p>1: I2C_DSR寄存器发送结束</p> <p>置“1”条件： EMPTYF=1的条件下，SCL的第9个上升沿此位置“1”</p> <p>清“0”条件： 检测到停止条件</p> <p>写I2C_DTR</p> <p>TENDFCLR写“1”</p> <p>通信复位</p> <p>内部状态复位</p>	R
b2	SLADDR1F	从机地址寄存器1匹配一致标志	<p>0: 未检测到从机地址寄存器1一致地址</p> <p>1: 检测到从机地址寄存器1一致地址</p> <p>置“1”条件： 当I2C_SLR1.ADDRM0D1位为“0”时，接收到的从机地址和I2C_SLR1.SLADDR1[7:1]匹配时。</p> <p>当I2C_SLR1.ADDRM0D1位为“1”时，接收10位从机地址的第一个字节地址与0b11110+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。</p> <p>清“0”条件： 检测到停止条件</p> <p>SLADDR1FCLR写“1”</p> <p>通信复位</p> <p>内部状态复位</p>	R

b1	SLADDR0F	从机地址寄存器0匹配一致标志	<p>0: 未检测到从机地址寄存器0一致地址</p> <p>1: 检测到从机地址寄存器0一致地址</p> <p>置“1”条件:</p> <p>当I2C_SLR0.ADDRMODE0位为“0”时, 接收到的从机地址和I2C_SLR0.SLADDR0[7:1]匹配时。</p> <p>当I2C_SLR0.ADDRMODE0位为“1”时, 接收10位从机地址的第一个字节地址与0b11110+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。</p> <p>清“0”条件:</p> <p>检测到停止条件</p> <p>SLADDR0FCLR写“1”</p> <p>通信复位</p> <p>内部状态复位</p>	R
b0	STARTF	开始条件/重新开始条件标志位	<p>0: 总线未检测到开始条件</p> <p>1: 总线检测到开始条件</p> <p>置“1”条件</p> <p>1) 检测到开始条件</p> <p>清“0”条件</p> <p>1) 检测到停止条件</p> <p>2) STARTFCLR写“1”</p> <p>3) 通信复位</p> <p>内部状态复位</p>	R

27.6.9 I2C 状态清零寄存器 (I2C_CLR)

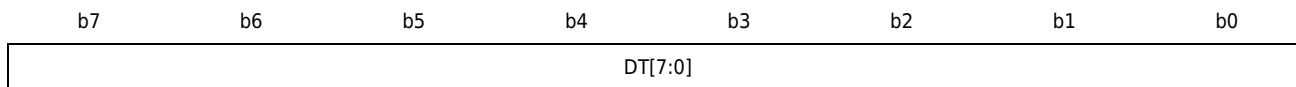
复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								SMBALRTFCLR	SMBHOSTFCLR	SMBDEFAULTFCLR	GENCALLFCLR	Res			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	TMOUTFCLR	Res	NACKFCLR	Res	RFREQCLR	ARLOFCLR	Res		RFULLFCLR	Res	STOPFCLR	TENDFCLR	SLADDR1FCLR	SLADDR0FCLR	STARTFCLR

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	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	Res	保留位	写入时写“0”	W
b14	TMOUTFCLR	超时标志位	写“1”清除TMOUTF标志位	W
b13	Res	保留位	写入时写“0”	W
b12	NACKFCLR	NACK标志位	写“1”清除NACKF标志位	W
b11	Res	保留位	写入时写“0”	W
b10	RFREQCLR	接收FIFO数据请求标志位	写“1”清除RFREQ	W
b9	ARLOFCLR	仲裁失败标志位	写“1”清除ARLOF标志位	W
b8~b7	Res	保留位	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	Res	保留位	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0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

27.6.10 I2C 数据发送寄存器 (I2C_DTR)

复位值: 0xFF



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请求发生时, 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 TEMPTIF 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。

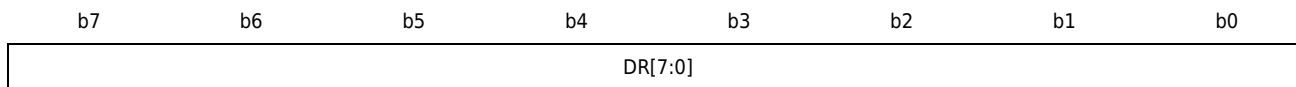
当 FEN 为 1 时选择外部数据总线到 FIFO 数据通路以及 FIFO 到 I2C_DSR 的数据通路。

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。

27.6.11 I2C 数据接收寄存器 (I2C_DRR)

复位值: 0x00



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 27-12 图 27-13) 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 2bytes 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。

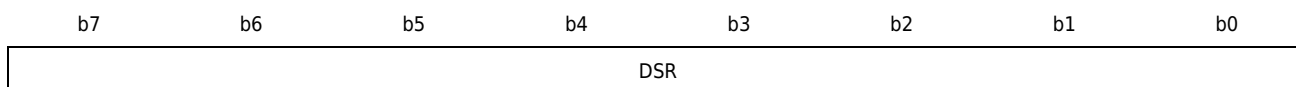
I2C 接收通路的选择通过 I2C 中 FIFO 状态寄存器里面的 FEN 位来选择。

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。

27.6.12 I2C 数据移位寄存器 (I2C_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。

27.6.13 I2C 时钟控制寄存器 (I2C_CCR)

复位值: 0x0000 1F1F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Res								FMPLUSEN	Res				CKDIV[2:0]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
SHIGHW[7:0]								SLOWW[7:0]								

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”，写入时写“0”	RW
b23	FMPLUSEN	Fast plus模式使能位	0: Fast plus模式禁止 1: Fast plus模式使能	RW
b22~b19	Res	保留位	读出时为“0”，写入时写“0”	RW
b18-b16	CKDIV[2:0]	I2C基准时钟频率设定位	000: I2C基准时钟频率=PCLK0/1 001: I2C基准时钟频率=PCLK0/2 010: I2C基准时钟频率=PCLK0/4 011: I2C基准时钟频率=PCLK0/8 100: I2C基准时钟频率=PCLK0/16 101: I2C基准时钟频率=PCLK0/32 110: I2C基准时钟频率=PCLK0/64 111: I2C基准时钟频率=PCLK0/128	RW
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	RW
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	RW

注意:

1. 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。
2. 在主机模式下，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)，对于标准模式为 250ns，对于快速模式为 100ns，对于超快速模式为 50ns。
3. 波特率:
 - 1) I2C_FLTR.DNFEN=0, I2C_CCR.CKDIV=0b000:
波特率=1/{[(SHIGHW+3) + (SLOWW+3)]/φI2C+SCL 上升时间+SCL 下降时间}
 - 2) I2C_FLTR.DNFEN=1, I2C_CCR.CKDIV=0b000:
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}
 - 3) I2C_FLTR.DNFEN=0, I2C_CCR.CKDIV!=0b000:
波特率=1/{[(SHIGHW+2) + (SLOWW+2)]/φI2C+SCL 上升时间+SCL 下降时间}

4) I2C_FLTR.DNFEN=1, I2C_CCR.CKDIV!= 0b000:

波特率=1/{[(SHIGHW+2+滤波能力) + (SLOWW+2+滤波能力)]/ΦI2C+SCL 上升时间+SCL 下降时间}

4. SCL 线的上升时间[tr]和下降时间[tf]取决于总线的总电容[Cb]和上拉电阻[Rp], 详细内容请参照 I2C 总线规格书。
5. ΦI2C: I2C 分频过后的时钟。

27.6.14 I2C 滤波控制寄存器 (I2C_FLTR)

复位值: 0x0000 0010

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

位/位域	标记	位名	功能	读写
b31~b6	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
b3~b2	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

27.6.15 I2C FIFO 控制器寄存器 (I2C_FSTR)

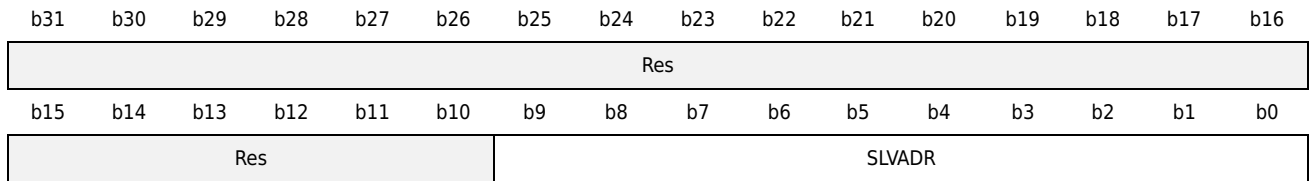
复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Res																
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Res								RFST	TFST	NACK TFFLUS H	RF FLUSH	TF FLUS HH	FEN			

位/位域	标记	位名	功能	读写
b31~b7	Res	保留位	读出时为“0”，写入时写“0”	RW
b7~b6	RFST	接收FIFO状态位	00: FIFO空 01: FIFO半满 10: FIFO满 11: 保留 通信复位	R
b5~b4	TFST	发送FIFO状态位	00: FIFO空 01: FIFO半空 10: FIFO满 11: 保留位 通信复位	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 系统复位	RW
b2	RFLUSH	RX FIFO指针复位	0: 写入无效 1: 复位RX FIFO指针 系统复位	RW
b1	TFLUSH	TX FIFO指针复位	0: 写入无效 1: 复位TX FIFO指针 系统复位	RW
b0	FEN	FIFO功能允许位	0: FIFO功能禁止 1: FIFO功能允许 系统复位	RW

27.6.16 I2C 从机地址接收器寄存器 (I2C_SLVADRR)

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b10	Res	保留位	读出时为“0”，写入时写“0”	RW
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

28 可编程逻辑阵列 (PLA)

28.1 概述

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 异步输出。
- 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。

28.2 功能说明

28.2.1 PLA 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

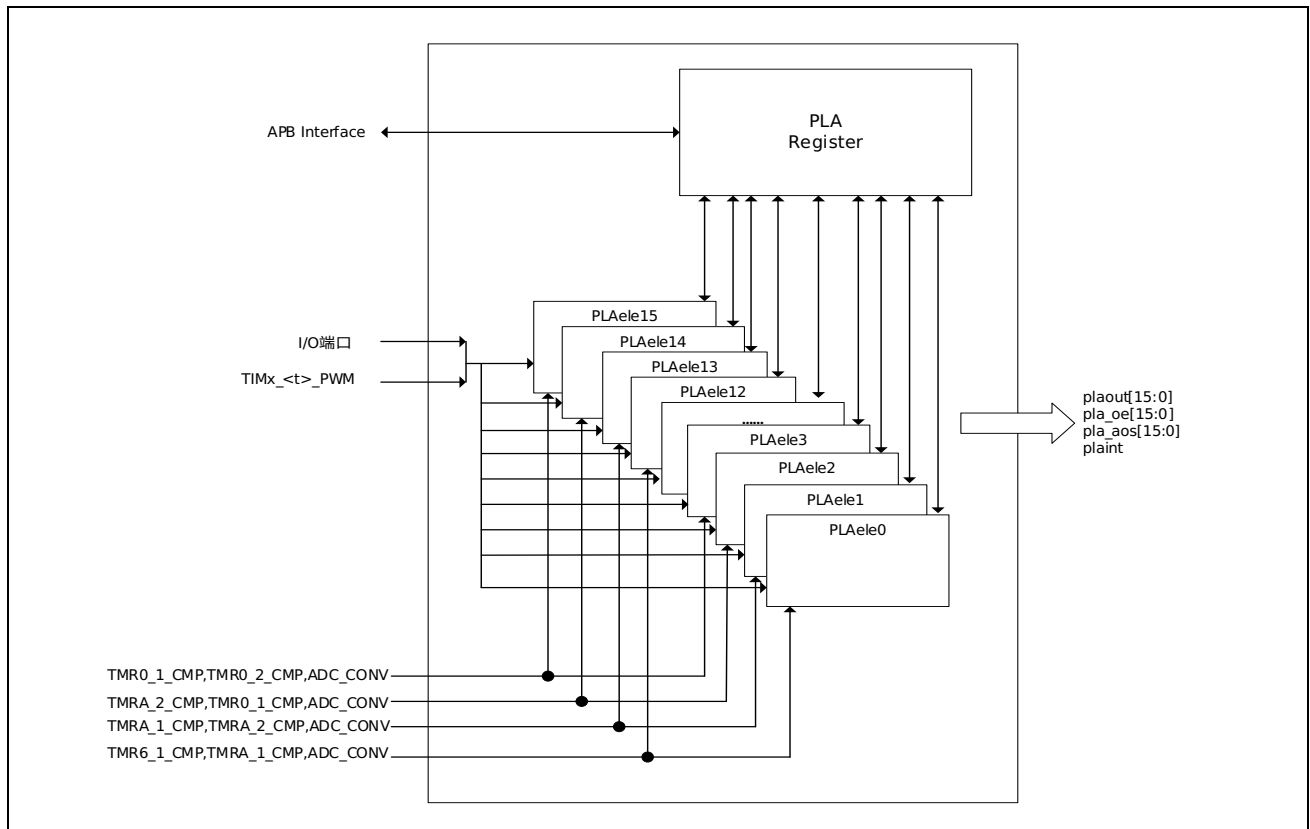


图 28-1 PLA 模块框图

28.2.2 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 13-2.PLA 单元框图显示了 PLA 单元的框图，每个 PLA 单元包含两个多路选择器和一个 LUT。

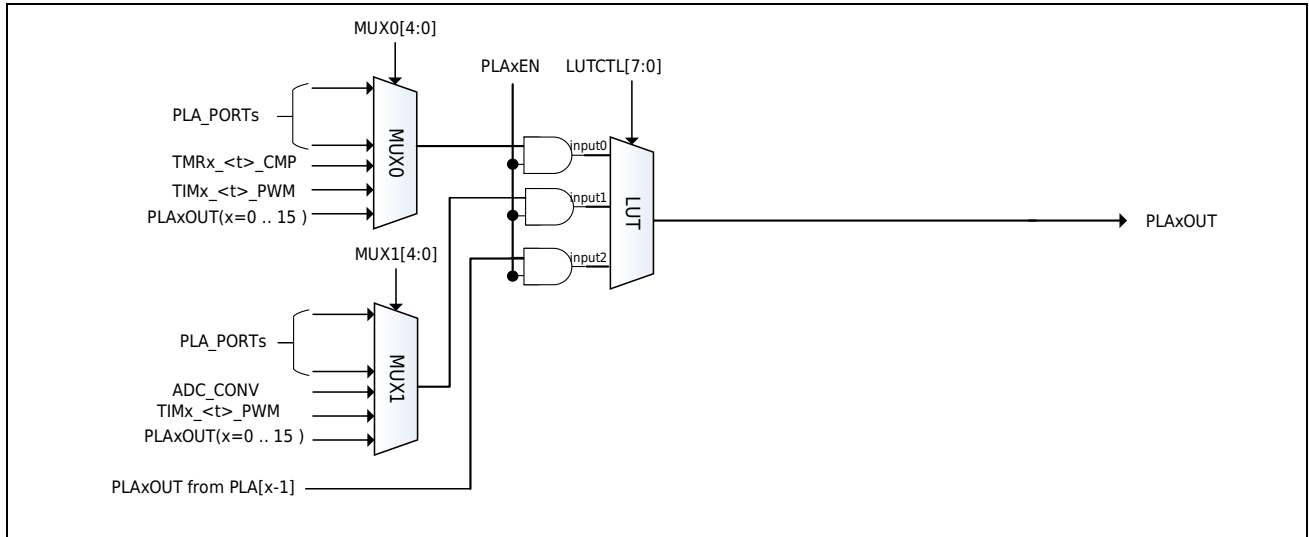


图 28-2 PLA 单元框图

28.2.3 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（TMRx_<t>_CMP），定时器通道（TMRx_<t>_PWM），ADC 转换信号（ADC_CONV）和 PLA 单元异步输出（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 28-1 PLAxMUX0 输入选择和表 28-2 PLAxMUX1 输入选择显示了 PLAxMUX0 和 PLAxMUX1 的输入选择。

表 28-1 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	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	TMRA_1_PWM3	TMRA_1_PWM4	TMRA_1_PWM3	TMRA_1_PWM4	TMRA_1_PWM3	TMRA_1_PWM4	TMRA_1_PWM3	TMRA_1_PWM4	TMRA_1_PWM3	TMRA_1_PWM4	TMRA_1_PWM3	TMRA_1_PWM4	TMRA_1_PWM3	TMRA_1_PWM4	TMRA_1_PWM3	TMRA_1_PWM4
00101	TMR6_1_PWMA	TMR6_1_PWMA	TMR6_2_PWMA	TMR6_3_PWMA	TMR6_1_PWMA	TMR6_1_PWMA	TMR6_2_PWMA	TMR6_3_PWMA	TMR6_1_PWMA	TMR6_1_PWMA	TMR6_2_PWMA	TMR6_3_PWMA	TMR6_1_PWMA	TMR6_1_PWMA	TMR6_2_PWMA	TMR6_3_PWMA
00110	TMR6_2_PWMA	TMR6_4_PWMA	TMR6_4_PWMA	TMRA_1_PWM1	TMR6_2_PWMA	TMR6_4_PWMA	TMR6_4_PWMA	TMRA_1_PWM1	TMR6_2_PWMA	TMR6_4_PWMA	TMR6_4_PWMA	TMRA_1_PWM1	TMR6_2_PWMA	TMR6_4_PWMA	TMR6_4_PWMA	TMRA_1_PWM1
00111	TMR6_3_PWMA	TMRA_1_PWM1	TMRA_1_PWM2	TMRA_1_PWM2	TMR6_3_PWMA	TMRA_1_PWM1	TMRA_1_PWM2	TMRA_1_PWM2	TMR6_3_PWMA	TMRA_1_PWM1	TMRA_1_PWM2	TMRA_1_PWM2	TMR6_3_PWMA	TMRA_1_PWM1	TMRA_1_PWM2	TMRA_1_PWM2
01000	PLAIN0	PLAIN4	PLAIN0	PLAIN20	PLAIN0	PLAIN4	PLAIN0	PLAIN20	PLAIN0	PLAIN4	PLAIN0	PLAIN20	PLAIN0	PLAIN4	PLAIN0	PLAIN20
01001	PLAIN2	PLAIN5	PLAIN20	PLAIN2	PLAIN2	PLAIN5	PLAIN20	PLAIN2	PLAIN2	PLAIN5	PLAIN20	PLAIN2	PLAIN2	PLAIN5	PLAIN20	PLAIN2
01010	PLAIN4	PLAIN8	PLAIN8	PLAIN6	PLAIN4	PLAIN8	PLAIN8	PLAIN6	PLAIN4	PLAIN8	PLAIN8	PLAIN6	PLAIN4	PLAIN8	PLAIN8	PLAIN6
01011	PLAIN6	PLAIN10	PLAIN9	PLAIN7	PLAIN6	PLAIN10	PLAIN9	PLAIN7	PLAIN6	PLAIN10	PLAIN9	PLAIN7	PLAIN6	PLAIN10	PLAIN9	PLAIN7
01100	PLAIN8	PLAIN12	PLAIN14	PLAIN10	PLAIN8	PLAIN12	PLAIN14	PLAIN10	PLAIN8	PLAIN12	PLAIN14	PLAIN10	PLAIN8	PLAIN12	PLAIN14	PLAIN10
01101	PLAIN10	PLAIN13	PLAIN15	PLAIN11	PLAIN10	PLAIN13	PLAIN15	PLAIN11	PLAIN10	PLAIN13	PLAIN15	PLAIN11	PLAIN10	PLAIN13	PLAIN15	PLAIN11
01110	PLAIN12	PLAIN16	PLAIN16	PLAIN18	PLAIN12	PLAIN16	PLAIN16	PLAIN18	PLAIN12	PLAIN16	PLAIN16	PLAIN18	PLAIN12	PLAIN16	PLAIN16	PLAIN18
01111	PLAIN14	PLAIN18	PLAIN17	PLAIN19	PLAIN14	PLAIN18	PLAIN17	PLAIN19	PLAIN14	PLAIN18	PLAIN17	PLAIN19	PLAIN14	PLAIN18	PLAIN17	PLAIN19
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

表 28-2 PLAxMUX1 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	TMR6_4_PWMA	TMR6_2_PWMA	TMR6_1_PWMA	TMR6_1_PWMA	TMR6_4_PWMA	TMR6_2_PWMA	TMR6_1_PWMA	TMR6_1_PWMA	TMR6_4_PWMA	TMR6_2_PWMA	TMR6_1_PWMA	TMR6_1_PWMA	TMR6_4_PWMA	TMR6_2_PWMA	TMR6_1_PWMA	TMR6_1_PWMA
00110	TMRA_1_PWM1	TMR6_3_PWMA	TMR6_3_PWMA	TMR6_2_PWMA	TMRA_1_PWM1	TMR6_3_PWMA	TMR6_3_PWMA	TMR6_2_PWMA	TMRA_1_PWM1	TMR6_3_PWMA	TMR6_3_PWMA	TMR6_2_PWMA	TMRA_1_PWM1	TMR6_3_PWMA	TMR6_3_PWMA	TMR6_2_PWMA
00111	TMRA_1_PWM2	TMRA_1_PWM2	TMRA_1_PWM1	TMR6_4_PWMA	TMRA_1_PWM2	TMRA_1_PWM2	TMRA_1_PWM1	TMR6_4_PWMA	TMRA_1_PWM2	TMRA_1_PWM2	TMRA_1_PWM1	TMR6_4_PWMA	TMRA_1_PWM2	TMRA_1_PWM2	TMRA_1_PWM1	TMR6_4_PWMA
01000	PLAIN1	PLAIN6	PLAIN2	PLAIN0	PLAIN1	PLAIN6	PLAIN2	PLAIN0	PLAIN1	PLAIN6	PLAIN2	PLAIN0	PLAIN1	PLAIN6	PLAIN2	PLAIN0
01001	PLAIN3	PLAIN7	PLAIN3	PLAIN1	PLAIN3	PLAIN7	PLAIN3	PLAIN1	PLAIN3	PLAIN7	PLAIN3	PLAIN1	PLAIN3	PLAIN7	PLAIN3	PLAIN1
01010	PLAIN5	PLAIN9	PLAIN10	PLAIN4	PLAIN5	PLAIN9	PLAIN10	PLAIN4	PLAIN5	PLAIN9	PLAIN10	PLAIN4	PLAIN5	PLAIN9	PLAIN10	PLAIN4
01011	PLAIN6	PLAIN11	PLAIN11	PLAIN5	PLAIN6	PLAIN11	PLAIN11	PLAIN5	PLAIN6	PLAIN11	PLAIN11	PLAIN5	PLAIN6	PLAIN11	PLAIN11	PLAIN5
01100	PLAIN9	PLAIN14	PLAIN12	PLAIN8	PLAIN9	PLAIN14	PLAIN12	PLAIN8	PLAIN9	PLAIN14	PLAIN12	PLAIN8	PLAIN9	PLAIN14	PLAIN12	PLAIN8
01101	PLAIN11	PLAIN15	PLAIN13	PLAIN9	PLAIN11	PLAIN15	PLAIN13	PLAIN9	PLAIN11	PLAIN15	PLAIN13	PLAIN9	PLAIN11	PLAIN15	PLAIN13	PLAIN9
01110	PLAIN13	PLAIN17	PLAIN18	PLAIN16	PLAIN13	PLAIN17	PLAIN18	PLAIN16	PLAIN13	PLAIN17	PLAIN18	PLAIN16	PLAIN13	PLAIN17	PLAIN18	PLAIN16
01111	PLAIN15	PLAIN19	PLAIN19	PLAIN17	PLAIN15	PLAIN19	PLAIN19	PLAIN17	PLAIN15	PLAIN19	PLAIN19	PLAIN17	PLAIN15	PLAIN19	PLAIN19	PLAIN17
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

28.2.4 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 28-3 所示。

表 28-3 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 $\{input2 \wedge input1 \wedge input0\}$ 的逻辑时, 需设置 $LUTCTL[7:0] = 8'b10010110$ 。

应用实例: 欲实现 pla_ele0 单元的 $\{input2 \wedge input1 \wedge input0\}$ 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 $\{input2 \wedge input1 \wedge input0\}$ 的逻辑,

PLA0_MUXS=0x00000088, 选通 MUX0 01000 和 MUX1 01000,

PLA_GCTL=0x00000001, 使能 PLA0 单元。

28.2.5 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 28-4 PLA 输出端口对应表

引脚名称	输入/输出	功能
PA2/PA8/PB3	O	PLA0OUT
PA7/PA9/PB0/PB4	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/PA12/PC3	O	PLA7OUT
PA13/PB12	O	PLA8OUT
PB13	O	PLA9OUT
PB14	O	PLA10OUT
PB15	O	PLA11OUT
PC4/PC6	O	PLA12OUT
PC5/PC7	O	PLA13OUT
PC8/PC11	O	PLA14OUT
PC9/PC12	O	PLA15OUT

28.3 寄存器说明

表 28-5 PLA 基地址

名称	基地址	描述
PLA	0x4004 FC00	PLA基地址

表 28-6 寄存器一览

寄存器	描述	偏移地址	位宽	复位值
PLA_GCTL	全局控制寄存器	0x00	32	0x0000 0000
PLA_STAT	状态寄存器	0x0C	32	0x0000 0000
PLAx_MUXS	多路选择器选择寄存器	$0x10 + 0x0C * x$	32	0x0000 0000
PLAx_LUTCTL	LUT控制寄存器	$0x14 + 0x0C * x$	32	0x0000 0000
PLAx_CTL	控制寄存器	$0x18 + 0x0C * x$	32	0x0000 0000

注*1: 地址计算公式中 $x = 0 \sim 15$

28.3.1 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
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PLA15EN	PLA14EN	PLA13EN	PLA12EN	PLA11EN	PLA10EN	PLA9EN	PLA8EN	PLA7EN	PLA6EN	PLA5EN	PLA4EN	PLA3EN	PLA2EN	PLA1EN	PLA0EN

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为 0, 写时请写 0	RW
b15	PLA15EN	PLA15 单元使能	0: 禁止 PLA15 单元 1: 允许 PLA15 单元	RW
b14	PLA14EN	PLA14 单元使能	0: 禁止 PLA14 单元 1: 允许 PLA14 单元	RW
b13	PLA13EN	PLA13 单元使能	0: 禁止 PLA13 单元 1: 允许 PLA13 单元	RW
b12	PLA12EN	PLA12 单元使能	0: 禁止 PLA12 单元 1: 允许 PLA12 单元	RW
b11	PLA11EN	PLA11 单元使能	0: 禁止 PLA11 单元 1: 允许 PLA11 单元	RW
b10	PLA10EN	PLA10 单元使能	0: 禁止 PLA10 单元 1: 允许 PLA10 单元	RW
b9	PLA9EN	PLA9 单元使能	0: 禁止 PLA9 单元 1: 允许 PLA9 单元	RW
b8	PLA8EN	PLA8 单元使能	0: 禁止 PLA8 单元 1: 允许 PLA8 单元	RW
b7	PLA7EN	PLA7 单元使能	0: 禁止 PLA7 单元 1: 允许 PLA7 单元	RW
b6	PLA6EN	PLA6 单元使能	0: 禁止 PLA6 单元 1: 允许 PLA6 单元	RW
b5	PLA5EN	PLA5 单元使能	0: 禁止 PLA5 单元 1: 允许 PLA5 单元	RW
b4	PLA4EN	PLA4 单元使能	0: 禁止 PLA4 单元 1: 允许 PLA4 单元	RW
b3	PLA3EN	PLA3 单元使能	0: 禁止 PLA3 单元 1: 允许 PLA3 单元	RW
b2	PLA2EN	PLA2 单元使能	0: 禁止 PLA2 单元 1: 允许 PLA2 单元	RW
b1	PLA1EN	PLA1 单元使能	0: 禁止 PLA1 单元 1: 允许 PLA1 单元	RW
b0	PLA0EN	PLA0 单元使能	0: 禁止 PLA0 单元 1: 允许 PLA0 单元	RW

28.3.2 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
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PLA15OUT	PLA14OUT	PLA13OUT	PLA12OUT	PLA11OUT	PLA10OUT	PLA9OUT	PLA8OUT	PLA7OUT	PLA6OUT	PLA5OUT	PLA4OUT	PLA3OUT	PLA2OUT	PLA1OUT	PLA0OUT

位/位域	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

28.3.3 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
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MUX0 [4]	MUX1 [4]	Res						MUX0[3:0]				MUX1[3:0]			

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为 0	RW
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 参考表 28-1 PLAxMUX0 输入选择	RW
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 参考表 28-2 PLAxMUX1 输入选择	RW
b13~b8	Res	保留位	读出时为 0,写入时写 0	RW
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 参考表 28-1 PLAxMUX0 输入选择	RW
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 参考表 28-2 PLAxMUX1 输入选择	RW

28.3.4 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
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								LUTCTL[7:0]							

位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	读出时为 0,写入时写 0	RW
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	RW

28.3.5 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
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								OSEL	OEN	Res					

位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	读出时为 0,写入时写 0	RW
b7	OSEL	输出选择	0: LUT 的结果不作为 PLAx 的输出 1: LUT 的结果作为 PLAx 的输出	RW
b6	OEN	输出使能	0: PLAx 输出禁止 1: PLAx 输出允许	RW
b5~b0	Res	保留位	读出时为 0,写入时写 0	RW

28.4 注意事项

用户禁止访问 0x4004 FD00~0x4004 FFFF 空间。

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 Combinational 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 Combinational Loop 情况。

PLA 模块仅支持单个单元的同步配置，多个单元间的时序关系需要根据实际情况由用户设置。

29 串行外设接口 (SPI)

29.1 简介

本产品搭载 1 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。

29.2 SPI 主要特性

表 29-1 SPI 的特性要点

要点	描述
串行通信功能	<ul style="list-style-type: none"> 支持4线式SPI模式和3线式时钟同步运行模式 支持全双工和只发送两种通信方式 可调整通信时钟SCK的极性和相位
数据格式	<ul style="list-style-type: none"> 可选择数据移位顺序:MSB开始/LSB开始 可选择数据宽度:4/5/6/7/8/9/10/11/12/13/14/15/16/20/24/32位 单次最多可传送或接收4帧宽度为32位的数据
波特率	<ul style="list-style-type: none"> 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 从机模式下允许的最大波特率为PCLK1的6分频
数据缓冲	<ul style="list-style-type: none"> 带有16字节的数据缓冲区域 支持双重缓冲
错误监测	<ul style="list-style-type: none"> 模式故障错误监测 数据过载错误监测 数据欠载错误监测 奇偶校验错误监测
片选信号控制	<ul style="list-style-type: none"> 每个通道配置四根片选信号线 可对片选信号和通信时钟的相对时序关系进行调整 可对连续两次通信之间的片选信号无效时间进行调整 极性可调
主机模式下的传输控制	<ul style="list-style-type: none"> 通过将数据写入数据寄存器启动传输 通信自动挂起功能
中断	<ul style="list-style-type: none"> 接收数据区域已满 发送数据区域已空 SPI错误 (模式/过载/欠载/奇偶校验) SPI空置 传输完成 (仅为事件源)
低功耗控制	<ul style="list-style-type: none"> 可设置模块停止
其他功能	<ul style="list-style-type: none"> SPI初始化功能

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】。

29.3 SPI 功能说明

29.3.1 SPI 系统框图

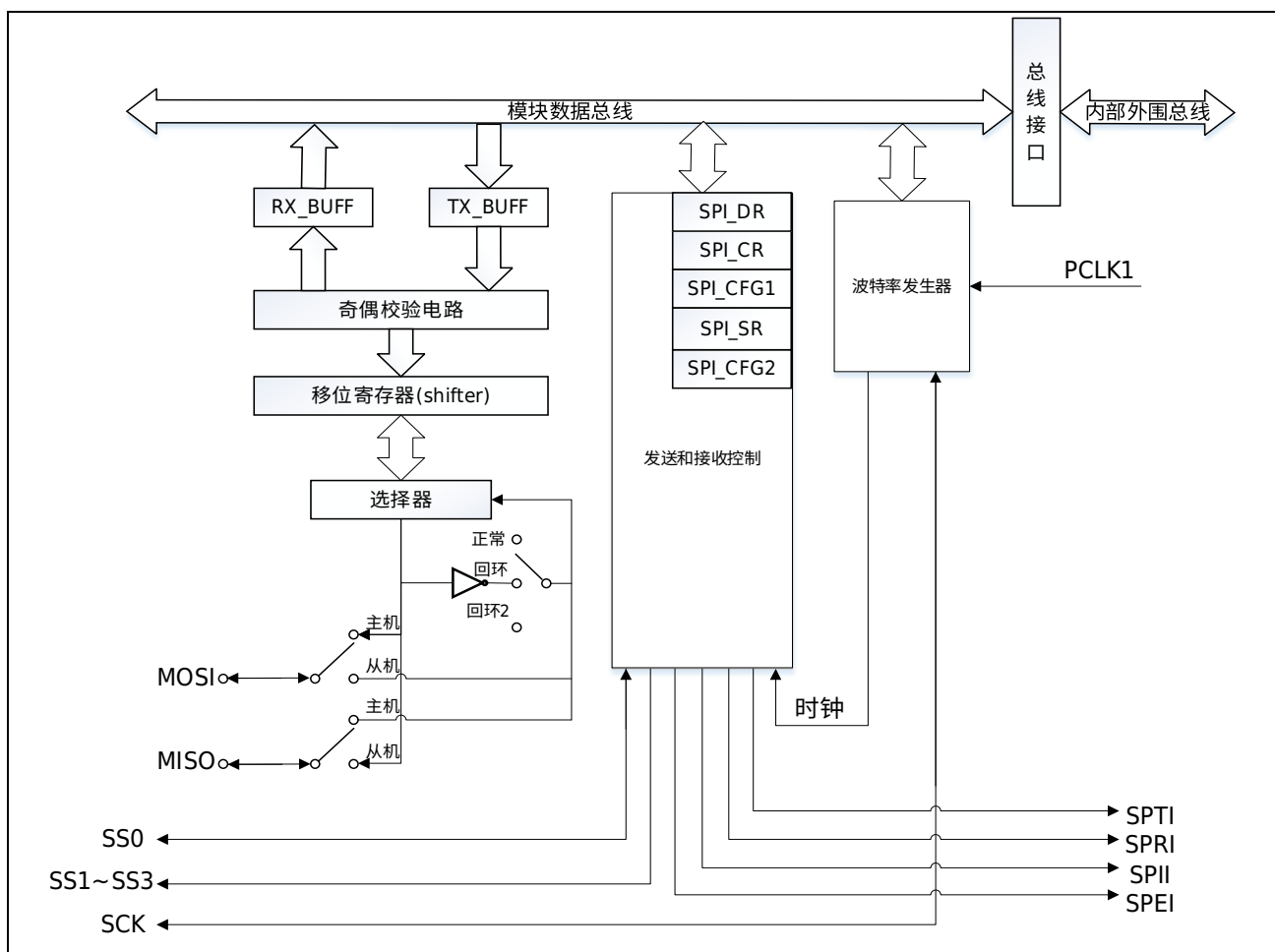


图 29-1 系统框图

表 29-2 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

29.3.2 SPI 动作系统说明

29.3.2.1 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 29-3 所示。

表 29-3 主机模式时 SPI 管脚状态说明

模式		管脚名	管脚状态
SPI动作 (SPIMDS=0)	主机模式 (MSTR=1、 MODFE=0)	SCK	输出
		SS0~SS3	输出
		MOSI	输出
		MISO	输入
时钟同步运行 (SPIMDS=1)	主机模式 (MSTR=1)	SCK	输出
		SS0~SS3 (不使用)	Hi-Z (可作为通用I/O)
		MOSI	输出
		MISO	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 GPIO 寄存器【通用控制寄存器 (PCRxy)】。

29.3.2.2 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 29-4 所示。

表 29-4 从机模式时 SPI 管脚状态说明

模式		管脚名	管脚状态
SPI动作 (SPIMDS=0)	从机模式 (MSTR=0、 MODFE=0)	SCK	输入
		SS0	输入
		SS1~SS3 (不使用)	Hi-Z (可作为通用I/O)
		MOSI	输入
		MISO	输出/Hi-Z
时钟同步运行 (SPIMDS=1)	从机模式 (MSTR=0)	SCK	输入
		SS0~SS3 (不使用)	Hi-Z (可作为通用I/O)
		MOSI	输入
		MISO	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 GPIO 寄存器【通用控制寄存器 (PCRxy)】。

29.3.2.3 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。

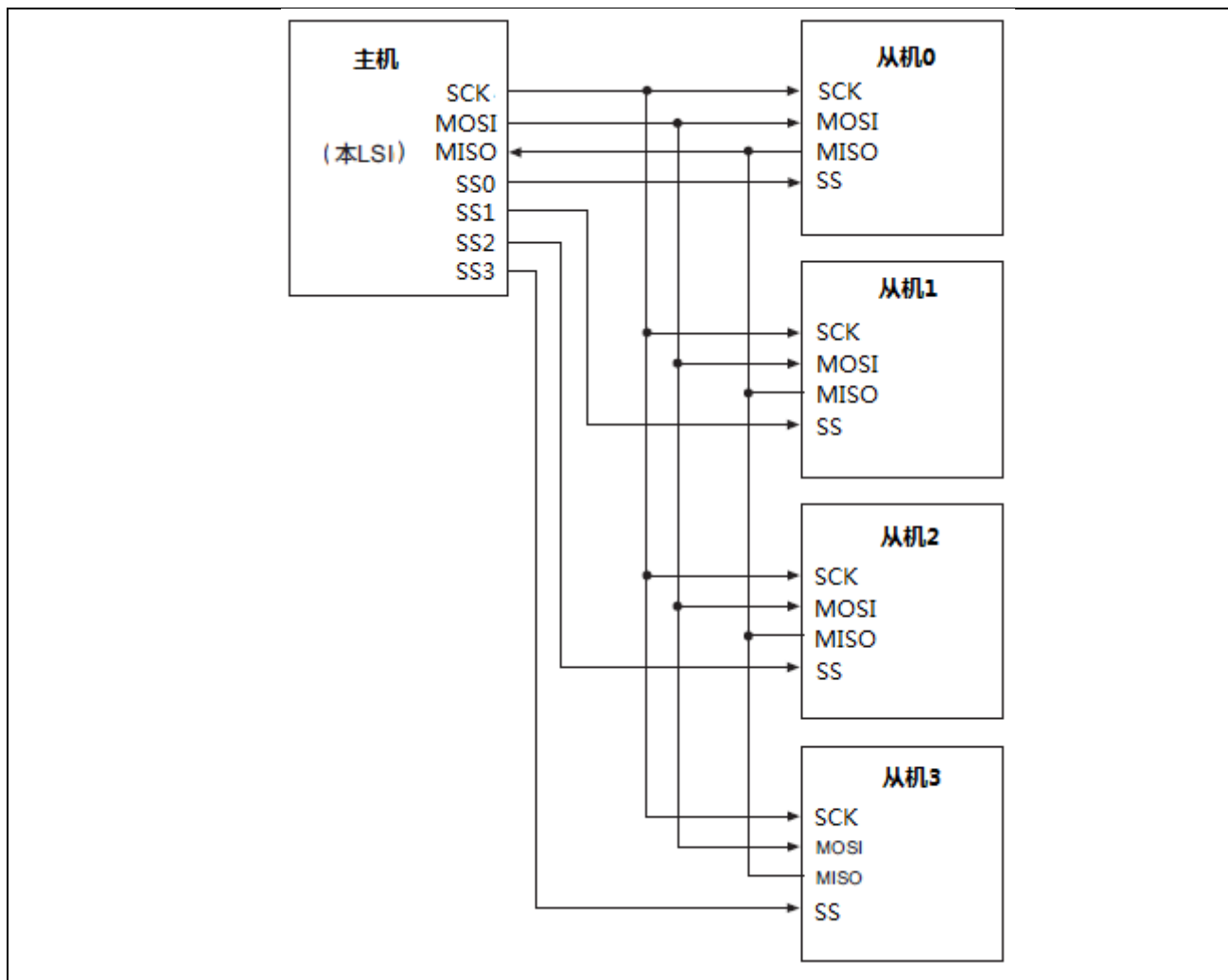


图 29-2 主机模式结构

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。

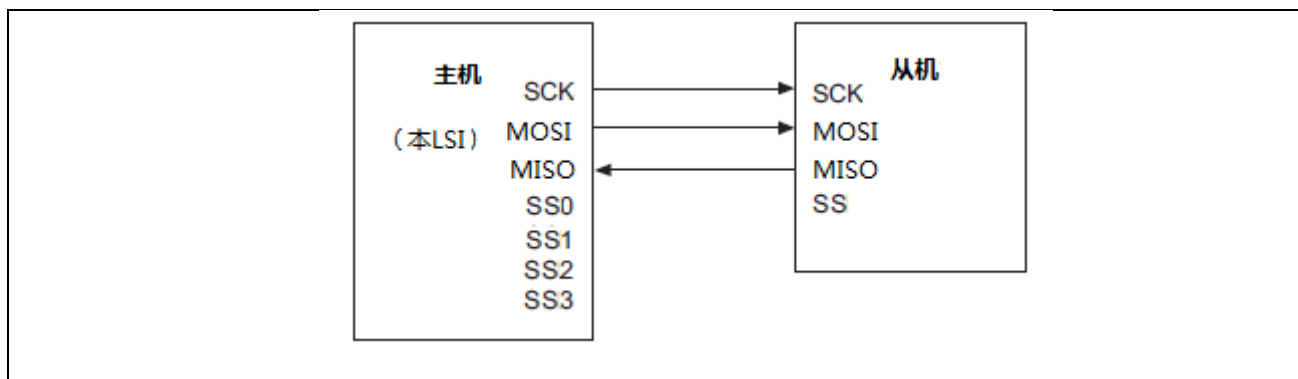


图 29-3 三线式时钟同步运行

29.3.3 数据通信说明

29.3.3.1 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。

$$\text{波特率} = \frac{f_{\text{PCLK1}}}{2^{N+1} \times (M+1)}$$

表 29-5 部分设定值的位速

MBR[1:0] 位的设定值 (N)	CLKDIV[3:0] 位的设定值 (M)	分频比	波特率			
			PCLK1=5 MHz	PCLK1=10 MHz	PCLK1=20 MHz	PCLK1=40 MHz
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

29.3.3.2 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 DSIZ[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.DSIZ[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 29-4 所示。

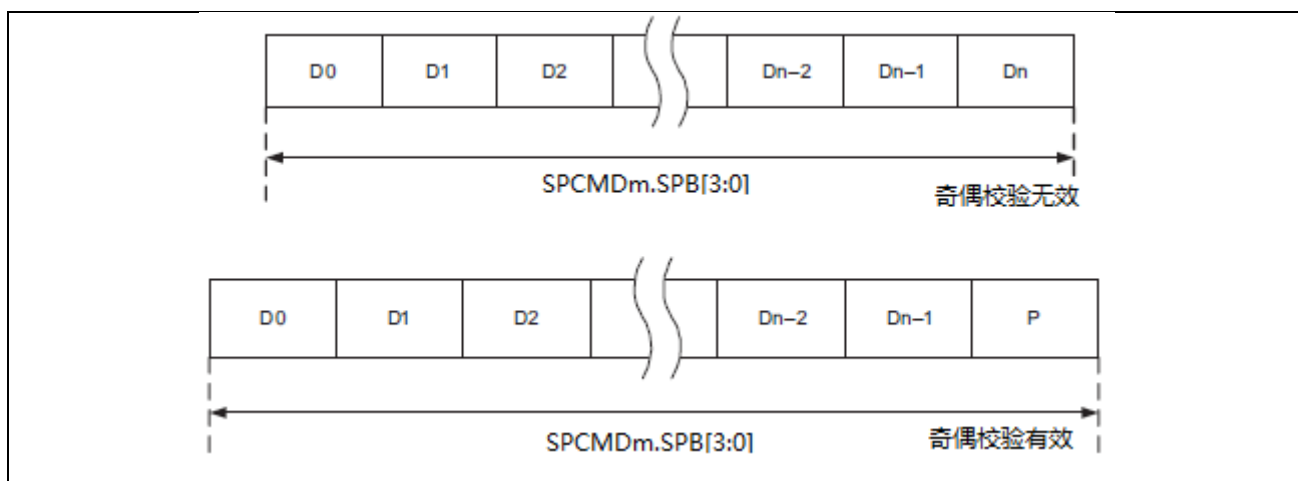


图 29-4 数据格式

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：

1) 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。

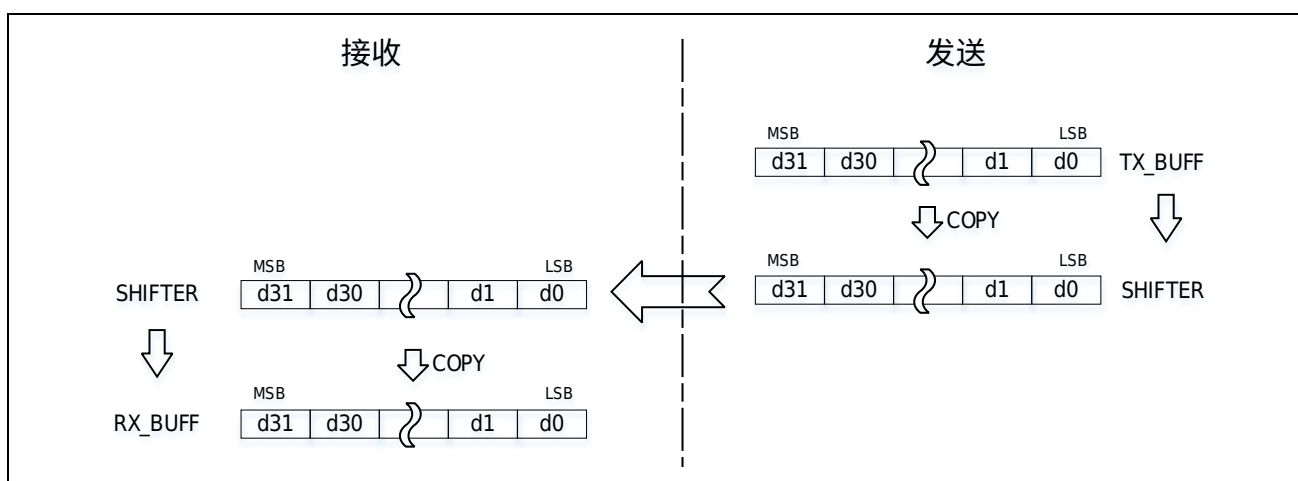


图 29-5 MSB 先传，奇偶校验无效

2) LSB 先传，奇偶校验无效

发送时，数据 d31~d0 按照 d0~d31 的顺序从 TX_BUFF 复制到 shifter，按照 d0~d31 的顺序从 shifter 的最高位移出；

接收时，数据 d0~d31 从 shifter 的最低位移入，等到数据全部移入后，按照 d31~d0 的顺序从 shifter 复制到 RX_BUFF。

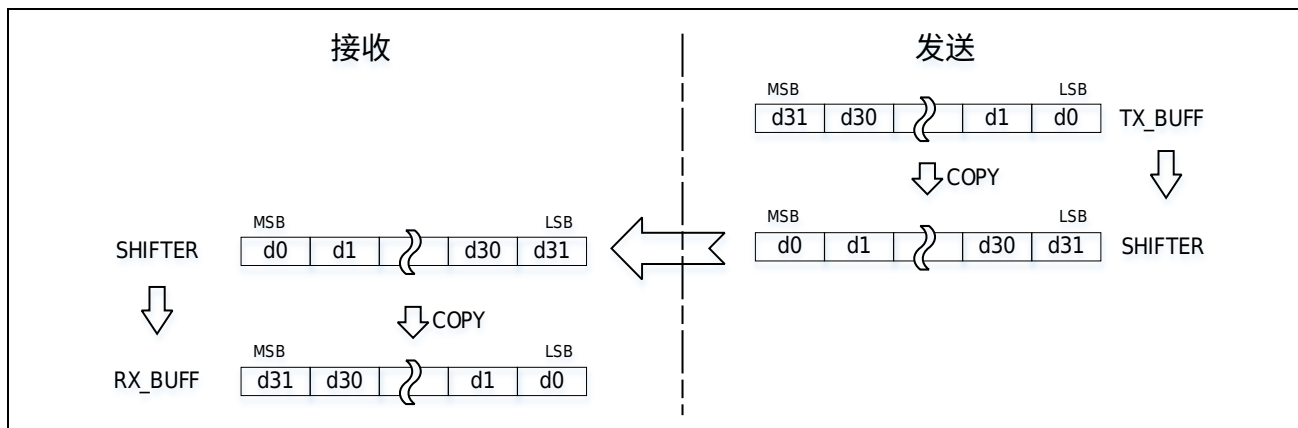


图 29-6 LSB 先传，奇偶校验无效

3) 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。

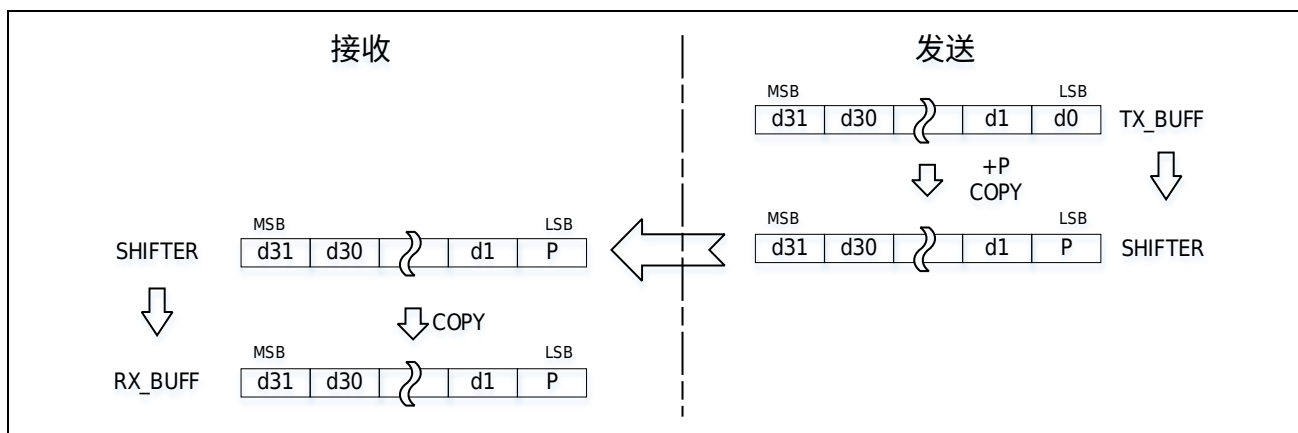


图 29-7 MSB 先传，奇偶校验有效时

4) 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。

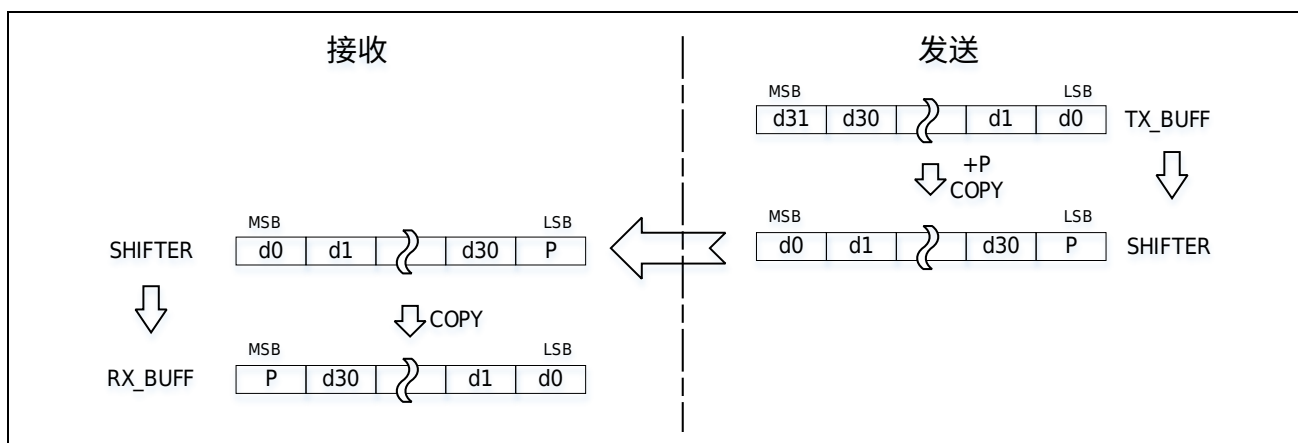


图 29-8 LSB 先传，奇偶校验有效时

29.3.3.3 传送格式

1) 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 29-9 是 CPHA=0 时 SPI 的传送时序图。当 SSI 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 SSI 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。

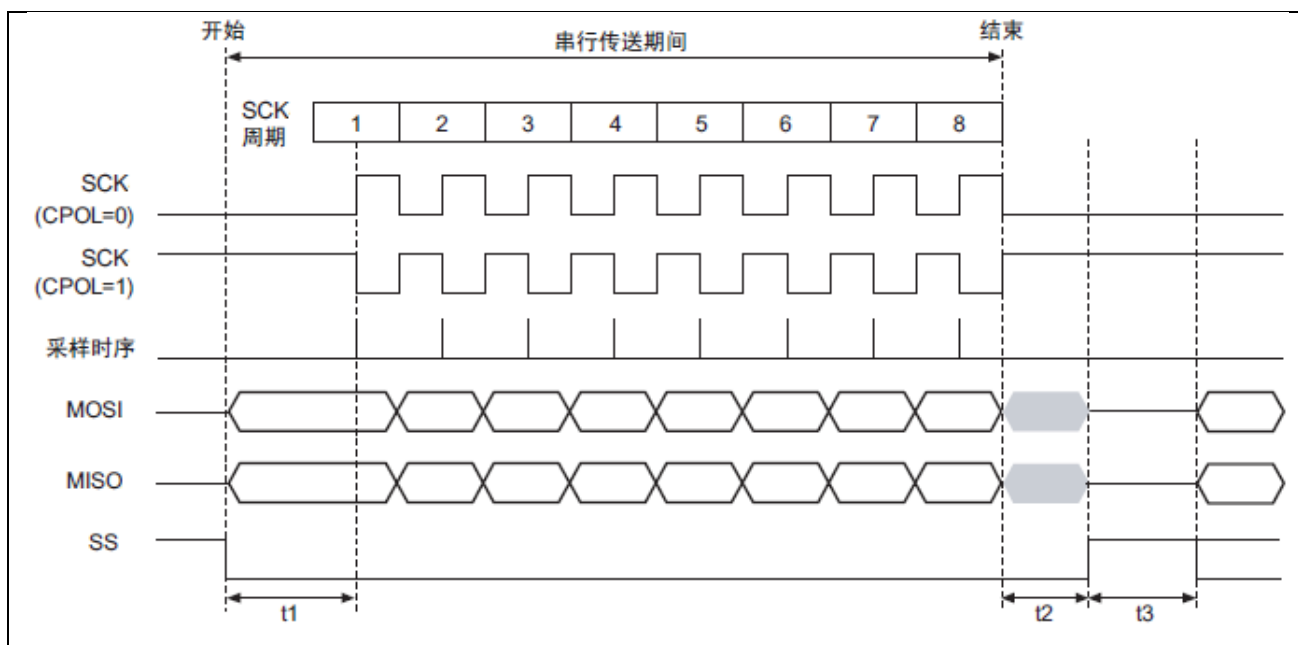


图 29-9 数据传送格式图 (CPHA=0)

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。

2) 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 29-10 是 CPHA=1 时 SPI 的传送时序图。MOSI/ MISO 在 SSi 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。

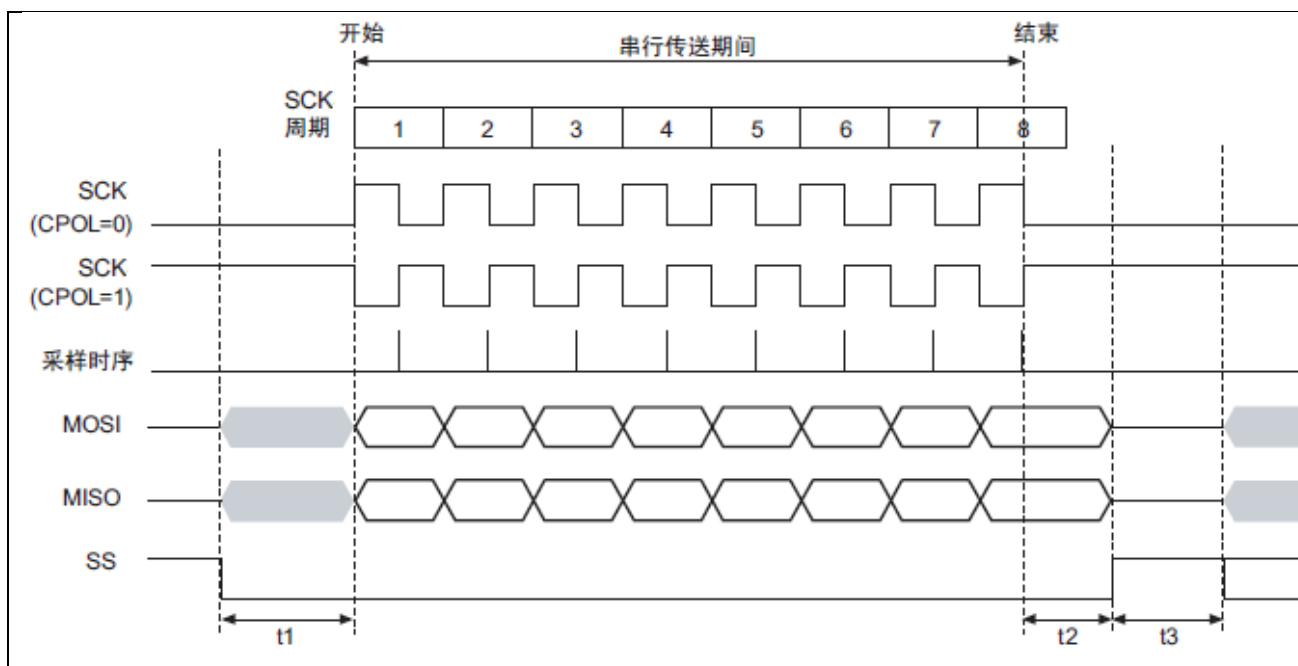


图 29-10 数据传送格式 (CPHA=1)

t1、t2、t3 和 CPHA 位为 0 的情况相同。

29.3.3.4 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。

1) 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 29-11 所示，SPI_CFG1.FTHLV[1:0]位为 0b00，SPI_CFG2.CPHA 位为 1 并且 SPI_CFG2.CPOL 位为 0，SPI 进行 8 位串行传送。

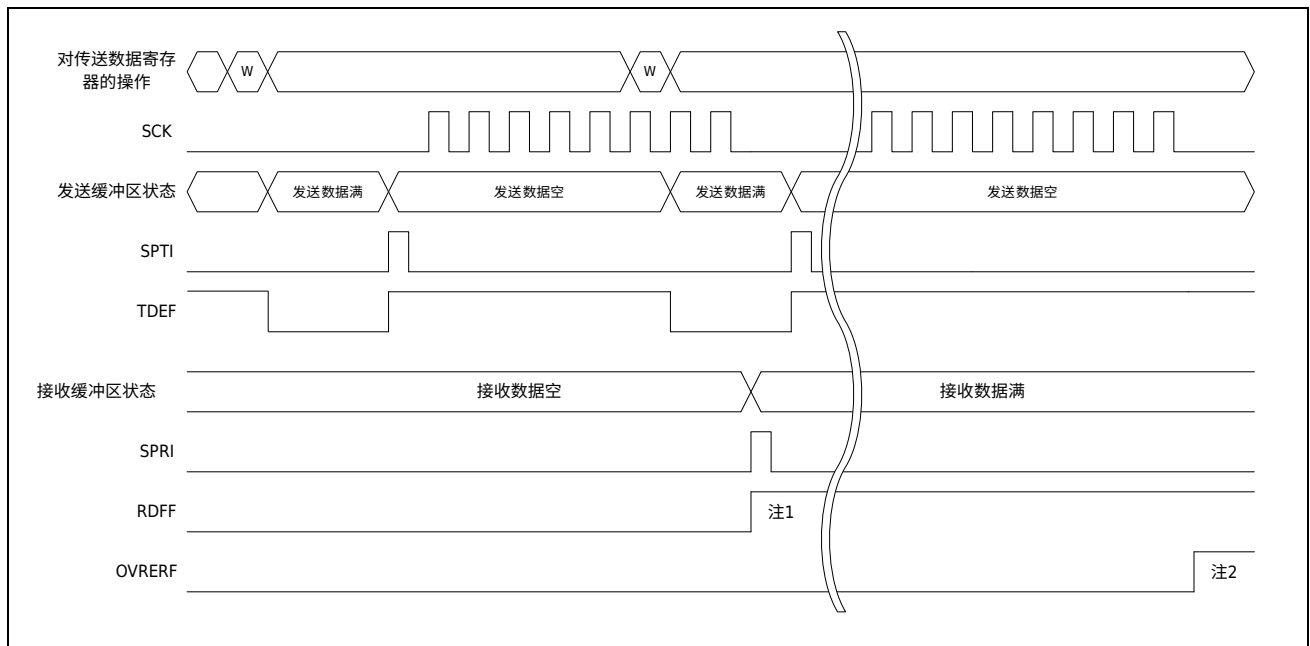


图 29-11 全双工同步串行通信

注：

1. 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)。
2. 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。

2) 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 29-12 所示，SPI_CFG1.FTHLV[1:0]位为 0b00，SPI_CFG2.CPHA 位为 1 并且 SPI_CFG2.CPOL 位为 0，SPI 进行 8 位串行传送。

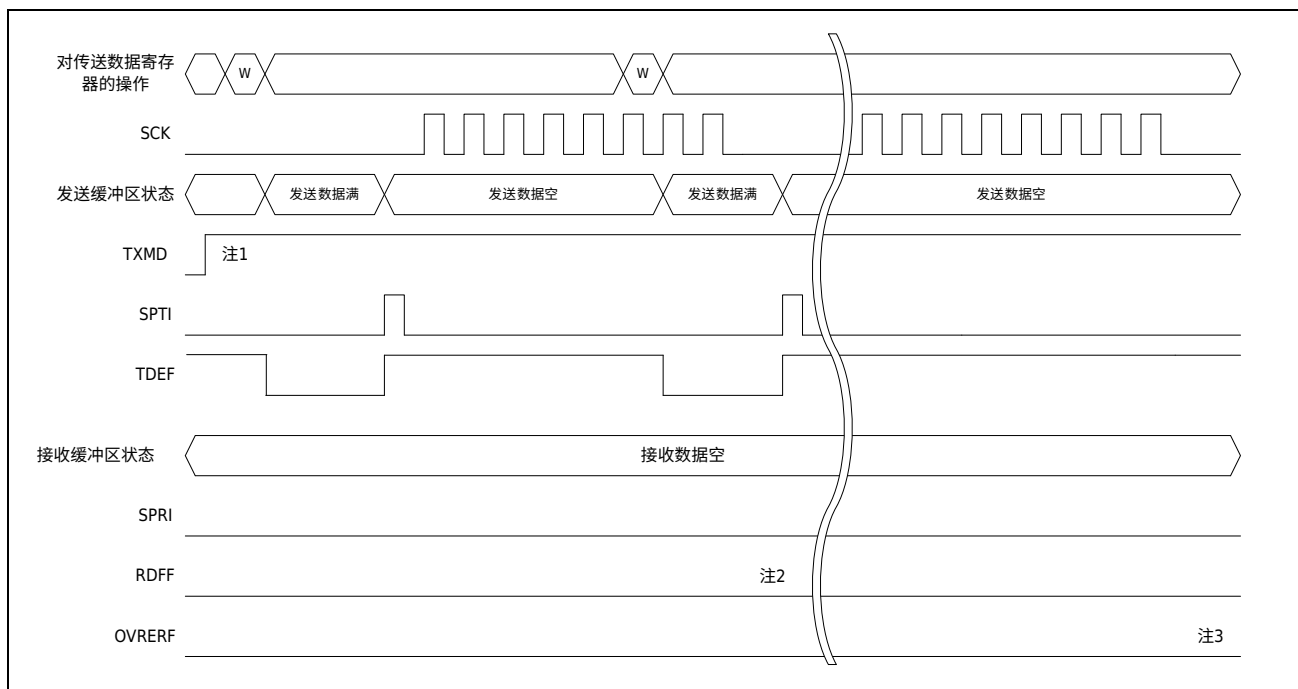


图 29-12 只进行发送通信

注:

1. 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 *RDFS* 为 0) 且未出现数据过载错误 (即 *OVRERF* 为 0)。
2. 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, *RDFS* 将始终保持 0 的状态。
3. 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。

29.3.3.5 通信间隔时间

1) 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 $1 \cdot SCK + 2 \cdot PCLK1$ 。如图 29-13 所示, *SPI_CFG2.CPHA* 位为 1 并且 *SPI_CFG2.CPOL* 位为 0, SPI 进行串行传送。

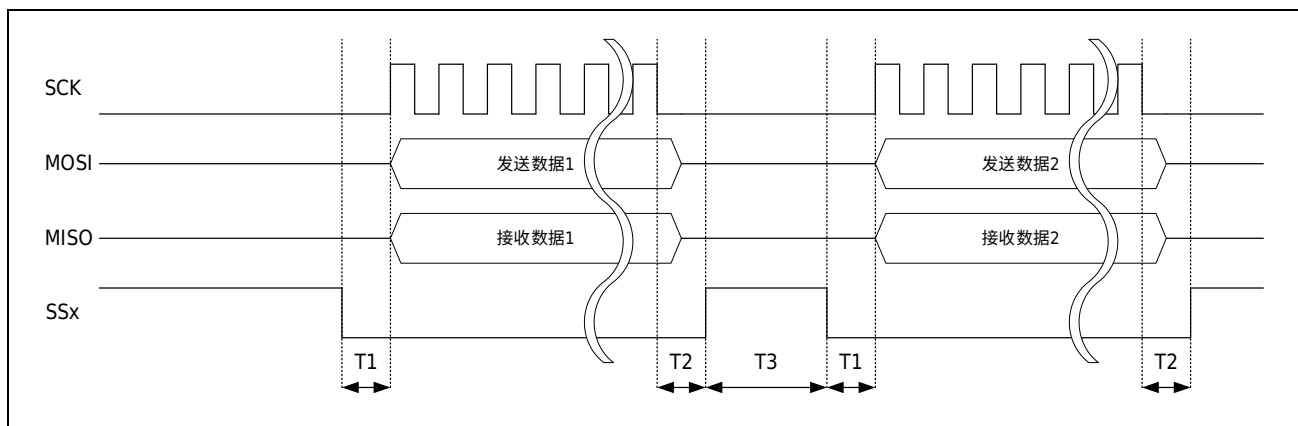


图 29-13 常规通信

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），t2 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），t3 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）。

注：

- 1) 当通信速率为 PCLK1 的 2 分频（SPI_CFG2.MBR[1:0]=0b00 且 SPI_CFG2.CLKDIV[3:0]=0b0000）时，请将 t3 配置为 $2 * SCK + 2 * PCLK1$ 及以上，即 SPI_CFG2.MIDIE=1 且 SPI_CFG1.MIDI[2:0]=1~7。
- 2) 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 29-14 所示，SPI_CFG2.CPHA 位为 1 并且 SPI_CFG2.CPOL 位为 0，SPI 进行串行传送。

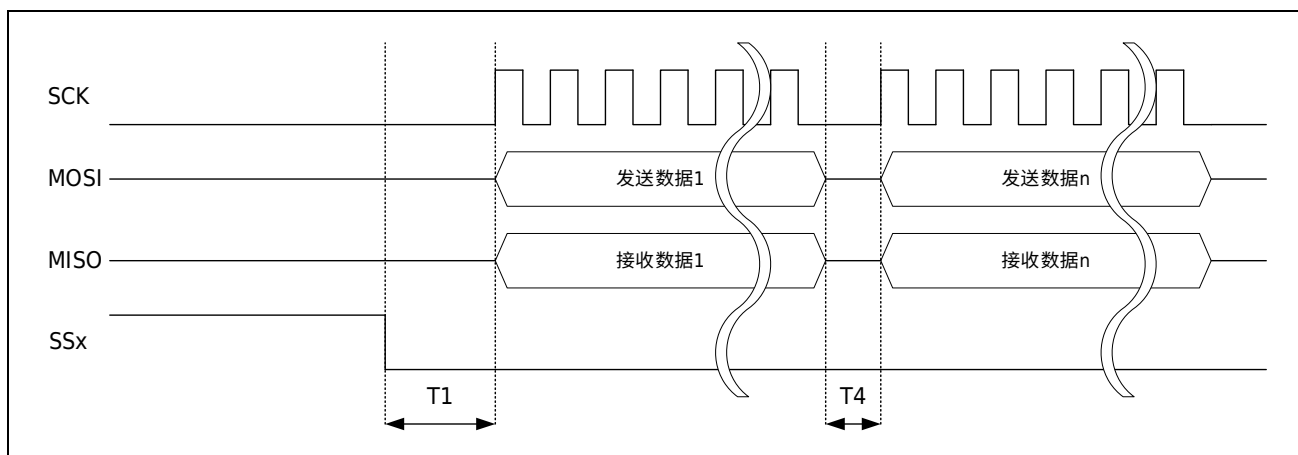


图 29-14 连续通信

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:

- 1) 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。
- 2) 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。
- 3) 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。
- 4) 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。
- 5) 连续通信模式下, 也可以通过 MSS1 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。

29.3.4 运行说明

29.3.4.1 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 29-6 所示。

表 29-6 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	2种 (注3)
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

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)
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	无	有	无	有

注:

- 1) 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。
- 2) 当 SPI_CR.PAE 位为 0 时, 不进行奇偶校验错误的检测。
- 3) 当 SPI_CFG2.CPHA 位为 0 时, 仅在连续传输模式下有效, 具体设定步骤请参考【时钟同步运行模式中的从机动作】

29.3.4.2 SPI 运行模式时的主机动作

- 1) 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【传送格式】。

- 2) 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, 各种迟延时间的设定等;

- 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。

29.3.4.3 SPI 运行模式时的从机动作

1) 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【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` 将无法开始正常串行传送。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。

2) `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。

29.3.4.4 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。

1) `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【传送格式】。

2) 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，各种延迟时间的设定等；
- 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。

29.3.4.5 时钟同步运行模式中的从机动作

1) 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，必须按照以下的顺序设置才能进行正常通信。

- ① 进行通信设置，并将 SPI_CR 的 SPE 位设置为 1；
- ② 将 SPI_CFG2.CPHA 设置为 1；
- ③ 将 SPI_CFG1.CTMDS 设置为 1；
- ④ 将需要发送的数据写入 SPI_DR；
- ⑤ 将 SPI_CFG2.CPHA 设置为 0；

⑥ 等待主机时钟开始通信。

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。

2) 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。

29.3.4.6 几种 SPI 动作的处理流程

1) 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 硬件产生 SPI 闲置状态中断；
- ⑥ 将 SPE 置 0，停止 SPI 动作，同时将 IDIE 清零。

2) 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。

3) 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 SS0 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。

29.3.5 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。

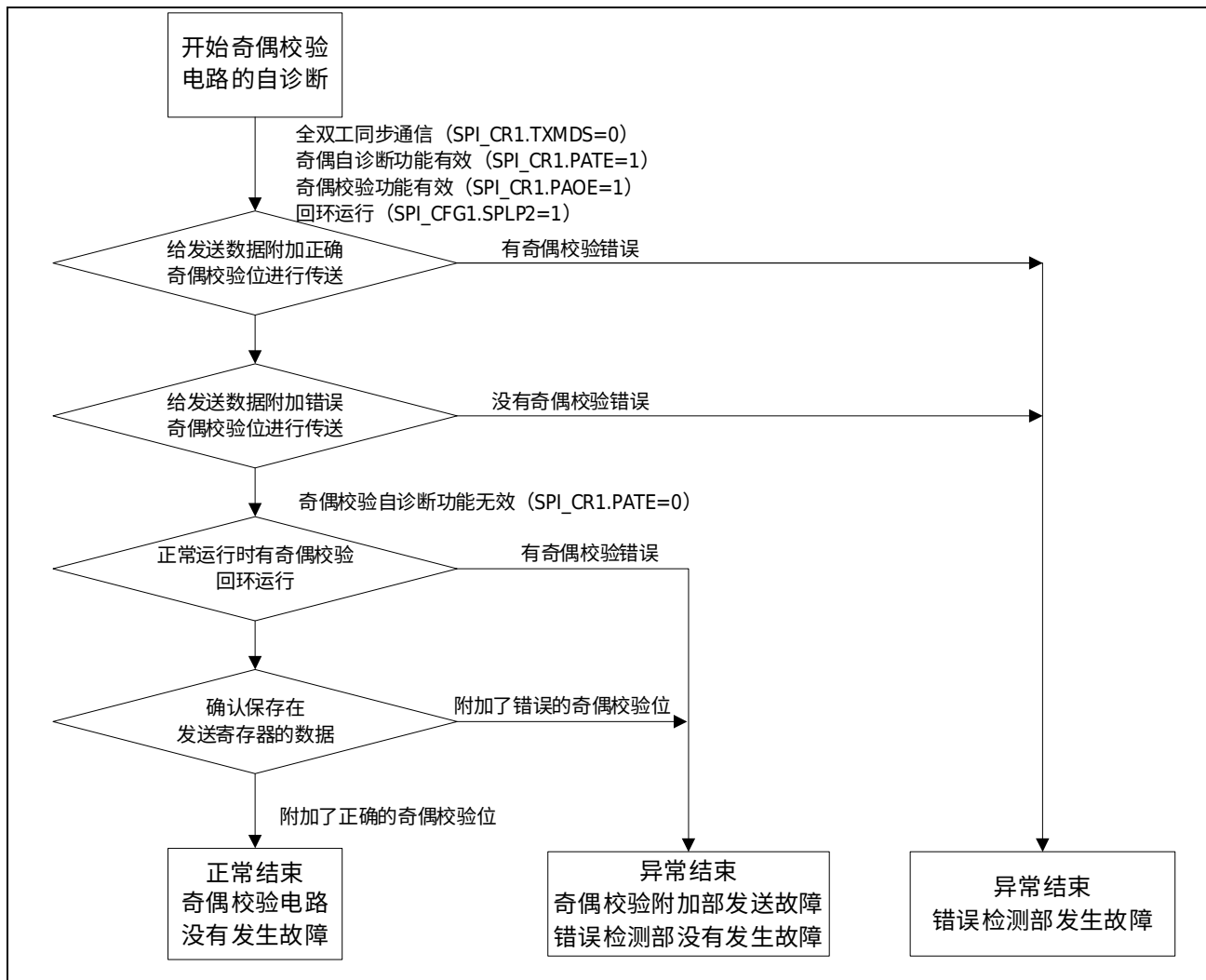


图 29-15 奇偶校验流程

29.3.6 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 29-7 所示。

表 29-7 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
		写数据丢失	
②	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
③	在从机模式中，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	
④	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
		丢失发送和接收数据	
		停止SPI功能	
⑤	在接收缓冲器满的状态下结束串行传送	保持接收缓冲器内容	过载错误
		接收数据丢失	
⑥	在进行全双工同步串行通信，并且奇偶校验功能有效的状态下，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

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。

29.3.6.1 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。

29.3.6.2 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.MODFREF 清零后，再把 SPI_CR.SPE 置 1。

29.3.6.3 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。

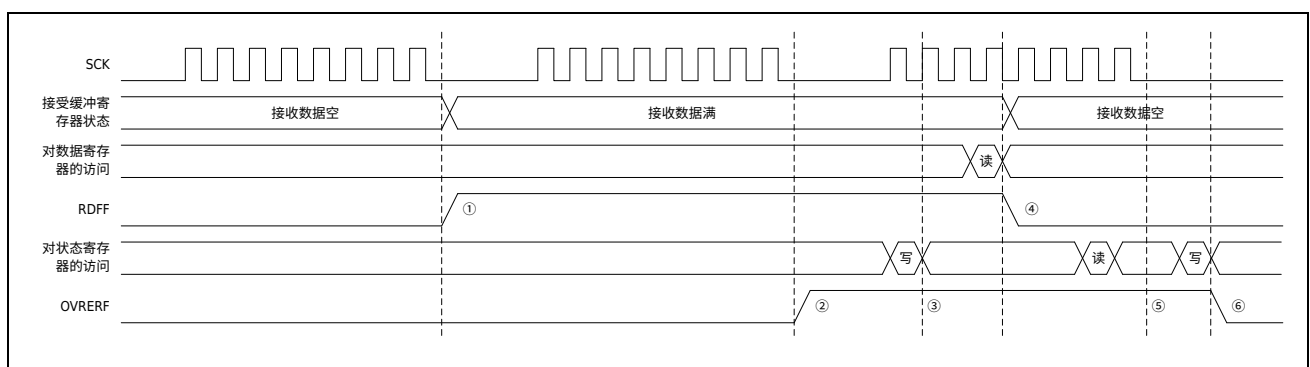


图 29-16 过载错误处理

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 29-17 和图 29-18。

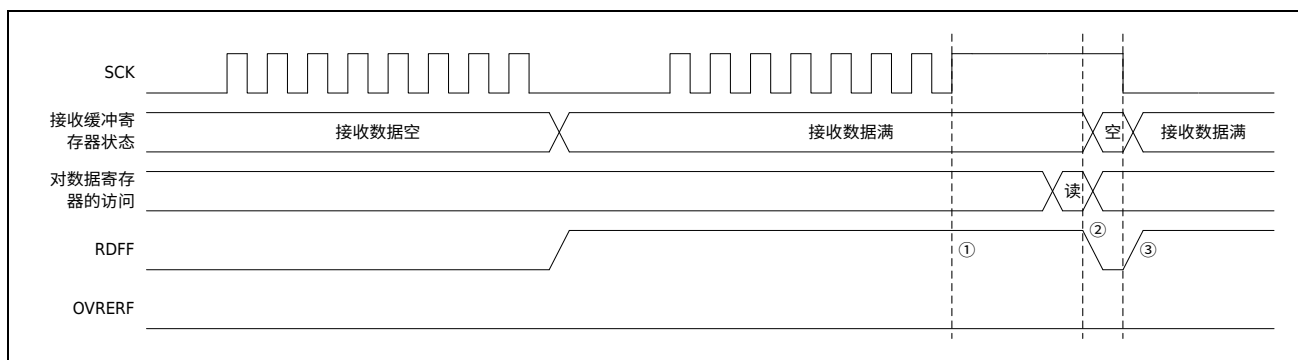


图 29-17 启用时钟自动停止功能时的动作示意图 (CPHA=1)

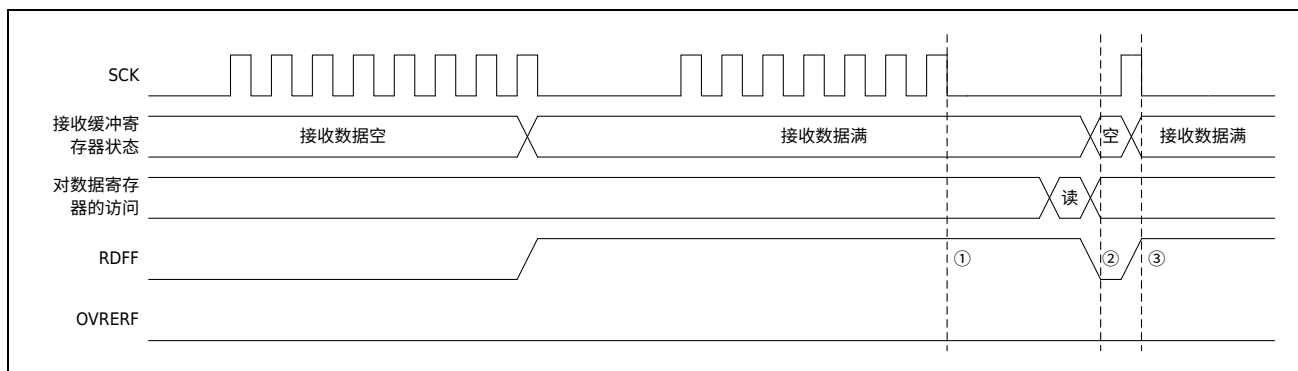


图 29-18 启用时钟自动停止功能时的动作示意图 (CPHA=0)

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。

29.3.6.4 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 29-19 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。

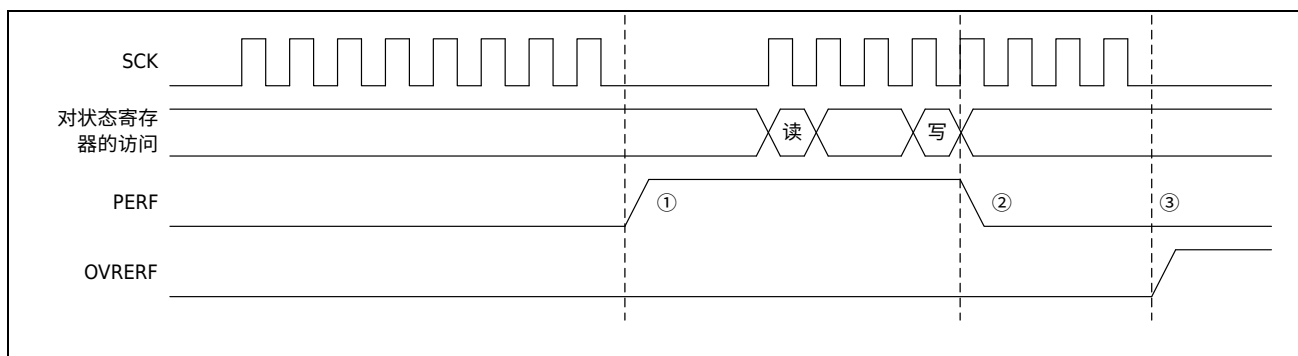


图 29-19 奇偶校验错误

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 PERF 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。

29.3.7 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。

29.3.7.1 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。

29.3.7.2 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。

29.4 中断和事件

29.4.1 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 29-8 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 29-8 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

29.4.2 事件

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）】。

29.5 寄存器说明

表 29-9 SPI 基地址

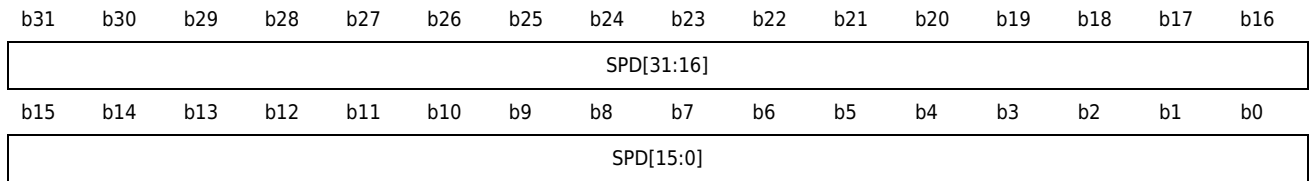
名称	基地址	描述
SPI	0x4001 C000	SPI基地址

表 29-10 SPI 寄存器列表一览表

寄存器	描述	偏移地址	位宽	复位值
SPI_DR	SPI数据寄存器	0x00	32	0x0000 0000
SPI_CR	SPI控制寄存器	0x04	32	0x0000 0000
SPI_CFG1	SPI通信配置寄存器1	0x0C	32	0x0000 0010
SPI_SR	SPI状态寄存器	0x14	32	0x0000 0020
SPI_CFG2	SPI通信配置寄存器2	0x18	32	0x0000 0F1D

29.5.1 SPI 数据寄存器 (SPI_DR)

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b0	SPD[31:0]	串行数据	SPI数据存储	RW

29.5.2 SPI 控制寄存器 (SPI_CR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
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	Res	TXMDS	SPIMDS

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”，写入时写“0”	RW
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)	RW
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	RW
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	RW
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	RW
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	RW
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	RW
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	RW
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	RW
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	RW
b6	SPE	SPI功能允许	0: SPI功能无效 1: SPI功能有效	RW
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)	RW
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)	RW
b3	MSTR	SPI主从模式选择	0: 从机模式 1: 主机模式	RW
b2	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
b0	SPIMDS	SPI模式选择	0: SPI运行 (4线式) 1: 时钟同步运行 (3线式)	RW

29.5.3 SPI 通信配置寄存器 1 (SPI_CFG1)

复位值: 0x0000 0010

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	MIDI[2:0]			Res	MSSDL[2:0]			Res	MSSI[2:0]			Res			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CLKDIV[3:0]				SS3P V	SS2P V	SS1P V	SS0P V	Res	SPRD TD	Res			CTMD S	FTHLV[1:0]	

位/位域	标记	位名	功能	读写
b31	Res	保留位	读出时为“0”，写入时写“0”	RW
b30~b28	MIDI[2:0]	主机下次存取数据间隔闲置时间设定位	000: 1个SCK+2个PCLK1 001: 2个SCK+2个PCLK1 010: 3个SCK+2个PCLK1 011: 4个SCK+2个PCLK1 100: 5个SCK+2个PCLK1 101: 6个SCK+2个PCLK1 110: 7个SCK+2个PCLK1 111: 8个SCK+2个PCLK1	RW
b27	Res	保留位	读出时为“0”，写入时写“0”	RW
b26~b24	MSSDL[2:0]	主机SS无效延迟设定位	000: 1个SCK 001: 2个SCK 010: 3个SCK 011: 4个SCK 100: 5个SCK 101: 6个SCK 110: 7个SCK 111: 8个SCK	RW
b23	Res	保留位	读出时为“0”，写入时写“0”	RW
b22~b20	MSSI[2:0]	主机SS闲置时间设定位	000: 1个SCK 001: 2个SCK 010: 3个SCK 011: 4个SCK 100: 5个SCK 101: 6个SCK 110: 7个SCK 111: 8个SCK	RW
b19~b16	Res	保留位	读出时为“0”，写入时写“0”	RW
b15~b12	CLKDIV	通讯基准时钟选择位	0000: PCLK1/2 0001: PCLK1/4 0010: PCLK1/6 0011: PCLK1/8 0100: PCLK1/10 0101: PCLK1/12 0110: PCLK1/14 0111: PCLK1/16 1000: PCLK1/18	RW

			1001: PCLK1/20	
			1010: PCLK1/22	
			1011: PCLK1/24	
			1100: PCLK1/26	
			1101: PCLK1/28	
			1110: PCLK1/30	
			1111: PCLK1/32	
b11	SS3PV	SS3信号 极性设定	0: SS3信号的低电平有效 1: SS3信号的高电平有效	RW
b10	SS2PV	SS2信号 极性设定	0: SS2信号的低电平有效 1: SS2信号的高电平有效	RW
b9	SS1PV	SS1信号 极性设定	0: SS1信号的低电平有效 1: SS1信号的高电平有效	RW
b8	SS0PV	SS0信号 极性设定	0: SS0信号的低电平有效 1: SS0信号的高电平有效	RW
b7	Res	保留位	读出时为“0”，写入时写“0”	RW
b6	SPRDTD	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)	RW
b5~b3	Res	保留位	读出时为“0b010”，写入时写“0b010”	RW
b2	CTMDS	连续通信模式选择	0: 常规通信模式 1: 连续通信模式	RW
b1~b0	FTHLV[1:0]	帧数设定位	00: 1 帧 01: 2 帧 10: 3 帧 11: 4 帧	RW

29.5.4 SPI 状态寄存器 (SPI_SR)

复位值: 0x0000 0020

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								RDF	FMIDLF	TDEF	UDRERF	PERF	MODFERF	IDLNF	OVRERF

位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	读出时为“0”，写入时写“0”	RW
b7	RDF	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	FMIDLF	连续模式主机IDLE状态标志	0: 连续模式主机处于正常动作状态 1: 连续模式主机处于空闲状态	RW
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。	RW
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	RW
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	RW
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	RW

29.5.5 SPI 通信配置寄存器 2 (SPI_CFG2)

复位值: 0x0000 0F1D

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
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]			Res	MBR[1:0]		CPOL	CPHA	

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”，写入时写“0”	RW
b15	MSSIE	SCK延迟允许	0: SCK 延迟为1 个SCK 1: SCK 延迟为SSI的设置值	RW
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的设置值	RW
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的设置值	RW
b12	LSBF	SPI LSB first 位	0: MSB first 1: LSB first	RW
b11~b8	DSIZE[3:0]	SPI数据长度设定位	0000: 4位 0001: 5位 0010: 6位 0011: 7位 0100: 8位 0101: 9位 0110: 10位 0111: 11位 1000: 12位 1001: 13位 1010: 14位 1011: 15位 1100: 16位 1101: 20位 1110: 24位 1111: 32位	RW
b7~b5	SSA[2:0]	SS信号有效设定位	000: SS0 001: SS1 010: SS2 011: SS3 其他: 禁止设定	RW
b4	Res	保留位	读出时为“1”，写入时写“1”	RW
b3~b2	MBR[1:0]	位速率分频设定位	00: 选择基准通讯时钟 01: 选择基准通讯时钟的2分频 10: 选择基准通讯时钟的4分频 11: 选择基准通讯时钟的8分频	RW
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	RW

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	RW
----	------	----------	--	----

30 控制器局域网（MCAN）

30.1 概述

控制器局域网（CAN）子系统由两个 CAN 模块、一个共享消息 RAM 存储器组成。关于这三个组成部分的基地址，请参考章节【存储器映射】。

两个 CAN 模块（MCAN1 和 MCAN2）均符合 ISO 11898-1: 2015（CAN 协议规范第 2.0 版 A、B 部分）和 CAN FD 协议规范第 1.0 版（CAN with Flexible Data-Rate Specification Version 1.0）。

2KB 的消息 RAM 存储器可实现过滤器（Rx Filter）、接收 FIFO（Rx FIFO）、接收缓冲区（Rx Buffer）、发送事件 FIFO（Tx Event FIFO）、发送缓冲区（Tx Buffer）功能。该消息 RAM 在 MCAN1 和 MCAN2 模块之间共用。

30.2 主要特性

- 适用 ISO 11898-1:2015 标准
- CAN FD 最多支持收发 64 个数据字节
- 支持 CAN 错误日志
- 支持 AUTOSAR 和 SAE J1939
- 支持接收过滤功能
- 两个可配置的接收 FIFO
- 接收到高优先级消息时单独发出信号指示
- 最多 64 个专用接收缓冲区
- 最多 32 个专用发送缓冲区
- 可配置的发送 FIFO 或队列
- 可配置的发送事件 FIFO
- 主机 CPU 可直接访问消息 RAM
- MCAN1 和 MCAN2 两个模块共享同一个消息 RAM
- 支持可编程回环测试模式
- 支持可屏蔽模块中断
- 两个时钟域：CAN 内核时钟和 AHB 总线接口时钟

30.3 功能说明

30.3.1 MCAN 系统框图

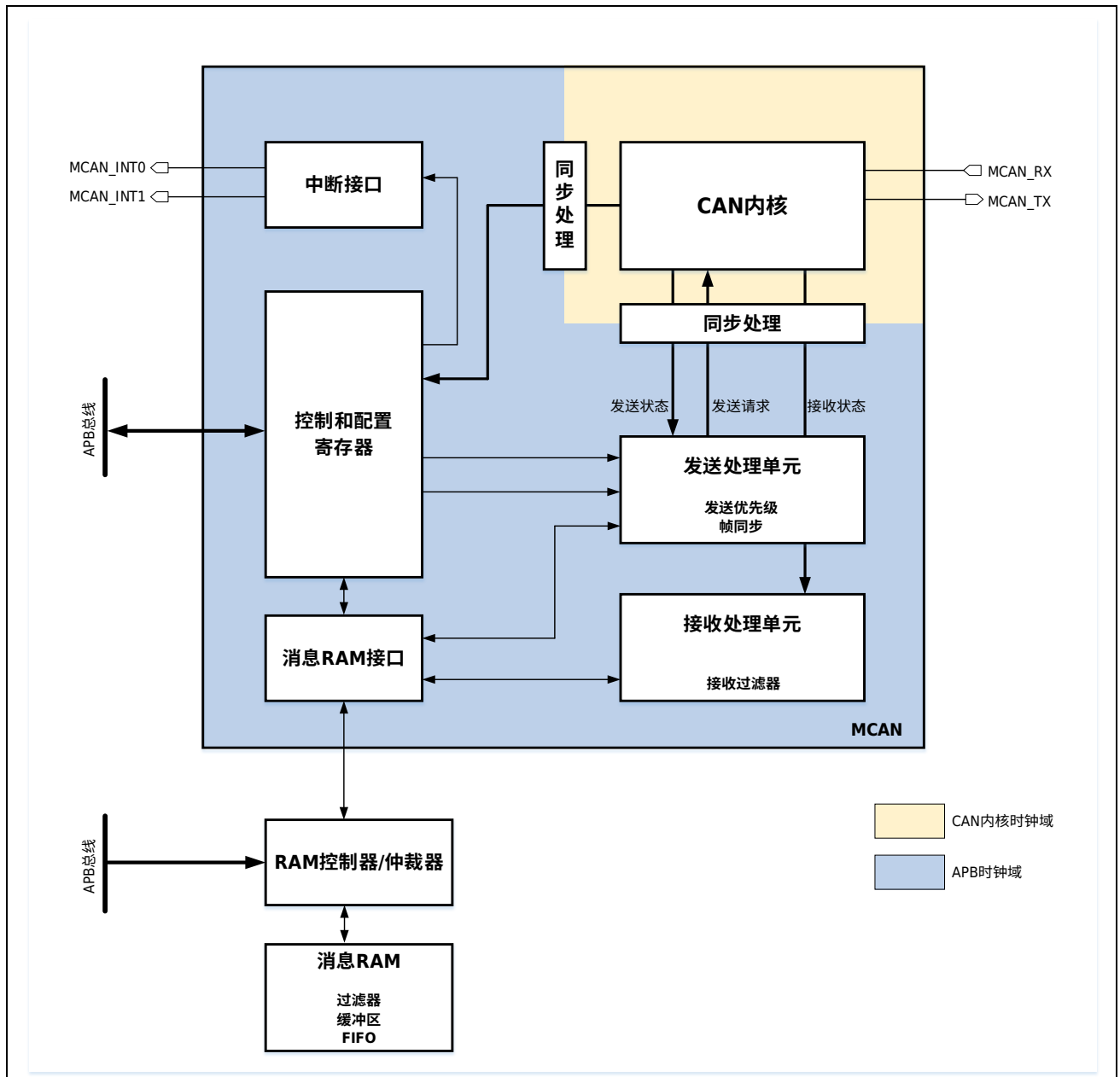


图 30-1 MCAN 系统框图

CAN 内核

CAN 内核包含协议控制器、接收与发送移位寄存器。它可处理 ISO 11898-1:2015 的所有协议功能，并支持 11 位标准 ID 和 29 位扩展 ID。

同步

同步模块将来自 AHB 时钟域的信号同步到 CAN 内核时钟域，以及将来自 CAN 内核时钟域的信号同步到 AHB 时钟域。

AHB 时钟域的时钟为 MCAN 控制逻辑时钟，CAN 内核时钟域的时钟为 MCAN 通信时钟，请参考章节【工作时钟规格】。为了保证 MCAN 通信的稳定性，控制逻辑时钟频率必须大于等于通信时钟频率。

发送处理单元

该处理单元控制消息从外部消息 RAM 到 CAN 内核的传输。最多可配置 32 个发送缓冲区进行发送。发送缓冲区可用作专用发送缓冲区、发送 FIFO（或队列），或二者的组合。发送事件 FIFO 将发送时间戳与对应的消息 ID 存储在一起。另外还支持取消发送。

接收处理单元

该处理单元控制接收到的消息从 CAN 内核到外部消息 RAM 的传输。接收处理单元支持两个接收 FIFO（每个 FIFO 的大小均可配置）和最多 64 个专用接收缓冲区，用于存储所有通过接收过滤的消息。与接收 FIFO 有所不同，专用接收缓冲区仅用于存储特定 ID 的消息。接收时间戳与其消息存储在一起。对于 11 位标准 ID，最多可定义 128 个过滤器；对于 29 位扩展 ID，最多可定义 64 个过滤器。

AHB 接口

将 MCAN 连接至 AHB 总线。

消息 RAM

接口通过 RAM 控制器和仲裁器将 MCAN 访问连接到外部 2KB 消息 RAM。

双中断线

MCAN 提供 MCAN_INT0 和 MCAN_INT1 两条中断线。对 MCAN_ILE.EINT0 和 MCAN_ILE.EINT1 位进行编程，可分别使能或禁止这两条中断线。

30.3.2 引脚说明

表 30-1 MCAN 引脚说明

引脚名	方向	功能描述
MCAN_RX	输入	MCAN接收引脚
MCAN_TX	输出	MCAN发送引脚

30.3.3 工作模式

30.3.3.1 软件初始化

软件初始化是通过置位 MCAN_CCCR.INIT 开始的。软件复位、硬件复位或进入 Bus_Off 会置位 MCAN_CCCR.INIT。当 MCAN_CCCR.INIT 被置位时，CAN 总线上传入和传出的消息都将停止，MCAN_TX 引脚的状态变为隐性（高电平）。错误逻辑管理单元（EML）的计数器保持不变。将 MCAN_CCCR.INIT 置位不会改变任何配置寄存器。将 MCAN_CCCR.INIT 清零可以完成软件初始化，随后，位流处理单元（BSP）等待总线上出现 11 个连续隐性位的序列（Bus_Idle），以此将其自身与 CAN 总线上的数据传输进行同步，然后才能参与总线活动并开始消息传输。

仅当 MCAN_CCCR.INIT 和 MCAN_CCCR.CCE 均置位时，才能访问 MCAN 配置寄存器。MCAN 配置寄存器如表 30-2 所示。

表 30-2 MCAN 配置寄存器列表

寄存器	说明
MCAN_DBTP	MCAN 数据位时间和预分频寄存器
MCAN_TEST	MCAN 测试寄存器
MCAN_RWD	MCAN RAM 看门狗寄存器
MCAN_CCCR	MCAN CC 控制寄存器
MCAN_NBTP	MCAN 标称位时间和预分频寄存器
MCAN_TSCC	MCAN 时间戳计数器配置寄存器
MCAN_TOCC	MCAN 超时计数器配置寄存器
MCAN_TDCCR	MCAN 发送延迟补偿寄存器
MCAN_GFC	MCAN 全局过滤器配置寄存器
MCAN_SIDFC	MCAN 标准 ID 过滤器配置寄存器
MCAN_XIDFC	MCAN 扩展 ID 过滤器配置寄存器
MCAN_XIDAM	MCAN 扩展 ID 与掩码寄存器
MCAN_RXF0C	MCAN 接收 FIFO0 配置寄存器
MCAN_RXBC	MCAN 接收缓冲区配置寄存器
MCAN_RXF1C	MCAN 接收 FIFO1 配置寄存器
MCAN_RXESC	MCAN 接收缓冲区和 FIFO 元素大小配置寄存器
MCAN_TXBC	MCAN 发送缓冲区配置寄存器

寄存器	说明
MCAN_TXESC	MCAN 发送缓冲区元素大小配置寄存器
MCAN_TXEFC	MCAN 发送事件 FIFO 配置寄存器

MCAN_CCCR.CCE 仅在 MCAN_CCCR.INIT 置位时才能被置位或清零，在 MCAN_CCCR.INIT 清零时自动清零。

MCAN_CCCR.CCE 置位后，以下寄存器会被复位：

- MCAN_HPMS——高优先级消息状态
- MCAN_RXF0S——接收 FIFO 0 状态
- MCAN_RXF1S——接收 FIFO 1 状态
- MCAN_TXFQS——发送 FIFO/队列状态
- MCAN_TXBRP——发送缓冲区请求挂起
- MCAN_TXBTO——发送缓冲区发送发生
- MCAN_TXBCF——发送缓冲区取消完成
- MCAN_TXEFS——发送事件 FIFO 状态
- MCAN_TXBAR——发送缓冲区添加请求
- MCAN_TXBCR——发送缓冲区取消请求

仅当 MCAN_CCCR.INIT 和 MCAN_CCCR.CCE 均已置位时，主机才能将 MCAN_CCCR.TEST 和 MCAN_CCCR.MON 置位，主机可随时复位 MCAN_CCCR.TEST 和 MCAN_CCCR.MON。仅当 MCAN_CCCR.INIT 和 MCAN_CCCR.CCE 均已置位时，才能将 MCAN_CCCR.DAR 置位或清零。

30.3.3.2 正常工作

MCAN 完成初始化且 MCAN_CCCR.INIT 清零后，MCAN 会将其自身与 CAN 总线同步并准备好进行通信。

接收消息时，通过接收过滤器接收到的消息（包含消息 ID 和 DLC）会存储到专用的接收缓冲区，或接收 FIFO0，或接收 FIFO1。

发送消息时，先初始化或更新专用发送缓冲区和（或）发送 FIFO（队列）。不支持在接收到远程帧后自动发送。

30.3.3.3 CAN FD 操作

CAN FD 帧传输有两种类型，第一种是不带比特率切换的 CAN FD 帧；第二种是带比特率切换的 CAN FD 帧，其控制场（Control field）、数据场（Data field）和 CRC 场（CRC field）的使用比帧开始和结束时更高的比特率传输。

经典 CAN（Classical CAN）标准帧中的 r0 位（IDE 位之后，标准帧格式中的保留位）、扩展帧中的 r1 位（RTR 位之后，扩展帧格式中的第一个保留位），在 CAN FD 帧格式中解码为 FDF 位。FDF 位为隐性（逻辑 1）表示 CAN FD 帧，为显性（逻辑 0）表示经典 CAN 帧。在 CAN FD 帧中，FDF 位之后的 res

和 BRS (Bit Rate Switch, 比特率切换) 两个位, 决定是否切换 CAN FD 帧中的比特率, 并通过 res 显性和 BRS 隐性指示已切换比特率。res 位隐性的编码保留, 供将来协议扩展使用。如果 MCAN 接收到的帧中 FDF 位和 res 位都是隐性, 则会通过将 MCAN_PSR.PXE 置位的方式发出协议异常事件。如果使能协议异常处理 (MCAN_CCCR.PXHD=0), MCAN 将在下一采样点将工作状态从接收 (MCAN_PSR.ACT=0b10) 变为同步 (MCAN_PSR.ACT=0b00)。如果禁止协议异常处理 (MCAN_CCCR.PXHD=1), MCAN 会将隐性 res 位当作格式错误处理, 并将以错误帧进行响应。

CAN FD 模式通过编程 MCAN_CCCR.FDOE 来使能。如果 MCAN_CCCR.FDOE=1, 则会使能 CAN FD 帧的发送和接收。经典 CAN 帧始终可进行发送和接收。是发送 CAN FD 帧还是经典 CAN 帧, 可通过相应发送缓冲区元素中的 FDF 位配置。如果 MCAN_CCCR.FDOE=0, 接收到的帧会被当作经典 CAN 帧, 从而会在接收到 CAN FD 帧时发送错误帧。如果 CAN FD 模式已禁止, 即使发送缓冲区元素的 FDF 位置 1, 也不会发送 CAN FD 帧。仅当 MCAN_CCCR.INIT 和 MCAN_CCCR.CCE 均置位时, 才能更改 MCAN_CCCR.FDOE 和 MCAN_CCCR.BRSE。

如果 MCAN_CCCR.FDOE=0, 则会忽略 FDF 和 BRS 位的设置, 帧以经典 CAN 格式发送。如果 MCAN_CCCR.FDOE=1 且 MCAN_CCCR.BRSE=0, 则仅会评估发送缓冲区元素的 FDF 位。如果 MCAN_CCCR.FDOE=1 且 MCAN_CCCR.BRSE=1, 则会使能比特率切换发送 CAN FD 帧。所有 FDF 和 BRS 位置位的发送缓冲区元素均会以比特率切换的 CAN FD 格式发送。

建议仅当以下条件满足时, 才在 CAN 操作期间切换模式:

- CAN FD 数据阶段 (Data phase) 的故障率显著高于 CAN FD 仲裁阶段 (Arbitration phase) 的故障率。在这种情况下, 应禁止发送的 CAN FD 比特率切换。
- 系统启动期间, 所有节点在被验证能够以 CAN FD 格式进行通信之前, 都将发送经典 CAN 消息。如果通过验证, 所有节点会切换为 CAN FD 操作。
- CAN 局部网络 (CAN Partial Networking) 中的唤醒消息必须以经典 CAN 格式发送。
- EOL 编程 (End-of-line programming) 时, 并非所有节点都支持 CANFD, 非 CANFD 的节点会保持在静默模式直到编程结束。随后, 所有节点切换回经典 CAN 通信。

在 CANFD 格式中, DLC 代码 0 到 8 的编码 (对应的数据字节长度) 与经典 CAN 相同, 代码 9 到 15 的编码与经典 CAN 不同, 如表 30-3 所示。

表 30-3 MCAN 中的 DLC 编码

DLC 编码		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
数据 字节数	经典 CAN	0	1	2	3	4	5	6	7	8	8	8	8	8	8	8	8
	CAN FD	0	1	2	3	4	5	6	7	8	12	16	20	24	32	48	64

在 CAN FD 帧中, 如果 BRS 位为隐性, 则位时长将在 BRS (比特率切换) 位后, 在帧内切换。在 BRS 位之前, 在 CAN FD 仲裁阶段 (Arbitration phase), 位时长按照寄存器 MCAN_NBTP (Nominal Bit Timing & Prescaler Register, 标称位时长和预分频寄存器) 的定义执行。在接下来的 CAN FD 数据阶段 (Data

phase)，位时长按照寄存器 MCAN_DBTP (Data Bit Timing & Prescaler Register，数据位时长和预分频寄存器) 的定义执行。位时长在 CRC 分隔符处，或检测到错误时从数据阶段 (Data phase) 位时长切换回来，以先发生的事件为准。

CAN FD 数据阶段 (Data phase) 的最大可配置比特率取决于 CAN 通信时钟频率。举例来说，如果 CAN 通信时钟频率为 20MHz，最短可配置的位时间为 4 个时间片 (tq)，则数据阶段的比特率为 5Mbps。

在带比特率切换和不带比特率切换的 CAN FD 两种数据帧格式中，位 ESI (Error Status Indicator，错误状态指示符) 的值，由发送开始时发送节点的错误状态决定。如果发送节点为错误被动状态，则 ESI 以隐性位发送，否则以显性位发送。

30.3.3.4 发送延迟补偿

在 CAN FD 传输的数据阶段 (Data phase)，只有一个节点进行发送，其他所有节点都是接收节点。总线的长度没有影响。当通过引脚 MCAN_TX 发送时，MCAN 通过引脚 MCAN_RX 从其本地 CAN 收发器接收发送的数据。接收数据因发送延迟而延迟，如果该延迟大于 TSEG1 (采样点之前的时间段)，则会检测到位错误。为了使数据阶段 (Data phase) 位时间可以比发送延迟更短，引入了延迟补偿。如果没有发送延迟补偿，CAN FD 帧数据阶段 (Data phase) 的比特率将受到发送延迟的限制。

MCAN 的协议单元实现了延迟补偿机制来补偿发送延迟，从而实现在 CAN FD 数据阶段 (Data phase) 以更高的比特率进行传输，而不受特定 CAN 收发器延迟的影响。

为了检查发送节点的数据阶段 (Data phase) 是否存在位错误，会将延迟的发送数据与第二采样点 SSP 处的接收数据进行比较。如果检测到位错误，发送节点将在下一个常规采样点对该位错误作出反应。在仲裁阶段，延迟补偿始终被禁用。

发送延迟补偿支持数据位时间短于发送延迟的配置，ISO 11898-1:2015 对此进行了详细描述，它通过置位 MCAN_DBTP.TDC 使能。

在第二采样点 SSP 将接收到的位与发送的位进行比较。SSP 的位置定义为，从 MCAN 发送输出引脚 MCAN_TX 通过收发器到接收输入引脚 MCAN_RX 测得的延迟，与由 MCAN_TDCR.TDCO 配置的发送延迟补偿偏移之和。发送延迟补偿偏移，用于调整 SSP 在已接收到的位中的位置 (例如数据阶段 Data phase 的位时间的一半)。第二采样点的位置向下取整到下一个整数 mtq (Minimum time quantum，即最小时间片，为一个 MCAN 通信时钟周期)。

MCAN_PSR.TDCV 显示实际的发送延迟补偿值。当 MCAN_CCCR.INIT 置位时，MCAN_PSR.TDCV 被清零；当 MCAN_DBTP.TDC 置位时，MCAN_PSR.TDCV 在每次发送 FD 帧时都会更新。

对于在 MCAN 中实现的发送延迟补偿，必须考虑以下边界条件：

- 测得的 MCAN_TX 到 MCAN_RX 的延迟与配置的发送延迟补偿偏移 MCAN_TDCR.TDCO 之和，必须小于数据阶段 (Data phase) 的 6 个位时间。

- 测得的 MCAN_TX 到 MCAN_RX 的延迟与配置的发送延迟补偿偏移 MCAN_TDCR.TDCO 之和，必须小于或等于 127mtq。如果此和大于 127mtq，则发送延迟补偿使用最大值 127mtq。
- 数据阶段（Data phase）在 CRC 分隔符的采样点结束，之后将不在 SSP 检查接收位。

如果通过编程 MCAN_DBTP.TDC=1 使能发送延迟补偿，则会在每个 CAN FD 帧的 FDF 位到 res 位的下降沿处开始测量。当在发送节点的 MCAN_RX 引脚上检测到该边沿时，停止测量。该测量的分辨率为一个 mtq。

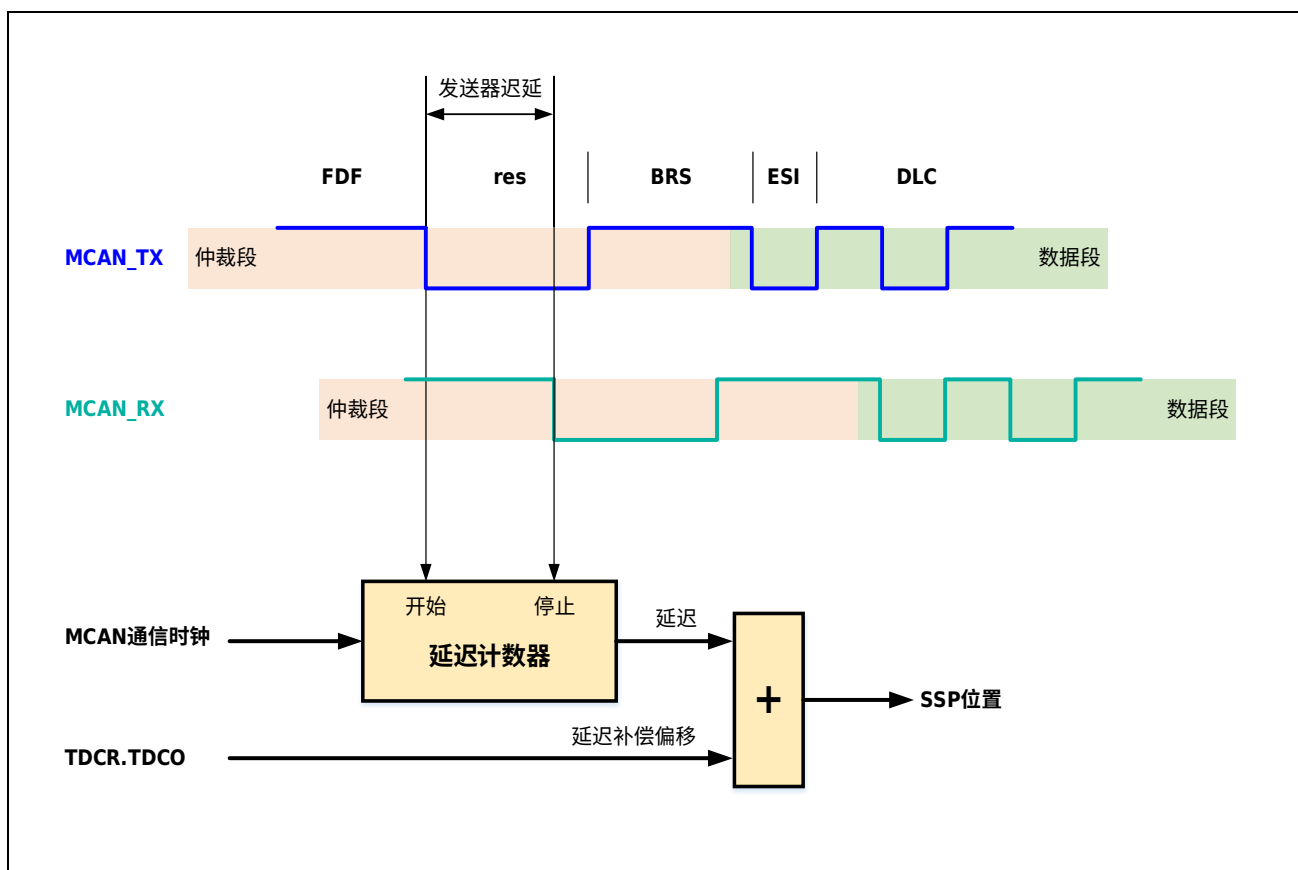


图 30-2 收发器延迟测量

为了避免接收到的 FDF 位中的显性毛刺，在接收到的 res 位的下降沿之前结束延迟补偿测量，从而导致 SSP 位置提前，可通过编程 MCAN_TDCR.TDCF 来使能发送延迟补偿滤波窗口，从而定义 SSP 位置的最小值。MCAN_RX 引脚上的显性边缘会导致 SSP 位置更加提前，但对于发送延迟测量而言这将被忽略。当 SSP 位置至少为 MCAN_TDCR.TDCF 且 MCAN_RX 引脚为低电平时，测量停止。

注：

MCAN 使用的发送延迟为实时测量值，不支持配置固定的发送延迟。

30.3.3.5 受限工作模式

在受限工作模式下，节点能够接收数据帧和远程帧，并能对有效帧进行确认，但不会发送数据帧、远程帧、主动错误帧或过载帧。在出现错误或过载情况时，不会发送显性位，而是等待出现总线空闲条件，以重新同步到 CAN 通信。错误记录 (MCAN_TDCR.CEL) 动作时，错误计数器 (MCAN_TDCR.REC、MCAN_TDCR.TEC) 被冻结。主机可通过置位 MCAN_CCCR.ASM 将 MCAN 设置为受限工作模式，仅当 MCAN_CCCR.CCE 和 MCAN_CCCR.INIT 均设为 1 时，主机才能将 MCAN_CCCR.ASM 置位。MCAN_CCCR.ASM 可随时通过主机清零。

当发送处理单元无法及时从消息 RAM 中读取数据时，会自动进入受限工作模式。要退出受限工作模式，主机 CPU 必须清零 MCAN_CCCR.ASM。

受限工作模式可用于适应不同 CAN 比特率的应用。在这种情况下，应用程序会测试不同的比特率，并在收到有效帧后退出受限操作模式。

注：

受限工作模式不得与回环模式（内部或外部）结合使用。

30.3.3.6 总线监听模式

配置 MCAN_CCCR.MON 为 1, 将 MCAN 设置为总线监听模式。在总线监听模式下 (见 ISO 11898-1:2015, 10.14 Bus monitoring), MCAN 能够接收有效数据帧和有效远程帧，但不能启动发送。在该模式下，MCAN 仅在 CAN 总线上发送隐性位，如果 MCAN 需要发送一个显性位 (ACK 位、过载标志、主动错误标志)，该位将在内部被改道发送，以便 MCAN 可以监听该显性位，但 CAN 总线可以保持隐性状态。在总线监听模式下，寄存器 MCAN_TXBRP 保持复位状态。

总线监听模式可用于分析 CAN 总线上的流量，同时又不会因发送显性位而对总线造成影响。图 30-3 显示了总线监听模式下 MCAN_TX 和 MCAN_RX 信号与 MCAN 的连接。

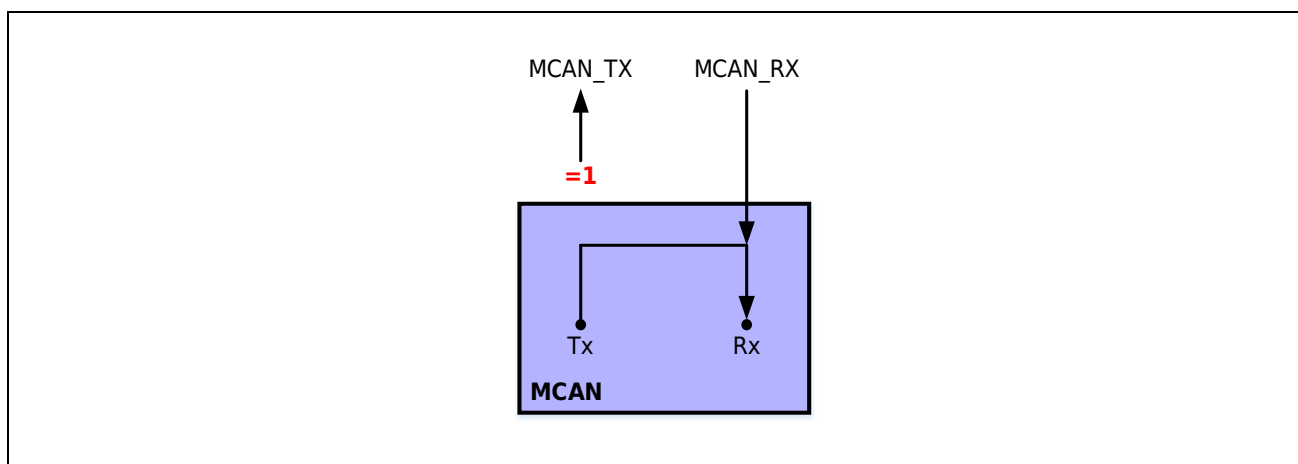


图 30-3 总线监听模式下的引脚控制

30.3.3.7 禁止自动重发 (DAR) 模式

根据 CAN 规范 (见 ISO 11898-1:2015, 8.3.4 Recovery Management), MCAN 可自动重发仲裁失败或在发送期间被错误干扰的帧。默认使能自动重发。

在 DAR 模式下, 所有发送在 CAN 总线上启动后都会自动取消。发送缓冲区发送请求挂起位 MCAN_TXBRP.TRPx 在成功发送后复位, 如果取消时发送尚未开始、发送已因仲裁失败而中止、或帧发送期间发生错误, 该位也会复位。

■ 成功发送:

- 对应的发送缓冲区发送已发生位 MCAN_TXBTO.TOx 置位
- 对应的发送缓冲区取消已完成位 MCAN_TXBCF.CFx 不置位

■ 即使已取消但也成功发送:

- 对应的发送缓冲区发送已发生位 MCAN_TXBTO.TOx 置位
- 对应的发送缓冲区取消已完成位 MCAN_TXBCF.CFx 置位

■ 仲裁失败或帧发送受到干扰:

- 对应的发送缓冲区发送已发生位 MCAN_TXBTO.TOx 不置位
- 对应的发送缓冲区取消已完成位 MCAN_TXBCF.CFx 置位

如果帧发送成功, 并且使能了发送事件的存储, 则会写入发送事件 FIFO 元素, 其中事件类型 ET=0b10 (即使已取消也会进行发送)。

30.3.3.8 掉电 (休眠模式)

MCAN 可通过 CC 控制寄存器 MCAN_CCCR.CSR 设置为掉电模式。只要时钟停止请求有效, MCAN_CCCR.CSR 位就读为 1。

当所有挂起的发送请求都完成后, MCAN 会一直等待直到检测到总线空闲状态。检测到总线空闲状态后, MCAN 将 MCAN_CCCR.INIT 设置为 1 以防止任何进一步的 CAN 传输。然后, MCAN 通过将 MCAN_CCCR.CSA 置 1 确认其已准备好进入掉电状态。在模块时钟 (控制逻辑时钟和通信时钟) 关闭之前, 可继续访问寄存器, 但 MCAN_CCCR.INIT 保持为 1。

注:

在 CAN 总线受到严重干扰的情况下, 可能永远不会达到空闲状态, 因此 MCAN 不会置位 MCAN_CCCR.INIT。这种情况可通过轮询 MCAN_PSR.ACT 来检测。如果 MCAN 没有进入空闲状态, 软件可以写 CCCR.INIT=1, 这将立即停止 MCAN 的 CAN 通信, 无论是否正在进行发送或接收。

要退出掉电模式, 应用程序必须在复位 CC 控制寄存器标志 MCAN_CCCR.CSR 之前开启模块时钟。MCAN 将通过复位 MCAN_CCCR.CSA 确认时钟已开启。之后, 应用程序可通过复位 MCAN_CCCR.INIT 重新开始 CAN 通信。

30.3.3.9 测试模式

要使能对 MCAN 测试寄存器 MCAN_TEST（请参见章节【MCAN 测试寄存器（MCAN_TEST）】）的写访问，必须将 MCAN_CCCR.TEST 置 1，从而能够配置测试模式和测试功能。

通过编程 MCAN_TEST.TX，发送引脚 MCAN_TX 可实现四种输出功能。除了串行数据输出这一默认功能之外，还可驱动 CAN 采样点信号来监测 MCAN 的位时长，并可输出恒定的显性或隐性电平。从 MCAN_TEST.RX 可读取引脚 MCAN_RX 的实际值。这两种功能（MCAN_TEST.TX 和 MCAN_TEST.RX）均可用于检查 CAN 总线的物理层。

由于 CAN 通信时钟与控制逻辑时钟之间的同步机制，从配置 MCAN_TEST.TX 到引脚 MCAN_TX 实现指定功能之间，可能有几个控制逻辑时钟周期的延迟。这一点还适用于通过 MCAN_TEST.RX 读取引脚 MCAN_RX 的情况。

注：

测试模式仅限用于生产测试和自检。对 MCAN_TX 引脚进行软件控制会干扰所有 CAN 协议功能。不建议实际应用时使用测试模式。

30.3.3.10 外部回环模式

配置 MCAN_TEST.LBCK 为 1，将 MCAN 设置为外部回环模式。在回环模式下，MCAN 将其自身发送的消息作为接收的消息来处理，并将消息存储（如果这些消息通过了接收过滤）到接收缓冲区或接收 FIFO 中。图 30-4 显示了外部回环模式下 MCAN_TX 和 MCAN_RX 引脚信号与 MCAN 的连接。

该模式用于硬件自检。为了不受外部影响，MCAN 在回环模式下忽略 ACK 错误（在数据帧或远程帧的 ACK 域采样到隐性位）。在此模式下，MCAN_TX 输出的信号在内部直接反馈到接收处理单元，MCAN_RX 输入引脚的实际值被忽略。从引脚 MCAN_TX 可以监听到发送的消息。

30.3.3.11 内部回环模式

配置 MCAN_TEST.LBCK 和 MCAN_CCCR.MON 为 1，将 MCAN 设置为内部回环模式。该模式可用于“热自检（Hot Selftest）”，也就是说，MCAN 可以进行检测，同时又不会影响接入的正在运行的 CAN 系统。在此模式下，引脚 MCAN_RX 与 MCAN 断开连接，引脚 MCAN_TX 保持隐性。图 30-4 显示了内部回环模式下 MCAN_TX 和 MCAN_RX 引脚信号与 MCAN 的连接。

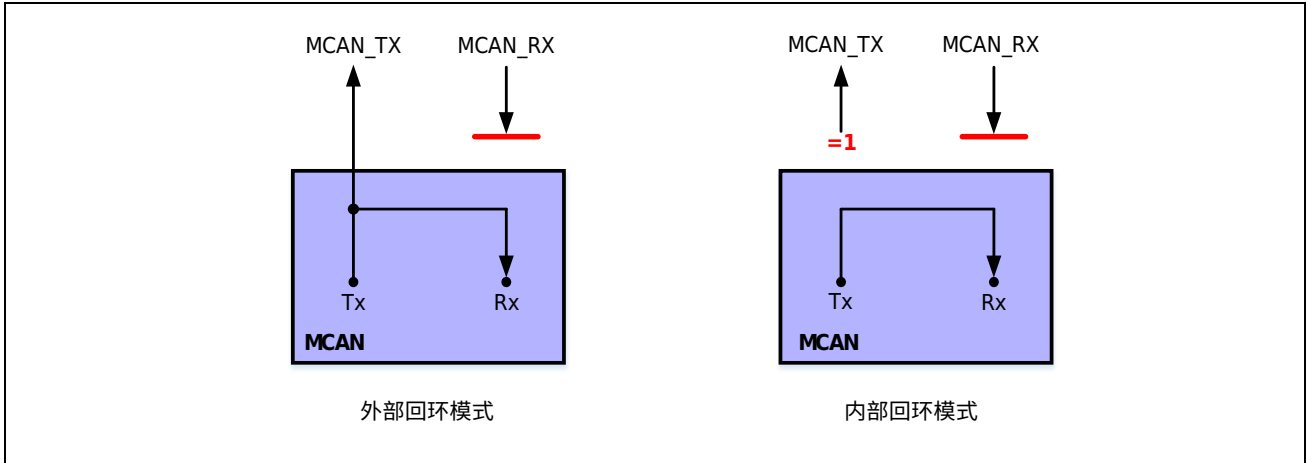


图 30-4 回环模式的引脚控制

30.3.4 时间戳生成

MCAN 提供了一个 16 位循环计数器来生成内部时间戳。计数时间单位可通过预分频器 MCAN_TSCC.TCP 配置为 CAN 位时间 1~16 倍。计数器值可通过 MCAN_TSCV.TCV 读取。对寄存器 MCAN_TSCV 进行写访问会将计数器复位为 0。时间戳计数器循环时，中断标志 MCAN_IR.TSW 会置位。

在接收到或发送 SOF (Start Of Frame, 帧起始) 时，计数值被捕获并存储到接收缓冲区或接收 FIFO (RXTS[15:0]) 或发送事件 FIFO (TXTS[15:0]) 元素的时间戳部分。

30.3.5 超时计数器

MCAN 提供了一个 16 位超时计数器来指示接收 FIFO0、接收 FIFO1 和发送事件 FIFO 的超时条件。它作为递减计数器工作，并且与时间戳计数器一样，使用 MCAN_TSCC.TCP 控制的预分频器。超时计数器通过寄存器 MCAN_TOCC 配置。实际计数器值可从 MCAN_TOCV.TOC 读取。仅当 MCAN_CCCR.INIT=0 时，才会启动超时计数器。当 MCAN_CCCR.INIT=1 时，超时计数器停止计数，例如当 CAN 进入 Bus_Off 状态时。

工作模式通过 MCAN_TOCC.TOS 选择。在连续模式下工作时，计数器在 MCAN_CCCR.INIT 复位时启动。

对 MCAN_TOCV 进行写操作，会将计数器预设为 MCAN_TOCC.TOP 配置的值，并继续递减计数。

当超时计数器由其中一个 FIFO 控制时，空 FIFO 会将计数器预设为 MCAN_TOCC.TOP 配置的值。存储了第一个 FIFO 元素后，开始递减计数。对 MCAN_TOCV 执行写入操作不起作用。

计数器达到 0 时，中断标志 MCAN_IR.TOO 置位。在连续模式下，计数器在达到 MCAN_TOCC.TOP 后立即重启。

注：

超时计数器的时钟信号来自 CAN 内核采样点信号。因此，由于 CAN 内核同步（重同步）机制的原因，超时计数器递减的时间点可能有所不同。如果在 CAN FD 中使用了比特率切换功能，则超时计数器在仲裁阶段 (Arbitration phase) 和数据阶段 (Data phase) 中使用不同的时钟计数。

30.3.6 接收处理

接收处理单元控制着接收过滤、已接收消息到接收缓冲区或其中一个接收 FIFO（共两个）的传输以及接收 FIFO 放入和获取索引。

30.3.6.1 接收过滤

MCAN 能够配置两组接收过滤器，一组供标准 ID 使用，另一组供扩展 ID 使用。这两组过滤器可分配给接收缓冲区、接收 FIFO0 或接收 FIFO1。对于接收过滤，每个过滤器列表从第 0 个元素开始执行比较，直到第一个匹配的元素。接收过滤在第一个匹配的元素处停止。不会为此消息评估剩下的过滤器元素。

■ 主要特性有：

- 每个过滤器元素可配置为
 - * ID 范围过滤器（起止范围）
 - * 一个或两个专用 ID 的过滤器
 - * ID 位屏蔽的经典过滤器
- 每个过滤器元素均可配置为实现接收过滤或拒绝过滤
- 每个过滤器元素均可单独使能/禁止
- 按顺序比较过滤器元素，到第一个匹配的过滤器元素后停止执行

■ 相关配置寄存器位：

- 全局过滤器配置 (MCAN_GFC)
- 标准 ID 过滤器配置 (MCAN_SIDFC)
- 扩展 ID 过滤器配置 (MCAN_XIDFC)
- 扩展 ID 与运算掩码 (MCAN_XIDAM)

■ 根据过滤器元素的配置 (SFEC/EFEC)，发生匹配时会触发下列操作之一：

- 将接收到的帧存储到接收 FIFO0 或 FIFO1 中
- 将接收到的帧存储到接收缓冲区中
- 拒绝接收到的帧
- 将高优先级消息中断标志 MCAN_IR.HPM 置 1
- 将高优先级消息中断标志 MCAN_IR.HPM 置 1，并将接收到的帧存储到接收 FIFO0 或 FIFO1 中

接收到完整的 ID 后，开始接收过滤。接收过滤完成后，如果找到匹配的接收缓冲区或接收 FIFO，消息处理单元便会开始将接收到的消息数据以 32 位的形式写入匹配的接收缓冲区或接收 FIFO 中。如果 CAN 协议控制器检测到错误条件（例如 CRC 错误），则会丢弃此消息，丢弃后的影响如下：

■ 接收缓冲区

匹配接收缓冲区的新数据标志不会置 1，但接收缓冲区（部分）会被接收到的数据覆盖。有关错误类型，请参见 MCAN_PSR.LEC 和 MCAN_PSR.DLEC。

■ 接收 FIFO

匹配接收 FIFO 的放入索引不会更新，但相关接收 FIFO 元素（部分）会被接收到的数据覆盖。有关错误类型，请参见 MCAN_PSR.LEC 和 MCAN_PSR.DLEC。如果匹配的接收 FIFO 在覆盖模式下工作，必须考虑接收 FIFO 覆盖模式中介绍的边界条件。

注：

当接收的消息写入两个接收 FIFO 之一或写入接收缓冲区中时，未匹配的已接收 ID 会独立于所用过滤器之外进行存储。接收过滤的过程很大程度上取决于所配置的过滤器元素的顺序。

30.3.6.2 ID 范围过滤器

该过滤器会将所有接收到的帧与 SF1ID/SF2ID 和 EF1ID/EF2ID 定义的范围内的消息 ID 进行匹配检查。范围过滤与扩展帧一起使用时，有两种可能的情况：

- EFT=0b00：应用范围过滤器之前，会将已接收帧的消息 ID 与扩展 ID 与运算掩码 (MCAN_XIDAM) 进行与运算
- EFT=0b11：不会为范围过滤使用扩展 ID 与运算掩码 (MCAN_XIDAM)

30.3.6.3 专用 ID 过滤器

可将过滤器元素配置为过滤一个或两个特定的消息 ID。要过滤一个特定的消息 ID，过滤器元素必须配置为 SF1ID=SF2ID 和 EF1ID=EF2ID。

30.3.6.4 ID 位屏蔽过滤器

ID 位屏蔽过滤用于通过屏蔽已接收消息 ID 的某些位来过滤消息 ID 组。进行 ID 位屏蔽过滤时，SF1ID/EF1ID 用作消息 ID 过滤器，SF2ID/EF2ID 用作过滤器 ID 掩码 (ID Mask)。

ID 掩码中为 0 的位将屏蔽掉已配置 ID 过滤器的对应位，即，已接收消息 ID 在该位位置的值不与接收过滤器对应位的值进行比较，只有对应掩码位为 1 的已接收消息 ID 的位才进行比较。

如果所有掩码位均为 1，则仅当接收到的消息 ID 和消息 ID 过滤器完全相同时，才会出现匹配。如果所有掩码位均为 0，则所有消息 ID 都匹配。

30.3.6.5 标准消息 ID 过滤

图 30-5 介绍了标准消息 ID（11 位 ID）的过滤流程。章节【标准消息 ID 过滤器元素】介绍了标准消息 ID 过滤器元素。

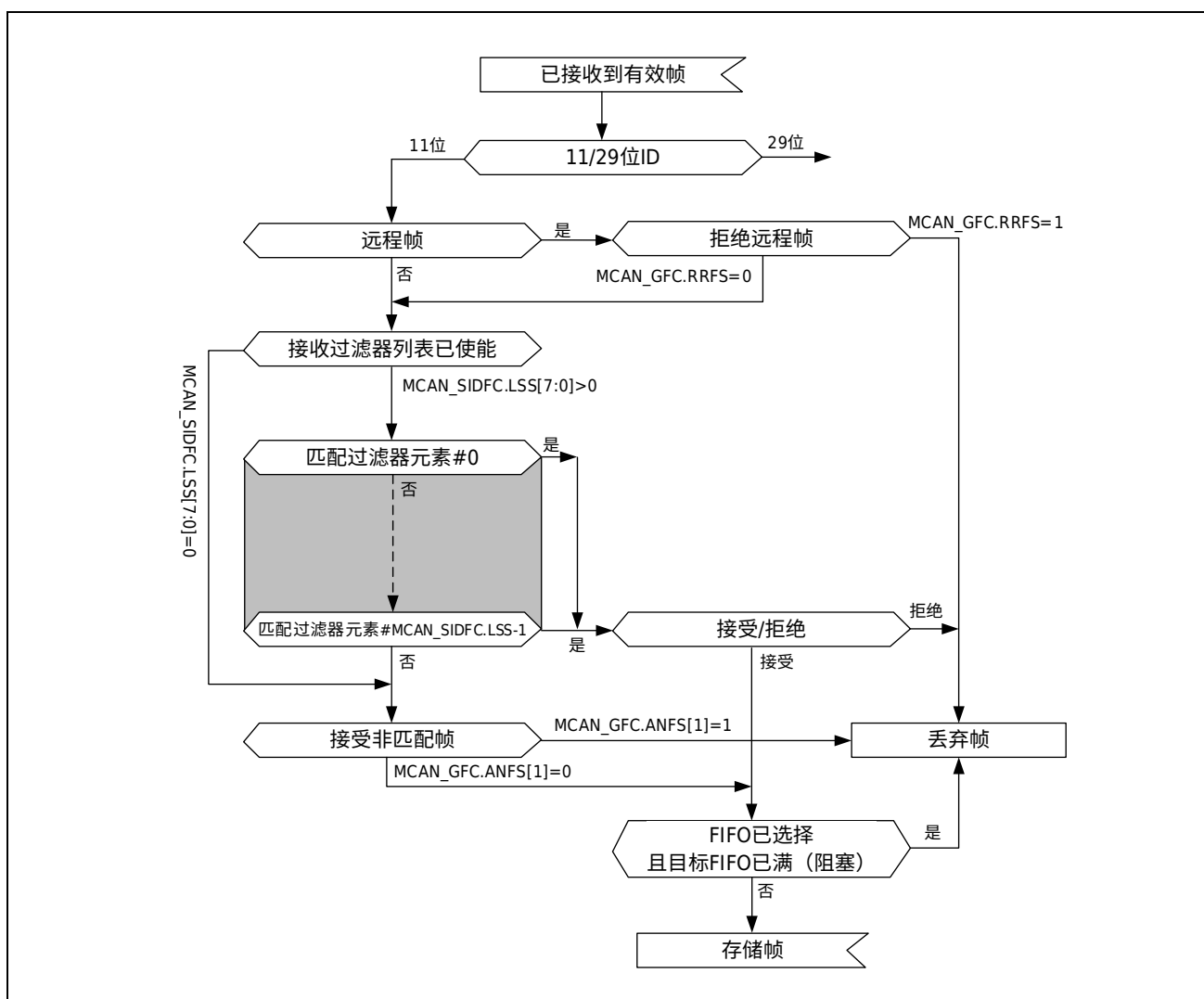


图 30-5 标准消息 ID 过滤器路径

在全局过滤器配置寄存器（MCAN_GFC）和标准 ID 过滤器配置寄存器（MCAN_SIDFC）控制下，接收消息的 ID、远程传输请求位（RTR）和 ID 扩展位（IDE）将与配置的过滤器元素列表进行比较。

30.3.6.6 扩展消息 ID 过滤

图 30-6 介绍了扩展消息 ID (29 位 ID) 的过滤流程。章节【扩展消息 ID 过滤器元素】介绍了扩展消息 ID 过滤器元素。

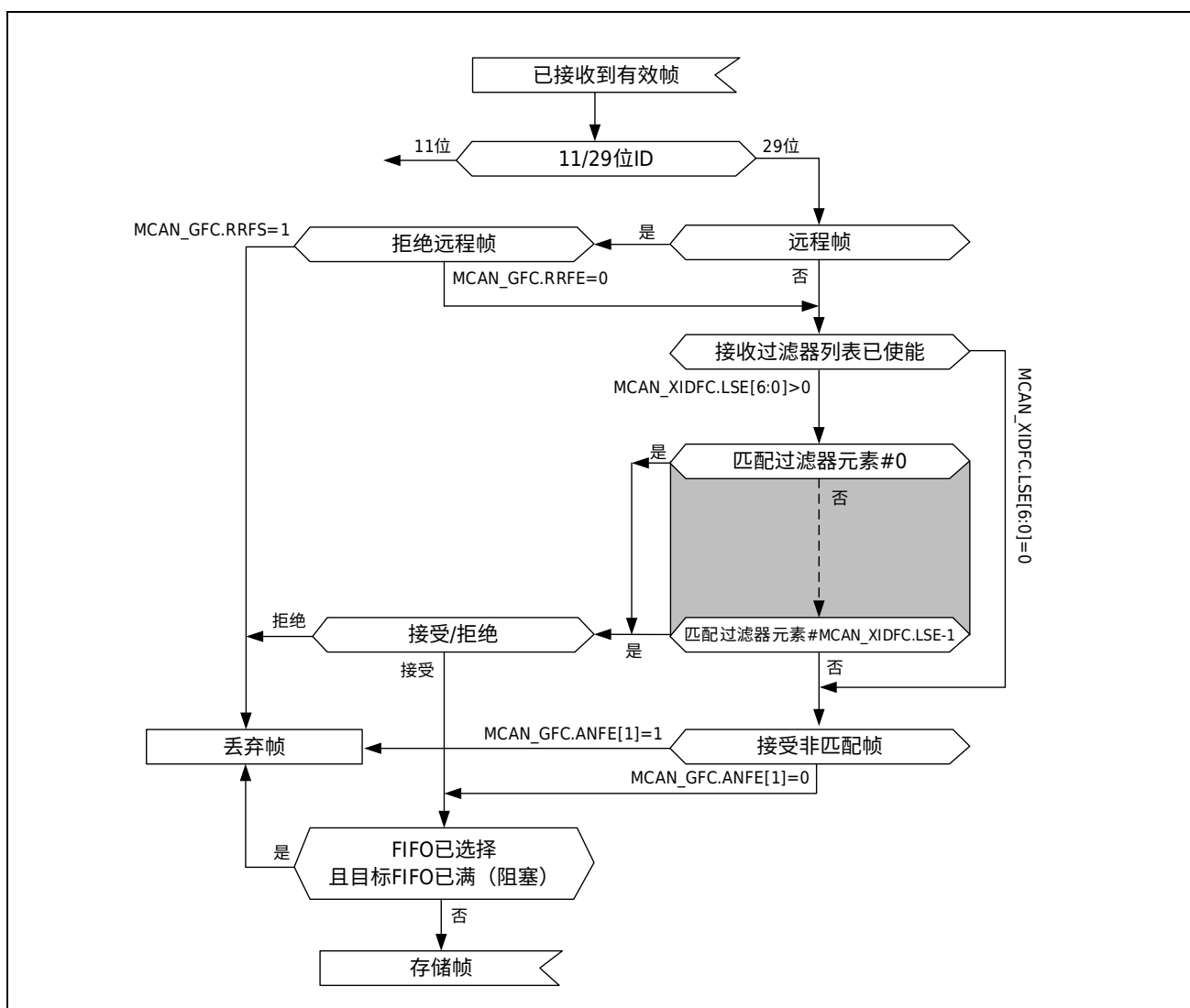


图 30-6 扩展消息 ID 过滤器路径

在全局过滤器配置寄存器 (MCAN_GFC) 和扩展 ID 过滤器配置寄存器 (MCAN_XIDFC) 控制下, 接收消息的 ID、远程传输请求位 (RTR) 和 ID 扩展位 (IDE) 将与配置的过滤器元素列表进行比较。扩展 ID 与掩码 (MCAN_XIDAM) 会在执行过滤器列表之前与接收到的 ID 进行与运算。

30.3.6.7 接收 FIFO

接收 FIFO0 和接收 FIFO1 分别最多可保存 64 个元素（实际元素个数与消息 RAM 容量有关）。两个接收 FIFO 的配置通过寄存器 MCAN_RXF0C 和 MCAN_RXF1C 完成。

通过接收过滤的消息会传输到匹配过滤器元素配置的接收 FIFO 中。有关可用于接收 FIFO0 和接收 FIFO1 的过滤器机制的说明，请参见章节【接收过滤】。章节【接收缓冲区和 FIFO 元素】介绍了接收缓冲区和 FIFO 元素。

为了避免接收 FIFO 溢出，可使用接收 FIFO 水位线（Watermark）。当接收 FIFO 填充级别达到 MCAN_RXFnC.FnWM 配置的接收 FIFO 水位线时，中断标志 MCAN_IR.RFnW 置位。当接收 FIFO 的放入索引和获取索引相遇时，接收 FIFO 满的条件产生，标志位 MCAN_RXFnS.FnF 置位，中断标志位 MCAN_IR.RFnF 置位。

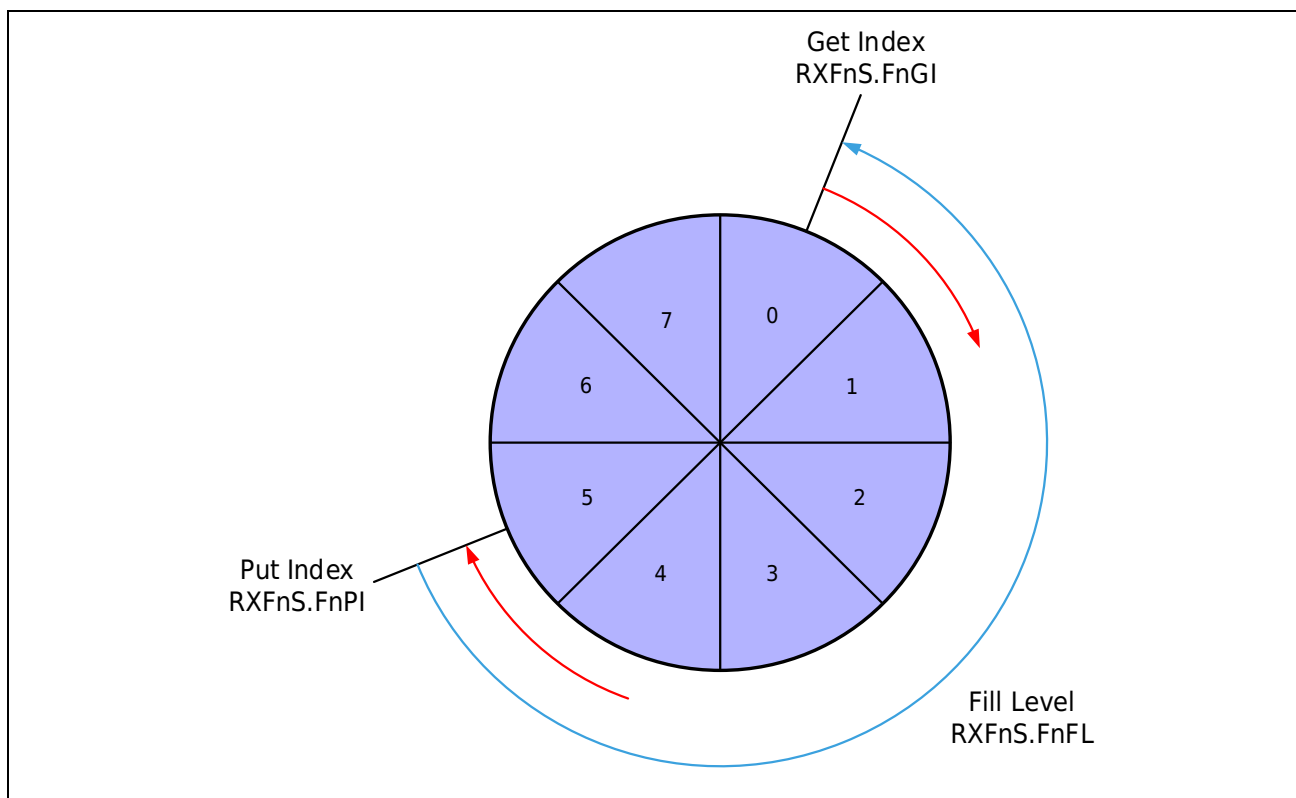


图 30-7 接收 FIFO 状态

当从接收 FIFO 中读取消息时，接收 FIFO 获取索引 MCAN_RXFnS.FnGI 乘以 FIFO 元素大小（字），再乘以 4 后，必须再加上起始地址 MCAN_RXFnC.FnSA。

表 30-4 接收缓冲器和 FIFO 元素大小

RXESC.RBDS[2:0] RXESC.FnDS[2:0]	数据场 (字节)	元素大小 (字)
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

30.3.6.8 接收 FIFO 阻塞模式

接收 FIFO 阻塞模式是通过 MCAN_RXFnC.FnOM=0 配置的，该模式是接收 FIFO 的默认工作模式。

达到接收 FIFO 已满条件时 (MCAN_RXFnS.FnPI=MCAN_RXFnS.FnGI)，标志位 RXFnS.FnF=1，中断标志 MCAN_IR.RFnF=1。此时，MCAN 不会继续向相应的接收 FIFO 写入消息，除非从相应的接收 FIFO 读取消息且其获取索引已递增。

如果在相应的接收 FIFO 已满时收到消息，此消息会被丢弃，并会通过 MCAN_RXFnS.RFnL=1 指示消息丢失。此外，中断标志 MCAN_IR.RFnL 也会置位。

30.3.6.9 接收 FIFO 覆盖模式

接收 FIFO 覆盖模式是通过 MCAN_RXFnC.FnOM=1 配置的。

达到接收 FIFO 已满条件时 (MCAN_RXFnS.FnPI=MCAN_RXFnS.FnGI)，标志位 RXFnS.FnF=1。此时，FIFO 接收的下一条消息将覆盖最早收到的消息，放入索引和获取索引都会加 1。

如果接收 FIFO 在覆盖模式下工作，并且接收 FIFO 已满，则应至少从获取索引加 1 处开始读取接收 FIFO 元素。这是因为，当 CPU 正在从消息 RAM 读取消息时，恰好有接收到的消息正在被写入消息 RAM，这可能导致 CPU 读到的数据与实际不一致，从接收 FIFO 读取数据时向获取索引中添加偏移量可避免此问题。偏移量取决于 CPU 访问接收 FIFO 的速度。图 30-8 显示了读取接收 FIFO 时获取索引的偏移量为 2。在这种情况下，存储在元素 1 和 2 中的两条消息会丢失。

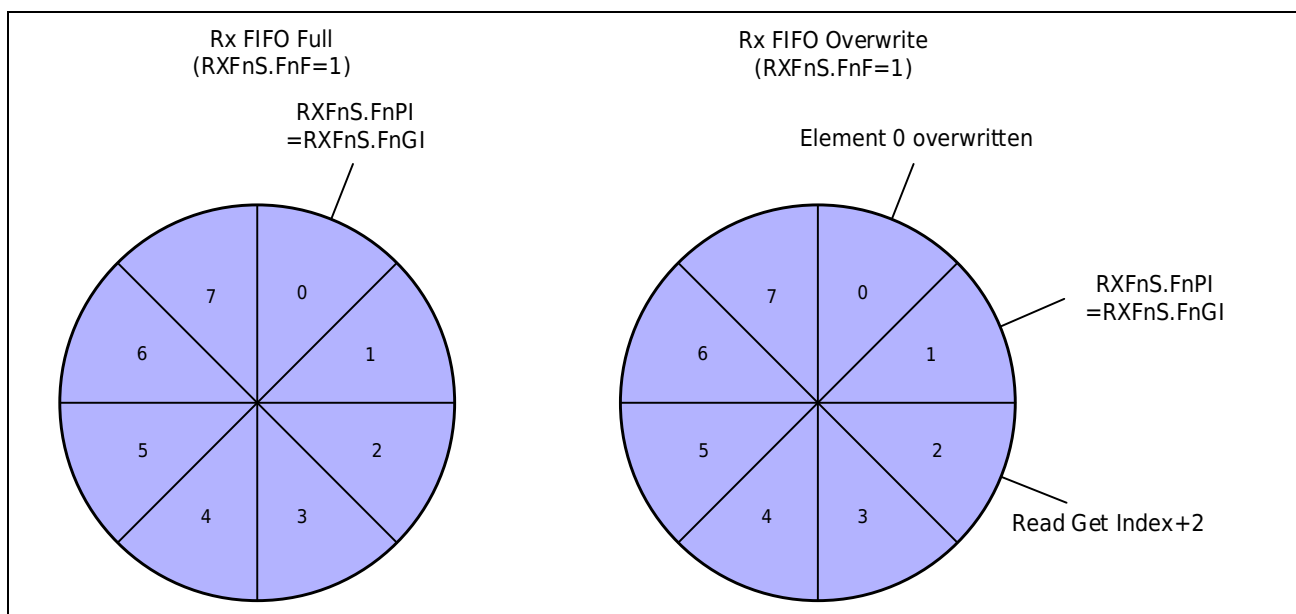


图 30-8 接收 FIFO 溢出处理

从接收 FIFO 读取数据后，必须将读取的最后一个元素的编号写入接收 FIFO 确认索引寄存器 MCAN_RXFnA.FnA 中，使增加后的获取索引指向下一个元素。如果放入索引未增加到该接收 FIFO 元素编号，接收 FIFO 已满标志会复位 (MCAN_RXFnS.FnF=0)。

30.3.6.10 专用接收缓冲区

MCAN 支持多达 64 个专用接收缓冲区。专用接收缓冲区的起始地址是通过 RXBC.RBSA 配置的。

要使用专用接收缓冲区，对于标准 ID 消息，必须配置标准消息 ID 过滤器元素的 SFEC 位为 0b1111、SFID2 位为 0b00 (请参见章节【标准消息 ID 过滤器元素】)；对于扩展 ID 消息，必须配置扩展消息 ID 过滤器元素的 EFEC 位为 0b1111、EFID2 位为 0b00 (请参见章节【扩展消息 ID 过滤器元素】)。表 30-5 为专用接收缓冲区的过滤器配置示例。

接收到的消息被过滤器元素接受后，会存储在过滤器元素引用的消息 RAM 中的接收缓冲区中。存储格式与接收 FIFO 元素的格式相同。此外，中断寄存器中的标志 MCAN_IR.DRX (专用接收缓冲区中存储的消息) 也会置 1。

表 30-5 接收缓冲区过滤器配置示例

过滤器元素	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID消息 1	00	00 0000
1	ID消息 2	00	00 0001
2	ID消息 3	00	00 0010

匹配的接收到的消息最后一个字写入消息 RAM 后，寄存器 MCAN_NDAT1/MCAN_NDAT2 中对应的新数据标志会置位。只要新数据标志保持置位的状态，相应的接收缓冲区就会被锁定，以免更新为新的匹配消息。新数据标志必须由主机向相应的位写 1 进行复位。

接收缓冲区新数据标志置位时，其对应的消息 ID 过滤器元素将不进行接收过滤，其它的过滤器元素会继续进行接收过滤。其它的消息 ID 过滤器元素可能导致接收到的消息存储到另一接收缓冲区或接收 FIFO 中，或者被拒绝，具体取决于过滤器配置。

30.3.6.11 接收缓冲区处理

- 复位中断标志 MCAN_IR.DRX
- 读取新数据寄存器
- 从消息 RAM 中读取消息
- 复位已处理消息的新数据标志

30.3.6.12 过滤调试消息

要过滤调试消息，需为三个调试消息分别配置一个标准或扩展消息 ID 过滤器元素。要使能过滤器元素过滤调试消息，必须将 SFEC/EFEC 编程为 0b111。在这种情况下，SFID1/SFID2 和 EFID1/EFID2 的含义不同 (请参见章节【标准消息 ID 过滤器元素】和章节【扩展消息 ID 过滤器元素】)。SFID2/EFID2[10:9] 控制着调试消息处理状态机，而 SFID2/EFID2[5:0] 控制着收到的调试消息的存储位置。表 30-6 为调试消息过滤器配置示例。

存储调试消息时，相应的新数据标志和 MCAN_IR.DRX 均不会置位，可通过 MCAN_RXF1S.DMS 监视调试消息的接收情况。

表 30-6 调试消息过滤器配置示例

过滤器元素	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID调试消息 A	01	11 1101
1	ID调试消息 B	10	11 1110
2	ID调试消息 C	11	11 1111

30.3.7 发送处理

发送处理单元处理专用发送缓冲区、发送 FIFO 和发送队列的发送请求。它控制着发送消息到 CAN 内核的传输、放入和获取索引以及发送事件 FIFO。最多可为消息发送设置 32 个发送缓冲区，每个发送缓冲区可以独立配置发送模式 (经典 CAN 模式或 CAN FD 模式)。发送缓冲区的描述见章节【发送缓冲区元素】。表 30-7 描述了发送帧的可能配置。

表 30-7 发送帧的可能配置

MCAN_CCCR		发送缓冲区元素		发送帧
BRSE	FDOE	FDI	BRS	
忽略	0	忽略	忽略	经典CAN
0	1	0	忽略	经典CAN
0	1	1	忽略	不切换比特率的FD

MCAN_CCCR		发送缓冲区元素		发送帧
BRSE	FDOE	FDF	BRS	
1	1	0	忽略	经典CAN
1	1	1	0	不切换比特率的FD
1	1	1	1	切换比特率的FD

注：

AUTOSAR 至少需要三个发送队列缓冲区，并支持发送取消。

当发送缓冲区请求挂起寄存器 MCAN_TXBRP 更新时，发送处理单元会启动发送扫描来检查优先级最高的挂起的发送请求（消息 ID 最小的发送缓冲区）。MCAN_TXBRP 更新的条件是，一个发送缓冲区发送完成，或主机通过写 MCAN_TXBAR 添加发送请求，或主机通过写 MCAN_TXBCR 取消已挂起的发送。

30.3.7.1 发送暂停

发送暂停功能用于 CAN 消息 ID（永久性地）被指定为特定值、并且无法轻易更改的 CAN 系统中。这些消息 ID 的 CAN 仲裁优先级可能高于其他已定义的消息，而在特定的应用中，其相对仲裁优先级应反转。这样可能会导致如下情况：一个 ECU 突发多条较高仲裁优先级的 CAN 消息造成另一个仲裁优先级较低的 ECU CAN 消息被延迟。

举例来说，如果 CAN ECU-1 启用了此功能，并且其应用软件请求发送四条消息，成功发送第一条消息后，将等待两个总线空闲的 CAN 位时间，然后才允许开始发送下一条请求的消息。如果其他 ECU 有挂起的消息，则会在空闲时间开始发送这些消息，不需要与 ECU-1 的下一条消息进行仲裁。接收到消息后，ECU-1 可在接收到的消息释放 CAN 总线后立即开始进行下一次发送。

此功能是通过 MCAN_CCCR.TXP 控制的。如果此位置位，则每次成功发送消息后，MCAN 都将暂停两个 CAN 位时间，然后再开始进行下一次发送。这样，即使网络中其他节点的 ID 优先级较低，也能发送消息。默认禁止此功能（MCAN_CCCR.TXP=0）。此功能可使来自单个节点突发的多条消息变得稀疏一些，避免发送较低优先级消息的 ECU 出现应用程序错误地认为自己请求过多传输。

30.3.7.2 专用发送缓冲区

专用发送缓冲区可完全在主机 CPU 的控制下发送消息。每个专用发送缓冲区都配置了特定的消息 ID。如果多个发送缓冲区配置为相同消息 ID，则会先发送编号最小的发送缓冲区中的消息，这些发送缓冲区应按升序请求发送，即先请求发送缓冲区编号最小的，或者，通过单次写入 MCAN_TXBAR 同时请求发送。

如果缓冲区中的消息有更新，通过置位 MCAN_TXBAR.ARn 请求发送。。已请求发送的消息，会根据消息 ID 先在内部与发送 FIFO 或发送队列中的消息进行仲裁，然后在外部与 CAN 总线上的消息进行仲裁，最后根据仲裁结果发送出去。

消息 RAM 中分配的专用发送缓冲区元素大小为若干个 32 位字（4 字节）。因此，消息 RAM 中专用发送缓冲区的起始地址，是通过将发送缓冲区索引（0~31）乘以元素大小（字），再乘以 4（转换为字节），然后与 MCAN_TXBC.TBSA 相加计算得到。

表 30-8 发送缓冲区、FIFO（队列）元素大小

TXESC.TBDS2[2:0]	数据场（字节）	元素大小（字）
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

30.3.7.3 发送 FIFO

发送 FIFO 模式是通过将 MCAN_TXBC.TFQM 配置为 0 实现的。存储在发送 FIFO 中的消息，是从获取索引 MCAN_TXFQS.TFGI 指向的消息开始发送的。每次发送后，获取索引会循环递增，直至发送 FIFO 为空。发送 FIFO 会按消息写入的顺序，发送来自不同发送缓冲区但 ID 相同的消息。MCAN 通过计算获取索引与放入索引之差，得到可用（空闲）的发送 FIFO 元素个数（TXFQS.TFFL）。

新消息必须写入以放入索引 MCAN_TXFQS.TFQPI 指向的发送缓冲区开始的发送 FIFO 中。添加发送请求会使放入索引递增，指向下一空闲的发送 FIFO 元素。当放入索引与获取索引相遇时，会指示发送 FIFO 已满（MCAN_TXFQS.TFQF=1）。这种情况下，在下一条消息已发送且获取索引递增之前，不应继续向发送 FIFO 写入消息。

如果有单条消息添加到发送 FIFO，通过对发送 FIFO 放入索引对应的 MCAN_TXBAR 位写 1 来请求发送。

如果有多条（假设有 n 条）消息添加到发送 FIFO，则会写入以放入索引开始的 n 个连续发送缓冲区中。随后通过 MCAN_TXBAR 请求发送，放入索引循环递增 n。请求发送的发送缓冲区数不应超过发送 FIFO 空闲级别（MCAN_TXFQS.TFFL）指示的空闲发送缓冲区数。

如果获取索引指向的发送缓冲区的发送请求已取消，那么获取索引会递增到下一个有发送请求挂起的发送缓冲区，并重新计算发送 FIFO 空闲级别（MCAN_TXFQS.TFFL）。如果取消其他任何发送缓冲区的发送，获取索引和 FIFO 空闲级别保持不变。

消息 RAM 中分配的发送 FIFO 元素大小为若干个 32 位字（4 字节）。因此，下一个可用（空闲）的发送 FIFO 的起始地址，是通过将放入索引 MCAN_TXFQS.TFQPI（0~31）乘以元素大小（字），再乘以 4，然后与 MCAN_TXBC.TBSA 相加计算得到。

30.3.7.4 发送队列

发送队列模式是通过将 MCAN_TXBC.TFQM 配置为 1 实现的。存储在发送队列中的消息，是先从消息 ID 最小（优先级最高）的开始发送的。如果多个发送队列配置为相同消息 ID，如果多个发送队列配置为相同消息 ID，由于发送顺序与存储消息的发送队列的编号有关，而这些发送队列的编号依赖于当前放入索引的状态，因此无法预测发送顺序。

新消息必须写入放入索引 MCAN_TXFQS.TFQPI 指向的发送缓冲区中。放入索引一直指向发送队列中编号最小的空闲缓冲区。如果发送队列已满 (MCAN_TXFQS.TFQF=1)，则放入索引无效，并且在至少有一个请求的消息已发出或挂起的发送请求已取消之前，不应继续向发送队列写入消息。

应用程序可使用寄存器 MCAN_TXBRP 来代替放入索引，并可将消息放入任何没有挂起发送请求的发送缓冲区中。

消息 RAM 中分配的发送队列缓冲区元素大小为若干个 32 位字 (4 字节)。因此，下一个可用 (空闲) 的发送队列缓冲区的起始地址，是通过将放入索引 MCAN_TXFQS.TFQPI (0~31) 乘以元素大小 (字)，再乘以 4，然后与 MCAN_TXBC.TBSA 相加计算得到。

30.3.7.5 混合使用专用发送缓冲区和发送 FIFO

在这种情况下，消息 RAM 中的发送缓冲区会被划分为一组专用发送缓冲区和一个发送 FIFO。专用发送缓冲区的数量是通过 MCAN_TXBC.NDTB 配置的。分配给发送 FIFO 的发送缓冲区数量是通过 MCAN_TXBC.TFQS 配置的。如果 MCAN_TXBC.TFQS 配置为 0，则仅会使用专用发送缓冲区。

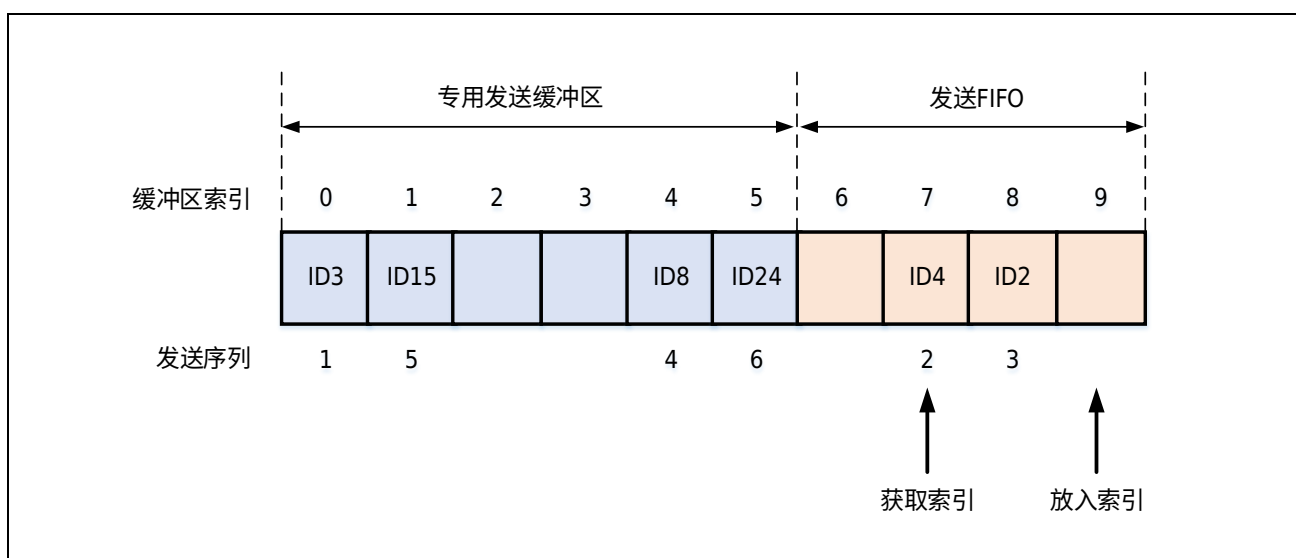


图 30-9 混合配置专用发送缓冲区和发送 FIFO 示例

发送优先级:

- 扫描所有激活了发送请求的专用发送缓冲区和最早挂起发送请求的发送 FIFO 缓冲区 (MCAN_TXEFS.TFGI 指向的缓冲区)
- 消息 ID 最小的缓冲区优先级最高，下次将发送该缓冲区的消息

30.3.7.6 混合使用专用发送缓冲区和发送队列

在这种情况下，消息 RAM 中的发送缓冲区会被划分为一组专用发送缓冲区和一个发送队列。专用发送缓冲区的数量是通过 MCAN_TXBC.NDTB 配置的。发送队列缓冲区的数量是通过 MCAN_TXBC.TFQS 配置的。如果 MCAN_TXBC.TFQS 配置为 0，则仅会使用专用发送缓冲区。

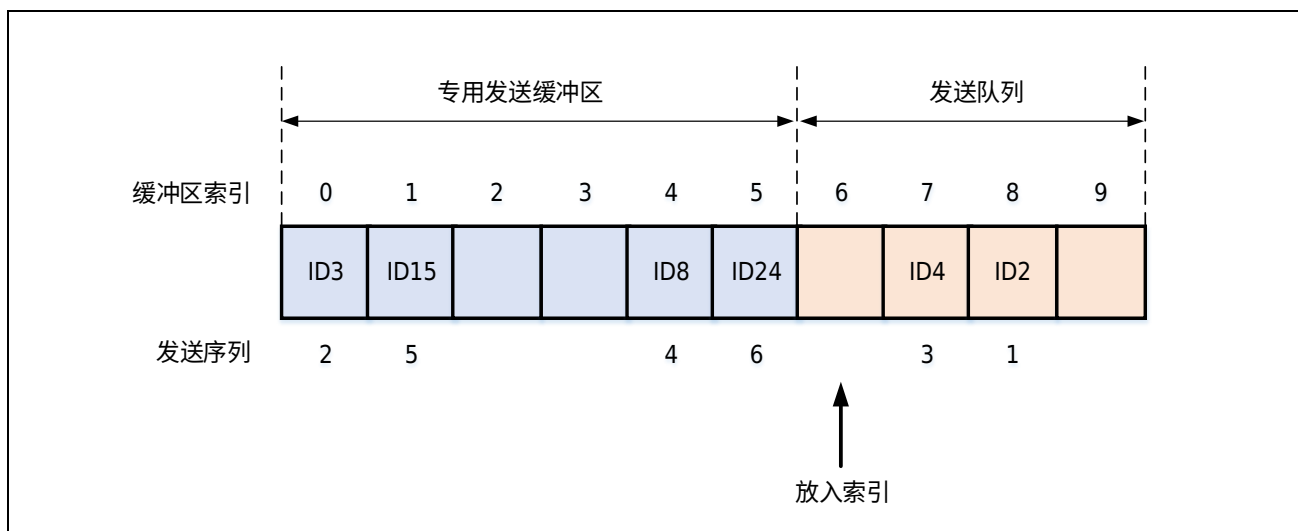


图 30-10 混合配置专用发送缓冲区和发送队列示例

发送优先级:

- 扫描所有激活了发送请求的专用发送缓冲区和发送队列缓冲区
- 消息 ID 最小的发送缓冲区优先级最高，下次将发送该缓冲区的消息

30.3.7.7 发送取消

MCAN 支持发送取消功能。这个功能特别适用于网关应用和基于 AUTOSAR 的应用。要取消专用发送缓冲区或发送队列缓冲区的发送请求，主机必须向寄存器 MCAN_TXBCR 中相应的位（发送缓冲区索引对应的位）写 1。发送取消不适用于发送 FIFO 模式。

成功取消后，寄存器 MCAN_TXBCF 的相应位会置 1。

如果在缓冲区发送过程中请求取消发送，则只要发送正在进行中，MCAN_TXBRP 相应的位就会保持置 1 状态。如果发送成功，MCAN_TXBTO 和 MCAN_TXBCF 相应的位会置 1。如果发送不成功，则不会重复发送，只会将 MCAN_TXBCF 相应的位置位。

注:

如果一个挂起的发送在开始发送之前被立即取消，即使该节点中有另一消息挂起，取消后也会有一个短暂的窗口期，在此期间不会开始任何发送。这样，另一节点便可以发送优先级可能比该节点中第二条消息低的消息。

30.3.7.8 发送事件处理

MCAN 用发送事件 FIFO 来支持发送事件的处理。MCAN 在 CAN 总线上发送消息后，消息 ID 和时间戳会存储在发送事件 FIFO 元素中。为了将发送事件关联到发送事件 FIFO 元素，已发送的发送缓冲区中的消息标记会被复制到发送事件 FIFO 元素中。

发送事件 FIFO 最多可配置 32 个元素，章节【发送事件 FIFO 元素】描述了发送事件 FIFO 元素。

发送事件 FIFO 的用途是将处理发送状态信息与处理发送消息分开，也就是让发送缓冲区仅保存要发送的消息，而将发送状态单独存储在发送事件 FIFO 中。这样做是有优势的，尤其是在处理动态管理的发送队列时，发送缓冲区可在发送成功后立即用于新消息。重写发送缓冲区之前，不需要保存该发送缓冲区的发送状态信息。

当 MCAN_IR.TEFF 指示发送事件 FIFO 已满时，在至少已读取一个元素且发送事件 FIFO 获取索引递增之前，不会继续向发送事件 FIFO 写入元素。如果在发送事件 FIFO 已满时发生发送事件，此事件会被丢弃，中断标志 MCAN_IR.TEFL 置位。

为了避免发送事件 FIFO 溢出，可使用发送事件 FIFO 水位线。当发送事件 FIFO 填充数量达到由 TXEFC.EFWM 配置的发送事件 FIFO 水位线时，中断标志 MCAN_IR.TEFW 置位。

从发送事件 FIFO 读取数据时，必须将发送事件 FIFO 获取索引 TXEFS.EFGI 乘以 2（发送事件 FIFO 元素大小为两个字），再乘以 4，然后与发送事件 FIFO 起始地址 TXEFC.EFSA 相加，以得到目标数据所在的地址。

30.3.8 FIFO 确认处理

接收 FIFO0、接收 FIFO1 和发送事件 FIFO 的获取索引，是通过相应 FIFO 确认索引寄存器进行写操作来控制的，请参见章节【MCAN 接收 FIFO0 确认寄存器 (MCAN_RXF0A)】、章节【MCAN 接收 FIFO1 确认寄存器 (MCAN_RXF1A)】和章节【MCAN 发送事件 FIFO 确认寄存器 (MCAN_TXEFA)】。对 FIFO 确认索引进行写操作会将 FIFO 获取索引设置为 FIFO 确认索引加 1，进而会更新 FIFO 填充级别。有两种用例：

1. 如果只从 FIFO 读取了一个元素（获取索引指向的元素），则应将获取索引值写入 FIFO 确认索引寄存器中。
2. 如果已从 FIFO 中读取了一个序列的元素，则仅需在该序列读取结束时，将最后一个元素的索引值写入 FIFO 确认索引寄存器，即可更新 FIFO 获取索引。

由于 CPU 可自由访问 MCAN 的消息 RAM，在以任意顺序（不考虑获取索引）读取 FIFO 元素时需要特别注意，特别是在从两个接收 FIFO 之一读取高优先级消息时。在这种情况下，不对 FIFO 确认索引寄存器进行写操作，否则会导致获取索引指向错误的位置，还会改变 FIFO 填充级别，一些较早的 FIFO 元素可能会丢失。

注:

应用程序必须确保写入到 FIFO 确认索引寄存器中的值是正确的, MCAN 不会检查数值是否正确。

30.3.9 消息 RAM

30.3.9.1 消息 RAM 配置

消息 RAM 的宽度为 32 位。MCAN 模块最多可在消息 RAM 中分配 512 个字, 如图 30-11 所示。实际应用中不要求对所有部分都进行配置, 各部分之间的顺序也没有限制。

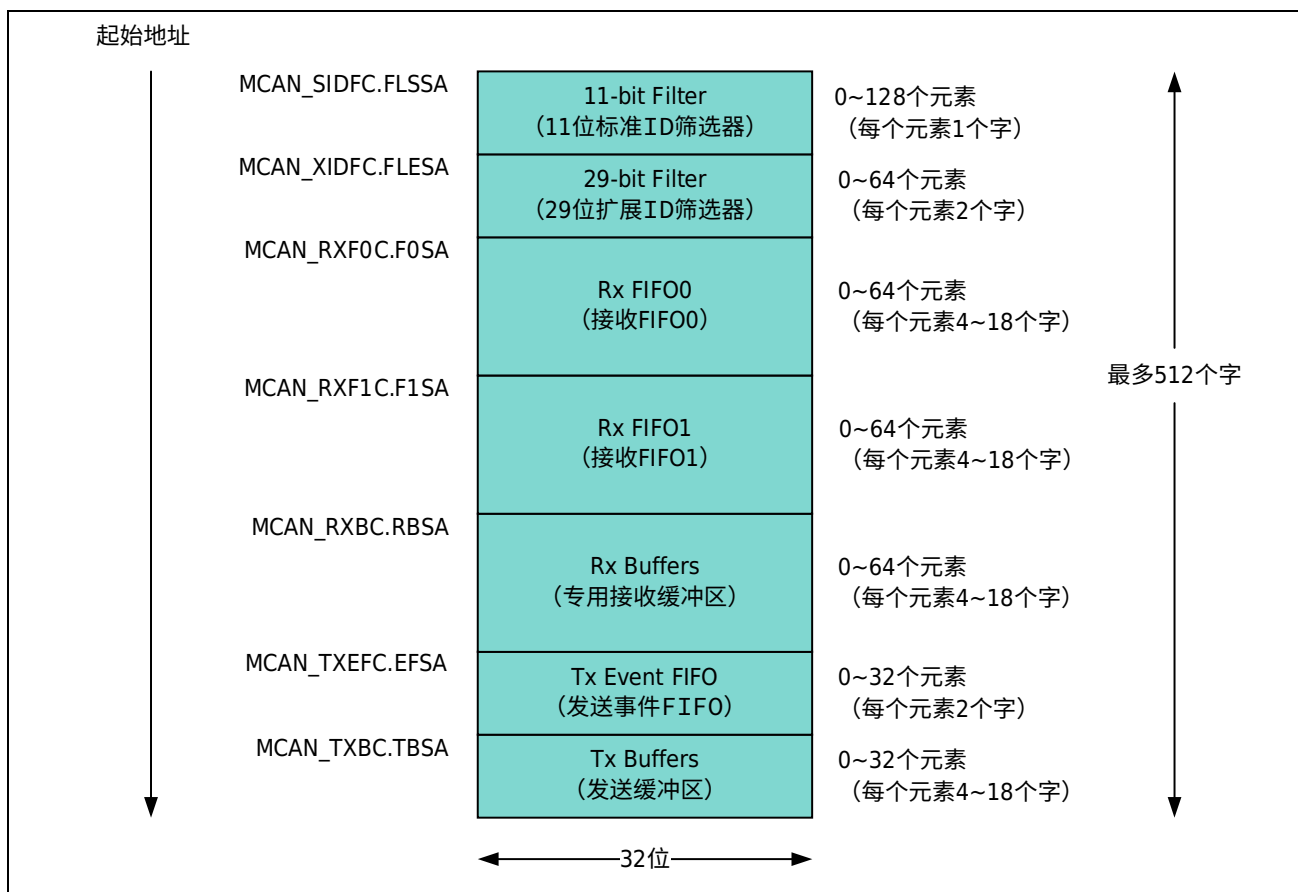


图 30-11 消息 RAM 配置

MCAN 访问消息 RAM 时需要按 32 位对齐的地址进行寻址, 而不是字节对齐的地址。可配置的起始地址是 32 位字地址, 即仅评估地址的 15 到 2 位, 忽略两个最低有效位。

消息 RAM 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。

注:

MCAN 不会检查消息 RAM 的配置是否存在错误, 因此配置不同部分的起始地址以及各部分元素的数量时必须多加留意, 以免造成数据入侵或丢失。

30.3.9.2 接收缓冲区和 FIFO 元素

最多可在消息 RAM 中配置 64 个接收缓冲区和 2 个接收 FIFO。每个接收 FIFO 均可配置为最多存储 64 个消息。接收缓冲区和接收 FIFO 元素的结构如表 30-9 所示，相关说明请见表 30-10。可通过寄存器 MCAN_RXESC 配置元素大小，CAN FD 消息的数据场（Data field）最多可达 64 字节。

表 30-9 接收缓冲区和 FIFO 元素

位	31	24	23	16	15	8	7	0	
R0	ESI	XTD	RTR	ID[28:0]					
R1	ANMF	FIDX[6:0]		Res	DF	BS	DLC[3:0]		RXTS[15:0]
R2	DB3[7:0]			DB2[7:0]		DB1[7:0]		DB0[7:0]	
R3	DB7[7:0]			DB6[7:0]		DB5[7:0]		DB4[7:0]	
...	
Rn	DBm[7:0]			DBm-1[7:0]		DBm-2[7:0]		DBm-3[7:0]	

表 30-10 接收缓冲区和 FIFO 元素说明

位域	标记	说明
R0[31]	ESI	错误状态指示符（Error State Indicator） 0：发送节点为错误主动状态 1：发送节点为错误被动状态
R0[30]	XTD	扩展 ID 标志位（Extended Identifier） 向主机指示接收到的帧的 ID 是标准 ID 还是扩展 ID。 0：11 位标准 ID 1：29 位扩展 ID
R0[29]	RTR	远程发送请求（Remote Transmission Request） 向主机指示接收到的是数据帧还是远程帧。 0：接收到的帧是数据帧 1：接收到的帧是远程帧 注： CAN FD 没有远程帧。在 CAN FD 帧中（DF=1），显性位 RRS（Remote Request Substitution）替代了 RTR 位。
R0[28:0]	ID[28:0]	ID（Identifier） 标准 ID 或扩展 ID 取决于 XTD 位，标准 ID 存储在 ID[28:18]中。
R1[31]	ANMF	接收到非匹配帧（Accepted Non-matching Frame） 可通过 MCAN_GFC.ANFS 和 MCAN_GFC.ANFE 使能接受非匹配帧。 0：接收到的帧与过滤器索引 FIDX 指向的过滤器元素匹配 1：接收到的帧与任何接收过滤器元素都不匹配
R1[30:24]	FIDX[6:0]	过滤器索引（Filter Index）

位域	标记	说明
		0~127: 匹配的接收过滤器元素的索引 (ANMF = 1 时无效) 范围为 0 到 MCAN_SIDFC.LSS]减 1 或 MCAN_XIDFC.LSE 减 1。
R1[21]	FDF	FD 格式 (FD Format) 0: 经典 CAN 帧格式 1: CAN FD 帧格式 (新的 DLC 编码和 CRC)
R1[20]	BRS	比特率切换 (Bit Rate Switch) 0: 接收帧没有比特率切换 1: 接收帧有比特率切换
R1[19:16]	DLC[3:0]	数据长度代码 (Data Length Code) 0~8: 经典 CAN 和 CAN FD: 接收到的帧包含 0~8 个数据字节 9~15: CAN: 接收到的帧包含 8 个数据字节 9~15: CAN FD: 接收到的帧包含 12/16/20/24/32/48/64 个数据字节
R1[15:0]	RXTS[15:0]	接收时间戳 (Rx Timestamp) 接收到 SOF (Start Of Frame) 时捕获的时间戳计数器值。 分辨率取决于时间戳计数器预分频器 MCAN_TSCC.TCP 的配置。
R2[31:24]	DB3[7:0]	数据字节 3 (Data Byte 3)
R2[23:16]	DB2[7:0]	数据字节 2 (Data Byte 2)
R2[15:8]	DB1[7:0]	数据字节 1 (Data Byte 1)
R2[7:0]	DB0[7:0]	数据字节 0 (Data Byte 0)
R3[31:24]	DB7[7:0]	数据字节 7 (Data Byte 7)
R3[23:16]	DB6[7:0]	数据字节 6 (Data Byte 6)
R3[15:8]	DB5[7:0]	数据字节 5 (Data Byte 5)
R3[7:0]	DB4[7:0]	数据字节 4 (Data Byte 4)
...
Rn[31:24]	DBm[7:0]	数据字节 m (Data Byte m)
Rn[23:16]	DBm-1[7:0]	数据字节 m-1 (Data Byte m-1)
Rn[15:8]	DBm-2[7:0]	数据字节 m-2 (Data Byte m-2)
Rn[7:0]	DBm-3[7:0]	数据字节 m-3 (Data Byte m-3)

30.3.9.3 发送缓冲区元素

发送缓冲区可配置为专用发送缓冲区和发送 FIFO（或队列）。如果发送缓冲区由专用发送缓冲区和发送 FIFO（或队列）共享，则专用发送缓冲区在发送缓冲区的起始处，其后是发送 FIFO（或队列）。发送处理单元会评估发送缓冲区配置 MCAN_TXBC.TFQS 和 MCAN_TXBC.NDTB，以区分专用发送缓冲区与发送 FIFO（队列）。发送缓冲区元素的结构如表 30-11 所示，相关说明请见表 30-12。可通过寄存器 MCAN_TXESC 配置元素大小，CAN FD 消息的数据场（Data field）最多可达 64 字节。

表 30-11 发送缓冲区元素

位	31			24			23			16			15			8			7			0		
T0	E	X	R	ID[28:0]																				
	S	T	T																					
	I	D	R																					
T1	MM[7:0]			E	R	F	B	DLC[3:0]			MM[15:8]			Res										
	F	e	s	C	e	D	R																	
T2	DB3[7:0]			DB2[7:0]			DB1[7:0]			DB0[7:0]														
T3	DB7[7:0]			DB6[7:0]			DB5[7:0]			DB4[7:0]														
...														
Tn	DBm[7:0]			DBm-1[7:0]			DBm-2[7:0]			DBm-3[7:0]														

表 30-12 发送缓冲区元素说明

位域	标记	说明
T0[31]	ESI	错误状态指示符（Error State Indicator） 0：CAN FD 格式的 ESI 位仅取决于错误被动标志 1：CAN FD 格式的 ESI 位以隐性位发送 注： 发送缓冲区的 ESI 位会与错误被动标志进行或运算，以确定将发送的 FD 帧 ESI 位的值。 按照 CAN FD 协议规范的要求，错误主动节点可选择发送隐性 ESI 位，但错误被动节点将始终发送隐性 ESI 位。
T0[30]	XTD	扩展 ID 标志位（Extended Identifier） 0：11 位标准 ID 1：29 位扩展 ID
T0[29]	RTR	远程发送请求（Remote Transmission Request） 0：发送数据帧 1：发送远程帧 注： 当 RTR = 1 时，即使 MCAN_CCCR.FDOE 使能以 CAN FD 格式进行发送，MCAN 也会按照 ISO 11898-1:2015 的规定发送远程帧。
T0[28:0]	ID[28:0]	ID (Identifier) 标准 ID 或扩展 ID，取决于 XTD 位，标准 ID 必须写在 ID[28:18]中。
T1[31:24]	MM[7:0]	消息标记（Message Marker）

位域	标记	说明
		在发送缓冲区配置期间由 CPU 写入。复制到发送事件 FIFO 元素中，用于标识发送消息状态。
T1[23]	EFC	事件 FIFO 控制 (Event FIFO Control) 0: 不存储发送事件 1: 存储发送事件
T1[21]	FD	FD 格式 (FD Format) 0: 以经典 CAN 格式发送帧 1: 以 CAN FD 格式发送帧
T1[20]	BRS	比特率切换 (Bit Rate Switch) 0: CAN FD 帧发送时不切换比特率 1: CAN FD 帧发送时切换比特率 注: 仅当 CAN FD 使能 (MCAN_CCCR.FDOE=1) 时，才会评估 ESI、FDF 和 BRS 位。仅当额外设置 MCAN_CCCR.BRSE =1 时，才会评估 BRS 位。
T1[19:16]	DLC[3:0]	数据长度代码 (Data Length Code) 0~8: 经典 CAN 和 CAN FD: 发送帧包含 0~8 个数据字节 9~15: CAN: 发送帧包含 8 个数据字节 9~15: CAN FD: 发送帧包含 12/16/20/24/32/48/64 个数据字节
T1[15:8]	MM[15:8]	宽消息标记的高字节 (High byte of Wide Message Marker) 在发送缓冲区配置期间由 CPU 写入。复制到发送事件 FIFO 元素中，用于标识发送消息状态。仅当 MCAN_CCCR.WMM=1 时有效。
T2[31:24]	DB3[7:0]	数据字节 3 (Data Byte 3) 注: 根据 MCAN_TXESC 寄存器配置的元素大小, 2~16 (Tn = 2 ~17) 个 32 位字用于存储 CAN 消息的数据场 (Data field)。
T2[23:16]	DB2[7:0]	数据字节 2 (Data Byte 2)
T2[15:8]	DB1[7:0]	数据字节 1 (Data Byte 1)
T2[7:0]	DB0[7:0]	数据字节 0 (Data Byte 0)
T3[31:24]	DB7[7:0]	数据字节 7 (Data Byte 7)
T3[23:16]	DB6[7:0]	数据字节 6 (Data Byte 6)
T3[15:8]	DB5[7:0]	数据字节 5 (Data Byte 5)
T3[7:0]	DB4[7:0]	数据字节 4 (Data Byte 4)
...
Tn[31:24]	DBm[7:0]	数据字节 m (Data Byte m)
Tn[23:16]	DBm-1[7:0]	数据字节 m-1 (Data Byte m-1)
Tn[15:8]	DBm-2[7:0]	数据字节 m-2 (Data Byte m-2)
Tn[7:0]	DBm-3[7:0]	数据字节 m-3 (Data Byte m-3)

30.3.9.4 发送事件 FIFO 元素

每个元素存储已发送消息相关的信息。通过读取发送事件 FIFO，主机 CPU 按消息的发送顺序获取这些信息。关于发送事件 FIFO 的状态信息，可从寄存器 MCAN_TXEFS 中获取。发送事件 FIFO 元素的结构如表 30-13 所示，相关说明请见表 30-14。

E1A: 当 MCAN_CCCR.WMM=0 时，E1A.TXTS[15:0]保存由 MCAN 的内部时间戳逻辑生成的 16 位时间戳。

E1B: 当使能 16 位宽消息标记 (MCAN_CCCR.WMM=1) 时，E1B.MM[15:8]为宽消息标记的高 8 位。

表 30-13 发送事件 FIFO 元素

位	31	24	23	16	15	8	7	0
E0	E S I	X T D	R T R	ID[28:0]				
E1A	MM[7:0]		ET [1:0]	FD F	BR S	DLC[3:0]	TXTS[15:0]	
E1B	MM[7:0]		ET [1:0]	FD F	BR S	DLC[3:0]	MM[15:8]	Res

表 30-14 发送事件 FIFO 元素说明

位域	标记	说明
E0[31]	ESI	错误状态指示符 (Error State Indicator) 0: 发送节点为错误主动状态 1: 发送节点为错误被动状态
E0[30]	XTD	扩展 ID 标志位 (Extended Identifier) 0: 11 位标准 ID 1: 29 位扩展 ID
E0[29]	RTR	远程发送请求 (Remote Transmission Request) 0: 发送帧为数据帧 1: 发送帧为远程帧
E0[28:0]	ID[28:0]	标识符 (Identifier) 标准 ID 或扩展 ID，取决于 XTD 位。标准 ID 存储在 ID[28:18]中。
E1A/B[31:24]	MM[7:0]	消息标记 (Message Marker) 从发送缓冲区复制到发送事件元素，用于标识发送消息状态。
E1A/B[23:22]	ET[1:0]	事件类型 (Event Type) 00: 保留 01: 发送事件 10: 取消后仍发送 (在 DAR 模式下发送时始终为该值) 11: 保留
E1A/B[21]	FDF	FD 格式 (FD Format) 0: 经典 CAN 帧格式 1: CAN FD 帧格式 (新的 DLC 编码和 CRC)

位域	标记	说明
E1A/B[20]	BRS	比特率切换 (Bit Rate Switch) 0: CAN FD 帧发送时没有切换比特率 1: CAN FD 帧发送时有切换比特率
E1A/B[19:16]	DLC[3:0]	数据长度代码 (Data Length Code) 0~8: 经典 CAN 和 CAN FD: 发送帧包含 0~8 个数据字节 9~15: 经典 CAN: 发送帧包含 8 个数据字节 9~15: CAN FD: 发送帧包含 12/16/20/24/32/48/64 个数据字节
E1A[15:0]	TXTS[15:0]	发送时间戳 (Tx Timestamp) 发送 SOF (Start Of Frame) 时捕获的时间戳计数器值。分辨率取决于时间戳计数器预分频器 MCAN_TSCC.TCP 的配置。
E1B[15:8]	MM[15:8]	宽消息标记的高字节 (High byte of Wide Message Marker) 在发送缓冲区配置期间由 CPU 写入。复制到发送事件 FIFO 元素中, 用于标识发送消息状态。

30.3.9.5 标准消息 ID 过滤器元素

最多可为 11 位标准 ID 配置 128 个过滤器元素 (每个元素大小为 1 个字)。其地址为过滤器元素索引 (0~127) 乘以 4 (转换成字节), 加上起始地址 MCAN_SIDFC.FLSSA。标准消息 ID 元素的结构如表 30-15 所示, 相关说明请见表 30-16。

表 30-15 标准消息 ID 过滤器元素

位	31	24	23	16	15	8	7	0	
S0	SFT [1:0]	SFEC [2:0]	SFID1[10:0]			Res	SFID2[10:0]		

表 30-16 标准消息 ID 过滤器元素说明

位域	标记	说明
S0[31:30]	SFT[1:0]	标准过滤器类型 (Standard Filter Type) 00: 从 SFID1 到 SFID2 的范围过滤器 (SFID2 ≥ SFID1) 01: SFID1 或 SFID2 定义的双 ID 过滤器 10: 经典过滤器: SFID1=ID, SFID2=掩码 11: 禁止该过滤器元素 注: 当 SFT=0b11 时, 该过滤器元素被禁止, 接收过滤继续进行 (与 SFEC=0b000 时的行为相同)。
S0[29:27]	SFEC[2:0]	标准过滤器元素配置 (Standard Filter Element Configuration) 所有使能的过滤器元素都用于标准帧的接收过滤。当匹配到一个已使能的过滤器元素, 或到达过滤器列表结尾处时, 停止接收过滤。如果 SFEC=0b100、0b101 或 0b110,

位域	标记	说明
		出现匹配时中断标志 MCAN_IR.HPM 置位，并产生中断（如果使能）。在这种情况下，寄存器 MCAN_HPMS 会更新为优先级匹配的状态。 000：禁止该过滤器元素 001：如果匹配，则存储到接收 FIFO0 中 010：如果匹配，则存储到接收 FIFO1 中 011：如果匹配，则拒绝 ID 100：如果匹配，则设置优先级，不存储 101：如果匹配，则设置优先级并存储到 FIFO0 中 110：如果匹配，则设置优先级并存储到 FIFO1 中 111：存储到接收缓冲区中或作为调试消息，忽略 SFT[1:0]的配置
S0[26:16]	SFID[10:0]	标准过滤器 ID1 (Standard Filter ID 1) 标准 ID 过滤器元素的第一个 ID。当为接收缓冲区或调试消息过滤时，此位域定义要存储的标准消息的 ID。接收的 ID 必须完全匹配，不使用掩码机制。
S0[10:0]	SFID2[10:0]	标准过滤器 ID2 (Standard Filter ID 2) 该位域具有不同含义，具体视 SFEC 的配置而定： - SFEC=0b001~0b110 时：标准 ID 过滤器元素的第二个 ID - SFEC=0b111 时：用于接收缓冲区或调试消息的过滤器
	SFID2[10:9]	确定接收的消息存储到接收缓冲区中，还是作为调试消息序列中的消息 A、B 或 C 进行处理。 00：将消息存储到接收缓冲区中 01：调试消息 A 10：调试消息 B 11：调试消息 C
	SFID2[5:0]	定义存储匹配消息的接收缓冲区到起始地址 RXBC.RBSA 的偏移（缓冲区索引）。

30.3.9.6 扩展消息 ID 过滤器元素

最多可为 29 位扩展 ID 配置 64 个过滤器元素（每个元素大小为 2 个字）。访问扩展消息 ID 过滤器元素时，其地址为过滤器元素索引（0~63）乘以 2，再乘以 4（转换成字节），然后加上起始地址 MCAN_XIDFC.FLESA。扩展消息 ID 元素的结构如表 30-17 所示，相关说明请见表 30-18。

表 30-17 扩展消息 ID 过滤器元素

位	31	24	23	16	15	8	7	0
F0	EFEC [2:0]		EFID1[28:0]					
F1	EFT [1:0]	Re s	EFID2[28:0]					

表 30-18 扩展消息 ID 过滤器元素说明

位域	标记	说明
F0[31:29]	EFEC[2:0]	<p>扩展过滤器元素配置 (Extended Filter Element Configuration)</p> <p>所有使能的过滤器元素都用于扩展帧的接收过滤。当匹配到一个已使能的过滤器元素，或到达过滤器列表结尾处时，停止接收过滤。如果 EFEC=0b100、0b101 或 0b110，出现匹配时中断标志 MCAN_IR.HPM 置位，并产生中断（若使能）。在这种情况下，寄存器 MCAN_HPMS 会更新为优先级匹配的状态。</p> <p>000：禁止该过滤器元素</p> <p>001：如果匹配，则存储到接收 FIFO0 中</p> <p>010：如果匹配，则存储到接收 FIFO1 中</p> <p>011：如果匹配，则拒绝 ID</p> <p>100：如果匹配，则设置优先级</p> <p>101：如果匹配，则设置优先级并存储到 FIFO0 中</p> <p>110：如果匹配，则设置优先级并存储到 FIFO1 中</p> <p>111：存储到接收缓冲区中或作为调试消息，忽略 EFT[1:0]的配置</p>
F0[28:0]	EFID1[28:0]	<p>扩展过滤器 ID1 (Extended Filter ID 1)</p> <p>扩展 ID 过滤器元素的第一个 ID。当为接收缓冲区或调试消息过滤时，此位域定义要存储的扩展消息的 ID。接收的 ID 必须完全匹配，仅使用 MCAN_XIDAM 掩码机制。</p>
F1[31:30]	EFT[1:0]	<p>扩展过滤器类型 (Extended Filter Type)</p> <p>00：从 EFID1 到 EFID2 的范围过滤器(EFID2 ≥ EFID1)</p> <p>01：EFID1 或 EFID2 定义的双 ID 过滤器</p> <p>10：经典过滤器：EFID1=ID，EFID2=掩码</p> <p>11：从 EFID1 到 EFID2 的范围过滤器 (EFID2 ≥ EFID1)，不使用 MCAN_XIDAM 掩码</p>
F1[28:0]	EFID2[28:0]	<p>扩展过滤器 ID2 (Extended Filter ID 2)</p> <p>该位域具有不同含义，具体视 EFEC 的配置而定：</p> <ul style="list-style-type: none"> - EFEC = 0b001~0b110 时：扩展 ID 过滤器元素的第二个 ID - EFEC = 0b111 时：用于接收缓冲区或调试消息的过滤器
	EFID2[10:9]	<p>确定接收的消息存储到接收缓冲区中，还是作为调试消息序列中的消息 A、B 或 C 进行处理。</p> <p>00：将消息存储到接收缓冲区中</p> <p>01：调试消息 A</p> <p>10：调试消息 B</p> <p>11：调试消息 C</p>
	EFID2[5:0]	<p>定义存储匹配消息的接收缓冲区到起始地址 MCAN_RXBC.RBSA 的偏移（缓冲区索引）。</p>

30.4 寄存器描述

表 30-19 MCAN/MCANRAM 基地址

名称	基地址	描述
MCAN1	0x4002 9000	MCAN1基地址
MCAN2	0x4002 9400	MCAN2基地址
MCANRAM	0x4007 3800	MCANRAM基地址

表 30-20 MCAN 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
MCAN_ENDN	MCAN字节序寄存器	0x0004	32	0x8765 4321
MCAN_DBTP	MCAN数据位时间和预分频寄存器	0x000C	32	0x0000 0A33
MCAN_TEST	MCAN测试寄存器	0x0010	32	0x0000 00X0
MCAN_RWD	MCAN RAM看门狗寄存器	0x0014	32	0x0000 0000
MCAN_CCCR	MCAN CC控制寄存器	0x0018	32	0x0000 0001
MCAN_NBTP	MCAN标称位时间和预分频寄存器	0x001C	32	0x0600 0A03
MCAN_TSCC	MCAN时间戳计数器配置寄存器	0x0020	32	0x0000 0000
MCAN_TSCV	MCAN时间戳计数器值寄存器	0x0024	32	0x0000 0000
MCAN_TOCC	MCAN超时计数器配置寄存器	0x0028	32	0xFFFF 0000
MCAN_TOCV	MCAN超时计数器值寄存器	0x002C	32	0x0000 FFFF
MCAN_ECR	MCAN错误计数器寄存器	0x0040	32	0x0000 0000
MCAN_PSR	MCAN协议状态寄存器	0x0044	32	0x0000 0707
MCAN_TDCR	MCAN发送延迟补偿寄存器	0x0048	32	0x0000 0000
MCAN_IR	MCAN中断寄存器	0x0050	32	0x0000 0000
MCAN_IE	MCAN中断许可寄存器	0x0054	32	0x0000 0000
MCAN_ILS	MCAN中断线选择寄存器	0x0058	32	0x0000 0000
MCAN_ILE	MCAN中断线使能寄存器	0x005C	32	0x0000 0000
MCAN_GFC	MCAN全局过滤器配置寄存器	0x0080	32	0x0000 0000
MCAN_SIDFC	MCAN标准ID过滤器配置寄存器	0x0084	32	0x0000 0000
MCAN_XIDFC	MCAN扩展ID过滤器配置寄存器	0x0088	32	0x0000 0000
MCAN_XIDAM	MCAN扩展ID与掩码寄存器	0x0090	32	0x1FFF FFFF
MCAN_HPMS	MCAN高优先级消息状态寄存器	0x0094	32	0x0000 0000
MCAN_NDAT1	MCAN新数据1寄存器	0x0098	32	0x0000 0000
MCAN_NDAT2	MCAN新数据2寄存器	0x009C	32	0x0000 0000
MCAN_RXFOC	MCAN接收FIFO0配置寄存器	0x00A0	32	0x0000 0000
MCAN_RXF0S	MCAN接收FIFO0状态寄存器	0x00A4	32	0x0000 0000
MCAN_RXFOA	MCAN接收FIFO0确认寄存器	0x00A8	32	0x0000 0000
MCAN_RXBC	MCAN接收缓冲区配置寄存器	0x00AC	32	0x0000 0000

寄存器	描述	偏移地址	位宽	复位值
MCAN_RXF1C	MCAN接收FIFO1配置寄存器	0x00B0	32	0x0000 0000
MCAN_RXF1S	MCAN接收FIFO1状态寄存器	0x00B4	32	0x0000 0000
MCAN_RXF1A	MCAN接收FIFO1确认寄存器	0x00B8	32	0x0000 0000
MCAN_RXESC	MCAN接收缓冲区和FIFO元素大小配置寄存器	0x00BC	32	0x0000 0000
MCAN_TXBC	MCAN发送缓冲区配置寄存器	0x00C0	32	0x0000 0000
MCAN_TXFQS	MCAN发送FIFO/队列状态寄存器	0x00C4	32	0x0000 0000
MCAN_TXESC	MCAN发送缓冲区元素大小配置寄存器	0x00C8	32	0x0000 0000
MCAN_TXBRP	MCAN发送缓冲区请求挂起寄存器	0x00CC	32	0x0000 0000
MCAN_TXBAR	MCAN发送缓冲区添加请求寄存器	0x00D0	32	0x0000 0000
MCAN_TXBCR	MCAN发送缓冲区取消请求寄存器	0x00D4	32	0x0000 0000
MCAN_TXBTO	MCAN发送缓冲区发送已发生寄存器	0x00D8	32	0x0000 0000
MCAN_TXBCF	MCAN发送缓冲区取消已完成寄存器	0x00DC	32	0x0000 0000
MCAN_TXBTIE	MCAN发送缓冲区发送中断使能寄存器	0x00E0	32	0x0000 0000
MCAN_TXBCIE	MCAN发送缓冲区取消已完成中断使能寄存器	0x00E4	32	0x0000 0000
MCAN_TXEFC	MCAN发送事件FIFO配置寄存器	0x00F0	32	0x0000 0000
MCAN_TXEFS	MCAN发送事件FIFO状态寄存器	0x00F4	32	0x0000 0000
MCAN_TXEFA	MCAN发送事件FIFO确认寄存器	0x00F8	32	0x0000 0000

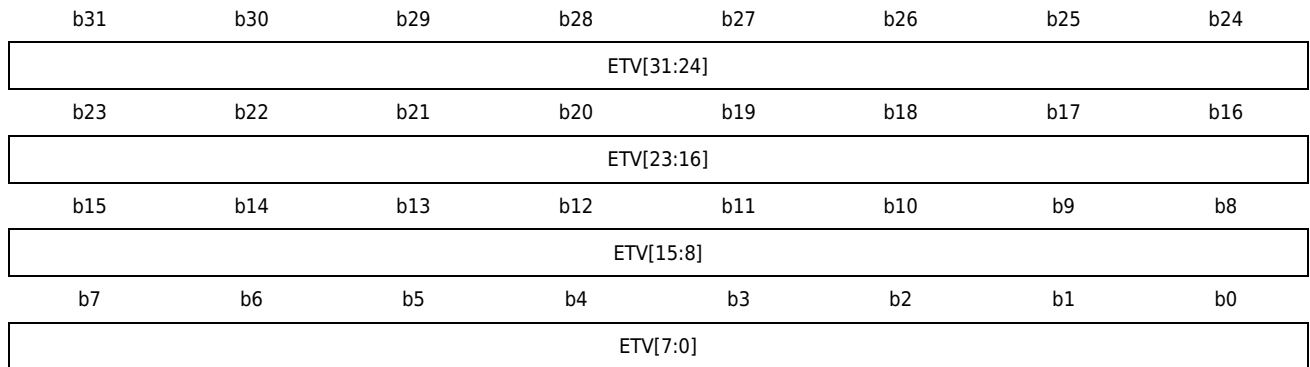
表 30-21 MCANRAM 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
MCANRAM_CKCR	MCAN消息RAM校验控制寄存器	0x00	32	0x00000000
MCANRAM_CKSR	MCAN消息RAM校验状态寄存器	0x04	32	0x00000000
MCANRAM_EIEN	MCAN消息RAM错误注入使能寄存器	0x08	32	0x00000000
MCANRAM_EIBIT0	MCAN消息RAM错误注入位使能寄存器0	0x0C	32	0x00000000
MCANRAM_EIBIT1	MCAN消息RAM错误注入位使能寄存器1	0x10	32	0x00000000
MCANRAM_ECCERRADDR	MCAN消息RAM校验错误地址寄存器	0x14	32	0x00000000

30.4.1 MCAN 字节序寄存器 (MCAN_ENDN)

MCAN Endianness Register

复位值: 0x8765 4321



位/位域	标记	位名	功能	读写
b31~b0	ETV[31:0]	字节序测试值	测试值为0x8765 4321.	R

30.4.2 MCAN 数据位时间和预分频寄存器 (MCAN_DBTP)

MCAN Data Bit Timing and Prescaler Register

复位值: 0x0000 0A33

仅当 MCAN_CCCR.CCE 和 MCAN_CCCR.INIT 位均被置位时, 才能对该寄存器进行写访问。CAN 位时间的可编程范围为 4 到 49 个时间片 (Time quanta, tq)。CAN 时间片的可编程范围为 1 到 32 个 MCAN 通信时钟周期 (Minimum time quantum, mtq)。 $tq = (DBRP + 1)mtq$ 。

DTSEG1 为 Prop_Seg 与 Phase_Seg1 之和。DTSEG2 为 Phase_Seg2。

因此, 位时间的长度为 (编程值) $[DTSEG1 + DTSEG2 + 3] tq$ 或 (功能值) $[Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] tq$ 。

信息处理时间 (IPT) 为零, 这意味着下一位数据在采样点之后的第一个时钟边沿可用。

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
TDC	Res		DBRP[4:0]				
b15	b14	b13	b12	b11	b10	b9	b8
Res			DTSEG1[4:0]				
b7	b6	b5	b4	b3	b2	b1	b0
DTSEG2[3:0]				DSJW[3:0]			

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为0, 写入时请写0	R
b23	TDC	发送延迟补偿	0: 禁止发送延迟补偿 1: 使能发送延迟补偿	RWA
b22~b21	Res	保留位	读出时为0, 写入时请写0	R
b20~b16	DBRP[4:0]	数据比特率预分频	生成位时间片的MCAN通信时钟预分频值, 位时间为该时间片的倍数。预分频有效值为0到31, 当TDC=1时, 范围被限制在0到1, 硬件实际使用值为编程值加1。	RWA
b15~b13	Res	保留位	读出时为0, 写入时请写0	R
b12~b8	DTSEG1[4:0]	采样点之前的数据时间	有效值为0到31, 硬件实际使用值为编程值加1。	RWA
b7~b4	DTSEG2[3:0]	采样点之后的数据时间	有效值为1到15, 硬件实际使用值为编程值加1。	RWA
b3~b0	DSJW[3:0]	数据 (重新) 同步跳转宽度	有效值为0到15, 硬件实际使用值为编程值加1。	RWA

注:

1. 当 MCAN 通信时钟为 8MHz 时, 复位值 0x0000 0A33 会将 MCAN 数据阶段 (Data phase) 的比特率配置为 500Kbps。
2. 通过 DBTP 配置的 CAN FD 数据阶段 (Data phase) 比特率不能小于通过 NBTP 配置的仲裁阶段 (Arbitration phase) 比特率。

30.4.3 MCAN 测试寄存器 (MCAN_TEST)

MCAN Test Register

复位值：0x0000 00X0（位 7 的复位值由 MCAN_RX 引脚的实际电平决定）

仅当 MCAN_CCCR.TEST 位被置 1 后，才能对该测试寄存器进行写访问。当 MCAN_CCCR.TEST 位被清 0 时，所有测试寄存器功能都会设为其复位值。

回环模式和软件控制 MCAN_TX 引脚属于硬件测试模式。TX 配置为非 0b00 时可能会干扰 CAN 总线上的消息传输。

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res		SVAL	TXBNS[4:0]				
b15	b14	b13	b12	b11	b10	b9	b8
Res		PVAL	TXBNP[4:0]				
b7	b6	b5	b4	b3	b2	b1	b0
RX	TX[1:0]		LBCK	Res			

位/位域	标记	位名	功能	读写
b31~b22	Res	保留位	读出时为0，写入时请写0	R
b21	SVAL	已开始有效	0: TXBNS的值无效 1: TXBNS的值有效	R
b20~b16	TXBNS[4:0]	已开始发送的缓冲区编号	最后开始发送的发送缓冲区编号。 当SVAL位被置1时有效，有效范围为0~31。	R
b15~b14	Res	保留位	读出时为0，写入时请写0	R
b13	PVAL	已准备有效	0: TXBNP的值无效 1: TXBNP的值有效	R
b12~b8	TXBNP[4:0]	已准备好发送的缓冲区编号	准备好发送的发送缓冲区编号。 当PVAL位被置1时有效，有效范围为0~31。	R
b7	RX	接收引脚	监测到的MCAN_RX引脚的实际值 0: CAN总线为显性 (MCAN_RX=0) 1: CAN总线为隐性 (MCAN_RX=1)	R
b6~b5	TX[1:0]	发送引脚控制	00: 复位值，MCAN_TX引脚由CAN内核控制，在CAN位时间结束时更新 01: 可以在MCAN_TX引脚监测采样点 10: MCAN_TX引脚输出显性电平 (0) 11: MCAN_TX引脚输出隐性电平 (1)	RWA
b4	LBCK	回环模式	0: 复位值，禁止回环模式 1: 使能回环模式 (详见【外部回环模式】和【内部回环模式】章节)	RWA
b3~b0	Res	保留位	读出时为0，写入时请写0	R

30.4.4 MCAN RAM 看门狗寄存器 (MCAN_RWD)

MCAN RAM Watchdog Register

复位值: 0x0000 0000

RAM 看门狗会监视消息 RAM 的 READY 信号输出。对消息 RAM 进行访问时, 会启动消息 RAM 看门狗计数器, 该计数器的起始值由 MCAN_RWD.WDC 位进行配置。

当消息 RAM 通过激活其 READY 信号输出, 指示访问成功完成时, 该计数器会重新载入 MCAN_RWD.WDC 的值。如果计数器递减为 0 之前消息 RAM 没有响应, 计数器将停止计数, 并将中断标志 MCAN_IR.WDI 位置 1。RAM 看门狗计数器由 MCAN 控制逻辑时钟计数。

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
WDV[7:0]							
b7	b6	b5	b4	b3	b2	b1	b0
WDC[7:0]							

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为0, 写入时请写0	R
b15~b8	WDV[7:0]	看门狗计数值	消息RAM看门狗计数器的实际值	R
b7~b0	WDC[7:0]	看门狗配置	消息RAM看门狗计数器的起始值。 如果使用初始值0x00, 则计数器被禁止。	RWA

30.4.5 MCAN CC 控制寄存器 (MCAN_CCCR)

MCAN CC Control Register

复位值: 0x0000 0001

寄存器各个位的设置条件请参见【软件初始化】章节。

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
NISO	TXP	EFBI	PXHD	WMM	Res	BRSE	FDOE
b7	b6	b5	b4	b3	b2	b1	b0
TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为0, 写入时请写0	R
b15	NISO	非ISO模式选择	0: 使用ISO 11898-1:2015 规定的CAN FD帧格式 1: 使用Bosch CAN FD规范V1.0规定的CAN FD帧格式	RWA
b14	TXP	传输暂停	若该位被置为1, 则MCAN在成功传输完一帧数据后会暂停两个位时间, 再进行下一帧数据的传输。 0: 禁止传输暂停 1: 使能传输暂停	RWA
b13	EFBI	总线同步期间的边沿滤波	0: 禁止边沿滤波 1: 需要两个连续的显性tq才能检测硬同步边沿	RWA
b12	PXHD	协议异常处理禁止	0: 使能协议异常处理 1: 禁止协议异常处理 注: 当协议异常处理被禁止时, MCAN将在检测到协议异常条件时传输错误帧。	RWA
b11	WMM	宽消息标记	当使用16位宽消息标记 (WMM=1) 时, 发送事件FIFO中的16位内部时间戳是无效的。0: 使用8位消息标记 1: 使用16位消息标记, 替换发送事件FIFO中的16位时间戳	RWA
b10	Res	保留位	读出时为0, 写入时请写0	R
b9	BRSE	比特率切换	0: 禁止发送时进行比特率切换 1: 使能发送时进行比特率切换 注: 当CAN FD模式无效 (FDOE=0) 时, BRSE的设置不起作用。	RWA
b8	FDOE	FD模式使能	0: 禁止FD模式 1: 使能FD模式	RWA
b7	TEST	测试模式使能	0: 正常模式, 测试寄存器MCAN_TEST保持复位值 1: 测试模式, 使能写访问测试寄存器MCAN_TEST	RW1A
b6	DAR	禁止自动重发	0: 使能允许自动重发未成功发送的消息 1: 禁止自动重发	RWA
b5	MON	总线监听模式	仅当CCE和INIT位均置1时, 主机才能将MON位置1。此位可随时通过主机复位。 0: 禁止总线监听模式	RW1A

			1: 使能总线监听模式	
b4	CSR	时钟停止请求	<p>当已请求时钟停止时, 在所有挂起的传输请求均已完成且CAN总线达到空闲状态之后, INIT位和CSA位将会依次被置1。</p> <p>0: 未请求时钟停止</p> <p>1: 已请求时钟停止</p>	RW
b3	CSA	时钟停止确认	<p>0: 未确认时钟停止请求</p> <p>1: 已确认时钟停止请求, 可以通过停止控制逻辑时钟和通信时钟将MCAN设为掉电状态。</p>	R
b2	ASM	受限操作模式	<p>仅当CCE和INIT均置1时, 主机才能将ASM位置1。此位可随时通过主机复位。有关受限工作模式的说明, 请参阅【受限工作模式】章节。</p> <p>0: 正常模式</p> <p>1: 受限工作模式</p>	RW1A
b1	CCE	配置更改使能	<p>0: CPU对受保护的配置寄存器无写访问权限</p> <p>1: CPU对受保护的配置寄存器有写访问权限 (当INIT=1时)</p>	RWA
b0	INIT	初始化	<p>0: 正常工作</p> <p>1: 启动初始化</p> <p>注:</p> <p>由于两个时钟域之间存在同步机制, INIT的值在写入后可能会有一定延迟才能被正确读取。因此, 在重新设定INIT值之前, 请确认之前的设定值确实已被写入。</p>	RW

30.4.6 MCAN 标称位时间和预分频寄存器 (MCAN_NBTP)

MCAN Nominal Bit Timing and Prescaler Register

复位值: 0x0600 0A03

仅当 MCAN_CCCR.CCE 和 MCAN_CCR.INIT 位均被置 1 时, 才能对该寄存器进行写访问。CAN 标称位时间的可编程范围为 4 到 385 个时间片 (Time quanta, tq)。CAN 时间片的可编程范围为 1 到 512 个 MCAN 通信时钟周期 (Minimum time quantum, mtq)。 $tq = (NBRP + 1)mtq$ 。

NTSEG1 为 Prop_Seg 与 Phase_Seg1 之和。NTSEG2 为 Phase_Seg2。

因此, 位时间的长度为 (编程值) $[NTSEG1 + NTSEG2 + 3] tq$ 或 (功能值) $[Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] tq$ 。

信息处理时间 (IPT) 为零, 这意味着下一位数据在采样点之后的第一个时钟边沿可用。

b31	b30	b29	b28	b27	b26	b25	b24
NSJW[6:0]							NBRP[8]
b23	b22	b21	b20	b19	b18	b17	b16
NBRP[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8
NTSEG1[7:0]							
b7	b6	b5	b4	b3	b2	b1	b0
Res	NTSEG2[6:0]						

位/位域	标记	位名	功能	读写
b31~b25	NSJW[6:0]	标称 (重) 同步跳转宽度	有效值为0到127, 硬件实际使用值为编程值加1。	RWA
b24~b16	NBRP[8:0]	标称比特率预分频	生成位时间片的MCAN通信时钟预分频值, 位时间为该时间片的倍数。预分频有效值为0到511, 硬件实际使用值为编程值加1。	RWA
b15~b8	NTSEG1[7:0]	采样点之前的标称时间段	有效值为1到255, 硬件实际使用值为编程值加1。	RWA
b7	Res	保留位	读出时为0, 写入时请写0	R
b6~b0	NTSEG2[6:0]	采样点之后的标称时间段	有效值为1到127, 硬件实际使用值为编程值加1。	RWA

注:

当 MCAN 时钟为 8MHz 时, 复位值 0x0600 0A03 会将 MCAN 的比特率配置为 500Kbps。

30.4.7 MCAN 时间戳计数器配置寄存器 (MCAN_TSCC)

MCAN Timestamp Counter Configuration Register

复位值: 0x0000 0000

该寄存器用于配置内部 16 位时间戳计数器。内部时间戳处理的介绍请参见章节【时间戳生成】。

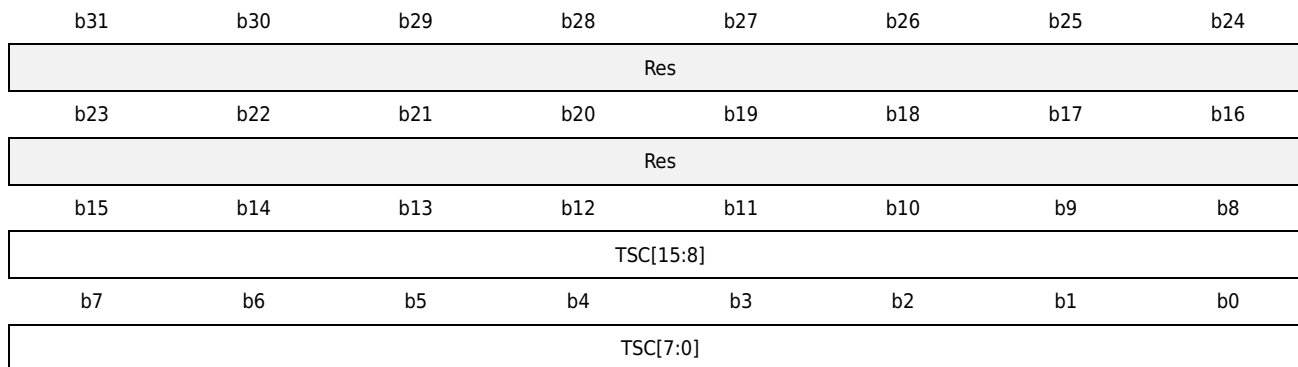
b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res				TCP[3:0]			
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res						TSS[1:0]	

位/位域	标记	位名	功能	读写
b31~b20	Res	保留位	读出时为0, 写入时请写0	R
b19~b16	TCP[3:0]	时间戳计数器预分频	将时间戳和超时计数器时间单位配置为CAN位时间的1~16倍。 有效设定值为0~15, 硬件实际使用值为编程值加1。	RWA
b15~b2	Res	保留位	读出时为0, 写入时请写0	R
b1~b0	TSS[1:0]	时间戳选择	00: 时间戳计数器的值一直为0x0000 01: 时间戳计数器的值根据TCP递增 10: 禁止设定 11: 时间戳计数器的值一直为0x0000 (等同于设定值“00”)	RWA

30.4.8 MCAN 时间戳计数器值寄存器 (MCAN_TSCV)

MCAN Timestamp Counter Value Register

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为0, 写入时请写0	R
b15~b0	TSC[15:0]	时间戳计数器	<p>在接收到或发送SOF (Start Of Frame) 时捕获的内部时间戳计数器值。</p> <p>当MCAN_TSCC.TSS=0b01时, 时间戳计数器以CAN位时间的1~16倍进行递增, 具体倍数由MCAN_TSCC.TCP的设定值决定。计数回卷时, 中断标志MCAN_IR.TSW会被置1。</p> <p>写操作会将计数器清零。</p> <p>注:</p> <p>1. “回卷”是指时间戳计数器的值从非零变为零, 并不是由于MCAN_TSCV的写操作引起的清零。</p> <p>2. 字节访问时, 对该寄存器的任何一个字节进行写操作都会导致时间戳寄存器被清零。</p>	RWC

30.4.9 MCAN 超时计数器配置寄存器 (MCAN_TOCC)

MCAN Timeout Counter Configuration Register

复位值: 0xFFFF 0000

超时计数器的介绍请参见章节【超时计数器】章节。

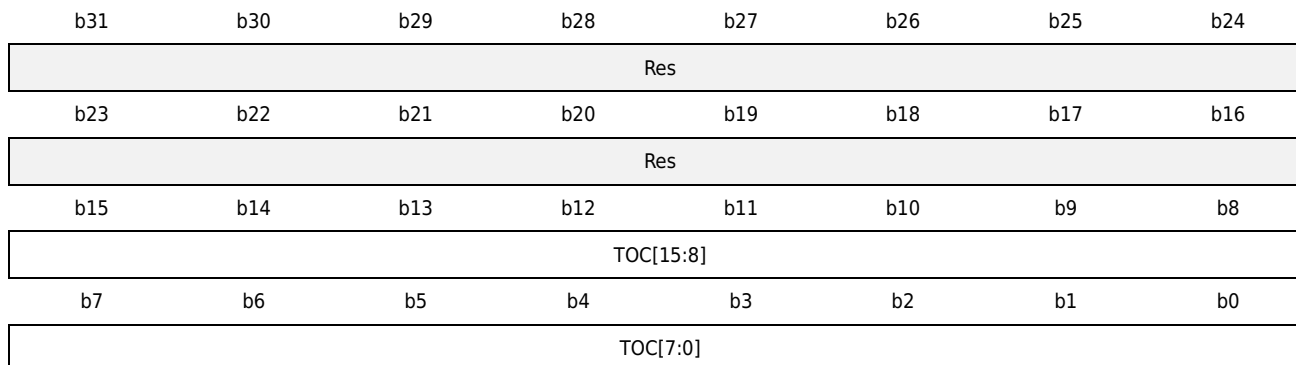
b31	b30	b29	b28	b27	b26	b25	b24
TOP[15:8]							
b23	b22	b21	b20	b19	b18	b17	b16
TOP[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res					TOS[1:0]		ETOC

位/位域	标记	位名	功能	读写
b31~b16	TOP[15:0]	超时周期	超时计数器（递减计数器）的起始值，配置超时周期。	RW
b15~b3	Res	保留位	读出时为0，写入时请写0	R
b2~b1	TOS[1:0]	超时选择	在连续模式下操作时，对MCAN_TOCV进行写访问会将计数器预设为MCAN_TOCC.TOP配置的值，并继续向下递减。当超时计数器由其中一个FIFO控制时，空FIFO会将计数器预设为MCAN_TOCC.TOP配置的值。当第一个FIFO元素被存储后，计数器开始递减计数。 00: 连续模式 01: 由发送事件FIFO控制超时 10: 由接收FIFO0控制超时 11: 由接收FIFO1控制超时	RWA
b0	ETOC	超时计数器使能	0: 禁止超时计数器 1: 使能超时计数器	RWA

30.4.10 MCAN 超时计数器值寄存器 (MCAN_TOCV)

MCAN Timeout Counter Value Register

复位值: 0x0000 FFFF



位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为0, 写入时请写0	R
b15~b0	TOC[15:0]	超时计数器	超时计数器以CAN位时间的1~16倍进行递减, 具体倍数由MCAN_TSCC.TCP的设定值决定。 当计数器递减到零时, 中断标志位MCAN_IR.TOO会被置1, 计数器停止计数。可以通过MCAN_TOCC.TOS来设置启动和复位(或重启)条件。 注: 字节访问时, 当MCAN_TOCC.TOS=0b00时, 对该寄存器的任何一个字节进行写操作都会将超时计数器预设为MCAN_TOCC.TOP配置的值。	RW

30.4.11 MCAN 错误计数器寄存器 (MCAN_ECR)

MCAN Error Counter Register

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
CEL[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8
RP	REC[6:0]						
b7	b6	b5	b4	b3	b2	b1	b0
TEC[7:0]							

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为0, 写入时请写0	R
b23~b16	CEL[7:0]	CAN错误记录	当CAN协议错误导致发送错误计数器TEC或接收错误计数器REC递增时, 该错误记录计数器CEL都会递增。计数器值达到0xFF时, 会停止计数。只有RP位置位但REC没有变化时, CEL不会递增。CEL在REC和TEC增加后递增。 对CEL进行读访问会复位计数器。计数器值达到0xFF时, 停止计数, TEC或REC下一次 RC 增加时中断标志MCAN_IR.ELO置位。 注: 字节访问时, 读取寄存器的字节2会将CEL清零, 读取字节3、1或0没有影响。	RC
b15	RP	接收错误被动	0: 接收错误计数器低于错误被动级别128 1: 接收错误计数器已达到错误被动级别128	R
b14~b8	REC[6:0]	接收错误计数器	接收错误计数器的实际状态, 计数值范围为0~127。	R
b7~b0	TEC[7:0]	发送错误计数器	发送错误计数器的实际状态, 计数值范围为0~255。	R

注:

如果 MCAN_CCCR.ASM 被置为 1, 则检测到 CAN 协议错误时, CAN 协议控制器不会使 TEC 和 REC 递增, 但 CEL 仍会递增。

30.4.12 MCAN 协议状态寄存器 (MCAN_PSR)

MCAN Protocol Status Register

复位值: 0x0000 0707

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res	TDCV[6:0]						
b15	b14	b13	b12	b11	b10	b9	b8
Res	PXE	RFDF	RBRS	RESI	DLEC[2:0]		
b7	b6	b5	b4	b3	b2	b1	b0
BO	EW	EP	ACT[1:0]		LEC[2:0]		

位/位域	标记	位名	功能	读写
b31~b23	Res	保留位	读出时为0, 写入时请写0	R
b22~b16	TDCV[6:0]	发送延迟补偿值	实际的发送延迟补偿值, 有效值为0~127个mtq。 第二采样点 (Secondary Sample Point, SSP) 的位置, 由MCAN_TX引脚到MCAN_RX引脚之间测得的延迟值与MCAN_TDCR.TDCO之和决定。在数据阶段 (Data phase), SSP位置是发送位的起始点与第二采样点间的mtq数。	R
b15	Res	保留位	读出时为0, 写入时请写0	R
b14	PXE	协议异常事件	0: 自上次读访问后未发生协议异常事件 1: 已发生协议异常事件 访问类型为RX: 进行读访问时复位。 注: 字节访问时, 读取寄存器的字节0会将PXE清零, 读取字节3/2/1没有影响。	R
b13	RFDF	收到CAN FD消息	此位的置位与接收过滤无关。 0: 自从上次被CPU清零后未收到CAN FD格式的消息 1: 收到FDF标志为1的CAN FD格式的消息 注: 字节访问时, 读取寄存器的字节0会将RFDF清零, 读取字节3/2/1没有影响。	R
b12	RBRS	上次收到的CAN FD消息的BRS标志	该位与RFDF一起置位, 与接收过滤器无关。 0: 上次收到的CAN FD消息的BRS标志未置位 1: 上次收到的CAN FD消息的BRS标志已置位 注: 字节访问时, 读取寄存器的字节0会将RBRS清零, 读取字节3/2/1没有影响。	R
b11	RESI	上次收到的CAN FD消息的ESI标志	该位与RFDF一起被设置, 与验收过滤器无关。 0: 上次收到的CAN FD消息的ESI标志未置位 1: 上次收到的CAN FD消息的ESI标志已置位 注: 字节访问时, 读取寄存器的字节0会将RESI清零, 读取字节3/2/1没有影响。	R
b10~b8	DLEC[2:0]	数据阶段最后一次错误码	在设置了BRS标志的CAN FD帧的数据阶段 (Data phase) 中发生的最后一个错误的类型。错误码与LEC相同。当设置了BRS标志的CAN FD帧传输 (接收或发送) 没有错误时, DLEC将被清零。 注: 字节访问时, 读取寄存器的字节0会将DLEC设置为0b111, 读取字节3/2/1没有影响。	R

b7	BO	Buss_Off状态	0: MCAN未处于Bus_Off状态 1: MCAN处于Bus_Off状态	R
b6	EW	警告状态	0: 两个错误计数器均小于错误警告 (Error_Warning) 上限96 1: 至少有一个错误计数器已达到错误警告 (Error_Warning) 上限96	R
b5	EP	错误被动	0: MCAN处于错误主动 (Error_Active) 状态。MCAN正常参与总线通信, 并在检测到错误时发送主动错误标志。 1: MCAN处于错误被动 (Error_Passive) 状态	R
b4~b3	ACT[1:0]	通信状态	监测MCAN的通信状态。 00: 同步中, 节点正在进行CAN通信同步 01: 空闲, 节点既不是接收节点也不是发送节点 10: 接收, 节点作为接收节点工作 11: 发送, 节点作为发送节点工作 注: 协议异常事件会将ACT设为0b00	R
b2~b0	LEC[2:0]	最后一次错误码	0: 没有错误 (No Error)。自成功接收或发送使LEC复位之后, 没有错误发生。 1: 填充错误 (Stuff Error)。在已接收到的消息中, 某一部分出现了5个以上的相同位, 这种情况是不允许的。 2: 格式错误 (Form Error)。已接收帧的固定格式部分有格式错误。 3: ACK错误 (ACK Error)。MCAN发送的消息没有被其他节点确认。 4: Bit1错误 (Bit1 Error)。在消息的发送过程中 (仲裁场Arbitration field除外), 设备发送了一个隐性电平 (逻辑值1), 但是监测到的总线电平为显性 (逻辑值0)。 5: Bit0错误 (Bit0 Error)。在消息的发送过程中 (包括ACK位、主动错误标志、过载标志), 设备发送了一个显性电平 (数据或ID的逻辑值0), 但是监测到的总线电平为隐性 (逻辑值1)。在总线Bus_Off恢复期间, 每次监测到11位隐性位序列时, 都会设置为该状态。这样, CPU就能够监测总线恢复序列的进程 (表明总线未卡在显性状态或持续受到干扰)。 6: CRC错误 (CRC Error)。已接收消息的CRC有错误。收到的CRC和用接收到的数据计算出的CRC不匹配。 7: 无变化 (No Change)。对协议状态寄存器的任何读访问会将LEC重新初始化为7。当LEC显示值为7时, 表示自上次CPU对协议状态寄存器进行读访问之后, 没有检测到CAN总线事件。 注: 1.当设置了BRS标志的CAN FD帧已经到达数据阶段 (Data phase) 时, 下一个CAN事件 (错误或有效帧) 将显示在DLEC中而不是LEC中。CAN FD CRC序列中的固定填充位的错误将显示为格式错误 (Form Error) 而不是填充错误 (Stuff Error)。 2.Bus_Off 恢复序列 (请参考 ISO11898-1:2015) 不能通过置位或清零MCAN_CCCR.INIT来缩短。如果设备进入 Bus_Off 状态, 它将自动置位MCAN_CCCR.INIT, 停止所有总线活动。一旦MCAN_CCCR.INIT被CPU清零, 设备在恢复正常操作之前, 将等待129个总线空闲 (129*11个连续的隐性位)。在Bus_Off恢复序列结束时, 错误管理计数器将被复位。在清零MCAN_CCCR.INIT后的等待期间, 每次监测到一个11位隐性序列时, 就向MCAN_PSR.LEC写入一个Bit0错误码, 使CPU能够随时检查CAN总线是否处于显性状态或持续被干扰, 并监测Bus_Off恢复序列。MCAN_ECR.REC用于对这些序列进行计数。 3.字节操作时, 读取寄存器的字节0会将LEC设置为0b111, 读取字节3/2/1没有影响。	RS

30.4.13 MCAN 发送延迟补偿寄存器 (MCAN_TDCR)

MCAN Transmitter Delay Compensation Register

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b15	Res	保留位	读出时为0, 写入时请写0	R
b14~b8	TDCO[6:0]	发送延迟补偿SSP偏移	该偏移值是指, 从MCAN_TX引脚到MCAN_RX引脚测得的延迟与第二采样点(SSP)之间的距离。有效设定值为0~127mtq。	RWA
b7	Res	保留位	读出时为0, 写入时请写0	R
b6~b0	TDCF[6:0]	发送延迟补偿滤波窗口宽度	定义SSP位置的最小值, 对于发送延迟的测量, 将忽略MCAN_RX引脚上会导致SSP位置提前的显性边沿。当TDCF设定值大于TDCO时, 该功能被启用。有效设定值为0~127mtq。	RWA

30.4.14 MCAN 中断寄存器 (MCAN_IR)

MCAN Interrupt Register

复位值: 0x0000 0000

当如下列表中的一个条件被检测到时, 对应标志位会置位。在主机将标志位清零之前, 保持置位状态。所有标志位写 1 清零, 写 0 无影响。硬复位会将寄存器清零。是否产生中断, 由寄存器 MCAN_IE 控制; 哪条中断线产生中断, 由寄存器 MCAN_ILS 控制。

b31	b30	b29	b28	b27	b26	b25	b24
Res		ARA	PED	PEA	WDI	BO	EW
b23	b22	b21	b20	b19	b18	b17	b16
EP	ELO	BEU	BEC	DRX	TOO	MRAF	TSW
b15	b14	b13	b12	b11	b10	b9	b8
TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
b7	b6	b5	b4	b3	b2	b1	b0
RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N

位/位域	标记	位名	功能	读写
b31~b30	Res	保留位	读出时为0, 写入时请写0	R
b29	ARA	访问保留地址	访问保留地址 (Access to Reserved Address) 0: 未对保留地址进行访问 1: 已对保留地址进行访问	RW1C
b28	PED	数据阶段协议错误 (使用数据位时间)	数据阶段协议错误 (使用数据位时间) (Protocol Error in Data Phase (Data Bit Time is used)) 0: 未检测到数据阶段协议错误 1: 检测到数据阶段协议错误 (MCAN_PSR.DLEC≠0/7)	RW1C
b27	PEA	仲裁阶段协议错误 (使用标称位时间)	仲裁阶段协议错误 (使用标称位时间) (Protocol Error in Arbitration Phase (Nominal Bit Time is used)) 0: 未检测到仲裁阶段协议错误 1: 检测到仲裁阶段协议错误 (MCAN_PSR.LEC≠0/7)	RW1C
b26	WDI	看门狗中断	看门狗中断 (Watchdog Interrupt) 0: 未发生消息RAM看门狗事件 1: 因READY信号缺失而发生消息RAM看门狗事件	RW1C
b25	BO	Bus_Off状态	Bus_Off 状态 (Bus_Off Status) 0: Bus_Off状态未改变 1: Bus_Off状态已改变	RW1C
b24	EW	警告状态	警告状态 (Warning Status) 0: 错误警告 (Error_Warning) 状态未改变 1: 错误警告 (Error_Warning) 状态已改变	RW1C
b23	EP	错误被动	错误被动 (Error Passive) 0: 错误被动 (Error_Passive) 状态未改变 1: 错误被动 (Error_Passive) 状态已改变	RW1C
b22	ELO	错误记录溢出	错误记录溢出 (Error Logging Overflow) 0: CAN错误记录计数器未溢出 1: CAN错误记录计数器已溢出	RW1C

			位错误未纠正 (Bit Error Uncorrected)	
b21	BEU	未纠正的错误位检出标志	0: 从消息RAM读取时未检测到错误 1: 当MCANRAM_CKCR.ECCMOD不为0时, 从消息RAM读取时检测到位错误且未纠正 注: BEU置位的同时MCAN_CCCR.INIT置位	RW
b20	BEC	纠正的错误位检出标志	0: 从消息RAM读取时未检测到错误 1: 当MCANRAM_CKCR.ECCMOD不为0时, 从消息RAM读取时检测到位错误且已纠正	RW
b19	DRX	消息存储到专用接收缓冲区	消息存储到专用接收缓冲区 (Message stored to Dedicated Rx Buffer) 每当收到的消息已存储到专用接收缓冲区时, 都会置位该标志。 0: 接收缓存区未更新 1: 至少有一条消息已存储到接收缓冲区	RW1C
b18	TOO	发生超时	发生超时 (Timeout Occurred) 0: 未发生超时 1: 已发生超时	RW1C
b17	MRAF	消息RAM访问失败	消息RAM访问失败 (Message RAM Access Failure) 接收处理中置位条件: 1) 在接收到后续消息的仲裁场 (Arbitration field) 之前, 尚未完成对已接收消息的接收过滤或存储。在这种情况下, 接收过滤或消息存储被终止, 接收处理单元将开始处理后续消息。 2) 无法向消息RAM写入消息, 消息存储将终止。 在这两种情况下, FIFO放入索引都不会更新, 专用接收缓冲区的新数据标志不会置位, 当下一条消息存储到该位置时, 已部分存储的消息会被覆盖。 发送处理中置位条件: 1) 发送处理单元无法及时从消息RAM读取消息。在这种情况下, 消息发送将终止。 在发送处理单元访问失败的情况下, MCAN被切换到受限工作模式 (详见章节【受限工作模式】)。要退出受限工作模式, 主机CPU必须复位MCAN_CCCR.ASM。 0: 未发生消息RAM访问失败 1: 发生了消息RAM访问失败	RW1C
b16	TSW	时间戳回卷	时间戳回卷 (Timestamp Wraparound) 0: 时间戳计数器未回卷 1: 时间戳计数器已回卷	RW1C
b15	TEFL	发送事件FIFO元素丢失	发送事件FIFO元素丢失 (Tx Event FIFO Element Lost) 0: 发送事件FIFO元素未丢失 1: 发送事件FIFO元素有丢失; 对空间大小为0的发送事件FIFO进行写操作时, 该位也会置位	RW1C
b14	TEFF	发送事件FIFO已满	发送事件FIFO已满 (Tx Event FIFO Full) 0: 发送事件FIFO未滿 1: 发送事件FIFO已滿	RW1C
b13	TEFW	发送事件FIFO到达水位线	发送事件FIFO到达水位线 (Tx Event FIFO Watermark Reached) 0: 发送事件FIFO的填充级别低于水位线 1: 发送事件FIFO的填充级别达到水位线	RW1C
b12	TEFN	发送事件FIFO新条目	发送事件FIFO新条目 (Tx Event FIFO New Entry) 0: 发送事件FIFO无变化 1: 发送处理单元写入了发送事件FIFO元素	RW1C
b11	TFE	发送FIFO为空	发送FIFO为空 (Tx FIFO Empty) 0: 发送FIFO非空 1: 发送FIFO为空	RW1C

b10	TCF	发送取消已完成	发送取消已完成 (Transmission Cancellation Finished) 0: 发送取消未完成 1: 发送取消已完成	RW1C
b9	TC	发送完成	发送完成 (Transmission Completed) 0: 发送未完成 1: 发送已完成	RW1C
b8	HPM	高优先级消息	高优先级消息 (High Priority Message) 0: 未收到高优先级消息 1: 已收到高优先级消息	RW1C
b7	RF1L	接收FIFO1消息丢失	接收FIFO1消息丢失 (Rx FIFO 1 Message Lost) 0: 接收FIFO1消息未丢失 1: 接收FIFO1消息有丢失; 对空间大小为0的接收FIFO1进行写操作时, 该位也会置位	RW1C
b6	RF1F	接收FIFO1已满	接收FIFO1已满 (Rx FIFO 1 Full) 0: 接收FIFO1未满 1: 接收FIFO1已满	RW1C
b5	RF1W	接收FIFO1到达水位线	接收FIFO1到达水位线 (Rx FIFO 1 Watermark Reached) 0: 接收FIFO1的填充级别低于水位线 1: 接收FIFO1的填充级别达到水位线	RW1C
b4	RF1N	接收FIFO1新消息	接收FIFO1新消息 (Rx FIFO 1 New Message) 0: 没有新消息写入接收FIFO1 1: 有新消息写入接收FIFO1	RW1C
b3	RF0L	接收FIFO0消息丢失	接收FIFO0消息丢失 (Rx FIFO 0 Message Lost) 0: 接收FIFO0消息未丢失 1: 接收FIFO0消息有丢失; 对空间大小为0的接收FIFO0进行写操作时, 该位也会置位	RW1C
b2	RF0F	接收FIFO0已满	接收FIFO0已满 (Rx FIFO 0 Full) 0: 接收FIFO0未满 1: 接收FIFO0已满	RW1C
b1	RF0W	接收FIFO0到达水位线	接收FIFO0到达水位线 (Rx FIFO 0 Watermark Reached) 0: 接收FIFO0的填充级别低于水位线 1: 接收FIFO0的填充级别达到水位线	RW1C
b0	RF0N	接收FIFO0新消息	接收FIFO0新消息 (Rx FIFO 0 New Message) 0: 没有新消息写入接收FIFO0 1: 有新消息写入接收FIFO0	RW1C

30.4.15 MCAN 中断使能寄存器 (MCAN_IE)

MCAN Interrupt Enable Register

复位值: 0x0000 0000

中断使能寄存器中的设置, 决定中断寄存器 (MCAN_IR) 中的哪些状态变化会指示在中断线上。

b31	b30	b29	b28	b27	b26	b25	b24
Res		ARAE	PEDE	PEAE	WDIE	BOE	EWE
b23	b22	b21	b20	b19	b18	b17	b16
EPE	ELOE	BEUE	BECE	DRXE	TOOE	MRAFE	TSWE
b15	b14	b13	b12	b11	b10	b9	b8
TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
b7	b6	b5	b4	b3	b2	b1	b0
RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE

位/位域	标记	位名	功能	读写
b31~b30	Res	保留位	读出时为0, 写入时请写0	R
b29	ARAE	访问保留地址中断使能	访问保留地址使能 (Access to Reserved Address Enable) 0: 禁止中断 1: 使能中断	RW
b28	PEDE	数据阶段协议错误中断使能	数据阶段的协议错误使能 (Protocol Error in Data Phase Enable) 0: 禁止中断 1: 使能中断	RW
b27	PEAE	仲裁阶段协议错误中断使能	仲裁阶段中的协议错误使能 (Protocol Error in Arbitration Phase Enable) 0: 禁止中断 1: 使能中断	RW
b26	WDIE	看门狗中断使能	看门狗中断使能 (Watchdog Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b25	BOE	Bus_Off状态中断使能	Bus_Off状态中断 (Bus_Off Status Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b24	EWE	警告状态中断使能	警告状态中断使能 (Warning Status Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b23	EPE	错误被动中断使能	错误被动中断使能 (Error Passive Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b22	ELOE	错误记录溢出中断使能	错误记录溢出中断使能 (Error Logging Overflow Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b21	BEUE	位错误未纠正中断使能	位错误未纠正中断使能 (Bit Error Uncorrected Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b20	BECE	位错误已纠正中断使能	位错误已纠正中断使能 (Bit Error Corrected Interrupt Enable)	

			0: 禁止中断 1: 使能中断	
b19	DRXE	消息存储到专用接收缓冲区中断使能	消息存储到专用接收缓冲区中断使能 (Message stored to Dedicated Rx Buffer Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b18	TOOE	发生超时中断使能	发生超时中断使能 (Timeout Occurred Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b17	MRAFE	消息RAM访问失败中断使能	消息 RAM 访问失败中断使能 (Message RAM Access Failure Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b16	TSWE	时间戳回卷中断使能	时间戳回卷中断使能 (Timestamp Wraparound Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b15	TEFLE	发送事件FIFO元素丢失中断使能	发送事件FIFO元素丢失中断使能 (Tx Event FIFO Element Lost Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b14	TEFFE	发送事件FIFO已满中断使能	发送事件FIFO满中断使能 (Tx Event FIFO Full Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b13	TEFWE	发送事件FIFO到达水位线中断使能	发送事件FIFO到达水位线中断使能(Tx Event FIFO Watermark Reached Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b12	TEFNE	发送事件FIFO新条目中断使能	发送事件 FIFO 新条目中断使能 (Tx Event FIFO New Entry Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b11	TFEE	发送FIFO为空中断使能	发送FIFO空中断使能 (Tx FIFO Empty Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b10	TCFE	发送取消已完成中断使能	发送取消已完成中断使能 (Transmission Cancellation Finished Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b9	TCE	发送完成中断使能	发送完成中断使能 (Transmission Completed Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b8	HPME	高优先级消息中断使能	高优先级消息中断使能 (High Priority Message Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b7	RF1LE	接收FIFO1消息丢失中断使能	接收FIFO1消息丢失中断使能 (Rx FIFO 1 Message Lost Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b6	RF1FE	接收FIFO1已满中断使能	接收FIFO1满中断使能 (Rx FIFO 1 Full Interrupt Enable) 0: 禁止中断 1: 使能中断	RW

b5	RF1WE	接收FIFO1到达水位线中断使能	接收FIFO1到达水位线中断使能 (Rx FIFO 1 Watermark Reached Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b4	RF1NE	接收FIFO1新消息中断使能	接收FIFO1新消息中断使能 (Rx FIFO 1 New Message Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b3	RF0LE	接收FIFO0消息丢失中断使能	接收FIFO0消息丢失中断使能 (Rx FIFO 0 Message Lost Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b2	RF0FE	接收FIFO0已满中断使能	接收FIFO0满中断使能 (Rx FIFO 0 Full Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b1	RF0WE	接收FIFO0到达水位线中断使能	接收FIFO0到达水位线中断使能 (Rx FIFO 0 Watermark Reached Interrupt Enable) 0: 禁止中断 1: 使能中断	RW
b0	RF0NE	接收FIFO0新消息中断使能	接收FIFO0新消息中断使能 (Rx FIFO 0 New Message Interrupt Enable) 0: 禁止中断 1: 使能中断	RW

30.4.16 MCAN 中断线选择寄存器 (MCAN_ILS)

MCAN Interrupt Line Select Register

复位值: 0x0000 0000

中断线选择寄存器, 将中断寄存器 (MCAN_IR) 中特定中断标志生成的中断分配到两条中断线之一。要产生中断, 必须通过 MCAN_ILE.EINT0 和 MCAN_ILE.EINT1 使能相应的中断线。

b31	b30	b29	b28	b27	b26	b25	b24
Res		ARAL	PEDL	PEAL	WDIL	BOL	EWL
b23	b22	b21	b20	b19	b18	b17	b16
EPL	ELOL	Res		DRXL	TOOL	MRAFL	TSWL
b15	b14	b13	b12	b11	b10	b9	b8
TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
b7	b6	b5	b4	b3	b2	b1	b0
RF1LL	RF1FL	RF1WL	RF1NL	RFOLL	RF0FL	RF0WL	RF0NL

位/位域	标记	位名	功能	读写
b31~b30	Res	保留位	读出时为0, 写入时请写0	R
b29	ARAL	访问保留地址中断线	访问保留地址中断线 (Access to Reserved Address Line) 0: 选择中断线0 1: 选择中断线1	RW
b28	PEDL	数据阶段协议错误中断线	数据阶段协议错误中断线 (Protocol Error in Data Phase Line) 0: 选择中断线0 1: 选择中断线1	RW
b27	PEAL	仲裁阶段协议错误中断线	仲裁阶段协议错误中断线 (Protocol Error in Arbitration Phase Line) 0: 选择中断线0 1: 选择中断线1	RW
b26	WDIL	看门狗中断线	看门狗中断线 (Watchdog Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b25	BOL	Bus_Off状态中断线	Bus_Off状态中断线 (Bus_Off Status Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b24	EWL	警告状态中断线	警告状态中断线 (Warning Status Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b23	EPL	错误被动中断线	错误被动中断线 (Error Passive Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b22	ELOL	错误记录溢出中断线	错误记录溢出中断线 (Error Logging Overflow Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b21~b20	Res	保留位	读出时为0, 写入时请写0	RW
b19	DRXL	消息存储到专用接收缓冲区	消息存储到专用接收缓冲区中断线 (Message stored to Dedicated Rx Buffer Interrupt Line)	RW

		中断线	0: 选择中断线0 1: 选择中断线1	
b18	TOOL	发生超时中断线	发生超时中断线 (Timeout Occurred Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b17	MRAFL	消息RAM访问失败中断线	消息RAM访问失败中断线 (Message RAM Access Failure Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b16	TSWL	时间戳回卷中断线	时间戳回卷中断线 (Timestamp Wraparound Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b15	TEFLL	发送事件FIFO元素丢失中断线	发送事件FIFO元素丢失中断线 (Tx Event FIFO Element Lost Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b14	TEFFL	发送事件FIFO已满中断线	发送事件FIFO已满中断线 (Tx Event FIFO Full Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b13	TEFWL	发送事件FIFO到达水位线中断线	发送事件FIFO到达水位线中断线 (Tx Event FIFO Watermark Reached Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b12	TEFNL	发送事件FIFO新条目中断线	发送事件FIFO新条目中断线 (Tx Event FIFO New Entry Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b11	TFEL	发送FIFO为空中断线	发送FIFO为空中断线 (Tx FIFO Empty Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b10	TCFL	发送取消已完成中断线	发送取消已完成中断线 (Transmission Cancellation Finished Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b9	TCL	发送完成中断线	发送完成中断线 (Transmission Completed Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b8	HPML	高优先级消息中断线	高优先级消息中断线 (High Priority Message Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b7	RF1LL	接收FIFO1消息丢失中断线	接收FIFO1消息丢失中断线 (Rx FIFO 1 Message Lost Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b6	RF1FL	接收FIFO1已满中断线	接收FIFO1已满中断线 (Rx FIFO 1 Full Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b5	RF1WL	接收FIFO1到达水位线中断线	接收FIFO1到达水位线中断线 (Rx FIFO 1 Watermark Reached Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b4	RF1NL	接收FIFO1新消息中断线	接收FIFO1新消息中断线 (Rx FIFO 1 New Message Interrupt Line) 0: 选择中断线0	RW

			1: 选择中断线1	
b3	RF0LL	接收FIFO0消息丢失中断线	接收FIFO0消息丢失中断线 (Rx FIFO 0 Message Lost Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b2	RF0FL	接收FIFO0已满中断线	接收FIFO0已满中断线 (Rx FIFO 0 Full Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b1	RF0WL	接收FIFO0到达水位线中断线	接收FIFO0到达水位线中断线 (Rx FIFO 0 Watermark Reached Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW
b0	RF0NL	接收FIFO0新消息中断线	接收FIFO0新消息中断线 (Rx FIFO 0 New Message Interrupt Line) 0: 选择中断线0 1: 选择中断线1	RW

30.4.17 MCAN 中断线使能寄存器 (MCAN_ILE)

MCAN Interrupt Line Enable Register

复位值: 0x0000 0000

连接到 CPU 的两条中断线中的每一条, 都可以通过设置 EINT0 和 EINT1 分别使能或禁用。

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res						EINT1	EINT0

位/位域	标记	位名	功能	读写
b31~b2	Res	保留位	读出时为0, 写入时请写0	R
b1	EINT1	中断线1使能	0: 中断线1禁止 1: 中断线1使能	RW
b0	EINT0	中断线0使能	0: 中断线0禁止 1: 中断线0使能	RW

30.4.18 MCAN 全局过滤器配置寄存器 (MCAN_GFC)

MCAN Global Filter Configuration Register

复位值: 0x0000 0000

用于消息 ID 过滤的全局设置。全局过滤器配置，控制标准消息和扩展消息的过滤路径，详见图 30-5 和图 30-6。

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

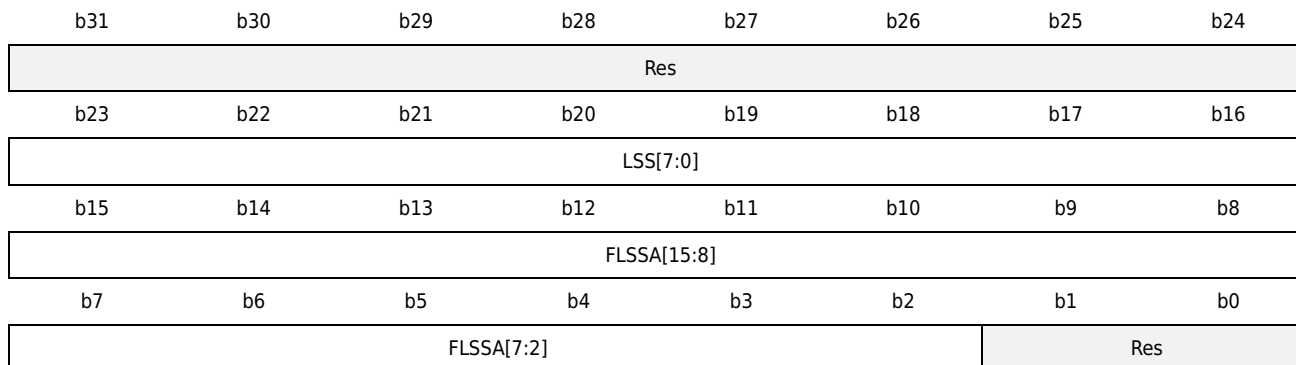
位/位域	标记	位名	功能	读写
b31~b6	Res	保留位	读出时为0，写入时请写0	R
b5~b4	ANFS[1:0]	接受不匹配的标准帧	定义如何处理，已接收的与过滤器列表任何元素都不匹配的11位标准ID消息。 00: 在接收FIFO0中接受不匹配的标准帧 01: 在接收FIFO1中接受不匹配的标准帧 10: 拒绝不匹配的标准帧 11: 拒绝不匹配的标准帧	RWA
b3~b2	ANFE[1:0]	接受不匹配的扩展帧	定义如何处理，已接收的与过滤器列表任何元素都不匹配的29位扩展ID消息。 00: 在接收FIFO0中接受不匹配的扩展帧 01: 在接收FIFO1中接受不匹配的扩展帧 10: 拒绝不匹配的扩展帧 11: 拒绝不匹配的扩展帧	RWA
b1	RRFS	拒绝远程标准帧	0: 过滤11位标准ID的远程帧 1: 拒绝所有11位标准ID的远程帧	RWA
b0	RRFE	拒绝远程扩展帧	0: 过滤29位扩展ID的远程帧 1: 拒绝所有29位扩展ID的远程帧	RWA

30.4.19 MCAN 标准 ID 过滤器配置寄存器 (MCAN_SIDFC)

MCAN Standard ID Filter Configuration Register

复位值: 0x0000 0000

用于 11 位标准 ID 过滤器的设置。标准 ID 过滤器配置，控制标准消息的过滤路径，见图 30-5。



位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为0，写入时请写0	R
b23~b16	LSS[7:0]	标准ID过滤器列表大小	0: 无标准ID过滤器 1~128: 标准ID过滤器元素数量 >128: 大于128的设定值被解析为128	RWA
b15~b2	FLSSA[15:2]	标准ID过滤器列表起始地址	设置标准ID过滤器列表的起始地址 (32位字地址，见图 30-11)	RWA
b1~b0	Res	保留位	读出时为0，写入时请写0	R

30.4.20 MCAN 扩展 ID 过滤器配置寄存器 (MCAN_XIDFC)

MCAN Extended ID Filter Configuration Register

复位值: 0x0000 0000

用于 29 位扩展 ID 过滤器的设置。扩展 ID 过滤器配置，控制扩展消息的过滤路径，见图 30-6。

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res	LSE[6:0]						
b15	b14	b13	b12	b11	b10	b9	b8
FLESA[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
FLESA[7:2]						Res	

位/位域	标记	位名	功能	读写
b31~b23	Res	保留位	读出时为0，写入时请写0	R
b22~b16	LES[7:0]	扩展ID过滤器列表大小	0: 无扩展ID过滤器 1~64: 扩展ID过滤器元素数量 >64: 大于64的设定值被解析为64	RWA
b15~b2	FLESA[15:2]	扩展ID过滤器列表起始地址	扩展ID过滤器列表的起始地址 (32位字地址，见图 30-11)	RWA
b1~b0	Res	保留位	读出时为0，写入时请写0	R

30.4.21 MCAN 扩展 ID 与掩码寄存器 (MCAN_XIDAM)

MCAN Extended ID AND Mask Register

复位值: 0x1FFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24
Res			EIDM[28:24]				
b23	b22	b21	b20	b19	b18	b17	b16
EIDM[23:16]							
b15	b14	b13	b12	b11	b10	b9	b8
EIDM[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
EIDM[7:0]							

位/位域	标记	位名	功能	读写
b31~b27	Res	保留位	读出时为0，写入时请写0	R
b28~b0	EIDM[28:0]	扩展ID掩码	对于扩展帧的接收过滤，会将EIDM与已接收帧的ID进行与运算。用于屏蔽SAE J1939中的29位ID。所有位的复位值都被设置为1，掩码无效。	RWA

30.4.22 MCAN 高优先级消息状态寄存器 (MCAN_HPMS)

MCAN High Priority Message Status Register

复位值: 0x0000 0000

配置为生成优先级事件的消息 ID 过滤器元素，每次匹配时都会更新此寄存器。这可用于监测收到的高优先级消息的状态，并可对这些消息进行快速访问。

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
FLST	FIDX[6:0]						
b7	b6	b5	b4	b3	b2	b1	b0
MSI[1:0]		BIDX[5:0]					

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为0，写入时请写0	R
b15	FLST	过滤器列表	指示匹配的过滤器列表。 0: 标准过滤器列表 1: 扩展过滤器列表	R
b14~b8	FIDX[6:0]	过滤器索引	匹配的过滤器元素的索引。范围为0到MCAN_SIDFC.LSS减1或MCAN_XIDFC.LSE减1。	R
b7~b6	MSI[1:0]	消息存储器指示符	00: 未选择FIFO 01: FIFO消息丢失 10: 消息存储在FIFO0中 11: 消息存储在FIFO1中	R
b5~b0	BIDX[5:0]	缓存区索引	存储消息的接收FIFO元素的索引。仅当MSI[1]=1时有效。	R

30.4.23 MCAN 新数据 1 寄存器 (MCAN_NDAT1)

MCAN New Data 1 Register

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
b23	b22	b21	b20	b19	b18	b17	b16
ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
b15	b14	b13	b12	b11	b10	b9	b8
ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
b7	b6	b5	b4	b3	b2	b1	b0
ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0

位/位域	标记	位名	功能	读写
b31~b0	ND31~0	新数据	<p>该寄存器保存接收缓冲区0到31的新数据标志。当接收帧更新接收缓冲区时，对应的标志位会置1。在主机将标志位清零之前，标志位保持置1状态。通过向对应位写1将标志位清零，写入0不起任何作用。硬复位会将寄存器清零。</p> <p>0: 接收缓冲区未更新 1: 接收缓冲区通过新消息更新</p>	RW

30.4.24 MCAN 新数据 2 寄存器 (MCAN_NDAT2)

MCAN New Data 2 Register

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56
b23	b22	b21	b20	b19	b18	b17	b16
ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
b15	b14	b13	b12	b11	b10	b9	b8
ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40
b7	b6	b5	b4	b3	b2	b1	b0
ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32

位/位域	标记	位名	功能	读写
b31~b0	ND63~32	新数据	<p>该寄存器保存接收缓冲区32到63的新数据标志。当接收帧更新接收缓冲区时，对应的标志位会置1。在主机将标志位清零之前，标志位保持置1状态。通过向对应位写1将标志位清零，写入0不起任何作用。硬复位会将寄存器清零。</p> <p>0: 接收缓冲区未更新 1: 接收缓冲区通过新消息更新</p>	RW

30.4.25 MCAN 接收 FIFO0 配置寄存器 (MCAN_RXF0C)

MCAN Rx FIFO 0 Configuration Register

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
F0OM	F0WM[6:0]						
b23	b22	b21	b20	b19	b18	b17	b16
Res	F0S[6:0]						
b15	b14	b13	b12	b11	b10	b9	b8
F0SA[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
F0SA[7:2]						Res	

位/位域	标记	位名	功能	读写
b31	F0OM	接收FIFO0工作模式	FIFO0可工作在阻塞模式 (Blocking mode) 或覆盖模式 (Overwrite mode) 0: 接收FIFO0为阻塞模式 1: 接收FIFO0为覆盖模式	RWA
b30~b24	F0WM[6:0]	接收FIFO0水位线	0: 禁止接收FIFO0水位线中断 1~64: 接收FIFO0水位线中断 (MCAN_IR.RF0W) 的级别 >64: 禁止接收FIFO0水位线中断	RWA
b23	Res	保留位	读出时为0, 写入时请写0	R
b22~b16	F0S[6:0]	接收FIFO0大小	0: 无接收FIFO0 1~64: 接收FIFO0元素数量 >64: 大于64的设定值被解析为64 接收FIFO0元素的索引为0到F0S减1	RWA
b15~b2	F0SA[15:2]	接收FIFO0起始地址	消息RAM中接收FIFO0的起始地址 (32位字地址, 见图 30-11)	RWA
b1~b0	Res	保留位	读出时为0, 写入时请写0	R

30.4.26 MCAN 接收 FIFO0 状态寄存器 (MCAN_RXF0S)

MCAN Rx FIFO 0 Status Register

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res						RF0L	F0F
b23	b22	b21	b20	b19	b18	b17	b16
Res		F0PI[5:0]					
b15	b14	b13	b12	b11	b10	b9	b8
Res		F0GI[5:0]					
b7	b6	b5	b4	b3	b2	b1	b0
Res	F0FL[6:0]						

位/位域	标记	位名	功能	读写
b31~b26	Res	保留位	读出时为0, 写入时请写0	R
b25	RF0L	接收FIFO0消息丢失	该位是中断标志MCAN_IR.RF0L的副本。当MCAN_IR.RF0L被复位时, 该位也被复位 0: 接收FIFO0无消息丢失 1: 接收FIFO0有消息丢失; 对空间大小为0的接收FIFO0进行写操作时, 该位也会置位 注: 当MCAN_RXF0C.F0OM=1时, 覆盖最早的消息, 不会置位该标志位。	R
b24	F0F	接收FIFO0满	0: 接收FIFO0未满 1: 接收FIFO0已满	R
b23~b22	Res	保留位	读出时为0, 写入时请写0	R
b15~b8	F0PI[5:0]	接收FIFO0放入索引	接收FIFO0写入索引指针, 范围0~63	R
b15~b14	Res	保留位	读出时为0, 写入时请写0	R
b13~b8	F0GI[5:0]	接收FIFO0获取索引	接收FIFO0读取索引指针, 范围0~63	R
b7	Res	保留位	读出时为0, 写入时请写0	R
b6~b0	F0FL[6:0]	接收FIFO0填充级别	接收FIFO0中存储的元素数量, 范围0~64	R

30.4.27 MCAN 接收 FIFO0 确认寄存器 (MCAN_RXF0A)

MCAN Rx FIFO 0 Acknowledge Register

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res		F0AI[5:0]					

位/位域	标记	位名	功能	读写
b31~b6	Res	保留位	读出时为0, 写入时请写0	R
b5~b0	F0AI[5:0]	接收FIFO0确认索引	主机从接收FIFO0读取消息或消息序列后, 必须将从接收FIFO0中读取的最后一个元素的缓冲区索引写入F0AI。此操作会将接收FIFO0获取索引MCAN_RXF0S.F0GI设为F0AI加1, 并会更新FIFO0填充级别MCAN_RXF0S.F0FL	RW

30.4.28 MCAN 接收缓冲区配置寄存器 (MCAN_RXBC)

MCAN Rx Buffer Configuration Register

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
RBSA[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
RBSA[7:2]						Res	

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为0, 写入时请写0	R
b15~b2	RBSA[15:2]	接收缓冲区的起始地址	配置消息RAM中接收缓冲区的起始地址 (32位地址), 也用于引用调试消息A、B、C	RWA
b1~b0	Res	保留位	读出时为0, 写入时请写0	R

30.4.29 MCAN 接收 FIFO1 配置寄存器 (MCAN_RXF1C)

MCAN Rx FIFO 1 Configuration Register

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
F1OM	F1WM[6:0]						
b23	b22	b21	b20	b19	b18	b17	b16
Res	F1S[6:0]						
b15	b14	b13	b12	b11	b10	b9	b8
F1SA[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
F1SA[7:2]						Res	

位/位域	标记	位名	功能	读写
b31	F1OM	接收FIFO1操作模式	FIFO1可工作在阻塞模式 (Blocking mode) 或覆盖模式 (Overwrite mode) 0: 接收FIFO1为阻塞模式 1: 接收FIFO1为覆盖模式	RWA
b30~b24	F1WM[6:0]	接收FIFO1水印	0: 禁止接收FIFO1水位线中断 1~64: 接收FIFO1水位线中断 (MCAN_IR.RF1W) 的级别 >64: 禁止接收FIFO1水位线中断	RWA
b23	Res	保留位	读出时为0, 写入时请写0	R
b22~b16	F1S[6:0]	接收FIFO1大小	0: 无接收FIFO1 1~64: 接收FIFO1元素数量 >64: 大于64的设定值被解析为64 接收FIFO1元素的索引为0到F1S减1	RWA
b15~b2	F1SA[15:2]	接收FIFO1起始地址	消息RAM中接收FIFO1的起始地址 (32位字地址, 见图 30-11)	RWA
b1~b0	Res	保留位	读出时为0, 写入时请写0	R

30.4.30 MCAN 接收 FIFO1 状态寄存器 (MCAN_RXF1S)

MCAN Rx FIFO 1 Status Register

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
DMS[1:0]		Res				RF1L	F1F
b23	b22	b21	b20	b19	b18	b17	b16
Res		F1PI[5:0]					
b15	b14	b13	b12	b11	b10	b9	b8
Res		F1GI[5:0]					
b7	b6	b5	b4	b3	b2	b1	b0
Res	F1FL[6:0]						

位/位域	标记	位名	功能	读写
b31~b30	DMS[1:0]	调试消息状态	00: 空闲状态, 等待接收调试消息 01: 已接收调试消息A 10: 已接收调试消息A、B 11: 已接收调试消息A、B、C	R
b29~b28	Res	保留位	读出时为0, 写入时请写0	R
b25	RF1L	接收FIFO1消息丢失	0: 接收FIFO1无元素丢失 1: 接收FIFO1有元素丢失; 对空间大小为0的接收FIFO1进行写操作时, 该位也会置位 注: 当MCAN_RXF1C.F1IOM=1时, 覆盖最早的消息, 不会置位该标志位。	R
b24	F1F	接收FIFO1满	0: 接收FIFO1未满 1: 接收FIFO1已满	R
b23~b22	Res	保留位	读出时为0, 写入时请写0	R
b15~b8	F1PI[5:0]	接收FIFO1放入索引	接收FIFO1写入索引指针, 范围0~63	R
b15~b14	Res	保留位	读出时为0, 写入时请写0	R
b13~b8	F1GI[5:0]	接收FIFO1获取索引	接收FIFO1读取索引指针, 范围0~63	R
b7	Res	保留位	读出时为0, 写入时请写0	R
b6~b0	F1FL[6:0]	接收FIFO1填充级别	接收FIFO1中存储的元素数量, 范围0~64	R

30.4.31 MCAN 接收 FIFO1 确认寄存器 (MCAN_RXF1A)

MCAN Rx FIFO 1 Acknowledge Register

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res		F1AI[5:0]					

位/位域	标记	位名	功能	读写
b31~b6	Res	保留位	读出时为0, 写入时请写0	R
b5~b0	F1AI[5:0]	接收FIFO1确认索引	主机从接收FIFO1读取消息或消息序列后, 必须将从接收FIFO1中读取的最后一个元素的缓冲区索引写入F1AI。此操作会将接收FIFO1获取索引MCAN_RXF1S.F1GI 设为F1AI加1, 并会更新FIFO1填充级别MCAN_RXF1S.F1FL	RW

30.4.32 MCAN 接收缓冲区和 FIFO 元素大小配置寄存器 (MCAN_RXESC)

MCAN Rx Buffer/ FIFO Element Size Configuration Register

复位值: 0x0000 0000

配置接收缓冲区和接收 FIFO 元素的数据字节数。大于 8 字节的数据场 (Data field) 仅适用于 CAN FD 模式。

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res					RBDS[2:0]		
b7	b6	b5	b4	b3	b2	b1	b0
Res	F1DS[2:0]			Res	F0DS[2:0]		

位/位域	标记	位名	功能	读写
b31~b11	Res	保留位	读出时为0, 写入时请写0	R
b10~b8	RBDS[2:0]	接收缓冲区数据场大小	000: 8字节 001: 12字节 010: 16字节 011: 20字节 100: 24字节 101: 32字节 110: 48字节 111: 64字节	RWA
b7	Res	保留位	读出时为0, 写入时请写0	R
b6~b4	F1DS[2:0]	接收FIFO1数据场大小	000: 8字节 001: 12字节 010: 16字节 011: 20字节 100: 24字节 101: 32字节 110: 48字节 111: 64字节	RWA
b3	Res	保留位	读出时为0, 写入时请写0	R
b2~b0	F0DS[2:0]	接收FIFO0数据场大小	000: 8字节 001: 12字节 010: 16字节 011: 20字节 100: 24字节 101: 32字节 110: 48字节 111: 64字节	RWA

30.4.33 MCAN 发送缓冲区配置寄存器 (MCAN_TXBC)

复位值: 0x0000 0000

仅当 MCAN_CCCR 寄存器的 CCE 位和 INIT 位均被置 1 时, 才能对该寄存器进行写访问。

b31	b30	b29	b28	b27	b26	b25	b24
Res	TFQM	TFQS[5:0]					
b23	b22	b21	b20	b19	b18	b17	b16
Res		NDTB[5:0]					
b15	b14	b13	b12	b11	b10	b9	b8
TBSA[13:6]							
b7	b6	b5	b4	b3	b2	b1	b0
TBSA[5:0]						Res	

位/位域	标记	位名	功能	读写
b31	Res	保留位	读出时为0, 写入时请写0	R
b30	TFQM	发送FIFO/队列模式	0: 发送FIFO模式 1: 发送队列模式	RW
b29~b24	TFQS[5:0]	发送FIFO/队列大小	0: 无发送FIFO或队列 1-32: 发送FIFO队列的缓冲区大小 >32: 大于32的设定值被解析为32	RW
b23~b22	Res	保留位	读出时为0, 写入时请写0	R
b21~b16	NDTB[5:0]	专用发送缓冲区大小	0: 无专用发送缓冲区 1-32: 专用发送缓冲区大小 >32: 大于32的设定值被解析为32	RW
b15~b2	TBSA[13:0]	发送缓冲区起始地址	消息RAM中发送缓冲区部分的起始地址 (32位字地址, 见图 30-11)	RW
b1~b0	Res	保留位	读出时为0, 写入时请写0	R

注意:

TFQS 和 NDTB 之和不得大于 32。MCAN 没有对该设置错误的监测。消息 RAM 中的发送缓冲区部分从专用发送缓冲区开始。

30.4.34 MCAN 发送 FIFO/队列状态寄存器 (MCAN_TXFQS)

MCAN Tx FIFO/Queue Status Register

复位值: 0x0000 0000

发送 FIFO 或队列的状态与寄存器 MCAN_TXBRP 中挂起的发送请求有关。因此，由于正在运行的发送扫描 (TXBRP 尚未更新)，添加或取消请求的效果可能会延迟。

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res		TFQF	TFQPI[4:0]				
b15	b14	b13	b12	b11	b10	b9	b8
Res			TFGI[4:0]				
b7	b6	b5	b4	b3	b2	b1	b0
Res		TFFL[5:0]					

位/位域	标记	位名	功能	读写
b31~b22	Res	保留位	读出时为0，写入时请写0	R
b21	TFQF	发送FIFO/队列满	0: 发送FIFO或队列未满 1: 发送FIFO或队列已满	R
b20~b16	TFQPI[4:0]	发送FIFO/队列放入索引	发送FIFO或队列的放入索引指针，范围0-31	R
b15~b13	Res	保留位	读出时为0，写入时请写0	R
b12~b8	TFGI[4:0]	发送FIFO获取索引	发送FIFO的读取索引指针，范围0-31。若已配置为发送队列模式 (MCAN_TXBC.TFQM=1)，则该位读取值为0	R
b7~b6	Res	保留位	读出时为0，写入时请写0	R
b5~b0	TFFL[5:0]	发送FIFO空闲级别	从TFGI开始的连续空闲的发送FIFO元素数量，范围0-32。若已配置为发送队列模式 (MCAN_TXBC.TFQM=1)，则该位读取值为0	R

注:

在专用发送缓冲区与发送 FIFO 或发送队列组合配置的情况下，放入索引和获取索引是指，从第一个专用发送缓冲区开始的发送缓冲区编号。

例如：对于有 12 个专用发送缓冲区和 20 个发送 FIFO 缓冲区的组合配置，放入索引 15 指向发送 FIFO 的第 4 个缓冲区。

30.4.35 MCAN 发送缓冲区元素大小配置寄存器 (MCAN_TXESC)

MCAN Tx Buffer Element Size Configuration Register

复位值: 0x0000 0000

配置发送缓冲区元素的数据字节数。大于 8 字节的数据场 (Data field) 仅适用于 CAN FD 模式。

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res					TBDS[2:0]		

位/位域	标记	位名	功能	读写
b31~b3	Res	保留位	读出时为0, 写入时请写0	R
b2~b1	TBDS[2:0]	发送缓冲区数据场大小	000: 8字节 001: 12字节 010: 16字节 011: 20字节 100: 24字节 101: 32字节 110: 48字节 111: 64字节	RW

注:

如果发送缓冲区元素的数据长度 (由 DLC 配置), 大于 TBDS 配置的数据长度, 则发送缓冲区中未定义的数据字节将会以 0xCC (填充字节) 发送。

30.4.36 MCAN 发送缓冲区请求挂起寄存器 (MCAN_TXBRP)

MCAN Tx Buffer Request Pending Register

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
b23	b22	b21	b20	b19	b18	b17	b16
TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
b15	b14	b13	b12	b11	b10	b9	b8
TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
b7	b6	b5	b4	b3	b2	b1	b0
TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0

位/位域	标记	位名	功能	读写
b31~b0	TRP31~0	发送请求挂起标志	<p>每个发送缓冲区都有自己的发送请求挂起位。这些位通过寄存器MCAN_TXBAR置位；在请求的发送已完成，或请求的发送已通过寄存器MCAN_TXBCR取消后复位。</p> <p>只有MCAN_TXBC配置了的发送缓冲区对应的TXBRP位才会置位。TXBRP位置位后会启动发送扫描（见章节【发送处理】），以检查优先级最高的挂起的发送请求（即消息ID最小的发送缓冲区）。</p> <p>取消发送请求会复位相应的TXBRP位。如果请求取消时发送已经开始，则无论发送是否成功，在发送结束时TXBRP位都会被复位。TXBRP位复位后，对应的取消请求位也会立即复位。</p> <p>请求取消后，以下情况通过寄存器MCAN_TXBCF指示取消已完成：</p> <ul style="list-style-type: none"> • 发送成功及相应的MCAN_TXBTO位置位后 • 取消时发送尚未开始 • 发送因仲裁失败而中止 • 帧发送期间发生错误 <p>在DAR模式下，所有发送在失败后都会自动取消。所有失败的发送，对应的MCAN_TXBCF位都会置位。</p> <p>0: 无发送请求挂起 1: 发送请求挂起</p> <p>注： 在发送扫描期间置位的TXBRP位，在本次发送扫描期间被忽略，在此期间，如果请求取消这些发送，则发送被立即取消，对应的TXBRP位复位。</p>	R

30.4.37 MCAN 发送缓冲区添加请求寄存器 (MCAN_TXBAR)

MCAN Tx Buffer Add Request Register

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
b23	b22	b21	b20	b19	b18	b17	b16
AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
b15	b14	b13	b12	b11	b10	b9	b8
AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
b7	b6	b5	b4	b3	b2	b1	b0
AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0

位/位域	标记	位名	功能	读写
b31~b0	AR31~0	添加请求	<p>每个发送缓冲区都有自己的添加请求位，写1置位相应的添加请求位，写0无影响。主机可一次设置多个发送缓冲区的发送请求。只有MCAN_TXBC配置了的发送缓冲区对应的TXBAR位才会置位。没有运行发送扫描时，这些位立即复位，否则将保持置位到发送扫描完成。</p> <p>0: 未添加发送请求 1: 已添加发送请求</p> <p>注: 如果添加请求的发送缓冲区已有发送请求挂起 (MCAN_TXBRP寄存器相应位已置位)，该添加请求将被忽略。</p>	RW

30.4.38 MCAN 发送缓冲区取消请求寄存器 (MCAN_TXBCR)

MCAN Tx Buffer Cancellation Request Register

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
b23	b22	b21	b20	b19	b18	b17	b16
CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
b15	b14	b13	b12	b11	b10	b9	b8
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
b7	b6	b5	b4	b3	b2	b1	b0
CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

位/位域	标记	位名	功能	读写
b31~b0	CR31~0	取消请求	每个发送缓冲区都有自己的取消请求位，写1置位相应的取消请求位，写0无影响。主机可一次设置多个发送缓冲区的取消请求。只有MCAN_TXBC配置了的发送缓冲区对应的TXBCR位才会置位。这些位保持置位直到对应的MCAN_TXBRP位复位。 0: 无取消请求挂起 1: 有取消请求挂起	RW

30.4.39 MCAN 发送缓冲区发送已发生寄存器 (MCAN_TXBTO)

MCAN Tx Buffer Transmission Occurred Register

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
b23	b22	b21	b20	b19	b18	b17	b16
TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
b15	b14	b13	b12	b11	b10	b9	b8
TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
b7	b6	b5	b4	b3	b2	b1	b0
TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0

位/位域	标记	位名	功能	读写
b31~b0	TO31~0	发送已发生	每个发送缓冲区都有自己的发送已发生标志位。当发送缓冲区发送完成，MCAN_TXBRP相应位清零时，对应的发送已发生标志位置位。向MCAN_TXBAR相应位写1请求新的发送时，对应的发送已发生标志位复位。 0: 未进行发送 1: 已进行发送	R

30.4.40 MCAN 发送缓冲区取消已完成寄存器 (MCAN_TXBCF)

MCAN Tx Buffer Cancellation Finished Register

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
b23	b22	b21	b20	b19	b18	b17	b16
CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
b15	b14	b13	b12	b11	b10	b9	b8
CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
b7	b6	b5	b4	b3	b2	b1	b0
CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0

位/位域	标记	位名	功能	读写
b31~b0	CF31~0	取消已完成	每个发送缓冲区都有自己的取消已完成标志位。当通过MCAN_TXBCR请求的取消已完成，MCAN_TXBRP相应位清零时，对应的取消已完成标志位置位。如果MCAN_TXBRP相应位在取消时没有置位，取消已完成标志位会立即置位。向MCAN_TXBAR相应位写1请求新的发送时，对应的取消已完成标志位复位。 0: 无发送缓冲区取消 1: 发送缓冲区已取消	R

30.4.41 MCAN 发送缓冲区发送中断使能寄存器 (MCAN_TXBTIE)

MCAN Tx Buffer Transmission Interrupt Enable Register

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
b23	b22	b21	b20	b19	b18	b17	b16
TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
b15	b14	b13	b12	b11	b10	b9	b8
TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
b7	b6	b5	b4	b3	b2	b1	b0
TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0

位/位域	标记	位名	功能	读写
每个发送缓冲区都有自己的发送中断使能位。				
b31~b0	TIE31~0	发送中断使能	0: 禁止发送中断 1: 使能发送中断	RW

30.4.42 MCAN 发送缓冲区取消已完成中断使能寄存器 (MCAN_TXBCIE)

MCAN Tx Buffer Cancellation Finished Interrupt Enable Register

复位值: 0x0000 0000

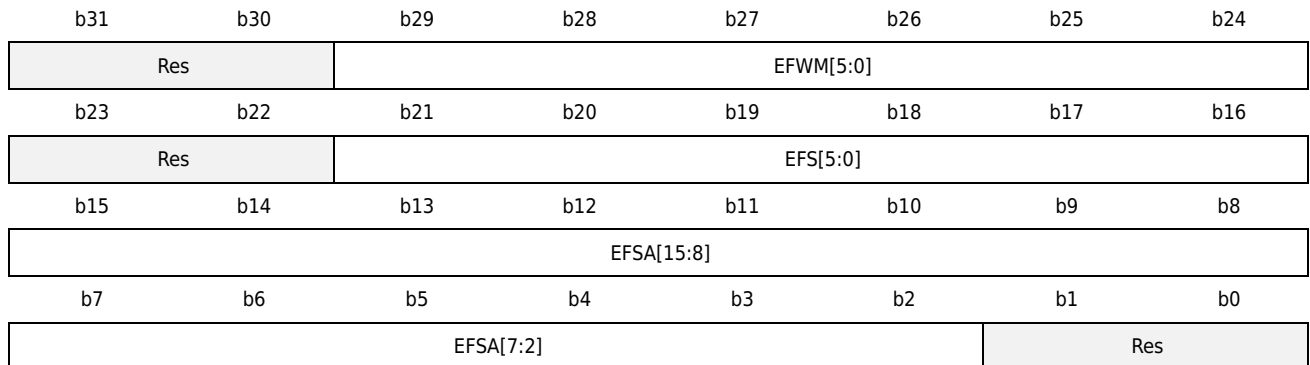
b31	b30	b29	b28	b27	b26	b25	b24
CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
b23	b22	b21	b20	b19	b18	b17	b16
CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
b15	b14	b13	b12	b11	b10	b9	b8
CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
b7	b6	b5	b4	b3	b2	b1	b0
CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0

位/位域	标记	位名	功能	读写
每个发送缓冲区都有自己的取消已完成中断使能位。				
b31~b0	CFIE31~0	取消已完成中断使能	0: 禁止取消已完成中断 1: 使能取消已完成中断	RW

30.4.43 MCAN 发送事件 FIFO 配置寄存器 (MCAN_TXEFC)

MCAN Tx Event FIFO Configuration Register

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b30	Res	保留位	读出时为0, 写入时请写0	R
b29~b24	EFWM[5:0]	事件FIFO水位线	0: 禁止水位线中断 1~32: 发送事件FIFO水位线中断 (MCAN_IR.TEFW) 的级别 > 32: 禁止水位线中断	RWA
b23~b22	Res	保留位	读出时为0, 写入时请写0	R
b21~b16	EFS[5:0]	事件FIFO大小	0: 禁止发送事件FIFO 1-32: 发送事件FIFO的元素数量 > 32: 大于32的设定值被解析为32 发送事件FIFO元素的索引为0到EFS减1	RWA
b15~b2	EFSA[15:2]	事件FIFO起始地址	消息RAM中事件FIFO的起始地址 (32位字地址, 见图 30-11)	RWA
b1~b0	Res	保留位	读出时为0, 写入时请写0	R

30.4.44 MCAN 发送事件 FIFO 状态寄存器 (MCAN_TXEFS)

MCAN Tx Event FIFO Status Register

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res						TEFL	EFF
b23	b22	b21	b20	b19	b18	b17	b16
Res			EFPI[4:0]				
b15	b14	b13	b12	b11	b10	b9	b8
Res			EFGI[4:0]				
b7	b6	b5	b4	b3	b2	b1	b0
Res		EFFL[5:0]					

位/位域	标记	位名	功能	读写
b31~b26	Res	保留位	读出时为0, 写入时请写0	R
b25	TEFL	发送事件FIFO元素丢失	该位是中断标志MCAN_IR. TEFL的副本。当MCAN_IR. TEFL被复位时, 该位也被复位 0: 发送事件FIFO无元素丢失 1: 发送事件FIFO有元素丢失; 对空间大小为0的发送事件FIFO进行写操作时, 该位也会置位	R
b24	EFF	事件FIFO已满	0: 发送事件FIFO未满 1: 发送事件FIFO已满	R
b23~b21	Res	保留位	读出时为0, 写入时请写0	R
b20~b16	EFPI[4:0]	事件FIFO放入索引	发送事件FIFO写入索引指针, 范围0~31	R
b15~b13	Res	保留位	读出时为0, 写入时请写0	R
b12~b8	EFGI[4:0]	事件FIFO获取索引	发送事件FIFO读取索引指针, 范围0~31	R
b7~b6	Res	保留位	读出时为0, 写入时请写0	R
b5~b0	EFFL[5:0]	事件FIFO填充级别	接收FIFO0中存储的元素个数, 范围0~32	R

30.4.45 MCAN 发送事件 FIFO 确认寄存器 (MCAN_TXEFA)

MCAN Tx Event FIFO Acknowledge Register

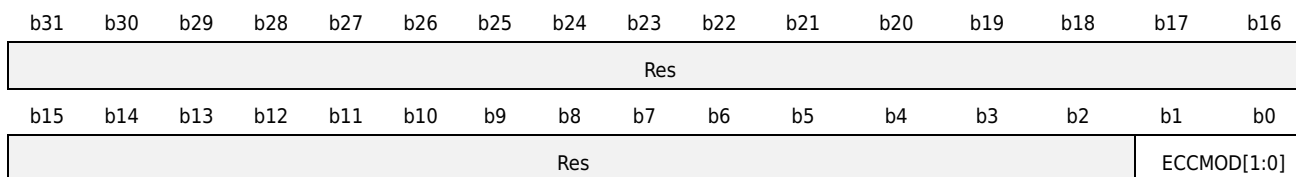
复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b5	Res	保留位	读出时为0, 写入时请写0	R
b4~b0	EFAI[4:0]	事件FIFO确认索引	主机从发送事件FIFO读取元素或元素序列后, 必须将从事件FIFO中读取的最后一个元素的索引写入EFAI。此操作会将发送事件FIFO获取索引MCAN_TXEFS.EFGI设为EFAI加 1, 并会更新事件FIFO填充级别MCAN_TXEFS.EFFL。	RW

30.4.46 MCAN 消息 RAM 校验控制寄存器 (MCANRAM_CKCR)

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1~b0	ECCMOD[1:0]	消息RAM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; 若2位错误, ECC检错, 产生2位错误标志, 产生中断. 10: 若1位错误, ECC纠错, 产生1位错误标志, 不产生中断; 若2位错误, ECC检错, 产生2位错误标志, 产生中断. 11: 若1位错误, ECC纠错, 产生1位错误标志, 产生中断; 若2位错误, ECC检错, 产生2位错误标志, 产生中断.	RW

注:

- 在 ECCMOD 设置不为 0 时, 当存取数据前, 必须先设置 ECCMOD, 然后对所用 RAM 空间以字为单位进行初始化。

30.4.47 MCAN 消息 RAM 校验状态寄存器 (MCANRAM_CKSR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														RAM2ERR	RAM1ERR

位/位域	标记	位名	功能	读写
b31~b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1	RAM_2ERR	消息RAM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	RW (注1)
b0	RAM_1ERR	消息RAM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	RW (注1)

注 1: 写 1 清 0。

30.4.48 MCAN 消息 RAM 错误注入使能寄存器 (MCANRAM_EIEN)

复位值: 0x0000 0000

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

位/位域	标记	位名	功能	读写
b31~b1	Res	保留位	读出时为“0”,写入时写“0”	RW
b0	EIEN	错误注入使能	0: 错误注入禁止 1: 错误注入有效	RW

30.4.49 MCAN 消息 RAM 错误注入位使能寄存器 0 (MCANRA_EIBIT0)

复位值: 0x0000 0000

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]															

位/位域	标记	位名	功能	读写
b31~b0	EIBIT[31:0]	错误注入位使能	控制消息RAM输出数据的0~31位是否取反 0: 不取反 1: 取反	RW

30.4.50 MCAN 消息 RAM 错误注入位使能寄存器 1 (MCANRAM_EIBIT1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res									EIBIT[38:32]						

位/位域	标记	位名	功能	读写
b31~b7	Res	保留位	读出时为“0”,写入时写“0”	RW
b6~b0	EIBIT[38:32]	错误注入位使能	控制消息RAM输出数据的32~38位是否取反 0: 不取反 1: 取反	RW

30.4.51 MCAN 消息 RAM 校验错误地址寄存器 (MCANRAM_ECCERRADDR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						ECCERRADDR [9:0]									

位/位域	标记	位名	功能	读写
b31~b10	Res	保留位	读出时为“0”,写入时写“0”	RW
b9~b0	ECCERRADDR [8:0]	校验错误地址	记录消息RAM发生1bit或2bit ECC错误时的访问地址	R

30.5 注意事项

30.5.1 CAN 使用注意事项

在使用任意一个 CAN 之前，需要把两路 CAN 的时钟都打开。

30.5.2 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，或者导致 CAN 协议未定义的错误，总线异常等不可预期的情况，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（包括但不限于）

- 1) 增加 CAN 总线接口电气隔离
- 2) 共收发器的信号地
- 3) 使用屏蔽双绞线缆并正确接地
- 4) 提高 CAN 传输线双绞程度
- 5) 增加信号保护器
- 6) 改进网络拓扑
- 7) 应用层软件抗干扰机制

30.5.3 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 MCAN 控制器异常动作。

31 CRC 运算 (CRC)

31.1 主要特性

- 一种执行标准：ISO/ IEC13239
- 两种编码方式：CRC-16、CRC-32
- 三种写入位宽：8bit、16bit、32bit
- 两种工作模式：CRC 编码模式、CRC 校验模式
- CRC-16 多项式： $x^{16}+x^{12}+x^5+1$ ，16 位初值为 0xFFFF
- CRC-32 多项式： $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$ ，32 位初值为 0xFFFF FFFF

31.2 功能说明

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 0x3322 1100、0x7766 5544；

31.2.1 功能框图

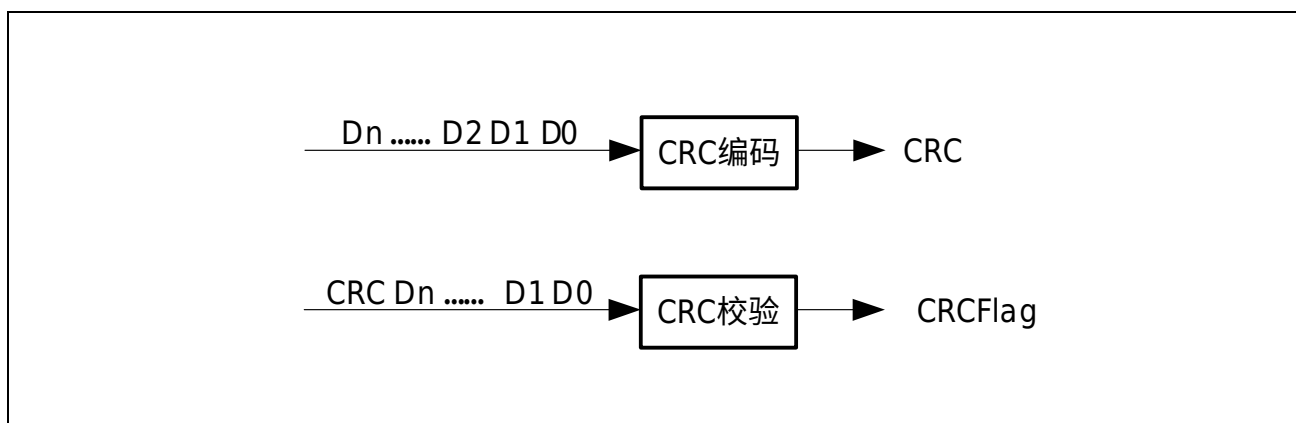


图 31-1 CRC 应用示意图

31.2.2 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 0b0，选择 CRC16；
2. 向 CRC_RESULT[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_RESULT[15:0]，即可获得 16 位 CRC 编码值。

31.2.3 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 0b0，选择 CRC16；
2. 向 CRC_RESULT[15:0]写入 0xFFFF，初始化 CRC 计算；
3. 将待校验的数据按 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 8/16 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。

31.2.4 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 0b1，选择 CRC32；
2. 向 CRC_RESULT[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_RESULT[31:0]，即可获得 32 位 CRC 编码值。

31.2.5 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 0b1，选择 CRC32；
2. 向 CRC_RESULT[31:0]写入 0xFFFF FFFF，初始化 CRC 计算；

3. 将待校验的数据按 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。

31.3 寄存器描述

表 31-1 CRC 基地址

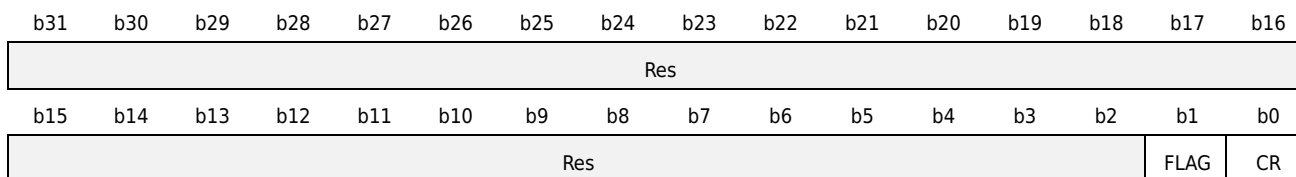
名称	基地址	描述
CRC	0x4000 8C00	CRC基地址

表 31-2 CRC 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
CRC_CR	CRC控制寄存器	0x00	32	0x0000 0001
CRC_RESLT	CRC结果寄存器	0x04	32	0x0000 0000
CRC_DAT	CRC数据寄存器	0x80~0xFF	32	0x0000 0000

31.3.1 控制寄存器 (CRC_CR)

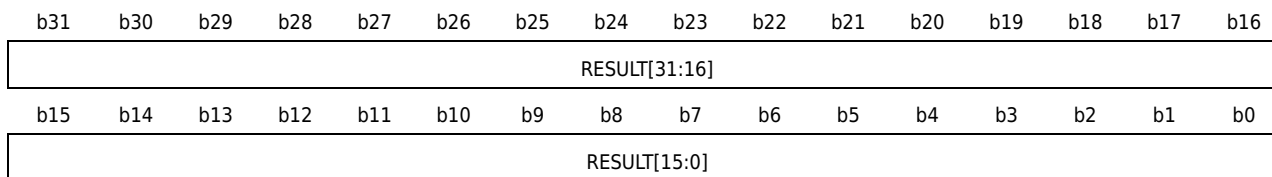
复位值: 0x0000 0001



位/位域	标记	位名	功能	读写
b30~b2	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW

31.3.2 结果寄存器 (CRC_RESULT)

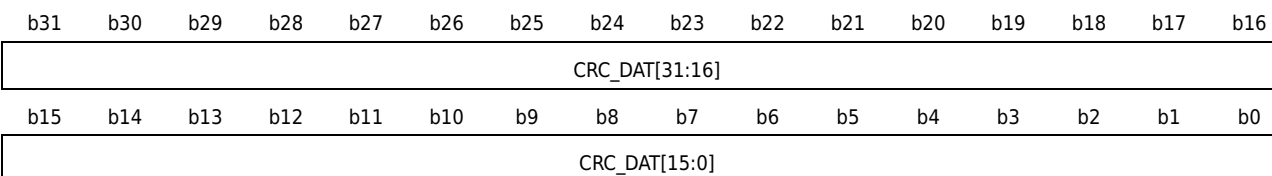
复位值: 0x0000 0000



位/位域	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];	RW

31.3.3 数据寄存器 (CRC_DAT)

复位值: 0x0000 0000



位/位域	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。	W

32 调试控制器 (DBGC)

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

32.1 概述

本 MCU 的内核是 Cortex-M4，该内核包含用于高级调试功能的硬件。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

32.2 DBGC 系统框图

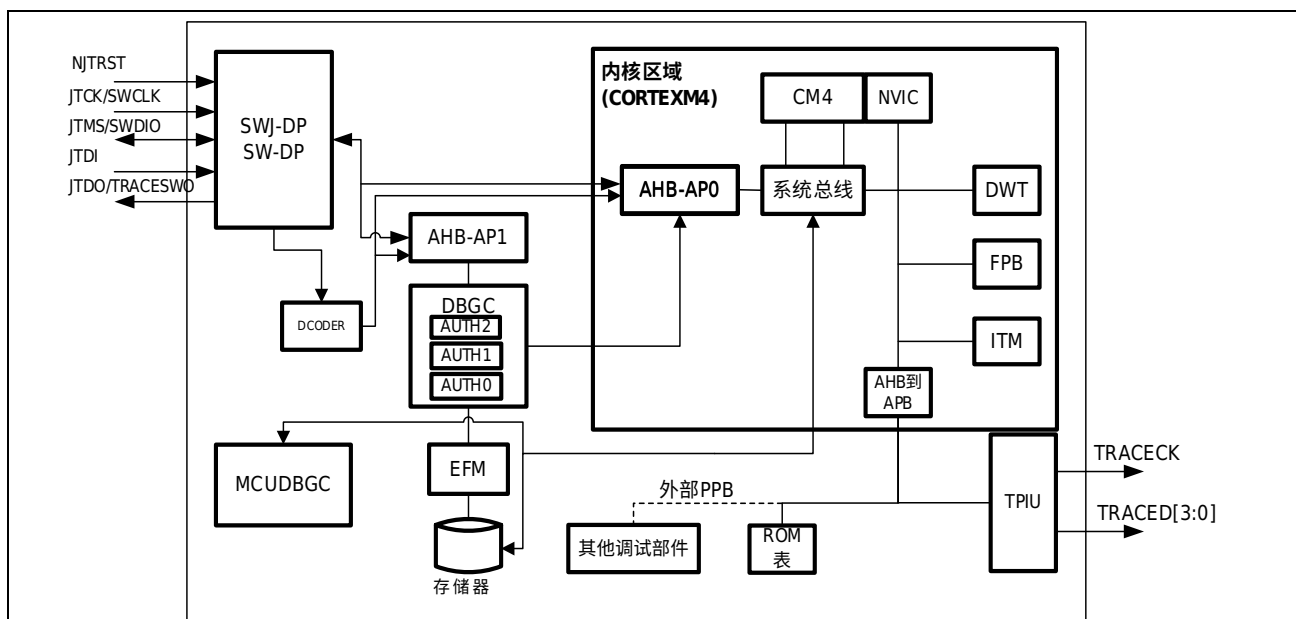


图 32-1 调试控制系统

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
- 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》。

32.3 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。

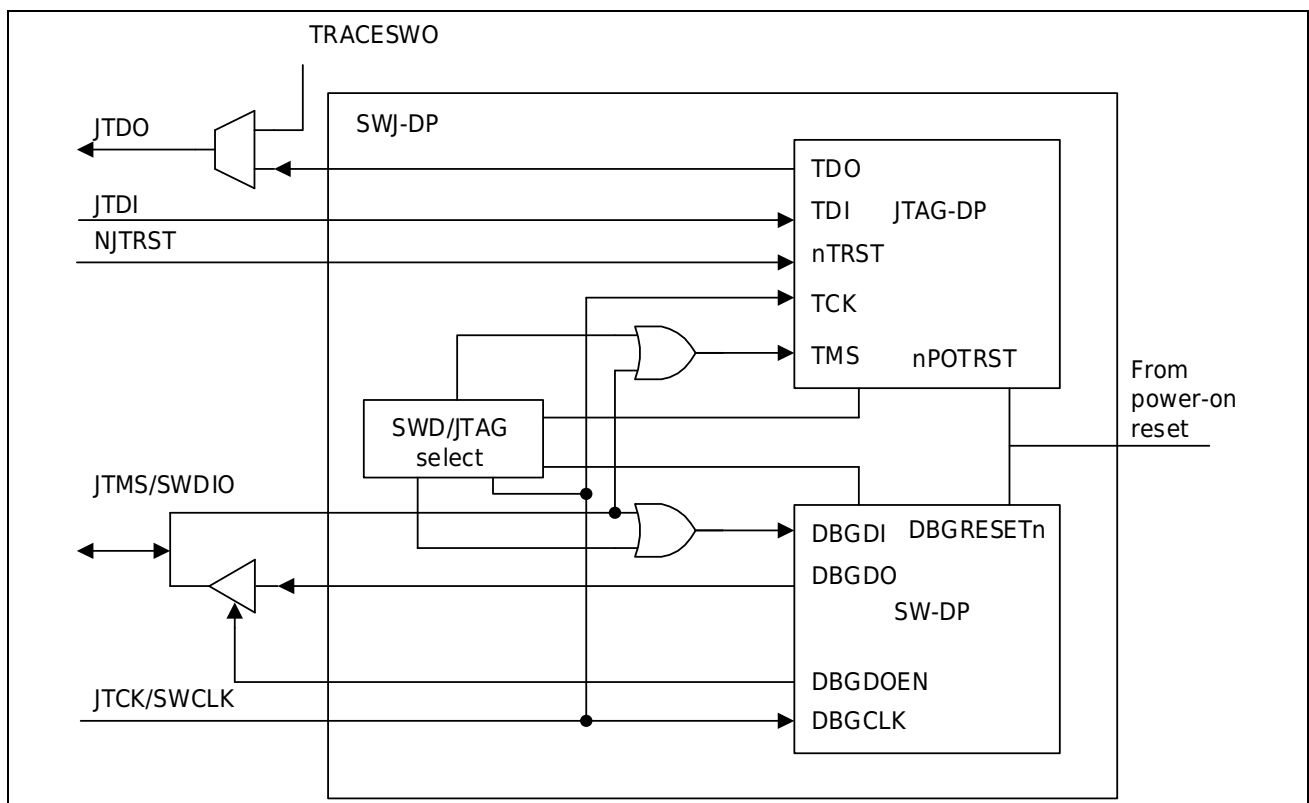


图 32-2 调试控制系统

32.3.1 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

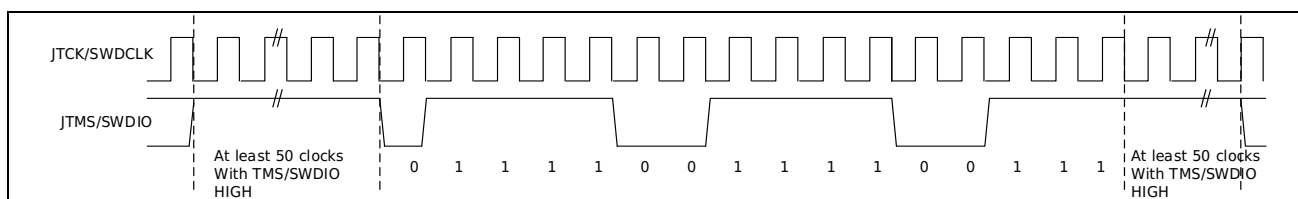


图 32-3 JTAG-DP 到 SW-DP 切换时序

32.4 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。

32.4.1 SWJ 调试端口引脚

MCU 的 5 个普通 I/O 口可用作 SWJ-DP 接口引脚。

表 32-1 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	输入	JTAG模式选择	输入输出	串行线数据输入/输出
JTCK/SWCLK	输入	JTAG时钟	输入	串行时钟
JTDI	输入	JTAG数据输入	-	-
JTDO/TRACESWO	输出	JTAG数据输出	-	TRACESWO (如果使能异步跟踪)
NJTRST	输入	JTAG复位	-	-

32.4.2 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 (GPIO)】章节特殊控制寄存器 PSPCR。

表 32-2 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JTAG-DP和禁止SW-DP	可释放	可释放	可释放	可释放	可释放

32.4.3 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：内部上拉

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。

32.4.4 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、JTCK 和 JTDO）。

32.5 寄存器说明

表 32-3 DBGC 基地址

名称	基地址	描述
DBGC	0xE004 2000	DBGC 基地址

表 32-4 DBGC 寄存器列表

寄存器	描述	偏移地址	位宽	复位值	访问主机
CHIPID	芯片专属标志寄存器	0x000C	32	0x5848 0334	调试IDE
MCUDBGSTAT	DBG状态寄存器	0x001C	32	0x0000 0000	CPU/调试IDE*
MCUSTPCTL	外设调试暂停寄存器	0x0020	32	0x0000 003B	CPU/调试IDE*
MCUTRACECTL	调试组件配置寄存器	0x0024	32	0x0000 0000	CPU/调试IDE*
MCUSTPCTL2	外设调试暂停寄存器2	0x0028	32	0x0000 0000	CPU/调试IDE*

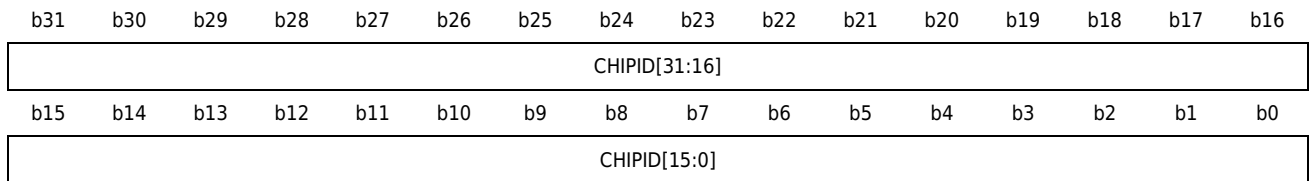
注意：

- 寄存器位于 *PPB* 区域，只能在特权模式下 *CPU* 才能进行访问。

32.5.1 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: 0x5848 0334

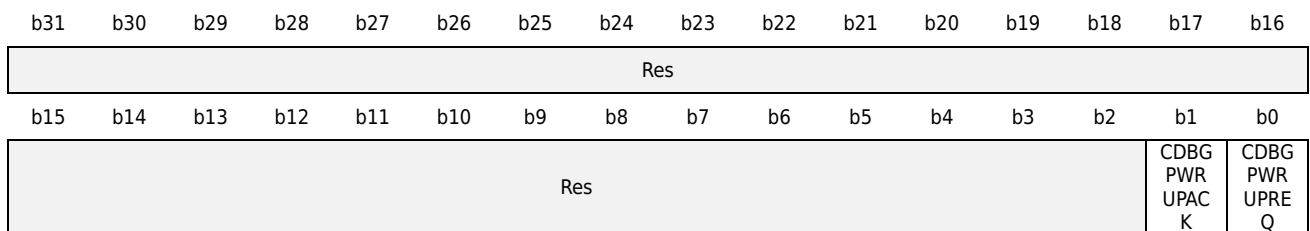


位/位域	标记	位名	功能	读写
b31~b0	CHIPID[31:0]	芯片专属标志	CHIPID[31:16]:XH的对应ASCII码, 0x5848 CHIPID[15:0]:产品型号, 0x0334	R

32.5.2 DBG 状态寄存器 (MCUDBGSTAT)

DBG 调试上电状态确认寄存器。

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b2	Res	保留位	读出时为“0”，写入时写“0”	RW
b1	CDBGPWRUPACK	调试器上电反馈	0: 无反馈 1: 调试上电反馈	RW
b0	CDBGPWRUPREQ	调试器上电请求	0: 无上电请求 1: 上电请求	RW

32.5.3 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
Res							M24S TP	M23S TP	M22S TP	M21S TP	M20S TP	M19S TP	M18S TP	M17S TP	M16S TP
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
M15S TP	Res	M12S TP	Res				M07S TP	M06S TP	Res			RTC STP	WDT STP	SWD TSTP	

位/位域	标记	位名	功能	读写
b31~b25	Res	保留位	读出时为“0”，写入时写“0”	RW
b24	M24STP	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	RW
b23	M23STP	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	RW
b22	M22STP	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	RW
b21	M21STP	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	RW
b20	M20STP	HRPWM_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	RW
b19	M19STP	HRPWM_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	RW
b18	M18STP	HRPWM_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	RW
b17	M17STP	HRPWM_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	RW
b16	M16STP	HRPWM_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	RW
b15	M15STP	HRPWM_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	RW
b14~b13	Res	保留位	读出时为“0”，写入时写“0”	RW
b12	M12STP	TMR4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	RW
b11~b8	Res	保留位	读出时为“0”，写入时写“0”	RW
b7	M07STP	TMR0_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	RW
b6	M06STP	TMR0_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	RW
b5~b3	Res	保留位	读出时所有位为“1”，写入时所有位写“1”	RW
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	RW
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	RW

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	RW
----	---------	------------	---	----

32.5.4 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
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													TRAC E_ IOE N	TRACE_ MODE	

位/位域	标记	位名	功能	读写
b31~b3	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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]	RW

32.5.5 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	
Res																
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Res												M36S TP	M35S TP	M34S TP	M33S TP	M32S TP

位/位域	标记	位名	功能	读写
b31~b15	Res	保留位	读出时为“0”，写入时写“0”	RW
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	RW
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	RW
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	RW
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	RW
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	RW

32.6 SW 调试端口

32.6.1 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 10K 欧）。

32.7 TPIU（跟踪端口接口单元）

32.7.1 简介

TPIU 是 ITM 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。

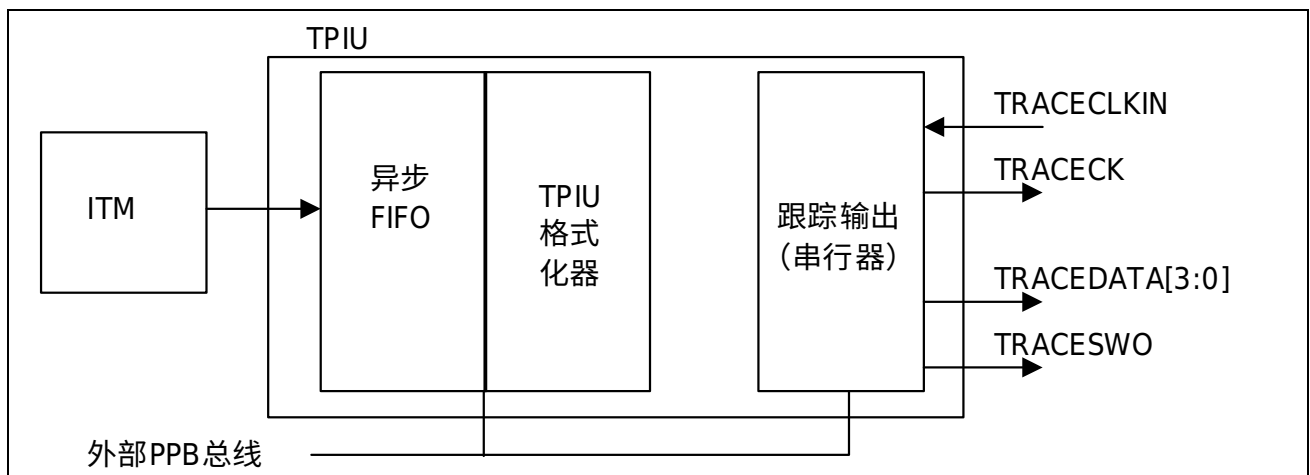


图 32-4 TPIU 框图

32.7.2 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。仅在使用串线行模式时异步模式才可用（在 JTAG 模式下不可用）。

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。

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。

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。

32.7.3 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。

32.7.4 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）。

32.7.5 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
Rev1.00	2024/06/12	初版发布。