

HC32A448 系列

32 位 ARM[®] Cortex[®]-M4 微控制器

参考手册

Rev1.31
2026 年 2 月

说明事项

版权所有©2026 小华半导体有限公司。保留所有权利

本文件及附件包含的信息有关知识产权权益全部属于小华半导体有限公司（以下简称“XHSC”）；客户对本文件及附件包含的信息只享有内部使用权，未经 XHSC 书面允许，任何单位和个人不得擅自摘抄、复制、改动或以其他任何形式使用本文件的部分或全部内容，并不得以任何形式进行传播。

商标声明

 小华半导体有限公司  小华半导体和其他商标均为 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。

注意事项

- XHSC 保留随时更改、更正、增强、修改产品和/或本文档的权利，恕不另行通知。用户可在下单前获取最新相关信息。XHSC 产品依据购销基本合同中载明的销售条款和条件进行销售。
- 客户应针对您的应用选择合适的 XHSC 产品，并设计、验证和测试您的应用，以确保您的应用满足相应标准以及任何安全、安保或其它要求。客户应对此独自承担全部责任。
- XHSC 在此确认未以明示或暗示方式授予任何知识产权许可。
- XHSC 产品的转售，若其条款与此处规定不同，XHSC 对此类产品的任何保修承诺无效。
- 本通知中的信息取代并替换先前版本中的信息。

小华半导体有限公司

地址：	上海市浦东新区中科路 1867 号 A 座 4 楼
网址：	https://www.xhsc.com.cn
邮箱：	XHSC_MCU@xhsc.com.cn （商务咨询） XHSC_FAE@xhsc.com.cn （技术咨询）

前言

位/位域

寄存器包含位（一个位宽）或者位域（多个位宽）的字段，位域通常描述成[y:x]，从 x 位开始到 y 位结束。没有特殊说明的情况下，位域字段使用的是十进制数。

不建议使用没有指定的字段，否则可能导致不可预知后果。

寄存器地址

每个寄存器的地址可以通过系统地址划分的基地址以及寄存器的偏移地址（详见各模块的寄存器总表）获得。

寄存器属性

寄存器说明中使用以下缩写词：

缩写词	说明
RW	读写，可以读写该位。
RO	只读，只能读取该位。
WO	只写，只能写入该位。
RWA	读写，可以读写该位。 仅解锁或者安全模式下可写。
RWC	读写，可以读写该位。 写入任意值将该位清零。
RC	只读，只能读取该位。 读取后该位自动清零。
RS	只读，只能读取该位。 读取后该位自动置位。

数据格式

- 0x 前缀表示十六进制数据
- 0b 前缀表示二进制数据
- 数字没有前缀表示十进制数据

Reserved (Res.)

标记为 Reserved 的寄存器和位/位域是为了将来使用预留，必须保持复位值（寄存器另有说明的场景除外）。Reserved 位的读取值在不同设备或者特性中可能不同。

复位值 (Reset)

复位值表示寄存器重置之后的值。寄存器在使用前需要进行初始化重置，否则可能导致不可预知后果。

外设可用性说明

HC32A448 系列 32 位 ARM® Cortex®-M4 微控制器所支持的外设以及其数目，请参见相应的《数据手册》。

安全声明

由于使用某个功能或者协议，可能会存在潜在的安全问题，需要进行声明，提醒用户慎用，规避安全风险。

目录

前言.....	iii
1 产品简介.....	37
2 存储器映射 (Memory Mapping)	38
2.1 存储器映射.....	38
2.2 外部空间映射.....	42
2.3 位段空间.....	43
2.4 地址重映射.....	43
2.5 寄存器.....	44
2.5.1 寄存器总表.....	44
2.5.2 访问保护寄存器 (MMF_REMPRT)	45
2.5.3 重映射寄存器 x (MMF_REMCrx) (x=0~1)	46
3 总线架构 (BUS)	47
3.1 简介.....	47
3.2 总线架构.....	48
3.3 总线功能.....	49
4 复位控制 (RMU)	50
4.1 简介.....	50
4.2 功能说明.....	50
4.2.1 复位方式和复位标志位.....	50
4.2.2 复位时序.....	52
4.2.3 复位方式判断.....	57
4.2.4 模块功能复位.....	57
4.2.5 复位条件.....	58
4.3 寄存器.....	60
4.3.1 寄存器总表.....	60
4.3.2 功能复位控制寄存器 0 (RMU_FRST0)	61
4.3.3 功能复位控制寄存器 1 (RMU_FRST1)	62
4.3.4 功能复位控制寄存器 2 (RMU_FRST2)	63
4.3.5 功能复位控制寄存器 3 (RMU_FRST3)	64
4.3.6 复位控制寄存器 0 (RMU_PRSTCR0)	65
4.3.7 复位标志寄存器 0 (RMU_RSTF0)	66
5 时钟控制器 (CMU)	69
5.1 简介.....	69
5.2 主要特性.....	69
5.3 功能说明.....	71
5.3.1 系统框图.....	71
5.3.2 工作时钟规格.....	72
5.3.3 晶振电路.....	73
5.3.4 内部 RC 时钟.....	76
5.3.5 PLL 时钟.....	77
5.3.6 时钟切换步骤.....	78
5.3.7 时钟输出功能.....	79

5.3.8 时钟频率测量.....	80
5.4 寄存器.....	82
5.4.1 寄存器总表.....	82
5.4.2 CMU XTAL 配置寄存器 (CMU_XTALCFGR)	84
5.4.3 CMU XTAL 稳定配置寄存器 (CMU_XTALSTBCR)	85
5.4.4 CMU XTAL 小数分频配置寄存器 (CMU_XTALDIVR)	86
5.4.5 CMU XTAL 小数分频控制寄存器 (CMU_XTALDIVCR)	86
5.4.6 CMU XTAL 控制寄存器 (CMU_XTALCR)	87
5.4.7 CMU XTAL 振荡故障控制寄存器 (CMU_XTALSTDCR)	88
5.4.8 CMU XTAL 振荡故障状态寄存器 (CMU_XTALSTDSR)	89
5.4.9 CMU XTAL32 配置寄存器 (CMU_XTAL32CFGR)	89
5.4.10 CMU XTAL32 滤波寄存器 (CMU_XTAL32NFR)	90
5.4.11 CMU XTAL32 控制寄存器 (CMU_XTAL32CR)	90
5.4.12 CMU HRC 校准寄存器 (CMU_HRCTRM)	91
5.4.13 CMU HRC 控制寄存器 (CMU_HRCCR)	92
5.4.14 CMU MRC 校准寄存器 (CMU_MRCTRM)	93
5.4.15 CMU MRC 控制寄存器 (CMU_MRCCR)	94
5.4.16 CMU LRC 校准寄存器 (CMU_LRCTRM)	95
5.4.17 CMU LRC 控制寄存器 (CMU_LRCCR)	96
5.4.18 CMU PLLH 配置寄存器 (CMU_PLLHCFGR)	97
5.4.19 CMU PLLH 控制寄存器 (CMU_PLLHCR)	99
5.4.20 CMU 时钟源稳定状态寄存器 (CMU_OSCSTBSR)	100
5.4.21 CMU 系统时钟源切换寄存器 (CMU_CKSWR)	101
5.4.22 CMU 时钟分频配置寄存器 (CMU_SCFGR)	102
5.4.23 CMU CAN 时钟配置寄存器 (CMU_CANCKCFGR)	105
5.4.24 CMU ADC/DAC 时钟配置寄存器 (CMU_PERICKSEL)	106
5.4.25 CMU 调试时钟配置寄存器 (CMU_TPIUCKCFGR)	107
5.4.26 CMU MCO1 时钟输出配置寄存器 (CMU_MCO1CFGR)	108
5.4.27 CMU MCO2 时钟输出配置寄存器 (CMU_MCO2CFGR)	109
5.4.28 FCM 下限比较值寄存器 (FCM_LVR)	110
5.4.29 FCM 上限比较值寄存器 (FCM_UVR)	110
5.4.30 FCM 计数器值寄存器 (FCM_CNTR)	111
5.4.31 FCM 开始停止寄存器 (FCM_STR)	111
5.4.32 FCM 测量对象控制寄存器 (FCM_MCCR)	112
5.4.33 FCM 测量基准控制寄存器 (FCM_RCCR)	113
5.4.34 FCM 中断复位控制寄存器 (FCM_RIER)	114
5.4.35 FCM 标志寄存器 (FCM_SR)	115
5.4.36 FCM 标志位清除寄存器 (FCM_CLR)	115
6 电源控制 (PWC)	116
6.1 简介.....	116
6.2 功能说明.....	116
6.2.1 电源分布.....	116
6.2.2 电源电压检测单元 (PVD) 说明.....	117
6.2.3 唤醒定时器.....	122
6.2.4 动作模式与低功耗模式.....	122

6.2.5 降低功耗的方法.....	129
6.3 寄存器.....	131
6.3.1 寄存器总表.....	131
6.3.2 电源模式控制寄存器 0 (PWC_PWRC0)	133
6.3.3 电源模式控制寄存器 1 (PWC_PWRC1)	134
6.3.4 电源模式控制寄存器 2 (PWC_PWRC2)	135
6.3.5 电源模式控制寄存器 3 (PWC_PWRC3)	135
6.3.6 电源模式控制寄存器 4 (PWC_PWRC4)	136
6.3.7 掉电唤醒使能寄存器 0 (PWC_PDWKE0)	137
6.3.8 掉电唤醒使能寄存器 1 (PWC_PDWKE1)	138
6.3.9 掉电唤醒使能寄存器 2 (PWC_PDWKE2)	139
6.3.10 掉电唤醒事件边沿选择寄存器 (PWC_PDWKES)	140
6.3.11 掉电唤醒标志寄存器 0 (PWC_PDWKF0)	141
6.3.12 掉电唤醒标志寄存器 1 (PWC_PDWKF1)	142
6.3.13 电源模式控制寄存器 5 (PWC_PWRC5)	143
6.3.14 电源模式控制寄存器 6 (PWC_PWRC6)	143
6.3.15 RAM 功耗控制寄存器 0 (PWC_RAMPC0)	144
6.3.16 RAM 运行条件寄存器 (PWC_RAMOPM)	145
6.3.17 外设 RAM 低功耗控制寄存器 (PWC_PRAMLPC)	146
6.3.18 PVD 控制寄存器 0 (PWC_PVDCR0)	147
6.3.19 PVD 控制寄存器 1 (PWC_PVDCR1)	148
6.3.20 PVD 滤波控制寄存器 (PWC_PVDFCR)	149
6.3.21 PVD 电平控制寄存器 (PWC_PVDLCR)	150
6.3.22 PVD 中断控制寄存器 (PWC_PVDICR)	151
6.3.23 PVD 检测状态寄存器 (PWC_PVDDSR)	152
6.3.24 功能时钟控制 0 (PWC_FCG0)	153
6.3.25 功能时钟控制 1 (PWC_FCG1)	155
6.3.26 功能时钟控制 2 (PWC_FCG2)	156
6.3.27 功能时钟控制 3 (PWC_FCG3)	157
6.3.28 PWC_FCG0 保护控制 (PWC_FCG0PC)	159
6.3.29 功能保护控制寄存器 (PWC_FPRC)	160
6.3.30 STOP 模式控制寄存器 (PWC_STPMCR)	161
6.3.31 唤醒计时器控制寄存器 (PWC_WKTCR)	162
7 初始化配置 (ICG)	163
7.1 简介.....	163
7.2 寄存器.....	163
7.2.1 寄存器总表.....	163
7.2.2 初始化配置寄存器 0 (ICG0)	164
7.2.3 初始化配置寄存器 1 (ICG1)	166
7.2.4 初始化配置寄存器 3 (ICG3)	166
8 嵌入式 Flash (EFM)	167
8.1 简介.....	167
8.2 主要特性.....	167
8.3 功能说明.....	168
8.3.1 嵌入式 Flash.....	168

8.3.2 读接口.....	168
8.3.3 Flash 读加速缓存.....	169
8.3.4 Flash 编程和擦除操作.....	170
8.3.5 一次性可编程字节 (OTP)	173
8.3.6 引导交换.....	174
8.3.7 中断.....	175
8.4 寄存器.....	176
8.4.1 寄存器总表.....	176
8.4.2 Flash 访问写保护寄存器 (EFM_FAPRT)	177
8.4.3 Flash 密钥 1 寄存器 (EFM_KEY1)	177
8.4.4 Flash 密钥 2 寄存器 (EFM_KEY2)	178
8.4.5 Flash 停止寄存器 (EFM_FSTP)	178
8.4.6 Flash 读模式寄存器 (EFM_FRMC)	179
8.4.7 Flash 擦写模式寄存器 (EFM_FWMC)	180
8.4.8 Flash 状态寄存器 (EFM_FSR)	181
8.4.9 Flash 状态清除寄存器 (EFM_FSCLR)	183
8.4.10 Flash 中断许可寄存器 (EFM_FITE)	184
8.4.11 Flash 引导交换状态寄存器 (EFM_FSWP)	185
8.4.12 芯片专属标志寄存器 (EFM_CHIPID)	186
8.4.13 UNIQUE ID 寄存器 (EFM_UQID0)	186
8.4.14 UNIQUE ID 寄存器 (EFM_UQID1)	186
8.4.15 UNIQUE ID 寄存器 (EFM_UQID2)	187
8.4.16 Flash 写保护锁定寄存器 (EFM_WLOCK)	188
8.4.17 Flash 写保护寄存器 (EFM_FONWPRT)	189
8.5 注意事项.....	190
9 内置 SRAM (SRAM)	191
9.1 简介.....	191
9.2 功能说明.....	191
9.2.1 SRAM 空间分配.....	191
9.2.2 ECC 校验错误注入.....	191
9.3 寄存器.....	193
9.3.1 寄存器总表.....	193
9.3.2 SRAM 校验控制寄存器 (SRAMC_CKCR)	194
9.3.3 SRAM 校验控制保护寄存器 (SRAMC_CKPR)	196
9.3.4 SRAM 校验状态寄存器 (SRAMC_CKSR)	197
9.3.5 SRAM0 错误注入使能寄存器 (SRAM0_EIEN)	198
9.3.6 SRAM0 错误注入位使能寄存器 0 (SRAM0_EIBIT0)	198
9.3.7 SRAM0 错误注入位使能寄存器 1 (SRAM0_EIBIT1)	199
9.3.8 SRAM0 错误地址寄存器 (SRAM0_ECCERRADDR)	199
9.3.9 Ret SRAM 错误注入使能寄存器 (SRAMB_EIEN)	200
9.3.10 Ret SRAM 错误注入位使能寄存器 0 (SRAMB_EIBIT0)	200
9.3.11 Ret SRAM 错误注入位使能寄存器 1 (SRAMB_EIBIT1)	201
9.3.12 Ret SRAM 错误地址寄存器 (SRAMB_ECCERRADDR)	201
10 通用 IO (GPIO)	202
10.1 简介.....	202

10.2 主要特性.....	202
10.3 功能说明.....	203
10.3.1 功能框图.....	203
10.3.2 通用输入输出 GPIO 功能.....	203
10.3.3 周边功能.....	204
10.3.4 双周边功能.....	204
10.3.5 Event Port 输入输出功能.....	204
10.3.6 外部中断 EIRQ 输入功能.....	204
10.3.7 模拟功能.....	204
10.3.8 通用控制.....	205
10.4 寄存器.....	206
10.4.1 寄存器总表.....	206
10.4.2 通用输入寄存器 (PIDRx)	208
10.4.3 通用输出数据寄存器 (PODRx)	208
10.4.4 通用输出许可寄存器 (POERx)	209
10.4.5 通用输出置位寄存器 (POSRx)	209
10.4.6 通用输出复位寄存器 (PORRx)	210
10.4.7 通用输出翻转寄存器 (POTRx)	210
10.4.8 特殊控制寄存器 (PSPCR)	211
10.4.9 公共控制寄存器 (PCCR)	212
10.4.10 写保护寄存器 (PWPR)	213
10.4.11 通用控制寄存器 (PCRxy)	214
10.4.12 功能选择寄存器 (PFSRxy)	216
10.4.13 Event Port 方向选择寄存器 (PEVNTDIRRm)	217
10.4.14 Event Port 输入数据寄存器 (PEVNTIDRm)	217
10.4.15 Event Port 输出数据寄存器 (PEVNTODRm)	218
10.4.16 Event Port 输出数据复位寄存器 (PEVNTORRm)	218
10.4.17 Event Port 输出数据置位寄存器 (PEVNTOSRm)	219
10.4.18 Event Port 上升沿输入许可寄存器 (PEVNTRISRm)	219
10.4.19 Event Port 下降沿输入许可寄存器 (PEVNTFALRm)	220
10.4.20 Event Port 输入滤波控制寄存器 (PEVNTNFCR)	221
10.5 注意事项.....	222
11 中断控制器 (INTC)	223
11.1 简介.....	223
11.2 向量表.....	224
11.2.1 中断向量表.....	224
11.2.2 中断事件表.....	227
11.3 功能说明.....	240
11.3.1 功能框图.....	240
11.3.2 不可屏蔽中断.....	240
11.3.3 外部中断.....	240
11.3.4 中断事件多路器.....	241
11.3.5 软件中断.....	241
11.3.6 事件输入.....	241
11.3.7 WFE 事件管理.....	241

11.3.8 噪声滤波器.....	242
11.3.9 低功耗模式返回.....	242
11.4 寄存器.....	243
11.4.1 寄存器总表.....	243
11.4.2 不可屏蔽中断使能寄存器 (INTC_NMIER)	244
11.4.3 不可屏蔽中断标志寄存器 (INTC_NMIFR)	245
11.4.4 不可屏蔽中断标志清除寄存器 (INTC_NMIFCR)	246
11.4.5 外部中断控制寄存器 (INTC_EIRQCRx) (x=0~15)	247
11.4.6 外部中断标志寄存器 (INTC_EIFR)	248
11.4.7 外部中断标志清除寄存器 (INTC_EIFCR)	248
11.4.8 中断事件选择寄存器 (INTC_INTSELx) (x=0~15)	249
11.4.9 中断事件使能寄存器 (INTC_INTENx) (x=0~15)	250
11.4.10 停止模式唤醒事件使能寄存器 (INTC_WKEN)	251
11.4.11 软件中断事件寄存器 (INTC_SWIER)	252
11.4.12 事件使能寄存器 (INTC_EVTER)	252
11.4.13 中断请求使能寄存器 (INTC_IER)	253
11.5 注意事项.....	253
12 自动运行系统 (AOS)	254
12.1 简介.....	254
12.2 主要特性.....	254
12.3 功能说明.....	255
12.3.1 功能框图.....	255
12.3.2 AOS 源事件和目标.....	255
12.3.3 专用触发源.....	257
12.3.4 公共触发源.....	257
12.3.5 可编程逻辑运算单元 (PLU)	257
12.4 寄存器.....	259
12.4.1 寄存器总表.....	259
12.4.2 软件触发事件寄存器 (INTSFTRG)	260
12.4.3 DCU 触发源选择寄存器 (DCU_TRGSELx) (x=1~4)	261
12.4.4 DMA1 传输启动触发源选择寄存器 (DMA1_TRGSELx) (x=0~5)	262
12.4.5 DMA2 传输启动触发源选择寄存器 (DMA2_TRGSELx) (x=0~5)	263
12.4.6 DMA 通道重置触发源选择寄存器 (DMA_RC_TRGSEL)	264
12.4.7 Timer6 硬件触发事件选择寄存器 (TMR6_TRGSELx) (x=0~1)	265
12.4.8 Timer4 硬件触发事件选择寄存器 (TMR4_TRGSELx) (x=0~2)	266
12.4.9 Event Port 触发源选择寄存器 (PEVNT_TRGSEL12/PEVNT_TRGSEL34)	267
12.4.10 Timer0 硬件触发事件选择寄存器 (TMR0_TRGSEL)	268
12.4.11 TimerA 硬件触发事件选择寄存器 (TMRA_TRGSELx) (x=0~3)	269
12.4.12 ADC1 转换开始片内触发源选择寄存器 (ADC1_TRGSELx) (x=0~1)	270
12.4.13 ADC2 转换开始片内触发源选择寄存器 (ADC2_TRGSELx) (x=0~1)	271
12.4.14 ADC3 转换开始片内触发源选择寄存器 (ADC3_TRGSELx) (x=0~1)	272
12.4.15 公共触发源选择寄存器 (AOS_COMTRGSELx) (x=1~2)	273
12.4.16 PLU 控制寄存器 (AOS_PLUX_CR) (x=0~3)	274
12.4.17 PLU 触发源选择寄存器 A (AOS_PLUX_TRGSELA) (x=0~3)	275
12.4.18 PLU 触发源选择寄存器 B (AOS_PLUX_TRGSELB) (x=0~3)	275

12.4.19 PLU 触发源选择寄存器 C (AOS_PLUX_TRGSELC) (x=0~3)	276
12.4.20 PLU 触发源选择寄存器 D (AOS_PLUX_TRGSELD) (x=0~3)	276
13 存储保护单元 (MPU)	277
13.1 简介	277
13.2 功能说明	278
13.2.1 区域范围设置	278
13.2.2 权限设置	278
13.2.3 MPU 动作选择	278
13.2.4 启动 MPU	278
13.3 操作示例	279
13.3.1 只允许部分空间访问	279
13.3.2 只禁止部分空间访问	279
13.4 寄存器	280
13.4.1 寄存器总表	280
13.4.2 区域范围描述寄存器 (MPU_RGDn) (n=0~15)	281
13.4.3 状态标志寄存器 (MPU_SR)	282
13.4.4 标志清除寄存器 (MPU_ECLR)	283
13.4.5 写保护寄存器 (MPU_WP)	283
13.4.6 IP 访问保护寄存器 (MPU_IPPR)	284
13.4.7 MSP 保护开始地址寄存器 (MPU_MSPPBA)	286
13.4.8 MSP 保护控制寄存器 (MPU_MSPPCTL)	286
13.4.9 PSP 保护开始地址寄存器 (MPU_PSPPBA)	287
13.4.10 PSP 保护控制寄存器 (MPU_PSPPCTL)	287
13.4.11 SMPU1 区域使能寄存器 (MPU_S1RGE)	288
13.4.12 SMPU1 区域写权限寄存器 (MPU_S1RGWP)	289
13.4.13 SMPU1 区域读权限寄存器 (MPU_S1RGRP)	290
13.4.14 SMPU1 控制寄存器 (MPU_S1CR)	291
13.4.15 SMPU2 区域使能寄存器 (MPU_S2RGE)	292
13.4.16 SMPU2 区域写权限寄存器 (MPU_S2RGWP)	293
13.4.17 SMPU2 区域读权限寄存器 (MPU_S2RGRP)	294
13.4.18 SMPU2 控制寄存器 (MPU_S2CR)	295
14 键盘扫描控制模块 (KEYSCAN)	296
14.1 简介	296
14.2 主要特性	296
14.3 功能说明	297
14.3.1 功能框图	297
14.3.2 按键识别功能	297
14.3.3 键盘扫描功能	298
14.4 寄存器	299
14.4.1 寄存器总表	299
14.4.2 扫描控制寄存器 (KEYSCAN_SCR)	300
14.4.3 扫描使能寄存器 (KEYSCAN_SER)	301
14.4.4 扫描状态寄存器 (KEYSCAN_SSR)	301
14.5 注意事项	302
15 内部时钟校准器 (CTC)	303

15.1 简介.....	303
15.2 主要特性.....	303
15.3 功能说明.....	303
15.3.1 功能框图.....	303
15.3.2 参考时钟.....	303
15.3.3 频率校准.....	308
15.3.4 中断.....	309
15.4 操作示例.....	309
15.4.1 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%.....	309
15.5 寄存器.....	311
15.5.1 寄存器总表.....	311
15.5.2 时钟校准控制寄存器 1 (CTC_CR1)	312
15.5.3 时钟校准控制寄存器 2 (CTC_CR2)	314
15.5.4 时钟校准状态寄存器 (CTC_STR)	315
15.5.5 时钟校准计数器 (CTC_CNT)	316
16 DMA 控制器 (DMA)	317
16.1 简介.....	317
16.2 主要特性.....	317
16.3 功能说明.....	318
16.3.1 DMA 框图.....	318
16.3.2 使能 DMA 控制器.....	318
16.3.3 通道选择和通道优先级.....	318
16.3.4 启动 DMA.....	318
16.3.5 数据块.....	319
16.3.6 传输地址控制.....	319
16.3.7 传输次数.....	319
16.3.8 连锁传输.....	319
16.3.9 不连续地址传输.....	320
16.3.10 通道重置.....	321
16.3.11 传输提前终止.....	323
16.3.12 中断.....	323
16.4 操作示例.....	324
16.4.1 存储器到存储器的传输.....	324
16.4.2 存储器到外围电路的传输.....	325
16.4.3 存储器到存储器的连锁传输.....	326
16.5 寄存器.....	328
16.5.1 寄存器总表.....	328
16.5.2 DMA 使能寄存器 (DMA_EN)	330
16.5.3 中断状态寄存器 0 (DMA_INTSTAT0)	330
16.5.4 中断状态寄存器 1 (DMA_INTSTAT1)	331
16.5.5 中断屏蔽寄存器 0 (DMA_INTMASK0)	332
16.5.6 中断屏蔽寄存器 1 (DMA_INTMASK1)	333
16.5.7 中断复位寄存器 0 (DMA_INTCLR0)	334
16.5.8 中断复位寄存器 1 (DMA_INTCLR1)	335
16.5.9 通道使能寄存器 (DMA_CHEN)	336

16.5.10 传输请求状态寄存器 (DMA_REQSTAT)	337
16.5.11 通道状态观测寄存器 (DMA_CHSTAT)	338
16.5.12 通道重置控制寄存器 (DMA_RCFGCTL)	339
16.5.13 软件触发寄存器 (DMA_SWREQ)	341
16.5.14 通道使能复位寄存器 (DMA_CHENCLR)	342
16.5.15 传输源地址寄存器 (DMA_SARx) (x=0~5)	342
16.5.16 传输目标地址寄存器 (DMA_DARx) (x=0~5)	343
16.5.17 数据控制寄存器 (DMA_DTCTLx) (x=0~5)	343
16.5.18 重复区域大小寄存器 (DMA_RPTx) (x=0~5)	344
16.5.19 重复区域大小寄存器 B (DMA_RPTBx) (x=0~5)	345
16.5.20 源地址不连续传输控制寄存器 (DMA_SNSEQCTLx) (x=0~5)	346
16.5.21 源地址不连续传输控制寄存器 B (DMA_SNSEQCTLBx) (x=0~5)	347
16.5.22 目标地址不连续传输控制寄存器 (DMA_DNSEQCTLx) (x=0~5)	348
16.5.23 目标地址不连续传输控制寄存器 B (DMA_DNSEQCTLBx) (x=0~5)	349
16.5.24 链指针寄存器 (DMA_LL Px) (x=0~5)	350
16.5.25 通道控制寄存器 (DMA_CHCTLx) (x=0~5)	350
16.5.26 传输源地址监视寄存器 (DMA_MONSARx) (x=0~5)	351
16.5.27 传输目标地址监视寄存器 (DMA_MONDARx) (x=0~5)	352
16.5.28 数据控制监视寄存器 (DMA_MONDTCTLx) (x=0~5)	353
16.5.29 重复区域计数器监视寄存器 (DMA_MONRPTx) (x=0~5)	354
16.5.30 源地址不连续传输计数器监视寄存器 (DMA_MONSNSEQCTLx) (x=0~5)	355
16.5.31 目标地址不连续传输计数器监视寄存器 (DMA_MONDNSEQCTLx) (x=0~5)	356
16.6 注意事项.....	356
17 电压比较器 (CMP)	357
17.1 简介.....	357
17.2 主要特性.....	357
17.3 功能说明.....	357
17.3.1 功能框图.....	357
17.3.2 输入选择.....	359
17.3.3 电压比较.....	359
17.3.4 窗口比较.....	360
17.3.5 输入扫描.....	360
17.3.6 窗口输出.....	362
17.3.7 噪声滤波器.....	363
17.3.8 外部管脚输出.....	363
17.3.9 中断和事件.....	363
17.4 寄存器.....	364
17.4.1 寄存器总表.....	364
17.4.2 比较器工作模式寄存器 (CMP_MDR)	365
17.4.3 比较器滤波和中断寄存器 (CMP_FIR)	366
17.4.4 比较器输出控制寄存器 (CMP_OCR)	367
17.4.5 比较器正负端输入选择寄存器 (CMP_PMSR)	368
17.4.6 比较器输出空白窗口设定寄存器 1 (CMP_BWSR1)	369
17.4.7 比较器输出空白窗口设定寄存器 2 (CMP_BWSR2)	370
17.4.8 比较器输入扫描控制寄存器 (CMP_SCCR)	371

17.4.9 比较器输入扫描监视寄存器 (CMP_SCMR)	372
17.5 注意事项.....	373
17.5.1 模块停止功能.....	373
17.5.2 模块停止时的功耗.....	373
17.5.3 停止低功耗模式时的功耗.....	373
18 模数转换模块 (ADC)	374
18.1 简介.....	374
18.2 主要特性.....	374
18.3 功能说明.....	375
18.3.1 功能框图.....	375
18.3.2 ADC 时钟.....	376
18.3.3 通道选择.....	376
18.3.4 触发源选择.....	377
18.3.5 序列 A 单次扫描模式.....	378
18.3.6 序列 A 连续扫描模式.....	378
18.3.7 序列 A 数据缓冲模式.....	379
18.3.8 双序列扫描模式.....	380
18.3.9 模拟看门狗功能.....	382
18.3.10 模拟输入的采样时间和转换时间.....	383
18.3.11 ADC 数据寄存器自动清除功能.....	383
18.3.12 转换数据平均计算功能.....	384
18.3.13 过采样.....	384
18.3.14 多 ADC 协同工作模式.....	384
18.3.15 中断和事件.....	388
18.4 寄存器.....	390
18.4.1 寄存器总表.....	390
18.4.2 启动寄存器 (ADC_STR)	392
18.4.3 控制寄存器 0 (ADC_CR0)	393
18.4.4 控制寄存器 1 (ADC_CR1)	395
18.4.5 控制寄存器 2 (ADC_CR2)	396
18.4.6 转换开始触发寄存器 (ADC_TRGSR)	397
18.4.7 通道选择寄存器 A (ADC_CHSELRA)	399
18.4.8 通道选择寄存器 B (ADC_CHSELRB)	400
18.4.9 平均通道选择寄存器 (ADC_AVCHSELR)	401
18.4.10 扩展通道选择寄存器 (ADC_EXCHSELR)	402
18.4.11 采样周期寄存器 (ADC_SSTRx)	403
18.4.12 通道映射控制寄存器 (ADC_CHMUXRx) (x=0~3)	404
18.4.13 中断状态寄存器 (ADC_ISR)	406
18.4.14 中断许可寄存器 (ADC_ICR)	407
18.4.15 中断状态复位寄存器 (ADC_ISCLRR)	407
18.4.16 协同模式控制寄存器 (ADC_SYNCCR)	408
18.4.17 数据寄存器 (ADC_DRx)	410
18.4.18 模拟看门狗控制寄存器 (ADC_AWDCR)	412
18.4.19 模拟看门狗状态寄存器 (ADC_AWDSR)	413
18.4.20 模拟看门狗状态复位寄存器 (ADC_AWDSCLRR)	414

18.4.21 模拟看门狗窗口 x 阈值寄存器 (ADC_AWDxDy) (x=0~1, y=0~1)	415
18.4.22 模拟看门狗窗口 x 比较通道选择寄存器 (ADC_AWDxCHSR) (x=0~1)	416
18.5 注意事项	417
18.5.1 数据寄存器读取时注意事项	417
18.5.2 扫描完成中断处理注意事项	417
18.5.3 模块停止与低功耗设定的注意事项	417
18.5.4 ADC 转换模拟通道输入的引脚设定	417
18.5.5 噪声控制	417
19 数模转换器 (DAC)	418
19.1 简介	418
19.2 主要特性	418
19.3 功能说明	418
19.3.1 功能框图	418
19.3.2 DAC 转换	419
19.3.3 同步转换	419
19.3.4 外部数据转换	419
19.3.5 ADC 转换优先模式	419
19.4 寄存器	421
19.4.1 寄存器总表	421
19.4.2 数据寄存器 (DAC_DADy) (y=1~2)	422
19.4.3 控制寄存器 (DAC_DACR)	423
19.4.4 模拟输出控制寄存器 (DAC_DAOCR)	424
19.4.5 ADC 转换优先控制寄存器 (DAC_DAADPCR)	425
19.5 注意事项	426
19.5.1 模块停止功能的设置	426
19.5.2 模块停止状态下的 DAC 动作	426
19.5.3 停止低功耗模式下的 DAC 动作	426
19.5.4 掉电低功耗模式下的 DAC 动作	426
19.5.5 使用输出放大器	426
20 高级控制定时器 (Timer6)	427
20.1 简介	427
20.2 主要特性	427
20.3 功能说明	428
20.3.1 Timer6 框图	428
20.3.2 基本动作	428
20.3.3 时钟源选择	430
20.3.4 计数方向	431
20.3.5 计数器刷新	431
20.3.6 数字滤波	431
20.3.7 软件同步	432
20.3.8 硬件同步	433
20.3.9 脉宽测量	434
20.3.10 周期测量	435
20.3.11 缓存功能	435
20.3.12 通用 PWM 输出	440

20.3.13 正交编码计数.....	444
20.3.14 周期间隔响应.....	448
20.3.15 EMB 控制.....	449
20.3.16 功能汇总表.....	450
20.3.17 中断和事件.....	451
20.4 操作示例.....	453
20.4.1 基本计数及中断动作.....	453
20.4.2 比较输出及中断动作.....	453
20.4.3 捕获输入及中断动作.....	453
20.4.4 缓存传送动作（周期基准值）	453
20.4.5 缓存传送动作（通用比较基准值）	454
20.4.6 缓存传送动作（专用比较基准值）	454
20.4.7 缓存传送动作（死区基准值）	454
20.4.8 同步启动动作（软件方式）	455
20.4.9 同步启动动作（硬件方式）	455
20.4.10 正交编码计数动作（2 相）	455
20.4.11 正交编码计数动作（3 相）	455
20.4.12 单路 PWM 输出.....	456
20.4.13 互补 PWM 输出（软件死区）	456
20.4.14 互补 PWM 输出（硬件死区）	456
20.4.15 EMB 监控及中断动作.....	457
20.5 寄存器.....	458
20.5.1 寄存器总表.....	458
20.5.2 通用计数值寄存器（TMR6_CNTER）	460
20.5.3 通用刷新值寄存器（TMR6_UPDAR）	460
20.5.4 通用周期基准值寄存器（TMR6_PERmR）（m=A~C）	461
20.5.5 通用比较基准值寄存器（TMR6_GCMmR）（m=A~F）	461
20.5.6 专用比较基准值寄存器（TMR6_SCMmR）（m=A~F）	462
20.5.7 死区时间基准值寄存器（TMR6_DTmnR）（m=U/D, n=A/B）	462
20.5.8 通用控制寄存器（TMR6_GCONR）	463
20.5.9 中断控制寄存器（TMR6_ICONR）	465
20.5.10 缓存控制寄存器（TMR6_BCONR）	467
20.5.11 死区控制寄存器（TMR6_DCONR）	470
20.5.12 端口控制寄存器 A（TMR6_PCNAR）	471
20.5.13 端口控制寄存器 B（TMR6_PCNBR）	474
20.5.14 滤波控制寄存器（TMR6_FCNGR）	477
20.5.15 有效周期寄存器（TMR6_VPERR）	478
20.5.16 状态标志寄存器（TMR6_STFLR）	479
20.5.17 硬件启动事件选择寄存器（TMR6_HSTAR）	481
20.5.18 硬件停止事件选择寄存器（TMR6_HSTPR）	483
20.5.19 硬件清零事件选择寄存器（TMR6_HCLRR）	485
20.5.20 硬件刷新事件选择寄存器（TMR6_HUPDR）	487
20.5.21 硬件捕获事件选择寄存器 A（TMR6_HCPAR）	489
20.5.22 硬件捕获事件选择寄存器 B（TMR6_HCPBR）	491
20.5.23 硬件递加事件选择寄存器（TMR6_HCUPR）	493

20.5.24 硬件递减事件选择寄存器 (TMR6_HCDOR)	495
20.5.25 公共滤波控制寄存器 (TMR6_FCNTR)	497
20.5.26 公共软件同步启动控制寄存器 (TMR6_SSTAR)	498
20.5.27 公共软件同步停止控制寄存器 (TMR6_SSTPR)	499
20.5.28 公共软件同步清零控制寄存器 (TMR6_SCLRR)	499
20.5.29 公共软件同步刷新控制寄存器 (TMR6_SUPDR)	500
20.6 注意事项.....	501
21 通用控制定时器 (Timer4)	503
21.1 简介.....	503
21.2 主要特性.....	503
21.3 功能说明.....	504
21.3.1 Timer4 框图.....	504
21.3.2 基本动作.....	505
21.3.3 缓存功能.....	507
21.3.4 通用 PWM 输出.....	512
21.3.5 周期间隔响应.....	516
21.3.6 EMB 控制.....	518
21.3.7 监测输出.....	518
21.3.8 硬件启动.....	519
21.3.9 同步启动.....	519
21.3.10 中断和事件.....	520
21.4 寄存器.....	522
21.4.1 寄存器总表.....	522
21.4.2 计数值寄存器 (TMR4_CNTR)	525
21.4.3 周期基准寄存器 (TMR4_CPSR)	525
21.4.4 控制状态寄存器 (TMR4_CCSR)	526
21.4.5 有效周期寄存器 (TMR4_CVPR)	528
21.4.6 通用比较基准寄存器 (TMR4_OCCRm)	529
21.4.7 通用控制状态寄存器 (TMR4_OCSRn)	530
21.4.8 通用扩展控制寄存器 (TMR4_OCERn)	531
21.4.9 通用模式控制寄存器 (TMR4_OCMRm)	533
21.4.10 专用比较基准寄存器 (TMR4_SCCRm)	538
21.4.11 专用控制状态寄存器 (TMR4_SCSRm)	539
21.4.12 专用扩展控制寄存器 (TMR4_SCER)	541
21.4.13 专用模式控制寄存器 (TMR4_SCMRm)	542
21.4.14 PWM 基本控制寄存器 (TMR4_POCRn)	543
21.4.15 PWM 状态控制寄存器 (TMR4_PSCR)	544
21.4.16 PWM 滤波控制寄存器 (TMR4_PFSRn)	548
21.4.17 PWM 死区控制寄存器 A (TMR4_PDARn)	548
21.4.18 PWM 死区控制寄存器 B (TMR4_PDBRn)	548
21.4.19 重载控制状态寄存器 (TMR4_RCSR)	549
21.4.20 专用中断控制寄存器 (TMR4_SCIR)	551
21.4.21 专用标志控制寄存器 (TMR4_SCFR)	552
22 紧急刹车模块 (EMB)	553
22.1 简介.....	553

22.2 功能说明.....	553
22.2.1 EMB 通道功能框图.....	553
22.2.2 EMB 结构框图.....	555
22.2.3 外部端口输入电平变化时控制 PWM 信号输出.....	555
22.2.4 PWM 输出端口电平发生同相（同高或同低）时停止 PWM 信号输出.....	556
22.2.5 根据电压比较器比较结果停止 PWM 信号输出.....	557
22.2.6 系统错误发生时停止 PWM 信号输出.....	557
22.2.7 写寄存器软件控制 PWM 信号输出.....	558
22.3 寄存器.....	559
22.3.1 寄存器总表.....	559
22.3.2 EMB 控制寄存器 1 (EMB_CTL1) (Group0)	560
22.3.3 EMB 控制寄存器 1 (EMB_CTL1) (Group1~3)	562
22.3.4 EMB 控制寄存器 2 (EMB_CTL2) (Group0)	564
22.3.5 EMB 控制寄存器 2 (EMB_CTL2) (Group1~3)	565
22.3.6 EMB 软件输出使能控制寄存器 (EMB_SOE)	566
22.3.7 EMB 状态寄存器 (EMB_STAT)	567
22.3.8 EMB 状态复位寄存器 (EMB_STATCLR)	568
22.3.9 EMB 中断许可寄存器 (EMB_INTEN)	569
22.3.10 EMB 控制 PWM 输出释放方式选择寄存器 (EMB_RLSSEL)	570
23 通用定时器 (TimerA)	571
23.1 简介.....	571
23.2 主要特性.....	571
23.3 功能说明.....	572
23.3.1 TimerA 框图.....	572
23.3.2 对称单元.....	572
23.3.3 基本动作.....	573
23.3.4 时钟源选择.....	575
23.3.5 数字滤波.....	575
23.3.6 同步启动.....	575
23.3.7 缓存功能.....	576
23.3.8 级联计数.....	577
23.3.9 通用 PWM 输出.....	578
23.3.10 正交编码计数.....	579
23.3.11 中断和事件.....	582
23.4 寄存器.....	583
23.4.1 寄存器总表.....	583
23.4.2 通用计数值寄存器 (TMRA_CNTER)	584
23.4.3 周期基准值寄存器 (TMRA_PERAR)	585
23.4.4 比较基准值寄存器 (TMRA_CMPARm) (m=1~8)	586
23.4.5 控制状态寄存器 L (TMRA_BCSTRL)	587
23.4.6 控制状态寄存器 H (TMRA_BCSTRH)	589
23.4.7 中断控制寄存器 (TMRA_ICONR)	590
23.4.8 事件控制寄存器 (TMRA_ECONR)	591
23.4.9 滤波控制寄存器 (TMRA_FCONR)	592
23.4.10 状态标志寄存器 (TMRA_STFLR)	593

23.4.11 缓存控制寄存器 (TMRA_BCONRm) (m=1~4)	596
23.4.12 捕获控制寄存器 (TMRA_CCONRm) (m=1~8)	597
23.4.13 端口控制寄存器 (TMRA_PCONRm) (m=1~8)	599
23.4.14 硬件触发事件选择寄存器 (TMRA_HCONR)	601
23.4.15 硬件递加事件选择寄存器 (TMRA_HCUPR)	603
23.4.16 硬件递减事件选择寄存器 (TMRA_HCDOR)	605
24 通用定时器 (Timer0)	607
24.1 简介.....	607
24.2 功能说明.....	607
24.2.1 Timer0 框图.....	607
24.2.2 时钟源选择.....	607
24.2.3 基本计数.....	608
24.2.4 硬件触发.....	608
24.2.5 软件同步启动.....	608
24.2.6 中断和事件.....	609
24.3 寄存器.....	610
24.3.1 寄存器总表.....	610
24.3.2 计数值寄存器 m (TMR0_CNTmR) (m=A/B)	611
24.3.3 基准值寄存器 m (TMR0_CMPmR) (m=A/B)	611
24.3.4 基本控制寄存器 (TMR0_BCONR)	612
24.3.5 状态标志寄存器 (TMR0_STFLR)	615
24.3.6 计数同步使能寄存器 (TMR_SYNENR)	616
24.4 注意事项.....	618
25 实时时钟 (RTC)	619
25.1 简介.....	619
25.2 主要特性.....	619
25.3 功能说明.....	620
25.3.1 RTC 框图.....	620
25.3.2 上电设定.....	620
25.3.3 系统低功耗模式切换.....	620
25.3.4 时钟误差补偿.....	620
25.3.5 中断.....	620
25.4 操作示例.....	621
25.4.1 RTC 计数开始设定.....	621
25.4.2 读出计数寄存器.....	621
25.4.3 写入计数寄存器.....	621
25.4.4 闹钟设定.....	621
25.4.5 1Hz 输出.....	622
25.5 寄存器.....	623
25.5.1 寄存器总表.....	623
25.5.2 控制寄存器 0 (RTC_CR0)	624
25.5.3 控制寄存器 1 (RTC_CR1)	625
25.5.4 控制寄存器 2 (RTC_CR2)	626
25.5.5 控制寄存器 3 (RTC_CR3)	627
25.5.6 秒计数寄存器 (RTC_SEC)	628

25.5.7 分计数寄存器 (RTC_MIN)	628
25.5.8 时计数寄存器 (RTC_HOUR)	629
25.5.9 日计数寄存器 (RTC_DAY)	630
25.5.10 周计数寄存器 (RTC_WEEK)	631
25.5.11 月计数寄存器 (RTC_MON)	631
25.5.12 年计数寄存器 (RTC_YEAR)	632
25.5.13 分闹钟寄存器 (RTC_ALMMIN)	632
25.5.14 时闹钟寄存器 (RTC_ALMHOUR)	633
25.5.15 周闹钟寄存器 (RTC_ALMWEEK)	633
25.5.16 时钟误差补偿寄存器 (RTC_ERRCRH、RTC_ERRCRL)	634
26 看门狗计数器 (WDT/SWDT)	636
26.1 简介.....	636
26.2 主要特性.....	636
26.3 功能说明.....	636
26.3.1 启动看门狗.....	636
26.3.2 硬件启动方式.....	636
26.3.3 软件启动方式.....	637
26.3.4 刷新动作.....	638
26.3.5 标志位.....	638
26.3.6 中断复位.....	638
26.3.7 计数下溢.....	639
26.3.8 刷新错误.....	640
26.4 寄存器.....	641
26.4.1 寄存器总表.....	641
26.4.2 控制寄存器 (SWDT_CR、WDT_CR)	642
26.4.3 状态寄存器 (SWDT_SR、WDT_SR)	643
26.4.4 刷新寄存器 (SWDT_RR、WDT_RR)	644
26.5 注意事项.....	644
27 通用同步异步收发器 (USART)	645
27.1 简介.....	645
27.2 主要特性.....	645
27.3 功能说明.....	647
27.3.1 系统框图.....	647
27.3.2 UART 功能.....	648
27.3.3 多处理器通信.....	655
27.3.4 UART-LIN.....	658
27.3.5 智能卡.....	660
27.3.6 时钟同步模式.....	663
27.3.7 数字滤波功能.....	667
27.3.8 中断.....	667
27.4 寄存器.....	669
27.4.1 寄存器总表.....	669
27.4.2 状态寄存器 (USART_SR)	670
27.4.3 发送数据寄存器 (USART_TDR)	674
27.4.4 接收数据寄存器 (USART_RDR)	674

27.4.5 波特率寄存器 (USART_BRR)	675
27.4.6 控制寄存器 1 (USART_CR1)	676
27.4.7 控制寄存器 2 (USART_CR2)	679
27.4.8 控制寄存器 3 (USART_CR3)	682
27.4.9 预分频寄存器 (USART_PR)	684
27.4.10 LIN 波特率测量计数寄存器 (USART_LBMC)	685
27.4.11 USART1 滤波控制寄存器 (USART1_NFC)	686
27.5 注意事项.....	687
27.5.1 UART 注意事项.....	687
27.5.2 时钟同步模式注意事项.....	687
27.5.3 其他注意事项.....	687
28 集成电路总线 (I2C)	688
28.1 简介.....	688
28.2 主要特性.....	688
28.3 功能说明.....	688
28.3.1 I2C 系统框图.....	688
28.3.2 I2C 协议.....	689
28.3.3 地址匹配.....	696
28.3.4 SMBus 动作.....	699
28.3.5 复位.....	699
28.3.6 可编程数字滤波.....	700
28.3.7 中断和事件.....	700
28.4 操作示例.....	701
28.4.1 I2C 初始化流程.....	701
28.5 寄存器.....	703
28.5.1 寄存器总表.....	703
28.5.2 I2C 控制寄存器 1 (I2C_CR1)	704
28.5.3 I2C 控制寄存器 2 (I2C_CR2)	706
28.5.4 I2C 控制寄存器 3 (I2C_CR3)	707
28.5.5 I2C 控制寄存器 4 (I2C_CR4)	708
28.5.6 I2C 从机地址寄存器 0 (I2C_SLR0)	709
28.5.7 I2C 从机地址寄存器 1 (I2C_SLR1)	711
28.5.8 I2C SCL 电平超时控制寄存器 (I2C_SLTR)	712
28.5.9 I2C 状态寄存器 (I2C_SR)	713
28.5.10 I2C 状态清零寄存器 (I2C_CLR)	718
28.5.11 I2C 数据发送寄存器 (I2C_DTR)	719
28.5.12 I2C 数据接收寄存器 (I2C_DRR)	720
28.5.13 I2C 时钟控制寄存器 (I2C_CCR)	721
28.5.14 I2C 滤波控制寄存器 (I2C_FLTR)	722
28.5.15 I2C FIFO 状态寄存器 (I2C_FSTR)	723
28.5.16 I2C 从机地址接收器寄存器 (I2C_SLVADRR)	724
29 串行外设接口 (SPI)	725
29.1 简介.....	725
29.2 主要特性.....	725
29.3 功能说明.....	726

29.3.1 系统框图.....	726
29.3.2 SPI 动作系统说明.....	727
29.3.3 数据通信说明.....	729
29.3.4 运行说明.....	735
29.3.5 奇偶校验位自诊断.....	739
29.3.6 错误检测.....	740
29.3.7 SPI 的初始化.....	743
29.3.8 中断和事件.....	744
29.4 寄存器.....	745
29.4.1 寄存器总表.....	745
29.4.2 SPI 数据寄存器 (SPI_DR)	746
29.4.3 SPI 控制寄存器 (SPI_CR)	747
29.4.4 SPI 通信配置寄存器 1 (SPI_CFG1)	749
29.4.5 SPI 状态寄存器 (SPI_SR)	751
29.4.6 SPI 通信配置寄存器 2 (SPI_CFG2)	752
30 四线式串行外设接口 (QSPI)	754
30.1 简介.....	754
30.2 主要特性.....	754
30.3 内存映射.....	755
30.3.1 内部总线空间.....	755
30.3.2 ROM 空间及总线的地址宽度.....	755
30.4 功能说明.....	757
30.4.1 基本框图.....	757
30.4.2 QSPI 总线.....	757
30.4.3 QSPI 总线的时序调整.....	759
30.4.4 用于 ROM 访问的 SPI 指令介绍.....	761
30.4.5 QSPI 总线周期的安排.....	769
30.4.6 XIP 控制.....	771
30.4.7 QSIO2 和 QSIO3 的管脚状态.....	772
30.4.8 直接通信模式.....	772
30.4.9 中断.....	773
30.5 寄存器.....	774
30.5.1 寄存器总表.....	774
30.5.2 QSPI 控制寄存器 (QSPI_CR)	775
30.5.3 QSPI 片选控制寄存器 (QSPI_CSCR)	776
30.5.4 QSPI 格式控制寄存器 (QSPI_FCR)	777
30.5.5 QSPI 状态寄存器 (QSPI_SR)	778
30.5.6 QSPI 直接通信指令寄存器 (QSPI_DCOM)	779
30.5.7 QSPI 指令代码寄存器 (QSPI_CCMD)	779
30.5.8 QSPI XIP 模式代码寄存器 (QSPI_XCMD)	780
30.5.9 QSPI 标志清除寄存器 (QSPI_CLR)	780
30.5.10 QSPI 外部扩展地址寄存器 (QSPI_EXAR)	780
30.6 注意事项.....	781
30.6.1 QSPI 寄存器的设置.....	781
30.6.2 模块停止位的设置.....	781

31 外部存储器控制器 (EXMC)	782
31.1 简介	782
31.2 主要特性	782
31.3 功能说明	783
31.3.1 控制器架构	783
31.3.2 基本访问规范	783
31.3.3 地址映射	785
31.3.4 协议接口	786
31.3.5 SMC-SRAM/PSRAM/NOR Flash 控制器	787
31.4 寄存器	803
31.4.1 寄存器总表	803
31.4.2 EXMC 公共寄存器	804
31.4.3 SMC-SRAM/PSRAM/NOR Flash 寄存器	805
31.5 注意事项	814
32 控制器局域网 (MCAN)	815
32.1 简介	815
32.2 主要特性	815
32.3 功能说明	816
32.3.1 MCAN 系统框图	816
32.3.2 引脚说明	817
32.3.3 工作模式	817
32.3.4 时间戳生成	824
32.3.5 超时计数器	824
32.3.6 接收处理	825
32.3.7 发送处理	830
32.3.8 FIFO 确认处理	834
32.3.9 消息 RAM	835
32.4 寄存器	845
32.4.1 寄存器总表	845
32.4.2 MCAN 字节序寄存器 (MCAN_ENDN)	847
32.4.3 MCAN 数据位时间和预分频寄存器 (MCAN_DBTP)	848
32.4.4 MCAN 测试寄存器 (MCAN_TEST)	849
32.4.5 MCAN RAM 看门狗寄存器 (MCAN_RWD)	850
32.4.6 MCAN CC 控制寄存器 (MCAN_CCCR)	851
32.4.7 MCAN 标称位时间和预分频寄存器 (MCAN_NBTP)	853
32.4.8 MCAN 时间戳计数器配置寄存器 (MCAN_TSCC)	854
32.4.9 MCAN 时间戳计数器值寄存器 (MCAN_TSCV)	855
32.4.10 MCAN 超时计数器配置寄存器 (MCAN_TOCC)	856
32.4.11 MCAN 超时计数器值寄存器 (MCAN_TOCV)	857
32.4.12 MCAN 错误计数器寄存器 (MCAN_ECR)	858
32.4.13 MCAN 协议状态寄存器 (MCAN_PSR)	859
32.4.14 MCAN 发送延迟补偿寄存器 (MCAN_TDCR)	862
32.4.15 MCAN 中断寄存器 (MCAN_IR)	863
32.4.16 MCAN 中断使能寄存器 (MCAN_IE)	866
32.4.17 MCAN 中断线选择寄存器 (MCAN_ILS)	868

32.4.18 MCAN 中断线使能寄存器 (MCAN_ILE)	870
32.4.19 MCAN 全局过滤器配置寄存器 (MCAN_GFC)	871
32.4.20 MCAN 标准 ID 过滤器配置寄存器 (MCAN_SIDFC)	872
32.4.21 MCAN 扩展 ID 过滤器配置寄存器 (MCAN_XIDFC)	872
32.4.22 MCAN 扩展 ID 与掩码寄存器 (MCAN_XIDAM)	873
32.4.23 MCAN 高优先级消息状态寄存器 (MCAN_HPMS)	873
32.4.24 MCAN 新数据 1 寄存器 (MCAN_NDAT1)	874
32.4.25 MCAN 新数据 2 寄存器 (MCAN_NDAT2)	874
32.4.26 MCAN 接收 FIFO0 配置寄存器 (MCAN_RXF0C)	875
32.4.27 MCAN 接收 FIFO0 状态寄存器 (MCAN_RXF0S)	876
32.4.28 MCAN 接收 FIFO0 确认寄存器 (MCAN_RXF0A)	877
32.4.29 MCAN 接收缓冲区配置寄存器 (MCAN_RXBC)	877
32.4.30 MCAN 接收 FIFO1 配置寄存器 (MCAN_RXF1C)	878
32.4.31 MCAN 接收 FIFO1 状态寄存器 (MCAN_RXF1S)	879
32.4.32 MCAN 接收 FIFO1 确认寄存器 (MCAN_RXF1A)	880
32.4.33 MCAN 接收缓冲区和 FIFO 元素大小配置寄存器 (MCAN_RXESC)	881
32.4.34 MCAN 发送缓冲区配置寄存器 (MCAN_TXBC)	882
32.4.35 MCAN 发送 FIFO/队列状态寄存器 (MCAN_TXFQS)	883
32.4.36 MCAN 发送缓冲区元素大小配置寄存器 (MCAN_TXESC)	884
32.4.37 MCAN 发送缓冲区请求挂起寄存器 (MCAN_TXBRP)	885
32.4.38 MCAN 发送缓冲区添加请求寄存器 (MCAN_TXBAR)	886
32.4.39 MCAN 发送缓冲区取消请求寄存器 (MCAN_TXBCR)	886
32.4.40 MCAN 发送缓冲区发送已发生寄存器 (MCAN_TXBTO)	887
32.4.41 MCAN 发送缓冲区取消已完成寄存器 (MCAN_TXBCF)	887
32.4.42 MCAN 发送缓冲区发送中断使能寄存器 (MCAN_TXBTIE)	888
32.4.43 MCAN 发送缓冲区取消已完成中断使能寄存器 (MCAN_TXBCIE)	888
32.4.44 MCAN 发送事件 FIFO 配置寄存器 (MCAN_TXEFC)	889
32.4.45 MCAN 发送事件 FIFO 状态寄存器 (MCAN_TXEFS)	890
32.4.46 MCAN 发送事件 FIFO 确认寄存器 (MCAN_TXEFA)	891
32.5 注意事项	891
32.5.1 CAN 总线抗干扰措施	891
32.5.2 CAN 控制器噪声制约	891
33 加密协处理模块 (CPM)	892
33.1 简介	892
33.2 功能说明	893
33.2.1 加解密算法处理器 (AES)	893
33.2.2 安全散列算法 (HASH)	897
33.2.3 真随机数发生器 (TRNG)	900
33.3 寄存器	901
33.3.1 寄存器总表	901
33.3.2 AES 寄存器	902
33.3.3 HASH 寄存器	905
33.3.4 TRNG 寄存器	907
34 数据计算单元 (DCU)	909
34.1 简介	909

34.2 主要特性.....	909
34.3 功能说明.....	909
34.3.1 加法模式.....	909
34.3.2 减法模式.....	909
34.3.3 硬件触发启动模式.....	910
34.3.4 比较模式.....	910
34.3.5 三角波输出模式.....	911
34.3.6 递增锯齿波输出模式.....	912
34.3.7 递减锯齿波输出模式.....	913
34.3.8 中断和事件.....	914
34.4 寄存器.....	915
34.4.1 寄存器总表.....	915
34.4.2 DCU 控制寄存器 (DCU_CTL)	916
34.4.3 DCU 标志寄存器 (DCU_FLAG)	917
34.4.4 DCU 数据寄存器 (DCU_DATAx) (x=0~2)	918
34.4.5 DCU 标志复位寄存器 (DCU_FLAGCLR)	921
34.4.6 DCU 中断和事件寄存器 (DCU_INTEVTSEL)	922
34.5 注意事项.....	923
35 CRC 运算 (CRC)	924
35.1 简介.....	924
35.2 主要特性.....	924
35.3 功能说明.....	924
35.3.1 工作模式.....	924
35.3.2 编码方式.....	924
35.3.3 写入位宽.....	925
35.4 操作示例.....	925
35.4.1 CRC-16 编码模式.....	925
35.4.2 CRC-16 检验模式.....	925
35.4.3 CRC-32 编码模式.....	925
35.4.4 CRC-32 检验模式.....	925
35.5 寄存器.....	926
35.5.1 寄存器总表.....	926
35.5.2 控制寄存器 (CRC_CR)	927
35.5.3 结果寄存器 (CRC_RESULT)	927
35.5.4 数据寄存器 (CRC_DAT)	928
36 调试控制器 (DBG)	929
36.1 简介.....	929
36.2 功能说明.....	929
36.2.1 功能框图.....	929
36.2.2 参考文档.....	930
36.2.3 SWJ-DP 调试端口 (SWD 和 JTAG)	930
36.2.4 引脚排列和调试端口引脚.....	931
36.2.5 SW 调试端口-SW 协议简介.....	932
36.2.6 TPIU (跟踪端口接口单元)	932
36.3 寄存器.....	935

36.3.1 寄存器总表.....	935
36.3.2 芯片专属标志寄存器（CHIPID）	936
36.3.3 DBG 状态寄存器（MCUDBGSTAT）	936
36.3.4 外设调试暂停寄存器（MCUSTPCTL）	937
36.3.5 调试组件配置寄存器（MCUTRACECTL）	938
36.3.6 外设调试暂停寄存器 2（MCUSTPCTL2）	939
A 术语表.....	940
修订记录.....	941

图目录

图 2-1 QSPI 地址空间分配.....	42
图 3-1 总线架构图.....	48
图 4-1 上电复位时序图.....	52
图 4-2 NRST 复位时序图.....	53
图 4-3 欠压复位时序图.....	53
图 4-4 可编程电压检测 1 复位时序图.....	54
图 4-5 可编程电压检测 2 复位时序图.....	54
图 4-6 看门狗和专用看门狗复位时序图.....	55
图 4-7 掉电唤醒复位时序图.....	55
图 4-8 软件复位时序图.....	55
图 4-9 MPU 错误复位时序图.....	56
图 4-10 RAM 奇偶校验复位时序图.....	56
图 4-11 RAM ECC 复位时序图.....	56
图 4-12 时钟频率异常复位时序图.....	56
图 4-13 外部高速振荡异常复位时序图.....	57
图 4-14 Cortex-M4 Lockup 复位时序图.....	57
图 4-15 模块功能复位.....	58
图 5-1 时钟系统框图.....	71
图 5-2 时钟频率测量框图.....	72
图 5-3 外部高速振荡器连接示例.....	74
图 5-4 外部时钟输入的连接示例.....	74
图 5-5 外部高速振荡器故障检测示例.....	74
图 5-6 系统时钟选择 XTAL，检测到 XTAL 振荡故障示例.....	75
图 5-7 外部低速振荡器连接示例.....	76
图 5-8 时钟源切换流程图.....	78
图 5-9 时钟分频切换流程图.....	79
图 5-10 时钟频率测量时序图.....	80
图 6-1 电源构成图.....	117
图 6-2 上电复位、掉电复位波形.....	117
图 6-3 欠压复位波形.....	118
图 6-4 PVD1 中断/复位框图.....	119
图 6-5 PVD2 中断/复位框图.....	119
图 6-6 电源监视 1 中断时序图.....	120
图 6-7 电源监视 1 复位时序图.....	120
图 6-8 电源监视 2 中断时序图.....	121
图 6-9 电源监视 2 复位时序图.....	121
图 6-10 内部电压采样示意图.....	122
图 6-11 PTWKn 结构框图.....	129
图 8-1 Flash 地址分布.....	168
图 8-2 启动扇区交换功能 1.....	174
图 8-3 启动交换功能 2.....	175
图 9-1 错误注入功能示意图.....	192
图 10-1 端口基本结构示意图.....	203
图 11-1 中断系统功能框图.....	240
图 11-2 中断请求/事件输入选择.....	241
图 11-3 数字滤波器工作示意图.....	242
图 12-1 AOS 模块框图.....	255
图 12-2 PLU 框图.....	258
图 13-1 只允许部分空间访问示意图.....	279
图 13-2 只禁止部分空间访问示意图.....	279

图 14-1	KEYSCAN 系统框图.....	297
图 14-2	键盘扫描功能示意图.....	298
图 15-1	CTC 功能框图.....	303
图 15-2	CTC 校准示意图.....	309
图 16-1	DMA 结构图.....	318
图 16-2	连锁传输示意图.....	320
图 16-3	不连续地址传输示意图（以源地址为例）	321
图 16-4	不连续式重置示意图.....	323
图 16-5	应用例 1：存储器到存储器传输.....	325
图 16-6	应用例 2：存储器到外围电路的传输.....	326
图 17-1	CMP 功能框图.....	357
图 17-2	电压比较示意图.....	359
图 17-3	窗口比较示意图.....	360
图 17-4	输入扫描示意图.....	361
图 17-5	窗口输出示意图一.....	362
图 17-6	窗口输出示意图二.....	362
图 17-7	窗口输出示意图三.....	363
图 18-1	ADC 框图.....	375
图 18-2	通道映射示意图.....	377
图 18-3	内部模拟信号选择.....	377
图 18-4	序列 A 单次扫描模式.....	378
图 18-5	连续扫描.....	379
图 18-6	数据缓冲模式.....	380
图 18-7	双序列扫描模式（序列 A 从被中断通道重新启动）	381
图 18-8	双序列扫描模式（序列 A 从第一个通道重新启动）	381
图 18-9	模拟看门狗保护区域（比较模式）	382
图 18-10	ADC 转换时间示意图.....	383
图 18-11	平均功能有效时的转换动作.....	384
图 18-12	单次并行触发模式（三个 ADC）	385
图 18-13	单次延迟触发模式（三个 ADC）	386
图 18-14	循环并行触发模式（三个 ADC）	387
图 18-15	循环延迟触发模式（两个 ADC）	387
图 18-16	循环延迟触发模式（三个 ADC）	388
图 18-17	ADC 中断和事件输出时序.....	389
图 19-1	DAC 转换通道框图.....	418
图 19-2	DAC 转换示意图.....	419
图 19-3	ADC 转换优先模式动作示意图.....	420
图 20-1	Timer6 基本框图.....	428
图 20-2	锯齿波波形（递加计数）	429
图 20-3	三角波波形.....	429
图 20-4	比较输出动作.....	430
图 20-5	捕获输入动作.....	430
图 20-6	硬件刷新动作.....	431
图 20-7	捕获输入端口的滤波功能.....	432
图 20-8	软件同步动作.....	433
图 20-9	硬件同步动作.....	434
图 20-10	脉宽测量.....	435
图 20-11	周期测量.....	435
图 20-12	单缓存方式比较输出时序.....	436
图 20-13	双缓存方式捕获输入时序.....	437
图 20-14	锯齿波模式时计数缓存动作.....	438
图 20-15	三角波模式时计数缓存动作 1.....	439

图 20-16 三角波模式时计数缓存动作 2.....	439
图 20-17 单边对齐独立 PWM.....	440
图 20-18 双边对称独立 PWM.....	441
图 20-19 软件设定 GCMBR 互补 PWM 波输出.....	441
图 20-20 硬件设定 GCMBR 互补 PWM 波输出.....	442
图 20-21 双边非对称 PWM 输出.....	442
图 20-22 单边对齐独立 PWM.....	443
图 20-23 带死区时间双边对称互补 PWM.....	444
图 20-24 位置模式-基本计数.....	445
图 20-25 位置计数模式-相位差计数（1 倍计数）	445
图 20-26 位置计数模式-相位差计数（2 倍计数）	445
图 20-27 位置计数模式-相位差计数（4 倍计数）	446
图 20-28 位置计数模式-方向计数.....	446
图 20-29 公转计数模式-Z 相计数.....	446
图 20-30 公转计数模式-位置溢出计数.....	447
图 20-31 公转计数模式-混合计数.....	447
图 20-32 公转计数模式-混合计数 Z 相屏蔽动作例 1.....	447
图 20-33 公转计数模式-混合计数 Z 相屏蔽动作例 2.....	448
图 20-34 周期间隔有效请求信号动作.....	448
图 20-35 锯齿波模式时中断&事件输出例.....	452
图 21-1 Timer4 基本框图.....	504
图 21-2 Timer4 锯齿波波形.....	505
图 21-3 Timer4 三角波波形.....	505
图 21-4 Timer4 锯齿波模式计数动作.....	506
图 21-5 Timer4 三角波模式计数动作.....	506
图 21-6 锯齿波模式波形输出例.....	507
图 21-7 三角波模式波形输出例.....	507
图 21-8 缓存无效时修改锯齿波计数周期.....	508
图 21-9 缓存使能时修改锯齿波计数周期.....	508
图 21-10 缓存使能时修改三角波计数周期.....	509
图 21-11 OCCR 缓冲数据传输（周期间隔响应链接禁止时）	509
图 21-12 OCCR 缓冲数据传输（周期间隔响应链接使能）	510
图 21-13 输出比较缓冲数据传输（OCMR 缓冲使能）	510
图 21-14 SCCR 缓冲传输操作（周期间隔响应链接传输禁止时）	511
图 21-15 SCCR 缓冲传输操作（周期间隔响应链接传输使能时）	512
图 21-16 锯齿波独立 PWM 输出例.....	512
图 21-17 三角波独立 PWM 输出例.....	513
图 21-18 锯齿波扩展 PWM 输出.....	513
图 21-19 软件实现互补 PWM 输出.....	514
图 21-20 死区定时器模式下的互补 PWM 输出.....	515
图 21-21 脉宽异常时的死区定时器模式下的波形输出.....	515
图 21-22 死区定时器滤波模式下的互补 PWM 输出.....	516
图 21-23 周期间隔响应时序图.....	517
图 21-24 专用事件输出信号周期间隔响应输出.....	518
图 21-25 PWM 周期输出监测示例.....	519
图 21-26 比较启动模式下专用事件输出 TMR4_<t>_ADSM 时序.....	519
图 21-27 延时启动模式下专用事件输出信号的输出时序.....	521
图 22-1 EMB 通道功能框图.....	554
图 22-2 EMB 结构框图.....	555
图 23-1 TimerA 基本框图.....	572
图 23-2 锯齿波波形（递加计数）	573
图 23-3 三角波波形.....	573

图 23-4 比较输出动作.....	573
图 23-5 捕获输入动作.....	574
图 23-6 时钟输入端口的滤波功能.....	575
图 23-7 软件同步动作.....	576
图 23-8 锯齿波模式时缓存动作.....	577
图 23-9 锯齿波模式时缓存与特殊缓存动作.....	577
图 23-10 32 位级联计数动作.....	578
图 23-11 单边对齐 PWM 输出例.....	578
图 23-12 双边对称 PWM 输出例.....	579
图 23-13 位置模式-基本计数.....	580
图 23-14 位置计数模式-相位差计数（1 倍计数）	580
图 23-15 位置计数模式-相位差计数（2 倍计数）	580
图 23-16 位置计数模式-相位差计数（4 倍计数）	580
图 23-17 位置计数模式-方向计数.....	581
图 23-18 公转计数模式-Z 相计数.....	581
图 23-19 公转计数模式-位置溢出计数.....	581
图 23-20 公转计数模式-混合计数.....	582
图 24-1 Timer0 基本框图.....	607
图 24-2 Timer0 计数时序图.....	608
图 25-1 RTC 基本框图.....	620
图 26-1 硬件启动例.....	637
图 26-2 软件启动例.....	637
图 26-3 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）	638
图 26-4 计数器下溢动作例.....	639
图 26-5 计数器刷新动作例.....	640
图 27-1 USART 系统框图.....	647
图 27-2 UART 数据格式.....	648
图 27-3 UART 发送数据图例 1.....	650
图 27-4 UART 发送数据图例 2.....	650
图 27-5 UART 内部同步和采样时序.....	651
图 27-6 UART 接收数据图例 1.....	652
图 27-7 UART 接收数据图例 2.....	652
图 27-8 多处理器通信图例.....	655
图 27-9 多处理器模式数据格式.....	655
图 27-10 多处理器模式发送数据图例.....	656
图 27-11 多处理器模式接收数据图例 1.....	657
图 27-12 多处理器模式接收数据图例 2.....	658
图 27-13 LIN 总线数据行为.....	658
图 27-14 智能卡连接示意图.....	660
图 27-15 智能卡模式同步时序和采样时序图.....	661
图 27-16 智能卡模式数据格式.....	661
图 27-17 智能卡模式发送数据图例.....	662
图 27-18 智能卡模式接收数据图例.....	663
图 27-19 时钟同步模式数据格式.....	664
图 27-20 时钟同步模式发送数据图例 1（LSB/无 CTS 功能）	665
图 27-21 时钟同步模式发送数据图例 2（LSB/CTS 功能）	665
图 27-22 时钟同步模式接收数据图例 1（LSB/无 RTS 功能）	666
图 27-23 时钟同步模式接收数据图例 2（LSB/RTS 功能）	666
图 28-1 I2C 系统框图.....	688
图 28-2 I2C 总线的结构例.....	689
图 28-3 I2C 总线的时序图.....	690
图 28-4 I2C 总线的数据格式.....	690

图 28-5 7 位地址格式的主机发送数据时序图 (例)	691
图 28-6 7 位地址格式的主机接收数据的时序图 (例)	692
图 28-7 7 位地址格式的从机发送模式时序图 (例)	692
图 28-8 7 位地址格式从机接收模式时序图 (例)	693
图 28-9 SCL 同步时序	694
图 28-10 从机发送时序图 (1)	694
图 28-11 从机发送时序图 (2)	695
图 28-12 从机接收时序图	695
图 28-13 快速 ACK/NACK 时序图	696
图 28-14 选择 7 位地址格式时的时序	696
图 28-15 选择 10 位地址格式时的时序	697
图 28-16 广播地址匹配时序图	697
图 28-17 SMBus 主机地址匹配时序图	698
图 28-18 SMBus 报警响应地址匹配时序图	698
图 28-19 SMBus 默认地址时序图	698
图 28-20 数字滤波电路框图	700
图 29-1 SPI 系统框图	726
图 29-2 主机模式结构	728
图 29-3 三线式时钟同步运行	729
图 29-4 数据格式	730
图 29-5 MSB 先传, 奇偶校验无效	730
图 29-6 LSB 先传, 奇偶校验无效	730
图 29-7 MSB 先传, 奇偶校验有效	731
图 29-8 LSB 先传, 奇偶校验有效	731
图 29-9 数据传送格式图 (CPHA=0)	732
图 29-10 数据传送格式图 (CPHA=1)	732
图 29-11 全双工同步串行通信	733
图 29-12 只进行发送通信	733
图 29-13 常规通信	734
图 29-14 连续通信	734
图 29-15 奇偶校验流程	740
图 29-16 过载错误处理	742
图 29-17 启用时钟自动停止功能时的动作示意图 (CPHA=1)	742
图 29-18 启用时钟自动停止功能时的动作示意图 (CPHA=0)	743
图 29-19 奇偶校验错误	743
图 30-1 默认区域设定及 AHB 总线空间内存映射关系图	755
图 30-2 QSPI-ROM 空间内存映像图	756
图 30-3 QSPI 的模块构成图	757
图 30-4 扩展式 SPI 协议动作示意图 1 (快速读模式)	758
图 30-5 扩展式 SPI 协议动作示意图 2 (四线式输入输出快速读模式)	758
图 30-6 二线式 SPI 协议动作示意图 (快速读模式)	758
图 30-7 四线式 SPI 协议动作示意图 (快速读模式)	759
图 30-8 串行接口的基本时序图	759
图 30-9 基准时钟选择 HCLK 三分频时输出时钟占空比补正示意图	760
图 30-10 QSSN 建立时间配置示意图	760
图 30-11 QSSN 保持时间配置示意图	761
图 30-12 数据接收延迟示意图	761
图 30-13 标准读总线周期示意图	762
图 30-14 快速读总线周期示意图	763
图 30-15 选择 XIP 模式的快速读总线周期示意图	763
图 30-16 二线式输出快速读总线周期示意图	764
图 30-17 选择 XIP 模式的二线式输出快速读总线周期示意图	764

图 30-18 二线式输入输出快速读总线周期示意图.....	765
图 30-19 选择 XIP 模式的二线式输入输出快速读总线周期示意图.....	765
图 30-20 四线式输出快速读总线周期示意图.....	766
图 30-21 选择 XIP 模式的四线式输出快速读总线周期示意图.....	766
图 30-22 四线式输入输出快速读总线周期示意图.....	767
图 30-23 选择 XIP 模式的四线式输入输出快速读总线周期示意图.....	767
图 30-24 进入 4-Bytes 模式指令总线周期示意图.....	768
图 30-25 退出 4-Bytes 模式指令总线周期示意图.....	768
图 30-26 写许可指令总线周期示意图.....	769
图 30-27 独立转换的单一闪存数据读取操作示意图.....	769
图 30-28 预读取功能有效时数据读取操作示意图.....	770
图 30-29 使用 QSPI 总线周期延长功能的数据读取操作示意图.....	770
图 30-30 XIP 模式控制示意图.....	771
图 31-1 EXMC 架构图.....	783
图 31-2 外部空间地址分配.....	785
图 31-3 SMC 的地址空间划分.....	786
图 31-4 SMC 状态图.....	788
图 31-5 SMC 初始设定流程.....	788
图 31-6 单次读动作基本时序（异步方式（RSYN=0b0）&8 位位宽（MW=0b00））.....	790
图 31-7 单次读动作基本时序（异步方式（RSYN=0b0）&16 位位宽（MW=0b01））.....	791
图 31-8 单次读动作基本时序（同步方式（RSYN=0b1）&8 位位宽（MW=0b00））.....	792
图 31-9 单次读动作基本时序（同步方式（RSYN=0b1）&16 位位宽（MW=0b01））.....	793
图 31-10 地址数据线复用单次读动作基本时序（8 位位宽（MW=0b00））.....	794
图 31-11 地址数据线复用单次读动作基本时序（16 位位宽（MW=0b01））.....	795
图 31-12 单次写动作基本时序（异步方式（WSYN=0b0）&8 位位宽（MW=0b00）&BLSS=0b0）.....	796
图 31-13 单次写动作基本时序（异步方式（WSYN=0b0）&16 位位宽（MW=0b01）&BLSS=0b1）.....	797
图 31-14 单次写动作基本时序（同步方式（WSYN=0b1）&8 位位宽（MW=0b00）&BLSS=0b1）.....	798
图 31-15 单次写动作基本时序（同步方式（WSYN=0b1）&16 位位宽（MW=0b01）&BLSS=0b0）.....	799
图 31-16 地址数据线复用单次写动作基本时序（异步方式（WSYN=0b0）&16 位位宽（MW=0b01））.....	800
图 31-17 地址数据线复用单次写动作基本时序（同步方式（WSYN=0b1）&16 位位宽（MW=0b01））.....	801
图 31-18 外部反馈时钟访问.....	814
图 32-1 MCAN 系统框图.....	816
图 32-2 收发器延迟测量.....	821
图 32-3 总线监听模式下的引脚控制.....	822
图 32-4 外部回环模式的引脚控制.....	823
图 32-5 内部回环模式的引脚控制.....	824
图 32-6 标准消息 ID 过滤器路径.....	826
图 32-7 扩展消息 ID 过滤器路径.....	827
图 32-8 接收 FIFO 状态.....	828
图 32-9 接收 FIFO 溢出处理.....	829
图 32-10 混合配置专用发送缓冲区和发送 FIFO 示例.....	833
图 32-11 混合配置专用发送缓冲区和发送队列示例.....	833
图 32-12 消息 RAM 配置.....	835
图 33-1 AES 的加解密示意图.....	893
图 33-2 AES 的加密流程图.....	893
图 33-3 HASH 算法流程图.....	898
图 33-4 TRNG 系统框图.....	900
图 34-1 硬件触发启动示意图.....	910
图 34-2 三角波输出模式波形.....	911
图 34-3 三角波输出模式.....	912
图 34-4 递增锯齿波输出模式波形.....	912
图 34-5 递增锯齿波输出模式.....	913

图 34-6 递减锯齿波输出模式波形..... 913

图 34-7 递减锯齿波输出模式..... 914

图 36-1 调试控制系统框图..... 929

图 36-2 调试控制系统..... 930

图 36-3 JTAG-DP 到 SW-DP 切换时序..... 931

图 36-4 TPIU 框图..... 933

表目录

表 2-1 存储器映射.....	38
表 2-2 目标地址配置示例.....	43
表 2-3 存储器映射 (Memory Mapping) 寄存器偏移地址.....	44
表 4-1 复位方式及产生条件.....	50
表 4-2 复位方式及复位标志.....	51
表 4-3 各模块复位条件.....	58
表 4-4 复位控制 (RMU) 寄存器偏移地址.....	60
表 5-1 时钟源主要特性.....	69
表 5-2 各个内部时钟的规格.....	72
表 5-3 时钟频率测量 (FCM) 寄存器偏移地址.....	82
表 5-4 时钟控制器 (CMU) 寄存器偏移地址.....	82
表 5-5 时钟控制器 (CMU) 寄存器偏移地址.....	82
表 5-6 时钟控制器 (CMU) 寄存器偏移地址.....	83
表 6-1 BOR 配置.....	118
表 6-2 PVD1/PVD2 特性.....	118
表 6-4 运行模式.....	122
表 6-5 低功耗模式.....	122
表 6-6 低功耗模式的运行条件及各模块在低功耗模式下的状态.....	123
表 6-7 运行模式说明.....	125
表 6-8 掉电模式子模式.....	127
表 6-10 RAM 模块与 RAM 掉电控制位.....	130
表 6-11 寄存器保护列表.....	131
表 6-12 电源控制 (PWC) 寄存器偏移地址.....	131
表 6-13 电源控制 (PWC) 寄存器偏移地址.....	132
表 6-14 电源控制 (PWC) 寄存器偏移地址.....	132
表 7-1 初始化配置 (ICG) 寄存器偏移地址.....	163
表 8-1 CPU 时钟频率和 Flash 读等待周期对照表.....	168
表 8-2 Flash 实际读周期数.....	169
表 8-3 OTP 地址分布表.....	173
表 8-4 嵌入式 Flash (EFM) 寄存器偏移地址.....	176
表 9-1 SRAM 空间分配.....	191
表 9-2 内置 SRAM (SRAM) 寄存器偏移地址.....	193
表 10-1 32-bit 访问时 PORT 寄存器一览.....	206
表 10-2 通用 IO (GPIO) 寄存器偏移地址.....	206
表 10-3 Event Port 寄存器偏移地址.....	207
表 11-1 外部中断输入管脚.....	223
表 11-2 中断向量表.....	224
表 11-4 中断控制器 (INTC) 寄存器偏移地址.....	243
表 12-1 AOS 目标列表.....	256
表 12-2 自动运行系统 (AOS) 寄存器偏移地址.....	259
表 13-1 MPU 模块简介.....	277
表 13-2 存储保护单元 (MPU) 寄存器偏移地址.....	280
表 14-1 KEYSKAN 管脚说明.....	297
表 14-2 键盘扫描控制模块 (KEYSCAN) 寄存器偏移地址.....	299
表 15-1 HRC 目标频率为 20MHz 时的测量误差.....	304
表 15-2 HRC 目标频率为 16MHz 时的测量误差.....	306
表 15-3 内部时钟校准器 (CTC) 寄存器偏移地址.....	311
表 16-1 通道重置说明.....	322
表 16-2 DMA 控制器 (DMA) 寄存器偏移地址.....	328
表 17-1 CMP 引脚一览表.....	358

表 17-2 CMP 正端输入一览表.....	359
表 17-3 CMP 负端输入一览表.....	359
表 17-4 电压比较器 (CMP) 寄存器偏移地址.....	364
表 17-15 定时器窗口 PWM 一览表.....	369
表 18-1 各 ADC 单元规格.....	375
表 18-2 序列 A 和 B 的各种竞争.....	381
表 18-3 ADC 转换时间.....	383
表 18-4 模数转换模块 (ADC) 寄存器偏移地址.....	390
表 18-30 不同的 ADC 单元 CHx 映射关系表.....	405
表 19-1 DAC 引脚.....	418
表 19-2 数模转换器 (DAC) 寄存器偏移地址.....	421
表 19-11 DAC 转换和模拟输出控制.....	424
表 20-1 Timer6 的基本功能及特性.....	427
表 20-2 Timer6 端口列表.....	428
表 20-3 不同模式下的功能对比表.....	450
表 20-4 高级控制定时器 (Timer6) 寄存器偏移地址.....	458
表 20-5 Timer6 公共寄存器偏移地址.....	459
表 20-62 计数器 (CNTNR) 控制优先级.....	501
表 20-63 PWMA 端口输出控制优先级.....	501
表 20-64 PWMB 端口输出控制优先级.....	502
表 21-1 Timer4 的基本功能及特性.....	503
表 21-2 Timer4 端口列表.....	504
表 21-3 通用控制定时器 (Timer4) 寄存器偏移地址.....	522
表 21-19 OCCR*h 计数匹配一览表.....	533
表 21-22 OCCR*h/l 计数匹配一览表.....	535
表 22-1 EMB 端口分配.....	555
表 22-2 EMB Group 对照表.....	556
表 22-3 紧急刹车模块 (EMB) 寄存器偏移地址.....	559
表 23-1 TimerA 的基本功能及特性.....	571
表 23-2 TimerA 端口列表.....	572
表 23-3 内部触发事件 TRGSEL 选择关系对应表.....	574
表 23-4 通用定时器 (TimerA) 寄存器偏移地址.....	583
表 24-1 通用定时器 (Timer0) 寄存器偏移地址.....	610
表 24-2 TMR_SYNENR 寄存器偏移地址.....	610
表 25-1 RTC 主要特性.....	619
表 25-2 实时时钟 (RTC) 寄存器偏移地址.....	623
表 26-1 看门狗计数器的基本特性.....	636
表 26-2 看门狗计数器 (WDT) 寄存器偏移地址.....	641
表 26-3 看门狗计数器 (SWDT) 寄存器偏移地址.....	641
表 27-2 USART 管脚说明.....	647
表 27-3 UART 接收器的容差.....	651
表 27-5 UART 中断/事件表.....	655
表 27-6 多处理器模式中断/事件表.....	658
表 27-7 LIN 中断/事件表.....	660
表 27-8 智能卡模式中断/事件表.....	663
表 27-9 时钟同步模式中断/事件表.....	667
表 27-10 USART 整体中断一览表.....	667
表 27-11 USART1 滤波控制寄存器偏移地址.....	669
表 27-12 通用同步异步收发器 (USART) 寄存器偏移地址.....	669
表 27-21 波特率计算公式 (小数波特率无效 FBME=0)	675
表 27-22 波特率计算公式 (小数波特率有效 FBME=1)	675
表 28-1 输入/输出引脚.....	689

表 28-2 中断一览表.....	700
表 28-3 事件信号输出一览表.....	701
表 28-4 集成电路总线 (I2C) 寄存器偏移地址.....	703
表 29-1 SPI 主要特性.....	725
表 29-2 管脚说明.....	726
表 29-3 主机模式时 SPI 管脚状态说明.....	727
表 29-4 从机模式时 SPI 管脚状态说明.....	727
表 29-5 部分设定值的位速.....	729
表 29-6 SPI 模式和寄存器设定关系.....	735
表 29-7 错误检测对应表.....	740
表 29-8 SPI 中断源和触发源说明.....	744
表 29-9 串行外设接口 (SPI) 寄存器偏移地址.....	745
表 30-1 QSPI 主要规格.....	754
表 30-2 QSPI 管脚.....	757
表 30-3 参考指令一览表.....	761
表 30-4 QSIO2 和 QSIO3 的管脚状态.....	772
表 30-5 四线式串行外设接口 (QSPI) 寄存器偏移地址.....	774
表 31-1 AHB 访问宽度和存储器位宽对应访问方式表.....	783
表 31-2 SMC 协议接口.....	786
表 31-3 EXMC 端口功能分配.....	787
表 31-4 单次读动作基本设定例.....	789
表 31-5 地址数据线复用单次读动作基本设定例.....	794
表 31-6 单次写动作基本设定例.....	796
表 31-7 地址数据线复用异步方式单次写动作基本设定例.....	800
表 31-8 地址数据线复用同步方式单次写动作基本设定例.....	801
表 31-9 EXMC 公共寄存器偏移地址.....	803
表 31-10 外部存储器控制器 (EXMC) SMC 寄存器偏移地址.....	803
表 31-29 CS0 设定和访问地址对应关系表.....	809
表 32-1 MCAN 引脚说明.....	817
表 32-2 MCAN 配置寄存器列表.....	817
表 32-3 MCAN 中的 DLC 编码.....	819
表 32-4 接收缓冲器和 FIFO 元素大小.....	828
表 32-5 接收缓冲区过滤器配置示例.....	830
表 32-6 调试消息过滤器配置示例.....	830
表 32-7 发送帧的可能配置.....	831
表 32-8 发送缓冲区、FIFO (队列) 元素大小.....	832
表 32-9 接收缓冲区和 FIFO 元素.....	836
表 32-10 接收缓冲区和 FIFO 元素说明.....	836
表 32-11 发送缓冲区元素.....	838
表 32-12 发送缓冲区元素说明.....	838
表 32-13 发送事件 FIFO 元素.....	840
表 32-14 发送事件 FIFO 元素说明.....	841
表 32-15 标准消息 ID 过滤器元素.....	842
表 32-16 标准消息 ID 过滤器元素说明.....	842
表 32-17 扩展消息 ID 过滤器元素.....	843
表 32-18 扩展消息 ID 过滤器元素说明.....	843
表 32-19 控制器局域网 (MCAN) 寄存器偏移地址.....	845
表 33-1 AES 加解密运行时间.....	894
表 33-2 128 位操作寄存器示例.....	895
表 33-3 192 位操作寄存器示例.....	896
表 33-4 256 位操作寄存器示例.....	897
表 33-5 加解密算法处理器 (AES) 寄存器偏移地址.....	901

表 33-6 安全散列算法（HASH）寄存器偏移地址..... 901

表 33-7 真随机数发生器（TRNG）寄存器偏移地址..... 901

表 34-1 数据计算单元（DCU）寄存器偏移地址..... 915

表 35-1 CRC 运算（CRC）寄存器偏移地址.....926

表 36-1 SWJ 调试端口引脚.....931

表 36-2 灵活的 SWJ-DP 引脚分配..... 931

表 36-6 调试控制器（DBGC）寄存器偏移地址.....935

1 产品简介

HC32A448 系列是基于 32 位 ARM® Cortex®-M4 微控制器，最高工作频率 200MHz 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。

HC32A448 系列集成了高速片上存储器，包括最大 256KB 的 Flash，最大 68KB 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。

HC32A448 系列集成了丰富的外设功能。包括：3 个独立的 12-bit 2.5Msps ADC；2 个 12 位 DAC；4 个高速电压比较器（CMP）；2 个多功能 PWM Timer（Timer6）支持 4 路互补 PWM 输出；3 个电机 PWM Timer（Timer4）支持 24 路互补 PWM 输出；4 个 16-bit 通用 Timer（TimerA）及 1 个 32-bit 通用 Timer（TimerA）支持正交编码输入及 40 路 Duty 独立可设 PWM 输出；11 个串行通信接口（I2C/ USART/ SPI）；1 个 QSPI 接口；2 路 CAN FD 控制器（MCAN）；1 个外部扩展总线控制器，包括 EXMC（支持 SMC）控制器。

HC32A448 系列支持宽电压范围（1.8~3.6V），宽温度范围（-40~105℃）和各种低功耗模式。支持低功耗模式的快速唤醒，Power Down 模式唤醒最快至 25μs。

典型应用

HC32A448 系列提供 48pin、64pin、80pin 的 LQFP 封装，适用于汽车电子节点应用领域，例如座椅控制器、尾门控制器、空调控制面板、AVAS 低速报警器等应用。

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。关于芯片的规格，请参阅对应的《数据手册》。

2 存储器映射 (Memory Mapping)

2.1 存储器映射

该 MCU 支持 4GB 的线性地址空间，地址从 0x00000000 到 0xFFFFFFFF，其中包含程序和数据。详细存储器映射请参阅下表。

表 2-1 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	-	0xE0100000	0xFFFFFFFF	511MB	Reserved	-	自定义空间
	私有外设 外部总线	0xE00FF000	0xE00FFFFF	4KB	ROMTABLE	-	-
		0xE0042400	0xE00FEFFF	755KB	BLANK	-	-
		0xE0042000	0xE00423FF	1KB	DBGC	-	调试控制寄存器区域
		0xE0041000	0xE0041FFF	4KB	BLANK	-	-
		0xE0040000	0xE0040FFF	4KB	TPIU	-	-
		私有外设 内部总线	0xE000F000	0xE003FFFF	196KB	BLANK	-
	0xE000E000		0xE000EFFF	4KB	SCS	-	系统控制空间 NVIC/ MPU 等
	0xE0003000		0xE000DFFF	44KB	BLANK	-	-
	0xE0002000		0xE0002FFF	4KB	FPB	-	-
	0xE0001000		0xE0001FFF	4KB	DWT	-	-
	0xE0000000		0xE0000FFF	4KB	ITM	-	-
	外部设备	-	0xA0000000	0xDFFFFFFF	1024MB	Reserved	-
外部 存储	AHB5 时钟：HCLK	0x98000000	0x9FFFFFFF	128MB	QSPI	-	-
		0x88200000	0x97FFFFFFF	254MB	BLANK	-	-
	AHB5 时钟：EXCLK	0x88100000	0x881FFFFF	1MB	BLANK	-	-
		0x88000800	0x880FFFFF	1022KB	BLANK	-	-
		0x88000400	0x880007FF	1KB	BLANK	-	-
		0x88000000	0x880003FF	1KB	SMCR	-	-
		0x80000000	0x87FFFFFFF	128MB	BLANK	-	-
		0x60000000	0x7FFFFFFF	512MB	SMC	-	-
外设	-	0x44000000	0x5FFFFFFF	448MB	Reserved	-	-
		0x42000000	0x43FFFFFFF	32MB	PeriBitBand	-	-
		0x40100000	0x41FFFFFFF	31MB	Reserved	-	-
		0x400C0000	0x400FFFFF	256KB	BLANK	-	-
	AHB2 时钟：PCLK0	0x40080000	0x400BFFFF	256KB	BLANK	-	-
		0x40078800	0x4007FFFF	30KB	BLANK	-	-
		0x40078400	0x400787FF	1KB	BLANK	-	-
		0x40078000	0x400783FF	1KB	BLANK	-	-
	AHB3 时钟：PCLK0	0x40071800	0x40077FFF	26KB	BLANK	-	-
		0x40071400	0x400717FF	1KB	BLANK	-	-
		0x40071000	0x400713FF	1KB	BLANK	-	-
		0x40060000	0x40070FFF	68KB	BLANK	-	-
	AHB1 时钟：HCLK	0x40059000	0x4005FFFF	28KB	BLANK	-	-
		0x40058C00	0x40058FFF	1KB	BLANK	-	-
		0x40058800	0x40058BFF	1KB	BLANK	-	-
		0x40058400	0x400587FF	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	AHB1 时钟: HCLK	0x40058000	0x400583FF	1KB	BLANK	-	-
		0x40057C00	0x40057FFF	1KB	BLANK	-	-
		0x40057800	0x40057BFF	1KB	BLANK	-	-
		0x40057400	0x400577FF	1KB	BLANK	-	-
		0x40057000	0x400573FF	1KB	BLANK	-	-
		0x40056C00	0x40056FFF	1KB	DCU_4	-	-
		0x40056800	0x40056BFF	1KB	DCU_3	-	-
		0x40056400	0x400567FF	1KB	DCU_2	-	-
		0x40056000	0x400563FF	1KB	DCU_1	-	-
		0x40055C00	0x40055FFF	1KB	BLANK	-	-
		0x40055800	0x40055BFF	1KB	BLANK	-	-
		0x40055400	0x400557FF	1KB	PERIC	-	周边模块控制寄存器
		0x40055000	0x400553FF	1KB	BLANK	-	-
		0x40054400	0x40054FFF	3KB	BLANK	-	禁止访问
		0x40054000	0x400543FF	1KB	CMU	带保护	-
		0x40053800	0x40053FFF	2KB	GPIO	-	-
		0x40053400	0x400537FF	1KB	DMA_2	-	-
		0x40053000	0x400533FF	1KB	DMA_1	-	-
		0x40052000	0x40052FFF	4KB	BLANK	-	-
		0x40051000	0x40051FFF	4KB	INTC	带保护	-
		0x40050C00	0x40050FFF	1KB	KEYSCAN	-	-
		0x40050800	0x40050BFF	1KB	RAMIF	带保护	-
		0x40050400	0x400507FF	1KB	BLANK	-	-
		0x40050000	0x400503FF	1KB	MPU	带保护	-
	APB4 时钟: PCLK3	0x4004FC00	0x4004FFFF	1KB	BLANK	-	-
		0x4004F800	0x4004FBFF	1KB	BLANK	-	-
		0x4004F400	0x4004F7FF	1KB	BLANK	-	-
		0x4004F000	0x4004F3FF	1KB	BLANK	-	-
		0x4004EC00	0x4004EFFF	1KB	BLANK	-	-
		0x4004E800	0x4004EBFF	1KB	BLANK	-	-
		0x4004E400	0x4004E7FF	1KB	BLANK	-	-
		0x4004E000	0x4004E3FF	1KB	BLANK	-	-
		0x4004D000	0x4004DFFF	1KB	BLANK	-	-
		0x4004CC00	0x4004CFFF	1KB	CMU	带保护	-
		0x4004C800	0x4004CBFF	1KB	BLANK	-	-
		0x4004C400	0x4004C7FF	1KB	WKTMR	带保护	-
		0x4004C000	0x4004C3FF	1KB	RTC	带保护	-
		0x4004AC00	0x4004BFFF	5KB	BLANK	-	-
		0x4004A800	0x4004ABFF	1KB	BLANK	-	-
		0x4004A400	0x4004A7FF	1KB	BLANK	-	-
		0x4004A000	0x4004A3FF	1KB	BLANK	-	-
		0x40049C00	0x40049FFF	1KB	CTC	-	-
		0x40049800	0x40049BFF	1KB	BLANK	-	-
		0x40049400	0x400497FF	1KB	SWDT	带保护	-
		0x40049000	0x400493FF	1KB	WDT	带保护	-
		0x40048800	0x40048FFF	2KB	BLANK	-	-

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	APB4 时钟: PCLK3	0x40048400	0x400487FF	1KB	FCM	-	-
		0x40048000	0x400483FF	1KB	FCG, CMU	带保护	-
	APB3 时钟: PCLK4	0x40042400	0x40047FFF	23KB	BLANK	-	-
		0x40042000	0x400423FF	1KB	TRNG	带保护	-
		0x40041C00	0x40041FFF	1KB	BLANK	-	-
		0x40041800	0x40041BFF	1KB	BLANK	-	-
		0x40041400	0x400417FF	1KB	BLANK	-	-
		0x40041000	0x400413FF	1KB	DAC	-	-
		0x40040C00	0x40040FFF	1KB	BLANK	-	-
		0x40040800	0x40040BFF	1KB	ADC_3	-	-
		0x40040400	0x400407FF	1KB	ADC_2	-	-
		0x40040000	0x400403FF	1KB	ADC_1	-	-
	APB5 时钟: PCLK0	0x4003C000	0x4003FFFF	16KB	Timer6_1, 2	-	-
		0x4003BC00	0x4003BFFF	1KB	BLANK	-	-
		0x4003B800	0x4003BBFF	1KB	I2C_2	-	-
		0x4003B400	0x4003B7FF	1KB	I2C_1	-	-
		0x4003B000	0x4003B3FF	1KB	BLANK	-	-
		0x4003AC00	0x4003AFFF	1KB	TimerA_4	-	-
		0x4003A800	0x4003ABFF	1KB	TimerA_3	-	-
		0x4003A400	0x4003A7FF	1KB	TimerA_2	-	-
		0x4003A000	0x4003A3FF	1KB	TimerA_1	-	-
		0x40038F00	0x40039FFF	4352B	BLANK	-	-
		0x40038E00	0x40038EFF	256B	Timer4_3	-	-
		0x40038C00	0x40038DFF	512B	CMP_3, 4	-	-
		0x40038A00	0x40038BFF	512B	BLANK	-	-
		0x40038800	0x400389FF	512B	CMP_1, 2	-	-
		0x40038400	0x400387FF	1KB	Timer4_2	-	-
		0x40038000	0x400383FF	1KB	Timer4_1	-	-
	APB2 时钟: PCLK1	0x4002B800	0x40037FFF	50KB	BLANK	-	-
		0x4002B000	0x4002B7FF	2KB	CAN_RAM	-	-
		0x40029800	0x4002AFFF	6KB	BLANK	-	-
		0x40029400	0x400297FF	1KB	MCAN_2	-	-
		0x40029000	0x400293FF	1KB	MCAN_1	-	-
		0x40028800	0x40028FFF	2KB	BLANK	-	-
		0x40028400	0x400287FF	1KB	BLANK	-	-
		0x40028000	0x400283FF	1KB	BLANK	-	-
		0x40027C00	0x40027FFF	1KB	BLANK	-	-
		0x40027800	0x40027BFF	1KB	BLANK	-	-
		0x40027400	0x400277FF	1KB	BLANK	-	-
		0x40027000	0x400273FF	1KB	BLANK	-	-
		0x40026C00	0x40026FFF	1KB	BLANK	-	-
		0x40026800	0x40026BFF	1KB	BLANK	-	-
		0x40026400	0x400267FF	1KB	BLANK	-	-
		0x40026000	0x400263FF	1KB	TimerA_5	-	-
		0x40025800	0x40025FFF	2KB	BLANK	-	-
		0x40025400	0x400257FF	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	APB2 时钟: PCLK1	0x40025000	0x400253FF	1KB	BLANK	-	-
		0x40024C00	0x40024FFF	1KB	BLANK	-	-
		0x40024800	0x40024BFF	1KB	BLANK	-	-
		0x40024400	0x400247FF	1KB	Timer0_2	-	-
		0x40024000	0x400243FF	1KB	Timer0_1	-	-
		0x40022800	0x40023FFF	6KB	BLANK	-	-
		0x40022400	0x400227FF	1KB	BLANK	-	-
		0x40022000	0x400223FF	1KB	BLANK	-	-
		0x40021C00	0x40021FFF	1KB	BLANK	-	-
		0x40021800	0x40021BFF	1KB	BLANK	-	-
		0x40021400	0x400217FF	1KB	USART_6	-	-
		0x40021000	0x400213FF	1KB	USART_5	-	-
		0x40020C00	0x40020FFF	1KB	USART_4	-	-
		0x40020800	0x40020BFF	1KB	BLANK	-	-
		0x40020400	0x400207FF	1KB	BLANK	-	-
		0x40020000	0x400203FF	1KB	SPI_3	-	-
	APB1 时钟: PCLK1	0x4001E800	0x4001FFFF	6KB	BLANK	-	-
		0x4001E400	0x4001E7FF	1KB	BLANK	-	-
		0x4001E000	0x4001E3FF	1KB	BLANK	-	-
		0x4001DC00	0x4001DFFF	1KB	BLANK	-	-
		0x4001D800	0x4001DBFF	1KB	BLANK	-	-
		0x4001D400	0x4001D7FF	1KB	USART_3	-	-
		0x4001D000	0x4001D3FF	1KB	USART_2	-	-
		0x4001CC00	0x4001CFFF	1KB	USART_1	-	-
		0x4001C800	0x4001CBFF	1KB	BLANK	-	-
		0x4001C400	0x4001C7FF	1KB	SPI_2	-	-
		0x4001C000	0x4001C3FF	1KB	SPI_1	-	-
		0x40018000	0x4001BFFF	16KB	BLANK	-	-
		0x40017C00	0x40017FFF	1KB	EMB	-	-
		0x40010C00	0x40017BFF	28KB	BLANK	-	-
		0x40010800	0x40010BFF	1KB	AOS	-	内部触发事件寄存器区域
		0x40010400	0x400107FF	1KB	EFM	带保护	-
		0x40010000	0x400103FF	1KB	BLANK	-	-
	AHB4 时钟: PCLK1	0x40009400	0x4000FFFF	27KB	BLANK	-	-
		0x40009000	0x400093FF	1KB	BLANK	-	-
		0x40008C00	0x40008FFF	1KB	CRC	带保护	-
		0x40008800	0x40008BFF	1KB	BLANK	-	-
		0x40008400	0x400087FF	1KB	HASH	带保护	-
		0x40008000	0x400083FF	1KB	AES	带保护	-
		0x40000000	0x40007FFF	32KB	Reserved	-	-
		0x24000000	0x3FFFFFFF	448MB	Reserved	-	-
SRAM	SRAM 时钟: HCLK	0x22000000	0x23FFFFFF	32MB	SRAMBitBand	-	-
		0x20100000	0x21FFFFFF	31MB	Reserved	-	-
		0x200F1000	0x200FFFFF	60KB	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
SRAM	SRAM 时钟: HCLK	0x200F0000	0x200F0FFF	4KB	Ret SRAM	带保护	ECC RAM
		0x20060000	0x200EFFFF	576KB	Reserved	-	-
		0x20040000	0x2005FFFF	128KB	Reserved	-	-
		0x20008000	0x2003FFFF	224KB	Reserved	-	-
		0x20000000	0x20007FFF	32KB	SRAM0	-	ECC RAM
CODE	SRAM 时钟: HCLK	0x1FFF8000	0x1FFFFFFF	32KB	SRAMH	-	-
		0x03004000	0x1FFF7FFF	471.94MB	BLANK	-	-
	OTP, Flash 时钟: HCLK	0x0300204C	0x03003FFF	8116B	BLANK	-	-
		0x03002040	0x0300204B	12B	数据安全保护	-	用于配置数据安全保护
		0x03002004	0x0300203F	60B	BLANK	-	-
		0x03002000	0x03002003	4B	引导交换	-	-
		0x03001000	0x03001FFF	4KB	BLANK	-	-
		0x03000A00	0x03000FFF	1.5KB	OTP	-	-
		0x03000000	0x030009FF	2.5KB	BLANK	-	-
		-	0x02100000	0x02FFFFFF	15M	BLANK	-
	REMAP 时钟: HCLK	0x02080000	0x020FFFFF	512KB	REMAP1	-	地址重映射区域 1
		0x02000000	0x0207FFFF	512KB	REMAP0	-	地址重映射区域 0
	-	0x00080000	0x01FFFFFF	31.5M	BLANK	-	-
	Flash 时钟: HCLK	0x00040000	0x0007FFFF	256KB	BLANK	-	-
		0x00000000	0x0003FFFF	256KB	Embedded Flash 0	-	-

 **说明**

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

2. 总线说明请参见《总线架构 (BUS) 》章节相关描述。

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。具体寄存器及说明请参见《存储保护单元 (MPU) 》章节相关描述。

2.2 外部空间映射

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

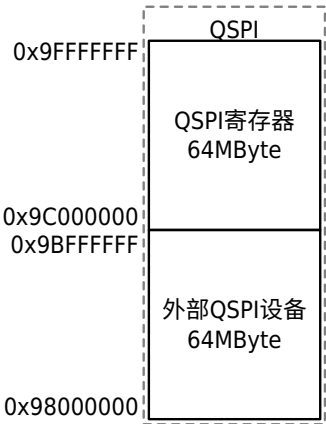


图 2-1 QSPI 地址空间分配

2.3 位段空间

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

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

2.4 地址重映射

本 MCU 提供了存储器地址重映射功能，有 2 个重映射地址供自由配置，分别使用 MMF_REMCRO 和 MMF_REMCRI 来配置重映射目标地址，目标地址可以是主闪存地址也可以是 SRAMH 地址。

- 重映射地址 0：
0x02000000~0x02080000（视设定重映射空间大小而定 MMF_REMCRO/1.RMSIZE）
- 重映射地址 1：
0x02080000~0x02100000（视设定重映射空间大小而定 MMF_REMCRO/1.RMSIZE）

当使用重映射功能有效（MMF_REMCRO.EN=1 或 MMF_REMCRI.EN=1）时，地址对应表如下：

表 2-2 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]=0b01110 情形 (重映射空间：16KB)	0x02000000~0x02003FFF	全 0	RMTADDR[16:2]	CPUADDR[13:0]
RMSIZE[4:0]=0b01111 情形 (重映射空间：32KB)	0x02000000~0x02007FFF	全 0	RMTADDR[16:3]	CPUADDR[14:0]
RMSIZE[4:0]=0b10000 情形 (重映射空间：64KB)	0x02080000~0x0208FFFF	全 0	RMTADDR[16:4]	CPUADDR[15:0]
RMSIZE[4:0]=0b10001 情形 (重映射空间：128KB)	0x02080000~0x0209FFFF	全 0	RMTADDR[16:5]	CPUADDR[16:0]

举例：

- 使用重映射地址 0 功能，设定源地址为主闪存 FLASH 地址 0x00008000，重映射空间 32K，寄存器 MMF_REMCRO 需设定为 0x8000800F。
- 使用重映射地址 1 功能，设定源地址为高速 SRAM 地址 0x1FFF8000，重映射空间为 16K，寄存器 MMF_REMCRI 需设定为 0x9FFF800E。

 **注意**

源地址的起始地址应设为重映射空间的整数倍。

2.5 寄存器

2.5.1 寄存器总表

基地址: 0x40010500

表 2-3 存储器映射 (Memory Mapping) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x0000	MMF_REMPRT	访问保护寄存器	32
0x0004+0x4*x	MMF_REMCrx	重映射寄存器 x (x=0~1)	32

2.5.2 访问保护寄存器 (MMF_REMPRT)

Offset	Bit Position																															
0x0000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																MMF_REMPRT															
Access																																

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	MMF_REMPRT	保护寄存器	寄存器 MMF_REMCRO 和 MMF_REMCR1 写保护：先对 MMF_REMPRT[15:0]写入 0x0123 接着写入 0x3210 解除保护。  说明 ● 寄存器 MMF_REMCRO 和 MMF_REMCR1 写保护状态时，读寄存器为 0； ● 寄存器 MMF_REMCRO 和 MMF_REMCR1 解除写保护状态时，读寄存器为 1。	RW

2.5.3 重映射寄存器 x (MMF_REMCrx) (x=0~1)

偏移地址 (Offset) : 0x0004+0x4*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name		Reserved	RMTADDR																		Reserved								RMSIZE			
Access	RW		RW																										RW			

位/位域	标记	位名	功能描述	读写
31	EN	重映射有效位	0b0: 重映射无效 0b1: 重映射有效	RW
30:29	Reserved	保留	-	-
28:12	RMTADDR	源地址	有效位数和 RMSIZE[4:0]设定有关。 设定可参考 地址重映射 。	RW
11:5	Reserved	保留	-	-
4:0	RMSIZE	重映射空间大小	设定重映射区域的大小： 0b01100: 4KB 0b01101: 8KB 0b01110: 16KB 0b01111: 32KB 0b10000: 64KB 0b10001: 128KB 0b10010: 256KB (128KB 产品禁止设置) - 其他: 禁止设置	RW

3 总线架构 (BUS)

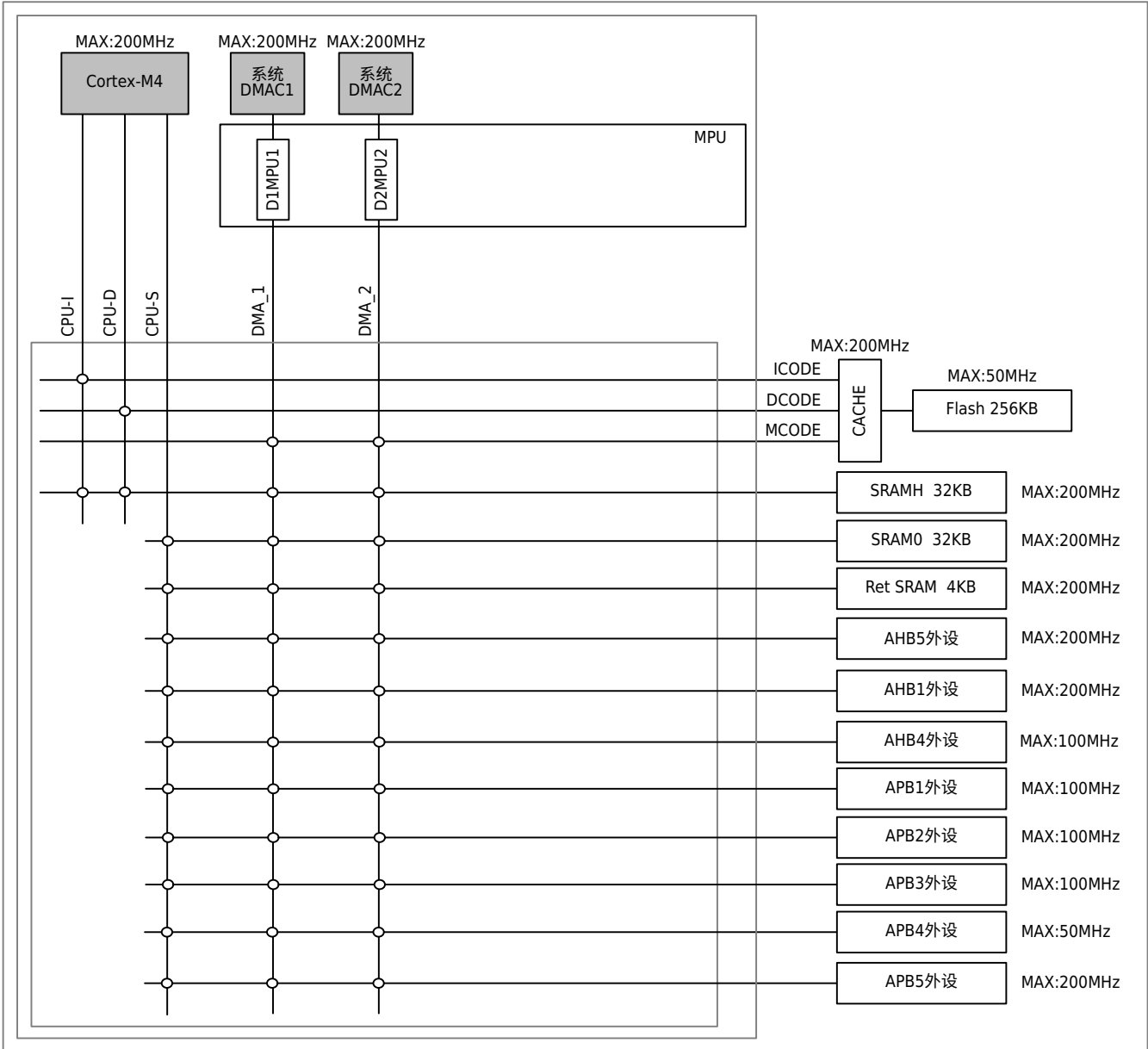
3.1 简介

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

- 主机总线
 - ▶ Cortex-M4 内核 CPU-I 总线，CPU-D 总线，CPU-S 总线
 - ▶ 系统 DMA_1 总线，系统 DMA_2 总线
- 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 32KB）总线
 - ▶ 系统 SRAM（SRAM0 32KB）总线
 - ▶ 系统 SRAM（Ret SRAM 4KB）总线
 - ▶ APB1 外设总线（AOS/EFM/EMB/SPI/USART）
 - ▶ APB2 外设总线（TimerA/Timer0/SPI/USART/MCAN）
 - ▶ APB3 外设总线（ADC/DAC/TRNG）
 - ▶ APB4 外设总线（FCM/WDT/SWDT/PWC/CTC/RTC/WKTM）
 - ▶ APB5 外设总线（Timer4/TimerA/Timer6/CMP/I2C）
 - ▶ AHB1 外设总线（DCU/CMU/GPIO/DMA/INTC/KEYSCAN/MPU）
 - ▶ AHB4 外设总线（AES/HASH/CRC）
 - ▶ AHB5 外设总线（SMC/SMCR/QSPI）

借助总线矩阵，可以实现主机总线到从机总线高效率的并发访问。

3.2 总线架构



总线矩阵

图 3-1 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_1 总线, DMA_2 总线

系统 DMA_1/系统 DMA_2 专用总线, DMA_1/DMA_2 通过此总线访问数据存储器 and 外设, 访问对象为 Flash/SRAMH/SRAM0/Ret SRAM/所有外设以及外部扩展空间。

3.3 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, CPU-S 访问 APB 外设, DMA_1 访问 SRAM0, DMA_2 访问 Ret SRAM, 这些访问可以同时进行。

4 复位控制 (RMU)

4.1 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)

4.2 功能说明

4.2.1 复位方式和复位标志位

复位方式及产生条件如下表所示。

表 4-1 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

复位方式	产生条件
RAM ECC 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, 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, 复位标志位如下表所示。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 4-2 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 1 复位	电压检测 2 复位	看门狗复位	专用看门狗复位	掉电唤醒复位	软件复位	MPU 错误复位	RAM 奇偶错误复位	RAM ECC 复位	时钟频率异常复位	外部高速振荡器异常停振复位
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)	×	×	×	-	-	-	-	×	-	-	-	-	-	✓

复位标志	复位方式													
	上电复位	引脚复位	欠压复位	电压检测 1 复位	电压检测 2 复位	看门狗复位	专用看门狗复位	掉电唤醒复位	软件复位	MPU 错误复位	RAM 奇偶错误复位	RAM ECC 复位	时钟频率异常复位	外部高速振荡器异常停振复位
Cortex-M4 Lockup 复位 (RMU_RSTF0.LKUPRF)	x	x	x	-	-	-	-	x	-	-	-	-	-	-

说明

“√”表示置位；“x”表示清零；“-”表示不变。

4.2.2 复位时序

4.2.2.1 上电复位

上电复位是上电复位电路引起的内部复位，时序如下所示。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见《电源控制 (PWC) 》章节的“[上电复位/掉电复位动作说明](#)”。

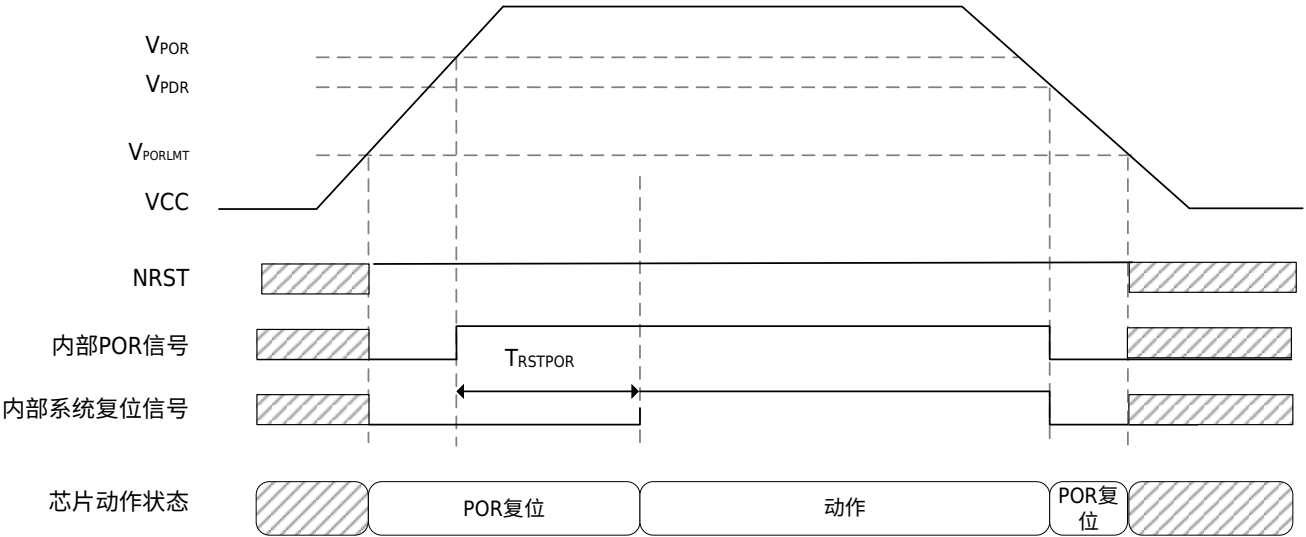


图 4-1 上电复位时序图

4.2.2.2 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，复位时序如下所示。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。

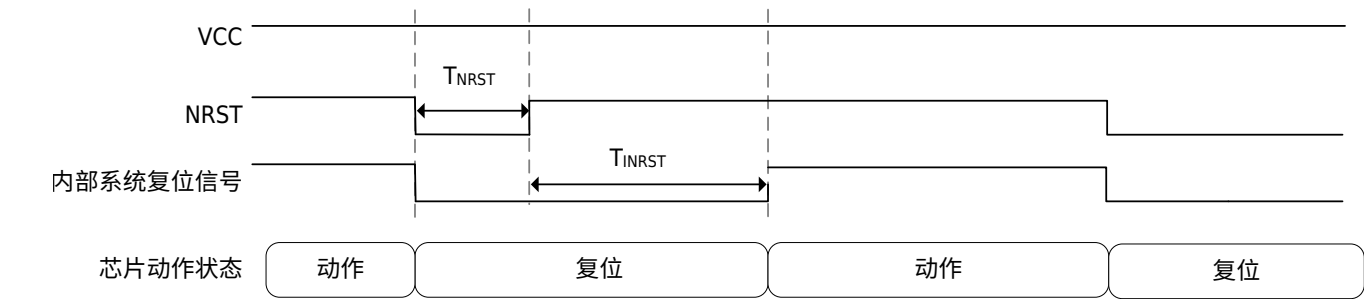


图 4-2 NRST 复位时序图

4.2.2.3 欠压复位

欠压复位是电压监测电路引起的内部复位，时序如下所示。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} ， $RMU\_RSTF0.BORF$ 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见《电源控制 (PWC) 》章节的“[欠压复位 \(BOR\) 说明](#)”。

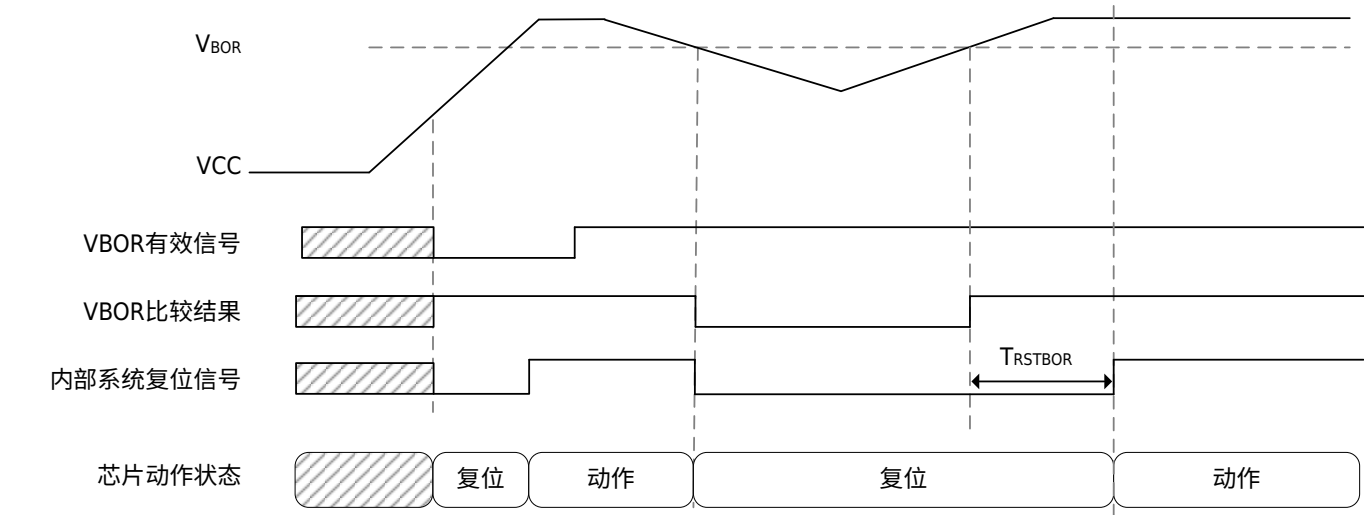


图 4-3 欠压复位时序图

4.2.2.4 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， $RMU\_RSTF0.PVD1RF$ 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。

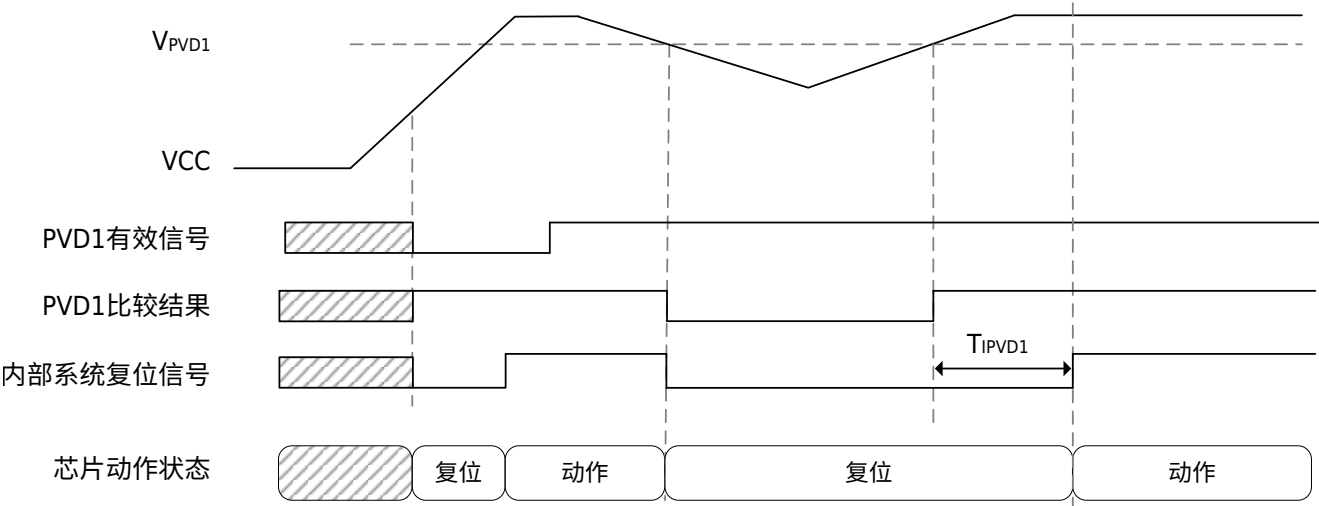


图 4-4 可编程电压检测 1 复位时序图

当 V_{CC} 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, $RMU\_RSTF0.PVD2RF$ 置位。当 V_{CC} 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。

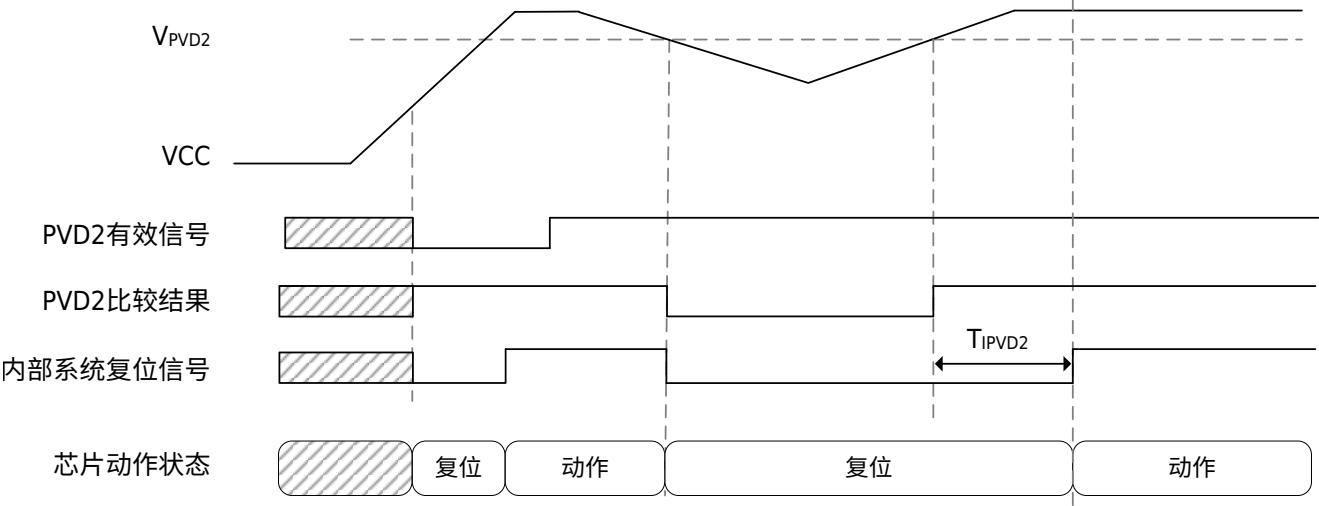


图 4-5 可编程电压检测 2 复位时序图

可编程电压检测 1、可编程电压检测 2 的复位设定请参见《电源控制 (PWC)》章节的“[可编程电压检测 1 \(PVD1\)](#)”、[可编程电压检测 2 \(PVD2\)](#)”。

4.2.2.5 看门狗复位、专用看门狗复位

看门狗复位是看门狗定时器产生的内部复位, 专用看门狗复位是专用看门狗定时器产生的内部复位, 复位时序如下所示。

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见《看门狗计数器 (WDT/SWDT)》章节相关描述。

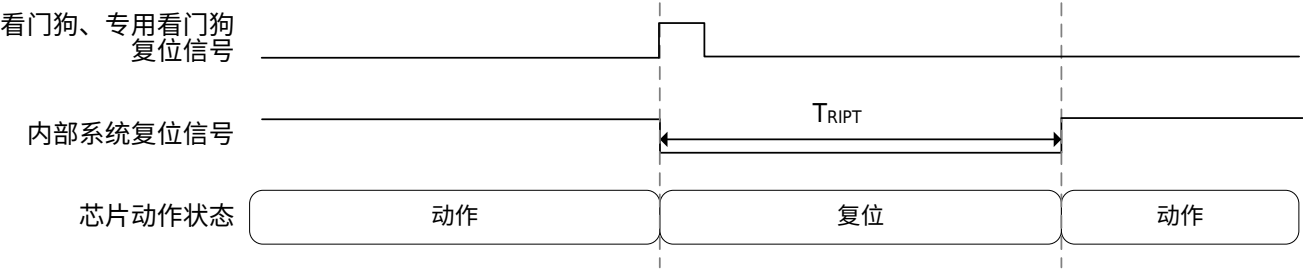


图 4-6 看门狗和专用看门狗复位时序图

4.2.2.6 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，时序如下所示。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，请参见《电源控制 (PWC) 》章节的“掉电模式”。

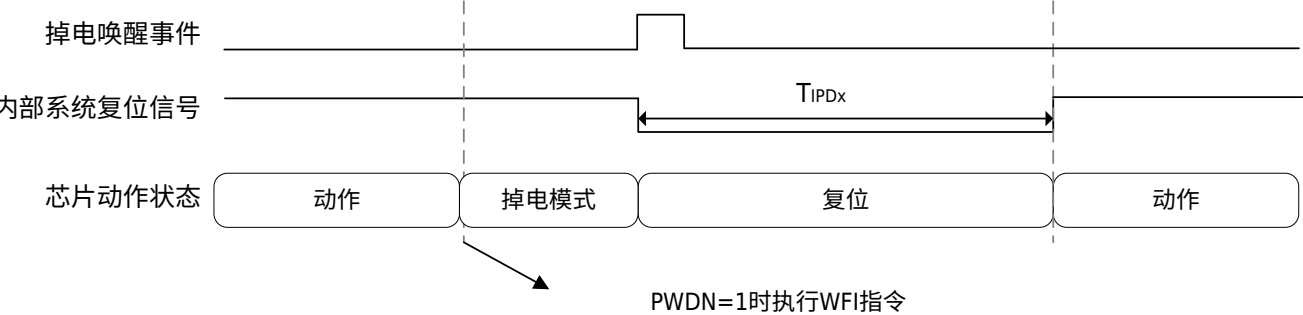


图 4-7 掉电唤醒复位时序图

4.2.2.7 软件复位

通过写 ARM 寄存器 AIRCR.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。

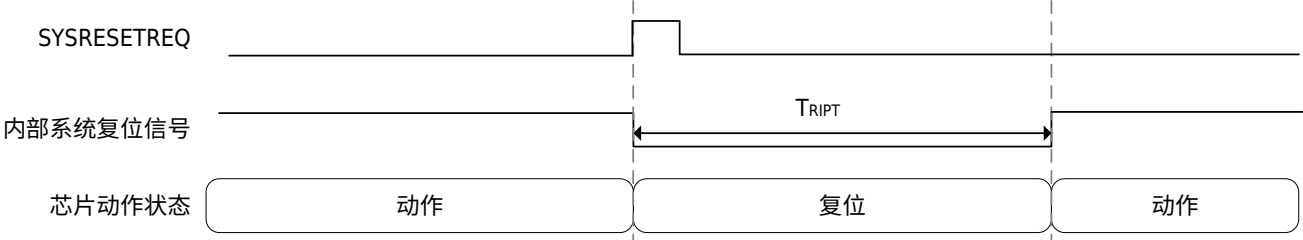


图 4-8 软件复位时序图

4.2.2.8 MPU 错误复位

MPU 错误复位将 RMU_RSTF0.MPUERF 置位，时序如下所示。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，请参见《存储保护单元 (MPU) 》章节相关描述。

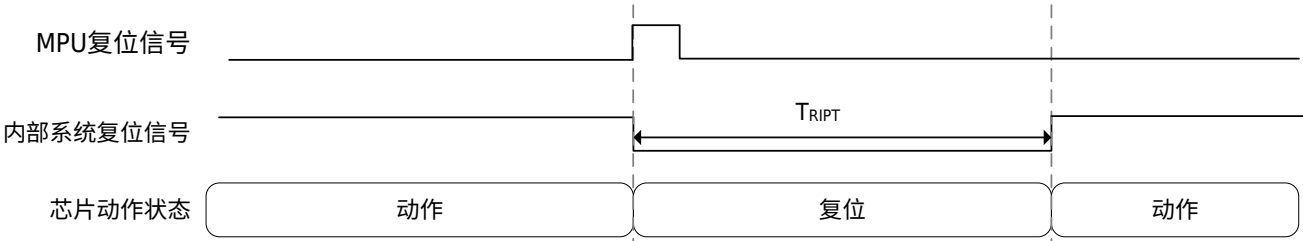


图 4-9 MPU 错误复位时序图

4.2.2.9 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，其复位时序如下所示。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，请参见《内置 SRAM (SRAM) 》章节相关描述。

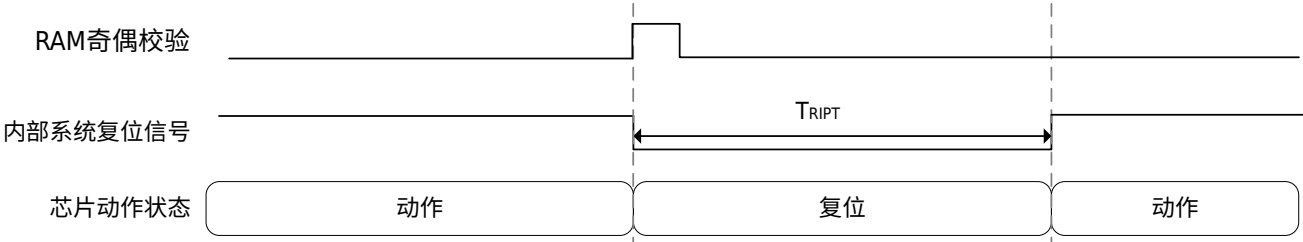


图 4-10 RAM 奇偶校验复位时序图

4.2.2.10 RAM ECC 复位

RAM ECC 校验发生错误时，可产生 RAM ECC 复位，其复位时序如下所示。RAM ECC 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。
RAM ECC 复位的设定，请参见《内置 SRAM (SRAM) 》章节相关描述。

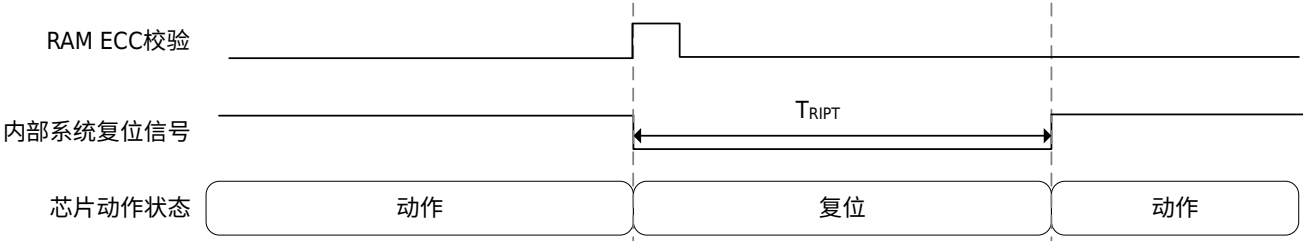


图 4-11 RAM ECC 复位时序图

4.2.2.11 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，其复位时序如下所示。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，请参见《时钟控制器 (CMU) 》章节的“[时钟频率测量](#)”。

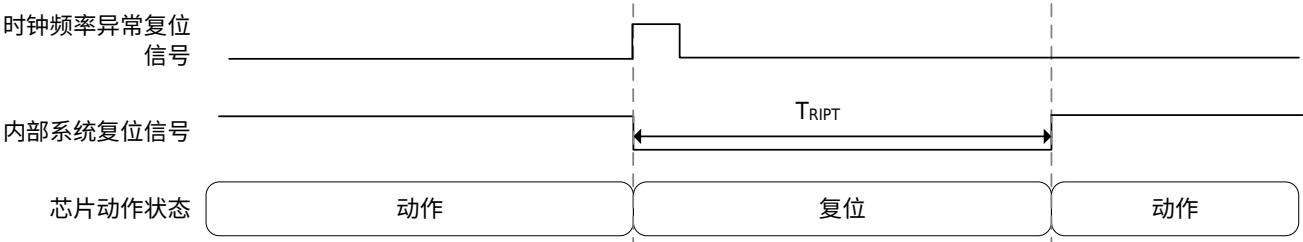


图 4-12 时钟频率异常复位时序图

4.2.2.12 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，请参见《时钟控制器 (CMU)》章节的“外部高速振荡器故障检测”。

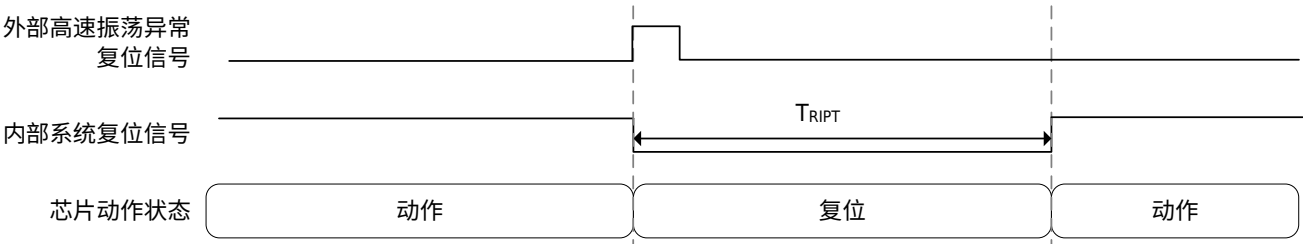


图 4-13 外部高速振荡异常复位时序图

4.2.2.13 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，并在几个时钟周期延时之后复位整个芯片，时序如下所示。Cortex-M4 Lockup 复位将 RMU_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。

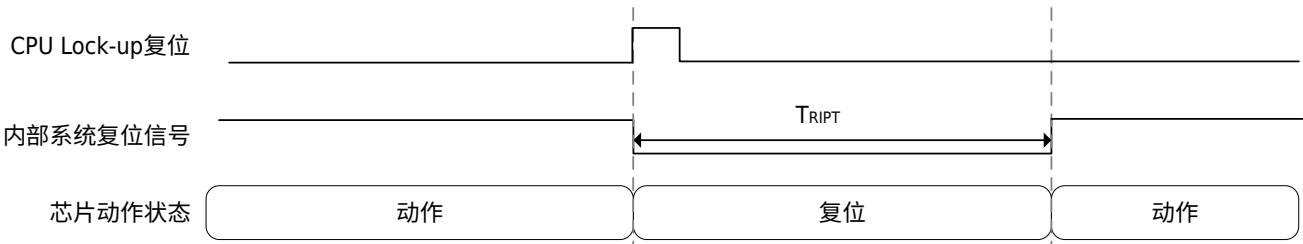


图 4-14 Cortex-M4 Lockup 复位时序图

4.2.3 复位方式判断

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.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。

4.2.4 模块功能复位

除上述多种芯片复位方式以外，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。

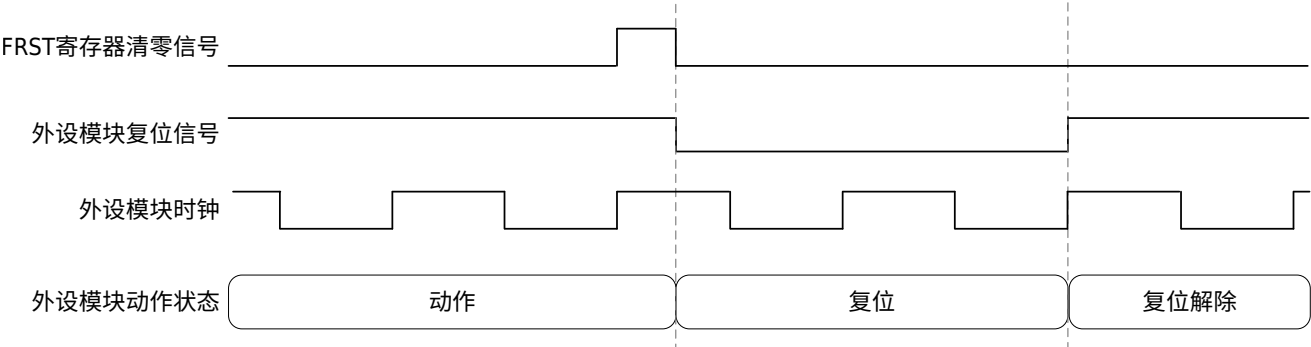


图 4-15 模块功能复位

4.2.5 复位条件

表 4-3 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 SRAM0_ECCERRADDR SRAMB_ECCERRADDR	1. 上电复位 2. 掉电唤醒复位
	上記以外的其他寄存器	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 PWC_PWRC1.PDTS CMU_XTALCFGR	下记复位以外的所有复位源： ● 掉电模式 1 唤醒复位 ● 掉电模式 2 唤醒复位 ● 掉电模式 4 唤醒复位
	PWC_PDWKFO PWC_PDWKF1	掉电唤醒复位以外的所有复位源
	PWC_PVDLCR PWC_PVDCCR1 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
电源控制 (PWC) 时钟控制 (CMU)	PWC_PVDICR.PVD1EDGS	1. 上电复位
	PWC_PVDDSR.PVD1DETFLG	2. 引脚复位
	PWC_PVDICR.PVD2EDGS	3. 欠压复位
	PWC_PVDDSR.PVD2DETFLG	4. 看门狗复位
		5. 专用看门狗复位
		6. 掉电唤醒复位
	PWC_WKTCR	上电复位
	上記以外的其他寄存器	所有复位源
以上模块以外的寄存器		所有复位源

4.3 寄存器

4.3.1 寄存器总表

基地址：0x4004CCE0

表 4-4 复位控制 (RMU) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	RMU_FRST0	功能复位控制寄存器 0	32
0x04	RMU_FRST1	功能复位控制寄存器 1	32
0x08	RMU_FRST2	功能复位控制寄存器 2	32
0x0C	RMU_FRST3	功能复位控制寄存器 3	32
0x10	RMU_PRSTCR0	复位控制寄存器 0	8
0x14	RMU_RSTF0	复位标志寄存器 0	32

4.3.2 功能复位控制寄存器 0 (RMU_FRST0)

Offset	Bit Position																																
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0xFFFFFFFF																																
Name	Reserved				DCU4	DCU3	DCU2	DCU1	CRC	TRNG	HASH	AES	Reserved	CTC	AOS	FCM	DMA2	DMA1	KEYSCAN	Reserved													
Access					RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	KEYSCAN													

位/位域	标记	位名	功能描述	读写
31:28	Reserved	保留	所有位读出为 1，写入时必须写 1	-
27	DCU4	DCU4 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
26	DCU3	DCU3 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
25	DCU2	DCU2 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
24	DCU1	DCU1 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
23	CRC	CRC 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
22	TRNG	TRNG 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
21	HASH	HASH 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
20	AES	AES 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
19	Reserved	保留	读出为 1，写入时必须写 1	-
18	CTC	CTC 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
17	AOS	AOS 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
16	FCM	FCM 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
15	DMA2	DMA2 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
14	DMA1	DMA1 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
13	KEYSCAN	KEYSCAN 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
12:0	Reserved	保留	所有位读出为 1，写入时必须写 1	-

4.3.3 功能复位控制寄存器 1 (RMU_FRST1)

Offset	Bit Position																																
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0xFFBFFFFF																																
Name	Reserved																	Reserved															
															SPI3	SPI2	SPI1															QSPI	Reserved
Access															RW	RW	RW															RW	

位/位域	标记	位名	功能描述	读写
31:23	Reserved	保留	所有位读出于 1，写入时必须写 1	-
22	Reserved	保留	读出于 0，写入时必须写 0	-
21:19	Reserved	保留	所有位读出于 1，写入时必须写 1	-
18	SPI3	SPI3 复位控制	● 0b0：功能复位有效 ● 0b1：功能复位解除	RW
17	SPI2	SPI2 复位控制	● 0b0：功能复位有效 ● 0b1：功能复位解除	RW
16	SPI1	SPI1 复位控制	● 0b0：功能复位有效 ● 0b1：功能复位解除	RW
15:4	Reserved	保留	所有位读出于 1，写入时必须写 1	-
3	QSPI	QSPI 复位控制	● 0b0：功能复位有效 ● 0b1：功能复位解除	RW
2:0	Reserved	保留	所有位读出于 1，写入时必须写 1	-

4.3.4 功能复位控制寄存器 2 (RMU_FRST2)

Offset	Bit Position																																		
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Reset	0xFFFFFFFF																																		
Name	Reserved												TMRA	Reserved			EMB	Reserved			TMR0	Reserved	TMR4	Reserved											
Access													RW				RW				RW		RW												

位/位域	标记	位名	功能描述	读写
31:21	Reserved	保留	所有位读出于 1，写入时必须写 1	-
20	TMRA	TimerA 复位控制	● 0b0：功能复位有效 ● 0b1：功能复位解除	RW
19:16	Reserved	保留	所有位读出于 1，写入时必须写 1	-
15	EMB	EMB 复位控制	● 0b0：功能复位有效 ● 0b1：功能复位解除	RW
14:13	Reserved	保留	所有位读出于 1，写入时必须写 1	-
12	TMR0	Timer0 复位控制	● 0b0：功能复位有效 ● 0b1：功能复位解除	RW
11	Reserved	保留	读出于 1，写入时必须写 1	-
10	TMR4	Timer4 复位控制	● 0b0：功能复位有效 ● 0b1：功能复位解除	RW
9:1	Reserved	保留	所有位读出于 1，写入时必须写 1	-
0	TMR6	Timer6 复位控制	● 0b0：功能复位有效 ● 0b1：功能复位解除	RW

 **说明**

TMR_SYNENR 不受 RMU_FRST 寄存器的控制。

4.3.5 功能复位控制寄存器 3 (RMU_FRST3)

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved						USART6	USART5	USART4	USART3	USART2	USART1	Reserved			SMC	Reserved						CMP34	CMP12	Reserved			DAC	Reserved	ADC3	ADC2	ADC1
Access							RW	RW	RW	RW	RW	RW				RW							RW	RW				RW		RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:26	Reserved	保留	所有位读出的 1，写入时必须写 1	-
25	USART6	USART6 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
24	USART5	USART5 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
23	USART4	USART4 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
22	USART3	USART3 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
21	USART2	USART2 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
20	USART1	USART1 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
19:17	Reserved	保留	所有位读出的 1，写入时必须写 1	-
16	SMC	SMC 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
15:10	Reserved	保留	所有位读出的 1，写入时必须写 1	-
9	CMP34	CMP 单元 3 和单元 4 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
8	CMP12	CMP 单元 1 和单元 2 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
7:5	Reserved	保留	所有位读出的 1，写入时必须写 1	-
4	DAC	DAC 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
3	Reserved	保留	读出的 1，写入时必须写 1	-
2	ADC3	ADC3 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
1	ADC2	ADC2 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW
0	ADC1	ADC1 复位控制	0b0: 功能复位有效 0b1: 功能复位解除	RW

4.3.6 复位控制寄存器 0 (RMU_PRSTCR0)

Offset	Bit Position							
0x10	7	6	5	4	3	2	1	0
Reset	0x40							
Name	Reserved		LKUPREN	Reserved				
Access			RW					

位/位域	标记	位名	功能描述	读写
7	Reserved	保留	-	-
6	Reserved	保留	读出为 1，写入时必须写 1	-
5	LKUPREN	Cortex-M4 Lockup 复位使能	0b0: Cortex-M4 Lockup 复位无效 0b1: Cortex-M4 Lockup 复位使能	RW
4:0	Reserved	保留	-	-

4.3.7 复位标志寄存器 0 (RMU_RSTF0)

Offset	Bit Position																																			
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reset	0xFFFFFFFF（根据复位方式不同，复位值不同）																																			
Name	CLRF	MULTIRF	Reserved																LKUPRF	XTALERF	CKFERF	RAECRF	RAPERF	MPUERF	SWRF	PDRF	SWDRF	WDRF	PVD2RF	PVD1RF	BORF	PINRF	PORF			
Access	RW	RW																	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31	CLRFR	清除复位标志	写 1 清零复位标志, 写 0 无效; 读出恒为 0  说明 置位动作必须在读取 RMU_RSTF0 之后进行。	RW
30	MULTIRF	2 个以上复位发生标志	发生两个及两个以上复位时, 由硬件置位。通过置位 CLRFR 清零。 读取: ● 0b0: 未发生两个及两个以上复位 ● 0b1: 发生两个及两个以上复位	RW
29:15	Reserved	保留	-	-
14	LKUPRF	Cortex-M4 Lockup 复位标志	发生 Cortex-M4 Lockup 复位时, 由硬件置位。通过置位 CLRFR 清零。 读取: ● 0b0: 未发生复位 ● 0b1: 发生复位	RW
13	XTALERF	外部高速振荡器异常停振复位标志	发生外部高速振荡器异常停振复位时, 由硬件置位。通过置位 CLRFR 清零。 读取: ● 0b0: 未发生复位 ● 0b1: 发生复位	RW
12	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 CLRFR 清零。 读取: ● 0b0: 未发生复位 ● 0b1: 发生复位	RW
11	RAECRF	RAM ECC 复位标志	发生 RAM ECC 复位时, 由硬件置位。通过置位 CLRFR 清零。 读取: ● 0b0: 未发生复位 ● 0b1: 发生复位	RW
10	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 CLRFR 清零。 读取: ● 0b0: 未发生复位 ● 0b1: 发生复位	RW

位/位域	标记	位名	功能描述	读写
9	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。 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW
8	SWRF	软件复位标志	发生软件复位时，由硬件置位。 通过置位 CLRf 清零。 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW
7	PDRF	掉电模式复位	发生掉电模式复位时，由硬件置位。 通过置位 CLRf 清零。 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW
6	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 CLRf 清零。 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW
5	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 CLRf 清零。 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW
4	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 CLRf 清零。 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW
3	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 CLRf 清零。 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW
2	BORF	欠压复位标志	发生欠压复位时，由硬件置位。 通过置位 CLRf 清零。 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW
1	PINRF	NRST 引脚复位标志	发生引脚复位时，由硬件置位。 通过置位 CLRf 清零。 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW

位/位域	标记	位名	功能描述	读写
0	PORF	上电复位标志	发生上电复位时，由硬件置位。 通过置位 CLRF 清零。 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW

5 时钟控制器 (CMU)

5.1 简介

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

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

AHB、APB 和 Cortex-M4 时钟都源自系统时钟。系统时钟的最大运行时钟频率可以达到 200MHz，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。

5.2 主要特性

表 5-1 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	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：1~4 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 4MHz~25MHz PLLH 倍频系数：25~150 倍 VCO 振荡频率：600MHz~1200MHz PLLHP 输出分频比：2~16 任意分频 PLLHQ 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

时钟源	规格
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

5.3 功能说明

5.3.1 系统框图

5.3.1.1 时钟系统框图

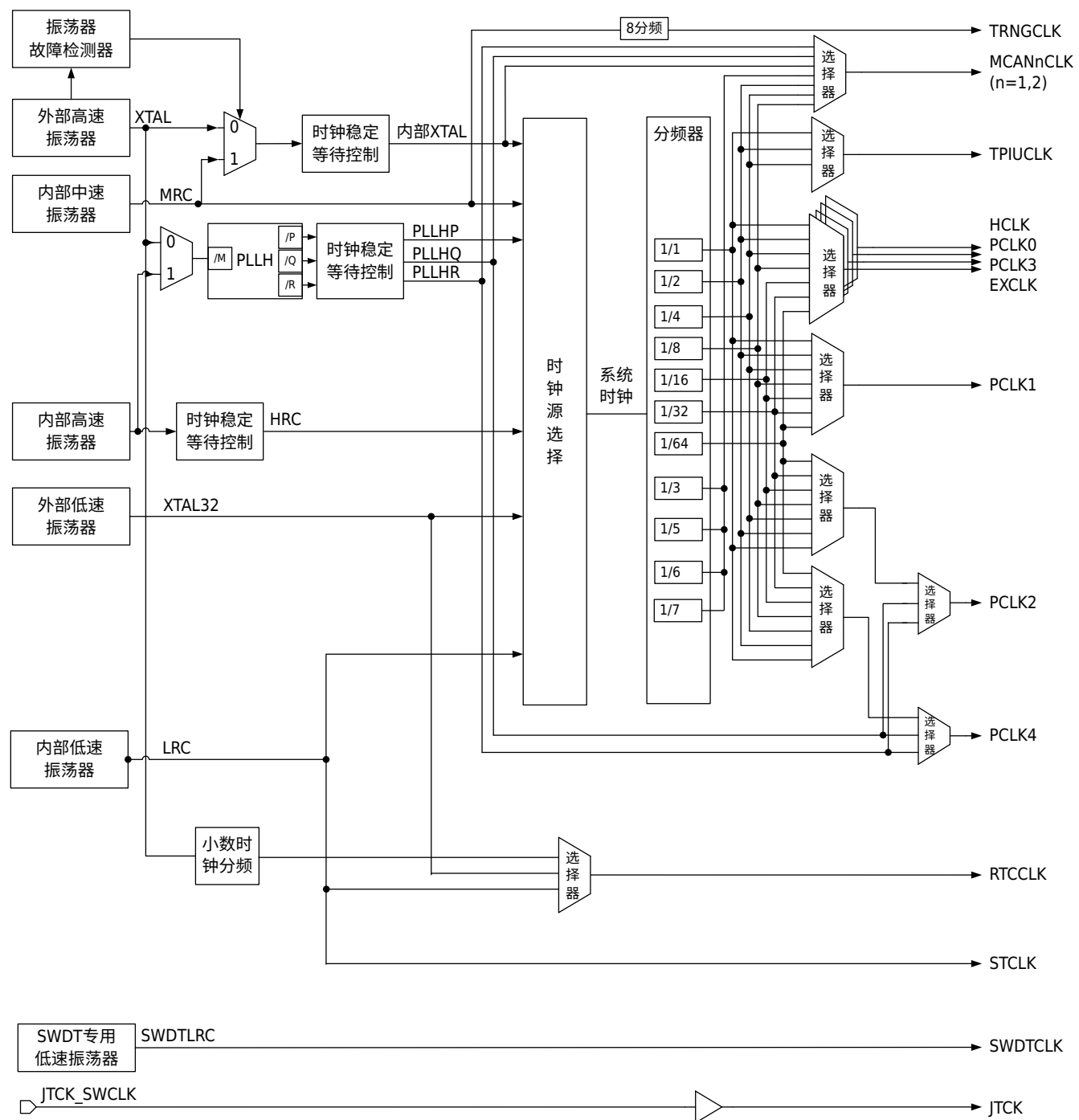


图 5-1 时钟系统框图

5.3.1.2 时钟频率测量框图

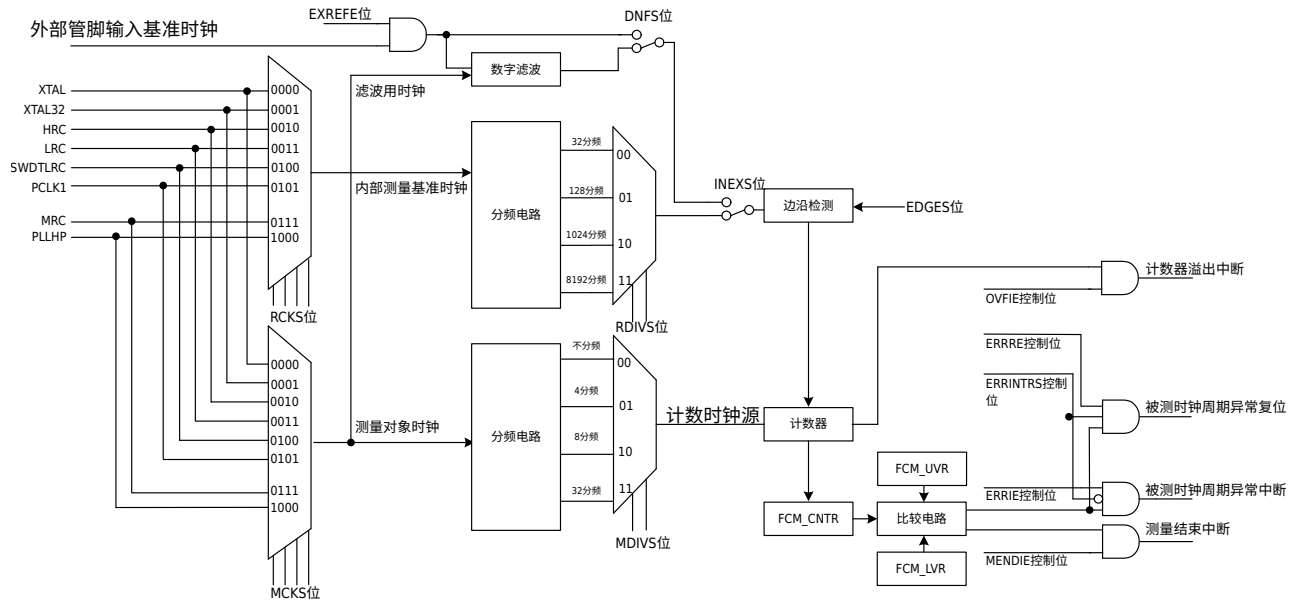


图 5-2 时钟频率测量框图

5.3.2 工作时钟规格

表 5-2 各个内部时钟的规格

时钟	作用范围	规格
HCLK	CPU、DMA _n (n=1~2)、EFM、SRAM0、SRAMH、Ret SRAM、MPU、GPIO、DCU _n (n=1~4)、INTC、QSPI	最高频率 200MHz。 由 CMU_SCFGR.HCLKS 位配置。 可选系统时钟的分频：1, 2, 4, 8, 16, 32, 64。
PCLK0	Timer6 _n (n=1~2)、Timer4 _n (n=1~3)、TimerA _n (n=1~4)、I2C _n (n=1~2)、CMP _n (n=1~4)	最高频率 200MHz。 由 CMU_SCFGR.PCLK0S 位配置。 可选系统时钟的分频：1, 2, 4, 8, 16, 32, 64。
PCLK1	USART _n (n=1~6)、SPI _n (n=1~3)、Timer0 _n (n=1~2)、TimerA_5、EMB、CRC、HASH、AES、MCAN _n (n=1~2) 控制逻辑	最高频率 100MHz。 由 CMU_SCFGR.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.PCLK2S 位和 CMU_PERICKSEL.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	最高频率 50MHz。 由 CMU_SCFGR.PCLK3S 位配置。 可选系统时钟的分频：1, 2, 4, 8, 16, 32, 64。

时钟	作用范围	规格
PCLK4	ADCn (n=1~3) (控制逻辑)、DAC (控制逻辑)、TRNG	最高频率 100MHz。 由 CMU_SCFGR.PCLK4S 位和 CMU_PERICKSEL.PERICKSEL 位配置。 可选系统时钟的分频：1, 2, 4, 8, 16, 32, 64。 可选独立时钟源：PLLHQ, PLLHR。
EXCLK	SMC	最高频率 60MHz。 由 CMU_SCFGR.EXCKS 位配置。 可选系统时钟的分频：1, 2, 4, 8, 16, 32, 64。
MCANnCLK	MCANn (n=1~2) 通信时钟	最高频率 80MHz。 由 CMU_CANCKCFGR.MCANnCKS (n=1~2) 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。
TPIUCLK	Cortex-M4 调试跟踪器用时钟	最高频率 50MHz。 由 CMU_TPIUCKCFGR.TPIUCKS 位配置。 可选系统时钟的分频：1, 2, 4。
TRNGCLK	TRNG 动作时钟	频率为 1MHz。 固定为 MRC 的 8 分频，使用 TRNG 功能时禁止停止 MRC 时钟。

各时钟之间需遵守下列规则：

1. HCLK 频率≥PCLK0、PCLK1、PCLK2、PCLK3、PCLK4 频率
2. PCLK0 频率≥PCLK1 频率，PCLK0 频率≥PCLK3 频率
3. PCLK2 频率:PCLK4 频率=1:8, 1:4, 1:2, 1:1, 2:1, 4:1, 8:1

5.3.3 晶振电路

5.3.3.1 外部高速振荡器

5.3.3.1.1 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.XTALSTP 位打开和关闭。

CMU_OSCSTBSR.XTALSTBF 位用来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。

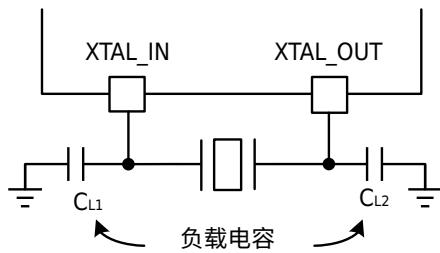


图 5-3 外部高速振荡器连接示例

5.3.3.1.2 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.XTALMS 位置 1 和 CMU_XTALCR.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_OUT/XTAL_EXT 引脚。此时 XTAL_IN 引脚可根据寄存器设定配置成 GPIO。

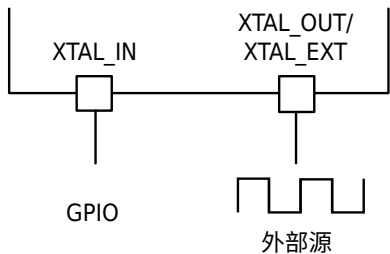


图 5-4 外部时钟输入的连接示例

5.3.3.2 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.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.XTALSTDE 位打开。

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

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

如果外部高速振荡器发生故障，动作波形如下图所示。操作流程请参见“检测到 XTAL 故障检测动作”。

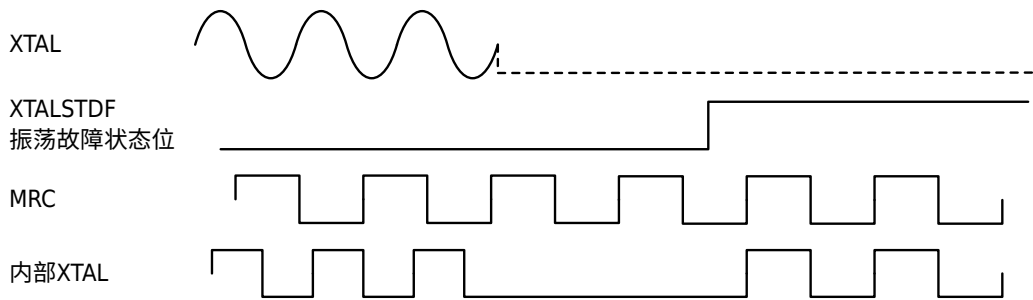


图 5-5 外部高速振荡器故障检测示例

5.3.3.2.1 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。详细信息请参见《紧急刹车模块 (EMB)》章节相关描述。

系统时钟选择成 XTAL，检测到 XTAL 故障时，动作例如下所示：

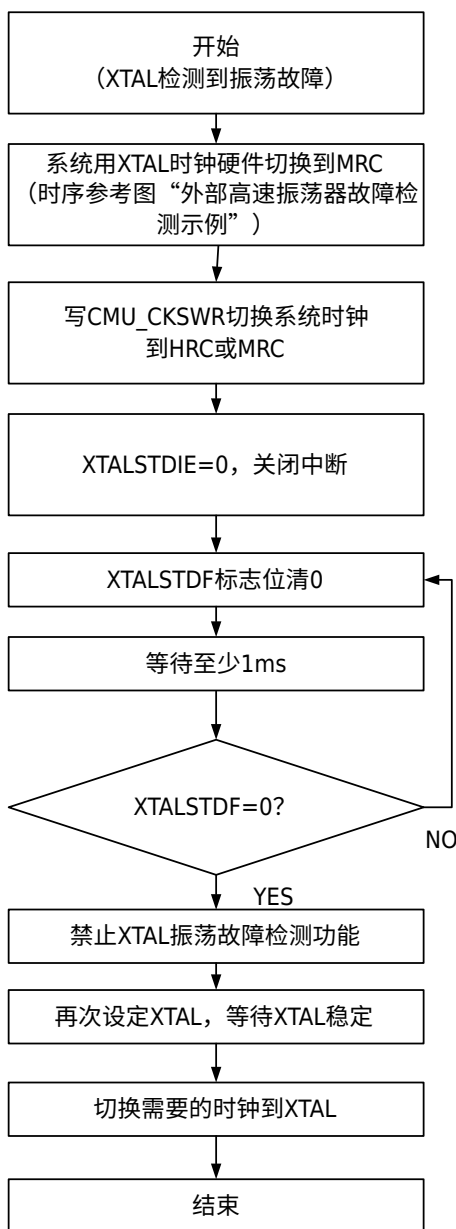


图 5-6 系统时钟选择 XTAL，检测到 XTAL 振荡故障示例

5.3.3.2.2 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，详细信息请参见《中断控制器 (INTC)》章节。

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，复位动作请参见《复位控制 (RMU)》章节。

5.3.3.3 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：

$$\text{分频数} = \frac{10000000\text{Hz}}{32768\text{Hz}} = \frac{78125 \text{ (10进制)}}{256 \text{ (10进制)}} = \frac{0x1312D \text{ (16进制)}}{0x100 \text{ (16进制)}}$$

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 0x1312D100，再通过 CMU_XTALDIVCR 设定开始。

外部高速振荡器 8MHz 分频到 32.768kHz 设定例：

$$\text{分频数} = \frac{8000000\text{Hz}}{32768\text{Hz}} = \frac{31250 \text{ (10进制)}}{128 \text{ (10进制)}} = \frac{0x7A12 \text{ (16进制)}}{0x80 \text{ (16进制)}}$$

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 0x07A12080，再通过 CMU_XTALDIVCR 设定开始。

5.3.3.4 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.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。

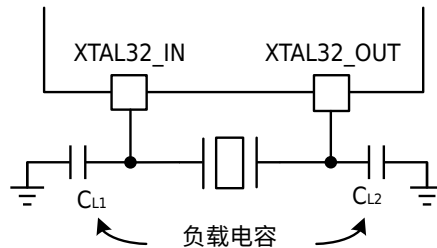


图 5-7 外部低速振荡器连接示例

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，稳定时间请参见《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.XTAL32STP 位设 1，关闭外部低速振荡器。

5.3.4 内部 RC 时钟

5.3.4.1 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 晶振快，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，保证精度请参见《数据手册》“电气特性-工作条件-内部时钟源特性-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 振荡器的精度。用户可通过寄存器对 HRC 频率进行微调。

CMU_OSCSTBSR.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.HRCSTP 位打开或关闭。

5.3.4.2 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，保证精度请参见《数据手册》“电气特性-工作条件-内部时钟源特性-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.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”。



注意

使用 TRNG 模块时禁止关闭 MRC 时钟。

5.3.4.3 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 和 KEYSCAN 使用。

LRC 振荡器的启动速度快。

频率校准

因为生产工艺不同，不同芯片的 RC 振荡器频率也不同，因此会对每个器件进行出厂校准，保证精度请参见《数据手册》“电气特性-工作条件-内部时钟源特性-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 振荡器的精度。用户可通过寄存器对 LRC 频率进行微调。

LRC 可通过 CMU_LRCCR.LRCSTP 位打开或关闭。

5.3.4.4 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，保证精度请参见《数据手册》“电气特性-工作条件-内部时钟源特性-SWDT 专用内部低速（SWDTLRC）振荡器”章节。

5.3.5 PLL 时钟

本产品配备了一个 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 200 MHz）
 - ▶ 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。
- PLLH 的分频系数 M、N、P、Q、R 可独立配置。



注意

1. 在 HRC 或 XTAL 振荡器稳定后，再对 PLL 进行配置。
2. 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。
3. 当进入掉电和停止模式后，PLL 将由硬件禁止。

5.3.6 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，切换步骤请参见“时钟源切换”。只有在目标时钟源已稳定的状态下，才可以从一个时钟源切换到另一个时钟源。

时钟切换时需要正确配置 Flash 的等待周期，防止系统时钟频率大于 Flash 的最大动作频率。详细配置请参见《嵌入式 Flash (EFM) 》的“CPU 时钟和 Flash 读取时间之间的关系”章节。

时钟源切换

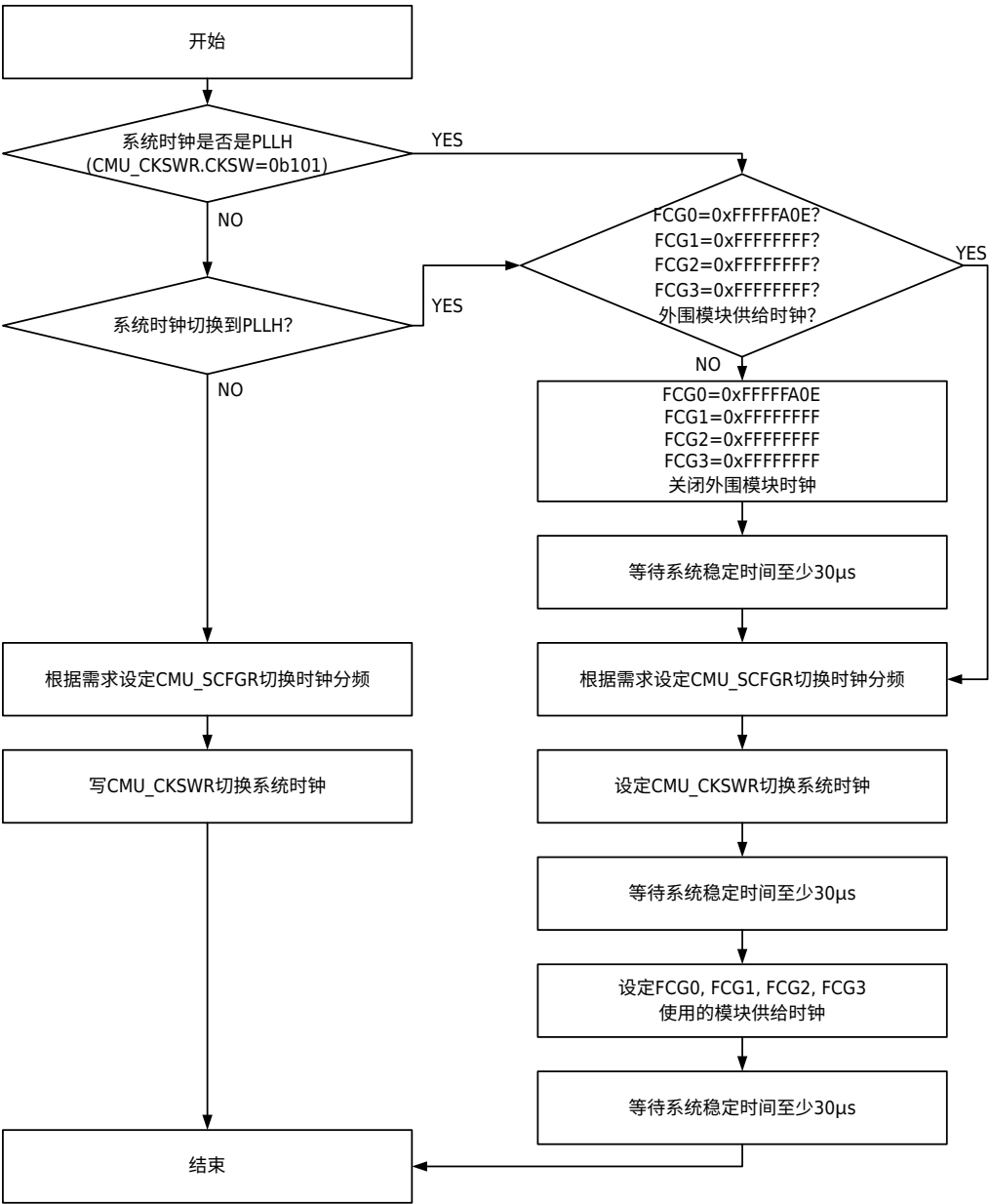


图 5-8 时钟源切换流程图

时钟分频切换

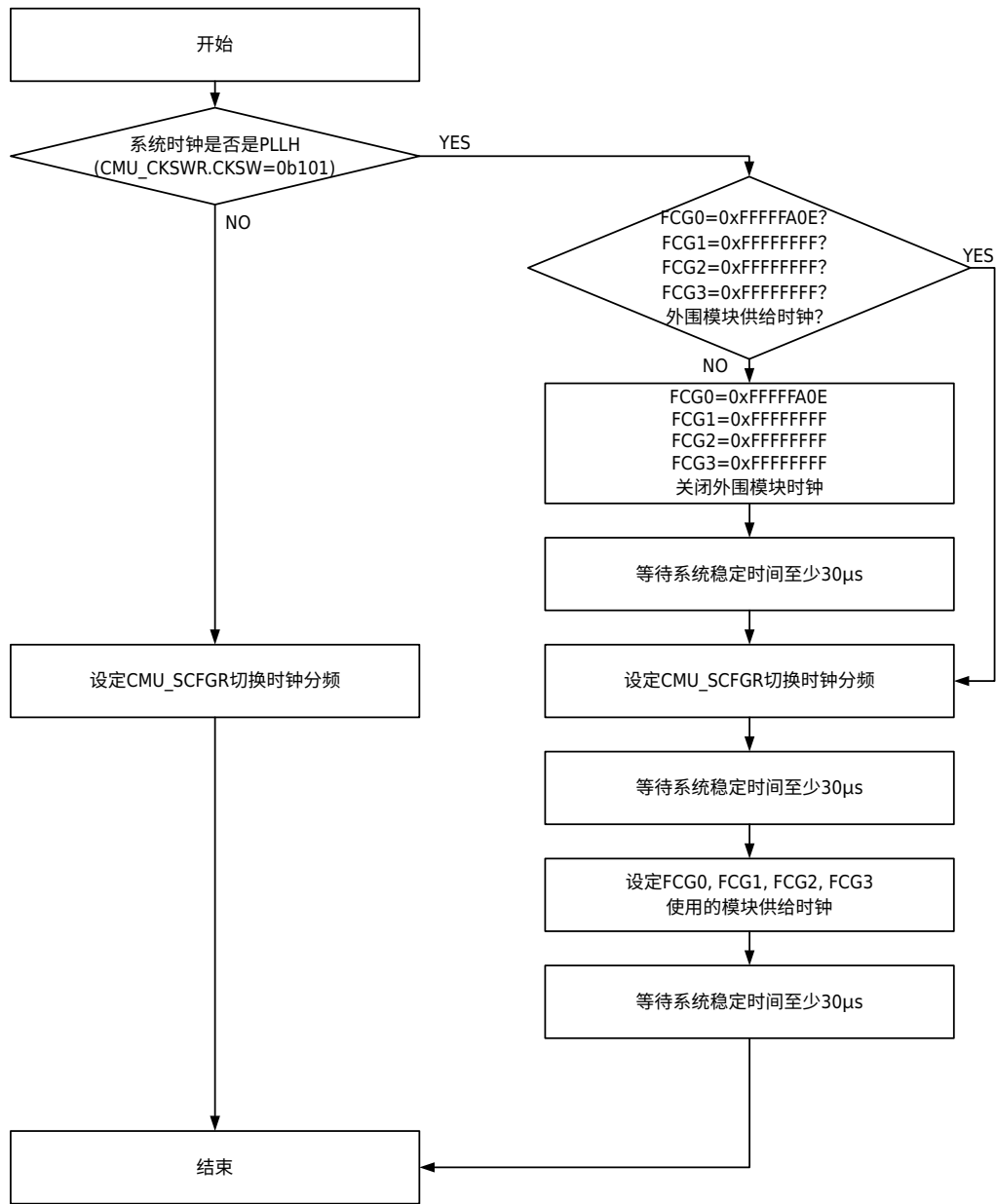


图 5-9 时钟分频切换流程图

5.3.7 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)。

5.3.8 时钟频率测量

5.3.8.1 时钟频率测量功能

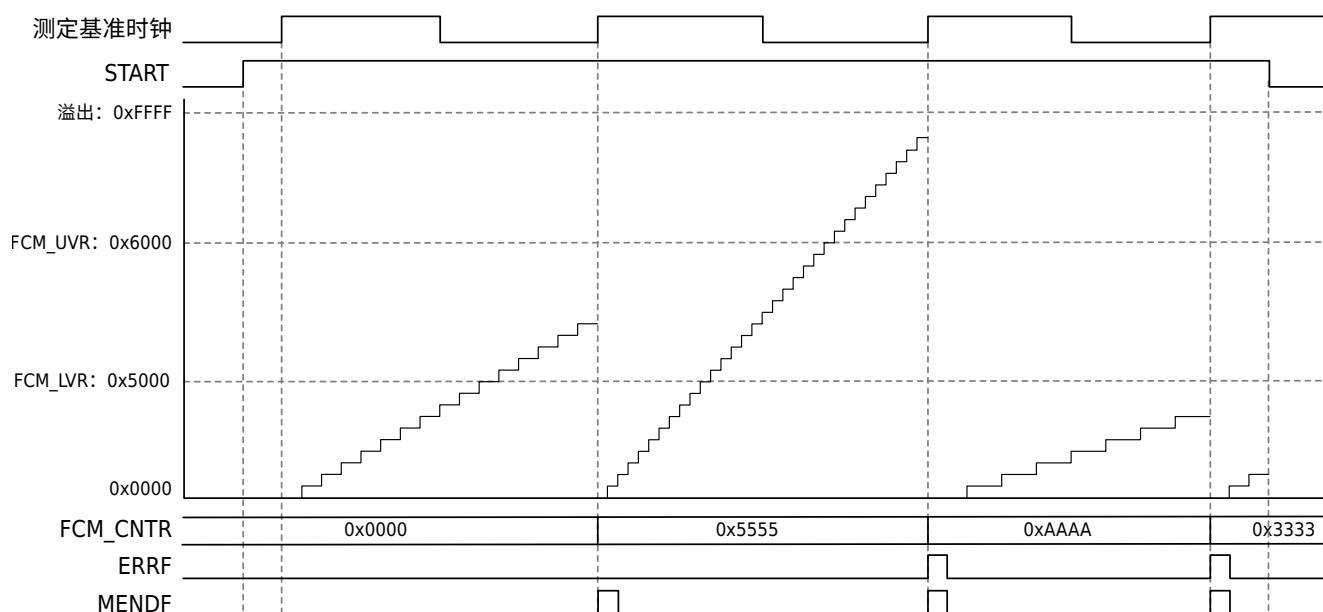


图 5-10 时钟频率测量时序图

- 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。
- FCM_STR.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。
- 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。
- FCM_STR 的 START 位写 0 后，计数器计数停止并清零。

5.3.8.2 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。

5.3.8.3 中断/复位功能

时钟频率测量电路支持三种中断请求：

- 频率异常中断
- 频率测量结束中断
- 计数器溢出中断

时钟频率测量电路支持一种复位请求：

- 频率异常复位

5.4 寄存器

5.4.1 寄存器总表

FCM 基地址: 0x40048400

表 5-3 时钟频率测量 (FCM) 寄存器偏移地址

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

CMU 基地址 1: 0x4004CC00

表 5-4 时钟控制器 (CMU) 寄存器偏移地址

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

CMU 基地址 2: 0x40048000

表 5-5 时钟控制器 (CMU) 寄存器偏移地址

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

CMU 基地址 3: 0x40054000

表 5-6 时钟控制器 (CMU) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x0A2	CMU_XTALSTBCR	CMU XTAL 稳定配置寄存器	8
0x032	CMU_XTALCR	CMU XTAL 控制寄存器	8
0x040	CMU_XTALSTDCR	CMU XTAL 振荡故障控制寄存器	8
0x041	CMU_XTALSTDSR	CMU XTAL 振荡故障状态寄存器	8
0x036	CMU_HRCCR	CMU HRC 控制寄存器	8
0x062	CMU_HRCTRM	CMU HRC 校准寄存器	8
0x038	CMU_MRCCR	CMU MRC 控制寄存器	8
0x061	CMU_MRCTRM	CMU MRC 校准寄存器	8
0x100	CMU_PLLHCFGR	CMU PLLH 配置寄存器	32
0x02A	CMU_PLLHCR	CMU PLLH 控制寄存器	8
0x03C	CMU_OSCSTBSR	CMU 时钟源稳定状态寄存器	8
0x026	CMU_CKSWR	CMU 系统时钟源切换寄存器	8
0x020	CMU_SCFGR	CMU 时钟分频配置寄存器	32
0x018	CMU_CANCKCFGR	CMU CAN 时钟配置寄存器	16
0x010	CMU_PERICKSEL	CMU ADC/DAC 时钟配置寄存器	16
0x03F	CMU_TPIUCKCFGR	CMU 调试时钟配置寄存器	8
0x03D	CMU_MCO1CFGR	CMU MCO1 时钟输出配置寄存器	8
0x03E	CMU_MCO2CFGR	CMU MCO2 时钟输出配置寄存器	8

5.4.2 CMU XTAL 配置寄存器 (CMU_XTALCFGR)

Offset	Bit Position							
0x50	7	6	5	4	3	2	1	0
Reset	0x80							
Name	Reserved	XTALMS	XTALDRV		Reserved			
Access		RW	RW					

位/位域	标记	位名	功能描述	读写
7	Reserved	保留	读出为 1，写入时必须写 1	-
6	XTALMS	XTAL 模式选择位	0b0：振荡器模式 0b1：外部时钟输入模式	RW
5:4	XTALDRV	XTAL 驱动能力选择	0b00：高驱动能力（20MHz<晶振≤25MHz） 0b01：中驱动能力（16MHz<晶振≤20MHz） 0b10：低驱动能力（8MHz<晶振≤16MHz） 0b11：超低驱动能力（4MHz≤晶振≤8MHz）	RW
3:0	Reserved	保留	-	-

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

Offset	Bit Position							
0x0A2	7	6	5	4	3	2	1	0
Reset	0x05							
Name	Reserved				XTALSTB			
Access					RW			

位/位域	标记	位名	功能描述	读写
7:4	Reserved	保留	-	-
3:0	XTALSTB	XTAL 稳定时间选择	● 0b0001: 稳定计数器 35 周期 ● 0b0010: 稳定计数器 67 周期 ● 0b0011: 稳定计数器 131 周期 ● 0b0100: 稳定计数器 259 周期 ● 0b0101: 稳定计数器 547 周期 ● 0b0110: 稳定计数器 1059 周期 ● 0b0111: 稳定计数器 2147 周期 ● 0b1000: 稳定计数器 4291 周期 ● 0b1001: 稳定计数器 8163 周期 ● 其他: 禁止设置  说明 稳定计数器的一个计数周期=LRC 周期/8 仅支持在 CMU_XTALCR.XTALSTP=1 且 CMU_OSCSTBSR.XTALSTBF=0 的状态下配置此寄存器。	RW

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

Offset	Bit Position																															
0x18	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x03D09040																															
Name	Reserved				NUMER																Reserved				DENOM							
Access					RW																				RW							

位/位域	标记	位名	功能描述	读写
31:29	Reserved	保留	-	-
28:12	NUMER	XTAL 小数分频分子设定值	XTAL 小数分频分子设定值，使用方法请参见“ 外部高速振荡器小数分频 ”	RW
11	Reserved	保留	-	-
10:0	DENOM	XTAL 小数分频分母设定值	XTAL 小数分频分母设定值，使用方法请参见“ 外部高速振荡器小数分频 ”  说明 请将此值设定为 2^N。例如，0x100 表示分母为 256；0x80 表示分母为 128。	RW

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

Offset	Bit Position																																
0x1C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																																FRADIVEN
Access																																	RW

位/位域	标记	位名	功能描述	读写
31:1	Reserved	保留	-	-
0	FRADIVEN	XTAL 小数分频启停控制	0b0: XTAL 小数分频停止 0b1: XTAL 小数分频开始	RW

5.4.6 CMU XTAL 控制寄存器 (CMU_XTALCR)

Offset	Bit Position							
0x032	7	6	5	4	3	2	1	0
Reset	0x01							
Name	Reserved							XTALSTP
Access								RW

位/位域	标记	位名	功能描述	读写
7:1	Reserved	保留	-	-
0	XTALSTP	XTAL 振荡器启停控制	0b0: XTAL 振荡器振荡 0b1: XTAL 振荡器停止	RW

 **注意**

1. XTAL 选作系统时钟或者 PLLH 时钟源时，禁止通过 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。

5.4.7 CMU XTAL 振荡故障控制寄存器 (CMU_XTALSTDCR)

Offset	Bit Position							
0x040	7	6	5	4	3	2	1	0
Reset	0x00							
Name	XTALSTDE	Reserved				XTALSTDRIS	XTALSTDRE	XTALSTDIE
Access	RW					RW	RW	RW

位/位域	标记	位名	功能描述	读写
7	XTALSTDE	XTAL 振荡故障检测功能允许	0b0: 禁止 XTAL 振荡故障检测 0b1: 允许 XTAL 振荡故障检测  说明 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
6:3	Reserved	保留	-	-
2	XTALSTDRIS	XTAL 振荡故障复位中断选择	0b0: XTAL 振荡故障产生中断 0b1: XTAL 振荡故障产生复位  说明 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
1	XTALSTDRE	XTAL 振荡故障复位允许	0b0: 禁止 XTAL 振荡故障复位 0b1: 允许 XTAL 振荡故障复位	RW
0	XTALSTDIE	XTAL 振荡故障中断允许	0b0: 禁止 XTAL 振荡故障中断 0b1: 允许 XTAL 振荡故障中断  说明 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，禁止通过 XTALSTP 写 1 停止 XTAL 振荡器。

5.4.8 CMU XTAL 振荡故障状态寄存器 (CMU_XTALSTDSR)

Offset	Bit Position							
0x041	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved							XTALSTDF
Access								RW

位/位域	标记	位名	功能描述	读写
7:1	Reserved	保留	-	-
0	XTALSTDF	XTAL 振荡故障状态位	0b0: 未检测到 XTAL 振荡故障 0b1: 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

5.4.9 CMU XTAL32 配置寄存器 (CMU_XTAL32CFGR)

Offset	Bit Position							
0x58	7	6	5	4	3	2	1	0
Reset	0xXX							
Name	Reserved					XTAL32DRV		
Access						RW		

位/位域	标记	位名	功能描述	读写
7:3	Reserved	保留	-	-
2:0	XTAL32DRV	XTAL32 驱动能力选择	0b000: 中驱动能力 0b001: 高驱动能力 - 其他: 禁止设置  说明 使用方法请参见《数据手册》“电气特性-工作条件-外部时钟源特性-晶振/陶瓷谐振器产生的低速外部时钟”。	RW

5.4.10 CMU XTAL32 滤波寄存器 (CMU_XTAL32NFR)

Offset	Bit Position							
0x68	7	6	5	4	3	2	1	0
Reset	0xXX							
Name	Reserved						XTAL32NF	
Access							RW	

位/位域	标记	位名	功能描述	读写
7:2	Reserved	保留	-	-
1:0	XTAL32NF	XTAL32 振荡器滤波选择	0b00: RUN 模式/停止模式/掉电模式, XTAL32 的 3μs 滤波有效 0b01: 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 0b10: 禁止设置 0b11: RUN 模式/停止模式/掉电模式, XTAL32 的 3μs 滤波无效	RW

5.4.11 CMU XTAL32 控制寄存器 (CMU_XTAL32CR)

Offset	Bit Position							
0x54	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved						XTAL32STP	
Access							RW	

位/位域	标记	位名	功能描述	读写
7:1	Reserved	保留	-	-
0	XTAL32STP	XTAL32 振荡器开启停止位	0b0: XTAL32 振荡器振荡 0b1: XTAL32 振荡器停止	RW

**注意**

XTAL32 选作系统时钟源时, 禁止通过 XTAL32STP 写 1 停止 XTAL32 振荡器。

5.4.12 CMU HRC 校准寄存器 (CMU_HRCTRM)

Offset	Bit Position							
0x062	7	6	5	4	3	2	1	0
Reset	0x00							
Name	HRCTRM							
Access	RW							

位/位域	标记	位名	功能描述	读写
7:0	HRCTRM	HRC 频率校准	● 0b10000000: -128 ● 0b10000001: -127 ● ● 0b11111111: -1 ● 0b00000000: 中心 Code ● 0b00000001: +1 ● ● 0b01111110: +126 ● 0b01111111: +127	RW

 **说明**

频率校准需在 HRC 频率保证范围内。

5.4.13 CMU HRC 控制寄存器 (CMU_HRCCR)

Offset	Bit Position							
0x036	7	6	5	4	3	2	1	0
Reset	由 ICG1.HRCSTOP 值决定							
Name	Reserved							HRCSTP
Access								RW

位/位域	标记	位名	功能描述	读写
7:1	Reserved	保留	-	-
0	HRCSTP	HRC 振荡器开启 停止位	● 0b0: HRC 振荡器振荡 ● 0b1: HRC 振荡器停止  说明 根据 ICG1.HRCSTOP 的配置，复位后 HRC 振荡或停止。	RW

 **注意**

1. HRC 选作系统时钟源或 PLLH 时钟源时，禁止通过 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。

5.4.14 CMU MRC 校准寄存器 (CMU_MRCTRM)

Offset	Bit Position							
0x061	7	6	5	4	3	2	1	0
Reset	0x00							
Name	MRCTRM							
Access	RW							

位/位域	标记	位名	功能描述	读写
7:0	MRCTRM	MRC 频率校准	● 0b10000000: -128 ● 0b10000001: -127 ● ● 0b11111111: -1 ● 0b00000000: 中心 Code ● 0b00000001: +1 ● ● 0b01111110: +126 ● 0b01111111: +127	RW

 **说明**

频率校准需在 MRC 频率保证范围内。

5.4.15 CMU MRC 控制寄存器 (CMU_MRCCR)

Offset	Bit Position							
0x038	7	6	5	4	3	2	1	0
Reset	0x80							
Name	Reserved							MRCSTP
Access								RW

位/位域	标记	位名	功能描述	读写
7	Reserved	保留	读出为 1，写入时必须写 1	-
6:1	Reserved	保留	-	-
0	MRCSTP	MRC 振荡器开启停止位	● 0b0：MRC 振荡器振荡 ● 0b1：MRC 振荡器停止  说明 ● XTAL 振荡故障功能有效时，本位同时清零，MRC 振荡。 ● 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，禁止通过 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。

5.4.16 CMU LRC 校准寄存器 (CMU_LRCTRM)

Offset	Bit Position							
0x74	7	6	5	4	3	2	1	0
Reset	0x00							
Name	LRCTRM							
Access	RW							

位/位域	标记	位名	功能描述	读写
7:0	LRCTRM	LRC 频率校准	● 0b10000000: -128 ● 0b10000001: -127 ● ● 0b11111111: -1 ● 0b00000000: 中心 Code ● 0b00000001: +1 ● ● 0b01111110: +126 ● 0b01111111: +127	RW

 **说明**

频率校准需在 LRC 频率保证范围内。

5.4.17 CMU LRC 控制寄存器 (CMU_LRCCR)

Offset	Bit Position							
0x6C	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved							LRCSTP
Access								RW

位/位域	标记	位名	功能描述	读写
7:1	Reserved	保留	-	-
0	LRCSTP	LRC 振荡器开启 停止位	0b0: LRC 振荡器振荡 0b1: LRC 振荡器停止	RW

 **注意**

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

5.4.18 CMU PLLH 配置寄存器 (CMU_PLLHCFGR)

Offset	Bit Position																															
0x100	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x11101300																															
Name	PLLHP				PLLHQ				PLLHR				Reserved				PLLHN				PLLSRC				Reserved				PLLHM			
Access	RW				RW				RW								RW				RW								RW			

位/位域	标记	位名	功能描述	读写
31:28	PLLHP	PLLHP 时钟分频系数	PLLHP 输出时钟频率=PLLH 的 VCO 频率/PLLHP ● 0b0000: 禁止设置 ● 0b0001: 2 分频 ● 0b0010: 3 分频 ● ● 0b1110: 15 分频 ● 0b1111: 16 分频  说明 在 PLLH 停止条件下写 PLLHP，设置 PLLHP 时钟的频率。	RW
27:24	PLLHQ	PLLHQ 时钟分频系数	PLLHQ 输出时钟频率=PLLH 的 VCO 频率/PLLHQ ● 0b0000: 禁止设置 ● 0b0001: 2 分频 ● 0b0010: 3 分频 ● ● 0b1110: 15 分频 ● 0b1111: 16 分频  说明 在 PLLH 停止条件下写 PLLHQ，设置 PLLHQ 时钟的频率。	RW
23:20	PLLHR	PLLHR 时钟分频系数	PLLHR 输出时钟频率=PLLH 的 VCO 频率/PLLHR ● 0b0000: 禁止设置 ● 0b0001: 2 分频 ● 0b0010: 3 分频 ● ● 0b1110: 15 分频 ● 0b1111: 16 分频  说明 在 PLLH 停止条件下写 PLLHR，设置 PLLHR 时钟的频率。	RW
19:16	Reserved	保留	-	-

位/位域	标记	位名	功能描述	读写
15:8	PLLHN	PLLH 倍频系数	PLLH 的 VCO 频率=PLLH 的 VCO 输入频率*PLLHN ● 0b00011000: 25 ● 0b00011001: 26 ● 0b00011010: 27 ● ● 0b10010100: 149 ● 0b10010101: 150 ● 其他: 禁止设置  说明 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 600MHz 到 1200MHz 之间。	RW
7	PLLSRC	PLLH 输入时钟源选择	● 0b0: 选择外部高速振荡器 ● 0b1: 选择内部高速振荡器	RW
6:2	<i>Reserved</i>	<i>保留</i>	-	-
1:0	PLLHM	PLLH 输入时钟分频系数	● 0b00: 1 分频 ● 0b01: 2 分频 ● 0b10: 3 分频 ● 0b11: 4 分频  说明 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 8MHz 到 25MHz 之间。	RW

5.4.19 CMU PLLH 控制寄存器 (CMU_PLLHCR)

Offset	Bit Position							
0x02A	7	6	5	4	3	2	1	0
Reset	0x01							
Name	Reserved							PLLHOFF
Access								RW

位/位域	标记	位名	功能描述	读写
7:1	Reserved	保留	-	-
0	PLLHOFF	PLLH 使能	0b0: PLLH 动作开始 0b1: PLLH 停止  说明 如果 PLLH 时钟用作系统时钟时，请不要将此位设成 1。	RW

 **注意**

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

5.4.20 CMU 时钟源稳定状态寄存器 (CMU_OSCSTBSR)

Offset	Bit Position							
0x03C	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved		PLLHSTBF	Reserved	XTALSTBF	Reserved		HRCSTBF
Access			RO		RO			RO

位/位域	标记	位名	功能描述	读写
7:6	Reserved	保留	-	-
5	PLLHSTBF	PLLH 稳定标志位	0b0: PLLH 停止或者未稳定 0b1: PLLH 稳定	RO
4	Reserved	保留	-	-
3	XTALSTBF	XTAL 稳定标志位	0b0: XTAL 停止或者未稳定 0b1: XTAL 稳定	RO
2:1	Reserved	保留	-	-
0	HRCSTBF	HRC 稳定标志位	0b0: HRC 停止或者未稳定 0b1: HRC 稳定	RO

5.4.21 CMU 系统时钟源切换寄存器 (CMU_CKSWR)

Offset	Bit Position							
0x026	7	6	5	4	3	2	1	0
Reset	0x01							
Name	Reserved					CKSW		
Access						RW		

位/位域	标记	位名	功能描述	读写
7:3	Reserved	保留	-	-
2:0	CKSW	系统时钟源切换	● 0b000: 选择 HRC 时钟作为系统时钟 ● 0b001: 选择 MRC 时钟作为系统时钟 ● 0b010: 选择 LRC 时钟作为系统时钟 ● 0b011: 选择 XTAL 时钟作为系统时钟 ● 0b100: 选择 XTAL32 时钟作为系统时钟 ● 0b101: 选择 PLLH 作为系统时钟 ● 其他: 禁止设置  说明 ● 切换的目标时钟源, 需保证在时钟稳定状态。操作步骤请参见“时钟源切换”。 ● PWC_STPMCRC.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

5.4.22 CMU 时钟分频配置寄存器 (CMU_SCFGR)

Offset	Bit Position																																																							
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																								
Reset	0x00000000																																																							
Name	Reserved				HCLKS				Reserved				EXCKS				Reserved				PCLK4S				Reserved				PCLK3S				Reserved				PCLK2S				Reserved				PCLK1S				Reserved				PCLK0S			
Access					RW								RW								RW								RW								RW								RW								RW			

位/位域	标记	位名	功能描述	读写
31:27	Reserved	保留	-	-
26:24	HCLKS	HCLK 时钟分频选择位	0b000: 系统时钟的 1 分频 0b001: 系统时钟的 2 分频 0b010: 系统时钟的 4 分频 0b011: 系统时钟的 8 分频 0b100: 系统时钟的 16 分频 0b101: 系统时钟的 32 分频 0b110: 系统时钟的 64 分频 0b111: 禁止设置  说明 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
23	Reserved	保留	-	-
22:20	EXCKS	EXMC 时钟分频选择位	0b000: 系统时钟的 1 分频 0b001: 系统时钟的 2 分频 0b010: 系统时钟的 4 分频 0b011: 系统时钟的 8 分频 0b100: 系统时钟的 16 分频 0b101: 系统时钟的 32 分频 0b110: 系统时钟的 64 分频 0b111: 禁止设置  说明 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
19	Reserved	保留	-	-
18:16	PCLK4S	PCLK4 时钟分频选择位	0b000: 系统时钟的 1 分频 0b001: 系统时钟的 2 分频 0b010: 系统时钟的 4 分频 0b011: 系统时钟的 8 分频 0b100: 系统时钟的 16 分频 0b101: 系统时钟的 32 分频 0b110: 系统时钟的 64 分频 0b111: 禁止设置	RW

位/位域	标记	位名	功能描述	读写
18:16	PCLK4S	PCLK4 时钟分频选择位	 说明 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
15	Reserved	保留	-	-
14:12	PCLK3S	PCLK3 时钟分频选择位	● 0b000：系统时钟的 1 分频 ● 0b001：系统时钟的 2 分频 ● 0b010：系统时钟的 4 分频 ● 0b011：系统时钟的 8 分频 ● 0b100：系统时钟的 16 分频 ● 0b101：系统时钟的 32 分频 ● 0b110：系统时钟的 64 分频 ● 0b111：禁止设置  说明 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。	RW
11	Reserved	保留	-	-
10:8	PCLK2S	PCLK2 时钟分频选择位	● 0b000：系统时钟的 1 分频 ● 0b001：系统时钟的 2 分频 ● 0b010：系统时钟的 4 分频 ● 0b011：系统时钟的 8 分频 ● 0b100：系统时钟的 16 分频 ● 0b101：系统时钟的 32 分频 ● 0b110：系统时钟的 64 分频 ● 0b111：禁止设置  说明 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。	RW
7	Reserved	保留	-	-
6:4	PCLK1S	PCLK1 时钟分频选择位	● 0b000：系统时钟的 1 分频 ● 0b001：系统时钟的 2 分频 ● 0b010：系统时钟的 4 分频 ● 0b011：系统时钟的 8 分频 ● 0b100：系统时钟的 16 分频 ● 0b101：系统时钟的 32 分频 ● 0b110：系统时钟的 64 分频 ● 0b111：禁止设置  说明 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。	RW
3	Reserved	保留	-	-

位/位域	标记	位名	功能描述	读写
2:0	PCLK0S	PCLK0 时钟分频选择位	● 0b000: 系统时钟的 1 分频 ● 0b001: 系统时钟的 2 分频 ● 0b010: 系统时钟的 4 分频 ● 0b011: 系统时钟的 8 分频 ● 0b100: 系统时钟的 16 分频 ● 0b101: 系统时钟的 32 分频 ● 0b110: 系统时钟的 64 分频 ● 0b111: 禁止设置 说明 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。	RW

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

Offset	Bit Position															
0x018	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00DD															
Name	Reserved								MCAN2CKS				MCAN1CKS			
Access									RW				RW			

位/位域	标记	位名	功能描述	读写
15:8	Reserved	保留	-	-
7:4	MCAN2CKS	MCAN2 通讯时钟选择	0b0001: 系统时钟 2 分频 0b0010: 系统时钟 3 分频 0b0011: 系统时钟 4 分频 0b0100: 系统时钟 5 分频 0b0101: 系统时钟 6 分频 0b0110: 系统时钟 7 分频 0b0111: 系统时钟 8 分频 0b1000: PLLHQ 0b1001: PLLHR 0b1101: XTAL - 其他: 禁止设置  说明 - 切换的目标时钟源为 PLLH 时, 需保证在 PLLH 时钟处于稳定状态。 - 系统时钟选择 PLLH 时, 需要关闭各外围模块时钟, 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。 - 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
3:0	MCAN1CKS	MCAN1 通讯时钟选择	0b0001: 系统时钟 2 分频 0b0010: 系统时钟 3 分频 0b0011: 系统时钟 4 分频 0b0100: 系统时钟 5 分频 0b0101: 系统时钟 6 分频 0b0110: 系统时钟 7 分频 0b0111: 系统时钟 8 分频 0b1000: PLLHQ 0b1001: PLLHR 0b1101: XTAL - 其他: 禁止设置	RW

位/位域	标记	位名	功能描述	读写
3:0	MCAN1CKS	MCAN1 通讯时钟选择	 说明 - 切换的目标时钟源为 PLLH 时，需保证在 PLLH 时钟处于稳定状态。 - 系统时钟选择 PLLH 时，需要关闭各外围模块时钟，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。 - 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

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

Offset	Bit Position															
0x010	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved											PERICKSEL				
Access												RW				

位/位域	标记	位名	功能描述	读写
15:4	Reserved	保留	-	-
3:0	PERICKSEL	ADC/DAC 时钟源选择	0b0000: CMU_SCFGR 设定的 PCLK2/ PCLK4 0b1000: PCLK2/PCLK4 配置为 PLLHQ 0b1001: PCLK2/PCLK4 配置为 PLLHR - 其他: 禁止设置  说明 - TRNG 时钟源不受该位控制，当 TRNG 动作时，该位必须设置为 0b0000。 - 切换的目标时钟源为 PLLH 时，需保证在 PLLH 时钟处于稳定状态。	RW

5.4.25 CMU 调试时钟配置寄存器 (CMU_TPIUCKCFGR)

Offset	Bit Position							
0x03F	7	6	5	4	3	2	1	0
Reset	0x00							
Name	TPIUCKOE	Reserved					TPIUCKS	
Access	RW						RW	

位/位域	标记	位名	功能描述	读写
7	TPIUCKOE	TPIU 时钟供给允许位	● 0b0：禁止 ● 0b1：允许	RW
6:2	Reserved	保留	-	-
1:0	TPIUCKS	TPIU 时钟分频选择位	● 0b00：1 分频 ● 0b01：2 分频 ● 0b10：4 分频 ● 0b11：禁止设置	RW

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

Offset	Bit Position							
0x03D	7	6	5	4	3	2	1	0
Reset	0x00							
Name	MCO1EN	MCO1DIV			MCO1SEL			
Access	RW	RW			RW			

位/位域	标记	位名	功能描述	读写
7	MCO1EN	MCO_1 输出许可	● 0b0: 禁止 ● 0b1: 允许	RW
6:4	MCO1DIV	MCO_1 分频选择	● 0b000: 1 分频 ● 0b001: 2 分频 ● 0b010: 4 分频 ● 0b011: 8 分频 ● 0b100: 16 分频 ● 0b101: 32 分频 ● 0b110: 64 分频 ● 0b111: 128 分频	RW
3:0	MCO1SEL	MCO_1 时钟源选择	● 0b0000: HRC 时钟 ● 0b0001: MRC 时钟 ● 0b0010: LRC 时钟 ● 0b0011: XTAL 时钟 ● 0b0100: XTAL32 时钟 ● 0b0110: PLLHP ● 0b1000: PLLHQ ● 0b1011: 系统时钟 ● 其他: 禁止设置	RW

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

Offset	Bit Position							
0x03E	7	6	5	4	3	2	1	0
Reset	0x00							
Name	MCO2EN	MCO2DIV			MCO2SEL			
Access	RW	RW			RW			

位/位域	标记	位名	功能描述	读写
7	MCO2EN	MCO_2 输出许可	0b0: 禁止 0b1: 允许	RW
6:4	MCO2DIV	MCO_2 分频选择	0b000: 1 分频 0b001: 2 分频 0b010: 4 分频 0b011: 8 分频 0b100: 16 分频 0b101: 32 分频 0b110: 64 分频 0b111: 128 分频	RW
3:0	MCO2SEL	MCO_2 时钟源选择	0b0000: HRC 时钟 0b0001: MRC 时钟 0b0010: LRC 时钟 0b0011: XTAL 时钟 0b0100: XTAL32 时钟 0b0110: PLLHP 0b1000: PLLHQ 0b1011: 系统时钟 - 其他: 禁止设置	RW

5.4.28 FCM 下限比较值寄存器 (FCM_LVR)

Offset	Bit Position																															
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																LVR															
Access																																

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	LVR	下限比较值	START 位为 0 时配置此寄存器	RW

5.4.29 FCM 上限比较值寄存器 (FCM_UVR)

Offset	Bit Position																															
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																UVR															
Access																																

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	UVR	上限比较值	START 位为 0 时配置此寄存器	RW

5.4.30 FCM 计数器值寄存器 (FCM_CNTR)

Offset	Bit Position																															
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																CNTR															
Access																																

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	CNTR	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)	RO

5.4.31 FCM 开始停止寄存器 (FCM_STR)

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															RW START
Access																																RW

位/位域	标记	位名	功能描述	读写
31:1	Reserved	保留	-	-
0	START	频率测量开始位	0b0: 频率测量停止 0b1: 频率测量开始	RW

5.4.32 FCM 测量对象控制寄存器 (FCM_MCCR)

Offset	Bit Position																															
0x10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								MCKS		Reserved	MDIVS				
Access																									RW			RW				

位/位域	标记	位名	功能描述	读写
31:8	Reserved	保留	-	-
7:4	MCKS	测量对象时钟选择位	0b0000: XTAL 0b0001: XTAL32 0b0010: HRC 0b0011: LRC 0b0100: SWDTLRC 0b0101: PCLK1 0b0111: MRC 0b1000: PLLHP - 其他: 禁止设置	RW
3:2	Reserved	保留	-	-
1:0	MDIVS	测量对象分频选择	0b00: 不分频 0b01: 4 分频 0b10: 8 分频 0b11: 32 分频	RW

5.4.33 FCM 测量基准控制寄存器 (FCM_RCCR)

Offset	Bit Position																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															
Reset	0x00000000																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
Name	Reserved																EXREFE	Reserved	EDGES	Reserved	DNFS	INEXS	RCKS				Reserved	RDIVS																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
Access																	RW		RW		RW	RW																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15	EXREFE	外部管脚输入参考时钟 FCMREF 允许位	0b0: 禁止外部管脚输入参考时钟 FCMREF 0b1: 允许外部管脚输入参考时钟 FCMREF	RW
14	Reserved	保留	-	-
13:12	EDGES	测量基准边沿选择位	0b00: 上升沿 0b01: 下降沿 0b10: 上升和下降沿 0b11: 禁止设置	RW
11:10	Reserved	保留	-	-
9:8	DNFS	数字滤波器功能选择位	0b00: 无滤波功能 0b01: MCKS 位选择的时钟作为滤波时钟 0b10: MCKS 位选择的时钟的 4 分频作为滤波时钟 0b11: MCKS 位选择的时钟的 16 分频作为滤波时钟	RW
7	INEXS	测量基准, 内部时钟和端子选择位	0b0: 外部管脚输入参考时钟 FCMREF 0b1: RCKS 选择位选择的时钟	RW
6:3	RCKS	测量基准时钟选择位	0b0000: XTAL 0b0001: XTAL32 0b0010: HRC 0b0011: LRC 0b0100: SWDTLRC 0b0101: PCLK1 0b0111: MRC 0b1000: PLLHP - 其他: 禁止设置	RW
2	Reserved	保留	-	-
1:0	RDIVS	测量基准分频选择	0b00: 32 分频 0b01: 128 分频 0b10: 1024 分频 0b11: 8192 分频	RW

5.4.34 FCM 中断复位控制寄存器 (FCM_RIER)

Offset	Bit Position																															
0x18	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								ERRRE	Reserved	ERRINTRS	Reserved	OVFIE	MENDIE	ERRIE	
Access																									RW		RW		RW	RW	RW	

位/位域	标记	位名	功能描述	读写
31:8	Reserved	保留	-	-
7	ERRRE	频率异常复位允许位	● 0b0：禁止 ● 0b1：允许	RW
6:5	Reserved	保留	-	-
4	ERRINTRS	频率异常中断复位选择位	● 0b0：频率异常发生中断 ● 0b1：频率异常发生复位	RW
3	Reserved	保留	-	-
2	OVFIE	计数器溢出中断允许位	● 0b0：禁止计数器溢出中断 ● 0b1：允许计数器溢出中断	RW
1	MENDIE	测量结束中断允许位	● 0b0：禁止测量结束发生中断 ● 0b1：允许测量结束发生中断	RW
0	ERRIE	频率异常中断允许位	● 0b0：禁止频率异常发生中断 ● 0b1：允许频率异常发生中断	RW

5.4.35 FCM 标志寄存器 (FCM_SR)

Offset	Bit Position																																							
0x1C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Reset	0x00000000																																							
Name	Reserved																												RO		OVF		RO		MENDF		RO		ERRF	
Access																																								

位/位域	标记	位名	功能描述	读写
31:3	Reserved	保留	-	-
2	OVF	计数器溢出标志位	0b0: 计数器未溢出 0b1: 计数器溢出	RO
1	MENDF	测量结束标志位	0b0: 测量中 0b1: 测量结束	RO
0	ERRF	频率异常标志位	0b0: 无频率异常发生 0b1: 发生频率异常	RO

5.4.36 FCM 标志位清除寄存器 (FCM_CLR)

Offset	Bit Position																																	
0x20	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved																												OVFCLR		MENDFCLR		ERRFCLR	
Access																													WO		WO		WO	

位/位域	标记	位名	功能描述	读写
31:3	Reserved	保留	-	-
2	OVFCLR	计数器溢出标志清除	写 1 清除标志, 写 0 无效	WO
1	MENDFCLR	测量结束标志清除	写 1 清除标志, 写 0 无效	WO
0	ERRFCLR	频率异常标志清除	写 1 清除标志, 写 0 无效	WO

6 电源控制 (PWC)

6.1 简介

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

芯片的工作电压 (VCC) 为 1.8V 到 3.6V。电压调节器 (LDO) 为 VDD 域和 VDDR 域供电, VDDR 电压调节器 (RLDO) 在掉电模式为 VDDR 域供电。芯片通过功耗控制逻辑 (PWCL) 提供了高速、超低速两种运行模式, 睡眠、停止和掉电等三种低功耗模式。

电源电压检测单元 (PVD) 提供了上电复位 (POR)、掉电复位 (PDR)、欠压复位 (BOR)、可编程电压检测 1 (PVD1)、可编程电压检测 2 (PVD2) 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。

6.2 功能说明

6.2.1 电源分布

6.2.1.1 电源分布构成

芯片由 VCC 域、VDD 电源域、AVCC 电源域和 VDDR 域构成。

- VCC 域通过 VCC/VSS 引脚供电, 由功耗控制逻辑 (PWCL)、电源电压检测单元 (PVD)、IO 电平保持电路、电压调节器 (LDO)、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 电源域由 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。
- AVCC 电源域为模拟电源域, 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。
- VDDR 域由 4KB 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, 除此以外的模式中通过 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。

下图是芯片的电源分布图。

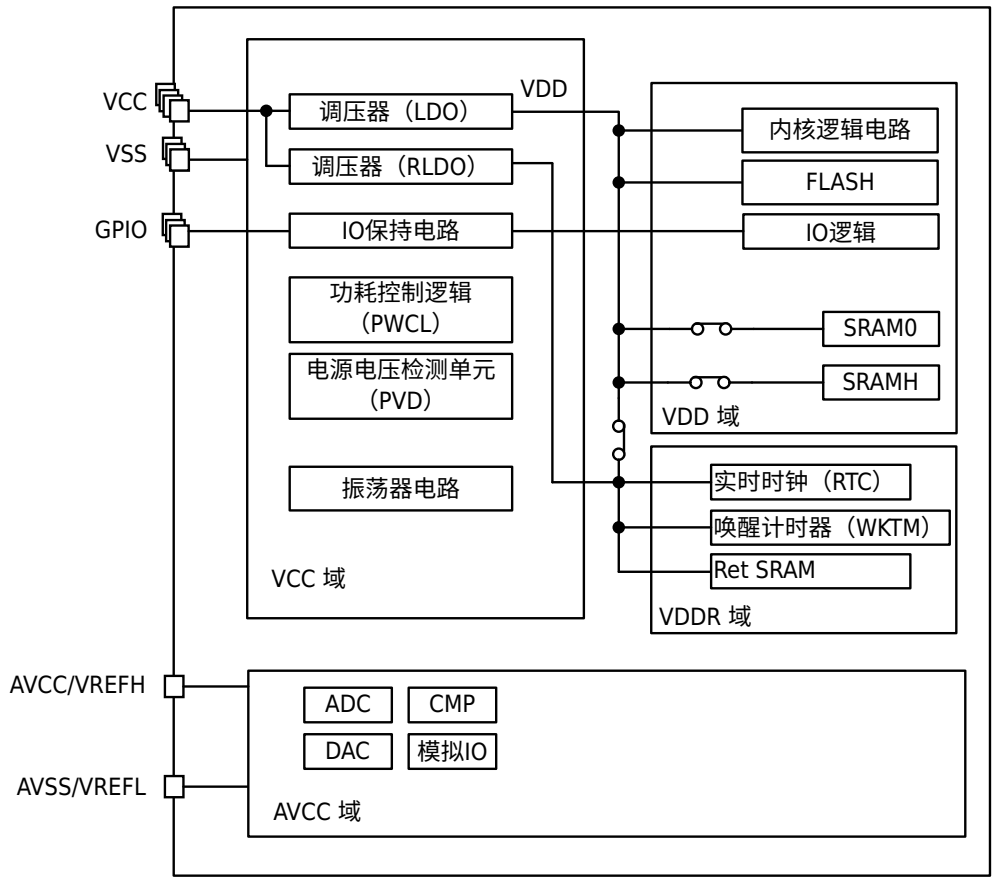


图 6-1 电源构成图

6.2.2 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。

6.2.2.1 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、掉电复位波形如下图所示，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，请参见《数据手册》“电气特性”章节。

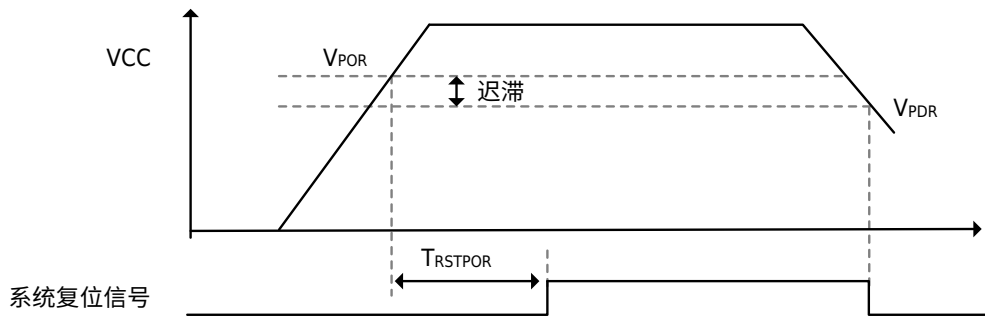


图 6-2 上电复位、掉电复位波形

6.2.2.2 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。

VBOR 阈值通过初始化配置位（ICG）的 BOR_LEV、BORDIS 进行配置。BORDIS=0b0 时，BOR 检测电压可以从 4 个阈值中选择。BORDIS 被配置成 0b1 时，芯片通过上电复位、掉电复位进行复位控制。

表 6-1 BOR 配置

BORDIS	BOR_LEV	说明
0b1	0bXX	BOR 无效
0b0	0b00	BOR 有效、选择 BOR 阈值 0（VBOR0）
0b0	0b01	BOR 有效、选择 BOR 阈值 1（VBOR1）
0b0	0b10	BOR 有效、选择 BOR 阈值 2（VBOR2）
0b0	0b11	BOR 有效、选择 BOR 阈值 3（VBOR3）

BOR 阈值的电气特性，请参见《数据手册》“电气特性”章节。

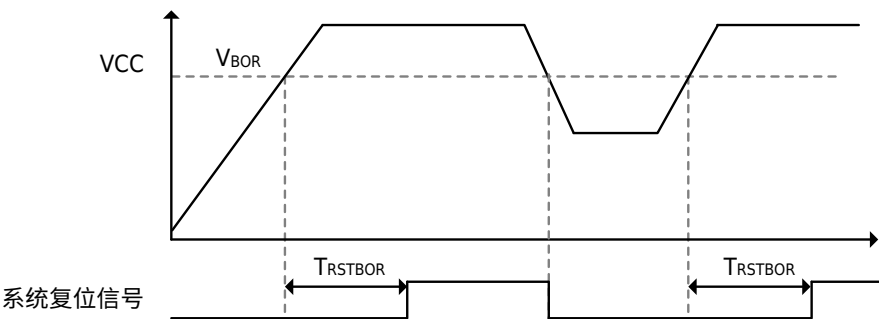


图 6-3 欠压复位波形

6.2.2.3 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下表所示。

表 6-2 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，PWC_PVDLCR.PVD2LVL=0b111）
检测电压点	由 PVD1LVL 配置	由 PVD2LVL 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。

项目	PVD1	PVD2
中断	配置成电压检测 1 非可屏蔽中断	配置成电压检测 2 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）

6.2.2.4 PVD1、PVD2 中断/复位框图

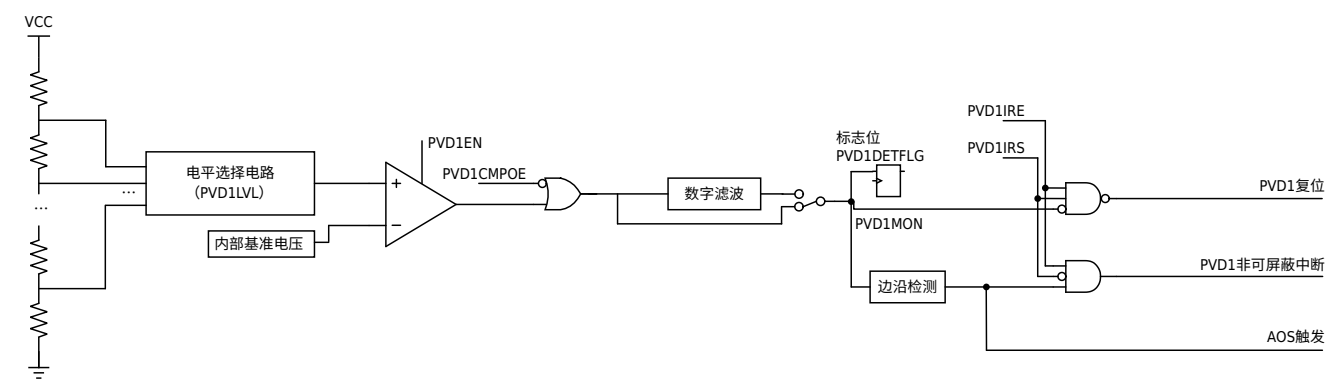


图 6-4 PVD1 中断/复位框图

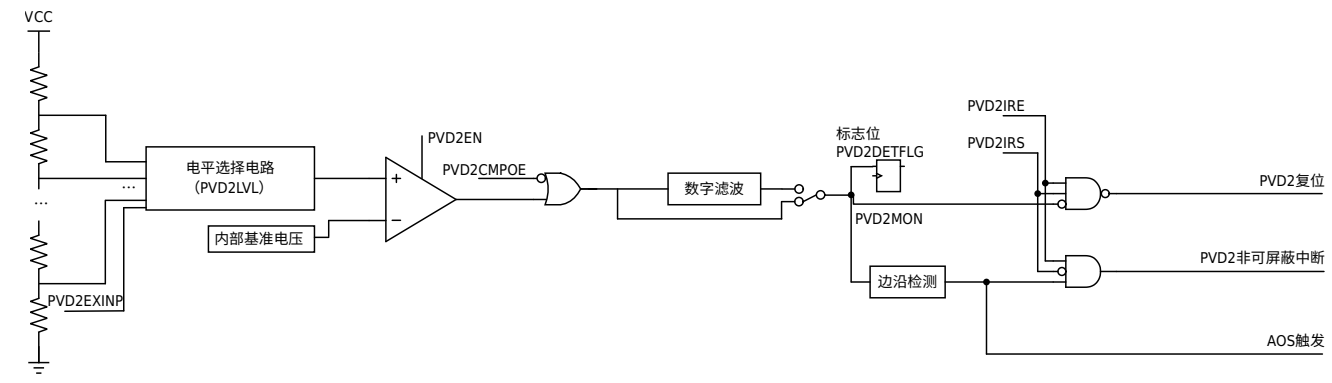


图 6-5 PVD2 中断/复位框图

6.2.2.5 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

6.2.2.6 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
 - 必须将数字滤波器无效。
2. 掉电模式
 - 必须将数字滤波器无效。
 - PVD1IRS 置 0，选择 PVD1 产生中断；选择复位功能时，不能进入掉电模式。

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

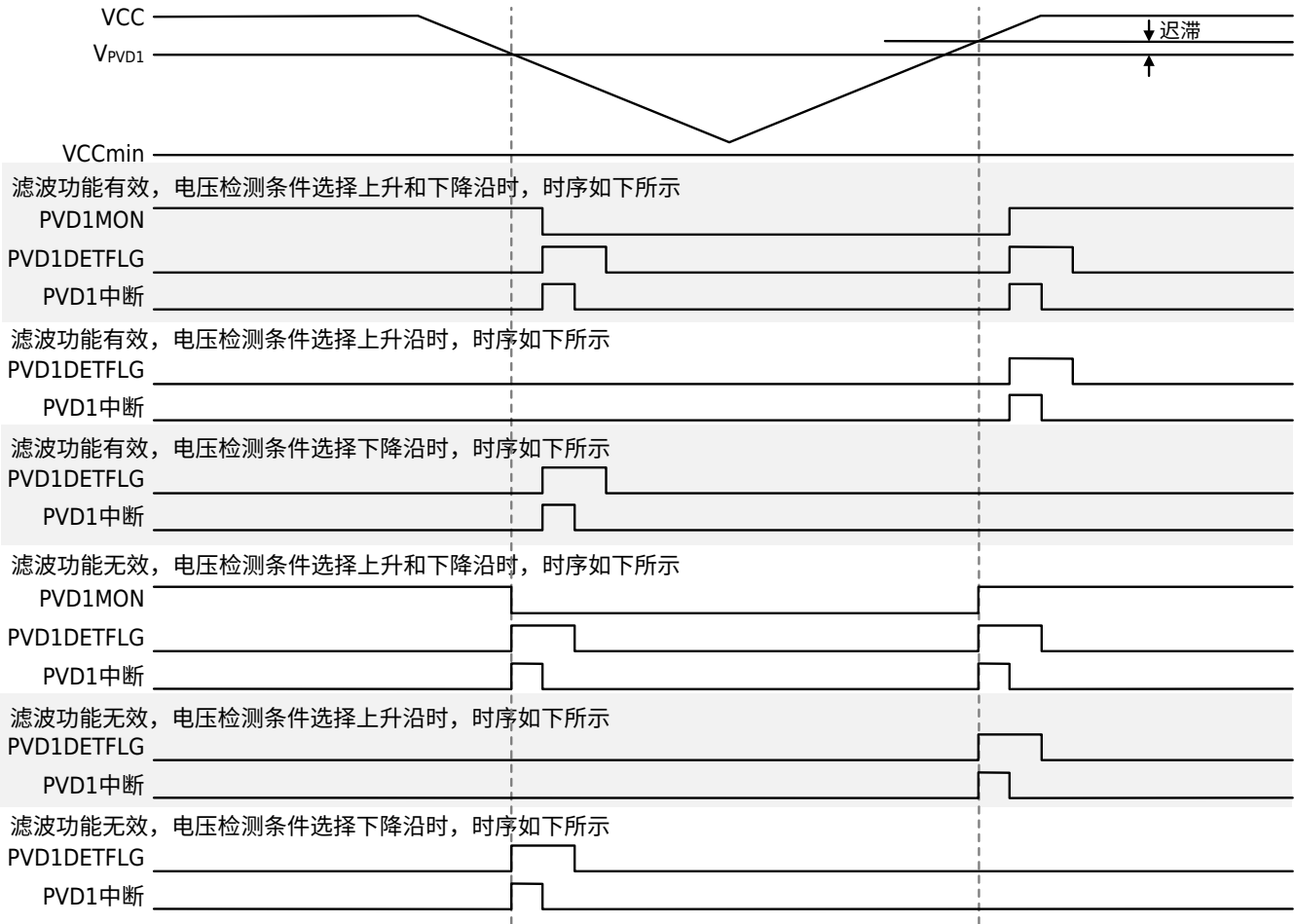


图 6-6 电源监视 1 中断时序图

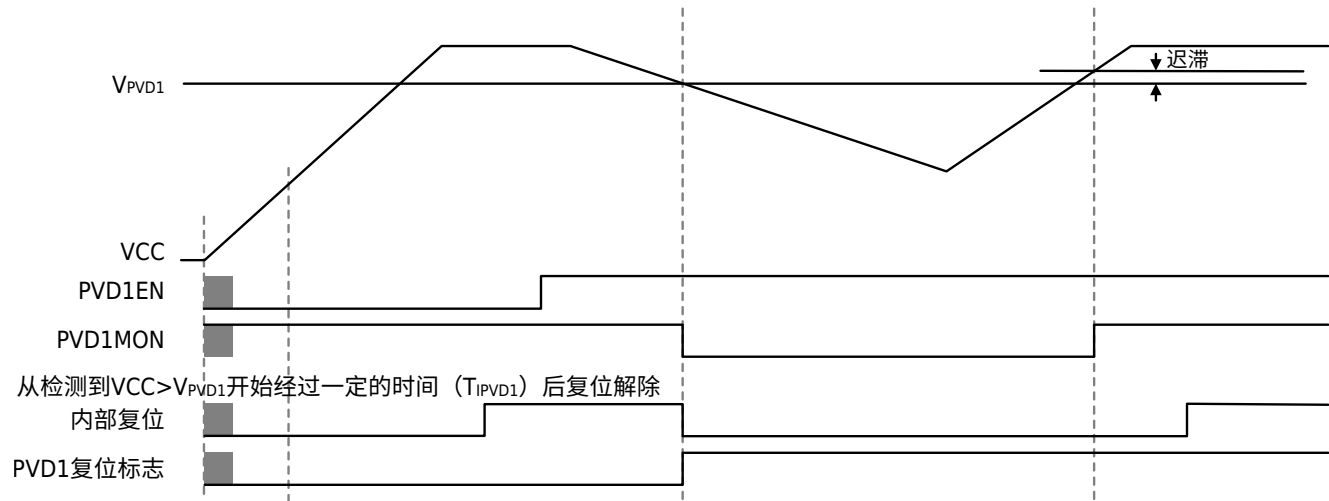


图 6-7 电源监视 1 复位时序图

6.2.2.7 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
 - 必须将数字滤波器无效。
2. 掉电模式
 - 必须将数字滤波器无效。

- PVD2IRS 置 0，选择 PVD2 产生中断；选择复位功能时，不能进入掉电模式。

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

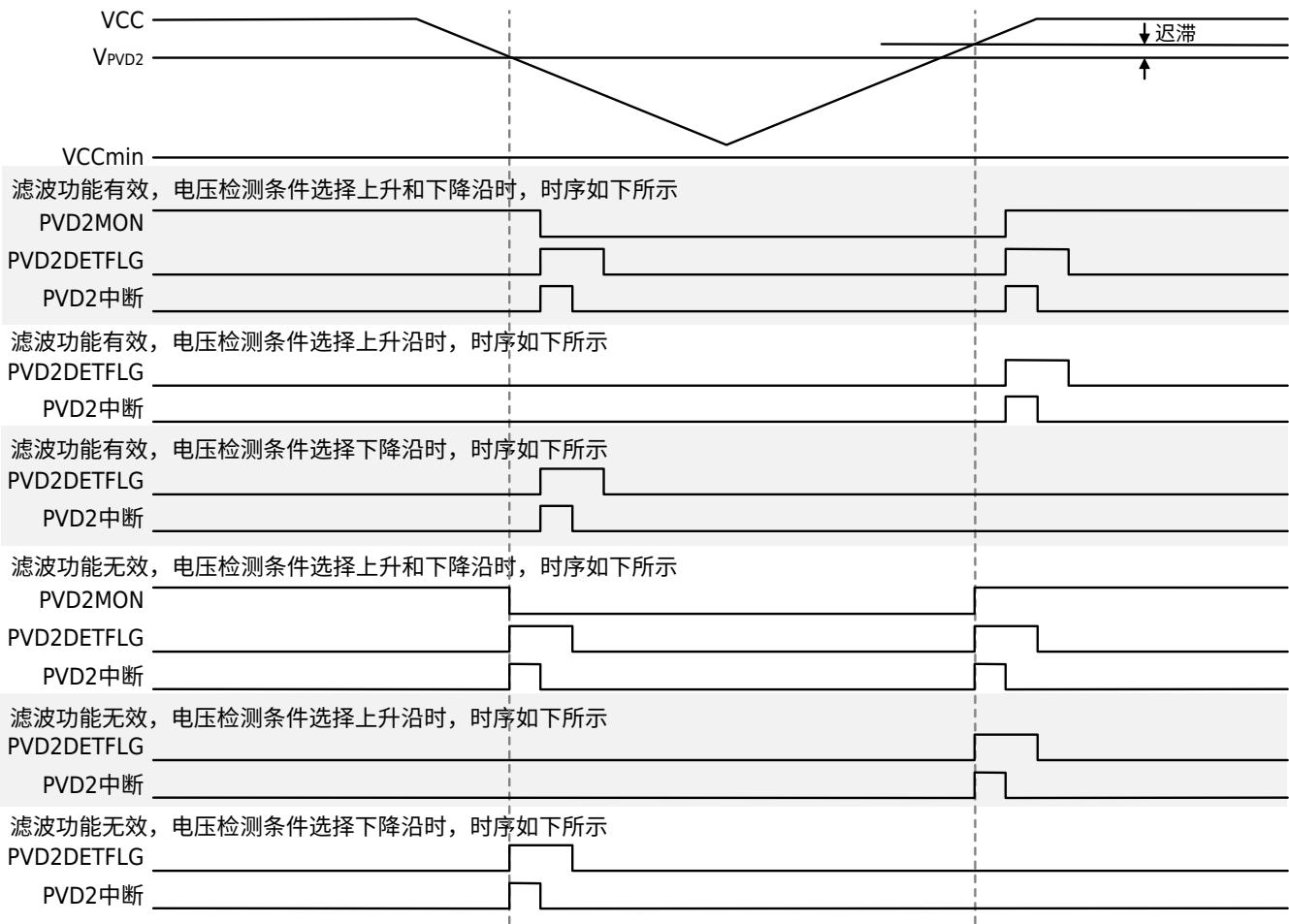


图 6-8 电源监视 2 中断时序图

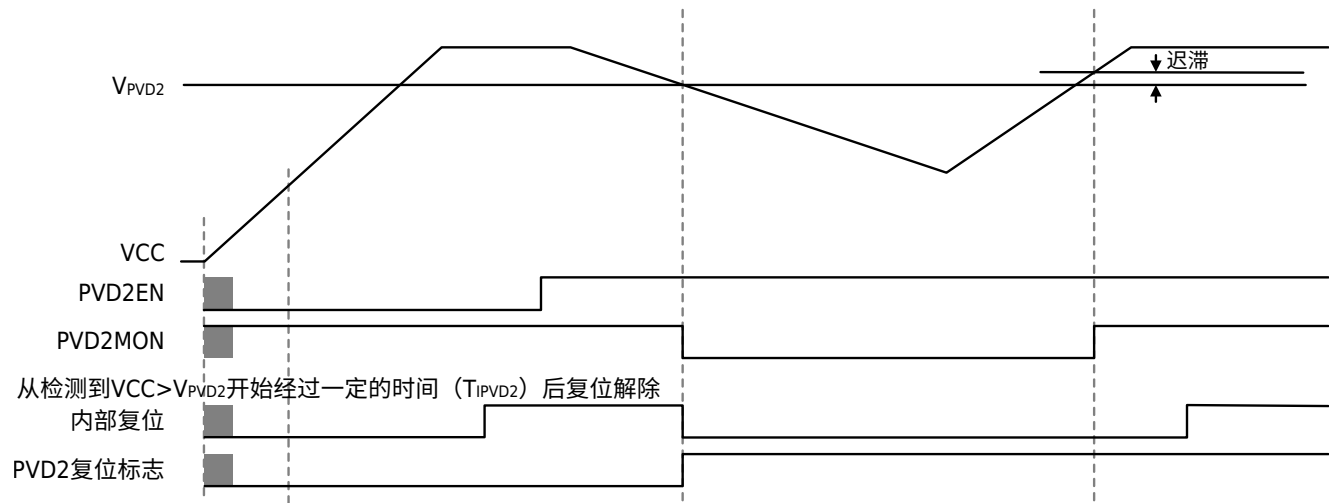


图 6-9 电源监视 2 复位时序图

6.2.2.8 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 1.15V。

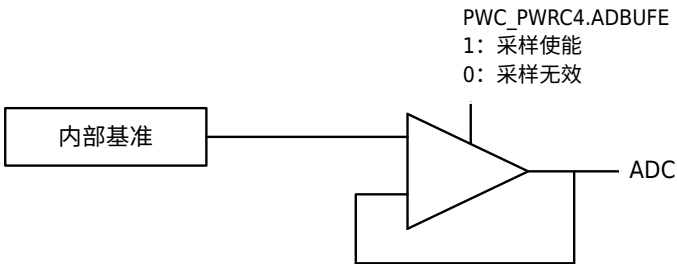


图 6-10 内部电压采样示意图

● 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=0b1，使能内部电压测量功能；
- 2. ADC_EXCHSELR.EXCHSEL=0b1，使 ADC 选通内部电压测量通道；
- 3. 等待 50μs 后使用 ADC 测量内部基准电压。

6.2.3 唤醒定时器

芯片内置了唤醒定时器 WKTMR，该计数器可以选择内部低速振荡器（LRC）、外部低速振荡器作为时钟源，在 RTC 使用外部低速振荡器作为时钟动作时还可以选择 64Hz 的内部时钟信号作为时钟源。该计数器是加计数的计数器，在 PWC_WKTCR.WKTCE 置位后，计数器开始计数，当计数值等于 PWC_WKTCR.WKTCMP[11:0] 设定值的一个周期后，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。WKTMR 寄存器值上电复位后不定，上电复位后，设置 PWC_WKTCR。在启动 WKTMR 前，要确保计数时钟处于开启状态。

6.2.4 动作模式与低功耗模式

6.2.4.1 模式概述

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，芯片进入高速运行模式。在运行模式下，CPU 通过 HCLK 提供时钟，并执行程序代码。芯片提供了高速运行模式、超低速运行模式两种运行模式。芯片可以配置的运行模式如下表所示。

表 6-4 运行模式

运行模式	说明
高速运行模式（High Speed Run Mode）	主频≤200MHz
超低速运行模式（Ultra Low Speed Run Mode）	主频≤8MHz

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下表所示。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。

表 6-5 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

模式		说明
掉电模式 (Power DownMode)	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

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

用户可以通过下列方法优化运行模式下的功耗:

1. 设定最适的运行模式
2. 降低系统时钟速度
3. 关闭不使用的时钟源
4. 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
5. 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下表所示。

表 6-6 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=0b0, PWC_PWRC0.PWDN=0b0, WFI	PWC_STPMCR.STOP=0b1, PWC_PWRC0.PWDN=0b0, WFI	PWC_STPMCR.STOP=0b1, PWC_PWRC0.PWDN=0b1, WFI
解除	任意中断或者复位	停止模式下可以使用的中断或复位	掉电模式下可以使用的唤醒事件或者复位
外部高速振荡器	工作可设	停止	停止
外部低速振荡器	工作可设	工作可设	工作可设
内部高速振荡器	工作可设	停止	掉电
内部中速振荡器	工作可设	停止	掉电
内部低速振荡器	工作可设	工作可设	工作可设
WDT 专用时钟振荡器	工作可设	工作可设	掉电
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, 可以维持掉电或者睡眠	掉电
Ret SRAM	工作可设 可以设置工作, 掉电, 睡眠	停止 (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
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

项目	睡眠模式	停止模式	掉电模式
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
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~PA15, PB0~PB15, PC0~PC8, PC10~PC13, PD0~PD2, PD8~PD11, PE2~PE4, PE12~PE15, PH2	工作可设	保持	保持或者高阻
PC14-PC15	工作可设	作为外部低速振荡器的管脚使用时, 保持振荡器动作; 设为 GPIO 或者其他外设功能时, 请设定保持两个管脚为同样的电平	设为 GPIO 或者其他外设功能时, PC14 和 PC15 的状态可以设置为保持或者高阻, 请设定保持两个管脚为同样的电平
PH0-PH1	工作可设	作为外部高速振荡器使用时, 振荡器停止震荡, 管脚状态保持进入 STOP 模式前的状态; 设为 GPIO 或者其他外设功能时, 保持 STOP 之前的状态	作为外部高速振荡器使用时, 振荡器停止震荡, 管脚状态保持进入掉电模式前的状态; 设为 GPIO 或者其他外设功能时, 保持掉电模式之前的状态
NRST 复位管脚	芯片外面通过电路上拉到 VCC	芯片外面通过电阻上拉到 VCC	芯片外面通过电阻上拉到 VCC
MD	工作可设	保持	保持 芯片外面通过电阻接到地
JTAG	工作可设 作为 JTAG 功能时, 内置上拉电路有效	保持 作为 JTAG 功能时, 内置上拉电路有效	保持 作为 JTAG 功能时, 内置上拉电路有效

6.2.4.2 运行模式

芯片有高速、超低速等两种运行模式。根据系统的速度要求, 设置最佳的运行模式, 使内核电压、驱动能力适配于系统时钟频率, 从而达到降低功耗的目的。如下表所示, 根据 PWC_PWRC2.DVS 位, 可以让芯片工作在相应的运行模式。芯片选择动作在超低速模式时, FLASH 和 RAM 也需要设定工作在低内核电压下, 因此需要设置寄存器位 FRMC.LVM=0b1、PWC_RAMOPM=0x9062。

表 6-7 运行模式说明

运行模式	频率范围	寄存器设定	
		PWC_PWRC2.DVS	PWC_PWRC3.DDAS
高速运行模式	≤200MHz	0b11	0b1111
超低速运行模式*	≤8MHz	0b10	0b0000

 **说明**

*: 超低速模式只支持 $T_A = -40^{\circ}\text{C} \sim 85^{\circ}\text{C}$ 。

高速运行模式与超低速运行模式之间的切换需要遵循如下流程。

- **高速模式向超低速模式切换**
 1. 设置在超低速模式下使用的时钟源，确保时钟源符合超低速模式下的频率要求；
 2. 关闭在超低速模式下不需要使用的时钟源及模块，确认 Flash 不处于编程或者擦除状态；
 3. 设置 FLASH 的 FRMC.LVM=0b1，RAM 动作模式寄存器 PWC_RAMOPM 设置成 0x9062；
 4. 确认 FRMC.LVM=0b1，PWC_RAMOPM=0x9062；
 5. 设置 PWC_PWRC3.DDAS 为 0x0，PWC_PWRC2.DVS 为 0b10；
 6. 等待 TSWMD1 (30μs) ；
 7. 芯片在超低速模式下动作。
- **超低速模式向高速模式切换**
 1. 设置 PWC_PWRC3.DDAS 为 0xF，PWC_PWRC2.DVS 根据系统频率要求设置为 0b11；
 2. 等待 TSWMD2 (30μs) ；
 3. 设置 FLASH 的 FRMC.LVM=0b0，RAM 动作模式寄存器 PWC_RAMOPM 设置成 0x8043；
 4. 确认 FRMC.LVM=0b0，PWC_RAMOPM=0x8043；
 5. 芯片在高速模式下动作。

6.2.4.3 睡眠模式

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

通过 ICG 设置为自动启动时，如果 ICG0.WDTSLOFF 位为 0b1，看门狗在睡眠模式下停止计数；如果 WDTSLOFF 位为 0b0，看门狗在睡眠模式下继续计数。当 ICG 未设置为自动启动，通过软件启动的方式启动看门狗时，如果 WDT_CR.SLOFF 位为 0b1，看门狗在睡眠模式下停止计数；如果 WDT_CR.SLOFF 位为 0b0，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.SWDTSLPOFF 位为 0b1，专用看门狗在睡眠模式下停止计数；如果 SWDTSLPOFF 位为 0b0，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=0b0 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。

6.2.4.4 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。

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。

通过 ICG 设置为自动启动时, 如果 ICG0.SWDTSLPOFF 位为 0b1, 专用看门狗在停止模式下停止计数; 如果 SWDTSLPOFF 位为 0b0, 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 配置为 0b00。

在停止模式下, ADC 和 DAC 也会发生功耗, 除非在进入停止模式前将其禁止。要禁止 DAC, 需要将 DACR.DAE, DACR.DA1E, 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.CKSMRC 位来选择唤醒后的时钟。CKSMRC=0b1 时, 唤醒后的系统时钟源选择为 MRC; CKSMRC=0b0 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 的时钟频率之比为 1:1 或 2:1 或 4:1。若 HCLK 与任一外设时钟的频率之比超过 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)。

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

芯片在超低速模式下需要设置 PWC_PWRC1.STPDAS=0b11 后再进入 STOP 模式; 如果 PWC_PWRC1.STPDAS=0b00 时进入 STOP 模式, 在 STOP 模式中芯片会消耗更多的电流。

通过非可屏蔽中断解除停止模式时, 需要设置 INTC_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 INTC_WKEN 寄存器的相应位使能该中断的唤醒许可。

● 进入停止模式

在 PWC_STPMCR.STOP=0b1, PWC_PWRC0.PWDN=0b0 时执行 WFI 指令即可进入停止模式。

芯片的外设和时钟源在停止模式下的状态请参见表“[低功耗模式的运行条件及各模块在低功耗模式下的状态](#)”。

● 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: 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、实时时钟闹钟中断、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。

6.2.4.5 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 SWDT 自动启动时:

- 如果 ICG0.SWDTSLPOFF 为 0b1, 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 ICG0.SWDTSLPOFF 为 0b0, 芯片将进入停止模式而非掉电模式, 专用看门狗的振荡器和专用看门狗将继续运行。

通过软件刷新寄存器设置 SWDT 启动时:

- 如果 SWDT_CR.SLPOFF 为 0b1, 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 SWDT_CR.SLPOFF 为 0b0, 芯片将进入停止模式而非掉电模式, 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。

芯片的 VCAP_1 管脚使用的电容如下：VCAP_1 管脚管脚可以使用 0.094μF、0.1μF、0.2μF 或者 0.22μF 电容。

从掉电模式唤醒时，内核电压建立过程中需要给 VCAP_1 充电。一方面，较小的 VCAP_1 总容量能够缩短充电时间，为应用带来快速响应能力；另一方面，较大的 VCAP_1 总容量会延长充电时间，但是也提供更强的电磁兼容性（EMC）。用户可以根据电磁兼容性和系统响应速度的要求，选择较小或者较大的电容。芯片的 VCAP_1 总容量必须与 PWC_PWRC1.PDTS 位的赋值相匹配。VCAP_1 的总容量为 0.2μF 或者 0.22μF 时，需要在进入掉电模式之前确保 PWC_PWRC1.PDTS 位清零。VCAP_1 的总容量为 0.094μF 或者 0.1μF 时，需要在进入掉电模式之前确保 PWC_PWRC1.PDTS 位置位。

通过设置 PWC_PWRC0.PDMDS，可以进一步降低掉电模式的功耗。掉电模式的子模式如下表所示。

- 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，请参见《数据手册》的“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 6-8 掉电模式子模式

掉电模式	PDMDS	功耗	唤醒时间	说明
掉电模式 1	0b00	I_{PD1}	T_{PD1}	VCC 域电源电压检测单元有效
掉电模式 2	0b01	I_{PD2}	T_{PD2}	VCC 域 POR、PDR 检测电路有效，BOR、PVD1、PVD2 无效
掉电模式 3	0b10	I_{PD3}	T_{PD3}	VCC 域 POR、PDR，BOR、PVD1、PVD2 无效
掉电模式 4	0b11	I_{PD4}	T_{PD4}	VCC 域 POR、PDR，BOR、PVD1、PVD2 无效

功耗和唤醒时间的关系： $I_{PD1} > I_{PD2} > I_{PD3} = I_{PD4}$ ， $T_{PD1} < T_{PD2} < T_{PD3} < T_{PD4}$

- 进入掉电模式

在 PWC_STPMCR.STOP=0b1，PWC_PWRC0.PWDN=0b1 时执行 WFI 指令即可进入掉电模式。

● 解除掉电模式

掉电模式可以通过掉电模式唤醒事件或者复位解除。能够用于唤醒掉电模式的复位有管脚复位、上电复位和欠压复位。能够用于掉电模式唤醒的事件包括：WKUPn_0/1/2/3 (n=0/1/2/3) 唤醒事件、实时时钟的闹钟和定时事件、电压检测 1 唤醒事件、电压检测 2 唤醒事件、唤醒定时器唤醒事件。

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

在掉电模式 3 中 POR、PDR，BOR、PVD1、PVD2 电路都处于无效状态，从掉电模式 3 唤醒后，除 PWC_PDWKFO/PWC_PDWKFI/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_PDWKFI) 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_PDWKFO、PWC_PDWKFI、PWC_PVDCR0、PWC_PVDCR1、PWC_PVDFCR、PWC_PVDLCR、PWC_PVDICR、PWC_PVDDSR
掉电模式 3	PWC_PDWKFO、PWC_PDWKFI、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=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=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=0b01，芯片管脚将保持进入掉电模式之前的状态，唤醒后即使用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 的结构框图如下图所示。

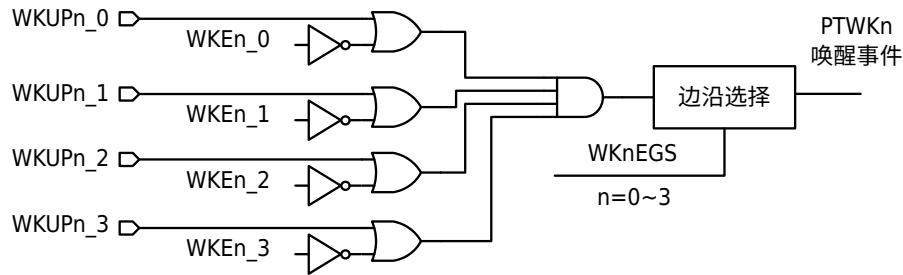


图 6-11 PTWK_n 结构框图

6.2.5 降低功耗的方法

6.2.5.1 降低系统时钟速度

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

6.2.5.2 关闭不使用的时钟源

芯片的系统时钟有以下时钟源：

- 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。详细信息请参见《时钟控制器（CMU）》。

6.2.5.3 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。

6.2.5.4 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。下表是 RAM 模块与掉电控制位的对应关系，通过设置 PWC_RAMPC0.RAMPDC_n 位可以使对应的 RAM 掉电。

表 6-10 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： 0x20000000~0x20007FFF	PWC_RAMPC0.RAMPDC0
SRAMH	系统 RAM，地址范围： 0x1FFF8000~0x1FFFFFFF	PWC_RAMPC0.RAMPDC10
Ret SRAM	掉电模式保持 RAM，地址范围： 0x200F0000~0x200F0FFF	PWR_PWRC5.SRAMBSD
CAN RAM	CAN 用 RAM	PWC_PRAMLPC.PRAMDC0
CACHERAM	Cache 用 RAM	PWC_PRAMLPC.PRAMDC2

6.3 寄存器

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 6-11 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_RAMOPM、PWC_PRAMLPC、RMU_RSTF0、RMU_FRST0、RMU_FRSTR1、RMU_FRST2、RMU_FRST3、RMU_PRSTCR0、PWC_WKTC
PWC_FPRC.FPRCB3	PWC_PVDCR0、PWC_PVDCR1、PWC_PVDFCR、PWC_PVDLCR、PWC_PVDICR、PWC_PVDDSR
PWC_FCG0PC.PRT0	PWC_FCG0

6.3.1 寄存器总表

PWC 基地址 1: 0x4004CC00

表 6-12 电源控制 (PWC) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	PWC_PWRC0	电源模式控制寄存器 0	8
0x04	PWC_PWRC1	电源模式控制寄存器 1	8
0x08	PWC_PWRC2	电源模式控制寄存器 2	8
0x0C	PWC_PWRC3	电源模式控制寄存器 3	8
0x10	PWC_PWRC4	电源模式控制寄存器 4	8
0x14	PWC_PVDCR0	PVD 控制寄存器 0	8
0x18	PWC_PVDCR1	PVD 控制寄存器 1	8
0x1C	PWC_PVDFCR	PVD 滤波控制寄存器	8
0x20	PWC_PVDLCR	PVD 电平控制寄存器	8
0x28	PWC_PDWKE0	掉电唤醒使能寄存器 0	8
0x2C	PWC_PDWKE1	掉电唤醒使能寄存器 1	8
0x40	PWC_PDWKE2	掉电唤醒使能寄存器 2	8

偏移地址	寄存器	描述	位宽
0x34	PWC_PDWKES	掉电唤醒事件边沿选择寄存器	8
0x38	PWC_PDWKFO	掉电唤醒标志寄存器 0	8
0x3C	PWC_PDWKFI	掉电唤醒标志寄存器 1	8
0x40	PWC_PWRC5	电源模式控制寄存器 5	8
0x44	PWC_PWRC6	电源模式控制寄存器 6	8
0xC0	PWC_PVDICR	PVD 中断控制寄存器	8
0xC4	PWC_PVDDSR	PVD 检测状态寄存器	8
0xC8	PWC_RAMPC0	RAM 功耗控制寄存器 0	32
0xCC	PWC_RAMOPM	RAM 运行条件寄存器	16
0xD0	PWC_PRAMLPC	外设 RAM 低功耗控制寄存器	32

PWC 基地址 2: 0x40054000

表 6-13 电源控制 (PWC) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00C	PWC_STPMCR	STOP 模式控制寄存器	16
0x3FE	PWC_FPRC	功能保护控制寄存器	8

PWC 基地址 3: 0x4004C400

表 6-14 电源控制 (PWC) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x50	PWC_WKTCR	唤醒计时器控制寄存器	16
0x00	PWC_FCG0	功能时钟控制 0	32
0x04	PWC_FCG1	功能时钟控制 1	32
0x08	PWC_FCG2	功能时钟控制 2	32
0x0C	PWC_FCG3	功能时钟控制 3	32
0x10	PWC_FCG0PC	PWC_FCG0 保护控制	32

6.3.2 电源模式控制寄存器 0 (PWC_PWRC0)

Offset	Bit Position							
0x00	7	6	5	4	3	2	1	0
Reset	0x00							
Name	PWDN	Reserved	IORTN		Reserved		PDMDS	
Access	RW		RW				RW	

位/位域	标记	位名	功能描述	读写
7	PWDN	掉电模式控制位	0b0: 掉电模式无效 0b1: 掉电模式使能	RW
6	Reserved	保留	-	-
5:4	IORTN	掉电模式下 IO 保持控制	0b00: 掉电模式中 IO 保持状态, 掉电唤醒后硬件解除 IO 保持状态 0b01: 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 0b10/0b11: 掉电模式中及掉电唤醒后 IO 为高阻	RW
3:2	Reserved	保留	-	-
1:0	PDMDS	掉电模式选择控制	0b00: 掉电模式 1 0b01: 掉电模式 2 0b10: 掉电模式 3 0b11: 掉电模式 4	RW

6.3.3 电源模式控制寄存器 1 (PWC_PWRC1)

Offset	Bit Position							
0x04	7	6	5	4	3	2	1	0
Reset	0x00							
Name	STPDAS		Reserved		PDTS	VHRCSD	VPLLSD	
Access	RW				RW	RW	RW	

位/位域	标记	位名	功能描述	读写
7:6	STPDAS	STOP 模式 LDO 驱动选择	0b00: 高速模式模式进入 STOP 模式时设定的驱动能力 0b11: 超低速模式进入 STOP 模式时设定的驱动能力 - 其他: 禁止设置	RW
5:4	Reserved	保留	-	-
3	PDTS	掉电唤醒时间控制位	0b0: 当 VCAP_1 总电容是 0.2μF 时或者 0.22μF 时, 进入掉电模式之前, PDTS 位必须清零 0b1: VCAP_1 总电容是 0.094μF 时或者 0.1μF 时, 进入掉电模式之前, PDTS 位必须置位	RW
2	VHRCSD	HRC 电源关闭	0b0: HRC 电源使能 0b1: HRC 电源关闭  说明 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
1:0	VPLLSD	PLL 电源关闭	0b00: PLL 电源使能 0b11: PLL 电源关闭 - 其他: 禁止设置  说明 在 PLLH 关闭并等待 50μs 后, 设置 VPLLSD=0b11 后关闭 PLL 用电源, 进一步降低功耗。设置 VPLLSD=0b00 后, 需要等待 25μs 后再启动 PLLH 模块。	RW

6.3.4 电源模式控制寄存器 2 (PWC_PWRC2)

Offset	Bit Position							
0x08	7	6	5	4	3	2	1	0
Reset	0xF0							
Name	Reserved		DVS		Reserved			
Access			RW					

位/位域	标记	位名	功能描述	读写
7:6	Reserved	保留	所有位读出的 1，写入时必须写 1	-
5:4	DVS	动作模式下电压选择	0b00: 选择超低速动作电压 0b11: 选择高速动作模式电压 - 其他: 禁止设置	RW
3:0	Reserved	保留	-	-

6.3.5 电源模式控制寄存器 3 (PWC_PWRC3)

Offset	Bit Position							
0x0C	7	6	5	4	3	2	1	0
Reset	0x0F							
Name	Reserved				DDAS			
Access					RW			

位/位域	标记	位名	功能描述	读写
7:4	Reserved	保留	-	-
3:0	DDAS	电源驱动选择	0x0: 超低速驱动模式 0xF: 高速动作模式 - 其他: 禁止设置	RW

6.3.6 电源模式控制寄存器 4 (PWC_PWRC4)

Offset	Bit Position							
0x10	7	6	5	4	3	2	1	0
Reset	0x00							
Name	ADBUFE	Reserved						
Access	RW							

位/位域	标记	位名	功能描述	读写
7	ADBUFE	内部电压采样使能	● 0b0: 无效 ● 0b1: 有效  说明 使用 ADC 采样芯片内部电压时，该位需要设置为 1。	RW
6:0	Reserved	保留	-	-

6.3.7 掉电唤醒使能寄存器 0 (PWC_PDWKE0)

Offset	Bit Position							
0x28	7	6	5	4	3	2	1	0
Reset	0x00							
Name	WKE13	WKE12	WKE11	WKE10	WKE03	WKE02	WKE01	WKE00
Access	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
7	WKE13	WKUP1_3 唤醒事件使能	● 0b0: 禁止 ● 0b1: 使能	RW
6	WKE12	WKUP1_2 唤醒事件使能	● 0b0: 禁止 ● 0b1: 使能	RW
5	WKE11	WKUP1_1 唤醒事件使能	● 0b0: 禁止 ● 0b1: 使能	RW
4	WKE10	WKUP1_0 唤醒事件使能	● 0b0: 禁止 ● 0b1: 使能	RW
3	WKE03	WKUP0_3 唤醒事件使能	● 0b0: 禁止 ● 0b1: 使能	RW
2	WKE02	WKUP0_2 唤醒事件使能	● 0b0: 禁止 ● 0b1: 使能	RW
1	WKE01	WKUP0_1 唤醒事件使能	● 0b0: 禁止 ● 0b1: 使能	RW
0	WKE00	WKUP0_0 唤醒事件使能	● 0b0: 禁止 ● 0b1: 使能	RW

6.3.8 掉电唤醒使能寄存器 1 (PWC_PDWKE1)

Offset	Bit Position							
0x2C	7	6	5	4	3	2	1	0
Reset	0x00							
Name	WKE33	WKE32	WKE31	WKE30	WKE23	WKE22	WKE21	WKE20
Access	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
7	WKE33	WKUP3_3 唤醒事件使能	● 0b0: 禁止 ● 0b1: 使能	RW
6	WKE32	WKUP3_2 唤醒事件使能	● 0b0: 禁止 ● 0b1: 使能	RW
5	WKE31	WKUP3_1 唤醒事件使能	● 0b0: 禁止 ● 0b1: 使能	RW
4	WKE30	WKUP3_0 唤醒事件使能	● 0b0: 禁止 ● 0b1: 使能	RW
3	WKE23	WKUP2_3 唤醒事件使能	● 0b0: 禁止 ● 0b1: 使能	RW
2	WKE22	WKUP2_2 唤醒事件使能	● 0b0: 禁止 ● 0b1: 使能	RW
1	WKE21	WKUP2_1 唤醒事件使能	● 0b0: 禁止 ● 0b1: 使能	RW
0	WKE20	WKUP2_0 唤醒事件使能	● 0b0: 禁止 ● 0b1: 使能	RW

6.3.9 掉电唤醒使能寄存器 2 (PWC_PDWKE2)

Offset	Bit Position							
0x40	7	6	5	4	3	2	1	0
Reset	0x00							
Name	WKTMWKE	Reserved	RTCALMWKE	RTCPRDWKE	Reserved		VD2WKE	VD1WKE
Access	RW		RW	RW			RW	RW

位/位域	标记	位名	功能描述	读写
7	WKTMWKE	WKT M 唤醒事件使能	0b0: 禁止 0b1: 使能	RW
6	Reserved	保留	-	-
5	RTCALMWKE	RTC 闹钟唤醒事件使能	0b0: 禁止 0b1: 使能	RW
4	RTCPRDWKE	RTC 周期唤醒事件使能	0b0: 禁止 0b1: 使能	RW
3:2	Reserved	保留	-	-
1	VD2WKE	PVD2 唤醒事件使能	0b0: 禁止 0b1: 使能	RW
0	VD1WKE	PVD1 唤醒事件使能	0b0: 禁止 0b1: 使能	RW

6.3.10 掉电唤醒事件边沿选择寄存器 (PWC_PDWKES)

Offset	Bit Position							
0x34	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved		VD2EGS	VD1EGS	WK3EGS	WK2EGS	WK1EGS	WK0EGS
Access			RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
7:6	Reserved	保留	-	-
5	VD2EGS	VD2 边沿选择	0b0: VCC<VPVD2 0b1: VCC>VPVD2	RW
4	VD1EGS	VD1 边沿选择	0b0: VCC<VPVD1 0b1: VCC>VPVD1	RW
3	WK3EGS	PTWK3 边沿选择	0b0: 下降沿 0b1: 上升沿	RW
2	WK2EGS	PTWK2 边沿选择	0b0: 下降沿 0b1: 上升沿	RW
1	WK1EGS	PTWK1 边沿选择	0b0: 下降沿 0b1: 上升沿	RW
0	WK0EGS	PTWK0 边沿选择	0b0: 下降沿 0b1: 上升沿	RW

6.3.11 掉电唤醒标志寄存器 0 (PWC_PDWKFO)

Offset	Bit Position							
0x38	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved		VD2WKF	VD1WKF	PTWK3F	PTWK2F	PTWK1F	PTWK0F
Access			RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
7:6	Reserved	保留	-	-
5	VD2WKF	PVD2 唤醒标志位	掉电唤醒后, 需要写 0 清除标志 读取: ● 0b0: 未发生唤醒事件 ● 0b1: 发生唤醒事件	RW
4	VD1WKF	PVD1 唤醒标志位	掉电唤醒后, 需要写 0 清除标志 读取: ● 0b0: 未发生唤醒事件 ● 0b1: 发生唤醒事件	RW
3	PTWK3F	PTWK3 唤醒标志位	掉电唤醒后, 需要写 0 清除标志 读取: ● 0b0: 未发生唤醒事件 ● 0b1: 发生唤醒事件	RW
2	PTWK2F	PTWK2 唤醒标志位	掉电唤醒后, 需要写 0 清除标志 读取: ● 0b0: 未发生唤醒事件 ● 0b1: 发生唤醒事件	RW
1	PTWK1F	PTWK1 唤醒标志位	掉电唤醒后, 需要写 0 清除标志 读取: ● 0b0: 未发生唤醒事件 ● 0b1: 发生唤醒事件	RW
0	PTWK0F	PTWK0 唤醒标志位	掉电唤醒后, 需要写 0 清除标志 读取: ● 0b0: 未发生唤醒事件 ● 0b1: 发生唤醒事件	RW

6.3.12 掉电唤醒标志寄存器 1 (PWC_PDWKF1)

Offset	Bit Position							
0x3C	7	6	5	4	3	2	1	0
Reset	0x00							
Name	WKTMWKF	Reserved	RTCALMWKF	RTCPRDWKF	Reserved			
Access	RW		RW	RW				

位/位域	标记	位名	功能描述	读写
7	WKTMWKF	WKTM 唤醒标志位	掉电唤醒后，需要写 0 清除标志 读取： ● 0b0：未发生唤醒事件 ● 0b1：发生唤醒事件	RW
6	Reserved	保留	-	-
5	RTCALMWKF	RTC 闹钟唤醒标志位	掉电唤醒后，需要写 0 清除标志 读取： ● 0b0：未发生唤醒事件 ● 0b1：发生唤醒事件	RW
4	RTCPRDWKF	RTC 周期唤醒标志位	掉电唤醒后，需要写 0 清除标志 读取： ● 0b0：未发生唤醒事件 ● 0b1：发生唤醒事件	RW
3:0	Reserved	保留	-	-

6.3.13 电源模式控制寄存器 5 (PWC_PWRC5)

Offset	Bit Position							
0x40	7	6	5	4	3	2	1	0
Reset	0x00							
Name	CSDIS	Reserved					SRAMBSD	VVDRSD
Access	RW						RW	RW

位/位域	标记	位名	功能描述	读写
7	CSDIS	电流源无效控制	0b0: 电流源有效 0b1: 电流源无效  说明 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
6:2	Reserved	保留	-	-
1	SRAMBSD	Ret SRAM (保持 RAM) 掉电控制	0b0: Ret SRAM 不掉电 0b1: Ret SRAM 掉电	RW
0	VVDRSD	RLDO 关闭控制位	0b0: RLDO 有效 0b1: RLDO 无效  说明 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

6.3.14 电源模式控制寄存器 6 (PWC_PWRC6)

Offset	Bit Position							
0x44	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved						RTCKSEL	
Access							RW	

位/位域	标记	位名	功能描述	读写
7:2	Reserved	保留	-	-
1:0	RTCKSEL	RTC 时钟选择	0b00: LRC 0b01: XTAL 小数分频 - 其他: 禁止设置	RW

6.3.15 RAM 功耗控制寄存器 0 (PWC_RAMPC0)

Offset	Bit Position																																	
0xC8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved																						RAMPDC10	Reserved										RAMPDC0
Access																							RW											RW

位/位域	标记	位名	功能描述	读写
31:11	Reserved	保留	-	-
10	RAMPDC10	RAM 掉电控制位 10	● 0b0: SRAMH 动作 ● 0b1: SRAMH 掉电	RW
9:1	Reserved	保留	-	-
0	RAMPDC0	RAM 掉电控制位 0	● 0b0: SRAM0 动作 ● 0b1: SRAM0 掉电	RW

6.3.16 RAM 运行条件寄存器 (PWC_RAMOPM)

Offset	Bit Position															
0xCC	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x8043															
Name	PWC_RAMOPM															
Access	RW															

位/位域	标记	位名	功能描述	读写
15:0	PWC_RAMOPM	RAM 动作模式选择位	● 芯片工作在高速运行模式时，PWC_RAMOPM 设置成 0x8043 ● 芯片工作在超低速模式动作时，PWC_RAMOPM 设置成 0x9062	RW

6.3.17 外设 RAM 低功耗控制寄存器 (PWC_PRAMLPC)

Offset	Bit Position																																
0xD0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																														PRAMPDC2	Reserved	PRAMPDC0
Access																															RW		RW

位/位域	标记	位名	功能描述	读写
31:3	Reserved	保留	-	-
2	PRAMPDC2	CACHE_RAM 掉电控制	● 0b0: RAM 动作 ● 0b1: RAM 掉电	RW
1	Reserved	保留	-	-
0	PRAMPDC0	CAN RAM 掉电控制	● 0b0: RAM 动作 ● 0b1: RAM 掉电	RW

6.3.18 PVD 控制寄存器 0 (PWC_PVDCR0)

Offset	Bit Position							
0x14	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved	PVD2EN	PVD1EN	Reserved				EXVCCINEN
Access		RW	RW					RW

位/位域	标记	位名	功能描述	读写
7	Reserved	保留	-	-
6	PVD2EN	电压检测 2 允许	0b0: 电压检测 2 电路无效 0b1: 电压检测 2 电路有效	RW
5	PVD1EN	电压检测 1 允许	0b0: 电压检测 1 电路无效 0b1: 电压检测 1 电路有效	RW
4:1	Reserved	保留	-	-
0	EXVCCINEN	外部 VCC 电压输入使能	0b0: 外部 VCC 电压输入无效 0b1: 外部 VCC 电压输入有效	RW

6.3.19 PVD 控制寄存器 1 (PWC_PVDCR1)

Offset	Bit Position							
0x18	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved	PVD2CMPOE	PVD2IRS	PVD2IRE	Reserved	PVD1CMPOE	PVD1IRS	PVD1IRE
Access		RW	RW	RW		RW	RW	RW

位/位域	标记	位名	功能描述	读写
7	Reserved	保留	-	-
6	PVD2CMPOE	PVD2 比较结果输出使能	0b0: 禁止输出比较器 2 的比较结果 0b1: 允许输出比较器 2 的比较结果	RW
5	PVD2IRS	PVD2 中断复位选择	0b0: 在 VCC 变化满足检测条件时产生 PVD2 中断 0b1: 在下降过程中经过 VPVD2 产生 PVD2 复位  说明 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 位和 PVD2IRS 位同时置 0。	RW
4	PVD2IRE	PVD2 中断复位使能	0b0: 禁止 0b1: 允许  说明 请在 PWC_PVDCR0.PVD2EN 位为 1 且 PVD2CMPOE 位为 1 的情况下将该位写 1。	RW
3	Reserved	保留	-	-
2	PVD1CMPOE	PVD1 比较结果输出使能	0b0: 禁止输出比较器 1 的比较结果 0b1: 允许输出比较器 1 的比较结果	RW
1	PVD1IRS	PVD1 中断复位选择	0b0: 在 VCC 变化满足检测条件时产生 PVD1 中断 0b1: 在下降过程中经过 VPVD1 产生 PVD1 复位  说明 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 位和 PVD2IRS 位同时置 0。	RW
0	PVD1IRE	PVD1 中断复位使能	0b0: 禁止 0b1: 允许  说明 请在 PWC_PVDCR0.PVD1EN 位为 1 且 PVD1CMPOE 位为 1 的情况下将该位写 1。	RW

6.3.20 PVD 滤波控制寄存器 (PWC_PVDFCR)

Offset	Bit Position							
0x1C	7	6	5	4	3	2	1	0
Reset	0x11							
Name	Reserved	PVD2NFCKS		PVD2NFDIS	Reserved	PVD1NFCKS		PVD1NFDIS
Access		RW		RW		RW		RW

位/位域	标记	位名	功能描述	读写
7	<i>Reserved</i>	<i>保留</i>	-	-
6:5	PVD2NFCKS	PVD2 数字滤波采样时钟选择	0b00: 2 个滤波时钟周期 0b01: 4 个滤波时钟周期 0b10: 8 个滤波时钟周期 0b11: 16 个滤波时钟周期 滤波时钟周期=LRC 周期/8  说明 只能在 PVD2NFDIS 位为 1 时改写该位。	RW
4	PVD2NFDIS	PVD2 数字滤波器屏蔽	0b0: 数字滤波器有效 0b1: 数字滤波器无效	RW
3	<i>Reserved</i>	<i>保留</i>	-	-
2:1	PVD1NFCKS	PVD1 数字滤波采样时钟选择	0b00: 2 个滤波时钟周期 0b01: 4 个滤波时钟周期 0b10: 8 个滤波时钟周期 0b11: 16 个滤波时钟周期 滤波时钟周期=LRC 周期/8  说明 只能在 PVD1NFDIS 位为 1 时改写该位。	RW
0	PVD1NFDIS	PVD1 数字滤波器屏蔽	0b0: 数字滤波器有效 0b1: 数字滤波器无效	RW

6.3.21 PVD 电平控制寄存器 (PWC_PVDLCR)

Offset	Bit Position							
0x20	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved	PVD2LVL			Reserved	PVD1LVL		
Access		RW				RW		

位/位域	标记	位名	功能描述	读写
7	Reserved	保留	-	-
6:4	PVD2LVL	PVD2 阈值电压选择	● 0b000: 2.1V ● 0b001: 2.3V ● 0b010: 2.5V ● 0b011: 2.6V ● 0b100: 2.7V ● 0b101: 2.8V ● 0b110: 2.9V ● 0b111: 1.1V (仅在 PWC_PVDCR0.EXVCCINEN=0b1 时有效, 其它情况请不要设定该值)	RW
3	Reserved	保留	-	-
2:0	PVD1LVL	PVD1 阈值电压选择	● 0b000: 2.0V ● 0b001: 2.1V ● 0b010: 2.3V ● 0b011: 2.5V ● 0b100: 2.6V ● 0b101: 2.7V ● 0b110: 2.8V ● 0b111: 2.9V	RW

6.3.22 PVD 中断控制寄存器 (PWC_PVDICR)

Offset	Bit Position							
0xC0	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved	PVD2EDGS		Reserved		PVD1EDGS		Reserved
Access		RW				RW		

位/位域	标记	位名	功能描述	读写
7	Reserved	保留	-	-
6:5	PVD2EDGS	PVD2 检测条件选择	0b00: 当检测到 $VCC < VPVD2$ (下降) 时 0b01: 当检测到 $VCC \geq VPVD2$ (上升) 时 0b10: 当检测到 $VCC < VPVD2$ (下降) 时或者当检测到 $VCC \geq VPVD2$ (上升) 时 0b11: 禁止设置	RW
4:3	Reserved	保留	-	-
2:1	PVD1EDGS	PVD1 检测条件选择	0b00: 当检测到 $VCC < VPVD1$ (下降) 时 0b01: 当检测到 $VCC \geq VPVD1$ (上升) 时 0b10: 当检测到 $VCC < VPVD1$ (下降) 时或者当检测到 $VCC \geq VPVD1$ (上升) 时 0b11: 禁止设置	RW
0	Reserved	保留	-	-

6.3.23 PVD 检测状态寄存器 (PWC_PVDDSR)

Offset	Bit Position							
0xC4	7	6	5	4	3	2	1	0
Reset	0x11							
Name	Reserved		PVD2DETF LG	PVD2MON	Reserved		PVD1DETF LG	PVD1MON
Access			RW	RO			RW	RO

位/位域	标记	位名	功能描述	读写
7:6	Reserved	保留	-	-
5	PVD2DETF LG	PVD2 检测标志位	读取该位后写 0 清除本标志位；写 1 无效 读取： 0b0：PVD2 未检测到 VCC 经过 VPVD2 0b1：PVD2 检测到 VCC 经过 VPVD2  说明 当 PWC_PVDCR0.PVD2EN 位为 1 且 PWC_PVDCR1.PVD2CMPOE 位为 1 时，该位有效。	RW
4	PVD2MON	PVD2 监视位	0b0：VCC≤VPVD2 或者外部输入比较电压≤PVD2 内部基准电压 0b1：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	RO
3:2	Reserved	保留	-	-
1	PVD1DETF LG	PVD1 检测标志位	读取该位后写 0 清除本标志位；写 1 无效 读取： 0b0：PVD1 未检测到 VCC 经过 VPVD1 0b1：PVD1 检测到 VCC 经过 VPVD1  说明 当 PWC_PVDCR0.PVD1EN 位为 1 且 PWC_PVDCR1.PVD1CMPOE 位为 1 时，该位有效。	RW
0	PVD1MON	PVD1 监视位	0b0：VCC<VPVD1 0b1：PVD1 无效时或者 VCC>VPVD1	RO

6.3.24 功能时钟控制 0 (PWC_FCG0)

Offset	Bit Position																															
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFA0E																															
Name	Reserved				DCU4	DCU3	DCU2	DCU1	CRC	TRNG	HASH	AES	Reserved	CTC	AOS	FCM	DMA2	DMA1	KEYSCAN	Reserved	SRAMB	Reserved				SRAM0	Reserved				SRAMH	
Access					RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW		RW					RW					RW	

位/位域	标记	位名	功能描述	读写
31:28	Reserved	保留	所有位读出的 1，写入时必须写 1	-
27	DCU4	DCU4 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
26	DCU3	DCU3 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
25	DCU2	DCU2 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
24	DCU1	DCU1 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
23	CRC	CRC 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
22	TRNG	TRNG 功能控制	0b0: CPM 中的真随机发生器 TRNG 功能使能 0b1: CPM 中的真随机发生器 TRNG 功能禁止	RW
21	HASH	HASH 功能控制	0b0: CPM 中的安全散列算法模块 HASH 功能使能 0b1: CPM 中的安全散列算法模块 HASH 功能禁止	RW
20	AES	AES 功能控制	0b0: CPM 中的加解密算法处理器 AES 功能使能 0b1: CPM 中的加解密算法处理器 AES 功能禁止	RW
19	Reserved	保留	读出的 1，写入时必须写 1	-
18	CTC	CTC 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
17	AOS	AOS 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
16	FCM	FCM 功能控制	0b0: CMU 中的时钟频率测量模块 FCM 功能使能 0b1: CMU 中的时钟频率测量模块 FCM 功能禁止	RW
15	DMA2	DMA2 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
14	DMA1	DMA1 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
13	KEYSCAN	KEYSCAN 功能控制	0b0: 功能使能 0b1: 功能禁止	RW

位/位域	标记	位名	功能描述	读写
12:11	<i>Reserved</i>	<i>保留</i>	<i>所有位读出于 1，写入时必须写 1</i>	-
10	SRAMB	Ret SRAM 功能控制	● 0b0：功能使能 ● 0b1：功能禁止	RW
9:5	<i>Reserved</i>	<i>保留</i>	<i>读出于 0b10000，写入时必须写 0b10000</i>	-
4	SRAM0	SRAM0 功能控制	● 0b0：功能使能 ● 0b1：功能禁止	RW
3:1	<i>Reserved</i>	<i>保留</i>	<i>所有位读出于 1，写入时必须写 1</i>	-
0	SRAMH	SRAMH 功能控制	● 0b0：功能使能 ● 0b1：功能禁止	RW

6.3.25 功能时钟控制 1 (PWC_FCG1)

Offset	Bit Position																																																	
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																		
Reset	0xFFFFFFFF																																																	
Name	Reserved																	Reserved																																
Access															RW	RW	RW													RW	RW	RW																		

位/位域	标记	位名	功能描述	读写
31:19	Reserved	保留	所有位读出于 1，写入时必须写 1	-
18	SPI3	SPI3 功能控制	● 0b0：功能使能 ● 0b1：功能禁止	RW
17	SPI2	SPI2 功能控制	● 0b0：功能使能 ● 0b1：功能禁止	RW
16	SPI1	SPI1 功能控制	● 0b0：功能使能 ● 0b1：功能禁止	RW
15:6	Reserved	保留	所有位读出于 1，写入时必须写 1	-
5	I2C2	I2C2 功能控制	● 0b0：功能使能 ● 0b1：功能禁止	RW
4	I2C1	I2C1 功能控制	● 0b0：功能使能 ● 0b1：功能禁止	RW
3	QSPI	QSPI 功能控制	● 0b0：功能使能 ● 0b1：功能禁止	RW
2	Reserved	保留	读出于 1，写入时必须写 1	-
1	MCAN2	MCAN2 功能控制	● 0b0：功能使能 ● 0b1：功能禁止	RW
0	MCAN1	MCAN1 功能控制	● 0b0：功能使能 ● 0b1：功能禁止	RW

6.3.26 功能时钟控制 2 (PWC_FCG2)

Offset	Bit Position																																	
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0xFFFFFFFF																																	
Name	Reserved								TMRA_5	TMRA_4	TMRA_3	TMRA_2	TMRA_1	Reserved				EMB	Reserved	TMR0_2	TMR0_1	TMR4_3	TMR4_2	TMR4_1	Reserved								TMR6_2	TMR6_1
Access									RW	RW	RW	RW	RW					RW		RW	RW	RW	RW	RW									RW	RW

位/位域	标记	位名	功能描述	读写
31:25	Reserved	保留	所有位读出的 1，写入时必须写 1	-
24	TMRA_5	TimerA_5 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
23	TMRA_4	TimerA_4 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
22	TMRA_3	TimerA_3 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
21	TMRA_2	TimerA_2 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
20	TMRA_1	TimerA_1 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
19:16	Reserved	保留	所有位读出的 1，写入时必须写 1	-
15	EMB	EMB 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
14	Reserved	保留	读出的 1，写入时必须写 1	-
13	TMRO_2	Timer0_2 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
12	TMRO_1	Timer0_1 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
11	TMRA_3	Timer4_3 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
10	TMRA_2	Timer4_2 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
9	TMRA_1	Timer4_1 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
8:2	Reserved	保留	所有位读出的 1，写入时必须写 1	-
1	TMRA_2	Timer6_2 功能控制	0b0: 功能使能 0b1: 功能禁止	RW
0	TMRA_1	Timer6_1 功能控制	0b0: 功能使能 0b1: 功能禁止	RW

6.3.27 功能时钟控制 3 (PWC FCG3)

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved						USART6	USART5	USART4	USART3	USART2	USART1	Reserved		SMC	Reserved						CMP34	CMP12	Reserved		DAC	Reserved	ADC3	ADC2	ADC1		
Access							RW	RW	RW	RW	RW	RW			RW							RW	RW			RW		RW	RW	RW		

位/位域	标记	位名	功能描述	读写
31:26	Reserved	保留	所有位读出于 1，写入时必须写 1	-
25	USART6	USART6 功能控制	0b0：功能使能 0b1：功能禁止	RW
24	USART5	USART5 功能控制	0b0：功能使能 0b1：功能禁止	RW
23	USART4	USART4 功能控制	0b0：功能使能 0b1：功能禁止	RW
22	USART3	USART3 功能控制	0b0：功能使能 0b1：功能禁止	RW
21	USART2	USART2 功能控制	0b0：功能使能 0b1：功能禁止	RW
20	USART1	USART1 功能控制	0b0：功能使能 0b1：功能禁止	RW
19:17	Reserved	保留	所有位读出于 1，写入时必须写 1	-
16	SMC	SMC 功能控制	0b0：外部存储控制器（EXMC）的 SMC 控制器功能使能 0b1：外部存储控制器（EXMC）的 SMC 控制器功能禁止	RW
15:10	Reserved	保留	所有位读出于 1，写入时必须写 1	-
9	CMP34	CMP34 功能控制	0b0：功能使能 0b1：功能禁止	RW
8	CMP12	CMP12 功能控制	0b0：功能使能 0b1：功能禁止	RW
7:5	Reserved	保留	所有位读出于 1，写入时必须写 1	-
4	DAC	DAC 功能控制	0b0：功能使能 0b1：功能禁止	RW
3	Reserved	保留	读出于 1，写入时必须写 1	-
2	ADC3	ADC3 功能控制	0b0：功能使能 0b1：功能禁止	RW
1	ADC2	ADC2 功能控制	0b0：功能使能 0b1：功能禁止	RW

位/位域	标记	位名	功能描述	读写
0	ADC1	ADC1 功能控制	● 0b0: 功能使能 ● 0b1: 功能禁止	RW

6.3.28 PWC_FCG0 保护控制 (PWC_FCG0PC)

Offset	Bit Position																															
0x10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	FCG0PCWE																Reserved															PRT0
Access	RW																															RW

位/位域	标记	位名	功能描述	读写
31:16	FCG0PCWE	PWC_FCG0PC 写使能	在写入 PRT0 时，必须同时向 FCG0PCWE 写入 0xA5A5；读出恒为 0	RW
15:1	Reserved	保留	-	-
0	PRT0	PWC_FCG0 写使能控制位	0b0: PWC_FCG0 写无效 0b1: PWC_FCG0 写使能	RW

 **说明**

向本寄存器写入 0xA5A50001 可以将 PRT0 置 1；写入 0xA5A50000 可以将 PRT0 清 0；禁止写入其他值修改 PRT0。

6.3.29 功能保护控制寄存器 (PWC_FPRC)

Offset	Bit Position															
0x3FE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	PWC_FPRCWE								FPRCB7	FPRCB6	FPRCB5	FPRCB4	FPRCB3	Reserved	FPRCB1	FPRCB0
Access	RW								RW	RW	RW	RW	RW		RW	RW

位/位域	标记	位名	功能描述	读写
15:8	PWC_FPRCWE	PWC_FPRC 寄存器写使能	在写入 FPRCB7~FPRCB0 时，必须同时向 PWC_FPRCWE 写入 0xA5；读出恒为 0	RW
7	FPRCB7	FPRC 位 7	预留，读出为 0，写入时必须写 0	RW
6	FPRCB6	FPRC 位 6	预留，读出为 0，写入时必须写 0	RW
5	FPRCB5	FPRC 位 5	预留，读出为 0，写入时必须写 0	RW
4	FPRCB4	FPRC 位 4	预留，读出为 0，写入时必须写 0	RW
3	FPRCB3	FPRC 位 3	保护对象请参见“寄存器保护列表”： ● 0b0：对象写无效 ● 0b1：对象写使能	RW
2	Reserved	保留	预留，读出为 0，写入时必须写 0	-
1	FPRCB1	FPRC 位 1	保护对象请参见“寄存器保护列表”： ● 0b0：对象写无效 ● 0b1：对象写使能	RW
0	FPRCB0	FPRC 位 0	保护对象请参见“寄存器保护列表”： ● 0b0：对象写无效 ● 0b1：对象写使能	RW

6.3.30 STOP 模式控制寄存器 (PWC_STPMCR)

Offset	Bit Position															
0x00C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	STOP	EXBUSOE	Reserved												CKSMRC	FLNWT
Access	RW	RW													RW	RW

位/位域	标记	位名	功能描述	读写
15	STOP	STOP 模式选择位	0b0: STOP 模式无效 0b1: STOP 模式有效	RW
14	EXBUSOE	EXMC 地址总线和输出控制信号的输出使能	掉电模式和停止模式下外部总线的地址总线信号和输出控制信号的状态控制: 0b0: 高阻 0b1: 保持掉电模式和停止模式前的状态	RW
13:2	Reserved	保留	-	-
1	CKSMRC	时钟切换到 MRC 选项	0b0: 维持进入 STOP 模式之前的系统时钟及分频 0b1: STOP 模式唤醒时系统时钟切换到 MRC、SCKCFGR 寄存器被初始化	RW
0	FLNWT	FLASH 稳定等待控制	0b0: STOP 模式唤醒时等待 FLASH 稳定 0b1: STOP 模式唤醒时不等待 FLASH 稳定	RW

6.3.31 唤醒计时器控制寄存器 (PWC_WKTCR)

Offset	Bit Position															
0x50	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	WKTCE	WKCKS		WKOVF	WKTMCMP											
Access	RW	RW		RW	RW											

位/位域	标记	位名	功能描述	读写
15	WKTCE	WKTM 使能	0b0: WKTM 停止 0b1: 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
14:13	WKCKS	WKTM 时钟选择	0b00: 64Hz 时钟 0b01: XTAL32 0b10: LRC 0b11: 禁止设置	RW
12	WKOVF	计时器比较结果标志	0b0: 计数器与 WKTMCMP 值不一致 0b1: 计数器与 WKTMCMP 值一致 写 0 清除本标志位。	RW
11:0	WKTMCMP	WKTM 比较位	WKTM 计数器的比较值	RW

7 初始化配置 (ICG)

7.1 简介

芯片复位解除后，硬件电路会读取 FLASH 地址 0x00000400~0x0000045F 把数据加载到初始化配置寄存器。地址 0x00000408~0x0000040B，0x00000410~0x0000041F，0x00000438~0x0000045F 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 块 0 扇区 0；Flash 引导交换有效，并且 Flash 块 0 扇区 0 的 OTP 没有锁存（0x03000A80~0x03000A83 数据全 1）时，该区域位于 Flash 块 0 扇区 1；否则该区域存在 FLASH 块 0 扇区 0。用户可通过编程或擦除扇区 0 来修改初始化配置 (ICG) 寄存器。地址 0x00000420~0x00000437 为数据安全保护使能区，详细内容请参见《嵌入式 FLASH (EFM) 》中“功能说明-FLASH 编程和擦除操作-数据安全保护”。寄存器复位值由 FLASH 数据确定。

7.2 寄存器

7.2.1 寄存器总表

基地址：0x00000400

表 7-1 初始化配置 (ICG) 寄存器偏移地址

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

7.2.2 初始化配置寄存器 0 (ICG0)

Offset	Bit Position																																			
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reset	0xFFFFFFFF																																			
Name	Reserved		WDTSLPOFF		WDTWDPT				WDTCKS				WDTPERI		WDTITS		WDTAUS		Reserved		SWDTSLOFF		SWDTWDPT				SWDTCKS				SWDTPERI		SWDTITS		SWDTAUS	
Access			RO		RO				RO				RO		RO		RO				RO		RO				RO				RO		RO		RO	

位/位域	标记	位名	功能描述	读写
31:29	Reserved	保留	-	-
28	WDTSLPOFF	WDT 在 Sleep 模式下计数停止	0b0: WDT 在睡眠模式下计数不停止 0b1: WDT 在睡眠模式下计数停止	RO
27:24	WDTWDPT	刷新允许区域计数值百分比	WDT 计数值刷新允许区间: 0b0000: 0%~100% 0b0001: 0%~25% 0b0010: 25%~50% 0b0011: 0%~50% 0b0100: 50%~75% 0b0101: 0%~25%, 50%~75% 0b0110: 25%~75% 0b0111: 0%~75% 0b1000: 75%~100% 0b1001: 0%~25%, 75%~100% 0b1010: 25%~50%, 75%~100% 0b1011: 0%~50%, 75%~100% 0b1100: 50%~100% 0b1101: 0%~25%, 50%~100% 0b1110: 25%~100% 0b1111: 0%~100%	RO
23:20	WDTCKS	WDT 计数时钟	0b0010: PCLK3/4 0b0110: PCLK3/64 0b0111: PCLK3/128 0b1000: PCLK3/256 0b1001: PCLK3/512 0b1010: PCLK3/1024 0b1011: PCLK3/2048 0b1101: PCLK3/8192 - 其他: Reserved	RO
19:18	WDTPERI	WDT 计数溢出周期	0b00: 256 周期 0b01: 4096 周期 0b10: 16384 周期 0b11: 65536 周期	RO

位/位域	标记	位名	功能描述	读写
17	WDTITS	WDT 中断选择	0b0: 中断请求 0b1: 复位请求	RO
16	WDTAUTS	WDT 自动启动	0b0: 复位后, WDT 自动启动 (硬件启动) 0b1: 复位后, WDT 停止状态	RO
15:13	Reserved	保留	-	-
12	SWDTSLPPOFF	SWDT 在 Sleep/Stop 模式下计数停止	0b0: SWDT 在睡眠模式和停止模式下计数不停止 0b1: SWDT 在睡眠模式和停止模式下计数停止	RO
11:8	SWDTWDPT	刷新允许区域计数值百分比	SWDT 计数值刷新允许区间: 0b0000: 0%~100% 0b0001: 0%~25% 0b0010: 25%~50% 0b0011: 0%~50% 0b0100: 50%~75% 0b0101: 0%~25%, 50%~75% 0b0110: 25%~75% 0b0111: 0%~75% 0b1000: 75%~100% 0b1001: 0%~25%, 75%~100% 0b1010: 25%~50%, 75%~100% 0b1011: 0%~50%, 75%~100% 0b1100: 50%~100% 0b1101: 0%~25%, 50%~100% 0b1110: 25%~100% 0b1111: 0%~100%	RO
7:4	SWDTCKS	SWDT 计数时钟	0b0000: SWDTCLK 0b0100: SWDTCLK/16 0b0101: SWDTCLK/32 0b0110: SWDTCLK/64 0b0111: SWDTCLK/128 0b1000: SWDTCLK/256 0b1011: SWDTCLK/2048 - 其他: Reserved	RO
3:2	SWDTPERI	SWDT 计数溢出周期	0b00: 256 周期 0b01: 4096 周期 0b10: 16384 周期 0b11: 65536 周期	RO
1	SWDTITS	SWDT 中断选择	0b0: 中断请求 0b1: 复位请求	RO
0	SWDTAUTS	SWDT 自动启动	0b0: 复位后, SWDT 自动启动 (硬件启动) 0b1: 复位后, SWDT 停止状态	RO

7.2.3 初始化配置寄存器 1 (ICG1)

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved														BORDIS	BOR_LEV	Reserved						HRCSTOP	Reserved						HRCFREQSEL		
Access															RO	RO							RO							RO		

位/位域	标记	位名	功能描述	读写
31:19	Reserved	保留	-	-
18	BORDIS	BOR 动作选择	0b0: 复位后允许 BOR 动作 0b1: 复位后禁止 BOR 动作	RO
17:16	BOR_LEV	BOR 阈值电压选择	0b00: 1.9V 0b01: 2.0V 0b10: 2.1V 0b11: 2.3V	RO
15:9	Reserved	保留	-	-
8	HRCSTOP	HRC 振荡停止位	0b0: HRC 振荡 0b1: HRC 停止	RO
7:1	Reserved	保留	所有位读出为 1, 写入时必须写 1	-
0	HRCFREQSEL	HRC 频率选择	0b0: 20MHz 0b1: 16MHz	RO

7.2.4 初始化配置寄存器 3 (ICG3)

Offset	Bit Position																															
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved																DBUSPRT															
Access																	RO															

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	DBUSPRT	对 128KBytes 区域 D-BUS 读保护功能	当 DBUSPRT[15:0]=0x4450 时, 对 0x00000000~0x0001FFFF 区域的 D-BUS 读保护使能; 否则 D-BUS 读保护无效。	RO

8 嵌入式 Flash (EFM)

8.1 简介

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

8.2 主要特性

- 最大 9KBytes 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

8.3 功能说明

8.3.1 嵌入式 Flash

Flash 具有以下主要特性：

- 容量高达 256KBytes，由一块 256KBytes 的 Flash 构成，共 32 个扇区，每个扇区为 8KBytes。扇区 0 为可配置为 OTP 区域。
- OTP（One Time Program）区域共 9KBytes，其中 8KBytes 配置在地址 0x00000000~0x0001FFFF，1KBytes 配置在地址 0x03000C00~0x03000FFF。
地址 0x03000A80~0x03000AFF 为 OTP 数据锁存区。0x03000A00~0x03000A03 为 OTP 使能区。
- 128 位宽数据读取，读缓存 128 位宽缓冲，加速代码执行。
- 编程单位为 4Bytes。擦除单位为 8KBytes。

	
0x03002040	数据安全配置区
0x03002000	引导交换使能
0x03000FFF	
0x03000A00	OTP区域
0x0003FFFF	
0x0003E000	扇区31
0x00001FFF	
0x00000000	扇区0

图 8-1 Flash 地址分布

OTP 区域 0x03000A00~0x03000FFF 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 0x03002000~0x03002003 和数据安全配置区 0x03002040~0x0300204B 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 0x03002040~0x0300204B 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。

8.3.2 读接口

8.3.2.1 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 动作频率设定合适的等待周期数（EFM_FRMC.FLWT）。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。等待周期数如下表所示。

表 8-1 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
150MHz<HCLK≤200MHz	EFM_FRMC.FLWT=0b0011
100MHz<HCLK≤150MHz	EFM_FRMC.FLWT=0b0010
50MHz<HCLK≤100MHz	EFM_FRMC.FLWT=0b0001
HCLK≤50MHz	EFM_FRMC.FLWT=0b0000

CPU 频率提高步骤：

- 步骤 1.** 向寄存器 EFM_FRMC.FLWT 写入新的读等待周期设定值。
- 步骤 2.** 读取寄存器 EFM_FRMC，检查新的等待周期是否设定成功。
- 步骤 3.** 通过设定系统时钟源切换寄存器 CMU_CKSWR.CKSW 或者系统时钟配置寄存器 CMU_SCFGR.HCLKS 来提升 CPU 时钟频率。
- 步骤 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：

- 步骤 1.** 通过设定系统时钟源切换寄存器 CMU_CKSWR.CKSW 或者系统时钟配置寄存器 CMU_SCFGR.HCLKS 来降低 CPU 时钟频率。
- 步骤 2.** 读取寄存器 CMU_CKSWR 或者 CMU_SCFGR，检查新的设定是否成功。
- 步骤 3.** 向寄存器 EFM_FRMC.FLWT 写入新的读等待周期设定值。
- 步骤 4.** 读取寄存器 EFM_FRMC，检查新的等待周期是否设定成功。

8.3.2.2 D-BUS 读保护功能

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

D-BUS 读保护使能时，用户需要注意以下事项：

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

8.3.3 Flash 读加速缓存

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

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

系统提供 1.125KBytes 空间作为缓存存储器，可以有效地减少因指令跳转而产生的时间损耗。通过 EFM_FRMC.ICACHE/DCACHE 位置 1，来使缓存功能有效。每当出现指令或数据缺失（即请求的指令或数据未存在于当前使用的指令行或缓存存储器中）时，系统会将新读取的数据行（128 位）复制到缓存存储器中。如果 CPU 请求的指令或者数据已存在于缓存中，则无需任何延时即可立即获取。缓存存储器存满后，采用 LRU（最近最少使用）策略确定缓存存储器中待替换的数据。

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

表 8-2 Flash 实际读周期数

EFM_FRMC. FLWT 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

8.3.4 Flash 编程和擦除操作

8.3.4.1 编程与擦除操作概述

Flash 支持编程，扇区擦除，全擦除操作。

Flash 编程，扇区/全擦除地址末位必须以 4 对齐（末位地址为：0x0，0x4，0x8，0xC）。编程单位是 4Bytes。扇区擦除单位为 8KBytes。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，需要关闭缓存功能和预取指功能。以下分别介绍编程和擦除操作的设定步骤。

8.3.4.2 解锁 EFM_KEY1 寄存器

复位解除后，Flash 擦写模式寄存器（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：

步骤 1. 解除 Flash 寄存器访问写保护（EFM_FAPRT 先写 0x0123，再写 0x3210）

步骤 2. 解除 EFM_KEY1 锁定（EFM_KEY1 先写 0x01234567，再写 0xFEDCBA98）

如对 EFM_KEY1 写入错误的序列，则会发生引发一次总线错误，并且自锁，EFM_FWMC 寄存器将写禁止直至系统发生复位。

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。

8.3.4.3 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）。

8.3.4.4 单编程无回读功能

单编程无回读模式设定步骤如下：

步骤 1. 解除 Flash 的寄存器写保护。（EFM_FAPRT 先写 0x0123，再写 0x3210）

步骤 2. 解除 EFM_KEY1 锁定。（EFM_KEY1 先写 0x01234567，再写 0xFEDCBA98）

步骤 3. 设定单次编程模式。（EFM_FWMC.PEMOD=0b001）

步骤 4. 解除写保护。（EFM_FONWPRT 对应位设定为 1）

步骤 5. 对编程地址写入 32 位数据。

步骤 6. 等待 Flash 操作结束并处于空闲状态。（EFM_FSR.OPTEND=1，EFM_FSR.RDY=1）

步骤 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。

步骤 8. 清除操作结束标志位。（EFM_FSR.OPTEND）

对已锁存的 OTP 地址发行单编程不回读写操作，编程不成功，标志位 EFM_FSR.OTPWERR 置位。

8.3.4.5 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.MISMATCH 验证写入数据正确。

单编程回读模式设定步骤如下：

步骤 1. 解除 Flash 的寄存器写保护。（EFM_FAPRT 先写 0x0123，再写 0x3210）

- 步骤 2. 解除 EFM_KEY1 锁定。(EFM_KEY1 先写 0x01234567, 再写 0xFEDCBA98)
 - 步骤 3. 设定单次编程回读模式。(EFM_FWMC.PEMOD=0b010)
 - 步骤 4. 解除写保护。(EFM_FONWPRT 对应位设定为 1)
 - 步骤 5. 对编程地址写入 32 位数据。
 - 步骤 6. 等待 Flash 操作结束并处于空闲状态。(EFM_FSR.OPTEND=1, EFM_FSR.RDY=1)
 - 步骤 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。
 - 步骤 8. 清除操作结束标志位 (EFM_FSR.OPTEND)。
- 对已锁存的 OTP 地址发行单编程回读写操作, 编程不成功, 标志位 EFM_FSR.OTPWERR 置位。

8.3.4.6 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:

- 步骤 1. 解除 Flash 的寄存器写保护。(EFM_FAPRT 先写 0x0123, 再写 0x3210)
- 步骤 2. 解除 EFM_KEY1 锁定。(EFM_KEY1 先写 0x01234567, 再写 0xFEDCBA98)
- 步骤 3. 设定连续编程模式。(EFM_FWMC.PEMOD=0b011)
- 步骤 4. 解除写保护。(EFM_FONWPRT 对应位设定为 1)
- 步骤 5. 对编程地址写入 32 位数据。
执行程序在 Flash 以外区域运行。
- 步骤 6. 等待操作结束标志位 (EFM_FSR.OPTEND) 置位。
- 步骤 7. 清除操作结束标志位, 直至读到 EFM_FSR.OPTEND 为 0。
- 步骤 8. 重复步骤 5~7, 直至所有数据写完。
- 步骤 9. 修改连续编程模式为只读模式来退出连续编程。(EFM_FWMC.PEMOD=0b000)
- 步骤 10. 等待 Flash 处于空闲状态。(EFM_FSR.RDY=1)
- 步骤 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 地址发行连续编程写操作, 编程不成功, 标志位 EFM_FSR.OTPWERR 置位。

8.3.4.7 扇区擦除功能

对 Flash 进行扇区擦除操作后, 该扇区内地址 (8KBytes 空间) 数据刷新为全 1。

扇区擦除操作设定步骤如下:

- 步骤 1. 解除 Flash 的寄存器写保护。(EFM_FAPRT 先写 0x0123, 再写 0x3210)
- 步骤 2. 解除 EFM_KEY1 锁定。(EFM_KEY1 先写 0x01234567, 再写 0xFEDCBA98)
- 步骤 3. 设定扇区擦除模式 (EFM_FWMC.PEMOD=0b100)。
- 步骤 4. 解除写保护。(EFM_FONWPRT 对应位设定为 1)
- 步骤 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。
- 步骤 6. 等待 Flash 操作结束并处于空闲状态。(EFM_FSR.OPTEND=1, EFM_FSR.RDY=1)
- 步骤 7. 清除操作结束标志位。(EFM_FSR.OPTEND)

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

8.3.4.8 全擦除功能

EFM 提供了单块 Flash 全擦除和两块 Flash 同时全擦除两种擦除方式。

全擦除操作设定步骤如下:

- 步骤 1. 解除 Flash 的寄存器写保护。(EFM_FAPRT 先写 0x0123, 再写 0x3210)
- 步骤 2. 解除 EFM_KEY1 锁定。(EFM_KEY1 先写 0x01234567, 再写 0xFEDCBA98)

步骤 3. 单块 Flash 全擦除时, 设定 EFM_FWMC.PEMOD=0b101。

步骤 4. 解除写保护。(EFM_FONWPRT 所有位都设定为 1)

步骤 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。

步骤 6. 等待 Flash 操作结束并处于空闲状态。(EFM_FSR.OPTEND=1, EFM_FSR.RDY=1)

步骤 7. 清除操作结束标志位。(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。

8.3.4.9 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 0x00000430-0x00000433 编程写入数据 0xAF180402, 同时地址 0x03002040-0x0300204B 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 0x03002040-0x0300204B 的数据无法读出。

激活保护级别 1 后, 可通过密码认证方式或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 0x00000434-0x00000437 编程写入数据 0xA85173AE, 保护级别 2 使能。

保护级别 2 使能后:

- 调试接口, ISP 接口和测试接口无法访问 MCU 资源。
- 用户程序无法对扇区 0 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 0x00000420-0x00000423, 0x00000424-0x00000427, 0x00000428-0x0000042B 编程写入数据都为 0x42545048 时, 保护级别 3 使能。

保护级别 3 使能后:

- 调试接口, ISP 接口和测试接口无法改写 Flash 数据。
- 用户程序无法对扇区 0 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。

8.3.4.10 总线保持功能

通过设定寄存器 EFM_FWMC.BUSHLCTL 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 BUSHLCTL 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。

8.3.5 一次性可编程字节 (OTP)

本 MCU 提供最大 9KBytes 的 OTP 领域，地址分布如下。

分为 1 个 8KBytes，16 个 64Bytes。

表 8-3 OTP 地址分布表

Sector	OTP 块数据地址	容量	OTP 块锁存地址
0	0x00000000~0x00001FFF	8KB	0x03000A80~0x03000A83
1	0x03000C00~0x03000C3F	64B	0x03000AC0~0x03000AC3
2	0x03000C40~0x03000C7F	64B	0x03000AC4~0x03000AC7
3	0x03000C80~0x03000CBF	64B	0x03000AC8~0x03000ACB
4	0x03000CC0~0x03000CFF	64B	0x03000ACC~0x03000ACF
5	0x03000D00~0x03000D3F	64B	0x03000AD0~0x03000AD3
6	0x03000D40~0x03000D7F	64B	0x03000AD4~0x03000AD7
7	0x03000D80~0x03000DBF	64B	0x03000AD8~0x03000ADB
8	0x03000DC0~0x03000DFF	64B	0x03000ADC~0x03000ADF
9	0x03000E00~0x03000E3F	64B	0x03000AE0~0x03000AE3
10	0x03000E40~0x03000E7F	64B	0x03000AE4~0x03000AE7
11	0x03000E80~0x03000EBF	64B	0x03000AE8~0x03000AEB
12	0x03000EC0~0x03000EFF	64B	0x03000AEC~0x03000AEF
13	0x03000F00~0x03000F3F	64B	0x03000AF0~0x03000AF3
14	0x03000F40~0x03000F7F	64B	0x03000AF4~0x03000AF7
15	0x03000F80~0x03000FBF	64B	0x03000AF8~0x03000AFB
16	0x03000FC0~0x03000FFF	64B	0x03000AFC~0x03000AFF

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 0x03000A00~0x03000A03 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 0x00000000~0x00001FFF（扇区 0）共享 8KBytes 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 0x10325476，再写 0xEFCDAB89）。如对 EFM_KEY2 寄存器写入错误的序列，则会发生引发一次总线错误，并且 EFM_KEY2 寄存器会锁定直至发生复位。

对 OTP 区域操作请按以下步骤进行：

步骤 1. 解除 Flash 的寄存器写保护。（EFM_FAPRT 先写 0x0123，再写 0x3210）。

- 步骤 2. 解除 EFM_KEY1 锁定。（EFM_KEY1 先写 0x01234567，再写 0xFEDCBA98）。
- 步骤 3. 设定编程模式。（EFM_FWMC.PEMOD=0b001/0b010/0b011）。
- 步骤 4. 根据需要对 OTP 地址写入数据。
- 步骤 5. 解除 EFM_KEY2 锁定。（EFM_KEY2 先写 0x10325476，再写 0xEFCDAB89）。
- 步骤 6. 对 OTP 使能地址 0x03000A00~0x03000A03 的 32-bit 任意 Bit 写 0。（推荐写入 32-bit 全 0）
- 步骤 7. 对 OTP 锁存地址任意 Bit 写入 0。（推荐写入 32-bit 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，若 OTP 使能地址（0x03000A00~0x03000A03），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、扇区擦除和全擦除，将会产生 OTPWERR0 错误标志。
- OTP 锁存地址写完后，如需回归到 EFM_KEY2 锁定状态，可对 EFM_FWMC.KEY2LOCK 写入 1。

8.3.6 引导交换

用户如要升级引导程序，需要对扇区 0（0x00000000~0x00001FFF）进行擦写，如擦写时遇到不可期的意外（如复位、掉电）时，有可能会致整个芯片不能正常启动。在对扇区 0 进行擦除前，预先把新的引导程序写入扇区 1（0x00002000~0x00003FFF），然后对 EFM 地址 0x03002000 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。启动交换操作如下图所示。

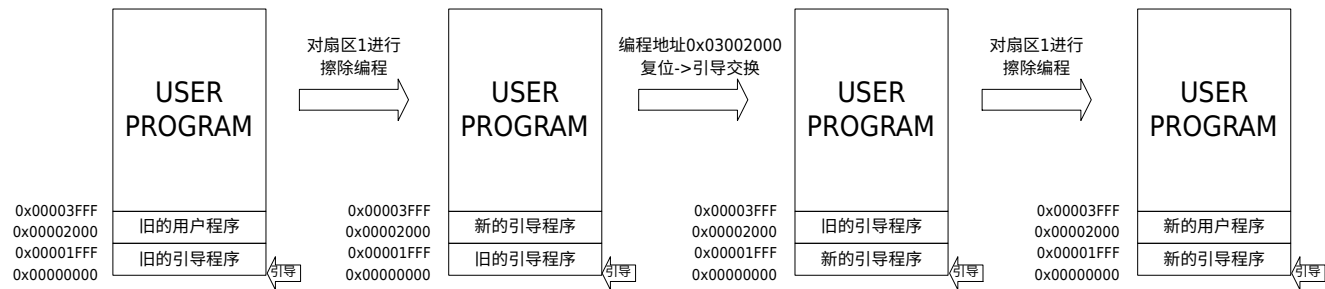


图 8-2 启动扇区交换功能 1

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 0x03002000 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 0x03002000 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，预先把新的引导程序写入扇区 1，然后对 0x03002000 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。

**注意**

1. 需要在擦除后对数据安全配置区 0x03002040~0x0300204B 重新编程。
2. 引导交换使能地址 0x03002000~0x03002003 和数据安全配置区 0x03002040~0x0300204B 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 0x03002040~0x0300204B 编程。

操作流程如下所示：

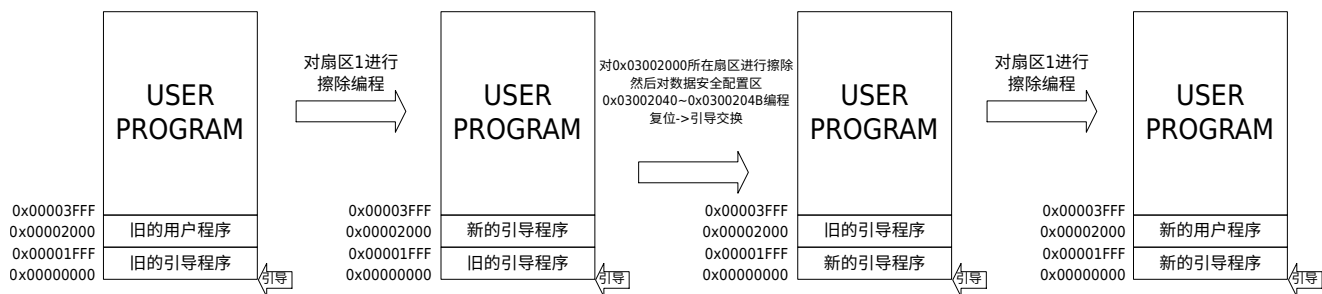


图 8-3 启动交换功能 2

若扇区 0 的 OTP 没有锁存（0x03000A80~0x03000A83 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（0x03000A80~0x03000A83）无法进行编程。

若扇区 0 的 OTP 已锁存（0x03000A80~0x03000A83 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。

8.3.7 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（EFM_FSR.OTPWERR=1）。
- ▶ OTP 功能启用后，对 0x03000A00~0x03000FFF 空间进行擦除，全擦除操作（EFM_FSR.OTPWERR=1）。
- ▶ 对写保护扇区发出编程，擦除，全擦除操作（EFM_FSR.PRTWERR=1）。
- ▶ 编程地址非以 4 对齐或者数据大小非 32 位（EFM_FSR.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（EFM_FSR.MISMATCH=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（EFM_FSR.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（EFM_FSR.OPTEND=1）。
- ▶ 擦除模式：扇区擦除，全擦除结束（EFM_FSR.OPTEND=1）。

● 清零:

寄存器 EFM_FSCLR 对应清除位置 1，状态位清零。

8.4 寄存器

8.4.1 寄存器总表

基地址：0x40010400

表 8-4 嵌入式 Flash (EFM) 寄存器偏移地址

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

8.4.2 Flash 访问写保护寄存器 (EFM_FAPRT)

Offset	Bit Position																															
0x0000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																FAPRT															
Access																	RW															

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	FAPRT	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 0x00000000。 EFM 寄存器访问保护无效时，该寄存器读出值为 0x00000001。	RW

8.4.3 Flash 密钥 1 寄存器 (EFM_KEY1)

Offset	Bit Position																															
0x0004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	KEY1																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	KEY1	密钥 1 寄存器	EFM_FWMC 写保护寄存器。 对 EFM_KEY1 先写 0x01234567，再写 0xFEDCBA98，解除对 EFM_FWMC 的写保护。 该寄存器读出值为 0x00000000。	RW

8.4.4 Flash 密钥 2 寄存器 (EFM_KEY2)

Offset	Bit Position																															
0x0008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	KEY2																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	KEY2	密钥 2 寄存器	对 OTP 锁存地址的写保护寄存器。 对 EFM_KEY2 先写 0x10325476，再写 0xEFCDAB89，解除对 OTP 锁存地址的写保护。 该寄存器读出值为 0x00000000。	RW

8.4.5 Flash 停止寄存器 (EFM_FSTP)

Offset	Bit Position																															
0x0014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															FSTP
Access																																RW

位/位域	标记	位名	功能描述	读写
31:1	Reserved	保留	-	-
0	FSTP	Flash 停止模式控制	0b0：Flash 活动状态 0b1：Flash 处于停止模式  说明 当该位由 1 设为 0 后，请在确认 FSR.RDY 位为 1 后，进行 Flash 访问。	RW

8.4.6 Flash 读模式寄存器 (EFM_FRMC)

Offset	Bit Position																															
0x0018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved												CRST	PREFE	DCACHE	ICACHE	Reserved							LVM	Reserved				FLWT			
Access													RW	RW	RW	RW								RW					RW			

位/位域	标记	位名	功能描述	读写
31:20	Reserved	保留	-	-
19	CRST	缓存复位	0b0: 缓存数据解除复位 0b1: 缓存数据复位 (缓冲、缓存、预取指)	RW
18	PREFE	ICODE 预取指许可位	0b0: 关闭 ICODE 预取指功能 0b1: ICODE 预取指功能使能	RW
17	DCACHE	DCODE 缓存许可位	0b0: 关闭 DCODE 缓存功能 0b1: DCODE 缓存功能使能	RW
16	ICACHE	ICODE 缓存许可位	0b0: 关闭 ICODE 缓存功能 0b1: ICODE 缓存功能使能	RW
15:9	Reserved	保留	-	-
8	LVM	低电压读模式	0b0: 正常电压读模式 0b1: 低电压读模式  说明 超低速运行模式时需要置 LVM 为“1”。	RW
7:4	Reserved	保留	-	-
3:0	FLWT	Flash 读插入的等待周期	0b0000: 不插入读等待周期 0b0001: 插入 1 个读等待周期 0b0010: 插入 2 个读等待周期 0b1110: 插入 14 个读等待周期 0b1111: 插入 15 个读等待周期	RW

8.4.7 Flash 擦写模式寄存器 (EFM_FWMC)

Offset	Bit Position																															
0x001C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00030000																															
Name	Reserved														KEY2LOCK	KEY1LOCK	Reserved						BUSHLCTL	Reserved						PEMOD		
Access															RW	RW							RW							RW		

位/位域	标记	位名	功能描述	读写
31:18	Reserved	保留	-	-
17	KEY2LOCK	Flash 密钥 2 保护状态位	写 1 锁定 EFM_KEY2 寄存器，写 0 无效 读取： 0b0：EFM_KEY2 序列解锁 0b1：EFM_KEY2 序列未解锁  说明 当 EFM_KEY2 写入错误序列后，该位保持为 1 直至复位。	RW
16	KEY1LOCK	Flash 密钥 1 保护状态位	写 1 锁定 EFM_KEY1 寄存器，写 0 无效 读取： 0b0：EFM_KEY1 序列解锁 0b1：EFM_KEY1 序列未解锁  说明 当 EFM_KEY1 写入错误序列后，该位保持为 1 直至复位。	RW
15:9	Reserved	保留	-	-
8	BUSHLCTL	Flash 编程/擦除期间总线控制	0b0：Flash 编程擦除期间，总线被占用 0b1：Flash 编程擦除期间，总线释放	RW
7:3	Reserved	保留	-	-
2:0	PEMOD	Flash 编程/擦除模式	0b000：只读模式 0b001：单编程无回读模式 0b010：单编程回读模式 0b011：连续编程模式 0b100：扇区擦除模式 0b101：单块 Flash 全擦除模式 0b110：禁止设置 0b111：只读模式	RW

8.4.8 Flash 状态寄存器 (EFM_FSR)

Offset	Bit Position																															
0x0020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000100																															
Name	Reserved																								RDY	Reserved	COLERR	OPTEND	MISMTCH	PGSZERR	PRTWERR	OTPWERR
Access																									RO		RO	RO	RO	RO	RO	RO

位/位域	标记	位名	功能描述	读写
31:9	Reserved	保留	-	-
8	RDY	Flash 忙/空闲状态	0b0: Flash 忙状态 0b1: Flash 空闲状态	RO
7:6	Reserved	保留	-	-
5	COLERR	Flash 读写冲突错误标志位	0b0: 未产生标志 0b1: 产生标志 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	RO
4	OPTEND	Flash 空间编程/擦除/全擦除操作结束标志位	0b0: 未产生标志 0b1: 产生标志 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	RO
3	MISMTCH	Flash 空间单编程回读值不一致标志位	0b0: 未产生标志 0b1: 产生标志 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	RO
2	PGSZERR	Flash 空间编程地址和大小不对齐标志位	0b0: 未产生标志 0b1: 产生标志 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	RO

位/位域	标记	位名	功能描述	读写
1	PRTWERR	Flash 写保护错误标志位	● 0b0: 未产生标志 ● 0b1: 产生标志 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	RO
0	OTPWERR	Flash 的 OTP 擦写错误标志位	● 0b0: 未产生标志 ● 0b1: 产生标志 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	RO

8.4.9 Flash 状态清除寄存器 (EFM_FSCLR)

Offset	Bit Position																																	
0x0024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved																												COLERRCLR	OPTENDCLR	MISMTCHCLR	PGSZERRCLR	PRTWERRCLR	OTPWERRCLR
Access																													RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:6	Reserved	保留	-	-
5	COLERRCLR	Flash 读写冲突 错误标志清除	写 1 清除 FSR.COLERR 位，写 0 无效；读出恒为 0	RW
4	OPTENDCLR	Flash 空间编程/ 擦除/全擦除操作 结束标志清除	写 1 清除 FSR.OPTEND 位，写 0 无效；读出恒为 0	RW
3	MISMTCHCLR	Flash 空间单编 程回读值不一致 标志清除	写 1 清除 FSR.MISMTCH 位，写 0 无效；读出恒为 0	RW
2	PGSZERRCLR	Flash 空间编程 地址和大小不对 齐标志清除	写 1 清除 FSR.PGSZERR 位，写 0 无效；读出恒为 0	RW
1	PRTWERRCLR	Flash 写保护错 误标志清除	写 1 清除 FSR.PRTWERR 位，写 0 无效；读出恒为 0	RW
0	OTPWERRCLR	Flash 的 OTP 擦 写错误标志清除	写 1 清除 FSR.OTPWERR 位，写 0 无效；读出恒为 0	RW

8.4.10 Flash 中断许可寄存器 (EFM_FITE)

Offset	Bit Position																																
0x0028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																														COLERRITE	OPTENDITE	PEERRITE
Access																															RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:3	Reserved	保留	-	-
2	COLERRITE	读写冲突错误中断许可	● 0b0：中断不许可 ● 0b1：中断许可	RW
1	OPTENDITE	操作结束中断许可	● 0b0：中断不许可 ● 0b1：中断许可	RW
0	PEERRITE	编程/擦除错误中断许可	● 0b0：中断不许可 ● 0b1：中断许可	RW

8.4.11 Flash 引导交换状态寄存器 (EFM_FSWP)

Offset	Bit Position																															
0x002C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0XXXXXXXXX																															
Name	Reserved																															FSWP
Access																																

位/位域	标记	位名	功能描述	读写
31:1	Reserved	保留	-	-
0	FSWP	扇区 0 和扇区 1 地址交换位	● 0b0: 复位后 CPU 从扇区 0 启动 ● 0b1: 复位后 CPU 从扇区 1 启动 说明 寄存器初始值由 Flash 地址 0x03002000~0x03002003 的值决定，其数据为 0x005A5A5A 时，并且扇区 0 的 OTP 没有锁存（0x03000A80~0x03000A83 数据全 1），初值为 1，其余值时为 0。	RO

8.4.12 芯片专属标志寄存器 (EFM_CHIPID)

Offset	Bit Position																															
0x0040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x58480448																															
Name	CHIPID																															
Access	RO																															

位/位域	标记	位名	功能描述	读写
31:0	CHIPID	芯片专属标志	CHIPID[31:16]: XH 的 ASCII 码 0x5848 CHIPID[15:0]: 产品型号, 例如 0x0448	RO

8.4.13 UNIQUE ID 寄存器 (EFM_UQID0)

Offset	Bit Position																															
0x0050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	LOT_ID[7:0]								WAFER_ID								X_LOCATION								Y_LOCATION							
Access	RO								RO								RO								RO							

位/位域	标记	位名	功能描述	读写
31:24	LOT_ID[7:0]	LOT ID	芯片 LOT ID[7:0]	RO
23:16	WAFER_ID	Wafer ID	Wafer ID	RO
15:8	X_LOCATION	X 坐标	芯片在 Wafer 上的 X 坐标	RO
7:0	Y_LOCATION	Y 坐标	芯片在 Wafer 上的 Y 坐标	RO

8.4.14 UNIQUE ID 寄存器 (EFM_UQID1)

Offset	Bit Position																															
0x0054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	LOT_ID[39:8]																															
Access	RO																															

位/位域	标记	位名	功能描述	读写
31:0	LOT_ID[39:8]	LOT ID	芯片 LOT ID[39:8]	RO

8.4.15 UNIQUE ID 寄存器 (EFM_UQID2)

Offset	Bit Position																															
0x0058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0XXXXXXXXX																															
Name	Reserved																								RO LOT_ID[47:40]							
Access																																

位/位域	标记	位名	功能描述	读写
31:8	Reserved	保留	-	-
7:0	LOT_ID[47:40]	LOT ID	芯片 LOT ID[47:40]	RO

8.4.16 Flash 写保护锁定寄存器 (EFM_WLOCK)

Offset	Bit Position																															
0x0180	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															WLOCK0
Access																																RW

位/位域	标记	位名	功能描述	读写
31:1	Reserved	保留	-	-
0	WLOCK0	FONWPRT 寄存器锁定位	0b0: FONWPRT 寄存器控制的扇区写禁止未 lockup 0b1: FONWPRT 寄存器控制的扇区写禁止 lockup  说明 该位一旦设定为 1，只能通过复位才能恢复成 0。	RW

8.4.17 Flash 写保护寄存器 (EFM_FONWPRT)

Offset	Bit Position																															
0x0190	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	FONWPRT31	FONWPRT30	FONWPRT29	FONWPRT28	FONWPRT27	FONWPRT26	FONWPRT25	FONWPRT24	FONWPRT23	FONWPRT22	FONWPRT21	FONWPRT20	FONWPRT19	FONWPRT18	FONWPRT17	FONWPRT16	FONWPRT15	FONWPRT14	FONWPRT13	FONWPRT12	FONWPRT11	FONWPRT10	FONWPRT9	FONWPRT8	FONWPRT7	FONWPRT6	FONWPRT5	FONWPRT4	FONWPRT3	FONWPRT2	FONWPRT1	FONWPRT0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:0	FONWPRTx (x=31~0)	Flash 扇区 x 的 写许可位	● 0b0: Flash 扇区 x 的写禁止 ● 0b1: Flash 扇区 x 的写许可 说明 WLOCK[0]=0 时, 该位支持写入 0 或 1。 WLOCK[0]=1 时, 该位只支持写 0, 写 1 无效。	RW

8.5 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（EFM_FWMC.BUSHLDC2=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。

9 内置 SRAM (SRAM)

9.1 简介

本产品带有 64KB 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 (200MHz) 下执行。
Ret SRAM 可在 Power down 模式下提供 4KB 的数据保持空间。
SRAM0 和 Ret SRAM 带有 ECC 校验 (Error Checking and Correcting) , ECC 校验为纠一检二码, 即可以纠正一位错误, 检查两位错误。SRAMH 带有奇偶校验 (Parity check) , 每字节数据带有一位校验位。

9.2 功能说明

9.2.1 SRAM 空间分配

表 9-1 SRAM 空间分配

名称	容量	地址范围	校验方式
SRAM0	32KB	0x20000000~0x20007FFF	ECC
SRAMH	32KB	0x1FFF8000~0x1FFFFFFF	Even-parity check
Ret SRAM	4KB	0x200F0000~0x200F0FFF	ECC

 说明

- 在允许产生 RAM 奇偶校验错误产生 NMI 中断或复位的情况下, 当存取数据时, 必须对所用 RAM (SRAMH) 空间以字为单位进行初始化; 当从 SRAMH 空间执行指令时, 必须对所用 RAM 空间 (SRAMH) +3 字的区域以字为单位进行初始化。
- 在允许产生 RAM ECC 校验错误产生 NMI 中断或复位的情况下, 当存取数据时, 必须对所用 RAM (SRAM0, Ret SRAM) 空间以字为单位进行初始化。

9.2.2 ECC 校验错误注入

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 1-bit 错误和多 bit 错误, 由于 ECC 电路只能检出不超过 2-bit 错误, 因此 3-bit 及以上的错误有可能不会被检出。

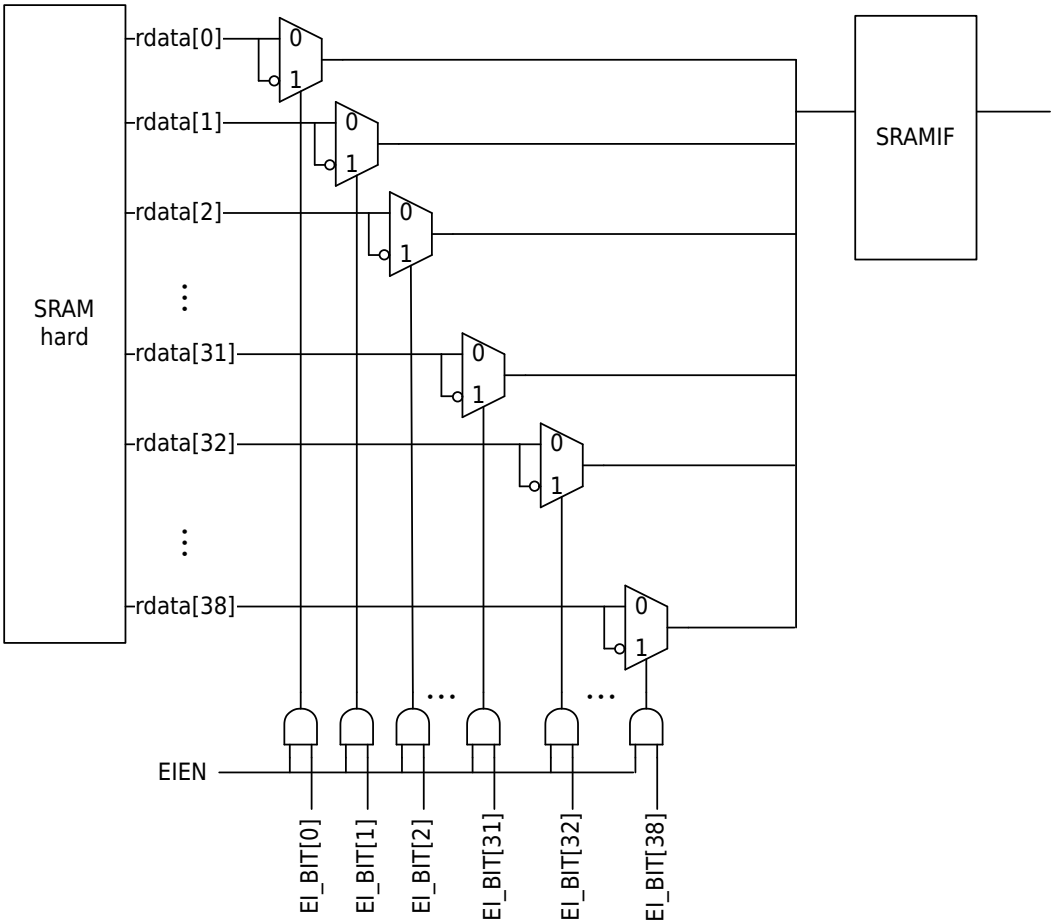


图 9-1 错误注入功能示意图



说明

0x20000000~0x20000002 不支持错误注入和 ECC 校验。

9.3 寄存器

9.3.1 寄存器总表

基地址：0x40050800

表 9-2 内置 SRAM (SRAM) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x08	SRAMC_CKCR	SRAM 校验控制寄存器	32
0x0C	SRAMC_CKPR	SRAM 校验控制保护寄存器	32
0x10	SRAMC_CKSR	SRAM 校验状态寄存器	32
0x14	SRAM0_EIEN	SRAM0 错误注入使能寄存器	32
0x18	SRAM0_EIBIT0	SRAM0 错误注入位使能寄存器 0	32
0x1C	SRAM0_EIBIT1	SRAM0 错误注入位使能寄存器 1	32
0x20	SRAM0_ECCERRADDR	SRAM0 错误地址寄存器	32
0x24	SRAMB_EIEN	Ret SRAM 错误注入使能寄存器	32
0x28	SRAMB_EIBIT0	Ret SRAM 错误注入位使能寄存器 0	32
0x2C	SRAMB_EIBIT1	Ret SRAM 错误注入位使能寄存器 1	32
0x30	SRAMB_ECCERRADDR	Ret SRAM 错误地址寄存器	32

9.3.2 SRAM 校验控制寄存器 (SRAMC_CKCR)



说明

- 在允许产生 RAM 奇偶校验错误产生 NMI 中断或复位的情况下，当存取数据时，必须对所用 RAM 空间 (SRAMH) 以字为单位进行初始化；当从 SRAMH 空间执行指令时，必须对所用 RAM 空间 (SRAMH) +3 字的区域以字为单位进行初始化。
- 在允许产生 RAM ECC 校验错误产生 NMI 中断或复位的情况下，当存取数据时，必须对所用 RAM 空间 (SRAM0, Ret SRAM) 以字为单位进行初始化。

Offset	Bit Position																																					
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	Reserved				BECCMOD				ECCMOD				Reserved				BECCMOD				ECCMOD				Reserved												PYOAD	
Access					RW				RW								RW				RW																RW	

位/位域	标记	位名	功能描述	读写
31:28	Reserved	保留	-	-
27:26	BECCMOD	Ret SRAM 的 ECC 校验允许位	0b00: 禁止 ECC 校验功能 0b01: - 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 0b10: - 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 0b11: - 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
25:24	ECCMOD	SRAM0 的 ECC 校验允许位	0b00: 禁止 ECC 校验功能 0b01: - 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 0b10: - 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 0b11: - 若 1 位错误, ECC 纠错, 产生 1 位错误标志, 产生中断/复位	RW

位/位域	标记	位名	功能描述	读写
25:24	ECCMOD	SRAM0 的 ECC 校验允许位	► 若 2 位错误, ECC 检错, 产生 2 位错误标志, 产生中断/复位	RW
23:18	Reserved	保留	-	-
17	BECCOAD	Ret SRAM ECC 校验出错后操作	● 0b0: Non-maskable interrupt ● 0b1: Reset	RW
16	ECCOAD	SRAM0 ECC 校验出错后操作	● 0b0: Non-maskable interrupt ● 0b1: Reset	RW
15:1	Reserved	保留	-	-
0	PYOAD	奇偶校验出错后操作	● 0b0: Non-maskable interrupt ● 0b1: Reset	RW

9.3.3 SRAM 校验控制保护寄存器 (SRAMC_CKPR)

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								CKPRKW				CKPRC			
Access																									RW				RW			

位/位域	标记	位名	功能描述	读写
31:8	Reserved	保留	-	-
7:1	CKPRKW	写入关键码	当 CKPRC 需要置 1 或者清零时，必须同时向 CKPRKW 写入 0x3B	RW
0	CKPRC	SRAM 校验控制寄存器写使能	● 0b0：SRAM 校验控制寄存器写入禁止 ● 0b1：SRAM 校验控制寄存器写入使能  说明 该位控制 SRAM_CKCR 寄存器的写操作。当 CKPRC 为 1 时，允许对 SRAM_CKCR 进行写操作；当 CKPRC 为 0 时，不允许对 SRAM_CKCR 进行写操作。	RW



说明

向本寄存器写入 0x00000077 可以将 CKPRC 置 1；写入 0x00000076 可以将 CKPRC 清零。

9.3.4 SRAM 校验状态寄存器 (SRAMC_CKSR)

Offset	Bit Position																															
0x10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								CACHE_PYERR	SRAMB_2ERR	SRAMB_1ERR	SRAM0_2ERR	SRAM0_1ERR	SRAMH_PYERR	Reserved	
Access																									RW	RW	RW	RW	RW	RW		

位/位域	标记	位名	功能描述	读写
31:9	Reserved	保留	-	-
8	CACHE_PYERR	CACHERAM 奇偶校验错误标志	写 1 清除标志, 写 0 无效 读取: ● 0b0: 未产生奇偶校验错误 ● 0b1: 产生奇偶校验错误	RW
7	SRAMB_2ERR	Ret SRAM 发生 ECC 2-bit 错误标志	写 1 清除标志, 写 0 无效 读取: ● 0b0: 未产生 2-bit ECC 错误 ● 0b1: 产生 2-bit ECC 错误	RW
6	SRAMB_1ERR	Ret SRAM 发生 ECC 1-bit 错误标志	写 1 清除标志, 写 0 无效 读取: ● 0b0: 未产生 1-bit ECC 错误 ● 0b1: 产生 1-bit ECC 错误	RW
5	SRAM0_2ERR	SRAM0 发生 ECC 2-bit 错误标志	写 1 清除标志, 写 0 无效 读取: ● 0b0: 未产生 2-bit ECC 错误 ● 0b1: 产生 2-bit ECC 错误	RW
4	SRAM0_1ERR	SRAM0 发生 ECC 1-bit 错误标志	写 1 清除标志, 写 0 无效 读取: ● 0b0: 未产生 1-bit ECC 错误 ● 0b1: 产生 1-bit ECC 错误	RW
3	SRAMH_PYERR	SRAMH 奇偶校验错误标志	写 1 清除标志, 写 0 无效 读取: ● 0b0: 未产生奇偶校验错误 ● 0b1: 产生奇偶校验错误	RW
2:0	Reserved	保留	-	-

9.3.5 SRAM0 错误注入使能寄存器 (SRAM0_EIEN)

Offset	Bit Position																															
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															EIEN
Access																																RW

位/位域	标记	位名	功能描述	读写
31:1	Reserved	保留	-	-
0	EIEN	错误注入使能	0b0: 错误注入禁止 0b1: 错误注入使能	RW

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

Offset	Bit Position																															
0x18	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	EIBIT[31:0]																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	EIBIT[31:0]	错误注入位使能	控制 SRAM0 输出数据的 31~0 位是否取反： 0b0: 不取反，该位不注入 ECC 错误 0b1: 取反，该位注入 ECC 错误 说明 每个 Bit 对应一个输出数据位。	RW

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

Offset	Bit Position																															
0x1C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								EIBIT[38:32]							
Access																									RW							

位/位域	标记	位名	功能描述	读写
31:7	Reserved	保留	-	-
6:0	EIBIT[38:32]	错误注入位使能	控制 SRAM0 输出数据的 38~32 位是否取反： 0b0：不取反，该位不注入 ECC 错误 0b1：取反，该位注入 ECC 错误  说明 每个 Bit 对应一个输出数据位。	RW



说明

SRAM0 输出数据的 38~32 位为 ECC 位。

9.3.8 SRAM0 错误地址寄存器 (SRAM0_ECCERRADDR)

Offset	Bit Position																															
0x20	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																ECCERRADDR															
Access																	RO															

位/位域	标记	位名	功能描述	读写
31:15	Reserved	保留	-	-
14:0	ECCERRADDR	校验错误地址	记录 SRAM0 产生 1-bit 或者 2-bit ECC 错误时的偏移地址	RO

9.3.9 Ret SRAM 错误注入使能寄存器 (SRAMB_EIEN)

Offset	Bit Position																															
0x24	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															EIEN
Access																																RW

位/位域	标记	位名	功能描述	读写
31:1	Reserved	保留	-	-
0	EIEN	错误注入使能	0b0: 错误注入禁止 0b1: 错误注入使能	RW

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

Offset	Bit Position																															
0x28	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	EIBIT[31:0]																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	EIBIT[31:0]	错误注入位使能	控制 Ret SRAM 输出数据的 31~0 位是否取反： 0b0: 不取反，该位不注入 ECC 错误 0b1: 取反，该位注入 ECC 错误  说明 每个 Bit 对应一个输出数据位。	RW

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

Offset	Bit Position																															
0x2C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								EIBIT[38:32]							
Access																									RW							

位/位域	标记	位名	功能描述	读写
31:7	Reserved	保留	-	-
6:0	EIBIT[38:32]	错误注入位使能	控制 Ret SRAM 输出数据的 38~32 位是否取反： 0b0：不取反，该位不注入 ECC 错误 0b1：取反，该位注入 ECC 错误  说明 每个 Bit 对应一个输出数据位。	RW



说明

Ret SRAM 输出数据的 38~32 位为 ECC 位。

9.3.12 Ret SRAM 错误地址寄存器 (SRAMB_ECCERRADDR)

Offset	Bit Position																															
0x30	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																				ECCERRADDR											
Access																					RO											

位/位域	标记	位名	功能描述	读写
31:12	Reserved	保留	-	-
11:0	ECCERRADDR	校验错误地址	记录 Ret SRAM 产生 1-bit 或者 2-bit ECC 错误时的偏移地址	RO

10 通用 IO (GPIO)

10.1 简介

本章中使用的一些略称：

- Px (x=A~E, H) 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~E, H, 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。

10.2 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 41 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）

10.3 功能说明

10.3.1 功能框图

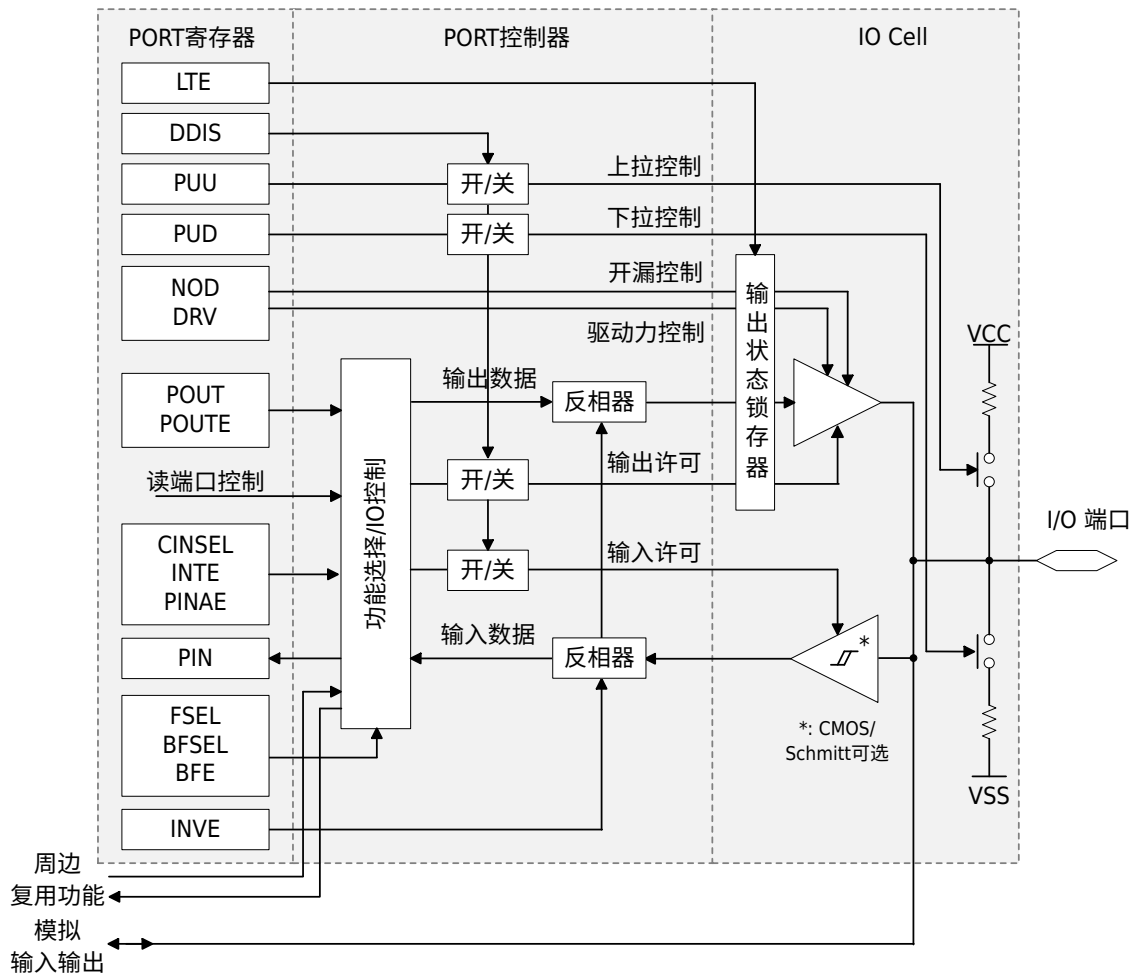


图 10-1 端口基本结构示意图

详细 GPIO 端口数目、5V 耐压、驱动能力配置请参见《数据手册》的“引脚配置及功能”章节。

10.3.2 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.FSEL 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.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，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 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 (PFSRxy.FSEL=0x0), 且处于高阻态 (输出禁止 POUTE=0)。

10.3.3 周边功能

通过功能选择寄存器 PFSRxy.FSEL, 每个端口可以配置最多 41 个功能。其中包括 FSEL=0x0 对应的通用输出 GPO 功能。各端口具体配置的功能请参见《数据手册》的“引脚配置及功能-引脚功能说明”。

JTAG/SWD 调试功能, 通过寄存器 PSPCR 选择。PSPCR.SPFE[z] (z=0~4) 为 1 时, 对应端口的 PFSRxy.FSEL 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。

10.3.4 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 选择好一个功能, 再通过设置公共控制寄存器 PCCR.BFSEL 选择第二个功能, 并设置 PFSRxy.BFE 为 1。例如: 设置 PFSRxy.FSEL=0x2, PCCR.BFSEL=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。

10.3.5 Event Port 输入输出功能

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

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

作为被触发对象时, 设置 AOS 模块 PEVNT_TRGSEL12/PEVNT_TRGSEL34 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 AOS 功能控制位使能。

10.3.6 外部中断 EIRQ 输入功能

除 PB11/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。详细请参见《中断控制器 (INTC)》章节相关描述。

10.3.7 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。

10.3.8 通用控制

以下所述通用控制功能，如无特别说明，它们与端口具体选择的功能即 PFSRxy.FSEL 的设置无关。

上/下拉电阻

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



说明

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，内部处于弱下拉状态。在 I/O 端口处于输出状态时，下拉功能将自动无效。



说明

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。

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。

本功能只在端口处于输出状态时才有效。

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。

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。

10.4 寄存器

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

表 10-1 32-bit 访问时 PORT 寄存器一览

地址	Bit31 ... Bit16	Bit15 ... Bit0
0x40053800+0x10*n	Reserved	PIDRx
0x40053804+0x10*n	POERx	PODRx
0x40053808+0x10*n	PORRx	POSRx
0x4005380C+0x10*n	Reserved	POTRx
0x40053BF4	Reserved	PSPCR
0x40053BF8	Reserved	PCCR
0x40053BFC	Reserved	PWPR
0x40053C00+0x40*n+0x04*y	PFSRxy	PCRxy



说明

除特殊说明，本节所有寄存器偏移地址计算公式中：

- x=A~C 对应 n=0~2, y=0~15。
- x=D 对应 n=3, y=0~2, 8~11。
- x=E 对应 n=4, y=0~4, 12~15。
- x=H 对应 n=5, y=0~2。

10.4.1 寄存器总表

基地址：0x40053800

表 10-2 通用 IO (GPIO) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00+0x10*n	PIDRx	通用输入数据寄存器 (x=A~E, H)	16
0x04+0x10*n	PODRx	通用输出数据寄存器 (x=A~E, H)	16
0x06+0x10*n	POERx	通用输出许可寄存器 (x=A~E, H)	16
0x08+0x10*n	POSRx	通用输出置位寄存器 (x=A~E, H)	16
0x0A+0x10*n	PORRx	通用输出复位寄存器 (x=A~E, H)	16
0x0C+0x10*n	POTRx	通用输出翻转寄存器 (x=A~E, H)	16
0x3F4	PSPCR	特殊控制寄存器	16
0x3F8	PCCR	公共控制寄存器	16
0x3FC	PWPR	写保护寄存器	16

偏移地址	寄存器	描述	位宽
0x400+0x40*n+0x4*y	PCRxy	通用控制寄存器 (x=A~E, H)	16
0x402+0x40*n+0x4*y	PFSRxy	功能选择寄存器 (x=A~E, H)	16

Event Port 基地址：0x40010800

表 10-3 Event Port 寄存器偏移地址

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

 说明

Event Port 寄存器偏移地址中 m=1~4。

10.4.2 通用输入寄存器 (PIDRx)

偏移地址 (Offset) : 0x00+0x10*n

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xXXXX															
Name	PIN15	PIN14	PIN13	PIN12	PIN11	PIN10	PIN9	PIN8	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
Access	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

位/位域	标记	位名	功能描述	读写
15:0	PINy (y=15~0)	输入状态	分别对应 Px15~Px0 端口： 0b0: I/O 端口输入状态为低电平 0b1: I/O 端口输入状态为高电平	RO


说明

端口数字功能未被禁止 (PCRxy.DDIS=0) 时, 通过读取此寄存器可以获取端口的输入状态, 与功能选择寄存器的 PFSRxy.FSEL 设定值无关。

不存在端口对应位的读出值不定。

端口数字功能被禁止 (PCRxy.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。

10.4.3 通用输出数据寄存器 (PODRx)

偏移地址 (Offset) : 0x04+0x10*n

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	POUT15	POUT14	POUT13	POUT12	POUT11	POUT10	POUT9	POUT8	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
15:0	POUTy (y=15~0)	输出数据	分别对应 Px15~Px0 端口： 0b0: 输出低电平 0b1: 输出高电平	RW


说明

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。

10.4.4 通用输出许可寄存器 (POERx)

偏移地址 (Offset) : 0x06+0x10*n

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	POUTE15	POUTE14	POUTE13	POUTE12	POUTE11	POUTE10	POUTE9	POUTE8	POUTE7	POUTE6	POUTE5	POUTE4	POUTE3	POUTE2	POUTE1	POUTE0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
15:0	POUTEy (y=15~0)	输出许可	分别对应 Px15~Px0 端口： ● 0b0：输出禁止 ● 0b1：输出许可	RW

说明

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。

禁止将不存在的端口对应位设置为 1。

10.4.5 通用输出置位寄存器 (POSRx)

偏移地址 (Offset) : 0x08+0x10*n

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	POS15	POS14	POS13	POS12	POS11	POS10	POS9	POS8	POS7	POS6	POS5	POS4	POS3	POS2	POS1	POS0
Access	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

位/位域	标记	位名	功能描述	读写
15:0	POSy (y=15~0)	输出高	分别对应 Px15~Px0 端口： ● 0b0：对应的 PODRx.POUTy 无变化 ● 0b1：对应的 PODRx.POUTy 置 1	WO

说明

此寄存器的读出值始终为 0。32-bit 访问时，同一 I/O 的 PORy 与 POSy 同时写 1 时，PORy 优先级更高，即对应 POUTy 清零。

10.4.6 通用输出复位寄存器 (PORRx)

偏移地址 (Offset) : 0x0A+0x10*n

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	POR15	POR14	POR13	POR12	POR11	POR10	POR9	POR8	POR7	POR6	POR5	POR4	POR3	POR2	POR1	POR0
Access	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

位/位域	标记	位名	功能描述	读写
15:0	PORy (y=15~0)	输出低	分别对应 Px15~Px0 端口： ● 0b0：对应的 PODRx.POUTy 无变化 ● 0b1：对应的 PODRx.POUTy 清零	WO

说明

此寄存器的读出值始终为 0。

10.4.7 通用输出翻转寄存器 (POTRx)

偏移地址 (Offset) : 0x0C+0x10*n

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	POT15	POT14	POT13	POT12	POT11	POT10	POT9	POT8	POT7	POT6	POT5	POT4	POT3	POT2	POT1	POT0
Access	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

位/位域	标记	位名	功能描述	读写
15:0	POTy (y=15~0)	输出翻转	分别对应 Px15~Px0 端口： ● 0b0：对应的 PODRx.POUTy 无变化 ● 0b1：对应的 PODRx.POUTy 取反	WO

说明

此寄存器的读出值始终为 0。

10.4.8 特殊控制寄存器 (PSPCR)

Offset	Bit Position															
0x3F4	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x001F															
Name	Reserved											SPFE[4:0]				
Access												RW				

位/位域	标记	位名	功能描述	读写
15:5	Reserved	保留	-	-
4	SPFE[4]	特殊功能选择	● 0b0: NJTRST 功能无效 ● 0b1: NJTRST 功能有效	RW
3	SPFE[3]	特殊功能选择	● 0b0: JTDI 功能无效 ● 0b1: JTDI 功能有效	RW
2	SPFE[2]	特殊功能选择	● 0b0: JTDO_TRACESWO 功能无效 ● 0b1: JTDO_TRACESWO 功能有效	RW
1	SPFE[1]	特殊功能选择	● 0b0: JTMS_SWDIO 功能无效 ● 0b1: JTMS_SWDIO 功能有效	RW
0	SPFE[0]	特殊功能选择	● 0b0: JTCK_SWCLK 功能无效 ● 0b1: JTCK_SWCLK 功能有效	RW

 **说明**

PSPCR.SPFE 功能选择位优先级高于 PFSRxy.FSEL 功能选择位。

10.4.9 公共控制寄存器 (PCCR)

Offset	Bit Position															
0x3F8	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1000															
Name	Reserved	RDWT			Reserved						BFSEL					
Access		RW									RW					

位/位域	标记	位名	功能描述	读写															
15	Reserved	保留	-	-															
14:12	RDWT	读端口等待	设置读寄存器 PIDRx, PCRxy 时插入的等待周期数	RW															
			<table><tr><th>工作频率</th><th>最小等待周期</th><th>设定范围</th></tr><tr><td>0<HCLK≤50MHz</td><td>0 周期</td><td>0~7</td></tr><tr><td>50MHz<HCLK≤100MHz</td><td>1 周期</td><td>1~7</td></tr><tr><td>100MHz<HCLK≤150MHz</td><td>2 周期</td><td>2~7</td></tr><tr><td>150MHz<HCLK≤200MHz</td><td>3 周期</td><td>3~7</td></tr></table>		工作频率	最小等待周期	设定范围	0<HCLK≤50MHz	0 周期	0~7	50MHz<HCLK≤100MHz	1 周期	1~7	100MHz<HCLK≤150MHz	2 周期	2~7	150MHz<HCLK≤200MHz	3 周期	3~7
			工作频率		最小等待周期	设定范围													
			0<HCLK≤50MHz		0 周期	0~7													
			50MHz<HCLK≤100MHz		1 周期	1~7													
			100MHz<HCLK≤150MHz		2 周期	2~7													
150MHz<HCLK≤200MHz	3 周期	3~7																	
11:6	Reserved	保留	-	-															
5:0	BFSEL	副功能选择	各端口的功能配置请参见《数据手册》的“引脚配置及功能-引脚功能说明”。	RW															

10.4.10 写保护寄存器 (PWPR)

Offset	Bit Position															
0x3FC	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	WP								Reserved							WE
Access	WO															RW

位/位域	标记	位名	功能描述	读写
15:8	WP	写保护码	当 WE 需要置 1 时，必须同时向 WP 写入 0xA5；读出恒为 0	WO
7:1	Reserved	保留	-	-
0	WE	写许可	0b0：PSPCR、PCCR、PCRxy、PFSRxy 寄存器写禁止 0b1：PSPCR、PCCR、PCRxy、PFSRxy 寄存器写许可	RW

 **说明**

向本寄存器写入 0xA501 可以将 WE 置 1；写入其他值会将 WE 清零。

10.4.11 通用控制寄存器 (PCRxy)

偏移地址 (Offset) : $0x400+0x40*n+0x4*y$

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0X00 ⁽¹⁾															
Name	DDIS	LTE	PINAE	INTE	Reserved	CINSEL	INVE	PIN	PUD	PUU	DRV	Reserved	NOD	POUTE	POUT	
Access	RW	RW	RW	RW		RW	RW	RO	RW	RW	RW		RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
15	DDIS	数字功能禁止	0b0: 数字功能有效 0b1: 数字功能禁止	RW
14	LTE	输出状态锁存	0b0: 输出锁存无效 0b1: 输出锁存有效	RW
13	PINAE	输入常开	0b0: 输入 MOS 常开无效 0b1: 输入 MOS 常开有效	RW
12	INTE	外部中断许可	0b0: 外部中断输入禁止 0b1: 外部中断输入许可	RW
11	Reserved	保留	-	-
10	CINSEL	输入模式选择	0b0: Schmitt 输入 0b1: CMOS 输入	RW
9	INVE	反相许可	0b0: 输入输出数据不反相 0b1: 输入输出数据反相	RW
8	PIN	输入状态	0b0: I/O 端口输入状态为低电平 0b1: I/O 端口输入状态为高电平 与寄存器 PIDRx.PINy 功能一致	RO
7	PUD	下拉许可	0b0: 内部下拉 (pulldown) 电阻无效 0b1: 内部下拉 (pulldown) 电阻有效	RW
6	PUU	上拉许可	0b0: 内部上拉 (pullup) 电阻无效 0b1: 内部上拉 (pullup) 电阻有效	RW
5:4	DRV	驱动模式选择	0b00: 低驱动力模式 0b01: 中驱动力模式 0b10/0b11: 高驱动力模式	RW
3	Reserved	保留	-	-
2	NOD	NMOS 开漏	0b0: 正常 CMOS 输出模式 0b1: NMOS 开漏输出	RW
1	POUTE	输出许可	0b0: 输出禁止 0b1: 输出许可 与寄存器 POERx.POUTEy 功能一致	RW

位/位域	标记	位名	功能描述	读写
0	POUT	输出数据	0b0: 输出低电平 0b1: 输出高电平 与寄存器 PODRx.POUTy 功能一致	RW

 **说明**

1. 外部低速振荡器 XTAL32 (XTAL32_IN, XTAL32_OUT) 复用端口 PCRC14 和 PCRC15 的复位值为 0x8100。

2. 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。

3. LTE 设为 1 输出锁存有效时，端口当前的输出状态保持，直至 LTE 写成 0。本功能主要在端口功能切换时使用。为避免功能切换时端口输出意想之外的毛刺而导致系统误动作，在功能切换前，先将 LTE 写 1 锁存住端口的输出状态，再改写寄存器选择寄存器切换功能，最后再将 LTE 写 0 解除锁存，端口状态更新为新功能。

4. 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，和其他周边的输入输出功能。

5. 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。

10.4.12 功能选择寄存器 (PFSRxy)

偏移地址 (Offset) : 0x402+0x40*n+0x4*y

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved							BFE	Reserved		FSEL					
Access								RW			RW					

位/位域	标记	位名	功能描述	读写
15:9	Reserved	保留	-	-
8	BFE	副功能许可	控制 PCCR.BFSEL 选择的副功能是否有效 ● 0b0: 副功能禁止 ● 0b1: 副功能有效	RW
7:6	Reserved	保留	-	-
5:0	FSEL	功能选择	各端口的功能配置请参见《数据手册》的“引脚配置及功能-引脚功能说明”。	RW

说明

- I/O 端口可以通过 FSEL 选择配置在该端口上多个功能中的一个。FSEL 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。
- 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 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 端口复位后为外部低速振荡器接口，初始状态为数字功能禁止状态。在选择数字功能时，需要先将寄存器 CMU_XTAL32CR.XTAL32STP 置 1，关闭外部低速振荡器，再将相应寄存器 PCRxy 的 DDIS 位写 0 使能数字功能。

10.4.13 Event Port 方向选择寄存器（PEVNTDIRm）

偏移地址（Offset）：0x100+0x1C*(m-1)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																PDIR[15:0]															
Access																	RW															

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	PDIR[15:0]	方向选择	● 0b0：Event Port 为输入功能 ● 0b1：Event Port 为输出功能	RW

10.4.14 Event Port 输入数据寄存器（PEVNTIDRm）

偏移地址（Offset）：0x104+0x1C*(m-1)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																PIN[15:0]															
Access																	RO															

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	PIN[15:0]	端口输入状态	● 0b0：事件触发时 Event Port 输入状态为低电平 ● 0b1：事件触发时 Event Port 输入状态为高电平	RO

说明

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。

10.4.15 Event Port 输出数据寄存器（PEVNTODRm）

偏移地址（Offset）：0x108+0x1C*(m-1)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																POUT[15:0]															
Access																	RW															

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	POUT[15:0]	端口输出值	0b0：Event Port 输出低电平 0b1：Event Port 输出高电平	RW

说明

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。

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

偏移地址（Offset）：0x10C+0x1C*(m-1)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																POR[15:0]															
Access																	RW															

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	POR[15:0]	输出值复位	0b0：事件触发时对应 PEVNTODRm.POUT 无变化 0b1：事件触发时对应 PEVNTODRm.POUT 复位	RW

说明

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

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

偏移地址 (Offset) : 0x110+0x1C*(m-1)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																POS[15:0]															
Access																	RW															

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	POS[15:0]	输出值置位	0b0: 事件触发时对应 PEVNTODRm.POUT 无变化 0b1: 事件触发时对应 PEVNTODRm.POUT 置位	RW

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

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

偏移地址 (Offset) : 0x114+0x1C*(m-1)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																RIS[15:0]															
Access																	RW															

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	RIS[15:0]	上升沿检测许可	0b0: EVNTPmn 上升沿事件检出无效 0b1: EVNTPmn 上升沿事件检出有效 PEVNTRISRm.RIS[n]对应 EVNTPmn	RW

 **说明**
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。

10.4.19 Event Port 下降沿输入许可寄存器 (PEVNTFALRm)

偏移地址 (Offset) : 0x118+0x1C*(m-1)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																FAL[15:0]															
Access																	RW															

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	FAL[15:0]	下降沿检测许可	● 0b0: EVNTPmn 下降沿事件检出无效 ● 0b1: EVNTPmn 下降沿事件检出有效 PEVNTRISRm.FAL[n]对应 EVNTPmn	RW

 **说明**

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。

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

Offset	Bit Position																															
0x170	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved				DIVS4		NFEN4		Reserved				DIVS3		NFEN3		Reserved				DIVS2		NFEN2		Reserved				DIVS1		NFEN1	
Access					RW		RW						RW		RW						RW		RW						RW		RW	

位/位域	标记	位名	功能描述	读写
31:27	Reserved	保留	-	-
26:25	DIVS4	Event Port4 数字滤波采样时钟选择	0b00: PCLK1 0b01: PCLK1/8 0b10: PCLK1/32 0b11: PCLK1/64	RW
24	NFEN4	Event Port4 数字滤波许可	0b0: 数字滤波无效 0b1: 数字滤波有效	RW
23:19	Reserved	保留	-	-
18:17	DIVS3	Event Port3 数字滤波采样时钟选择	0b00: PCLK1 0b01: PCLK1/8 0b10: PCLK1/32 0b11: PCLK1/64	RW
16	NFEN3	Event Port3 数字滤波许可	0b0: 数字滤波无效 0b1: 数字滤波有效	RW
15:11	Reserved	保留	-	-
10:9	DIVS2	Event Port2 数字滤波采样时钟选择	0b00: PCLK1 0b01: PCLK1/8 0b10: PCLK1/32 0b11: PCLK1/64	RW
8	NFEN2	Event Port2 数字滤波许可	0b0: 数字滤波无效 0b1: 数字滤波有效	RW
7:3	Reserved	保留	-	-
2:1	DIVS1	Event Port1 数字滤波采样时钟选择	0b00: PCLK1 0b01: PCLK1/8 0b10: PCLK1/32 0b11: PCLK1/64	RW
0	NFEN1	Event Port1 数字滤波许可	0b0: 数字滤波无效 0b1: 数字滤波有效	RW

10.5 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=0b1）。
3. 请在输出锁存有效时（PCRxy.LTE=0b1）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。

11 中断控制器 (INTC)

11.1 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（睡眠模式和停止模式）；控制外部中断和软件中断。

INTC 主要特性如下：

1. 可屏蔽中断：INTC 配备了 257 个中断事件，处理后作为中断请求 (IRQ) 发送给 NVIC。NVIC 支持 130 个 IRQ，每个 IRQ 对应一个或多个中断事件。
-  **说明**
更多关于异常和 NVIC 编程的说明，请参考《Arm Cortex-M4 Processor Technical Reference Manual》。
2. NVIC 可编程优先级：16 个可编程优先级（使用了 4 位中断优先级）。
3. 不可屏蔽中断：可以选择多种系统中断事件作为不可屏蔽中断请求，且各中断事件配备了独立的使能、挂起、清除挂起寄存器。
4. 配备 16 个外部管脚中断事件。
5. 配备多种外设中断事件请求，详细信息请参见“中断事件表”。
6. 配备 32 个软件中断事件。
7. 中断可唤醒系统睡眠模式和停止模式。

表 11-1 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

管脚名	I/O	说明
EIRQ15	输入	外部中断事件 15

11.2 向量表

11.2.1 中断向量表

表 11-2 中断向量表

向量地址	向量序号	IRQ 序号	中断源	说明
ARM 内核中断				
0x00000000	0	-	ARM core	Initial Stack Pointer
0x00000004	1	-	ARM core	Initial Program Counter
0x00000008	2	-	ARM core	Non-maskable Interrupt(NMI)
0x0000000C	3	-	ARM core	Hard Fault
0x00000010	4	-	ARM core	Mem Manage Fault
0x00000014	5	-	ARM core	Bus Fault
0x00000018	6	-	ARM core	Usage Fault
0x0000001C	7	-	ARM core	Reserved
0x00000020	8	-	ARM core	Reserved
0x00000024	9	-	ARM core	Reserved
0x00000028	10	-	ARM core	Reserved
0x0000002C	11	-	ARM core	Supervisor Call(SVCall)
0x00000030	12	-	ARM core	Debug Monitor
0x00000034	13	-	ARM core	Reserved
0x00000038	14	-	ARM core	Pendable request for system service(PendSV)
0x0000003C	15	-	ARM core	System tick timer(SysTick)
非 ARM 内核中断				
0x00000040	16	0	INT_INTSEL0	寄存器 INTC_INTSEL0 选择的中断事件/软件中断 0
0x00000044	17	1	INT_INTSEL1	寄存器 INTC_INTSEL1 选择的中断事件/软件中断 1
0x00000048	18	2	INT_INTSEL2	寄存器 INTC_INTSEL2 选择的中断事件/软件中断 2
0x0000004C	19	3	INT_INTSEL3	寄存器 INTC_INTSEL3 选择的中断事件/软件中断 3
0x00000050	20	4	INT_INTSEL4	寄存器 INTC_INTSEL4 选择的中断事件/软件中断 4
0x00000054	21	5	INT_INTSEL5	寄存器 INTC_INTSEL5 选择的中断事件/软件中断 5
0x00000058	22	6	INT_INTSEL6	寄存器 INTC_INTSEL6 选择的中断事件/软件中断 6
0x0000005C	23	7	INT_INTSEL7	寄存器 INTC_INTSEL7 选择的中断事件/软件中断 7
0x00000060	24	8	INT_INTSEL8	寄存器 INTC_INTSEL8 选择的中断事件/软件中断 8
0x00000064	25	9	INT_INTSEL9	寄存器 INTC_INTSEL9 选择的中断事件/软件中断 9
0x00000068	26	10	INT_INTSEL10	寄存器 INTC_INTSEL10 选择的中断事件/软件中断 10
0x0000006C	27	11	INT_INTSEL11	寄存器 INTC_INTSEL11 选择的中断事件/软件中断 11
0x00000070	28	12	INT_INTSEL12	寄存器 INTC_INTSEL12 选择的中断事件/软件中断 12
0x00000074	29	13	INT_INTSEL13	寄存器 INTC_INTSEL13 选择的中断事件/软件中断 13
0x00000078	30	14	INT_INTSEL14	寄存器 INTC_INTSEL14 选择的中断事件/软件中断 14
0x0000007C	31	15	INT_INTSEL15	寄存器 INTC_INTSEL15 选择的中断事件/软件中断 15
0x00000080	32	16	PORT	外部中断 0/软件中断 16
0x00000084	33	17	PORT	外部中断 1/软件中断 17
0x00000088	34	18	PORT	外部中断 2/软件中断 18
0x0000008C	35	19	PORT	外部中断 3/软件中断 19

向量地址	向量序号	IRQ 序号	中断源	说明
0x00000090	36	20	PORT	外部中断 4/软件中断 20
0x00000094	37	21	PORT	外部中断 5/软件中断 21
0x00000098	38	22	PORT	外部中断 6/软件中断 22
0x0000009C	39	23	PORT	外部中断 7/软件中断 23
0x000000A0	40	24	PORT	外部中断 8/软件中断 24
0x000000A4	41	25	PORT	外部中断 9/软件中断 25
0x000000A8	42	26	PORT	外部中断 10/软件中断 26
0x000000AC	43	27	PORT	外部中断 11/软件中断 27
0x000000B0	44	28	PORT	外部中断 12/软件中断 28
0x000000B4	45	29	PORT	外部中断 13/软件中断 29
0x000000B8	46	30	PORT	外部中断 14/软件中断 30
0x000000BC	47	31	PORT	外部中断 15/软件中断 31
0x000000C0	48	32	DMA_1	传输错误中断
0x000000C4	49	33	DMA_1	通道 0 传输完成中断/通道 0 数据块传输完成中断
0x000000C8	50	34	DMA_1	通道 1 传输完成中断/通道 1 数据块传输完成中断
0x000000CC	51	35	DMA_1	通道 2 传输完成中断/通道 2 数据块传输完成中断
0x000000D0	52	36	DMA_1	通道 3 传输完成中断/通道 3 数据块传输完成中断
0x000000D4	53	37	DMA_1	通道 4 传输完成中断/通道 4 数据块传输完成中断
0x000000D8	54	38	DMA_1	通道 5 传输完成中断/通道 5 数据块传输完成中断
0x000000DC	55	39	EFM	编程（擦除）错误中断/读写冲突错误中断
0x000000E0	56	40	EFM	操作结束中断
0x000000E4	57	41	QSPI	FLASH 访问错误中断
0x000000E8	58	42	DCU_1	DCU_1 中断
0x000000EC	59	43	DCU_2	DCU_2 中断
0x000000F0	60	44	DCU_3	DCU_3 中断
0x000000F4	61	45	DCU_4	DCU_4 中断
0x000000F8	62	46	DMA_2	传输错误中断
0x000000FC	63	47	DMA_2	通道 0 传输完成中断/通道 0 数据块传输完成中断
0x00000100	64	48	DMA_2	通道 1 传输完成中断/通道 1 数据块传输完成中断
0x00000104	65	49	DMA_2	通道 2 传输完成中断/通道 2 数据块传输完成中断
0x00000108	66	50	DMA_2	通道 3 传输完成中断/通道 3 数据块传输完成中断
0x0000010C	67	51	DMA_2	通道 4 传输完成中断/通道 4 数据块传输完成中断
0x00000110	68	52	DMA_2	通道 5 传输完成中断/通道 5 数据块传输完成中断
0x00000114	69	53	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
0x00000118	70	54	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
0x0000011C	71	55	RTC	闹钟中断/定周期中断
0x00000120	72	56	CMU	XTAL 振荡故障中断
0x00000124	73	57	PWC	WKTM 比较匹配中断
0x00000128	74	58	SWDT	计数下溢/刷新错误中断
0x0000012C	75	59	Timer6_1	通用计数比较匹配中断/输入捕获中断
0x00000130	76	60	Timer6_1	计数周期匹配中断
0x00000134	77	61	Timer6_1	死区时间错误中断
0x00000138	78	62	Timer6_1	专用计数比较匹配中断
0x0000013C	79	63	TimerA_1	计数周期匹配中断
0x00000140	80	64	TimerA_1	计数比较匹配中断
0x00000144	81	65	Timer6_2	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
0x00000148	82	66	Timer6_2	计数周期匹配中断
0x0000014C	83	67	Timer6_2	死区时间错误中断
0x00000150	84	68	Timer6_2	专用计数比较匹配中断
0x00000154	85	69	TimerA_2	计数周期匹配中断
0x00000158	86	70	TimerA_2	计数比较匹配中断
0x0000015C	87	71	TimerA_3	计数周期匹配中断
0x00000160	88	72	TimerA_3	计数比较匹配中断
0x00000164	89	73	TimerA_4	计数周期匹配中断
0x00000168	90	74	TimerA_4	计数比较匹配中断
0x0000016C	91	75	Timer4_1	通用计数比较匹配中断
0x00000170	92	76	Timer4_1	计数周期匹配中断
0x00000174	93	77	Timer4_1	重载计数匹配中断
0x00000178	94	78	Timer4_1	专用计数比较匹配中断
0x0000017C	95	79	Timer4_2	通用计数比较匹配中断
0x00000180	96	80	Timer4_2	计数周期匹配中断
0x00000184	97	81	Timer4_2	重载计数匹配中断
0x00000188	98	82	Timer4_2	专用计数比较匹配中断
0x0000018C	99	83	Timer4_3	通用计数比较匹配中断
0x00000190	100	84	Timer4_3	计数周期匹配中断
0x00000194	101	85	Timer4_3	重载计数匹配中断
0x00000198	102	86	Timer4_3	专用计数比较匹配中断
0x0000019C	103	87	I2C_1	I2C_1 中断
0x000001A0	104	88	I2C_2	I2C_2 中断
0x000001A4	105	89	CMP	CMP1 中断
0x000001A8	106	90	CMP	CMP2 中断
0x000001AC	107	91	CMP	CMP3 中断
0x000001B0	108	92	CMP	CMP4 中断
0x000001B4	109	93	USART_1	USART_1 中断 (TCI 及 RX 线唤醒停止模式中断除外)
0x000001B8	110	94	USART_1	USART_1_TCI 中断
0x000001BC	111	95	USART_2	USART_2 中断 (TCI 中断除外)
0x000001C0	112	96	USART_2	USART_2_TCI 中断
0x000001C4	113	97	SPI_1	SPI_1 中断
0x000001C8	114	98	TimerA_5	计数周期匹配中断
0x000001CC	115	99	TimerA_5	计数比较匹配中断
0x000001D0	116	100	PORT	Event Port1 中断
0x000001D4	117	101	PORT	Event Port2 中断
0x000001D8	118	102	PORT	Event Port3 中断
0x000001DC	119	103	PORT	Event Port4 中断
0x000001E0	120	104	USART_3	USART_3 中断 (TCI 中断除外)
0x000001E4	121	105	USART_3	USART_3_TCI 中断
0x000001E8	122	106	USART_4	USART_4 中断 (TCI 中断除外)
0x000001EC	123	107	USART_4	USART_4_TCI 中断
0x000001F0	124	108	SPI_2	SPI_2 中断
0x000001F4	125	109	SPI_3	SPI_3 中断
0x000001F8	126	110	EMB	Group0 中断
0x000001FC	127	111	EMB	Group1 中断

向量地址	向量序号	IRQ 序号	中断源	说明
0x00000200	128	112	EMB	Group2 中断
0x00000204	129	113	EMB	Group3 中断
0x00000208	130	114	USART_5	USART_5 中断 (TCI 中断除外)
0x0000020C	131	115	USART_5	USART_5_TCI 中断
0x00000210	132	116	USART_6	USART_6 中断 (TCI 中断除外)
0x00000214	133	117	USART_6	USART_6_TCI 中断
0x00000218	134	118	MCAN_1	MCAN_1 中断 0
0x0000021C	135	119	MCAN_1	MCAN_1 中断 1
0x00000220	136	120	MCAN_2	MCAN_2 中断 0
0x00000224	137	121	MCAN_2	MCAN_2 中断 1
0x00000228	138	122	USART_1	RX 线唤醒停止模式中断
0x00000234	141	125	CMU	FCM 时钟频率测量中断
0x00000238	142	126	WDT	计数下溢/刷新错误中断
0x0000023C	143	127	CTC	HRC 频率校准值溢出中断
0x00000240	144	128	ADC_1	ADC_1 中断
0x00000244	145	129	ADC_2	ADC_2 中断
0x00000248	146	130	ADC_3	ADC_3 中断
0x0000024C	147	131	TRNG	TRNG 中断



说明

INTC_INTSEL 寄存器选择的中断事件编号请参见[中断事件表](#)。

IRQ 序号 123~124、132~239 对应向量未配置任何中断事件，请勿使用。

11.2.2 中断事件表

中断事件由系统或外设产生，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。

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]

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)
16	0x010	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
17	0x011	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
18	0x012	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
19	0x013	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
20	0x014	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
21	0x015	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
22	0x016	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
23	0x017	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
24	0x018	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
25	0x019	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
26	0x01A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
27	0x01B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
28	0x01C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
29	0x01D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
30	0x01E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
31	0x01F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
32	0x020	DMA_1	DMA_1_ERR	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[32]
33	0x021		DMA_1_TC0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[33]
34	0x022		DMA_1_BTC0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
35	0x023		DMA_1_TC1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[34]
36	0x024		DMA_1_BTC1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
37	0x025		DMA_1_TC2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[35]
38	0x026		DMA_1_BTC2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
39	0x027		DMA_1_TC3	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[36]
40	0x028		DMA_1_BTC3	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
41	0x029		DMA_1_TC4	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[37]
42	0x02A		DMA_1_BTC4	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
43	0x02B		DMA_1_TC5	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[38]
44	0x02C		DMA_1_BTC5	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
45	0x02D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
46	0x02E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
47	0x02F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
48	0x030	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
49	0x031	EFM	EFM_PEERR	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[39]
50	0x032		EFM_RDCOL	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
51	0x033		EFM_OPTEND	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[40]
52	0x034	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
53	0x035	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
54	0x036	QSPI	QSPI_INTR	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[41]
55	0x037	DCU	DCU1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[42]
56	0x038		DCU2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[43]
57	0x039		DCU3	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[44]
58	0x03A		DCU4	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[45]
59	0x03B	-	-	-	-	-	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)
60	0x03C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
61	0x03D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
62	0x03E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
63	0x03F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
64	0x040	DMA_2	DMA_2_ERR	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[46]
65	0x041		DMA_2_TC0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[47]
66	0x042		DMA_2 BTC0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
67	0x043		DMA_2_TC1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[48]
68	0x044		DMA_2 BTC1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
69	0x045		DMA_2_TC2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[49]
70	0x046		DMA_2 BTC2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
71	0x047		DMA_2_TC3	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[50]
72	0x048		DMA_2 BTC3	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
73	0x049		DMA_2_TC4	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[51]
74	0x04A		DMA_2 BTC4	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
75	0x04B		DMA_2_TC5	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[52]
76	0x04C		DMA_2 BTC5	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
77	0x04D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
78	0x04E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
79	0x04F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
80	0x050	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
81	0x051	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
82	0x052	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
83	0x053	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
84	0x054	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
85	0x055	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
86	0x056	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
87	0x057	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
88	0x058	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
89	0x059	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
90	0x05A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
91	0x05B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
92	0x05C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
93	0x05D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
94	0x05E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
95	0x05F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
96	0x060	Timer0_1	TMR0_1_CMPA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[53]
97	0x061		TMR0_1_CMPB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
98	0x062		TMR0_1_OVFA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
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[54]
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)	↑

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)
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[55]
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[56]
126	0x07E	WKTm	WKTm_PRD	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[57]
127	0x07F	SWDT	SWDT_REFUDF	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[58]
128	0x080	Timer6_1	TMR6_1_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[59]
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)	↑
134	0x086	Timer6_1	TMR6_1_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[60]
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[61]
137	0x089		TMR6_1_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[62]
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[63]
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[64]
142	0x08E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
143	0x08F	-	-	-	-	-	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)
144	0x090	Timer6_2	TMR6_2_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[65]
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[66]
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[67]
153	0x099		TMR6_2_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[68]
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[69]
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[70]
158	0x09E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
159	0x09F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
160	0x0A0	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
161	0x0A1	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
162	0x0A2	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
163	0x0A3	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
164	0x0A4	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
165	0x0A5	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
166	0x0A6	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
167	0x0A7	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
168	0x0A8	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
169	0x0A9	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
170	0x0AA	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
171	0x0AB	TimerA_3	TMRA_3_OVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[71]
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[72]
174	0x0AE	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
175	0x0AF	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
176	0x0B0	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
177	0x0B1	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
178	0x0B2	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
179	0x0B3	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
180	0x0B4	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
181	0x0B5	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
182	0x0B6	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
183	0x0B7	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
184	0x0B8	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
185	0x0B9	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
186	0x0BA	-	-	-	-	-	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)
187	0x0BB	TimerA_4	TMRA_4_OVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[73]
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[74]
190	0x0BE	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
191	0x0BF	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
192	0x0C0	Timer4_1	TMR4_1_GCMUH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[75]
193	0x0C1		TMR4_1_GCMUL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
194	0x0C2		TMR4_1_GCMVH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
195	0x0C3		TMR4_1_GCMVL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
196	0x0C4		TMR4_1_GCMWH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
197	0x0C5		TMR4_1_GCMWL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
198	0x0C6		TMR4_1_GCMXH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
199	0x0C7		TMR4_1_GCMXL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
200	0x0C8		TMR4_1_GOVF	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[76]
201	0x0C9		TMR4_1_GUDF	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
202	0x0CA		TMR4_1_GRLU	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[77]
203	0x0CB		TMR4_1_GRLV	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
204	0x0CC		TMR4_1_GRLW	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
205	0x0CD		TMR4_1_GRLX	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
206	0x0CE		TMR4_1_SCM0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[78]
207	0x0CF		TMR4_1_SCM1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
208	0x0D0		TMR4_1_SCM2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
209	0x0D1		TMR4_1_SCM3	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
210	0x0D2		TMR4_1_SCM4	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
211	0x0D3		TMR4_1_SCM5	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
212	0x0D4		TMR4_1_SCM6	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
213	0x0D5		TMR4_1_SCM7	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
214	0x0D6	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
215	0x0D7	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
216	0x0D8	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
217	0x0D9	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
218	0x0DA	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
219	0x0DB	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
220	0x0DC	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
221	0x0DD	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
222	0x0DE	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
223	0x0DF	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
224	0x0E0	Timer4_2	TMR4_2_GCMUH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[79]
225	0x0E1		TMR4_2_GCMUL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
226	0x0E2		TMR4_2_GCMVH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
227	0x0E3		TMR4_2_GCMVL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
228	0x0E4		TMR4_2_GCMWH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
229	0x0E5		TMR4_2_GCMWL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
230	0x0E6		TMR4_2_GCMXH	L	✓	✓	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)
231	0x0E7	Timer4_2	TMR4_2_GCMXL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
232	0x0E8		TMR4_2_GOVF	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[80]
233	0x0E9		TMR4_2_GUDF	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
234	0x0EA		TMR4_2_GRLU	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[81]
235	0x0EB		TMR4_2_GRLV	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
236	0x0EC		TMR4_2_GRLW	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
237	0x0ED		TMR4_2_GRLX	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
238	0x0EE		TMR4_2_SCM0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[82]
239	0x0EF		TMR4_2_SCM1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
240	0x0F0		TMR4_2_SCM2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
241	0x0F1		TMR4_2_SCM3	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
242	0x0F2		TMR4_2_SCM4	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
243	0x0F3		TMR4_2_SCM5	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
244	0x0F4		TMR4_2_SCM6	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
245	0x0F5		TMR4_2_SCM7	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
246	0x0F6	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
247	0x0F7	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
248	0x0F8	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
249	0x0F9	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
250	0x0FA	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
251	0x0FB	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
252	0x0FC	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
253	0x0FD	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
254	0x0FE	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
255	0x0FF	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
256	0x100	Timer4_3	TMR4_3_GCMUH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[83]
257	0x101		TMR4_3_GCMUL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
258	0x102		TMR4_3_GCMVH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
259	0x103		TMR4_3_GCMVL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
260	0x104		TMR4_3_GCMWH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
261	0x105		TMR4_3_GCMWL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
262	0x106		TMR4_3_GCMXH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
263	0x107		TMR4_3_GCMXL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
264	0x108		TMR4_3_GOVF	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[84]
265	0x109		TMR4_3_GUDF	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
266	0x10A		TMR4_3_GRLU	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[85]
267	0x10B		TMR4_3_GRLV	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
268	0x10C		TMR4_3_GRLW	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
269	0x10D		TMR4_3_GRLX	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
270	0x10E		TMR4_3_SCM0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[86]
271	0x10F		TMR4_3_SCM1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
272	0x110		TMR4_3_SCM2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
273	0x111		TMR4_3_SCM3	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
274	0x112		TMR4_3_SCM4	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)
275	0x113	Timer4_3	TMR4_3_SCM5	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
276	0x114		TMR4_3_SCM6	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
277	0x115		TMR4_3_SCM7	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
278	0x116	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
279	0x117	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
280	0x118	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
281	0x119	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
282	0x11A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
283	0x11B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
284	0x11C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
285	0x11D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
286	0x11E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
287	0x11F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
288	0x120	I2C_1	I2C_1_RXI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[87]
289	0x121		I2C_1_TXI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
290	0x122		I2C_1_TEI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
291	0x123		I2C_1_EEI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
292	0x124	I2C_2	I2C_2_RXI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[88]
293	0x125		I2C_2_TXI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
294	0x126		I2C_2_TEI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
295	0x127		I2C_2_EEI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
296	0x128	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
297	0x129	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
298	0x12A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
299	0x12B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
300	0x12C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
301	0x12D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
302	0x12E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
303	0x12F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
304	0x130	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
305	0x131	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
306	0x132	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
307	0x133	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
308	0x134	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
309	0x135	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
310	0x136	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
311	0x137	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
312	0x138	CMP	CMP_1_IRQ	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[89]
313	0x139		CMP_2_IRQ	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[90]
314	0x13A		CMP_3_IRQ	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[91]
315	0x13B		CMP_4_IRQ	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[92]
316	0x13C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
317	0x13D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
318	0x13E	-	-	-	-	-	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)
319	0x13F	-	-	-	-	-	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[93]
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[94]
327	0x147	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
328	0x148	USART_2	USART_2_EI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[95]
329	0x149		USART_2_RI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
330	0x14A		USART_2_TI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
331	0x14B		USART_2_RTO	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
332	0x14C		USART_2_TENDI	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
333	0x14D		USART_2_TCI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[96]
334	0x14E	SPI_1	SPI_1_SPRI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[97]
335	0x14F		SPI_1_SPTI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
336	0x150		SPI_1_SPII	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
337	0x151		SPI_1_SPEI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
338	0x152		SPI_1_SPEND	P	-	✓	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[98]
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[99]
343	0x157	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
344	0x158	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
345	0x159	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
346	0x15A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
347	0x15B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
348	0x15C	EVENT port	EVENT_PORT1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[100]
349	0x15D		EVENT_PORT2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[101]
350	0x15E		EVENT_PORT3	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[102]
351	0x15F		EVENT_PORT4	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[103]
352	0x160	USART_3	USART_3_BRKWKPI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[104]
353	0x161		USART_3_EI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
354	0x162		USART_3_RI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
355	0x163		USART_3_TI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
356	0x164		-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
357	0x165		USART_3_TENDI	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[104]
358	0x166		USART_3_TCI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[105]
359	0x167	-	-	-	-	-	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)
360	0x168	USART_4	USART_4_EI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[106]
361	0x169		USART_4_RI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
362	0x16A		USART_4_TI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
363	0x16B		USART_4_RTO	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
364	0x16C		USART_4_TENDI	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
365	0x16D		USART_4_TCI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[107]
366	0x16E	SPI_2	SPI_2_SPRI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[108]
367	0x16F		SPI_2_SPTI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
368	0x170		SPI_2_SPII	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
369	0x171		SPI_2_SPEI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
370	0x172		SPI_2_SPEND	P	-	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	-
371	0x173	SPI_3	SPI_3_SPRI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[109]
372	0x174		SPI_3_SPTI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
373	0x175		SPI_3_SPII	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
374	0x176		SPI_3_SPEI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
375	0x177		SPI_3_SPEND	P	-	✓	-	-
376	0x178	EMB	EMB_GR0	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[110]
377	0x179		EMB_GR1	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[111]
378	0x17A	EMB	EMB_GR2	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[112]
379	0x17B		EMB_GR3	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[113]
380	0x17C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
381	0x17D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
382	0x17E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
383	0x17F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
384	0x180	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
385	0x181	USART_5	USART_5_EI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[114]
386	0x182		USART_5_RI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
387	0x183		USART_5_TI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
388	0x184		USART_5_RTO	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
389	0x185		USART_5_TENDI	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
390	0x186		USART_5_TCI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[115]
391	0x187	USART_6	USART_6_BRKWKPI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[116]
392	0x188		USART_6_EI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
393	0x189		USART_6_RI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
394	0x18A		USART_6_TI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
395	0x18B		USART_6_TENDI	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
396	0x18C		USART_6_TCI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[117]
397	0x18D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
398	0x18E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
399	0x18F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
400	0x190	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
401	0x191	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
402	0x192	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
403	0x193	-	-	-	-	-	-	-

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)
404	0x194	-	-	-	-	-	-	-
405	0x195	-	-	-	-	-	-	-
406	0x196	-	-	-	-	-	-	-
407	0x197	-	-	-	-	-	-	-
408	0x198	MCAN	MCAN_1_INT0	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[118]
409	0x199		MCAN_1_INT1	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[119]
410	0x19A		MCAN_2_INT0	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[120]
411	0x19B		MCAN_2_INT1	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[121]
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)	-
447	0x1BF	-	-	-	-	-	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)
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[122]
465	0x1D1	PVD	PVD_PVD1	P	-	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	-
466	0x1D2		PVD_PVD2	P	-	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	-
467	0x1D3	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
468	0x1D4	FCM	FCMFERRI	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[125]
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[126]
472	0x1D8	CTC	CTC_ERR	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[127]
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[128]
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[129]
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[130]
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)	↑

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)
492	0x1EC	TRNG	TRNG_END	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[131]
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)	-



说明

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 INTSFTTRG.STRG 位产生。
4. “↑”表示 IRQ 序号同上。

11.3 功能说明

11.3.1 功能框图

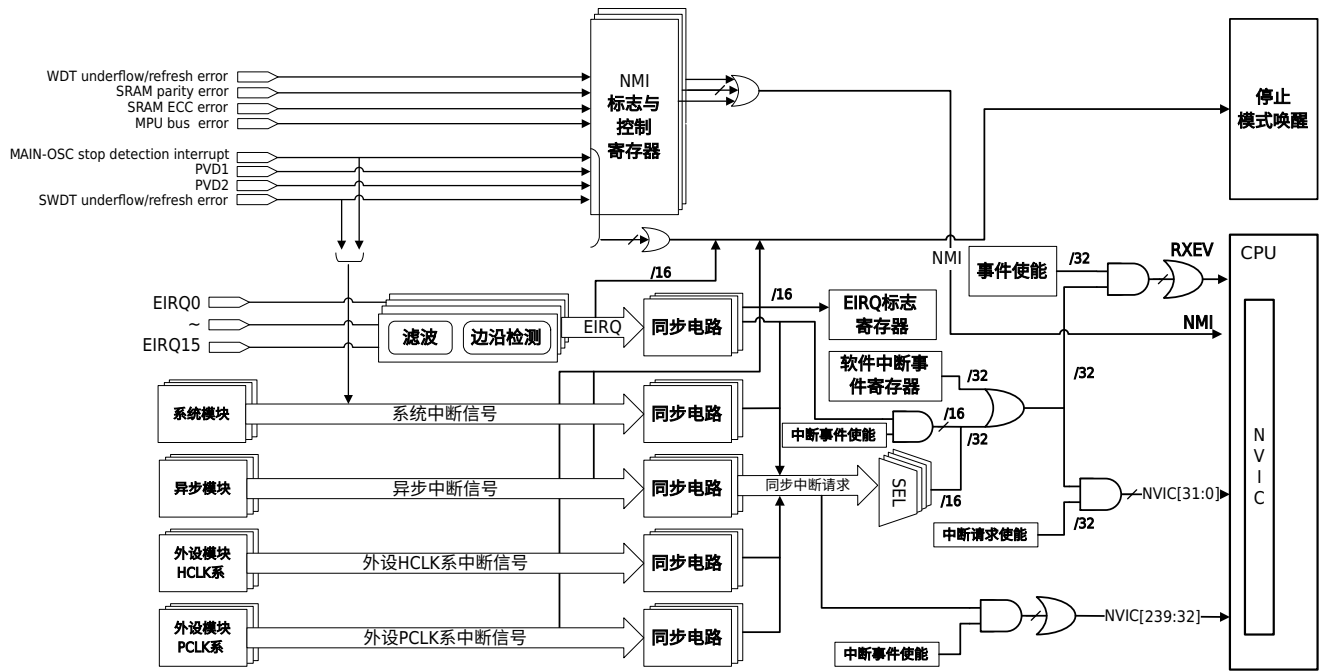


图 11-1 中断系统功能框图

11.3.2 不可屏蔽中断

不可屏蔽中断源如下：

- XTAL 振荡故障中断
- WDT 下溢/刷新错误中断
- SWDT 下溢/刷新错误中断
- 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，除非复位。

11.3.3 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 位，选择触发边沿或电平；设定 EISMPCLK 位，选择数字滤波器采样时钟；设定 EFEN 位，使能数字滤波器。

11.3.4 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见“中断事件表”。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。

11.3.5 软件中断

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

11.3.6 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：

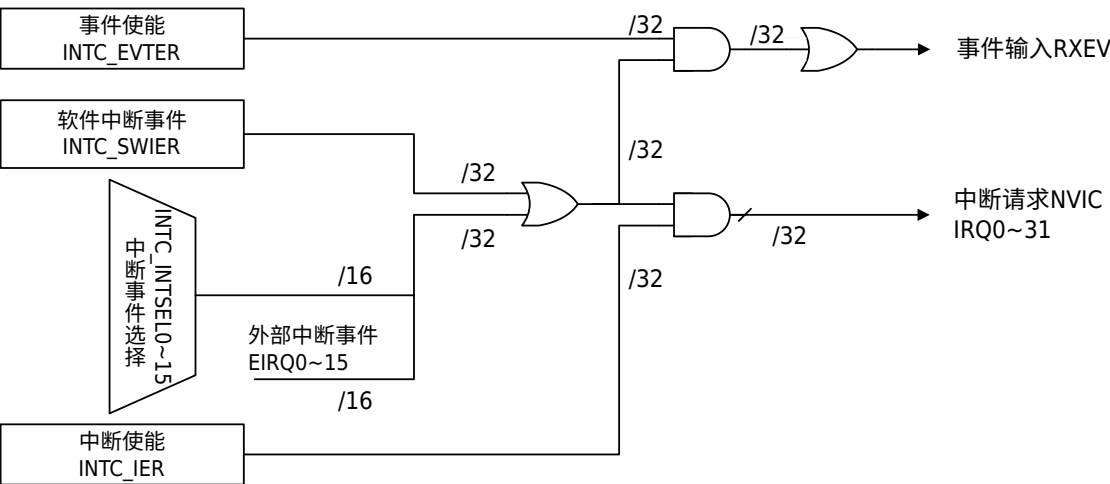


图 11-2 中断请求/事件输入选择

11.3.7 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 INTC_IER 寄存器（但不在 NVIC 中使能该中断请求）。INTC_EVTER 寄存器设定为非使能，同时使能 Cortex-M4 内核控制寄存器 SCR 中的 SEVONPEND 位。执行 WFE 使内核进入休眠模式。当选择的中断事件发生时，内核由休眠模式被唤醒，但不进入中断处理程序。
- 事件输入唤醒内核

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

11.3.8 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.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.EISMPCLK 位设定。

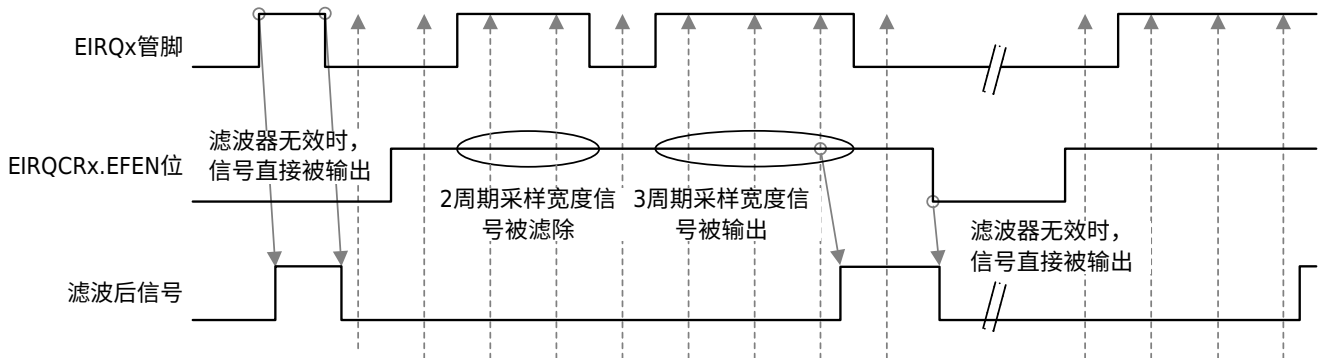


图 11-3 数字滤波器工作示意图

模拟滤波器通过 INTC_EIRQCRx.NOCEN 位使能。使能后以 INTC_EIRQCRx.NOCSEL 位选择的滤波宽度对信号进行滤波。

数字滤波器和模拟滤波器彼此独立，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。

11.3.9 低功耗模式返回

睡眠模式返回

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

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：XTAL 振荡故障中断、SWDT 下溢/刷新错误中断。
- 在 NVIC 中使能该中断请求

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、NRST 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。详细说明请参见《电源控制 (PWC)》章节相关描述。

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。

11.4 寄存器

11.4.1 寄存器总表

基地址：0x40051000

表 11-4 中断控制器 (INTC) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x0004	INTC_NMIER	不可屏蔽中断使能寄存器	32
0x0008	INTC_NMIFR	不可屏蔽中断标志寄存器	32
0x000C	INTC_NMIFCR	不可屏蔽中断标志清除寄存器	32
0x0010+0x4*x	INTC_EIRQCRx	外部中断控制寄存器 x (x=0~15)	32
0x0054	INTC_EIFR	外部中断标志寄存器	32
0x0058	INTC_EIFCR	外部中断标志清除寄存器	32
0x005C+0x4*x	INTC_INTSELx	中断事件选择寄存器 x (x=0~15)	32
0x009C+0x4*x	INTC_INTENx	中断事件使能寄存器 x (x=0~15)	32
0x0050	INTC_WKEN	停止模式唤醒事件使能寄存器	32
0x00DC	INTC_SWIER	软件中断事件寄存器	32
0x00E0	INTC_EVTER	事件使能寄存器	32
0x00E4	INTC_IER	中断请求使能寄存器	32

11.4.2 不可屏蔽中断使能寄存器 (INTC_NMIER)

Offset	Bit Position																															
0x0004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																				WDTEN	BUSERREN	RECCERREN	RPARERREN	Reserved	XATLSTPEN	Reserved	PVD2EN	PVD1EN	SWDTEN	Reserved	
Access																					RW	RW	RW	RW		RW		RW	RW	RW		

位/位域	标记	位名	功能描述	读写
31:12	Reserved	保留	-	-
11	WDTEN	WDT 下溢/刷新 错误中断选择	0b0: 禁止中断作为不可屏蔽中断源 0b1: 选择中断作为不可屏蔽中断源	RW
10	BUSERREN	MPU 主总线错误 中断选择	0b0: 禁止中断作为不可屏蔽中断源 0b1: 选择中断作为不可屏蔽中断源	RW
9	RECCERREN	SRAM ECC 校验 错误中断选择	0b0: 禁止中断作为不可屏蔽中断源 0b1: 选择中断作为不可屏蔽中断源	RW
8	RPARERREN	SRAM 奇偶校验 错误中断选择	0b0: 禁止中断作为不可屏蔽中断源 0b1: 选择中断作为不可屏蔽中断源	RW
7:6	Reserved	保留	-	-
5	XATLSTPEN	XTAL 振荡故障 中断选择	0b0: 禁止中断作为不可屏蔽中断源 0b1: 选择中断作为不可屏蔽中断源	RW
4	Reserved	保留	-	-
3	PVD2EN	低电压检测 PVD2 中断选择	0b0: 禁止中断作为不可屏蔽中断源 0b1: 选择中断作为不可屏蔽中断源	RW
2	PVD1EN	低电压检测 PVD1 中断选择	0b0: 禁止中断作为不可屏蔽中断源 0b1: 选择中断作为不可屏蔽中断源	RW
1	SWDTEN	SWDT 下溢/刷新 错误中断选择	0b0: 禁止中断作为不可屏蔽中断源 0b1: 选择中断作为不可屏蔽中断源	RW
0	Reserved	保留	-	-

11.4.3 不可屏蔽中断标志寄存器 (INTC_NMIFR)

Offset	Bit Position																																							
0x0008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Reset	0x00000000																																							
Name	Reserved																				RO	WDTF	RO	BUSERRF	RO	RECCERRF	RO	RPARERRF	Reserved	RO	XTALSTPF	Reserved	RO	PVD2F	RO	PVD1F	RO	SWDTF	Reserved	
Access																					RO		RO		RO		RO				RO		RO		RO		RO			

位/位域	标记	位名	功能描述	读写
31:12	Reserved	保留	-	-
11	WDTF	WDT 下溢/刷新错误中断标志	0b0: 没有发生 WDT 下溢/刷新错误中断申请 0b1: 发生 WDT 下溢/刷新错误中断申请	RO
10	BUSERRF	MPU 主总线错误中断标志	0b0: 没有发生 MPU 主总线错误中断申请 0b1: 发生 MPU 主总线错误中断申请	RO
9	RECCERRF	SRAM ECC 校验错误中断标志	0b0: 没有发生 SRAM ECC 校验错误中断申请 0b1: 发生 SRAM ECC 校验错误中断申请	RO
8	RPARERRF	SRAM 奇偶校验错误中断标志	0b0: 没有发生 SRAM 奇偶校验错误中断申请 0b1: 发生 SRAM 奇偶校验错误中断申请	RW
7:6	Reserved	保留	-	-
5	XTALSTPF	XTAL 振荡故障中断标志	0b0: 没有发生 XTAL 振荡故障中断申请 0b1: 发生 XTAL 振荡故障中断申请	RO
4	Reserved	保留	-	-
3	PVD2F	低电压检测 PVD2 中断标志	0b0: 没有发生低电压检测 PVD2 中断申请 0b1: 发生低电压检测 PVD2 中断申请	RO
2	PVD1F	低电压检测 PVD1 中断标志	0b0: 没有发生低电压检测 PVD1 中断申请 0b1: 发生低电压检测 PVD1 中断申请	RO
1	SWDTF	SWDT 下溢/刷新错误中断标志	0b0: 没有发生 SWDT 下溢/刷新错误中断申请 0b1: 发生 SWDT 下溢/刷新错误中断申请	RO
0	Reserved	保留	-	-

11.4.4 不可屏蔽中断标志清除寄存器 (INTC_NMIFCR)

Offset	Bit Position																																		
0x000C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Reset	0x00000000																																		
Name	Reserved																								WDTFCLR	BUSERRFCLR	RECCERRFCLR	RPARERRFCLR	Reserved	XATLSTPFCLR	Reserved	PVD2FCLR	PVD1FCLR	SWDTFCLR	Reserved
Access																									WO	WO	WO	WO		WO		WO	WO	WO	

位/位域	标记	位名	功能描述	读写
31:12	Reserved	保留	-	-
11	WDTFCLR	WDT 下溢/刷新 错误中断标志清除	写 1 清除标志，写 0 无效；读出恒为 0	WO
10	BUSERRFCLR	MPU 主总线错误 中断标志清除	写 1 清除标志，写 0 无效；读出恒为 0	WO
9	RECCERRFCLR	SRAM ECC 校验 错误中断标志清除	写 1 清除标志，写 0 无效；读出恒为 0	WO
8	RPARERRFCLR	SRAM 奇偶校验 错误中断标志清除	写 1 清除标志，写 0 无效；读出恒为 0	RW
7:6	Reserved	保留	-	-
5	XATLSTPFCLR	XTAL 振荡故障 中断标志清除	写 1 清除标志，写 0 无效；读出恒为 0	WO
4	Reserved	保留	-	-
3	PVD2FCLR	低电压检测 PVD2 中断标志清除	写 1 清除标志，写 0 无效；读出恒为 0	WO
2	PVD1FCLR	低电压检测 PVD1 中断标志清除	写 1 清除标志，写 0 无效；读出恒为 0	WO
1	SWDTFCLR	SWDT 下溢/刷新 错误中断标志清除	写 1 清除标志，写 0 无效；读出恒为 0	WO
0	Reserved	保留	-	-

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

偏移地址 (Offset) : 0x0010+0x4*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00020000																															
Name	Reserved															NOCEN	Reserved	NOCSEL	Reserved					EFEN	Reserved	EISMPCLK	Reserved			EIRQTRG		
Access																RW		RW						RW		RW				RW		

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15	NOCEN	EIRQ 模拟滤波器使能	0b0: 禁止模拟滤波器功能 0b1: 允许模拟滤波器功能 滤波宽度由 NOCSEL 决定。	RW
14	Reserved	保留	-	-
13:12	NOCSEL	模拟滤波滤波宽度选择	0b00: 滤波宽度档位 1 0b01: 滤波宽度档位 2 0b10: 滤波宽度档位 3 0b11: 滤波宽度档位 4 各档位具体值请参见《数据手册》的“电气特性-EIRQ 滤波特性”章节相关描述。	RW
11:8	Reserved	保留	-	-
7	EFEN	EIRQ 数字滤波器使能	0b0: 禁止数字滤波器功能 0b1: 允许数字滤波器功能	RW
6	Reserved	保留	-	-
5:4	EISMPCLK	数字滤波采样周期选择	0b00: 1 × PCLK3 周期 0b01: 8 × PCLK3 周期 0b10: 32 × PCLK3 周期 0b11: 64 × PCLK3 周期	RW
3:2	Reserved	保留	-	-
1:0	EIRQTRG	触发选择	0b00: 下降沿 0b01: 上升沿 0b10: 双边沿 0b11: 低电平	RW

11.4.6 外部中断标志寄存器 (INTC_EIFR)

Offset	Bit Position																															
0x0054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																RO EIF															
Access																																

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	EIF	EIRQ 标志位	对应 EIF15~EIF0： ● 0b0：EIRQ 事件未发生，或写 EIFCR 位清除 ● 0b1：选择的 EIRQ 事件发生	RO

11.4.7 外部中断标志清除寄存器 (INTC{EIFCR)

Offset	Bit Position																															
0x0058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																EIFCLR															
Access																	WO															

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	EIFCLR	EIRQ 标志清除位	对应 EIFCLR15~EIFCLR0，写 1 清除 EIFR 寄存器对应位，写 0 无效；读出恒为 0	WO

11.4.8 中断事件选择寄存器 (INTC_INTSELx) (x=0~15)

偏移地址 (Offset) : 0x005C+0x4*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000001FF																															
Name	Reserved																								INTSEL							
Access																																

位/位域	标记	位名	功能描述	读写
31:9	Reserved	保留	-	-
8:0	INTSEL	中断事件请求选择	选择 0x000~0x1FE 对应的中断事件请求，详见 中断事件表	RW

11.4.9 中断事件使能寄存器 (INTC_INTENx) (x=0~15)

偏移地址 (Offset) : 0x009C+0x4*x

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	INTENy																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	INTENy (y=31~0)	中断事件使能	对应中断事件序号 z (z=x*32+y) 0b0: 中断事件无效 0b1: 中断事件有效  说明 - 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。 - 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

11.4.10 停止模式唤醒事件使能寄存器 (INTC_WKEN)

Offset	Bit Position																																
0x0050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	CMP4WKEN	CMP3WKEN	CMP2WKEN	Reserved	RXWKEN	Reserved	TMR0CMPWKEN	RTCPRDWKEN	RTCALMWKEN	WKTMWKEN	CMP1WKEN	Reserved	SWDTWKEN	EIRQWKEN																			
Access	RW	RW	RW		RW		RW	RW	RW	RW	RW		RW																				RW

位/位域	标记	位名	功能描述	读写
31	CMP4WKEN	CMP4 停止模式唤醒使能	0b0: 唤醒禁止 0b1: 唤醒许可	RW
30	CMP3WKEN	CMP3 停止模式唤醒使能	0b0: 唤醒禁止 0b1: 唤醒许可	RW
29	CMP2WKEN	CMP2 停止模式唤醒使能	0b0: 唤醒禁止 0b1: 唤醒许可	RW
28:27	Reserved	保留	-	-
26	RXWKEN	USART_1 RX 线停止模式唤醒使能	0b0: 唤醒禁止 0b1: 唤醒许可	RW
25:24	Reserved	保留	-	-
23	TMR0CMPWKEN	TMR0_1_CMPA 停止模式唤醒使能	0b0: 唤醒禁止 0b1: 唤醒许可	RW
22	RTCPRDWKEN	RTC_PRD 停止模式唤醒使能	0b0: 唤醒禁止 0b1: 唤醒许可	RW
21	RTCALMWKEN	RTC_ALM 停止模式唤醒使能	0b0: 唤醒禁止 0b1: 唤醒许可	RW
20	WKTMWKEN	WKT_M_CMP 停止模式唤醒使能	0b0: 唤醒禁止 0b1: 唤醒许可	RW
19	CMP1WKEN	CMP1 停止模式唤醒使能	0b0: 唤醒禁止 0b1: 唤醒许可	RW
18:17	Reserved	保留	-	-
16	SWDTWKEN	SWDT_REFUDF 停止模式唤醒使能	0b0: 唤醒禁止 0b1: 唤醒许可	RW
15:0	EIRQWKEN	EIRQ 停止模式唤醒使能	对应 EIRQWKEN15~EIRQWKEN0: 0b0: 唤醒禁止 0b1: 唤醒许可	RW

11.4.11 软件中断事件寄存器 (INTC_SWIER)

Offset	Bit Position																															
0x00DC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	SWIE																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	SWIE	软件中断/事件	对应 SWIE31~SWIE0： ● 0b0：不发生软件中断/事件 ● 0b1：发生软件中断/事件  说明 由“0”写“1”后发生一次软件中断/事件。再次发生必须先写“0”清除。	RW

11.4.12 事件使能寄存器 (INTC_EVTER)

Offset	Bit Position																															
0x00E0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	EVTE																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	EVTE	事件使能	对应 EVTE31~EVTE0： ● 0b0：事件选择禁止 ● 0b1：事件选择许可	RW

11.4.13 中断请求使能寄存器 (INTC_IER)

Offset	Bit Position																															
0x00E4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	IEN																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	IEN	中断使能	IEN31~IEN0 分别对应 NVIC 中断请求 31~0： ● 0b0：中断禁止 ● 0b1：中断许可 禁止时，各中断请求对应的中断事件将不能被 NVIC 接收。	RW

11.5 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) 》。

12 自动运行系统 (AOS)

12.1 简介

自动运行系统 (Automatic Operation System) 用于在不借助 CPU 的情况下实现外设之间的联动。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)。

AOS 中还配备了 4 个可编程逻辑运算单元 (PLU)，用于 PORT 和 AOS 源的逻辑运算。用户可以根据需求选择一个或多个 AOS 源来触发同一个 AOS 目标。

12.2 主要特性

AOS 有多个触发源，详见 [AOS 源事件](#)。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。需要选择多个触发源时可以使用 AOS 配备的逻辑运算单元 (PLU)，最多可以同时选择 4 个触发源。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 6 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：

- 4 个 DCU 触发目标
- 13 个 DMA 触发目标
- 1 个通用定时器 0 (Timer0) 触发目标
- 2 个通用控制定时器 4 (Timer4) 触发目标
- 2 个高级控制定时器 (Timer6) 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

12.3 功能说明

12.3.1 功能框图

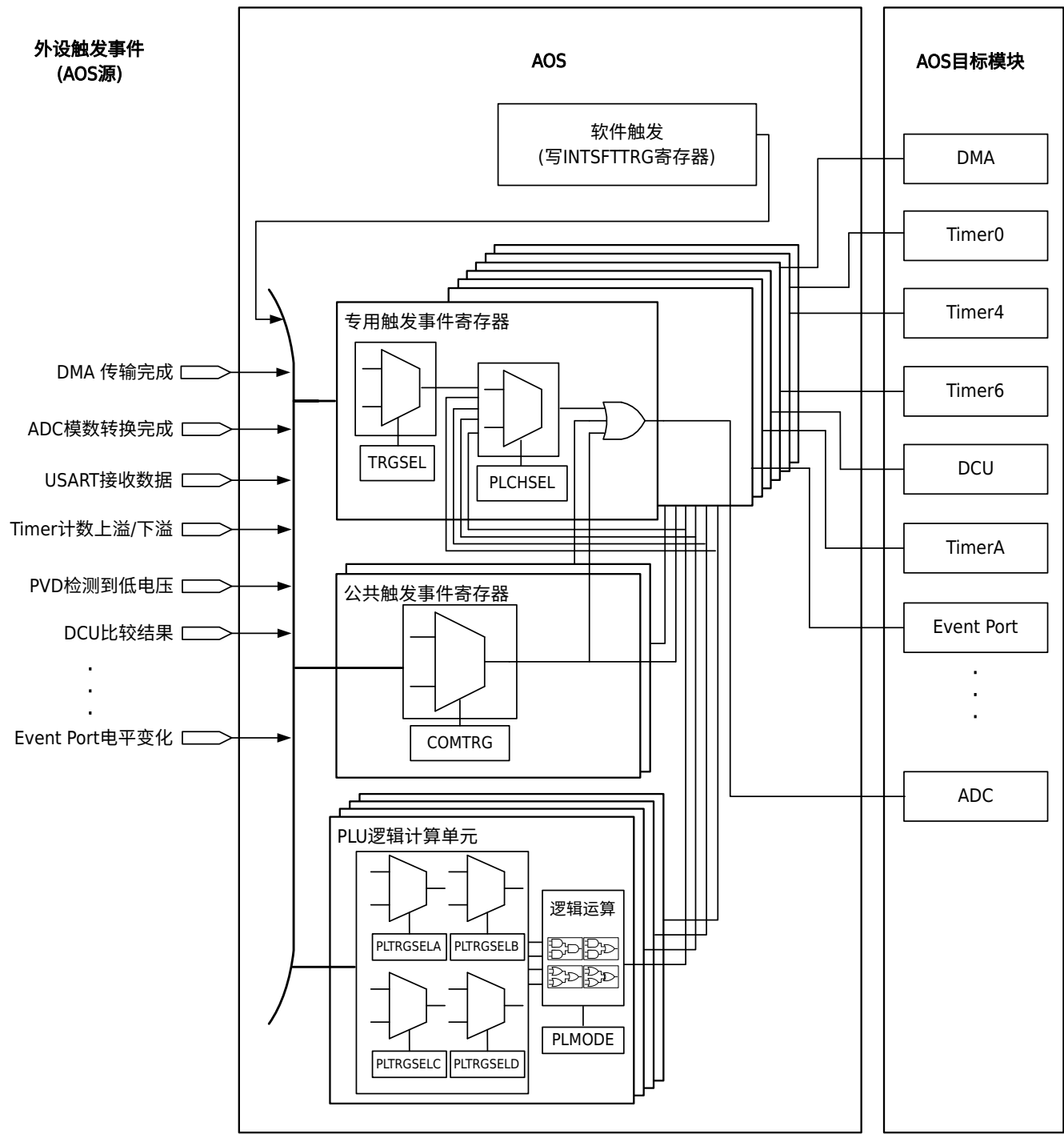


图 12-1 AOS 模块框图

12.3.2 AOS 源事件和目标

AOS 源事件编号见《中断控制器（INTC）》章节的中断事件表，表中“触发源”列中打“√”的事件可以用作 AOS 源。AOS 目标如下表所示。

表 12-1 AOS 目标列表

模块	序号	动作
DCU_1	0	触发加/减法运算，三角波/锯齿波模式下触发数据递增或递减
DCU_2	1	触发加/减法运算，三角波/锯齿波模式下触发数据递增或递减
DCU_3	2	触发加/减法运算，三角波/锯齿波模式下触发数据递增或递减
DCU_4	3	触发加/减法运算，三角波/锯齿波模式下触发数据递增或递减
DMA_1	4	触发通道 0 开始传输
	5	触发通道 1 开始传输
	6	触发通道 2 开始传输
	7	触发通道 3 开始传输
	8	触发通道 4 开始传输
	9	触发通道 5 开始传输
DMA_2	10	触发通道 0 开始传输
	11	触发通道 1 开始传输
	12	触发通道 2 开始传输
	13	触发通道 3 开始传输
	14	触发通道 4 开始传输
	15	触发通道 5 开始传输
DMA_1/2	16	触发事件触发通道重置
Timer6	17	触发计数器启动/停止/清零/刷新/捕获/递增/递减
	18	触发计数器启动/停止/清零/刷新/捕获/递增/递减
Timer4	19	触发单元 1 计数器启动/停止/清零/捕获
	20	触发单元 2 计数器启动/停止/清零/捕获
	21	触发单元 3 计数器启动/停止/清零/捕获
Event Port	22	触发 Event Port1/2 动作
	23	触发 Event Port3/4 动作
Timer0	24	触发计数器启动/停止/清零/捕获
TimerA	25	触发单元 1/5 计数器启动/停止/清零/递增/递减
		触发单元 2 计数器输入捕获
	26	触发单元 2 计数器启动/停止/清零/递增/递减
		触发单元 1/5 计数器输入捕获

模块	序号	动作
TimerA	27	触发单元 4 计数器输入捕获
		触发单元 3 计数器启动/停止/清零/递增/递减
	28	触发单元 4 计数器启动/停止/清零/递增/递减
		触发单元 3 计数器输入捕获
ADC_1	29、30	触发序列 A/B 开始模数转换
ADC_2	31、32	触发序列 A/B 开始模数转换
ADC_3	33、34	触发序列 A/B 开始模数转换

12.3.3 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。

12.3.4 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。因此使用时当其他 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。

12.3.5 可编程逻辑运算单元（PLU）

AOS 中一共配备了 4 个可编程逻辑运算单元（PLU），每个单元有 4 个输入（PLINA、PLINB、PLINC、PLIND）和 1 个输出（PLOUT）。

四个输入 PLINA~D 的源不仅可以是 AOS 触发源，也可以是 PORT 输入，或者是固定值。逻辑运算后的结果不仅可以用于 AOS 触发目标模块，也可以通过 PORT 输出。

设置 AOS_PLUx_CR.PLINySEL（x=0~3，y=A、B、C、D）位可以选择 PLINA~D 的输入源。设置 AOS_PLUx_CR.PLMODE 位可以选择需要的逻辑运算关系。

PLUx_INA、PLUx_INB、PLUx_INC、PLUx_IND 和 PLUx_OUT 的配置请参见《数据手册》中“引脚功能说明”章节。

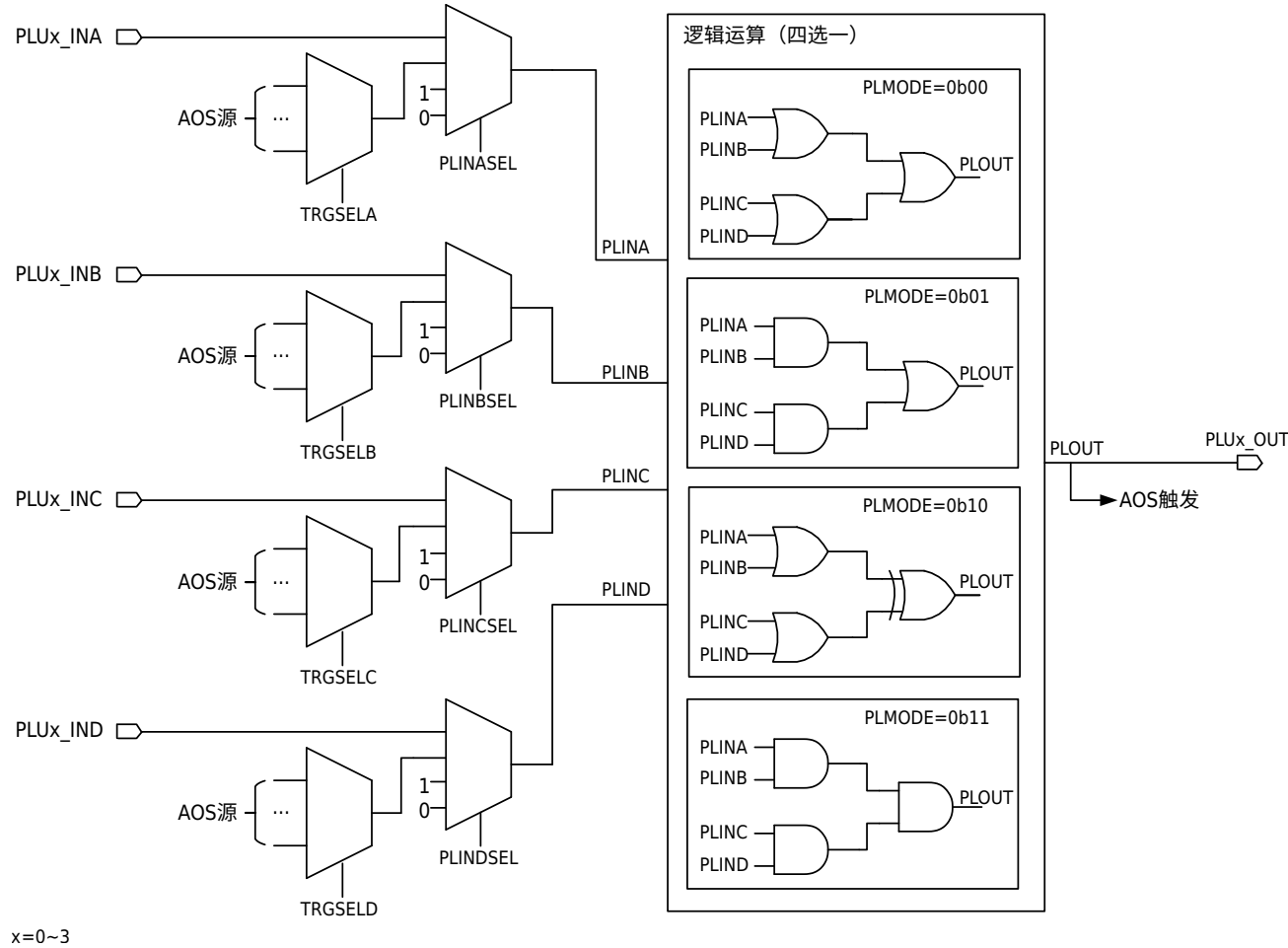


图 12-2 PLU 框图

12.4 寄存器

12.4.1 寄存器总表

基地址：0x40010800

表 12-2 自动运行系统 (AOS) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	INTSFTRG	软件触发事件寄存器	32
0x04*x	DCU_TRGSELx	DCU 触发源选择寄存器 (x=1~4)	32
0x14+0x04*x	DMA1_TRGSELx	DMA1 传输启动触发源选择寄存器 (x=0~5)	32
0x2C+0x04*x	DMA2_TRGSELx	DMA2 传输启动触发源选择寄存器 (x=0~5)	32
0x44	DMA_RC_TRGSEL	DMA 通道重置触发源选择寄存器	32
0x48+0x04*x	TMR6_TRGSELx	Timer6 硬件触发事件选择寄存器 (x=0~1)	32
0x50+0x04*x	TMR4_TRGSELx	Timer4 硬件触发事件选择寄存器 (x=0~2)	32
0x5C	PEVNT_TRGSEL12	Event Port1/2 触发源选择寄存器	32
0x60	PEVNT_TRGSEL34	Event Port3/4 触发源选择寄存器	32
0x64	TMR0_TRGSEL	Timer0 硬件触发事件选择寄存器	32
0x68+0x04*x	TMRA_TRGSELx	TimerA 硬件触发事件选择寄存器 (x=0~3)	32
0x78+0x04*x	ADC1_TRGSELx	ADC1 转换开始片内触发源选择寄存器 (x=0~1)	32
0x80+0x04*x	ADC2_TRGSELx	ADC2 转换开始片内触发源选择寄存器 (x=0~1)	32
0x88+0x04*x	ADC3_TRGSELx	ADC3 转换开始片内触发源选择寄存器 (x=0~1)	32
0x90	AOS_COMTRGSEL1	公共触发源选择寄存器 1	32
0x94	AOS_COMTRGSEL2	公共触发源选择寄存器 2	32
0x200+0x004*x	AOS_PLUX_CR	PLU 控制寄存器 (x=0~3)	32
0x210+0x010*x	AOS_PLUX_TRGSELA	PLU 触发源选择寄存器 A (x=0~3)	32
0x214+0x010*x	AOS_PLUX_TRGSELB	PLU 触发源选择寄存器 B (x=0~3)	32
0x218+0x010*x	AOS_PLUX_TRGSELC	PLU 触发源选择寄存器 C (x=0~3)	32
0x21C+0x010*x	AOS_PLUX_TRGSELD	PLU 触发源选择寄存器 D (x=0~3)	32

12.4.2 软件触发事件寄存器（INTSFTRG）

该寄存器每次写 1 将产生一次触发事件。

Offset	Bit Position																															
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															STRG
Access																																WO

位/位域	标记	位名	功能描述	读写
31:1	Reserved	保留	-	-
0	STRG	软件触发	写 1 产生一次软件触发事件，写 0 无效	WO

12.4.3 DCU 触发源选择寄存器 (DCU_TRGSELx) (x=1~4)

在 DCU 选择硬件触发启动模式后，请用该寄存器来选择触发事件源。当 PLCHSEL[2:0]设为 0b000 时，请将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。当 PLCHSEL[2:0]设为 0b001、0b010、0b011、0b100 时，请将触发事件的编号写入对应的 PLU 触发源选择寄存器 AOS_PLU_TRGSELA/B/C/D 中，并设置 PLU 控制寄存器 AOS_PLU_CR。当编号对应的外设事件发生后，DCU 将由该事件触发启动并进行运算。

每个 DCU 单元对应一个寄存器。

偏移地址 (Offset) : 0x04*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000001FF																															
Name	COMEN	Reserved												PLCHSEL	Reserved								TRGSEL									
Access	RW													RW									RW									

位/位域	标记	位名	功能描述	读写
31	COMEN[1]	公共触发使能	0b0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 0b1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	RW
30	COMEN[0]	公共触发使能	0b0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 0b1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	RW
29:19	Reserved	保留	-	-
18:16	PLCHSEL	PLU 选择位	0b000: 使用 TRGSEL[8:0]位选择的事件触发本 AOS 目标 0b001: 使用 PLU0 运算后的事件触发本 AOS 目标 0b010: 使用 PLU1 运算后的事件触发本 AOS 目标 0b011: 使用 PLU2 运算后的事件触发本 AOS 目标 0b100: 使用 PLU3 运算后的事件触发本 AOS 目标 - 其他: 禁止设置	RW
15:9	Reserved	保留	-	-
8:0	TRGSEL	触发源选择	写入要选择的外设电路事件的编号	RW

12.4.4 DMA1 传输启动触发源选择寄存器（DMA1_TRGSELx）（x=0~5）

在 DMA1 使能后，请用该寄存器来选择触发事件源。当 PLCHSEL[2:0]设为 0b000 时，请将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。当 PLCHSEL[2:0]设为 0b001、0b010、0b011、0b100 时，请将触发事件的编号写入对应的 PLU 触发源选择寄存器 AOS_PLU_TRGSELA/B/C/D 中，并设置 PLU 控制寄存器 AOS_PLU_CR。当编号对应的外设事件发生时，DMA1 将由该事件触发启动并进行传输。

每个数据传输通道对应一个寄存器。

偏移地址（Offset）：0x14+0x04*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000001FF																															
Name	COMEN	Reserved												PLCHSEL	Reserved								TRGSEL									
Access	RW													RW									RW									

位/位域	标记	位名	功能描述	读写
31	COMEN[1]	公共触发使能	0b0：禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 0b1：允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	RW
30	COMEN[0]	公共触发使能	0b0：禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 0b1：允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	RW
29:19	Reserved	保留	-	-
18:16	PLCHSEL	PLU 选择位	0b000：使用 TRGSEL[8:0]位选择的事件触发本 AOS 目标 0b001：使用 PLU0 运算后的事件触发本 AOS 目标 0b010：使用 PLU1 运算后的事件触发本 AOS 目标 0b011：使用 PLU2 运算后的事件触发本 AOS 目标 0b100：使用 PLU3 运算后的事件触发本 AOS 目标 - 其他：禁止设置	RW
15:9	Reserved	保留	-	-
8:0	TRGSEL	触发源选择	写入要选择的外设电路事件的编号	RW

12.4.5 DMA2 传输启动触发源选择寄存器 (DMA2_TRGSELx) (x=0~5)

在 DMA2 使能后，请用该寄存器来选择触发事件源。当 PLCHSEL[2:0]设为 0b000 时，请将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。当 PLCHSEL[2:0]设为 0b001、0b010、0b011、0b100 时，请将触发事件的编号写入对应的 PLU 触发源选择寄存器 AOS_PLU_TRGSELA/B/C/D 中，并设置 PLU 控制寄存器 AOS_PLU_CR。当编号对应的外设事件发生时，DMA2 将由该事件触发启动并进行传输。

每个数据传输通道对应一个寄存器。

偏移地址 (Offset) : 0x2C+0x04*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000001FF																															
Name	COMEN	Reserved												PLCHSEL	Reserved								TRGSEL									
Access	RW													RW									RW									

位/位域	标记	位名	功能描述	读写
31	COMEN[1]	公共触发使能	0b0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 0b1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	RW
30	COMEN[0]	公共触发使能	0b0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 0b1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	RW
29:19	Reserved	保留	-	-
18:16	PLCHSEL	PLU 选择位	0b000: 使用 TRGSEL[8:0]位选择的事件触发本 AOS 目标 0b001: 使用 PLU0 运算后的事件触发本 AOS 目标 0b010: 使用 PLU1 运算后的事件触发本 AOS 目标 0b011: 使用 PLU2 运算后的事件触发本 AOS 目标 0b100: 使用 PLU3 运算后的事件触发本 AOS 目标 - 其他: 禁止设置	RW
15:9	Reserved	保留	-	-
8:0	TRGSEL	触发源选择	写入要选择的外设电路事件的编号	RW

12.4.6 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。当 PLCHSEL[2:0]设为 0b000 时，请将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。当 PLCHSEL[2:0]设为 0b001、0b010、0b011、0b100 时，请将触发事件的编号写入对应的 PLU 触发源选择寄存器 AOS_PLU_TRGSELA/B/C/D 中，并设置 PLU 控制寄存器 AOS_PLU_CR。当编号对应的外设事件发生时，DMA 将由该事件触发进行通道重置。

DMA1 和 DMA2 共用这个寄存器。

Offset	Bit Position																															
0x44	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000001FF																															
Name	COMEN	Reserved												PLCHSEL	Reserved										TRGSEL							
Access		RW													RW																	

位/位域	标记	位名	功能描述	读写
31	COMEN[1]	公共触发使能	0b0：禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 0b1：允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	RW
30	COMEN[0]	公共触发使能	0b0：禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 0b1：允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	RW
29:19	Reserved	保留	-	-
18:16	PLCHSEL	PLU 选择位	0b000：使用 TRGSEL[8:0]位选择的事件触发本 AOS 目标 0b001：使用 PLU0 运算后的事件触发本 AOS 目标 0b010：使用 PLU1 运算后的事件触发本 AOS 目标 0b011：使用 PLU2 运算后的事件触发本 AOS 目标 0b100：使用 PLU3 运算后的事件触发本 AOS 目标 - 其他：禁止设置	RW
15:9	Reserved	保留	-	-
8:0	TRGSEL	触发源选择	写入要选择的外设电路事件的编号	RW

12.4.7 Timer6 硬件触发事件选择寄存器 (TMR6_TRGSELx) (x=0~1)

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。当 PLCHSEL[2:0]设为 0b000 时，请将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。当 PLCHSEL[2:0]设为 0b001、0b010、0b011、0b100 时，请将触发事件的编号写入对应的 PLU 触发源选择寄存器 AOS_PLU_TRGSELA/B/C/D 中，并设置 PLU 控制寄存器 AOS_PLU_CR。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 2 个寄存器。

偏移地址 (Offset)：0x48+0x04*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000001FF																															
Name	COMEN	Reserved												PLCHSEL	Reserved								TRGSEL									
Access	RW													RW									RW									

位/位域	标记	位名	功能描述	读写
31	COMEN[1]	公共触发使能	0b0：禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 0b1：允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	RW
30	COMEN[0]	公共触发使能	0b0：禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 0b1：允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	RW
29:19	Reserved	保留	-	-
18:16	PLCHSEL	PLU 选择位	0b000：使用 TRGSEL[8:0]位选择的事件触发本 AOS 目标 0b001：使用 PLU0 运算后的事件触发本 AOS 目标 0b010：使用 PLU1 运算后的事件触发本 AOS 目标 0b011：使用 PLU2 运算后的事件触发本 AOS 目标 0b100：使用 PLU3 运算后的事件触发本 AOS 目标 - 其他：禁止设置	RW
15:9	Reserved	保留	-	-
8:0	TRGSEL	触发源选择	写入要选择的外设电路事件的编号	RW

12.4.8 Timer4 硬件触发事件选择寄存器 (TMR4_TRGSELx) (x=0~2)

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。当 PLCHSEL[2:0]设为 0b000 时，请将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。当 PLCHSEL[2:0]设为 0b001、0b010、0b011、0b100 时，请将触发事件的编号写入对应的 PLU 触发源选择寄存器 AOS_PLU_TRGSELA/B/C/D 中，并设置 PLU 控制寄存器 AOS_PLU_CR。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。

每个 Timer4 单元对应一个寄存器。

偏移地址 (Offset) : 0x50+0x04*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000001FF																															
Name	COMEN	Reserved											PLCHSEL	Reserved											TRGSEL							
Access	RW												RW												RW							

位/位域	标记	位名	功能描述	读写
31	COMEN[1]	公共触发使能	0b0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 0b1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	RW
30	COMEN[0]	公共触发使能	0b0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 0b1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	RW
29:19	Reserved	保留	-	-
18:16	PLCHSEL	PLU 选择位	0b000: 使用 TRGSEL[8:0]位选择的事件触发本 AOS 目标 0b001: 使用 PLU0 运算后的事件触发本 AOS 目标 0b010: 使用 PLU1 运算后的事件触发本 AOS 目标 0b011: 使用 PLU2 运算后的事件触发本 AOS 目标 0b100: 使用 PLU3 运算后的事件触发本 AOS 目标 - 其他: 禁止设置	RW
15:9	Reserved	保留	-	-
8:0	TRGSEL	触发源选择	写入要选择的外设电路事件的编号	RW

12.4.9 Event Port 触发源选择寄存器（PEVNT_TRGSEL12/PEVNT_TRGSEL34）

将触发事件的编号写入该寄存器中，请用该寄存器来选择触发事件源。当 PLCHSEL[2:0]设为 0b000 时，请将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。当 PLCHSEL[2:0]设为 0b001、0b010、0b011、0b100 时，请将触发事件的编号写入对应的 PLU 触发源选择寄存器 AOS_PLU_TRGSELA/B/C/D 中，并设置 PLU 控制寄存器 AOS_PLU_CR。Event Port 被触发输出指定电平或者锁存端口状态。

Event Port1/2 共用 PEVNT_TRGSEL12 寄存器，Event Port3/4 共用 PEVNT_TRGSEL34 寄存器。

偏移地址（Offset）：0x5C（PEVNT_TRGSEL12），0x60（PEVNT_TRGSEL34）

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000001FF																															
Name	COMEN	Reserved										PLCHSEL	Reserved										TRGSEL									
Access	RW											RW											RW									

位/位域	标记	位名	功能描述	读写
31	COMEN[1]	公共触发使能	0b0：禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 0b1：允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	RW
30	COMEN[0]	公共触发使能	0b0：禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 0b1：允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	RW
29:19	Reserved	保留	-	-
18:16	PLCHSEL	PLU 选择位	0b000：使用 TRGSEL[8:0]位选择的事件触发本 AOS 目标 0b001：使用 PLU0 运算后的事件触发本 AOS 目标 0b010：使用 PLU1 运算后的事件触发本 AOS 目标 0b011：使用 PLU2 运算后的事件触发本 AOS 目标 0b100：使用 PLU3 运算后的事件触发本 AOS 目标 - 其他：禁止设置	RW
15:9	Reserved	保留	-	-
8:0	TRGSEL	触发源选择	写入要选择的外设电路事件的编号	RW

12.4.10 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。当 PLCHSEL[2:0]设为 0b000 时，请将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。当 PLCHSEL[2:0]设为 0b001、0b010、0b011、0b0100 时，请将触发事件的编号写入对应的 PLU 触发源选择寄存器 AOS_PLU_TRGSELA/B/C/D 中，并设置 PLU 控制寄存器 AOS_PLU_CR。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。

Offset	Bit Position																																					
0x64	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x000001FF																																					
Name	COMEN	Reserved												PLCHSEL	Reserved								TRGSEL															
Access	RW													RW									RW															

位/位域	标记	位名	功能描述	读写
31	COMEN[1]	公共触发使能	0b0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 0b1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	RW
30	COMEN[0]	公共触发使能	0b0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 0b1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	RW
29:19	Reserved	保留	-	-
18:16	PLCHSEL	PLU 选择位	0b000: 使用 TRGSEL[8:0]位选择的事件触发本 AOS 目标 0b001: 使用 PLU0 运算后的事件触发本 AOS 目标 0b010: 使用 PLU1 运算后的事件触发本 AOS 目标 0b011: 使用 PLU2 运算后的事件触发本 AOS 目标 0b100: 使用 PLU3 运算后的事件触发本 AOS 目标 - 其他: 禁止设置	RW
15:9	Reserved	保留	-	-
8:0	TRGSEL	触发源选择	写入要选择的外设电路事件的编号	RW

12.4.11 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。当 PLCHSEL[2:0]设为 0b000 时，请将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。当 PLCHSEL[2:0]设为 0b001、0b010、0b011、0b100 时，请将触发事件的编号写入对应的 PLU 触发源选择寄存器 AOS_PLU_TRGSELA/B/C/D 中，并设置 PLU 控制寄存器 AOS_PLU_CR。当编号对应的外设事件发生时，TimerA 将由该事件触发动作（包括启动/停止/清零/输入捕获）。

所有 TimerA 单元共用这 4 个寄存器。

偏移地址（Offset）：0x68+0x04*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000001FF																															
Name	COMEN	Reserved												PLCHSEL	Reserved										TRGSEL							
Access	RW													RW											RW							

位/位域	标记	位名	功能描述	读写
31	COMEN[1]	公共触发使能	0b0：禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 0b1：允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	RW
30	COMEN[0]	公共触发使能	0b0：禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 0b1：允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	RW
29:19	Reserved	保留	-	-
18:16	PLCHSEL	PLU 选择位	0b000：使用 TRGSEL[8:0]位选择的事件触发本 AOS 目标 0b001：使用 PLU0 运算后的事件触发本 AOS 目标 0b010：使用 PLU1 运算后的事件触发本 AOS 目标 0b011：使用 PLU2 运算后的事件触发本 AOS 目标 0b100：使用 PLU3 运算后的事件触发本 AOS 目标 - 其他：禁止设置	RW
15:9	Reserved	保留	-	-
8:0	TRGSEL	触发源选择	写入要选择的外设电路事件的编号	RW

12.4.12 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。当 PLCHSEL[2:0]设为 0b000 时，请将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。当 PLCHSEL[2:0]设为 0b001、0b010、0b011、0b100 时，请将触发事件的编号写入对应的 PLU 触发源选择寄存器 AOS_PLU_TRGSELA/B/C/D 中，并设置 PLU 控制寄存器 AOS_PLU_CR。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。

偏移地址（Offset）：0x78+0x04*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000001FF																															
Name	COMEN	Reserved												PLCHSEL	Reserved								TRGSEL									
Access	RW													RW									RW									

位/位域	标记	位名	功能描述	读写
31	COMEN[1]	公共触发使能	0b0：禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 0b1：允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	RW
30	COMEN[0]	公共触发使能	0b0：禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 0b1：允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	RW
29:19	Reserved	保留	-	-
18:16	PLCHSEL	PLU 选择位	0b000：使用 TRGSEL[8:0]位选择的事件触发本 AOS 目标 0b001：使用 PLU0 运算后的事件触发本 AOS 目标 0b010：使用 PLU1 运算后的事件触发本 AOS 目标 0b011：使用 PLU2 运算后的事件触发本 AOS 目标 0b100：使用 PLU3 运算后的事件触发本 AOS 目标 - 其他：禁止设置	RW
15:9	Reserved	保留	-	-
8:0	TRGSEL	触发源选择	写入要选择的外设电路事件的编号	RW

12.4.13 ADC2 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。当 PLCHSEL[2:0]设为 0b000 时，请将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。当 PLCHSEL[2:0]设为 0b001、0b010、0b011、0b100 时，请将触发事件的编号写入对应的 PLU 触发源选择寄存器 AOS_PLU_TRGSELA/B/C/D 中，并设置 PLU 控制寄存器 AOS_PLU_CR。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 的序列 A 和序列 B 共用这 2 个寄存器。

偏移地址 (Offset) : 0x80+0x04*x

Offset	Bit Position																																					
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x000001FF																																					
Name	COMEN	Reserved											PLCHSEL	Reserved											TRGSEL													
Access	RW												RW												RW													

位/位域	标记	位名	功能描述	读写
31	COMEN[1]	公共触发使能	0b0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 0b1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	RW
30	COMEN[0]	公共触发使能	0b0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 0b1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	RW
29:19	Reserved	保留	-	-
18:16	PLCHSEL	PLU 选择位	0b000: 使用 TRGSEL[8:0]位选择的事件触发本 AOS 目标 0b001: 使用 PLU0 运算后的事件触发本 AOS 目标 0b010: 使用 PLU1 运算后的事件触发本 AOS 目标 0b011: 使用 PLU2 运算后的事件触发本 AOS 目标 0b100: 使用 PLU3 运算后的事件触发本 AOS 目标 - 其他: 禁止设置	RW
15:9	Reserved	保留	-	-
8:0	TRGSEL	触发源选择	写入要选择的外设电路事件的编号	RW

12.4.14 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。当 PLCHSEL[2:0]设为 0b000 时，请将触发事件的编号写入该寄存器的 TRGSEL[8:0]位中。当 PLCHSEL[2:0]设为 0b001、0b010、0b011、0b100 时，请将触发事件的编号写入对应的 PLU 触发源选择寄存器 AOS_PLU_TRGSELA/B/C/D 中，并设置 PLU 控制寄存器 AOS_PLU_CR。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。

偏移地址 (Offset) : 0x88+0x04*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000001FF																															
Name	COMEN	Reserved												PLCHSEL	Reserved								TRGSEL									
Access	RW													RW									RW									

位/位域	标记	位名	功能描述	读写
31	COMEN[1]	公共触发使能	0b0: 禁止 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标 0b1: 允许 AOS_COMTRGSEL2 的公共触发事件触发本 AOS 目标	RW
30	COMEN[0]	公共触发使能	0b0: 禁止 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标 0b1: 允许 AOS_COMTRGSEL1 的公共触发事件触发本 AOS 目标	RW
29:19	Reserved	保留	-	-
18:16	PLCHSEL	PLU 选择位	0b000: 使用 TRGSEL[8:0]位选择的事件触发本 AOS 目标 0b001: 使用 PLU0 运算后的事件触发本 AOS 目标 0b010: 使用 PLU1 运算后的事件触发本 AOS 目标 0b011: 使用 PLU2 运算后的事件触发本 AOS 目标 0b100: 使用 PLU3 运算后的事件触发本 AOS 目标 - 其他: 禁止设置	RW
15:9	Reserved	保留	-	-
8:0	TRGSEL	触发源选择	写入要选择的外设电路事件的编号	RW

12.4.15 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。

偏移地址（Offset）：0x90（AOS_COMTRGSEL1），0x94（AOS_COMTRGSEL2）

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000001FF																															
Name	Reserved																								COMTRG							
Access																									RW							

位/位域	标记	位名	功能描述	读写
31:9	Reserved	保留	-	-
8:0	COMTRG	触发源选择	写入要选择的外设电路事件的编号	RW

12.4.16 PLU 控制寄存器 (AOS_PLUX_CR) (x=0~3)

AOS 中一共配备了 4 个可编程逻辑运算单元 (PLU)，每个单元有 4 个输入 (PLINA、PLINB、PLINC、PLIND) 和 1 个输出 (PLOUT)。

逻辑运算的输入不仅可以是 AOS 触发源，也可以是 PORT 输入。通过 PLINySEL (y=A、B、C、D) 位可以选择输入源或者固定值。通过 PLMODE 位可以选择需要的逻辑运算关系。

偏移地址 (Offset) : 0x200+0x004*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																PLINDSEL	PLINCSEL	PLINBSEL	PLINASEL	Reserved						PLMODE					
Access																	RW	RW	RW	RW							RW					

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:14	PLINDSEL	PLIND 输入源选择	0b00: PLIND 固定为 0 0b01: PLIND 固定为 1 0b10: PLIND 为 AOS_PLU_TRGSELD 选择的触发源 0b11: PLIND 为 PLUX_IND 端口输入值	RW
13:12	PLINCSEL	PLINC 输入源选择	0b00: PLINC 固定为 0 0b01: PLINC 固定为 1 0b10: PLINC 为 AOS_PLU_TRGSELC 选择的触发源 0b11: PLINC 为 PLUX_INC 端口输入值	RW
11:10	PLINBSEL	PLINB 输入源选择	0b00: PLINB 固定为 0 0b01: PLINB 固定为 1 0b10: PLINB 为 AOS_PLU_TRGSELB 选择的触发源 0b11: PLINB 为 PLUX_INB 端口输入值	RW
9:8	PLINASEL	PLINA 输入源选择	0b00: PLINA 固定为 0 0b01: PLINA 固定为 1 0b10: PLINA 为 AOS_PLU_TRGSELA 选择的触发源 0b11: PLINA 为 PLUX_INA 端口输入值	RW
7:2	Reserved	保留	-	-
1:0	PLMODE	逻辑运算选择	0b00: OR-OR, 即 (PLINA PLINB) (PLINC PLIND) 0b01: AND-OR, 即 (PLINA & PLINB) (PLINC & PLIND) 0b10: OR-XOR, 即 (PLINA PLINB) ^ (PLINC PLIND) 0b11: AND-AND, 即 (PLINA & PLINB) & (PLINC & PLIND)	RW



注意

当使用 PLU 运算结果作为 AOS 触发源时，请将 PLINySEL (y=A,B,C,D) 设为 0b00 或 0b10，且 PLMODE 设为 0b00。

12.4.17 PLU 触发源选择寄存器 A (AOS_PLUX_TRGSELA) (x=0~3)

当 PLU 的 PLINA 选择触发源输入时，请将对应触发事件的编号写入 AOS_PLUX_TRGSELA。

偏移地址 (Offset) : 0x210+0x010*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000001FF																															
Name	Reserved																								PLTRGSEL							
Access																									RW							

位/位域	标记	位名	功能描述	读写
31:9	Reserved	保留	-	-
8:0	PLTRGSEL	触发源选择	写入要选择的外设电路事件的编号	RW

12.4.18 PLU 触发源选择寄存器 B (AOS_PLUX_TRGSELB) (x=0~3)

当 PLU 的 PLINB 选择触发源输入时，请将对应触发事件的编号写入 AOS_PLUX_TRGSELB。

偏移地址 (Offset) : 0x214+0x010*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000001FF																															
Name	Reserved																								PLTRGSEL							
Access																									RW							

位/位域	标记	位名	功能描述	读写
31:9	Reserved	保留	-	-
8:0	PLTRGSEL	触发源选择	写入要选择的外设电路事件的编号	RW

12.4.19 PLU 触发源选择寄存器 C (AOS_PLUX_TRGSELC) (x=0~3)

当 PLU 的 PLINC 选择触发源输入时，请将对应触发事件的编号写入 AOS_PLUX_TRGSELC。

偏移地址 (Offset) : 0x218+0x010*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000001FF																															
Name	Reserved																				PLTRGSEL											
Access																					RW											

位/位域	标记	位名	功能描述	读写
31:9	Reserved	保留	-	-
8:0	PLTRGSEL	触发源选择	写入要选择的外设电路事件的编号	RW

12.4.20 PLU 触发源选择寄存器 D (AOS_PLUX_TRGSELD) (x=0~3)

当 PLU 的 PLIND 选择触发源输入时，请将对应触发事件的编号写入 AOS_PLUX_TRGSELD。

偏移地址 (Offset) : 0x21C+0x010*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000001FF																															
Name	Reserved																				PLTRGSEL											
Access																					RW											

位/位域	标记	位名	功能描述	读写
31:9	Reserved	保留	-	-
8:0	PLTRGSEL	触发源选择	写入要选择的外设电路事件的编号	RW

13 存储保护单元（MPU）

13.1 简介

MPU 可以提供对存储器的保护，通过阻止非授权的访问，可以提高系统的安全性。

本芯片内置了 1 个针对 CPU 的 MPU 单元，1 个针对 CPU 主栈指针的 MPU 单元，1 个针对 CPU 线程栈指针的 MPU 单元，2 个针对 DMA 的 MPU 单元和 1 个针对 IP 的 MPU 单元。

表 13-1 MPU 模块简介

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
DMPU	系统 DMA_1 MPU: SMPU1	系统 DMA_1 的存储保护单元 16 区域，8 区域系统 DMA 专用，8 区域所有 DMA 共用
	系统 DMA_2 MPU: SMPU2	系统 DMA_2 的存储保护单元 16 区域，8 区域系统 DMA 专用，8 区域所有 DMA 共用
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。

DMPU 包含 SMPU1/SMPU2，分别提供系统 DMA_1/系统 DMA_2 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。

13.2 功能说明

13.2.1 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。

13.2.2 权限设置

每个区域包括背景区域可以针对各个 DMA 独立设置允许读/禁止读和允许写/禁止写。如果不同区域之间发生地址重叠，则设定的禁止优先。

13.2.3 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

13.2.4 启动 MPU

MSPMPU/PSPMPU/SMPU1/SMPU2 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。

13.3 操作示例

13.3.1 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。

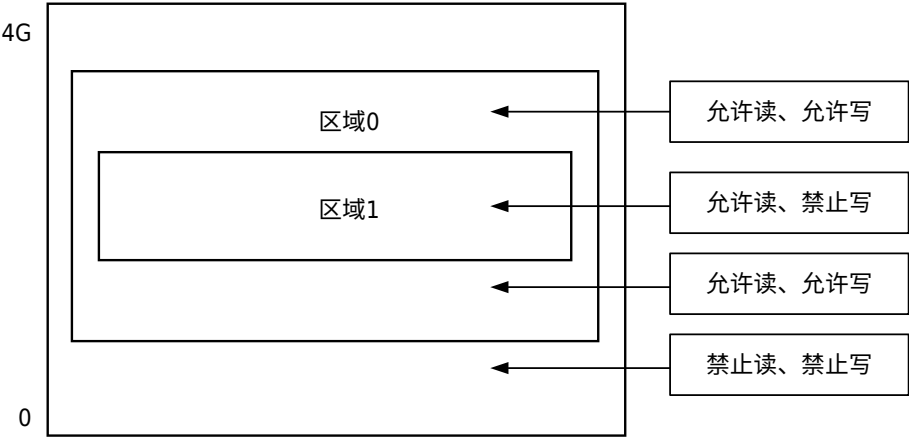


图 13-1 只允许部分空间访问示意图

13.3.2 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。

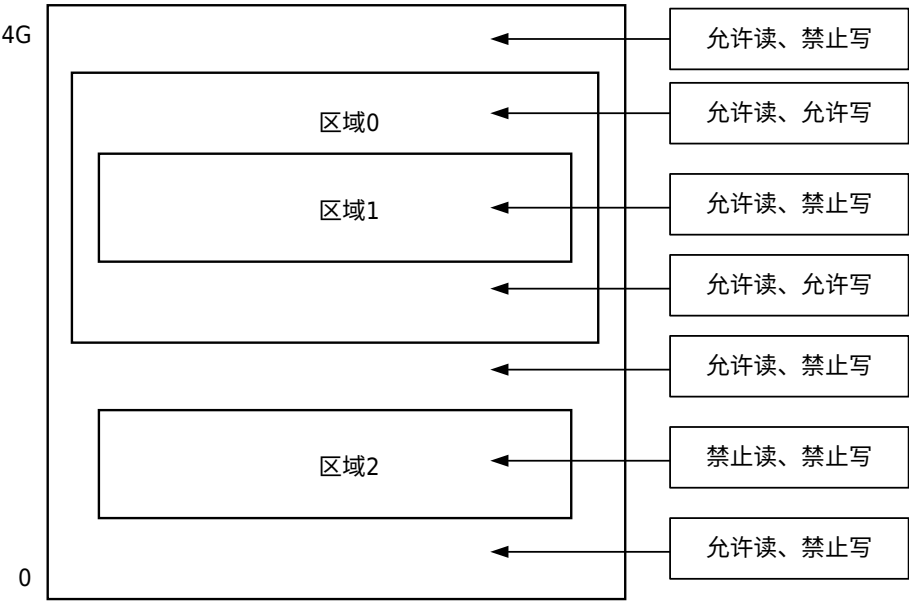


图 13-2 只禁止部分空间访问示意图

13.4 寄存器

13.4.1 寄存器总表

本模块的寄存器只能通过 CPU 来设置。

基地址：0x40050000

表 13-2 存储保护单元 (MPU) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00+0x04*n	MPU_RGDn	区域范围描述寄存器 (n=0~15) ⁽¹⁾	32
0x40	MPU_SR	状态标志寄存器	32
0x44	MPU_ECLR	标志清除寄存器	32
0x48	MPU_WP	写保护寄存器	32
0x4C	MPU_IPPR	IP 访问保护寄存器 ⁽¹⁾	32
0x50	MPU_MSPPBA	MSP 保护开始地址寄存器 ⁽¹⁾	32
0x54	MPU_MSPPCTL	MSP 保护控制寄存器 ⁽¹⁾	32
0x58	MPU_PSPPBA	PSP 保护开始地址寄存器 ⁽¹⁾	32
0x5C	MPU_PSPPCTL	PSP 保护控制寄存器 ⁽¹⁾	32
0x60	MPU_S1RGE	SMPU1 区域使能寄存器 ⁽¹⁾	32
0x64	MPU_S1RGWP	SMPU1 区域写权限寄存器 ⁽¹⁾	32
0x68	MPU_S1RGRP	SMPU1 区域读权限寄存器 ⁽¹⁾	32
0x6C	MPU_S1CR	SMPU1 控制寄存器 ⁽¹⁾	32
0x70	MPU_S2RGE	SMPU2 区域使能寄存器 ⁽¹⁾	32
0x74	MPU_S2RGWP	SMPU2 区域写权限寄存器 ⁽¹⁾	32
0x78	MPU_S2RGRP	SMPU2 区域读权限寄存器 ⁽¹⁾	32
0x7C	MPU_S2CR	SMPU2 控制寄存器 ⁽¹⁾	32

说明

1. 将 MPU_WP.MPUWE 置 1 时，允许写入。

13.4.2 区域范围描述寄存器 (MPU_RGDn) (n=0~15)

偏移地址 (Offset) : 0x00+0x04*n

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	MPURGnADDR[31:5]																												MPURGnSIZE			
Access	RW																												RW			

位/位域	标记	位名	功能描述	读写
31:5	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。 区域基地址总长 32bits，其中 MPURGnADDR[MPURGnSIZE:0]固定为 0。 例如： 区域大小为 32Bytes 时，区域基地址为 MPURGnADDR[31:5]+MPURGnADDR[4:0]（低 5 位 固定为 0）。 区域大小为 64Bytes 时，区域基地址为 MPURGnADDR[31:6]+MPURGnADDR[5:0]（低 6 位 固定为 0）。	RW
4:0	MPURGnSIZE	区域大小	设定区域 n 的大小： ● 0~3：禁止设置 ● 4：32Bytes ● 5：64Bytes ● ● 30：2GBytes ● 31：4GBytes	RW

13.4.3 状态标志寄存器 (MPU_SR)

Offset	Bit Position																																			
0x40	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reset	0x00000000																																			
Name	Reserved																												MSPEF		PSPEF		SMPU2EAF		SMPU1EAF	
Access																													RO	RO	RO	RO				

位/位域	标记	位名	功能描述	读写
31:4	Reserved	保留	-	-
3	MSPEF	MSP 错误标志	0b0: MSP 指针未超出设定范围 0b1: MSP 指针超出了设定范围	RO
2	PSPEF	PSP 错误标志	0b0: PSP 指针未超出设定范围 0b1: PSP 指针超出了设定范围	RO
1	SMPU2EAF	SMPU2 错误标志	0b0: 系统 DMA_2 未发生错误访问 0b1: 系统 DMA_2 发生了错误访问	RO
0	SMPU1EAF	SMPU1 错误标志	0b0: 系统 DMA_1 未发生错误访问 0b1: 系统 DMA_1 发生了错误访问	RO

 **注意**

对本寄存器的写操作会被忽略，清除错误标志请使用 MPU_ECLR。

13.4.4 标志清除寄存器 (MPU_ECLR)

Offset	Bit Position																																	
0x44	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved																										MSPECLR		PSPECLR		SMPU2ECLR		SMPU1ECLR	
Access																											RW		RW		RW		RW	

位/位域	标记	位名	功能描述	读写
31:4	Reserved	保留	-	-
3	MSPECLR	MSP 错误标志清除	写 1 将错误标志清 0，写 0 无效；读出恒为 0	RW
2	PSPECLR	PSP 错误标志清除	写 1 将错误标志清 0，写 0 无效；读出恒为 0	RW
1	SMPU2ECLR	SMPU2 错误标志清除	写 1 将错误标志清 0，写 0 无效；读出恒为 0	RW
0	SMPU1ECLR	SMPU1 错误标志清除	写 1 将错误标志清 0，写 0 无效；读出恒为 0	RW

13.4.5 写保护寄存器 (MPU_WP)

Offset	Bit Position																																
0x48	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																WKEY																MPUWE
Access																	WO																RW

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:1	WKEY	写入码	在写入 MPUWE 时，必须同时向 WKEY 写入 0x4B52	WO
0	MPUWE	MPU 寄存器写允许	0b0：MPU 地址寄存器/控制寄存器不允许写入 0b1：MPU 地址寄存器/控制寄存器允许写入	RW

 **说明**
向本寄存器写入 0x96A5 可以将 MPUWE 置 1；写入 0x96A4 可以将 MPUWE 清 0；禁止写入其他值修改 MPUWE。

13.4.6 IP 访问保护寄存器 (MPU_IPPR)



说明

1. 特权模式下不受本寄存器影响可以读写访问对象 IP。

Offset	Bit Position																															
0x4C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	BUSERRE	Reserved	MSTPWRP	MSTPRDP	SYSCWRP	SYSCRDP	INTCWRP	INTCRDP	SRAMCWRP	SRAMCRDP	DMPUWRP	DMPURDP	RTCWRP	RTCRDP	BKSRAMWRP	BKSRAMRDP	SWDTPWRP	SWDTRDP	WDTWRP	WDTRDP	Reserved	EFMWRP	EFMRDP	CRCWRP	CRCRDP	TRNGWRP	TRNGRDP	HASHWRP	HASHRDP	AESWRP	AESRDP	
Access	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31	BUSERRE	总线错误允许	0b0: 忽略对保护 IP 的访问 0b1: 发生对保护 IP 的访问时, 返回总线错误	RW
30	Reserved	保留	-	-
29	MSTPWRP	MSTP 写保护	0b0: 允许对 PWC_FCG0/1/2/3、PWC_FCG0PC 的写操作 0b1: 禁止对 PWC_FCG0/1/2/3、PWC_FCG0PC 的写操作	RW
28	MSTPRDP	MSTP 读保护	0b0: 允许对 PWC_FCG0/1/2/3、PWC_FCG0PC 的读操作 0b1: 禁止对 PWC_FCG0/1/2/3、PWC_FCG0PC 的读操作	RW
27	SYSCWRP	SYSC 写保护	0b0: 允许对 RMU/CMU/PWC 的写操作 0b1: 禁止对 RMU/CMU/PWC 的写操作 说明 BUSERRE 设置为 1 时, 对 RMU/PWC 的写操作不会发生总线错误。	RW
26	SYSCRDP	SYSC 读保护	0b0: 允许对 RMU/CMU/PWC 的读操作 0b1: 禁止对 RMU/CMU/PWC 的读操作 说明 BUSERRE 设置为 1 时, 对 RMU/PWC 的写操作不会发生总线错误。	RW
25	INTCWRP	INTC 写保护	0b0: 允许对 INTC 的写操作 0b1: 禁止对 INTC 的写操作	RW
24	INTCRDP	INTC 读保护	0b0: 允许对 INTC 的读操作 0b1: 禁止对 INTC 的读操作	RW
23	SRAMCWRP	SRAMC 写保护	0b0: 允许对 SRAMC 的写操作 0b1: 禁止对 SRAMC 的写操作	RW
22	SRAMCRDP	SRAMC 读保护	0b0: 允许对 SRAMC 的读操作 0b1: 禁止对 SRAMC 的读操作	RW

位/位域	标记	位名	功能描述	读写
21	DMPUWRP	DMPU 写保护	● 0b0: 允许对 DMPU 的写操作 ● 0b1: 禁止对 DMPU 的写操作	RW
20	DMPURDP	DMPU 读保护	● 0b0: 允许对 DMPU 的读操作 ● 0b1: 禁止对 DMPU 的读操作	RW
19	RTCWRP	RTC 写保护	● 0b0: 允许对 RTC 的写操作 ● 0b1: 禁止对 RTC 的写操作	RW
18	RTCRDP	RTC 读保护	● 0b0: 允许对 RTC 的读操作 ● 0b1: 禁止对 RTC 的读操作	RW
17	BKSRAMWRP	BKSRAM 写保护	● 0b0: 允许对 Ret SRAM 的写操作 ● 0b1: 禁止对 Ret SRAM 的写操作	RW
16	BKSRAMRDP	BKSRAM 读保护	● 0b0: 允许对 Ret SRAM 的读操作 ● 0b1: 禁止对 Ret SRAM 的读操作	RW
15	SWDTWRP	SWDT 写保护	● 0b0: 允许对 SWDT 的写操作 ● 0b1: 禁止对 SWDT 的写操作	RW
14	SWDTRDP	SWDT 读保护	● 0b0: 允许对 SWDT 的读操作 ● 0b1: 禁止对 SWDT 的读操作	RW
13	WDTWRP	WDT 写保护	● 0b0: 允许对 WDT 的写操作 ● 0b1: 禁止对 WDT 的写操作	RW
12	WDTRDP	WDT 读保护	● 0b0: 允许对 WDT 的读操作 ● 0b1: 禁止对 WDT 的读操作	RW
11:10	Reserved	保留	-	-
9	EFMWRP	EFM 写保护	● 0b0: 允许对 EFM 的写操作 ● 0b1: 禁止对 EFM 的写操作	RW
8	EFMRDP	EFM 读保护	● 0b0: 允许对 EFM 的读操作 ● 0b1: 禁止对 EFM 的读操作	RW
7	CRCWRP	CRC 写保护	● 0b0: 允许对 CRC 的写操作 ● 0b1: 禁止对 CRC 的写操作	RW
6	CRCRDP	CRC 读保护	● 0b0: 允许对 CRC 的读操作 ● 0b1: 禁止对 CRC 的读操作	RW
5	TRNGWRP	TRNG 写保护	● 0b0: 允许对 TRNG 的写操作 ● 0b1: 禁止对 TRNG 的写操作	RW
4	TRNGRDP	TRNG 读保护	● 0b0: 允许对 TRNG 的读操作 ● 0b1: 禁止对 TRNG 的读操作	RW
3	HASHWRP	HASH 写保护	● 0b0: 允许对 HASH 的写操作 ● 0b1: 禁止对 HASH 的写操作	RW
2	HASHRDP	HASH 读保护	● 0b0: 允许对 HASH 的读操作 ● 0b1: 禁止对 HASH 的读操作	RW
1	AESWRP	AES 写保护	● 0b0: 允许对 AES 的写操作 ● 0b1: 禁止对 AES 的写操作	RW
0	AESRDP	AES 读保护	● 0b0: 允许对 AES 的读操作 ● 0b1: 禁止对 AES 的读操作	RW

13.4.7 MSP 保护开始地址寄存器 (MPU_MSPPPBA)

Offset	Bit Position																															
0x50	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	MSPPBA[31:2]																															Reserved
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:2	MSPPPBA[31:2]	允许范围开始地址	设定 MSP 指针允许范围开始地址	RW
1:0	Reserved	保留	-	-

13.4.8 MSP 保护控制寄存器 (MPU_MSPPPCTL)

Offset	Bit Position																															
0x54	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000003																															
Name	MSPPE	MSPACT	Reserved														MSPPSIZE										Reserved					
Access	RW	RW															RW															

位/位域	标记	位名	功能描述	读写
31	MSPPE	MSP 保护使能	0b0: MSP 指针保护无效 0b1: MSP 指针保护有效	RW
30	MSPPPACT	MSP 保护动作选择	0b0: 不可屏蔽中断 0b1: 复位	RW
29:16	Reserved	保留	-	-
15:2	MSPPPSIZE	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
1:0	Reserved	保留	所有位读出为 1, 写入时必须写 1	-

13.4.9 PSP 保护开始地址寄存器 (MPU_PSPBPA)

Offset	Bit Position																																
0x58	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	PSPBPA[31:2]																																Reserved
Access	RW																																

位/位域	标记	位名	功能描述	读写
31:2	PSPBPA[31:2]	允许范围开始地址	设定 PSP 指针允许范围开始地址	RW
1:0	Reserved	保留	-	-

13.4.10 PSP 保护控制寄存器 (MPU_PSPCTL)

Offset	Bit Position																															
0x5C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000003																															
Name	PSPPE	PSPPACT	Reserved												PSPPSIZE												Reserved					
Access	RW	RW													RW																	

位/位域	标记	位名	功能描述	读写
31	PSPPE	PSP 保护使能	0b0: PSP 指针保护无效 0b1: PSP 指针保护有效	RW
30	PSPPACT	PSP 保护动作选择	0b0: 不可屏蔽中断 0b1: 复位	RW
29:16	Reserved	保留	-	-
15:2	PSPPSIZE	允许范围大小	设定 PSP 指针允许范围, 设定值 4Bytes~64KBytes 当 PSPPSIZE=0x0 时, PSP 指针允许范围为 4Bytes, 当 PSPPSIZE=0x3FFF 时, PSP 指针允许范围为 64KBytes	RW
1:0	Reserved	保留	所有位读出为 1, 写入时必须写 1	-

13.4.11 SMPU1 区域使能寄存器 (MPU_S1RGE)

Offset	Bit Position																															
0x60	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																S1RG15E	S1RG14E	S1RG13E	S1RG12E	S1RG11E	S1RG10E	S1RG9E	S1RG8E	S1RG7E	S1RG6E	S1RG5E	S1RG4E	S1RG3E	S1RG2E	S1RG1E	S1RG0E
Access																	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15	S1RG15E	SMPU1 区域 15 使能	0b0: SMPU1 的区域 15 无效 0b1: SMPU1 的区域 15 有效	RW
14	S1RG14E	SMPU1 区域 14 使能	0b0: SMPU1 的区域 14 无效 0b1: SMPU1 的区域 14 有效	RW
13	S1RG13E	SMPU1 区域 13 使能	0b0: SMPU1 的区域 13 无效 0b1: SMPU1 的区域 13 有效	RW
12	S1RG12E	SMPU1 区域 12 使能	0b0: SMPU1 的区域 12 无效 0b1: SMPU1 的区域 12 有效	RW
11	S1RG11E	SMPU1 区域 11 使能	0b0: SMPU1 的区域 11 无效 0b1: SMPU1 的区域 11 有效	RW
10	S1RG10E	SMPU1 区域 10 使能	0b0: SMPU1 的区域 10 无效 0b1: SMPU1 的区域 10 有效	RW
9	S1RG9E	SMPU1 区域 9 使能	0b0: SMPU1 的区域 9 无效 0b1: SMPU1 的区域 9 有效	RW
8	S1RG8E	SMPU1 区域 8 使能	0b0: SMPU1 的区域 8 无效 0b1: SMPU1 的区域 8 有效	RW
7	S1RG7E	SMPU1 区域 7 使能	0b0: SMPU1 的区域 7 无效 0b1: SMPU1 的区域 7 有效	RW
6	S1RG6E	SMPU1 区域 6 使能	0b0: SMPU1 的区域 6 无效 0b1: SMPU1 的区域 6 有效	RW
5	S1RG5E	SMPU1 区域 5 使能	0b0: SMPU1 的区域 5 无效 0b1: SMPU1 的区域 5 有效	RW
4	S1RG4E	SMPU1 区域 4 使能	0b0: SMPU1 的区域 4 无效 0b1: SMPU1 的区域 4 有效	RW
3	S1RG3E	SMPU1 区域 3 使能	0b0: SMPU1 的区域 3 无效 0b1: SMPU1 的区域 3 有效	RW
2	S1RG2E	SMPU1 区域 2 使能	0b0: SMPU1 的区域 2 无效 0b1: SMPU1 的区域 2 有效	RW
1	S1RG1E	SMPU1 区域 1 使能	0b0: SMPU1 的区域 1 无效 0b1: SMPU1 的区域 1 有效	RW

位/位域	标记	位名	功能描述	读写
0	S1RG0E	SMPU1 区域 0 使能	0b0: SMPU1 的区域 0 无效 0b1: SMPU1 的区域 0 有效	RW

13.4.12 SMPU1 区域写权限寄存器 (MPU_S1RGWP)

Offset	Bit Position																															
0x64	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved															S1RG15WP	S1RG14WP	S1RG13WP	S1RG12WP	S1RG11WP	S1RG10WP	S1RG9WP	S1RG8WP	S1RG7WP	S1RG6WP	S1RG5WP	S1RG4WP	S1RG3WP	S1RG2WP	S1RG1WP	S1RG0WP	
Access																RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15	S1RG15WP	SMPU1 区域 15 写权限	0b0: SMPU1 的区域 15 允许系统 DMA_1 写 0b1: SMPU1 的区域 15 禁止系统 DMA_1 写	RW
14	S1RG14WP	SMPU1 区域 14 写权限	0b0: SMPU1 的区域 14 允许系统 DMA_1 写 0b1: SMPU1 的区域 14 禁止系统 DMA_1 写	RW
13	S1RG13WP	SMPU1 区域 13 写权限	0b0: SMPU1 的区域 13 允许系统 DMA_1 写 0b1: SMPU1 的区域 13 禁止系统 DMA_1 写	RW
12	S1RG12WP	SMPU1 区域 12 写权限	0b0: SMPU1 的区域 12 允许系统 DMA_1 写 0b1: SMPU1 的区域 12 禁止系统 DMA_1 写	RW
11	S1RG11WP	SMPU1 区域 11 写权限	0b0: SMPU1 的区域 11 允许系统 DMA_1 写 0b1: SMPU1 的区域 11 禁止系统 DMA_1 写	RW
10	S1RG10WP	SMPU1 区域 10 写权限	0b0: SMPU1 的区域 10 允许系统 DMA_1 写 0b1: SMPU1 的区域 10 禁止系统 DMA_1 写	RW
9	S1RG9WP	SMPU1 区域 9 写权限	0b0: SMPU1 的区域 9 允许系统 DMA_1 写 0b1: SMPU1 的区域 9 禁止系统 DMA_1 写	RW
8	S1RG8WP	SMPU1 区域 8 写权限	0b0: SMPU1 的区域 8 允许系统 DMA_1 写 0b1: SMPU1 的区域 8 禁止系统 DMA_1 写	RW
7	S1RG7WP	SMPU1 区域 7 写权限	0b0: SMPU1 的区域 7 允许系统 DMA_1 写 0b1: SMPU1 的区域 7 禁止系统 DMA_1 写	RW
6	S1RG6WP	SMPU1 区域 6 写权限	0b0: SMPU1 的区域 6 允许系统 DMA_1 写 0b1: SMPU1 的区域 6 禁止系统 DMA_1 写	RW
5	S1RG5WP	SMPU1 区域 5 写权限	0b0: SMPU1 的区域 5 允许系统 DMA_1 写 0b1: SMPU1 的区域 5 禁止系统 DMA_1 写	RW
4	S1RG4WP	SMPU1 区域 4 写权限	0b0: SMPU1 的区域 4 允许系统 DMA_1 写 0b1: SMPU1 的区域 4 禁止系统 DMA_1 写	RW
3	S1RG3WP	SMPU1 区域 3 写权限	0b0: SMPU1 的区域 3 允许系统 DMA_1 写 0b1: SMPU1 的区域 3 禁止系统 DMA_1 写	RW

位/位域	标记	位名	功能描述	读写
2	S1RG2WP	SMPU1 区域 2 写权限	0b0: SMPU1 的区域 2 允许系统 DMA_1 写 0b1: SMPU1 的区域 2 禁止系统 DMA_1 写	RW
1	S1RG1WP	SMPU1 区域 1 写权限	0b0: SMPU1 的区域 1 允许系统 DMA_1 写 0b1: SMPU1 的区域 1 禁止系统 DMA_1 写	RW
0	S1RG0WP	SMPU1 区域 0 写权限	0b0: SMPU1 的区域 0 允许系统 DMA_1 写 0b1: SMPU1 的区域 0 禁止系统 DMA_1 写	RW

13.4.13 SMPU1 区域读权限寄存器 (MPU_S1RGRP)

Offset	Bit Position																															
0x68	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																SIRG15RP	SIRG14RP	SIRG13RP	SIRG12RP	SIRG11RP	SIRG10RP	SIRG9RP	SIRG8RP	SIRG7RP	SIRG6RP	SIRG5RP	SIRG4RP	SIRG3RP	SIRG2RP	SIRG1RP	SIRG0RP
Access																	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15	S1RG15RP	SMPU1 区域 15 读权限	0b0: SMPU1 的区域 15 允许系统 DMA_1 读 0b1: SMPU1 的区域 15 禁止系统 DMA_1 读	RW
14	S1RG14RP	SMPU1 区域 14 读权限	0b0: SMPU1 的区域 14 允许系统 DMA_1 读 0b1: SMPU1 的区域 14 禁止系统 DMA_1 读	RW
13	S1RG13RP	SMPU1 区域 13 读权限	0b0: SMPU1 的区域 13 允许系统 DMA_1 读 0b1: SMPU1 的区域 13 禁止系统 DMA_1 读	RW
12	S1RG12RP	SMPU1 区域 12 读权限	0b0: SMPU1 的区域 12 允许系统 DMA_1 读 0b1: SMPU1 的区域 12 禁止系统 DMA_1 读	RW
11	S1RG11RP	SMPU1 区域 11 读权限	0b0: SMPU1 的区域 11 允许系统 DMA_1 读 0b1: SMPU1 的区域 11 禁止系统 DMA_1 读	RW
10	S1RG10RP	SMPU1 区域 10 读权限	0b0: SMPU1 的区域 10 允许系统 DMA_1 读 0b1: SMPU1 的区域 10 禁止系统 DMA_1 读	RW
9	S1RG9RP	SMPU1 区域 9 读权限	0b0: SMPU1 的区域 9 允许系统 DMA_1 读 0b1: SMPU1 的区域 9 禁止系统 DMA_1 读	RW
8	S1RG8RP	SMPU1 区域 8 读权限	0b0: SMPU1 的区域 8 允许系统 DMA_1 读 0b1: SMPU1 的区域 8 禁止系统 DMA_1 读	RW
7	S1RG7RP	SMPU1 区域 7 读权限	0b0: SMPU1 的区域 7 允许系统 DMA_1 读 0b1: SMPU1 的区域 7 禁止系统 DMA_1 读	RW
6	S1RG6RP	SMPU1 区域 6 读权限	0b0: SMPU1 的区域 6 允许系统 DMA_1 读 0b1: SMPU1 的区域 6 禁止系统 DMA_1 读	RW
5	S1RG5RP	SMPU1 区域 5 读权限	0b0: SMPU1 的区域 5 允许系统 DMA_1 读 0b1: SMPU1 的区域 5 禁止系统 DMA_1 读	RW

位/位域	标记	位名	功能描述	读写
4	S1RG4RP	SMPU1 区域 4 读权限	0b0: SMPU1 的区域 4 允许系统 DMA_1 读 0b1: SMPU1 的区域 4 禁止系统 DMA_1 读	RW
3	S1RG3RP	SMPU1 区域 3 读权限	0b0: SMPU1 的区域 3 允许系统 DMA_1 读 0b1: SMPU1 的区域 3 禁止系统 DMA_1 读	RW
2	S1RG2RP	SMPU1 区域 2 读权限	0b0: SMPU1 的区域 2 允许系统 DMA_1 读 0b1: SMPU1 的区域 2 禁止系统 DMA_1 读	RW
1	S1RG1RP	SMPU1 区域 1 读权限	0b0: SMPU1 的区域 1 允许系统 DMA_1 读 0b1: SMPU1 的区域 1 禁止系统 DMA_1 读	RW
0	S1RG0RP	SMPU1 区域 0 读权限	0b0: SMPU1 的区域 0 允许系统 DMA_1 读 0b1: SMPU1 的区域 0 禁止系统 DMA_1 读	RW

13.4.14 SMPU1 控制寄存器 (MPU_S1CR)

Offset	Bit Position																															
0x6C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								SMPU1E	Reserved			SMPU1ACT	SMPU1BWP	SMPU1BRP	
Access																									RW				RW	RW	RW	

位/位域	标记	位名	功能描述	读写
31:8	Reserved	保留	-	-
7	SMPU1E	SMPU1 使能	0b0: SMPU1 无效 0b1: SMPU1 有效	RW
6:4	Reserved	保留	-	-
3:2	SMPU1ACT	SMPU1 动作选择	设定系统 DMA_1 发生被禁止的访问时的动作 0b00: 读访问读到 0, 写访问忽略 0b01: 读访问读到 0, 写访问忽略; 产生总线错误 0b10: 读访问读到 0, 写访问忽略; 产生不可屏蔽中断 0b11: 复位	RW
1	SMPU1BWP	SMPU1 背景写权限设置	0b0: SMPU1 背景空间允许系统 DMA_1 写 0b1: SMPU1 背景空间禁止系统 DMA_1 写	RW
0	SMPU1BRP	SMPU1 背景读权限设置	0b0: SMPU1 背景空间允许系统 DMA_1 读 0b1: SMPU1 背景空间禁止系统 DMA_1 读	RW



说明

多个区域设定重叠时, 优先级为: 设定的禁止>设定的允许。

13.4.15 SMPU2 区域使能寄存器 (MPU_S2RGE)

Offset	Bit Position																															
0x70	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																S2RG15E	S2RG14E	S2RG13E	S2RG12E	S2RG11E	S2RG10E	S2RG9E	S2RG8E	S2RG7E	S2RG6E	S2RG5E	S2RG4E	S2RG3E	S2RG2E	S2RG1E	S2RG0E
Access																	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15	S2RG15E	SMPU2 区域 15 使能	0b0: SMPU2 的区域 15 无效 0b1: SMPU2 的区域 15 有效	RW
14	S2RG14E	SMPU2 区域 14 使能	0b0: SMPU2 的区域 14 无效 0b1: SMPU2 的区域 14 有效	RW
13	S2RG13E	SMPU2 区域 13 使能	0b0: SMPU2 的区域 13 无效 0b1: SMPU2 的区域 13 有效	RW
12	S2RG12E	SMPU2 区域 12 使能	0b0: SMPU2 的区域 12 无效 0b1: SMPU2 的区域 12 有效	RW
11	S2RG11E	SMPU2 区域 11 使能	0b0: SMPU2 的区域 11 无效 0b1: SMPU2 的区域 11 有效	RW
10	S2RG10E	SMPU2 区域 10 使能	0b0: SMPU2 的区域 10 无效 0b1: SMPU2 的区域 10 有效	RW
9	S2RG9E	SMPU2 区域 9 使能	0b0: SMPU2 的区域 9 无效 0b1: SMPU2 的区域 9 有效	RW
8	S2RG8E	SMPU2 区域 8 使能	0b0: SMPU2 的区域 8 无效 0b1: SMPU2 的区域 8 有效	RW
7	S2RG7E	SMPU2 区域 7 使能	0b0: SMPU2 的区域 7 无效 0b1: SMPU2 的区域 7 有效	RW
6	S2RG6E	SMPU2 区域 6 使能	0b0: SMPU2 的区域 6 无效 0b1: SMPU2 的区域 6 有效	RW
5	S2RG5E	SMPU2 区域 5 使能	0b0: SMPU2 的区域 5 无效 0b1: SMPU2 的区域 5 有效	RW
4	S2RG4E	SMPU2 区域 4 使能	0b0: SMPU2 的区域 4 无效 0b1: SMPU2 的区域 4 有效	RW
3	S2RG3E	SMPU2 区域 3 使能	0b0: SMPU2 的区域 3 无效 0b1: SMPU2 的区域 3 有效	RW
2	S2RG2E	SMPU2 区域 2 使能	0b0: SMPU2 的区域 2 无效 0b1: SMPU2 的区域 2 有效	RW
1	S2RG1E	SMPU2 区域 1 使能	0b0: SMPU2 的区域 1 无效 0b1: SMPU2 的区域 1 有效	RW

位/位域	标记	位名	功能描述	读写
0	S2RG0E	SMPU2 区域 0 使能	0b0: SMPU2 的区域 0 无效 0b1: SMPU2 的区域 0 有效	RW

13.4.16 SMPU2 区域写权限寄存器 (MPU_S2RGWP)

Offset	Bit Position																															
0x74	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																S2RG15WP	S2RG14WP	S2RG13WP	S2RG12WP	S2RG11WP	S2RG10WP	S2RG9WP	S2RG8WP	S2RG7WP	S2RG6WP	S2RG5WP	S2RG4WP	S2RG3WP	S2RG2WP	S2RG1WP	S2RG0WP
Access																	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15	S2RG15WP	SMPU2 区域 15 写权限	0b0: SMPU2 的区域 15 允许系统 DMA_2 写 0b1: SMPU2 的区域 15 禁止系统 DMA_2 写	RW
14	S2RG14WP	SMPU2 区域 14 写权限	0b0: SMPU2 的区域 14 允许系统 DMA_2 写 0b1: SMPU2 的区域 14 禁止系统 DMA_2 写	RW
13	S2RG13WP	SMPU2 区域 13 写权限	0b0: SMPU2 的区域 13 允许系统 DMA_2 写 0b1: SMPU2 的区域 13 禁止系统 DMA_2 写	RW
12	S2RG12WP	SMPU2 区域 12 写权限	0b0: SMPU2 的区域 12 允许系统 DMA_2 写 0b1: SMPU2 的区域 12 禁止系统 DMA_2 写	RW
11	S2RG11WP	SMPU2 区域 11 写权限	0b0: SMPU2 的区域 11 允许系统 DMA_2 写 0b1: SMPU2 的区域 11 禁止系统 DMA_2 写	RW
10	S2RG10WP	SMPU2 区域 10 写权限	0b0: SMPU2 的区域 10 允许系统 DMA_2 写 0b1: SMPU2 的区域 10 禁止系统 DMA_2 写	RW
9	S2RG9WP	SMPU2 区域 9 写权限	0b0: SMPU2 的区域 9 允许系统 DMA_2 写 0b1: SMPU2 的区域 9 禁止系统 DMA_2 写	RW
8	S2RG8WP	SMPU2 区域 8 写权限	0b0: SMPU2 的区域 8 允许系统 DMA_2 写 0b1: SMPU2 的区域 8 禁止系统 DMA_2 写	RW
7	S2RG7WP	SMPU2 区域 7 写权限	0b0: SMPU2 的区域 7 允许系统 DMA_2 写 0b1: SMPU2 的区域 7 禁止系统 DMA_2 写	RW
6	S2RG6WP	SMPU2 区域 6 写权限	0b0: SMPU2 的区域 6 允许系统 DMA_2 写 0b1: SMPU2 的区域 6 禁止系统 DMA_2 写	RW
5	S2RG5WP	SMPU2 区域 5 写权限	0b0: SMPU2 的区域 5 允许系统 DMA_2 写 0b1: SMPU2 的区域 5 禁止系统 DMA_2 写	RW
4	S2RG4WP	SMPU2 区域 4 写权限	0b0: SMPU2 的区域 4 允许系统 DMA_2 写 0b1: SMPU2 的区域 4 禁止系统 DMA_2 写	RW
3	S2RG3WP	SMPU2 区域 3 写权限	0b0: SMPU2 的区域 3 允许系统 DMA_2 写 0b1: SMPU2 的区域 3 禁止系统 DMA_2 写	RW

位/位域	标记	位名	功能描述	读写
2	S2RG2WP	SMPU2 区域 2 写权限	0b0: SMPU2 的区域 2 允许系统 DMA_2 写 0b1: SMPU2 的区域 2 禁止系统 DMA_2 写	RW
1	S2RG1WP	SMPU2 区域 1 写权限	0b0: SMPU2 的区域 1 允许系统 DMA_2 写 0b1: SMPU2 的区域 1 禁止系统 DMA_2 写	RW
0	S2RG0WP	SMPU2 区域 0 写权限	0b0: SMPU2 的区域 0 允许系统 DMA_2 写 0b1: SMPU2 的区域 0 禁止系统 DMA_2 写	RW

13.4.17 SMPU2 区域读权限寄存器 (MPU_S2RGRP)

Offset	Bit Position																															
0x78	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																S2RG15RP	S2RG14RP	S2RG13RP	S2RG12RP	S2RG11RP	S2RG10RP	S2RG9RP	S2RG8RP	S2RG7RP	S2RG6RP	S2RG5RP	S2RG4RP	S2RG3RP	S2RG2RP	S2RG1RP	S2RG0RP
Access																	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15	S2RG15RP	SMPU2 区域 15 读权限	0b0: SMPU2 的区域 15 允许系统 DMA_2 读 0b1: SMPU2 的区域 15 禁止系统 DMA_2 读	RW
14	S2RG14RP	SMPU2 区域 14 读权限	0b0: SMPU2 的区域 14 允许系统 DMA_2 读 0b1: SMPU2 的区域 14 禁止系统 DMA_2 读	RW
13	S2RG13RP	SMPU2 区域 13 读权限	0b0: SMPU2 的区域 13 允许系统 DMA_2 读 0b1: SMPU2 的区域 13 禁止系统 DMA_2 读	RW
12	S2RG12RP	SMPU2 区域 12 读权限	0b0: SMPU2 的区域 12 允许系统 DMA_2 读 0b1: SMPU2 的区域 12 禁止系统 DMA_2 读	RW
11	S2RG11RP	SMPU2 区域 11 读权限	0b0: SMPU2 的区域 11 允许系统 DMA_2 读 0b1: SMPU2 的区域 11 禁止系统 DMA_2 读	RW
10	S2RG10RP	SMPU2 区域 10 读权限	0b0: SMPU2 的区域 10 允许系统 DMA_2 读 0b1: SMPU2 的区域 10 禁止系统 DMA_2 读	RW
9	S2RG9RP	SMPU2 区域 9 读权限	0b0: SMPU2 的区域 9 允许系统 DMA_2 读 0b1: SMPU2 的区域 9 禁止系统 DMA_2 读	RW
8	S2RG8RP	SMPU2 区域 8 读权限	0b0: SMPU2 的区域 8 允许系统 DMA_2 读 0b1: SMPU2 的区域 8 禁止系统 DMA_2 读	RW
7	S2RG7RP	SMPU2 区域 7 读权限	0b0: SMPU2 的区域 7 允许系统 DMA_2 读 0b1: SMPU2 的区域 7 禁止系统 DMA_2 读	RW
6	S2RG6RP	SMPU2 区域 6 读权限	0b0: SMPU2 的区域 6 允许系统 DMA_2 读 0b1: SMPU2 的区域 6 禁止系统 DMA_2 读	RW
5	S2RG5RP	SMPU2 区域 5 读权限	0b0: SMPU2 的区域 5 允许系统 DMA_2 读 0b1: SMPU2 的区域 5 禁止系统 DMA_2 读	RW

位/位域	标记	位名	功能描述	读写
4	S2RG4RP	SMPU2 区域 4 读权限	0b0: SMPU2 的区域 4 允许系统 DMA_2 读 0b1: SMPU2 的区域 4 禁止系统 DMA_2 读	RW
3	S2RG3RP	SMPU2 区域 3 读权限	0b0: SMPU2 的区域 3 允许系统 DMA_2 读 0b1: SMPU2 的区域 3 禁止系统 DMA_2 读	RW
2	S2RG2RP	SMPU2 区域 2 读权限	0b0: SMPU2 的区域 2 允许系统 DMA_2 读 0b1: SMPU2 的区域 2 禁止系统 DMA_2 读	RW
1	S2RG1RP	SMPU2 区域 1 读权限	0b0: SMPU2 的区域 1 允许系统 DMA_2 读 0b1: SMPU2 的区域 1 禁止系统 DMA_2 读	RW
0	S2RG0RP	SMPU2 区域 0 读权限	0b0: SMPU2 的区域 0 允许系统 DMA_2 读 0b1: SMPU2 的区域 0 禁止系统 DMA_2 读	RW

13.4.18 SMPU2 控制寄存器 (MPU_S2CR)

Offset	Bit Position																															
0x7C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								SMPU2E	Reserved	SMPU2ACT	SMPU2BWP	SMPU2BRP			
Access																									RW		RW	RW	RW			

位/位域	标记	位名	功能描述	读写
31:8	Reserved	保留	-	-
7	SMPU2E	SMPU2 使能	0b0: SMPU2 无效 0b1: SMPU2 有效	RW
6:4	Reserved	保留	-	-
3:2	SMPU2ACT	SMPU2 动作选择	设定系统 DMA_2 发生被禁止的访问时的动作 0b00: 读访问读到 0, 写访问忽略 0b01: 读访问读到 0, 写访问忽略; 产生总线错误 0b10: 读访问读到 0, 写访问忽略; 产生不可屏蔽中断 0b11: 复位	RW
1	SMPU2BWP	SMPU2 背景写权限设置	0b0: SMPU2 背景空间允许系统 DMA_2 写 0b1: SMPU2 背景空间禁止系统 DMA_2 写	RW
0	SMPU2BRP	SMPU2 背景读权限设置	0b0: SMPU2 背景空间允许系统 DMA_2 读 0b1: SMPU2 背景空间禁止系统 DMA_2 读	RW



说明

多个区域设定重叠时, 优先级为: 设定的禁止>设定的允许。

14 键盘扫描控制模块 (KEYSCAN)

14.1 简介

本产品搭载键盘扫描控制模块 (KEYSCAN) 1 个单元。KEYSCAN 模块支持键盘阵列 (行和列) 扫描, 列是由独立的扫描输出 KEYOUT_m ($m=0\sim7$) 驱动, 而行 KEYIN_n ($n=0\sim15$) 则作为 EIRQ_n ($n=0\sim15$) 输入被检测。本模块通过行扫描查询法实现按键识别功能。

14.2 主要特性

- EIRQ₀~EIRQ₁₅ 可以独立选择作为键盘阵列的行输入
- KEYOUT 可以通过寄存器选择
- 间隔一定时间依次输出低电平来扫描键盘阵列
- 扫描时间可以设置
- EIRQ 中断检出时停止扫描, 根据 SSR.INDEX 值和 EIRQ 中断标志 (INTC_EIFR.EIF) 来定位被按下的键

14.3 功能说明

14.3.1 功能框图

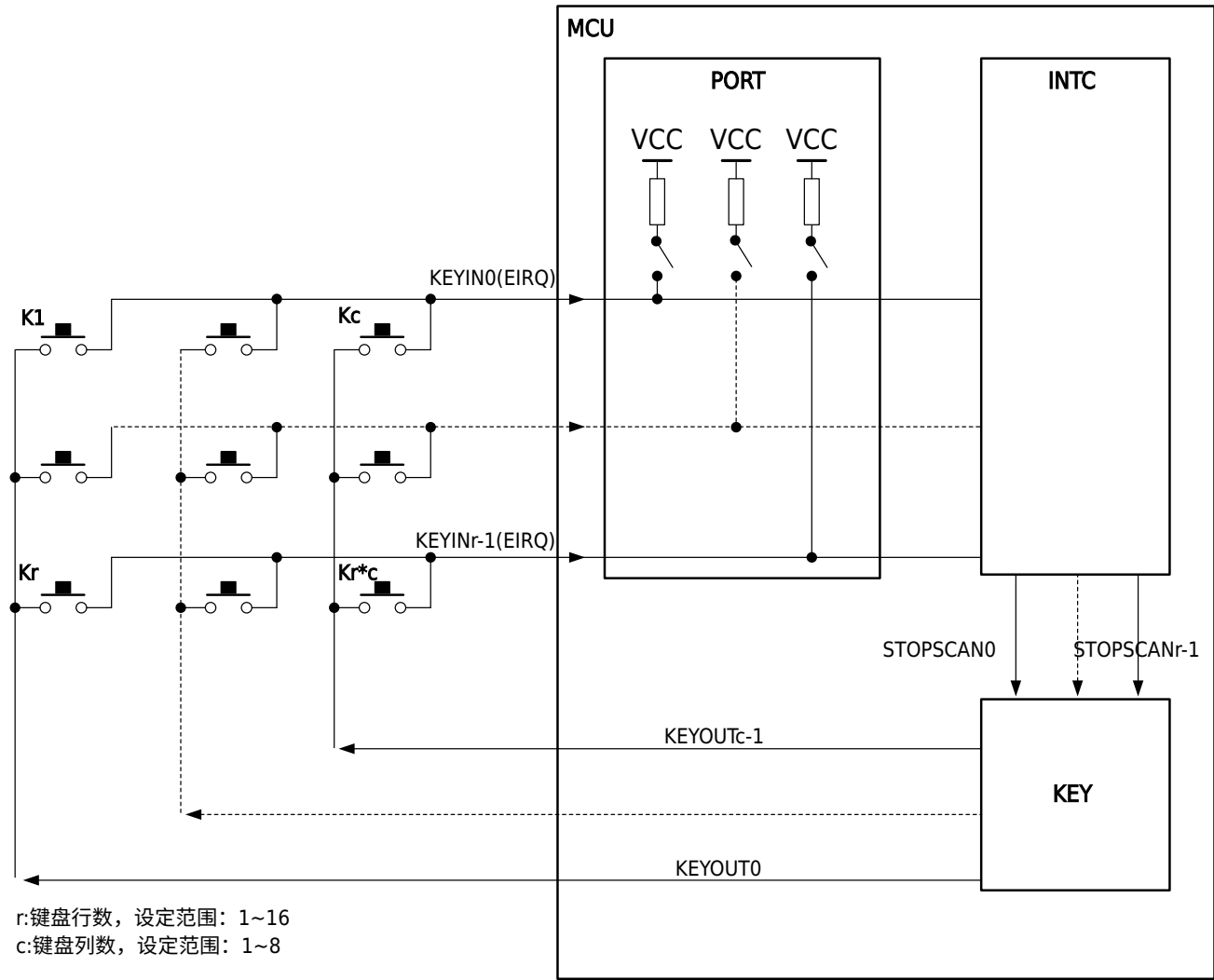


图 14-1 KEYSCAN 系统框图

表 14-1 KEYSCAN 管脚说明

管脚名	方向	功能描述
KEYINn (n=0~15)	输入	键盘行输入信号
KEYOUTm (m=0~7)	输出	键盘列输出信号

14.3.2 按键识别功能

当有键被按下时，键盘的行和列短接，行产生下降沿，从而产生 EIRQ 中断标志，通过比较中断标志位 (INTC_EIFR.EIF) 和 SSR.INDEX[2:0]的值来定位当前被按下的键。

通过寄存器 SCR.KEYINSEL[15:0]，KEYIN 可以从 EIRQ0~EIRQ15 中独立选择，通过寄存器 SCR.KEYOUTSEL[2:0]可以选择使用的 KEYOUT 管脚，从而可以灵活的选择键盘的行列数目，最大可以支持 16 行*8 列的键盘阵列。

14.3.3 键盘扫描功能

键盘扫描功能为：不断循环的给键盘阵列的列输出低电平，从而当有键被按下时，会产生对应的 EIRQ 中断标志。

当 KEYSCAN_SER.SEN 设置为 1 后，KEYOUT0 输出低电平，KEYOUT1~KEYOUTn（n 通过 KEYSCAN_SCR.KEYOUTSEL[2:0] 设定）为 Hi-Z，经过 KEYSCAN_SCR.T_LLEVEL[4:0] 设定的时间后，KEYOUT0~KEYOUTn 管脚全部为 Hi-Z，经过 KEYSCAN_SCR.T_HIZ[2:0] 设定的时间后，KEYOUT1 输出低电平，其余 KEYOUT 管脚为 Hi-Z，依次类推循环。有键被按下并产生 EIRQ 中断标志时，键盘扫描功能停止，对应的中断标志清零后，扫描自动重启。

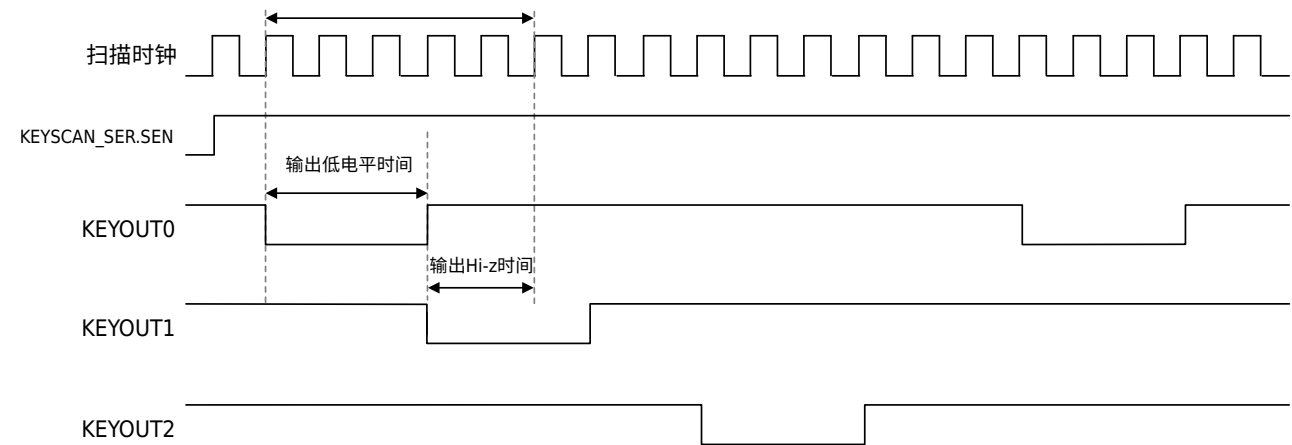


图 14-2 键盘扫描功能示意图

14.4 寄存器

14.4.1 寄存器总表

基地址：0x4001F400

表 14-2 键盘扫描控制模块 (KEYSCAN) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	KEYSCAN_SCR	扫描控制寄存器	32
0x04	KEYSCAN_SER	扫描使能寄存器	32
0x08	KEYSCAN_SSR	扫描状态寄存器	32

14.4.2 扫描控制寄存器 (KEYSCAN_SCR)



说明

该寄存器只在 KEYCAN_SER.SEN=0 时设定有效。

Offset	Bit Position																															
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	T_HIZ		T_LLEVEL				Reserved		CKSEL		Reserved		KEYOUTSEL		KEYINSEL																	
Access	RW		RW						RW				RW		RW																	

位/位域	标记	位名	功能描述	读写
31:29	T_HIZ	输出 Hi-Z 时间	KEYOUT 输出 Hi-Z 时间 (扫描时钟数) 0b000: 4 个扫描时钟数 0b001: 8 个扫描时钟数 0b010: 16 个扫描时钟数 0b011: 32 个扫描时钟数 0b100: 64 个扫描时钟数 0b101: 256 个扫描时钟数 0b110: 512 个扫描时钟数 0b111: 1024 个扫描时钟数 扫描周期 = 输出低电平时间 + 输出 Hi-Z 时间	RW
28:24	T_LLEVEL	输出低电平时间	KEYOUT 输出低电平时间 (扫描时钟数) 输出低电平时间 = 2^{T_LLEVEL} 个扫描时钟数 扫描周期 = 输出低电平时间 + 输出 Hi-Z 时间  说明 0b00000 和 0b00001 禁止设定, 最大可以设定值为 0b11000。	RW
23:22	Reserved	保留	-	-
21:20	CKSEL	扫描时钟源选择位	0b00: 系统时钟 HCLK 0b01: 内部低速振荡器 LRC 0b10: 外部低速振荡器 XTAL32 0b11: 禁止设置	RW
19	Reserved	保留	-	-
18:16	KEYOUTSEL	KEYOUT 输出选择位	0b000: 禁止设置 0b001: KEYOUT0~KEYOUT1 0b010: KEYOUT0~KEYOUT2 0b011: KEYOUT0~KEYOUT3 0b100: KEYOUT0~KEYOUT4 0b101: KEYOUT0~KEYOUT5 0b110: KEYOUT0~KEYOUT6 0b111: KEYOUT0~KEYOUT7	RW

位/位域	标记	位名	功能描述	读写
15:0	KEYINSEL	行输入选择位	对应 KEYINSEL15~KEYINSEL0，选择的行作为键盘阵列的行使用，并作为 EIRQn（n=15~0）被检测： ● 0b0：不用作键盘阵列的行使用 ● 0b1：用作键盘阵列的行使用	RW

14.4.3 扫描使能寄存器 (KEYSCAN_SER)

Offset	Bit Position																															
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															SEN
Access																																RW

位/位域	标记	位名	功能描述	读写
31:1	Reserved	保留	-	-
0	SEN	扫描使能位	● 0b0：扫描禁止 ● 0b1：扫描使能	RW

14.4.4 扫描状态寄存器 (KEYSCAN_SSR)

说明

该寄存器只在 KEYCAN_SER.SEN=1 时读取出来的数据才有意义。

Offset	Bit Position																															
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															RO INDEX
Access																																

位/位域	标记	位名	功能描述	读写
31:3	Reserved	保留	-	-
2:0	INDEX	当前工作的 SCAN 管脚索引位	● 0b000：当前工作的 SCAN 管脚为 KEYOUT0 ● 0b001：当前工作的 SCAN 管脚为 KEYOUT1 ● 0b010：当前工作的 SCAN 管脚为 KEYOUT2 ● 0b011：当前工作的 SCAN 管脚为 KEYOUT3 ● 0b100：当前工作的 SCAN 管脚为 KEYOUT4 ● 0b101：当前工作的 SCAN 管脚为 KEYOUT5 ● 0b110：当前工作的 SCAN 管脚为 KEYOUT6 ● 0b111：当前工作的 SCAN 管脚为 KEYOUT7	RO

14.5 注意事项

本模块驱动键盘列，而键盘行检测则由中断控制模块 (INTC) 的外部 EIRQ 功能实现，EIRQ 需要选择下降沿检测，并开启数字滤波功能，设定合适的滤波时间。

STOP 模式下如果使用此功能，则需设定好扫描相关参数后，选择内部低速振荡器 LRC 或者外部低速振荡器 XTAL32 时钟作为扫描时钟。

如果使用芯片内部上拉电阻，则请参见《数据手册》中“电气特性-I/O 端口特性”章节相关描述，选择合适的扫描时间以及滤波时间。

15 内部时钟校准器 (CTC)

15.1 简介

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 的频率可以得到一个精准的 HRC 时钟。

15.2 主要特性

- 三个外部参考时钟源: XTAL、XTAL32、CTCREF
- 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 位校准偏差值和 6 位校准值
- 用于提示校准失败的错误中断

15.3 功能说明

15.3.1 功能框图

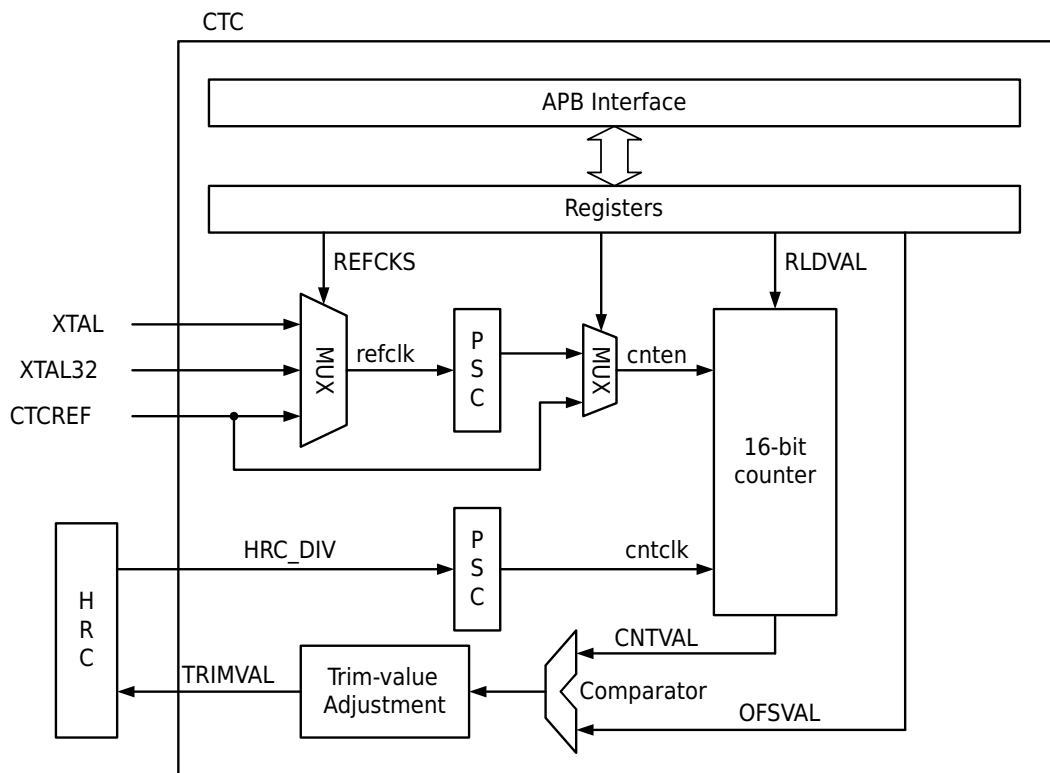


图 15-1 CTC 功能框图

15.3.2 参考时钟

CTC 有以下三个参考时钟, 通过 CTC_CR1.REFCKS 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.REFPSC 和 HRC_PSC 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:

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

其中：

- 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。

CTC 单次校准时只能选择从 CTCREF 输入基准信号，并且通过 CTC_CR1.REFEDG 位来选择基准信号的有效边沿。此时 CTC_CR1.REFPSC 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。

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

其中：

- 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。

下面两个表分别列举了 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。



说明

- “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 15-1 HRC 目标频率为 20MHz 时的测量误差

F_{ref}		1kHz							
PSC_{ref}		2	4	8	16	64	256	1024	4096
PSC_{hrc}	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%
F_{ref}		32.768kHz							
PSC_{ref}		2	4	8	16	64	256	1024	4096
PSC_{hrc}	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

PSC _{hrc}	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%
F _{ref}		100kHz							
PSC _{ref}		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%
F _{ref}		1MHz							
PSC _{ref}		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
F _{ref}		8MHz							
PSC _{ref}		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%

PSC _{hrc}	32	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.559%
	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
F _{ref}		20MHz							
PSC _{ref}		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

表 15-2 HRC 目标频率为 16MHz 时的测量误差

F _{ref}		1kHz							
PSC _{ref}		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
	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%
F _{ref}		32.768kHz							
PSC _{ref}		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

PSC _{hrc}	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%
F _{ref}		100kHz							
PSC _{ref}		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%
F _{ref}		1MHz							
PSC _{ref}		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
F _{ref}		8MHz							
PSC _{ref}		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
F _{ref}		20MHz							
PSC _{ref}		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

15.3.3 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。

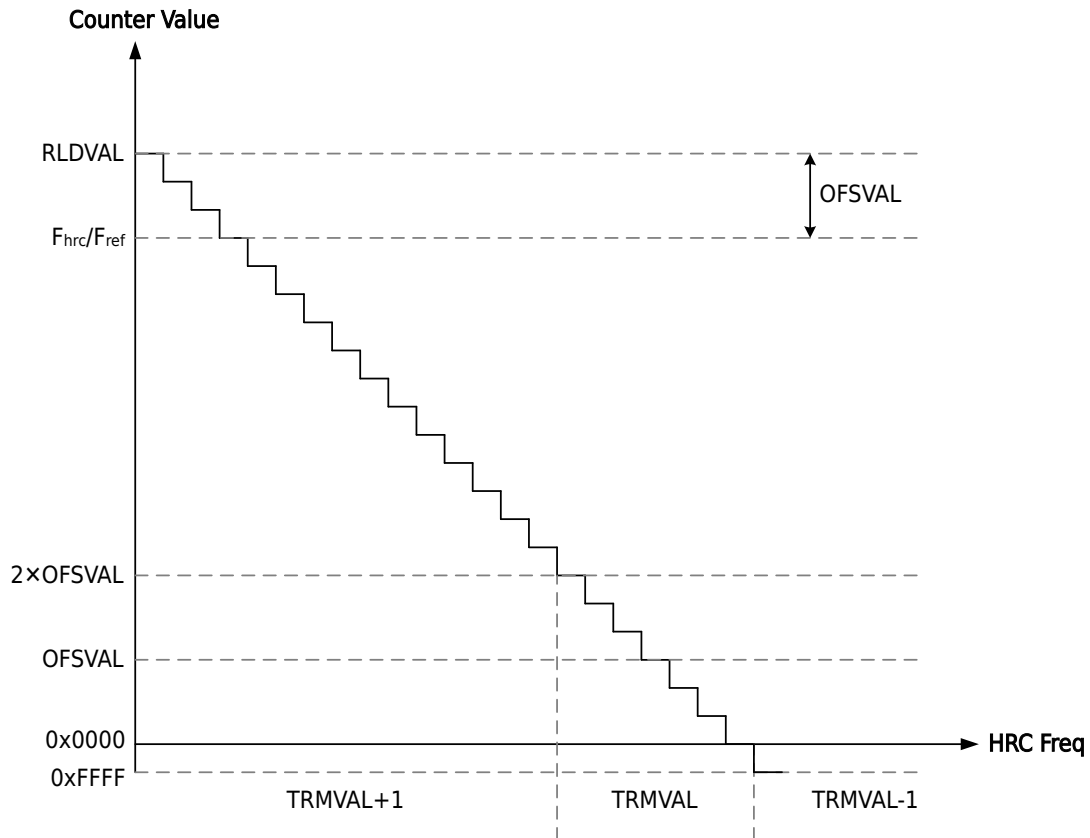


图 15-2 CTC 校准示意图

16 位校准计数器停止计数后，HRC 频率校准开始。由于在设定校准计数重载值 RLDVAL 时加上了校准目标偏差值 OFSVAL，因此如果 16 位校准计数器的值大于 $2 \times \text{OFSVAL}$ 且没有发生下溢，说明当前频率比期望频率慢，需要增大 CTC_CR1 寄存器中 TRMVAL 的值来提高频率。反之，如果 16 位校准计数器的值发生下溢，则说明当前频率比期望频率快，需要减小 TRMVAL 的值以降低频率。而如果 16 位校准计数器的值小于 $2 \times \text{OFSVAL}$ ，说明当前频率处于期望频率范围内，就不需要调整 TRMVAL 的值。

- Counter > ($\text{OFSVAL} \times 2$) 时停止计数，TRMVAL 的值自动加 1。
- $0 \leq \text{Counter} \leq (\text{OFSVAL} \times 2)$ 时停止计数，TRMVAL 的值不变，CTC_STR.TRIMOK 位自动变为 1。
- Counter = 0xFFFF，CTC_CR1 寄存器中的 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。

15.3.4 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 寄存器中的 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。具体设定方法请参见《中断控制器 (INTC)》章节。

15.4 操作示例

15.4.1 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%

XTAL32 设为 16 分频，HRC 设为不分频，经查表 15-2 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 = 0x1EAB0027，其中：

- $\text{OFSVAL} = (16 \div 0.032768) \times 16 \times 0.5\% \approx 39$ (0x27)

- $RLDVAL = (16 \div 0.032768) \times 16 + 39 \approx 7851$ (0x1EAB)
2. 设定寄存器 CTC_CR1 = 0x000000E3, 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 16 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, 继续等待一段时间后继续本步骤。
 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。

**说明**

频率校准是通过 CTC_CR1.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.CTCBSY 位为 0 后再进行修改。

15.5 寄存器

15.5.1 寄存器总表

基地址：0x40049C00

表 15-3 内部时钟校准器 (CTC) 寄存器偏移地址

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

15.5.2 时钟校准控制寄存器 1 (CTC_CR1)

Offset	Bit Position																																		
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Reset	0x00000000																																		
Name	Reserved										TRMVAL						Reserved		REFEDG		Reserved		HRCPSC			CTCEN		ERRIE		REFCKS		Reserved		REFPSC	
Access											RW								RW				RW			RW		RW		RW				RW	

位/位域	标记	位名	功能描述	读写
31:22	Reserved	保留	-	-
21:16	TRMVAL	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
15:14	Reserved	保留	-	-
13:12	REFEDG	参考信号边沿选择	用于 REFCKS=0b01 时选择 CTCREF 的有效边沿。 ● 0b00: 选择 CTCREF 的上升沿到上升沿进行校准 ● 0b01: 选择 CTCREF 的上升沿到下降沿进行校准 ● 0b10: 选择 CTCREF 的下降沿到上升沿进行校准 ● 0b11: 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
11	Reserved	保留	-	-
10:8	HRCPSC	HRC 分频数选择	● 0b000: 不分频 ● 0b001: 2 分频 ● 0b010: 4 分频 ● 0b011: 8 分频 ● 0b100: 32 分频 ● 0b101: 128 分频 ● 0b110: 512 分频 ● 0b111: 2048 分频	RW

位/位域	标记	位名	功能描述	读写
10:8	HRCPSC	HRC 分频数选择	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
7	CTCEN	校准使能	用于启动频率校准。 ● 0b0: 停止频率校准 ● 0b1: 启动频率校准  说明 ● 置位条件: 软件置位 ● 清零条件: ▶ 软件清零 ▶ 校准发生上溢或者下溢时自动清零 ▶ 单次校准结束时自动清零	RW
6	ERRIE	错误中断允许	用于选择发生校准上溢或者下溢时是否产生错误中断。 ● 0b0: 禁止发生错误中断 ● 0b1: 允许发生错误中断	RW
5:4	REFCKS	参考时钟选择	用于为频率校准选择参考时钟源。 ● 0b00: 选择 CTCREF 作为参考时钟 (连续校准) ● 0b01: 选择 CTCREF 作为参考信号 (单次校准) ● 0b10: 选择 XTAL32 作为参考时钟 (连续校准) ● 0b11: 选择 XTAL 作为参考时钟 (连续校准) CTCEN 位为 1 时禁止修改 REFCKS 的值。如需修改请将 CTCEN 位写 0 并确认 CTC_STR.CTCBSY 位为 0 后再修改。	RW
3	<i>Reserved</i>	<i>保留</i>	-	-
2:0	REFPSC	参考时钟分频数选择	REFCKS=0b00/0b10/0b11 时选择参考时钟的分频数。 ● 0b000: 2 分频 ● 0b001: 4 分频 ● 0b010: 8 分频 ● 0b011: 16 分频 ● 0b100: 64 分频 ● 0b101: 256 分频 ● 0b110: 1024 分频 ● 0b111: 4096 分频 CTCEN 位为 1 时禁止修改 REFPSC 的值。如需修改请将 CTCEN 位写 0 并确认 CTC_STR.CTCBSY 位为 0 后再修改。	RW

15.5.3 时钟校准控制寄存器 2 (CTC_CR2)

Offset	Bit Position																															
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	RLDVAL																Reserved								OFSVAL							
Access	RW																								RW							

位/位域	标记	位名	功能描述	读写
31:16	RLDVAL	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：$RLDVAL = (F_{hrc} \div PSC_{hrc}) \div (F_{ref} \div PSC_{ref}) + OFSVAL$ 单次校准：$RLDVAL = ((F_{hrc} \div PSC_{hrc}) \times T_{trim}) + OFSVAL$ ● F_{hrc}: HRC 目标频率; ● F_{ref}: 参考时钟频率; ● T_{trim}: 单次校准周期; ● OFSVAL: 校准计数偏差值; ● PSC_{hrc}: HRC 分频数; ● PSC_{ref}: 参考时钟分频数。  说明 CTCEN 位为 1 时禁止修改 RLDVAL 的值。如需修改请将 CTCEN 位写 0 并确认 CTC_STR.CTCBSY 位为 0 后再修改。	RW
15:8	Reserved	保留	-	-
7:0	OFSVAL	校准计数偏差值	OFSVAL 设定的是 HRC 在校准时允许的偏差值。 连续校准：$OFSVAL = (F_{hrc} \div PSC_{hrc}) \div (F_{ref} \div PSC_{ref}) \times TA$ 单次校准：$OFSVAL = ((F_{hrc} \div PSC_{hrc}) \times T_{trim}) \times TA$ ● F_{hrc}: HRC 目标频率; ● F_{ref}: 参考时钟频率; ● T_{trim}: 单次校准周期; ● TA: 校准精度, 例如 0.5%; ● PSC_{hrc}: HRC 分频数; ● PSC_{ref}: 参考时钟分频数。  说明 CTCEN 位为 1 时禁止修改 OFSVAL 的值。如需修改请将 CTCEN 位写 0 并确认 CTC_STR.CTCBSY 位为 0 后再修改。	RW

15.5.4 时钟校准状态寄存器 (CTC_STR)

Offset	Bit Position																																			
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reset	0x00000000																																			
Name	Reserved																																CTCBSY	TRMUDF	TRMOVF	TRIMOK
Access																																	RO	RO	RO	RO

位/位域	标记	位名	功能描述	读写
31:4	Reserved	保留	-	-
3	CTCBSY	CTC 状态标志	0b0: CTC 处于停止状态 0b1: CTC 处于工作状态 CTC_CR1.CTCEN=1 时置位，在软件将 CTCEN 清零或者发生校准错误时清零。	RO
2	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。 0b0: 未发生校准下溢 0b1: 发生了校准下溢 在 CTC_CR1.CTCEN 写 1 时清零。	RO
1	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。 0b0: 未发生校准上溢 0b1: 发生了校准上溢 在 CTC_CR1.CTCEN 写 1 时清零。	RO
0	TRIMOK	校准成功标志	0b0: HRC 频率未校准到目标范围 0b1: HRC 频率已校准到目标范围 在 CTC_CR1.CTCEN 写 1 时清零。	RO

15.5.5 时钟校准计数器 (CTC_CNT)

Offset	Bit Position															
0x0C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	CTCCNT															
Access	RO															

位/位域	标记	位名	功能描述	读写
15:0	CTCCNT	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。	RO

16 DMA 控制器 (DMA)

16.1 简介

DMA 用于在存储器和外围功能模块之间传送数据，能够在 CPU 不参与的情况下，实现存储器之间、存储器和外围功能模块之间以及外围功能模块之间的数据交换。

16.2 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 2 个 DMA 控制单元，共 12 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 8-bit、16-bit 或 32-bit
- 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、重复跳转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，可作为其他外围模块的触发源
- 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
- 支持软件触发启动传输和软件触发通道重置
- 不使用时可设置进入模块停止状态以降低功耗
- DMA 访问时 AHB 总线中的 HPROT 值可通过寄存器设置

16.3 功能说明

16.3.1 DMA 框图

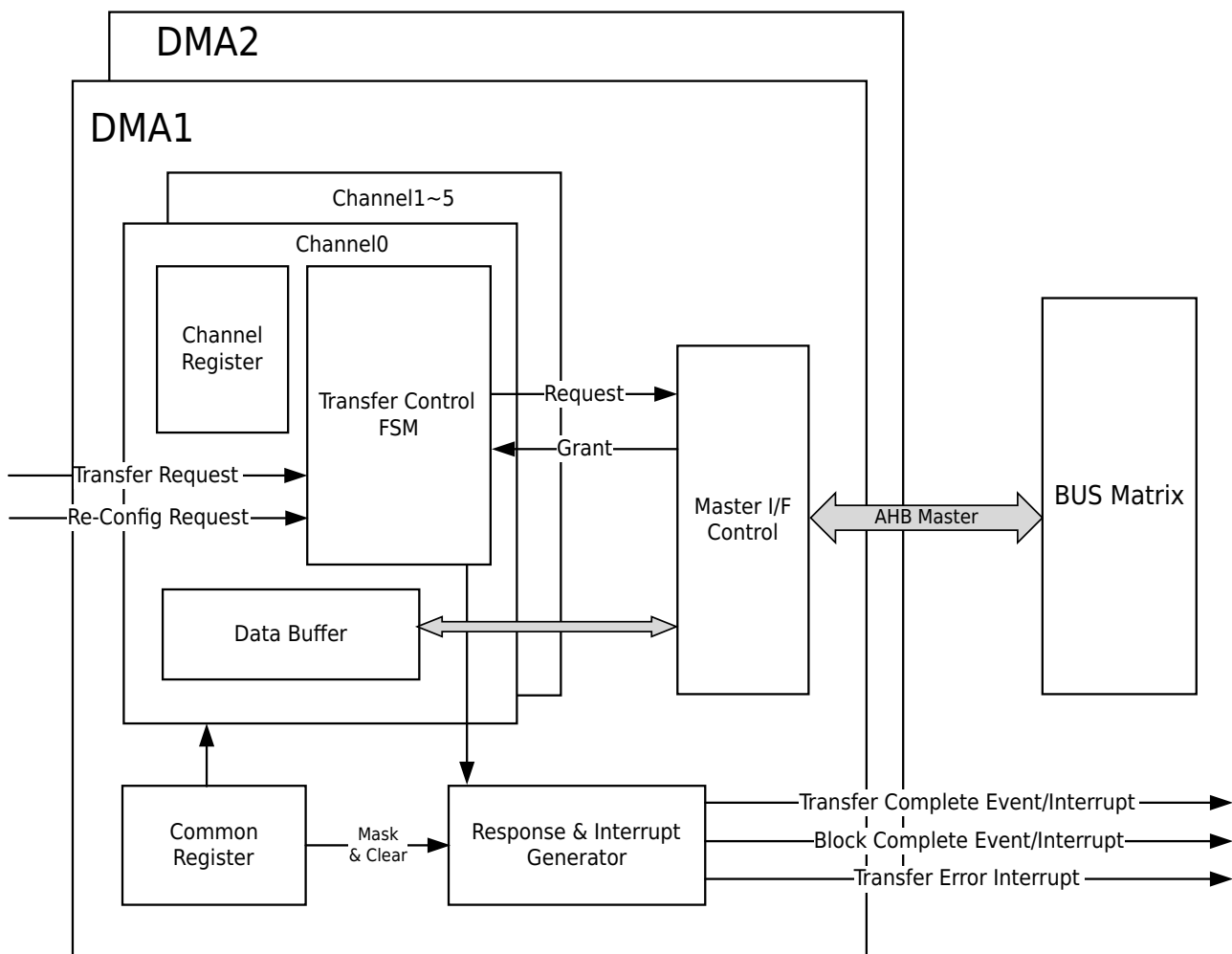


图 16-1 DMA 结构图

16.3.2 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要写 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。

16.3.3 通道选择和通道优先级

每个 DMA 控制单元中包含 6 个通道，每个通道可以独立配置传输功能。

通道优先级顺序为：通道 0>通道 1>...>通道 5。

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，高优先级通道需等当前通道传输完成后才会启动。

16.3.4 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=0~5) 进行配置，各通道独立配置启动请求源。DMA 可以通过软件写软件触发寄存器 DMA_SWREQ.SWREQ 启动 DMA 并开始

传输。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。

16.3.5 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 8-bit、16-bit 或 32-bit。

16.3.6 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、重复跳转或者不连续跳转。

- **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 8-bit 时，地址将每次增加/减少 1，为 16-bit 时每次增加/减少 2，为 32-bit 时每次增加/减少 4。
- **重复跳转：**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。地址重复跳转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。
- **不连续跳转：**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。当地址重复跳转与不连续跳转的条件同时满足时，执行地址重复跳转。

16.3.7 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.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。



说明

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。

16.3.8 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.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.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。

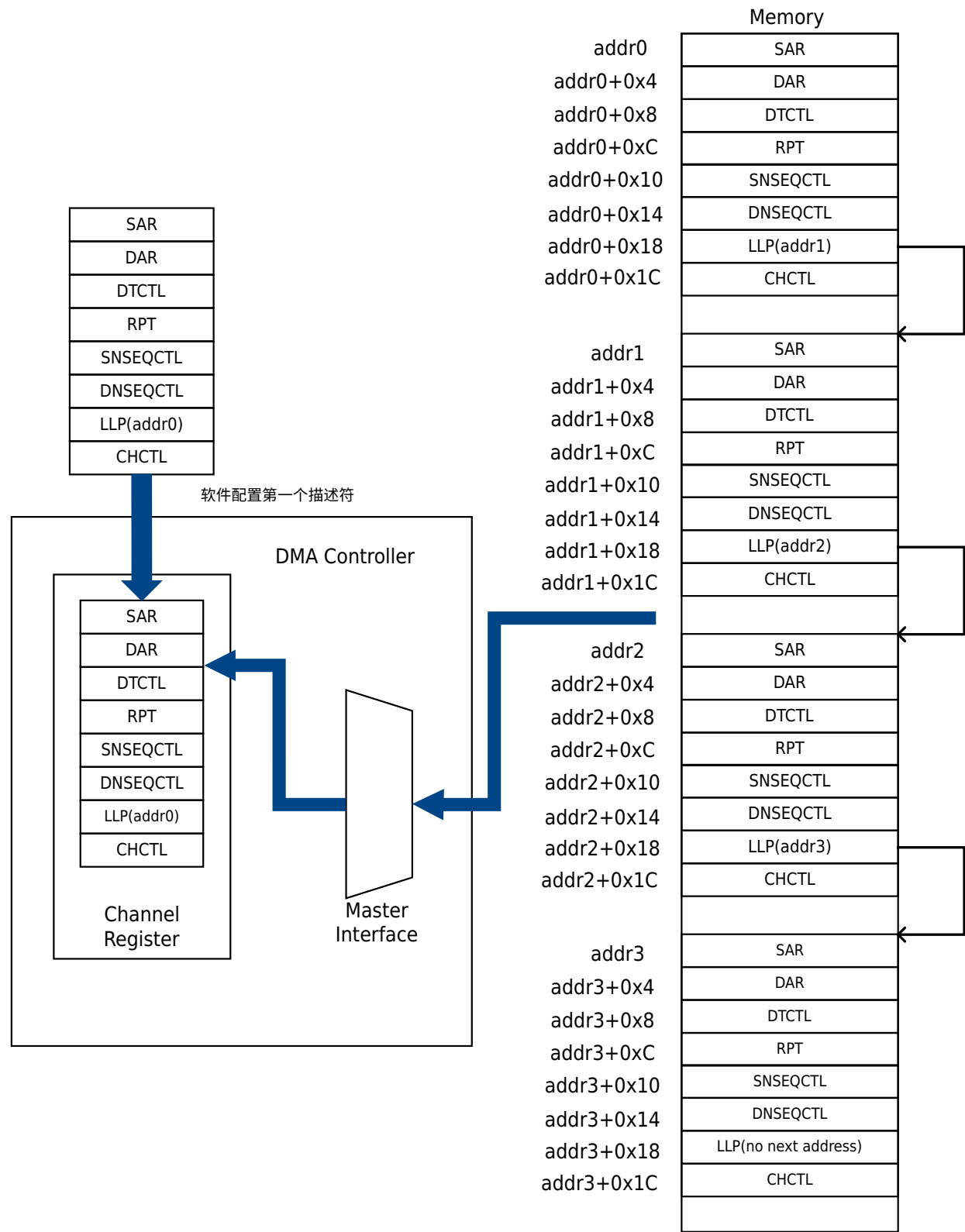


图 16-2 连锁传输示意图

16.3.9 不连续地址传输

使用不连续地址传输能够实现源地址和目标地址在传输完一定量的数据后按照一定的偏移量跳转。以源地址不连续传输为例：首先根据需要将通道控制寄存器 DMA_CHCTLx.SNSEQEN 置 1 使不连续地址传输有效，跳

转的方向按照 DMA_CHCTLx.SINC 的设定向前或向后跳转，然后配置寄存器 DMA_SNSEQCTLx 设置具体的跳转方式。目标地址的控制与源地址相似，对应控制寄存器分别为 DMA_CHCTLx.DNSEQEN、DMA_CHCTLx.DINC、DMA_DNSEQCTLx。传输过程按照下图方式进行。

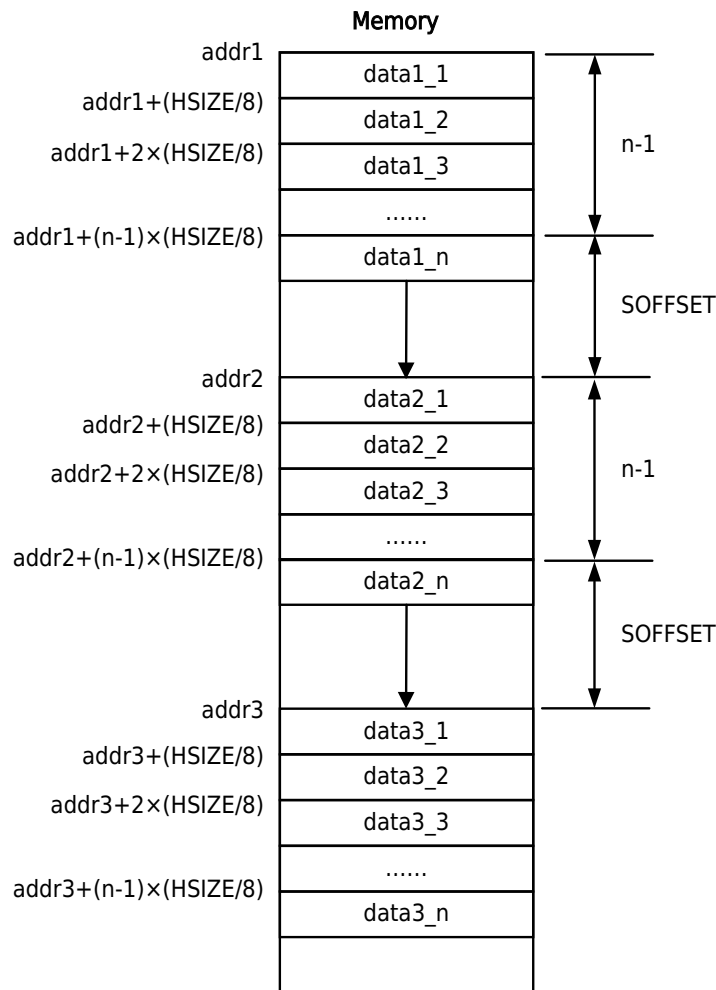


图 16-3 不连续地址传输示意图（以源地址为例）

$n = \text{SNSCNT}$
 $\text{addr2} = \text{addr1} + (n-1) \times (\text{HSIZE}/8) + \text{SOFFSET} \times (\text{HSIZE}/8)$
 $\text{addr3} = \text{addr2} + (n-1) \times (\text{HSIZE}/8) + \text{SOFFSET} \times (\text{HSIZE}/8)$
其中 HSIZE 为数据宽度 (bit)，可以是字节 (8bits)，半字 (16bits) 和字 (32bits)。

16.3.10 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，或通过软件写软件触发寄存器 DMA_SWREQ.SWRCFGREQ 触发通道重置功能。当选择的重置请求源输入或软件写寄存器产生重置请求时，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 16-1 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，下一次源地址重复跳转发生后的数值	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，下一次目标地址重复跳转发生后的数值		更新为寄存器 DMA_DARx 的初始设定值

 **说明**

当重置功能有效时，该通道使用寄存器 DMA_RPTBx 和 DMA_SNSEQCTLBx，DMA_DNSEQCTLBx 控制传输地址的重复跳转和不连续跳转。寄存器 DMA_RPTx 和 DMA_SNSEQCTLx，DMA_DNSEQCTLx 无效。

下图所示 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。

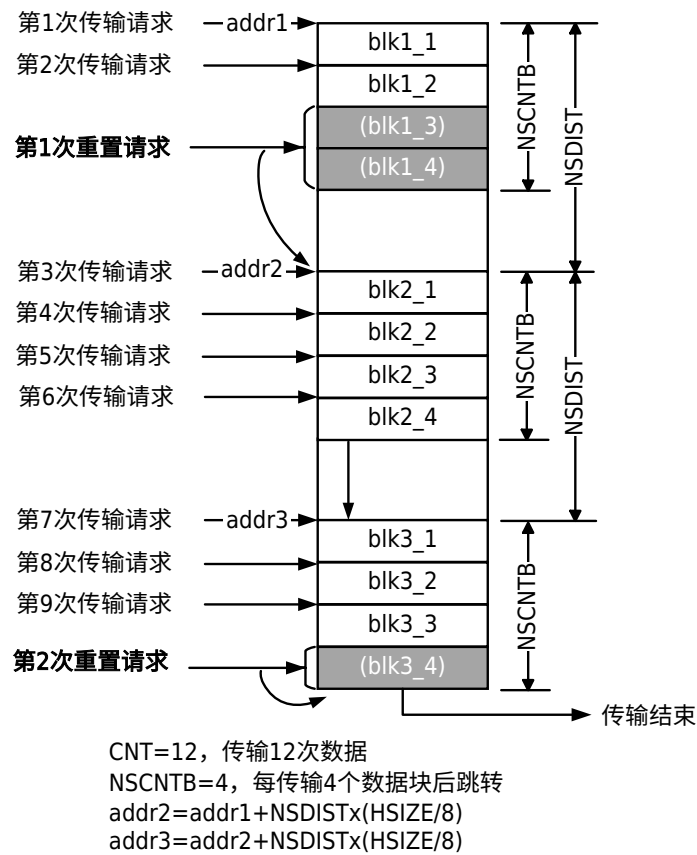


图 16-4 不连续式重置示意图

16.3.11 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。

16.3.12 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，可用作其他外围电路的触发电源，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。

16.4 操作示例

16.4.1 存储器到存储器的传输

目标

将 12 个数据从 RAM 地址 0x20000000 传送到 0x20001000，数据宽度为 32-bit。

寄存器设置

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 0x20000000。
 - b. 写 DMA_DAR0 配置目标地址为 SRAM 区域 0x20001000。
 - 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：
 - 1) 源地址和目标地址连锁传输无效。
 - 2) 源地址重复传输有效，目标地址更新方式为自增。
 - 3) 数据宽度为 32-bit。
 - 4) 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。

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 0x20000000 并继续传输剩余的 2 个地址。第二次传输完成后，传输次数 DMA_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。

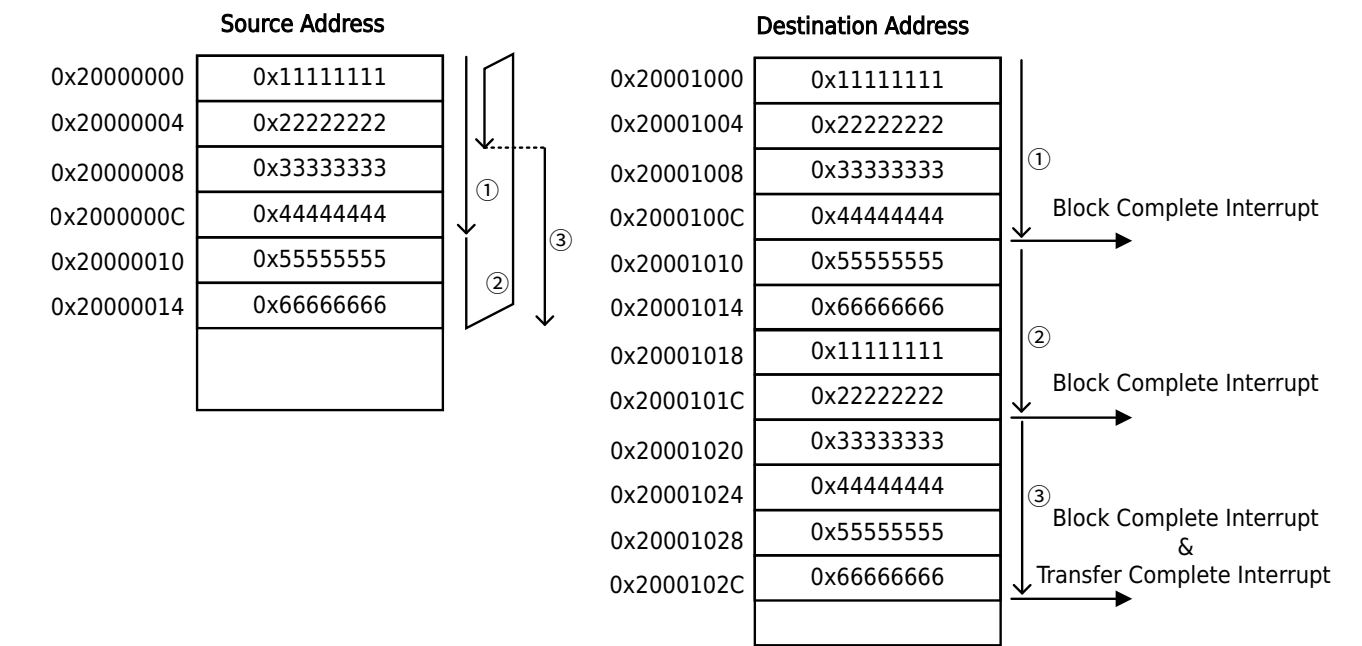


图 16-5 应用例 1：存储器到存储器传输

16.4.2 存储器到外围电路的传输

目标

将 10 个宽度为 16-bit 的数据从 RAM 地址 0x20000000 传送到通信模块的发送缓冲寄存器中，通信模块每发送完一个数据产生一次传送请求。当最后一个数据发送完成后 DMA 产生一个传输完成中断。

寄存器设定

1. DMA_EN.EN 写 1 使能 DMA 控制器。
2. 配置 DMA_INTMSK1 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，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 0x20000000。
 - b. 写 DMA_DAR0 配置目标地址为外围电路的寄存器地址 0x40000000。
 - 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：
 - 1) 源地址和目标地址连锁传输无效。
 - 2) 源地址更新方式为自增，目标地址为固定。
 - 3) 源地址和目标地址数据宽度为 16-bit。
 - 4) 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。

传输过程

通道使能后，DMA 等待来自通信模块的传输请求。当传输请求产生后，DMA 把 RAM 中的数据传送至通信模块的发送缓冲寄存器中，并等待第二次来自通信模块的传输请求，由于屏蔽了块传输完成中断，DMA 此时不产生中断请求。当 10 个数据全部传送完成后，DMA 产生一个传输完成中断，并且通道使能位 DMA_CHEN.CHEN0 将被自动清零。

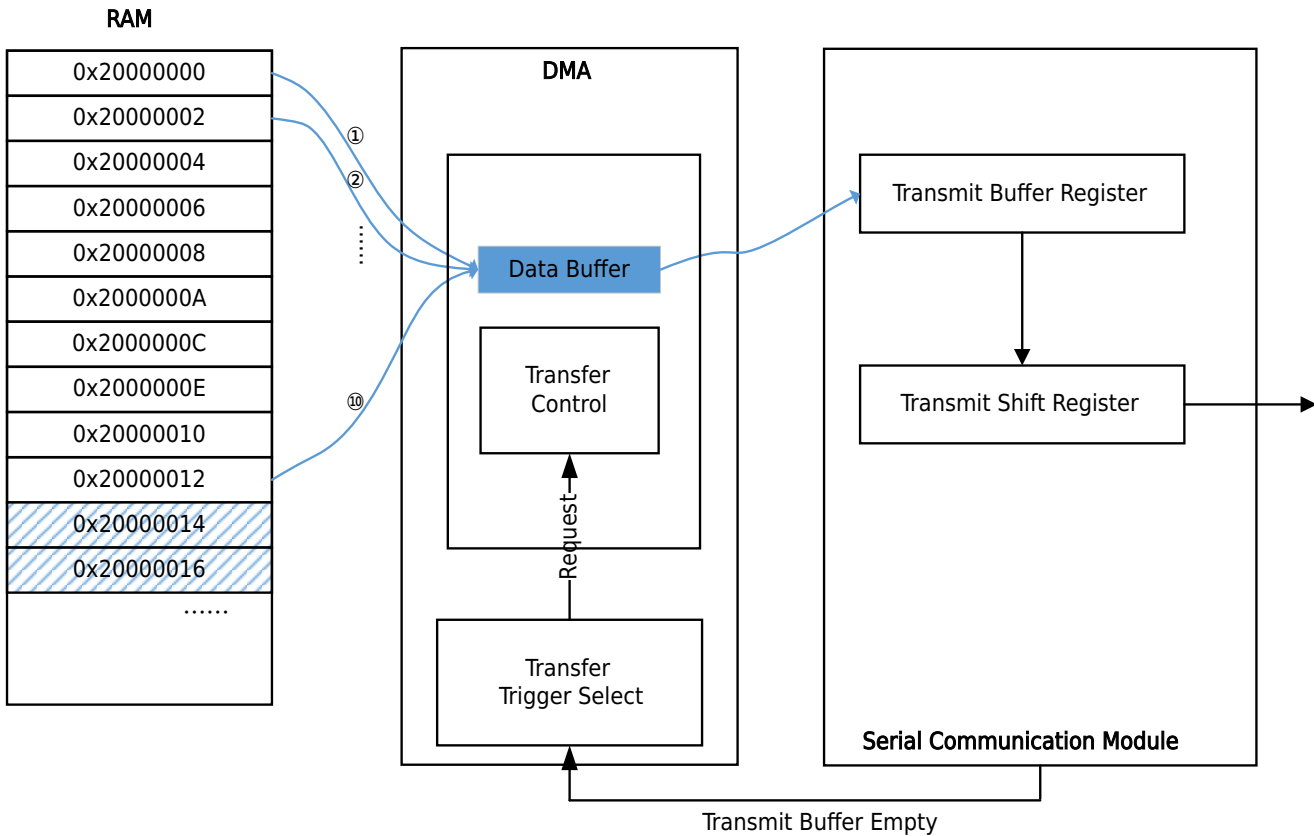


图 16-6 应用例 2：存储器到外围电路的传输

16.4.3 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 0x20000000 传到 0x20001000。第二块数据为 20 个 16 位数据，从 0x20000100 传到 0x20001100。第三块数据为 40 个 8 位数据，从 0x20000200 传到 0x20001200。

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 0x20000000。
 - b. 写 DMA_DAR0 配置目标地址为 SRAM 区域 0x20001000。
 - 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 0x20002000。
 - 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：
 - 1) 连锁传输有效。
 - 2) 连锁传输模式为直接启动下一次传输。
 - 3) 源地址和目标地址更新方式为自增。
 - 4) 数据宽度为 32-bit。
 - 5) 中断使能无效。
3. 在 RAM 空间的 0x20002000 地址中配置第二次传输的描述符（descriptor1），包括：
 - a. 0x20002000 中写入 32 位数据 0x20000100，此为第二次传输的源地址。
 - b. 0x20002004 中写入 32 位数据 0x20001100，此为第二次传输的目标地址。
 - c. 0x20002008 中配置数据块的大小为 20。

- d. 0x20002018 中写入 32 位数据 0x20002020，此为第三次传输的描述符 (descriptor2) 所在地址。
- e. 0x2000201C 中写入第二次传输的控制数据，该控制数据实现：
 - 1) 连锁传输有效。
 - 2) 连锁传输模式为直接启动下一次传输。
 - 3) 源地址和目标地址更新方式为自增。
 - 4) 数据宽度为 16-bit。
 - 5) 中断使能无效。
- 4. 在 RAM 空间的 0x20002020 地址中配置第三次传输的描述符 (descriptor2)，包括：
 - a. 0x20002020 中写入 32 位数据 0x20000200，此为第三次传输的源地址。
 - b. 0x20002024 中写入 32 位数据 0x20001200，此为第三次传输的目标地址。
 - c. 0x20002028 配置数据块的大小为 40。
 - d. 0x20002038 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. 0x2000203C 中写入第三次传输的控制数据实现：
 - 1) 连锁传输无效。
 - 2) 源地址和目标地址更新方式为自增。
 - 3) 数据宽度为 8-bit。
 - 4) 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 INTSFTRG.STRG 为 1，发送一个启动请求，DMA 开始传输数据。

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。

16.5 寄存器

16.5.1 寄存器总表

- DMA1 基地址: 0x40053000
- DMA2 基地址: 0x40053400

表 16-2 DMA 控制器 (DMA) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	DMA_EN	DMA 使能寄存器	32
0x04	DMA_INTSTAT0	中断状态寄存器 0	32
0x08	DMA_INTSTAT1	中断状态寄存器 1	32
0x0C	DMA_INTMASK0	中断屏蔽寄存器 0	32
0x10	DMA_INTMASK1	中断屏蔽寄存器 1	32
0x14	DMA_INTCLR0	中断复位寄存器 0	32
0x18	DMA_INTCLR1	中断复位寄存器 1	32
0x1C	DMA_CHEN	通道使能寄存器	32
0x20	DMA_REQSTAT	传输请求状态寄存器	32
0x24	DMA_CHSTAT	通道状态观测寄存器	32
0x2C	DMA_RCFGCTL	通道重置控制寄存器	32
0x30	DMA_SWREQ	软件触发寄存器	32
0x34	DMA_CHENCLR	通道使能复位寄存器	32
0x40+0x40*x	DMA_SARx	传输源地址寄存器	32
0x44+0x40*x	DMA_DARx	传输目标地址寄存器	32
0x48+0x40*x	DMA_DTCTLx	数据控制寄存器	32
0x4C+0x40*x	DMA_RPTx	重复区域大小寄存器	32
0x4C+0x40*x	DMA_RPTBx	重复区域大小寄存器 B	32
0x50+0x40*x	DMA_SNSEQCTLx	源地址不连续传输控制寄存器	32
0x50+0x40*x	DMA_SNSEQCTLBx	源地址不连续传输控制寄存器 B	32
0x54+0x40*x	DMA_DNSEQCTLx	目标地址不连续传输控制寄存器	32
0x54+0x40*x	DMA_DNSEQCTBx	目标地址不连续传输控制寄存器 B	32
0x58+0x40*x	DMA_LL Px	链指针寄存器	32
0x5C+0x40*x	DMA_CHCTLx	通道控制寄存器	32
0x60+0x40*x	DMA_MONSARx	传输源地址监视寄存器	32

偏移地址	寄存器	描述	位宽
0x64+0x40*x	DMA_MONDARx	传输目标地址监视寄存器	32
0x68+0x40*x	DMA_MONDTCTLx	数据控制监视寄存器	32
0x6C+0x40*x	DMA_MONRPTx	重复区域计数器监视寄存器	32
0x70+0x40*x	DMA_MONSNSEQCTLx	源地址不连续传输计数器监视寄存器	32
0x74+0x40*x	DMA_MONDNSEQCTLx	目标地址不连续传输计数器监视寄存器	32

 **说明**

表中的 x 代表存在多个寄存器，取值为 0~5。

16.5.2 DMA 使能寄存器 (DMA_EN)

Offset	Bit Position																															
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															EN
Access																																RW

位/位域	标记	位名	功能描述	读写
31:1	Reserved	保留	-	-
0	EN	DMA 使能位	0b0: 禁止 0b1: 使能	RW

16.5.3 中断状态寄存器 0 (DMA_INTSTAT0)

Offset	Bit Position																															
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved											REQERR				Reserved											TRNERR					
Access												RO															RO					

位/位域	标记	位名	功能描述	读写
31:22	Reserved	保留	-	-
21:16	REQERR	传输请求溢出错误中断位	对应 REQERR5~REQERR0: 0b0: 该通道没有发生传输请求溢出错误 0b1: 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  说明 每个 Bit 对应一个传输通道。	RO
15:6	Reserved	保留	-	-
5:0	TRNERR	传输错误中断位	对应 TRNERR5~TRNERR0: 0b0: 该通道没有发生传输错误 0b1: 该通道发生了传输错误  说明 每个 Bit 对应一个传输通道。	RO

16.5.4 中断状态寄存器 1 (DMA_INTSTAT1)

Offset	Bit Position																															
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved										BTC						Reserved										TC					
Access											RO																RO					

位/位域	标记	位名	功能描述	读写
31:22	Reserved	保留	-	-
21:16	BTC	块传输完成中断位	对应 BTC5~BTC0，该中断在完成一个数据块的传输后发生： ● 0b0：该通道没有块传输中断发生 ● 0b1：该通道发生块传输中断  说明 每个 Bit 对应一个传输通道。	RO
15:6	Reserved	保留	-	-
5:0	TC	传输完成中断位	对应 TC5~TC0，该中断在完成传输 DMA_DTCTLx.CNT 设置的次数后发生： ● 0b0：该通道没有传输完成中断发生 ● 0b1：该通道发生传输完成中断  说明 每个 Bit 对应一个传输通道。	RO

16.5.5 中断屏蔽寄存器 0 (DMA_INTMASK0)

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved								MSKREQERR								Reserved								MSKTRNERR							
Access									RW																RW							

位/位域	标记	位名	功能描述	读写
31:22	Reserved	保留	-	-
21:16	MSKREQERR	传输请求溢出中断屏蔽	对应 MSKREQERR5~MSKREQERR0： ● 0b0：不屏蔽传输请求溢出中断 ● 0b1：屏蔽传输请求溢出中断  说明 每个 Bit 对应一个传输通道。	RW
15:6	Reserved	保留	-	-
5:0	MSKTRNERR	传输错误中断屏蔽	对应 MSKTRNERR5~MSKTRNERR0： ● 0b0：不屏蔽传输错误中断 ● 0b1：屏蔽传输错误中断  说明 每个 Bit 对应一个传输通道。	RW

16.5.6 中断屏蔽寄存器 1 (DMA_INTMASK1)

Offset	Bit Position																															
0x10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved										MSKBTC					Reserved										MSKTC						
Access											RW															RW						

位/位域	标记	位名	功能描述	读写
31:22	Reserved	保留	-	-
21:16	MSKBTC	块传输完成中断屏蔽	对应 MSKBTC5~MSKBTC0： ● 0b0：不屏蔽块传输完成中断 ● 0b1：屏蔽块传输完成中断  说明 每个 Bit 对应一个传输通道。	RW
15:6	Reserved	保留	-	-
5:0	MSKTC	传输完成中断屏蔽	对应 MSKTC5~MSKTC0： ● 0b0：不屏蔽传输完成中断 ● 0b1：屏蔽传输完成中断  说明 每个 Bit 对应一个传输通道。	RW

16.5.7 中断复位寄存器 0 (DMA_INTCLR0)

Offset	Bit Position																															
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved								CLRREQERR								Reserved								CLRTRNERR							
Access									WO																WO							

位/位域	标记	位名	功能描述	读写
31:22	Reserved	保留	-	-
21:16	CLRREQERR	传输请求溢出错误中断复位	对应 CLRREQERR5~CLRREQERR0: 写 1 复位传输请求溢出错误中断状态位, 写 0 无效; 读出恒为 0  说明 每个 Bit 对应一个传输通道。	WO
15:6	Reserved	保留	-	-
5:0	CLRTRNERR	传输错误中断复位	对应 CLRTRNERR5~CLRTRNERR0: 写 1 复位传输错误中断状态位, 写 0 无效; 读出恒为 0  说明 每个 Bit 对应一个传输通道。	WO

16.5.8 中断复位寄存器 1 (DMA_INTCLR1)

Offset	Bit Position																															
0x18	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved								CLR BTC								Reserved								CLR RTC							
Access									WO																WO							

位/位域	标记	位名	功能描述	读写
31:22	Reserved	保留	-	-
21:16	CLRBTC	块传输完成中断复位	对应 CLRBTC5~CLRBTC0： 写 1 复位块传输完成中断状态位，写 0 无效；读出恒为 0  说明 每个 Bit 对应一个传输通道。	WO
15:6	Reserved	保留	-	-
5:0	CLRRTC	传输完成中断复位	对应 CLRRTC5~CLRRTC0： 写 1 复位传输完成中断状态位，写 0 无效；读出恒为 0  说明 每个 Bit 对应一个传输通道。	WO

16.5.9 通道使能寄存器 (DMA_CHEN)

Offset	Bit Position																															
0x1C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								CHEN							
Access																									RW							

位/位域	标记	位名	功能描述	读写
31:6	Reserved	保留	-	-
5:0	CHEN	通道使能位	对应 CHEN5~CHEN0: ● 0b0: 该通道无效 ● 0b1: 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。  说明 ● 每个 Bit 对应一个传输通道。 ● 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

16.5.10 传输请求状态寄存器 (DMA_REQSTAT)

Offset	Bit Position																																
0x20	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																RCFGREQ	Reserved								CHREQ							
Access																	RO									RO							

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15	RCFGREQ	通道重置请求标志	● 0b0: 无通道重置请求 ● 0b1: 有通道重置请求  说明 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。	RO
14:6	Reserved	保留	-	-
5:0	CHREQ	通道传输请求标志位	对应 CHREQ5~CHREQ0: ● 0b0: 该通道无传输请求 ● 0b1: 该通道有传输请求  说明 每个 Bit 对应一个传输通道。 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。	RO

16.5.11 通道状态观测寄存器 (DMA_CHSTAT)

Offset	Bit Position																															
0x24	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved										CHACT					Reserved										RCFGACT	DMAACT					
Access											RO															RO	RO					

位/位域	标记	位名	功能描述	读写
31:22	Reserved	保留	-	-
21:16	CHACT	传输动作中通道 监视位	对应 CHACT5~CHACT0： ● 0b0：该通道空闲中 ● 0b1：该通道处于动作中  说明 每个 Bit 对应一个传输通道。	RO
15:2	Reserved	保留	-	-
1	RCFGACT	DMA 通道重置动 作中监视位	● 0b0：DMA 未处于通道重置动作中 ● 0b1：DMA 处于通道重置动作中	RO
0	DMAACT	DMA 动作中监视 位	● 0b0：DMA 未处于传输动作中 ● 0b1：DMA 处于传输动作中	RO

16.5.12 通道重置控制寄存器 (DMA_RCFGCTL)

Offset	Bit Position																															
0x2C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved										CNTMD		DARMD		SARMD		Reserved		RCFGCHS				Reserved				RCFGLLP		RCFGEN			
Access											RW		RW		RW				RW								RW		RW			

位/位域	标记	位名	功能描述	读写
31:22	Reserved	保留	-	-
21:20	CNTMD	剩余传输次数计数器重置方式	0b00: 保持不变 0b01: 按源地址方式 0b10/0b11: 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
19:18	DARMD	目标地址重置方式	0b00: 保持不变 0b01: 不连续式重置, 下次传输的目标地址更新为 $\text{addr\_base} + (\text{DMA\_DNSEQCTLBx.DNSDIST} \times \text{DMA\_CHCTLx.HSIZE}(\text{bit}) / 8)$, 其中: addr_base 表示当前不连续传输区域的首地址 0b10/0b11: 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
17:16	SARMD	源地址重置方式	0b00: 保持不变 0b01: 不连续式重置, 下次传输的源地址更新为 $\text{addr\_base} + (\text{DMA\_SNSEQCTLBx.SNSDIST} \times \text{DMA\_CHCTLx.HSIZE}(\text{bit}) / 8)$, 其中: addr_base 表示当前不连续传输区域的首地址 0b10/0b11: 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
15:12	Reserved	保留	-	-

位/位域	标记	位名	功能描述	读写
11:8	RCFGCHS	重置通道选择	● 0x0：通道 0 ● 0x1：通道 1 ● ... ● 0x5：通道 5 ● 其他：禁止设置	RW
7:2	Reserved	保留	-	-
1	RCFGLLP	链指针式通道重置	● 0b0：链指针式重置无效 ● 0b1：链指针式重置有效  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 Bit21-Bit16 全部无效。	RW
0	RCFGEN	通道重置许可	● 0b0：禁止事件触发通道配置寄存器强制更新 ● 0b1：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。

16.5.13 软件触发寄存器 (DMA_SWREQ)

Offset	Bit Position																																
0x30	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	SWRCFGWP								SWREQWP								SWRCFGREQ	Reserved								SWREQ							
Access	WO								WO								WO									WO							

位/位域	标记	位名	功能描述	读写
31:24	SWRCFGWP	软件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	WO
23:16	SWREQWP	软件通道触发请求写保护	软件触发请求写保护，SWREQWP 写入值为 0xA1 解除写保护	WO
15	SWRCFGREQ	软件通道重置请求	写 1 (SWRCFGWP 同时写 0xA2) 产生一次软件通道重置请求，写 0 无效  说明 软件触发通道重置功能，与 AOS 触发效果相同。	WO
14:6	Reserved	保留	-	-
5:0	SWREQ	软件通道触发请求	对应 SWREQ5~SWREQ0： 写 1 (SWREQWP 同时写 0xA1) 产生一次软件启动传输请求，写 0 无效  说明 每个 Bit 对应一个传输通道。 软件触发启动并进行传输，与 AOS 触发效果相同。	WO

16.5.14 通道使能复位寄存器 (DMA_CHENCLR)

Offset	Bit Position																															
0x34	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								CHENCLR							
Access																									WO							

位/位域	标记	位名	功能描述	读写
31:6	Reserved	保留	-	-
5:0	CHENCLR	通道使能位复位	写 1 复位通道使能位 CHEN，写 0 无效；读出恒为 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  说明 每个 Bit 对应一个传输通道。	WO

16.5.15 传输源地址寄存器 (DMA_SARx) (x=0~5)

偏移地址 (Offset) : 0x40+0x40*x

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	SAR																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	SAR	传输源地址	设置传输源地址  说明 - 传输数据宽度为 16-bit 时，即 DMA_CHCTLx.HSIZE=0b01，SAR[0]无效。 - 传输数据宽度为 32-bit 时，即 DMA_CHCTLx.HSIZE=0b10/0b11，SAR[1:0]无效。	RW

16.5.16 传输目标地址寄存器 (DMA_DARx) (x=0~5)

偏移地址 (Offset) : 0x44+0x40*x

Offset	Bit Position																																					
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	DAR																																					
Access	RW																																					

位/位域	标记	位名	功能描述	读写
31:0	DAR	传输目标地址	设置传输目标地址  说明 - 传输数据宽度为 16-bit 时, 即 DMA_CHCTLx.HSIZE=0b01, DAR[0]无效。 - 传输数据宽度为 32-bit 时, 即 DMA_CHCTLx.HSIZE=0b10/0b11, DAR[1:0]无效。	RW

16.5.17 数据控制寄存器 (DMA_DTCTLx) (x=0~5)

偏移地址 (Offset) : 0x48+0x40*x

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000001																															
Name	CNT																Reserved								BLKSIZE							
Access	RW																								RW							

位/位域	标记	位名	功能描述	读写
31:16	CNT	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
15:10	Reserved	保留	-	-
9:0	BLKSIZE	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.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

16.5.18 重复区域大小寄存器 (DMA_RPTx) (x=0~5)

偏移地址 (Offset) : 0x4C+0x40*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved						DRPT[9:0]						Reserved						SRPT[9:0]													
Access							RW												RW													

位/位域	标记	位名	功能描述	读写
31:26	Reserved	保留	-	-
25:16	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 个数据后目标地址载入为 DMA_DARx 寄存器初始设定值。寄存器设为 10 则每传输 10 个数据后地址载入初始值，设为 0 则每传输 1024 个数据后地址载入初始值。每个数据的宽度由 DMA_CHCTLx.HSIZE 决定。	RW
15:10	Reserved	保留	-	-
9:0	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 个数据后源地址载入为 DMA_SARx 寄存器初始设定值。寄存器设为 10 则每传输 10 个数据后地址载入初始值，设为 0 则每传输 1024 个数据后地址载入初始值。每个数据的宽度由 DMA_CHCTLx.HSIZE 决定。	RW

 说明

- 1. 该寄存器配置源地址和目标地址重复区域的大小。使用重复传输需要配置 DMA_CHCTLx.SRPTEN/DRPREN 位有效，并且配置 DMA_CHCTLx.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复跳转无效。
- 2. 该寄存器在本通道的重置功能打开时无效。

16.5.19 重复区域大小寄存器 B (DMA_RPTBx) (x=0~5)

偏移地址 (Offset) : 0x4C+0x40*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved				DRPTB[9:0]								Reserved				SRPTB[9:0]															
Access					RW												RW															

位/位域	标记	位名	功能描述	读写
31:26	Reserved	保留	-	-
25:16	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载入 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。  说明 本通道重置功能有效 (DMA_RCFGCTL.RCFGEN=1) 时, DRPTB[9:0]的设定值必须大于 0。	RW
15:10	Reserved	保留	-	-
9:0	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 个数据块后源地址载入 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。  说明 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


说明

- 该寄存器配置源地址和目标地址重复区域的大小。使用重复传输需要配置 DMA_CHCTLx.SRPTEN/DRPREN 位有效, 并且配置 DMA_CHCTLx.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复跳转无效。
- 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。

16.5.20 源地址不连续传输控制寄存器 (DMA_SNSEQCTLx) (x=0~5)

偏移地址 (Offset) : 0x50+0x40*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	SNSCNT[11:0]												SOFFSET[19:0]																			
Access	RW												RW																			

位/位域	标记	位名	功能描述	读写
31:20	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
19:0	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，详情请参见 不连续地址传输示意图 。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 32-bit 时，地址偏移量为 10×4=40；若数据宽度为 16-bit 时，偏移量为 10×2=20；若数据宽度为 8-bit 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

 说明

- 1. 使用源设备不连续传输需要配置 DMA_CHCTLx.SNSEQEN 位有效，并且配置 DMA_CHCTLx.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。

16.5.21 源地址不连续传输控制寄存器 B (DMA_SNSEQCTLBx) (x=0~5)

偏移地址 (Offset) : 0x50+0x40*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	SNSCNTB[11:0]												SNSDIST[19:0]																			
Access	RW												RW																			

位/位域	标记	位名	功能描述	读写
31:20	SNSCNTB[11:0]	源地址跳转的数据量	设置源地址跳转前传输的数据量大小。源设备在每传输 SNSCNTB 个数据块后源地址按 SNSDIST 指定地址间距跳转。数据块大小由 DMA_DTCTLx.BLKSIZE 与 DMA_CHCTLx.HSIZE 决定。  说明 本通道重置功能有效 (DMA_RCFGCTL.RCFGGEN=1) 时, SNSCNTB[11:0]的设定值必须大于 0。	RW
19:0	SNSDIST[19:0]	源不连续区域地址间距	当不连续地址传输时, 设置源设备两个不连续区域的间距。跳转的方向根据通道控制寄存器 DMA_CHCTLx.SINC 的值向前或向后跳转, 详情请参见不连续式重置示意图。当 DMA_CHCTLx.SINC 设为地址固定时, 不连续地址传输无效。地址间距将根据数据的宽度 (DMA_CHCTLx.HSIZE) 设置的 bit 数和 SNSDIST 的值进行计算。 地址间距= SNSDIST × (DMA_CHCTLx.HSIZE(bit) / 8) 例如, 当 SNSDIST 设为 10, 数据宽度为 32-bit 时, 地址间距为 10×4=40; 若数据宽度为 16-bit 时, 间距为 10×2=20; 若数据宽度为 8-bit 时, 间距为 10×1=10。 下次传输的源地址=当前源不连续区域首地址±地址间距	RW

 说明

1. 使用源设备不连续传输需要配置 DMA_CHCTLx.SNSEQEN 位有效, 并且配置 DMA_CHCTLx.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_SNSEQCTLx。通道重置功能关闭时无效。

16.5.22 目标地址不连续传输控制寄存器 (DMA_DNSEQCTLx) (x=0~5)

偏移地址 (Offset) : 0x54+0x40*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	DNSCNT[11:0]												DOFFSET[19:0]																			
Access	RW												RW																			

位/位域	标记	位名	功能描述	读写
31:20	DNSCNT[11:0]	目标地址跳转的数据量	设置目标地址跳转前传输的数据量大小。 目标设备在每传输 DNSCNT 个数据后目标地址按 DOFFSET 指定偏移量跳转。寄存器设为 10 则每传输 10 个数据后地址跳转, 设为 0 则每传 4096 个数据后地址跳转。	RW
19:0	DOFFSET[19:0]	目标地址跳转的地址偏移量	当不连续地址传输时, 设置目标地址跳转的偏移量。偏移量是相对于当前传输地址, 即跳转前最后一次传输地址而言的。跳转的方向根据通道控制寄存器 DMA_CHCTLx.DINC 的值向前或向后跳转, 详情请参见 不连续地址传输示意图 。当 DMA_CHCTLx.DINC 设为地址固定时, 不连续地址传输将无效。 跳转地址将根据数据的宽度 (DMA_CHCTLx.HSIZE) 设置的 bit 数和 DOFFSET 的值进行计算。 地址偏移量=DOFFSET × (DMA_CHCTLx.HSIZE(bit) / 8) 例如, 当 DOFFSET 设为 10, 数据宽度为 32-bit 时, 地址偏移量为 10×4=40; 若数据宽度为 16-bit 时, 偏移量为 10×2=20; 若数据宽度为 8-bit 时, 偏移量为 10×1=10。 下次传输的目标地址=当前传输的目标地址±地址偏移量	RW

 **说明**

1. 使用目标设备不连续传输需要配置 DMA_CHCTLx.DNSEQEN 位有效, 并且配置 DMA_CHCTLx.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。

16.5.23 目标地址不连续传输控制寄存器 B (DMA_DNSEQCTLBx) (x=0~5)

偏移地址 (Offset) : 0x54+0x40*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	DNSCNTB[11:0]												DNSDIST[19:0]																			
Access	RW												RW																			

位/位域	标记	位名	功能描述	读写
31:20	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。  说明 本通道重置功能有效 (DMA_RCFGCTL.RCFGGEN=1) 时, DNSCNTB[11:0]的设定值必须大于 0。	RW
19:0	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, 详情请参见不连续式重置示意图。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 32-bit 时, 地址间距为 10×4=40; 若数据宽度为 16-bit 时, 间距为 10×2=20; 若数据宽度为 8-bit 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

 **说明**

1. 使用目标设备不连续传输需要配置 DMA_CHCTLx.DNSEQEN 位有效, 并且配置 DMA_CHCTLx.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。

16.5.24 链指针寄存器 (DMA_LL Px) (x=0~5)

偏移地址 (Offset) : 0x58+0x40*x

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	LLP[31:2]																															Reserved
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:2	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
1:0	Reserved	保留	-	-

16.5.25 通道控制寄存器 (DMA_CHCTLx) (x=0~5)

偏移地址 (Offset) : 0x5C+0x40*x

Offset	Bit Position																																	
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00001000																																	
Name	Reserved															HPROT[3:2]	Reserved	IE	LLPRUN	LLPEN	HSIZE	DNSEQEN	SNSEQEN	DRPTEN	SRPTEN	DINC	SINC							
Access																RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW							

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:14	HPROT[3:2]	AHB 总线中的 HPROT 值	HPROT[3:2]位控制 AHB 总线的 HPROT[3:2]。 当使用 DMA 访问 SMC 时, 请将 HPROT[3:2]设为 0b11; 其他情况请将 HPROT[3:2]设为 0b00。	RW
13	Reserved	保留	-	-
12	IE	中断使能位	配置该通道是否产生中断 ● 0b0: 该通道不产生中断 ● 0b1: 该通道产生中断	RW
11	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 ● 0b0: 不立即传输, 等待下一个传输请求产生后开 始传输 ● 0b1: 新描述符载入后, 立即开始传输	RW
10	LLPEN	连锁传输使能	● 0b0: 连锁传输无效 ● 0b1: 连锁传输有效	RW

位/位域	标记	位名	功能描述	读写
9:8	HSIZE	传输数据的宽度	0b00: 8-bit 0b01: 16-bit 0b10/0b11: 32-bit	RW
7	DNSEQEN	目标地址不连续传输使能	0b0: 不允许不连续地址传输 0b1: 允许不连续地址传输	RW
6	SNSEQEN	源地址不连续传输使能	0b0: 不允许不连续地址传输 0b1: 允许不连续地址传输	RW
5	DRPTEN	目标地址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 0b0: 不重新载入初始值 0b1: 重新载入初始值, 进行地址重复传输	RW
4	SRPTEN	源地址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 0b0: 不重新载入初始值 0b1: 重新载入初始值, 进行地址重复传输	RW
3:2	DINC	目标地址的更新方式	0b00: 固定 0b01: 递增 0b10/0b11: 递减	RW
1:0	SINC	源地址的更新方式	0b00: 固定 0b01: 递增 0b10/0b11: 递减	RW

16.5.26 传输源地址监视寄存器 (DMA_MONSARx) (x=0~5)

偏移地址 (Offset) : 0x60+0x40*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	RO SAR																															
Access	RO																															

位/位域	标记	位名	功能描述	读写
31:0	SAR	传输源地址	设置传输源地址  说明 - 传输数据宽度为 16-bit 时, 即 DMA_CHCTLx.HSIZE=0b01, SAR[0]无效。 - 传输数据宽度为 32-bit 时, 即 DMA_CHCTLx.HSIZE=0b10/0b11, SAR[1:0]无效。	RO



说明

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。DMA_MONSARx.SAR 的更新方式为: 按寄存器设置的固定/递增/递减/重复跳转/不连续跳转方式更新为下一次传输的地址。

16.5.27 传输目标地址监视寄存器 (DMA_MONDARx) (x=0~5)

偏移地址 (Offset) : 0x64+0x40*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	RO DAR																															
Access	RO																															

位/位域	标记	位名	功能描述	读写
31:0	DAR	传输目标地址	设置传输目标地址 说明 ● 传输数据宽度为 16-bit 时，即 DMA_CHCTLx.HSIZE=0b01，DAR[0]无效。 ● 传输数据宽度为 32-bit 时，即 DMA_CHCTLx.HSIZE=0b10/0b11，DAR[1:0]无效。	RO

说明

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_MONSARx) 会在 DMA 每完成一个数据块传输后更新。DMA_MONDARx.DAR 的更新方式为：按寄存器设置的固定/递增/递减/重复跳转/不连续跳转方式更新为下一次传输的地址。

16.5.28 数据控制监视寄存器 (DMA_MONDTCTLx) (x=0~5)

偏移地址 (Offset) : 0x68+0x40*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000001																															
Name	CNT[15:0]																Reserved				BLKSIZE[9:0]											
Access	RO																				RO											

位/位域	标记	位名	功能描述	读写
31:16	CNT[15:0]	传输次数	剩余传输次数  说明 传输次数为 0 时 (DMA_DTCTLx.CNT=0x0000) , 该位域保持为 0。	RO
15:10	Reserved	保留	-	-
9:0	BLKSIZE[9:0]	数据块的大小	数据块的大小  说明 该位域为 0 时, 表示每次传输 1024 个数据。	RO

16.5.29 重复区域计数器监视寄存器 (DMA_MONRPTx) (x=0~5)

偏移地址 (Offset) : 0x6C+0x40*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved						DRPT[9:0]						Reserved						SRPT[9:0]													
Access							RO												RO													

位/位域	标记	位名	功能描述	读写
31:26	Reserved	保留	-	-
25:16	DRPT[9:0]	目标地址重复区域大小	-	RO
15:10	Reserved	保留	-	-
9:0	SRPT[9:0]	源地址重复区域大小	-	RO

 **说明**

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。

DMA_MONRPTx.SRPT[9:0]和 DMA_MONRPTx.DRPT[9:0]的更新方式为：

- 通道重置无效时：减去块大小，减到 0 时重新载入 DMA_RPTx 设定值。
- 通道重置有效时：减 1，减到 0 时重新载入初始值。

16.5.30 源地址不连续传输计数器监视寄存器 (DMA_MONSNSEQCTLx) (x=0~5)

偏移地址 (Offset) : 0x70+0x40*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	SNSCNT[11:0]												SOFFSET[19:0]																			
Access	RO												RO																			

位/位域	标记	位名	功能描述	读写
31:20	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	RO
19:0	SOFFSET[19:0]	源地址跳转的地址偏移量	源地址跳转的偏移量	RO

说明

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。

DMA_MONSNSEQCTLx.SNSCNT[11:0]的更新方式为：

- 通道重置无效时：减去块大小，减到 0 时重新载入 DMA_SNSEQCTLx 初始值。
- 通道重置有效时：减 1，减到 0 时重新载入初始值。

16.5.31 目标地址不连续传输计数器监视寄存器（DMA_MONDNSEQCTLx）（x=0~5）

偏移地址（Offset）：0x74+0x40*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	DNSCNT[11:0]												DOFFSET[19:0]																			
Access	RO												RO																			

位/位域	标记	位名	功能描述	读写
31:20	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	RO
19:0	DOFFSET[19:0]	目标地址跳转的地址偏移量	目标地址跳转的偏移量	RO

说明

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。

DMA_MONDNSEQCTLx.DNSCNT[11:0]的更新方式为：

- 通道重置无效时：减去块大小，减到 0 时重新载入 DMA_DNSEQCTLx 初始值。
- 通道重置有效时：减 1，减到 0 时重新载入初始值。

16.6 注意事项

- DMA 的寄存器只支持 32-bit 读写，8-bit/16-bit 读写操作无效。
- 禁止 DMA 访问会发生总线错误的区域（保留地址空间、受保护的地址空间等）。

17 电压比较器 (CMP)

17.1 简介

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 4 个比较通道: CMP1 和 CMP2、CMP3 和 CMP4。

17.2 主要特性

- 4 个比较通道可独立进行普通比较
- 同组的两个比较通道组合使用可实现最多 2 个窗口比较
- 每个比较通道的正/负端电压均有多个输入源 (IO、DAC) 供选择
- 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

17.3 功能说明

17.3.1 功能框图

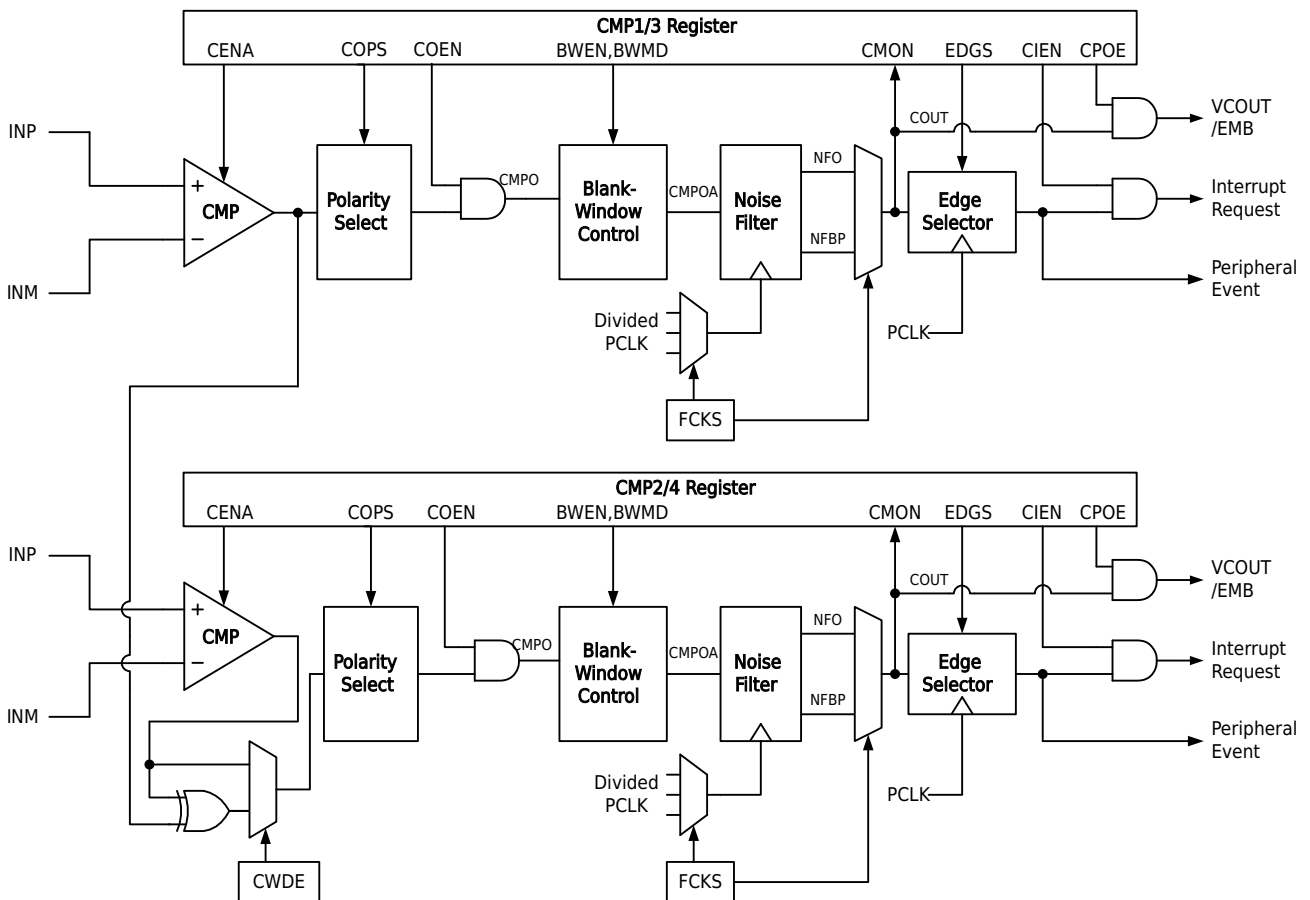


图 17-1 CMP 功能框图

表 17-1 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
CMP1_INP2	输入	CMP1 正端模拟输入
CMP1_INP3	输入	CMP1 正端模拟输入
CMP1_INP4	输入	CMP1 正端模拟输入
CMP1234_INM1	输入	CMP1~4 负端模拟输入
CMP1_INM2	输入	CMP1 负端模拟输入
CMP2_INP1	输入	CMP2 正端模拟输入
CMP2_INP2	输入	CMP2 正端模拟输入
CMP2_INP3	输入	CMP2 正端模拟输入
CMP2_INP4	输入	CMP2 正端模拟输入
CMP2_INM2	输入	CMP2 负端模拟输入
CMP3_INP1	输入	CMP3 正端模拟输入
CMP3_INP2	输入	CMP3 正端模拟输入
CMP3_INP3	输入	CMP3 正端模拟输入
CMP3_INP4	输入	CMP3 正端模拟输入
CMP3_INM2	输入	CMP3 负端模拟输入
CMP4_INP1	输入	CMP4 正端模拟输入
CMP4_INP2	输入	CMP4 正端模拟输入
CMP4_INP3	输入	CMP4 正端模拟输入
CMP4_INP4	输入	CMP4 正端模拟输入
CMP4_INM2	输入	CMP4 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
VCOUT3	输出	CMP3 的外部端口输出
VCOUT4	输出	CMP4 的外部端口输出
VCOUT	输出	CMP1~4 的外部端口输出

17.3.2 输入选择

CMP1~4 的模拟输入分别按照下表进行选择。

表 17-2 CMP 正端输入一览表

比较通道	CVSL=0b0001 (INP1)	CVSL=0b0010 (INP2)	CVSL=0b0100 (INP3)	CVSL=0b1000 (INP4)
CMP1	CMP1_INP1	CMP1_INP2	CMP1_INP3	CMP1_INP4
CMP2	CMP2_INP1	CMP2_INP2	CMP2_INP3	CMP2_INP4
CMP3	CMP3_INP1	CMP3_INP2	CMP3_INP3	CMP3_INP4
CMP4	CMP4_INP1	CMP4_INP2	CMP4_INP3	CMP4_INP4

表 17-3 CMP 负端输入一览表

比较通道	RVSL=0b0001 (INM1)	RVSL=0b0010 (INM2)	RVSL=0b0100 (INM3)	RVSL=0b1000 (INM4)
CMP1	CMP1234_INM1	CMP1_INM2	DAO1	DAO2
CMP2	CMP1234_INM1	CMP2_INM2	DAO1	DAO2
CMP3	CMP1234_INM1	CMP3_INM2	DAO1	DAO2
CMP4	CMP1234_INM1	CMP4_INM2	DAO1	DAO2

17.3.3 电压比较

CMP1~4 可以独立进行电压比较。以 CMP1 为例，设定步骤如下：

- 步骤 1. 使能 CMP1 模块功能 (PWC_FCG3.CMP12 写 0) ；
- 步骤 2. 设定 CMP_PMSR.CVSL，选择正端电压；
设定 CMP_PMSR.RVSL，选择负端电压；
- 步骤 3. 将 CMP_MDR.CENA 设为 1，启动比较器；
- 步骤 4. 等待比较器的稳定时间 t_{CMP} （具体数值请参见《数据手册》的“电气特性”）；
- 步骤 5. 设定 CMP_FIR.FCKS，设置数字滤波的采样周期；
设定 CMP_FIR.EDGS 和 CMP_FIR.CIEN，设置中断发生的边沿；
- 步骤 6. 设定 CMP_OCR.CPOE、CMP_OCR.COPS，设置 VCOUT 的输出；
- 步骤 7. 将 CMP_OCR.COEN 设为 1，允许比较器输出。

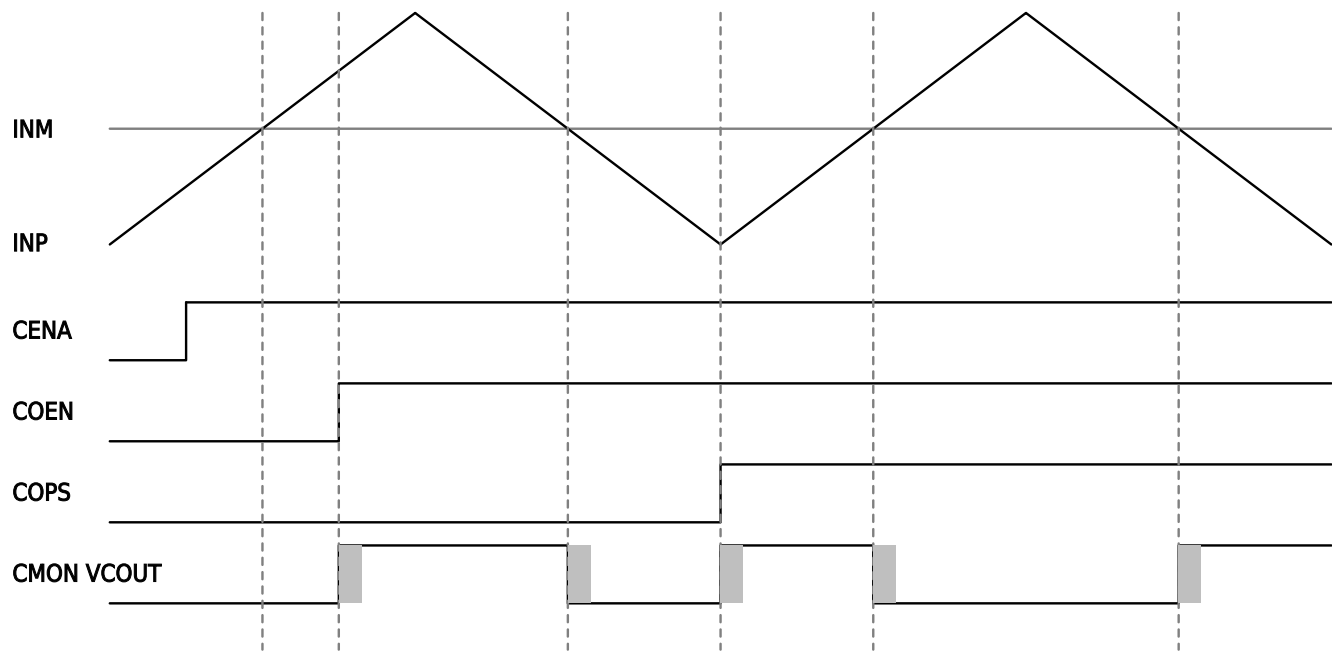


图 17-2 电压比较示意图

如上图所示，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.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。

17.3.4 窗口比较

CMP1 与 CMP2 或者 CMP3 与 CMP4 同时运行时可实现窗口比较。以 CMP1 与 CMP2 为例，设定步骤如下：

- 步骤 1. 使能 CMP1 与 CMP2 模块功能 (PWC_FCG3.CMP12 写 0)；
- 步骤 2. 设定 CMP1_PMSR.CVSL，选择正端电压；
设定 CMP1_PMSR.RVSL，选择的负端电压作为窗口下限；
- 步骤 3. 设定 CMP2_PMSR.CVSL，选择与 CMP1 相同的正端电压；
设定 CMP2_PMSR.RVSL，选择的负端电压作为窗口上限；
- 步骤 4. 将 CMP2_MDR.CWDE 设为 1，启动窗口比较；
- 步骤 5. 将 CMP1_MDR.CENA 和 CMP2_MDR.CENA 设为 1，启动比较器；
- 步骤 6. 等待稳定时间 t_{CMP} （具体数值请参见《数据手册》的“电气特性”）；
- 步骤 7. 设定 CMP2_FIR.FCKS，设置数字滤波器；
设定 CMP2_FIR.EDGS、CMP2_FIR.CIEN，设置中断和边沿检测条件；
- 步骤 8. 设定 CMP2_OCR.CPOE、CMP2_OCR.COPS，设置 VCOUT 的输出；
- 步骤 9. 将 CMP1_OCR.COEN、CMP2_OCR.COEN 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。

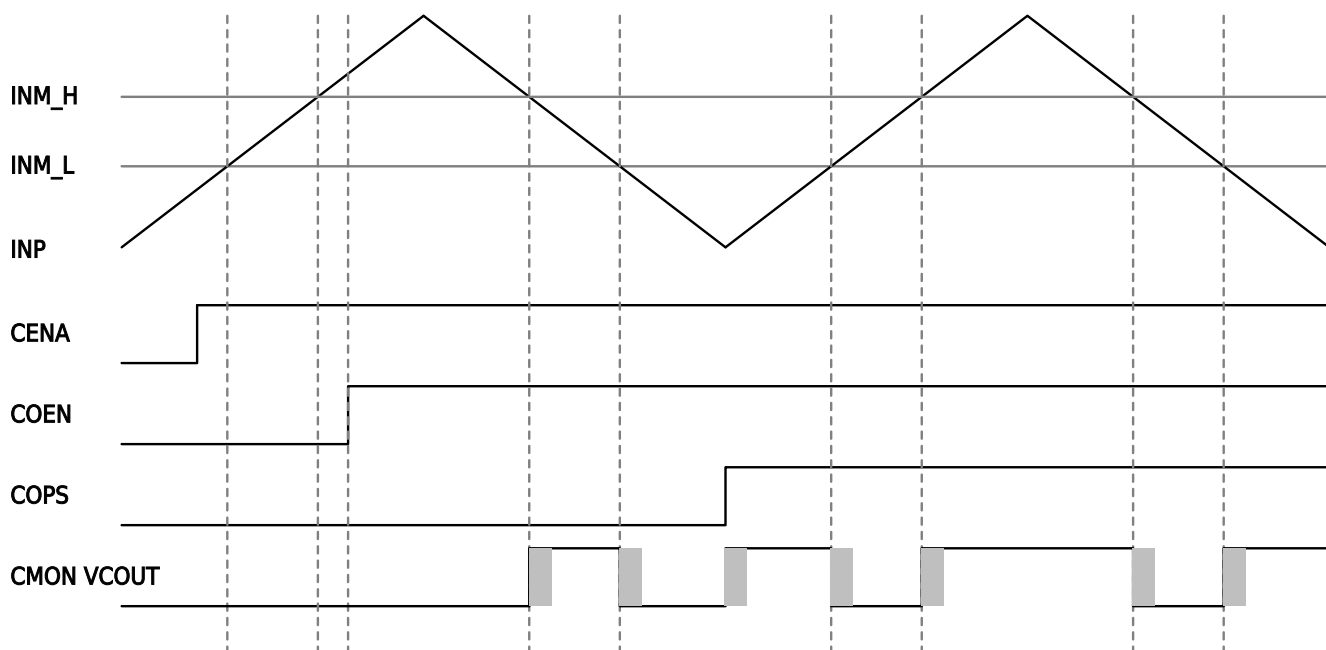


图 17-3 窗口比较示意图

如上图所示，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.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。

17.3.5 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.CSMD 位选择扫描正端或者负端输入，然后设置 CMP_SCCR.SISL 位选择具体参与扫描的正/负端输入。CMP_SCCR.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.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。

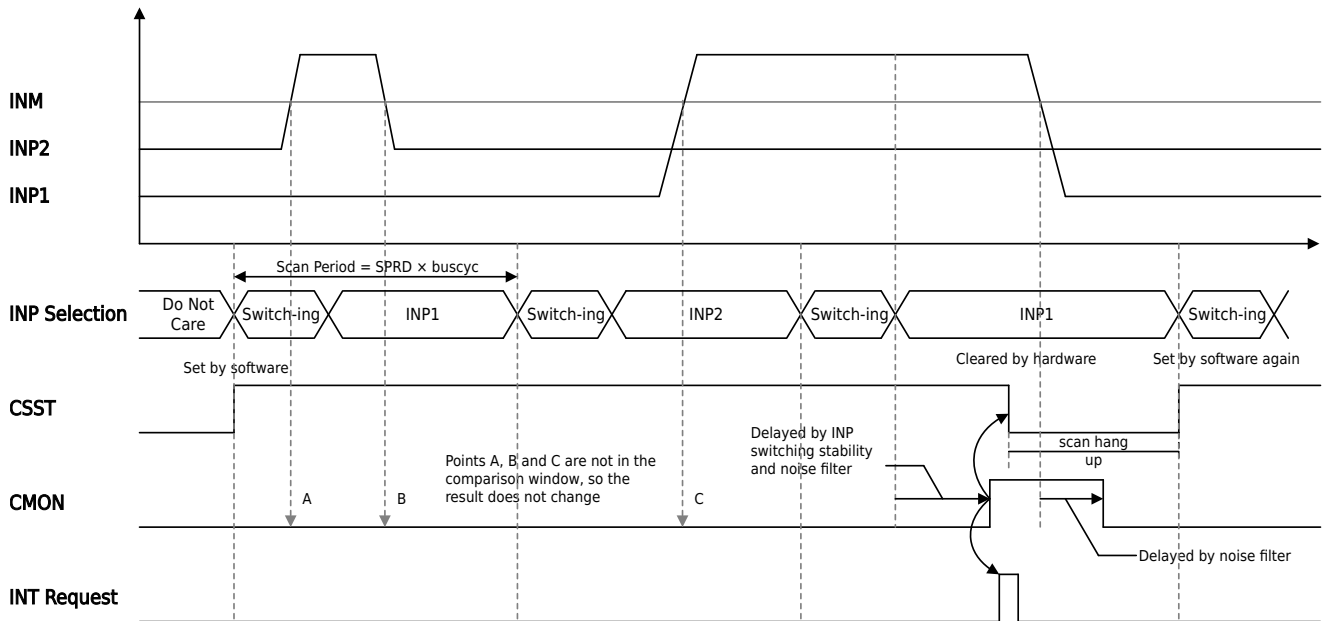


图 17-4 输入扫描示意图

上图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：

- 步骤 1. 设定 CMP_MDR.CSMD 为 0b10 或者 0b11；
- 步骤 2. CSMD 设为 0b10 时，设定 CMP_PMSR.RVSL，选择一个 INM 输入；
CSMD 设为 0b11 时，设定 CMP_PMSR.CVSL，选择一个 INP 输入；
- 步骤 3. 设定 CMP_SCCR.SISL，选择扫描对象端口；
设定 CMP_SCCR.SSTB，设置输出稳定时间；
设定 CMP_SCCR.SPRD，设置扫描周期；
- 步骤 4. 设定 CMP_FIR.EDGS，选择检测上升沿；
设定 CMP_FIR.CIEN，允许中断；
- 步骤 5. 设定 CMP_MDR.CENA，启动比较器；
- 步骤 6. 等待稳定时间 t_{CMP} （具体数值请参见《数据手册》的“电气特性”）；
- 步骤 7. 设定 CMP_OCR.COEN，允许比较器输出；
- 步骤 8. 设定 CMP_MDR.CSST，启动扫描。



说明

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。

17.3.6 窗口输出

CMP 可将 Timer4、Timer6、TimerA 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。窗口输出有电平和边沿两种模式。

1. 电平模式

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

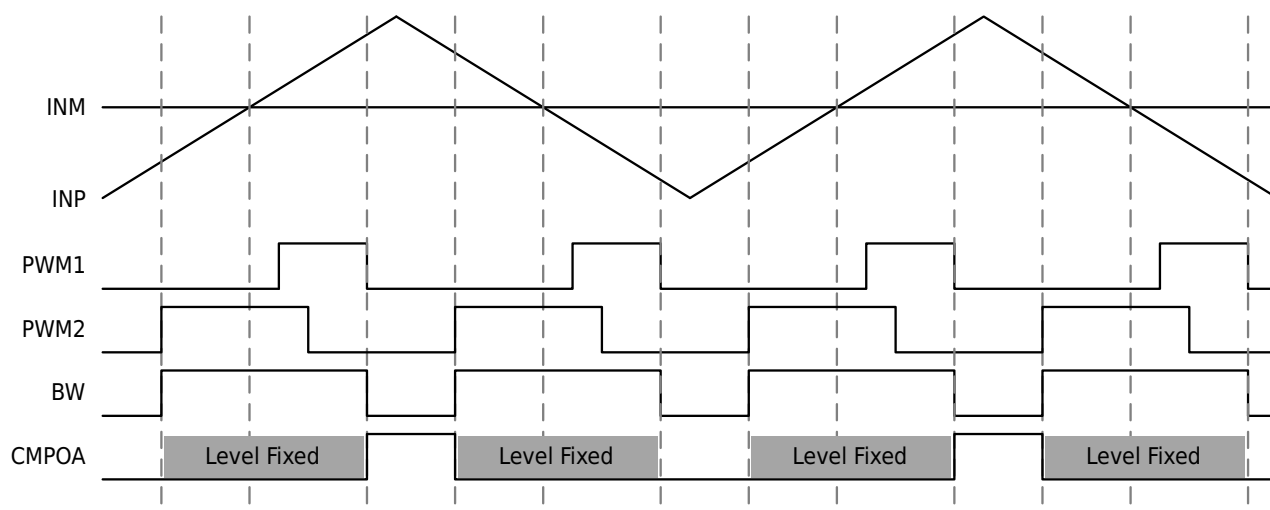


图 17-5 窗口输出示意图一

如上图所示，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。

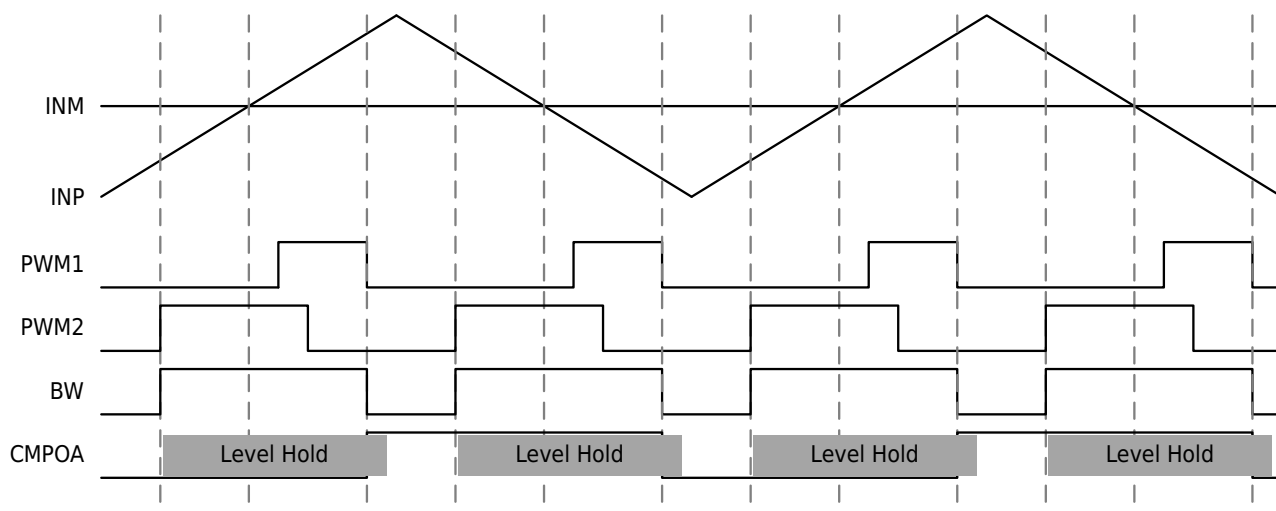


图 17-6 窗口输出示意图二

如上图所示，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_OCR.BWEN 位为 1) 并且选择边沿模式 (CMP_OCR.BWMD 位为 1) 时, 通过 CMP_BWSR1.CTWS 位选择一个或多个定时器 PWM 作为 Blank-Window, 并通过 CMP_BWSR2.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.BWOL 位设定的状态, 并维持由 CMP_BWSR2.MSKW 位设定的 N 个 CMPCLK 周期。

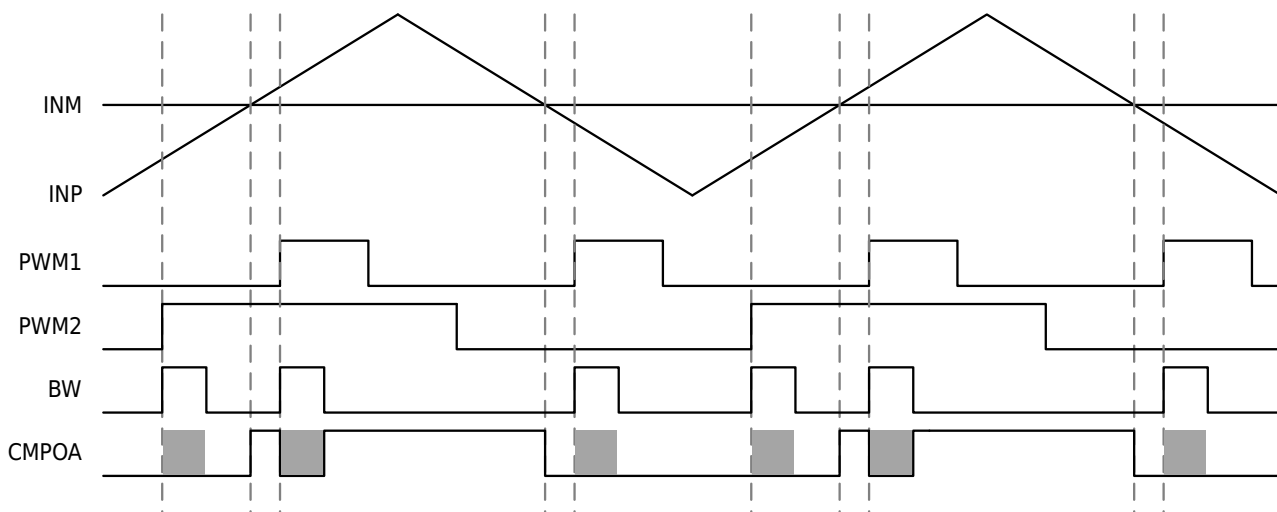


图 17-7 窗口输出示意图三

如上图所示, 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。

17.3.7 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.FCKS 位选择, 当 FCKS 位设为 0b00 时噪声滤波器关闭。

17.3.8 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 VCOUNT。通过 CMP_OCR.CPOE、COPS 位可以分别设定是否允许 VCOUNT 输出以及输出极性 (正相输出或者反相输出)。复位后 VCOUNT 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 VCOUNT 输出功能。

17.3.9 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.CIEN 位设置成 1 并通过 CMP_FIR.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和反相输出后的信号。

CMP 中断还可以唤醒停止低功耗模式。在进入停止模式前必须设定中断有效 (CIEN=1) 并且关闭数字滤波和边沿检测, 中断将在比较器输出由低变高时发生并唤醒停止低功耗模式。如果设定了反相输出, 那么中断将在反相输出由低变高时发生并唤醒停止低功耗模式。系统回到工作模式后, 请先设定中断无效 (CIEN=0) 后再进行后续操作。

外设触发事件和中断请求的产生条件相同, 但必须通过目标外设的硬件触发源寄存器启动或关闭事件触发功能, 与 CIEN 位无关。

17.4 寄存器

17.4.1 寄存器总表

- CMP1 基地址：0x40038800
- CMP2 基地址：0x40038900
- CMP3 基地址：0x40038C00
- CMP4 基地址：0x40038D00

表 17-4 电压比较器 (CMP) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	CMP_MDR	比较器工作模式寄存器	8
0x01	CMP_FIR	比较器滤波和中断寄存器	8
0x02	CMP_OCR	比较器输出控制寄存器	8
0x04	CMP_PMSR	比较器正负端输入选择寄存器	32
0x10	CMP_BWSR1	比较器输出空白窗口设定寄存器 1	32
0x14	CMP_BWSR2	比较器输出空白窗口设定寄存器 2	16
0x18	CMP_SCCR	比较器输入扫描控制寄存器	32
0x1C	CMP_SCMR	比较器输入扫描监视寄存器	32

17.4.2 比较器工作模式寄存器 (CMP_MDR)

Offset	Bit Position							
0x00	7	6	5	4	3	2	1	0
Reset	0x00							
Name	CMON	Reserved		CSST	CSMD		CWDE	CENA
Access	RO			RW	RW		RW	RW

位/位域	标记	位名	功能描述	读写
7	CMON	比较结果监视	普通比较模式： ● 0b0：正端电压<负端电压 ● 0b1：正端电压>负端电压 窗口比较模式（此位只对 CMP2 和 CMP4 有效）： ● 0b0：正端电压<下限负端电压，或者正端电压>上限负端电压 ● 0b1：下限负端电压<正端电压<上限负端电压	RO
6:5	Reserved	保留	-	-
4	CSST	输入扫描启动	● 0b0：停止输入扫描 ● 0b1：开始输入扫描	RW
3:2	CSMD	输入扫描模式	● 0b0x：关闭输入扫描 ● 0b10：扫描正端输入 INP ● 0b11：扫描负端输入 INM	RW
1	CWDE	窗口比较选择	● 0b0：普通比较模式 ● 0b1：窗口比较模式  说明 此位只存在于 CMP2 和 CMP4 中。设为窗口比较模式后，对窗口比较结果的滤波、边沿检测以及输出等控制均由 CMP2 和 CMP4 完成。	RW
0	CENA	比较器工作许可	● 0b0：电压比较器停止 ● 0b1：电压比较器工作  说明 每次将 CENA 由 0 设为 1 后，需要等待约 300ns 的工作稳定时间才可以进行后续操作。	RW

17.4.3 比较器滤波和中断寄存器 (CMP_FIR)

Offset	Bit Position							
0x01	7	6	5	4	3	2	1	0
Reset	0x00							
Name	CRF	CFF	EDGS		CIEN	FCKS		
Access	RW	RW	RW		RW	RW		

位/位域	标记	位名	功能描述	读写
7	CRF	比较器输出上升沿标志	写 1 清除标志, 写 0 无效 读取: 0b0: 未检测到比较器输出 COUT 的上升沿 0b1: 检测到比较器输出 COUT 的上升沿	RW
6	CFF	比较器输出下降沿标志	写 1 清除标志, 写 0 无效 读取: 0b0: 未检测到比较器输出 COUT 的下降沿 0b1: 检测到比较器输出 COUT 的下降沿	RW
5:4	EDGS	比较器输出边沿检测选择	0b00: 检测比较器输出 COUT 的高电平 0b01: 检测比较器输出 COUT 的上升沿 0b10: 检测比较器输出 COUT 的下降沿 0b11: 检测比较器输出 COUT 的上升沿和下降沿  说明 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
3	CIEN	比较器中断许可	0b0: 禁止比较器中断 0b1: 允许比较器中断	RW
2:0	FCKS	比较器结果滤波采样选择	0b000: 不使用噪声滤波器 0b001: 使用噪声滤波器, 通过 PCLK0 进行采样 0b010: 使用噪声滤波器, 通过 PCLK0/2 进行采样 0b011: 使用噪声滤波器, 通过 PCLK0/4 进行采样 0b100: 使用噪声滤波器, 通过 PCLK0/8 进行采样 0b101: 使用噪声滤波器, 通过 PCLK0/16 进行采样 0b110: 使用噪声滤波器, 通过 PCLK0/32 进行采样 0b111: 使用噪声滤波器, 通过 PCLK0/64 进行采样  说明 请在比较器输出禁止 (即 CMP_OCR.COEN 为 0) 时改写 FCKS[2:0]。FCKS[2:0]从“0b000”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

17.4.4 比较器输出控制寄存器 (CMP_OCR)

Offset	Bit Position							
0x02	7	6	5	4	3	2	1	0
Reset	0x00							
Name	BWOL		BWMD	BWEN	Reserved	CPOE	COPS	COEN
Access	RW		RW	RW		RW	RW	RW

位/位域	标记	位名	功能描述	读写
7:6	BWOL	Blank-Window 输出电平	0b00: Blank-Window 有效 (输出关闭) 期间比较器输出低电平 0b01: Blank-Window 有效 (输出关闭) 期间比较器输出高电平 0b1x: Blank-Window 有效 (输出关闭) 期间比较器输出保持	RW
5	BWMD	Blank-Window 模式	0b0: 电平模式 0b1: 边沿模式	RW
4	BWEN	Blank-Window 使能	0b0: 禁止 0b1: 使能	RW
3	Reserved	保留	-	-
2	CPOE	VCOUT 输出允许	0b0: 禁止 VCOUT 输出 0b1: 允许 VCOUT 输出	RW
1	COPS	比较器输出极性选择	0b0: 比较器正相输出 0b1: 比较器反相输出  说明 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
0	COEN	比较器输出允许	0b0: 禁止比较器输出 (比较器输出固定成低电平) 0b1: 允许比较器输出	RW

17.4.5 比较器正负端输入选择寄存器 (CMP_PMSR)

Offset	Bit Position																															
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved												CVSL		Reserved												RVSL					
Access													RW														RW					

位/位域	标记	位名	功能描述	读写
31:20	Reserved	保留	-	-
19:16	CVSL	正端输入选择	选择电压比较器的正端输入电压，所选输入详见 CMP 正端输入一览表： ● 0b0000：无正端电压输入 ● 0b0001：选择 INP1 作为正端电压 ● 0b0010：选择 INP2 作为正端电压 ● 0b0100：选择 INP3 作为正端电压 ● 0b1000：选择 INP4 作为正端电压 ● 其他：禁止设置  说明 当扫描正端输入有效（即 CMP_MDR.CSMD=0b10）时，本寄存器设定无效。 请在比较器输出禁止（即 CMP_OCR.COEN 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 CMP_OCR.COEN 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。	RW
15:4	Reserved	保留	-	-
3:0	RVSL	负端输入选择	选择电压比较器的负端输入电压，所选输入详见 CMP 负端输入一览表： ● 0b0000：无负端电压输入 ● 0b0001：选择 INM1 作为负端电压 ● 0b0010：选择 INM2 作为负端电压 ● 0b0100：选择 DAO1 作为负端电压 ● 0b1000：选择 DAO2 作为负端电压 ● 其他：禁止设置  说明 当扫描负端输入有效（即 CMP_MDR.CSMD=0b11）时，本寄存器设定无效。 请在比较器输出禁止（即 CMP_OCR.COEN 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 CMP_OCR.COEN 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。	RW

17.4.6 比较器输出空白窗口设定寄存器 1 (CMP_BWSR1)

Offset	Bit Position																															
0x10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	CTWP15	CTWP14	CTWP13	CTWP12	CTWP11	CTWP10	CTWP9	CTWP8	CTWP7	CTWP6	CTWP5	CTWP4	CTWP3	CTWP2	CTWP1	CTWP0	CTWS15	CTWS14	CTWS13	CTWS12	CTWS11	CTWS10	CTWS9	CTWS8	CTWS7	CTWS6	CTWS5	CTWS4	CTWS3	CTWS2	CTWS1	CTWS0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:16	CTWPn (n=15~0)	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) : 0b0: 窗口信号为高电平时比较器输出关闭 0b1: 窗口信号为低电平时比较器输出关闭	RW
15:0	CTWSn (n=15~0)	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 信号详见下表: 0b0: 不选择定时器 PWM 输出作为窗口信号 0b1: 选择定时器 PWM 输出作为窗口信号	RW

表 17-15 定时器窗口 PWM 一览表

CTWS Bit	CMP1	CMP2	CMP3	CMP4
15	TMR4_1_OXL	TMR4_1_OXL	TMR4_2_OXL	TMR4_2_OXL
14	TMR4_1_OXH	TMR4_1_OXH	TMR4_2_OXH	TMR4_2_OXH
13	TMR4_1_OWL	TMR4_1_OWL	TMR4_2_OWL	TMR4_2_OWL
12	TMR4_1_OWH	TMR4_1_OWH	TMR4_2_OWH	TMR4_2_OWH
11	TMR4_1_OVL	TMR4_1_OVL	TMR4_2_OVL	TMR4_2_OVL
10	TMR4_1_OVH	TMR4_1_OVH	TMR4_2_OVH	TMR4_2_OVH
9	TMR4_1_OUL	TMR4_1_OUL	TMR4_2_OUL	TMR4_2_OUL
8	TMR4_1_OUH	TMR4_1_OUH	TMR4_2_OUH	TMR4_2_OUH
7	TMR6_2_PWMA	TMR6_2_PWMA	TMR6_2_PWMB	TMR6_2_PWMB
6	TMR6_1_PWMA	TMR6_1_PWMA	TMR6_1_PWMB	TMR6_1_PWMB
5	TMRA_2_PWM3	TMRA_4_PWM3	TMRA_3_PWM3	TMRA_4_PWM3
4	TMRA_2_PWM2	TMRA_4_PWM2	TMRA_3_PWM2	TMRA_4_PWM2
3	TMRA_2_PWM1	TMRA_4_PWM1	TMRA_3_PWM1	TMRA_4_PWM1
2	TMRA_1_PWM3	TMRA_3_PWM3	TMRA_1_PWM3	TMRA_2_PWM3
1	TMRA_1_PWM2	TMRA_3_PWM2	TMRA_1_PWM2	TMRA_2_PWM2
0	TMRA_1_PWM1	TMRA_3_PWM1	TMRA_1_PWM1	TMRA_2_PWM1

17.4.7 比较器输出空白窗口设定寄存器 2 (CMP_BWSR2)

Offset	Bit Position															
0x14	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved						TWEG		MSKW							
Access							RW		RW							

位/位域	标记	位名	功能描述	读写
15:10	Reserved	保留	-	-
9:8	TWEG	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。 0b00: 无有效边沿 0b01: 窗口信号的上升沿处比较器输出关闭 0b10: 窗口信号的下降沿处比较器输出关闭 0b11: 窗口信号的变化沿处比较器输出关闭	RW
7:0	MSKW	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。 可设置范围是 0~255。 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

17.4.8 比较器输入扫描控制寄存器 (CMP_SCCR)

Offset	Bit Position															
0x18	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reset	0x00000000															
Name	Reserved		SSTB				SPRD				Reserved					
Access			RW				RW									

位/位域	标记	位名	功能描述	读写
31:30	Reserved	保留	-	-
29:24	SSTB	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
23:16	SPRD	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
15:4	Reserved	保留	-	-
3:0	SISL	输入选择	正端输入扫描时，Bit3~Bit0 分别对应 CMP 的正端输入 INP4~INP1； 负端输入扫描时，Bit3~Bit0 分别对应 CMP 的负端输入 INM4~INM1。 ● 0b0：输入不作为扫描对象 ● 0b1：输入作为扫描对象  注意 如需改写 SISL，请先停止扫描。	RW

17.4.9 比较器输入扫描监视寄存器 (CMP_SCMR)

Offset	Bit Position																															
0x1C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved												CVST				Reserved												RVST			
Access													RO																RO			

位/位域	标记	位名	功能描述	读写
31:20	Reserved	保留	-	-
19:16	CVST	CMP 正端输入状态	用于在输入扫描时监视 CMP 的输入状态，Bit19~Bit16 分别对应 CMP 的正端输入 INP4~INP1： ● 0b0：不在比较 INPn (n=4~1) ● 0b1：正在比较 INPn (n=4~1)	RO
15:4	Reserved	保留	-	-
3:0	RVST	CMP 负端输入状态	用于在输入扫描时监视 CMP 的输入状态，Bit3~Bit0 分别对应 CMP 的负端输入 INM4~INM1： ● 0b0：不在比较 INMn (n=4~1) ● 0b1：正在比较 INMn (n=4~1)	RO

17.5 注意事项

17.5.1 模块停止功能

CMP 有模块停止功能，通过设定模块停止寄存器可将模块的数字部分关闭。

CMP1 和 CMP2 共用一个模块停止控制位 PWC_FCG3.Bit8。

CMP3 和 CMP4 共用一个模块停止控制位 PWC_FCG3.Bit9。

所有比较通道初始均为停止状态，设定模块工作时才可以访问各自的寄存器。相关寄存器设定请参见《电源控制 (PWC)》章节“动作模式与低功耗模式”。

17.5.2 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.CENA 位清 0。

17.5.3 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.CENA 位清 0。

18 模数转换模块 (ADC)

18.1 简介

12 位 ADC 是一种采用逐次逼近方式的模拟数字转换器。本 MCU 搭载 3 个 ADC 单元，单元 1 最大支持 16 个通道，单元 2 最大支持 8 个通道，单元 3 最大支持 12 个通道，可以转换来自外部引脚和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。

18.2 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：2.5Mps ($PCLK2=60MHz$ ，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
 - ▶ 最大 24 个外部模拟输入
 - ▶ 1 个内部模拟信号：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
- ▶ 上述所有事件输出都可启动 DMA

18.3 功能说明

18.3.1 功能框图

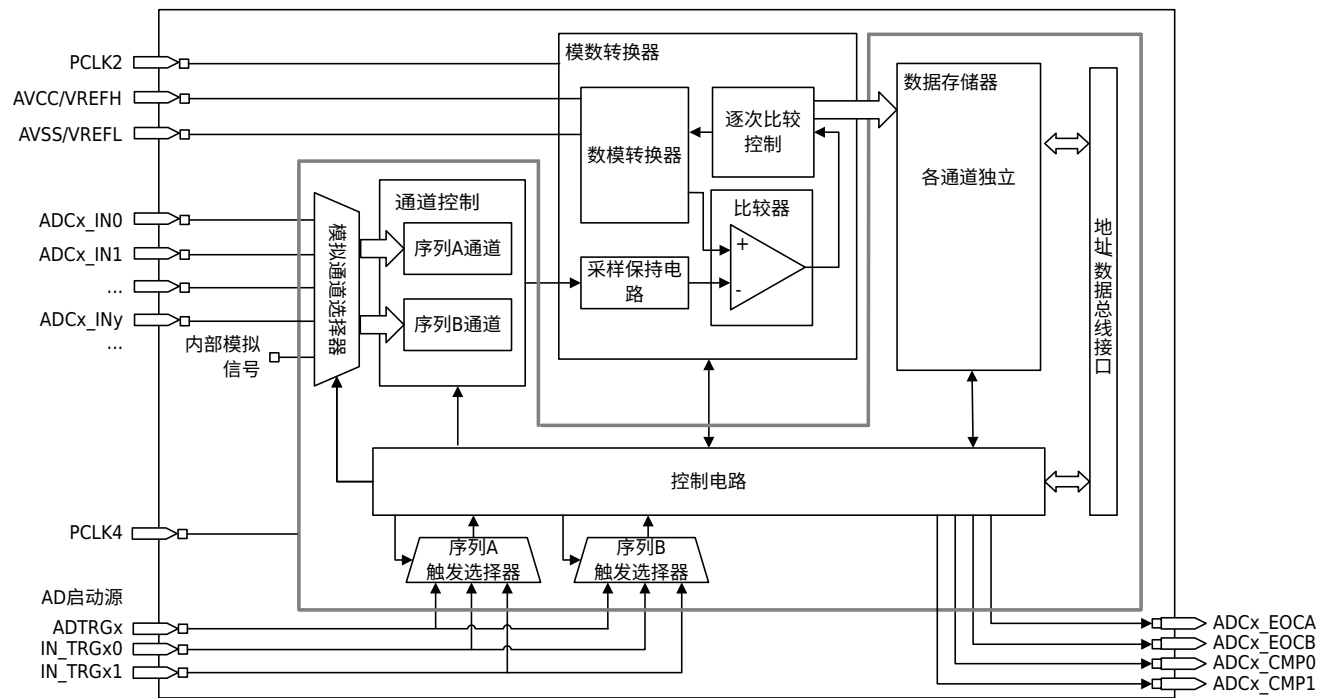


图 18-1 ADC 框图

本芯片搭载了 3 个 ADC 单元，每个单元的配置有所不同，具体参考下表：

表 18-1 各 ADC 单元规格

项目		单元 1 (ADC1)	单元 2 (ADC2)	单元 2 (ADC3)
电源/基准电压		AVCC/VREFH		
		AVSS/VREFL		
模拟通道*1	CH0	ADC1_IN0	ADC12_IN4	ADC3_IN0
	CH1	ADC1_IN1	ADC12_IN5	ADC3_IN1
	CH2	ADC1_IN2	ADC123_IN6	ADC3_IN2
	CH3	ADC1_IN3	ADC123_IN7	ADC3_IN3
	CH4	ADC12_IN4	ADC123_IN8	ADC3_IN4
	CH5	ADC12_IN5	ADC123_IN9	ADC3_IN5
	CH6	ADC123_IN6	ADC12_IN10	ADC123_IN6
	CH7	ADC123_IN7	ADC12_IN11/ 内部模拟信号	ADC123_IN7
	CH8	ADC123_IN8	-	ADC123_IN8

项目		单元 1 (ADC1)	单元 2 (ADC2)	单元 2 (ADC3)
模拟通道*1	CH9	ADC123_IN9	-	ADC123_IN9
	CH10	ADC12_IN10	-	ADC3_IN10
	CH11	ADC12_IN11	-	ADC3_IN11/ 内部模拟信号
	CH12	ADC1_IN12	-	-
	CH13	ADC1_IN13	-	-
	CH14	ADC1_IN14	-	-
	CH15	ADC1_IN15/ 内部模拟信号	-	-
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



说明

ADC 中虚拟通道 CH0~CH15 与物理通道 ADCx_INy（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。

18.3.2 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 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。

18.3.3 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的物理通道间的映射。虚拟通道是指 ADC 模块中假定的通道，如寄存器 ADC_CHSELRA 设置为 0x1 表示序列 A 选择转换 CH0，这个 CH0 就是虚拟通道，而寄存器 ADC_DR0，是虚拟通道 CH0 的转换结果寄存器。物理通道是指实际存在的模拟通道，即外部引脚的模拟输入 ADCx_INy。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。



说明

本章中若无特别说明，通道 n 或 CHn 均表示虚拟通道。

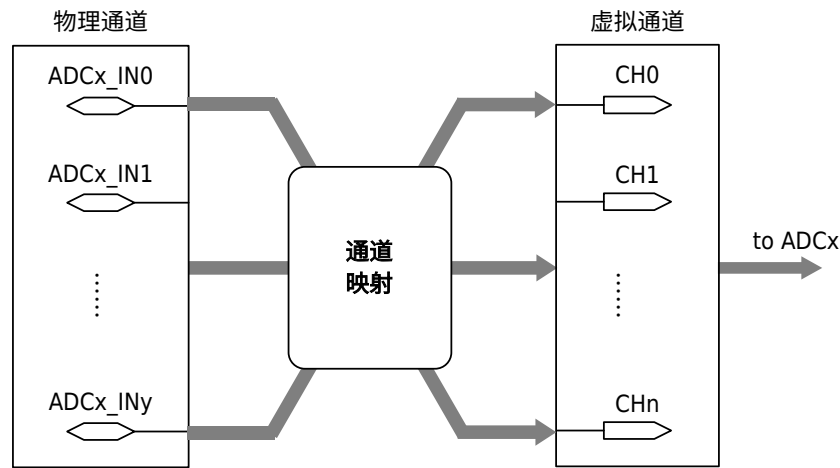


图 18-2 通道映射示意图

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，ADC_CHSELRB 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，内部模拟信号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=7，x=3 时 y=11) 共用一个 CH。设置 ADC_CHMUX 同时改变内部模拟信号和 ADCx_INy 与 CHn 的映射关系。当需要转换内部模拟信号或 ADCx_INy 时，需要先设置扩展通道寄存器 ADC_EXCHSELR，然后将 ADC_CHSELRA/ADC_CHSELRB 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，表示转换内部模拟信号。

复位后内部模拟信号是处于关闭状态，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，允许内部模拟信号输出。

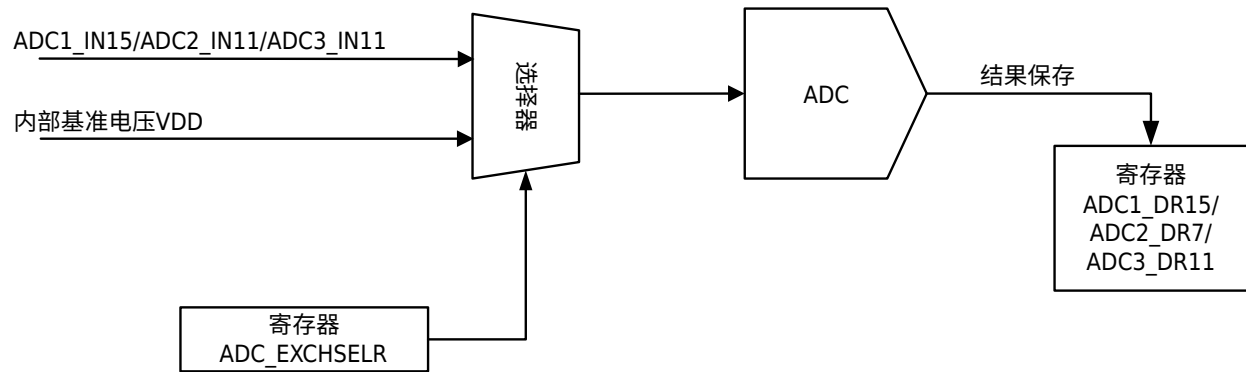


图 18-3 内部模拟信号选择

另外，大部分的物理通道可以输入到多个 ADC 单元，如 ADC123_IN6 可输入到 ADC1、ADC2、ADC3 三个单元，ADC12_IN4 可输入 ADC1、ADC2 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。

18.3.4 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_TRGx0/1。其中，端口 ADTRGx 下降沿输入有效。IN_TRGx0/1 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。

18.3.5 序列 A 单次扫描模式

ADC 控制寄存器 ADC_CR0.MS 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.EOCAL 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。

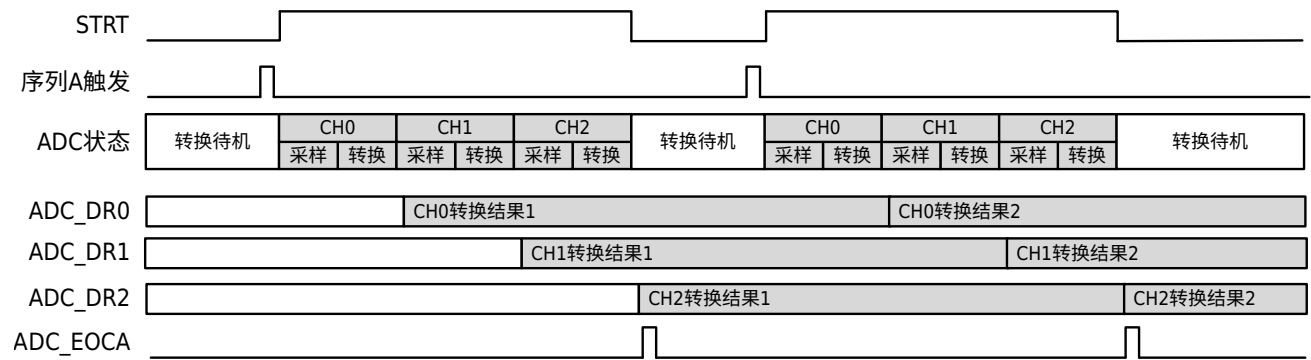


图 18-4 序列 A 单次扫描模式

序列 A 单次扫描模式的软件流程：

- 步骤 1.** 确认 ADC_STR.STRT 为 0，ADC 处于转换待机状态；
- 步骤 2.** ADC 控制寄存器 ADC_CR0.MS 设为 0b000 选择序列 A 单次扫描模式；
- 步骤 3.** 设置序列 A 通道选择寄存器 ADC_CHSELRA；
- 步骤 4.** 设置采样时间寄存器 ADC_SSTR；
- 步骤 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；
- 步骤 6.** 查询序列 A 转换结束标志位 EOCAL；
- 步骤 7.** 读取各通道数据寄存器 ADC_DR；
- 步骤 8.** 写 0 清除 EOCAL 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。

18.3.6 序列 A 连续扫描模式

ADC 控制寄存器 ADC_CR0.MS 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。

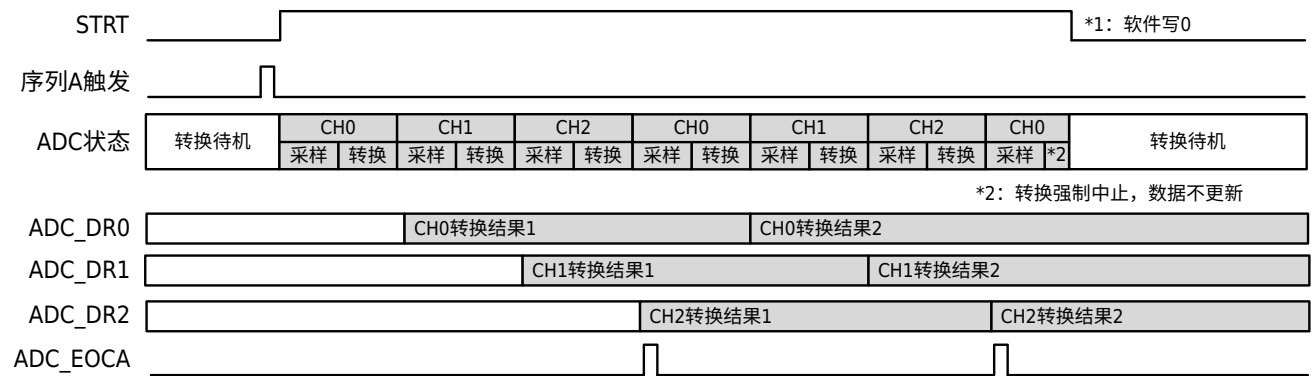


图 18-5 连续扫描

序列 A 连续扫描模式的软件流程：

- 步骤 1. 确认 ADC_STR.STRT 为 0，ADC 处于转换待机状态；
 - 步骤 2. ADC 控制寄存器 ADC_CR0.MS 设为 0b001 选择序列 A 连续扫描模式；
 - 步骤 3. 设置序列 A 通道选择寄存器 ADC_CHSELRA；
 - 步骤 4. 设置采样时间寄存器 ADC_SSTR；
 - 步骤 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；
 - 步骤 6. 查询序列 A 转换结束标志位 EOCAF；
 - 步骤 7. 读取各通道数据寄存器 ADC_DR；
 - 步骤 8. 写 0 清除 EOCAF 标志位，为下次转换准备；
 - 步骤 9. 不需要继续转换时，对 START 位写 0，并读 START 确认为 0 以判断 ADC 进入转换待机状态。
- 上面的步骤 6~8 的查询方式也可以替换成中断方式，利用 ADC_EOCA 中断来处理转换数据。或者利用 ADC_EOCA 事件启动 DMA 读取数据。

 **注意**

由于是连续转换，每次扫描的间隔比较短，特别是只选择 1 个通道转换时。推荐使用 ADC_EOCA 事件启动 DMA 读取数据，避免查询方式下处理不及时导致数据丢失。

18.3.7 序列 A 数据缓冲模式

ADC 控制寄存器 ADC_CR0.MS 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.EOCAF 置 1，并产生序列 A 转换结束事件 ADC_EOCA，可以用此事件启动 DMA。若 ADC_ICR.EOCAIEN 为 1，中断许可的状态下，同时还产生序列 A 转换结束中断请求。

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。

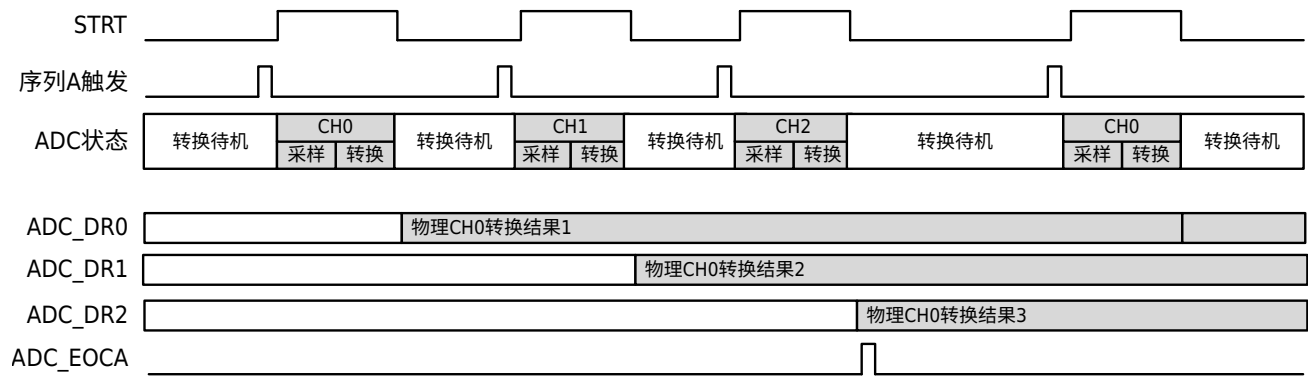


图 18-6 数据缓冲模式

序列 A 数据缓冲模式的软件流程：

- 步骤 1. 确认 ADC_STR.STRT 为 0，ADC 处于转换待机状态；
- 步骤 2. ADC 控制寄存器 ADC_CR0.MS 设为 0b100 选择序列 A 数据缓冲模式；
- 步骤 3. 设置序列 A 通道选择寄存器 ADC_CHSELRA；
- 步骤 4. 设置通道映射控制寄存器 ADC_CHMUXR；
- 步骤 5. 设置采样时间寄存器 ADC_SSTR；
- 步骤 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；
- 步骤 7. 查询单次转换结束 ADC_STR.STRT=0；
- 步骤 8. 重复步骤 6~7 直到所有通道转换结束；
- 步骤 9. 查询序列 A 转换结束标志位 EOCAF；
- 步骤 10. 读取各通道数据寄存器 ADC_DR；
- 步骤 11. 写 0 清除 EOCAF 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_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。

18.3.8 双序列扫描模式

ADC 控制寄存器 ADC_CR0.MS 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=0b010 时，序列 A 和 B 等效于两个独立的单次扫描序列。MS=0b011 序列 A 为连续扫描模式，B 为单次扫描模式。MS=0b110 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 选择触发源，由 ADC_CHSELRA 选择转换的通道。序列 B 由 ADC_TRGSR.TRGSELB 选择触发源，由 ADC_CHSELRB 选择转换的通道。

当序列 A 全部通道转换结束时，序列 A 转换结束标志位 ADC_ISR.EOCAF 置 1，并产生序列 A 转换结束事件 ADC_EOCA，若 ADC_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.EOCBF 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 18-2 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	

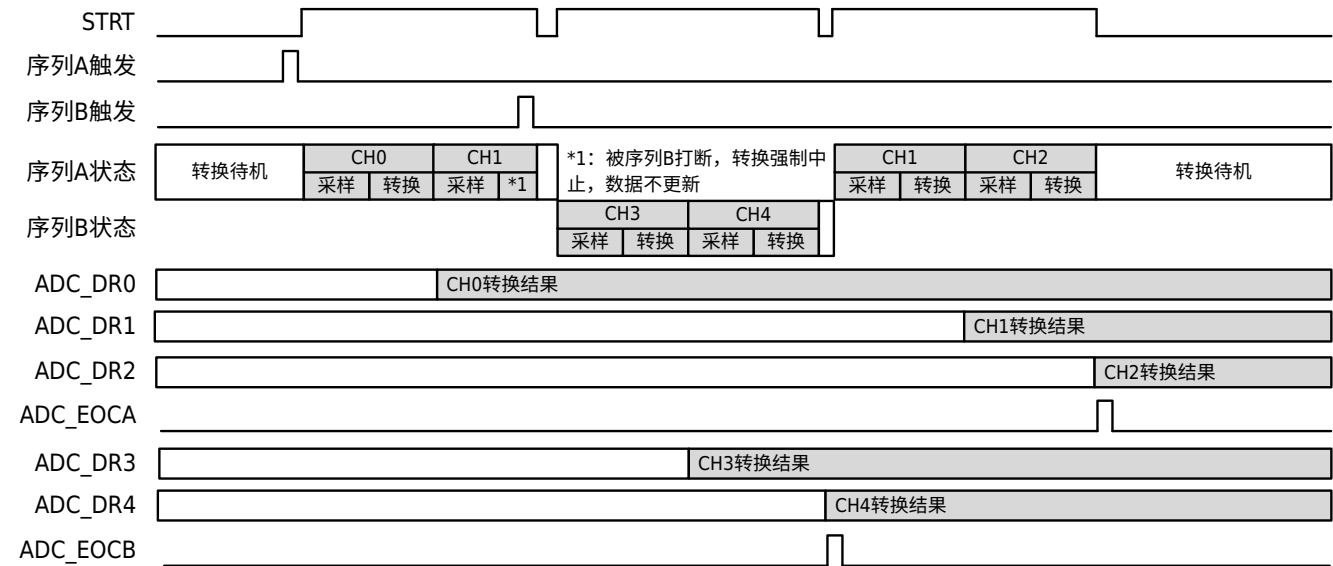


图 18-7 双序列扫描模式 (序列 A 从被中断通道重新启动)

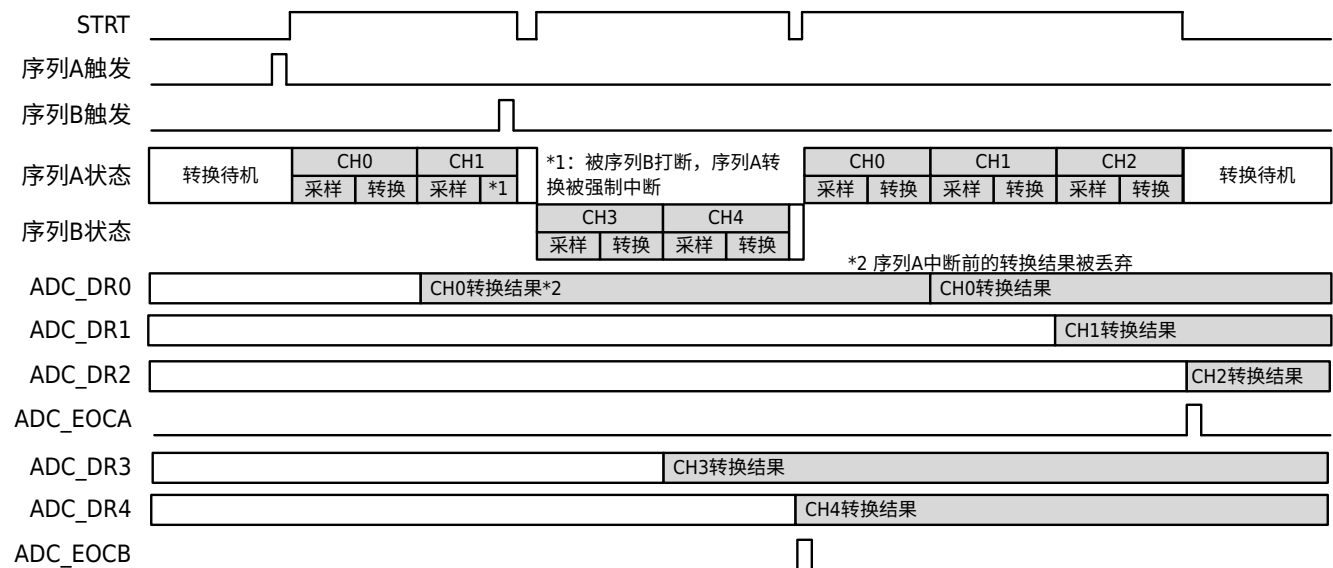


图 18-8 双序列扫描模式 (序列 A 从第一个通道重新启动)

双序列扫描模式的软件流程：

- 步骤 1. 确认 ADC_STR.STRT 为 0，ADC 处于转换待机状态；
- 步骤 2. ADC 控制寄存器 ADC_CR0.MS 设为 0b010 或 0b011 选择双序列扫描模式；
- 步骤 3. 设置寄存器 ADC_CR1.RSCHSEL 选择序列 A 被打断后启动方式；
- 步骤 4. 设置序列 A 通道选择寄存器 ADC_CHSELRA；
- 步骤 5. 设置序列 B 通道选择寄存器 ADC_CHSELRB；
- 步骤 6. 设置采样时间寄存器 ADC_SSTR；
- 步骤 7. 设置寄存器 ADC_TRGSR 选择序列 A 和 B 触发条件；
- 步骤 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。

18.3.9 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。

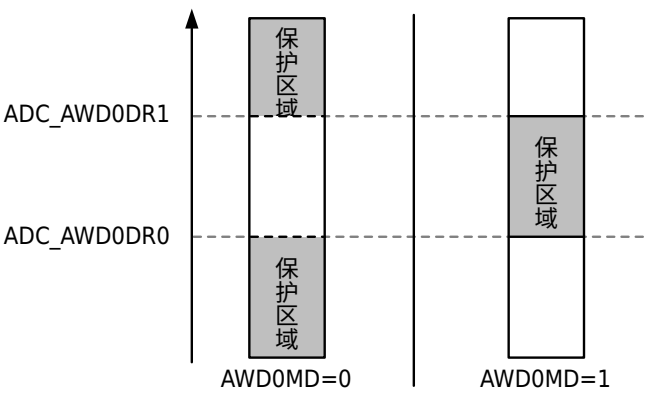


图 18-9 模拟看门狗保护区域（比较模式）

使用模拟看门狗功能的软件流程，以比较窗口 0 为例：

- 步骤 1. 设置阈值寄存器 AD_AWD0DR0、ADC_AWD0DR1；
- 步骤 2. 设置比较通道寄存器 ADC_AWD0CHSR，选择需要比较的通道；
- 步骤 3. 设置 ADC_AWDCR.AWD0MD 选择比较模式；
- 步骤 4. 设置 ADC_AWDCR.AWD0IEN 中断许可位；
- 步骤 5. 设置 ADC_AWDCR.AWD0EN 允许看门狗窗口 0 比较功能；
- 步骤 6. 根据前文，设置扫描模式，启动 ADC 进行转换；
- 步骤 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 寄存器选择组合方式。

18.3.10 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。

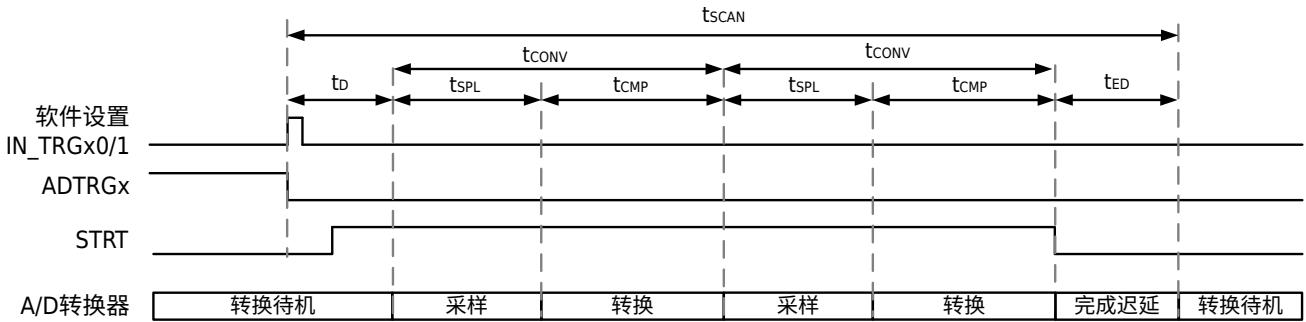


图 18-10 ADC 转换时间示意图

表 18-3 ADC 转换时间

标记	说明		条件			
			同步周边触发	异步周边触发 ⁽¹⁾	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	ADC_SSTRx.SST * 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_{TD}	最小连续触发时间间隔		$\sum t_{CONV} + 6 \text{ PCLK2}$			

说明

1. 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。

18.3.11 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。

- 当 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。

18.3.12 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 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。

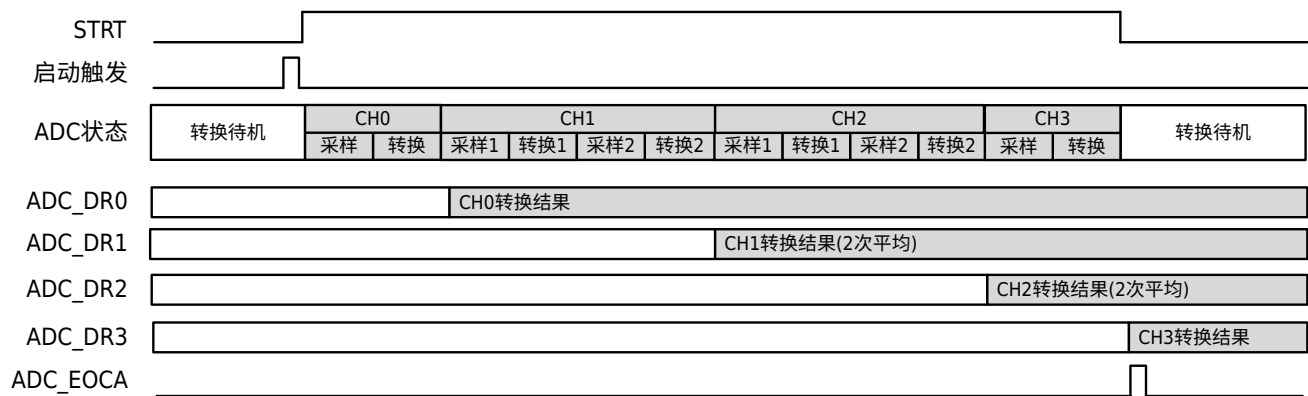


图 18-11 平均功能有效时的转换动作

上图中, 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。

18.3.13 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 16-bit 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 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 设置, 最大可以设置向右移位 8 位。

18.3.14 多 ADC 协同工作模式

在搭载有两个或三个 ADC 单元的芯片上, 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 设置无效。所有 ADC 单元均由 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 单元协同工作, 也可以 ADC1、ADC2、ADC3 三个 ADC 单元协同工作。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 单元，且只触发一次。

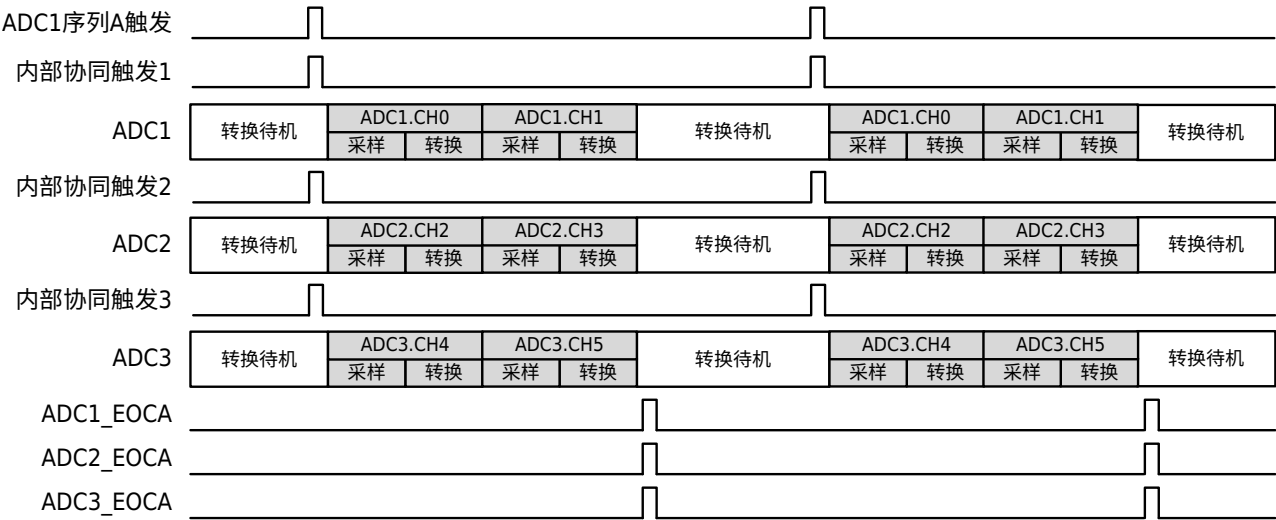


图 18-12 单次并行触发模式（三个 ADC）

**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 单元采样，否则精度不做保证，下同。

本模式的软件设定流程如下：

- 步骤 1.** 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 0，确认协同工作无效。
- 步骤 2.** 设置 ADC1 单元：

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 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
- 步骤 3.** 设置 ADC2 单元：

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，通道选择寄存器 ADC2_CHSELRA，通道采样时间寄存器 ADC2_SSTR
- **说明**

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。
- 步骤 4.** 设置 ADC3 单元（三个 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，通道选择寄存器 ADC3_CHSELRA，通道采样时间寄存器 ADC3_SSTR



说明

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。

- 步骤 5.** 设置协同模式控制寄存器 ADC_SYNCCCR.SYNCMD，写 0b010（ADC1、ADC2 两个 ADC 协同工作）或者写 0b011（ADC1、ADC2、ADC3 三个 ADC 协同工作）；
- 步骤 6.** 协同工作许可寄存器 ADC_SYNCCCR.SYNCEN 写 1，协同工作有效；
- 步骤 7.** 等待 ADC1 序列 A 触发源输入，在 ADC1、ADC2、ADC3 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 单元只触发一次。

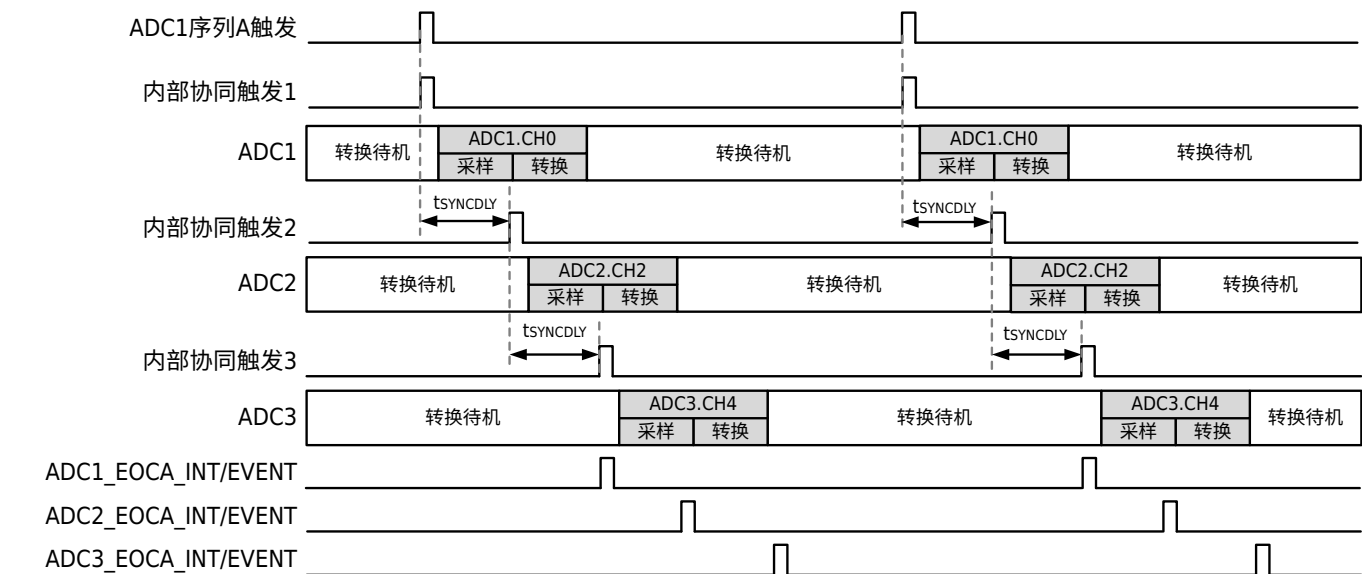


图 18-13 单次延迟触发模式（三个 ADC）



注意

- ADC1 序列 A 触发第一次输入后，ADC3 协同触发发生前，再次输入 ADC1 序列 A 触发将被忽略。
- 若各 ADC 单元转换的是同一模拟通道，需要错开采样时间，即延迟时间 $t_{SYNCDLY}$ 与通道开采样时间 t_{SPL} 需满足： $t_{SYNCDLY} > t_{SPL}$ 。

本模式的软件设定流程如下：

- 步骤 1.** 协同工作许可寄存器 ADC_SYNCCCR.SYNCEN 写 0，确认协同工作无效；
- 步骤 2.** 设置 ADC1、ADC2、ADC3 单元（参考单次平行模式）；
- 步骤 3.** 设置协同模式控制寄存器 ADC_SYNCCCR.SYNCDLY，设置两个 ADC 的启动延迟；
- 步骤 4.** 设置协同模式控制寄存器 ADC_SYNCCCR.SYNCMD，写 0b000（ADC1、ADC2 两个 ADC 协同工作）或者写 0b001（ADC1、ADC2、ADC3 三个 ADC 协同工作）；
- 步骤 5.** 协同工作许可寄存器 ADC_SYNCCCR.SYNCEN 写 1，协同工作有效；
- 步骤 6.** 等待 ADC1 序列 A 触发源输入，在 ADC1、ADC2、ADC3 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 单元，且之后每经过指定延迟之后会再次同时触发所有 ADC 单元。直至用户主动软件停止 ADC1 单元，或者禁止协同工作模式。

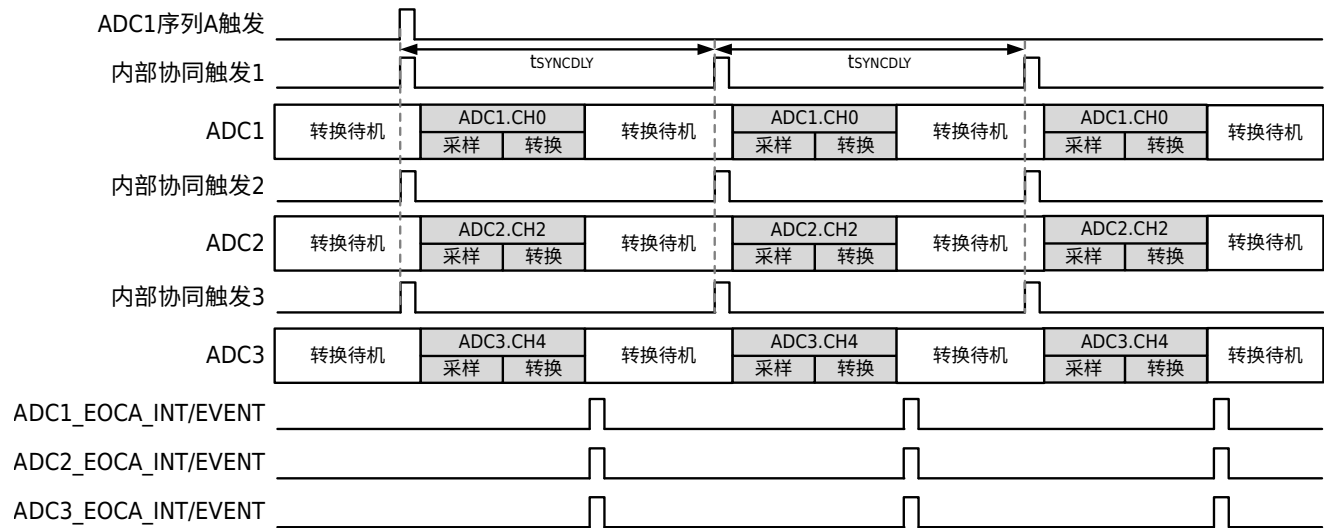


图 18-14 循环并行触发模式 (三个 ADC)



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:

- 步骤 1. 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 0, 确认协同工作无效;
- 步骤 2. 设置 ADC1、ADC2、ADC3 单元, 参考单次并行模式。ADC_CR0.MS 设置为 0b000 (序列 A 单次扫描模式);
- 步骤 3. 设置协同模式控制寄存器 ADC_SYNCCR.SYNCDLY, 设置每次并行触发的延迟;
- 步骤 4. 设置协同模式控制寄存器 ADC_SYNCCR.SYNCMD, 写 0b110 (ADC1、ADC2 两个 ADC 协同工作) 或者写 0b111 (ADC1、ADC2、ADC3 三个 ADC 协同工作);
- 步骤 5. 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 1, 协同工作有效;
- 步骤 6. 等待 ADC1 序列 A 触发源输入, 在 ADC1、ADC2、ADC3 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 单元, 或者禁止协同工作模式。

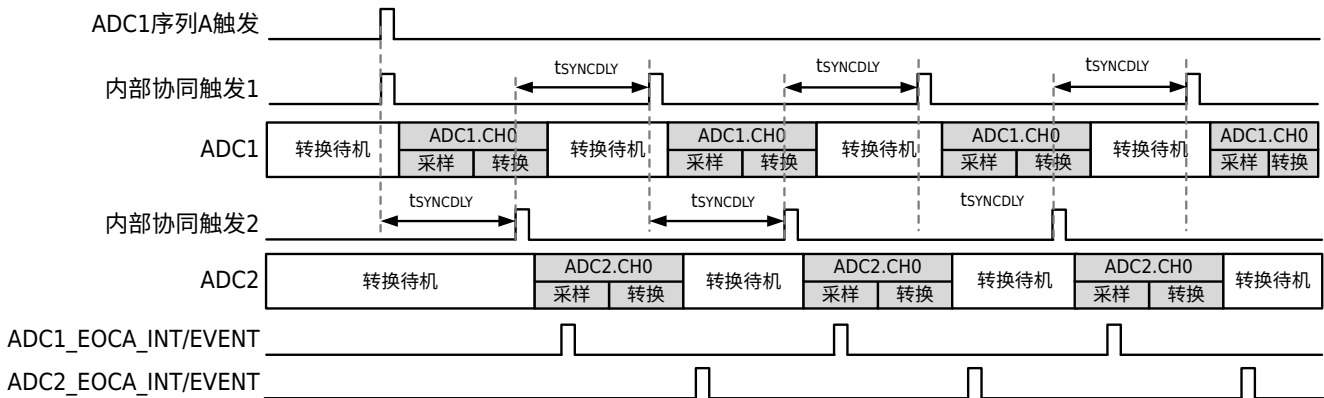


图 18-15 循环延迟触发模式 (两个 ADC)

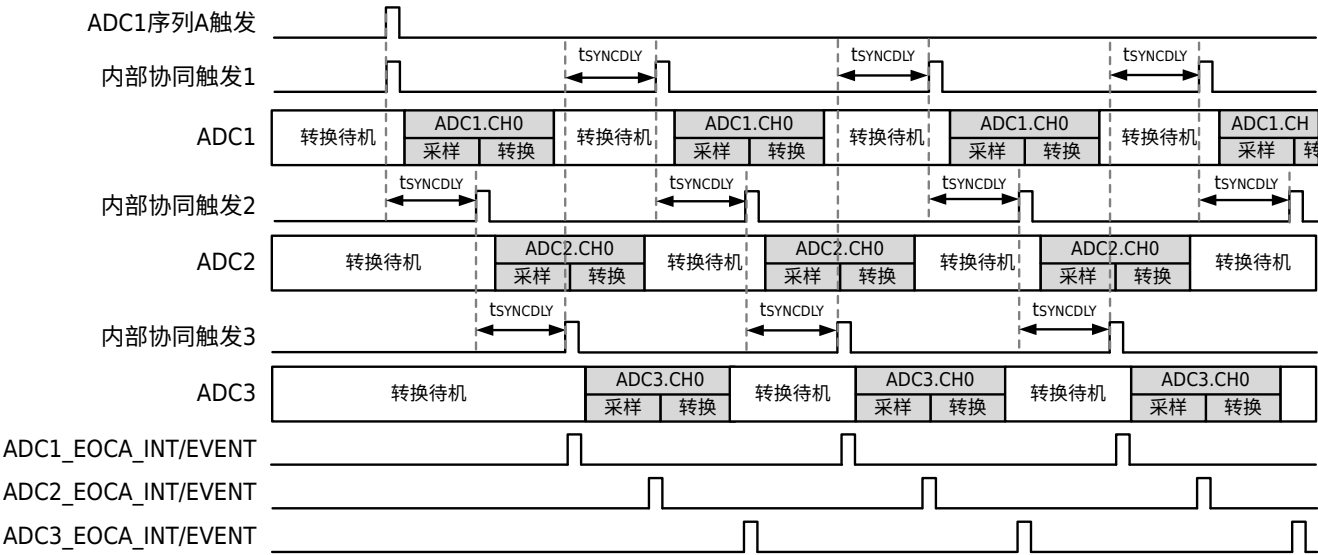


图 18-16 循环延迟触发模式 (三个 ADC)



注意

两个 ADC 协同工作时，延迟时间 $t_{\text{SYNCDELAY}}$ 与一次扫描转换的时间 t_{SCAN} 需满足： $t_{\text{SYNCDELAY}} > t_{\text{SCAN}}/2$ 。三个 ADC 协同工作时需满足： $t_{\text{SYNCDELAY}} > t_{\text{SCAN}}/3$ 。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 $t_{\text{SYNCDELAY}} > t_{\text{SPL}}$ 。

本模式的软件设定流程如下：

- 步骤 1.** 协同工作许可寄存器 ADC_SYNCCCR.SYNCEN 写 0，确认协同工作无效；
- 步骤 2.** 设置 ADC1、ADC2、ADC3 单元，参考循环并行触发模式；
- 步骤 3.** 设置协同模式控制寄存器 ADC_SYNCCCR.SYNCDELAY，设置每次触发的延迟；
- 步骤 4.** 设置协同模式控制寄存器 ADC_SYNCCCR.SYNCMD，写 0b100（ADC1、ADC2 两个 ADC 协同工作）或者写 0b101（ADC1、ADC2、ADC3 三个 ADC 协同工作）；
- 步骤 5.** 协同工作许可寄存器 ADC_SYNCCCR.SYNCEN 写 1，协同工作有效；
- 步骤 6.** 等待 ADC1 序列 A 触发源输入，在 ADC1、ADC2、ADC3 完成转换后处理结果。

18.3.15 中断和事件

ADC 模块可以产生以下几种事件输出，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。

- 序列 A 扫描结束 ADC_EOCA，对应中断许可寄存器 ADC_ICR.EOCAIEN
- 序列 B 扫描结束 ADC_EOCB，对应中断许可寄存器 ADC_ICR.EOCBIEN
- 模拟看门狗窗口 0 比较 ADC_CMP0，对应中断许可寄存器 ADC_AWDCR.AWD0IEN
- 模拟看门狗窗口 1 比较 ADC_CMP1，对应中断许可寄存器 ADC_AWDCR.AWD1IEN

上述所有事件输出，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 的设定请参见《DMA 控制器 (DMA)》章节相关描述。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。

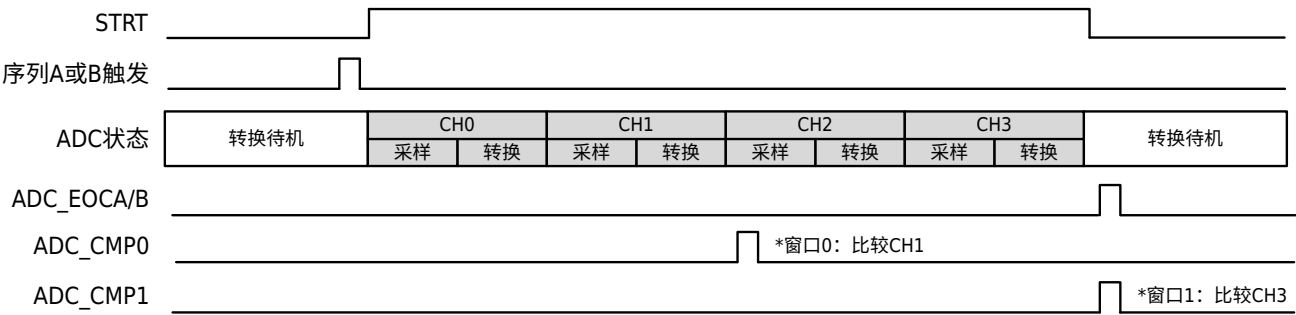


图 18-17 ADC 中断和事件输出时序

18.4 寄存器

18.4.1 寄存器总表

- ADC1 基地址：0x40040000
- ADC2 基地址：0x40040400
- ADC3 基地址：0x40040800

表 18-4 模数转换模块 (ADC) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	ADC_STR	启动寄存器	8
0x02	ADC_CR0	控制寄存器 0	16
0x04	ADC_CR1	控制寄存器 1	16
0x06	ADC_CR2	控制寄存器 2	16
0x0A	ADC_TRGSR	转换开始触发寄存器	16
0x0C	ADC_CHSELRA	通道选择寄存器 A	32
0x10	ADC_CHSELRB	通道选择寄存器 B	32
0x14	ADC_AVCHSELR	平均通道选择寄存器	32
0x18	ADC_EXCHSELR	扩展通道选择寄存器	8
0x20+0x1*x	ADC_SSTRx	采样周期寄存器 ⁽¹⁾	8
0x38	ADC_CHMUXR0	通道映射控制寄存器 0	16
0x3A	ADC_CHMUXR1	通道映射控制寄存器 1	16
0x3C	ADC_CHMUXR2	通道映射控制寄存器 2	16
0x3E	ADC_CHMUXR3	通道映射控制寄存器 3	16
0x44	ADC_ISR	中断状态寄存器	8
0x45	ADC_ICR	中断许可寄存器	8
0x46	ADC_ISCLRR	中断状态复位寄存器	8
0x4C	ADC_SYNCRR	协同模式控制寄存器	16
0x50+0x2*x	ADC_DRx	数据寄存器 ⁽¹⁾	16
0xA0	ADC_AWDCR	模拟看门狗控制寄存器	16
0xA2	ADC_AWDSR	模拟看门狗状态寄存器	8
0xA3	ADC_AWDSCLRR	模拟看门狗状态复位寄存器	8
0xA4+0x8*x+0x2*y	ADC_AWDxDry	模拟看门狗窗口 x 阈值寄存器 (x=0~1, y=0~1)	16

偏移地址	寄存器	描述	位宽
0xA8+0x8*x	ADC_AWDxCHSR	模拟看门狗窗口 x 比较通道选择寄存器 (x=0~1)	8

说明

1. ADC_SSTRx 和 ADC_DRx 中的“x”表示通道，ADC1 中 x=0~15，ADC2 中 x=0~7，ADC3 中 x=0~11。

18.4.2 启动寄存器 (ADC_STR)

Offset	Bit Position							
0x00	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved							STRT
Access								RW

位/位域	标记	位名	功能描述	读写
7:1	Reserved	保留	-	-
0	STRT	ADC 转换开始	● 0b0: 停止转换 ● 0b1: 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  说明 ● 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 ADC 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

18.4.3 控制寄存器 0 (ADC_CR0)



说明

请在 ADC_STR.STRT 为 0 时设置本寄存器。

Offset	Bit Position															
0x02	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved					AVCNT			DFMT	CLREN	ACCSEL	Reserved		MS		
Access						RW			RW	RW	RW			RW		

位/位域	标记	位名	功能描述	读写
15:11	Reserved	保留	-	-
10:8	AVCNT	次数选择	0b000: 连续转换 2 次平均 0b001: 连续转换 4 次平均 0b010: 连续转换 8 次平均 0b011: 连续转换 16 次平均 0b100: 连续转换 32 次平均 0b101: 连续转换 64 次平均 0b110: 连续转换 128 次平均 0b111: 连续转换 256 次平均	RW
7	DFMT	数据格式	0b0: 转换数据右对齐 0b1: 转换数据左对齐 说明 过采样模式下该 Bit 位设置无效。	RW
6	CLREN	数据寄存器自动清除	0b0: 自动清除禁止 0b1: 自动清除许可 说明 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
5:4	ACCSEL	分辨率选择	0b00: 12 位分辨率 0b01: 10 位分辨率 0b10: 8 位分辨率 0b11: 禁止设置 说明 过采样模式下需要设置为 0b00。	RW
3	Reserved	保留	-	-

位/位域	标记	位名	功能描述	读写
2:0	MS	模式选择	● 0b000: 序列 A 单次扫描模式, 序列 B 无效 ● 0b001: 序列 A 连续扫描模式, 序列 B 无效 ● 0b010: 序列 A 单次扫描模式, 序列 B 单次扫描模式 ● 0b011: 序列 A 连续扫描模式, 序列 B 单次扫描模式 ● 0b100: 序列 A 数据缓冲模式, 序列 B 无效 ● 0b110: 序列 A 数据缓冲模式, 序列 B 单次扫描模式 ● 其他: 禁止设置	RW

18.4.4 控制寄存器 1 (ADC_CR1)



说明

请在 ADC_STR.STRT 为 0 时设置本寄存器。

Offset	Bit Position															
0x04	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved												RSCHSEL	Reserved		
Access													RW			

位/位域	标记	位名	功能描述	读写
15:3	Reserved	保留	-	-
2	RSCHSEL	序列 A 重启通道选择	0b0: 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 0b1: 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
1:0	Reserved	保留	-	-

18.4.5 控制寄存器 2 (ADC_CR2)



说明

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

Offset	Bit Position															
0x06	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved			OVSMOD		OVSS			Reserved							
Access				RW		RW										

位/位域	标记	位名	功能描述	读写
15:13	Reserved	保留	-	-
12	OVSMOD	过采样模式	0b0: 普通模式 0b1: 过采样模式	RW
11:8	OVSS	移位选择	0b0000: 不移位 0b0001: 向右移 1 位 0b0010: 向右移 2 位 0b0011: 向右移 3 位 0b0100: 向右移 4 位 0b0101: 向右移 5 位 0b0110: 向右移 6 位 0b0111: 向右移 7 位 0b1000: 向右移 8 位 - 其他: 禁止设置 说明 OVSMOD=0 (普通模式) 时, 该位设置无效。	RW
7:0	Reserved	保留	-	-

18.4.6 转换开始触发寄存器 (ADC_TRGSR)



说明

请在 ADC_STR.STRT 为 0 时设置本寄存器。

Offset	Bit Position															
0x0A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	TRGENB	Reserved				TRGSELB		TRGENA		Reserved				TRGSELA		
Access	RW					RW		RW						RW		

位/位域	标记	位名	功能描述	读写
15	TRGENB	序列 B 触发使能	0b0: 序列 B 片内或外部引脚触发禁止 0b1: 序列 B 片内或外部引脚触发许可 说明 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
14:10	Reserved	保留	-	-
9:8	TRGSELB	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： 0b00: ADTRGx (x=1~3, 代表 ADC 单元编号) 0b01: IN_TRGx0 0b10: IN_TRGx1 0b11: IN_TRGx0 + IN_TRGx1 说明 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
7	TRGENA	序列 A 触发使能	0b0: 序列 A 片内或外部引脚触发禁止 0b1: 序列 A 片内或外部引脚触发许可 说明 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
6:2	Reserved	保留	-	-

位/位域	标记	位名	功能描述	读写
1:0	TRGSELA	序列 A 触发条件选择	序列 A 的触发条件： ● 0b00：ADTRGx（x=1~3，代表 ADC 单元编号） ● 0b01：IN_TRGx0 ● 0b10：IN_TRGx1 ● 0b11：IN_TRGx0 + IN_TRGx1  说明 ADC 空闲中对 ADC_STR.STRT 写 1 软件触发，无视 TRGENA 和 TRGSELA 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

18.4.7 通道选择寄存器 A (ADC_CHSELRA)



说明

请在 ADC_STR.STRT 为 0 时设置本寄存器。

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	CHSELA																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	CHSELA	转换通道选择	序列 A 的通道选择，每个 Bit 代表一个通道，CHSELA[x]代表通道 CHx，可选择任意组合。 0b0：未选择对应通道 0b1：选择对应通道 注意 请不要在序列 A、序列 B 中选择相同的通道。 ADC1 支持通道 0~15，ADC2 支持通道 0~7，ADC3 支持通道 0~11。不支持的通道需要设置为 0。	RW

18.4.8 通道选择寄存器 B (ADC_CHSELRB)



说明

请在 ADC_STR.STRT 为 0 时设置本寄存器。

Offset	Bit Position																															
0x10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	CHSELB																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	CHSELB	转换通道选择	序列 B 的通道选择，每个 Bit 代表一个通道，CHSELB[x]代表通道 CHx，可选择任意组合。 0b0：未选择对应通道 0b1：选择对应通道 注意 只有在双序列扫描模式时才有效。请不要在序列 A、序列 B 中选择相同的通道。 ADC1 支持通道 0~15，ADC2 支持通道 0~7，ADC3 支持通道 0~11。不支持的通道需要设置为 0。	RW

18.4.9 平均通道选择寄存器 (ADC_AVCHSELR)



说明

请在 ADC_STR.STRT 为 0 时设置本寄存器。

Offset	Bit Position																															
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	RW AVCHSEL																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	AVCHSEL	平均通道选择	每个 Bit 代表一个通道，AVCHSEL[x]代表通道 CHx，可选择任意组合。 ● 0b0：未选择对应通道 ● 0b1：选择对应通道 注意 当 AVCHSEL 与 ADC_CHSELRA 或 ADC_CHSELRB 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。 ADC1 支持通道 0~15，ADC2 支持通道 0~7，ADC3 支持通道 0~11。不支持的通道需要设置为 0。	RW

18.4.10 扩展通道选择寄存器 (ADC_EXCHSELR)



说明

请在 ADC_STR.STRT 为 0 时设置本寄存器。

Offset	Bit Position							
0x18	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved							EXCHSEL
Access								RW

位/位域	标记	位名	功能描述	读写
7:1	Reserved	保留	-	-
0	EXCHSEL	扩展通道选择	扩展通道模拟输入源选择： 0b0：选择外部引脚 0b1：选择内部模拟信号 说明 内部模拟信号的设置请参见《电源控制（PWC）》章节相关描述。	RW

18.4.11 采样周期寄存器 (ADC_SSTRx)



说明

请在 ADC_STR.STRT 为 0 时设置本寄存器。

偏移地址 (Offset) : 0x20+0x1*x (“x”表示通道, ADC1 中 x=0~15, ADC2 中 x=0~7, ADC3 中 x=0~11)

Offset	Bit Position							
-	7	6	5	4	3	2	1	0
Reset	0x0B							
Name	SST							
Access	RW							

位/位域	标记	位名	功能描述	读写
7:0	SST	采样周期数	0x00~0x04: 禁止设置 0x05~0xFF: 分别对应 5~255 个 PCLK2 时钟周期 通道 CH0~15 分别由 ADC_SSTRx (x=0~15) 来设置。 说明 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。 $SST \geq (R_{AIN}+R_{ADC}) * C_{ADC} * \ln(2^{N+2}) * f_{ADC} + 1$ 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)。具体信息请参见《数据手册》“电气特性”章节相关描述。	RW



说明

- 内部基准电压的采样时间不要小于 1 μ s。

18.4.12 通道映射控制寄存器 (ADC_CHMUXRx) (x=0~3)



说明

请在 ADC_STR.STRT 为 0 时设置本寄存器。

单元 2 仅配置了 ADC_CHMUXRn (n=0~1)，单元 3 仅配置了 ADC_CHMUXRn (n=0~2)。

ADC_CHMUXR0 寄存器

Offset	Bit Position															
0x38	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x3210															
Name	CH03MUX				CH02MUX				CH01MUX				CH00MUX			
Access	RW				RW				RW				RW			

ADC_CHMUXR1 寄存器

Offset	Bit Position															
0x3A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x7654															
Name	CH07MUX				CH06MUX				CH05MUX				CH04MUX			
Access	RW				RW				RW				RW			

ADC_CHMUXR2 寄存器

Offset	Bit Position															
0x3C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xBA98															
Name	CH11MUX				CH10MUX				CH09MUX				CH08MUX			
Access	RW				RW				RW				RW			

ADC_CHMUXR3 寄存器

Offset	Bit Position															
0x3E	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFEDC															
Name	CH15MUX				CH14MUX				CH13MUX				CH12MUX			
Access	RW				RW				RW				RW			

位/位域	标记	位名	功能描述	读写
15:12	CHxMUX	通道 x 映射选择 (x=0~15)	选择 ADC 单元 CHx 映射对象，设定值选择详见 不同的 ADC 单元 CHx 映射关系表  注意 不存在的通道对应位读出时为 0，写入时写 0。 请不要设置到不存在的模拟输入上。	RW
11:8	CHxMUX			RW
7:4	CHxMUX			RW
3:0	CHxMUX			RW

表 18-30 不同的 ADC 单元 CHx 映射关系表

设定值	ADC1 映射对象	ADC2 映射对象	ADC3 映射对象
0x0	ADC1_IN0	ADC12_IN4	ADC3_IN0
0x1	ADC1_IN1	ADC12_IN5	ADC3_IN1
0x2	ADC1_IN2	ADC123_IN6	ADC3_IN2
0x3	ADC1_IN3	ADC123_IN7	ADC3_IN3
0x4	ADC12_IN4	ADC123_IN8	ADC3_IN4
0x5	ADC12_IN5	ADC123_IN9	ADC3_IN5
0x6	ADC123_IN6	ADC12_IN10	ADC123_IN6
0x7	ADC123_IN7	ADC12_IN11/内部模拟信号	ADC123_IN7
0x8	ADC123_IN8	-	ADC123_IN8
0x9	ADC123_IN9	-	ADC123_IN9
0xA	ADC12_IN10	-	ADC3_IN10
0xB	ADC12_IN11	-	ADC3_IN11/内部模拟信号
0xC	ADC1_IN12	-	-
0xD	ADC1_IN13	-	-
0xE	ADC1_IN14	-	-
0xF	ADC1_IN15/内部模拟信号	-	-

18.4.13 中断状态寄存器 (ADC_ISR)

Offset	Bit Position							
0x44	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved			SASTPDF	Reserved		EOCBF	EOCAF
Access				RO			RO	RO

位/位域	标记	位名	功能描述	读写
7:5	Reserved	保留	-	-
4	SASTPDF	序列 A 被打断标志	● 0b0: 序列 A 未被打断 ● 0b1: 序列 A 被打断  说明 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。	RO
3:2	Reserved	保留	-	-
1	EOCBF	序列 B 转换完成标志	● 0b0: 序列 B 所选通道未扫描完成 ● 0b1: 序列 B 所选通道全部扫描完成  说明 序列 B 所选通道全部扫描完成后置 1。	RO
0	EOCAF	序列 A 转换完成标志	● 0b0: 序列 A 所选通道未扫描完成 ● 0b1: 序列 A 所选通道全部扫描完成  说明 序列 A 所选通道全部扫描完成后置 1。	RO

18.4.14 中断许可寄存器 (ADC_ICR)

Offset	Bit Position							
0x45	7	6	5	4	3	2	1	0
Reset	0x03							
Name	Reserved						EOCBIE	EOCAIE
Access							RW	RW

位/位域	标记	位名	功能描述	读写
7:2	Reserved	保留	-	-
1	EOCBIE	序列 B 转换完成中断使能	0b0: 中断禁止 0b1: 中断许可	RW
0	EOCAIE	序列 A 转换完成中断使能	0b0: 中断禁止 0b1: 中断许可	RW

18.4.15 中断状态复位寄存器 (ADC_ISCLRR)

Offset	Bit Position							
0x46	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved			CLRSASTPDF	Reserved		CLREOCBF	CLREOCAF
Access				WO			WO	WO

位/位域	标记	位名	功能描述	读写
7:5	Reserved	保留	-	-
4	CLRSASTPDF	序列 A 被打断标志复位	写 1 复位 SASTPDF 状态位, 写 0 无效; 读出恒为 0	WO
3:2	Reserved	保留	-	-
1	CLREOCBF	序列 B 转换完成标志复位	写 1 复位 EOCBF 状态位, 写 0 无效; 读出恒为 0	WO
0	CLREOCAF	序列 A 转换完成标志复位	写 1 复位 EOCAF 状态位, 写 0 无效; 读出恒为 0	WO

18.4.16 协同模式控制寄存器 (ADC_SYNCRR)



说明

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。

Offset	Bit Position															
0x4C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0C00															
Name	SYNCDLY								Reserved	SYNCDMD[2:0]			Reserved			SYNCEN
Access	RW									RW						RW

位/位域	标记	位名	功能描述	读写
15:8	SYNCDLY	同步延迟时间	延迟触发模式时，两个 ADC 的启动延迟时间 t_{SYNCDLY}。 ● 0x00：禁止设置 ● 0x01： $t_{\text{SYNCDLY}} = \text{PCLK2 时钟周期} \times 1$ ● ● 0xFF： $t_{\text{SYNCDLY}} = \text{PCLK2 时钟周期} \times 255$ 说明 请在 SYNCEN 为 0 时设置本寄存器。 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： ● 单次延迟触发模式： $t_{\text{SYNCDLY}} > t_{\text{SPL}}$ ● 两个 ADC 循环延迟触发模式： $t_{\text{SYNCDLY}} > t_{\text{SPL}}$，且 $t_{\text{SYNCDLY}} > t_{\text{SCAN}}/2$ ● 三个 ADC 循环延迟触发模式： $t_{\text{SYNCDLY}} > t_{\text{SPL}}$，且 $t_{\text{SYNCDLY}} > t_{\text{SCAN}}/3$ ● 单次并行触发模式：本寄存器设置无效。 ● 循环并行触发模式： $t_{\text{SYNCDLY}} > t_{\text{SCAN}}$	RW
7	Reserved	保留	-	-
6:4	SYNCDMD[2:0]	同步模式选择	SYNCDMD[2]: ● 0b0：单次触发 ● 0b1：循环触发 SYNCDMD[1]: ● 0b0：延迟触发模式 ● 0b1：并行触发模式 SYNCDMD[0]: ● 0b0：ADC1 和 ADC2 同步工作，ADC3 独立工作 ● 0b1：ADC1，ADC2 和 ADC3 同步工作	RW

位/位域	标记	位名	功能描述	读写
6:4	SYNCMD[2:0]	同步模式选择	 说明 请在 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。	RW
3:1	Reserved	保留	-	-
0	SYNCEN	同步模式许可	● 0b0：同步模式无效 ● 0b1：同步模式有效  说明 同步模式只支持序列 A。 在 SYNCEN 写 1 之前，请将参与同步的几个 ADC 的序列 B 关闭 (ADC_CR0.MS[1]=0)，并给序列 A 选择相同数目的通道，设置相同的通道采样时间 ADC_SSTRx。以避免各 ADC 扫描时间 t_{SCAN} 不一致，引起后续同步失败。 软件对 ADC1_STR.STRT 写 0 强制停止转换时，SYNCEN 自动清 0。	RW

18.4.17 数据寄存器 (ADC_DRx)

本寄存器是用于存储各通道 ADC 转换数据的只读寄存器。根据过采样模式、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。

偏移地址 (Offset) : 0x50+0x2*x (“x”表示通道, ADC1 中 x=0~15, ADC2 中 x=0~7, ADC3 中 x=0~11)

过采样模式

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	RO AD[15:0]															
Access	RO															

数据右对齐-12 位分辨率

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved				RO AD[11:0]											
Access					RO											

数据右对齐-10 位分辨率

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved						RO AD[9:0]									
Access							RO									

数据右对齐-8 位分辨率

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved								RO AD[7:0]							
Access									RO							

数据左对齐-12 位分辨率

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	RO AD[11:0]												Reserved			
Access																

数据左对齐-10 位分辨率

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	RO AD[9:0]												Reserved			
Access																

数据左对齐-8 位分辨率

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	RO AD[7:0]												Reserved			
Access																

18.4.18 模拟看门狗控制寄存器 (ADC_AWDCR)

Offset	Bit Position															
0xA0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved						AWDCM		Reserved	AWD1MD	AWD1IEN	AWD1EN	Reserved	AWD0MD	AWD0IEN	AWD0EN
Access							RW			RW	RW	RW		RW	RW	RW

位/位域	标记	位名	功能描述	读写
15:10	Reserved	保留	-	-
9:8	AWDCM	看门狗窗口组合选择	0b00: 窗口组合无效, ADC_CMP1 输出窗口 1 独立比较结果 0b01: 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 0b10: 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 0b11: 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  说明 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
7	Reserved	保留	-	-
6	AWD1MD	看门狗窗口 1 比较模式	窗口 1 保护区域选择: 0b0: 转换结果 < AWD1DR0, 或转换结果 > AWD1DR1 0b1: 转换结果 ≥ AWD1DR0, 且转换结果 ≤ AWD1DR1	RW
5	AWD1IEN	看门狗窗口 1 中断使能	0b0: 看门狗窗口 1 比较中断 ADC_CMP1 无效 0b1: 看门狗窗口 1 比较中断 ADC_CMP1 有效	RW
4	AWD1EN	看门狗窗口 1 比较功能使能	0b0: 看门狗窗口 1 比较功能无效 0b1: 看门狗窗口 1 比较功能有效	RW
3	Reserved	保留	-	-
2	AWD0MD	看门狗窗口 0 比较模式	窗口 0 保护区域选择: 0b0: 转换结果 < AWD0DR0, 或转换结果 > AWD0DR1 0b1: 转换结果 ≥ AWD0DR0, 且转换结果 ≤ AWD0DR1	RW
1	AWD0IEN	看门狗窗口 0 中断使能	0b0: 看门狗窗口 0 比较中断 ADC_CMP0 无效 0b1: 看门狗窗口 0 比较中断 ADC_CMP0 有效	RW
0	AWD0EN	看门狗窗口 0 比较功能使能	0b0: 看门狗窗口 0 比较功能无效 0b1: 看门狗窗口 0 比较功能有效	RW

18.4.19 模拟看门狗状态寄存器 (ADC_AWDSR)

Offset	Bit Position							
0xA2	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved			AWDCMF	Reserved		AWD1F	AWD0F
Access				RO			RO	RO

位/位域	标记	位名	功能描述	读写
7:5	Reserved	保留	-	-
4	AWDCMF	看门狗窗口组合比较状态位	0b0: 窗口 0 与窗口 1 比较结果不满足组合条件 0b1: 窗口 0 与窗口 1 比较结果满足组合条件  说明 看门狗窗口组合比较功能有效（即 ADC_AWDCR.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。	RO
3:2	Reserved	保留	-	-
1	AWD1F	看门狗窗口 1 比较状态位	0b0: 转换结果不满足比较条件 0b1: 转换结果满足比较条件  说明 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。	RO
0	AWD0F	看门狗窗口 0 比较状态位	0b0: 转换结果不满足比较条件 0b1: 转换结果满足比较条件  说明 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。	RO

18.4.20 模拟看门狗状态复位寄存器 (ADC_AWDSCCLR)

Offset	Bit Position							
0xA3	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved			CLRAWDCMF	Reserved		CLRAW1F	CLRAW0F
Access				WO			WO	WO

位/位域	标记	位名	功能描述	读写
7:5	Reserved	保留	-	-
4	CLRAWDCMF	看门狗窗口组合比较状态位复位	写 1 复位 AWD1F 状态位，写 0 无效；读出恒为 0	WO
3:2	Reserved	保留	-	-
1	CLRAW1F	看门狗窗口 1 比较状态位复位	写 1 复位 AWD1F 状态位，写 0 无效；读出恒为 0	WO
0	CLRAW0F	看门狗窗口 0 比较状态位复位	写 1 复位 AWD0F 状态位，写 0 无效；读出恒为 0	WO

18.4.21 模拟看门狗窗口 x 阈值寄存器 (ADC_AWDxDRy) (x=0~1, y=0~1)

偏移地址 (Offset) : 0xA4+0x8*x+0x2*y

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000 (AWDxDR0); 0xFFFF (AWDxDR1)															
Name	AWDDR															
Access	RW															

位/位域	标记	位名	功能描述	读写
15:0	AWDDR	比较数据	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  说明 AWD0DR0、AWD0DR1、AWD1DR0、AWD1DR1 未使用的位需补 0。	RW

18.4.22 模拟看门狗窗口 x 比较通道选择寄存器 (ADC_AWDxCHSR) (x=0~1)

偏移地址 (Offset) : 0xA8+0x8*x

Offset	Bit Position							
-	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved			AWDCH				
Access				RW				

位/位域	标记	位名	功能描述	读写
7:5	Reserved	保留	-	-
4:0	AWDCH	看门狗比较通道选择	选择窗口 x 的比较通道 ● 0x00: CH0 (通道 0) ● 0x01: CH1 (通道 1) ● ● 0x0F: CH15 (通道 15) ● 其他: 禁止设置  注意 请不要设置成不存在的通道。ADC1 支持通道 0~15，ADC2 支持通道 0~7，ADC3 支持通道 0~11。	RW

18.5 注意事项

18.5.1 数据寄存器读取时注意事项

ADC 数据寄存器 ADC_DR 请以半字单位访问。请不要以字节单位访问数据寄存器。

18.5.2 扫描完成中断处理注意事项

当对同一通道进行连续两次扫描转换时，从第一次转换完成中断处理到第二次转完成中断处理期间，如果 CPU 没有及时将第一次的转换数据读取，那么第二次的转换数据会将第一次的转换数据覆盖。

18.5.3 模块停止与低功耗设定的注意事项

通过设定寄存器 PWC_FCG，可以设定 ADC 模块停止，降低功耗。ADC 初始状态为停止状态。需要 ADC 模块工作时请先设定 PWC_FCG 寄存器相应位取消停止，并等待 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。

具体描述请参见《电源控制 (PWC)》章节“动作模式与低功耗模式”。

18.5.4 ADC 转换模拟通道输入的引脚设定

当芯片引脚设定为 ADC 模拟通道输入时，请先禁用对应引脚的数字功能 (PCRxy.DDIS)。具体描述请参见《通用 IO (GPIO)》。

18.5.5 噪声控制

为防止浪涌等异常电压破坏模拟输入引脚，推荐使用《数据手册》的“电气特性”章节所示保护电路。

19 数模转换器（DAC）

19.1 简介

本 MCU 搭载了 1 个 12 位的数模转换器单元 DAC。DAC 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。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。

19.2 主要特性

- 两个 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
- 两个转换通道可实现同步转换
- 转换数据计算单元（DCU）的数据可输出三角波和锯齿波
- 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

19.3 功能说明

19.3.1 功能框图

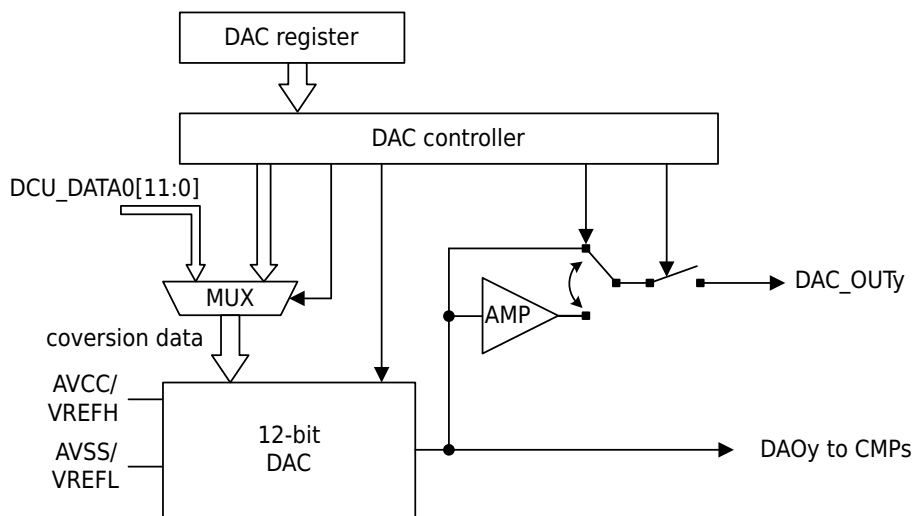


图 19-1 DAC 转换通道框图

表 19-1 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
DAC_OUTy	输出	DAC 转换模拟输出（y=1~2）

说明

本章节后文提到 y 均指通道编号。

19.3.2 DAC 转换

每个转换通道都可以独立进行 DAC 转换。DAC_DACR.DA1E 设置成 1 时，对应通道的 DAC 转换开始并从 DAC_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 DAC_DAOCR 寄存器将 DAC_OUTy 输出关闭。DAC 转换动作时序如下：

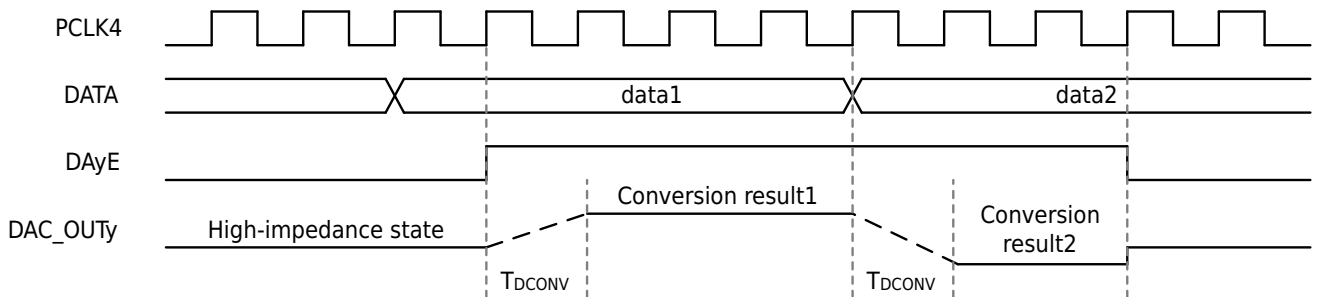


图 19-2 DAC 转换示意图

以使用 DAC 通道 1 进行单通道转换为例，对 DAC 转换过程进行介绍。

1. 设置 DAC 转换数据 (DAC_DADR1) 和数据格式 (DAC_DACR.DPSEL)。
2. 设置 DAC_DACR.DA1E 为 1，DAC 转换开始。经过转换时间 T_{DCONV} 后，转换结果从端口 DAC_OUT1 输出并一直保持到 DAC_DADR1 的值被改写或者 DAC_DACR.DA1E 被设置成 0。输出的模拟电压值可由以下公式计算：

$$DAC_{output} = \frac{Conversion}{4096} \times VREFH$$
 DAC_DACR.DA1E 为 1 时，改写 DAC_DADR1 的值将触发一次新的 DAC 转换。同样地，经过转换时间 T_{DCONV} 后，从端口 DAC_OUT1 输出新的转换结果。
3. 设置 DAC_DACR.DA1E 位为 0，关闭 DAC 通道 1，DAC_OUT1 输出高阻态。

19.3.3 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。

以使用 DAC 进行两通道同步转换为例，对同步转换过程进行介绍。

1. 设置 DAC 转换数据 (DAC_DADR1、DAC_DADR2) 和数据格式 (DAC_DACR.DPSEL)。
2. 设置 DAC_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 DAC_OUT1 和 DAC_OUT2 输出并一直保持到 DADR 的值被改写，或者 DAC_DACR.DAE 被设置成 0。
3. DAC_DACR.DAE 为 1 的状态下，同时改写 DAC_DADR1 和 DAC_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 DAC_OUT1 和 DAC_OUT2 输出新的转换结果。
4. 设置 DAC_DACR.DAE 为 0，两个通道同步关闭，DAC_OUT1 和 DAC_OUT2 输出高阻态。

19.3.4 外部数据转换

DAC 除了可以转换 DAC_DADRx 中的数据，还可以转换来自数据计算单元 (DCU) 的数据。DAC 的通道 1 和通道 2 分别对应 DCU 的单元 1 和单元 2。设置 DAC_DACR.EXTDSLy 为 1 即选择转换来自相应 DCU 单元的数据寄存器 0 的低 12 位数据。

转换外部数据时禁止使用输出放大器和 ADC 转换优先模式。

19.3.5 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 DAC_DAADPCR.ADPEN 为 1，DAC 进入 ADC 转换优先模式：

- 如果在 ADC 转换期间改写 DAC_DADRy 的值，DAC 转换并不会立即开始，而是要等到 ADC 转换完成后才开始。所以从改写 DAC_DADRy 到实际开始 DAC 转换最长需要等待一次 ADC 转换的时间，在此期间 DAC_DADRy 的值与模拟输出值并不一致。
- 如果改写 DAC_DADRy 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。

DAC_DAADPCR.ADCSLn（n=1~3）用于选择 ADC 转换优先的 ADC 通道。ADC 转换优先模式动作时序如下：

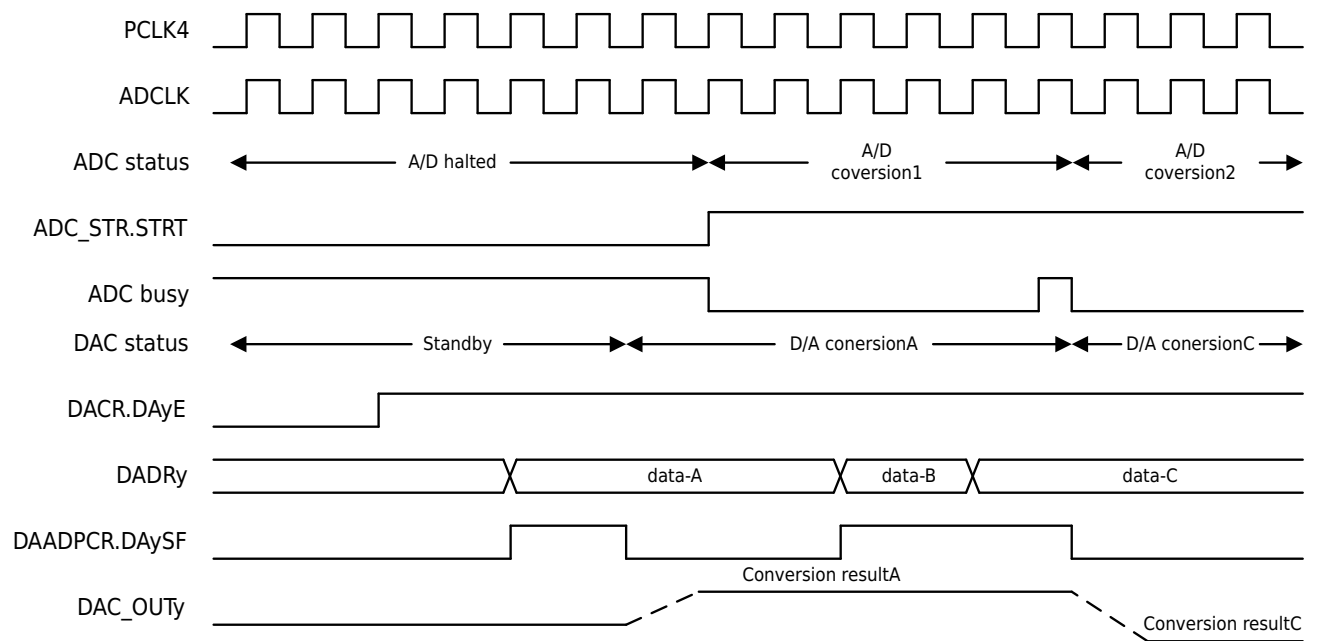


图 19-3 ADC 转换优先模式动作示意图



说明

选择转换外部数据（DAC_DACR.EXTDSLy=1）时，ADC 转换优先模式无效。

下面以 DAC 通道 1 为例，对 ADC 转换优先模式的设定步骤进行介绍。

1. 确认 ADC 处于停止状态，设置 DAC_DAADPCR.ADCSL1。
2. 确认 ADC 处于停止状态，设置 DAC_DACR.DA1E 为 1。
3. 将转换数据写入 DAC_DADR1。
 - 转换数据 A 被写入 DAC_DADR1 时，ADC_STR.STRT 为 0，ADC 处于停止状态，DAC 转换在 2 个 ADCLK 周期后开始。
 - 转换数据 B 被写入 DAC_DADR1 时，ADC_STR.STRT 为 1，ADC 正在进行 ADC 转换，DAC 转换要等到 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 DAC_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（DAC_DAADPCR.DAySF），确认当前数据转换完成后再改写 DAC_DADRy 的值。

19.4 寄存器

19.4.1 寄存器总表

DAC 基地址：0x40041000

表 19-2 数模转换器 (DAC) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	DAC_DADR1	数据寄存器 1	16
0x02	DAC_DADR2	数据寄存器 2	16
0x04	DAC_DACR	控制寄存器	16
0x06	DAC_DAADPCR	ADC 转换优先控制寄存器	16
0x1C	DAC_DAOCR	模拟输出控制寄存器	16

19.4.2 数据寄存器 (DAC_DADRy) (y=1~2)

该寄存器用于存放 DAC 变换的数据，每个通道对应有一个。只要允许 DAC 转换，DAC_DADRy 中的值就被转换成模拟电压并从模拟端口输出。对 DAC_DADR1 进行 32 位操作可实现该单元双通道转换数据同时设置。

偏移地址 (Offset)：0x00 (DAC_DADR1)，0x02 (DAC_DADR2)

数据右对齐 (DACR.DPSEL=0 时)

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved				DADR											
Access																

位/位域	标记	位名	功能描述	读写
15:12	Reserved	保留	-	-
11:0	DADR	变换数据	变换数据设定值	RW

数据左对齐 (DACR.DPSEL=1 时)

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	DADR												Reserved			
Access																

位/位域	标记	位名	功能描述	读写
15:4	DADR	变换数据	变换数据设定值	RW
3:0	Reserved	保留	-	-

19.4.3 控制寄存器 (DAC_DACR)

Offset	Bit Position															
0x04	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved			EXTDSL2	EXTDSL1	DAAMP2	DAAMP1	DPSEL	Reserved					DA2E	DA1E	DAE
Access				RW	RW	RW	RW	RW						RW	RW	RW

位/位域	标记	位名	功能描述	读写
15:13	Reserved	保留	-	-
12	EXTDSL2	通道 2 外部数据选择	0b0: 选择 DAC_DADR2 作为通道 2 的转换数据 0b1: 选择外部数据作为通道 2 的转换数据	RW
11	EXTDSL1	通道 1 外部数据选择	0b0: 选择 DAC_DADR1 作为通道 1 的转换数据 0b1: 选择外部数据作为通道 1 的转换数据	RW
10	DAAMP2	通道 2 输出放大器使能	0b0: 禁止通道 2 输出放大器 0b1: 启动通道 2 输出放大器	RW
9	DAAMP1	通道 1 输出放大器使能	0b0: 禁止通道 1 输出放大器 0b1: 启动通道 1 输出放大器	RW
8	DPSEL	数据寄存器格式选择	0b0: 右对齐格式 0b1: 左对齐格式	RW
7:3	Reserved	保留	-	-
2	DA2E	通道 2 使能	DAE=0 时此位设定有效: 0b0: 通道 2 禁止 0b1: 通道 2 使能	RW
1	DA1E	通道 1 使能	DAE=0 时此位设定有效: 0b0: 通道 1 禁止 0b1: 通道 1 使能	RW
0	DAE	通道总使能	0b0: 通道 1, 2 禁止 0b1: 通道 1, 2 使能	RW



说明

- 该寄存器用于控制 DAC 转换和 DAC 输出，具体信息请参见“[模拟输出控制寄存器 \(DAC_DAOCR\)](#)”相关描述。
- DA1E=0 且 DAE=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。
- 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。

19.4.4 模拟输出控制寄存器 (DAC_DAOCR)

Offset	Bit Position															
0x06	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	DAODIS2	DAODIS1	Reserved													
Access	RW	RW														

位/位域	标记	位名	功能描述	读写
15	DAODIS2	DAC_OUT2 端口输出控制	● 0b0: 允许端口输出模拟电压 ● 0b1: 禁止端口输出模拟电压	RW
14	DAODIS1	DAC_OUT1 端口输出控制	● 0b0: 允许端口输出模拟电压 ● 0b1: 禁止端口输出模拟电压	RW
13:0	Reserved	保留	-	-

表 19-11 DAC 转换和模拟输出控制

DAE	DAyE	DAAMP	DAODIS	DAC 转换	输出放大器	DAC_OUTy 输出	DAO 输出
0b0	0b0	-	-	停止	停止	高阻态	高阻态
	0b1	0b0	0b0	启动	停止	普通输出	普通输出
			0b1	启动	停止	高阻态	普通输出
		0b1	0b0	启动	启动	放大输出	普通输出
			0b1	启动	启动	高阻态	普通输出
0b1	0b0	0b0	0b0	启动	停止	普通输出	普通输出
			0b1	启动	停止	高阻态	普通输出
		0b1	0b0	启动	启动	放大输出	普通输出
			0b1	启动	启动	高阻态	普通输出
	0b1	0b0	0b0	启动	停止	普通输出	普通输出
			0b1	启动	停止	高阻态	普通输出
		0b1	0b0	启动	启动	放大输出	普通输出
			0b1	启动	启动	高阻态	普通输出

19.4.5 ADC 转换优先控制寄存器 (DAC_DAADPCR)

Offset	Bit Position															
0x1C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	ADPEN	Reserved					DA2SF	DA1SF	Reserved					ADCSL3	ADCSL2	ADCSL1
Access	RW						RO	RO						RW	RW	RW

位/位域	标记	位名	功能描述	读写
15	ADPEN	ADC 转换优先模式选择	0b0: ADC 转换优先模式无效 0b1: ADC 转换优先模式有效	RW
14:10	Reserved	保留	-	-
9	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: 0b0: 通道 2 数据更新完成 0b1: 通道 2 数据更新中	RO
8	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: 0b0: 通道 1 数据更新完成 0b1: 通道 1 数据更新中	RO
7:3	Reserved	保留	-	-
2	ADCSL3	ADC_3 优先选择	0b0: 不选择 ADC_3 作为 ADC 转换优先对象 0b1: 选择 ADC_3 作为 ADC 转换优先对象	RW
1	ADCSL2	ADC_2 优先选择	0b0: 不选择 ADC_2 作为 ADC 转换优先对象 0b1: 选择 ADC_2 作为 ADC 转换优先对象	RW
0	ADCSL1	ADC_1 优先选择	0b0: 不选择 ADC_1 作为 ADC 转换优先对象 0b1: 选择 ADC_1 作为 ADC 转换优先对象	RW

 **说明**

- 该寄存器用于 ADC 转换优先模式的控制。
- ADPEN 设定成 1 时 ADC 转换优先模式有效，请在设定 ADPEN 的同时选择作为优先对象的 ADC 单元。
- DA1SF、DA2SF 为 DA 转换状态标志位，反映的是当前该通道数据寄存器中的数据是否已被转换。

19.5 注意事项

19.5.1 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 都是停止的，需要解除模块停止状态后才可以访问寄存器。具体描述请参见《电源控制 (PWC)》章节“动作模式与低功耗模式”。

19.5.2 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 DAC_DACR 的 DAE 和 DAE 位全部设为 0。

19.5.3 停止低功耗模式下的 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 DAC_DACR 的 DAE 和 DAE 位全部设为 0。

19.5.4 掉电低功耗模式下的 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。

19.5.5 使用输出放大器

使用输出放大器时请使用以下的初始化程序。

步骤 1. 给 DAC_DAD_{Ry} 写全 0。

步骤 2. 设置 DAC_DACR.DAAMP_y 为 1。

步骤 3. 设置 DAC_DACR.DAE 或者 DAE 为 1。

步骤 4. 等待 3μs 后将变换数据写入 DAC_DAD_{Ry}。

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。

20 高级控制定时器（Timer6）

20.1 简介

高级控制定时器 6（Timer6）是一个 16 位计数宽度的高性能定时器，能在各种复杂应用场景中提供丰富、灵活的搭配组合和各种中断、事件、PWM 输出。该定时器支持锯齿波和三角波两种波形模式，可生成各种 PWM 波形（单边对齐独立 PWM、双边对称独立 PWM、双边对称互补 PWM、双边非对称 PWM 等）；单元间可实现软件同步和硬件同步（同步启动、停止、清零、刷新等）；各基准值寄存器支持缓存功能（单级缓存和双级缓存）；支持脉宽测量和周期测量；支持 2 相正交编码计数和 3 相正交编码计数；支持 EMB 控制。本系列产品中搭载 2 个单元的 Timer6。

20.2 主要特性

Timer6 基本的功能及特性如下表所示。

表 20-1 Timer6 的基本功能及特性

波形模式	锯齿波（递加、递减计数）
	三角波（递加递减计数）
基本功能	捕获输入
	软件同步
	硬件同步
	缓存功能
	脉宽测量
	周期测量
	正交编码计数
	通用 PWM 输出
	EMB 控制
中断输出	计数比较匹配中断
	计数周期匹配中断
	死区时间错误中断
事件输出	计数比较匹配事件
	计数周期匹配事件

20.3 功能说明

20.3.1 Timer6 框图

Timer6 的基本框图如下图所示。框图中所示“<t>”表示单元编号，即“<t>”为 1~2，本章节后文提到“<t>”时均指单元编号，不再赘述。

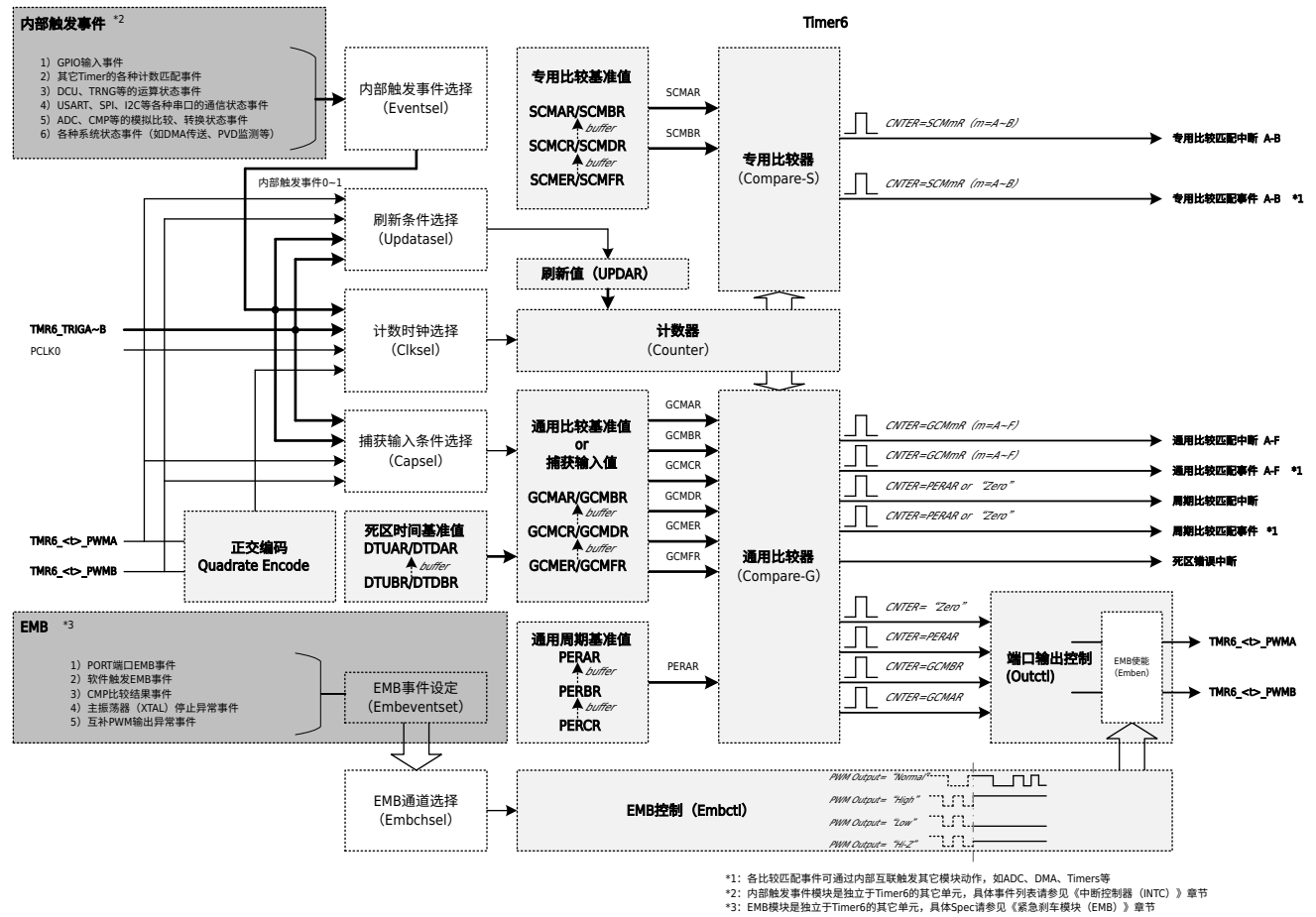


图 20-1 Timer6 基本框图

表 20-2 所示为 Timer6 的输入输出端口列表。

表 20-2 Timer6 端口列表

端口名	方向	功能
TMR6_<t>_PWMA	输入/输出	● 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 ● 硬件启动、停止、清零、刷新条件输入端口
TMR6_<t>_PWMB		
TMR6_TRIGA	输入/输出	● 硬件计数时钟输入端口或捕获输入端口 ● 硬件启动、停止、清零、刷新条件输入端口
TMR6_TRIGB		

20.3.2 基本动作

20.3.2.1 波形模式

Timer6 有 2 种基本计数波形模式，锯齿波模式和三角波模式。两种波形模式的基本波形如下所示。

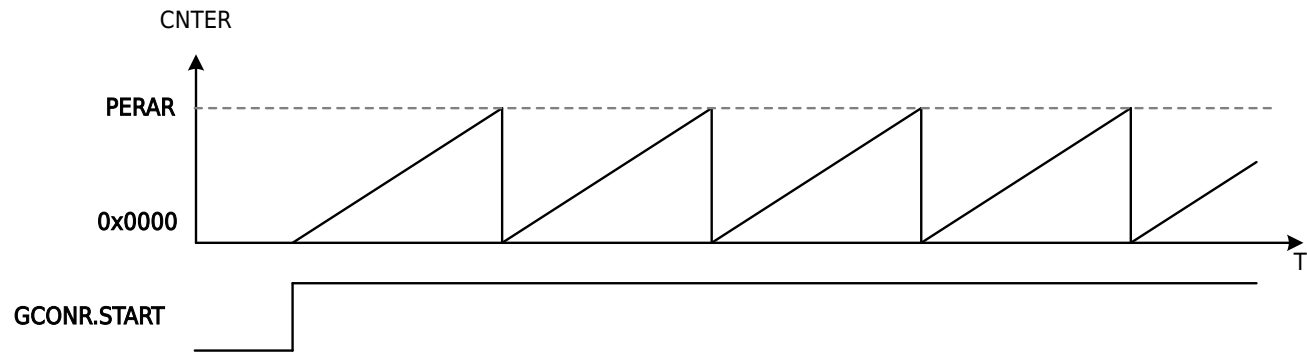


图 20-2 锯齿波波形 (递加计数)

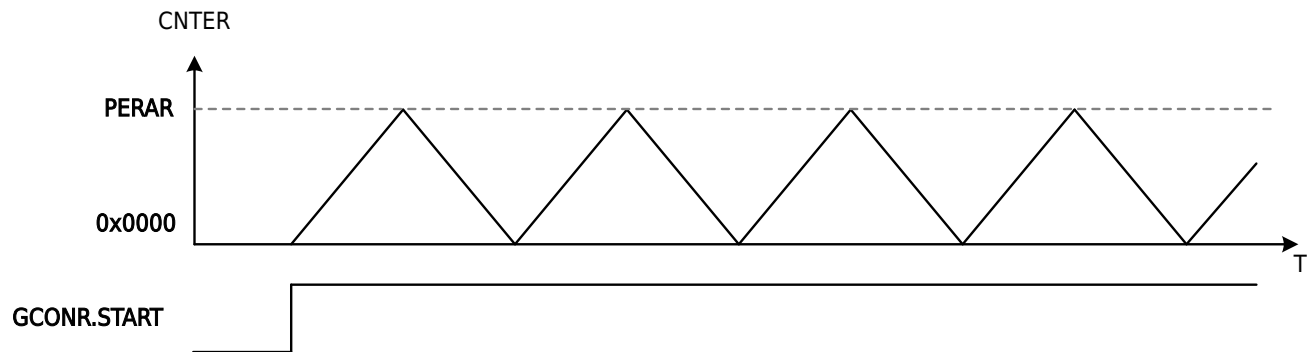


图 20-3 三角波波形

20.3.2.2 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。下图为比较输出的动作例。

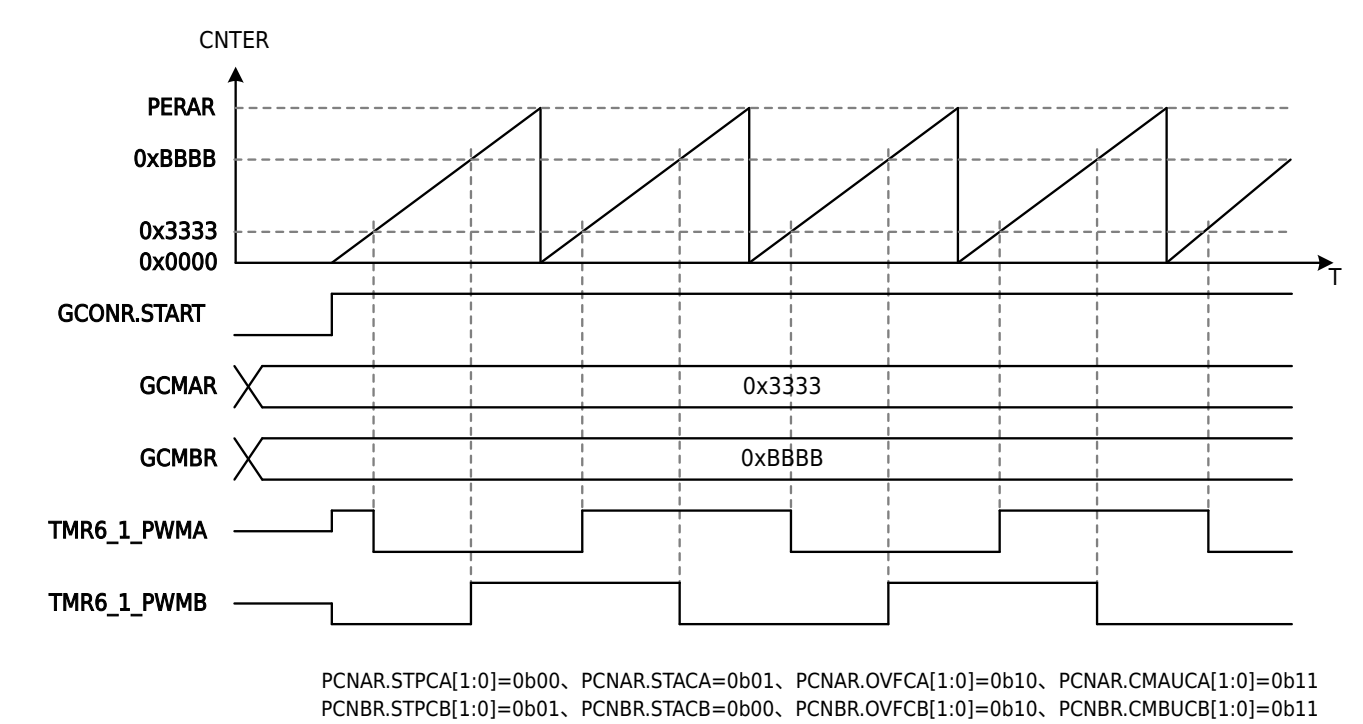


图 20-4 比较输出动作

20.3.2.3 捕获输入

每个单元都具有捕获输入功能，具备 2 组捕获输入寄存器（GCMAR、GCMBR），用于保存捕获到的当前计数值。设定端口控制寄存器 PCNA.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 0~1、TMR6_TRIGA~B 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。下图为捕获输入的动作例。

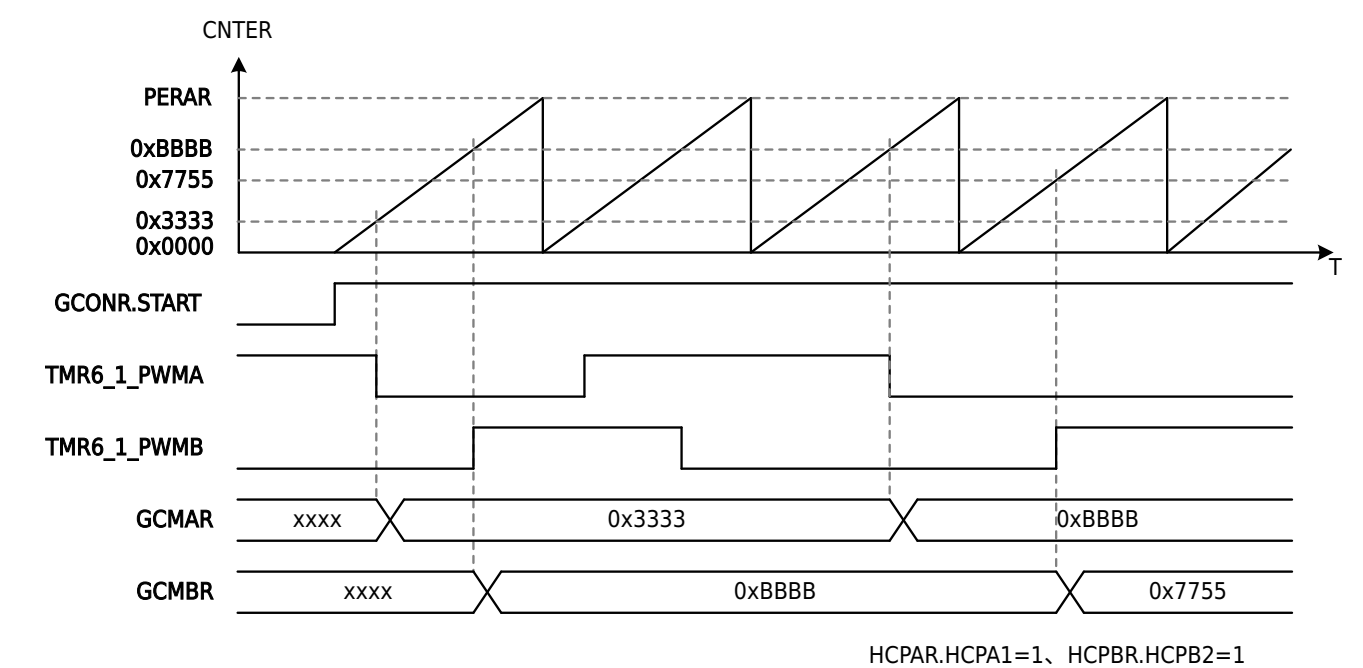


图 20-5 捕获输入动作

20.3.3 时钟源选择

Timer6 的计数时钟可以有以下几种选择：

1. PCLK0 的 1、2、4、8、16、32、64、128、256、512、1024 分频 (GCONR.CKDIV 设定)
2. 内部触发事件输入 0~1 (HCUPR.HCUPn 或 HCDOR.HCDO_n 设定) (n=8~9)
3. TMR6_TRIGA~B 的端口输入 (HCUPR.HCUPn 或 HCDOR.HCDO_n 设定) (n=16~19)
4. TMR6_<t>_PWMA 和 TMR6_<t>_PWMB 的正交编码关系输入 (HCUPR.HCUPn 或 HCDOR.HCDO_n 设定) (n=0~7)

计数时钟源选择 1 时为软件计数模式，其余为硬件计数模式。

硬件计数模式下：

- 各时钟互相独立，可分别设定有效或无效。
- 软件时钟自动无效。

20.3.4 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。

20.3.4.1 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=0b0 (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=0b1 (递加计数)，则定时器计数到下溢后变为递加计数模式。

在计数停止时，设定 GCONR.DIR 位，计数开始后直至上溢或下溢时，GCONR.DIR 的设定反映到计数中。

20.3.4.2 三角波计数方向

三角波模式时，计数过程中设定计数方向无效，计数至计数峰点或计数谷点时，自动改变计数方向。

20.3.5 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 (CNT_{ER}) 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 (CNT_{ER}) 的值更新为通用刷新值寄存器 (UPDAR) 中指定的值。下图是锯齿波模式在计数中的硬件刷新动作例。

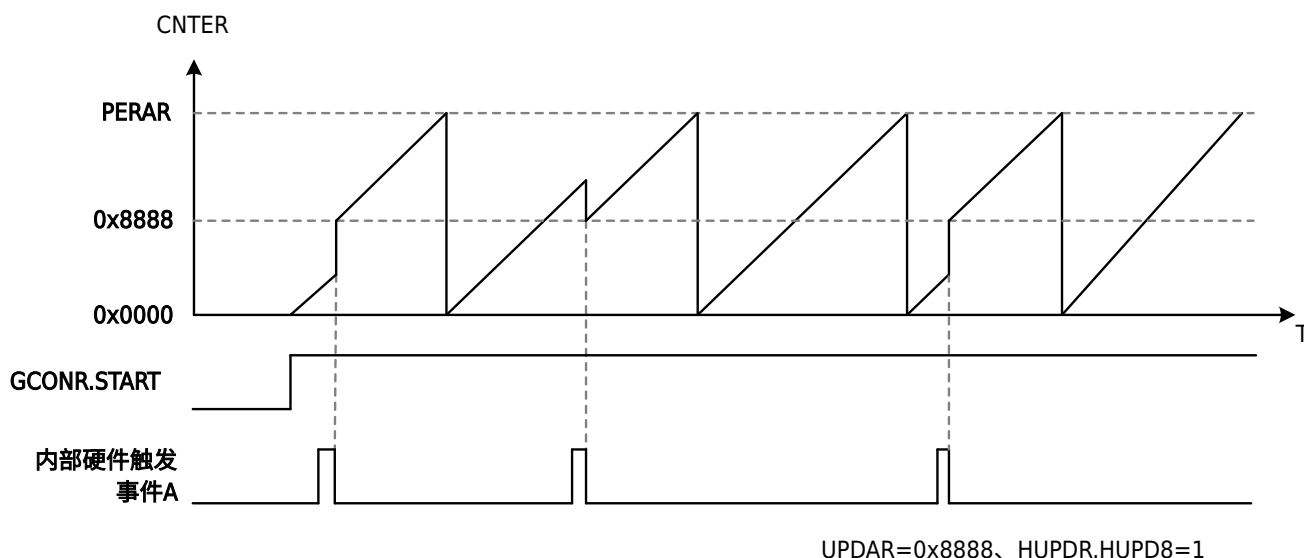


图 20-6 硬件刷新动作

20.3.6 数字滤波

TMR6_<t>_PWMA、TMR6_<t>_PWMB、TMR6_TRIGA~B 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（FCNTR）设定；TMR6_TRIGA~B 端口是一组单元间共用的端口通过设定公共滤波控制寄存器（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公共滤波控制寄存器（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。其动作例如下图所示。

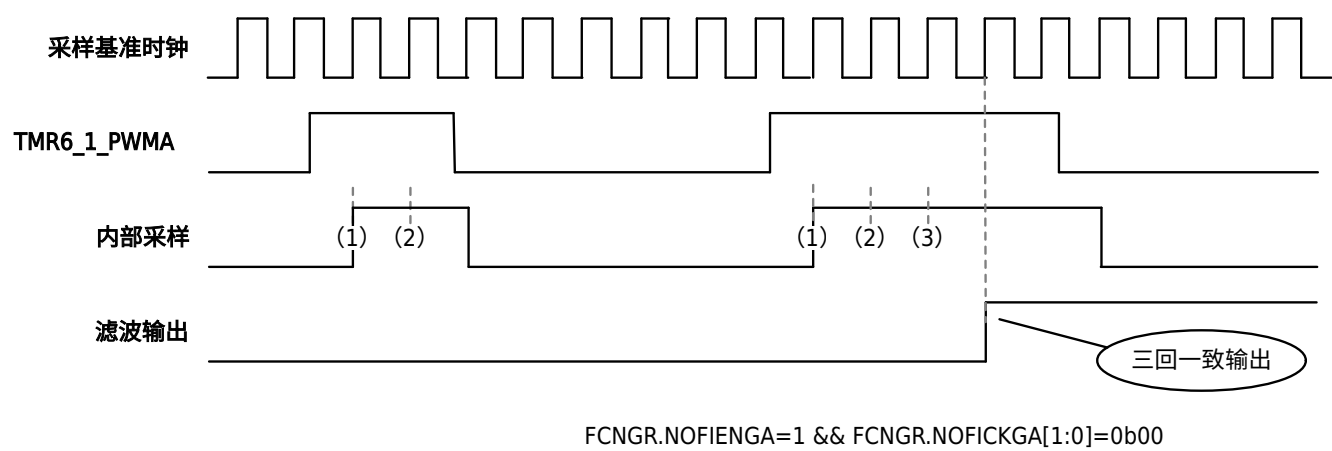


图 20-7 捕获输入端口的滤波功能

20.3.7 软件同步

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，产品配置了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。

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。

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。

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。

如下图所示，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。

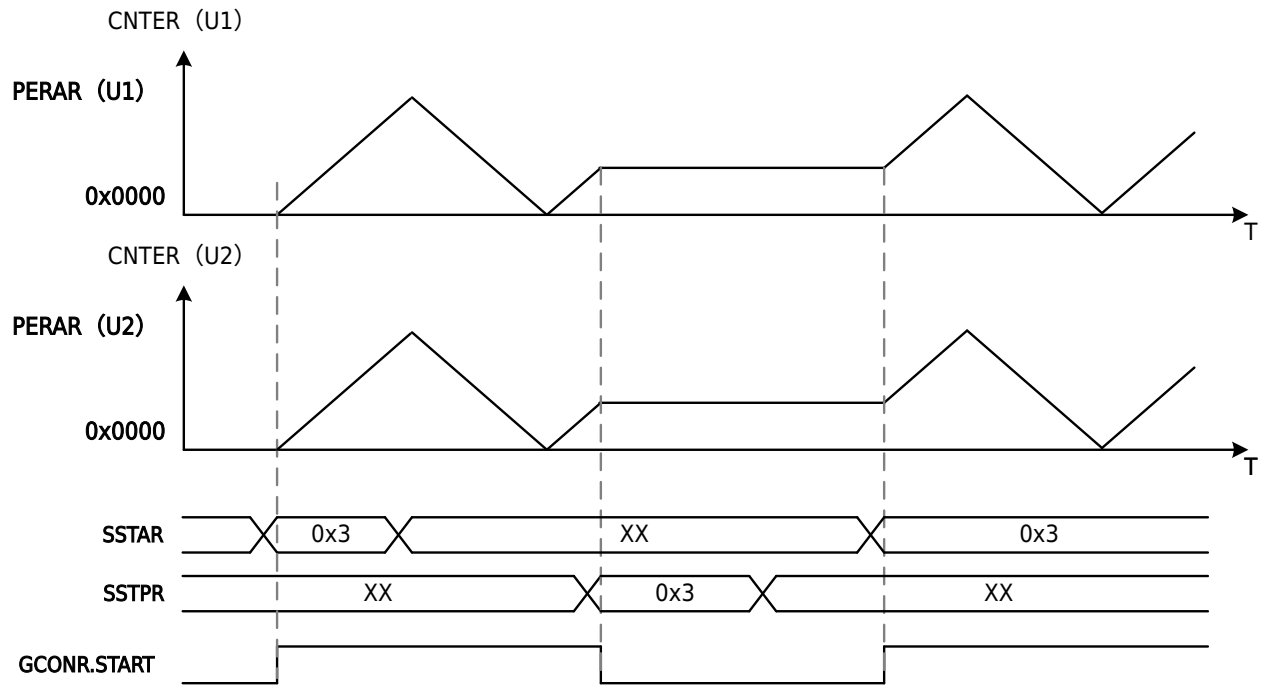


图 20-8 软件同步动作

20.3.8 硬件同步

每个单元除独立拥有 2 个通用输入端口（TMR6_<t>_PWMA、TMR6_<t>_PWMB）外，还共同拥有 2 个通用外部触发输入端口（TMR6_TRIGA、TMR6_TRIGB）及 2 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中对应的编号设定来选择，具体的寄存器内容详见《自动运行系统（AOS）》的“Timer6 硬件触发事件选择寄存器（TMR6_TRGSELx）（x=0~1）”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 0 寄存器（PWC_FCG0）的外围电路触发功能使能位置 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。

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。

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。

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。

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。

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。

下图是单元 1~2 的硬件同步动作例。

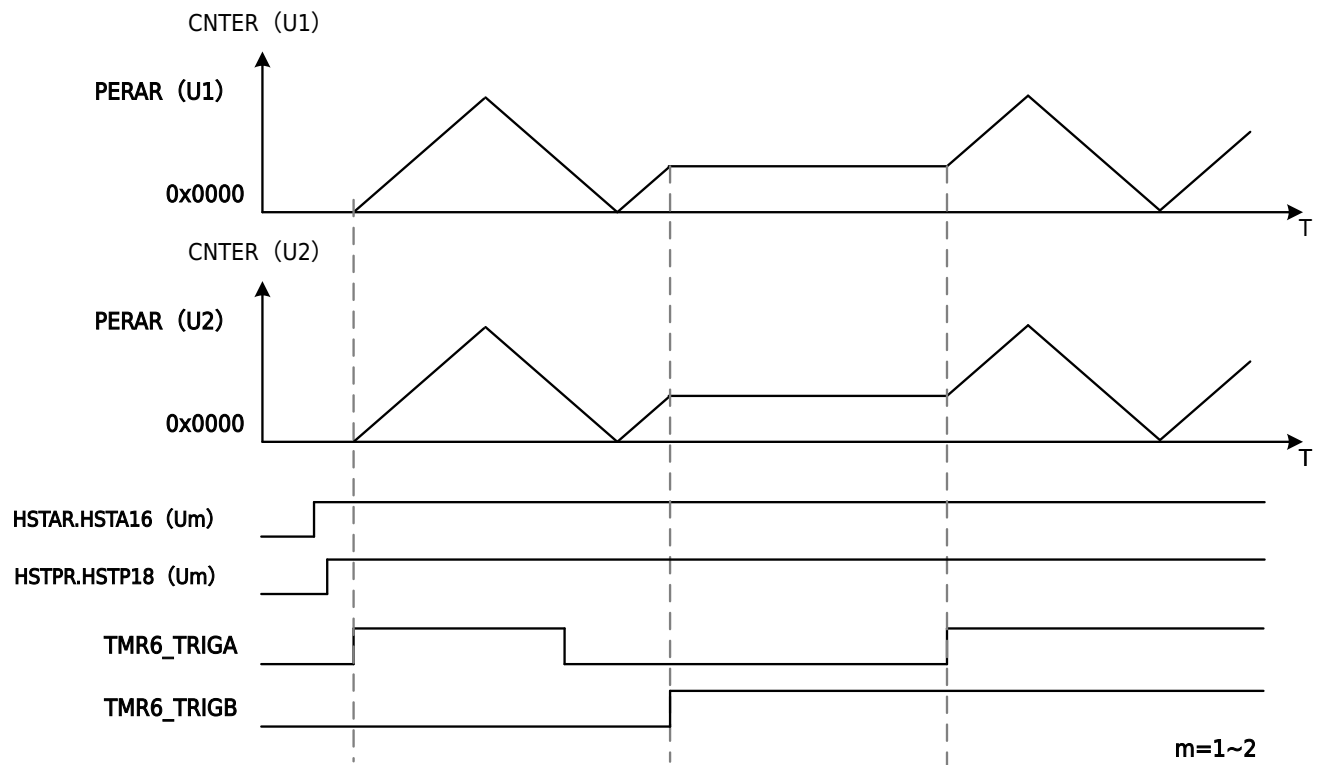


图 20-9 硬件同步动作

20.3.9 脉宽测量

在使用 TMR6_TRIGA~B 端口、TMR6_<t>_PWMA~B 端口的硬件触发相关功能（参考“[硬件同步](#)”）时，每个单元都可以实现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。对应动作如下图所示。

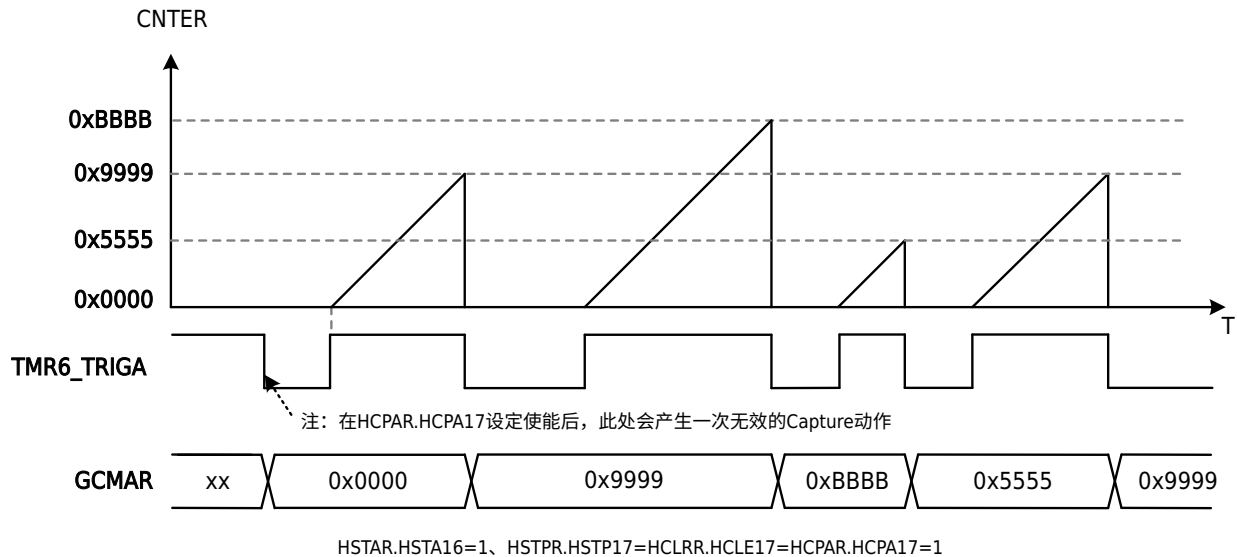


图 20-10 脉宽测量

20.3.10 周期测量

在使用 TMR6_TRIGA~B 端口、TMR6_<t>_PWMA~B 端口的硬件触发相关功能（参考“[硬件同步](#)”）时，每个单元都可以实现独立的周期测量功能。

例如，将计数器的硬件启动条件、硬件清零条件、GCMBR 寄存器的捕获输入条件均设为 TMR6_TRIGB 的上升沿，就可以实现连续的周期测量。对应动作如下图所示。

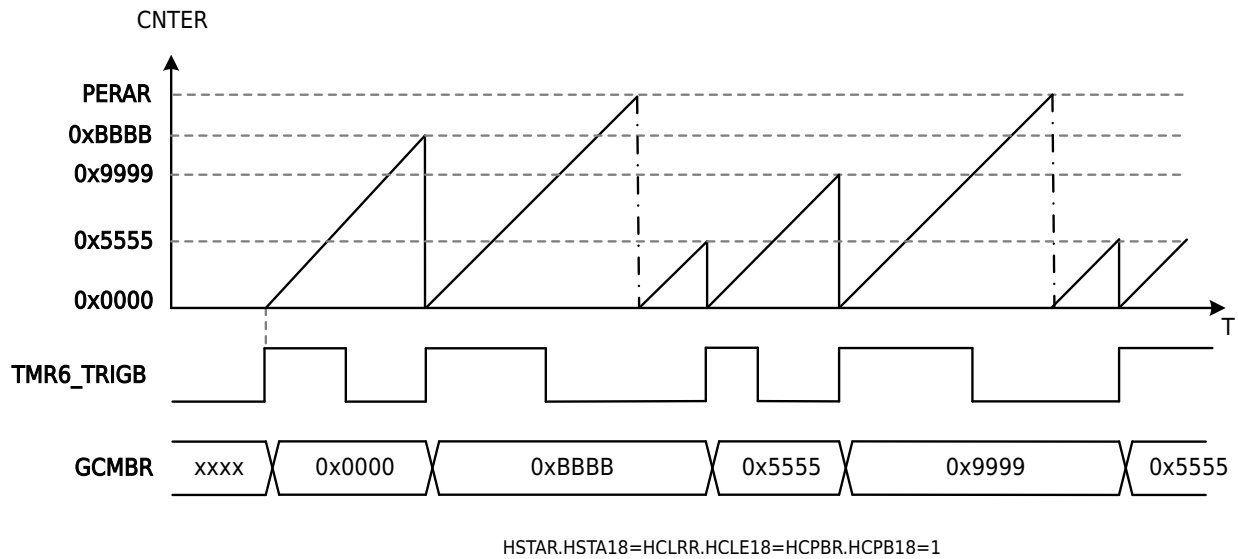


图 20-11 周期测量

20.3.11 缓存功能

Timer6 的通用周期基准值、通用比较基准值、专用比较基准值、死区时间基准值等都具有缓存功能，可在计数期间硬件实现周期变化、占空比变化、死区变化等。周期基准值、通用比较基准值和专用比较基准值等有单缓存、双缓存功能，死区时间基准值有单缓存功能。

20.3.11.1 单缓存动作

单缓存动作是指通过设定缓存控制寄存器（BCONR.BENA<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：

- 1. 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
- 2. 通用比较基准值缓存寄存器 (GCMCR、GCMDR) 的值自动传送到通用比较基准值寄存器 (GCMAR、GCMBR) 中 (比较输出时)
- 3. 通用比较基准值寄存器 (GCMAR、GCMBR) 的值自动传送到通用比较基准值缓存寄存器 (GCMCR、GCMDR) 中 (捕获输入时)
- 4. 专用比较基准值缓存寄存器 (SCMCR、SCMDR) 的值自动传送到专用比较基准值寄存器 (SCMAR、SCMBR) 中
- 5. 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

下图是单元 1 比较输出动作 (PCNAR.CAPMDA=0) 时、通用比较基准值寄存器的单缓存方式的时序图。从图中可以看到，在计数期间改变通用比较基准值寄存器 (GCMAR) 的值可以调整输出占空比，改变通用周期基准值寄存器 (PERAR) 的值可以调整输出周期。

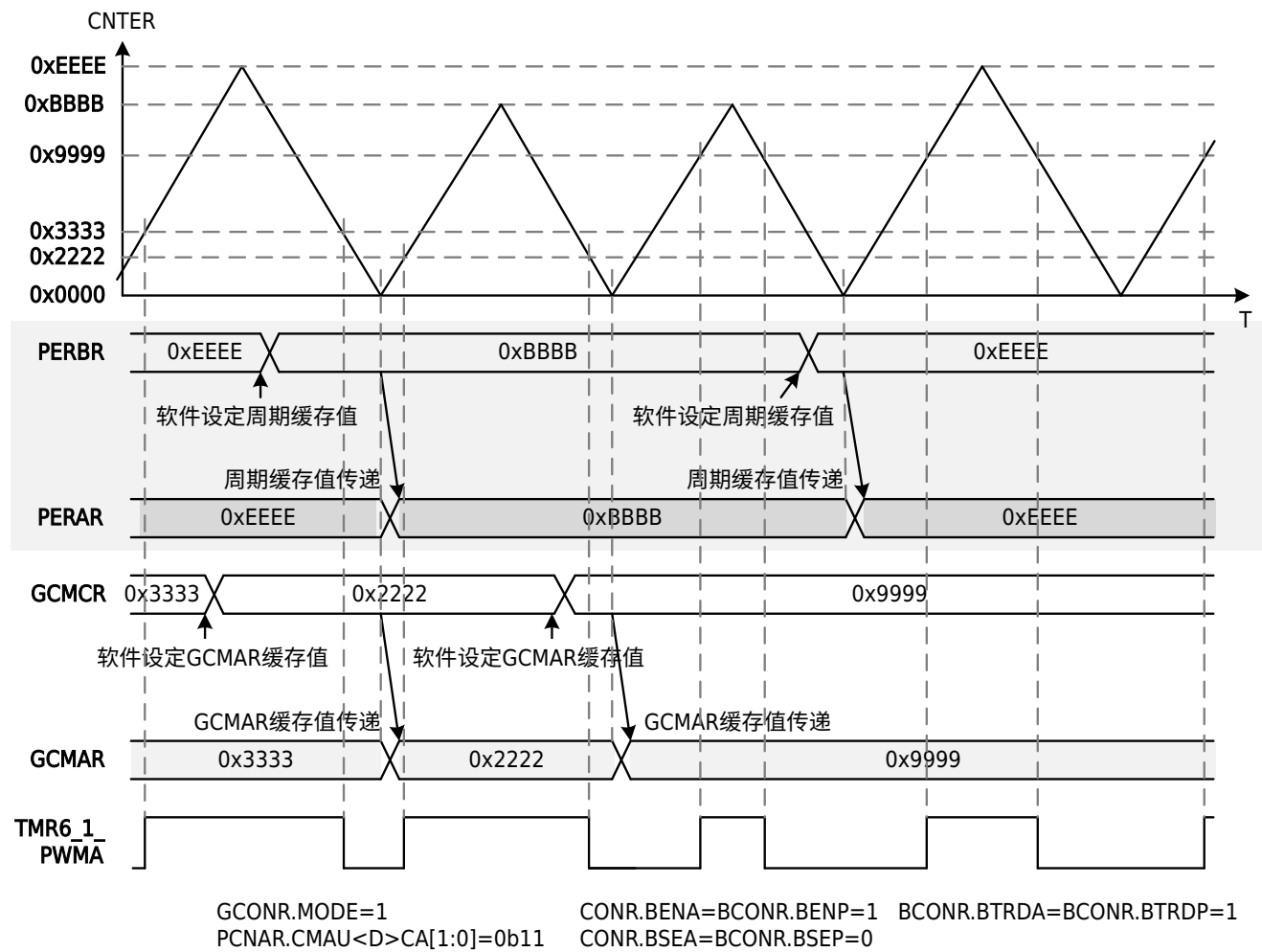


图 20-12 单缓存方式比较输出时序

20.3.11.2 双缓存动作

双缓存动作是指通过设定缓存控制寄存器 (BCONR.BENA<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：

- 1. 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

- 2. 通用比较基准值缓存寄存器（GCMCR、GCMDR）的值自动传送到通用比较基准值寄存器（GCMAR、GCMBR）、通用比较基准值双缓存寄存器（GCMER、GCMFR）的值自动传送到通用比较基准值缓存寄存器（GCMCR、GCMDR）中（比较输出时）
- 3. 通用比较基准值缓存寄存器（GCMCR、GCMDR）的值自动传送到通用比较基准值双缓存寄存器（GCMER、GCMFR）中、通用比较基准值寄存器（GCMAR、GCMBR）的值自动传送到通用比较基准值缓存寄存器（GCMCR、GCMDR）中（捕获输入时）
- 4. 专用比较基准值缓存寄存器（SCMCR、SCMDR）的值自动传送到专用比较基准值寄存器（SCMAR、SCMBR）中、专用比较基准值双缓存寄存器（SCMER、SCMFR）的值自动传送到专用比较基准值缓存寄存器（SCMCR、SCMDR）中

下图是内部触发事件 0 触发捕获输入时、双缓存方式的时序图。

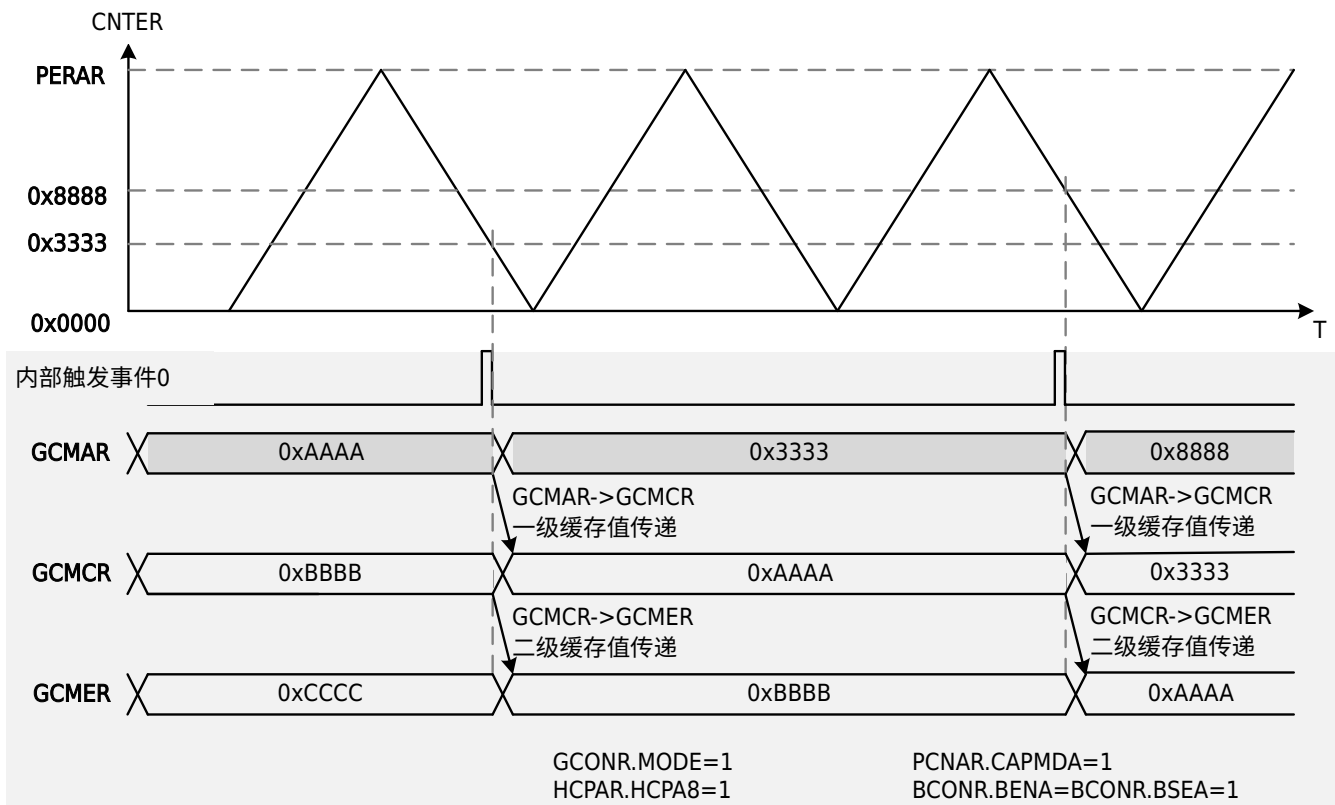


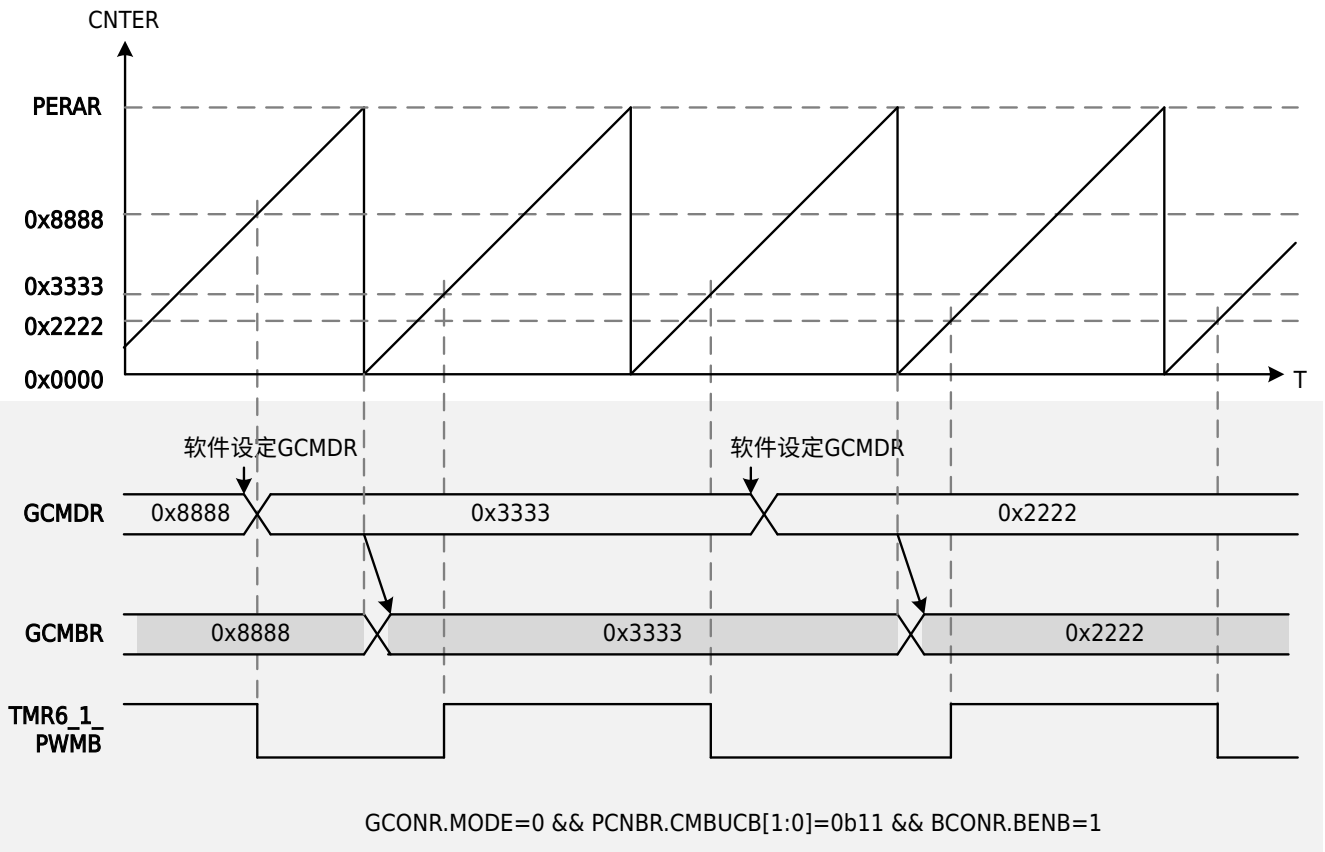
图 20-13 双缓存方式捕获输入时序

20.3.11.3 缓存传送节点

比较输出（锯齿波）缓存传送

在缓存功能有效（BCONR.BENA<P><SPA><SPB>=DCONR.DTBENU<D>=1）&&比较输出计数（PCNAR.CAPMDA=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。

下图是锯齿波递加计数时的缓存动作。



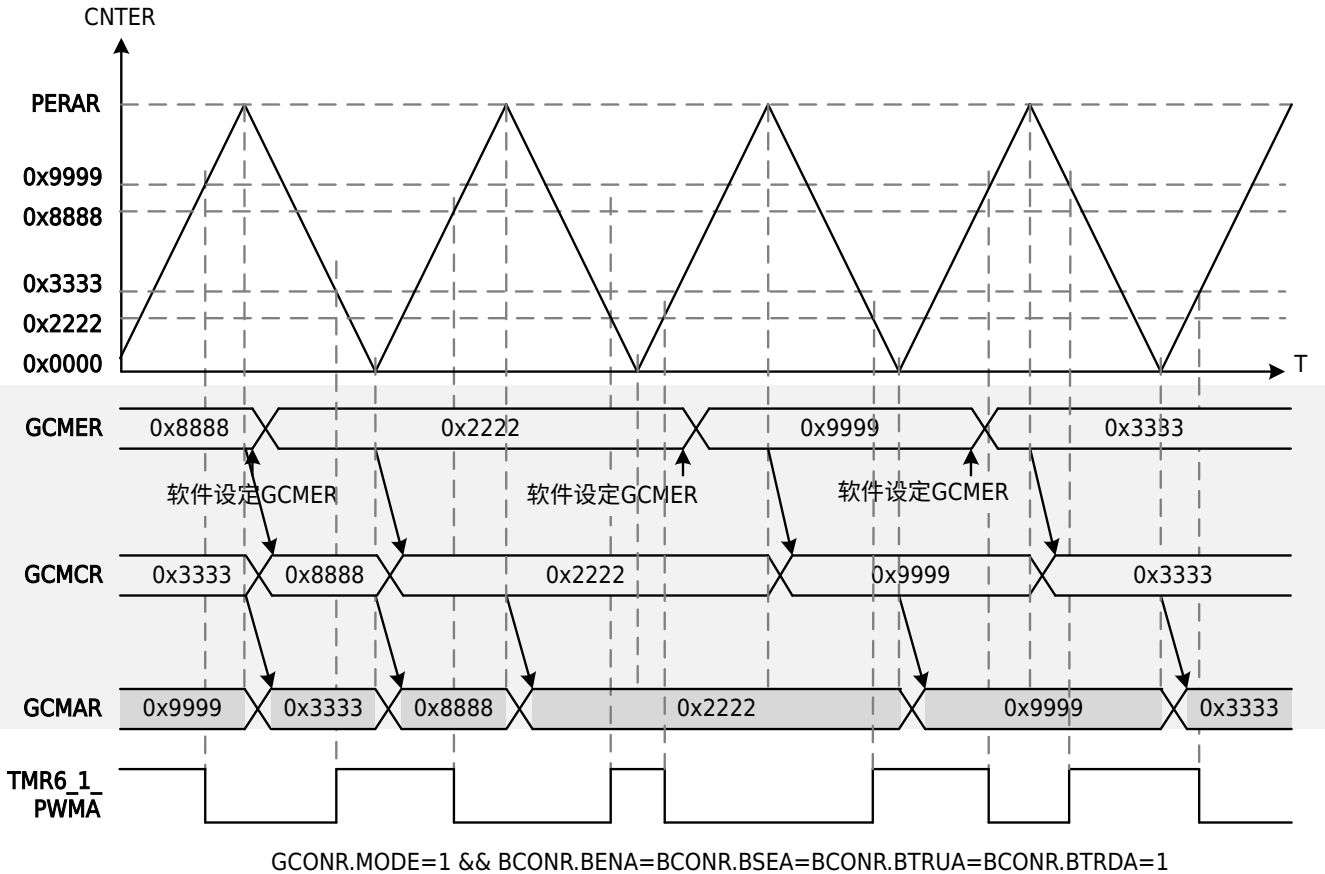
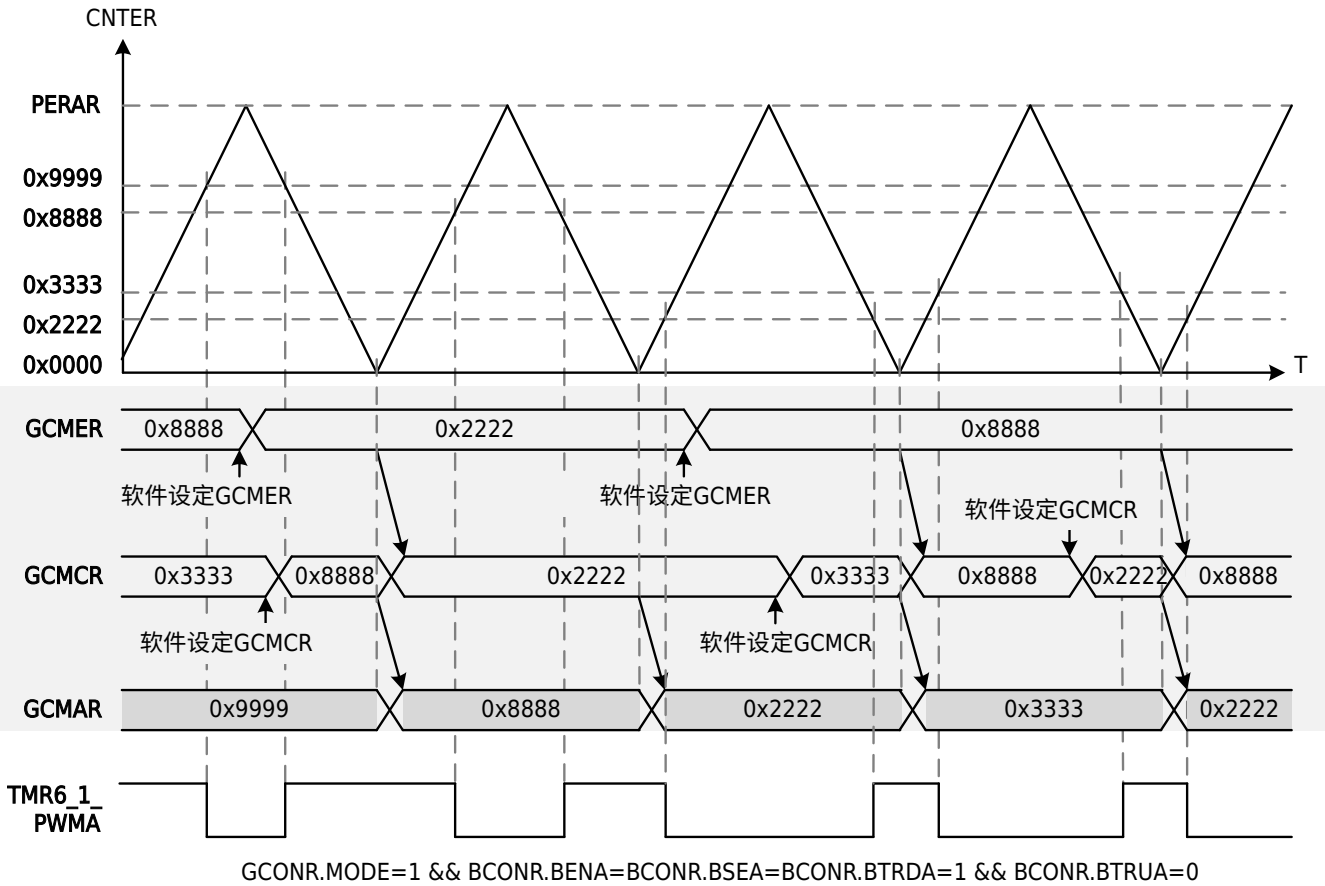
说明

- 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。

图 20-15 是三角波计数到谷点时的缓存动作，图 20-16 是三角波计数到峰点和谷点时均发生缓存动作的示意图。



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)。

20.3.12 通用 PWM 输出

20.3.12.1 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。

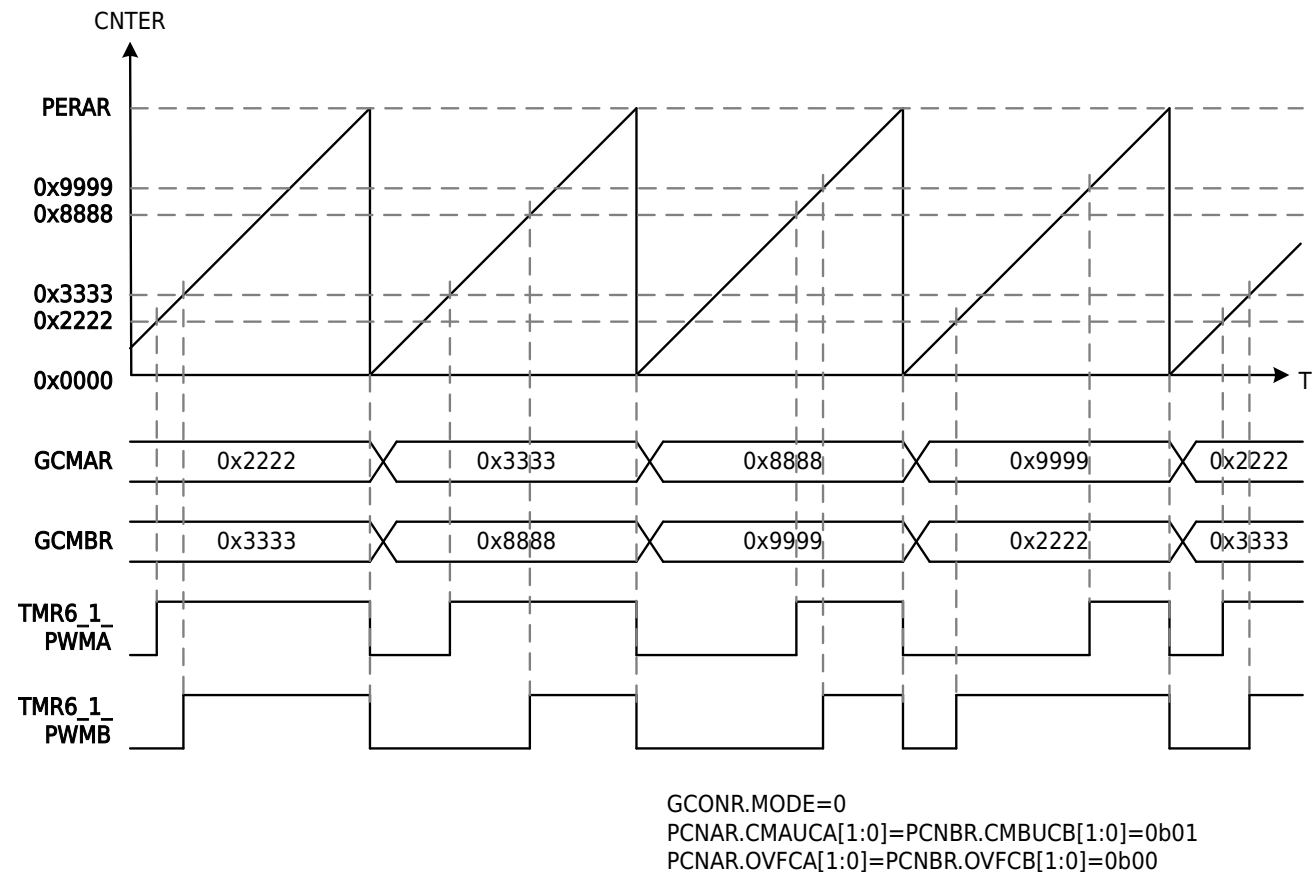


图 20-17 单边对齐独立 PWM

20.3.12.2 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。下图是 TMR6_<t>_PWMA 端口和 TMR6_<t>_PWMB 端口实现双边对称独立输出 PWM。

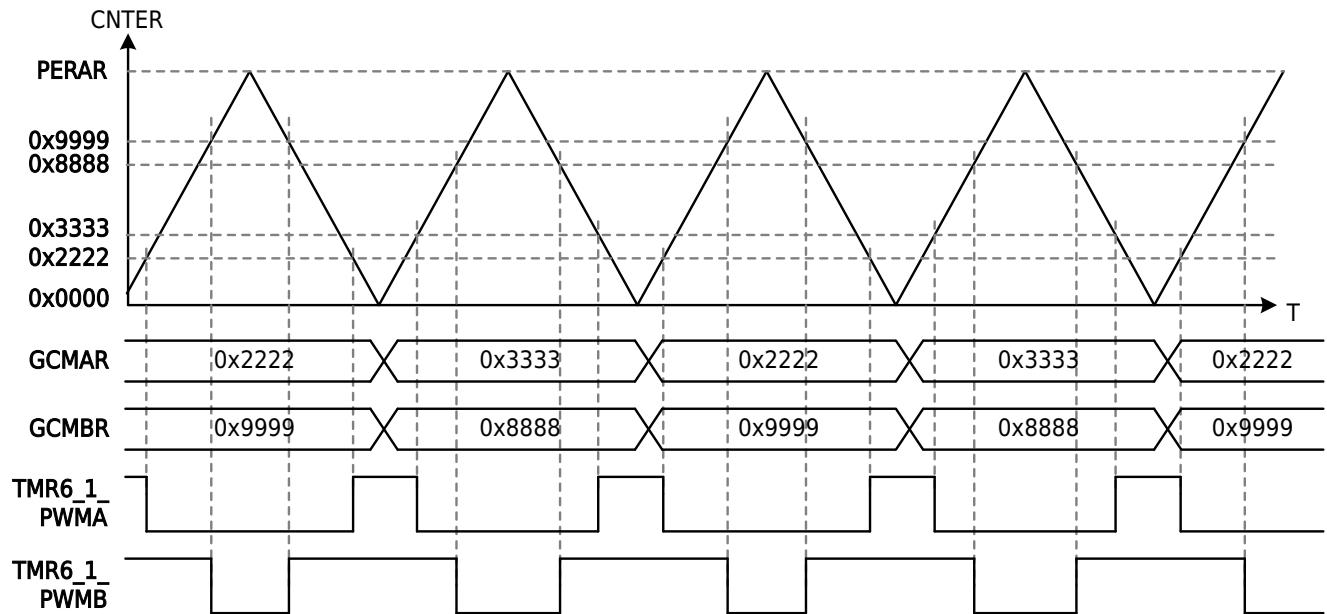


图 20-18 双边对称独立 PWM

20.3.12.3 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。

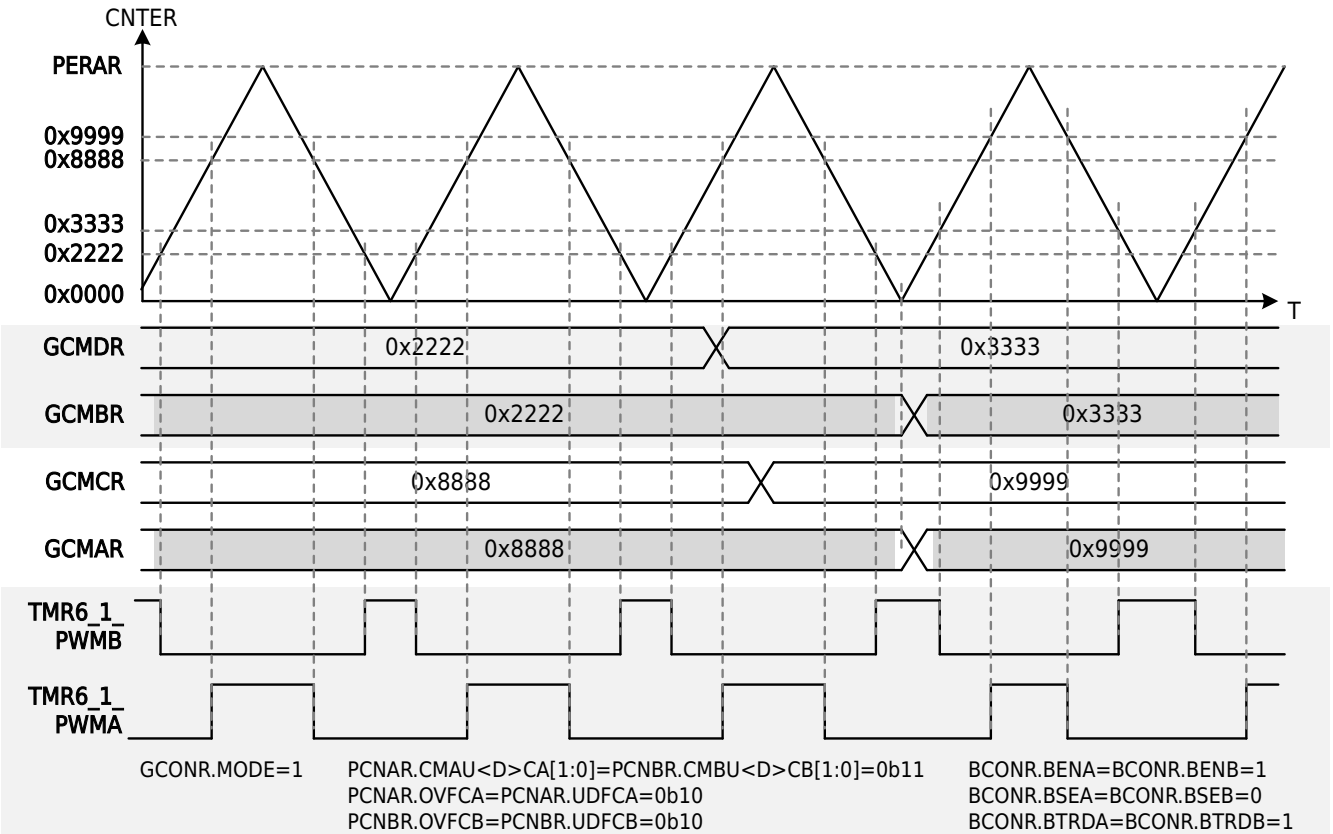


图 20-19 软件设定 GCMBR 互补 PWM 波输出

硬件设定 GCMBR 互补 PWM 输出

硬件设定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。

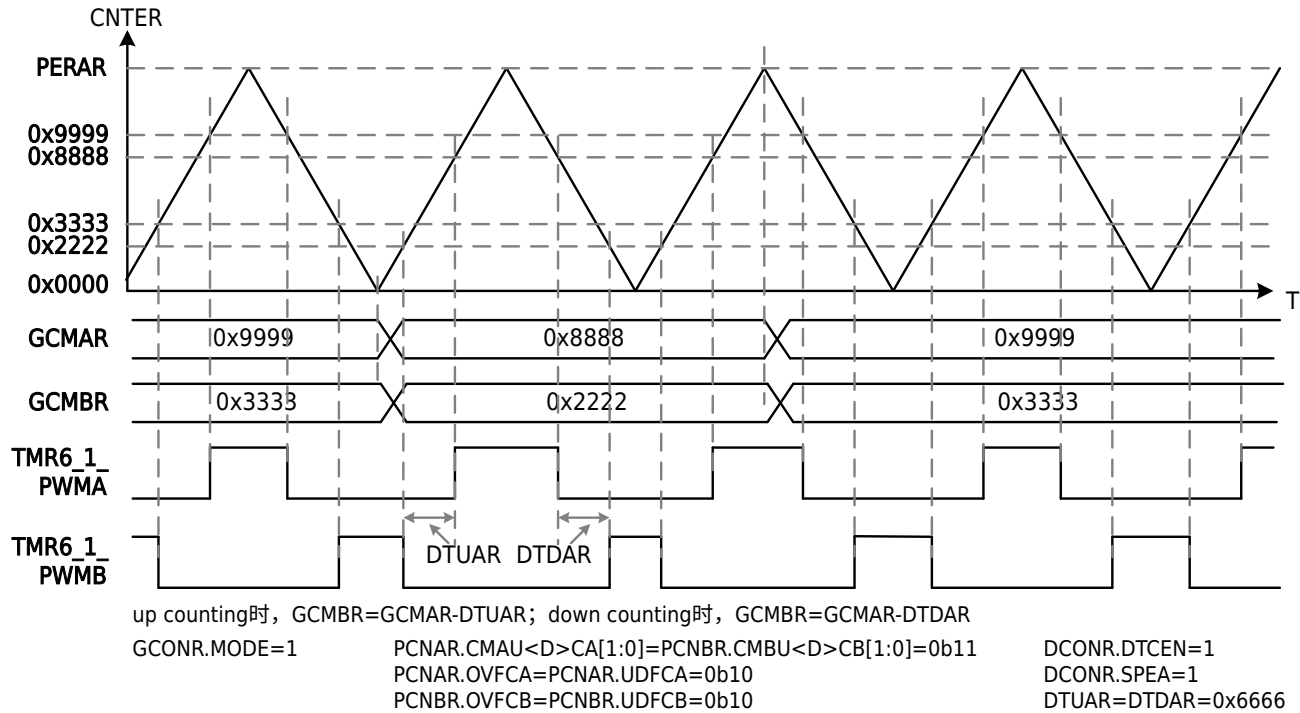


图 20-20 硬件设定 GCMBR 互补 PWM 波输出

20.3.12.4 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。下图是 TMR6_<t>_PWMA 端口的双边非对称 PWM 输出。

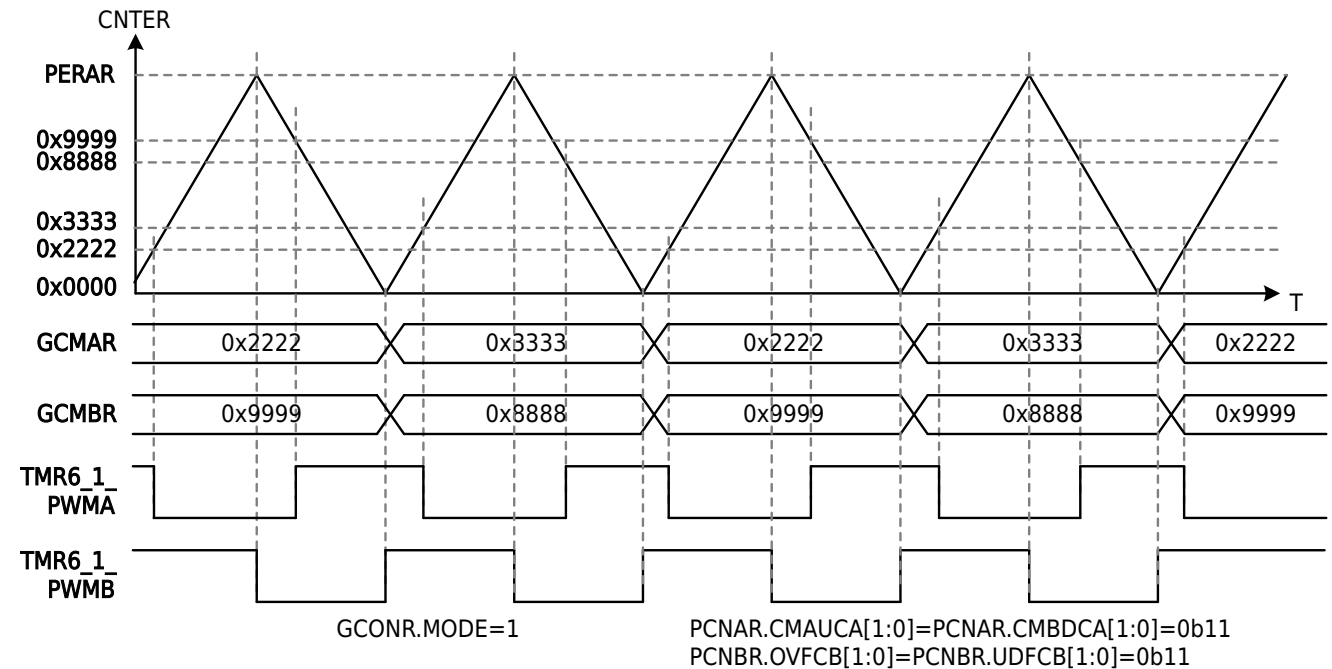


图 20-21 双边非对称 PWM 输出

20.3.12.5 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。图 20-22 是单元 1、单元 2 组合输出单边对齐独立 PWM；图 20-23 是单元 1、单元 2 组合输出双边对称互补 PWM。

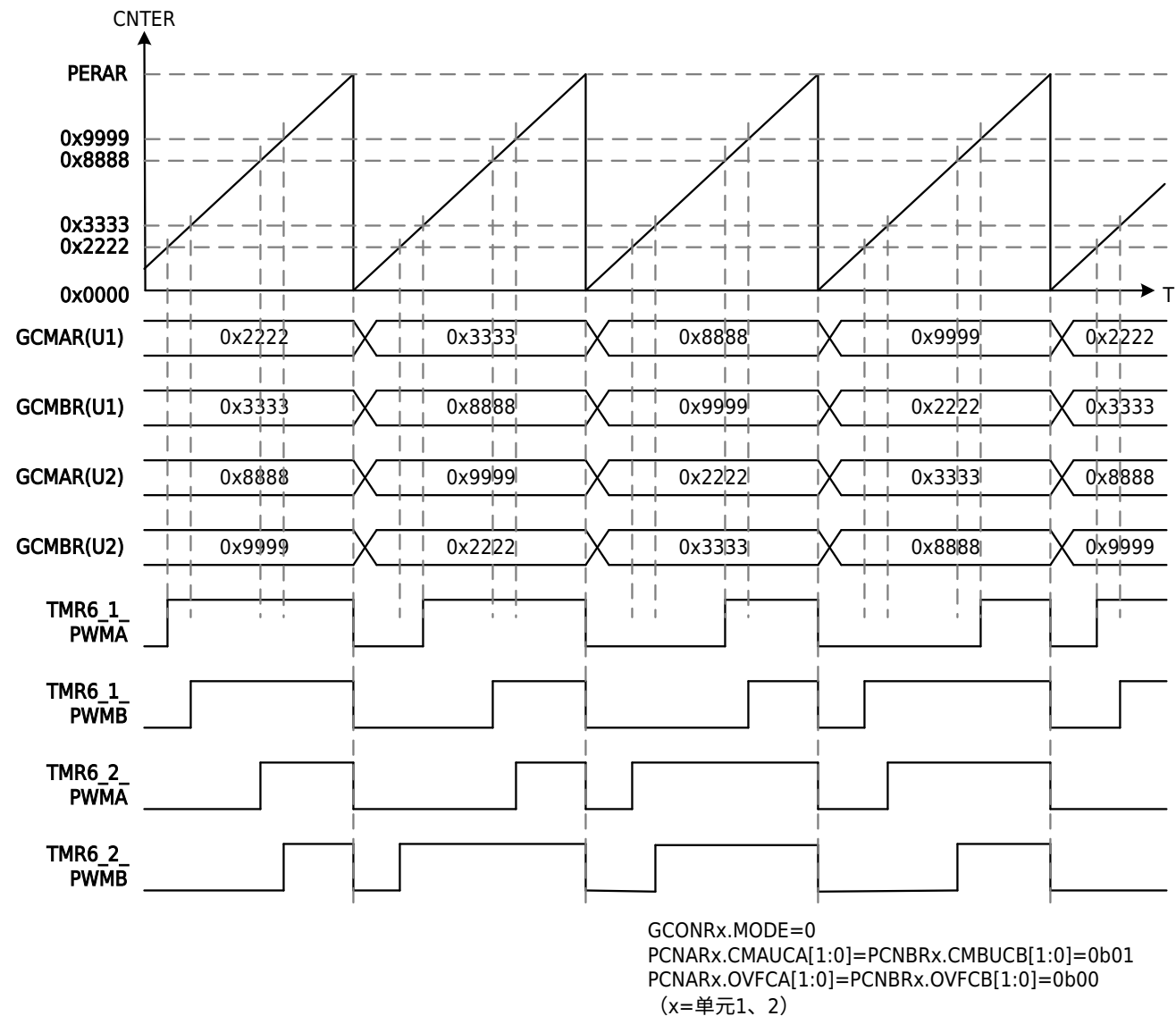


图 20-22 单边对齐独立 PWM

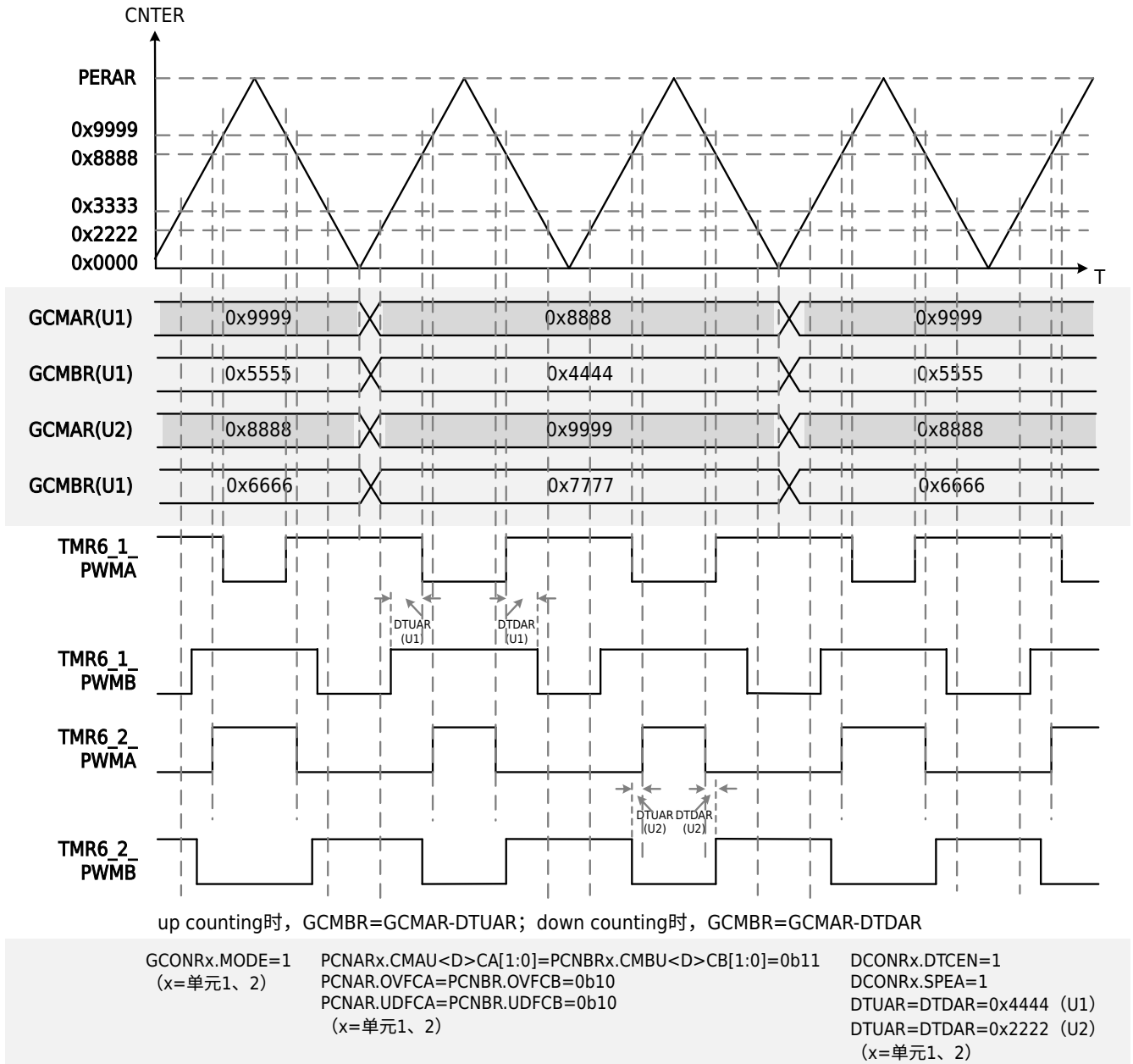


图 20-23 带死区时间双边对称互补 PWM

20.3.13 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。

20.3.13.1 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。

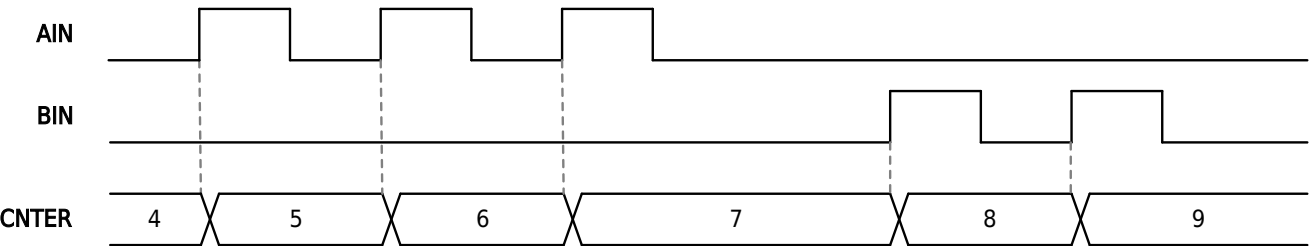
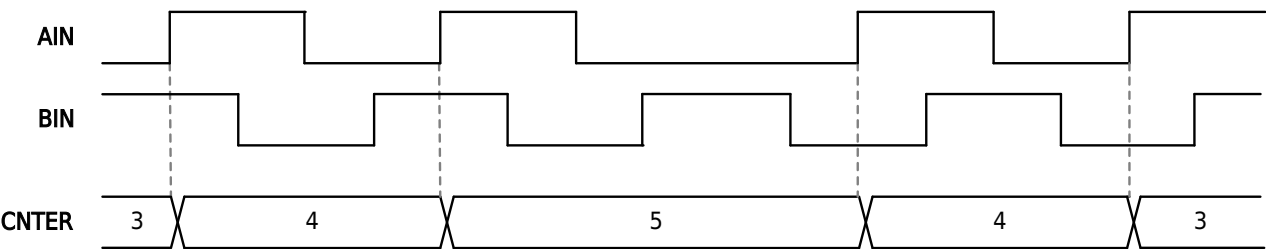


图 20-24 位置模式-基本计数

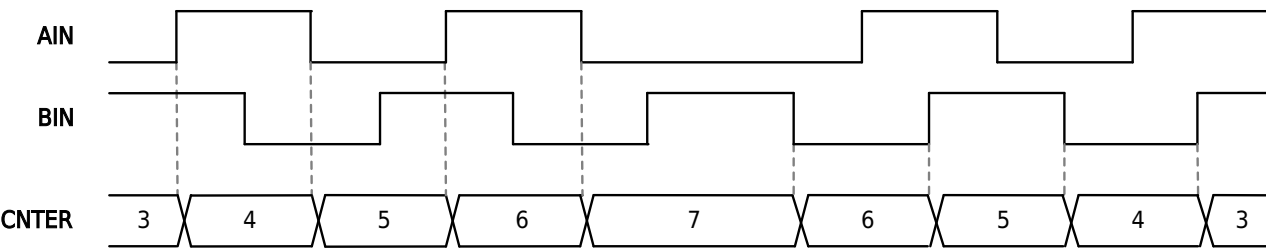
相位差计数

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。



HCUPR.HCUP6=0b1 && HCDOR.HCDO4=0b1

图 20-25 位置计数模式-相位差计数（1 倍计数）



HCUPR.HCUP6=HCUPR.HCUP5=0b1 && HCDOR.HCDO2=HCDOR.HCDO1=0b1

图 20-26 位置计数模式-相位差计数（2 倍计数）

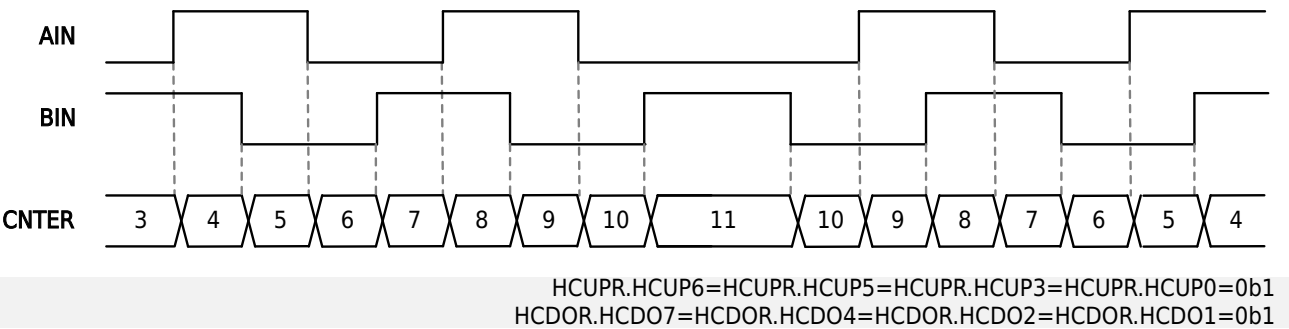


图 20-27 位置计数模式-相位差计数（4 倍计数）

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。

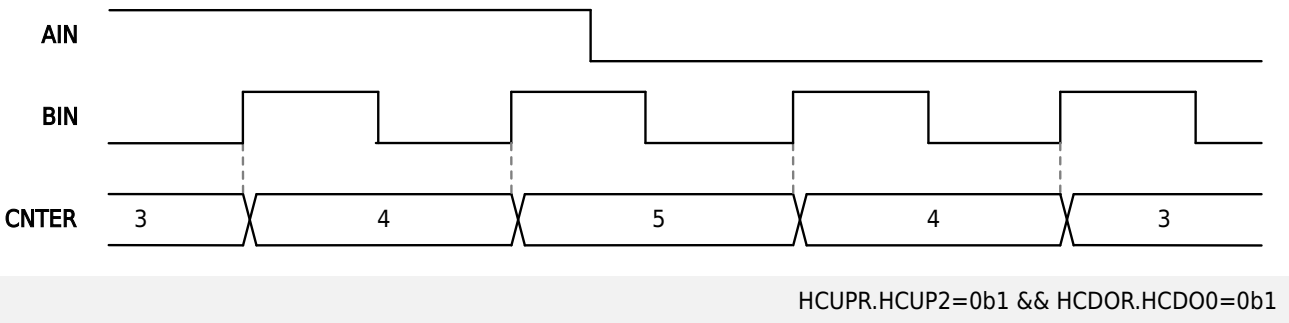


图 20-28 位置计数模式-方向计数

20.3.13.2 公转计数模式

正交编码公转计数模式，是指在 AIN、BIN 计数的基础上，加入 ZIN 的输入事件以实现对公转圈数等的判断。公转计数模式时根据公转定时器的计数方式，可实现 Z 相计数功能、位置溢出计数功能和混合计数功能。

Z 相计数

Z 相计数是指根据 ZIN 的输入，公转计数单元进行计数，同时将位置计数单元清零的计数动作。

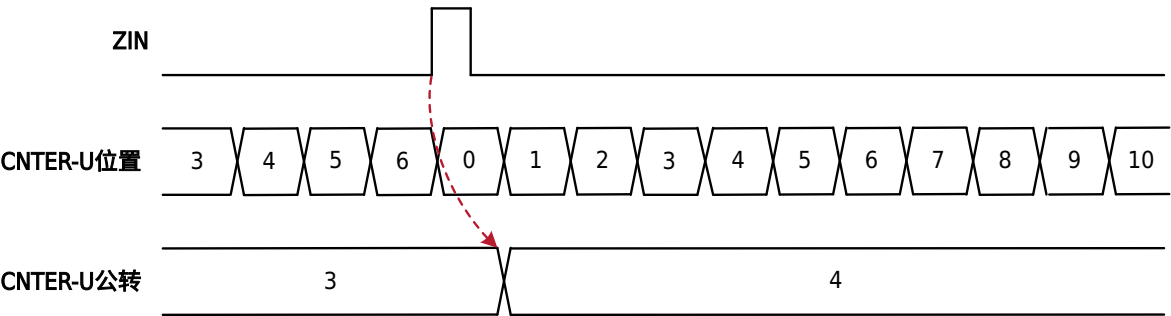


图 20-29 公转计数模式-Z 相计数

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计数器进行一次计数（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=0~1) 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《自动运行系统 (AOS)》章节“Timer6 硬件触发事件选择寄存器 (TMR6_TRGSELx) (x=0~1)”，具体的事件对应关系请参考《中断控制器 (INTC)》章节。

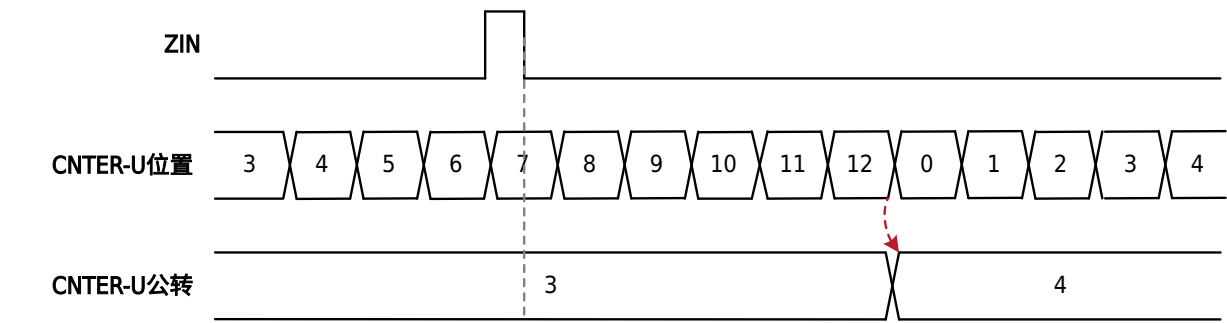


图 20-30 公转计数模式-位置溢出计数

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。

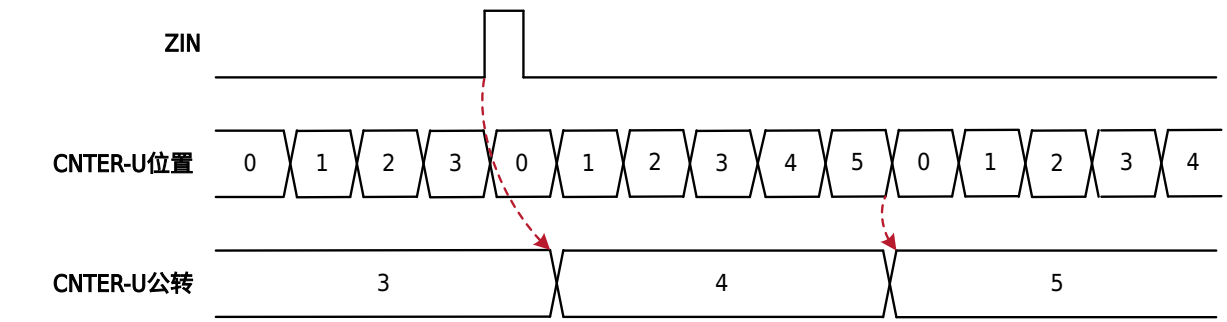


图 20-31 公转计数模式-混合计数

20.3.13.3 Z 相动作屏蔽

在公转计数模式的 Z 相计数功能或混合计数功能时，可以设定在位置定时器的上溢点或下溢点后的几个周期内 (GCONR.ZMSKVAL 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 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 GCONR.ZMSKREV 为 1 时，公转计数单元的 Z 相屏蔽功能使能。下图是公转计数模式混合计数时，在位置计数单元计数上溢后的 4 个计数周期内有 ZIN 相输入时，ZIN 相输入的动作无效，即公转计数单元不计数、位置计数单元不清零；之后再来的 ZIN 相输入正常动作。

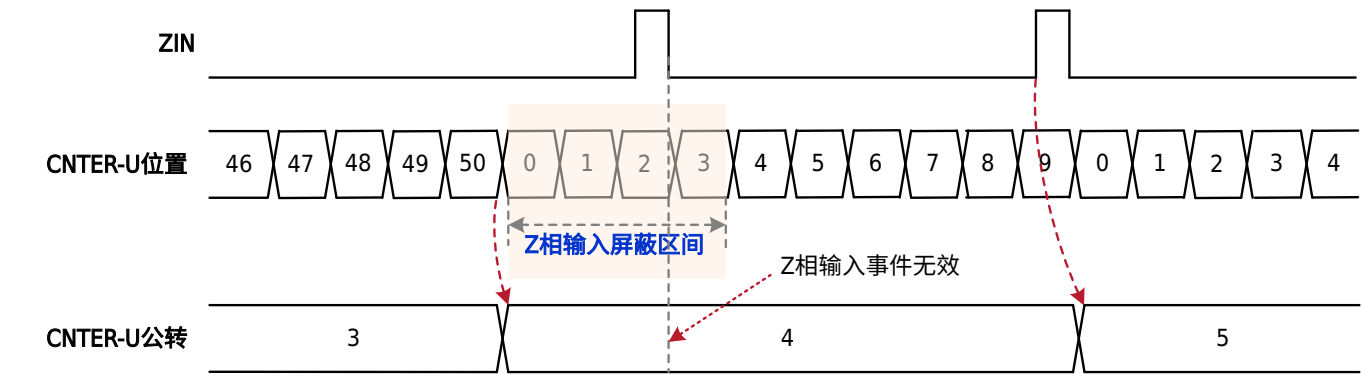


图 20-32 公转计数模式-混合计数 Z 相屏蔽动作例 1

下图是公转计数模式混合计数时，在位置计数单元计数上溢后的第 3 个周期，计数方向发生变化，此时设定的 4 个周期的屏蔽周期变为无效（实际 ZIN 相屏蔽功能维持了 3 个周期），开始向下计数。在位置计数单元发生计数下溢后，ZIN 相屏蔽功能重新开启，维持 4 个周期后变为无效。在 ZIN 相屏蔽期间，ZIN 相的输入功能无效，即公转计数单元不计数、位置计数单元不清零；之后再来的 ZIN 相输入正常动作。

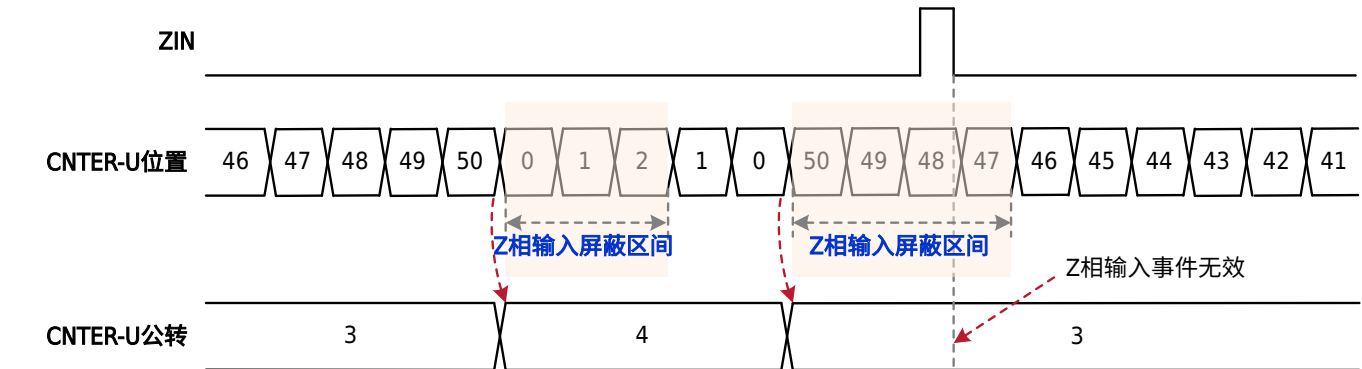


图 20-33 公转计数模式-混合计数 Z 相屏蔽动作例 2

20.3.14 周期间隔响应

Timer6 的 2 个专用比较基准值寄存器（SCMAR、SCMBR），在计数比较匹配时可分别输出专用比较匹配中断 A 信号、专用比较匹配中断 B 信号到 INTC 产生对应的中断；同时可分别输出专用比较匹配事件 A 信号、专用比较匹配事件 B 信号，用于和其他模块关联动作，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 VPERR.PCNTE 和 VPERR.SPPERIA/B 位使能。设定 VPERR.PCNTS 位来指定每隔多少个周期请求信号有效一次，其他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。下图是周期间隔有效请求信号的动作例。

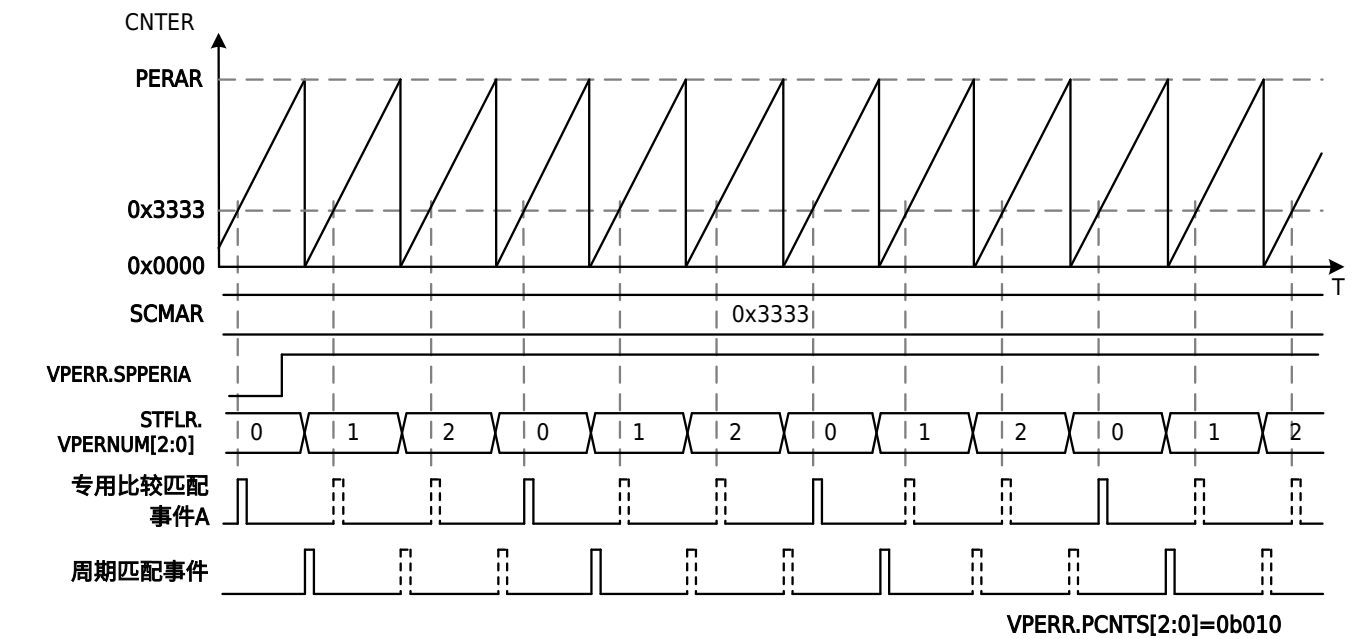


图 20-34 周期间隔有效请求信号动作

20.3.15 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 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=0b01 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=0b00；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）。

20.3.16 功能汇总表

Timer6 的锯齿波模式和三角波 A、B 模式下，主要功能的汇总表如下所示。

表 20-3 不同模式下的功能对比表

PWM 输出功能				锯齿波	三角波	相关主要寄存器
独立 PWM 输出	端口状态控制	启动时		支持	支持	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（Down Counting）		支持	支持	PCNAR.CMADCA
	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	PCNAR.EMBCA
互补 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（Down Counting）		支持	支持	PCNAR.CMADCA
	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=GCMAR-DTUAR GCNBR=GCMAR-DTDAR
	EMB			支持	支持	PCNAR.EMBCA

20.3.17 中断和事件

20.3.17.1 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。

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 ICONR.INTENA~INTENF 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 ICONR.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 ICONR.INTENSAU<D>或 INTENSBU<D>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。

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 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)。

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。此时若设定 ICONR.INTENDTE 位使能中断，则会在该时刻触发死区时间错误中断 (TMR6_<t>_GDTE)。

20.3.17.2 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。

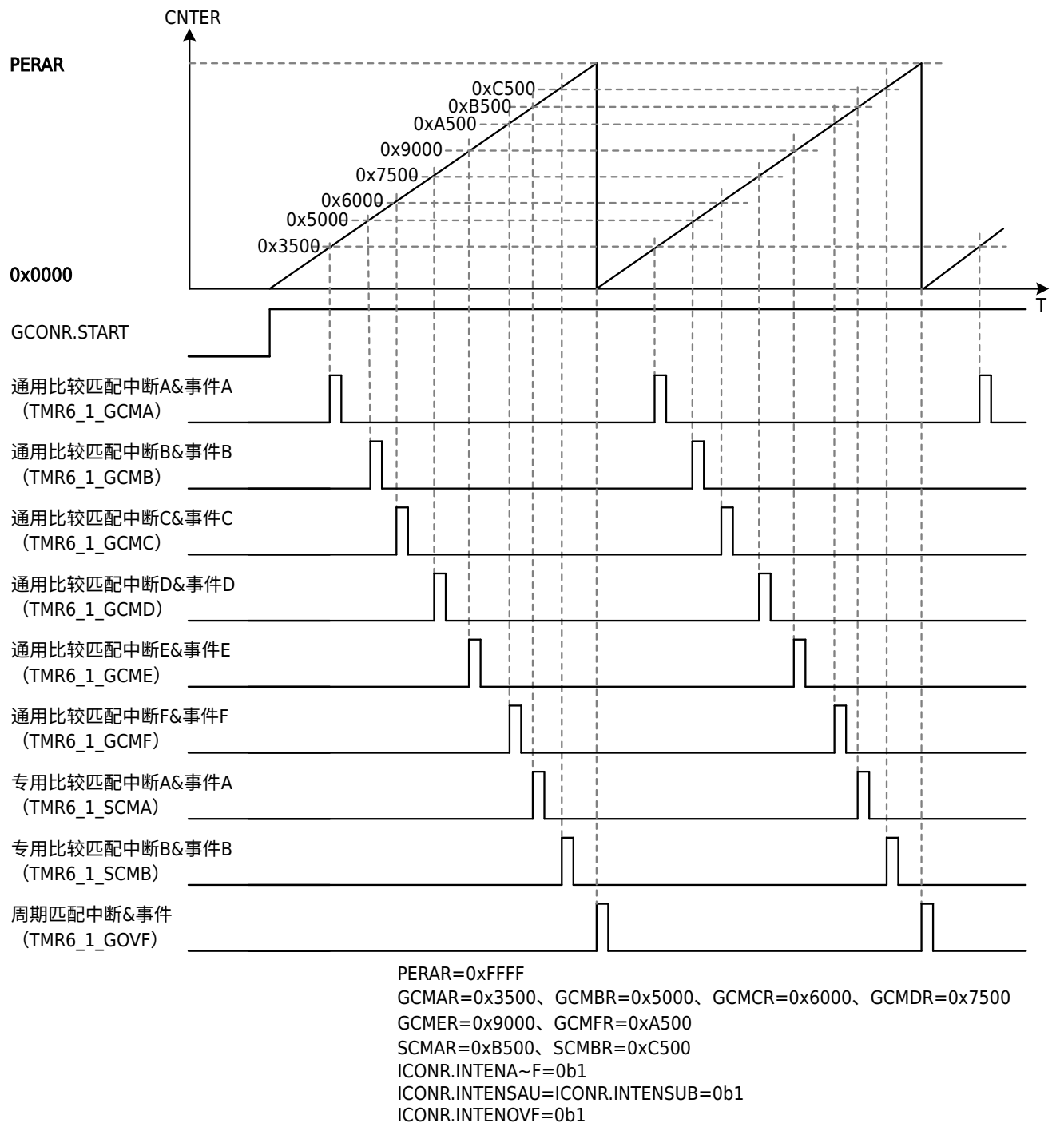


图 20-35 锯齿波模式时中断&事件输出例

20.4 操作示例

下面描述几种典型应用情况下，Timer6 相关寄存器的基本设定，供用户参考。

20.4.1 基本计数及中断动作

- 步骤 1. 设定通用周期基准值 (PERAR)；
- 步骤 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；
- 步骤 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；
- 步骤 4. 设定内部计数时钟分频 (GCONR.CKDIV)；
- 步骤 5. 设定波形模式 (GCONR.MODE)；
- 步骤 6. 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定)；
- 步骤 7. 启动计数器 (GCONR.START=1)。

20.4.2 比较输出及中断动作

- 步骤 1. 设定通用周期基准值 (PERAR)；
- 步骤 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)；
- 步骤 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；
- 步骤 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)；
- 步骤 5. 设定内部计数时钟分频 (GCONR.CKDIV)；
- 步骤 6. 设定波形模式 (GCONR.MODE)；
- 步骤 7. 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定)；
- 步骤 8. 设定各通道比较输出模式 (PCNAR.CAPMDA=0、PCNBR.CAPMDB=0)；
- 步骤 9. 设定各通道输出使能 (PCNAR.OUTENA=1、PCNBR.OUTENB=1)；
- 步骤 10. 启动计数器 (GCONR.START=1)。

20.4.3 捕获输入及中断动作

- 步骤 1. 设定通用周期基准值 (PERAR)；
- 步骤 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；
- 步骤 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)；
- 步骤 4. 设定内部计数时钟分频 (GCONR.CKDIV)；
- 步骤 5. 设定波形模式 (GCONR.MODE)；
- 步骤 6. 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定)；
- 步骤 7. 设定捕获输入模式 (PCNAR.CAPMDA=1、PCNBR.CAPMDB=1)；
- 步骤 8. 启动计数器 (GCONR.START=1)；
- 步骤 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。

20.4.4 缓存传送动作 (周期基准值)

- 步骤 1. 设定需要的通用周期基准值 (PERAR、PERBR、PERCR)；

- 步骤 2. 设定单、双缓存传送方式 (BCONR.BSEP) ;
- 步骤 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) ;
- 步骤 4. 设定内部计数时钟分频 (GCONR.CKDIV) ;
- 步骤 5. 设定波形模式 (GCONR.MODE) ;
- 步骤 6. 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定) ;
- 步骤 7. 设定缓存功能有效 (BCONR.BENP=1) ;
- 步骤 8. 启动计数器 (GCONR.START=1) ;
- 步骤 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)) 。

20.4.5 缓存传送动作 (通用比较基准值)

- 步骤 1. 设定需要的通用比较基准值 (GCMAR、GCMCR、GCMER、GCMBR、GCMDBR、GCMFR) ;
- 步骤 2. 设定各通道单、双缓存传送方式 (BCONR.BSEA、BCONR.BSEB) ;
- 步骤 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) ;
- 步骤 4. 设定内部计数时钟分频 (GCONR.CKDIV) ;
- 步骤 5. 设定波形模式 (GCONR.MODE) ;
- 步骤 6. 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定) ;
- 步骤 7. 设定各通道缓存功能有效 (BCONR.BENA=1、BCONR.BENB=1) ;
- 步骤 8. 启动计数器 (GCONR.START=1) ;
- 步骤 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)、GCMDBR->GCMBR (BCONR.BSEB=0 时)、GCMFR->GCMDBR->GCMBR (BCONR.BSEB=1 时)) 。

20.4.6 缓存传送动作 (专用比较基准值)

- 步骤 1. 设定需要的专用比较基准值 (SCMAR、SCMCR、SCMER、SCMBR、SCMDR、SCMFR) ;
- 步骤 2. 设定各通道单、双缓存传送方式 (BCONR.BSESPA、BCONR.BSESPB) ;
- 步骤 3. 设定各通道缓存传送时间点 (BCONR.BTRUSPA、BCONR.BTRDSPA、BCONR.BTRUSPB、BCONR.BTRDSPB, 每个通道的两个控制位互相独立, 可以同时选择, 均作为缓存传送时间点) (该设定仅在三角波模式时有效, 在锯齿波模式时无效) ;
- 步骤 4. 设定内部计数时钟分频 (GCONR.CKDIV) ;
- 步骤 5. 设定波形模式 (GCONR.MODE) ;
- 步骤 6. 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定) ;
- 步骤 7. 设定各通道缓存功能有效 (BCONR.BENSPA=1、BCONR.BENSPB=1) ;
- 步骤 8. 启动计数器 (GCONR.START=1) ;
- 步骤 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.BSESPA=0 时)、SCMER->SCMCR->SCMAR (BCONR.BSESPA=1 时)、SCMDR->SCMBR (BCONR.BSESPB=0 时)、SCMFR->SCMDR->SCMBR (BCONR.BSESPB=1 时)) 。

20.4.7 缓存传送动作 (死区基准值)

- 步骤 1. 设定需要的死区时间基准值 (DTUAR、DTUBR、DTDAR、DTDBR) ;
- 步骤 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) ;

- 步骤 3. 设定内部计数时钟分频 (GCONR.CKDIV) ;
- 步骤 4. 设定波形模式 (GCONR.MODE) ;
- 步骤 5. 设定计数方向 (仅在锯齿波模式 GCONR.MODE=0 时需要设定) ;
- 步骤 6. 设定缓存功能有效 (DCONR.DTBENU=1、DCONR.DTBEND=1) ;
- 步骤 7. 设定硬件死区功能有效 (DCONR.DTCEN=1) ;
- 步骤 8. 启动计数器 (GCONR.START=1) ;
- 步骤 9. 等待对应缓存传送时间点, 发生缓存动作 (DTUBR->DTUAR、DTDBR->DTDAR) 。

20.4.8 同步启动动作 (软件方式)

- 步骤 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;
- 步骤 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) 。

20.4.9 同步启动动作 (硬件方式)

- 步骤 1. 设定通用周期基准值 (PERAR) ;
- 步骤 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;
- 步骤 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.INTENUF)、计数匹配中断 (ICONR.INTENA~F、ICONR.INTENSAU、ICONR.INTENSAD、ICONR.INTENSBU、ICONR.INTENSBD) 等;
- 步骤 4. 设定硬件启动条件 (通过 HSTAR.HSTAx 选择, x=8~31) ;
- 步骤 5. 设定硬件启动使能 (HSTAR.STAS=1) ;
- 步骤 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) ;
- 步骤 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。

20.4.10 正交编码计数动作 (2 相)

- 步骤 1. 设定通用周期基准值 (PERAR) ;
- 步骤 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;
- 步骤 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.INTENUF)、计数匹配中断 (ICONR.INTENA~F、ICONR.INTENSAU、ICONR.INTENSAD、ICONR.INTENSBU、ICONR.INTENSBD) 等;
- 步骤 4. 设定需要的硬件向上计数条件 (通过 HCUPR.HCUPx 选择, x=0~7) ;
- 步骤 5. 设定需要的硬件向下计数条件 (通过 HCDOR.HCDOx 选择, x=0~7) ;
- 步骤 6. 启动计数器 (GCONR.START=1) ;
- 步骤 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。

20.4.11 正交编码计数动作 (3 相)

- 步骤 1. 参考"正交编码计数动作 (2 相)"章节的步骤 1~5, 对位置计数单元做设定;
- 步骤 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) ;
- 步骤 3. 设定位置计数单元硬件清零使能 (HCLRR.CLES=1) ;
- 步骤 4. 设定公转计数单元的通用周期基准值 (PERAR) ;
- 步骤 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;

- 步骤 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.INTENSBUS、ICONR.INTENSBD) 等；
- 步骤 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.HCUPx 选择，x=16~31，此处的设定事件应和步骤 2 中位置计数单元设定的事件一致)；
- 步骤 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)；
- 步骤 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)；
- 步骤 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《自动运行系统 (AOS)》章节“寄存器-Timer6 硬件触发事件选择寄存器 (TMR6_TRGSELx) (x=0~1)”，上溢事件编号请参见《中断控制器 (INTC)》章节)；
- 步骤 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《自动运行系统 (AOS)》章节“寄存器-Timer6 硬件触发事件选择寄存器 (TMR6_TRGSELx) (x=0~1)”，上溢事件编号请参见《中断控制器 (INTC)》章节)；
- 步骤 12.** 启动公转计数单元计数器 (GCONR.START=1)；
- 步骤 13.** 启动位置计数单元计数器 (GCONR.START=1)；
- 步骤 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。

20.4.12 单路 PWM 输出

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)。

20.4.13 互补 PWM 输出 (软件死区)

- 步骤 1.** 设定通用周期基准值 (PERAR)；
- 步骤 2.** 设定通用比较基准值 A (GCMAR)、通用比较基准值 B (GCMBR)；
- 步骤 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；
- 步骤 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)；
- 步骤 5.** 设定内部计数时钟分频 (GCONR.CKDIV)；
- 步骤 6.** 设定波形模式为三角波模式 (GCONR.MODE=1)；
- 步骤 7.** 设定比较输出模式 (PCNAR.CAPMDA=0、PCNBR.CAPMDB=0)；
- 步骤 8.** 设定输出使能 (PCNAR.OUTENA=1、PCNBR.OUTENB=1)；
- 步骤 9.** 启动计数器 (GCONR.START=1)。

20.4.14 互补 PWM 输出 (硬件死区)

- 步骤 1.** 设定通用周期基准值 (PERAR)；
- 步骤 2.** 设定通用比较基准值 A (GCMAR)、死区时间基准值 (DTUAR、DTDAR)；
- 步骤 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；
- 步骤 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)；
- 步骤 5.** 设定内部计数时钟分频 (GCONR.CKDIV)；
- 步骤 6.** 设定波形模式为三角波模式 (GCONR.MODE=1)；

- 步骤 7. 设定各通道比较输出模式 (PCNAR.CAPMDA=0、PCNBR.CAPMDB=0) ；
- 步骤 8. 设定各通道输出使能 (PCNAR.OUTENA=1、PCNBR.OUTENB=1) ；
- 步骤 9. 设定硬件死区功能有效 (DCONR.DTCEN=1) ；
- 步骤 10. 启动计数器 (GCONR.START=1) 。

20.4.15 EMB 监控及中断动作

- 步骤 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；
- 步骤 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) ；
- 步骤 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) ；
- 步骤 4. 设定选择从 EMB 模块输入的 EMB 事件源 (PCNAR.EMBSA、PCNBR.EMBSB) ；
- 步骤 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) ；
- 步骤 6. 启动计数器 (GCONR.START=1) , EMB 模块实时监控状态。

20.5 寄存器

20.5.1 寄存器总表

- Timer6_1 基地址：0x4003C000
- Timer6_2 基地址：0x4003C400



说明

Timer6 为定时器名称，Timer6_<t> (<t>：1~2) 为该定时器单元。

表 20-4 高级控制定时器 (Timer6) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x0000	TMR6_CNTER	通用计数值寄存器	32
0x0004	TMR6_UPDAR	通用刷新值寄存器	32
0x0040	TMR6_PERAR	通用周期基准值寄存器 A	32
0x0044	TMR6_PERBR	通用周期基准值寄存器 B	32
0x0048	TMR6_PERCR	通用周期基准值寄存器 C	32
0x0080	TMR6_GCMAR	通用比较基准值寄存器 A	32
0x0084	TMR6_GCMBR	通用比较基准值寄存器 B	32
0x0088	TMR6_GCMCR	通用比较基准值寄存器 C	32
0x008C	TMR6_GCMDR	通用比较基准值寄存器 D	32
0x0090	TMR6_GCMER	通用比较基准值寄存器 E	32
0x0094	TMR6_GCMFR	通用比较基准值寄存器 F	32
0x00C0	TMR6_SCMAR	专用比较基准值寄存器 A	32
0x00C4	TMR6_SCMBR	专用比较基准值寄存器 B	32
0x00C8	TMR6_SCMCR	专用比较基准值寄存器 C	32
0x00CC	TMR6_SCMDR	专用比较基准值寄存器 D	32
0x00D0	TMR6_SCMER	专用比较基准值寄存器 E	32
0x00D4	TMR6_SCMFR	专用比较基准值寄存器 F	32
0x0100	TMR6_DTUAR	死区时间基准值寄存器 UA	32
0x0104	TMR6_DTDAR	死区时间基准值寄存器 DA	32
0x0108	TMR6_DTUBR	死区时间基准值寄存器 UB	32
0x010C	TMR6_DTDBR	死区时间基准值寄存器 DB	32
0x0140	TMR6_GCONR	通用控制寄存器	32
0x0144	TMR6_ICONR	中断控制寄存器	32

偏移地址	寄存器	描述	位宽
0x0148	TMR6_BCONR	缓存控制寄存器	32
0x014C	TMR6_DCONR	死区控制寄存器	32
0x0154	TMR6_PCNAR	端口控制寄存器 A	32
0x0158	TMR6_PCNBR	端口控制寄存器 B	32
0x015C	TMR6_FCNGR	滤波控制寄存器	32
0x0160	TMR6_VPERR	有效周期寄存器	32
0x0164	TMR6_STFLR	状态标志寄存器	32
0x0180	TMR6_HSTAR	硬件启动事件选择寄存器	32
0x0184	TMR6_HSTPR	硬件停止事件选择寄存器	32
0x0188	TMR6_HCLRR	硬件清零事件选择寄存器	32
0x018C	TMR6_HUPDR	硬件刷新事件选择寄存器	32
0x0190	TMR6_HCPAR	硬件捕获事件选择寄存器 A	32
0x0194	TMR6_HCPBR	硬件捕获事件选择寄存器 B	32
0x0198	TMR6_HCUPR	硬件递加事件选择寄存器	32
0x019C	TMR6_HCDOR	硬件递减事件选择寄存器	32

Timer6 公共寄存器基地址：0x4003C300

表 20-5 Timer6 公共寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00EC	TMR6_FCNTR	公共滤波控制寄存器	32
0x00F0	TMR6_SSTAR	公共软件同步启动控制寄存器	32
0x00F4	TMR6_SSTPR	公共软件同步停止控制寄存器	32
0x00F8	TMR6_SCLRR	公共软件同步清零控制寄存器	32
0x00FC	TMR6_SUPDR	公共软件同步刷新控制寄存器	32



说明

- 触发选择寄存器 (TMR6_TRGSEL0~1) 是 2 个独立于单元的寄存器，为 2 个单元 Timer6 所共有。
- 软件同步寄存器 (TMR6_SSTAR、TMR6_SSTPR、TMR6_SCLLR、TMR6_SUPDR) 是 4 个独立于单元的寄存器，为 2 个单元 Timer6 所共有。
- 滤波控制寄存器 (TMR6_FCNTR) 是 1 个独立于单元的寄存器，为 2 个单元 Timer6 所共有。

20.5.2 通用计数值寄存器（TMR6_CNTER）

Offset	Bit Position																															
0x0000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																CNT															
Access																																

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	CNT	计数值	当前定时器的计数值	RW

20.5.3 通用刷新值寄存器（TMR6_UPDAR）

Offset	Bit Position																															
0x0004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																UPDA															
Access																	RW															

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	UPDA	刷新值	设定要更新进定时器的刷新值	RW

20.5.4 通用周期基准值寄存器 (TMR6_PERmR) (m=A~C)

偏移地址 (Offset) : 0x0040 (TMR6_PERAR) , 0x0044 (TMR6_PERBR) , 0x0048 (TMR6_PERCR)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000FFFF																															
Name	Reserved																PERm															
Access																																

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	PERm	计数周期值	设定每轮计数的计数周期值及对应缓存值	RW

20.5.5 通用比较基准值寄存器 (TMR6_GCMmR) (m=A~F)

偏移地址 (Offset) : 0x0080 (TMR6_GCMAR) , 0x0084 (TMR6_GCMBR) , 0x0088 (TMR6_GCMCR) , 0x008C (TMR6_GCMDR) , 0x0090 (TMR6_GCMER) , 0x0094 (TMR6_GCMFR)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000FFFF																															
Name	Reserved																GCMm															
Access																	RW															

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	GCMm	计数比较基准值	通用比较基准值设定, 与计数值相等时匹配信号有效	RW

20.5.6 专用比较基准值寄存器 (TMR6_SCmR) (m=A~F)

偏移地址 (Offset) : 0x00C0 (TMR6_SCMAR) , 0x00C4 (TMR6_SCMBR) , 0x00C8 (TMR6_SCmCR) , 0x00CC (TMR6_SCMDR) , 0x00D0 (TMR6_SCMER) , 0x00D4 (TMR6_SCmFR)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000FFFF																															
Name	Reserved																SCMm															
Access																	RW															

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	SCm	专用比较基准值	设定比较基准值及缓存值	RW

20.5.7 死区时间基准值寄存器 (TMR6_DTmnR) (m=U/D, n=A/B)

偏移地址 (Offset) : 0x0100 (TMR6_DTUAR) , 0x0104 (TMR6_DTDAR) , 0x0108 (TMR6_DTUBR) , 0x010C (TMR6_DTDBR)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000FFFF																															
Name	Reserved																DTmn															
Access																	RW															

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	DTmn	死区时间值	死区时间设定值及缓存值	RW

20.5.8 通用控制寄存器 (TMR6_GCONR)

Offset	Bit Position																															
0x0140	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000002																															
Name	Reserved												ZMSKVAL	ZMSKPOS	ZMSKREV	Reserved								OVSTP	CKDIV				Reserved	MODE	DIR	START
Access													RW	RW	RW									RW	RW					RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:20	Reserved	保留	-	-
19:18	ZMSKVAL	Z 相输入屏蔽周期数	0b00: Z 相输入屏蔽功能无效 0b01: 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 0b10: 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 0b11: 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
17	ZMSKPOS	Z 相输入位置定时器选择	0b0: 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 0b1: 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
16	ZMSKREV	Z 相输入公转定时器选择	0b0: 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 0b1: 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
15:9	Reserved	保留	-	-
8	OVSTP	计数溢出停止控制	0b0: 计数器在计数上溢或下溢后, 继续计数 0b1: 计数器在计数上溢或下溢后, 停止计数	RW
7:4	CKDIV	计数时钟选择	0b0000: PCLK0 0b0001: PCLK0/2 0b0010: PCLK0/4 0b0011: PCLK0/8 0b0100: PCLK0/16 0b0101: PCLK0/32 0b0110: PCLK0/64 0b0111: PCLK0/128 0b1000: PCLK0/256 0b1001: PCLK0/512 0b1010: PCLK0/1024 - 其他: 禁止设置	RW
3	Reserved	保留	-	-
2	MODE	计数模式	0b0: 锯齿波模式 0b1: 三角波模式	RW

位/位域	标记	位名	功能描述	读写
1	DIR	计数方向	● 0b0：递减计数 ● 0b1：递加计数	RW
0	START	定时器启动	● 0b0：定时器关闭 ● 0b1：定时器启动  说明 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

20.5.9 中断控制寄存器 (TMR6_ICONR)

Offset	Bit Position																																			
0x0144	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reset	0x00000000																																			
Name	Reserved												INTENSBD	INTENSBU	INTENSAD	INTENSAU	Reserved								INTENDTE	INTENUDF	INTENOVF	INTENF	INTENE	INTEND	INTENC	INTENB	INTENA			
Access													RW	RW	RW	RW									RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:20	Reserved	保留	-	-
19	INTENSBD	专用向下计数中断使能 B	中断条件：向下计数期间，SCMBR 寄存器和计数值 (CNTERR) 相等 ● 0b0：禁止 ● 0b1：使能	RW
18	INTENSBU	专用向上计数中断使能 B	中断条件：向上计数期间，SCMBR 寄存器和计数值 (CNTERR) 相等 ● 0b0：禁止 ● 0b1：使能	RW
17	INTENSAD	专用向下计数中断使能 A	中断条件：向下计数期间，SCMAR 寄存器和计数值 (CNTERR) 相等 ● 0b0：禁止 ● 0b1：使能	RW
16	INTENSAU	专用向上计数中断使能 A	中断条件：向上计数期间，SCMAR 寄存器与计数值 (CNTERR) 相等 ● 0b0：禁止 ● 0b1：使能	RW
15:9	Reserved	保留	-	-
8	INTENDTE	死区时间错误中断使能	中断条件：死区时间错误 ● 0b0：禁止 ● 0b1：使能	RW
7	INTENUDF	下溢中断使能	中断条件：计数器计数下溢 ● 0b0：禁止 ● 0b1：使能	RW
6	INTENOVF	上溢中断使能	中断条件：计数器计数上溢 ● 0b0：禁止 ● 0b1：使能	RW
5	INTENF	计数匹配中断使能 F	中断条件：GCMFR 寄存器与计数值 (CNTERR) 相等 ● 0b0：禁止 ● 0b1：使能	RW
4	INTENE	计数匹配中断使能 E	中断条件：GCMER 寄存器与计数值 (CNTERR) 相等 ● 0b0：禁止 ● 0b1：使能	RW

位/位域	标记	位名	功能描述	读写
3	INTEND	计数匹配中断使能 D	中断条件：GCMDR 寄存器与计数值 (CNTERR) 相等 ● 0b0：禁止 ● 0b1：使能	RW
2	INTENC	计数匹配中断使能 C	中断条件：GCMCR 寄存器与计数值 (CNTERR) 相等 ● 0b0：禁止 ● 0b1：使能	RW
1	INTENB	计数匹配中断使能 B	中断条件：GCMBR 寄存器与计数值 (CNTERR) 相等，或发生输入捕获 ● 0b0：禁止 ● 0b1：使能	RW
0	INTENA	计数匹配中断使能 A	中断条件：GCMAR 寄存器与计数值 (CNTERR) 相等，或发生输入捕获 ● 0b0：禁止 ● 0b1：使能	RW

20.5.10 缓存控制寄存器 (TMR6_BCONR)

Offset	Bit Position																																				
0x0148	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Reset	0x00000000																																				
Name	Reserved								BTRDSPB	BTRUSPB	BSESPB	BENSPB	BTRDSPA	BTRUSPA	BSESPA	BENSPA	Reserved								BTRDP	BTRUP	BSEP	BENP	BTRDB	BTRUB	BSEB	BENB	BTRDA	BTRUA	BSEA	BENA	
Access									RW	RW	RW	RW	RW	RW	RW	RW									RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:24	Reserved	保留	-	-
23	BTRDSPB	专用比较基准值缓存传送时间设定 DB	0b0: 三角波计数模式计数到谷点时, 缓存值不传送 0b1: 三角波计数模式计数到谷点时, 发生一次缓存值传送  说明 锯齿波模式时, 不需要该位控制。	RW
22	BTRUSPB	专用比较基准值缓存传送时间设定 UB	0b0: 三角波计数模式计数到峰点时, 缓存值不传送 0b1: 三角波计数模式计数到峰点时, 发生一次缓存值传送  说明 锯齿波模式时, 不需要该位控制。	RW
21	BSESPB	专用比较基准值缓存传送选择 B	0b0: 单缓存传送 (SCMDR->SCMBR) 0b1: 双缓存传送 (SCMFR->SCMDR->SCMBR)	RW
20	BENSPB	专用比较基准值缓存传送 B	0b0: 缓存传送无效 0b1: 缓存传送使能	RW
19	BTRDSPA	专用比较基准值缓存传送时间设定 DA	0b0: 三角波计数模式计数到谷点时, 缓存值不传送 0b1: 三角波计数模式计数到谷点时, 发生一次缓存值传送  说明 锯齿波模式时, 不需要该位控制。	RW
18	BTRUSPA	专用比较基准值缓存传送时间设定 UA	0b0: 三角波计数模式计数到峰点时, 缓存值不传送 0b1: 三角波计数模式计数到峰点时, 发生一次缓存值传送  说明 锯齿波模式时, 不需要该位控制。	RW
17	BSESPA	专用比较基准值缓存传送选择 A	0b0: 单缓存传送 (SCMCR->SCMAR) 0b1: 双缓存传送 (SCMER->SCMCR->SCMAR)	RW

位/位域	标记	位名	功能描述	读写
16	BENSPA	专用比较基准值缓存传送 A	0b0: 缓存传送无效 0b1: 缓存传送使能	RW
15:12	Reserved	保留	-	-
11	BTRDP	周期值缓存传送时间设定 D	0b0: 三角波计数模式计数到谷点时, 缓存值不传送 0b1: 三角波计数模式计数到谷点时, 发生一次缓存值传送  说明 锯齿波模式时, 不需要该位控制。	RW
10	BTRUP	周期值缓存传送时间设定 U	0b0: 三角波计数模式计数到峰点时, 缓存值不传送 0b1: 三角波计数模式计数到峰点时, 发生一次缓存值传送  说明 锯齿波模式时, 不需要该位控制。	RW
9	BSEP	周期值缓存传送选择	0b0: 单缓存传送 (PERBR->PERAR) 0b1: 双缓存传送 (PERCR->PERBR->PERAR)  说明 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。	RW
8	BENP	周期值缓存传送	0b0: 缓存传送无效 0b1: 缓存传送使能	RW
7	BTRDB	通用比较值缓存传送时间设定 DB	0b0: 三角波计数模式计数到谷点时, 缓存值不传送 0b1: 三角波计数模式计数到谷点时, 发生一次缓存值传送  说明 锯齿波模式时, 不需要该位控制。	RW
6	BTRUB	通用比较值缓存传送时间设定 UB	0b0: 三角波计数模式计数到峰点时, 缓存值不传送 0b1: 三角波计数模式计数到峰点时, 发生一次缓存值传送  说明 锯齿波模式时, 不需要该位控制。	RW
5	BSEB	通用比较值缓存传送选择 B	比较输出功能时: 0b0: 单缓存传送 (GCMDR->GCMBR) 0b1: 双缓存传送 (GCMFR->GCMDR->GCMBR) 捕获输入功能时: 0b0: 单缓存传送 (GCMBR->GCMDR) 0b1: 双缓存传送 (GCMBR->GCMDR->GCMFR)	RW

位/位域	标记	位名	功能描述	读写
4	BENB	通用比较值缓存 传送 B	0b0: 缓存传送无效 0b1: 缓存传送使能	RW
3	BTRDA	通用比较值缓存 传送时间设定 DA	0b0: 三角波计数模式计数到谷点时, 缓存值不传送 0b1: 三角波计数模式计数到谷点时, 发生一次缓存值传送  说明 锯齿波模式时, 不需要该位控制。	RW
2	BTRUA	通用比较值缓存 传送时间设定 UA	0b0: 三角波计数模式计数到峰点时, 缓存值不传送 0b1: 三角波计数模式计数到峰点时, 发生一次缓存值传送  说明 锯齿波模式时, 不需要该位控制。	RW
1	BSEA	通用比较值缓存 传送选择 A	比较输出功能时: 0b0: 单缓存传送 (GCMCR->GCMAR) 0b1: 双缓存传送 (GCMER->GCMCR->GCMAR) 捕获输入功能时: 0b0: 单缓存传送 (GCMAR->GCMCR) 0b1: 双缓存传送 (GCMAR->GCMCR->GCMER)	RW
0	BENA	通用比较值缓存 传送 A	0b0: 缓存传送无效 0b1: 缓存传送使能	RW

20.5.11 死区控制寄存器 (TMR6_DCONR)

Offset	Bit Position																															
0x014C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								DTBTRD	DTBTRU	DTBEND	DTBENU	Reserved	SEPA	DTCEN	
Access																									RW	RW	RW	RW		RW	RW	

位/位域	标记	位名	功能描述	读写
31:8	Reserved	保留	-	-
7	DTBTRD	死区时间值缓存 传送时间设定 D	0b0: 三角波计数模式计数到谷点时, 死区缓存值不传送 0b1: 三角波计数模式计数到谷点时, 发生一次死区缓存值传送  说明 锯齿波模式时, 不需要该位控制。	RW
6	DTBTRU	死区时间值缓存 传送时间设定 U	0b0: 三角波计数模式计数到峰点时, 死区缓存值不传送 0b1: 三角波计数模式计数到峰点时, 发生一次死区缓存值传送  说明 锯齿波模式时, 不需要该位控制。	RW
5	DTBEND	死区时间值缓存 传送 D	0b0: 缓存传送无效 0b1: 缓存传送使能 (DTDBR->DTDAR)	RW
4	DTBENU	死区时间值缓存 传送 U	0b0: 缓存传送无效 0b1: 缓存传送使能 (DTUBR->DTUAR)	RW
3:2	Reserved	保留	-	-
1	SEPA	分离设定	0b0: DTUAR 和 DTDAR 分别设定 0b1: DTUAR 和 DTDAR 的值自动相等	RW
0	DTCEN	死区功能	0b0: 死区功能无效 0b1: 死区功能有效	RW

20.5.12 端口控制寄存器 A (TMR6_PCNAR)

Offset	Bit Position																															
0x0154	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	CAPMDA	Reserved	OUTENA	Reserved	EMBSA	EMBRA	EMBCA	Reserved	FORCA	CMBDCA	CMBUCA	CMADCA	CMAUCA	UDFCA	OVFCA	STPCA	STACA															
Access	RW		RW		RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW															

位/位域	标记	位名	功能描述	读写
31	CAPMDA	功能模式选择 A	0b0: 比较输出功能 0b1: 捕获输入功能	RW
30:29	Reserved	保留	-	-
28	OUTENA	输出使能 A	0b0: Timer6 功能时的 TMR6_<t>_PWMA 端口输出无效 0b1: Timer6 功能时的 TMR6_<t>_PWMA 端口输出有效	RW
27:26	Reserved	保留	-	-
25:24	EMBSA	EMB 事件时端口状态选择 A	0b00: 选择 EMB 的 Group0 控制事件有效 - 其他: 禁止设置	RW
23:22	EMBRA	EMB 释放方式选择 A	0b00: EMBSA 选择的 EMB 控制事件无效时, 立即释放 TMR6_<t>_PWMA 端口 (One Shot) 0b01: EMBSA 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) 0b10: EMBSA 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) 0b11: EMBSA 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
21:20	EMBCA	EMB 事件时端口状态设定 A	0b00: 发生 EMBSA 选择的 EMB 控制事件时, TMR6_<t>_PWMA 端口正常输出 0b01: 发生 EMBSA 选择的 EMB 控制事件时, TMR6_<t>_PWMA 端口输出高阻态 0b10: 发生 EMBSA 选择的 EMB 控制事件时, TMR6_<t>_PWMA 端口输出低电平 0b11: 发生 EMBSA 选择的 EMB 控制事件时, TMR6_<t>_PWMA 端口输出高电平	RW
19:18	Reserved	保留	-	-

位/位域	标记	位名	功能描述	读写
17:16	FORCA	强制端口状态设定 A	0b0x: 不使用强制端口状态设定 0b10: 下周期开始, TMR6_<t>_PWMA 端口输出设定为低电平 0b11: 下周期开始, TMR6_<t>_PWMA 端口输出设定为高电平  说明 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。 该寄存器位可用于实现 PWM 输出占空比 0%或 100% 的控制。	RW
15:14	CMBDCA	向下计数&&比较值 B 匹配时端口状态设定 A	0b00: 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 0b01: 在向下计数期间, 定时器计数值与 GCMBR 相等时, TMR6_<t>_PWMA 端口输出设定为高电平 0b10: 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 0b11: 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	RW
13:12	CMBUCA	向上计数&&比较值 B 匹配时端口状态设定 A	0b00: 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 0b01: 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 0b10: 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 0b11: 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	RW
11:10	CMADCA	向下计数&&比较值 A 匹配时端口状态设定 A	0b00: 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 0b01: 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 0b10: 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 0b11: 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	RW

位/位域	标记	位名	功能描述	读写
9:8	CMAUCA	向上计数&&比较值 A 匹配时端口状态设定 A	● 0b00: 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 ● 0b01: 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 ● 0b10: 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 ● 0b11: 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	RW
7:6	UDFCA	计数下溢端口状态设定 A	● 0b00: 计数下溢时, TMR6_<t>_PWMA 端口输出设定为低电平 ● 0b01: 计数下溢时, TMR6_<t>_PWMA 端口输出设定为高电平 ● 0b10: 计数下溢时, TMR6_<t>_PWMA 端口输出保持先前状态 ● 0b11: 计数下溢时, TMR6_<t>_PWMA 端口输出设定为反转电平	RW
5:4	OVFCA	计数上溢端口状态设定 A	● 0b00: 计数上溢时, TMR6_<t>_PWMA 端口输出设定为低电平 ● 0b01: 计数上溢时, TMR6_<t>_PWMA 端口输出设定为高电平 ● 0b10: 计数上溢时, TMR6_<t>_PWMA 端口输出保持先前状态 ● 0b11: 计数上溢时, TMR6_<t>_PWMA 端口输出设定为反转电平	RW
3:2	STPCA	计数停止端口状态设定 A	● 0b00: 计数停止时, TMR6_<t>_PWMA 端口输出设定为低电平 ● 0b01: 计数停止时, TMR6_<t>_PWMA 端口输出设定为高电平 ● 0b10: 计数停止时, TMR6_<t>_PWMA 端口输出保持先前状态 ● 0b11: 计数停止时, TMR6_<t>_PWMA 端口输出保持先前状态	RW
1:0	STACA	计数开始端口状态设定 A	● 0b00: 计数开始时, TMR6_<t>_PWMA 端口输出设定为低电平 ● 0b01: 计数开始时, TMR6_<t>_PWMA 端口输出设定为高电平 ● 0b10: 计数开始时, TMR6_<t>_PWMA 端口输出保持先前状态 ● 0b11: 计数开始时, TMR6_<t>_PWMA 端口输出保持先前状态	RW

20.5.13 端口控制寄存器 B (TMR6_PCNBR)

Offset	Bit Position																															
0x0158	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	CAPMDB	Reserved	OUTENB	Reserved	EMBSB	EMBRB	EMBCB	Reserved	FORCB	CMBDCB	CMBUCB	CMADCB	CMAUCB	UDFCB	OVFCB	STPCB	STACB															
Access	RW		RW		RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW															

位/位域	标记	位名	功能描述	读写
31	CAPMDB	功能模式选择 B	0b0: 比较输出功能 0b1: 捕获输入功能	RW
30:29	Reserved	保留	-	-
28	OUTENB	输出使能 B	0b0: Timer6 功能时的 TMR6_<t>_PWMB 端口输出无效 0b1: Timer6 功能时的 TMR6_<t>_PWMB 端口输出有效	RW
27:26	Reserved	保留	-	-
25:24	EMBSB	EMB 事件时端口状态选择 B	0b00: 选择 EMB 的 Group0 控制事件有效 - 其他: 禁止设置	RW
23:22	EMBRB	EMB 释放方式选择 B	0b00: EMBSB 选择的 EMB 控制事件无效时, 立即释放 TMR6_<t>_PWMB 端口 (One Shot) 0b01: EMBSB 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) 0b10: EMBSB 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) 0b11: EMBSB 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
21:20	EMBCB	EMB 事件时端口状态设定 B	0b00: 发生 EMBSB 选择的 EMB 控制事件时, TMR6_<t>_PWMB 端口正常输出 0b01: 发生 EMBSB 选择的 EMB 控制事件时, TMR6_<t>_PWMB 端口输出高阻态 0b10: 发生 EMBSB 选择的 EMB 控制事件时, TMR6_<t>_PWMB 端口输出低电平 0b11: 发生 EMBSB 选择的 EMB 控制事件时, TMR6_<t>_PWMB 端口输出高电平	RW
19:18	Reserved	保留	-	-

位/位域	标记	位名	功能描述	读写
17:16	FORCB	强制端口状态设定 B	0b0x: 不使用强制端口状态设定 0b10: 下周期开始, TMR6_<t>_PWMB 端口输出设定为低电平 0b11: 下周期开始, TMR6_<t>_PWMB 端口输出设定为高电平  说明 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。 该寄存器位可用于实现 PWM 输出占空比 0%或 100% 的控制。	RW
15:14	CMBDCB	向下计数&&比较值 B 匹配时端口状态设定 B	0b00: 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 0b01: 在向下计数期间, 定时器计数值与 GCMBR 相等时, TMR6_<t>_PWMB 端口输出设定为高电平 0b10: 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 0b11: 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	RW
13:12	CMBUCB	向上计数&&比较值 B 匹配时端口状态设定 B	0b00: 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 0b01: 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 0b10: 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 0b11: 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	RW
11:10	CMADCB	向下计数&&比较值 A 匹配时端口状态设定 B	0b00: 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 0b01: 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 0b10: 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 0b11: 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	RW

位/位域	标记	位名	功能描述	读写
9:8	CMAUCB	向上计数&&比较值 A 匹配时端口状态设定 B	● 0b00: 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 ● 0b01: 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 ● 0b10: 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 ● 0b11: 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	RW
7:6	UDFCB	计数下溢端口状态设定 B	● 0b00: 计数下溢时, TMR6_<t>_PWMB 端口输出设定为低电平 ● 0b01: 计数下溢时, TMR6_<t>_PWMB 端口输出设定为高电平 ● 0b10: 计数下溢时, TMR6_<t>_PWMB 端口输出保持先前状态 ● 0b11: 计数下溢时, TMR6_<t>_PWMB 端口输出设定为反转电平	RW
5:4	OVFCB	计数上溢端口状态设定 B	● 0b00: 计数上溢时, TMR6_<t>_PWMB 端口输出设定为低电平 ● 0b01: 计数上溢时, TMR6_<t>_PWMB 端口输出设定为高电平 ● 0b10: 计数上溢时, TMR6_<t>_PWMB 端口输出保持先前状态 ● 0b11: 计数上溢时, TMR6_<t>_PWMB 端口输出设定为反转电平	RW
3:2	STPCB	计数停止端口状态设定 B	● 0b00: 计数停止时, TMR6_<t>_PWMB 端口输出设定为低电平 ● 0b01: 计数停止时, TMR6_<t>_PWMB 端口输出设定为高电平 ● 0b10: 计数停止时, TMR6_<t>_PWMB 端口输出保持先前状态 ● 0b11: 计数停止时, TMR6_<t>_PWMB 端口输出保持先前状态	RW
1:0	STACB	计数开始端口状态设定 B	● 0b00: 计数开始时, TMR6_<t>_PWMB 端口输出设定为低电平 ● 0b01: 计数开始时, TMR6_<t>_PWMB 端口输出设定为高电平 ● 0b10: 计数开始时, TMR6_<t>_PWMB 端口输出保持先前状态 ● 0b11: 计数开始时, TMR6_<t>_PWMB 端口输出保持先前状态	RW

20.5.14 滤波控制寄存器 (TMR6_FCNGR)

Offset	Bit Position																																							
0x015C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Reset	0x00000000																																							
Name	Reserved																										RW		NOFICKGB	RW		NOFIENGB	Reserved		RW		NOFICKGA	RW		NOFIENGA
Access																											RW			RW			RW			RW				

位/位域	标记	位名	功能描述	读写
31:7	Reserved	保留	-	-
6:5	NOFICKGB	滤波采样基准时钟选择 GB	0b00: PCLK0 0b01: PCLK0/4 0b10: PCLK0/16 0b11: PCLK0/64	RW
4	NOFIENGB	捕获输入端口滤波 GB	0b0: 该单元 TMR6_<t>_PWMB 输入端口滤波功能无效 0b1: 该单元 TMR6_<t>_PWMB 输入端口滤波功能使能	RW
3	Reserved	保留	-	-
2:1	NOFICKGA	滤波采样基准时钟选择 GA	0b00: PCLK0 0b01: PCLK0/4 0b10: PCLK0/16 0b11: PCLK0/64	RW
0	NOFIENGA	捕获输入端口滤波 GA	0b0: 该单元 TMR6_<t>_PWMA 输入端口滤波功能无效 0b1: 该单元 TMR6_<t>_PWMA 输入端口滤波功能使能	RW

20.5.15 有效周期寄存器 (TMR6_VPERR)

Offset	Bit Position																																	
0x0160	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved											PCNTS		PCNTE	Reserved									SPPERIB	SPPERIA	Reserved								
Access												RW		RW										RW	RW									

位/位域	标记	位名	功能描述	读写
31:21	Reserved	保留	-	-
20:18	PCNTS	有效周期选择	0b000: 有效周期选择功能无效 0b001: 每隔 1 个周期有效一次 0b010: 每隔 2 个周期有效一次 0b011: 每隔 3 个周期有效一次 0b100: 每隔 4 个周期有效一次 0b101: 每隔 5 个周期有效一次 0b110: 每隔 6 个周期有效一次 0b111: 每隔 7 个周期有效一次	RW
17:16	PCNTE	有效周期计数条件选择	0b00: 有效周期选择功能无效 0b01: 锯齿波计数上、下溢点或三角波波谷做为计数条件 0b10: 锯齿波计数上、下溢点或三角波波峰做为计数条件 0b11: 锯齿波计数上、下溢点或三角波波谷、波峰做为计数条件	RW
15:10	Reserved	保留	-	-
9	SPPERIB	专用信号有效周期选择 B	0b0: 有效周期选择功能无效 0b1: 有效周期选择功能使能	RW
8	SPPERIA	专用信号有效周期选择 A	0b0: 有效周期选择功能无效 0b1: 有效周期选择功能使能	RW
7:0	Reserved	保留	-	-

20.5.16 状态标志寄存器 (TMR6_STFLR)

Offset	Bit Position																																
0x0164	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x80000000																																
Name	DIRF	Reserved			CMBF2	CMAF2	Reserved			VPERNUM			Reserved					CMSBDF	CMSBUF	CMSADF	CMSAUF	DTEF	UDFF	OVSF	CMFF	CMEF	CMDF	CMCF	CMBF	CMAF			
Access	RO				RW	RW				RO									RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31	DIRF	计数方向	0b0: 递减计数 0b1: 递加计数	RO
30:28	Reserved	保留	-	-
27	CMBF2	再次捕获 B	0b0: 计数匹配 B 标志 CMBF 为 1 后未发生 TMR6_<t>_PWMB 捕获完成动作 0b1: 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
26	CMAF2	再次捕获 A	0b0: 计数匹配 A 标志 CMAF 为 1 后未发生 TMR6_<t>_PWMA 捕获完成动作 0b1: 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
25:24	Reserved	保留	-	-
23:21	VPERNUM	周期次数	有效周期选择功能使能时, 计数后的周期次数	RO
20:13	Reserved	保留	-	-
12	CMSBDF	向下计数专用比较基准值匹配 B	0b0: 向下计数时, SCMBR 寄存器的值与计数值不相等 0b1: 向下计数时, SCMBR 寄存器的值与计数值相等	RW
11	CMSBUF	向上计数专用比较基准值匹配 B	0b0: 向上计数时, SCMBR 寄存器的值与计数值不相等 0b1: 向上计数时, SCMBR 寄存器的值与计数值相等	RW
10	CMSADF	向下计数专用比较基准值匹配 A	0b0: 向下计数时, SCMAR 寄存器的值与计数值不相等 0b1: 向下计数时, SCMAR 寄存器的值与计数值相等	RW
9	CMSAUF	向上计数专用比较基准值匹配 A	0b0: 向上计数时, SCMAR 寄存器的值与计数值不相等 0b1: 向上计数时, SCMAR 寄存器的值与计数值相等	RW
8	DTEF	死区时间错误	0b0: 未发生死区时间错误 0b1: 发生死区时间错误	RW
7	UDFF	下溢匹配	0b0: 未发生锯齿波下溢或三角波计数到谷点 0b1: 发生锯齿波下溢或三角波计数到谷点	RW

位/位域	标记	位名	功能描述	读写
6	OVFF	上溢匹配	0b0: 未发生锯齿波上溢或三角波计数到峰点 0b1: 发生锯齿波上溢或三角波计数到峰点	RW
5	CMFF	计数匹配 F	0b0: GCMFR 寄存器的值与计数值不相等 0b1: GCMFR 寄存器的值与计数值相等	RW
4	CMEF	计数匹配 E	0b0: GCMEF 寄存器的值与计数值不相等 0b1: GCMEF 寄存器的值与计数值相等	RW
3	CMDF	计数匹配 D	0b0: GCMDR 寄存器的值与计数值不相等 0b1: GCMDR 寄存器的值与计数值相等	RW
2	CMCF	计数匹配 C	0b0: GCMCR 寄存器的值与计数值不相等 0b1: GCMCR 寄存器的值与计数值相等	RW
1	CMBF	计数匹配 B	0b0: 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 0b1: 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
0	CMAF	计数匹配 A	0b0: 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 0b1: 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

20.5.17 硬件启动事件选择寄存器 (TMR6_HSTAR)

Offset	Bit Position																																
0x0180	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved												HSTA19	HSTA18	HSTA17	HSTA16	Reserved						HSTA9	HSTA8	STAS	Reserved				HSTA3	HSTA2	HSTA1	HSTA0
Access													RW	RW	RW	RW							RW	RW	RW					RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:20	Reserved	保留	-	-
19	HSTA19	硬件启动条件 19	条件：TMR6_TRIGB 端口上采样到下降沿 ● 0b0：条件匹配时，硬件启动无效 ● 0b1：条件匹配时，硬件启动有效	RW
18	HSTA18	硬件启动条件 18	条件：TMR6_TRIGB 端口上采样到上升沿 ● 0b0：条件匹配时，硬件启动无效 ● 0b1：条件匹配时，硬件启动有效	RW
17	HSTA17	硬件启动条件 17	条件：TMR6_TRIGA 端口上采样到下降沿 ● 0b0：条件匹配时，硬件启动无效 ● 0b1：条件匹配时，硬件启动有效	RW
16	HSTA16	硬件启动条件 16	条件：TMR6_TRIGA 端口上采样到上升沿 ● 0b0：条件匹配时，硬件启动无效 ● 0b1：条件匹配时，硬件启动有效	RW
15:10	Reserved	保留	-	-
9	HSTA9	硬件启动条件 9	条件：内部硬件触发事件 1 有效 ● 0b0：条件匹配时，硬件启动无效 ● 0b1：条件匹配时，硬件启动有效	RW
8	HSTA8	硬件启动条件 8	条件：内部硬件触发事件 0 有效 ● 0b0：条件匹配时，硬件启动无效 ● 0b1：条件匹配时，硬件启动有效	RW
7	STAS	硬件启动使能	● 0b0：硬件启动无效 ● 0b1：硬件启动有效  说明 硬件启动有效时，SSTAR 的设定无效。	RW
6:4	Reserved	保留	-	-
3	HSTA3	硬件启动条件 3	条件：TMR6_<t>_PWMB 端口上采样到下降沿 ● 0b0：条件匹配时，硬件启动无效 ● 0b1：条件匹配时，硬件启动有效	RW
2	HSTA2	硬件启动条件 2	条件：TMR6_<t>_PWMB 端口上采样到上升沿 ● 0b0：条件匹配时，硬件启动无效 ● 0b1：条件匹配时，硬件启动有效	RW

位/位域	标记	位名	功能描述	读写
1	HSTA1	硬件启动条件 1	条件：TMR6_<t>_PWMA 端口上采样到下降沿 ● 0b0：条件匹配时，硬件启动无效 ● 0b1：条件匹配时，硬件启动有效	RW
0	HSTA0	硬件启动条件 0	条件：TMR6_<t>_PWMA 端口上采样到上升沿 ● 0b0：条件匹配时，硬件启动无效 ● 0b1：条件匹配时，硬件启动有效	RW

20.5.18 硬件停止事件选择寄存器 (TMR6_HSTPR)

Offset	Bit Position																															
0x0184	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved												HSTP19	HSTP18	HSTP17	HSTP16	Reserved						HSTP9	HSTP8	STPS	Reserved			HSTP3	HSTP2	HSTP1	HSTP0
Access													RW	RW	RW	RW							RW	RW	RW				RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:20	Reserved	保留	-	-
19	HSTP19	硬件停止条件 19	条件：TMR6_TRIGB 端口上采样到下降沿 ● 0b0：条件匹配时，硬件停止无效 ● 0b1：条件匹配时，硬件停止有效	RW
18	HSTP18	硬件停止条件 18	条件：TMR6_TRIGB 端口上采样到上升沿 ● 0b0：条件匹配时，硬件停止无效 ● 0b1：条件匹配时，硬件停止有效	RW
17	HSTP17	硬件停止条件 17	条件：TMR6_TRIGA 端口上采样到下降沿 ● 0b0：条件匹配时，硬件停止无效 ● 0b1：条件匹配时，硬件停止有效	RW
16	HSTP16	硬件停止条件 16	条件：TMR6_TRIGA 端口上采样到上升沿 ● 0b0：条件匹配时，硬件停止无效 ● 0b1：条件匹配时，硬件停止有效	RW
15:10	Reserved	保留	-	-
9	HSTP9	硬件停止条件 9	条件：内部硬件触发事件 1 有效 ● 0b0：条件匹配时，硬件停止无效 ● 0b1：条件匹配时，硬件停止有效	RW
8	HSTP8	硬件停止条件 8	条件：内部硬件触发事件 0 有效 ● 0b0：条件匹配时，硬件停止无效 ● 0b1：条件匹配时，硬件停止有效	RW
7	STPS	硬件停止使能	● 0b0：硬件停止无效 ● 0b1：硬件停止有效  说明 硬件停止有效时，SSTPR 的设定无效。	RW
6:4	Reserved	保留	-	-
3	HSTP3	硬件停止条件 3	条件：TMR6_<t>_PWMB 端口上采样到下降沿 ● 0b0：条件匹配时，硬件停止无效 ● 0b1：条件匹配时，硬件停止有效	RW
2	HSTP2	硬件停止条件 2	条件：TMR6_<t>_PWMB 端口上采样到上升沿 ● 0b0：条件匹配时，硬件停止无效 ● 0b1：条件匹配时，硬件停止有效	RW

位/位域	标记	位名	功能描述	读写
1	HSTP1	硬件停止条件 1	条件：TMR6_<t>_PWMA 端口上采样到下降沿 ● 0b0：条件匹配时，硬件停止无效 ● 0b1：条件匹配时，硬件停止有效	RW
0	HSTP0	硬件停止条件 0	条件：TMR6_<t>_PWMA 端口上采样到上升沿 ● 0b0：条件匹配时，硬件停止无效 ● 0b1：条件匹配时，硬件停止有效	RW

20.5.19 硬件清零事件选择寄存器 (TMR6_HCLRR)

Offset	Bit Position																																
0x0188	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved												HCLE19	HCLE18	HCLE17	HCLE16	Reserved					HCLE11	HCLE10	HCLE9	HCLE8	CLES	Reserved			HCLE3	HCLE2	HCLE1	HCLE0
Access													RW	RW	RW	RW						RW	RW	RW	RW	RW				RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:20	Reserved	保留	-	-
19	HCLE19	硬件清零条件 19	条件：TMR6_TRIGB 端口上采样到下降沿 ● 0b0：条件匹配时，硬件清零无效 ● 0b1：条件匹配时，硬件清零有效	RW
18	HCLE18	硬件清零条件 18	条件：TMR6_TRIGB 端口上采样到上升沿 ● 0b0：条件匹配时，硬件清零无效 ● 0b1：条件匹配时，硬件清零有效	RW
17	HCLE17	硬件清零条件 17	条件：TMR6_TRIGA 端口上采样到下降沿 ● 0b0：条件匹配时，硬件清零无效 ● 0b1：条件匹配时，硬件清零有效	RW
16	HCLE16	硬件清零条件 16	条件：TMR6_TRIGA 端口上采样到上升沿 ● 0b0：条件匹配时，硬件清零无效 ● 0b1：条件匹配时，硬件清零有效	RW
15:10	Reserved	保留	-	-
9	HCLE9	硬件清零条件 9	条件：内部硬件触发事件 1 有效 ● 0b0：条件匹配时，硬件清零无效 ● 0b1：条件匹配时，硬件清零有效	RW
8	HCLE8	硬件清零条件 8	条件：内部硬件触发事件 0 有效 ● 0b0：条件匹配时，硬件清零无效 ● 0b1：条件匹配时，硬件清零有效	RW
7	CLES	硬件清零使能	● 0b0：硬件清零无效 ● 0b1：硬件清零有效  说明 硬件清零有效时，SCLRR 的设置无效。	RW
6:4	Reserved	保留	-	-
3	HCLE3	硬件清零条件 3	条件：TMR6_<t>_PWMB 端口上采样到下降沿 ● 0b0：条件匹配时，硬件清零无效 ● 0b1：条件匹配时，硬件清零有效	RW
2	HCLE2	硬件清零条件 2	条件：TMR6_<t>_PWMB 端口上采样到上升沿 ● 0b0：条件匹配时，硬件清零无效 ● 0b1：条件匹配时，硬件清零有效	RW

位/位域	标记	位名	功能描述	读写
1	HCLE1	硬件清零条件 1	条件：TMR6_<t>_PWMA 端口上采样到下降沿 ● 0b0：条件匹配时，硬件清零无效 ● 0b1：条件匹配时，硬件清零有效	RW
0	HCLE0	硬件清零条件 0	条件：TMR6_<t>_PWMA 端口上采样到上升沿 ● 0b0：条件匹配时，硬件清零无效 ● 0b1：条件匹配时，硬件清零有效	RW

20.5.20 硬件刷新事件选择寄存器 (TMR6_HUPDR)

Offset	Bit Position																															
0x018C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved												HUPD19	HUPD18	HUPD17	HUPD16	Reserved						HUPD9	HUPD8	UPDS	Reserved			HUPD3	HUPD2	HUPD1	HUPD0
Access													RW	RW	RW	RW							RW	RW	RW				RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:20	Reserved	保留	-	-
19	HUPD19	硬件刷新条件 19	条件：TMR6_TRIGB 端口上采样到下降沿 ● 0b0：条件匹配时，硬件刷新无效 ● 0b1：条件匹配时，硬件刷新有效	RW
18	HUPD18	硬件刷新条件 18	条件：TMR6_TRIGB 端口上采样到上升沿 ● 0b0：条件匹配时，硬件刷新无效 ● 0b1：条件匹配时，硬件刷新有效	RW
17	HUPD17	硬件刷新条件 17	条件：TMR6_TRIGA 端口上采样到下降沿 ● 0b0：条件匹配时，硬件刷新无效 ● 0b1：条件匹配时，硬件刷新有效	RW
16	HUPD16	硬件刷新条件 16	条件：TMR6_TRIGA 端口上采样到上升沿 ● 0b0：条件匹配时，硬件刷新无效 ● 0b1：条件匹配时，硬件刷新有效	RW
15:10	Reserved	保留	-	-
9	HUPD9	硬件刷新条件 9	条件：内部硬件触发事件 1 有效 ● 0b0：条件匹配时，硬件刷新无效 ● 0b1：条件匹配时，硬件刷新有效	RW
8	HUPD8	硬件刷新条件 8	条件：内部硬件触发事件 0 有效 ● 0b0：条件匹配时，硬件刷新无效 ● 0b1：条件匹配时，硬件刷新有效	RW
7	UPDS	硬件刷新使能	● 0b0：硬件刷新无效 ● 0b1：硬件刷新有效  说明 硬件刷新有效时，SUPDR 的设定无效。	RW
6:4	Reserved	保留	-	-
3	HUPD3	硬件刷新条件 3	条件：TMR6_PWMB 端口上采样到下降沿 ● 0b0：条件匹配时，硬件刷新无效 ● 0b1：条件匹配时，硬件刷新有效	RW
2	HUPD2	硬件刷新条件 2	条件：TMR6_PWMB 端口上采样到上升沿 ● 0b0：条件匹配时，硬件刷新无效 ● 0b1：条件匹配时，硬件刷新有效	RW

位/位域	标记	位名	功能描述	读写
1	HUPD1	硬件刷新条件 1	条件：TMR6_<t>_PWMA 端口上采样到下降沿 ● 0b0：条件匹配时，硬件刷新无效 ● 0b1：条件匹配时，硬件刷新有效	RW
0	HUPD0	硬件刷新条件 0	条件：TMR6_<t>_PWMA 端口上采样到上升沿 ● 0b0：条件匹配时，硬件刷新无效 ● 0b1：条件匹配时，硬件刷新有效	RW

20.5.21 硬件捕获事件选择寄存器 A (TMR6_HCPAR)

Offset	Bit Position																																							
0x0190	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Reset	0x00000000																																							
Name	Reserved				HCPA25		HCPA24		Reserved				HCPA19		HCPA18		HCPA17		HCPA16		Reserved				HCPA9		HCPA8		Reserved				HCPA3		HCPA2		HCPA1		HCPA0	
Access					RW		RW						RW		RW		RW		RW						RW		RW						RW		RW		RW		RW	

位/位域	标记	位名	功能描述	读写
31:26	Reserved	保留	-	-
25	HCPA25	硬件捕获 A 条件 25	条件：内部信号 TMR6_<t>_XOR 上采样到下降沿 ● 0b0：条件匹配时，硬件捕获 A 无效 ● 0b1：条件匹配时，硬件捕获 A 有效	RW
24	HCPA24	硬件捕获 A 条件 24	条件：内部信号 TMR6_<t>_XOR 上采样到上升沿 ● 0b0：条件匹配时，硬件捕获 A 无效 ● 0b1：条件匹配时，硬件捕获 A 有效	RW
23:20	Reserved	保留	-	-
19	HCPA19	硬件捕获 A 条件 19	条件：TMR6_TRIGB 端口上采样到下降沿 ● 0b0：条件匹配时，硬件捕获 A 无效 ● 0b1：条件匹配时，硬件捕获 A 有效	RW
18	HCPA18	硬件捕获 A 条件 18	条件：TMR6_TRIGB 端口上采样到上升沿 ● 0b0：条件匹配时，硬件捕获 A 无效 ● 0b1：条件匹配时，硬件捕获 A 有效	RW
17	HCPA17	硬件捕获 A 条件 17	条件：TMR6_TRIGA 端口上采样到下降沿 ● 0b0：条件匹配时，硬件捕获 A 无效 ● 0b1：条件匹配时，硬件捕获 A 有效	RW
16	HCPA16	硬件捕获 A 条件 16	条件：TMR6_TRIGA 端口上采样到上升沿 ● 0b0：条件匹配时，硬件捕获 A 无效 ● 0b1：条件匹配时，硬件捕获 A 有效	RW
15:10	Reserved	保留	-	-
9	HCPA9	硬件捕获 A 条件 9	条件：内部硬件触发事件 1 有效 ● 0b0：条件匹配时，硬件捕获 A 无效 ● 0b1：条件匹配时，硬件捕获 A 有效	RW
8	HCPA8	硬件捕获 A 条件 8	条件：内部硬件触发事件 0 有效 ● 0b0：条件匹配时，硬件捕获 A 无效 ● 0b1：条件匹配时，硬件捕获 A 有效	RW
7:4	Reserved	保留	-	-
3	HCPA3	硬件捕获 A 条件 3	条件：TMR6_<t>_PWMB 端口上采样到下降沿 ● 0b0：条件匹配时，硬件捕获 A 无效 ● 0b1：条件匹配时，硬件捕获 A 有效	RW

位/位域	标记	位名	功能描述	读写
2	HCPA2	硬件捕获 A 条件 2	条件：TMR6_<t>_PWMB 端口上采样到上升沿 ● 0b0：条件匹配时，硬件捕获 A 无效 ● 0b1：条件匹配时，硬件捕获 A 有效	RW
1	HCPA1	硬件捕获 A 条件 1	条件：TMR6_<t>_PWMA 端口上采样到下降沿 ● 0b0：条件匹配时，硬件捕获 A 无效 ● 0b1：条件匹配时，硬件捕获 A 有效	RW
0	HCPA0	硬件捕获 A 条件 0	条件：TMR6_<t>_PWMA 端口上采样到上升沿 ● 0b0：条件匹配时，硬件捕获 A 无效 ● 0b1：条件匹配时，硬件捕获 A 有效	RW

 **说明**

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。

20.5.22 硬件捕获事件选择寄存器 B (TMR6_HCPBR)

Offset	Bit Position																																							
0x0194	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Reset	0x00000000																																							
Name	Reserved				HCPB25		HCPB24		Reserved				HCPB19		HCPB18		HCPB17		HCPB16		Reserved				HCPB9		HCPB8		Reserved				HCPB3		HCPB2		HCPB1		HCPB0	
Access					RW		RW						RW		RW		RW		RW						RW		RW						RW		RW		RW		RW	

位/位域	标记	位名	功能描述	读写
31:26	Reserved	保留	-	-
25	HCPB25	硬件捕获 B 条件 25	条件：内部信号 TMR6_<t>_XOR 上采样到下降沿 ● 0b0：条件匹配时，硬件捕获 B 无效 ● 0b1：条件匹配时，硬件捕获 B 有效	RW
24	HCPB24	硬件捕获 B 条件 24	条件：内部信号 TMR6_<t>_XOR 上采样到上升沿 ● 0b0：条件匹配时，硬件捕获 B 无效 ● 0b1：条件匹配时，硬件捕获 B 有效	RW
23:20	Reserved	保留	-	-
19	HCPB19	硬件捕获 B 条件 19	条件：TMR6_TRIGB 端口上采样到下降沿 ● 0b0：条件匹配时，硬件捕获 B 无效 ● 0b1：条件匹配时，硬件捕获 B 有效	RW
18	HCPB18	硬件捕获 B 条件 18	条件：TMR6_TRIGB 端口上采样到上升沿 ● 0b0：条件匹配时，硬件捕获 B 无效 ● 0b1：条件匹配时，硬件捕获 B 有效	RW
17	HCPB17	硬件捕获 B 条件 17	条件：TMR6_TRIGA 端口上采样到下降沿 ● 0b0：条件匹配时，硬件捕获 B 无效 ● 0b1：条件匹配时，硬件捕获 B 有效	RW
16	HCPB16	硬件捕获 B 条件 16	条件：TMR6_TRIGA 端口上采样到上升沿 ● 0b0：条件匹配时，硬件捕获 B 无效 ● 0b1：条件匹配时，硬件捕获 B 有效	RW
15:10	Reserved	保留	-	-
9	HCPB9	硬件捕获 B 条件 9	条件：内部硬件触发事件 1 有效 ● 0b0：条件匹配时，硬件捕获 B 无效 ● 0b1：条件匹配时，硬件捕获 B 有效	RW
8	HCPB8	硬件捕获 B 条件 8	条件：内部硬件触发事件 0 有效 ● 0b0：条件匹配时，硬件捕获 B 无效 ● 0b1：条件匹配时，硬件捕获 B 有效	RW
7:4	Reserved	保留	-	-
3	HCPB3	硬件捕获 B 条件 3	条件：TMR6_<t>_PWMB 端口上采样到下降沿 ● 0b0：条件匹配时，硬件捕获 B 无效 ● 0b1：条件匹配时，硬件捕获 B 有效	RW

位/位域	标记	位名	功能描述	读写
2	HCPB2	硬件捕获 B 条件 2	条件：TMR6_<t>_PWMB 端口上采样到上升沿 ● 0b0：条件匹配时，硬件捕获 B 无效 ● 0b1：条件匹配时，硬件捕获 B 有效	RW
1	HCPB1	硬件捕获 B 条件 1	条件：TMR6_<t>_PWMA 端口上采样到下降沿 ● 0b0：条件匹配时，硬件捕获 B 无效 ● 0b1：条件匹配时，硬件捕获 B 有效	RW
0	HCPB0	硬件捕获 B 条件 0	条件：TMR6_<t>_PWMA 端口上采样到上升沿 ● 0b0：条件匹配时，硬件捕获 B 无效 ● 0b1：条件匹配时，硬件捕获 B 有效	RW



说明

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。

20.5.23 硬件递加事件选择寄存器 (TMR6_HCUPR)

Offset	Bit Position																																			
0x0198	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reset	0x00000000																																			
Name	Reserved												HCUP19	HCUP18	HCUP17	HCUP16	Reserved						HCUP9	HCUP8	HCUP7	HCUP6	HCUP5	HCUP4	HCUP3	HCUP2	HCUP1	HCUP0				
Access													RW	RW	RW	RW							RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:20	Reserved	保留	-	-
19	HCUP19	硬件递加条件 19	条件：TMR6_TRIGB 端口上采样到下降沿 ● 0b0：条件匹配时，硬件递加无效 ● 0b1：条件匹配时，硬件递加有效	RW
18	HCUP18	硬件递加条件 18	条件：TMR6_TRIGB 端口上采样到上升沿 ● 0b0：条件匹配时，硬件递加无效 ● 0b1：条件匹配时，硬件递加有效	RW
17	HCUP17	硬件递加条件 17	条件：TMR6_TRIGA 端口上采样到下降沿 ● 0b0：条件匹配时，硬件递加无效 ● 0b1：条件匹配时，硬件递加有效	RW
16	HCUP16	硬件递加条件 16	条件：TMR6_TRIGA 端口上采样到上升沿 ● 0b0：条件匹配时，硬件递加无效 ● 0b1：条件匹配时，硬件递加有效	RW
15:10	Reserved	保留	-	-
9	HCUP9	硬件递加条件 9	条件：内部硬件触发事件 1 有效 ● 0b0：条件匹配时，硬件递加无效 ● 0b1：条件匹配时，硬件递加有效	RW
8	HCUP8	硬件递加条件 8	条件：内部硬件触发事件 0 有效 ● 0b0：条件匹配时，硬件递加无效 ● 0b1：条件匹配时，硬件递加有效	RW
7	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 ● 0b0：条件匹配时，硬件递加无效 ● 0b1：条件匹配时，硬件递加有效	RW
6	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 ● 0b0：条件匹配时，硬件递加无效 ● 0b1：条件匹配时，硬件递加有效	RW
5	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 ● 0b0：条件匹配时，硬件递加无效 ● 0b1：条件匹配时，硬件递加有效	RW

位/位域	标记	位名	功能描述	读写
4	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 ● 0b0: 条件匹配时, 硬件递加无效 ● 0b1: 条件匹配时, 硬件递加有效	RW
3	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 ● 0b0: 条件匹配时, 硬件递加无效 ● 0b1: 条件匹配时, 硬件递加有效	RW
2	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 ● 0b0: 条件匹配时, 硬件递加无效 ● 0b1: 条件匹配时, 硬件递加有效	RW
1	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 ● 0b0: 条件匹配时, 硬件递加无效 ● 0b1: 条件匹配时, 硬件递加有效	RW
0	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 ● 0b0: 条件匹配时, 硬件递加无效 ● 0b1: 条件匹配时, 硬件递加有效	RW

20.5.24 硬件递减事件选择寄存器 (TMR6_HCDOR)

Offset	Bit Position																																				
0x019C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Reset	0x00000000																																				
Name	Reserved												HCD019	HCD018	HCD017	HCD016	Reserved						HCD09	HCD08	HCD07	HCD06	HCD05	HCD04	HCD03	HCD02	HCD01	HCD00					
Access													RW	RW	RW	RW							RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:20	Reserved	保留	-	-
19	HCD019	硬件递减条件 19	条件：TMR6_TRIGB 端口上采样到下降沿 ● 0b0：条件匹配时，硬件递减无效 ● 0b1：条件匹配时，硬件递减有效	RW
18	HCD018	硬件递减条件 18	条件：TMR6_TRIGB 端口上采样到上升沿 ● 0b0：条件匹配时，硬件递减无效 ● 0b1：条件匹配时，硬件递减有效	RW
17	HCD017	硬件递减条件 17	条件：TMR6_TRIGA 端口上采样到下降沿 ● 0b0：条件匹配时，硬件递减无效 ● 0b1：条件匹配时，硬件递减有效	RW
16	HCD016	硬件递减条件 16	条件：TMR6_TRIGA 端口上采样到上升沿 ● 0b0：条件匹配时，硬件递减无效 ● 0b1：条件匹配时，硬件递减有效	RW
15:10	Reserved	保留	-	-
9	HCD09	硬件递减条件 9	条件：内部硬件触发事件 1 有效 ● 0b0：条件匹配时，硬件递减无效 ● 0b1：条件匹配时，硬件递减有效	RW
8	HCD08	硬件递减条件 8	条件：内部硬件触发事件 0 有效 ● 0b0：条件匹配时，硬件递减无效 ● 0b1：条件匹配时，硬件递减有效	RW
7	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 ● 0b0：条件匹配时，硬件递减无效 ● 0b1：条件匹配时，硬件递减有效	RW
6	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 ● 0b0：条件匹配时，硬件递减无效 ● 0b1：条件匹配时，硬件递减有效	RW
5	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 ● 0b0：条件匹配时，硬件递减无效 ● 0b1：条件匹配时，硬件递减有效	RW

位/位域	标记	位名	功能描述	读写
4	HCDO4	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 ● 0b0: 条件匹配时, 硬件递减无效 ● 0b1: 条件匹配时, 硬件递减有效	RW
3	HCDO3	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 ● 0b0: 条件匹配时, 硬件递减无效 ● 0b1: 条件匹配时, 硬件递减有效	RW
2	HCDO2	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 ● 0b0: 条件匹配时, 硬件递减无效 ● 0b1: 条件匹配时, 硬件递减有效	RW
1	HCDO1	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 ● 0b0: 条件匹配时, 硬件递减无效 ● 0b1: 条件匹配时, 硬件递减有效	RW
0	HCDO0	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 ● 0b0: 条件匹配时, 硬件递减无效 ● 0b1: 条件匹配时, 硬件递减有效	RW

20.5.25 公共滤波控制寄存器 (TMR6_FCNT)

Offset	Bit Position																															
0x00EC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																										NOFICKTB	NOFIENTB	Reserved	NOFICKTA	NOFIENTA	
Access																											RW	RW		RW	RW	

位/位域	标记	位名	功能描述	读写
31:7	Reserved	保留	-	-
6:5	NOFICKTB	滤波采样基准时钟选择 TB	0b00: PCLK0 0b01: PCLK0/4 0b10: PCLK0/16 0b11: PCLK0/64	RW
4	NOFIENTB	捕获输入端口滤波 TB	0b0: TMR6_TRIGB 端口输入滤波功能无效 0b1: TMR6_TRIGB 端口输入滤波功能使能	RW
3	Reserved	保留	-	-
2:1	NOFICKTA	滤波采样基准时钟选择 TA	0b00: PCLK0 0b01: PCLK0/4 0b10: PCLK0/16 0b11: PCLK0/64	RW
0	NOFIENTA	捕获输入端口滤波 TA	0b0: TMR6_TRIGA 端口输入滤波功能无效 0b1: TMR6_TRIGA 端口输入滤波功能使能	RW

20.5.26 公共软件同步启动控制寄存器 (TMR6_SSTAR)

Offset	Bit Position																															
0x00F0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																														SSTA2	SSTA1
Access																															RW	RW

位/位域	标记	位名	功能描述	读写
31:2	Reserved	保留	-	-
1	SSTA2	单元 2 软件启动	写 1 启动定时器单元 2，写 0 无效 读取： ● 0b0：定时器单元 2 未启动 ● 0b1：定时器单元 2 已启动	RW
0	SSTA1	单元 1 软件启动	写 1 启动定时器单元 1，写 0 无效 读取： ● 0b0：定时器单元 1 未启动 ● 0b1：定时器单元 1 已启动	RW

20.5.27 公共软件同步停止控制寄存器 (TMR6_SSTPR)

Offset	Bit Position																															
0x00F4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																												SSTP2		SSTP1	
Access																													WO		WO	

位/位域	标记	位名	功能描述	读写
31:2	Reserved	保留	-	-
1	SSTP2	单元 2 软件停止	写 1 停止定时器单元 2，写 0 无效	WO
0	SSTP1	单元 1 软件停止	写 1 停止定时器单元 1，写 0 无效	WO

20.5.28 公共软件同步清零控制寄存器 (TMR6_SCLRR)

Offset	Bit Position																																	
0x00F8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved																														SCLE2		SCLE1	
Access																															WO		WO	

位/位域	标记	位名	功能描述	读写
31:2	Reserved	保留	-	-
1	SCLE2	单元 2 软件清零	写 1 清零定时器单元 2，写 0 无效	WO
0	SCLE1	单元 1 软件清零	写 1 清零定时器单元 1，写 0 无效	WO

20.5.29 公共软件同步刷新控制寄存器（TMR6_SUPDR）

Offset	Bit Position																															
0x00FC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																												SUPD2		SUPD1	
Access																													WO		WO	

位/位域	标记	位名	功能描述	读写
31:2	Reserved	保留	-	-
1	SUPD2	单元 2 软件刷新	写 1 刷新定时器单元 2，写 0 无效	WO
0	SUPD1	单元 1 软件刷新	写 1 刷新定时器单元 1，写 0 无效	WO

20.6 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所示。

表 20-62 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所示。

表 20-63 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	↓

PWMA 端口输出控制事件		端口状态	优先级
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	↓
计数停止时		PCNAR.STPCA 设定状态	低

表 20-64 PWMB 端口输出控制优先级

PWMB 端口输出控制事件		端口状态	优先级
EMB 选择的事件发生		PCNBR.EMBCB 设定状态	高
设定强制端口输出		PCNBR.FORCB 设定状态	↓
计数开始时		PCNBR.STACB 设定状态	↓
锯齿波向上计数时	CNTER=PERAR	PCNBR.OVFCE 设定状态	↓
	CNTER=GCMBR	PCNBR.CMBUCB 设定状态	↓
	CNTER=GCMAR	PCNBR.CMAUCB 设定状态	↓
锯齿波向下计数时	CNTER=0	PCNBR.UDFCB 设定状态	↓
	CNTER=GCMBR	PCNBR.CMBDCB 设定状态	↓
	CNTER=GCMAR	PCNBR.CMADCB 设定状态	↓
三角波向上计数时	CNTER=GCMBR	PCNBR.CMBUCB 设定状态	↓
	CNTER=GCMAR	PCNBR.CMAUCB 设定状态	↓
	CNTER=PERAR	PCNBR.OVFCE 设定状态	↓
三角波向下计数时	CNTER=GCMBR	PCNBR.CMBDCB 设定状态	↓
	CNTER=GCMAR	PCNBR.CMADCB 设定状态	↓
	CNTER=0	PCNBR.UDFCB 设定状态	↓
计数停止时		PCNBR.STPCB 设定状态	低

21 通用控制定时器（Timer4）

21.1 简介

通用控制定时器 4（Timer4）是一个用于三相电机控制的定时器模块，提供各种不同应用的三相电机控制方案。该定时器支持三角波和锯齿波两种波形模式，可生成各种 PWM 波形；支持缓存功能；支持 EMB 控制。本系列产品中搭载 3 个单元的 Timer4。

21.2 主要特性

Timer4 基本的功能及特性如下表所示。

表 21-1 Timer4 的基本功能及特性

波形模式	锯齿波、三角波
基本功能	递加、递减计数方向
	缓存功能
	通用 PWM 输出
	通用比较匹配事件
	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
	软件同步启动计数
	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

21.3 功能说明

21.3.1 Timer4 框图

下图描述了通用控制定时器 Timer4 的基本架构。框图中所示“<t>”表示单元编号，即“<t>”为 1~3，本章节后文提到“<t>”时均指单元编号，不再赘述。

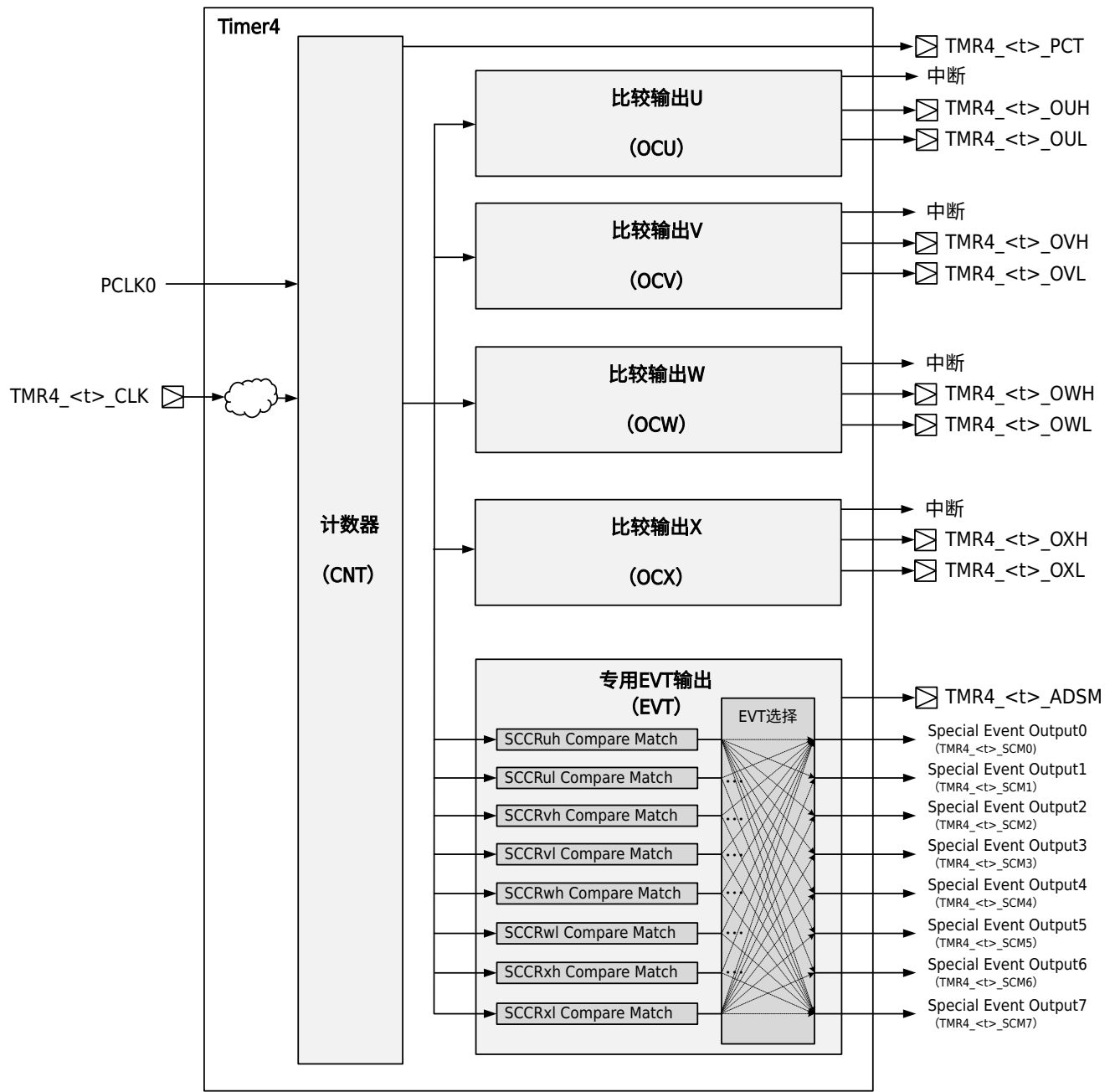


图 21-1 Timer4 基本框图

下表为 Timer4 的输入输出端口列表。

表 21-2 Timer4 端口列表

端口名	方向	功能
TMR4_<t>_CLK	输入	计数时钟输入端口
TMR4_<t>_PCT	输出	PWM 周期输出监测

端口名	方向	功能
TMR4_<t>_ADSM	输出	专用事件输出监测
TMR4_<t>_OUH	输出	PWM 输出端口
TMR4_<t>_OUL		
TMR4_<t>_OVH		
TMR4_<t>_OVL		
TMR4_<t>_OWH		
TMR4_<t>_OWL		
TMR4_<t>_OXH		
TMR4_<t>_OXL		

21.3.2 基本动作

21.3.2.1 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。

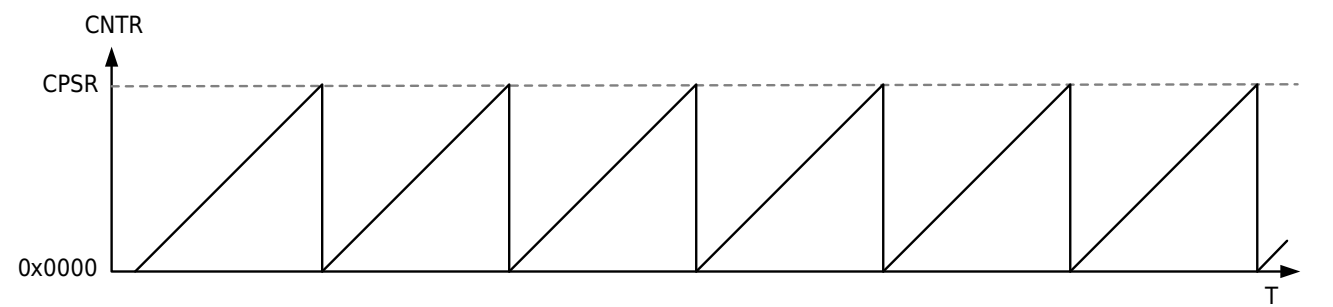


图 21-2 Timer4 锯齿波波形

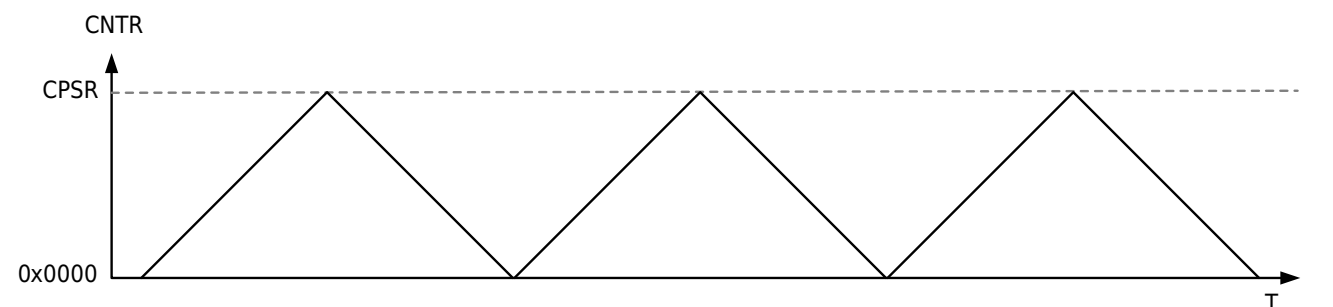


图 21-3 Timer4 三角波波形

21.3.2.2 计数动作

- 锯齿波计数操作与控制流程如下图所示：

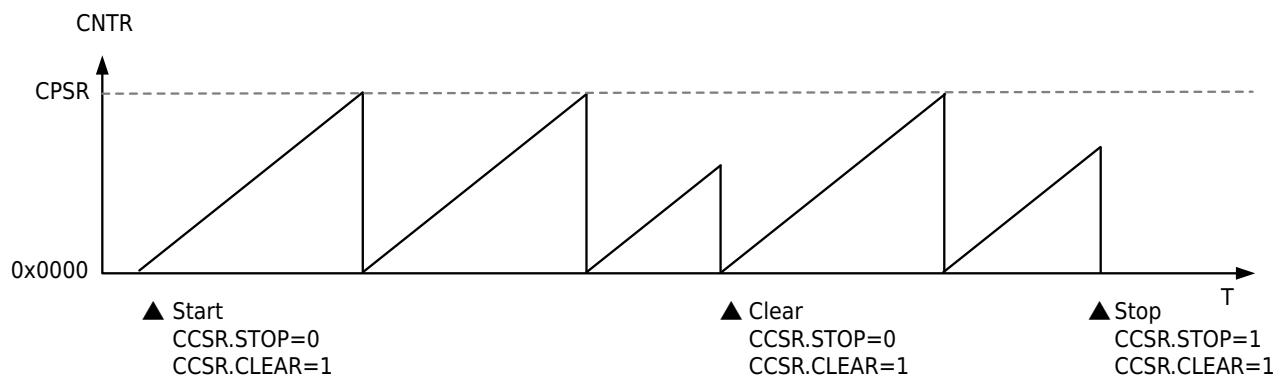


图 21-4 Timer4 锯齿波模式计数动作

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) × 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。
- 三角波计数操作与控制流程如下图所示:

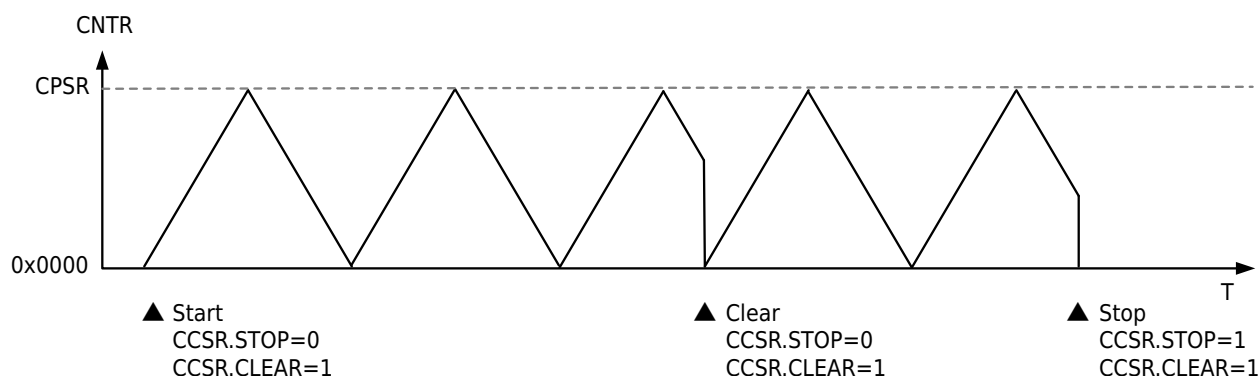


图 21-5 Timer4 三角波模式计数动作

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。

21.3.2.3 比较输出

● 下图是锯齿波计数模式下的比较输出模块的波形输出示例。

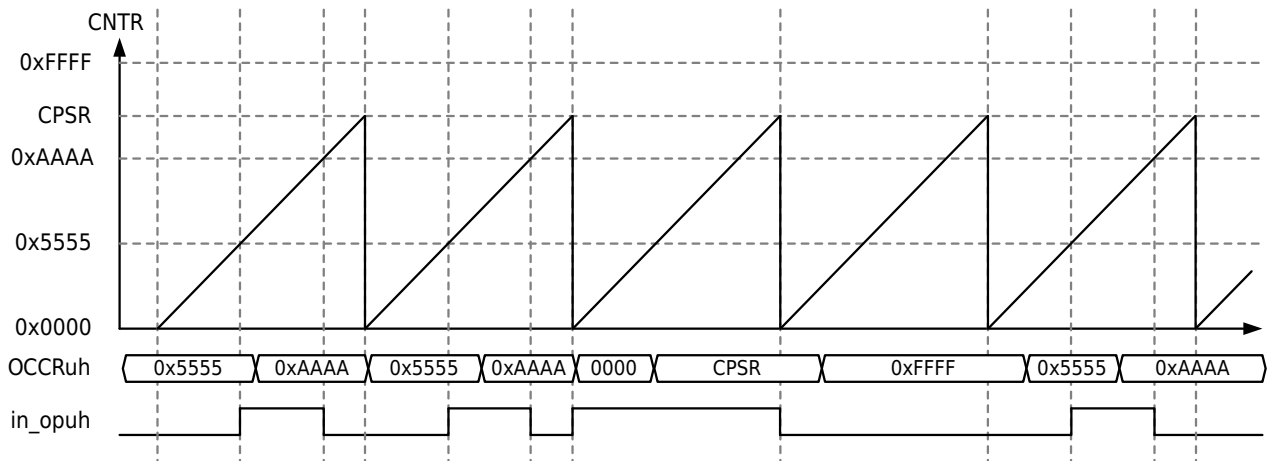


图 21-6 锯齿波模式波形输出例

● 下图是三角波计数模式下的比较输出模块的波形输出示例。

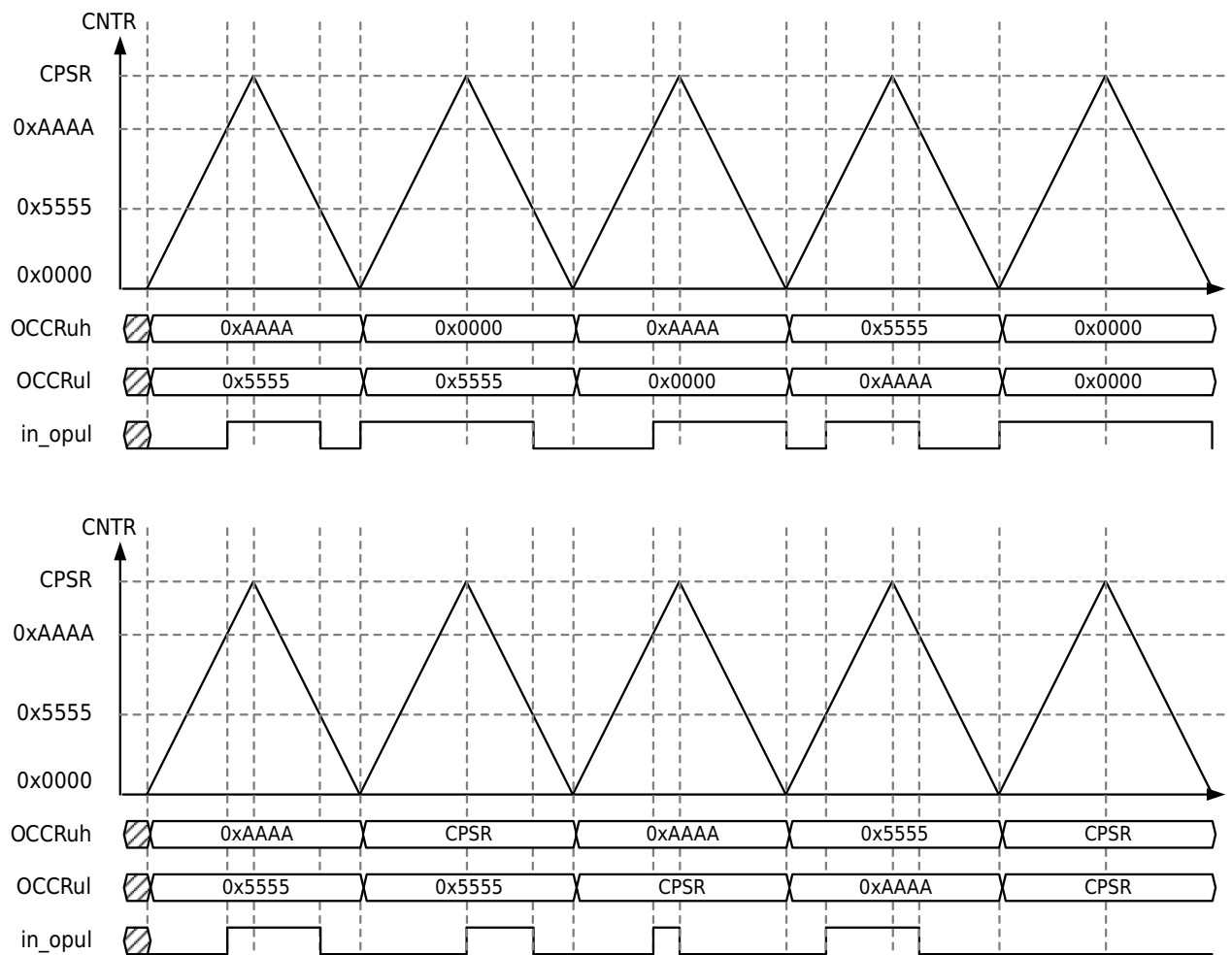


图 21-7 三角波模式波形输出例

21.3.3 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。

21.3.3.1 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收纳到缓冲寄存器。在以下条件下，数据从缓冲寄存器传输到 CPSR 寄存器中。

- 1. 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。
- 2. 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。

- 当锯齿波模式且禁止缓冲功能时修改计数峰值 CPSR 的操作：

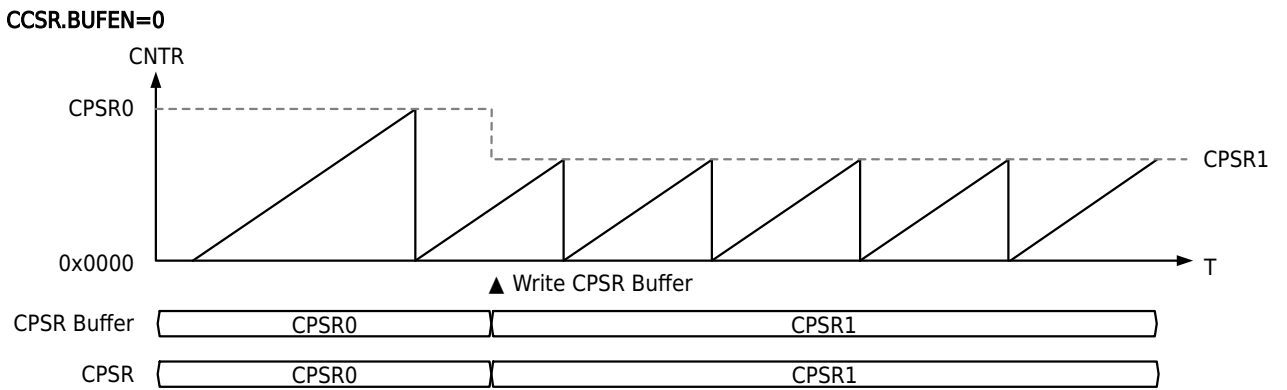


图 21-8 缓存无效时修改锯齿波计数周期



注意

当禁止缓冲功能时，写入数据立即从缓冲寄存器传输到 CPSR 计数器，计数周期在写入操作完成后立即改变。在此情况下，如果写入的值小于计数器当前计数值，计数器将持续进行递增计数，直到到达 "0xFFFF"，因此，需特别注意此种状况。

- 当锯齿波模式且使能缓冲功能时修改计数峰值 CPSR 的操作：

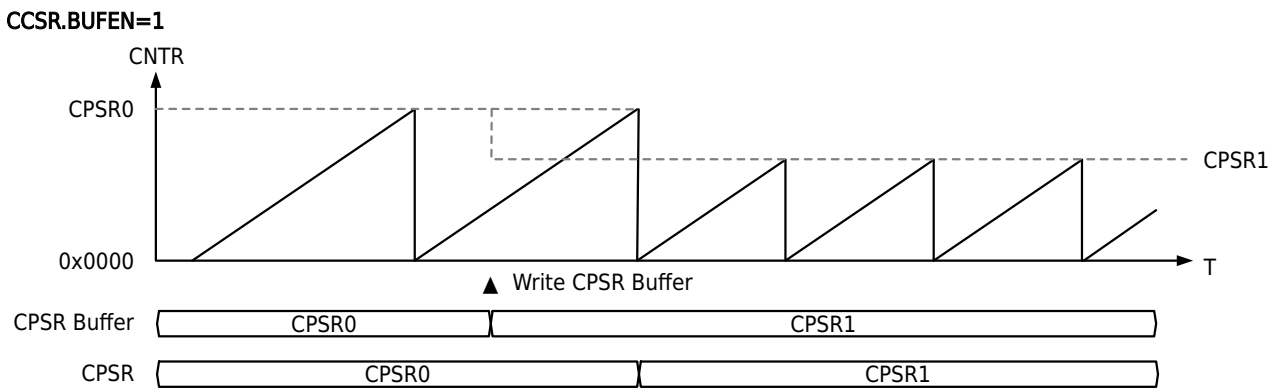


图 21-9 缓存使能时修改锯齿波计数周期

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。

- 当三角波模式且使能缓冲功能时修改计数峰值 CPSR 的操作：

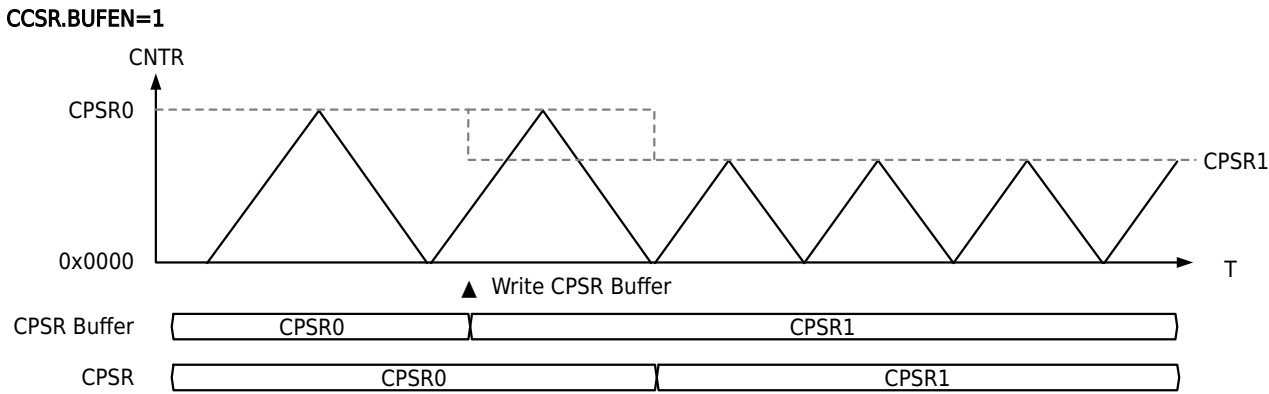


图 21-10 缓存使能时修改三角波计数周期

三角波计数模式下，当使能缓冲功能时，在计数器停止时或者计数器的下一次计数值为"0x0000"时，写入的数据从缓冲寄存器传输到 CPSR 寄存器。计数器周期的改变在写入操作完成后的下一个计数器周期开始。

21.3.3.2 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。

- 1. 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。

下图所示为通用输出比较 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)。

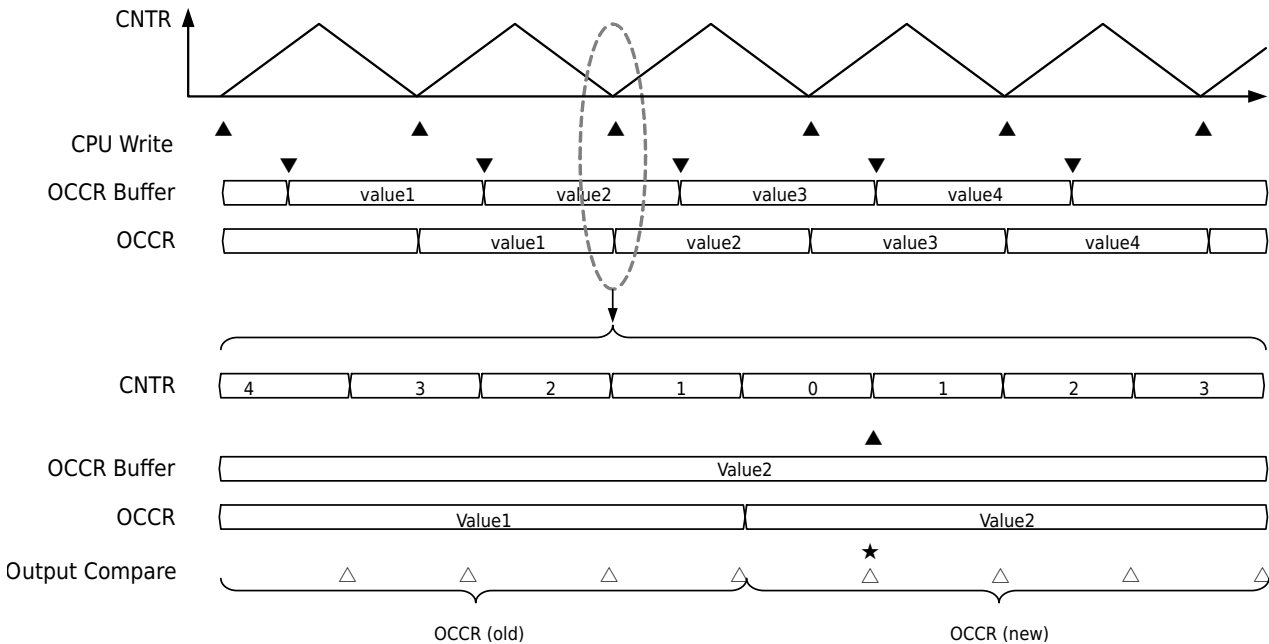


图 21-11 OCCR 缓冲数据传输 (周期间隔响应链接禁止时)

图中的上半部分是全局示意图，下半部分是传输操作时放大图。

计数器为三角波计数模式，下溢中断在标志▲（计数谷点）时刻产生。在▼时刻，CPU 改写 OCCR 寄存器，写入的数据存储在 OCCR 缓冲寄存器中。之后当计数器计数值为 0x0000 时，进行数据加载操作，数据从缓冲寄存器到 OCCR 寄存器，同时产生中断标志 IRQZF。

在△时刻，输出比较根据设定的 OCCR 寄存器值与计数值匹配事件，执行改变 PWM 输出和置位 OCSR.OCFx 位（x=L 或 H）。在时刻★（计数值=0x0000）之后，端口输出控制根据新的 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）。

2. 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。

下图所示为通用输出比较 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）。

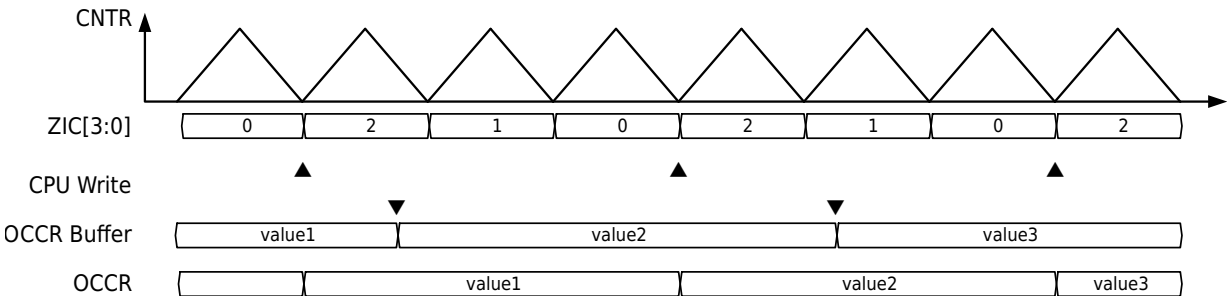


图 21-12 OCCR 缓冲数据传输（周期间隔响应链接使能）

计数器为三角波计数模式，周期间隔计数器（CVPR.ZIC）从 2 到 0 做递减计数，下溢中断在标志▲（计数谷点）时刻产生。在▼时刻，CPU 改写 OCCR 寄存器，写入的数据存储在 OCCR 缓冲寄存器中。之后当计数器计数值为 0x0000 且周期间隔计数器（CVPR.ZIC）为 0 时，进行数据加载操作，数据从缓冲寄存器到 OCCR 寄存器，同时产生中断标志 IRQZF。

图中示意了 OCCR 缓冲寄存器在计数谷点的传输操作，OCMR 的缓冲寄存器的缓存传送操作与之类似；同理，在计数峰点的传输操作也类似。新的数据在传输时刻之后即刻生效（新写入数据将控制 PWM 输出和中断标志置位）。

使用通道链接操作模式时，同时使能 OCCRuH 和 OCCRuL 缓冲功能可以产生各种不同的 PWM 输出波形。如下图所示，在输出比较寄存器 OCCRuH 和 OCCRuL 保持不变的情况下，改变 OCMR 寄存器值，使 in_opul 输出不同的波形信号。

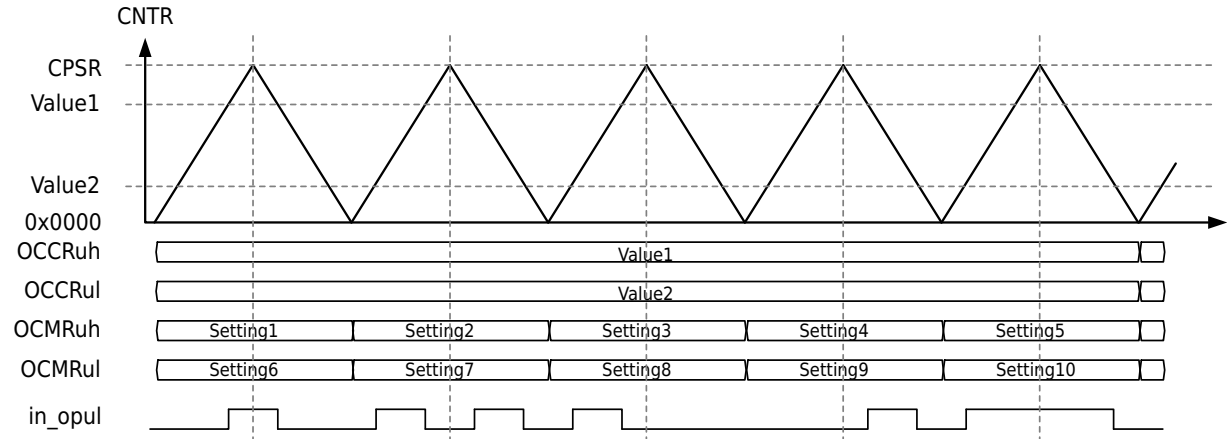


图 21-13 输出比较缓冲数据传输（OCMR 缓冲使能）

21.3.3.3 专用比较寄存器缓存功能

专用比较基准寄存器 (SCCR) 和专用模式控制寄存器 (SCMR) 都带有缓冲寄存器功能。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。

1. 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。

下图所示为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。

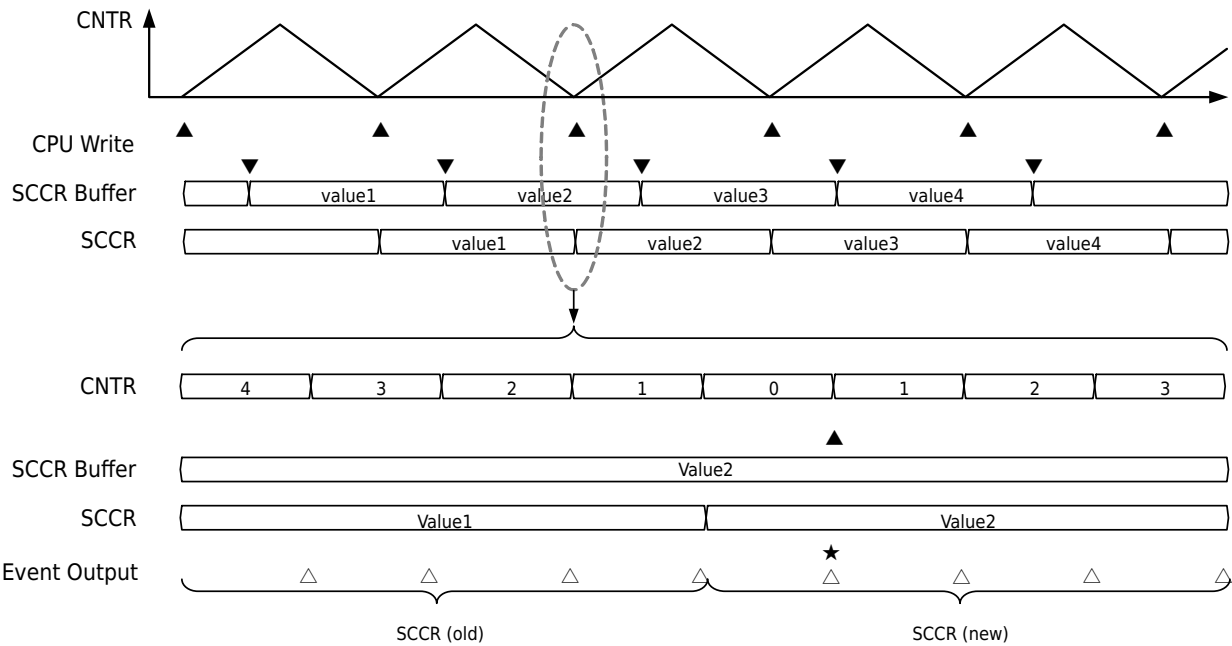


图 21-14 SCCR 缓冲传输操作（周期间隔响应链接传输禁止时）

图中的上半部分是全局示意图，下半部分缓冲寄存器传输操作的局部放大图。

计数器为三角波计数模式，下溢中断在标志▲（计数谷点）时刻产生。在▼时刻，CPU 改写 SCCR 寄存器，写入的数据存储在 SCCR 缓冲寄存器中。之后当计数器计数值为 0x0000 时，进行数据加载操作，数据从缓冲寄存器到 SCCR 寄存器，同时产生中断标志 IRQZF。

在△时刻，根据设定的 SCCR 寄存器值与计数值进行比较操作。在时刻★（计数值=0x0000）之后，专用事件输出信号根据新的 SCCR 数据执行操作。在时刻★之前根据 SCCR 数据执行操作。

图中示意了 SCCR 缓冲寄存器在计数谷点的传输操作，SCMR 的缓冲寄存器的缓存传送操作与之类似；同理，在计数峰点的传输操作也类似。新的数据在传输时刻之后即刻生效（新写入数据将专用事件输出信号和中断标志置位）。

2. 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。

下图所示为使能 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。

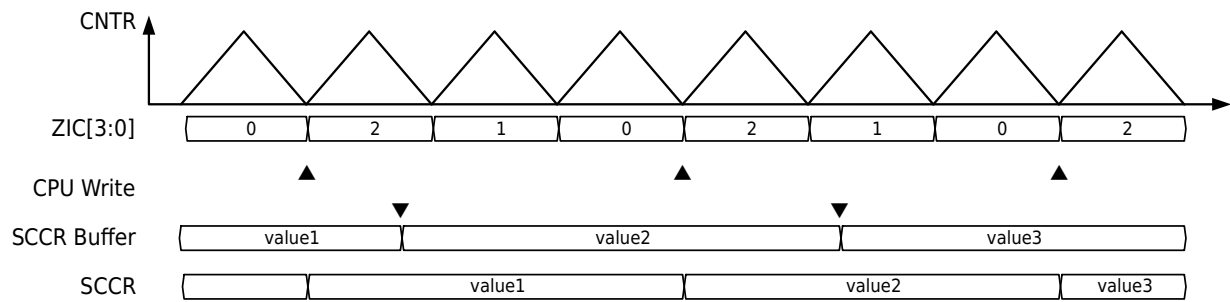


图 21-15 SCCR 缓冲传输操作（周期间隔响应链接传输使能时）

计数器为三角波计数模式，周期间隔计数器（CVPR.ZIC）从 2 到 0 做递减计数，下溢中断在标志▲（计数谷点）时刻产生。在▼时刻，CPU 改写 SCCR 寄存器，写入的数据存储在 SCCR 缓冲寄存器中。之后当计数器计数值为 0x0000 且周期间隔计数器（CVPR.ZIC）为 0 时，进行数据加载操作，数据从缓冲寄存器到 SCCR 寄存器，同时产生中断标志 IRQZF。

21.3.4 通用 PWM 输出

21.3.4.1 概述

本节后文提到的 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_<t>_O*H、TMR4_<t>_O*L 时，通配符“*”代表 U、V、W、X。

21.3.4.2 独立 PWM 输出

在直通模式（POCR.PWMMD=0b00）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）”。以下两张图分别是单元 1 的锯齿波和三角波下的独立 PWM 输出例。

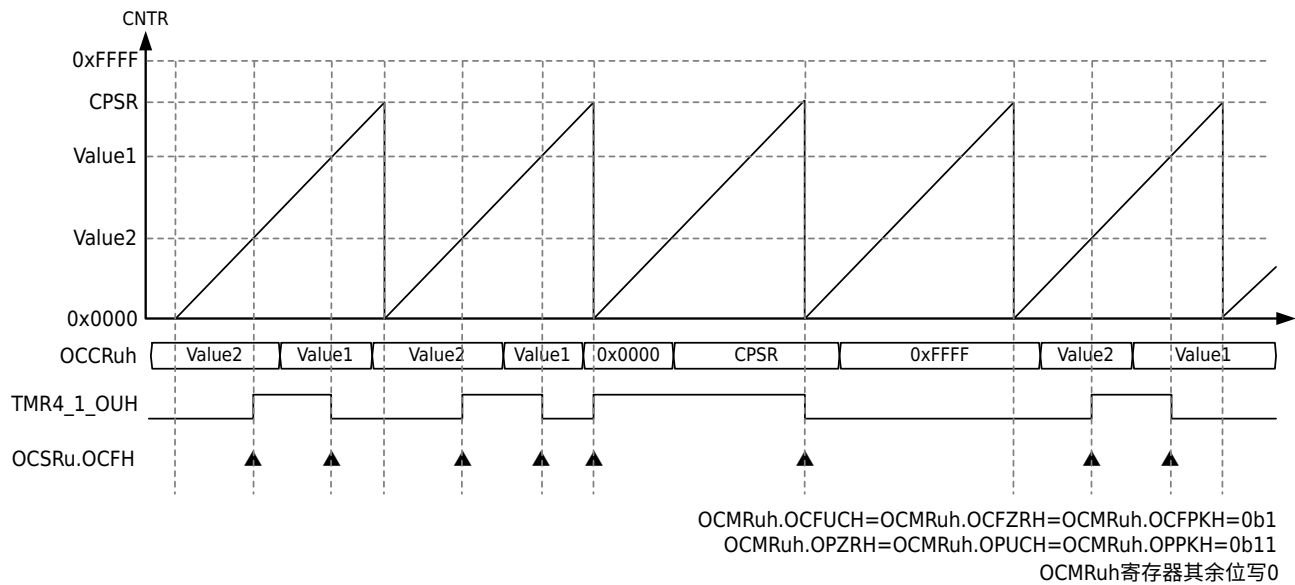


图 21-16 锯齿波独立 PWM 输出例

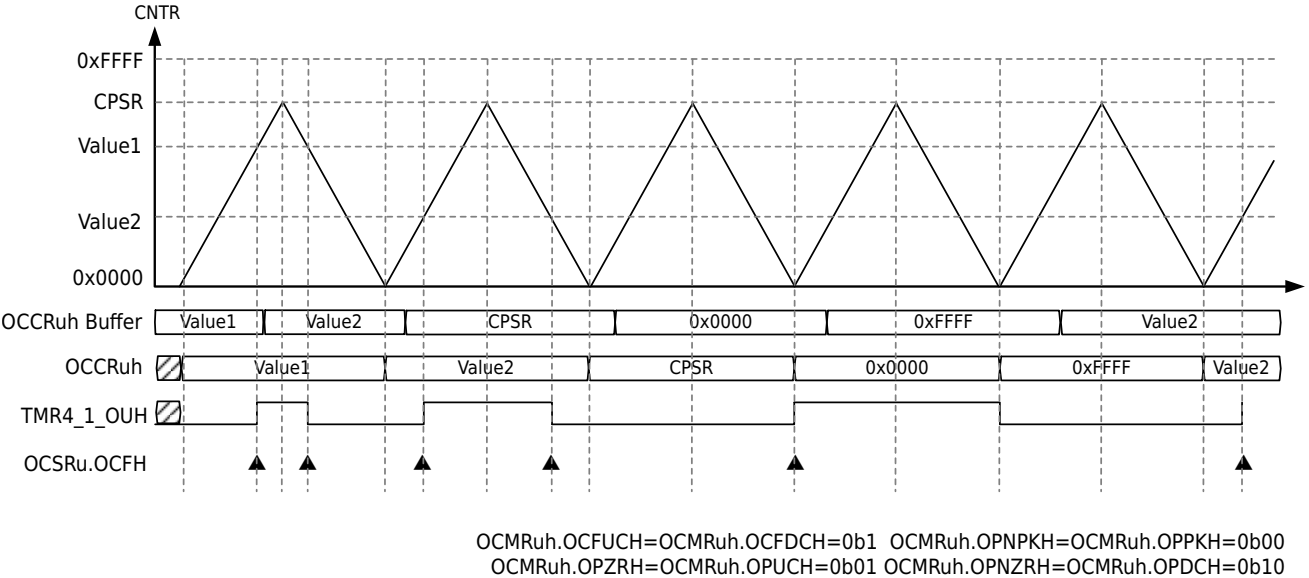


图 21-17 三角波独立 PWM 输出例



说明

直通模式是指通用比较基准寄存器（OCCR*h、OCCR*i）的值发生比较匹配所产生的内部输出信号（in_op*h、in_op*i）直接输出至对应的端口（TMR4_<t>_O*H、TMR4_<t>_O*L）上。

21.3.4.3 扩展 PWM 输出

在直通模式（POCR.PWMMD=0b00）下，TMR4_<t>_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_<t>_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）”。下图所示是该方式下 TMR4_<t>_OUH 和 TMR4_<t>_OUL 端口的 PWM 输出。

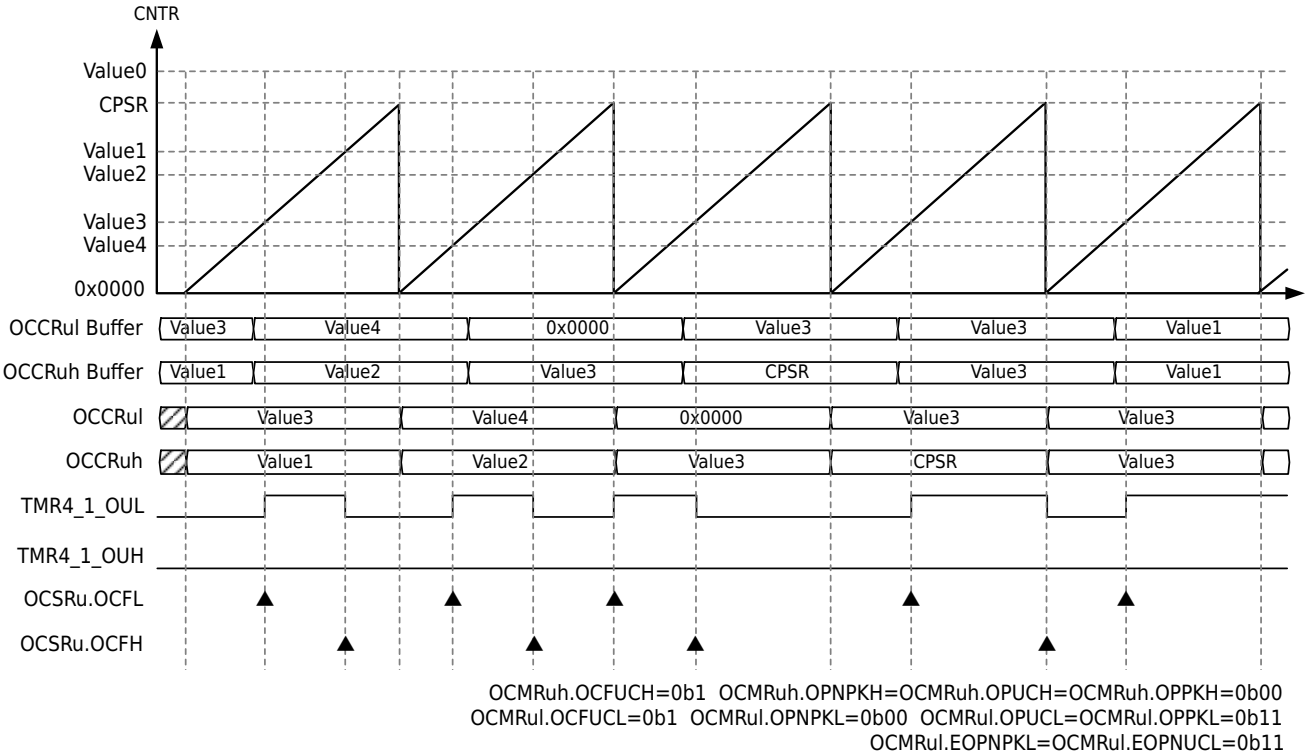


图 21-18 锯齿波扩展 PWM 输出



说明

独立 PWM 输出方式下，TMR4_<t>_O*L 的端口状态由 OCMRxI 寄存器的 Bit15~Bit4 位决定，仅与 OCCR*I 基准值有关。

21.3.4.4 互补 PWM 输出

软件设定实现互补 PWM 输出

在直通模式 (POCR.PWMMD=0b00) 下，直接设定 OCCR*h 和 OCCR*I 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。

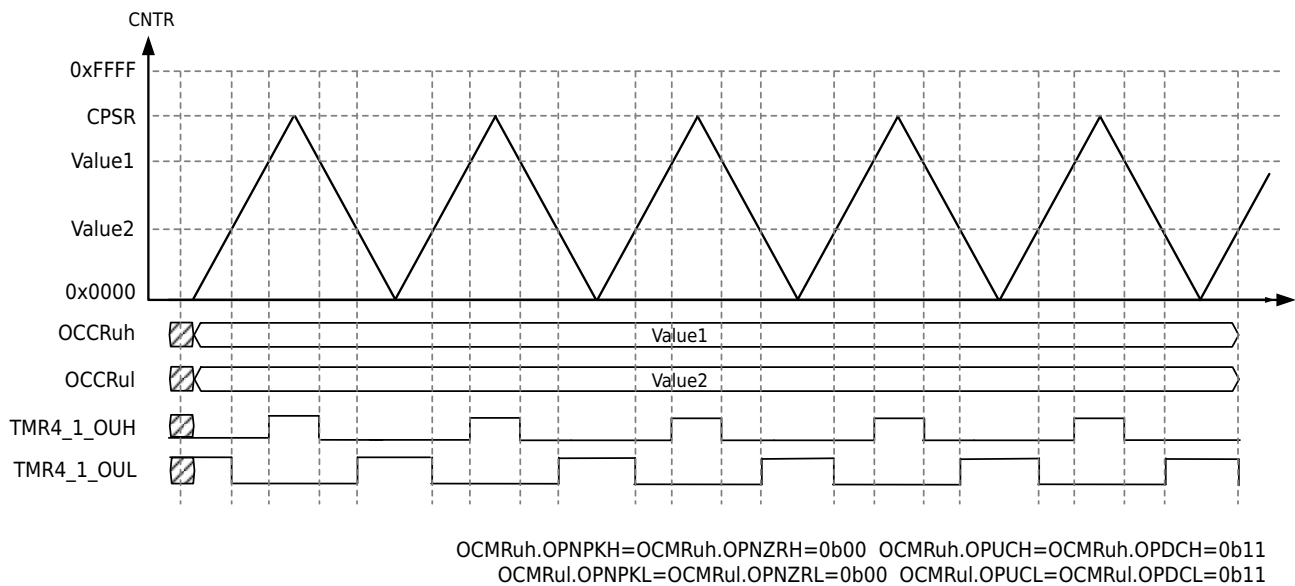


图 21-19 软件实现互补 PWM 输出

硬件设定实现互补 PWM 输出

在死区定时器模式 (POCR.PWMMD=0b01) 下，通用比较基准寄存器 (OCCR*I) 的值发生比较匹配产生的内部输出信号 (in_op*I) 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_<t>_O*H 端口输出的极性与 in_op*I 相同，TMR4_<t>_O*L 端口输出的极性与 in_op*I 相反。下图是死区定时器模式下的互补 PWM 输出例。

如果检测到 in_op*I 上升沿，TMR4_<t>_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_<t>_O*H 输出高电平；如果检测到 in_op*I 下降沿，TMR4_<t>_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_<t>_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。

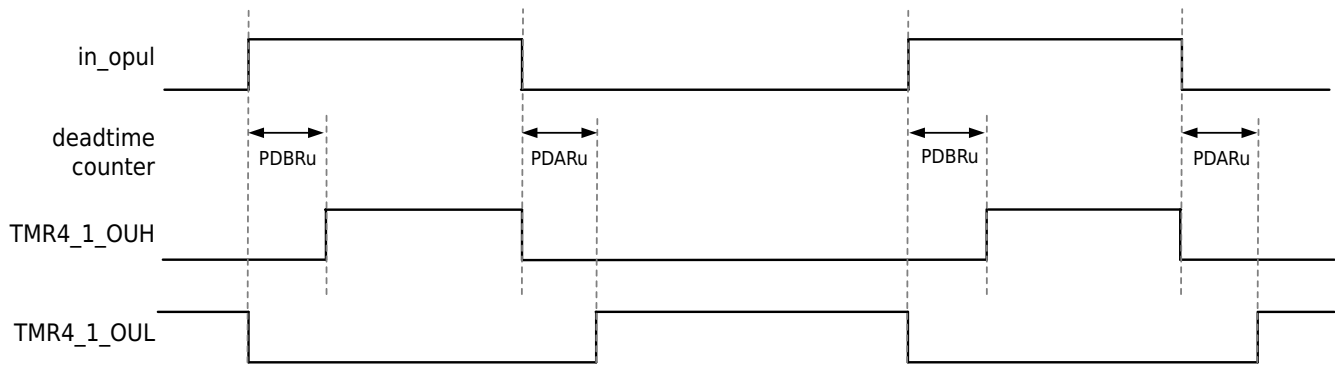


图 21-20 死区定时器模式下的互补 PWM 输出

in_opul 的高电平脉冲宽度小于 PDBR 设定的死区时间时，仅 TMR4_<t>_O*L 输出变为低电平。
TMR4_<t>_O*L 输出电平变为高的条件是当 in_opul 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_<t>_O*H 输出将持续保持为低电平。
in_opul 的低电平脉冲宽度小于 PDAR 设定的死区时间时，仅 TMR4_<t>_O*H 输出变为低电平。
TMR4_<t>_O*H 输出电平变为高的条件是当 in_opul 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_<t>_O*L 输出将持续保持为低电平。如下图所示。

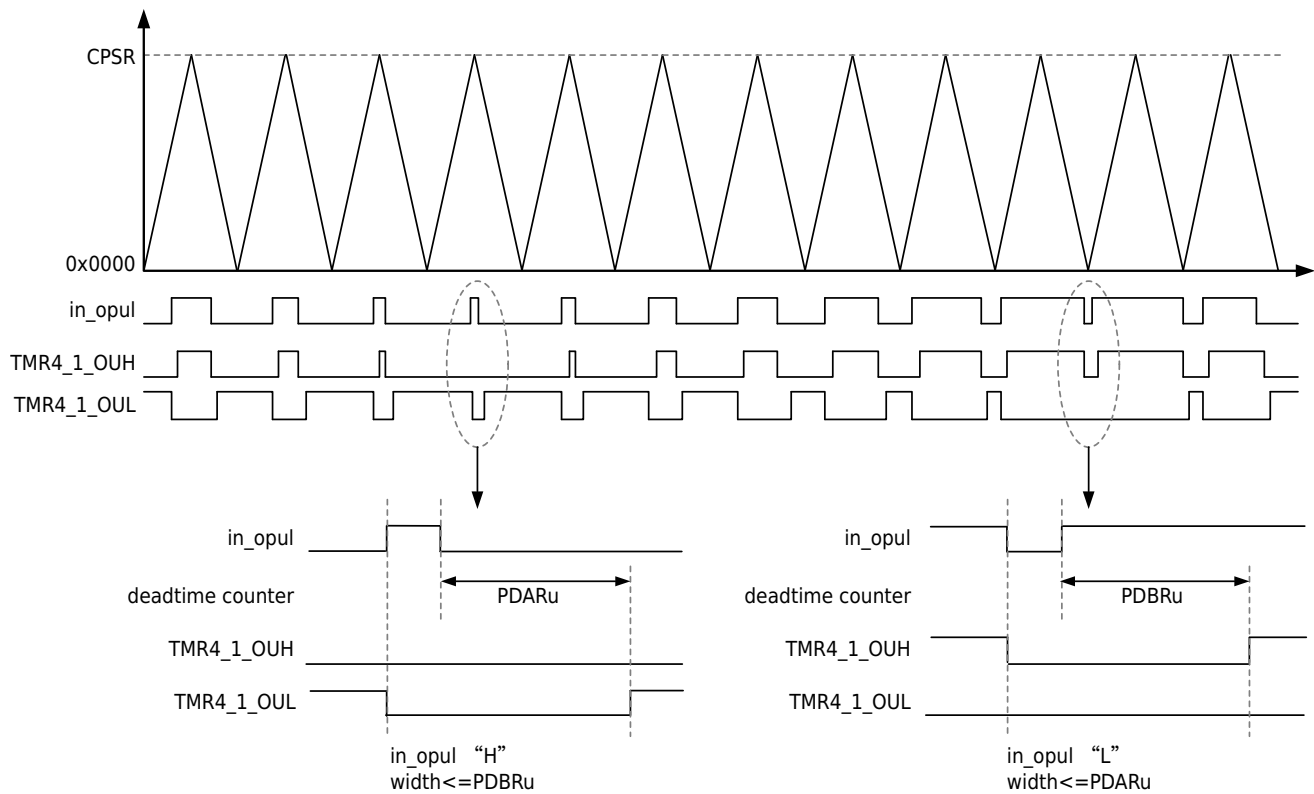


图 21-21 脉宽异常时的死区定时器模式下的波形输出

在上述硬件实现死区输出方式的基础上，还可以对内部的比较匹配信号（in_opul）进行脉冲宽度的监测，从而实现对 in_opul 信号的滤波控制。这种 in_opul 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=0b10）。

在死区计数器滤波模式下，滤波宽度由 PWM 滤波控制寄存器（PFSR）的设定值决定。当 in_opul 的脉冲宽度大于寄存器 PFSR 设定的时间时，滤波计数器将 in_opul 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_opul 上升沿，滤波计数器加载 PFSR 寄存器值并启动计量 in_opul 的高电平宽度，当 in_opul 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_<t>_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_<t>_O*H 输出高电平；如果检测到信号 in_op**l* 下降沿，滤波计数器加载 PFSR 寄存器值并启动计量 in_op**l* 的低电平宽度，当 in_op**l* 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_<t>_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_<t>_O*L 输出高电平。当 in_op**l* 的电平脉冲宽度小于寄存器 PFSR 设定的时间时，输出 TMR4_<t>_O*H 和 TMR4_<t>_O*L 将保持不变。

下图为死区计数器滤波模式下的互补 PWM 输出例。

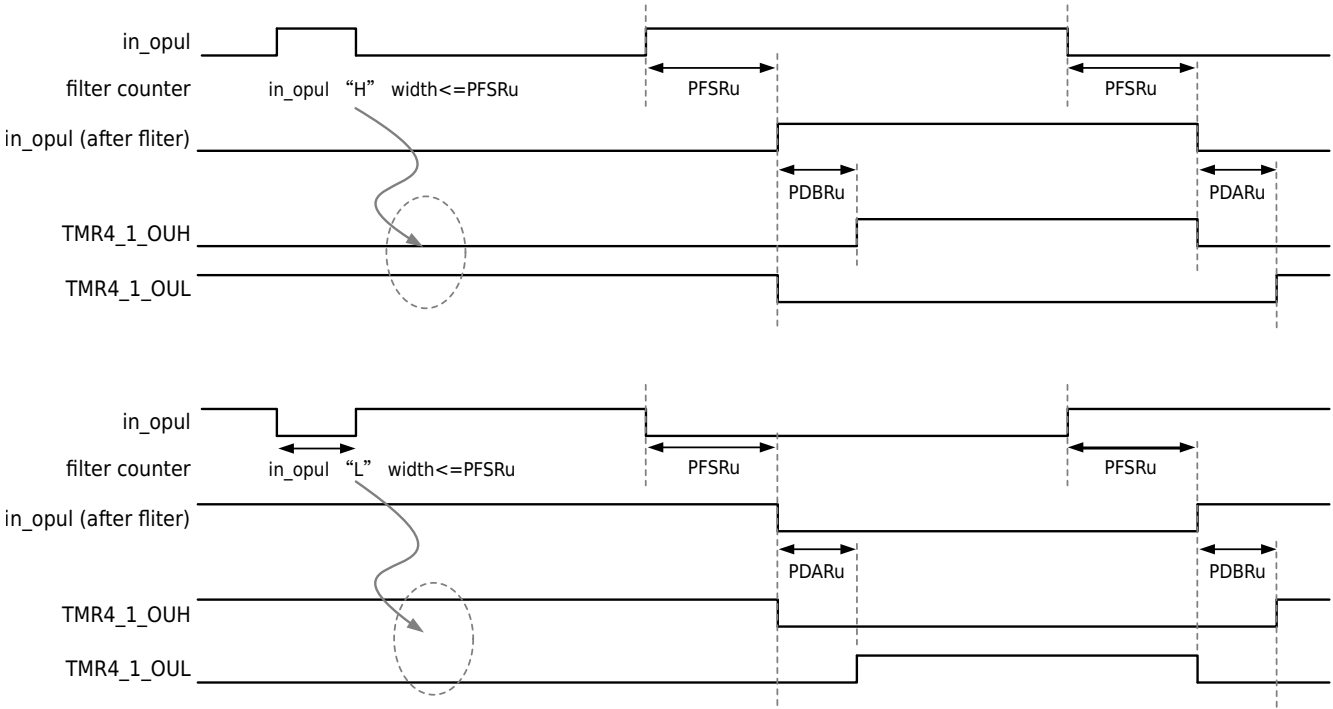


图 21-22 死区定时器滤波模式下的互补 PWM 输出

说明

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 \(TMR4_PSCR\)](#)”。

21.3.5 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) 作为递减计数器操作，在开始时载入 CVPR.ZIM 所设定的值，当 CVPR.ZIC=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) 作为递减计数器操作，在开始时载入 CVPR.PIM 所设定的值，当 CVPR.PIC=0 时，上溢标志位 (CCSR.IRQPF) 被置位为 1。

下图为周期间隔响应时的 IRQZF、IRQPF 的置位时序图。

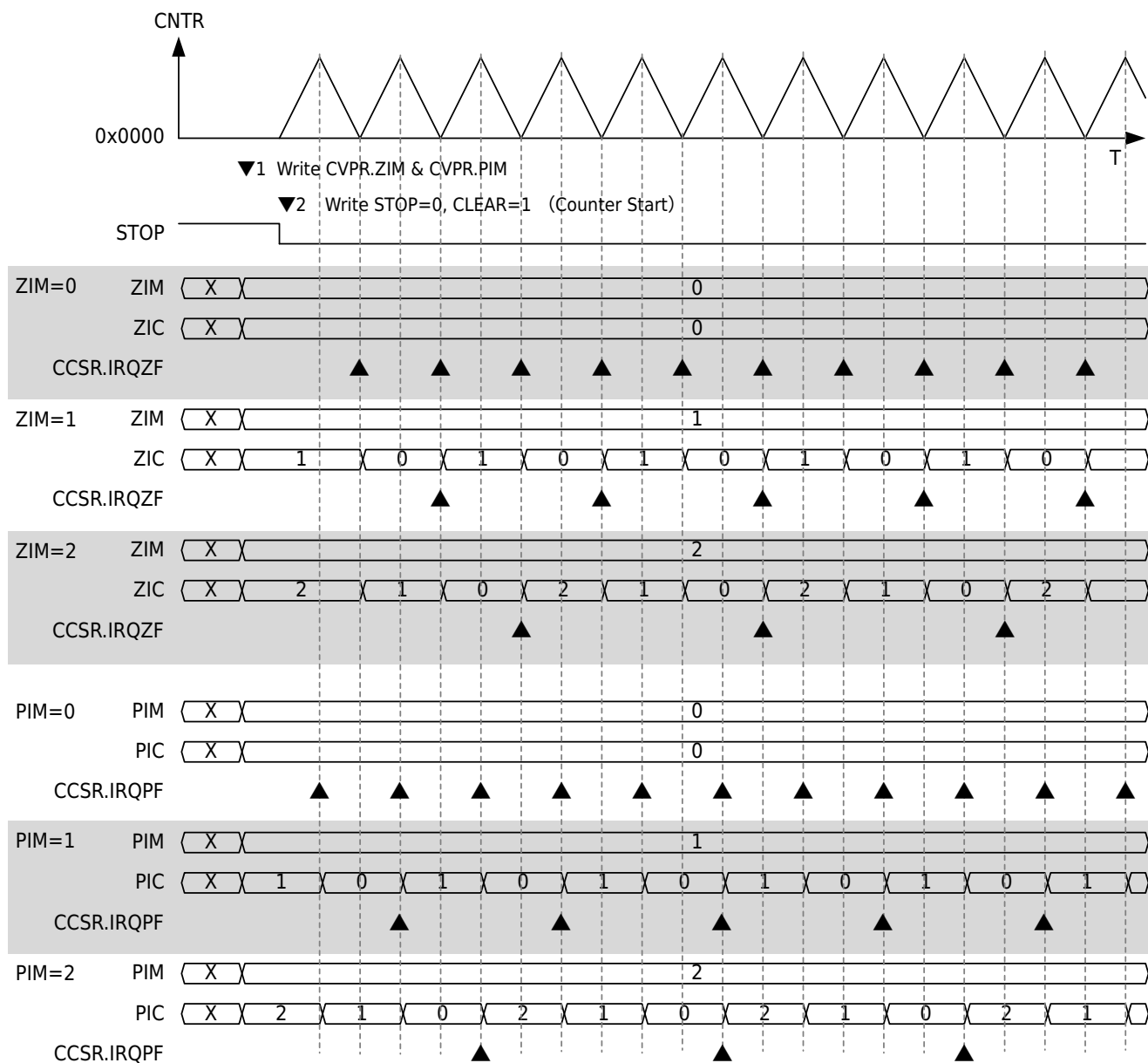


图 21-23 周期间隔响应时序图



说明

- 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。
- 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。下图为专用事件输出信号的周期间隔响应输出图。

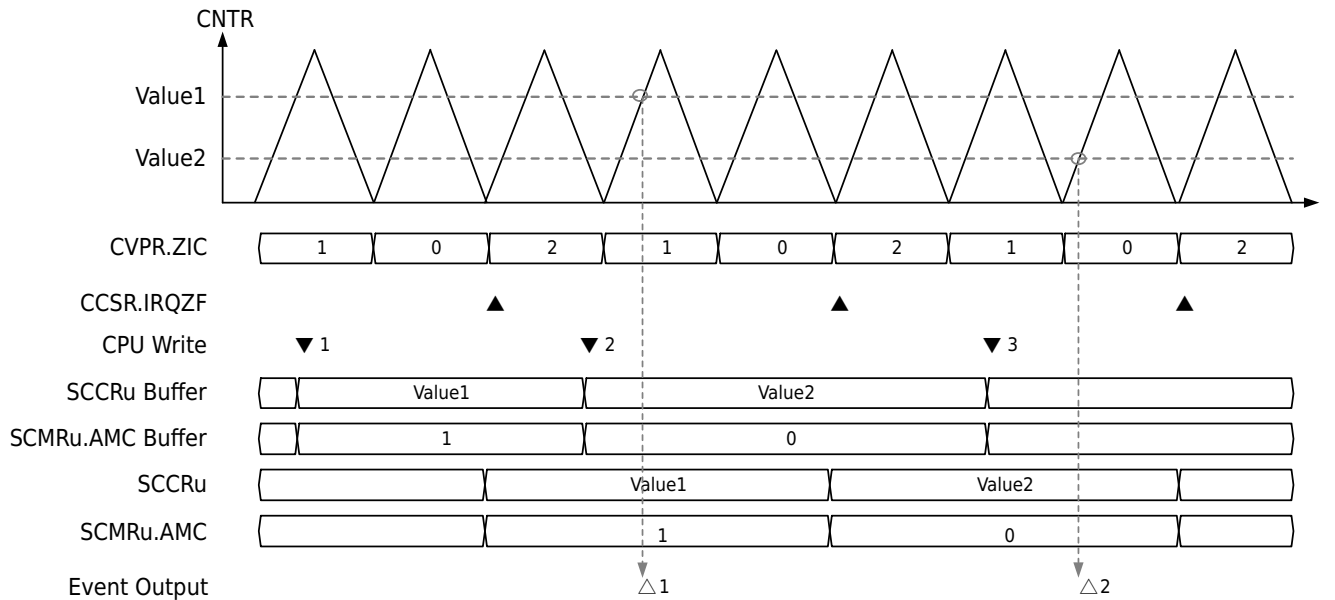


图 21-24 专用事件输出信号周期间隔响应输出

计数器是三角波计数模式，下溢中断屏蔽计数器（CVPR.ZIC）从 2-0 做递减计数。下溢中断在时刻▲产生。在时刻▼1，向 SCCRu 缓冲寄存器写入 value1。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=Value1，在时刻△1，专用事件输出信号置位。

在时刻▼2，向 SCCRu 缓冲寄存器写入 value2。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=Value2，在时刻△2，专用事件输出信号置位。

21.3.6 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_<t>_OUH 端口正常输出期间产生 EMB 事件，则 TMR4_<t>_OUH 端口上输出变为高阻态。

触发 EMB 异常事件的条件无效后，用户可通过写 EMB 状态复位寄存器（EMB_STATCLR0）来清除 EMB 异常事件，若此时 PSCR.AOE=1 则 PSCR.MOE 位将在下一个计数下溢点由硬件自动置 1（此处的下溢点为计数器的实际计数零点，不受寄存器 CVPR 的设定值影响），随后 PWM 端口恢复到正常输出；若此时 PSCR.AOE=0 则需要软件置位 PSCR.MOE，以恢复 PWM 端口到正常输出。

需要注意，硬件上控制 PWM 输出使使用的是 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，请参见《紧急刹车模块（EMB）》章节相关描述。

21.3.7 监测输出

每个 Timer4 单元都有一个 PWM 周期输出监测端口 TMR4_<t>_PCT 和专用事件输出监测端口 TMR4_<t>_ADSM。这些监测端口输出到外部，用户可根据端口的变化信息灵活对应用系统进行控制。

PWM 周期输出监测端口 TMR4_<t>_PCT 用来监测当前计数器的计数状态。在三角波模式时，每次计数到波峰或波谷时，监测端口 TMR4_<t>_PCT 上就发生一次翻转。下图为计数方向信号的示例。

TMR4_<t>_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。

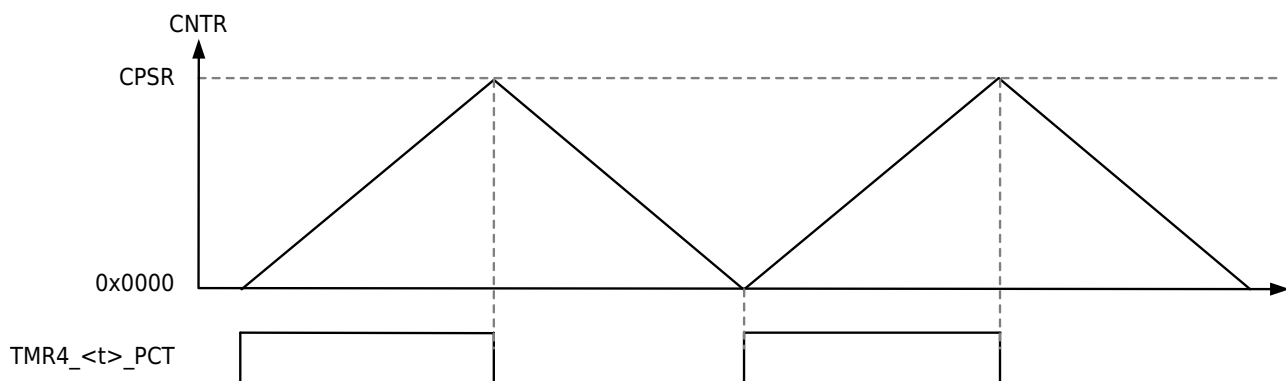


图 21-25 PWM 周期输出监测示例

专用事件输出监测端口 TMR4_<t>_ADSM 可以用来监测内部计数器的专用比较事件。通过专用扩展控制寄存器 (SCER) 的 SCER.EVTRS 位选择需要监测的专用比较输出通道, 在该通道上的比较基准值和计数器产生比较匹配时, 监测端口 TMR4_<t>_ADSM 上就会置位 (输出为高); 在该通道上的计数器计数归零时 (锯齿波溢出和三角波计数到波谷), 监测端口 TMR4_<t>_ADSM 上就会复位 (输出为低)。

需要注意, 当配置寄存器 TMR4_SCSRm 同时使能向上计数 EVT 和向下计数 EVT 条件下

(TMR4_SCSRm.UEN=1 且 TMR4_SCSRm.DEN=1), 输出到外部端口的同步信号只对向上计数 EVT 有效。

下图是比较启动模式下 (SCSRm.EVTMS=0) 输出到端口的同步信号示例, 向上计数 EVT 有效后

TMR4_<t>_ADSM 置位, 计数器下溢后 TMR4_<t>_ADSM 清零。

SCSRm.EVTMS=0

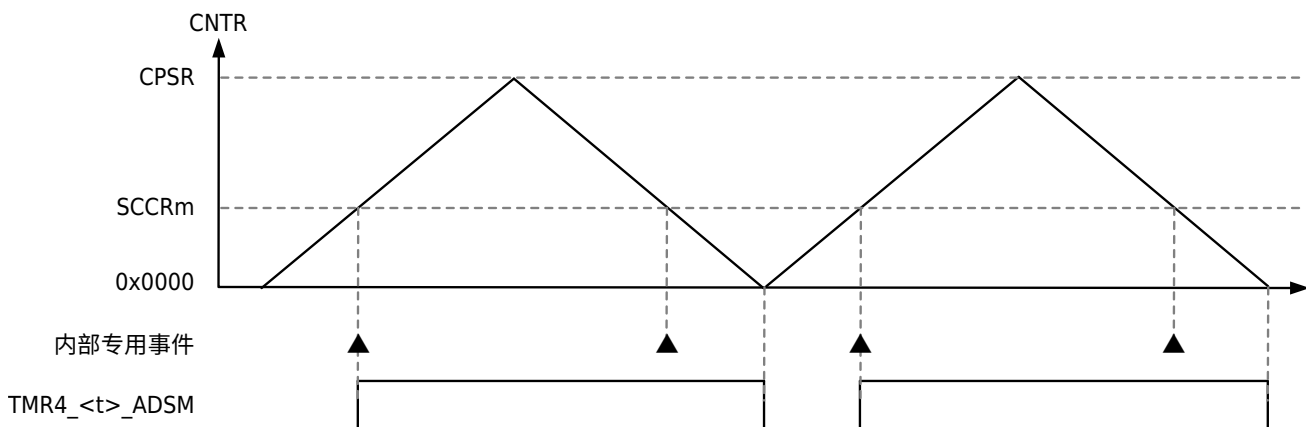


图 21-26 比较启动模式下专用事件输出 TMR4_<t>_ADSM 时序

21.3.8 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, 事件源可通过硬件触发事件选择寄存器中对应的编号设定来选择, 具体寄存器内容详见《自动运行系统 (AOS)》的“Timer4 硬件触发事件选择寄存器 (TMR4_TRGSELx) (x=0~2)”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 0 寄存器 (PWC_FCG0) 的外围电路触发功能使能位置 1。

21.3.9 同步启动

本产品所搭载的 3 个单元的 Timer4, 可以实现软件同步启动。单元 1 可以同步启动单元 2 和单元 3。

当被同步单元 m (m=2, 3) 中的 CCSR.SYNST 位设定为 1 时, 该单元同步启动功能有效。此时, 若软件设定单元 1 的 CCSR.STOP 位为 0, 被同步单元 m 的计数器与单元 1 的计数器同步启动计数。

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 Timer（包括 Timer0、Timer4、Timer6、TimerA）。使用方法和寄存器说明请参见《通用定时（Timer0）》章节描述。请不要在本定时器异步计数的情况下使用该功能。

21.3.10 中断和事件

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_<t>_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，事件编号请参见《中断控制器 (INTC)》章节“中断事件表”。

计数周期匹配中断及事件

锯齿波计数至上溢点或下溢点、三角波计数至谷点或峰点时，控制状态寄存器 (CCSR) 的 CCSR.IRQPF 或 CCSR.IRQZF 位会被置为 1。此时若设置 CCSR.IRQPEN 或 CCSR.IRQZEN 位使能中断，则在对应的时间点可触发计数周期匹配中断 (TMR4_<t>_GOVF 和 TMR4_<t>_GUDF)。

计数周期匹配事件可以用于触发别的模块，事件编号请参见《中断控制器 (INTC)》章节“中断事件表”。

重载计数匹配中断及事件

在重载功能有效时，TMR4_PFSRn 作为重载定时器的周期计数值，使能计数时从寄存器 TMR4_PFSRn 重载为计数器初始值，执行递减操作，完成一个周期后，产生重载计数中断请求，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_<t>_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，事件编号请参见《中断控制器 (INTC)》章节“中断事件表”。

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，若专用比较基准值 (SCCRM) 发生计数比较匹配事件，会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）用于触发其他模块。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_<t>_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*l (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，事件编号请参见《中断控制器 (INTC)》章节“中断事件表”。

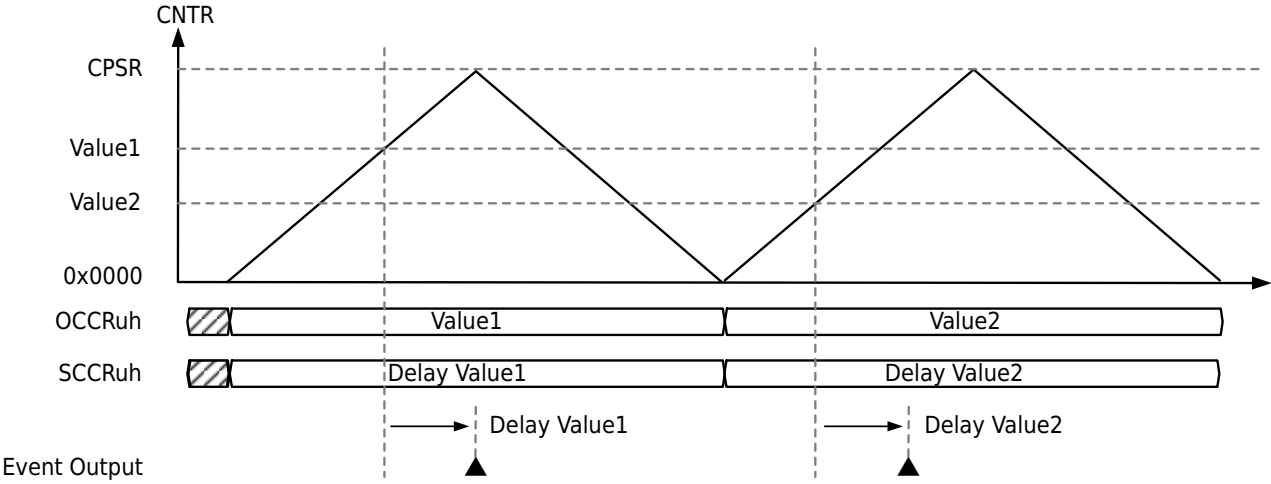


图 21-27 延时启动模式下专用事件输出信号的输出时序



说明

在延迟计数运行中，如果再次发生 OCCR 与计数器匹配事件，延迟计数器重新加载计数值并重新进行递减计数。因此，如果 OCCR 匹配事件时间间隔小于设定的延迟时间 SCCR 时，专用事件输出的请求信号可能一直不会产生。

21.4 寄存器

21.4.1 寄存器总表

- Timer4_1 基地址：0x40038000
- Timer4_2 基地址：0x40038400
- Timer4_3 基地址：0x40038E00



说明

Timer4 为定时器名称，Timer4_<t>（<t>：1~3）为该定时器单元。

表 21-3 通用控制定时器（Timer4）寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x0054	TMR4_CNTR	计数值寄存器	16
0x0050	TMR4_CPSR	周期基准寄存器	16
0x0058	TMR4_CCSR	控制状态寄存器	16
0x005A	TMR4_CVPR	有效周期寄存器	16
0x0000	TMR4_OCCRuh	通用比较基准寄存器 UH	16
0x0004	TMR4_OCCRul	通用比较基准寄存器 UL	16
0x0008	TMR4_OCCRvh	通用比较基准寄存器 VH	16
0x000C	TMR4_OCCRvl	通用比较基准寄存器 VL	16
0x0010	TMR4_OCCRwh	通用比较基准寄存器 WH	16
0x0014	TMR4_OCCRwl	通用比较基准寄存器 WL	16
0x0018	TMR4_OCCRxh	通用比较基准寄存器 XH	16
0x001C	TMR4_OCCRxl	通用比较基准寄存器 XL	16
0x0020	TMR4_OCSRu	通用控制状态寄存器 U	16
0x0024	TMR4_OCSRv	通用控制状态寄存器 V	16
0x0028	TMR4_OCSRw	通用控制状态寄存器 W	16
0x002C	TMR4_OCSRx	通用控制状态寄存器 x	16
0x0022	TMR4_OCERu	通用扩展控制寄存器 U	16
0x0026	TMR4_OCERv	通用扩展控制寄存器 V	16
0x002A	TMR4_OCERw	通用扩展控制寄存器 W	16
0x002E	TMR4_OCERx	通用扩展控制寄存器 X	16
0x0030	TMR4_OCMRuh	通用模式控制寄存器 UH	16
0x0034	TMR4_OCMRul	通用模式控制寄存器 UL	32

偏移地址	寄存器	描述	位宽
0x0038	TMR4_OCMRvh	通用模式控制寄存器 VH	16
0x003C	TMR4_OCMRvl	通用模式控制寄存器 VL	32
0x0040	TMR4_OCMRwh	通用模式控制寄存器 WH	16
0x0044	TMR4_OCMRwl	通用模式控制寄存器 WL	32
0x0048	TMR4_OCMRxh	通用模式控制寄存器 XH	16
0x004C	TMR4_OCMRxl	通用模式控制寄存器 XL	32
0x00B0	TMR4_SCCRuh	专用比较基准寄存器 UH	16
0x00B4	TMR4_SCCRul	专用比较基准寄存器 UL	16
0x00B8	TMR4_SCCRvh	专用比较基准寄存器 VH	16
0x00BC	TMR4_SCCRvl	专用比较基准寄存器 VL	16
0x00C0	TMR4_SCCRwh	专用比较基准寄存器 WH	16
0x00C4	TMR4_SCCRwl	专用比较基准寄存器 WL	16
0x00C8	TMR4_SCCRxh	专用比较基准寄存器 XH	16
0x00CC	TMR4_SCCRxl	专用比较基准寄存器 XL	16
0x00D0	TMR4_SCSRuh	专用控制状态寄存器 UH	16
0x00D4	TMR4_SCSRul	专用控制状态寄存器 UL	16
0x00D8	TMR4_SCSRvh	专用控制状态寄存器 VH	16
0x00DC	TMR4_SCSRvl	专用控制状态寄存器 VL	16
0x00E0	TMR4_SCSRwh	专用控制状态寄存器 WH	16
0x00E4	TMR4_SCSRwl	专用控制状态寄存器 WL	16
0x00E8	TMR4_SCCRxh	专用控制状态寄存器 XH	16
0x00EC	TMR4_SCSRxl	专用控制状态寄存器 XL	16
0x00D2	TMR4_SCMRuh	专用模式控制寄存器 UH	16
0x00D6	TMR4_SCMRul	专用模式控制寄存器 UL	16
0x00DA	TMR4_SCMRvh	专用模式控制寄存器 VH	16
0x00DE	TMR4_SCMRvl	专用模式控制寄存器 VL	16
0x00E2	TMR4_SCMRwh	专用模式控制寄存器 WH	16
0x00E6	TMR4_SCMRwl	专用模式控制寄存器 WL	16
0x00EA	TMR4_SCMRxh	专用模式控制寄存器 XH	16
0x00EE	TMR4_SCMRxl	专用模式控制寄存器 XL	16

偏移地址	寄存器	描述	位宽
0x00F0	TMR4_SCER	专用扩展控制寄存器	16
0x00A0	TMR4_POCRu	PWM 基本控制寄存器 U	16
0x00A4	TMR4_POCRv	PWM 基本控制寄存器 V	16
0x00A8	TMR4_POCRw	PWM 基本控制寄存器 W	16
0x00AC	TMR4_POCRx	PWM 基本控制寄存器 X	16
0x005C	TMR4_PSCR	PWM 状态控制寄存器	32
0x0082	TMR4_PFSRu	PWM 滤波控制寄存器 U	16
0x008A	TMR4_PFSRv	PWM 滤波控制寄存器 V	16
0x0092	TMR4_PFSRw	PWM 滤波控制寄存器 W	16
0x009A	TMR4_PFSRx	PWM 滤波控制寄存器 X	16
0x0084	TMR4_PDARu	PWM 死区控制寄存器 AU	16
0x0086	TMR4_PDBRu	PWM 死区控制寄存器 BU	16
0x008C	TMR4_PDARv	PWM 死区控制寄存器 AV	16
0x008E	TMR4_PDBRv	PWM 死区控制寄存器 BV	16
0x0094	TMR4_PDARw	PWM 死区控制寄存器 AW	16
0x0096	TMR4_PDBRw	PWM 死区控制寄存器 BW	16
0x009C	TMR4_PDARx	PWM 死区控制寄存器 AX	16
0x009E	TMR4_PDBRx	PWM 死区控制寄存器 BX	16
0x00F4	TMR4_RCSR	重载控制状态寄存器	32
0x00F8	TMR4_SCIR	专用中断控制寄存器	16
0x00FC	TMR4_SCFR	专用标志控制寄存器	16



说明

本节后续寄存器详细描述中, 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_<t>_OUH、TMR4_<t>_OUL、TMR4_<t>_OVH、TMR4_<t>_OVL、TMR4_<t>_OWH、TMR4_<t>_OWL、TMR4_<t>_OXH、TMR4_<t>_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_<t>_OUx、TMR4_<t>_OVx、TMR4_<t>_OWx、TMR4_<t>_Ox 的输出控制等, 其中 x=H 或 L, 且 H 或 L 的具体控制在这些寄存器中有相对应的对称位。

21.4.2 计数值寄存器（TMR4_CNTR）

Offset	Bit Position															
0x0054	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	CNTR															
Access	RW															

位/位域	标记	位名	功能描述	读写
15:0	CNTR	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  注意 在计数中时，不能向该寄存器写入值。	RW

21.4.3 周期基准寄存器（TMR4_CPSR）

Offset	Bit Position															
0x0050	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFF															
Name	CPSR															
Access	RW															

位/位域	标记	位名	功能描述	读写
15:0	CPSR	通用周期基准值	计数器的计数周期值  说明 从本地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

21.4.4 控制状态寄存器 (TMR4_CCSR)

Offset	Bit Position															
0x0058	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0040															
Name	ECKEN	Reserved	HST	SYNST	IRQZF	IRQZEN	IRQPF	IRQPEN	BUFEN	STOP	MODE	CLEAR	CKDIV			
Access	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW			

位/位域	标记	位名	功能描述	读写
15	ECKEN	时钟源选择	0b0: 内部 PCLK0 时钟 0b1: 外部 TMR4_<t>_CLK 端口输入时钟  说明 - 该位在计数器停止时设定。 - 当使用外部输入时钟进行操作时，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
14	Reserved	保留	-	-
13	HST	硬件启动使能	条件：TMR4_TRGSELx 寄存器中指定的事件发生 0b0: 条件匹配时，硬件启动无效 0b1: 条件匹配时，硬件启动有效，计数器开始计数	RW
12	SYNST	同步启动使能	0b0: 该单元与单元 1 的同步启动功能无效 0b1: 该单元与单元 1 的同步启动功能有效  说明 仅单元 2 和单元 3 支持 SYNST 功能位，单元 1 的该位为保留位，请保持初始值 0。	RW
11	IRQZF	下溢状态	0b0: 未发生计数下溢 0b1: 发生计数下溢  说明 - 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。 - 当计数器被总线复位或者写入 CLEAR=1，IRQZF 位将不会被置位。	RW
10	IRQZEN	下溢中断使能	0b0: 禁止 IRQZF 产生中断到 CPU 0b1: 允许 IRQZF 产生中断到 CPU	RW
9	IRQPF	上溢状态	0b0: 未发生计数上溢 0b1: 发生计数上溢  说明 - 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。 - 当计数器被总线复位或者写入 CLEAR=1，IRQPF 位将不会被置位。	RW

位/位域	标记	位名	功能描述	读写
8	IRQPEN	上溢中断使能	0b0: 禁止 IRQPF 产生中断到 CPU 0b1: 允许 IRQPF 产生中断到 CPU	RW
7	BUFEN	缓存使能	0b0: 禁止 CPSR 缓存功能 0b1: 使能 CPSR 缓存功能	RW
6	STOP	计数器使能	0b0: 计数器启动 0b1: 计数器停止	RW
5	MODE	波形模式	0b0: 锯齿波模式 (仅支持向上计数) 0b1: 三角波模式	RW
4	CLEAR	计数器清零	写 1 清零计数器, 写 0 无效; 读出恒为 0	RW
3:0	CKDIV	计数时钟分频	该位指示基本计数器的计数时钟分频 0b0000: 计数时钟为 PCLK0 0b0001: 计数时钟为 PCLK0/2 0b0010: 计数时钟为 PCLK0/4 0b0011: 计数时钟为 PCLK0/8 0b0100: 计数时钟为 PCLK0/16 0b0101: 计数时钟为 PCLK0/32 0b0110: 计数时钟为 PCLK0/64 0b0111: 计数时钟为 PCLK0/128 0b1000: 计数时钟为 PCLK0/256 0b1001: 计数时钟为 PCLK0/512 0b1010: 计数时钟为 PCLK0/1024 - 其他: 禁止设置  说明 时钟源为内部 PCLK0 时, 分频有效。	RW

21.4.5 有效周期寄存器 (TMR4_CVPR)

Offset	Bit Position															
0x005A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	PIC				ZIC				PIM				ZIM			
Access	RW				RW				RW				RW			

位/位域	标记	位名	功能描述	读写
15:12	PIC	上溢中断屏蔽状态	PIC 是一个向下计数器，当 CNTR 计数上溢中断时 PIC 的值减 1，读取 PIC 可以获得当前剩余需要屏蔽的上溢中断个数。 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： 1. 计数器清零动作发生 (CCSR.CLEAR 写 1) 2. PIC=0 且 CNTR 计数发生上溢中断	RW
11:8	ZIC	下溢中断屏蔽状态	ZIC 是一个向下计数器，当 CNTR 计数下溢中断时 ZIC 的值减 1，读取 ZIC 可以获得当前剩余需要屏蔽的下溢中断个数。 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： 1. 计数器清零动作发生 (CCSR.CLEAR 写 1) 2. ZIC=0 且 CNTR 计数发生下溢中断	RW
7:4	PIM	上溢中断屏蔽设定	设置需要屏蔽的上溢中断个数 满足以下之一，PIM 的值载入 PIC： 1. 计数器清零动作发生 (CCSR.CLEAR 写 1) 2. PIC=0 且 CNTR 计数发生上溢中断	RW
3:0	ZIM	下溢中断屏蔽设定	设置需要屏蔽的下溢中断个数 满足以下之一，ZIM 的值载入 ZIC： 1. 计数器清零动作发生 (CCSR.CLEAR 写 1) 2. ZIC=0 且 CNTR 计数发生下溢中断	RW

21.4.6 通用比较基准寄存器（TMR4_OCCRm）

偏移地址（Offset）：0x0000（TMR4_OCCRu_h），0x0004（TMR4_OCCRu_l），0x0008（TMR4_OCCRv_h），0x000C（TMR4_OCCRv_l），0x0010（TMR4_OCCRw_h），0x0014（TMR4_OCCRw_l），0x0018（TMR4_OCCRx_h），0x001C（TMR4_OCCRx_l）

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	OCCR															
Access	RW															

位/位域	标记	位名	功能描述	读写
15:0	OCCR	通用比较基准值	通用比较基准值	RW
			 说明 从本地地址区域读取数据时，读取的并非缓冲器寄存器的值，而是 OCCR 寄存器的值。	

21.4.7 通用控制状态寄存器 (TMR4_OCSRn)

偏移地址 (Offset) : 0x0020 (TMR4_OCSRu) , 0x0024 (TMR4_OCSRv) , 0x0028 (TMR4_OCSRw) , 0x002C (TMR4_OCSRx)

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFF00															
Name	Reserved								OCFL	OCFH	OCIEL	OCIEH	OCPL	OCPH	OCEL	OCEH
Access									RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
15:8	Reserved	保留	所有位读出为 1, 写入时必须写 1	-
7	OCFL	计数匹配标志 L	0b0: 计数器计数值与 OCCR*l 设定值不相等 0b1: 计数器计数值与 OCCR*l 设定值相等  说明 该位在 OCEL=1 时有效。	RW
6	OCFH	计数匹配标志 H	0b0: 计数器计数值与 OCCR*h 设定值不相等 0b1: 计数器计数值与 OCCR*h 设定值相等  说明 该位在 OCEH=1 时有效。	RW
5	OCIEL	计数匹配中断使能 L	0b0: OCFL 置位时, 不发生中断 0b1: OCFL 置位时, 发生中断	RW
4	OCIEH	计数匹配中断使能 H	0b0: OCFH 置位时, 不发生中断 0b1: OCFH 置位时, 发生中断	RW
3	OCPL	比较输出无效时端口状态 L	写入: 0b0: OCEL=0 时, in_op*l 输出低电平 0b1: OCEL=0 时, in_op*l 输出高电平 读取: 0b0: in_op*l 当前状态为低电平 0b1: in_op*l 当前状态为高电平	RW
2	OCPL	比较输出无效时端口状态 H	写入: 0b0: OCEH=0 时, in_op*h 输出低电平 0b1: OCEH=0 时, in_op*h 输出高电平 读取: 0b0: in_op*h 当前状态为低电平 0b1: in_op*h 当前状态为高电平	RW
1	OCEL	端口输出选择 L	0b0: 比较输出无效, in_op*l 端口状态由 OCPL 决定 0b1: 比较输出有效, in_op*l 端口状态由 OCMR*l 的设定和 OCFL 状态决定	RW
0	OCEH	端口输出选择 H	0b0: 比较输出无效, in_op*h 端口状态由 OCPH 决定 0b1: 比较输出有效, in_op*h 端口状态由 OCMR*h 的设定和 OCFH 状态决定	RW

21.4.8 通用扩展控制寄存器 (TMR4_OCERn)

偏移地址 (Offset) : 0x0022 (TMR4_OCERu) , 0x0026 (TMR4_OCERv) , 0x002A (TMR4_OCERw) , 0x002E (TMR4_OCERx)

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved		MCECL	MCECH	LMML	LMMH	LMCL	LMCH	MLBUFEN		MHBUFEN		CLBUFEN		CHBUFEN	
Access			RW	RW	RW	RW	RW	RW	RW		RW		RW		RW	

位/位域	标记	位名	功能描述	读写
15:14	Reserved	保留	-	-
13	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 0b0：禁止该条件作为匹配条件 0b1：使能该条件作为匹配条件，条件满足时 OCCR*I 计数匹配  说明 - 只能在输出比较功能无效 (OCSR.OCEL=0) 时修改该位。 - 具体请参考 TMR4_OCMRm 寄存器说明的注意事项。	RW
12	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 0b0：禁止该条件作为匹配条件 0b1：使能该条件作为匹配条件，条件满足时 OCCR*h 计数匹配  说明 - 只能在输出比较功能无效 (OCSR.OCEH=0) 时修改该位。 - 具体请参考 TMR4_OCMRm 寄存器说明的注意事项。	RW
11	LMML	OCMR*I 周期间隔响应功能链接 L	0b0：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 0b1：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=0b0000 (计数上溢时) 或 CVPR.ZIC=0b0000 (计数下溢时)	RW
10	LMMH	OCMR*h 周期间隔响应功能链接 H	0b0：周期间隔响应功能链接无效，OCMR*h 的缓存传送由 MLBUFEN 设定决定 0b1：周期间隔响应功能链接有效，OCMR*h 的缓存传送在 MLBUFEN 设定的基础上，还必须满足 CVPR.PIC=0b0000 (计数上溢时) 或 CVPR.ZIC=0b0000 (计数下溢时)	RW

位/位域	标记	位名	功能描述	读写
9	LMCL	OCCR*I 周期间隔响应功能链接 L	0b0: 周期间隔响应功能链接无效, OCCR*I 的缓存传送由 MLBUFEN 设定决定 0b1: 周期间隔响应功能链接有效, OCCR*I 的缓存传送在 MLBUFEN 设定的基础上, 还必须满足 CVPR.PIC=0b0000 (计数上溢时) 或 CVPR.ZIC=0b0000 (计数下溢时)	RW
8	LMCH	OCCR*h 周期间隔响应功能链接 H	0b0: 周期间隔响应功能链接无效, OCCR*h 的缓存传送由 MLBUFEN 设定决定 0b1: 周期间隔响应功能链接有效, OCCR*h 的缓存传送在 MLBUFEN 设定的基础上, 还必须满足 CVPR.PIC=0b0000 (计数上溢时) 或 CVPR.ZIC=0b0000 (计数下溢时)	RW
7:6	MLBUFEN	OCMR*I 缓存传送	0b00: OCMR*I 缓存寄存器的值直接写入 OCMR*I 0b01: OCMR*I 缓存寄存器的值在计数下溢时写入 OCMR*I 0b10: OCMR*I 缓存寄存器的值在计数上溢时写入 OCMR*I 0b11: OCMR*I 缓存寄存器的值在计数下溢或上溢时写入 OCMR*I	RW
5:4	MHBUFEN	OCMR*h 缓存传送	0b00: OCMR*h 缓存寄存器的值直接写入 OCMR*h 0b01: OCMR*h 缓存寄存器的值在计数下溢时写入 OCMR*h 0b10: OCMR*h 缓存寄存器的值在计数上溢时写入 OCMR*h 0b11: OCMR*h 缓存寄存器的值在计数下溢或上溢时写入 OCMR*h	RW
3:2	CLBUFEN	OCCR*I 缓存传送	0b00: OCCR*I 缓存寄存器的值直接写入 OCCR*I 0b01: OCCR*I 缓存寄存器的值在计数下溢时写入 OCCR*I 0b10: OCCR*I 缓存寄存器的值在计数上溢时写入 OCCR*I 0b11: OCCR*I 缓存寄存器的值在计数下溢或上溢时写入 OCCR*I	RW
1:0	CHBUFEN	OCCR*h 缓存传送	0b00: OCCR*h 缓存寄存器的值直接写入 OCCR*h 0b01: OCCR*h 缓存寄存器的值在计数下溢时写入 OCCR*h 0b10: OCCR*h 缓存寄存器的值在计数上溢时写入 OCCR*h 0b11: OCCR*h 缓存寄存器的值在计数下溢或上溢时写入 OCCR*h	RW

21.4.9 通用模式控制寄存器 (TMR4_OCMRm)

偏移地址 (Offset) : 0x0030 (TMR4_OCMRuh) , 0x0034 (TMR4_OCMRul) , 0x0038 (TMR4_OCMRvh) , 0x003C (TMR4_OCMRvl) , 0x0040 (TMR4_OCMRwh) , 0x0044 (TMR4_OCMRwl) , 0x0048 (TMR4_OCMRxh) , 0x004C (TMR4_OCMRxl)

TMR4_OCMRuh、TMR4_OCMRvh、TMR4_OCMRwh、TMR4_OCMRxh 寄存器:

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	OPNZRH		OPNPKH		OPZRH		OPUCH		OPPKH		OPDCH		OCFZRH		OCFUCH	
Access	RW		RW		RW		RW		RW		RW		RW		RW	

表 21-19 OCCR*h 计数匹配一览表

条件编号	OCCR*h 计数是否匹配
1	否
2	是

位/位域	标记	位名	功能描述	读写
15:14	OPNZRH	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 0b00: 满足条件时, in_op*h 保持不变 0b01: 满足条件时, in_op*h 输出高电平 0b10: 满足条件时, in_op*h 输出低电平 0b11: 满足条件时, in_op*h 输出反转	RW
13:12	OPNPKH	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 0b00: 满足条件时, in_op*h 保持不变 0b01: 满足条件时, in_op*h 输出高电平 0b10: 满足条件时, in_op*h 输出低电平 0b11: 满足条件时, in_op*h 输出反转	RW
11:10	OPZRH	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 0b00: 满足条件时, in_op*h 保持不变 0b01: 满足条件时, in_op*h 输出高电平 0b10: 满足条件时, in_op*h 输出低电平 0b11: 满足条件时, in_op*h 输出反转	RW
9:8	OPUCH	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 0b00: 满足条件时, in_op*h 保持不变 0b01: 满足条件时, in_op*h 输出高电平 0b10: 满足条件时, in_op*h 输出低电平 0b11: 满足条件时, in_op*h 输出反转	RW

位/位域	标记	位名	功能描述	读写
7:6	OPPKH	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 ● 0b00：满足条件时，in_op*h 保持不变 ● 0b01：满足条件时，in_op*h 输出高电平 ● 0b10：满足条件时，in_op*h 输出低电平 ● 0b11：满足条件时，in_op*h 输出反转	RW
5:4	OPDCH	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 ● 0b00：满足条件时，in_op*h 保持不变 ● 0b01：满足条件时，in_op*h 输出高电平 ● 0b10：满足条件时，in_op*h 输出低电平 ● 0b11：满足条件时，in_op*h 输出反转	RW
3	OCFZRH	下溢点 OCFH 状态 H	条件 (同时满足)：计数下溢，OCCR*h 计数匹配 ● 0b0：满足条件时，OCSR.OCFH 位保持不变 ● 0b1：满足条件时，OCSR.OCFH 位置位	RW
2	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 ● 0b0：满足条件时，OCSR.OCFH 位保持不变 ● 0b1：满足条件时，OCSR.OCFH 位置位	RW
1	OCFPKH	上溢点 OCFH 状态 H	条件 (同时满足)：计数上溢，OCCR*h 计数匹配 ● 0b0：满足条件时，OCSR.OCFH 位保持不变 ● 0b1：满足条件时，OCSR.OCFH 位置位	RW
0	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 ● 0b0：满足条件时，OCSR.OCFH 位保持不变 ● 0b1：满足条件时，OCSR.OCFH 位置位	RW

TMR4_OC MRul、TMR4_OC MRvl、TMR4_OC MRwl、TMR4_OC MRxl 寄存器:

Offset	Bit Position																																		
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Reset	0x00000000																																		
Name	EOPNZRL	EOPNPKL		EOPZRL		EOPUCL		EOPPKL		EOPDCL		EOPNUCL		EOPNDCL		OPNZRL		OPNPKL		OPZRL		OPUCL		OPPKL		OPDCL		OCFZRL		OCFUCL		OCFPKL		OCFDCL	
Access	RW	RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW	

表 21-22 OCCR*h/I 计数匹配一览表

条件编号	OCCR*h 计数是否匹配	OCCR*I 计数是否匹配
1	否	否
2	否	是
3	是	否
4	是	是

位/位域	标记	位名	功能描述	读写
31:30	EOPNZRL	下溢条件 3 端口状态 L	下溢条件 3 (同时满足): 计数下溢, OCCR*I 计数不匹配, OCCR*h 计数匹配 ● 0b00: 满足条件时, in_op*I 保持不变 ● 0b01: 满足条件时, in_op*I 输出高电平 ● 0b10: 满足条件时, in_op*I 输出低电平 ● 0b11: 满足条件时, in_op*I 输出反转	RW
29:28	EOPNPKL	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 ● 0b00: 满足条件时, in_op*I 保持不变 ● 0b01: 满足条件时, in_op*I 输出高电平 ● 0b10: 满足条件时, in_op*I 输出低电平 ● 0b11: 满足条件时, in_op*I 输出反转	RW
27:26	EOPZRL	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 ● 0b00: 满足条件时, in_op*I 保持不变 ● 0b01: 满足条件时, in_op*I 输出高电平 ● 0b10: 满足条件时, in_op*I 输出低电平 ● 0b11: 满足条件时, in_op*I 输出反转	RW
25:24	EOPUCL	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 ● 0b00: 满足条件时, in_op*I 保持不变 ● 0b01: 满足条件时, in_op*I 输出高电平 ● 0b10: 满足条件时, in_op*I 输出低电平 ● 0b11: 满足条件时, in_op*I 输出反转	RW
23:22	EOPPKL	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 ● 0b00: 满足条件时, in_op*I 保持不变 ● 0b01: 满足条件时, in_op*I 输出高电平 ● 0b10: 满足条件时, in_op*I 输出低电平	RW

位/位域	标记	位名	功能描述	读写
			0b11: 满足条件时, in_op*I 输出反转	
21:20	EOPDCL	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 0b00: 满足条件时, in_op*I 保持不变 0b01: 满足条件时, in_op*I 输出高电平 0b10: 满足条件时, in_op*I 输出低电平 0b11: 满足条件时, in_op*I 输出反转	RW
19:18	EOPNUCL	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 0b00: 满足条件时, in_op*I 保持不变 0b01: 满足条件时, in_op*I 输出高电平 0b10: 满足条件时, in_op*I 输出低电平 0b11: 满足条件时, in_op*I 输出反转	RW
17:16	EOPNDCL	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 0b00: 满足条件时, in_op*I 保持不变 0b01: 满足条件时, in_op*I 输出高电平 0b10: 满足条件时, in_op*I 输出低电平 0b11: 满足条件时, in_op*I 输出反转	RW
15:14	OPNZRL	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 0b00: 满足条件时, in_op*I 保持不变 0b01: 满足条件时, in_op*I 输出高电平 0b10: 满足条件时, in_op*I 输出低电平 0b11: 满足条件时, in_op*I 输出反转	RW
13:12	OPNPKL	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 0b00: 满足条件时, in_op*I 保持不变 0b01: 满足条件时, in_op*I 输出高电平 0b10: 满足条件时, in_op*I 输出低电平 0b11: 满足条件时, in_op*I 输出反转	RW
11:10	OPZRL	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 0b00: 满足条件时, in_op*I 保持不变 0b01: 满足条件时, in_op*I 输出高电平 0b10: 满足条件时, in_op*I 输出低电平 0b11: 满足条件时, in_op*I 输出反转	RW
9:8	OPUCL	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 0b00: 满足条件时, in_op*I 保持不变 0b01: 满足条件时, in_op*I 输出高电平 0b10: 满足条件时, in_op*I 输出低电平 0b11: 满足条件时, in_op*I 输出反转	RW

位/位域	标记	位名	功能描述	读写
7:6	OPPKL	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 ● 0b00：满足条件时，in_op*I 保持不变 ● 0b01：满足条件时，in_op*I 输出高电平 ● 0b10：满足条件时，in_op*I 输出低电平 ● 0b11：满足条件时，in_op*I 输出反转	RW
5:4	OPDCL	向下计数条件 2 端口状态 L	向下计数条件 2 (同时满足)：计数器向下计数，OCCR*I 计数匹配，OCCR*h 计数不匹配 ● 0b00：满足条件时，in_op*I 保持不变 ● 0b01：满足条件时，in_op*I 输出高电平 ● 0b10：满足条件时，in_op*I 输出低电平 ● 0b11：满足条件时，in_op*I 输出反转	RW
3	OCFZRL	下溢点 OCFL 状态 L	条件 (同时满足)：计数下溢，OCCR*I 计数匹配 ● 0b0：满足条件时，OCSR.OCFL 位保持不变 ● 0b1：满足条件时，OCSR.OCFL 位置位	RW
2	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 ● 0b0：满足条件时，OCSR.OCFL 位保持不变 ● 0b1：满足条件时，OCSR.OCFL 位置位	RW
1	OCFPKL	上溢点 OCFL 状态 L	条件 (同时满足)：计数上溢，OCCR*I 计数匹配 ● 0b0：满足条件时，OCSR.OCFL 位保持不变 ● 0b1：满足条件时，OCSR.OCFL 位置位	RW
0	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 ● 0b0：满足条件时，OCSR.OCFL 位保持不变 ● 0b1：满足条件时，OCSR.OCFL 位置位	RW



注意

- 从本地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_<t>_O*I 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 设定的条件：
 - CCSR.MODE=1 && 计数上溢 && OCCR*h=0xFFFF
 - OCER*.MCECH=1 && 计数上溢 && OCCR*h≥CNTR
- 下面 2 种情况满足时，也认为是符合 OCMR*I.OPPKL 设定的条件：
 - CCSR.MODE=1 && 计数上溢 && OCCR*I=0xFFFF
 - OCER*.MCECL=1 && 计数上溢 && OCCR*I≥CNTR

21.4.10 专用比较基准寄存器（TMR4_SCCRm）

偏移地址（Offset）：0x00B0（TMR4_SCCRuh），0x00B4（TMR4_SCCRul），0x00B8（TMR4_SCCRvh），0x00BC（TMR4_SCCRvl），0x00C0（TMR4_SCCRwh），0x00C4（TMR4_SCCRwl），0x00C8（TMR4_SCCRxh），0x00CC（TMR4_SCCRxl）

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	SCCR															
Access	RW															

位/位域	标记	位名	功能描述	读写
15:0	SCCR	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）  说明 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

21.4.11 专用控制状态寄存器 (TMR4_SCSRm)

偏移地址 (Offset) : 0x00D0 (TMR4_SCSRuh) , 0x00D4 (TMR4_SCSRul) , 0x00D8 (TMR4_SCSRvh) , 0x00DC (TMR4_SCSRvl) , 0x00E0 (TMR4_SCSRwh) , 0x00E4 (TMR4_SCSRwl) , 0x00E8 (TMR4_SCSRxh) , 0x00EC (TMR4_SCSRxl)

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	ZEN	UEN	PEN	DEN	Reserved		EVTDS	EVTMS	Reserved		LMC	EVTOS			BUFEN	
Access	RW	RW	RW	RW			RW	RW			RW	RW			RW	

位/位域	标记	位名	功能描述	读写
15	ZEN	下溢点 EVT 使能	0b0: 计数下溢时, EVT 无操作 0b1: 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
14	UEN	向上计数 EVT 使能	0b0: 向上计数时, EVT 无操作 0b1: 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
13	PEN	上溢点 EVT 使能	0b0: 计数上溢时, EVT 无操作 0b1: 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
12	DEN	向下计数 EVT 使能	0b0: 向下计数时, EVT 无操作 0b1: 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
11:10	Reserved	保留	-	-
9	EVTDS	EVT 延时对象选择	0b0: 在延时启动模式时, OCCR*h 作为延时比较匹配对象 0b1: 在延时启动模式时, OCCR*l 作为延时比较匹配对象  说明 该位在 EVTMS=0 时无效。	RW

位/位域	标记	位名	功能描述	读写
8	EVTMS	EVT 模式选择	0b0: 比较启动模式 (CNTR 和 SCCR 的比较结果触发) 0b1: 延时启动模式 (比较匹配事件经 SCCR 延迟后触发)	RW
7:6	Reserved	保留	-	-
5	LMC	周期间隔响应功能链接	0b0: 周期间隔响应功能链接无效, SCCR 的缓存传送由 BUFEN 设定决定 0b1: 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=0b0000 (计数上溢时) 或 CVPR.ZIC=0b0000 (计数下溢时)	RW
4:2	EVTOS	EVT 输出选择	0b000: Special Event 0 (TMR4_<t>_SCM0) 的 EVT 输出有效 0b001: Special Event 1 (TMR4_<t>_SCM1) 的 EVT 输出有效 0b010: Special Event 2 (TMR4_<t>_SCM2) 的 EVT 输出有效 0b011: Special Event 3 (TMR4_<t>_SCM3) 的 EVT 输出有效 0b100: Special Event 4 (TMR4_<t>_SCM4) 的 EVT 输出有效 0b101: Special Event 5 (TMR4_<t>_SCM5) 的 EVT 输出有效 0b110: Special Event 6 (TMR4_<t>_SCM6) 的 EVT 输出有效 0b111: Special Event 7 (TMR4_<t>_SCM7) 的 EVT 输出有效	RW
1:0	BUFEN	SCCR&SCMR 缓存传送	0b00: SCCR、SCMR 缓存寄存器的值直接写入 SCCR、SCMR 0b01: SCCR、SCMR 缓存寄存器的值在计数下溢时写入 SCCR、SCMR 0b10: SCCR、SCMR 缓存寄存器的值在计数上溢时写入 SCCR、SCMR 0b11: SCCR、SCMR 缓存寄存器的值在计数下溢或上溢时写入 SCCR、SCMR	RW

21.4.12 专用扩展控制寄存器 (TMR4_SCER)

Offset	Bit Position															
0x00F0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved							PCTS	Reserved				EVTRS			
Access								RW					RW			

位/位域	标记	位名	功能描述	读写
15:9	Reserved	保留	-	-
8	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 ● 0b0：禁止计数方向信号输出到 TMR4_<t>_PCT 端口 ● 0b1：允许计数方向信号输出到 TMR4_<t>_PCT 端口	RW
7:4	Reserved	保留	-	-
3:0	EVTRS	专用事件输出选择	● 0b0000：禁止专用事件输出到端口 ● 0b0001：Special Event 0 (TMR4_<t>_SCM0) 的 EVT 输出至 TMR4_<t>_ADSM 端口 ● 0b0010：Special Event 1 (TMR4_<t>_SCM1) 的 EVT 输出至 TMR4_<t>_ADSM 端口 ● 0b0011：Special Event 2 (TMR4_<t>_SCM2) 的 EVT 输出至 TMR4_<t>_ADSM 端口 ● 0b0100：Special Event 3 (TMR4_<t>_SCM3) 的 EVT 输出至 TMR4_<t>_ADSM 端口 ● 0b0101：Special Event 4 (TMR4_<t>_SCM4) 的 EVT 输出至 TMR4_<t>_ADSM 端口 ● 0b0110：Special Event 5 (TMR4_<t>_SCM5) 的 EVT 输出至 TMR4_<t>_ADSM 端口 ● 0b1000：Special Event 6 (TMR4_<t>_SCM6) 的 EVT 输出至 TMR4_<t>_ADSM 端口 ● 0b1001：Special Event 7 (TMR4_<t>_SCM7) 的 EVT 输出至 TMR4_<t>_ADSM 端口 ● 其他：禁止设置	RW

21.4.13 专用模式控制寄存器 (TMR4_SCMRm)

偏移地址 (Offset) : 0x00D2 (TMR4_SCMRuh) , 0x00D6 (TMR4_SCMRul) , 0x00DA (TMR4_SCMRvh) , 0x00DE (TMR4_SCMRvl) , 0x00E2 (TMR4_SCMRwh) , 0x00E6 (TMR4_SCMRwl) , 0x00EA (TMR4_SCMRxh) , 0x00EE (TMR4_SCMRxl)

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFF00															
Name	Reserved								MPCE	MZCE	Reserved			AMC		
Access									RW	RW				RW		

位/位域	标记	位名	功能描述	读写
15:8	Reserved	保留	所有位读出于 1，写入时必须写 1	-
7	MPCE	周期间隔响应使能	0b0: 禁止 AMC 与 CVPR.PIC 比较 0b1: 使能 AMC 与 CVPR.PIC 比较	RW
6	MZCE	周期间隔响应使能	0b0: 禁止 AMC 与 CVPR.ZIC 比较 0b1: 使能 AMC 与 CVPR.ZIC 比较	RW
5:4	Reserved	保留	-	-
3:0	AMC	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

说明

从本地址区域读取数据时，读取的并非缓冲器寄存器的值，而是 SCMR 寄存器的值。

21.4.14 PWM 基本控制寄存器 (TMR4_POCRn)

偏移地址 (Offset) : 0x00A0 (TMR4_POCRu) , 0x00A4 (TMR4_POCRv) , 0x00A8 (TMR4_POCRw) , 0x00AC (TMR4_POCRx)

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFF00															
Name	Reserved								LVL5		PWMMD		Reserved	DIVCK		
Access									RW		RW			RW		

位/位域	标记	位名	功能描述	读写
15:8	Reserved	保留	所有位读出的为 1, 写入时必须写 1	-
7:6	LVL5	PWM 输出极性控制	0b00: TMR4_<t>_O*H 和 TMR4_<t>_O*L 的输出都不反转 0b01: TMR4_<t>_O*H 和 TMR4_<t>_O*L 的输出都反转 0b10: TMR4_<t>_O*H 的输出反转, TMR4_<t>_O*L 的输出不反转 0b11: TMR4_<t>_O*H 的输出不反转, TMR4_<t>_O*L 的输出反转	RW
5:4	PWMMD	PWM 输出模式	0b00: 直通模式 0b01: 死区定时器模式 0b10: 死区定时器滤波模式 0b11: 禁止设置	RW
3	Reserved	保留	-	-
2:0	DIVCK	计数时钟分频	该位指示滤波计数器、死区计数器和重载计数器的计数时钟分频 0b000: 计数时钟为 PCLK0 0b001: 计数时钟为 PCLK0/2 0b010: 计数时钟为 PCLK0/4 0b011: 计数时钟为 PCLK0/8 0b100: 计数时钟为 PCLK0/16 0b101: 计数时钟为 PCLK0/32 0b110: 计数时钟为 PCLK0/64 0b111: 计数时钟为 PCLK0/128	RW

21.4.15 PWM 状态控制寄存器 (TMR4_PSCR)

[illegible]

位/位域	标记	位名	功能描述	读写
31:30	OSXL	TMR4_<t>_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 0b00：满足以上条件之一时，TMR4_<t>_OXL 端口正常输出 0b01：满足以上条件之一时，TMR4_<t>_OXL 端口输出为 Hi-Z 0b10：满足以上条件之一时，TMR4_<t>_OXL 端口输出固定为低电平 0b11：满足以上条件之一时，TMR4_<t>_OXL 端口输出固定为高电平  说明 只能在 Timer4 停止（CCSR.STOP=1）时修改该位。	RW
29:28	OSXH	TMR4_<t>_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 0b00：满足以上条件之一时，TMR4_<t>_OXH 端口正常输出 0b01：满足以上条件之一时，TMR4_<t>_OXH 端口输出为 Hi-Z 0b10：满足以上条件之一时，TMR4_<t>_OXH 端口输出固定为低电平 0b11：满足以上条件之一时，TMR4_<t>_OXH 端口输出固定为高电平  说明 只能在 Timer4 停止（CCSR.STOP=1）时修改该位。	RW
27:26	OSWL	TMR4_<t>_OWL 输出禁止状态	条件：1.发生 EMB 事件；2.软件写 MOE=0；3. MOE=1&OEWL=0 0b00：满足以上条件之一时，TMR4_<t>_OWL 端口正常输出 0b01：满足以上条件之一时，TMR4_<t>_OWL 端口输出为 Hi-Z 0b10：满足以上条件之一时，TMR4_<t>_OWL 端口输出固定为低电平 0b11：满足以上条件之一时，TMR4_<t>_OWL 端口输出固定为高电平  说明 只能在 Timer4 停止（CCSR.STOP=1）时修改该位。	RW

位/位域	标记	位名	功能描述	读写
25:24	OSWH	TMR4_<t>_OWH 输出禁止状态	条件: 1.发生 EMB 事件; 2.软件写 MOE=0; 3. MOE=1&OEWH=0 0b00: 满足以上条件之一时, TMR4_<t>_OWH 端口正常输出 0b01: 满足以上条件之一时, TMR4_<t>_OWH 端口输出为 Hi-Z 0b10: 满足以上条件之一时, TMR4_<t>_OWH 端口输出固定为低电平 0b11: 满足以上条件之一时, TMR4_<t>_OWH 端口输出固定为高电平  说明 只能在 Timer4 停止 (CCSR.STOP=1) 时修改该位。	RW
23:22	OSVL	TMR4_<t>_OVL 输出禁止状态	条件: 1.发生 EMB 事件; 2.软件写 MOE=0; 3. MOE=1&OEVL=0 0b00: 满足以上条件之一时, TMR4_<t>_OVL 端口正常输出 0b01: 满足以上条件之一时, TMR4_<t>_OVL 端口输出为 Hi-Z 0b10: 满足以上条件之一时, TMR4_<t>_OVL 端口输出固定为低电平 0b11: 满足以上条件之一时, TMR4_<t>_OVL 端口输出固定为高电平  说明 只能在 Timer4 停止 (CCSR.STOP=1) 时修改该位。	RW
21:20	OSVH	TMR4_<t>_OVH 输出禁止状态	条件: 1.发生 EMB 事件; 2.软件写 MOE=0; 3. MOE=1&OEVL=0 0b00: 满足以上条件之一时, TMR4_<t>_OVH 端口正常输出 0b01: 满足以上条件之一时, TMR4_<t>_OVH 端口输出为 Hi-Z 0b10: 满足以上条件之一时, TMR4_<t>_OVH 端口输出固定为低电平 0b11: 满足以上条件之一时, TMR4_<t>_OVH 端口输出固定为高电平  说明 只能在 Timer4 停止 (CCSR.STOP=1) 时修改该位。	RW
19:18	OSUL	TMR4_<t>_OUL 输出禁止状态	条件: 1.发生 EMB 事件; 2.软件写 MOE=0; 3. MOE=1&OEUL=0 0b00: 满足以上条件之一时, TMR4_<t>_OUL 端口正常输出 0b01: 满足以上条件之一时, TMR4_<t>_OUL 端口输出为 Hi-Z 0b10: 满足以上条件之一时, TMR4_<t>_OUL 端口输出固定为低电平 0b11: 满足以上条件之一时, TMR4_<t>_OUL 端口输出固定为高电平	RW

位/位域	标记	位名	功能描述	读写
19:18	OSUL	TMR4_<t>_O UL 输出禁止 状态	 说明 只能在 Timer4 停止 (CCSR.STOP=1) 时修改该位。	RW
17:16	OSUH	TMR4_<t>_O UH 输出禁止 状态	条件: 1.发生 EMB 事件; 2.软件写 MOE=0; 3. MOE=1&OEUH=0 0b00: 满足以上条件之一时, TMR4_<t>_OUH 端口正 常输出 0b01: 满足以上条件之一时, TMR4_<t>_OUH 端口输 出为 Hi-Z 0b10: 满足以上条件之一时, TMR4_<t>_OUH 端口输 出固定为低电平 0b11: 满足以上条件之一时, TMR4_<t>_OUH 端口输 出固定为高电平  说明 只能在 Timer4 停止 (CCSR.STOP=1) 时修改该位。	RW
15:12	Reserved	保留	-	-
11:10	ODT	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 0b0x: 满足条件时, 立即生效 0b10: 满足条件时, 计数器下次下溢时生效 0b11: 满足条件时, 计数器下次上溢时生效	RW
9	AOE	自动输出使能	0b0: EMB 事件清除后 MOE 位只能软件置 1 以恢复 PWM 正常输出 0b1: 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  说明 置位 EMB_STATCLR 或清零 EMB_SOE 可以清除 EMB 事件。	RW
8	MOE	主输出使能	0b0: TMR4_<t>_O*H/L 端口输出由 OS*H/L 位指定的 状态 0b1: OE*H、OE*L 位设定为使能时 TMR4_<t>_O*H、 TMR4_<t>_O*L 端口输出使能  说明 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
7	OEXL	TMR4_<t>_O XL 输出使能	0b0: TMR4_<t>_OXL 端口输出状态由 OSXL 位设定 0b1: MOE=1 时, TMR4_<t>_OWL 端口输出正常 PWM 波形	RW
6	OEXH	TMR4_<t>_O XH 输出使能	0b0: TMR4_<t>_OXH 端口输出状态由 OSXH 位设定 0b1: MOE=1 时, TMR4_<t>_OXH 端口输出正常 PWM 波形	RW
5	OEWL	TMR4_<t>_O WL 输出使能	0b0: TMR4_<t>_OWL 端口输出状态由 OSWL 位设定 0b1: MOE=1 时, TMR4_<t>_OWL 端口输出正常 PWM 波形	RW
4	OEWH	TMR4_<t>_O WH 输出使能	0b0: TMR4_<t>_OWH 端口输出状态由 OSWH 位设定 0b1: MOE=1 时, TMR4_<t>_OWH 端口输出正常 PWM 波形	RW

位/位域	标记	位名	功能描述	读写
3	OEVL	TMR4_<t>_OVL 输出使能	0b0: TMR4_<t>_OVL 端口输出状态由 OSVL 位设定 0b1: MOE=1 时, TMR4_<t>_OVL 端口输出正常 PWM 波形	RW
2	OEVL	TMR4_<t>_OVH 输出使能	0b0: TMR4_<t>_OVH 端口输出状态由 OSVH 位设定 0b1: MOE=1 时, TMR4_<t>_OVH 端口输出正常 PWM 波形	RW
1	OEUL	TMR4_<t>_OUL 输出使能	0b0: TMR4_<t>_OUL 端口输出状态由 OSUL 位设定 0b1: MOE=1 时, TMR4_<t>_OUL 端口输出正常 PWM 波形	RW
0	OEUH	TMR4_<t>_OUH 输出使能	0b0: TMR4_<t>_OUH 端口输出状态由 OSUH 位设定 0b1: MOE=1 时, TMR4_<t>_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:

MOE 位	OE*L/H 位	OS*L/H 位	TMR4_<t>_O*L/H 输出状态
0b1	0b0	0b00	正常输出 (Timer4 驱动)
		0b01	Hi-Z
		0b10	低电平
		0b11	高电平
	0b1	0bXX	正常输出 (Timer4 驱动)
0b0	0bX	0b00	正常输出 (Timer4 驱动)
		0b01	Hi-Z
		0b10	低电平
		0b11	高电平

21.4.16 PWM 滤波控制寄存器 (TMR4_PFSRn)

偏移地址 (Offset) : 0x0082 (TMR4_PFSRu) , 0x008A (TMR4_PFSRv) , 0x0092 (TMR4_PFSRw) , 0x009A (TMR4_PFSRx)

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	PFSR															
Access	RW															

位/位域	标记	位名	功能描述	读写
15:0	PFSR	滤波初始值	滤波计数初始值  说明 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 16 位重载计数器, 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

21.4.17 PWM 死区控制寄存器 A (TMR4_PDARn)

偏移地址 (Offset) : 0x0084 (TMR4_PDARu) , 0x008C (TMR4_PDARv) , 0x0094 (TMR4_PDARw) , 0x009C (TMR4_PDARx)

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	PDAR															
Access	RW															

位/位域	标记	位名	功能描述	读写
15:0	PDAR	死区初始值	死区计数初始值	RW

21.4.18 PWM 死区控制寄存器 B (TMR4_PDBRn)

偏移地址 (Offset) : 0x0086 (TMR4_PDBRu) , 0x008E (TMR4_PDBRv) , 0x0096 (TMR4_PDBRw) , 0x009E (TMR4_PDBRx)

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	PDBR															
Access	RW															

位/位域	标记	位名	功能描述	读写
15:0	PDBR	死区初始值	死区计数初始值	RW

21.4.19 重载控制状态寄存器 (TMR4_RCSR)

Offset	Bit Position																																					
0x00F4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	Reserved												RTSX	RTEX	RTICX	RO RTIFX	RW RTSW	RW RTEW	RW RTICW	RO RTIFW	RW RTSV	RW RTEV	RW RTICV	RO RTIFV	RW RTSU	RW RTEU	RW RTICU	RO RTIFU	RW RTIDX	RW RTIDW	RW RTIDV	RW RTIDU						
Access													RW	RW	RW	RO	RW	RW	RW	RO	RW	RW	RW	RO	RW	RW	RW	RO	RW	RW	RW	RO	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:20	Reserved	保留	-	-
19	RTSX	重载计数器停止 X	写 1 停止重载计数器 X 并清除 RTIFX，写 0 无效；读出恒为 0	RW
18	RTEX	重载计数器启动 X	写 1 启动重载计数器 X，写 0 无效 读取： ● 0b0：重载计数器 X 已停止 ● 0b1：重载计数器 X 已启动	RW
17	RTICX	清零计数匹配状态 X	写 1 清零 RTIFX 标志位，写 0 无效；读出恒为 0	RW
16	RTIFX	计数匹配状态 X	● 0b0：重载计数器计数值与 PFSRx 发生未比较匹配 ● 0b1：重载计数器计数值与 PFSRx 发生比较匹配	RO
15	RTSW	重载计数器停止 W	写 1 停止重载计数器 W 并清除 RTIFW，写 0 无效；读出恒为 0	RW
14	RTEW	重载计数器启动 W	写 1 启动重载计数器 W，写 0 无效 读取： ● 0b0：重载计数器 W 已停止 ● 0b1：重载计数器 W 已启动	RW
13	RTICW	清零计数匹配状态 W	写 1 清零 RTIFW 标志位，写 0 无效；读出恒为 0	RW
12	RTIFW	计数匹配状态 W	● 0b0：重载计数器计数值与 PFSRw 发生未比较匹配 ● 0b1：重载计数器计数值与 PFSRw 发生比较匹配	RO
11	RTSV	重载计数器停止 V	写 1 停止重载计数器 V 并清除 RTIFV，写 0 无效；读出恒为 0	RW
10	RTEV	重载计数器启动 V	写 1 启动重载计数器 V，写 0 无效 读取： ● 0b0：重载计数器 V 已停止 ● 0b1：重载计数器 V 已启动	RW
9	RTICV	清零计数匹配状态 V	写 1 清零 RTIFV 标志位，写 0 无效；读出恒为 0	RW
8	RTIFV	计数匹配状态 V	● 0b0：重载计数器计数值与 PFSRv 发生未比较匹配 ● 0b1：重载计数器计数值与 PFSRv 发生比较匹配	RO

位/位域	标记	位名	功能描述	读写
7	RTSU	重载计数器停止 U	写 1 停止重载计数器 U 并清除 RTIFU，写 0 无效；读出恒为 0	RW
6	RTEU	重载计数器启动 U	写 1 启动重载计数器 U，写 0 无效 读取： ● 0b0：重载计数器 U 已停止 ● 0b1：重载计数器 U 已启动	RW
5	RTICU	清零计数匹配状态 U	写 1 清零 RTIFU 标志位，写 0 无效；读出恒为 0	RW
4	RTIFU	计数匹配状态 U	● 0b0：重载计数器计数值与 PFSRu 发生未比较匹配 ● 0b1：重载计数器计数值与 PFSRu 发生比较匹配	RO
3	RTIDX	重载中断屏蔽 X	● 0b0：重载功能有效时，重载中断 X 输出有效 ● 0b1：重载功能有效时，重载中断 X 输出无效	RW
2	RTIDW	重载中断屏蔽 W	● 0b0：重载功能有效时，重载中断 W 输出有效 ● 0b1：重载功能有效时，重载中断 W 输出无效	RW
1	RTIDV	重载中断屏蔽 V	● 0b0：重载功能有效时，重载中断 V 输出有效 ● 0b1：重载功能有效时，重载中断 V 输出无效	RW
0	RTIDU	重载中断屏蔽 U	● 0b0：重载功能有效时，重载中断 U 输出有效 ● 0b1：重载功能有效时，重载中断 U 输出无效	RW

21.4.20 专用中断控制寄存器（TMR4_SCIR）

Offset	Bit Position															
0x00F8	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFF00															
Name	Reserved								ITEN7	ITEN6	ITEN5	ITEN4	ITEN3	ITEN2	ITEN1	ITEN0
Access									RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
15:8	Reserved	保留	-	-
7	ITEN7	专用匹配中断 7 使能	● 0b0：禁止 ● 0b1：使能	RW
6	ITEN6	专用匹配中断 6 使能	● 0b0：禁止 ● 0b1：使能	RW
5	ITEN5	专用匹配中断 5 使能	● 0b0：禁止 ● 0b1：使能	RW
4	ITEN4	专用匹配中断 4 使能	● 0b0：禁止 ● 0b1：使能	RW
3	ITEN3	专用匹配中断 3 使能	● 0b0：禁止 ● 0b1：使能	RW
2	ITEN2	专用匹配中断 2 使能	● 0b0：禁止 ● 0b1：使能	RW
1	ITEN1	专用匹配中断 1 使能	● 0b0：禁止 ● 0b1：使能	RW
0	ITEN0	专用匹配中断 0 使能	● 0b0：禁止 ● 0b1：使能	RW

21.4.21 专用标志控制寄存器（TMR4_SCFR）

Offset	Bit Position															
0x00FC	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFF00															
Name	Reserved								SF7	SF6	SF5	SF4	SF3	SF2	SF1	SF0
Access									RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
15:8	Reserved	保留	-	-
7	SF7	专用计数匹配标志位 7	● 0b0：未发生专用计数比较匹配事件 7 ● 0b1：发生专用计数比较匹配事件 7	RW
6	SF6	专用计数匹配标志位 6	● 0b0：未发生专用计数比较匹配事件 6 ● 0b1：发生专用计数比较匹配事件 6	RW
5	SF5	专用计数匹配标志位 5	● 0b0：未发生专用计数比较匹配事件 5 ● 0b1：发生专用计数比较匹配事件 5	RW
4	SF4	专用计数匹配标志位 4	● 0b0：未发生专用计数比较匹配事件 4 ● 0b1：发生专用计数比较匹配事件 4	RW
3	SF3	专用计数匹配标志位 3	● 0b0：未发生专用计数比较匹配事件 3 ● 0b1：发生专用计数比较匹配事件 3	RW
2	SF2	专用计数匹配标志位 2	● 0b0：未发生专用计数比较匹配事件 2 ● 0b1：发生专用计数比较匹配事件 2	RW
1	SF1	专用计数匹配标志位 1	● 0b0：未发生专用计数比较匹配事件 1 ● 0b1：发生专用计数比较匹配事件 1	RW
0	SF0	专用计数匹配标志位 0	● 0b0：未发生专用计数比较匹配事件 0 ● 0b1：发生专用计数比较匹配事件 0	RW

22 紧急刹车模块 (EMB)

22.1 简介

紧急刹车模块是在满足一定条件时产生控制事件输出给定时器，使得定时器停止或更改向外部电机输出 PWM 信号的功能模块，下列要因用于产生控制事件：

- 外部端口输入电平变化
- PWM 输出端口电平发生同相（同高或同低）
- 电压比较器比较结果
- 系统错误发生
- 写寄存器软件控制

22.2 功能说明

22.2.1 EMB 通道功能框图

EMB 用于向具有 PWM 功能的定时器模块（Timer4，Timer6）在满足一定条件时输出一个控制事件信号，通知定时器模块关闭当前 PWM 输出。EMB 模块具有 4 个 Group，其中 Group0 用于控制 Timer6，由 Timer6 寄存器设定选择使用；Group1~Group3 用于控制 Timer4，分别对应 Timer4 的 3 个单元。

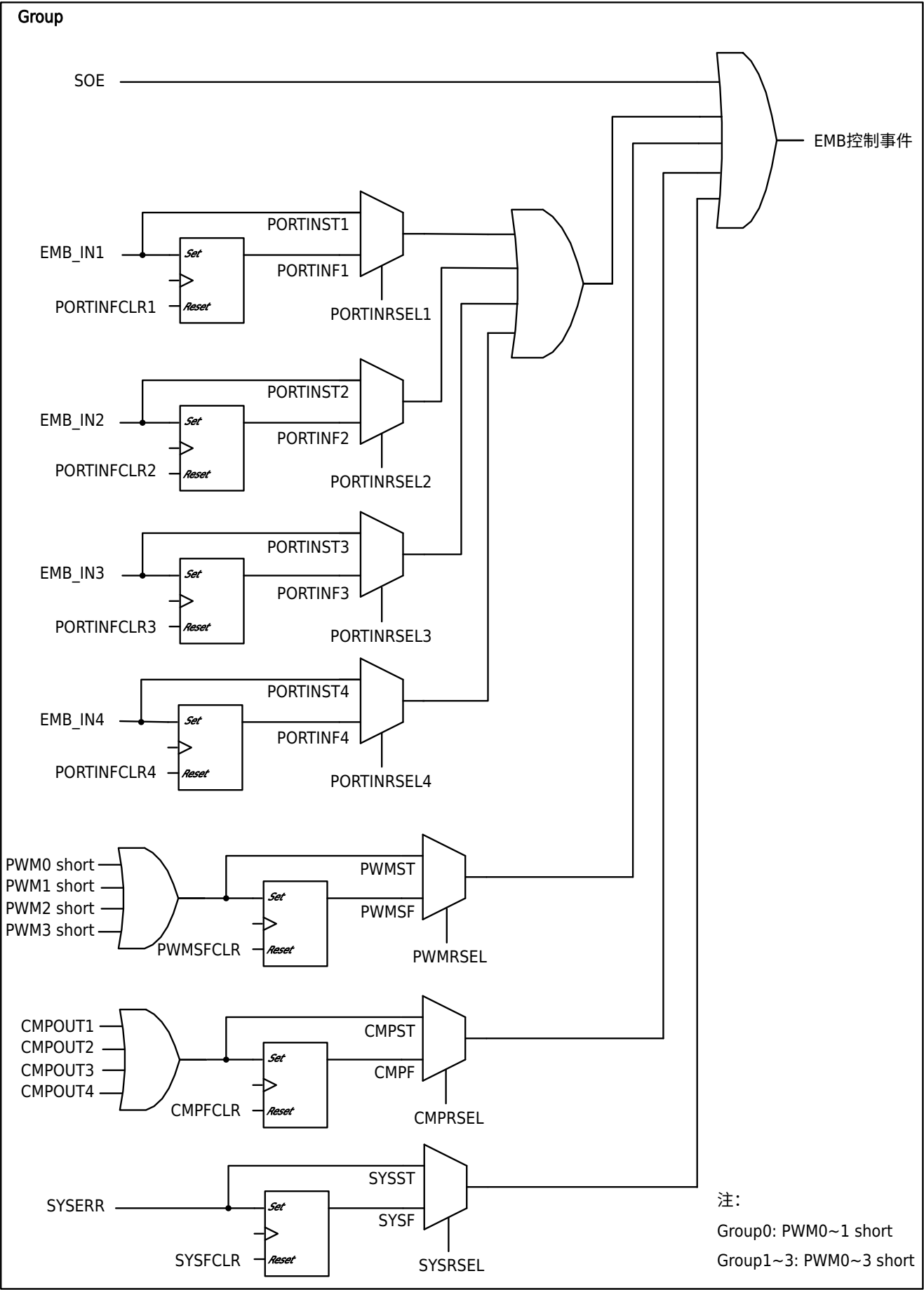


图 22-1 EMB 通道功能框图

22.2.2 EMB 结构框图

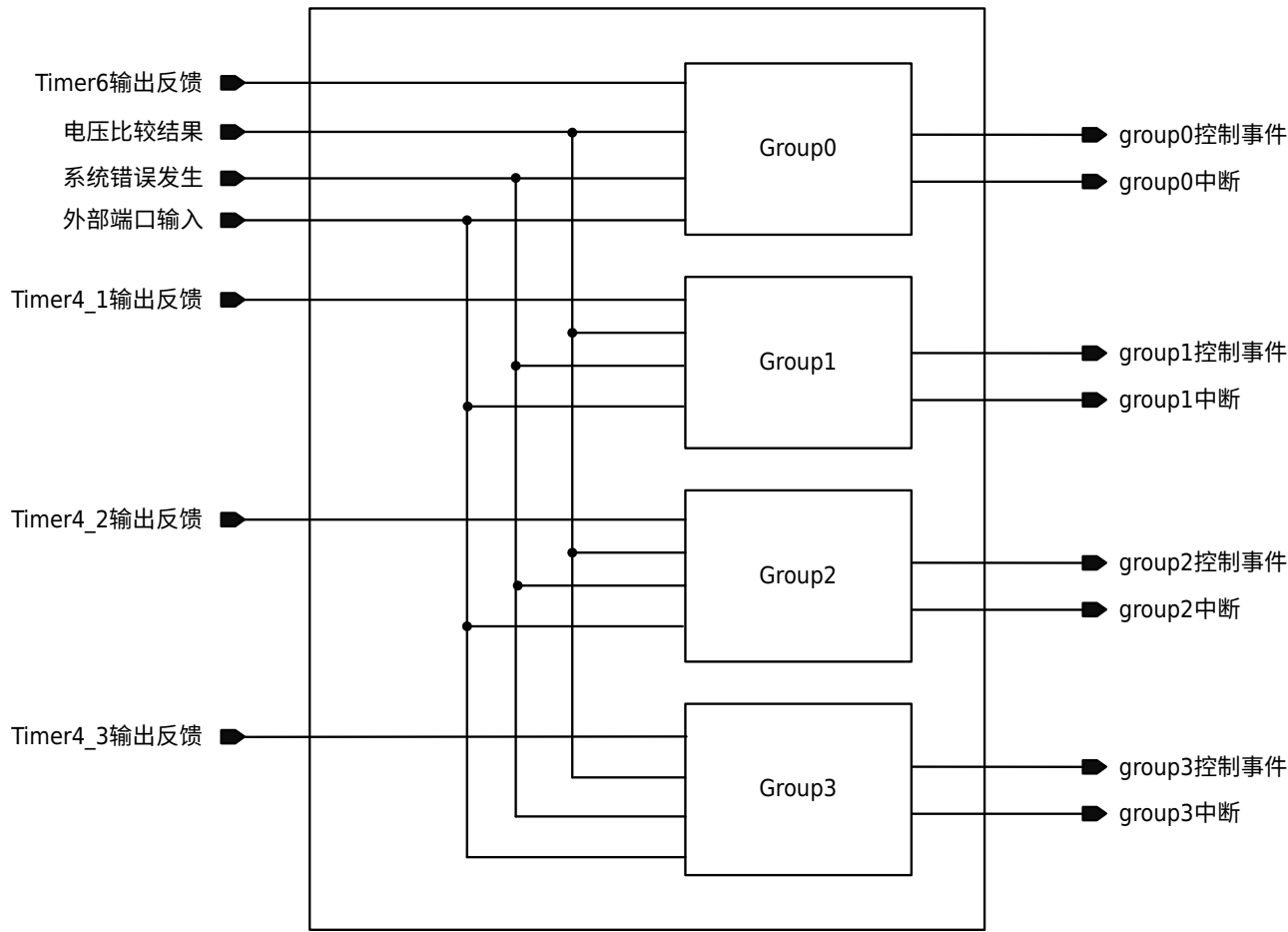


图 22-2 EMB 结构框图

22.2.3 外部端口输入电平变化时控制 PWM 信号输出

EMB 共有 4 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 4 个外部端口中的 1 个或多个有效，端口分配如下表所示。

表 22-1 EMB 端口分配

端口	PIN 名
EMB_IN1	PA11, PB2
EMB_IN2	PA6, PB12, PE15
EMB_IN3	PA7, PC2
EMB_IN4	PB15, PH2, PB9

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.PORTINENy (y=1~4) 置为有效，同时通过 EMB_CTL1.INVSELY (y=1~4) 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.NFENy (y=1~4) 和 EMB_CTL2.NFSELY (y=1~4) 使能滤波功能和设置滤波时钟。滤波器根据滤波时钟对输入信号进行采样后，采用三次比较一致的方式进行滤波，即当滤波时钟采样到端口上 3 次一致的电平时，该电平被当作有效电平传送到模块内部；小于 3 次一致的电平会被当作外部干扰滤掉，不传送到模块内部。

当端口上产生符合条件的有效电平时，EMB 状态寄存器的端口输入状态 EMB_STAT.PORTINSTy (y=1~4) 被置位，同时端口输入控制标志位 EMB_STAT.PORTINFy (y=1~4) 被置位。当中断许可寄存器 EMB_INTEN.PORTINEN=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_RLSSEL.PORTINRSELy (y=1~4) 选择由 EMB_STAT.PORTINSTy (y=1~4) 或者 EMB_STAT.PORTINFy (y=1~4) 向 Timer6 和 Timer4 输出。Timer6 和 Timer4 在接到控制事件后，可根据寄存器设定将输出端口置为高电平，低电平或高阻态。释放控制时，如果 EMB_RLSSEL.PORTINRSELy (y=1~4) =1，当端口输入电平转为无效时，EMB_STAT.PORTINSTy (y=1~4) 将自动清零，同时控制事件将立即释放。如果 EMB_RLSSEL.PORTINRSELy (y=1~4) =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_STATCLR.PORTINFCLR 使 EMB_STAT.PORTINF 清 0，将控制事件释放。

22.2.4 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.PWMSENy (Group=0, y=0~1 或 Group=1~3, y=0~3) 置为有效，同时通过 EMB 控制寄存器 EMB_CTL2.PWMLVy (Group=0, y=0~1 或 Group=1~3, y=0~3) 选择需要监控的有效电平。

完成设置后，EMB 监控 Timer6 和 Timer4 的互补 PWM 输出信号，当输出信号出现同高或同低情况时，EMB 状态寄存器的 PWM 输出状态 EMB_STAT.PWMST 被置位，同时 PWM 输出同相位控制标志位 EMB_STAT.PWMSF 被置位。当中断许可寄存器 EMB_INTEN.PWMEN=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_RLSSEL.PWMRSEL 选择由 EMB_STAT.PWMST 或者 EMB_STAT.PWMSF 向 Timer6 和 Timer4 输出。Timer6 和 Timer4 在接到控制事件后可根据寄存器设定将输出端口置为高电平、低电平或高阻态。

释放控制时，如果 EMB_RLSSEL.PWMRSEL=1，当 PWM 输出同高或同低状态解除，EMB_STAT.PWMST 将自动清零，同时控制事件将立即释放。如果 EMB_RLSSEL.PWMST=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_STATCLR.PWMSFCLR 使 EMB_STAT.PWMSF 清 0 后，将控制事件释放。

Group0 可用于监控 Timer6 的互补 PWM 输出信号，Group1 用于监控 Timer4_1 的互补 PWM 输出信号，Group2 用于监控 Timer4_2 的互补 PWM 输出信号，Group3 用于监控 Timer4_3 的互补 PWM 输出信号。

表 22-2 EMB Group 对照表

端口名	功能	对应 Group	EMB_CTL1 控制位	EMB_CTL2 控制位
TMR6_<t>_PWMA (<t>为 1~2)	Timer6 的互补 PWM 输出信号	Group0	PWMSEN1~PWMSEN0	PWMLV1~PWMLV0
TMR6_<t>_PWMB (<t>为 1~2)				
TMR4_1_OXH	Timer4_1 的互补 PWM 输出信号	Group1	PWMSEN3	PWMLV3
TMR4_1_OXL			PWMSEN2	PWMLV2
TMR4_1_OUH				
TMR4_1_OUL			PWMSEN1	PWMLV1
TMR4_1_OVH				
TMR4_1_OVL			PWMSEN0	PWMLV0
TMR4_1_OWH				
TMR4_1_OWL				

端口名	功能	对应 Group	EMB_CTL1 控制位	EMB_CTL2 控制位
TMR4_2_OXH	Timer4_2 的互补 PWM 输出信号	Group2	PWMSEN3	PWMLV3
TMR4_2_OXL				
TMR4_2_OUH			PWMSEN2	PWMLV2
TMR4_2_OUL				
TMR4_2_OVH			PWMSEN1	PWMLV1
TMR4_2_OVL				
TMR4_2_OWH			PWMSEN0	PWMLV0
TMR4_2_OWL				
TMR4_3_OXH	Timer4_3 的互补 PWM 输出信号	Group3	PWMSEN3	PWMLV3
TMR4_3_OXL				
TMR4_3_OUH			PWMSEN2	PWMLV2
TMR4_3_OUL				
TMR4_3_OVH			PWMSEN1	PWMLV1
TMR4_3_OVL				
TMR4_3_OWH			PWMSEN0	PWMLV0
TMR4_3_OWL				

22.2.5 根据电压比较器比较结果停止 PWM 信号输出

EMB 的每个 Group 能够根据电压比较器的 4 组比较结果向 Timer6 和 Timer4 发送控制事件信号。电压比较器输出结果的设定请参见《电压比较器 (CMP)》章节相关描述。

使用时首先将使能位，即 EMB 控制寄存器 EMB_CTL1.CMPENy (y=1~4) (Group0~3) 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_STAT.CMPST 被置位，同时 EMB 电压比较器控制标志位 EMB_STAT.CMPF 被置位。当中断许可寄存器 EMB_INTEN.CMPINTEN=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_RLSSEL.CMPRSEL 选择由 EMB_STAT.CMPST 或者 EMB_STAT.CMPF 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_RLSSEL.CMPRSEL=1，当电压比较器结果转为无效时，EMB_STAT.CMPST 将自动清零，同时控制事件将立即释放。如果 EMB_RLSSEL.CMPRSEL=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.CMPFCLR 使 EMB_STAT.CMPF 清 0 后，将控制事件释放。

22.2.6 系统错误发生时停止 PWM 信号输出

EMB 的每个 Group 能够在系统错误发生时向 Timer6 和 Timer4 发送通知信号。系统错误信号来自外部振荡器停止、SRAM 奇偶校验错误以及 ECC 校验错误、Cortex-M4 锁定信号和 PVD 检测。

使用时首先将使能位，即 EMB 控制寄存器 EMB_CTL1.SYSEN (Group0~3) 置为有效，设置 EMB_CTL1.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_STAT.SYSST 被置位，同时 EMB 系统错误控制标志位 EMB_STAT.SYSF 被置位。当中断许可寄存器 EMB_INTEN.SYSINTEN=1 时还将产生中断。

EMB 的控制事件通过 EMB 释放方式选择寄存器 EMB_RLSSEL.SYSRSEL 选择由 EMB_STAT.SYSST 或者 EMB_STAT.SYSF 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_RLSSEL.SYSRSEL=1，当系统错误信号转为无效时，EMB_STAT.SYSST 将自动清零，同时控制事件将立即释放。如果 EMB_RLSSEL.SYSRSEL=0，当系统错误信号转为无效后，需要写 EMB 状态复位寄存器 EMB_STATCLR.SYSFCLR 使 EMB_STAT.SYSF 清 0 后，将控制事件释放。

22.2.7 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 Timer6 和 Timer4 发送控制信号，软件控制 PWM 输出时不会产生中断请求。

22.3 寄存器

22.3.1 寄存器总表

- EMB Group0 基地址: 0x40017C00
- EMB Group1 基地址: 0x40017C20
- EMB Group2 基地址: 0x40017C40
- EMB Group3 基地址: 0x40017C60

表 22-3 紧急刹车模块 (EMB) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	EMB_CTL1	EMB 控制寄存器 1 (Group0)	32
0x00	EMB_CTL1	EMB 控制寄存器 1 (Group1~3)	32
0x04	EMB_CTL2	EMB 控制寄存器 2 (Group0)	32
0x04	EMB_CTL2	EMB 控制寄存器 2 (Group1~3)	32
0x08	EMB_SOE	EMB 软件输出使能控制寄存器	32
0x0C	EMB_STAT	EMB 状态寄存器	32
0x10	EMB_STATCLR	EMB 状态复位寄存器	32
0x14	EMB_INTEN	EMB 中断许可寄存器	32
0x18	EMB_RLSSEL	EMB 控制 PWM 输出释放方式选择寄存器	32

22.3.2 EMB 控制寄存器 1 (EMB_CTL1) (Group0)

该寄存器为单次写入寄存器，即复位之后仅能够写入一次。

Offset	Bit Position																															
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	PVDEN	LOCKUPEN	SRAMPYERREN	SRAMECCERREN	OSCSTPEN	Reserved	INVSEL4	INVSEL3	INVSEL2	INVSEL1	Reserved	PORTINEN4	PORTINEN3	PORTINEN2	PORTINEN1	Reserved										PWMSEN1	PWMSEN0	SYSEN	CMPEN4	CMPEN3	CMPEN2	CMPEN1
Access	RW	RW	RW	RW	RW		RW	RW	RW	RW		RW	RW	RW	RW											RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31	PVDEN	PVD 检测控制使能	0b0: PVD 检测结果输出控制无效 0b1: PVD 检测结果输出控制有效	RW
30	LOCKUPEN	Cortex-M4 Lockup 控制使能	0b0: Lockup 发生时输出控制无效 0b1: Lockup 发生时输出控制有效	RW
29	SRAMPYERREN	SRAM 奇偶校验错误控制使能	0b0: SRAM 奇偶校验错误发生时输出控制无效 0b1: SRAM 奇偶校验错误发生时输出控制有效	RW
28	SRAMECCERREN	SRAM ECC 校验错误控制使能	0b0: SRAM ECC 校验错误发生时输出控制无效 0b1: SRAM ECC 校验错误发生时输出控制有效	RW
27	OSCSTPEN	振荡器停止控制使能	0b0: 振荡器停止发生时输出控制无效 0b1: 振荡器停止发生时输出控制有效	RW
26	Reserved	保留	-	-
25	INVSEL4	端口输入 4 有效电平选择	0b0: 高电平有效 0b1: 低电平有效	RW
24	INVSEL3	端口输入 3 有效电平选择	0b0: 高电平有效 0b1: 低电平有效	RW
23	INVSEL2	端口输入 2 有效电平选择	0b0: 高电平有效 0b1: 低电平有效	RW
22	INVSEL1	端口输入 1 有效电平选择	0b0: 高电平有效 0b1: 低电平有效	RW
21:20	Reserved	保留	-	-
19	PORTINEN4	端口输入 4 控制使能	0b0: 端口输入控制无效 0b1: 端口输入控制有效	RW
18	PORTINEN3	端口输入 3 控制使能	0b0: 端口输入控制无效 0b1: 端口输入控制有效	RW

位/位域	标记	位名	功能描述	读写
17	PORTINEN2	端口输入 2 控制使能	0b0: 端口输入控制无效 0b1: 端口输入控制有效	RW
16	PORTINEN1	端口输入 1 控制使能	0b0: 端口输入控制无效 0b1: 端口输入控制有效	RW
15:7	<i>Reserved</i>	<i>保留</i>	-	-
6	PWMSEN1	TMR6_2_PWMA/B 短路输出控制使能	0b0: 短路时输出控制无效 0b1: 短路时输出控制有效	RW
5	PWMSEN0	TMR6_1_PWMA/B 短路输出控制使能	0b0: 短路时输出控制无效 0b1: 短路时输出控制有效	RW
4	SYSEN	系统错误控制使能	0b0: 系统错误发生时输出控制无效 0b1: 系统错误发生时输出控制有效	RW
3	CMPEN4	CMP4 电压比较器比较结果控制使能	0b0: 电压比较器比较结果输出控制无效 0b1: 电压比较器比较结果输出控制有效	RW
2	CMPEN3	CMP3 电压比较器比较结果控制使能	0b0: 电压比较器比较结果输出控制无效 0b1: 电压比较器比较结果输出控制有效	RW
1	CMPEN2	CMP2 电压比较器比较结果控制使能	0b0: 电压比较器比较结果输出控制无效 0b1: 电压比较器比较结果输出控制有效	RW
0	CMPEN1	CMP1 电压比较器比较结果控制使能	0b0: 电压比较器比较结果输出控制无效 0b1: 电压比较器比较结果输出控制有效	RW

22.3.3 EMB 控制寄存器 1 (EMB_CTL1) (Group1~3)

该寄存器为单次写入寄存器，即复位之后仅能够写入一次。

Offset	Bit Position																																	
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	PVDEN	LOCKUPEN	SRAMPYERREN	SRAMECCERREN	OSCSTPEN	Reserved	INVSEL4	INVSEL3	INVSEL2	INVSEL1	Reserved	PORTINEN4	PORTINEN3	PORTINEN2	PORTINEN1	Reserved								PWMSEN3	PWMSEN2	PWMSEN1	PWMSEN0	SYSEN	CMPEN4	CMPEN3	CMPEN2	CMPEN1		
Access	RW	RW	RW	RW	RW		RW	RW	RW	RW		RW	RW	RW	RW									RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31	PVDEN	PVD 检测控制使能	0b0: PVD 检测结果输出控制无效 0b1: PVD 检测结果输出控制有效	RW
30	LOCKUPEN	Cortex-M4 Lockup 控制使能	0b0: Lockup 发生时输出控制无效 0b1: Lockup 发生时输出控制有效	RW
29	SRAMPYERREN	SRAM 奇偶校验错误控制使能	0b0: SRAM 奇偶校验错误发生时输出控制无效 0b1: SRAM 奇偶校验错误发生时输出控制有效	RW
28	SRAMECCERREN	SRAM ECC 校验错误控制使能	0b0: SRAM ECC 校验错误发生时输出控制无效 0b1: SRAM ECC 校验错误发生时输出控制有效	RW
27	OSCSTPEN	振荡器停止控制使能	0b0: 振荡器停止发生时输出控制无效 0b1: 振荡器停止发生时输出控制有效	RW
26	Reserved	保留	-	-
25	INVSEL4	端口输入 4 有效电平选择	0b0: 高电平有效 0b1: 低电平有效	RW
24	INVSEL3	端口输入 3 有效电平选择	0b0: 高电平有效 0b1: 低电平有效	RW
23	INVSEL2	端口输入 2 有效电平选择	0b0: 高电平有效 0b1: 低电平有效	RW
22	INVSEL1	端口输入 1 有效电平选择	0b0: 高电平有效 0b1: 低电平有效	RW
21:20	Reserved	保留	-	-
19	PORTINEN4	端口输入 4 控制使能	0b0: 端口输入控制无效 0b1: 端口输入控制有效	RW
18	PORTINEN3	端口输入 3 控制使能	0b0: 端口输入控制无效 0b1: 端口输入控制有效	RW

位/位域	标记	位名	功能描述	读写
17	PORTINEN2	端口输入 2 控制使能	0b0: 端口输入控制无效 0b1: 端口输入控制有效	RW
16	PORTINEN1	端口输入 1 控制使能	0b0: 端口输入控制无效 0b1: 端口输入控制有效	RW
15:9	Reserved	保留	-	-
8	PWMSEN3	TMR4 <t>_OXH/L 短路输出控制使能	0b0: 短路时输出控制无效 0b1: 短路时输出控制有效	RW
7	PWMSEN2	TMR4 <t>_OUH/L 短路输出控制使能	0b0: 短路时输出控制无效 0b1: 短路时输出控制有效	RW
6	PWMSEN1	TMR4 <t>_OVH/L 短路输出控制使能	0b0: 短路时输出控制无效 0b1: 短路时输出控制有效	RW
5	PWMSEN0	TMR4 <t>_OWH/L 短路输出控制使能	0b0: 短路时输出控制无效 0b1: 短路时输出控制有效	RW
4	SYSEN	系统错误控制使能	0b0: 系统错误发生时输出控制无效 0b1: 系统错误发生时输出控制有效	RW
3	CMPEN4	CMP4 电压比较器比较结果控制使能	0b0: 电压比较器比较结果输出控制无效 0b1: 电压比较器比较结果输出控制有效	RW
2	CMPEN3	CMP3 电压比较器比较结果控制使能	0b0: 电压比较器比较结果输出控制无效 0b1: 电压比较器比较结果输出控制有效	RW
1	CMPEN2	CMP2 电压比较器比较结果控制使能	0b0: 电压比较器比较结果输出控制无效 0b1: 电压比较器比较结果输出控制有效	RW
0	CMPEN1	CMP1 电压比较器比较结果控制使能	0b0: 电压比较器比较结果输出控制无效 0b1: 电压比较器比较结果输出控制有效	RW



说明

- Group1: <t>为 1
- Group2: <t>为 2
- Group3: <t>为 3

22.3.4 EMB 控制寄存器 2 (EMB_CTL2) (Group0)

该寄存器为单次写入寄存器，即复位之后仅能够写入一次。

Offset	Bit Position																															
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved				NFEN4	NFSEL4		NFEN3	NFSEL3		NFEN2	NFSEL2		NFEN1	NFSEL1		Reserved										PWMLV1		PWMLV0			
Access					RW	RW		RW	RW		RW	RW		RW	RW												RW		RW			

位/位域	标记	位名	功能描述	读写
31:28	Reserved	保留	-	-
27	NFEN4	端口输入 4 数字滤波器使能	0b0: 滤波器无效 0b1: 滤波器有效	RW
26:25	NFSEL4	端口输入 4 数字滤波器滤波时钟选择	0b00: 使用 PCLK0 滤波 0b01: 使用 PCLK0 的 8 分频滤波 0b10: 使用 PCLK0 的 32 分频滤波 0b11: 使用 PCLK0 的 128 分频滤波	RW
24	NFEN3	端口输入 3 数字滤波器使能	0b0: 滤波器无效 0b1: 滤波器有效	RW
23:22	NFSEL3	端口输入 3 数字滤波器滤波时钟选择	0b00: 使用 PCLK0 滤波 0b01: 使用 PCLK0 的 8 分频滤波 0b10: 使用 PCLK0 的 32 分频滤波 0b11: 使用 PCLK0 的 128 分频滤波	RW
21	NFEN2	端口输入 2 数字滤波器使能	0b0: 滤波器无效 0b1: 滤波器有效	RW
20:19	NFSEL2	端口输入 2 数字滤波器滤波时钟选择	0b00: 使用 PCLK0 滤波 0b01: 使用 PCLK0 的 8 分频滤波 0b10: 使用 PCLK0 的 32 分频滤波 0b11: 使用 PCLK0 的 128 分频滤波	RW
18	NFEN1	端口输入 1 数字滤波器使能	0b0: 滤波器无效 0b1: 滤波器有效	RW
17:16	NFSEL1	端口输入 1 数字滤波器滤波时钟选择	0b00: 使用 PCLK0 滤波 0b01: 使用 PCLK0 的 8 分频滤波 0b10: 使用 PCLK0 的 32 分频滤波 0b11: 使用 PCLK0 的 128 分频滤波	RW
15:2	Reserved	保留	-	-
1	PWMLV1	TMR6_2_PWMA/B 输出有效电平选择	0b0: 低电平为有效电平 0b1: 高电平为有效电平	RW
0	PWMLV0	TMR6_1_PWMA/B 输出有效电平选择	0b0: 低电平为有效电平 0b1: 高电平为有效电平	RW

22.3.5 EMB 控制寄存器 2 (EMB_CTL2) (Group1~3)

该寄存器为单次写入寄存器，即复位之后仅能够写入一次。

Offset	Bit Position																															
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved				NFEN4	NFSEL4		NFEN3	NFSEL3		NFEN2	NFSEL2		NFEN1	NFSEL1		Reserved										PWMLV3	PWMLV2	PWMLV1	PWMLV0		
Access					RW	RW		RW	RW		RW	RW		RW	RW												RW	RW	RW	RW		

位/位域	标记	位名	功能描述	读写
31:28	Reserved	保留	-	-
27	NFEN4	端口输入 4 数字滤波器使能	0b0: 滤波器无效 0b1: 滤波器有效	RW
26:25	NFSEL4	端口输入 4 数字滤波器滤波时钟选择	0b00: 使用 PCLK0 滤波 0b01: 使用 PCLK0 的 8 分频滤波 0b10: 使用 PCLK0 的 32 分频滤波 0b11: 使用 PCLK0 的 128 分频滤波	RW
24	NFEN3	端口输入 3 数字滤波器使能	0b0: 滤波器无效 0b1: 滤波器有效	RW
23:22	NFSEL3	端口输入 3 数字滤波器滤波时钟选择	0b00: 使用 PCLK0 滤波 0b01: 使用 PCLK0 的 8 分频滤波 0b10: 使用 PCLK0 的 32 分频滤波 0b11: 使用 PCLK0 的 128 分频滤波	RW
21	NFEN2	端口输入 2 数字滤波器使能	0b0: 滤波器无效 0b1: 滤波器有效	RW
20:19	NFSEL2	端口输入 2 数字滤波器滤波时钟选择	0b00: 使用 PCLK0 滤波 0b01: 使用 PCLK0 的 8 分频滤波 0b10: 使用 PCLK0 的 32 分频滤波 0b11: 使用 PCLK0 的 128 分频滤波	RW
18	NFEN1	端口输入 1 数字滤波器使能	0b0: 滤波器无效 0b1: 滤波器有效	RW
17:16	NFSEL1	端口输入 1 数字滤波器滤波时钟选择	0b00: 使用 PCLK0 滤波 0b01: 使用 PCLK0 的 8 分频滤波 0b10: 使用 PCLK0 的 32 分频滤波 0b11: 使用 PCLK0 的 128 分频滤波	RW
15:4	Reserved	保留	-	-
3	PWMLV3	TMR4_<t>_OXH/L 输出有效电平选择	0b0: 低电平为有效电平 0b1: 高电平为有效电平	RW
2	PWMLV2	TMR4_<t>_OUH/L 输出有效电平选择	0b0: 低电平为有效电平 0b1: 高电平为有效电平	RW

位/位域	标记	位名	功能描述	读写
1	PWMLV1	TMR4_<t>_OVH/L 输出有效电平选择	0b0: 低电平为有效电平 0b1: 高电平为有效电平	RW
0	PWMLV0	TMR4_<t>_OWH/L 输出有效电平选择	0b0: 低电平为有效电平 0b1: 高电平为有效电平	RW

 **说明**

- Group1: <t>为 1
- Group2: <t>为 2
- Group3: <t>为 3

22.3.6 EMB 软件输出使能控制寄存器 (EMB_SOE)

Offset	Bit Position																															
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															SOE
Access																																RW

位/位域	标记	位名	功能描述	读写
31:1	Reserved	保留	-	-
0	SOE	软件控制输出	0b0: PWM 正常输出 0b1: PWM 停止输出	RW

22.3.7 EMB 状态寄存器 (EMB_STAT)

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved														PORTINST4	PORTINST3	PORTINST2	PORTINST1	Reserved	PORTINF4	PORTINF3	PORTINF2	PORTINF1	SYSST	CMPST	PWMST	Reserved	SYSF	CMPF	PWMSF	Reserved	
Access															RO	RO	RO	RO		RO	RO	RO	RO	RO	RO	RO		RO	RO	RO		

位/位域	标记	位名	功能描述	读写
31:18	Reserved	保留	-	-
17	PORTINST4	端口输入 4 控制状态	0b0: 端口输入控制处于无效电平状态 0b1: 端口输入控制处于有效电平状态	RO
16	PORTINST3	端口输入 3 控制状态	0b0: 端口输入控制处于无效电平状态 0b1: 端口输入控制处于有效电平状态	RO
15	PORTINST2	端口输入 2 控制状态	0b0: 端口输入控制处于无效电平状态 0b1: 端口输入控制处于有效电平状态	RO
14	PORTINST1	端口输入 1 控制状态	0b0: 端口输入控制处于无效电平状态 0b1: 端口输入控制处于有效电平状态	RO
13:12	Reserved	保留	-	-
11	PORTINF4	端口输入 4 控制标志位	EMB_STAT.PORTINST4 为 1 时置位 EMB_STATCLR.PORTINFCLR4 写 1 时清 0	RO
10	PORTINF3	端口输入 3 控制标志位	EMB_STAT.PORTINST3 为 1 时置位 EMB_STATCLR.PORTINFCLR3 写 1 时清 0	RO
9	PORTINF2	端口输入 2 控制标志位	EMB_STAT.PORTINST2 为 1 时置位 EMB_STATCLR.PORTINFCLR2 写 1 时清 0	RO
8	PORTINF1	端口输入 1 控制标志位	EMB_STAT.PORTINST1 为 1 时置位 EMB_STATCLR.PORTINFCLR1 写 1 时清 0	RO
7	SYSST	系统错误状态	0b0: 系统错误未发生 0b1: 系统错误发生	RO
6	CMPST	电压比较器状态	0b0: 电压比较器比较结果标志位未被置位 0b1: 电压比较器比较结果标志位被置位	RO
5	PWMST	PWM 输出状态	0b0: 没有发生 PWM 输出同相 0b1: 发生 PWM 输出同相	RO
4	Reserved	保留	-	-
3	SYSF	系统错误标志位	EMB_STAT.SYSST 为 1 时置位 EMB_STATCLR.SYSFCLR 写 1 时清 0	RO
2	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	RO
1	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	RO

位/位域	标记	位名	功能描述	读写
0	Reserved	保留	-	-

22.3.8 EMB 状态复位寄存器 (EMB_STATCLR)

Offset	Bit Position																															
0x10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																				PORTINFCLR4	PORTINFCLR3	PORTINFCLR2	PORTINFCLR1	Reserved				SYSFCLR	CMPFCLR	PWMSFCLR	Reserved
Access																					WO	WO	WO	WO								

位/位域	标记	位名	功能描述	读写
31:12	Reserved	保留	-	-
11	PORTINFCLR4	复位 EMB_STAT.PORTINF4	写 1 将 EMB_STAT.PORTINF4 清 0 (当 EMB_STAT.PORTINST4=0 时)，写 0 无效	WO
10	PORTINFCLR3	复位 EMB_STAT.PORTINF3	写 1 将 EMB_STAT.PORTINF3 清 0 (当 EMB_STAT.PORTINST3=0 时)，写 0 无效	WO
9	PORTINFCLR2	复位 EMB_STAT.PORTINF2	写 1 将 EMB_STAT.PORTINF2 清 0 (当 EMB_STAT.PORTINST2=0 时)，写 0 无效	WO
8	PORTINFCLR1	复位 EMB_STAT.PORTINF1	写 1 将 EMB_STAT.PORTINF1 清 0 (当 EMB_STAT.PORTINST1=0 时)，写 0 无效	WO
7:4	Reserved	保留	-	-
3	SYSFCLR	复位 EMB_STAT.SYSF	写 1 将 EMB_STAT.SYSF 清 0 (当 EMB_STAT.SYSST=0 时)，写 0 无效	WO
2	CMPFCLR	复位 EMB_STAT.CMPF	写 1 将 EMB_STAT.CMPF 清 0 (当 EMB_STAT.CMPST=0 时)，写 0 无效	WO
1	PWMSFCLR	复位 EMB_STAT.PWMSF	写 1 将 EMB_STAT.PWMSF 清 0 (当 EMB_STAT.PWMST=0 时)，写 0 无效	WO
0	Reserved	保留	-	-

22.3.9 EMB 中断许可寄存器 (EMB_INTEN)

Offset	Bit Position																															
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																				PORTININTEN4	PORTININTEN3	PORTININTEN2	PORTININTEN1	Reserved				SYSINTEN	CMPINTEN	PWMSINTEN	Reserved
Access																					RW	RW	RW	RW								

位/位域	标记	位名	功能描述	读写
31:12	Reserved	保留	-	-
11	PORTININTEN4	端口输入 4 控制中断使能	0b0: 端口输入控制不产生中断 0b1: 端口输入控制产生中断	RW
10	PORTININTEN3	端口输入 3 控制中断使能	0b0: 端口输入控制不产生中断 0b1: 端口输入控制产生中断	RW
9	PORTININTEN2	端口输入 2 控制中断使能	0b0: 端口输入控制不产生中断 0b1: 端口输入控制产生中断	RW
8	PORTININTEN1	端口输入 1 控制中断使能	0b0: 端口输入控制不产生中断 0b1: 端口输入控制产生中断	RW
7:4	Reserved	保留	-	-
3	SYSINTEN	系统错误控制中断使能	0b0: 系统错误控制不产生中断 0b1: 系统错误控制产生中断	RW
2	CMPINTEN	电压比较器比较结果控制中断使能	0b0: 电压比较器比较结果控制不产生中断 0b1: 电压比较器比较结果控制产生中断	RW
1	PWMSINTEN	PWM 输出同相位控制中断使能	0b0: PWM 输出同相位控制不产生中断 0b1: PWM 输出同相位控制产生中断	RW
0	Reserved	保留	-	-

22.3.10 EMB 控制 PWM 输出释放方式选择寄存器 (EMB_RLSSEL)

Offset	Bit Position																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															
0x18	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
Reset	0x00000000																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															
Name	Reserved																				PORTINRSEL4		PORTINRSEL3		PORTINRSEL2		PORTINRSEL1		Reserved				SYSRSEL		CMPRSEL		PWMRSEL		Reserved																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
Access																					RW	RW	RW	RW	RW	RW	RW	RW					RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:12	Reserved	保留	-	-
11	PORTINRSEL4	端口输入 4 控制释放方式选择	0b0: EMB_STAT.PORTINF4=0 时释放 PWM 输出控制 0b1: EMB_STAT.PORTINST4=0 时释放 PWM 输出控制	RW
10	PORTINRSEL3	端口输入 3 控制释放方式选择	0b0: EMB_STAT.PORTINF3=0 时释放 PWM 输出控制 0b1: EMB_STAT.PORTINST3=0 时释放 PWM 输出控制	RW
9	PORTINRSEL2	端口输入 2 控制释放方式选择	0b0: EMB_STAT.PORTINF2=0 时释放 PWM 输出控制 0b1: EMB_STAT.PORTINST2=0 时释放 PWM 输出控制	RW
8	PORTINRSEL1	端口输入 1 控制释放方式选择	0b0: EMB_STAT.PORTINF1=0 时释放 PWM 输出控制 0b1: EMB_STAT.PORTINST1=0 时释放 PWM 输出控制	RW
7:4	Reserved	保留	-	-
3	SYSRSEL	系统错误控制释放方式选择	0b0: EMB_STAT.SYSF=0 时释放 PWM 输出控制 0b1: EMB_STAT.SYSST=0 时释放 PWM 输出控制	RW
2	CMPRSEL	电压比较器比较结果控制释放方式选择	0b0: EMB_STAT.CMPF=0 时释放 PWM 输出控制 0b1: EMB_STAT.CMPST=0 时释放 PWM 输出控制	RW
1	PWMRSEL	PWM 输出控制释放方式选择	0b0: EMB_STAT.PWMSF=0 时释放 PWM 输出控制 0b1: EMB_STAT.PWMST=0 时释放 PWM 输出控制	RW
0	Reserved	保留	-	-

23 通用定时器 (TimerA)

23.1 简介

通用定时器 A (TimerA) 是一个具有 16/32 位计数宽度、8 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 40 路 PWM 输出。

23.2 主要特性

TimerA 基本的功能及特性如下表所示。

表 23-1 TimerA 的基本功能及特性

波形模式	锯齿波、三角波
基本功能	递加、递减计数方向
	同步启动计数器
	基准值缓存功能
	32 位级联计数
	正交编码计数
	8 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

23.3 功能说明

23.3.1 TimerA 框图

TimerA 的基本框图如所示。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。

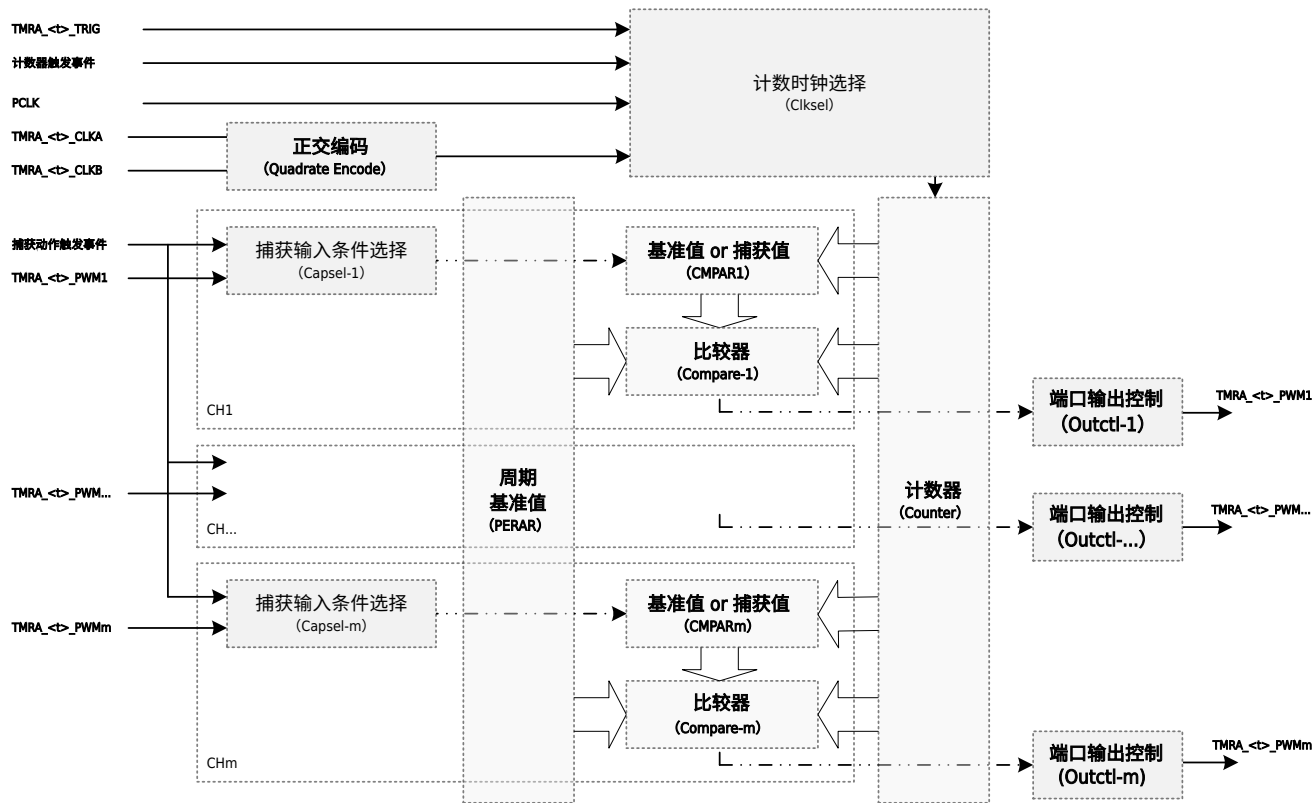


图 23-1 TimerA 基本框图

下表为 TimerA 的输入输出端口列表。

表 23-2 TimerA 端口列表

端口名	方向	功能
TMRA_<t>_PWMm	输入/输出	捕获输入事件端口或 PWM 输出端口（m=1~8）
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

23.3.2 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。

23.3.3 基本动作

23.3.3.1 波形模式

TimerA 有 2 种基本计数波形模式，锯齿波模式和三角波模式。两种波形模式的基本波形如下：

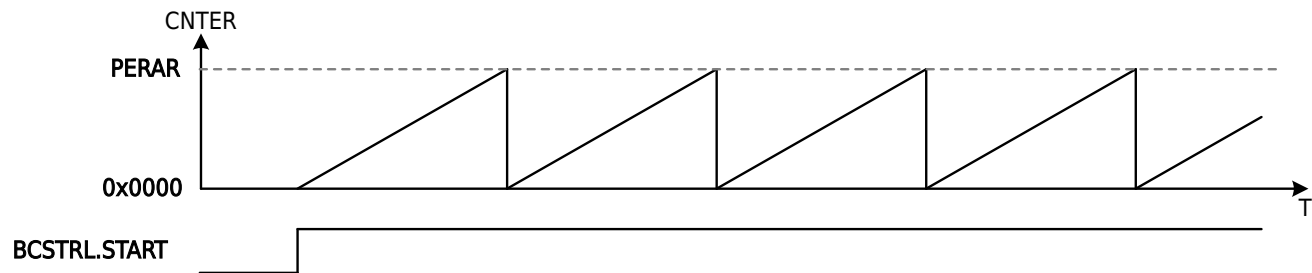


图 23-2 锯齿波波形 (递加计数)

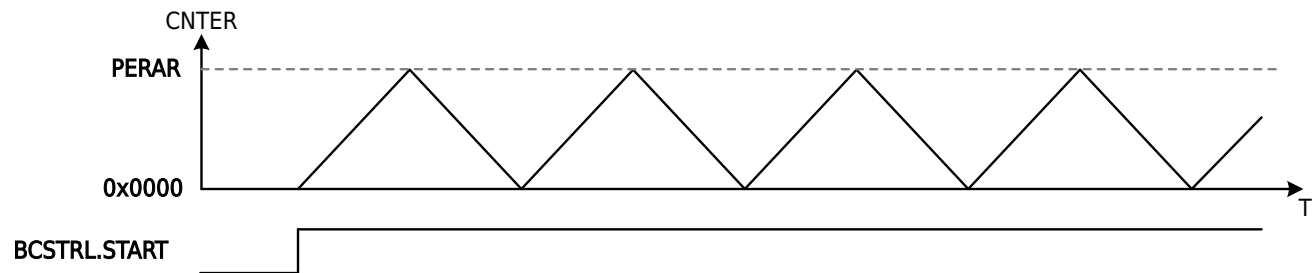


图 23-3 三角波波形

23.3.3.2 比较输出

各个 TimerA 单元内部均含有 8 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=1~8)。

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=1~8)。下图为单元 1 的比较输出动作例。

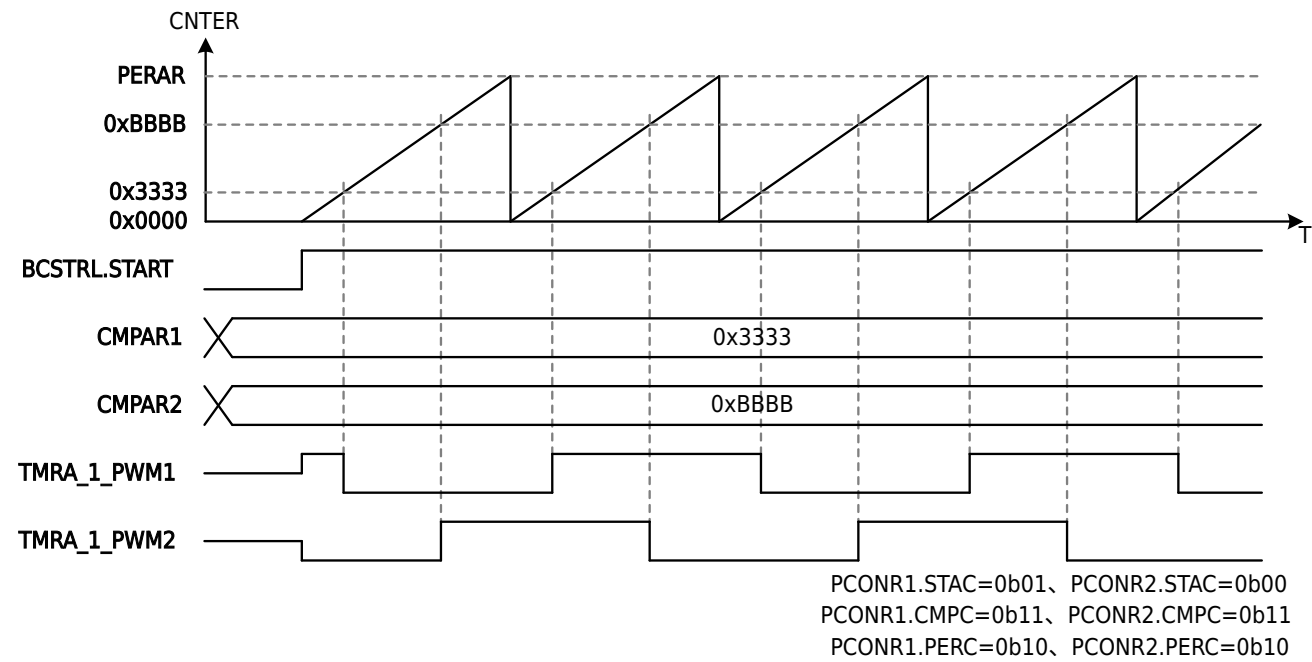


图 23-4 比较输出动作

23.3.3.3 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=1~8)。

捕获输入条件可以选择内部捕获动作触发事件（通过 TMRA_TRGSELx 寄存器选择，具体的寄存器内容详见《自动运行系统 (AOS)》章节“TimerA 硬件触发事件选择寄存器 (TMRA_TRGSELx) (x=0~3)”）、TMRA_<t>_PWMm 端口输入等，具体的条件选择可通过 CCONRm.HICP 位来设定 (m=1~8)。

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 23-3 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
TimerA_1	计数器触发事件：HCONR、HCUPR、HCDOR 所指的 TRGSEL 选择事件	TMRA_TRGSEL0
TimerA_2		TMRA_TRGSEL1
TimerA_3		TMRA_TRGSEL2
TimerA_4		TMRA_TRGSEL3
TimerA_5		TMRA_TRGSEL0
TimerA_1	捕获动作触发事件：CCONR 中所指的 TRGSEL 选择事件	TMRA_TRGSEL1
TimerA_2		TMRA_TRGSEL0
TimerA_3		TMRA_TRGSEL3
TimerA_4		TMRA_TRGSEL2
TimerA_5		TMRA_TRGSEL1

下图为捕获输入动作例。

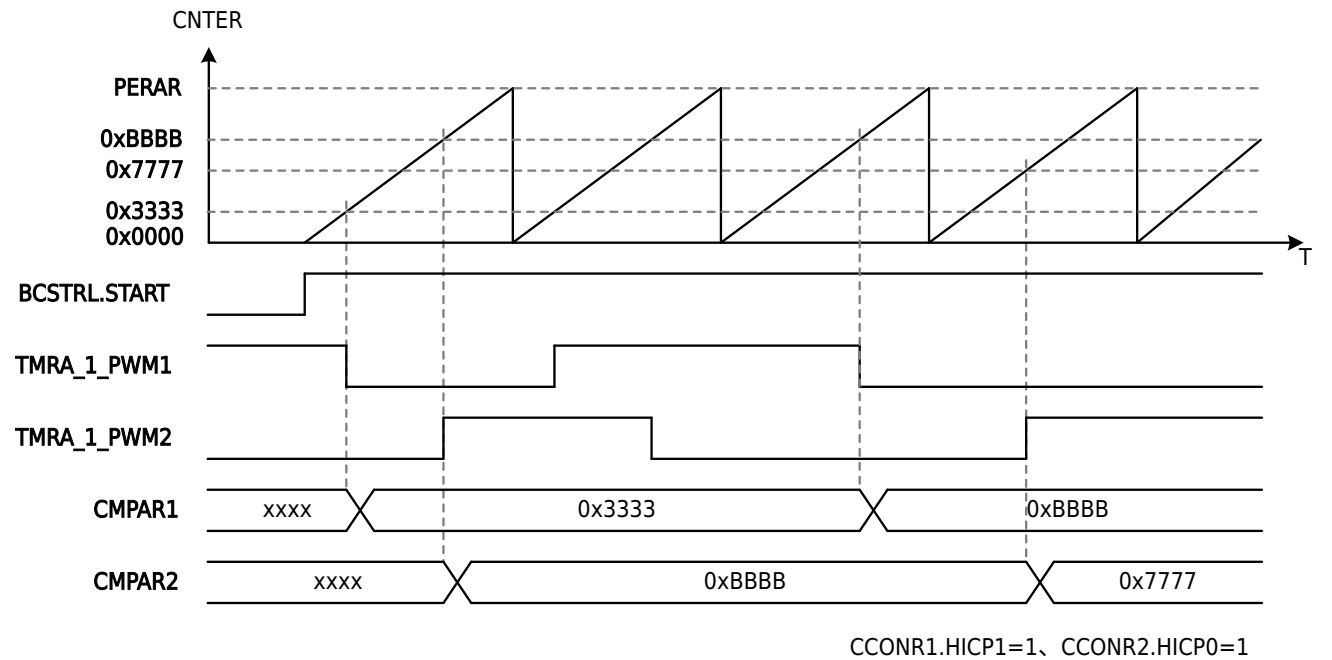


图 23-5 捕获输入动作

23.3.4 时钟源选择

TimerA 的计数时钟可以有以下几种选择：

1. PCLK 的 1、2、4、8、16、32、64、128、256、512、1024 分频 (BCSTR.LCKDIV 设定)
2. TMRA_<t>_TRIG 端口事件输入 (HCUPR.HCUPn 或 HCDOR.HCDO n 设定) (n=8~9)
3. 内部计数器触发事件输入 (HCUPR.HCUP10 或 HCDOR.HCDO10 设定)
4. 对称单元的计数上溢或计数下溢事件输入 (HCUPR.HCUPn 或 HCDOR.HCDO n 设定) (n=11~12)
5. TMRA_<t>_CLKA、TMRA_<t>_CLKB 的端口正交编码输入 (HCUPR.HCUPn 或 HCDOR.HCDO n 设定) (n=0~7)

计数时钟源选择 1 时为软件计数模式，选择其余时为硬件计数模式。

硬件计数模式下：

- 各时钟互相独立，可分别设定有效或无效。
- 软件时钟自动无效。

23.3.5 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。各端口的滤波功能的使能和滤波时钟的选择可通过设定滤波控制寄存器 (FCONR) 和捕获控制寄存器 (CCONRm) 的对应位来实现 (m=1~8)。

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。下图为 TMRA_1_CLKA 端口滤波动作例。

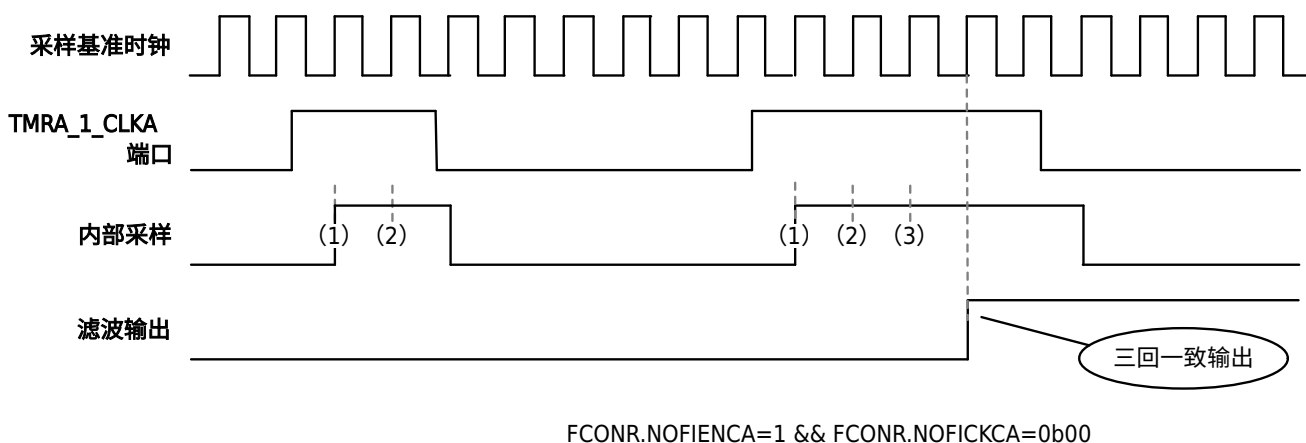


图 23-6 时钟输入端口的滤波功能

23.3.6 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 BCSTR.LSYNST 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 BCSTR.LSTART 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。

下图为设定单元 2 的 BCSTR.LSYNST=1 时的软件同步启动例。

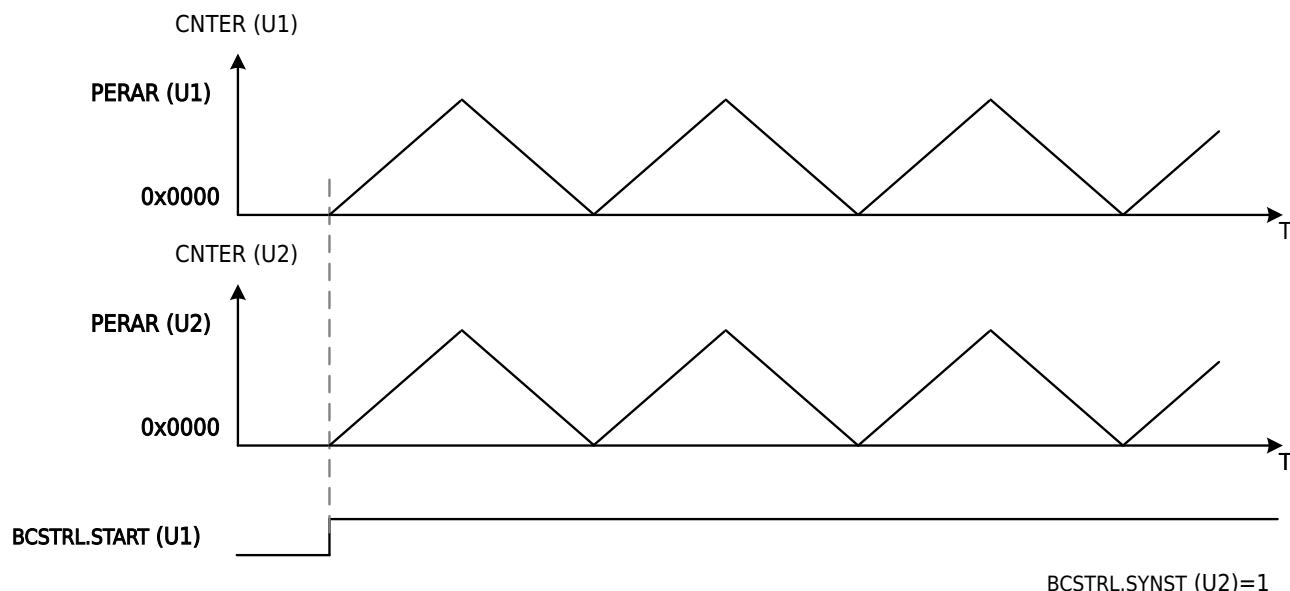


图 23-7 软件同步动作

另外，产品配置了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。

23.3.7 缓存功能

TimerA 的共计 8 个比较基准寄存器 (CMPARm) (m=1~8) 可以成对实现缓存功能。即 CMPAR2 作为 CMPAR1 的缓存值、CMPAR4 作为 CMPAR3 的缓存值、CMPAR6 作为 CMPAR5 的缓存值、CMPAR8 作为 CMPAR7 的缓存值。缓存控制寄存器 (BCONRm) (m=1~4) 分别实现对四组缓存功能的控制。

当缓存控制寄存器 BCONRm.BEN 位被置位时 (m=1~4)，缓存功能变为有效。计数器计数到特定时间点时就发生一次缓存传送 (CMPAR2/4/6/8->CMPAR1/3/5/7)。该“特定时间点”有以下几种情况：

1. 硬件计数模式时，计数到上溢点 (HCUPR!=0 时) 或下溢点 (HCDOR!=0 时) 或发生硬件清零计数器动作
2. 锯齿波计数模式 (BCSTRL.MODE=0) 时，计数器计数到上溢点 (BCSTRL.DIR=1 时) 或下溢点 (BCSTRL.DIR=0 时) 或发生硬件清零计数器动作
3. 三角波计数模式 (BCSTRL.MODE=1) 时，计数到峰点 (BCSTRL.DIR=1 && BCONRm.BSE0=1 时) (m=1~4)
4. 三角波计数模式 (BCSTRL.MODE=1) 时，计数到谷点 (BCSTRL.DIR=0 && BCONRm.BSE1=1 时) (m=1~4)

当缓存控制寄存器 BCONRm.BSEN 位被置位时，缓存特殊功能变为有效 (m=1~4)。对称单元的 CNTNR 被读时，本单元的 CMPAR 发生一次缓存传送 (CMPAR2/4/6/8->CMPAR1/3/5/7)。

下图是锯齿波模式时的缓存传送示意图。

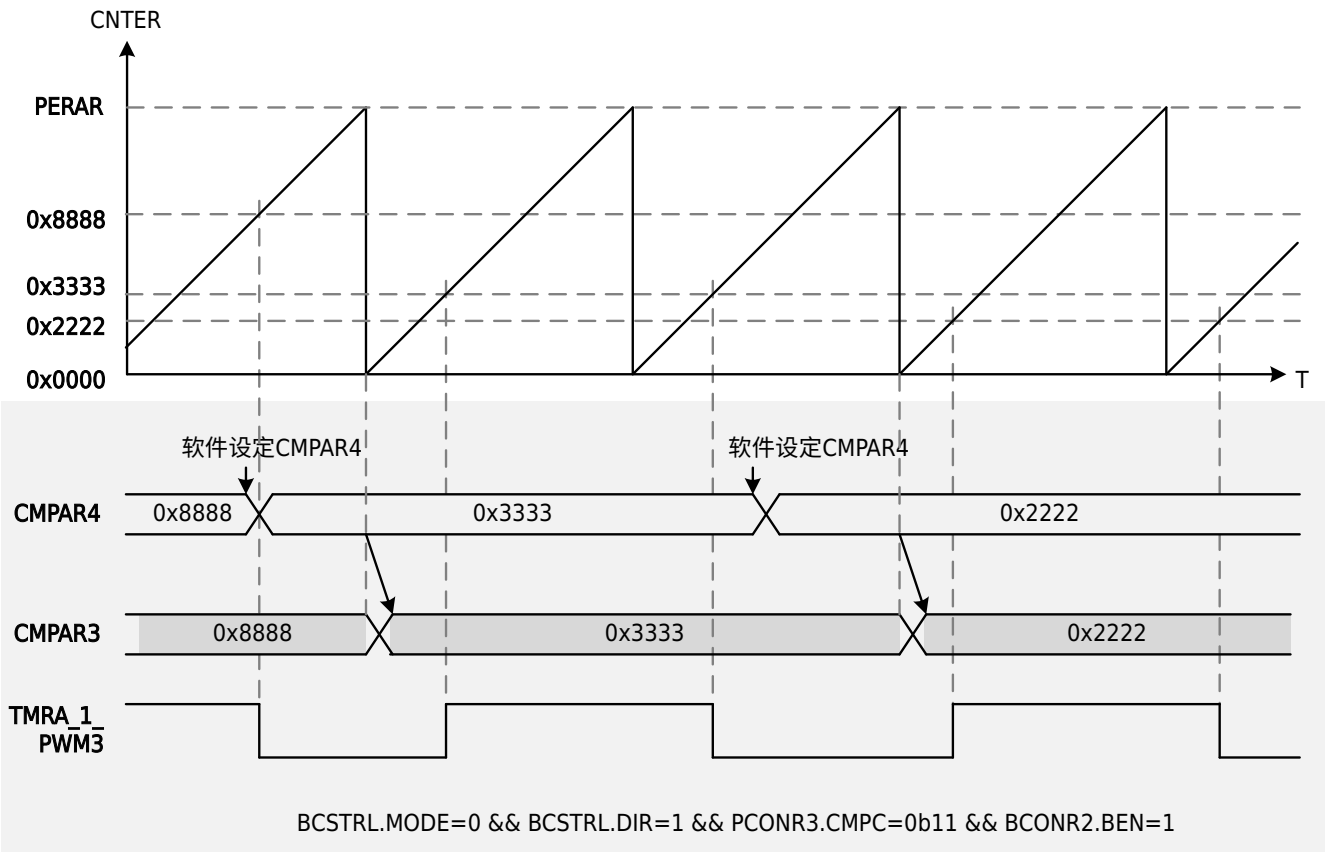


图 23-8 锯齿波模式时缓存动作

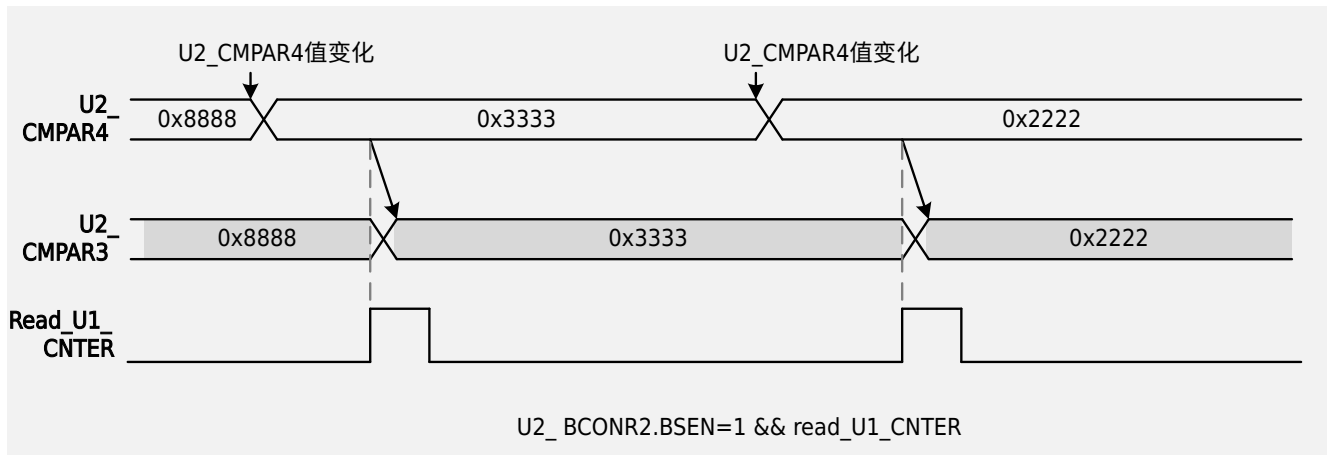


图 23-9 锯齿波模式时缓存与特殊缓存动作

23.3.8 级联计数

在“[计数时钟源选择](#)”章节中，当时钟源选择 4 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。下图是单元 1、2 级联计数的示意图。

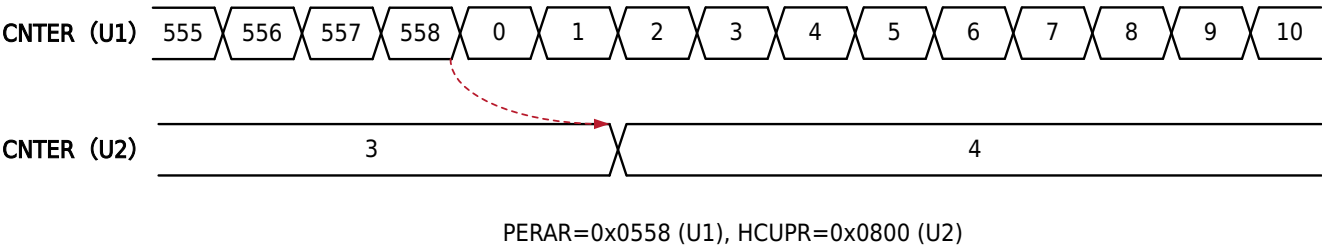


图 23-10 32 位级联计数动作

23.3.9 通用 PWM 输出

23.3.9.1 单边对齐 PWM 输出

锯齿波计数模式时，一个单元内的各个通道通过各种端口控制设定，可以实现单边对齐 PWM 输出（在计数周期点对齐）。例如，设定在比较基准值比较匹配时翻转（PCONR.CMPC=0b11）、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。
下图是锯齿波模式下，单边对齐 PWM 输出波形例。

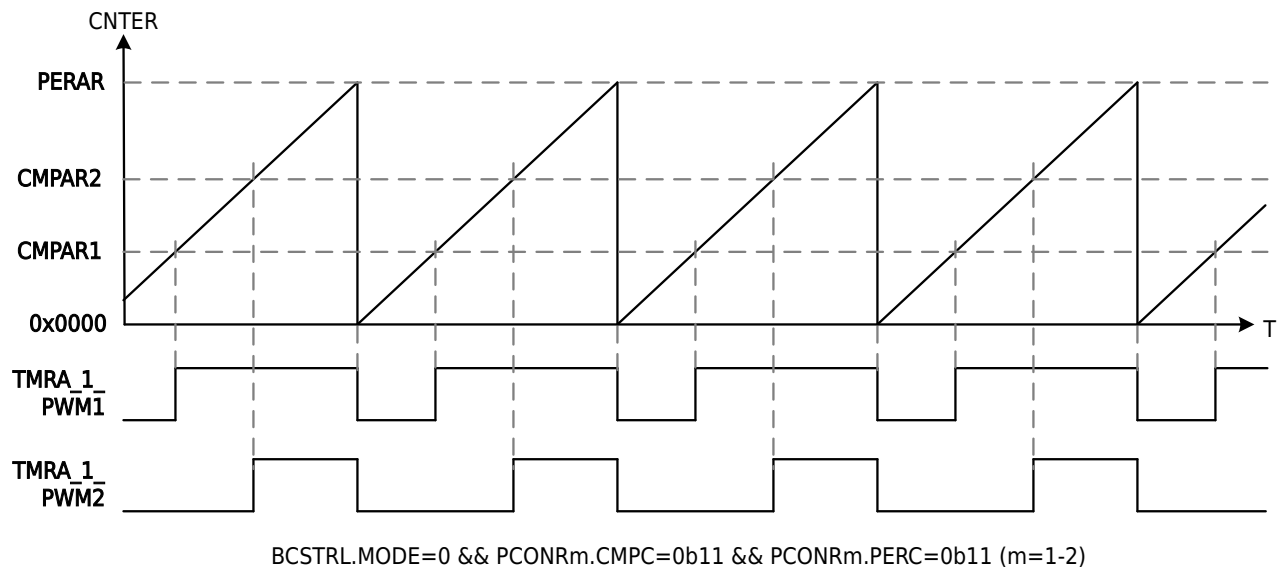


图 23-11 单边对齐 PWM 输出例

23.3.9.2 双边对称 PWM 输出

三角波计数模式时，一个单元内的各个通道通过各种端口控制设定，可以实现双边对称 PWM 输出（以计数峰点对称）。根据通道间的输出关系，可以实现独立的 PWM 输出或互补的 PWM 输出。
下图是三角波模式下，通道 1、2、3、4 的双边对称 PWM 输出波形例。其中，通道 1、2 可以作为一对互补 PWM 输出。

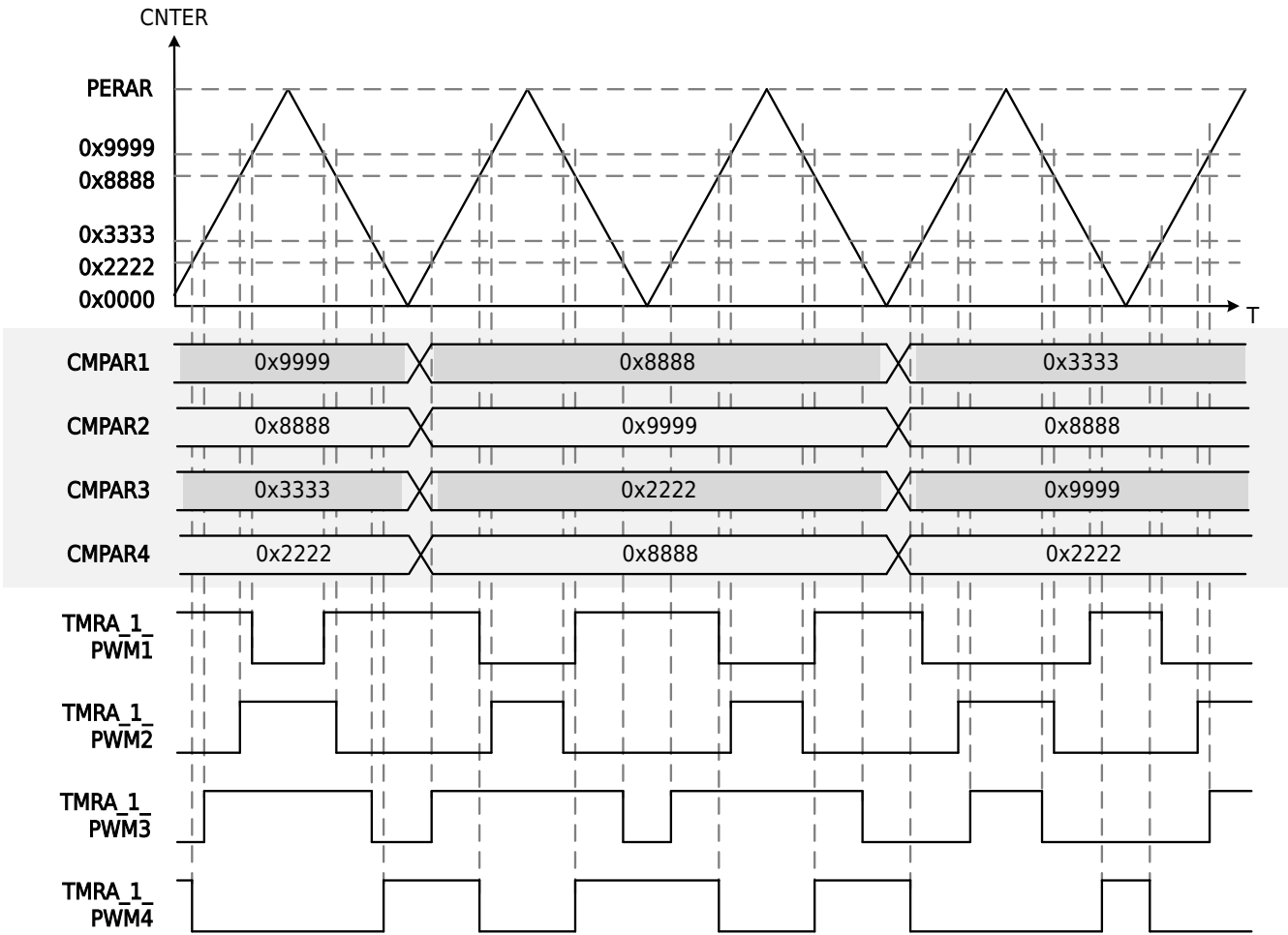


图 23-12 双边对称 PWM 输出例

23.3.10 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、通过设定公转单元的硬件递加事件选择寄存器（HCUPR）实现公转定时器计数。

23.3.10.1 位置计数模式

正交编码位置计数模式，是指根据 AIN、BIN 的输入实现基本计数功能、相位差计数功能和方向计数功能。

基本计数

基本计数动作是根据 AIN 或 BIN 端口的输入时钟进行计数。

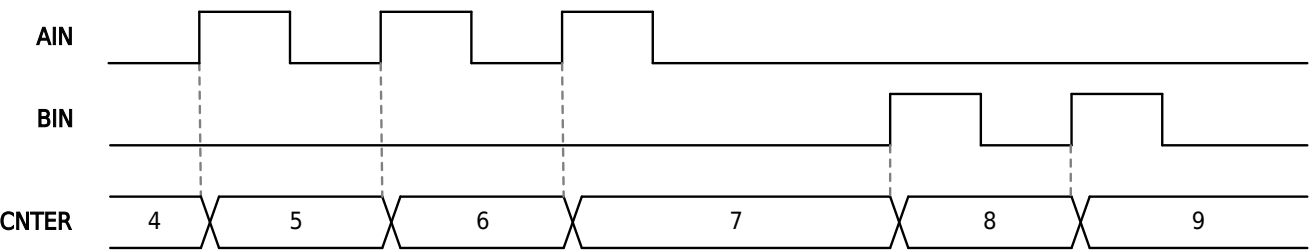
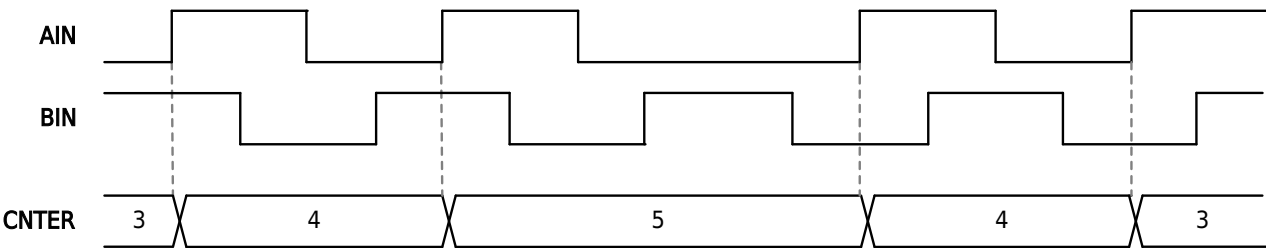


图 23-13 位置模式-基本计数

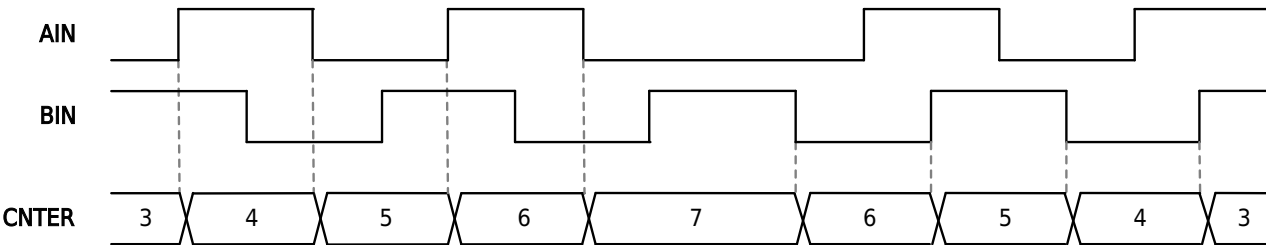
相位差计数

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。



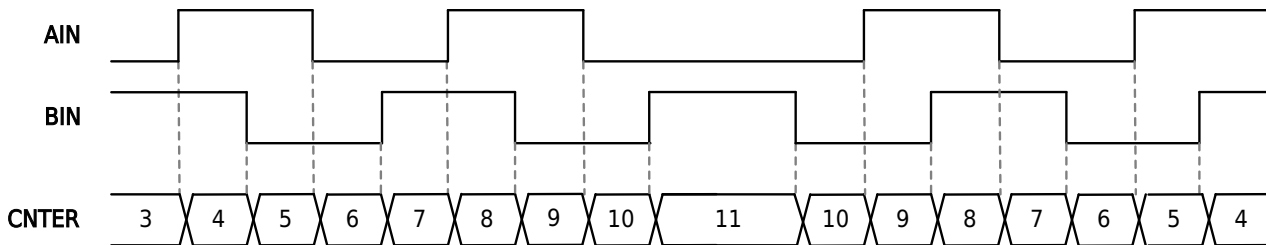
HCUPR.HCUP6=0b1 && HCDOR.HCDO4=0b1

图 23-14 位置计数模式-相位差计数 (1 倍计数)



HCUPR.HCUP6=HCUPR.HCUP5=0b1 && HCDOR.HCDO2=HCDOR.HCDO1=0b1

图 23-15 位置计数模式-相位差计数 (2 倍计数)



HCUPR.HCUP6=HCUPR.HCUP5=HCUPR.HCUP3=HCUPR.HCUP0=0b1
HCDOR.HCDO7=HCDOR.HCDO4=HCDOR.HCDO2=HCDOR.HCDO1=0b1

图 23-16 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。

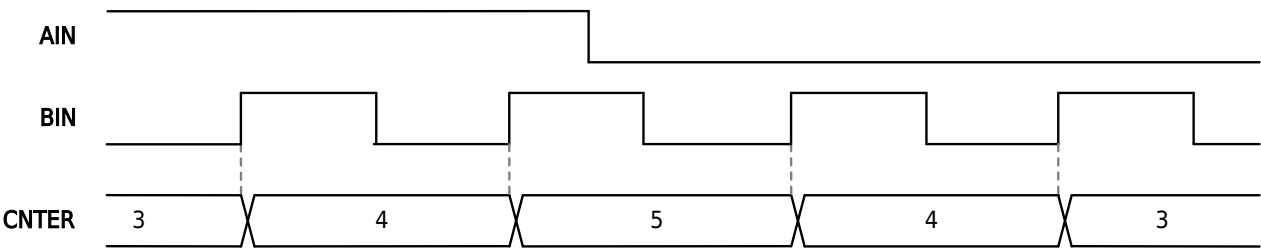


图 23-17 位置计数模式-方向计数

23.3.10.2 公转计数模式

正交编码公转模式，是指在 AIN、BIN 计数的基础上，加入 ZIN 的输入事件以实现对公转圈数等的判断。公转计数模式时根据公转定时器的计数方式，可实现 Z 相计数功能、位置溢出计数功能和混计数功能。

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。

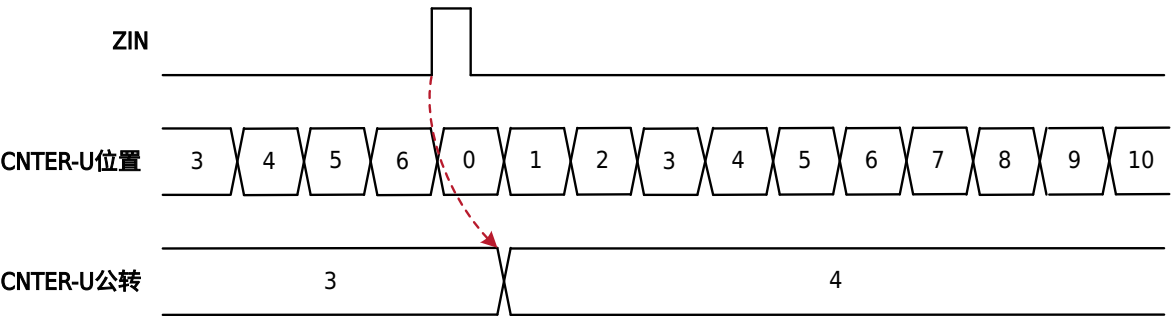


图 23-18 公转计数模式-Z 相计数

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。

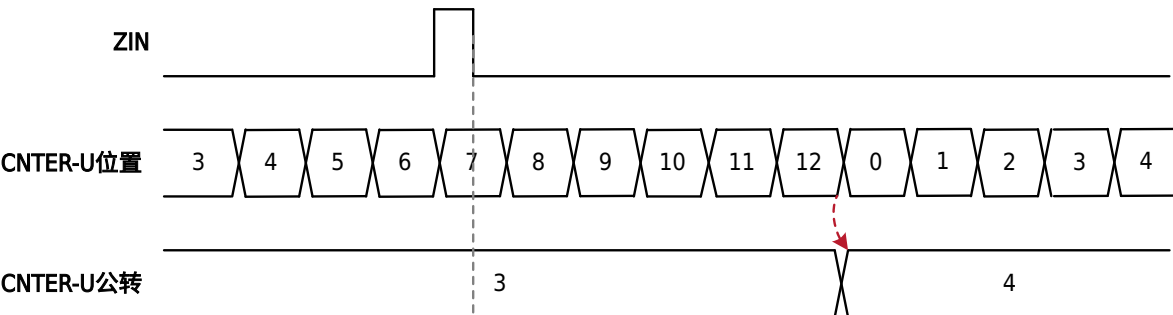


图 23-19 公转计数模式-位置溢出计数

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。

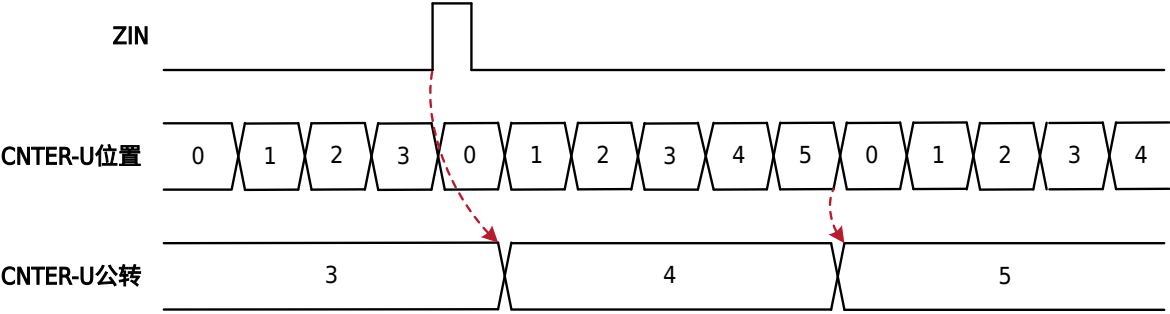


图 23-20 公转计数模式-混合计数

23.3.11 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。

23.3.11.1 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.CMPFm 位会被置为 1。此时，若中断控制寄存器 ICONR.ITENm 位设定为 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.ETENm 位设定为 1，则对应的事件请求 (TMRA_<t>_CMP) 会被触发 (m=1~8)。

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.ITENm 位设定为 1，则对应的中断请求 (TMRA_<t>_CMP) 被触发；若事件控制寄存器 ECONR.ETENm 位设定为 1，则对应的事件请求 (TMRA_<t>_CMP) 会被触发 (m=1~8)。

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。

23.3.11.2 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.OVFF 或 BCSTRH.UDFF 位会被置为 1。此时，若 BCSTRH.ITENOVF 或 BCSTRH.ITENUDF 位设定为 1 使能中断，则在对应的周期点可触发周期匹配中断 (TMRA_<t>_OVF 和 TMRA_<t>_UDF) 输出至中断模块 (INTC)；周期匹配事件则无对应的使能位控制，在对应的计数周期点就触发周期匹配事件 (TMRA_<t>_OVF 和 TMRA_<t>_UDF) 输出用于选择触发别的模块。

23.4 寄存器

23.4.1 寄存器总表

- TimerA_1 基地址：0x4003A000
- TimerA_2 基地址：0x4003A400
- TimerA_3 基地址：0x4003A800
- TimerA_4 基地址：0x4003AC00
- TimerA_5 基地址：0x40026000



说明

TimerA 为定时器名称，TimerA_<t> (<t>：1~5) 为该定时器单元。

表 23-4 通用定时器 (TimerA) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x0000	TMRA_CNTER	通用计数值寄存器	32/16
0x0004	TMRA_PERAR	周期基准值寄存器	32/16
0x0040+0x4*m	TMRA_CMPARm	比较基准值寄存器 m (m=1~8)	32/16
0x0080	TMRA_BCSTRL	控制状态寄存器 L	8
0x0081	TMRA_BCSTRH	控制状态寄存器 H	8
0x0090	TMRA_ICONR	中断控制寄存器	16
0x0094	TMRA_ECONR	事件控制寄存器	16
0x0098	TMRA_FCONR	滤波控制寄存器	16
0x009C	TMRA_STFLR	状态标志寄存器	16
0x00C0+0x8*m	TMRA_BCONRm	缓存控制寄存器 m (m=1~4)	16
0x0100+0x4*m	TMRA_CCONRm	捕获控制寄存器 m (m=1~8)	16
0x0140+0x4*m	TMRA_PCONRm	端口控制寄存器 m (m=1~8)	16
0x0084	TMRA_HCONR	硬件触发事件选择寄存器	16
0x0088	TMRA_HCUPR	硬件递加事件选择寄存器	16
0x008C	TMRA_HCDOR	硬件递减事件选择寄存器	16



说明

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 0x00000000，周期基准值寄存器、比较基准值寄存器的复位值为 0xFFFFFFFF。

23.4.2 通用计数值寄存器 (TMRA_CNTER)

单元 1:

Offset	Bit Position																															
0x0000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	CNT																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	CNT	计数值	当前定时器的计数值  说明 只能在定时器停止时写入。	RW

单元 2~5:

Offset	Bit Position															
0x0000	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	CNT															
Access	RW															

位/位域	标记	位名	功能描述	读写
15:0	CNT	计数值	当前定时器的计数值  说明 只能在定时器停止时写入。	RW

23.4.3 周期基准值寄存器 (TMRA_PERAR)

单元 1:

Offset	Bit Position																															
0x0004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	PER																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	PER	计数周期值	设定每轮计数的计数周期值	RW

单元 2~5:

Offset	Bit Position															
0x0004	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFF															
Name	PER															
Access	RW															

位/位域	标记	位名	功能描述	读写
15:0	PER	计数周期值	设定每轮计数的计数周期值	RW

23.4.4 比较基准值寄存器 (TMRA_CMPARm) (m=1~8)

偏移地址 (Offset) : 0x0040+0x4*m

单元 1:

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	CMP																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	CMP	计数比较基准值	设定比较基准值	RW

单元 2~5:

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFF															
Name	CMP															
Access	RW															

位/位域	标记	位名	功能描述	读写
15:0	CMP	计数比较基准值	设定比较基准值	RW

23.4.5 控制状态寄存器 L (TMRA_BCSTRL)

Offset	Bit Position							
0x0080	7	6	5	4	3	2	1	0
Reset	0x02							
Name	CKDIV				SYNST	MODE	DIR	START
Access	RW				RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
7:4	CKDIV	计数时钟选择	0b0000: PCLK 0b0001: PCLK/2 0b0010: PCLK/4 0b0011: PCLK/8 0b0100: PCLK/16 0b0101: PCLK/32 0b0110: PCLK/64 0b0111: PCLK/128 0b1000: PCLK/256 0b1001: PCLK/512 0b1010: PCLK/1024 - 其他: 禁止设置	RW
3	SYNST	同步启动使能	0b0: 对称单元中偶数单元与计数单元的同步启动无效 0b1: 对称单元中偶数单元与计数单元的同步启动有效  说明 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
2	MODE	计数模式	0b0: 锯齿波模式 0b1: 三角波模式	RW
1	DIR	计数方向	0b0: 计数器向下计数 0b1: 计数器向上计数  说明 三角波模式下，该位表示计数器方向状态。	RW
0	START	定时器启动	0b0: 定时器关闭 0b1: 定时器启动  说明 - 该位在硬件停止条件有效时，会自动变为 0。 - 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。

23.4.6 控制状态寄存器 H (TMRA_BCSTRH)

Offset	Bit Position							
0x0081	7	6	5	4	3	2	1	0
Reset	0x00							
Name	UDFF	OVFF	ITENUDF	ITENOVF	Reserved			OVSTP
Access	RW	RW	RW	RW				RW

位/位域	标记	位名	功能描述	读写
7	UDFF	下溢标志	0b0: 向下计数时, 未发生计数下溢 0b1: 向下计数时, 发生计数下溢	RW
6	OVFF	上溢标志	0b0: 向上计数时, 未发生计数上溢 0b1: 向上计数时, 发生计数上溢	RW
5	ITENUDF	下溢中断使能	0b0: 禁止 0b1: 使能	RW
4	ITENOVF	上溢中断使能	0b0: 禁止 0b1: 使能	RW
3:1	Reserved	保留	-	-
0	OVSTP	计数溢出停止控制	0b0: 计数器在计数上溢或下溢后, 继续计数 0b1: 计数器在计数上溢或下溢后, 停止计数  说明 溢出停止后再次启动定时器需初始化 TMRA_CNTER 值。	RW

23.4.7 中断控制寄存器 (TMRA_ICONR)

Offset	Bit Position															
0x0090	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved								ITEN8	ITEN7	ITEN6	ITEN5	ITEN4	ITEN3	ITEN2	ITEN1
Access									RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
15:8	Reserved	保留	-	-
7	ITEN8	计数匹配中断使能 8	中断条件: CMPAR8 寄存器与计数器值 (CNTERR) 相等, 或发生输入捕获 ● 0b0: 禁止中断输出 ● 0b1: 使能中断输出	RW
6	ITEN7	计数匹配中断使能 7	中断条件: CMPAR7 寄存器与计数器值 (CNTERR) 相等, 或发生输入捕获 ● 0b0: 禁止中断输出 ● 0b1: 使能中断输出	RW
5	ITEN6	计数匹配中断使能 6	中断条件: CMPAR6 寄存器与计数器值 (CNTERR) 相等, 或发生输入捕获 ● 0b0: 禁止中断输出 ● 0b1: 使能中断输出	RW
4	ITEN5	计数匹配中断使能 5	中断条件: CMPAR5 寄存器与计数器值 (CNTERR) 相等, 或发生输入捕获 ● 0b0: 禁止中断输出 ● 0b1: 使能中断输出	RW
3	ITEN4	计数匹配中断使能 4	中断条件: CMPAR4 寄存器与计数器值 (CNTERR) 相等, 或发生输入捕获 ● 0b0: 禁止中断输出 ● 0b1: 使能中断输出	RW
2	ITEN3	计数匹配中断使能 3	中断条件: CMPAR3 寄存器与计数器值 (CNTERR) 相等, 或发生输入捕获 ● 0b0: 禁止中断输出 ● 0b1: 使能中断输出	RW
1	ITEN2	计数匹配中断使能 2	中断条件: CMPAR2 寄存器与计数器值 (CNTERR) 相等, 或发生输入捕获 ● 0b0: 禁止中断输出 ● 0b1: 使能中断输出	RW
0	ITEN1	计数匹配中断使能 1	中断条件: CMPAR1 寄存器与计数器值 (CNTERR) 相等, 或发生输入捕获 ● 0b0: 禁止中断输出 ● 0b1: 使能中断输出	RW

23.4.8 事件控制寄存器 (TMRA_ECONR)

Offset	Bit Position															
0x0094	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved								ETEN8	ETEN7	ETEN6	ETEN5	ETEN4	ETEN3	ETEN2	ETEN1
Access									RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
15:8	Reserved	保留	-	-
7	ETEN8	计数匹配事件使能 8	事件条件：CMPAR8 寄存器与计数器值 (CNTERR) 相等，或发生输入捕获 ● 0b0：禁止事件输出 ● 0b1：使能事件输出	RW
6	ETEN7	计数匹配事件使能 7	事件条件：CMPAR7 寄存器与计数器值 (CNTERR) 相等，或发生输入捕获 ● 0b0：禁止事件输出 ● 0b1：使能事件输出	RW
5	ETEN6	计数匹配事件使能 6	事件条件：CMPAR6 寄存器与计数器值 (CNTERR) 相等，或发生输入捕获 ● 0b0：禁止事件输出 ● 0b1：使能事件输出	RW
4	ETEN5	计数匹配事件使能 5	事件条件：CMPAR5 寄存器与计数器值 (CNTERR) 相等，或发生输入捕获 ● 0b0：禁止事件输出 ● 0b1：使能事件输出	RW
3	ETEN4	计数匹配事件使能 4	事件条件：CMPAR4 寄存器与计数器值 (CNTERR) 相等，或发生输入捕获 ● 0b0：禁止事件输出 ● 0b1：使能事件输出	RW
2	ETEN3	计数匹配事件使能 3	事件条件：CMPAR3 寄存器与计数器值 (CNTERR) 相等，或发生输入捕获 ● 0b0：禁止事件输出 ● 0b1：使能事件输出	RW
1	ETEN2	计数匹配事件使能 2	事件条件：CMPAR2 寄存器与计数器值 (CNTERR) 相等，或发生输入捕获 ● 0b0：禁止事件输出 ● 0b1：使能事件输出	RW
0	ETEN1	计数匹配事件使能 1	事件条件：CMPAR1 寄存器与计数器值 (CNTERR) 相等，或发生输入捕获 ● 0b0：禁止事件输出 ● 0b1：使能事件输出	RW

23.4.9 滤波控制寄存器 (TMRA_FCONR)

Offset	Bit Position															
0x0098	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved	NOFICKCB		NOFIENCB		Reserved	NOFICKCA		NOFIENCA	Reserved				NOFICKTG		NOFIENTG
Access		RW		RW			RW		RW					RW		RW

位/位域	标记	位名	功能描述	读写
15	Reserved	保留	-	-
14:13	NOFICKCB	滤波采样基准时钟选择 CB	0b00: PCLK 0b01: PCLK/4 0b10: PCLK/16 0b11: PCLK/64	RW
12	NOFIENCB	捕获输入端口滤波 CB	0b0: TMRA_<t>_CLKB 端口输入滤波功能无效 0b1: TMRA_<t>_CLKB 端口输入滤波功能使能	RW
11	Reserved	保留	-	-
10:9	NOFICKCA	滤波采样基准时钟选择 CA	0b00: PCLK 0b01: PCLK/4 0b10: PCLK/16 0b11: PCLK/64	RW
8	NOFIENCA	捕获输入端口滤波 CA	0b0: TMRA_<t>_CLKA 端口输入滤波功能无效 0b1: TMRA_<t>_CLKA 端口输入滤波功能使能	RW
7:3	Reserved	保留	-	-
2:1	NOFICKTG	滤波采样基准时钟选择 TG	0b00: PCLK 0b01: PCLK/4 0b10: PCLK/16 0b11: PCLK/64	RW
0	NOFIENTG	捕获输入端口滤波 TG	0b0: TMRA_<t>_TRIG 输入端口滤波功能无效 0b1: TMRA_<t>_TRIG 输入端口滤波功能使能	RW



说明

对于 TimerA 单元 1~单元 4, PCLK 为 PCLK0; 对于 TimerA 单元 5, PCLK 为 PCLK1。

23.4.10 状态标志寄存器 (TMRA_STFLR)

Offset	Bit Position															
0x009C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	ICPF8	ICPF7	ICPF6	ICPF5	ICPF4	ICPF3	ICPF2	ICPF1	CM PF8	CM PF7	CM PF6	CM PF5	CM PF4	CM PF3	CM PF2	CM PF1
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
15	ICPF8	重复捕获标志 8	写 0 清除标志, 写 1 无效 读取: 0b0: 计数匹配标志位 CMPF8 为 1 后未发生捕获完成动作 0b1: 计数匹配标志位 CMPF8 为 1 后再次发生捕获完成动作	RW
14	ICPF7	重复捕获标志 7	写 0 清除标志, 写 1 无效 读取: 0b0: 计数匹配标志位 CMPF7 为 1 后未发生捕获完成动作 0b1: 计数匹配标志位 CMPF7 为 1 后再次发生捕获完成动作	RW
13	ICPF6	重复捕获标志 6	写 0 清除标志, 写 1 无效 读取: 0b0: 计数匹配标志位 CMPF6 为 1 后未发生捕获完成动作 0b1: 计数匹配标志位 CMPF6 为 1 后再次发生捕获完成动作	RW
12	ICPF5	重复捕获标志 5	写 0 清除标志, 写 1 无效 读取: 0b0: 计数匹配标志位 CMPF5 为 1 后未发生捕获完成动作 0b1: 计数匹配标志位 CMPF5 为 1 后再次发生捕获完成动作	RW
11	ICPF4	重复捕获标志 4	写 0 清除标志, 写 1 无效 读取: 0b0: 计数匹配标志位 CMPF4 为 1 后未发生捕获完成动作 0b1: 计数匹配标志位 CMPF4 为 1 后再次发生捕获完成动作	RW
10	ICPF3	重复捕获标志 3	写 0 清除标志, 写 1 无效 读取: 0b0: 计数匹配标志位 CMPF3 为 1 后未发生捕获完成动作 0b1: 计数匹配标志位 CMPF3 为 1 后再次发生捕获完成动作	RW

位/位域	标记	位名	功能描述	读写
9	ICPF2	重复捕获标志 2	写 0 清除标志, 写 1 无效 读取: 0b0: 计数匹配标志位 CMPF2 为 1 后未发生捕获完成动作 0b1: 计数匹配标志位 CMPF2 为 1 后再次发生捕获完成动作	RW
8	ICPF1	重复捕获标志 1	写 0 清除标志, 写 1 无效 读取: 0b0: 计数匹配标志位 CMPF1 为 1 后未发生捕获完成动作 0b1: 计数匹配标志位 CMPF1 为 1 后再次发生捕获完成动作	RW
7	CMPF8	计数匹配标志 8	写 0 清除标志, 写 1 无效 读取: 0b0: CMPAR8 寄存器的值与计数值不相等, 且未发生捕获完成动作 0b1: CMPAR8 寄存器的值与计数值相等, 或发生捕获完成动作	RW
6	CMPF7	计数匹配标志 7	写 0 清除标志, 写 1 无效 读取: 0b0: CMPAR7 寄存器的值与计数值不相等, 且未发生捕获完成动作 0b1: CMPAR7 寄存器的值与计数值相等, 或发生捕获完成动作	RW
5	CMPF6	计数匹配标志 6	写 0 清除标志, 写 1 无效 读取: 0b0: CMPAR6 寄存器的值与计数值不相等, 且未发生捕获完成动作 0b1: CMPAR6 寄存器的值与计数值相等, 或发生捕获完成动作	RW
4	CMPF5	计数匹配标志 5	写 0 清除标志, 写 1 无效 读取: 0b0: CMPAR5 寄存器的值与计数值不相等, 且未发生捕获完成动作 0b1: CMPAR5 寄存器的值与计数值相等, 或发生捕获完成动作	RW
3	CMPF4	计数匹配标志 4	写 0 清除标志, 写 1 无效 读取: 0b0: CMPAR4 寄存器的值与计数值不相等, 且未发生捕获完成动作 0b1: CMPAR4 寄存器的值与计数值相等, 或发生捕获完成动作	RW
2	CMPF3	计数匹配标志 3	写 0 清除标志, 写 1 无效 读取: 0b0: CMPAR3 寄存器的值与计数值不相等, 且未发生捕获完成动作 0b1: CMPAR3 寄存器的值与计数值相等, 或发生捕获完成动作	RW

位/位域	标记	位名	功能描述	读写
1	CMPF2	计数匹配标志 2	写 0 清除标志，写 1 无效 读取： ● 0b0：CMPAR2 寄存器的值与计数值不相等，且未发生捕获完成动作 ● 0b1：CMPAR2 寄存器的值与计数值相等，或发生捕获完成动作	RW
0	CMPF1	计数匹配标志 1	写 0 清除标志，写 1 无效 读取： ● 0b0：CMPAR1 寄存器的值与计数值不相等，且未发生捕获完成动作 ● 0b1：CMPAR1 寄存器的值与计数值相等，或发生捕获完成动作	RW

23.4.11 缓存控制寄存器 (TMRA_BCONRm) (m=1~4)

偏移地址 (Offset) : 0x00C0+0x8*m

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved												BSEN	BSE1	BSE0	BEN
Access													RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
15:4	Reserved	保留	-	-
3	BSEN	缓存特殊使能	0b0: 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 0b1: 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: $CMPAR_{2m} \rightarrow CMPAR_{2m-1}$	RW
2	BSE1	三角波缓存传送选择 1	0b0: 三角波计数模式计数到谷点时, 缓存值不传送 0b1: 三角波计数模式计数到谷点时, 缓存值传送, 即: $CMPAR_{2m} \rightarrow CMPAR_{2m-1}$  说明 仅当 BEN 为 1 时, 该位的配置有效。	RW
1	BSE0	三角波缓存传送选择 0	0b0: 三角波计数模式计数到峰点时, 缓存值不传送 0b1: 三角波计数模式计数到峰点时, 缓存值传送, 即: $CMPAR_{2m} \rightarrow CMPAR_{2m-1}$  说明 仅当 BEN 为 1 时, 该位的配置有效。	RW
0	BEN	缓存使能	0b0: $CMPAR_{2m-1}$ 基准值的缓存功能无效 0b1: $CMPAR_{2m-1}$ 基准值的缓存功能有效	RW



说明

- 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。

23.4.12 捕获控制寄存器 (TMRA_CCONRm) (m=1~8)

偏移地址 (Offset) : 0x0100+0x4*m

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved	NOFICKCP		NOFIENCP	HICP6	HICP5	HICP4	HICP3	Reserved	HICP2	HICP1	HICP0	Reserved			CAPMD
Access		RW		RW	RW	RW	RW	RW		RW	RW	RW				RW

位/位域	标记	位名	功能描述	读写
15	Reserved	保留	-	-
14:13	NOFICKCP	滤波采样基准时钟选择 CP	0b00: PCLK 0b01: PCLK/4 0b10: PCLK/16 0b11: PCLK/64	RW
12	NOFIENCP	捕获输入端口滤波 CP	0b0: TMRA_<t>_PWMm 端口输入滤波功能无效 0b1: TMRA_<t>_PWMm 端口输入滤波功能使能	RW
11	HICP6	捕获输入条件使能 6	0b0: TMRA_<t>_XOR 端口输入采样到下降沿时, 不发生捕获输入动作 0b1: TMRA_<t>_XOR 端口输入采样到下降沿时, 产生捕获输入动作  说明 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
10	HICP5	捕获输入条件使能 5	0b0: TMRA_<t>_XOR 端口输入采样到上升沿时, 不发生捕获输入动作 0b1: TMRA_<t>_XOR 端口输入采样到上升沿时, 产生捕获输入动作  说明 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

位/位域	标记	位名	功能描述	读写
9	HICP4	捕获输入条件使能 4	0b0: TMRA_<t>_TRIG 端口输入采样到下降沿时, 不发生捕获输入动作 0b1: TMRA_<t>_TRIG 端口输入采样到下降沿时, 产生捕获输入动作  说明 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
8	HICP3	捕获输入条件使能 3	0b0: TMRA_<t>_TRIG 端口输入采样到上升沿时, 不发生捕获输入动作 0b1: TMRA_<t>_TRIG 端口输入采样到上升沿时, 产生捕获输入动作  说明 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
7	Reserved	保留	-	-
6	HICP2	捕获输入条件使能 2	0b0: TMRA_TRGSEL 寄存器中指定的事件发生时, 不发生捕获输入动作 0b1: TMRA_TRGSEL 寄存器中指定的事件发生时, 产生捕获输入动作  说明 具体信息请参见《自动运行系统 (AOS) 》的“TimerA 硬件触发事件选择寄存器 (TMRA_TRGSELx) (x=0~3) ”。	RW
5	HICP1	捕获输入条件使能 1	0b0: TMRA_<t>_PWMm 端口输入采样到下降沿时, 不发生捕获输入动作 0b1: TMRA_<t>_PWMm 端口输入采样到下降沿时, 产生捕获输入动作	RW
4	HICP0	捕获输入条件使能 0	0b0: TMRA_<t>_PWMm 端口输入采样到上升沿时, 不发生捕获输入动作 0b1: TMRA_<t>_PWMm 端口输入采样到上升沿时, 产生捕获输入动作	RW
3:1	Reserved	保留	-	-
0	CAPMD	功能模式选择	0b0: 比较输出功能 0b1: 捕获输入功能	RW

**说明**

- 对于 TimerA 单元 1~单元 4, PCLK 为 PCLK0; 对于 TimerA 单元 5, PCLK 为 PCLK1。
- 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。

23.4.13 端口控制寄存器 (TMRA_PCONRm) (m=1~8)

偏移地址 (Offset) : 0x0140+0x4*m

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved			OUTEN	Reserved		FORC		PERC		CMPC		STPC		STAC	
Access				RW			RW		RW		RW		RW		RW	

位/位域	标记	位名	功能描述	读写
15:13	Reserved	保留	-	-
12	OUTEN	输出使能	0b0: PWM 输出功能时的 TMRA_<t>_PWMm 端口输出无效 0b1: PWM 输出功能时的 TMRA_<t>_PWMm 端口输出有效	RW
11:10	Reserved	保留	-	-
9:8	FORC	强制端口状态设定	0b0x: 不使用强制端口状态设定 0b10: 下周期开始, TMRA_<t>_PWMm 端口输出设定为低电平 0b11: 下周期开始, TMRA_<t>_PWMm 端口输出设定为高电平  说明 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。 该位可用于实现 PWM 输出占空比 0%或 100%的控制。	RW
7:6	PERC	周期值匹配时端口状态设定	0b00: 计数值与周期值相等时, TMRA_<t>_PWMm 端口输出设定为低电平 0b01: 计数值与周期值相等时, TMRA_<t>_PWMm 端口输出设定为高电平 0b10: 计数值与周期值相等时, TMRA_<t>_PWMm 端口输出保持先前状态 0b11: 计数值与周期值相等时, TMRA_<t>_PWMm 端口输出设定为反转电平	RW
5:4	CMPC	比较值匹配时端口状态设定	0b00: 计数值与 CMPARn 相等时, TMRA_<t>_PWMm 端口输出设定为低电平 0b01: 计数值与 CMPARn 相等时, TMRA_<t>_PWMm 端口输出设定为高电平 0b10: 计数值与 CMPARn 相等时, TMRA_<t>_PWMm 端口输出保持先前状态 0b11: 计数值与 CMPARn 相等时, TMRA_<t>_PWMm 端口输出设定为反转电平	RW

位/位域	标记	位名	功能描述	读写
3:2	STPC	计数停止时端口状态设定	● 0b00: 计数停止时, TMRA_<t>_PWMm 端口输出设定为低电平 ● 0b01: 计数停止时, TMRA_<t>_PWMm 端口输出设定为高电平 ● 0b10: 计数停止时, TMRA_<t>_PWMm 端口输出保持先前状态 ● 0b11: 计数停止时, TMRA_<t>_PWMm 端口输出保持先前状态	RW
1:0	STAC	计数开始时端口状态设定	● 0b00: 计数开始时, TMRA_<t>_PWMm 端口输出设定为低电平 ● 0b01: 计数开始时, TMRA_<t>_PWMm 端口输出设定为高电平 ● 0b10: 计数开始时, TMRA_<t>_PWMm 端口输出保持先前状态 ● 0b11: 计数开始时, TMRA_<t>_PWMm 端口输出保持先前状态  说明 该位设定只在不分频 (BCSTR.LCKDIV=0b0000) 的情况下有效, 其他分频请将该位设为 0b10 或 0b11。	RW

23.4.14 硬件触发事件选择寄存器 (TMRA_HCONR)

Offset	Bit Position															
0x0084	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	HCLE6	HCLE5	HCLE4	HCLE3	Reserved	HCLE2	HCLE1	HCLE0	Reserved	HSTP2	HSTP1	HSTP0	Reserved	HSTA2	HSTA1	HSTA0
Access	RW	RW	RW	RW		RW	RW	RW		RW	RW	RW		RW	RW	RW

位/位域	标记	位名	功能描述	读写
15	HCLE6	硬件清零条件 6	条件: TMRA_<t>_PWM3 端口输入采样到下降沿 ● 0b0: 条件匹配时, 硬件清零无效 ● 0b1: 条件匹配时, 硬件清零有效	RW
14	HCLE5	硬件清零条件 5	条件: TMRA_<t>_PWM3 端口输入采样到上升沿 ● 0b0: 条件匹配时, 硬件清零无效 ● 0b1: 条件匹配时, 硬件清零有效	RW
13	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 ● 0b0: 条件匹配时, 硬件清零无效 ● 0b1: 条件匹配时, 硬件清零有效	RW
12	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 ● 0b0: 条件匹配时, 硬件清零无效 ● 0b1: 条件匹配时, 硬件清零有效	RW
11	Reserved	保留	-	-
10	HCLE2	硬件清零条件 2	条件: TMRA_TRGSEL 寄存器中指定的事件发生 ● 0b0: 条件匹配时, 硬件清零无效 ● 0b1: 条件匹配时, 硬件清零有效  说明 具体信息请参见《自动运行系统 (AOS) 》的“TimerA 硬件触发事件选择寄存器 (TMRA_TRGSELx) (x=0~3) ”。	RW
9	HCLE1	硬件清零条件 1	条件: TMRA_<t>_TRIG 端口输入采样到下降沿 ● 0b0: 条件匹配时, 硬件清零无效 ● 0b1: 条件匹配时, 硬件清零有效	RW
8	HCLE0	硬件清零条件 0	条件: TMRA_<t>_TRIG 端口输入采样到上升沿 ● 0b0: 条件匹配时, 硬件清零无效 ● 0b1: 条件匹配时, 硬件清零有效	RW
7	Reserved	保留	-	-

位/位域	标记	位名	功能描述	读写
6	HSTP2	硬件停止条件 2	条件: TMRA_TRGSEL 寄存器中指定的事件发生 0b0: 条件匹配时, 硬件停止无效 0b1: 条件匹配时, 硬件停止有效  说明 具体信息请参见《自动运行系统 (AOS) 》的“TimerA 硬件触发事件选择寄存器 (TMRA_TRGSELx) (x=0~3) ”。	RW
5	HSTP1	硬件停止条件 1	条件: TMRA_<t>_TRIG 端口输入采样到下降沿 0b0: 条件匹配时, 硬件停止无效 0b1: 条件匹配时, 硬件停止有效	RW
4	HSTP0	硬件停止条件 0	条件: TMRA_<t>_TRIG 端口输入采样到上升沿 0b0: 条件匹配时, 硬件停止无效 0b1: 条件匹配时, 硬件停止有效	RW
3	Reserved	保留	-	-
2	HSTA2	硬件启动条件 2	条件: TMRA_TRGSEL 寄存器中指定的事件发生 0b0: 条件匹配时, 硬件启动无效 0b1: 条件匹配时, 硬件启动有效  说明 具体信息请参见《自动运行系统 (AOS) 》的“TimerA 硬件触发事件选择寄存器 (TMRA_TRGSELx) (x=0~3) ”。	RW
1	HSTA1	硬件启动条件 1	条件: - 同步启动功能无效时, TMRA_<t>_TRIG 端口输入采样到下降沿 - 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 (仅对称单元中的偶数单元支持该功能) 0b0: 条件匹配时, 硬件启动无效 0b1: 条件匹配时, 硬件启动有效	RW
0	HSTA0	硬件启动条件 0	条件: - 同步启动功能无效时, TMRA_<t>_TRIG 端口输入采样到上升沿 - 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 (仅对称单元中的偶数单元支持该功能) 0b0: 条件匹配时, 硬件启动无效 0b1: 条件匹配时, 硬件启动有效	RW

23.4.15 硬件递加事件选择寄存器 (TMRA_HCUPR)

Offset	Bit Position															
0x0088	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved			HCUP12	HCUP11	HCUP10	HCUP9	HCUP8	HCUP7	HCUP6	HCUP5	HCUP4	HCUP3	HCUP2	HCUP1	HCUP0
Access				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
15:13	Reserved	保留	-	-
12	HCUP12	硬件递加条件 12	条件：对称单元发生计数下溢 0b0：条件匹配时，硬件递加无效 0b1：条件匹配时，硬件递加有效	RW
11	HCUP11	硬件递加条件 11	条件：对称单元发生计数上溢 0b0：条件匹配时，硬件递加无效 0b1：条件匹配时，硬件递加有效	RW
10	HCUP10	硬件递加条件 10	条件：TMRA_TRGSEL 寄存器中指定的事件发生 0b0：条件匹配时，硬件递加无效 0b1：条件匹配时，硬件递加有效  说明 具体信息请参见《自动运行系统 (AOS)》的“TimerA 硬件触发事件选择寄存器 (TMRA_TRGSELx) (x=0~3)”。	RW
9	HCUP9	硬件递加条件 9	条件：TMRA_<t>_TRIG 端口上采样到下降沿 0b0：条件匹配时，硬件递加无效 0b1：条件匹配时，硬件递加有效	RW
8	HCUP8	硬件递加条件 8	条件：TMRA_<t>_TRIG 端口上采样到上升沿 0b0：条件匹配时，硬件递加无效 0b1：条件匹配时，硬件递加有效	RW
7	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 0b0：条件匹配时，硬件递加无效 0b1：条件匹配时，硬件递加有效	RW
6	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 0b0：条件匹配时，硬件递加无效 0b1：条件匹配时，硬件递加有效	RW
5	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 0b0：条件匹配时，硬件递加无效 0b1：条件匹配时，硬件递加有效	RW

位/位域	标记	位名	功能描述	读写
4	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 ● 0b0: 条件匹配时, 硬件递加无效 ● 0b1: 条件匹配时, 硬件递加有效	RW
3	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 ● 0b0: 条件匹配时, 硬件递加无效 ● 0b1: 条件匹配时, 硬件递加有效	RW
2	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 ● 0b0: 条件匹配时, 硬件递加无效 ● 0b1: 条件匹配时, 硬件递加有效	RW
1	HCUP1	硬件递加条件 1	条件: TMRA_<t>_CLKA 端口为低电平时, TMRA_<t>_CLKB 端口上采样到下降沿 ● 0b0: 条件匹配时, 硬件递加无效 ● 0b1: 条件匹配时, 硬件递加有效	RW
0	HCUP0	硬件递加条件 0	条件: TMRA_<t>_CLKA 端口为低电平时, TMRA_<t>_CLKB 端口上采样到上升沿 ● 0b0: 条件匹配时, 硬件递加无效 ● 0b1: 条件匹配时, 硬件递加有效	RW

23.4.16 硬件递减事件选择寄存器 (TMRA_HCDOR)

Offset	Bit Position															
0x008C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved			HCDO12	HCDO11	HCDO10	HCDO9	HCDO8	HCDO7	HCDO6	HCDO5	HCDO4	HCDO3	HCDO2	HCDO1	HCDO0
Access				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
15:13	Reserved	保留	-	-
12	HCDO12	硬件递减条件 12	条件：对称单元发生计数下溢 0b0：条件匹配时，硬件递减无效 0b1：条件匹配时，硬件递减有效	RW
11	HCDO11	硬件递减条件 11	条件：对称单元发生计数上溢 0b0：条件匹配时，硬件递减无效 0b1：条件匹配时，硬件递减有效	RW
10	HCDO10	硬件递减条件 10	条件：TMRA_TRGSEL 寄存器中指定的事件发生 0b0：条件匹配时，硬件递减无效 0b1：条件匹配时，硬件递减有效  说明 具体信息请参见《自动运行系统 (AOS) 》的“TimerA 硬件触发事件选择寄存器 (TMRA_TRGSELx) (x=0~3) ”。	RW
9	HCDO9	硬件递减条件 9	条件：TMRA_<t>_TRIG 端口上采样到下降沿 0b0：条件匹配时，硬件递减无效 0b1：条件匹配时，硬件递减有效	RW
8	HCDO8	硬件递减条件 8	条件：TMRA_<t>_TRIG 端口上采样到上升沿 0b0：条件匹配时，硬件递减无效 0b1：条件匹配时，硬件递减有效	RW
7	HCDO7	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 0b0：条件匹配时，硬件递减无效 0b1：条件匹配时，硬件递减有效	RW
6	HCDO6	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 0b0：条件匹配时，硬件递减无效 0b1：条件匹配时，硬件递减有效	RW
5	HCDO5	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 0b0：条件匹配时，硬件递减无效 0b1：条件匹配时，硬件递减有效	RW

位/位域	标记	位名	功能描述	读写
4	HCDO4	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 ● 0b0: 条件匹配时, 硬件递减无效 ● 0b1: 条件匹配时, 硬件递减有效	RW
3	HCDO3	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 ● 0b0: 条件匹配时, 硬件递减无效 ● 0b1: 条件匹配时, 硬件递减有效	RW
2	HCDO2	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 ● 0b0: 条件匹配时, 硬件递减无效 ● 0b1: 条件匹配时, 硬件递减有效	RW
1	HCDO1	硬件递减条件 1	条件: TMRA_<t>_CLKA 端口为低电平时, TMRA_<t>_CLKB 端口上采样到下降沿 ● 0b0: 条件匹配时, 硬件递减无效 ● 0b1: 条件匹配时, 硬件递减有效	RW
0	HCDO0	硬件递减条件 0	条件: TMRA_<t>_CLKA 端口为低电平时, TMRA_<t>_CLKB 端口上采样到上升沿 ● 0b0: 条件匹配时, 硬件递减无效 ● 0b1: 条件匹配时, 硬件递减有效	RW

24 通用定时器 (Timer0)

24.1 简介

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。

24.2 功能说明

24.2.1 Timer0 框图

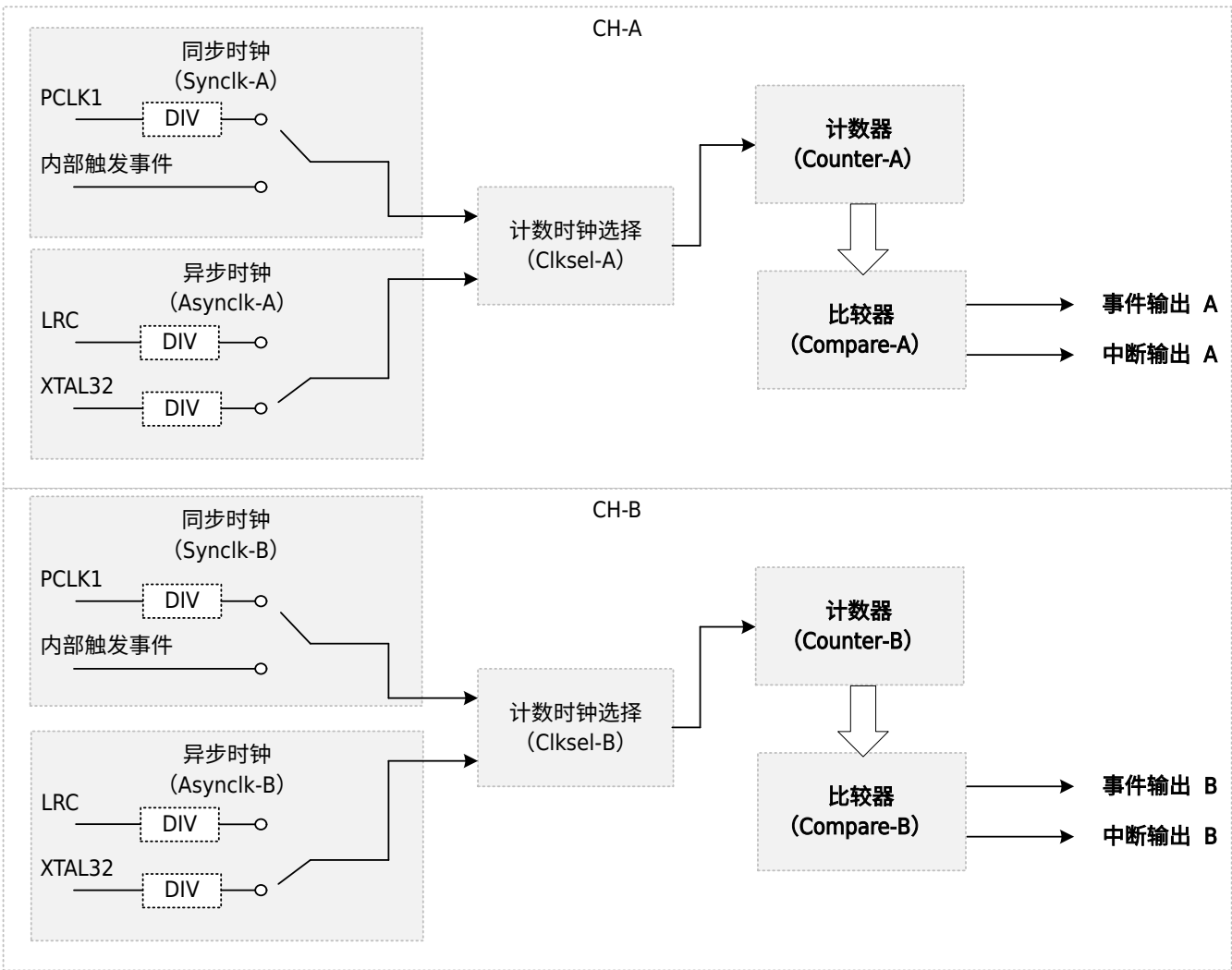


图 24-1 Timer0 基本框图

24.2.2 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。

同步计数时钟源

同步计数方式时 (BCONR.SYN=0)，时钟源可以有以下几种选择 (BCONR.SYN=0 设定)：

1. 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.SYN=0 & BCONR.CKDIVA=0 设定)；
2. 内部硬件触发事件输入作为同步计数时钟 (BCONR.SYN=1)。

异步计数时钟源

异步计数方式时 (BCONR.SYN=1)，时钟源可以有以下几种选择 (BCONR.ASYN=1 设定)：

1. 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.ASYN=0 & BCONR.CKDIVA=0 设定)；
2. 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.ASYN=1 & BCONR.CKDIVA=0 设定)。

24.2.3 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。

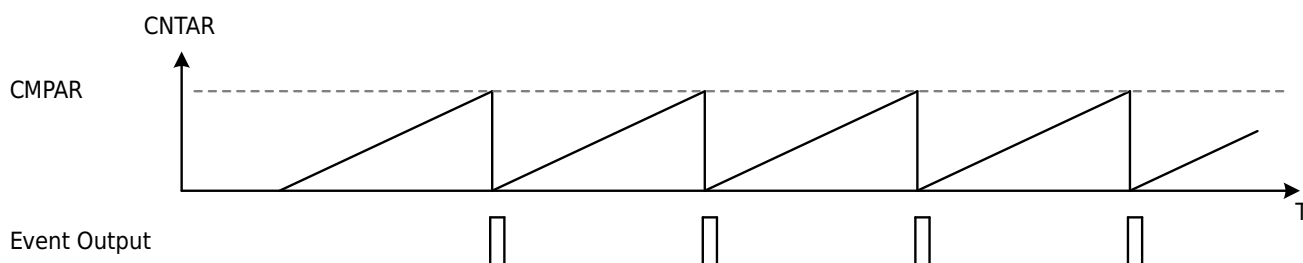


图 24-2 Timer0 计数时序图

24.2.4 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，寄存器内容请参见《自动运行系统 (AOS)》章节相关描述，具体的事件对应关系请参见《中断控制器 (INTC)》章节相关描述。使用内部硬件触发功能时，需要先将功能时钟控制 0 寄存器 (PWC_FCG0) 的 AOS 功能控制位使能。

24.2.5 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

- 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。
- 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。
- 当模块内部计数时钟选择 PCLK 的分频时钟时，Timer6 的启动时间和其他 Timer 不同步，请勿混用。

24.2.6 中断和事件

24.2.6.1 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_<t>_CMPn, <t>=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_<t>_CMPn, <t>=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_<t>_OVFn, <t>=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，具体请参见《中断控制器 (INTC)》章节相关描述。

24.2.6.2 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_<t>_CMPn 或 TMR0_<t>_OVFn, <t>=1~2; n=A、B) 输出信号，可以用于选择触发其他模块。

24.3 寄存器

24.3.1 寄存器总表

- Timer0_1 基地址：0x40024000
- Timer0_2 基地址：0x40024400



说明

Timer0 为定时器名称，Timer0_<t> (<t>：1~2) 为该定时器单元。

表 24-1 通用定时器 (Timer0) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x0000	TMRO_CNTAR	计数值寄存器 A	32
0x0004	TMRO_CNTBR	计数值寄存器 B	32
0x0008	TMRO_CMPAR	基准值寄存器 A	32
0x000C	TMRO_CMPBR	基准值寄存器 B	32
0x0010	TMRO_BCONR	基本控制寄存器	32
0x0014	TMRO_STFLR	状态标志寄存器	32

TMR_SYNENR 基地址：0x40055400

表 24-2 TMR_SYNENR 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x14	TMR_SYNENR	计数同步使能寄存器	32

24.3.2 计数值寄存器 m (TMR0_CNTmR) (m=A/B)

偏移地址 (Offset) : 0x0000 (CNTAR) , 0x0004 (CNTBR)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																CNTm															
Access																	RW															

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	CNTm	计数值	当前定时器的计数值  说明 在计数中时，不能向该寄存器中写入值。	RW

24.3.3 基准值寄存器 m (TMR0_CMPmR) (m=A/B)

偏移地址 (Offset) : 0x0008 (CMPAR) , 0x000C (CMPBR)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000FFFF																															
Name	Reserved																CMPm															
Access																	RW															

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	CMPm	基准值	设定计数基准值，产生比较匹配事件	RW

24.3.4 基本控制寄存器 (TMRO_BCONR)

Offset	Bit Position																															
0x0010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name																																
	HICPB	HCLEB	HSTPB	HSTAB	Reserved	ASYNCLKB	SYNCLKB	SYNSB		CKDIVB			OVENB	CMENB	CAPMDB	CSTB	HICPA	HCLEA	HSTPA	HSTAA	Reserved	ASYNCLKA	SYNCLKA	SYNSA		CKDIVA			OVENA	CMENA	CAPMDA	CSTA
Access	RW	RW	RW	RW		RW	RW	RW		RW			RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW		RW			RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31	HICPB	硬件触发捕获输入 B	条件：内部硬件触发事件有效 ● 0b0：条件匹配时，捕获输入无效 ● 0b1：条件匹配时，捕获输入有效	RW
30	HCLEB	硬件触发清零 B	条件：内部硬件触发事件有效 ● 0b0：条件匹配时，定时器清零无效 ● 0b1：条件匹配时，定时器清零有效	RW
29	HSTPB	硬件触发停止 B	条件：内部硬件触发事件有效 ● 0b0：条件匹配时，定时器停止无效 ● 0b1：条件匹配时，定时器停止有效	RW
28	HSTAB	硬件触发启动 B	条件：内部硬件触发事件有效 ● 0b0：条件匹配时，定时器启动无效 ● 0b1：条件匹配时，定时器启动有效	RW
27	Reserved	保留	-	-
26	ASYNCLKB	通道 B 异步计数时钟源选择	● 0b0：LRC ● 0b1：XTAL32  说明 当使能 UART 对应通道的接收超时功能时，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
25	SYNCLKB	通道 B 同步计数时钟源选择	● 0b0：PCLK1 ● 0b1：内部硬件触发事件	RW
24	SYNSB	通道 B 计数方式选择	● 0b0：同步计数方式 ● 0b1：异步计数方式	RW
23:20	CKDIVB	通道 B 计数时钟分频选择	● 0b0000：时钟源 ● 0b0001：时钟源/2 ● 0b0010：时钟源/4 ● 0b0011：时钟源/8 ● 0b0100：时钟源/16 ● 0b0101：时钟源/32 ● 0b0110：时钟源/64 ● 0b0111：时钟源/128 ● 0b1000：时钟源/256 ● 0b1001：时钟源/512	RW

位/位域	标记	位名	功能描述	读写
23:20	CKDIVB	通道 B 计数时钟分频选择	● 0b1010: 时钟源/1024 ● 其他: 禁止设置  说明 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。	RW
19	OVENB	计数溢出中断使能 B	中断条件: 计数溢出, 即计数值 (CNTBR) = 0xFFFF 时 ● 0b0: 禁止中断 ● 0b1: 使能中断	RW
18	CMENB	计数匹配中断使能 B	中断条件: CMPBR 寄存器与计数值 (CNTBR) 相等时, 或者发生捕获输入事件 ● 0b0: 禁止中断 ● 0b1: 使能中断	RW
17	CAPMDB	功能模式选择 B	● 0b0: 比较输出功能 ● 0b1: 捕获输入功能	RW
16	CSTB	通道 B 定时器启动	● 0b0: 通道 B 定时器关闭 ● 0b1: 通道 B 定时器启动  说明 该位在硬件触发停止条件有效时, 会自动变为 0。	RW
15	HICPA	硬件触发捕获输入 A	条件: 内部硬件触发事件有效 ● 0b0: 条件匹配时, 捕获输入无效 ● 0b1: 条件匹配时, 捕获输入有效	RW
14	HCLEA	硬件触发清零 A	条件: 内部硬件触发事件有效 ● 0b0: 条件匹配时, 定时器清零无效 ● 0b1: 条件匹配时, 定时器清零有效	RW
13	HSTPA	硬件触发停止 A	条件: 内部硬件触发事件有效 ● 0b0: 条件匹配时, 定时器停止无效 ● 0b1: 条件匹配时, 定时器停止有效	RW
12	HSTAA	硬件触发启动 A	条件: 内部硬件触发事件有效 ● 0b0: 条件匹配时, 定时器启动无效 ● 0b1: 条件匹配时, 定时器启动有效	RW
11	Reserved	保留	-	-
10	ASYNCLKA	通道 A 异步计数时钟源选择	● 0b0: LRC ● 0b1: XTAL32  说明 当使能 UART 对应通道的接收超时功能时, 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
9	SYNCLKA	通道 A 同步计数时钟源选择	● 0b0: PCLK1 ● 0b1: 内部硬件触发事件	RW
8	SYNSA	通道 A 计数方式选择	● 0b0: 同步计数方式 ● 0b1: 异步计数方式	RW

位/位域	标记	位名	功能描述	读写
7:4	CKDIVA	通道 A 计数时钟分频选择	● 0b0000: 时钟源 ● 0b0001: 时钟源/2 ● 0b0010: 时钟源/4 ● 0b0011: 时钟源/8 ● 0b0100: 时钟源/16 ● 0b0101: 时钟源/32 ● 0b0110: 时钟源/64 ● 0b0111: 时钟源/128 ● 0b1000: 时钟源/256 ● 0b1001: 时钟源/512 ● 0b1010: 时钟源/1024 ● 其他: 禁止设置  说明 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。	RW
3	OVENA	计数溢出中断使能 A	中断条件: 计数溢出, 即计数值 (CNTAR) = 0xFFFF 时 ● 0b0: 禁止中断 ● 0b1: 使能中断	RW
2	CMENA	计数匹配中断使能 A	中断条件: CMPAR 寄存器与计数值 (CNTAR) 相等时, 或者发生捕获输入事件 ● 0b0: 禁止中断 ● 0b1: 使能中断	RW
1	CAPMDA	功能模式选择 A	● 0b0: 比较输出功能 ● 0b1: 捕获输入功能	RW
0	CSTA	通道 A 定时器启动	● 0b0: 通道 A 定时器关闭 ● 0b1: 通道 A 定时器启动  说明 该位在硬件触发停止条件有效时, 会自动变为 0。	RW



说明

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, 具体请参见《通用同步异步收发器 (USART)》章节相关描述。

24.3.5 状态标志寄存器 (TMRO_STFLR)

Offset	Bit Position																															
0x0014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved													ICPB	OVFB	CMFB	Reserved													ICPA	OVFA	CMFA
Access														RW	RW	RW														RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:19	Reserved	保留	-	-
18	ICPB	重复捕获标志 B	0b0: 计数匹配标志位 CMFB 为 1 后未发生捕获输入动作 0b1: 计数匹配标志位 CMFB 为 1 后再次发生捕获输入动作	RW
17	OVFB	计数溢出标志 B	0b0: 计数值 (CNTBR) 未计数到 0xFFFF 0b1: 计数值 (CNTBR) 计数到 0xFFFF	RW
16	CMFB	计数匹配标志 B	0b0: CMPBR 寄存器的值与计数值 (CNTBR) 不相等且未发生捕获输入动作 0b1: CMPBR 寄存器的值与计数值 (CNTBR) 相等或发生捕获输入动作	RW
15:3	Reserved	保留	-	-
2	ICPA	重复捕获标志 A	0b0: 计数匹配标志位 CMFA 为 1 后未发生捕获输入动作 0b1: 计数匹配标志位 CMFA 为 1 后再次发生捕获输入动作	RW
1	OVFA	计数溢出标志 A	0b0: 计数值 (CNTAR) 未计数到 0xFFFF 0b1: 计数值 (CNTAR) 计数到 0xFFFF	RW
0	CMFA	计数匹配标志 A	0b0: CMPAR 寄存器的值与计数值 (CNTAR) 不相等且未发生捕获输入动作 0b1: CMPAR 寄存器的值与计数值 (CNTAR) 相等或发生捕获输入动作	RW



说明

该寄存器中的标志位只能写入 0，无法写入 1。

24.3.6 计数同步使能寄存器 (TMR_SYNENR)

Offset	Bit Position																															
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																	TMRAU5	TMRAU4	TMRAU3	TMRAU2	TMRAU1	TMR6U2	TMR6U1	Reserved	TMR4U3	TMR4U2	TMR4U1	TMR0U2B	TMR0U2A	TMR0U1B	TMR0U1A
Access																		RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	TMR0U2A	TMR0U1B

位/位域	标记	位名	功能描述	读写
31:15	Reserved	保留	-	-
14	TMRAU5	TimerA_5 软件启动使能位	0b0: TimerA 单元 5 软件启动无效 0b1: TimerA 单元 5 软件启动有效且定时器开始计数	RW
13	TMRAU4	TimerA_4 软件启动使能位	0b0: TimerA 单元 4 软件启动无效 0b1: TimerA 单元 4 软件启动有效且定时器开始计数	RW
12	TMRAU3	TimerA_3 软件启动使能位	0b0: TimerA 单元 3 软件启动无效 0b1: TimerA 单元 3 软件启动有效且定时器开始计数	RW
11	TMRAU2	TimerA_2 软件启动使能位	0b0: TimerA 单元 2 软件启动无效 0b1: TimerA 单元 2 软件启动有效且定时器开始计数	RW
10	TMRAU1	TimerA_1 软件启动使能位	0b0: TimerA 单元 1 软件启动无效 0b1: TimerA 单元 1 软件启动有效且定时器开始计数	RW
9	TMR6U2	Timer6_2 软件启动使能位	0b0: Timer6 单元 2 软件启动无效 0b1: Timer6 单元 2 软件启动有效且定时器开始计数	RW
8	TMR6U1	Timer6_1 软件启动使能位	0b0: Timer6 单元 1 软件启动无效 0b1: Timer6 单元 1 软件启动有效且定时器开始计数	RW
7	Reserved	保留	-	-
6	TMR4U3	Timer4_3 软件启动使能位	0b0: Timer4 单元 3 软件启动无效 0b1: Timer4 单元 3 软件启动有效且定时器开始计数	RW
5	TMR4U2	Timer4_2 软件启动使能位	0b0: Timer4 单元 2 软件启动无效 0b1: Timer4 单元 2 软件启动有效且定时器开始计数	RW
4	TMR4U1	Timer4_1 软件启动使能位	0b0: Timer4 单元 1 软件启动无效 0b1: Timer4 单元 1 软件启动有效且定时器开始计数	RW

位/位域	标记	位名	功能描述	读写
3	TMR0U2B	Timer0_2 通道 B 软件启动使能位	● 0b0: Timer0 单元 2 通道 B 软件启动无效 ● 0b1: Timer0 单元 2 通道 B 软件启动有效且定时器开始计数  说明 请在同步计数的情况下使用该功能。	RW
2	TMR0U2A	Timer0_2 通道 A 软件启动使能位	● 0b0: Timer0 单元 2 通道 A 软件启动无效 ● 0b1: Timer0 单元 2 通道 A 软件启动有效且定时器开始计数  说明 请在同步计数的情况下使用该功能。	RW
1	TMR0U1B	Timer0_1 通道 B 软件启动使能位	● 0b0: Timer0 单元 1 通道 B 软件启动无效 ● 0b1: Timer0 单元 1 通道 B 软件启动有效且定时器开始计数  说明 请在同步计数的情况下使用该功能。	RW
0	TMR0U1A	Timer0_1 通道 A 软件启动使能位	● 0b0: Timer0 单元 1 通道 A 软件启动无效 ● 0b1: Timer0 单元 1 通道 A 软件启动有效且定时器开始计数  说明 请在同步计数的情况下使用该功能。	RW



说明

- 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。

24.4 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，需间隔至少 6 个异步计数时钟。
4. 在选择异步计数的情况下，请将 BCONR.SYNCLKA设定为 0。
5. 在写 BCONR.CSTA时，请勿同时改写 BCONR.SYNSA (BCONR.SYNSA须保持当前值)，否则 BCONR.CSTA无法写入。
6. 用 Timer0 事件触发其他外设动作时，Timer0 对应通道的基准值寄存器设置须大于 5。

25 实时时钟 (RTC)

25.1 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。

25.2 主要特性

表 25-1 RTC 主要特性

计数时钟源	外部低速振荡器 (32.768kHz) 内部低速振荡器 (32.768kHz) 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	周期中断
	闹钟中断

25.3 功能说明

25.3.1 RTC 框图

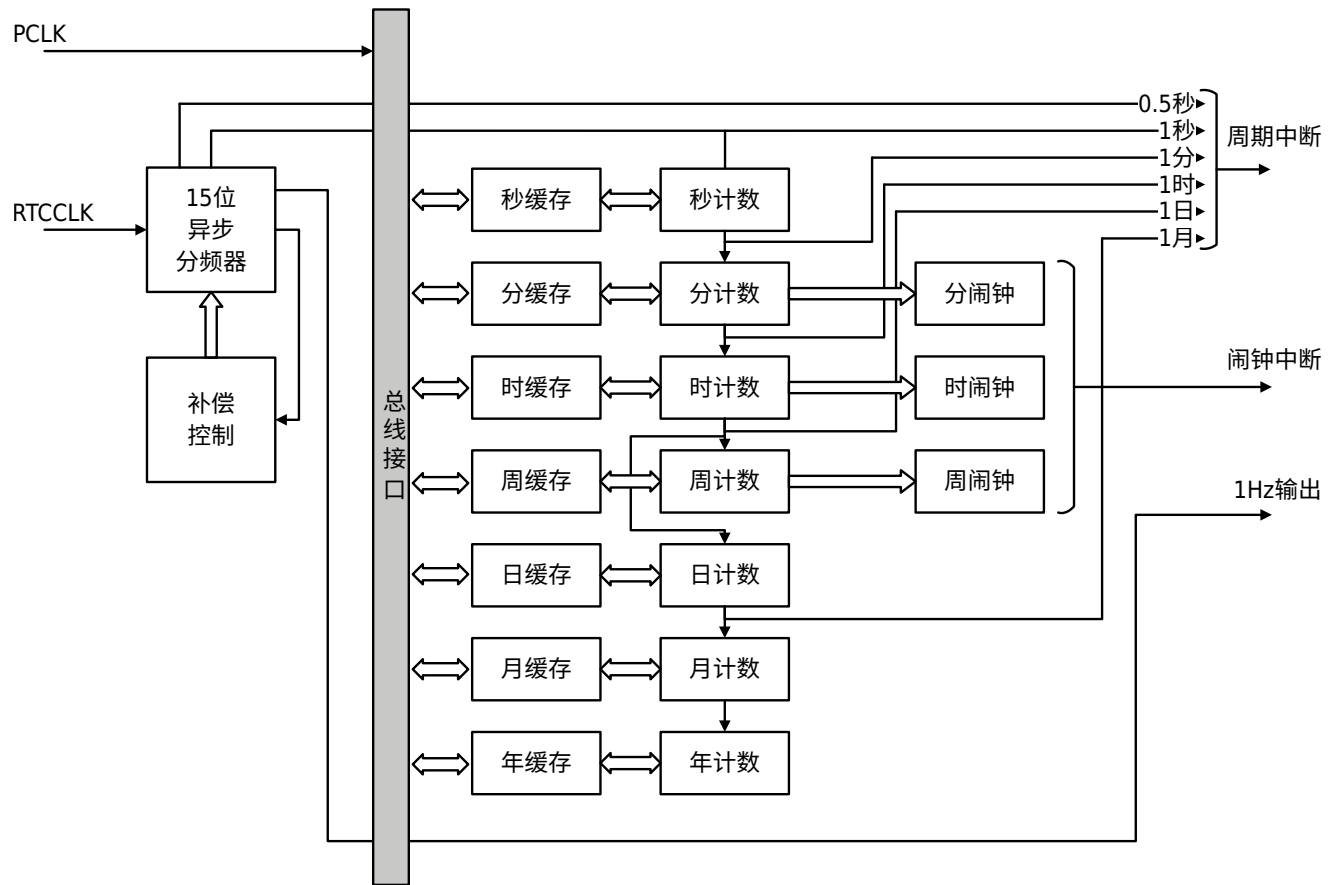


图 25-1 RTC 基本框图

25.3.2 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。

25.3.3 系统低功耗模式切换

在 RTC 计数开始后，系统如果立即切换为低功耗模式时，请执行下列任意一种确认后再进行模式切换。

- 在 RTC_CR1.START=1 设定后，经过 2 个以上的 RTC 计数时钟后再进行模式切换。
- 在 RTC_CR1.START=1 设定后，设定 RTC_CR2.RWREQ=1，查询 RTC_CR2.RWEN=1。设定日历计数寄存器，再设定 RTC_CR2.RWREQ=0，查询 RTC_CR2.RWEN=0，进行模式切换。

25.3.4 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。补偿方法请参见“[时钟误差补偿寄存器 \(RTC_ERRCRH、RTC_ERRCRL\)](#)”。

25.3.5 中断

RTC 支持 2 种中断类型。计时闹钟中断、周期中断。

闹钟中断

闹钟中断 RTC_ALM，在 RTC_CR2.ALMIE=1 且 RTC_CR2.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_ALMWEED) 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。

周期中断

周期中断 RTC_PRD，在 RTC_CR2.PRDI=1 时，选择的周期发生后，触发周期唤醒中断。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。

25.4 操作示例

25.4.1 RTC 计数开始设定

- 步骤 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；
- 步骤 2. 设定 RTC_CR1.START=0，停止计数；
- 步骤 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；
- 步骤 4. 设定 RTC_CR1，设定分、时、周、1Hz 时钟输出；
- 步骤 5. 设定秒，分，时，周，日，月，年的日历计数寄存器；
- 步骤 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；
- 步骤 7. 清除寄存器 RTC_CR2 中的标志寄存器位，并使能中断；
- 步骤 8. 设定 RTC_CR1.START=1，计数开始。

25.4.2 读出计数寄存器

- 步骤 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；
- 步骤 2. 查询直到 RTC_CR2.RWEN=1；
- 步骤 3. 读出全部或者部分秒，分，时，周，日，月，年计数寄存器值；
- 步骤 4. 设定 RTC_CR2.RWREQ=0；
- 步骤 5. 查询直到 RTC_CR2.RWEN=0。

25.4.3 写入计数寄存器

- 步骤 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；
- 步骤 2. 查询直到 RTC_CR2.RWEN=1；
- 步骤 3. 写入全部或者部分秒，分，时，周，日，月，年计数寄存器值；
- 步骤 4. 设定 RTC_CR2.RWREQ=0。注意，须在 1 秒内完成所有写操作；
- 步骤 5. 查询直到 RTC_CR2.RWEN=0。

25.4.4 闹钟设定

- 步骤 1. 设定 RTC_CR2.ALME=0，闹钟禁止；
- 步骤 2. 设定 RTC_CR2.ALMIE=1，闹钟中断许可；
- 步骤 3. 分闹钟 RTC_ALMMIN，时闹钟 RTC_ALMHOUR，周闹钟 RTC_ALMWEED 设定；
- 步骤 4. 设定 RTC_CR2.ALME=1，闹钟许可；
- 步骤 5. 等待闹钟中断；
- 步骤 6. 闹钟发生，RTC_CR2.ALMF=1，进入闹钟中断处理。

25.4.5 1Hz 输出

RTC 可输出 1Hz 时钟，提供三种精度输出方式：

1. 无时钟补偿的普通精度 1Hz 输出；
2. 每 32 秒内平均补偿的分布式补偿 1Hz 输出；
3. 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。

普通精度的 1Hz 输出

普通精度的 1Hz 输出设定如下：

- 步骤 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；
- 步骤 2. 设定 `RTC_CR1.START=0`，计数停止；
- 步骤 3. 1Hz 输出引脚设定；
- 步骤 4. `RTC_CR1.ONEHZOE=1`，时钟输出许可；
- 步骤 5. 设定 `RTC_CR1.START=1`，计数开始；
- 步骤 6. 等待 2 个计数周期以上；
- 步骤 7. 1Hz 输出开始。

分布式补偿 1Hz 输出

分布式补偿 1Hz 输出设定如下：

- 步骤 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；
- 步骤 2. 设定 `RTC_CR1.START=0`，计数停止；
- 步骤 3. 1Hz 输出引脚设定；
- 步骤 4. `RTC_CR1.ONEHZOE=1`，时钟输出许可；
- 步骤 5. `RTC_CR1.ONEHZSEL=0`，选择输出分布式补偿 1Hz 时钟；
- 步骤 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；
- 步骤 7. 时钟误差补偿寄存器 `RTC_ERRCRH.COMPEN=1`，误差补偿有效；
- 步骤 8. 设定 `RTC_CR1.START=1`，计数开始；
- 步骤 9. 等待 2 个计数周期以上；
- 步骤 10. 1Hz 输出开始。

均匀式补偿 1Hz 输出

均匀式补偿 1Hz 输出设定如下：

- 步骤 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；
- 步骤 2. 设定 `RTC_CR1.START=0`，计数停止；
- 步骤 3. RTC 输出引脚设定；
- 步骤 4. `RTC_CR1.ONEHZOE=1`，时钟输出许可；
- 步骤 5. `RTC_CR1.ONEHZSEL=1`，选择输出均匀式补偿 1Hz 时钟；
- 步骤 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；
- 步骤 7. 时钟误差补偿寄存器 `RTC_ERRCRH.COMPEN=1`，精度补偿有效；
- 步骤 8. 设定 `RTC_CR1.START=1`，计数开始；
- 步骤 9. 等待 2 个计数周期以上；
- 步骤 10. 1Hz 输出开始。

25.5 寄存器

25.5.1 寄存器总表

基地址：0x4004C000

表 25-2 实时时钟 (RTC) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x0000	RTC_CR0	控制寄存器 0	8
0x0004	RTC_CR1	控制寄存器 1	8
0x0008	RTC_CR2	控制寄存器 2	8
0x000C	RTC_CR3	控制寄存器 3	8
0x0010	RTC_SEC	秒计数寄存器	8
0x0014	RTC_MIN	分计数寄存器	8
0x0018	RTC_HOUR	时计数寄存器	8
0x001C	RTC_WEEK	周计数寄存器	8
0x0020	RTC_DAY	日计数寄存器	8
0x0024	RTC_MON	月计数寄存器	8
0x0028	RTC_YEAR	年计数寄存器	8
0x002C	RTC_ALMMIN	分闹钟寄存器	8
0x0030	RTC_ALMHOUR	时闹钟寄存器	8
0x0034	RTC_ALMWEEK	周闹钟寄存器	8
0x0038	RTC_ERRCRH	时钟误差补偿寄存器 H	8
0x003C	RTC_ERRCRL	时钟误差补偿寄存器 L	8

25.5.2 控制寄存器 0（RTC_CR0）

Offset	Bit Position							
0x0000	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved							RESET
Access								RW

位/位域	标记	位名	功能描述	读写
7:1	Reserved	保留	-	-
0	RESET	RTC 日历计数器 复位	写 1 初始化所有 RTC 寄存器，写 0 无效 读取： ● 0b0：正常计数状态或 RTC 软件复位结束 ● 0b1：RTC 处于复位状态	RW

25.5.3 控制寄存器 1 (RTC_CR1)

Offset	Bit Position							
0x0004	7	6	5	4	3	2	1	0
Reset	0x00							
Name	START	ONEHZSEL	ONEHZOE	Reserved	AMPM	PRDS		
Access	RW	RW	RW		RW	RW		

位/位域	标记	位名	功能描述	读写
7	START	RTC 计数开始	0b0: RTC 计数停止 0b1: RTC 计数开始  说明 用 RTC 事件作为触发源时，如需停止 RTC，请通过软件复位（对 RTC_CR0.RESET 写 1）对整个 RTC 进行复位。	RW
6	ONEHZSEL	1Hz 输出选择	0b0: 分布式补偿 1Hz 输出 0b1: 均匀式补偿 1Hz 输出  说明 RTC_ERRCRH.COMPEN=1 时，该位设定有效。	RW
5	ONEHZOE	1Hz 输出许可	0b0: 1Hz 输出禁止 0b1: 1Hz 输出许可	RW
4	Reserved	保留	-	-
3	AMPM	时制选择	0b0: 12 小时时制 0b1: 24 小时时制	RW
2:0	PRDS	周期中断选择	周期选择设定： 0b000: 不产生周期中断 0b001: 0.5 秒 0b010: 1 秒 0b011: 1 分钟 0b100: 1 小时 0b101: 1 天（每日 00 时 00 分 00 秒） 0b110/0b111: 1 月（每月 1 日 00 时 00 分 00 秒）  注意 在 START=1 计数过程中写入周期选择时，为防止误动作请将周期中断许可关闭。并且在写入后应将相关标志位清除。	RW

25.5.4 控制寄存器 2 (RTC_CR2)

Offset	Bit Position							
0x0008	7	6	5	4	3	2	1	0
Reset	0x00							
Name	ALME	ALMIE	PRDIE	Reserved	ALMF	PRDF	RWEN	RWREQ
Access	RW	RW	RW		RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
7	ALME	闹钟功能使能	0b0: 闹钟功能禁止 0b1: 闹钟功能许可	RW
6	ALMIE	闹钟中断使能	0b0: 闹钟中断禁止 0b1: 闹钟中断许可	RW
5	PRDIE	周期中断使能	0b0: 周期中断禁止 0b1: 周期中断许可	RW
4	Reserved	保留	-	-
3	ALMF	闹钟标志	写 0 清除标志, 写 1 无效 读取: 0b0: 闹钟不匹配 0b1: 闹钟匹配  说明 在 ALME=1 时有效, 闹钟匹配时, 一个计数时钟后置 1。	RW
2	PRDF	周期标志	写 0 清除标志, 写 1 无效 读取: 0b0: 周期不发生 0b1: 周期发生  说明 设定周期发生后, 该位置 1。	RW
1	RWEN	读出/写入允许	0b0: 读出/写入禁止 0b1: 读出/写入允许  说明 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
0	RWREQ	读出/写入请求	0b0: 正常计数模式 0b1: 读出/写入请求  说明 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。	RW

25.5.5 控制寄存器 3 (RTC_CR3)

Offset	Bit Position							
0x000C	7	6	5	4	3	2	1	0
Reset	0x00							
Name	RCKSEL	Reserved		LRCEN	Reserved			
Access	RW			RW				

位/位域	标记	位名	功能描述	读写
7	RCKSEL	RTC 计数时钟选择	0b0: 选择外部低速振荡器 (XTAL32) 0b1: 选择内部低速振荡器 (LRC) 或 XTAL 小数分频时钟  说明 在 RCKSEL=1 时, RTC 计数时钟由 PWC_PWRC6.RTCCKSEL 决定。 RTC 计数时钟选择 LRC 时, 计数偏差参考数据手册 LRC 振荡器特性。同时复位控制章节所示所有复位期间, LRC 频率偏低, 最大偏差-65%。	RW
6:5	Reserved	保留	-	-
4	LRCEN	内部低速振荡器使能	0b0: 内部低速振荡器 LRC 停止 0b1: 内部低速振荡器 LRC 工作  说明 内部低速振荡器作为 RTC 时钟源时, 请设定 LRCEN 位使能。	RW
3:0	Reserved	保留	-	-

25.5.6 秒计数寄存器（RTC_SEC）

Offset	Bit Position							
0x0010	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved	SECD			SECU			
Access		RW			RW			

位/位域	标记	位名	功能描述	读写
7	Reserved	保留	-	-
6:4	SECD	秒十位	秒十位计数值	RW
3:0	SECU	秒个位	秒个位计数值	RW

表示 0~59 秒，采用十进制计数。请写入十进制 0~59 的 BCD 码，写入超出范围的值将被忽略。

25.5.7 分计数寄存器（RTC_MIN）

Offset	Bit Position							
0x0014	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved	MIND			MINU			
Access		RW			RW			

位/位域	标记	位名	功能描述	读写
7	Reserved	保留	-	-
6:4	MIND	分十位	分十位计数值	RW
3:0	MINU	分个位	分个位计数值	RW

表示 0~59 分，采用十进制计数。请写入十进制 0~59 的 BCD 码，写入超出范围的值将被忽略。

25.5.8 时计数寄存器 (RTC_HOUR)

Offset	Bit Position							
0x0018	7	6	5	4	3	2	1	0
Reset	0x12							
Name	Reserved		HOURD		HOURU			
Access			RW		RW			

位/位域	标记	位名	功能描述	读写
7:6	Reserved	保留	-	-
5:4	HOURD	时十位	时十位计数值	RW
3:0	HOURU	时个位	时个位计数值	RW

24 小时制时，表示 0~23 小时。

12 小时制时，表示 0~12 小时。Bit5=0 表示 AM，则 01~12 表示上午；Bit5=1 表示 PM，则 01~12 表示下午。

请根据控制位 RTC_CR1.AMPM 的值，设定正确的十进制 0~23 或者 01~12，01~12 的 BCD 码，写入超出范围的值将被忽略。

具体时间取值可参考下表：

24 小时制 (AMPM=1)		12 小时制 (AMPM=0)	
时间	寄存器值	时间	寄存器值
00 时	0x00	AM 12 时	0x12
01 时	0x01	AM 01 时	0x01
02 时	0x02	AM 02 时	0x02
03 时	0x03	AM 03 时	0x03
04 时	0x04	AM 04 时	0x04
05 时	0x05	AM 05 时	0x05
06 时	0x06	AM 06 时	0x06
07 时	0x07	AM 07 时	0x07
08 时	0x08	AM 08 时	0x08
09 时	0x09	AM 09 时	0x09
10 时	0x10	AM 10 时	0x10
11 时	0x11	AM 11 时	0x11
12 时	0x12	PM 12 时	0x32
13 时	0x13	PM 01 时	0x21
14 时	0x14	PM 02 时	0x22
15 时	0x15	PM 03 时	0x23
16 时	0x16	PM 04 时	0x24
17 时	0x17	PM 05 时	0x25
18 时	0x18	PM 06 时	0x26

24 小时制 (AMPM=1)		12 小时制 (AMPM=0)	
时间	寄存器值	时间	寄存器值
19 时	0x19	PM 07 时	0x27
20 时	0x20	PM 08 时	0x28
21 时	0x21	PM 09 时	0x29
22 时	0x22	PM 10 时	0x30
23 时	0x23	PM 11 时	0x31

25.5.9 日计数寄存器 (RTC_DAY)

Offset	Bit Position							
0x0020	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved		DAYD		DAYU			
Access			RW		RW			

位/位域	标记	位名	功能描述	读写
7:6	Reserved	保留	-	-
5:4	DAYD	日十位	日十位计数值	RW
3:0	DAYU	日个位	日个位计数值	RW

表示 1~31 日，采用十进制计数，自动计算闰年和月份。请写入十进制 1~28、1~29、1~30 或 1~31 的 BCD 码，写入超出范围的值将被忽略。

具体日计数取值可参考下表：

月份	日计数值
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

25.5.10 周计数寄存器 (RTC_WEEK)

Offset	Bit Position							
0x001C	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved					WEEK		
Access						RW		

位/位域	标记	位名	功能描述	读写
7:3	Reserved	保留	-	-
2:0	WEEK	周	周计数值	RW

十进制 0~6 表示周日~周六。请写入十进制 0~6 的 BCD 码，写入超出范围的值将被忽略。

具体周计数取值可参考下表：

周	周计数值
周日	0x00
周一	0x01
周二	0x02
周三	0x03
周四	0x04
周五	0x05
周六	0x06

25.5.11 月计数寄存器 (RTC_MON)

Offset	Bit Position							
0x0024	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved			MON				
Access				RW				

位/位域	标记	位名	功能描述	读写
7:5	Reserved	保留	-	-
4:0	MON	月	月计数值	RW

表示 1~12 月，采用十进制计数。请写入十进制 1~12 的 BCD 码，写入超出范围的值将被忽略。

25.5.12 年计数寄存器 (RTC_YEAR)

Offset	Bit Position							
0x0028	7	6	5	4	3	2	1	0
Reset	0x00							
Name	YEAR				YEARU			
Access	RW				RW			

位/位域	标记	位名	功能描述	读写
7:4	YEARD	年十位	年十位计数值	RW
3:0	YEARU	年个位	年个位计数值	RW

表示 0~99 年，采用十进制计数，根据月进位计数。自动计算闰年如：00、04、08、...、92、96 等。请写入十进制 0~99 的 BCD 码，写入超出范围的值将被忽略。

25.5.13 分闹钟寄存器 (RTC_ALMMIN)

Offset	Bit Position							
0x002C	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved	ALMMIND			ALMMINU			
Access		RW			RW			

位/位域	标记	位名	功能描述	读写
7	Reserved	保留	-	-
6:4	ALMMIND	分闹钟十位	分闹钟十位计数值	RW
3:0	ALMMINU	分闹钟个位	分闹钟个位计数值	RW

表示 0~59 分，采用十进制计数。请写入十进制 0~59 的 BCD 码，写入超出范围的值将不会发生闹钟匹配。

25.5.14 时闹钟寄存器 (RTC_ALMHOUR)

Offset	Bit Position							
0x0030	7	6	5	4	3	2	1	0
Reset	0x12							
Name	Reserved		ALMHOURD		ALMHOURU			
Access			RW		RW			

位/位域	标记	位名	功能描述	读写
7:6	Reserved	保留	-	-
5:4	ALMHOURD	时闹钟十位	时闹钟十位计数值	RW
3:0	ALMHOURU	时闹钟个位	时闹钟个位计数值	RW

请根据控制位 RTC_CR1.AMPM 的值，设定正确的十进制 0~23 或者 01~12，21~32 的 BCD 码，写入超出范围的值将不会发生闹钟匹配。

25.5.15 周闹钟寄存器 (RTC_ALMWEEK)

Offset	Bit Position							
0x0034	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved	ALMWEEK						
Access		RW						

位/位域	标记	位名	功能描述	读写
7	Reserved	保留	-	-
6:0	ALMWEEK	周闹钟	周闹钟匹配值 Bit0~Bit6 分别对应周日~周六，对应位置 1 时，代表每周该日闹钟有效。 例如：设置 Bit0=1，Bit5=1 表示周日和周五闹钟设定有效。	RW

请设定正确的闹钟匹配值，否则不会发生闹钟匹配。

25.5.16 时钟误差补偿寄存器 (RTC_ERRCRH、RTC_ERRCRL)

RTC_ERRCRH

Offset	Bit Position							
0x0038	7	6	5	4	3	2	1	0
Reset	0x00							
Name	COMPEN	Reserved						COMP[8]
Access	RW							RW

位/位域	标记	位名	功能描述	读写
7	COMPEN	补偿使能	0b0: 时钟误差补偿无效 0b1: 时钟误差补偿有效	RW
6:1	Reserved	保留	-	-
0	COMP[8]	补偿值	与 COMP[7:0]共同设定补偿值	RW

RTC_ERRCRL

Offset	Bit Position							
0x003C	7	6	5	4	3	2	1	0
Reset	0x20							
Name	COMP[7:0]							
Access	RW							

位/位域	标记	位名	功能描述	读写
7:0	COMP[7:0]	补偿值	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。	RW

具体补偿精度请参考下表：

补偿值设定										补偿值
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

补偿值设定										补偿值
COMPEN	COMP[8:0]									
1	...									...
	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	无补偿

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 × 10⁶

= -11.29ppm

根据公式：

COMP[8:0] = $\left(\frac{\text{补偿目标值 [ppm]} \times 2^{15}}{10^6}\right)_{\text{取2的补码}} + 0b0001.00000$

如果补偿目标值为+20.3ppm，计算相应的寄存器值如下：

COMP[8:0] = (20.3 × 2¹⁵ / 10⁶)_{取2的补码} + 0b0001.00000

= (0.6651904)_{取2的补码} + 0b0001.00000

= 0b0000.10101 + 0b0001.00000

= 0b0001.10101

如果补偿目标值为-20.3ppm，计算相应的寄存器值如下：

COMP[8:0] = (-20.3 × 2¹⁵ / 10⁶)_{取2的补码} + 0b0001.00000

= (-0.6651904)_{取2的补码} + 0b0001.00000

= 0b1111.01011 + 0b0001.00000

= 0b0000.01011

26 看门狗计数器 (WDT/SWDT)

26.1 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。

26.2 主要特性

表 26-1 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

26.3 功能说明

26.3.1 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。

26.3.2 硬件启动方式

ICG0.WDTAUTS/SWDTAUTS 为 0 时，为硬件启动方式。选择硬件启动方式时，WDT/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。下图为硬件启动方式的动作例。

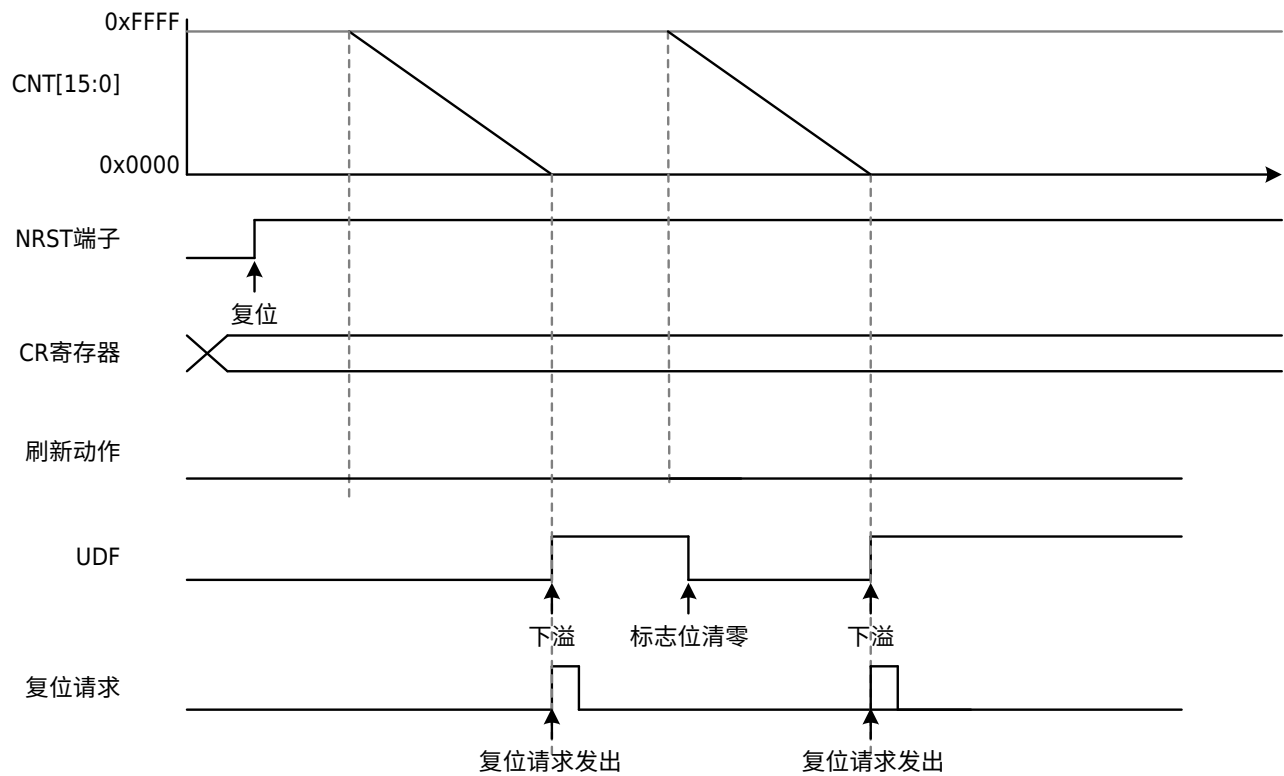


图 26-1 硬件启动例

26.3.3 软件启动方式

ICG0.WDTAUTS/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/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/SWDT_CR 设定只能 1 次，再次设定写入值无效。下图为软件启动方式的动作例。

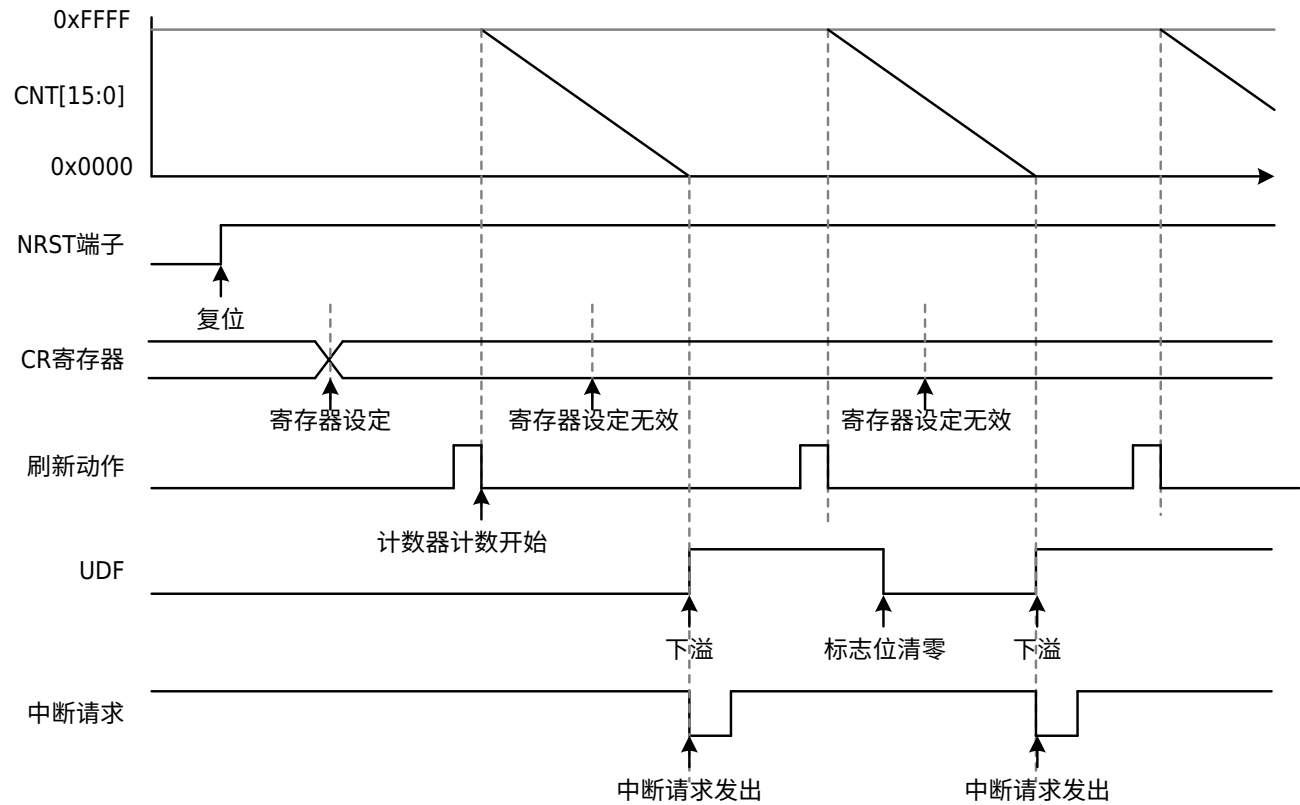


图 26-2 软件启动例

26.3.4 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。

下图为动作示例。

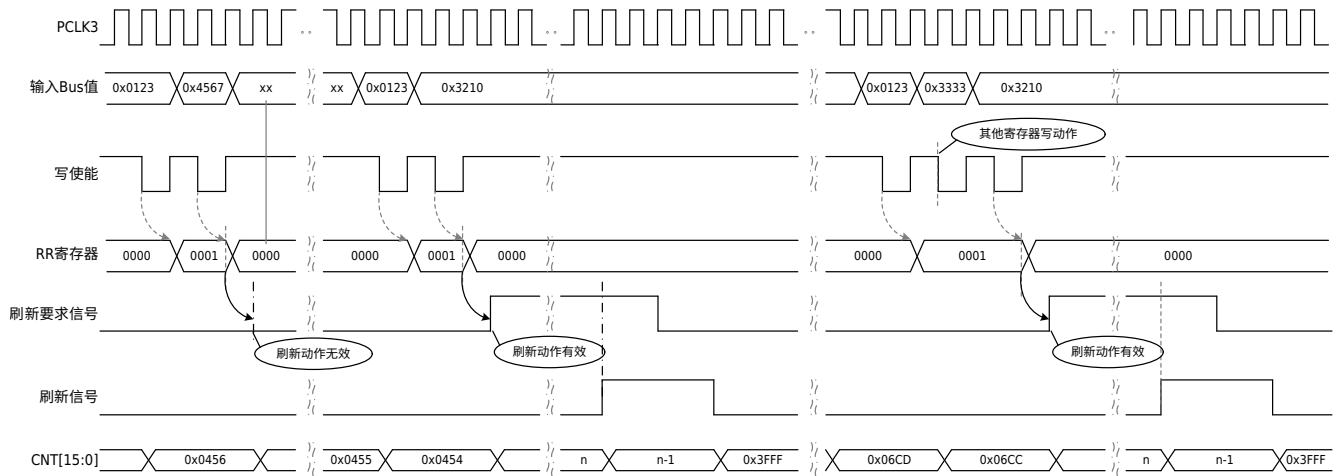


图 26-3 各种刷新动作时序示例（动作确认，刷新要求信号的下降沿等）

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。

26.3.5 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”。

说明

在发生刷新错误或者下溢错误的一定时间内，对标志位清零操作无效，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）。

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。用户对标志位清零后，需再次读取标志位查询清零操作是否成功。

标志位清零步骤：

- 步骤 1. 读取 WDT/SWDT_SR，查询标志位；
- 步骤 2. 对置 1 的标志位写入 0；
- 步骤 3. 读取 WDT/SWDT_SR，查询标志位是否清零成功，若标志位未清零回到步骤 2，直至清零成功。

26.3.6 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.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/SWDT_CR.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。

26.3.7 计数下溢

如下图所示动作例，在递减计数到零时产生下溢。

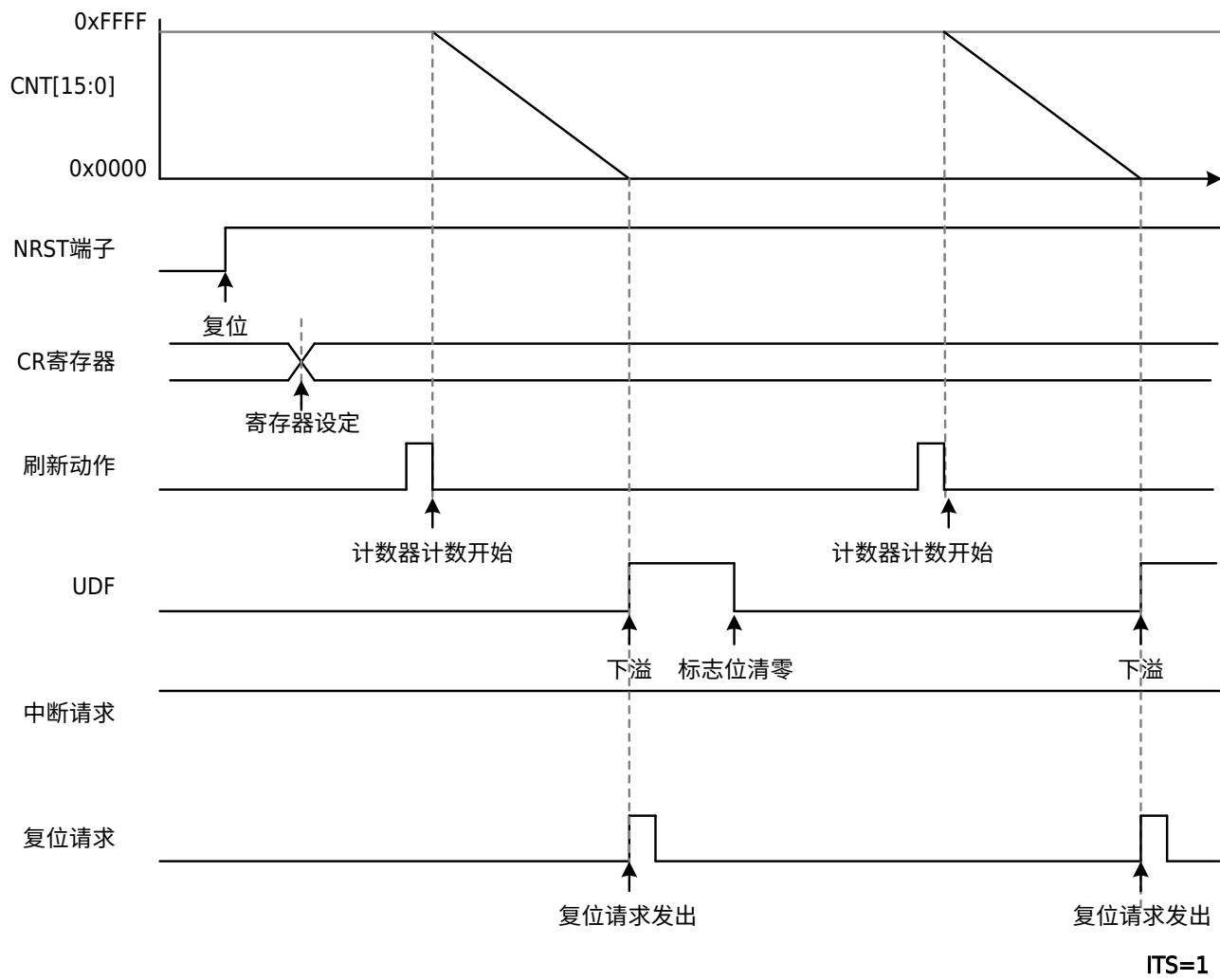


图 26-4 计数器下溢动作例

26.3.8 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。下图为刷新动作例。

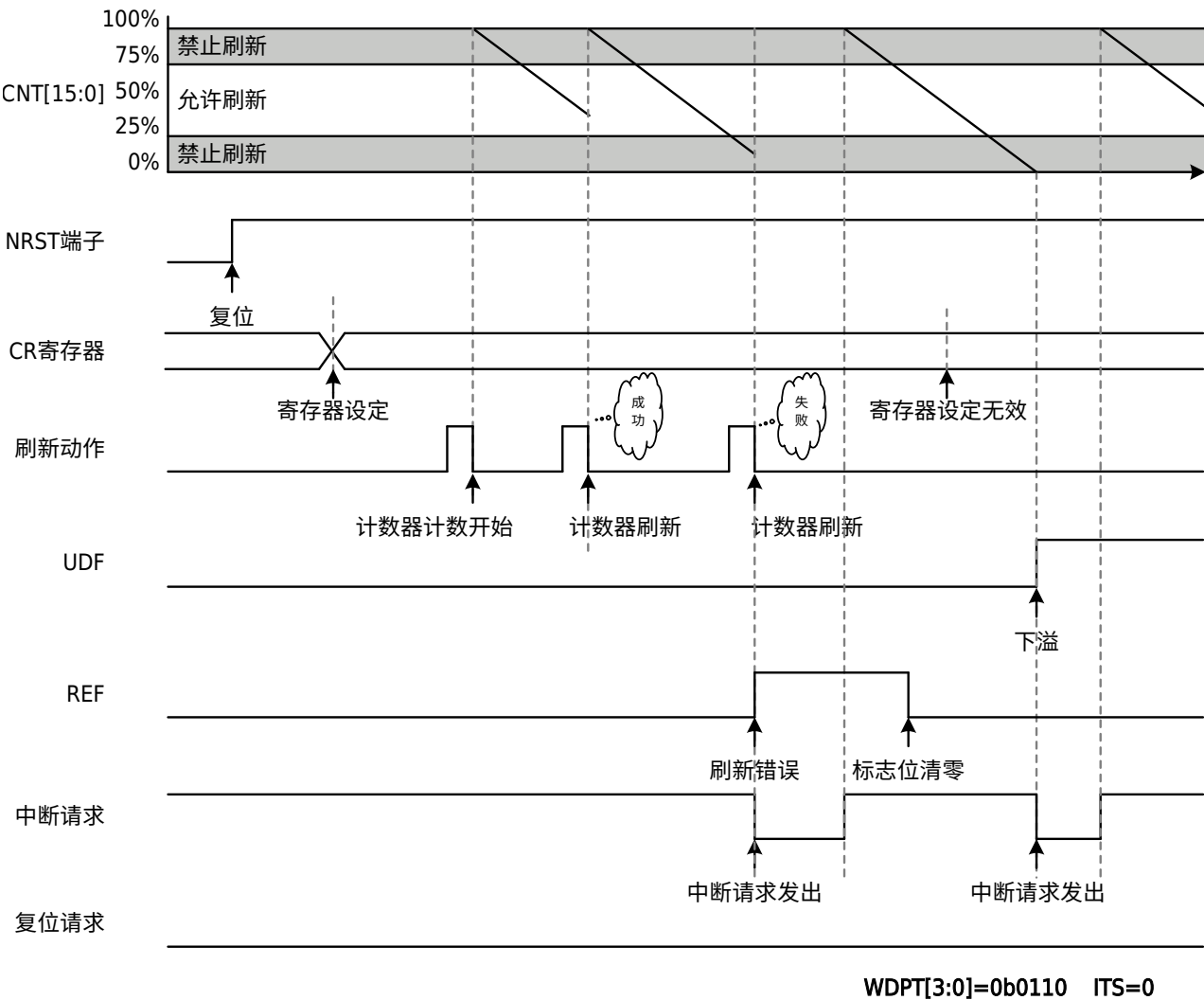


图 26-5 计数器刷新动作例

26.4 寄存器

26.4.1 寄存器总表

WDT 基地址：0x40049000

表 26-2 看门狗计数器 (WDT) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	WDT_CR	WDT 控制寄存器	32
0x04	WDT_SR	WDT 状态寄存器	32
0x08	WDT_RR	WDT 刷新寄存器	32

SWDT 基地址：0x40049400

表 26-3 看门狗计数器 (SWDT) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	SWDT_CR	SWDT 控制寄存器	32
0x04	SWDT_SR	SWDT 状态寄存器	32
0x08	SWDT_RR	SWDT 刷新寄存器	32

26.4.2 控制寄存器 (SWDT_CR、WDT_CR)

Offset	Bit Position																																
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x80010FF3																																
Name	ITS	Reserved														SLPOFF	Reserved						WDPT				CKS				Reserved		PERI
Access	RW															RW							RW				RW						RW

位/位域	标记	位名	功能描述	读写
31	ITS	刷新错误/溢出中断/复位选择	0b0: 中断请求 0b1: 复位请求	RW
30:17	Reserved	保留	-	-
16	SLPOFF	WDT/SWDT 在低功耗模式下计数禁止	WDT_CR: 0b0: WDT 在 Sleep 模式下计数许可 0b1: WDT 在 Sleep 模式下计数禁止 SWDT_CR: 0b0: SWDT 在 Sleep/Stop 模式下计数许可 0b1: SWDT 在 Sleep/Stop 模式下计数禁止	RW
15:12	Reserved	保留	-	-
11:8	WDPT	刷新允许区域计数值百分比	0b0000: 0%~100% 0b0001: 0%~25% 0b0010: 25%~50% 0b0011: 0%~50% 0b0100: 50%~75% 0b0101: 0%~25%, 50%~75% 0b0110: 25%~75% 0b0111: 0%~75% 0b1000: 75%~100% 0b1001: 0%~25%, 75%~100% 0b1010: 25%~50%, 75%~100% 0b1011: 0%~50%, 75%~100% 0b1100: 50%~100% 0b1101: 0%~25%, 50%~100% 0b1110: 25%~100% 0b1111: 0%~100%	RW
7:4	CKS	计数时钟	WDT_CR: 0b0010: PCLK3/4 0b0110: PCLK3/64 0b0111: PCLK3/128 0b1000: PCLK3/256 0b1001: PCLK3/512 0b1010: PCLK3/1024 0b1011: PCLK3/2048 0b1101: PCLK3/8192	RW

位/位域	标记	位名	功能描述	读写
7:4	CKS	计数时钟	- 其他：禁止设置 SWDT_CR： 0b0000：SWDTLRC 0b0100：SWDTLRC/16 0b0101：SWDTLRC/32 0b0110：SWDTLRC/64 0b0111：SWDTLRC/128 0b1000：SWDTLRC/256 0b1011：SWDTLRC/2048 - 其他：禁止设置	RW
3:2	Reserved	保留	-	-
1:0	PERI	计数周期	0b00：256 cycle 0b01：4096 cycle 0b10：16384 cycle 0b11：65536 cycle	RW

26.4.3 状态寄存器 (SWDT_SR、WDT_SR)

Offset	Bit Position																																					
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	Reserved																REF		UDF		CNT																	
Access																	RW	RW																				

位/位域	标记	位名	功能描述	读写
31:18	Reserved	保留	-	-
17	REF	刷新错误标志	0b0：没有刷新错误 0b1：发生刷新错误 对该位读出 1 后写入 0，该位清零。	RW
16	UDF	计数下溢标志	0b0：没有计数下溢 0b1：发生计数下溢 对该位读出 1 后写入 0，该位清零。	RW
15:0	CNT	计数值	计数器当前计数值	RO

26.4.4 刷新寄存器 (SWDT_RR、WDT_RR)

Offset	Bit Position																															
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																RF															
Access																																

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	RF	刷新值	依次写入 0x0123、0x3210 之后，完成刷新动作。 写入 0x0123 后，该寄存器读出于 0x00000001；其 余情况读出于 0x00000000。	RW

26.5 注意事项

SWDT 动作时，时钟 PCLK3 的动作频率必须大于或等于计数时钟频率的 4 倍，即 PCLK3 频率≥计数时钟频率×4。

27 通用同步异步收发器 (USART)

27.1 简介

本产品搭载通用同步异步收发器 (USART) 模块 6 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。与 Timer0 模块配合支持 UART 接收超时功能。USART1 支持通过 RX 线唤醒 STOP 模式功能。

27.2 主要特性

功能特性	模块支持情况					
	USART1	USART2	USART3	USART4	USART5	USART6
UART	✓	✓	✓	✓	✓	✓
多处理器通信	✓	✓	✓	✓	✓	✓
时钟同步通信	✓	✓	✓	✓	✓	✓
RX 线唤醒 STOP 模式功能	✓	-	-	-	-	-
小数波特率	✓	✓	✓	✓	✓	✓
LIN	-	-	✓	-	-	✓
智能卡	✓	✓	-	✓	✓	-
UART 接收超时功能	✓	✓	-	✓	✓	-

USART 主要特性

- 支持全双工异步通信，全双工时钟同步通信
- 支持 LIN 总线
- 支持智能卡接口 (ISO/IEC7816-3)
- 发送器和接收器具有独立使能位
- 内置双缓冲器
- LSB/MSB 可选
- 支持调制解调器操作 (CTS/RTS)
- 传输标志：发送数据寄存器空、发送数据完成、接收数据寄存器满、接收错误标志、UART 接收超时标志、LIN 唤醒信号检出标志、LIN 间隔段检出标志、LIN 总线错误标志

UART 主要特性

- 数据长度可编程：8 位/9 位
- 校验功能可配置：奇校验/偶校验/无校验
- 停止位可配置：1 位/2 位
- 时钟源可选：内部时钟源 (内部波特率生成器生成的时钟) /外部时钟源 (USARTn_CK 管脚输入的时钟)
- 接收错误：校验错误、帧错误、上溢错误
- 支持多个处理器间通信
- 内置数字滤波器

- 支持接收数据 TIMEOUT 功能
- 单元 1 支持停止模式唤醒功能
- 支持全双工/半双工通信方式

时钟同步模式主要特性

- 数据长度：8 位
- 接收错误：上溢错误
- 时钟源：内部时钟源（内部波特率生成器生成的时钟）/外部时钟源（USARTn_CK 管脚输入的时钟）
- 支持全双工通信方式

智能卡接口的主要特性

- 数据长度：8 位
- 检测到校验错误时能自动送出错误信号
- 支持数据重发

LIN 的主要特性

- 数据长度：8 位
- 支持唤醒信号的检测
- 支持 10/11 位同步间隔段（BF：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

27.3 功能说明

27.3.1 系统框图

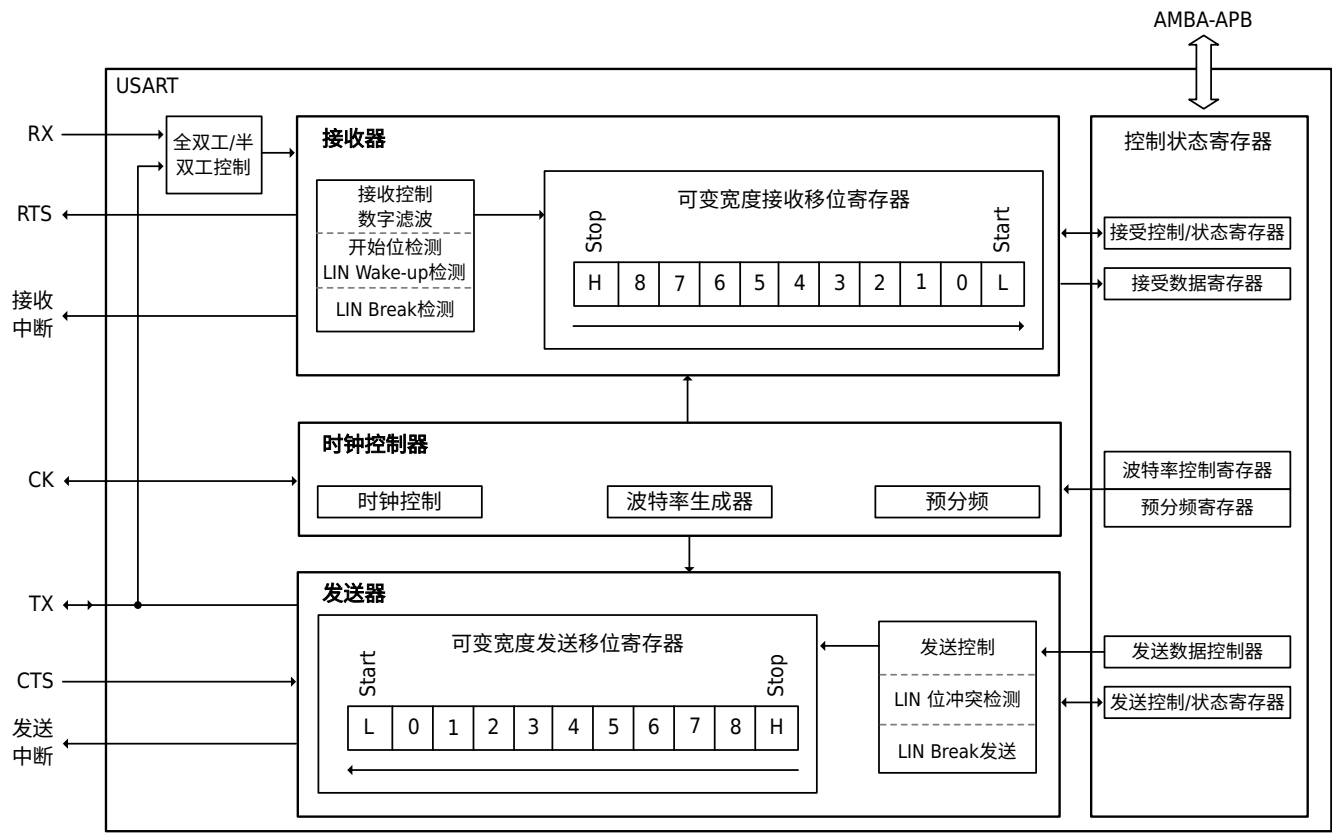


图 27-1 USART 系统框图

表 27-2 USART 管脚说明

管脚名	方向	功能描述
USARTn_CK	输入/输出	时钟
USARTn_TX	输入/输出	发送数据管脚 半双工时也作为接收数据管脚
USARTn_RX	输入	接收数据管脚
USARTn_CTS	输入	调制解调器操作管脚 清除发送管脚
USARTn_RTS	输出	调制解调器操作管脚 请求发送管脚

 **说明**
n 取值为 1~6。

27.3.2 UART 功能

27.3.2.1 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 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=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=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 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 位的设定选择为 PCLK1/128，PCLK1/256，PCLK1/512，PCLK1/1024。

外部时钟源

USARTn_CR2.CLKC 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：

$$B = \frac{C}{8 \times (2 - \text{OVER8}) \times (\text{DIV\_Integer} + 1)}$$

其中：

- B：波特率，单位：Mbps
- C：当 USARTn_PR.ULBREN 为 0 时，USARTn_PR.PSC 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 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（MHz），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 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。

27.3.2.2 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。

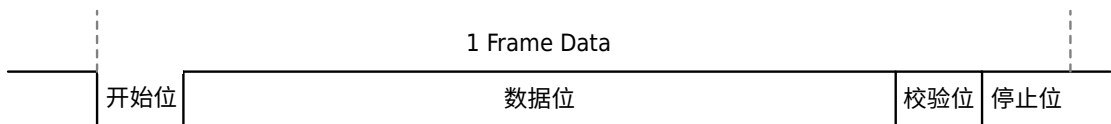


图 27-2 UART 数据格式

- 开始位（ST）：开始位固定由一位的低电平构成。
- 数据位（D）：数据位可以配置成 8 位或者 9 位。
- 校验位（P）：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 (SP)：** 停止位固定为高电平，可以配置成 1 位或者 2 位。

27.3.2.3 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)，且不正在接收数据
- 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

27.3.2.4 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

- 步骤 1. 将 USARTn_CR1 寄存器设定为复位值；
- 步骤 2. 设定 UART 所需要使用的管脚；
- 步骤 3. 通过 USARTn_CR2.CLKC 位选择时钟源；
- 步骤 4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器；
- 步骤 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）；
- 步骤 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；
- 步骤 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；



说明

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。

- 步骤 8. 如果需要连续发送数据时，重复步骤 7；
- 步骤 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。

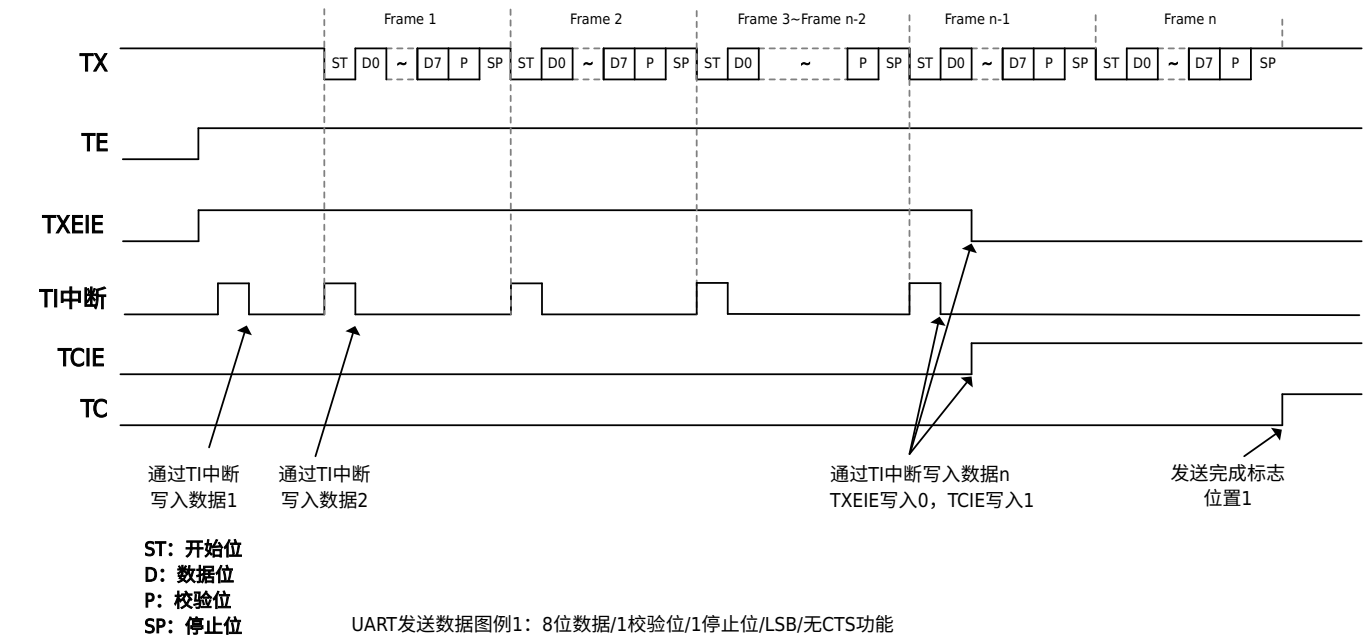


图 27-3 UART 发送数据图例 1

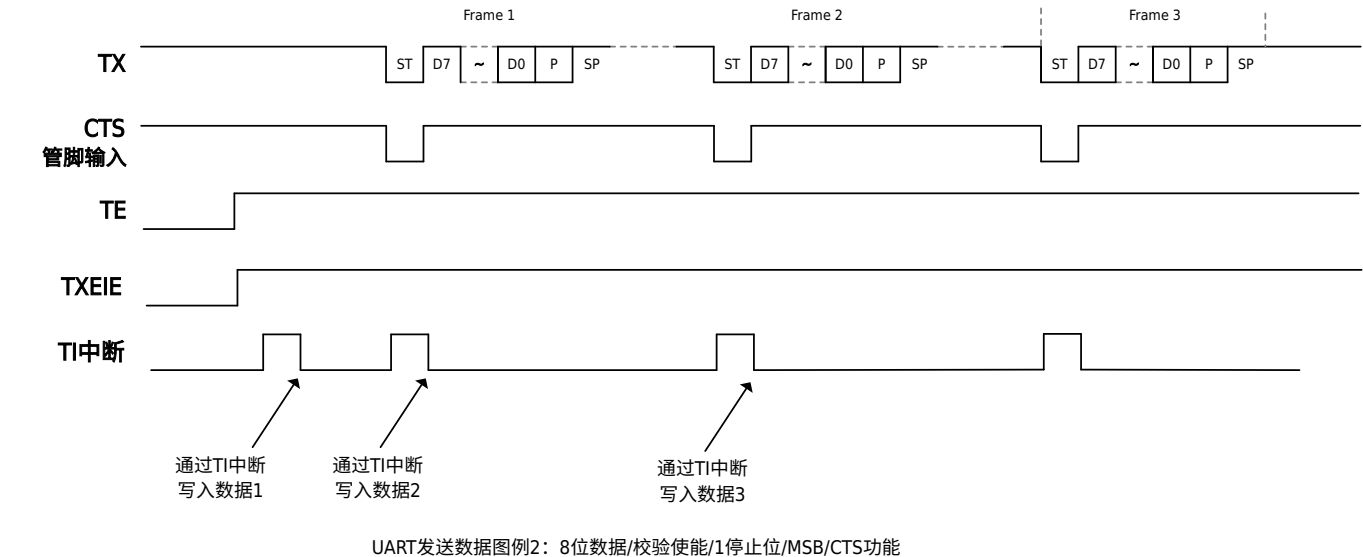


图 27-4 UART 发送数据图例 2

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。

27.3.2.5 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。

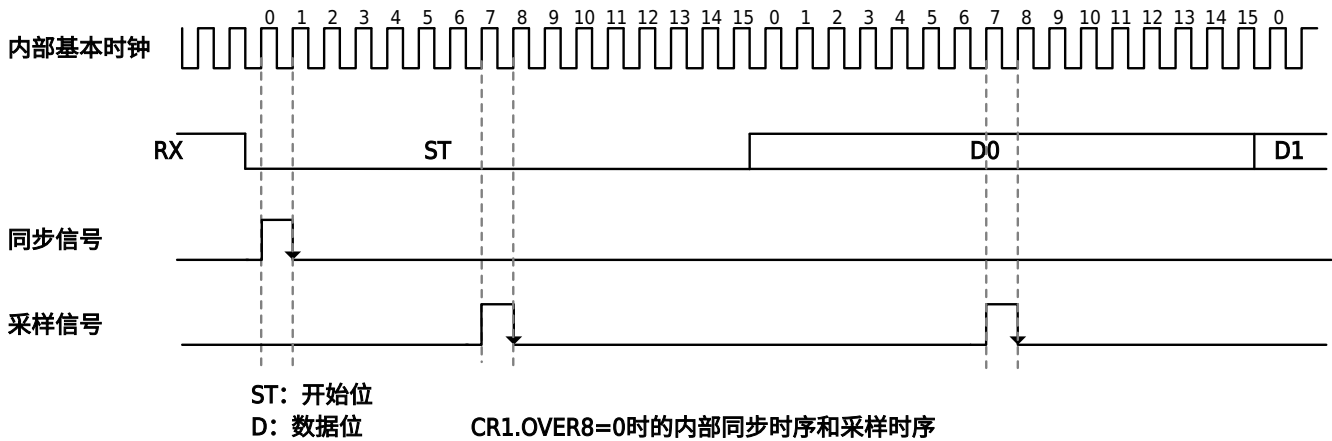


图 27-5 UART 内部同步和采样时序

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 USARTn_CR1.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 USARTn_CR1.OVER8 位定义的 8 倍或 16 倍过采样
- 由 USARTn_CR1.FBME 位定义的是否使用小数波特率

表 27-3 UART 接收器的容差

FL	DIV_Fraction 为 0		DIV_Fraction 不为 0	
	OVER8 位=0	OVER8 位=1	OVER8 位=0	OVER8 位=1
10	4.375%	3.75%	3.88%	3%
11	3.97%	3.41%	3.53%	2.73%
12	3.646%	3.125%	3.23%	2.5%

 **说明**
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，上表中指定的数据可能会略微不同。

接收数据设定步骤

- 步骤 1. 将 USARTn_CR1 寄存器设定为复位值；
- 步骤 2. 设定 UART 所需要使用的管脚；
- 步骤 3. 通过 USARTn_CR2.CLKC 位选择时钟源；
- 步骤 4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器；
- 步骤 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）；
- 步骤 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；
- 步骤 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

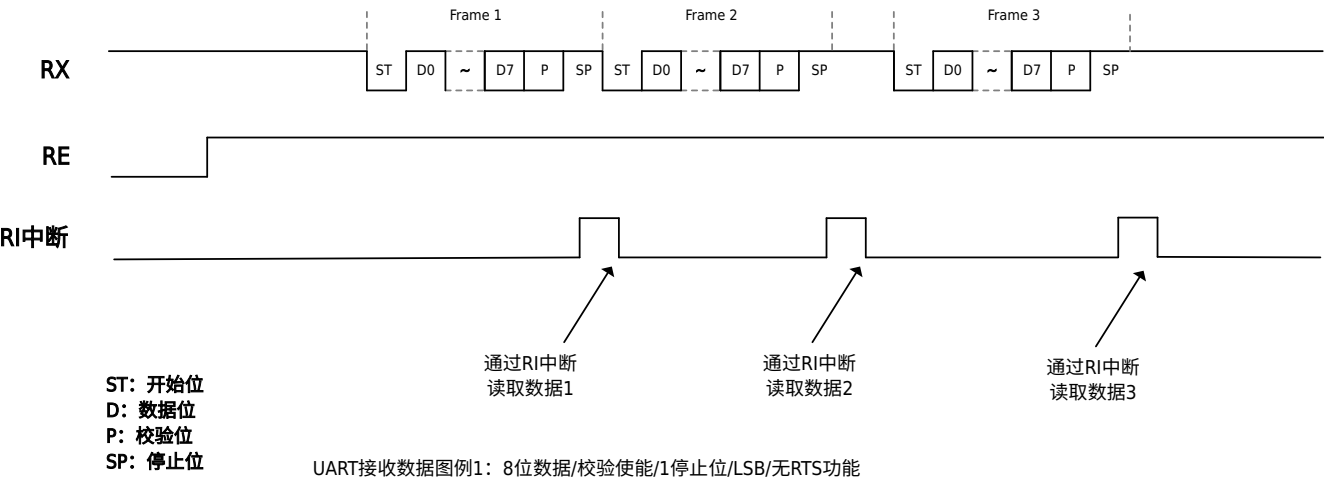


图 27-6 UART 接收数据图例 1

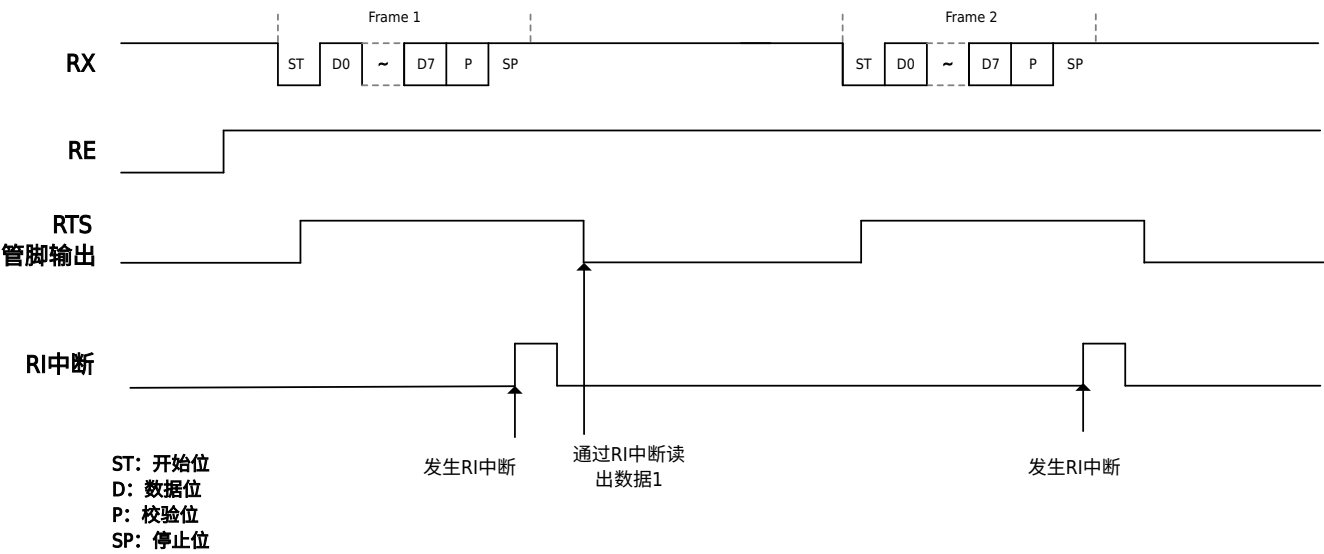


图 27-7 UART 接收数据图例 2

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_CR1.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_CR1.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。

27.3.2.6 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 USARTn_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，Timer0 计数方式需设置为异步计数方式，异步计数时钟源设置为 XTAL32（Timer0 计数时钟由 UART 波特率生成器产生的时钟 tick 提供，可以不存在 XTAL32 晶振），具体对应关系如下：

USART1	USART2	USART4	USART5
Timer0_1 A 通道	Timer0_1 B 通道	Timer0_2 A 通道	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：

$$CMPAR = \frac{RTB}{2^{CKDIRA}} - \alpha$$

其中：

- 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迟延
计数时钟不分频时，α=7；
计数时钟 2 分频时，α=5；
计数时钟 4、8、16 分频时，α=3；
计数时钟 32 分频及以上，α=2。
- RTB：Receive Timeout Bits，最小值=接收数据帧长度+α×2^{CKDIRA}
实际发生 TIMEOUT 的时间与 RTB 值之间有一定误差，误差≤2^{CKDIRA}。
- CKDIRA：TMR0_BCONR.CKDIRA位寄存器值

TIMEOUT 功能设定步骤

- 步骤 1. 将 USARTn_CR1 寄存器设定为复位值；
- 步骤 2. 设定 UART 所需要使用的管脚；
- 步骤 3. 通过 USARTn_CR2.CLKC 位选择时钟源（如果选择内部时钟源时需设置 CR2.CLKC=0b01）；
- 步骤 4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器；
- 步骤 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）；
- 步骤 6. USARTn_CR1.RTOE=1, 如果需要使用中断, 则设定 USARTn_CR1.RTOIE=1；
- 步骤 7. 设置 TMR0_BCONR.CSTA=0；
- 步骤 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；
- 步骤 9. 设置 TMR0_BCONR.HCLEA=1, TMR0_BCONR.HSTAA=1, TMR0_BCONR.ASYNCLKA=1, TMR0_BCONR.SYNCLKA=0, TMR0_BCONR.SYNSA=1；
- 步骤 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；
- 步骤 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_BCONR.SYNSA=1, 通过写 USARTn_CR1.CRTOF 清除 USARTn_SR.RTOF 状态位。

27.3.2.7 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:

- 步骤 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；
 - 步骤 2. 系统进入停止模式；
 - 步骤 3. 系统检测到 RX 线下降沿时, 从停止模式返回, 在 USART_1_WUPI 中断处理程序中关闭该功能。
- 需要注意的是, 当通信方需要唤醒本系统时, 需要发送一帧唤醒数据（建议为 0x00）, 该数据不会被 UART 接收且不置位相关的标志。并且通信方需要经过系统停止模式唤醒需要的时间后再进行 UART 通信数据的传输。

UART RX 线唤醒功能, 可以对 RX 线上的噪声进行过滤, 详细信息请参考 [USART1 滤波控制寄存器 \(USART1_NFC\)](#)。

27.3.2.8 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。
- 无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。

27.3.2.9 中断和事件

表 27-5 UART 中断/事件表

功能名称	使能位（仅中断）	标志位	可否作为事件源
接收错误中断	RIE	ORE、FE、PE	可
接收数据寄存器满中断	RIE	RXNE	可
发送数据寄存器空中断	TXEIE	TXE	可
发送完成中断	TCIE	TC	可
发送完成中断	TENDIE	TEND	不可
TIMEOUT 中断	RTOIE	RTOF	可
RX 线唤醒停止模式中断	INTC_WKEN.RXWKEN	-	不可

27.3.3 多处理器通信

27.3.3.1 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。

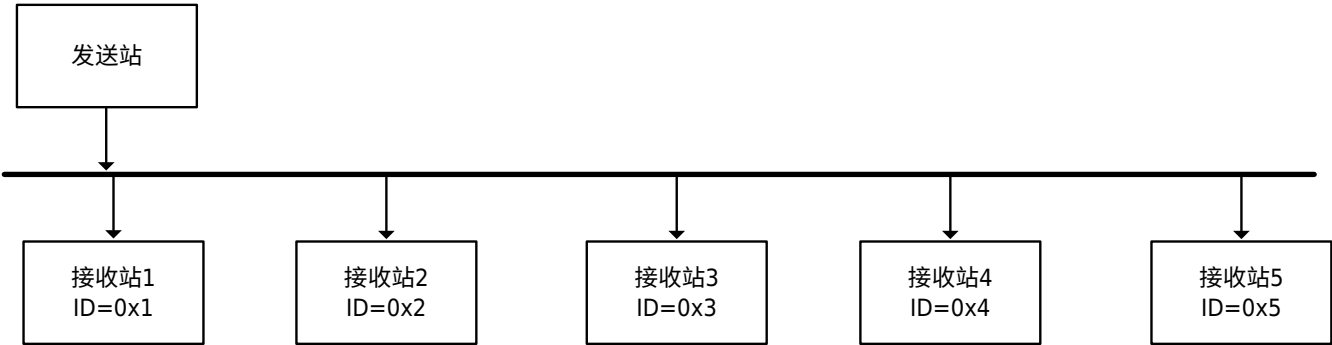


图 27-8 多处理器通信图例

27.3.3.2 数据格式

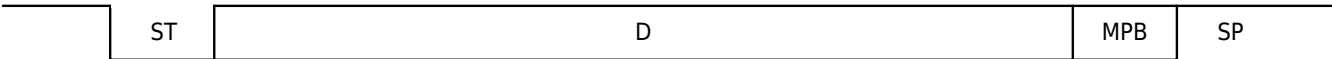


图 27-9 多处理器模式数据格式

- **开始位 (ST)：**开始位固定由一位的低电平构成。
- **数据位 (D)：**数据位可以配置成 8 位或者 9 位。
- **多处理器位 (MPB)：**多处理器位可以配置成发送通信数据或者发送 ID。
- **停止位 (SP)：**停止位固定为高电平，可以配置成 1 位或者 2 位。

27.3.3.3 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

- 步骤 1. 将 USARTn_CR1 寄存器设定为复位值；
- 步骤 2. 设定所需要使用的管脚；
- 步骤 3. 通过 USARTn_CR2.CLKC 位选择时钟源；
- 步骤 4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器；
- 步骤 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）；
- 步骤 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；
- 步骤 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；

 **说明**
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。

- 步骤 8. 设定 USARTn_TDR.MPID 位为 0（发送数据），写数据到 USARTn_TDR，发送数据；

 **说明**
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。

- 步骤 9. 如果需要连续发送数据，重复步骤 8，如果需要改变 ID 后再发送数据重复步骤 7 和 8；
- 步骤 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。

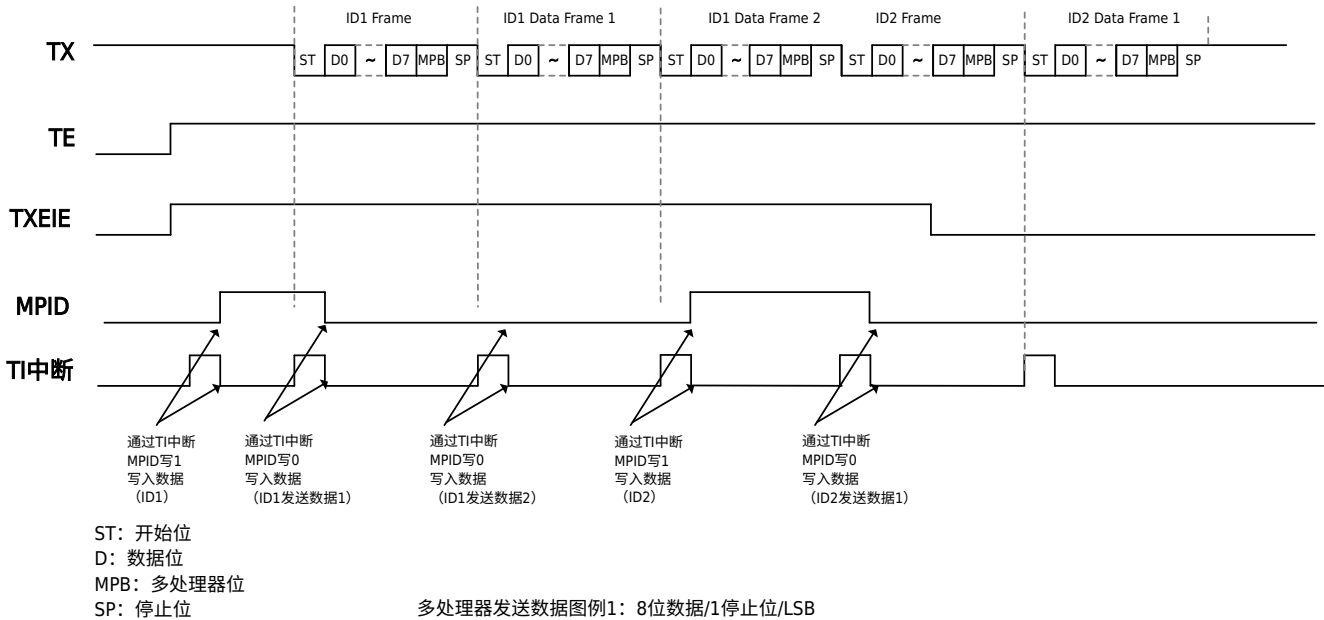


图 27-10 多处理器模式发送数据图例

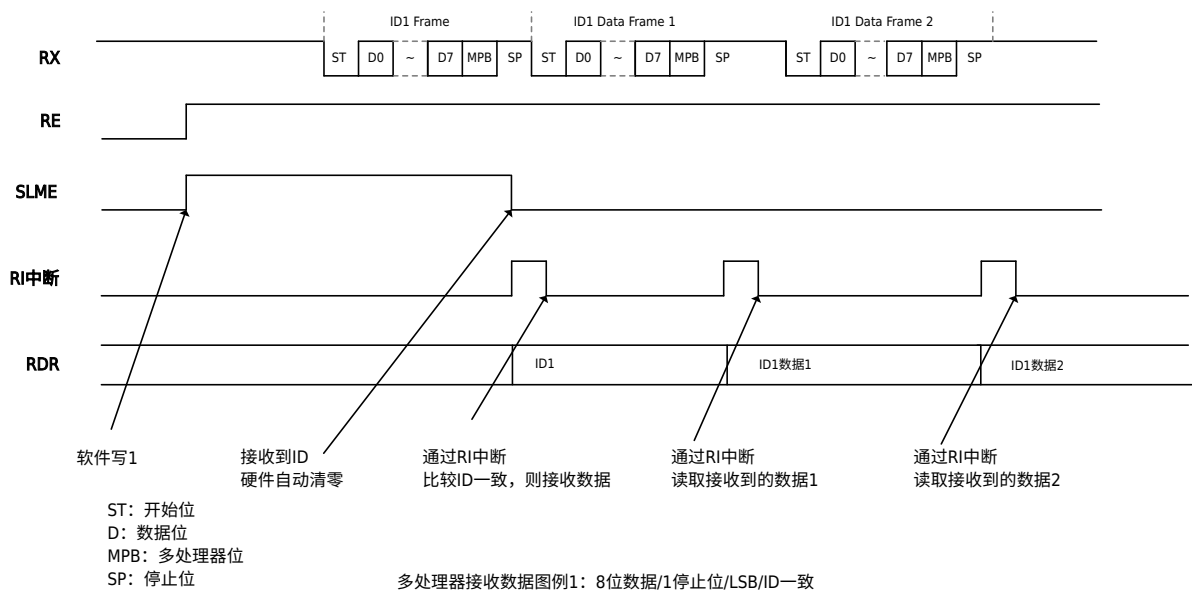
接收站动作

在多处理器模式时，接收站必须保证能收到每一个 ID 数据，并与自身的 ID 比较，如果一致则接收数据，不一致则进入静默模式（不接收数据，也不置位接收相关标志），直到接收到下一个 ID 数据。通过 USARTn_CR1.SLME 位来实现这个功能。
USARTn_CR1.SLME=0 时正常接收数据。

USARTn_CR1.SLME=1 时，除非接收到 MPB 位为 1（ID）的数据，否则不接收数据，不发生 RI 中断，错误标志 FE，ORE 也不置位。当接收到 MPB 位为 1 的数据时（ID），USARTn_CR1.SLME 位自动清零，正常接收数据和发生中断。

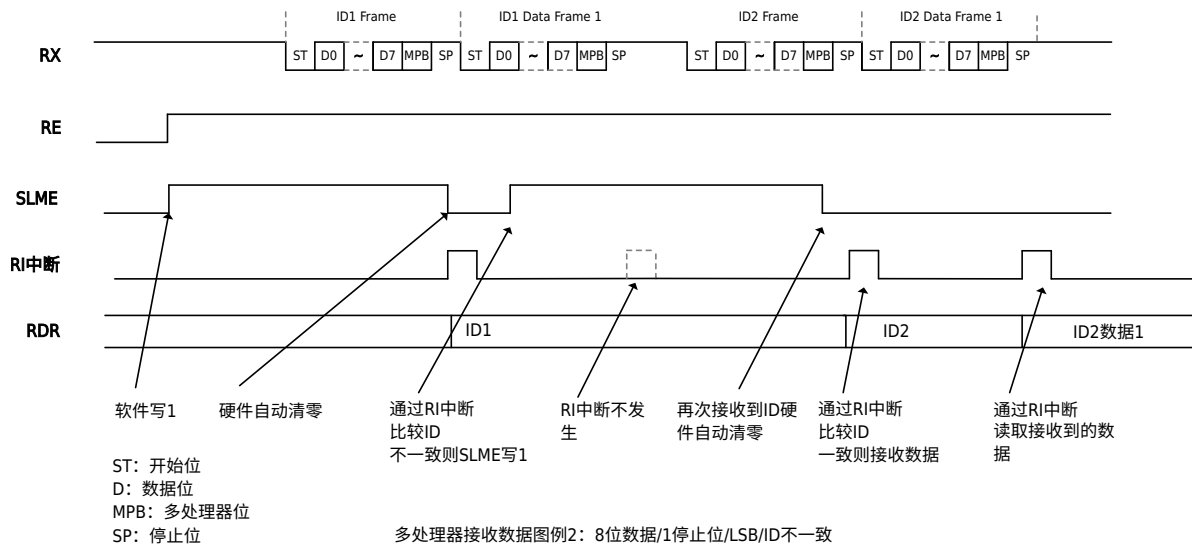
动作步骤：

- 步骤 1. 将 USARTn_CR1 寄存器设定为复位值；
- 步骤 2. 设定所需要使用的管脚；
- 步骤 3. 通过 USARTn_CR1.CLKC 位选择时钟源；
- 步骤 4. 设定 USARTn_CR1，USARTn_CR2，USARTn_CR3 寄存器；
- 步骤 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）；
- 步骤 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；
- 步骤 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；
- 步骤 8. 如果 USARTn_SR.MPB=1，USARTn_CR1.SLME 位自动清零，正常接收数据，软件比较接收的 ID 与自身的 ID。
 - 如果 ID 一致，则正常接收数据，发生中断，进行错误检测，与 UART 接收数据相同；
 - 如果 ID 不一致，软件再次将 USARTn_CR1.SLME 位写 1，重复步骤 8 的动作。



多处理器接收数据图例1：8位数据/1停止位/LSB/ID一致

图 27-11 多处理器模式接收数据图例 1



多处理器接收数据图例2：8位数据/1停止位/LSB/ID不一致

图 27-12 多处理器模式接收数据图例 2

27.3.3.4 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 27-6 多处理器模式中中断/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

27.3.4 UART-LIN

27.3.4.1 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。

27.3.4.2 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。

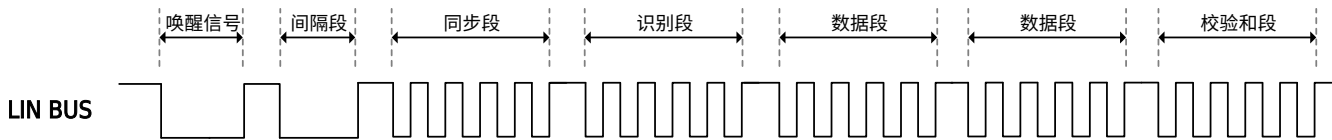


图 27-13 LIN 总线数据行为

27.3.4.3 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。

27.3.4.4 LIN 接收

唤醒信号帧的检测

USARTn_CR2.WKUPE=1 时，硬件自动检测唤醒信号，检测到 RX 线低电平宽度上大于等于 2.5 位数据宽度（19.2kbps 时为 130μs）时，置位 USARTn_SR.WKUP 标志，USARTn_CR2.WKUPE=1 时，产生对应的中断。检测到唤醒信号帧后需要清除 USARTn_SR.WKUP 标志，并读取 USARTn_RDR.RDR 寄存器。

USARTn_CR2.WKUPE 寄存器仅在系统需要等待唤醒信号时设定为 1，其他情况需要设定为 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

- 步骤 1. 将 USARTn_CR1 寄存器设定为复位值；
- 步骤 2. 设定 UART 所需要使用的管脚；
- 步骤 3. 通过 USARTn_CR2.CLKC 位选择时钟源；
- 步骤 4. 设定 USARTn_CR1（RE=1，RIE=0），USARTn_CR2，USARTn_CR3 寄存器；
- 步骤 5. （可选）唤醒信号帧的检测：
 - a. 设定 USARTn_CR2.WKUPE=1，等待唤醒信号；
 - b. 检测到唤醒信号，USARTn_CR2.WKUPE=0，等待同步段间隔段。
- 步骤 6. （可选）同步段间隔场检测：
 - a. 检测到同步间隔段后，清除 USARTn_SR.FE/LBD 标志，并确认接收到的数据为 0x00。
- 步骤 7. （可选）同步段频率测量与通信波特率计算：
 - a. 如果需要使用接收中断，则设置 USARTn_CR1.RIE=1；

- b. 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。接收数据若为 0x55，说明波特率误差在可接受范围内；若接收数据不为 0x55，用户校准波特率为必须步骤，用户根据 USARTn_LBMC 寄存器来重新校准波特率。

- 步骤 8. 接收识别段（过程与 UART 接收数据过程相同）；
- 步骤 9. 接收数据（过程与 UART 接收数据过程相同）；
- 步骤 10. 接收校验和（过程与 UART 接收数据过程相同）。

27.3.4.5 中断和事件

表 27-7 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
	LBDIE	LBD	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

27.3.5 智能卡

27.3.5.1 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。

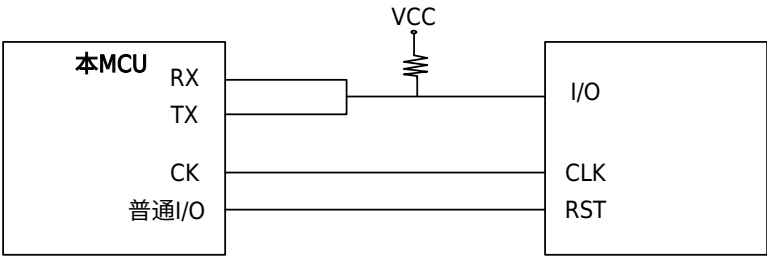


图 27-14 智能卡连接示意图

27.3.5.2 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 设定值。
通过设定寄存器 USARTn_CR2.CLKC 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：

$$B = \frac{C}{2 \times BCN \times (DIV\_Integer + 1)}$$

其中：

- B: 波特率, 单位: Mbps
 - C: 当 USARTn_PR.ULBREN 为 0 时, USARTn_PR.PSC 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 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
 - 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。

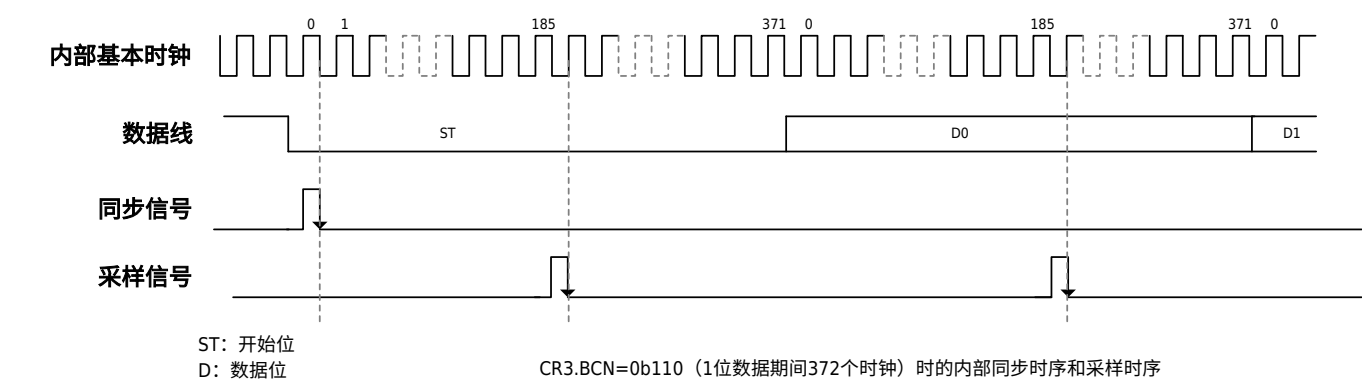


图 27-15 智能卡模式同步时序和采样时序图

接收容差计算公式:

$$RM = \{0.5 \times (1 - \frac{1}{BCN}) - 9.5CFD\} \times 100\%$$

其中:

- RM: 接收容差
- BCN: 一位数据传输所需要的时钟数 (USARTn_CR3.BCN 设定值)
- CFD: 时钟频率偏差

27.3.5.3 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。

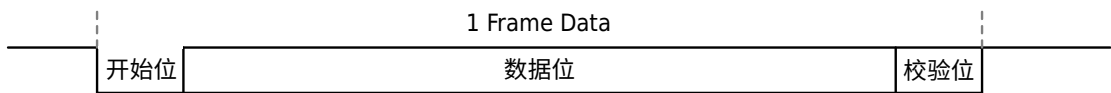


图 27-16 智能卡模式数据格式

- 起始位 (ST) : 起始位固定由一位的低电平构成。
- 数据位 (D) : 数据位固定为 8 位数据。
- 校验位 (P) : 校验位需配置成 1 位偶校验。

27.3.5.4 智能卡的初始化设定步骤

- 步骤 1. 将 USARTn_CR1 寄存器设定为复位值;
- 步骤 2. 设定所需要使用的管脚;
- 步骤 3. 状态寄存器确认, USARTn_SR 寄存器设定为复位值;
- 步骤 4. 设定 USARTn_CR1、USARTn_CR2、USARTn_CR3 寄存器;

- 步骤 5. 设定 USARTn_PR 选择预分频值，USARTn_BRR 寄存器设定通信波特率；
- 步骤 6. USARTn_CR2.CLKC 位设定时钟控制；
- 步骤 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。

27.3.5.5 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_CR1.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。

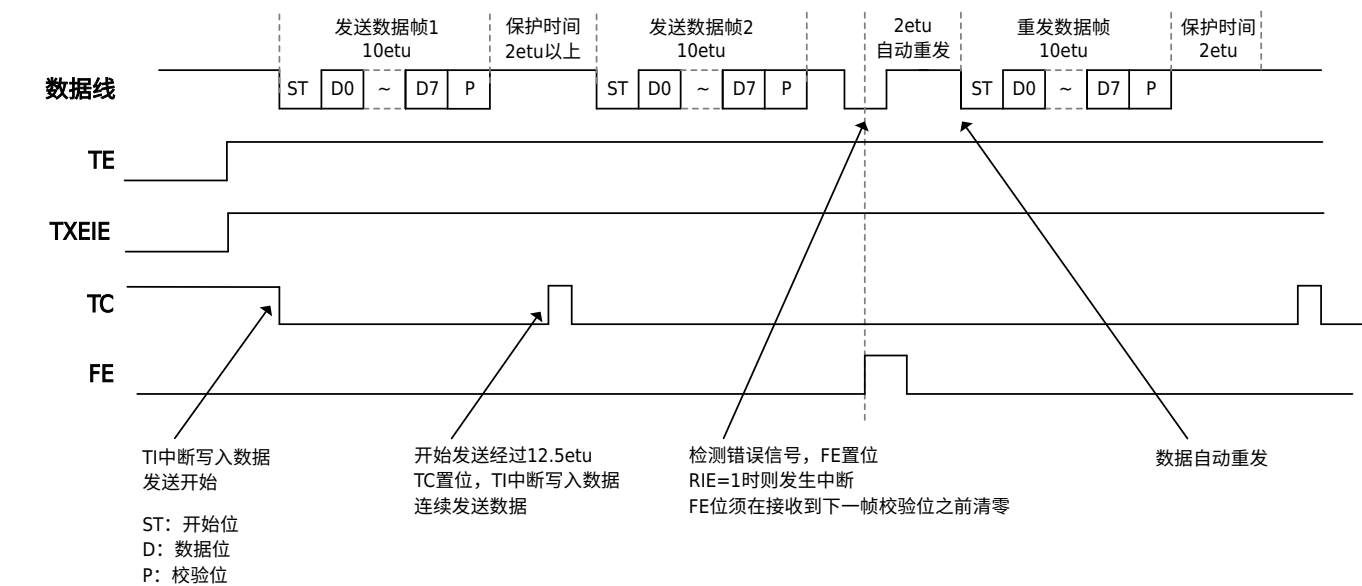


图 27-17 智能卡模式发送数据图例

接收说明

- 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。
- 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。
- 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。
- 接收数据时，检测上溢错误。

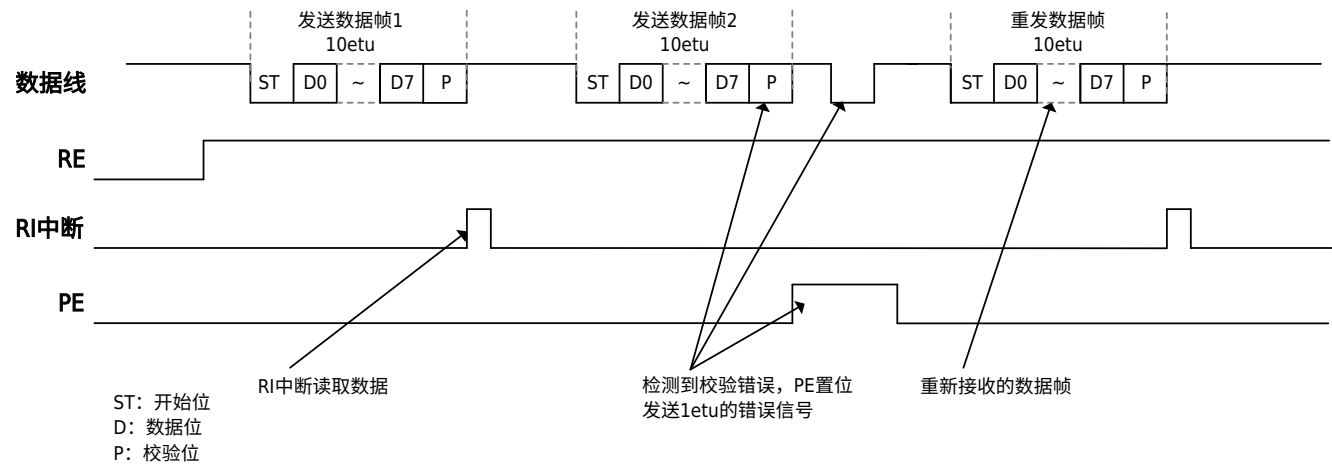


图 27-18 智能卡模式接收数据图例

27.3.5.6 中断和事件

表 27-8 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

27.3.6 时钟同步模式

27.3.6.1 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：

$$B = \frac{C}{4 \times (DIV\_Integer + 1)}$$

其中：

- B：波特率，单位：Mbps
- C：当 USARTn_PR.ULBREN 为 0 时，USARTn_PR.PSC 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 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 $PCLK1/6$ (MHz)，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。

27.3.6.2 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。

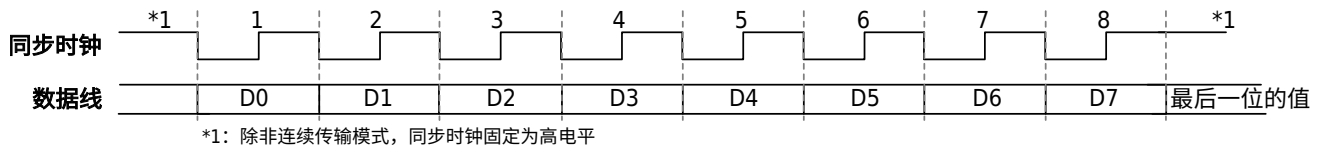


图 27-19 时钟同步模式数据格式

27.3.6.3 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。

27.3.6.4 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

- 步骤 1. 将 USARTn_CR1, USARTn_SR1 寄存器设定为复位值；
- 步骤 2. 设定所需要使用的管脚；
- 步骤 3. 通过 USARTn_CR2.CLKC 位选择时钟源；
- 步骤 4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器；
- 步骤 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）；

- 步骤 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；
- 步骤 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；
- 说明

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。
- 步骤 8. 如果需要连续发送数据时，重复步骤 7；
- 步骤 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。

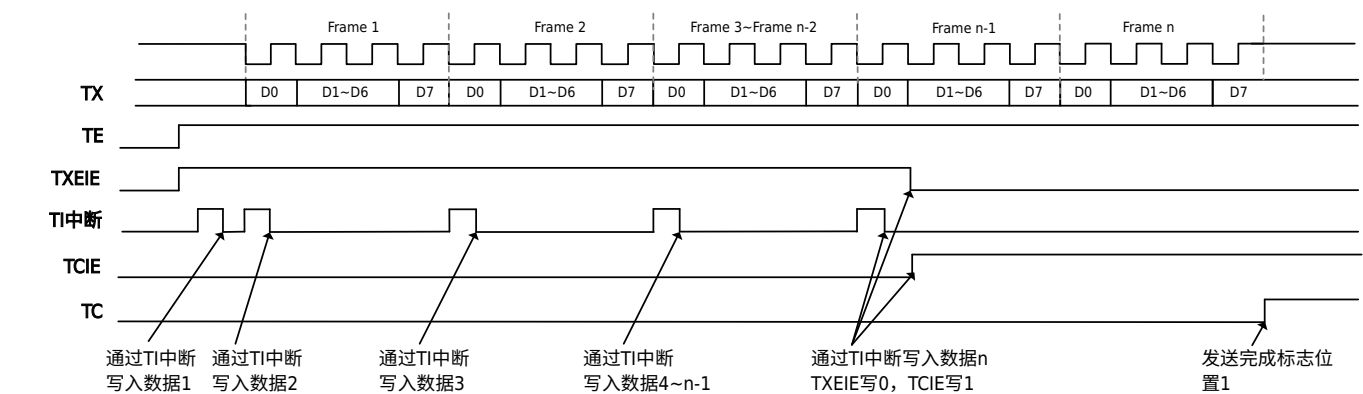


图 27-20 时钟同步模式发送数据图例 1 (LSB/无 CTS 功能)

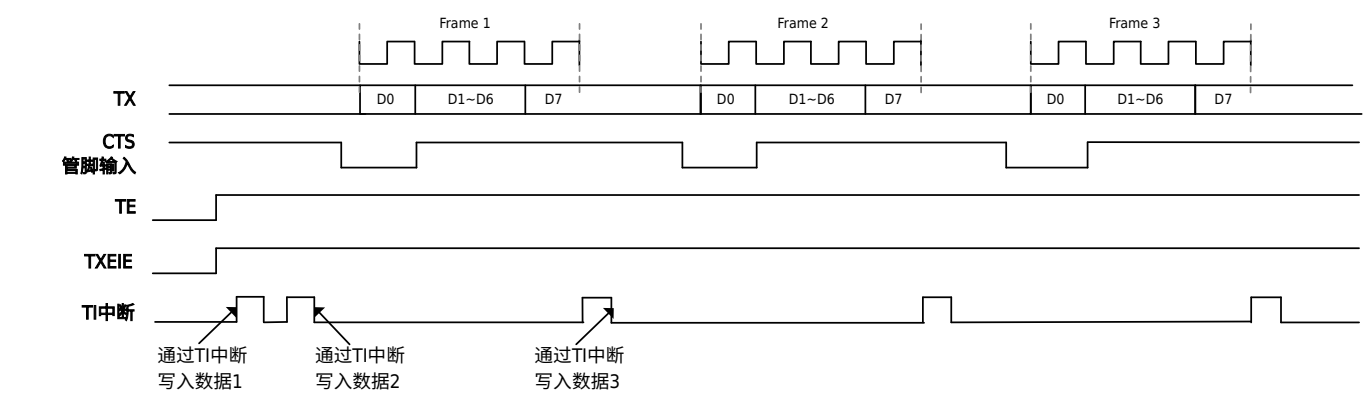


图 27-21 时钟同步模式发送数据图例 2 (LSB/CTS 功能)

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。

27.3.6.5 接收器

接收数据设定步骤

- 步骤 1. 将 USARTn_CR1，USARTn_SR 寄存器设定为复位值；
- 步骤 2. 设定所需要使用的管脚；
- 步骤 3. 通过 USARTn_CR2.CLKC 位选择时钟源；
- 步骤 4. 设定 USARTn_CR1，USARTn_CR2，USARTn_CR3 寄存器；

- 步骤 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）；
- 步骤 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；



说明

使用 RTS 功能时，RE=1 后 USARTn_RTS 输出低电平。

- 步骤 7.** 同步于输入同步时钟或者内部生成的同步时钟开始接收数据，接收数据到接收移位寄存器。
- 发生上溢错误时，数据丢失并置位 USARTn_SR.ORE 标志；
 - 无错误发生时，接收到的数据传送到 USARTn_RDR.RDR 寄存器中，置位 USARTn_SR.RXNE 标志，将当前接收到的数据在接收到下一帧数据最后一位前读取后重复步骤 7 可实现连续接收数据功能。



说明

使用 RTS 功能时，数据读取后 USARTn_RTS 输出低电平。

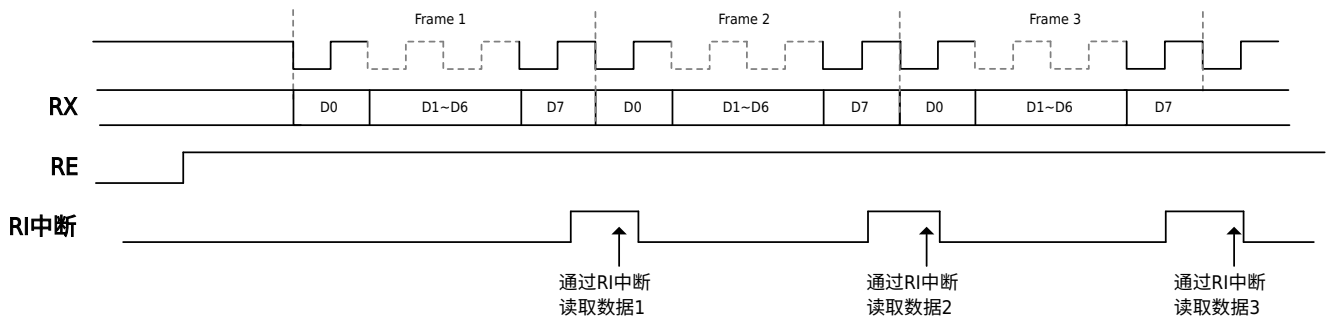


图 27-22 时钟同步模式接收数据图例 1 (LSB/无 RTS 功能)

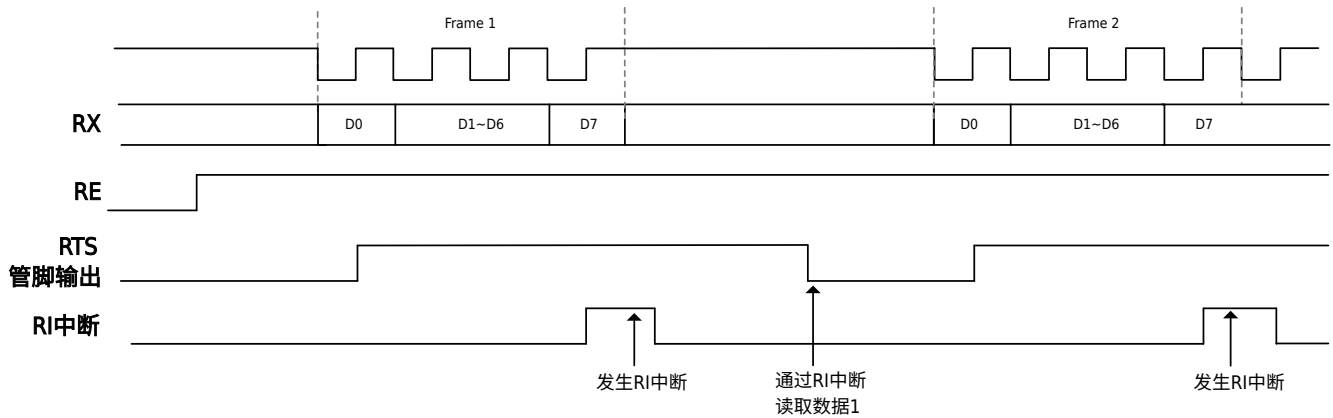


图 27-23 时钟同步模式接收数据图例 2 (LSB/RTS 功能)

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。

27.3.6.6 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，其他设定流程与发送器和接收器相同。

27.3.6.7 中断和事件

表 27-9 时钟同步模式中断/事件表

中断名称	使能位（仅中断）	标志位	可否作事件源
接收错误中断	RIE	ORE	可
接收数据寄存器满中断	RIE	RXNE	可
发送数据寄存器空中断	TXEIE	TXE	可
发送完成中断	TCIE	TC	可
发送完成中断	TENDIE	TEND	不可

27.3.7 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。

27.3.8 中断

表 27-10 USART 整体中断一览表

中断名称	符号	说明
错误中断	USART_n_EI n=1~6	UART/多处理器/智能卡/时钟同步模式接收错误中断 LIN 错误中断
接收数据寄存器满中断	USART_n_RI n=1~6	UART/多处理器/LIN/智能卡/时钟同步模式接收数据寄存器满中断 LIN 错误中断
发送数据寄存器空中断	USART_n_TI n=1~6	UART/多处理器/LIN/智能卡/时钟同步模式发送数据寄存器空中断
发送完成中断	USART_n_TCI n=1~6	UART/多处理器/LIN/智能卡/时钟同步模式发送完成中断 (向发送数据寄存器写入发送数据可清中断)
发送完成中断	USART_n_TENDI n=1~6	UART/多处理器/LIN/智能卡/时钟同步模式发送完成中断 (向寄存器 USARTn_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=3/6	LIN 唤醒信号/间隔段检测中断

中断名称	符号	说明
UART 接收 TIMEOUT 中断	USART_n_RTO n=1/2/4/5	UART 接收 TIMEOUT 中断
RX 线唤醒停止模式中断	USART_1_WUPI	USART1 RX 线唤醒停止模式中断

27.4 寄存器

本章将详细描述 USART 模块控制状态等相关寄存器。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。

27.4.1 寄存器总表

USART1_NFC 基地址：0x40055400

表 27-11 USART1 滤波控制寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x1C	USART1_NFC	USART1 滤波控制寄存器	32

- USART1 基地址：0x4001CC00
- USART2 基地址：0x4001D000
- USART3 基地址：0x4001D400
- USART4 基地址：0x40020C00
- USART5 基地址：0x40021000
- USART6 基地址：0x40021400

表 27-12 通用同步异步收发器 (USART) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	USART_SR	状态寄存器	32
0x04	USART_TDR	发送数据寄存器	16
0x06	USART_RDR	接收数据寄存器	16
0x08	USART_BRR	波特率寄存器	32
0x0C	USART_CR1	控制寄存器 1	32
0x10	USART_CR2	控制寄存器 2	32
0x14	USART_CR3	控制寄存器 3	32
0x18	USART_PR	预分频寄存器	32
0x1C	USART_LBMC	LIN 波特率测量计数寄存器	32

27.4.2 状态寄存器 (USART_SR)

Offset	Bit Position																																	
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x000000C0																																	
Name	Reserved																MPB	Reserved				TEND	LBD	WKUP	RTOF	TXE	TC	RXNE	BE	ORE	Reserved	FE	PE	
Access																	RO					RO	RO	RO	RO	RO	RO	RO	RO	RO	RO		RO	RO

位/位域	标记	位名	功能描述	读写
31:17	Reserved	保留	-	-
16	MPB	多处理器位标志	0b0: 当前接收数据为通信数据 0b1: 当前接收数据为 ID  说明 MPB 位只在多处理器模式时有效。	RO
15:12	Reserved	保留	-	-
11	TEND	发送完成标志	0b0: 发送数据中 0b1: 发送数据完成 UART 模式，多处理器模式，LIN 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 USART_CR1.CTEND 位写入 1。 智能卡模式 - TEND 置位条件： - 最后 1 字节的数据送出后经过特定时间后，FE=0 而且发送数据寄存器的值没有被更新。 - TEND 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 USART_CR1.CTEND 位写入 1。	RO
10	LBD	LIN 间隔段检出标志	仅 USART3/USART6 有效： 0b0: 未检测到间隔段 0b1: 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 USART_CR2.LBDL 寄存器设定的宽度后，并检测到 break delimiter 时。 LBD 清零条件： - 清零寄存器 USART_CR1.CLBD 位写入 1。	RO

位/位域	标记	位名	功能描述	读写
10	LBD	LIN 间隔段检出标志	 说明 RE=0 并不能复位 LBD 位。	RO
9	WKUP	LIN Wakeup 信号检出标志	仅 USART3/USART6 有效： ● 0b0：未检测到 Wakeup 信号 ● 0b1：检测到 Wakeup 信号 WKUP 置位条件： ● USART_CR2.WKUPE=1 时，检测到 RX 线低电平宽度上大于等于 2.5 位数据宽度（19.2kbps 时为 130μs）。 WKUP 清零条件： ● 清零寄存器 USART_CR1.CWKUP 位写入 1。  说明 RE=0 并不能复位 WKUP 位。 检测到 WKUP 信号为 1 后，表示本 LIN 节点已经从休眠模式退出，设定寄存器 USART_CR2.WKUPE=0 后，进行后续间隔场检测等操作。	RO
8	RTOF	UART 接收 TIMEOUT 标志位	仅 USART1/USART2/USART4/USART5 有效： ● 0b0：无 UART 接收 TIMEOUT ● 0b1：发生 UART 接收 TIMEOUT RTOF 置位条件： ● 从检测到最后一帧数据的 STOP 位起经过设定时间后没有检测到新的接收数据。 RTOF 清零条件： ● 清零寄存器 USART_CR1.CRTOF 位写入 1。  说明 RTOF 为由硬件置 1，且只有在 USART_CR1.RE=1 且 USART_CR1.RTOE=1 时置 1。USART_CR1.RE=0 时，TIMEOUT 功能有效，但 RTOF 不置 1。	RO
7	TXE	发送数据寄存器空标志	● 0b0：发送数据寄存器非空，硬件将 TXE 清 0 ● 0b1：发送数据寄存器为空，硬件将 TXE 置 1  说明 TXE 位在 UART/时钟同步模式/LIN 模式时有效。	RO
6	TC	发送完成标志	● 0b0：发送数据中 ● 0b1：发送数据完成 UART 模式，多处理器模式，LIN 模式，时钟同步模式 ● TC 置位条件： ▶ TE=0 发送禁止时； ▶ 送出一帧数据的最后一位时，发送数据寄存器的值没有被更新。 ● TC 清零条件： ▶ TE=1 时，向发送数据寄存器写入发送数据 智能卡模式 ● TC 置位条件： ▶ TE=0 发送禁止时；	RO

位/位域	标记	位名	功能描述	读写
6	TC	发送完成标志	► 最后 1 字节的数据送出后经过特定时间后，FE=0 而且发送数据寄存器的值没有被更新。 TC 置位的具体时序为：校验位送出后经过 2.5 位时间。 ● TC 清零条件： ► TE=1 时，向发送数据寄存器写入发送数据  说明 TE 位由 0 变化为 1 时 TC 为保持为 1。	RO
5	RXNE	接收数据寄存器不为空标志	● 0b0：未接收到数据 ● 0b1：准备好读取接收到的数据 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。	RO
4	BE	LIN 总线检错误标志	仅 USART3/USART6 有效： ● 0b0：无总线错误 ● 0b1：发生总线错误 BE 置位条件： ● 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）。 BE 清零条件： ● 清零寄存器 USART_CR1.CBE 位写入 1。  说明 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。	RO
3	ORE	接收上溢错误标志	● 0b0：无接收上溢错误 ● 0b1：发生接收上溢错误 ORE 置位条件： ● 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。 ORE 清零条件： ● 清零寄存器 USART_CR1.CORE 位写入 1。  说明 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。	RO
2	Reserved	保留	-	-

位/位域	标记	位名	功能描述	读写
1	FE	接收帧错误标志	● 0b0：无接收帧错误 ● 0b1：发生接收帧错误 UART 模式，多处理器模式，LIN 模式 ● FE 置位条件： ▶ 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。 ● FE 清零条件： ▶ 清零寄存器 USART_CR1.CFE 位写入 1。  说明 RE=0 并不能复位 FE 位。 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。 智能卡模式 ● FE 置位条件： ▶ 采样到低电平错误信号标志。 ● FE 清零条件： ▶ 清零寄存器 USART_CR1.CFE 位写入 1。  说明 RE=0 并不能复位 FE 位。	RO
0	PE	接收数据校验错误标志	● 0b0：无接收数据校验错误 ● 0b1：发生接收数据校验错误 PE 置位条件： ● 接收数据发生奇偶校验错误时。 PE 清零条件： ● 清零寄存器 USART_CR1.CPE 位写入 1。  说明 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。	RO

27.4.3 发送数据寄存器 (USART_TDR)

Offset	Bit Position															
0x04	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x01FF															
Name	Reserved						MPID	TDR								
Access							RW	RW								

位/位域	标记	位名	功能描述	读写
15:10	Reserved	保留	-	-
9	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： ● 0b0：发送数据 ● 0b1：发送 ID  说明 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
8:0	TDR	发送数据	发送数据寄存器  说明 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

27.4.4 接收数据寄存器 (USART_RDR)

Offset	Bit Position															
0x06	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved						RDR									
Access							RO									

位/位域	标记	位名	功能描述	读写
15:9	Reserved	保留	-	-
8:0	RDR	接收数据	接收数据寄存器  说明 最高位 RDR[8]仅在 UART 模式而且数据长度设定为 9 位时有效。	RO

27.4.5 波特率寄存器 (USART_BRR)

Offset	Bit Position																																				
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Reset	0x0000FFFF																																				
Name	Reserved																DIV_Integer				Reserved	DIV_Fraction															
Access																	RW					RW															

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:8	DIV_Integer	整数分频	整数分频寄存器 说明 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW
7	Reserved	保留	读出为 1，写入时必须写 1	-
6:0	DIV_Fraction	小数分频	小数分频寄存器 说明 该位只能在 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 27-21 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	$B = \frac{C}{2 \times \text{BCN} \times (\text{DIV\_Integer} + 1)}$	$E = \left\{ \frac{C}{2 \times \text{BCN} \times (\text{DIV\_Integer} + 1) \times B} - 1 \right\} \times 100\%$

表 27-22 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	$B = \frac{C \times (128 + \text{DIV\_Fraction})}{2 \times \text{BCN} \times (\text{DIV\_Integer} + 1) \times 256}$	$E = \left\{ \frac{C \times (128 + \text{DIV\_Fraction})}{2 \times \text{BCN} \times (\text{DIV\_Integer} + 1) \times 256 \times B} - 1 \right\} \times 100\%$

其中：

- B：波特率（Mbps）
- C：USART_PR.ULBREN 与 USART_PR.PSC 位设定的时钟（MHz）
- BCN：USART_CR3.BCN 寄存器设定值

27.4.6 控制寄存器 1 (USART_CR1)

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x80000000																															
Name	SBS	NFE	FBME	ML	Reserved	CTEND	MS	CLBD	CWKUP	CBE	CRTOF	CORE	Reserved	CFE	CPE	OVER8	Reserved	M	Reserved	PCE	PS	TENDIE	TXEIE	TCIE	RIE	SLME	TE	RE	RTOIE	RTOE		
Access	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW		RW	RW	RW		RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31	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： 0b0：开始位检测方式为 RX 管脚低电平 0b1：开始位检测方式为 RX 管脚下降沿  说明 非 UART 模式时该位须保持复位值。 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW
30	NFE	数字滤波使能位	0b0：禁止 0b1：使能  说明 时钟同步模式、LIN 模式检测同步帧时该位须保持复位值。 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW
29	FBME	小数波特率功能使能	0b0：禁止 0b1：使能  说明 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW
28	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： 0b0：LSB 方式 0b1：MSB 方式  说明 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW
27:26	Reserved	保留	-	-
25	CTEND	TEND 清零位	写 1 清除 TEND 标志，写 0 无效；读出恒为 0	RW
24	MS	通信模式选择位	0b0：UART 模式 0b1：时钟同步模式  说明 该位只能在 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
23	CLBD	LBD 清零位	仅 USART3/USART6 有效： 写 1 清除 LBD 标志，写 0 无效；读出恒为 0	RW

位/位域	标记	位名	功能描述	读写
22	CWKUP	WKUP 清零位	仅 USART3/USART6 有效： 写 1 清除 WKUP 标志，写 0 无效；读出恒为 0	RW
21	CBE	BE 清零位	仅 USART3/USART6 有效： 写 1 清除 BE 标志，写 0 无效；读出恒为 0	RW
20	CRTOF	RTOF 清零位	仅 USART1/USART2/USART4/USART5 有效： 写 1 清除 RTOF 标志，写 0 无效；读出恒为 0	RW
19	CORE	ORE 标志清零位	写 1 清除 ORE 标志，写 0 无效；读出恒为 0	RW
18	Reserved	保留	-	-
17	CFE	FE 标志清零位	写 1 清除 FE 标志，写 0 无效；读出恒为 0	RW
16	CPE	PE 标志清零位	写 1 清除 PE 标志，写 0 无效；读出恒为 0	RW
15	OVER8	UART 过采样模式	UART 过采样模式设定，即一位数据传输期间的基本时钟数： ● 0b0：16 位 ● 0b1：8 位  说明 智能卡模式、时钟同步模式时该位须保持复位值。 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW
14:13	Reserved	保留	-	-
12	M	数据长度设定位	UART 模式时，发送/接收数据长度设定位： ● 0b0：8 位 ● 0b1：9 位  说明 LIN 模式、智能卡模式、时钟同步模式时该位须保持复位值。 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW
11	Reserved	保留	-	-
10	PCE	校验使能位	UART 模式时，奇偶校验使能位： ● 0b0：无校验 ● 0b1：校验  说明 智能卡模式时该位必须为 1，LIN 模式、多处理器模式、时钟同步模式时该位必须保持复位值。 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW
9	PS	校验位	UART 模式时，奇偶校验选择位： ● 0b0：偶校验 ● 0b1：奇校验  说明 该位只在 PCE=1 时有效。 智能卡模式时该位必须保持为 0。 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW
8	TENDIE	发送完成中断使能位	● 0b0：TENDI 中断请求无效，TENDI 中断不发生 ● 0b1：TENDI 中断请求使能，TENDI 中断发生	RW

位/位域	标记	位名	功能描述	读写
7	TXEIE	发送数据寄存器空中断使能位	0b0: TI 中断请求无效, TI 中断不发生 0b1: TI 中断请求有效, TI 中断发生  说明 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。	RW
6	TCIE	发送完成中断使能位	0b0: TCI 中断请求无效, TCI 中断不发生 0b1: TCI 中断请求使能, TCI 中断发生  说明 非智能卡模式读取有效, 智能卡模式读取为 0。	RW
5	RIE	接收中断、错误中断使能位	0b0: 接收中断、错误中断请求无效, RI 和 EI 中断不发生 0b1: 接收中断、错误中断请求有效, RI 和 EI 中断发生	RW
4	SLME	静默模式使能位	多处理器操作时, 静默模式使能位: 0b0: 正常模式 0b1: 静默模式 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。  说明 该位只在 UART 多处理器模式时有效, 其他模式时该位必须保持复位值。	RW
3	TE	发送器使能位	0b0: 发送器禁止 0b1: 发送器使能  说明 时钟同步模式时该位只能在 TE=0&RE=0 (发送/接收禁止) 时写 1。	RW
2	RE	接收器使能位	0b0: 接收器禁止 0b1: 接收器使能  说明 时钟同步模式时该位只能在 TE=0&RE=0 (发送/接收禁止) 时写 1。	RW
1	RTOIE	UART TIMEOUT 中断使能位	仅 USART1/USART2/USART4/USART5 有效: 0b0: UART TIMEOUT 中断请求无效, RTOI 中断不发生 0b1: UART TIMEOUT 中断请求有效, RTOI 中断发生	RW
0	RTOE	UART TIMEOUT 功能使能位	仅 USART1/USART2/USART4/USART5 有效: 0b0: UART TIMEOUT 功能禁止 0b1: UART TIMEOUT 功能使能	RW

27.4.7 控制寄存器 2 (USART_CR2)

Offset	Bit Position																																			
0x10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reset	0x00000600																																			
Name	Reserved															SBKM	SBK	Reserved	LINEN	STOP	CLKC	Reserved	WKUPE	SBKL	LBDL	LBDIE	BEE	BEIE	WKUPIE	MPE						
																RW	RW		RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Access																																				

位/位域	标记	位名	功能描述	读写															
31:18	Reserved	保留	-	-															
17	SBKM	间隔段发送模式	仅 USART3/USART6 有效： 具体设置请参见 SBK 位描述。  说明 该位只在 LINEN=1 时有效。	RW															
16	SBK	间隔段发送使能位	仅 USART3/USART6 有效： <table><tr><th>SBK</th><th>SBKM</th><th>间隔段发送</th></tr><tr><td>0b0</td><td>0b0</td><td>不发送间隔段</td></tr><tr><td>0b0</td><td>0b1</td><td>不发送间隔段</td></tr><tr><td>0b1</td><td>0b0</td><td>SBK 写 1 时，自动发送间隔段，发送完成后，SBK 自动清零</td></tr><tr><td>0b1</td><td>0b1</td><td>写数据 0x00 到 USART_TDR.TDR，开始发送间隔段，发送完成后，SBK 自动清零</td></tr></table>  说明 该位只在 LINEN=1 时有效。	SBK	SBKM	间隔段发送	0b0	0b0	不发送间隔段	0b0	0b1	不发送间隔段	0b1	0b0	SBK 写 1 时，自动发送间隔段，发送完成后，SBK 自动清零	0b1	0b1	写数据 0x00 到 USART_TDR.TDR，开始发送间隔段，发送完成后，SBK 自动清零	RW
SBK	SBKM	间隔段发送																	
0b0	0b0	不发送间隔段																	
0b0	0b1	不发送间隔段																	
0b1	0b0	SBK 写 1 时，自动发送间隔段，发送完成后，SBK 自动清零																	
0b1	0b1	写数据 0x00 到 USART_TDR.TDR，开始发送间隔段，发送完成后，SBK 自动清零																	
15	Reserved	保留	-	-															
14	LINEN	LIN 功能使能	仅 USART3/USART6 有效： 0b0：禁止 LIN 功能 0b1：使能 LIN 功能  说明 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW															
13	STOP	停止位设定位	UART 模式时，停止位长度设定位： 0b0：1 停止位 0b1：2 停止位  说明 LIN 模式、智能卡模式、时钟同步模式时该位须保持复位值。 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW															

位/位域	标记	位名	功能描述	读写
12:11	CLKC	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 0b01: 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 0b10/0b11: 时钟源为外部输入时钟, 输入时钟的频率为波特率的 16 倍 (USART_CR1.OVER8=0) 或者 8 倍 (USART_CR1.OVER8=1) 时钟同步模式 0b00/0b01: 时钟源为内部波特率生成器生成的时钟, 输出到 USARTn_CK 管脚 0b10/0b11: 时钟源为外部输入时钟, 输入时钟的频率和波特率相同 智能卡模式 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 0b01: 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 0b10/0b11: 禁止设置  说明 该位只能在 TE=0&RE=0 (发送/接收禁止) 时设定。UART 模式使用接收 TIMEOUT 功能时, 该位必须设置为 0b01, 可以不使用 USARTn_CK 管脚。	RW
10:9	Reserved	保留	所有位读出为 1, 写入时必须写 1	-
8	WKUPE	LIN 唤醒信号检测使能位	仅 USART3/USART6 有效: 0b0: 禁止唤醒信号检测 0b1: 使能唤醒信号检测  说明 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
7:6	SBKL	LIN 间隔段低电平宽度设定位	仅 USART3/USART6 有效: 0b00: 10 位 0b01: 11 位 0b10: 13 位 0b11: 14 位  说明 非 UART-LIN 模式时该位必须保持复位值。该位只能在 TE=0&RE=0 (发送/接收禁止) 时设定。	RW

位/位域	标记	位名	功能描述	读写
5	LBDL	LIN 间隔段低电平宽度检测阈值设定位	仅 USART3/USART6 有效： ● 0b0: ≥ 10 位 ● 0b1: ≥ 11 位  说明 非 UART-LIN 模式时该位必须保持复位值。 该位只能在 TE=0&RE=0 (发送/接收禁止) 时设定。	RW
4	LBDIE	LIN 间隔段检测中断使能位	仅 USART3/USART6 有效： ● 0b0: LIN 间隔段检测中断请求无效，不发生中断 ● 0b1: LIN 间隔段检测中断请求有效，发生中断  说明 非 UART-LIN 模式时该位必须保持复位值。	RW
3	BEE	LIN 总线错误检测功能使能位	仅 USART3/USART6 有效： ● 0b0: 禁止 LIN 总线错误功能检测功能 ● 0b1: 使能 LIN 总线错误功能检测功能  说明 非 UART-LIN 模式时该位必须保持复位值。 该位只能在 TE=0&RE=0 (发送/接收禁止) 时设定。	RW
2	BEIE	LIN 总线错误中断使能位	仅 USART3/USART6 有效： ● 0b0: LIN 总线错误中断请求无效，不发生中断 ● 0b1: LIN 总线错误中断请求有效，发生中断  说明 非 UART-LIN 模式时该位必须保持复位值。	RW
1	WKUPIE	LIN 唤醒信号检测中断使能位	仅 USART3/USART6 有效： ● 0b0: LIN 唤醒信号检测中断请求无效，不发生中断 ● 0b1: LIN 唤醒信号检测中断请求有效，发生中断  说明 非 UART-LIN 模式时该位必须保持复位值。	RW
0	MPE	多处理器功能使能位	UART 模式时，多处理器功能使能位： ● 0b0: 禁止 ● 0b1: 使能  说明 非 UART 模式时该位必须保持复位值。 该位只能在 TE=0&RE=0 (发送/接收禁止) 时设定。	RW

27.4.8 控制寄存器 3 (USART_CR3)

Offset	Bit Position																																			
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reset	0x00000000																																			
Name	Reserved								BCN				Reserved										CTSE		RTSE		Reserved		SCEN		LOOP		HDSEL		Reserved	
Access									RW														RW		RW				RW		RW		RW			

位/位域	标记	位名	功能描述	读写
31:24	Reserved	保留	-	-
23:21	BCN	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： ● 0b000：32 ● 0b001：64 ● 0b010：禁止设置 ● 0b011：128 ● 0b100：禁止设置 ● 0b101：256 ● 0b110：372 ● 0b111：禁止设置  说明 非智能卡模式时该位必须保持复位值。 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。 USART3/USART6 不支持该功能，读出为 0。	RW
20:10	Reserved	保留	-	-
9	CTSE	CTS 功能使能位	● 0b0：CTS 功能无效 ● 0b1：CTS 功能有效  说明 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW
8	RTSE	RTS 功能使能位	● 0b0：RTS 功能无效 ● 0b1：RTS 功能有效  说明 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW
7:6	Reserved	保留	-	-
5	SCEN	智能卡模式使能位	● 0b0：禁止智能卡模式 ● 0b1：使能智能卡模式  说明 非智能卡模式时该位必须保持复位值。 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。 USART3/USART6 不支持该功能，读出为 0。	RW

位/位域	标记	位名	功能描述	读写
4	LOOP	LIN 回环模式使能位	仅 USART3/USART6 有效： ● 0b0：正常模式 ● 0b1：回环模式  说明 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW
3	HDSEL	UART 单线半双工模式使能位	● 0b0：UART 全双工模式 ● 0b1：UART 半双工模式  说明 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW
2:0	Reserved	保留	-	-

27.4.9 预分频寄存器 (USART_PR)

Offset	Bit Position																																	
0x18	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved																												ULBREN		LBMPSC		PSC	
Access																													RW		RW		RW	

位/位域	标记	位名	功能描述	读写
31:5	Reserved	保留	-	-
4	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： ● 0b0：超低波特率功能无效 ● 0b1：超低波特率功能有效 且根据 PSC 的值预分频器分频值为： ● 0b00：PCLK1/128 ● 0b01：PCLK1/256 ● 0b10：PCLK1/512 ● 0b11：PCLK1/1024  说明 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW
3:2	LBMPSC	LIN 波特率测量计数器时钟源选择	仅 USART3/USART6 有效： ● 0b00：PCLK1 ● 0b01：PCLK1/2 ● 0b10：PCLK1/4 ● 0b11：PCLK1/8  说明 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW
1:0	PSC	预分频器值	内部时钟源时，预分频器分频值选择位： ● 0b00：PCLK1 ● 0b01：PCLK1/4 ● 0b10：PCLK1/16 ● 0b11：PCLK1/64  说明 该位只能在 TE=0&RE=0（发送/接收禁止）时设定。	RW

27.4.10 LIN 波特率测量计数寄存器 (USART_LBMC)

Offset	Bit Position																															
0x1C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																LBMC															
Access																																

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	LBMC	LIN 波特率测量结果计数器	仅 USART3/USART6 有效。 LIN 从节点，用 USART_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。 USART_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。  说明 LBMC 的值在同步段接收完成后读取才有意义。	RO

27.4.11 USART1 滤波控制寄存器 (USART1_NFC)

Offset	Bit Position																															
0x1C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																												USART1_NFE		USART1_NFS	
Access																													RW		RW	

位/位域	标记	位名	功能描述	读写
31:3	Reserved	保留	-	-
2	USART1_NFE	USART1 滤波使能控制	用于控制 STOP 模式下，RX 线上的模拟滤波器的开关： ● 0b0：模拟滤波器禁止 ● 0b1：模拟滤波器使能，滤波范围请参考 USART1_NFS 位的设定	RW
1:0	USART1_NFS	USART1 滤波范围设置	用于控制 STOP 模式下，RX 线上的模拟滤波器的滤波范围： ● 0b00：滤波宽度档位 1 ● 0b01：滤波宽度档位 2 ● 0b10：滤波宽度档位 3 ● 0b11：滤波宽度档位 4 各档位具体值请参见《数据手册》“电气特性-工作条件-USART1 STOP 模式下 RX 滤波特性”章节相关描述。	RW

27.5 注意事项

27.5.1 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。

27.5.2 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。

27.5.3 其他注意事项

为了防止发送禁止时 TX 通信线 Hi-Z 状态，可以采用以下方法：

- 通信线上拉；
- 发送数据结束时，USARTn_CR1.TE=0 之前，将 TX 管脚设为普通 IO 输出；
- 发送数据开始前，USARTn_CR1.TE=1 之后，将 IO 设为 TX 功能。

28 集成电路总线 (I2C)

28.1 简介

I2C（集成电路总线）用作微控制器和 I2C 串行总线之间的接口。提供多主模式功能，可以控制所有 I2C 总线的协议、仲裁。支持标准模式、快速模式。还支持 SMBus 总线。

28.2 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
- 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

28.3 功能说明

28.3.1 I2C 系统框图

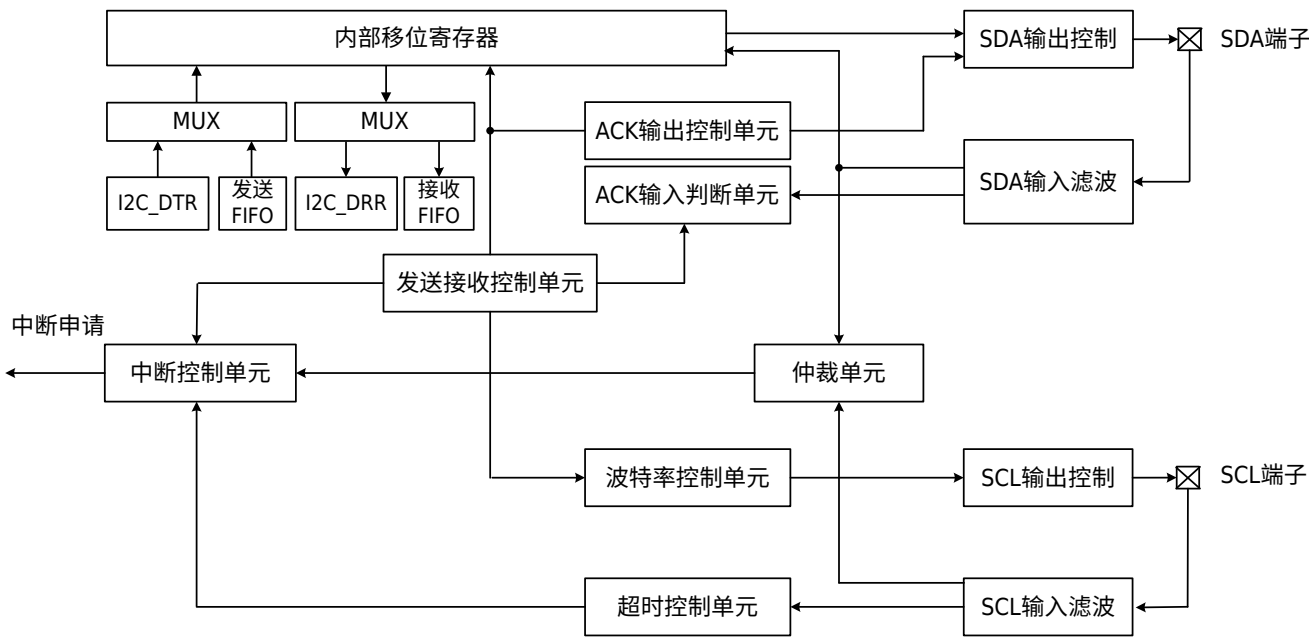


图 28-1 I2C 系统框图

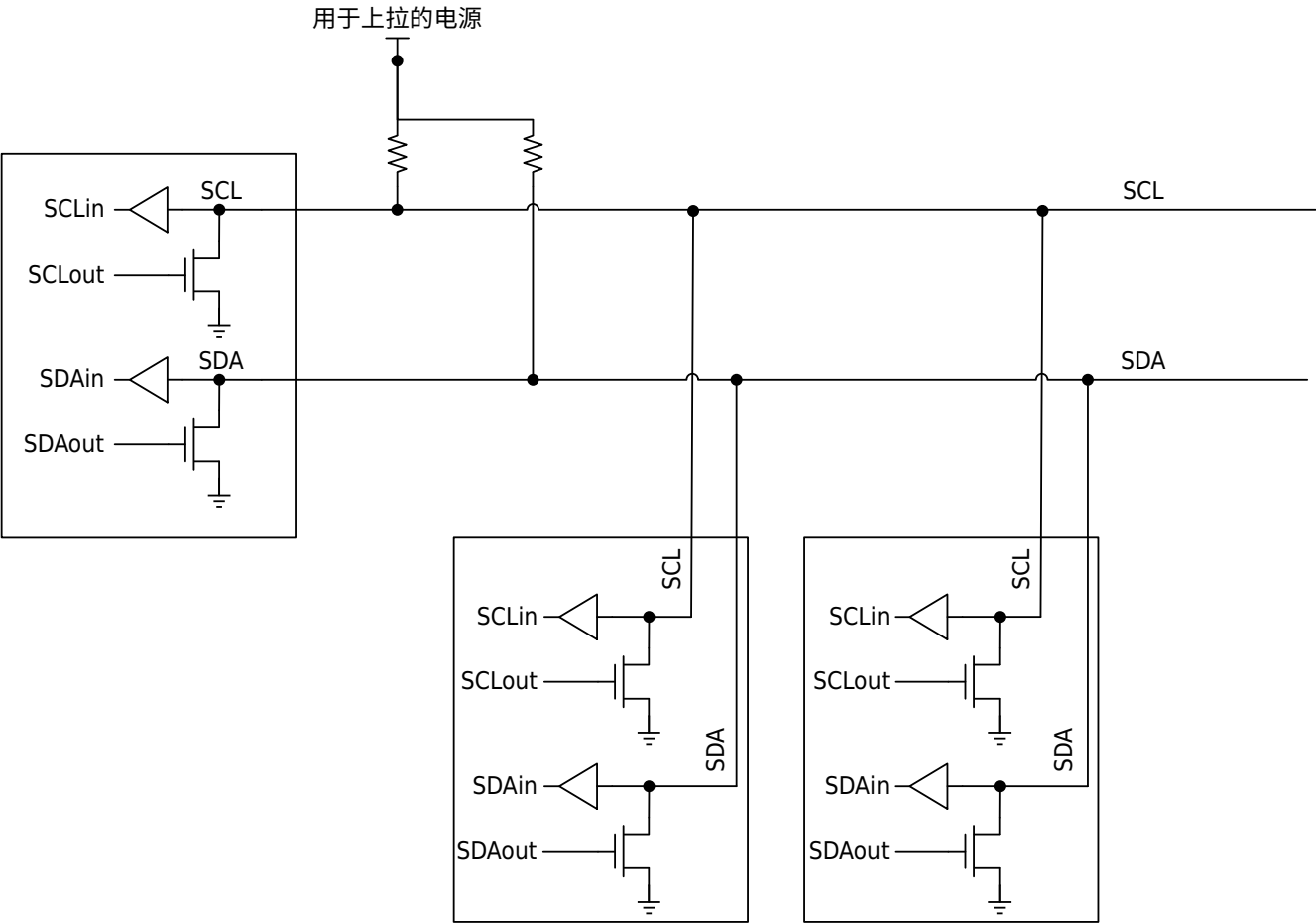


图 28-2 I2C 总线的结构例

表 28-1 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，需要将 SCL/SDA 输入电平设定为 Schmitt 电平（兼容 CMOS 电平）；当选择 SMBus 时，需要将 SCL/SDA 输入电平设定为 CMOS 电平。具体配置请参见《通用 IO（GPIO）》章节通用控制寄存器 PCRxy.CINSEL 位相关描述。

28.3.2 I2C 协议

28.3.2.1 数据传输

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 和 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

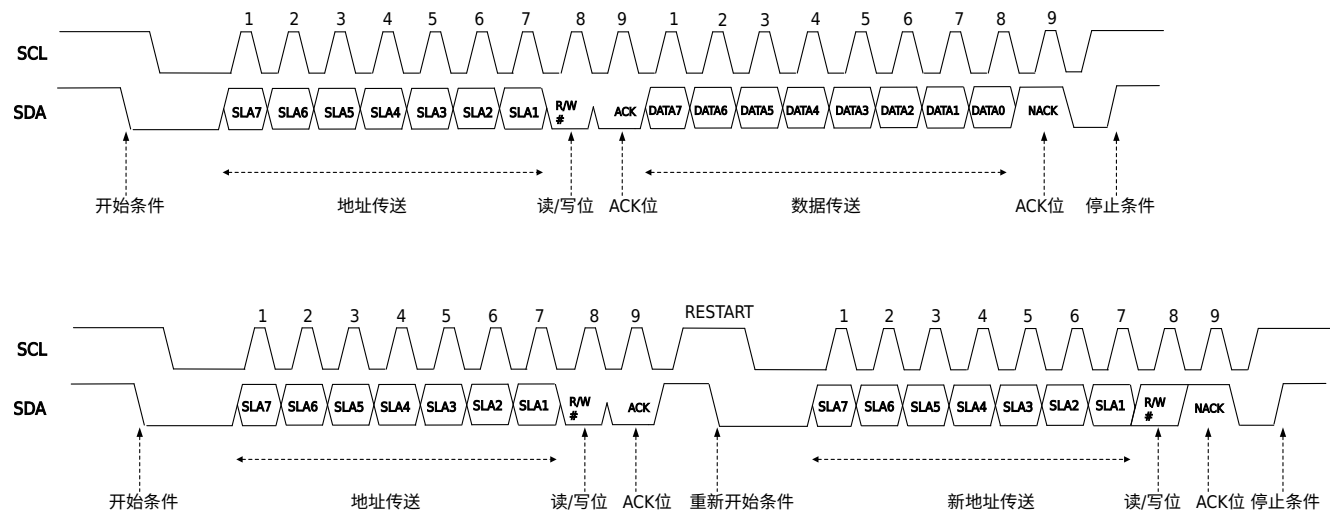


图 28-3 I2C 总线的时序图

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.EMPTFY 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。

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）。

7位地址格式

S	SLA（7位）	R/W #	ACK/ NACK	DATA（8位）	ACK	DATA（8位）	ACK/ NACK	P

10位地址格式

S	0b11110+SL A（2位）	W#	ACK	SLA（8位）	ACK	DATA（8位）	ACK	DATA（8位）	ACK/ NACK	P

S	0b11110+SL A（2位）	W#	ACK	SLA（8位）	ACK	Sr	0b11110+SL A（2位）	R	ACK	DATA（8位）	ACK	DATA（8位）	ACK/ NACK	P

图 28-4 I2C 总线的数据格式

- S：表示开始条件
- SLA：表示从机地址
- R/W#：表示发送和接收的方向。当 R/W#为 1 时，将数据从从机发送到主机；当 R/W#为 0 时，将数据从主机发送到从机
- Sr：表示重新开始条件
- DATA：表示发送和接收的数据
- P：表示停止条件

数据传送

地址匹配一致后，总线上的主机根据 R/W 定义的方向一帧一帧的传送数据。
所有的地址帧后传送的数据都视为数据帧。即使是 10 位地址格式的低 8 位地址也视为数据帧。
数据帧的长度是 8 位。SCL 的低电平 SDA 变化，SCL 的高电平 SDA 保持，每个时钟周期发送一位数据。数据帧后的第 9 个时钟是应答位，是接收方向发送方传送的握手信号。
如果总线上从机接收数据，在第 9 个时钟周期不响应主机，从机必须发送 NACK。如果总线上主机接收数据，第 9 个周期发送 NACK，从机接收到 NACK，从机停止发送数据。
无论主机还是从机发送了 NACK，数据传送终止。主机可以做下列任一动作：

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例如下图所示。

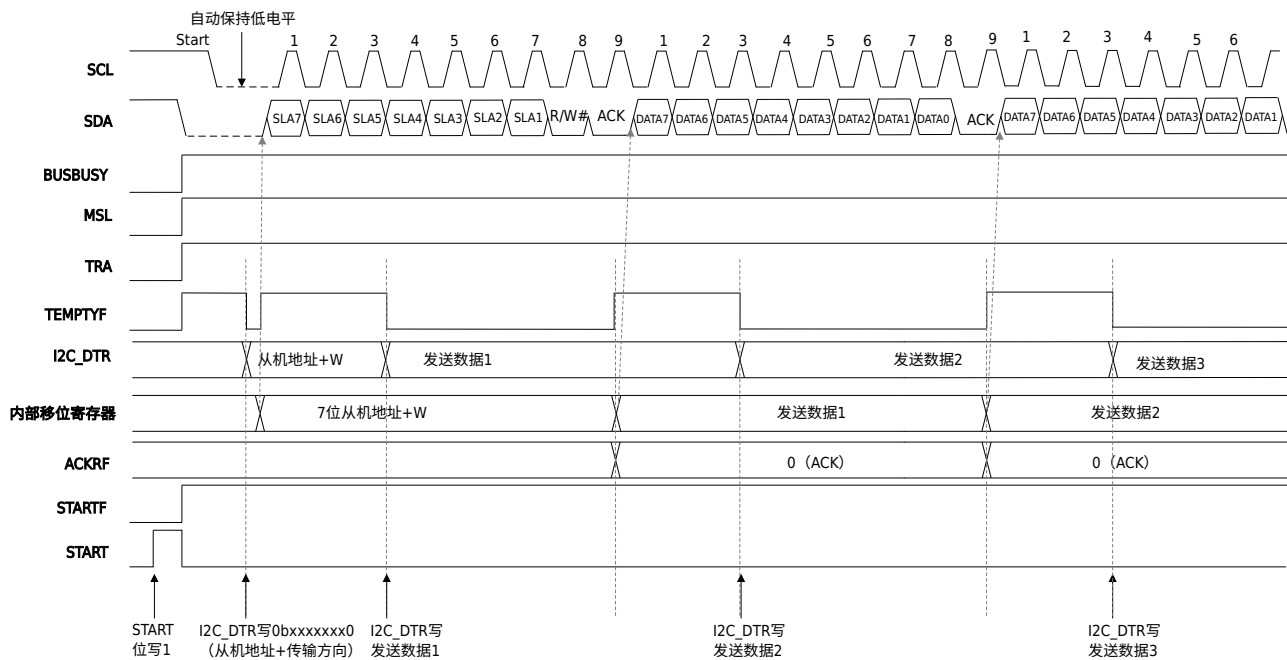


图 28-5 7 位地址格式的主机发送数据时序图（例）

- 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。

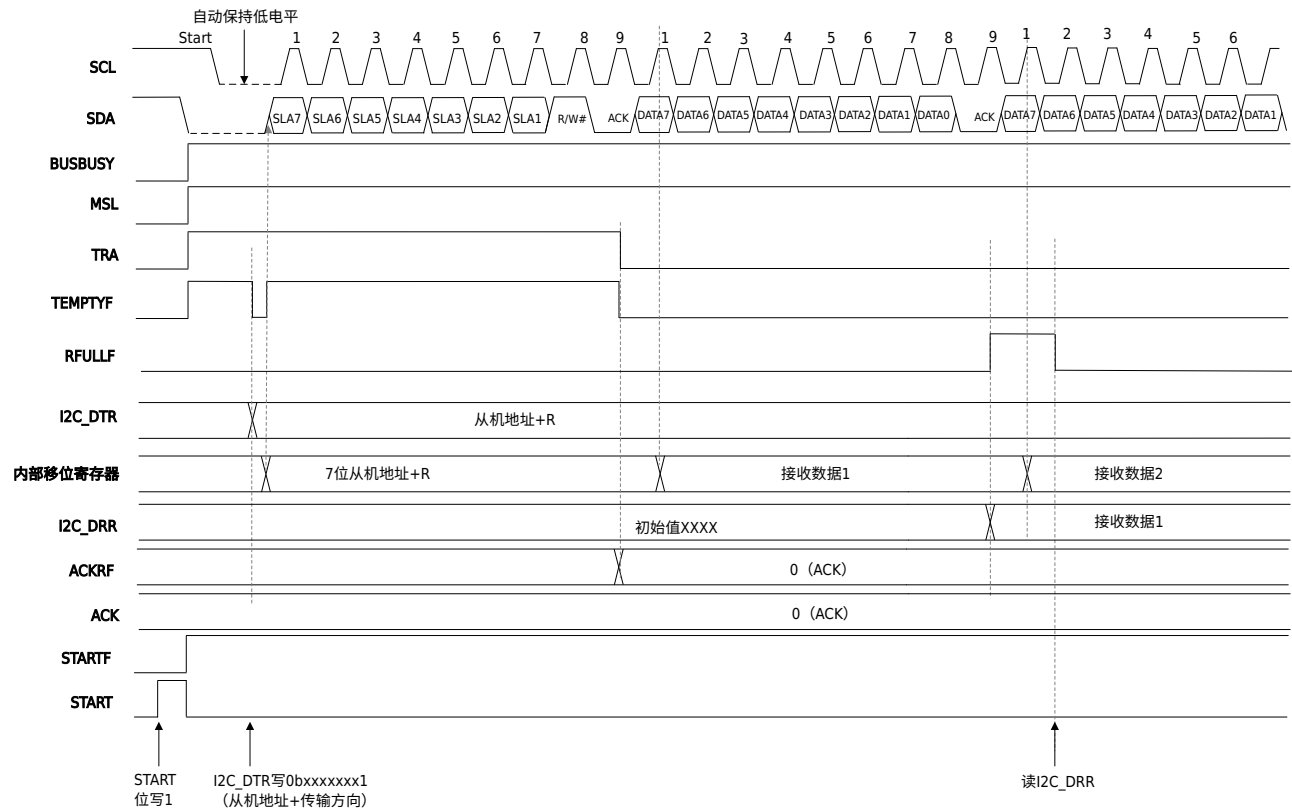


图 28-6 7 位地址格式的主机接收数据的时序图（例）

● 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。

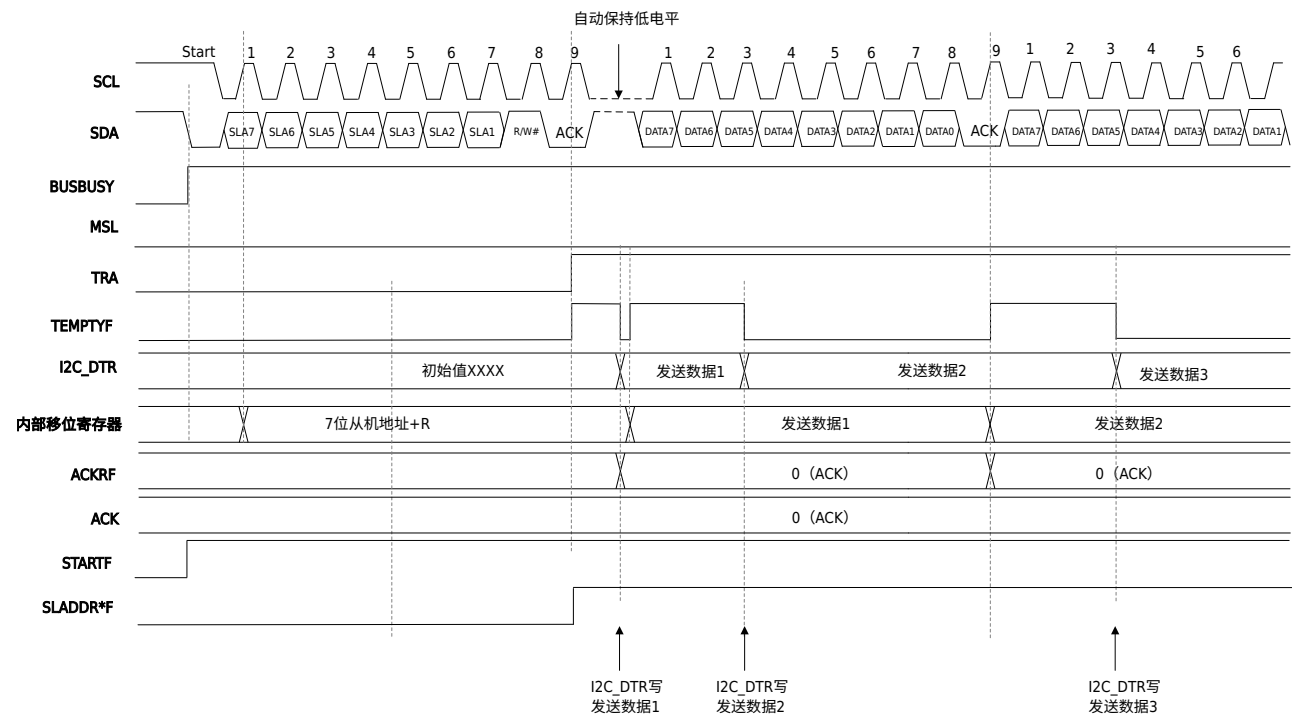


图 28-7 7 位地址格式的从机发送模式时序图（例）

● 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。

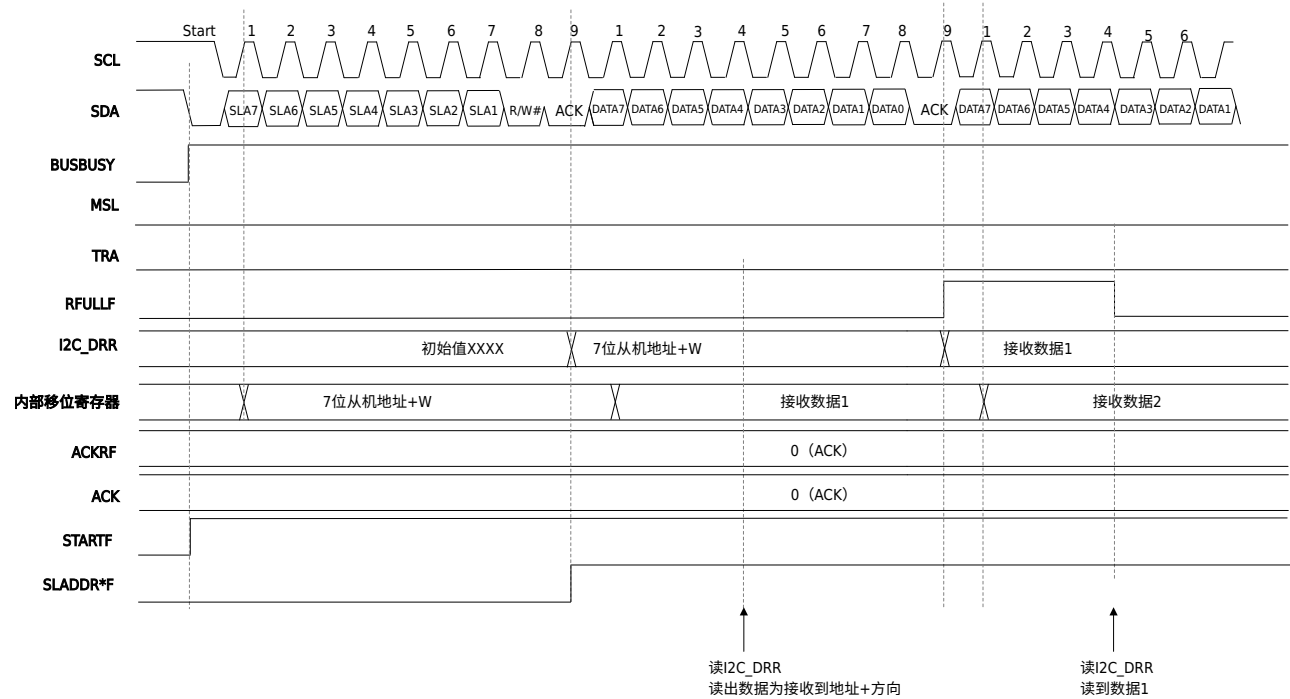


图 28-8 7 位地址格式从机接收模式时序图（例）

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。

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。

28.3.2.2 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。

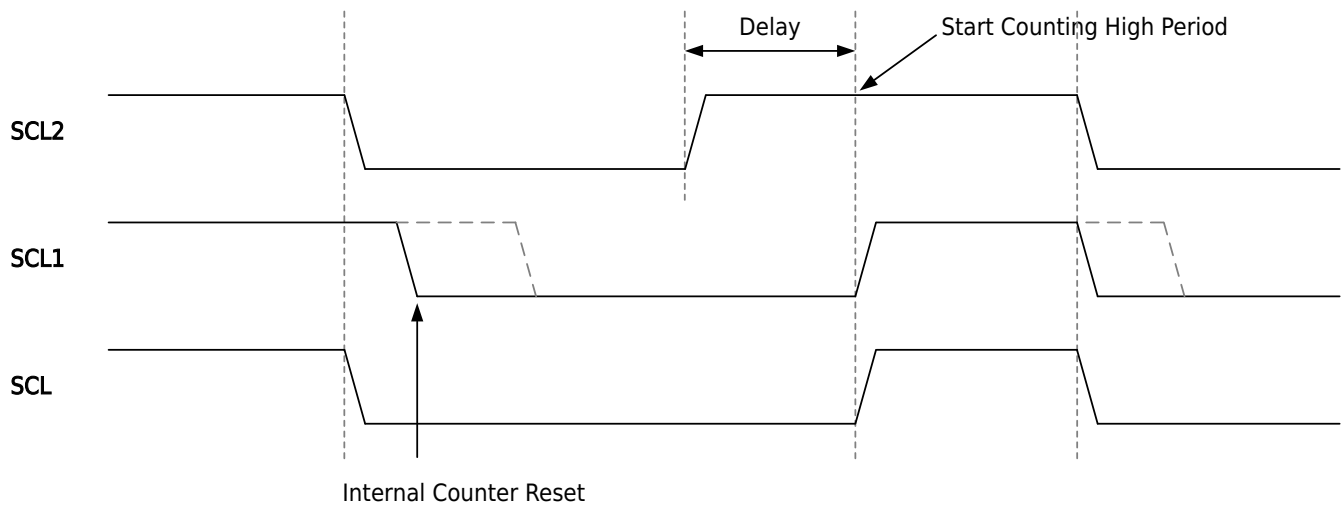


图 28-9 SCL 同步时序

28.3.2.3 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。

28.3.2.4 握手

数据传送过程中通过 SCL 时钟同步机制实现握手。从机在传送完一帧数据后（包含 ACK 位），将 SCL 时钟线维持在低电平。在这种情况下，SCL 时钟的低电平让主机进入等待状态，直到从机释放 SCL 线。

从机发送模式

在发送模式中（I2C_SR.TRA 位=1），如果内部移位寄存器为空状态并且未写发送数据（I2C_DTR 寄存器），就在第 9 时钟和下次传送的第 1 个时钟的低电平区间自动保持 SCL 线的低电平，动作时序如下图所示。

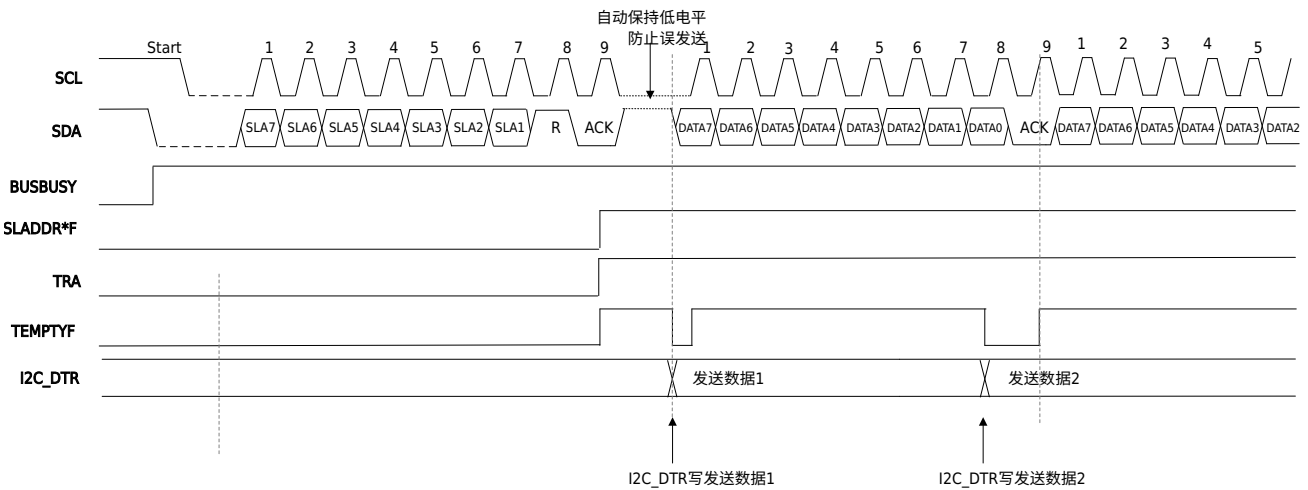


图 28-10 从机发送时序图（1）

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.EMPTTYF 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。

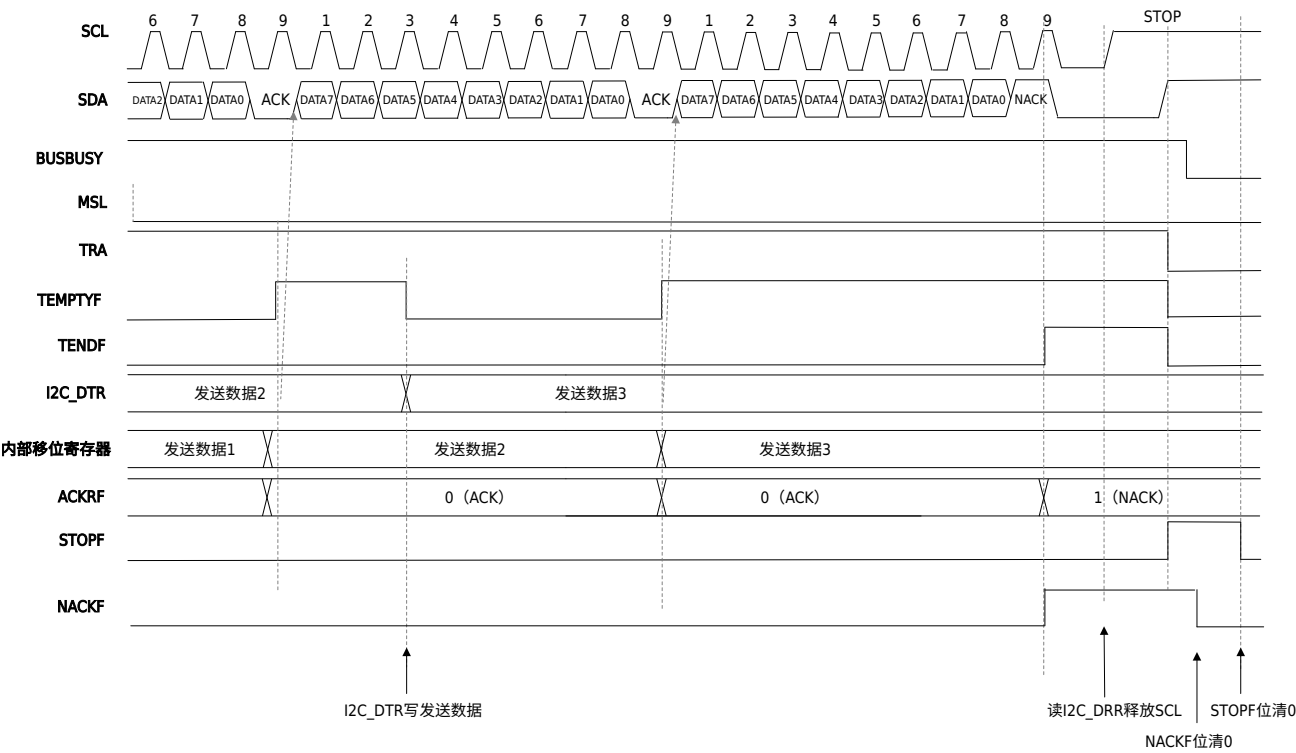


图 28-11 从机发送时序图 (2)

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。

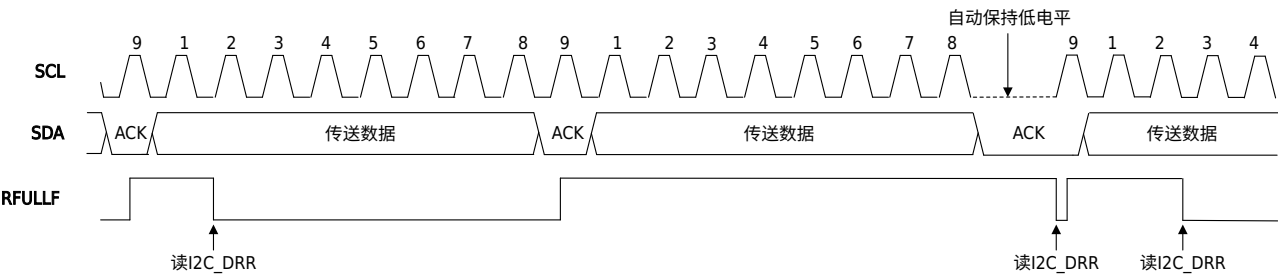


图 28-12 从机接收时序图

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。

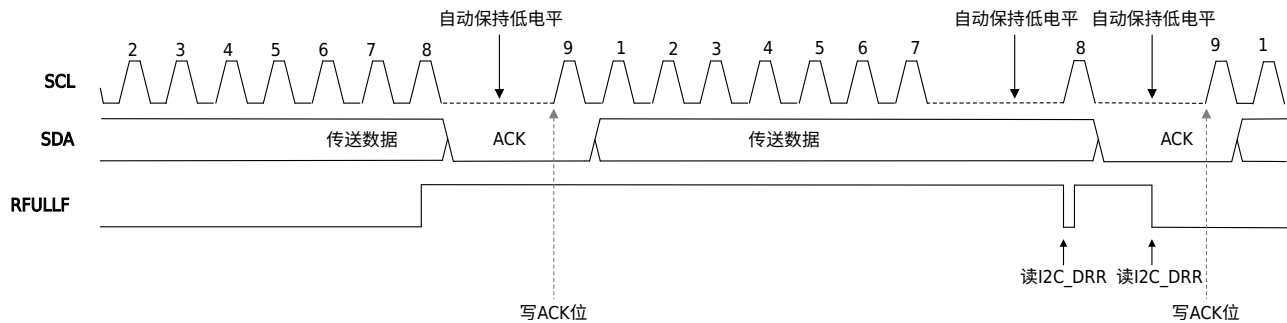


图 28-13 快速 ACK/NACK 时序图

28.3.3 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。

28.3.3.1 从机地址匹配

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 个时钟的上升沿将对应的 SLADDR1F、SLADDR0F 置 1，然后根据后续的 R/W# 位将 I2C_SR.RFULLF 标志或者 I2C_SR.TEMPTYF 标志置 1。由此，就能产生接收数据满中断或者发送数据空中断，并且能通过确认 I2C_SR.SLADDR1F、SLADDR0F 标志判断指定了哪个从机地址。

I2C_SR.SLADDR1F、SLADDR0F 标志变为 1 的时序分别如下图所示。

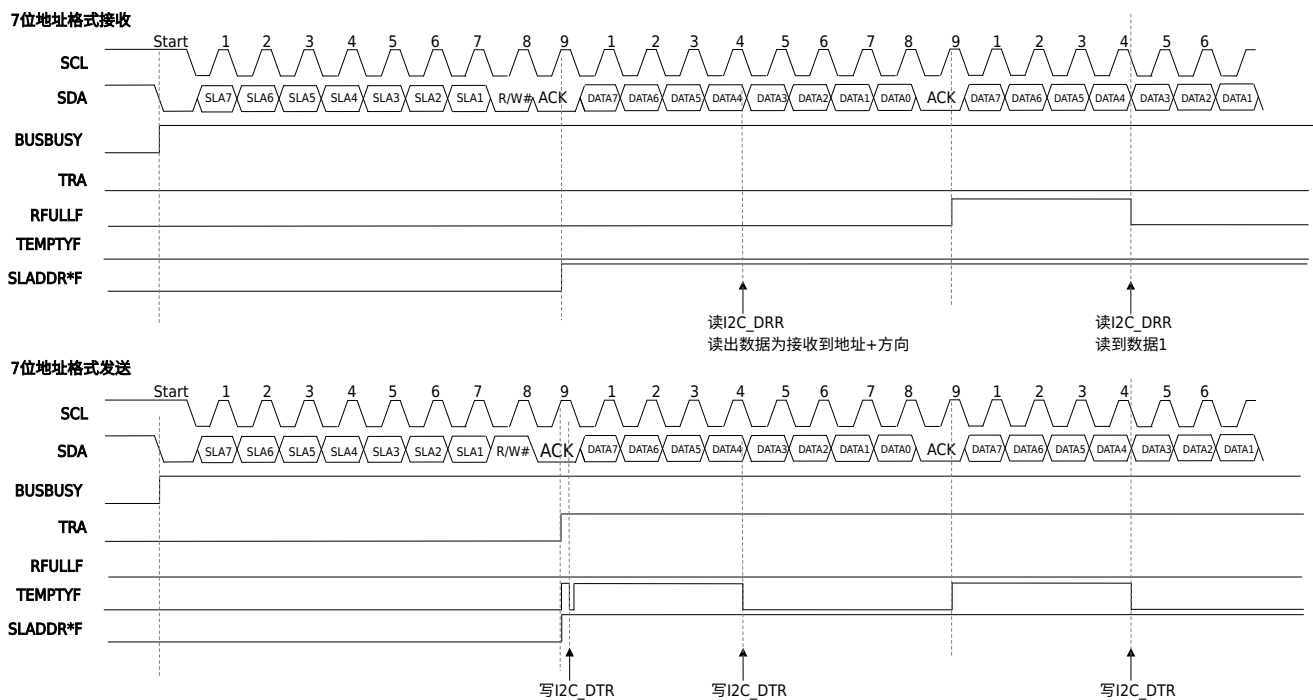


图 28-14 选择 7 位地址格式时的时序

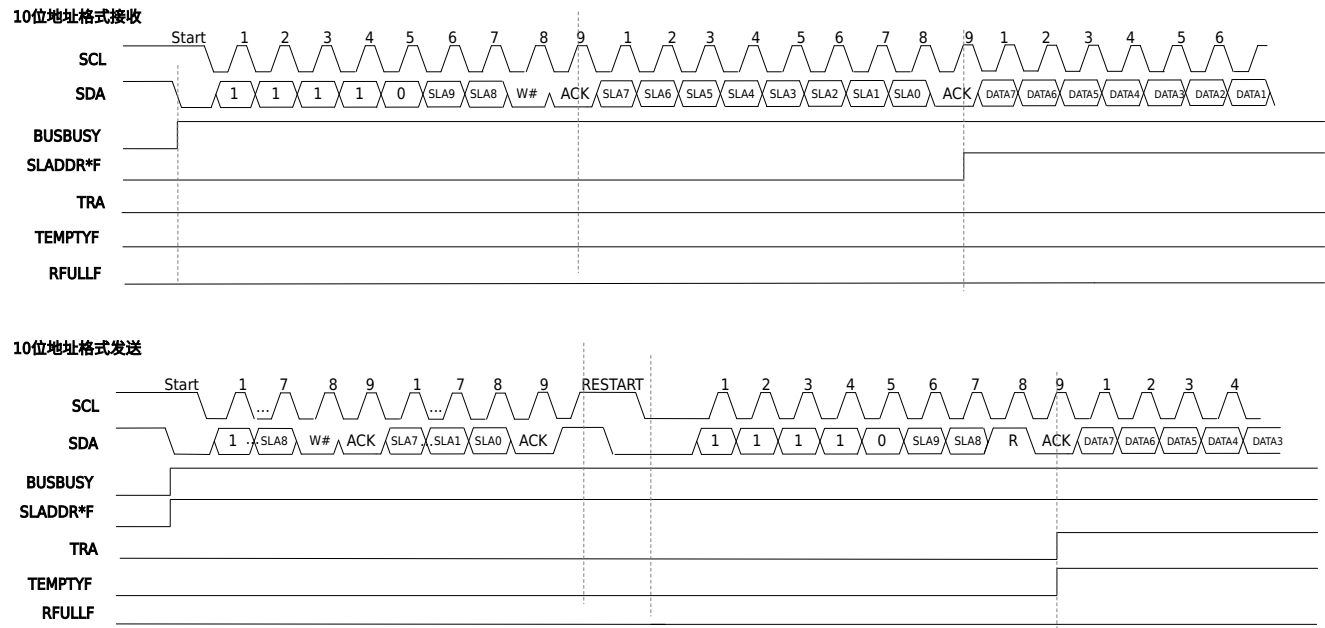


图 28-15 选择 10 位地址格式时的时序

28.3.3.2 广播地址匹配

当 I2C_CR1.GCEN 位为 1 时，能检测广播地址 (0b0000000+0[W])。

但是开始条件或者重新开始条件后的地址为 0b0000000+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 个时钟的上升沿将 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。

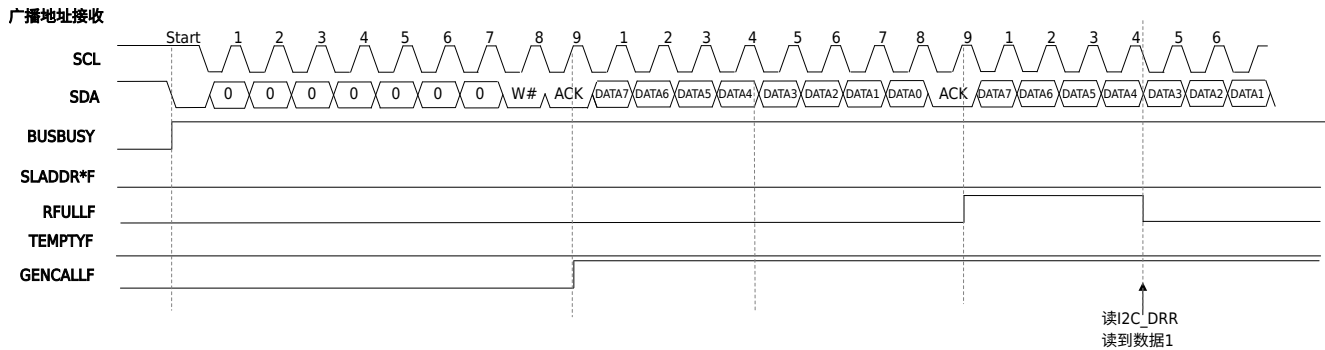


图 28-16 广播地址匹配时序图

28.3.3.3 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=0 且 I2C_SR.TRA=0) 中检测主机地址 (0b0001000)。

如果检测到 SMBUS 主机地址，就在 SCL 时钟的第 9 个时钟的上升沿将 I2C_SR.SMBHOSTF 标志置 1。

即使接在 SMBUS 主机地址 (0b0001000) 后面的位是 Rd 位 (R/W#位接收到 1)，也能检测 SMBUS 主机地址。SMBUS 主机地址检测后的运行和普通的从机模式运行相同。

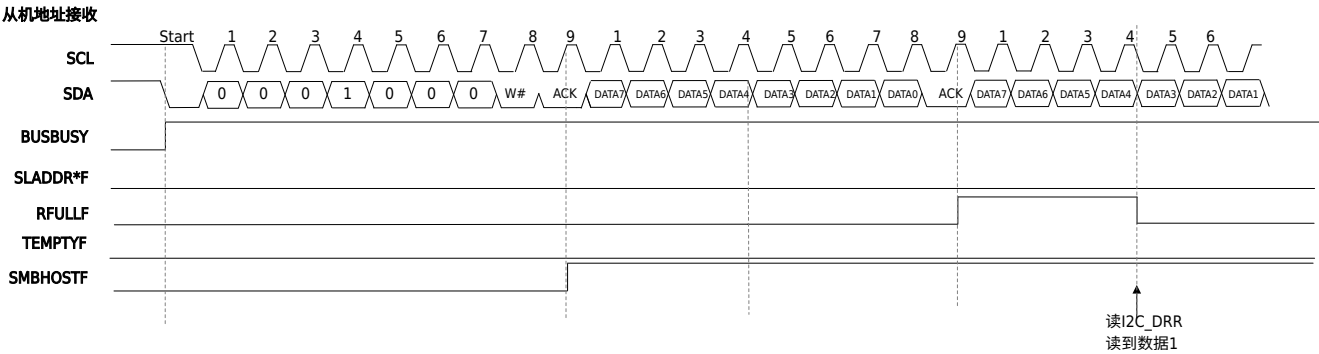


图 28-17 SMBus 主机地址匹配时序图

28.3.3.4 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=0 且 I2C_SR.TRA=0) 中检测 SMBUS 报警响应地址 (0b0001100)。

如果检测到 SMBUS 报警响应地址，就在 SCL 时钟的第 9 个时钟的上升沿将 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。

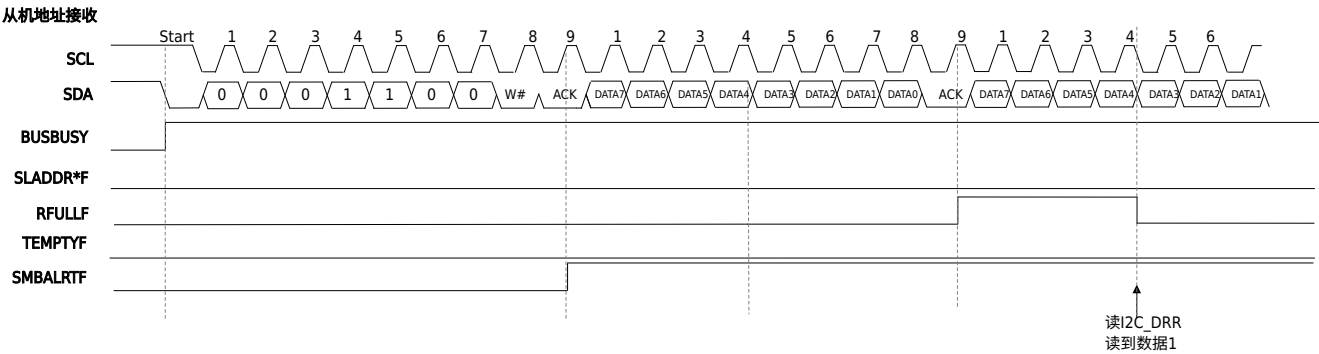


图 28-18 SMBus 报警响应地址匹配时序图

28.3.3.5 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=0 且 I2C_SR.TRA=0) 中检测 SMBUS 默认地址 (0b1100001)。

如果检测到 SMBUS 默认地址，就在 SCL 时钟的第 9 个时钟的上升沿将 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。

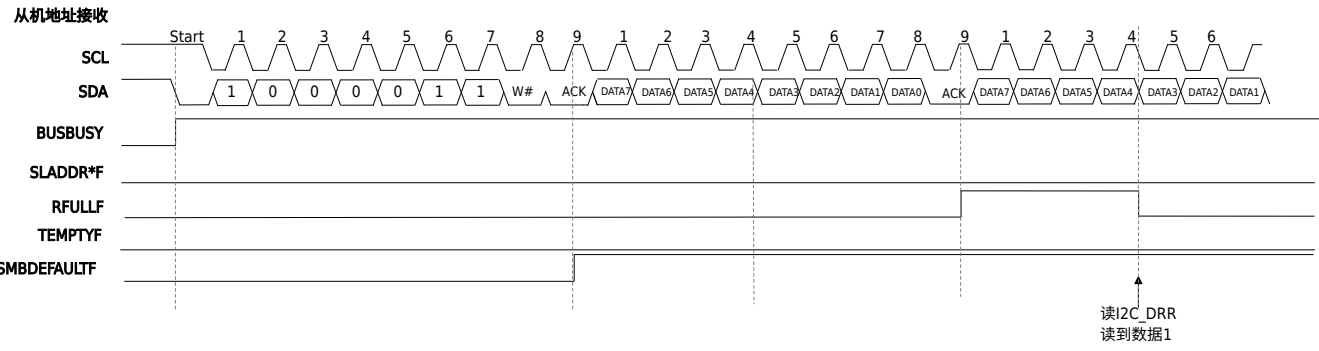


图 28-19 SMBus 默认地址时序图

28.3.4 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。

28.3.4.1 SMBus 超时测量

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 先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。

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: $T_{\text{LOW:SEXT}}$):

- 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] $T_{\text{LOW:SEXT}}$: 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 T_{TIMEOUT} : 25ms (min), 从机就需要释放总线。

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: $T_{\text{LOW:MEXT}}$):

- 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] $T_{\text{LOW:MEXT}}$: 10ms (max) 以内, 开始条件到停止条件的全部 $T_{\text{LOW:MEXT}}$ 的累加结果必须在 $T_{\text{LOW:SEXT}}$: 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] $T_{\text{LOW:MEXT}}$: 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 T_{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。

28.3.4.2 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。

28.3.5 复位

具有对通信模块进行复位的功能。有 2 种复位, 一种是通信复位: I2C 内部的全部寄存器 (I2C_FSTR[3:0]除外) 及内部状态复位; 另一种内部状态复位: I2C_SR (包括 BUSY 位), 内部移位寄存器和内部状态机进行复位。

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。



注意

在复位（I2C_CR1.PE=0 且 I2C_CR1.SWRST=1）过程中不能监视开始条件等的总线状态。

28.3.6 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。

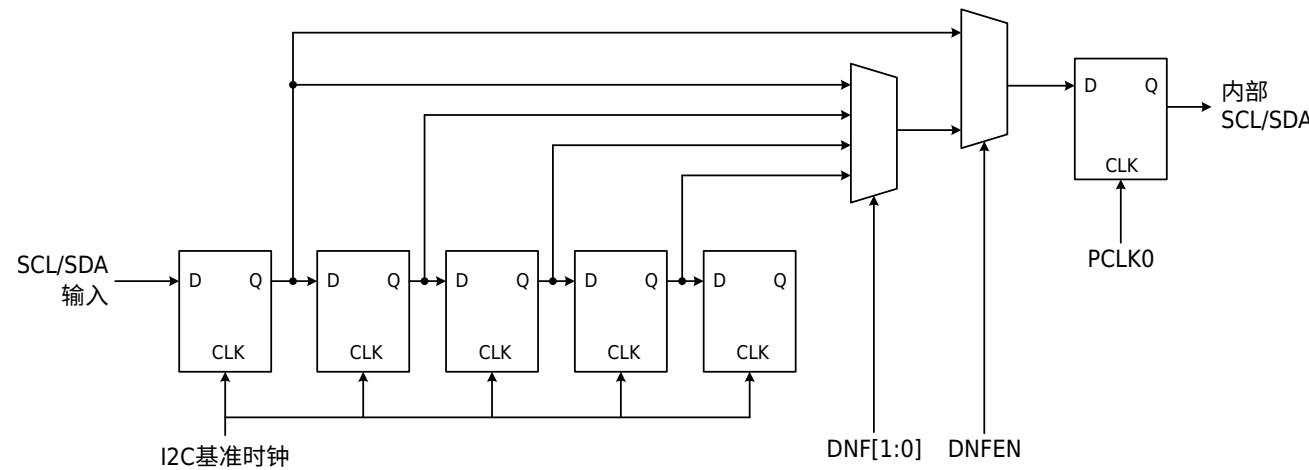


图 28-20 数字滤波电路框图

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。

28.3.7 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 28-2 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

名称	中断源	中断标志	中断条件
I2C_EEI	通信错误/通信事件	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 28-3 事件信号输出一览表

名称	事件源	事件条件
I2C_EEI	通信错误/通信事件	ARLOF=1
		SLADDR0F=1
		SLADDR1F=1
		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

28.4 操作示例

28.4.1 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：

- 步骤 1. I2C_CR1.PE 位设定为 0；
- 步骤 2. I2C_CR1.SWRST 设定为 1，通信复位；
- 步骤 3. I2C_CR1.PE 位设定为 1，内部状态复位；
- 步骤 4. 设定从机地址格式和地址；
- 步骤 5. 设定波特率；
- 步骤 6. 根据需要设定控制寄存器功能及中断；
- 步骤 7. I2C_CR1.SWRST 位设定为 0，解除内部状态复位；
- 步骤 8. 初始化结束。可发送接收数据。

28.5 寄存器

28.5.1 寄存器总表

- I2C1 基地址: 0x4003B400
- I2C2 基地址: 0x4003B800

表 28-4 集成电路总线 (I2C) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	I2C_CR1	I2C 控制寄存器 1	32
0x04	I2C_CR2	I2C 控制寄存器 2	32
0x08	I2C_CR3	I2C 控制寄存器 3	32
0x0C	I2C_CR4	I2C 控制寄存器 4	32
0x10	I2C_SLR0	I2C 从机地址寄存器 0	32
0x14	I2C_SLR1	I2C 从机地址寄存器 1	32
0x18	I2C_SLTR	I2C SCL 电平超时控制寄存器	32
0x1C	I2C_SR	I2C 状态寄存器	32
0x20	I2C_CLR	I2C 状态清零寄存器	32
0x24	I2C_DTR	I2C 数据发送寄存器	8
0x28	I2C_DRR	I2C 数据接收寄存器	8
0x2C	I2C_CCR	I2C 时钟控制寄存器	32
0x30	I2C_FLTR	I2C 滤波控制寄存器	32
0x34	I2C_FSTR	I2C FIFO 状态寄存器	32
0x38	I2C_SLVADRR	I2C 从机地址接收寄存器	32

28.5.2 I2C 控制寄存器 1 (I2C_CR1)

Offset	Bit Position																																																																																																																																																																																																																																																																																																																																																																	
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																																																																																																																																																																																																																																																																		
Reset	0x00000040																																																																																																																																																																																																																																																																																																																																																																	
Name	Reserved																SWRST	Reserved																ACK	STOP	START	RESTART	GCEN	Reserved	SMBHOSTEN	SMBDEFAULTEN	SMBALRTEN	SMBUS	PE																																																																																																																																																																																																																																																																																																																						
Access																	RW																	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写									
31:16	Reserved	保留	-	-									
15	SWRST	软件复位	● 0b0：解除复位 ● 0b1：软件复位 通过和 PE 位组合，选择内部状态复位或者通信复位： <table><tr><th>SWRST</th><th>PE</th><th>复位内容</th></tr><tr><td>0b1</td><td>0b0</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>0b1</td><td>0b1</td><td>内部状态复位：I2C_SR（包括 BUSY 位），内部移位寄存器和内部状态机进行复位</td></tr></table>	SWRST	PE	复位内容	0b1	0b0	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	0b1	0b1	内部状态复位：I2C_SR（包括 BUSY 位），内部移位寄存器和内部状态机进行复位	RW
SWRST	PE	复位内容											
0b1	0b0	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											
0b1	0b1	内部状态复位：I2C_SR（包括 BUSY 位），内部移位寄存器和内部状态机进行复位											
14:11	Reserved	保留	-	-									
10	ACK	发送应答	● 0b0：应答位发送“0”（发送 ACK） ● 0b1：应答位发送“1”（发送 NACK）	RW									
9	STOP	停止条件生成位	● 0b0：不生成停止条件 ● 0b1：生成停止条件 此位可软件置位和清零。 硬件清零条件： ● 检测到停止条件 ● 仲裁失败 ● 检测到开始条件 ● 通信复位	RW									
8	START	起始条件生成位	● 0b0：不生成起始条件 ● 0b1：生成起始条件 此位可软件置位和清零。 硬件清零条件： ● 检测到开始条件 ● 仲裁失败时 ● 通信复位	RW									

位/位域	标记	位名	功能描述	读写
7	RESTART	重复起始条件生成位	● 0b0: 不生成重复起始条件 ● 0b1: 生成重复起始条件 此位可软件置位和清零。 硬件清零条件: ● 检测到开始条件 ● 仲裁失败时 ● 通信复位	RW
6	GCEN	广播呼叫使能	● 0b0: 广播地址检测无效 ● 0b1: 广播地址检测有效	RW
5	<i>Reserved</i>	<i>保留</i>	-	-
4	SMBHOSTEN	允许匹配 SMBUS 主机地址位	● 0b0: 禁止匹配 SMBUS 主机地址 ● 0b1: 允许匹配 SMBUS 主机地址	RW
3	SMBDEFAULTEN	允许匹配 SMBUS 默认地址位	● 0b0: 禁止匹配 SMBUS 默认地址 ● 0b1: 允许匹配 SMBUS 默认地址	RW
2	SMBALRTEN	允许匹配 SMBUS 报警响应地址位	● 0b0: 禁止 SMBUS 报警响应地址 ● 0b1: 允许 SMBUS 报警响应地址	RW
1	SMBUS	SMBUS/I2C 总线模式选择位	● 0b0: I2C 总线模式 ● 0b1: SMBUS 总线模式	RW
0	PE	I2C 功能使能	通过和 SWRST 位组合, 选择内部状态复位或者通信复位: ● 0b0: I2C 功能禁止 ● 0b1: I2C 功能允许	RW

28.5.3 I2C 控制寄存器 2 (I2C_CR2)

Offset	Bit Position																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
Reset	0x00000000																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
Name	Reserved								SMBALRTIE				SMBHOSTIE				SMBDEFAULTIE				GENCALLIE				Reserved								TMOUTIE		Reserved		NACKIE		RFREQIE		Reserved		ARLOIE		Reserved		EMPTYIE		RFULLIE		Reserved		STOPIE		TENDIE		SLADDR1IE		SLADDR0IE		STARTIE																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														

位/位域	标记	位名	功能描述	读写
31:24	Reserved	保留	-	-
23	SMBALRTIE	SMBUS 报警响应地址匹配一致中断使能	0b0: 中断禁止 0b1: 中断允许	RW
22	SMBHOSTIE	SMBUS 主机地址匹配一致中断使能	0b0: 中断禁止 0b1: 中断允许	RW
21	SMBDEFAULTIE	SMBUS 默认地址匹配一致中断使能	0b0: 中断禁止 0b1: 中断允许	RW
20	GENCALLIE	广播呼叫地址匹配一致中断使能	0b0: 中断禁止 0b1: 中断允许	RW
19:15	Reserved	保留	-	-
14	TMOUTIE	超时中断使能	0b0: 中断禁止 0b1: 中断允许	RW
13	Reserved	保留	-	-
12	NACKIE	NACK 中断使能	0b0: 中断禁止 0b1: 中断允许	RW
11	RFREQIE	FIFO 接收请求中断使能	0b0: 中断禁止 0b1: 中断允许	RW
10	Reserved	保留	-	-
9	ARLOIE	仲裁失败中断使能	0b0: 中断禁止 0b1: 中断允许	RW
8	Reserved	保留	-	-
7	EMPTYIE	发送数据空中断使能	0b0: 中断禁止 0b1: 中断允许	RW
6	RFULLIE	接收数据满中断使能	0b0: 中断禁止 0b1: 中断允许	RW
5	Reserved	保留	-	-
4	STOPIE	停止条件中断使能	0b0: 中断禁止 0b1: 中断允许	RW

位/位域	标记	位名	功能描述	读写
3	TENDIE	发送一帧数据结束中断使能	0b0: 中断禁止 0b1: 中断允许	RW
2	SLADDR1IE	从机地址 1 匹配一致中断使能	0b0: 中断禁止 0b1: 中断允许	RW
1	SLADDR0IE	从机地址 0 匹配一致中断使能	0b0: 中断禁止 0b1: 中断允许	RW
0	STARTIE	开始条件/重新开始条件中断使能	0b0: 中断禁止 0b1: 中断允许	RW

28.5.4 I2C 控制寄存器 3 (I2C_CR3)

Offset	Bit Position																															
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000006																															
Name	Reserved																							FACKEN	Reserved			HTMOUT	LTMOUT	TMOUTEN		
Access																								RW				RW	RW	RW		

位/位域	标记	位名	功能描述	读写
31:8	Reserved	保留	-	-
7	FACKEN	RFULLF 标志位置位时间点选择	0b0: 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) 0b1: 在 SCL 时钟的第 8 个时钟上升时, RFULLF 标志位为“1” (在第 8 个时钟的下降沿, SCL 线保持低电平)  说明 通过写 ACK 位来解除保持的低电平。	RW
6:3	Reserved	保留	-	-
2	HTMOUT	高电平超时检测允许	0b0: 在 SCL 线为高电平时, 禁止超时检测 0b1: 在 SCL 线为高电平时, 允许超时检测	RW
1	LTMOUT	低电平超时检测允许	0b0: 在 SCL 线为低电平时, 禁止超时检测 0b1: 在 SCL 线为低电平时, 允许超时检测	RW
0	TMOUTEN	超时功能允许位	0b0: 检测 SCL 电平超时功能禁止 0b1: 检测 SCL 电平超时功能允许	RW

28.5.5 I2C 控制寄存器 4 (I2C_CR4)

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00301B07																															
Name	Reserved												SDADLY		Reserved		BUSFREE_CLREN		Reserved		BUSWAIT		Reserved									
Access													RW				RW				RW											

位/位域	标记	位名	功能描述	读写
31:22	Reserved	保留	-	-
21:20	Reserved	保留	所有位读出为 1, 写入时必须写 1	-
19:16	SDADLY	SDA 输出延迟计数器	0b0000: 没有输出延迟 0b0001: 1 个 I2C 基准时钟周期 0b0010: 2 个 I2C 基准时钟周期 0b1111: 15 个 I2C 基准时钟周期	RW
15:13	Reserved	保留	-	-
12	BUSFREE_CLREN	总线结束状态清除位	0b0: 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 0b1: 使能 I2C 总线 STOP 标志对 I2C_SLVADRR 从机接收地址寄存器以及 I2C_SR 状态寄存器中的 SMBALRTF, SMBHOSTF, SMBDEFAULTF, GENCALLF, MSL, TENDF, SLADDR1F, SLADDR0F, STARTF 位进行清除	RW
11	Reserved	保留	读出为 1, 写入时必须写 1	-
10	BUSWAIT	总线等待位	0b0: I2C_DRR 接收满、内部移位寄存器为空时, 在第 9 个时钟和下次传送的第 1 个时钟之间不保持低电平, 继续接收下一个数据 0b1: I2C_DRR 接收满、内部移位寄存器为空时, 在第 9 个时钟和下次传送的第 1 个时钟之间保持低电平, 通过读 I2C_DRR 寄存器来解除保持的低电平	RW
9:8	Reserved	保留	所有位读出为 1, 写入时必须写 1	-
7:3	Reserved	保留	-	-
2:0	Reserved	保留	所有位读出为 1, 写入时必须写 1	-

28.5.6 I2C 从机地址寄存器 0 (I2C_SLR0)

Offset	Bit Position																															
0x10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00001000																															
Name	Reserved		MASK0EN	MSLADDR0										ADDRMOD0	Reserved		SLADDR0EN	Reserved		SLADDR0												
Access			RW	RW										RW		RW			RW													

位/位域	标记	位名	功能描述	读写
31:27	Reserved	保留	-	-
26	MASK0EN	从机地址 0 掩码有效位	0b0: 从机地址寄存器 0 掩码设定值无效 0b1: 从机地址寄存器 0 掩码设定值有效	RW
25:24	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=0, MSLADDR0[9:8]设定的掩码无效。 - 当 MASK0EN=1, 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=0 时, MSLADDR0[9:8]位无效; - 当 ADDRMOD0=1 时, MSLADDR0[9:8]为 10 位从机地址的高两位掩码地址。	RW
23:16	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=0, MSLADDR0[7:0]设定的掩码无效。 - 当 MASK0EN=1, MSLADDR0[7:0]设定的掩码有效。其有效范围与 ADDRMOD0 所设值成正相关即: - 当 ADDRMOD0=0 时, MSLADDR0[7:1]为 7 位从机地址掩码, MSLADDR0[0]位无效; - 当 ADDRMOD0=1 时, MSLADDR0[7:0]为 10 位从机地址的低 8 位地址掩码。	RW
15	ADDRMOD0	7 位/10 位地址格式选择位	0b0: 选择 7 位地址格式 0b1: 选择 10 位地址格式	RW
14:13	Reserved	保留	-	-
12	SLADDR0EN	从机地址 0 有效位	0b0: 从机地址寄存器 0 设定值无效 0b1: 从机地址寄存器 0 设定值有效	RW
11:10	Reserved	保留	-	-
9:8	SLADDR0[9:8]	10 位从机地址的高位	设定从机地址: - 当 ADDRMOD0=0 时, SLADDR0[9:8]设定无效; - 当 ADDRMOD0=1 时, SLADDR0[9:8]为 10 位从机地址的高两位。	RW

位/位域	标记	位名	功能描述	读写
7:0	SLADDR0[7:0]	7 位地址/10 位地址的低位	设定从机地址： ● 当 ADDRMOD0=0 时，SLADDR0[7:1]为 7 位从机地址，SLADDR0[0]位无效； ● 当 ADDRMOD0=1 时，SLADDR0[7:0]为 10 位从机地址的低 8 位地址。	

28.5.7 I2C 从机地址寄存器 1 (I2C_SLR1)

Offset	Bit Position																															
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved		MASK1EN	MSLADDR1										ADDRMOD1	Reserved		SLADDR1EN	Reserved		SLADDR1												
Access														RW			RW													RW		

位/位域	标记	位名	功能描述	读写
31:27	Reserved	保留	-	-
26	MASK1EN	从机地址 1 掩码有效位	0b0: 从机地址寄存器 1 掩码设定值无效 0b1: 从机地址寄存器 1 掩码设定值有效	RW
25:24	MSLADDR1[9:8]	10 位从机地址的高位掩码地址	设定从机地址 1 的高位掩码地址: - 当 MASK1EN=0, MSLADDR1[9:8]设定的掩码无效。 - 当 MASK1EN=1, 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=0 时, MSLADDR1[9:8]位无效; - 当 ADDRMOD1=1 时, MSLADDR1[9:8]为 10 位从机地址的高两位掩码地址。	RW
23:16	MSLADDR1[7:0]	7 位地址/10 位地址的低位掩码地址	设定从机地址 1 的低位掩码地址: - 当 MASK1EN=0, MSLADDR1[7:0]设定的掩码无效。 - 当 MASK1EN=1, MSLADDR1[7:0]设定的掩码有效。其有效范围与 ADDRMOD1 所设值成正相关即: - 当 ADDRMOD1=0 时, MSLADDR1[7:1]为 7 位从机地址掩码, MSLADDR1[0]位无效; - 当 ADDRMOD1=1 时, MSLADDR1[7:0]为 10 位从机地址的低 8 位地址掩码。	RW
15	ADDRMOD1	7 位/10 位地址格式选择位	0b0: 选择 7 位地址格式 0b1: 选择 10 位地址格式	RW
14:13	Reserved	保留	-	-
12	SLADDR1EN	从机地址 1 有效位	0b0: 从机地址寄存器 1 设定值无效 0b1: 从机地址寄存器 1 设定值有效	RW
11:10	Reserved	保留	-	-
9:8	SLADDR1[9:8]	10 位从机地址的高位	设定从机地址: - 当 ADDRMOD1=0 时, SLADDR1[9:8]设定无效; - 当 ADDRMOD1=1 时, SLADDR1[9:8]为 10 位从机地址的高两位。	RW

位/位域	标记	位名	功能描述	读写
7:0	SLADDR1[7:0]	7 位地址/10 位地址的低位	设定从机地址： - 当 ADDRMOD1=0 时，SLADDR1[7:1]为 7 位从机地址，SLADDR1[0]位无效； - 当 ADDRMOD1=1 时，SLADDR1[7:0]为 10 位从机地址的低 8 位地址。	

28.5.8 I2C SCL 电平超时控制寄存器 (I2C_SLTR)

Offset	Bit Position																															
0x18	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	TOUTHIGH																TOUTLOW															
Access	RW																RW															

位/位域	标记	位名	功能描述	读写
31:16	TOUTHIGH	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
15:0	TOUTLOW	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

28.5.9 I2C 状态寄存器 (I2C_SR)

Offset	Bit Position																																
0x1C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x05000000																																
Name	RFREQ	Reserved		TFST		RFFULL	RFEMPTY	TFFULL	TFEMPTY	RO SMBALRTF	RO SMBHOSTF	RO SMBDEFAULTF	RO GENCALLF	Reserved	TRA	RO BUSY	RW MSL	Reserved	TMOUTF	Reserved	NACKF	Reserved	ACKRF	RO ARLOF	Reserved	TEMTYF	RO RFULLF	Reserved	RO STOPF	RO TENDF	RO SLADDR1F	RO SLADDR0F	RO STARTF
Access	RO			RO		RO	RO	RO	RO	RO	RO	RO	RO		RW	RO	RW		RO		RO		RO	RO		RO	RO		RO	RO	RO	RO	RO

位/位域	标记	位名	功能描述	读写
31	RFREQ	接收 FIFO 请求标志位	0b0: 当前接收 FIFO 无有效数据 0b1: 当前接收 FIFO 有有效数据	RO
30	Reserved	保留	-	-
29:28	TFST	发送 FIFO 状态标志位	0b00: 当前发送 FIFO 已空 0b01: 当前发送 FIFO 有 1Byte 有效数据 0b10: 当前发送 FIFO 有 2Bytes 有效数据 0b11: Reserved	RO
27	RFFULL	接收 FIFO 满标志位	0b0: 当前接收 FIFO 未滿 0b1: 当前接收 FIFO 已滿	RO
26	RFEMPTY	接收 FIFO 空标志位	0b0: 当前接收 FIFO 未空 0b1: 当前接收 FIFO 已空	RO
25	TFFULL	发送 FIFO 满标志位	0b0: 当前发送 FIFO 未滿 0b1: 当前发送 FIFO 已滿	RO
24	TFEMPTY	发送 FIFO 空标志位	0b0: 当前发送 FIFO 未空 0b1: 当前发送 FIFO 已空	RO
23	SMBALRTF	SMBUS 报警响应地址匹配一致标志位	0b0: 未匹配到 SMBUS 报警响应地址 0b1: 检测到主机地址 置位条件: - 接收到的地址和 0b0001100 匹配一致 清零条件: - SMBALRTFCLR 写“1” - 检测到停止条件 - 通信复位 - 内部状态复位	RO
22	SMBHOSTF	SMBUS 主机地址匹配一致标志位	0b0: 未匹配到 SMBUS 主机地址 0b1: 匹配到 SMBUS 主机地址 置位条件: - 接收到的地址和 0b0001000 匹配一致 清零条件: - SMBHOSTFCLR 写“1” - 检测到停止条件 - 通信复位 - 内部状态复位	RO

位/位域	标记	位名	功能描述	读写															
21	SMBDEFAULTF	SMBUS 默认地址匹配一致标志位	0b0：未匹配到 SMBUS 默认地址 0b1：匹配到 SMBUS 默认地址 置位条件： - 接收到的地址和 0b1100001 匹配一致 清零条件： - SMBDEFAULTFCLR 写“1” - 检测到停止条件 - 通信复位 - 内部状态复位	RO															
20	GENCALLF	广播呼叫地址匹配一致标志位	0b0：未匹配到广播呼叫地址 0b1：匹配到广播呼叫地址 置位条件： - 接收的从机地址与广播呼叫地址（All“0”）匹配一致 清零条件： - GENCALLFCLR 写“1” - 检测到停止条件 - 通信复位 - 内部状态复位	RO															
19	Reserved	保留	-	-															
18	TRA	发送接收选择位	0b0：接收数据 0b1：发送数据 通过和 MSL 位的组合，表示 I2C 的运行模式： <table border="1"><thead><tr><th>MSL</th><th>TRA</th><th>I2C 运行模式</th></tr></thead><tbody><tr><td>0b0</td><td>0b0</td><td>从机接收模式</td></tr><tr><td>0b0</td><td>0b1</td><td>从机发送模式</td></tr><tr><td>0b1</td><td>0b0</td><td>主机接收模式</td></tr><tr><td>0b1</td><td>0b1</td><td>主机发送模式</td></tr></tbody></table> 此位可软件置位和清零。 硬件置位条件： - 检测到开始条件 - 主机模式下，发送的 RW 位为 0 - 从机模式下，地址匹配且接收的 RW 位为 1 硬件清零条件： - 检测到停止条件 - 主机模式下，发送的 RW 位为 1 - 从机模式下，地址匹配且接收的 RW 位为 0 - 通信复位 - 内部状态复位	MSL	TRA	I2C 运行模式	0b0	0b0	从机接收模式	0b0	0b1	从机发送模式	0b1	0b0	主机接收模式	0b1	0b1	主机发送模式	RW
MSL	TRA	I2C 运行模式																	
0b0	0b0	从机接收模式																	
0b0	0b1	从机发送模式																	
0b1	0b0	主机接收模式																	
0b1	0b1	主机发送模式																	
17	BUSY	总线忙标志位	0b0：空闲状态，总线上无通信 0b1：占有状态，总线正在通信 置位条件： - 检测到总线上开始条件 清零条件： - 检测到总线停止条件 - 通信复位	RO															

位/位域	标记	位名	功能描述	读写
16	MSL	主从机选择位	● 0b0: 从机模式 ● 0b1: 主机模式 通过和 TRA 位的组合, 表示 I2C 的运行模式。 此位可软件置位和清零。 硬件置位条件: ● START 位为 1 的状态下, 检测到开始条件 硬件清零条件: ● 检测到停止条件 ● 仲裁失败 ● 通信复位 ● 内部状态复位	RW
15	Reserved	保留	-	-
14	TMOUTF	超时标志位	● 0b0: 未检测到 SCL 电平超时 ● 0b1: SCL 电平超时 置位条件: ● I2C_SLTR 设定的周期内, SCL 未翻转 清零条件: ● TMOUTFCLR 写“1” ● 通信复位 ● 内部状态复位	RO
13	Reserved	保留	-	-
12	NACKF	NACK 标志位	● 0b0: 未接收到 NACK ● 0b1: 接收到 NACK 置位条件: ● 发送模式下, 接收到 NACK 清零条件: ● NACKFCLR 写“1” ● 通信复位 ● 内部状态复位	RO
11	Reserved	保留	-	-
10	ACKRF	接收应答位	● 0b0: 接收到应答位为“0” (接收 ACK) ● 0b1: 接收到应答位为“1” (接收 NACK) 置位条件: ● 发送模式下, 接收到 NACK 清零条件: ● 发送模式下, 接收到 ACK ● 通信复位 ● 内部状态复位	RO
9	ARLOF	仲裁失败标志位	● 0b0: 未发生仲裁失败 ● 0b1: 仲裁失败 置位条件: ● 仲裁失败 清零条件: ● ARLOFCLR 写“1” ● 通信复位 ● 内部状态复位	RO
8	Reserved	保留	-	-

位/位域	标记	位名	功能描述	读写
7	EMPTYF	发送数据空标志位	● 0b0: I2C_DTR 寄存器满 ● 0b1: I2C_DTR 寄存器空 置位条件: ● I2C_DTR 数据传送到内部移位寄存器 ● TRA 位置 1 清零条件: ● 写 I2C_DTR ● TRA 位清 0 ● 通信复位 ● 内部状态复位	RO
6	RFULLF	接收数据满标志位	● 0b0: I2C_DRR 寄存器空 ● 0b1: I2C_DRR 寄存器满 置位条件: ● 接收的数据从内部移位寄存器传送到 I2C_DRR 清零条件: ● 读 I2C_DRR ● RFULLFCLR 写“1” ● 通信复位 ● 内部状态复位	RO
5	<i>Reserved</i>	<i>保留</i>	-	-
4	STOPF	停止条件标志位	● 0b0: 总线未检测到停止条件 ● 0b1: 总线检测到停止条件 置位条件: ● 检测到停止条件 清零条件: ● STOPFCLR 写“1” ● 通信复位 ● 内部状态复位	RO
3	TENDF	发送数据结束标志位	● 0b0: 内部移位寄存器发送中 ● 0b1: 内部移位寄存器发送结束 置位条件: ● EMPTYF=1 的条件下, SCL 的第 9 个上升沿此位置“1” 清零条件: ● 检测到停止条件 ● 写 I2C_DTR ● TENDFCLR 写“1” ● 通信复位 ● 内部状态复位	RO
2	SLADDR1F	从机地址寄存器 1 匹配一致标志位	● 0b0: 未检测到从机地址寄存器 1 一致地址 ● 0b1: 检测到从机地址寄存器 1 一致地址 置位条件: ● 当 I2C_SLR1.ADDRM0D1=0 时, 接收到的从机地址与 I2C_SLR1.SLADDR1[7:1]匹配一致 ● 当 I2C_SLR1.ADDRM0D1=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	RO

位/位域	标记	位名	功能描述	读写
2	SLADDR1F	从机地址寄存器 1 匹配一致标志位	并且第二个字节地址与 I2C_SLR1.SLADDR1[7:0]匹配一致 清零条件： ● 检测到停止条件 ● SLADDR1FCLR 写“1” ● 通信复位 ● 内部状态复位	RO
1	SLADDR0F	从机地址寄存器 0 匹配一致标志位	● 0b0：未检测到从机地址寄存器 0 一致地址 ● 0b1：检测到从机地址寄存器 0 一致地址 置位条件： ● 当 I2C_SLR0.ADDRM0=0 时，接收到的从机地址与 I2C_SLR0.SLADDR0[7:1]匹配一致 ● 当 I2C_SLR0.ADDRM0=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 清零条件： ● 检测到停止条件 ● SLADDR0FCLR 写“1” ● 通信复位 ● 内部状态复位	RO
0	STARTF	开始条件/重新开始条件标志位	● 0b0：总线未检测到开始条件 ● 0b1：总线检测到开始条件 置位条件： ● 检测到开始条件 清零条件： ● 检测到停止条件 ● STARTFCLR 写“1” ● 通信复位 ● 内部状态复位	RO

28.5.10 I2C 状态清零寄存器 (I2C_CLR)

Offset	Bit Position																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
0x20	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
Reset	0x00000000																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
Name	Reserved								SMBALRTFCLR				SMBHOSTFCLR				SMBDEFAULTFCLR				GENCALLFCLR				Reserved								TMOUTFCLR				Reserved				NACKFCLR				Reserved				RFREQCLR				ARLOFCLR				Reserved								RFULLFCLR				Reserved				STOPFCLR				TENDFCLR				SLADDR1FCLR				SLADDR0FCLR				STARTFCLR																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	</			

位/位域	标记	位名	功能描述	读写
31:24	Reserved	保留	-	-
23	SMBALRTFCLR	SMBUS 报警响应地址匹配一致标志清除	写 1 清除 SMBALRTF 标志位, 写 0 无效	WO
22	SMBHOSTFCLR	SMBUS 主机地址匹配一致标志清除	写 1 清除 SMBHOSTF 标志位, 写 0 无效	WO
21	SMBDEFAULTFCLR	SMBUS 默认地址匹配一致标志清除	写 1 清除 SMBDEFAULTF 标志位, 写 0 无效	WO
20	GENCALLFCLR	广播呼叫地址匹配一致标志清除	写 1 清除 GENCALLF 标志位, 写 0 无效	WO
19:15	Reserved	保留	-	-
14	TMOUTFCLR	超时标志清除	写 1 清除 TMOUTF 标志位, 写 0 无效	WO
13	Reserved	保留	-	-
12	NACKFCLR	NACK 标志清除	写 1 清除 NACKF 标志位, 写 0 无效	WO
11	Reserved	保留	-	-
10	RFREQCLR	接收 FIFO 数据请求标志清除	写 1 清除 RFREQF 标志位, 写 0 无效	WO
9	ARLOFCLR	仲裁失败标志清除	写 1 清除 ARLOF 标志位, 写 0 无效	WO
8:7	Reserved	保留	-	-
6	RFULLFCLR	接收数据满标志清除	写 1 清除 RFULLF 标志位, 写 0 无效	WO
5	Reserved	保留	-	-
4	STOPFCLR	停止条件标志清除	写 1 清除 STOPF 标志位, 写 0 无效	WO
3	TENDFCLR	发送数据结束标志清除	写 1 清除 TENDF 标志位, 写 0 无效	WO
2	SLADDR1FCLR	从机地址寄存器 1 匹配一致标志清除	写 1 清除 SLADDR1F 标志位, 写 0 无效	WO
1	SLADDR0FCLR	从机地址寄存器 0 匹配一致标志清除	写 1 清除 SLADDR0F 标志位, 写 0 无效	WO
0	STARTFCLR	开始条件/重新开始条件标志清除	写 1 清除 STARTF 标志位, 写 0 无效	WO

28.5.11 I2C 数据发送寄存器 (I2C_DTR)

Offset	Bit Position							
0x24	7	6	5	4	3	2	1	0
Reset	0xFF							
Name	DT							
Access	RW							

位/位域	标记	位名	功能描述	读写
7:0	DT	发送数据	如果内部移位寄存器为空，就将写在 I2C_DTR 寄存器的发送数据传送到内部移位寄存器，发送模式时开始发送数据到 SDA 上	RW

内部移位寄存器和 I2C_DTR 寄存器是双缓冲结构，在内部移位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。

I2C 工作在 FIFO 模式时，访问 I2C_DTR 寄存器地址可以直接访问发送 FIFO，如果当前 TEMPTYF 有效，针对 I2C_DTR 寄存器地址的写操作可以将当前的总线数据写入发送 FIFO。当发送 FIFO 有效并且当前 I2C 允许发送则可以将发送 FIFO 数据传送到内部移位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 到内部移位寄存器的数据通路。
- 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 寄存器到内部移位寄存器的数据通路。

默认选择非 FIFO 模式。

28.5.12 I2C 数据接收寄存器 (I2C_DRR)

Offset	Bit Position							
0x28	7	6	5	4	3	2	1	0
Reset	0x00							
Name	DR							
Access	RO							

位/位域	标记	位名	功能描述	读写
7:0	DR	接收数据	如果接收到 1 帧数据，就能将接收数据从内部移位寄存器转存到 I2C_DRR 寄存器，进而可以进入到下一个数据接收状态	RO

内部移位寄存器和 I2C_DRR 寄存器是双缓冲结构，在内部移位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[从机接收时序图](#)和[快速 ACK/NACK 时序图](#)）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：

- 如果接收到 1 帧数据，就能将接收数据从内部移位寄存器转存到接收 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。
- 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 时选择内部移位寄存器到接收 FIFO 数据通路以及接收 FIFO 到数据总线的通路。
- 当 FEN 为 0 时选择内部移位寄存器到 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。

28.5.13 I2C 时钟控制寄存器 (I2C_CCR)

Offset	Bit Position																															
0x2C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00001F1F																															
Name	Reserved														CKDIV		SHIGHW								SLOWW							
Access															RW		RW								RW							

位/位域	标记	位名	功能描述	读写
31:19	Reserved	保留	-	-
18:16	CKDIV	I2C 基准时钟频率设定位	0b000: I2C 基准时钟频率=PCLK0 0b001: I2C 基准时钟频率=PCLK0/2 0b010: I2C 基准时钟频率=PCLK0/4 0b011: I2C 基准时钟频率=PCLK0/8 0b100: I2C 基准时钟频率=PCLK0/16 0b101: I2C 基准时钟频率=PCLK0/32 0b110: I2C 基准时钟频率=PCLK0/64 0b111: I2C 基准时钟频率=PCLK0/128	RW
15:8	SHIGHW	设定 SCL 高电平宽度位	设定 SCL 时钟的高电平宽度	RW
7:0	SLOWW	设定 SCL 低电平宽度位	设定 SCL 时钟的低电平宽度	RW

波特率计算如下:

I2C_FLTR.DNFEN	I2C_CCR.CKDIV	波特率
0b0	0b000	$B = \frac{1}{\frac{(SHIGHW + 3) + (SLOWW + 3)}{\Phi_{I2C}} + t_R + t_F}$
0b1	0b000	$B = \frac{1}{\frac{(SHIGHW + 3 + DNF) + (SLOWW + 3 + DNF)}{\Phi_{I2C}} + t_R + t_F}$
0b0	非 0	$B = \frac{1}{\frac{(SHIGHW + 2) + (SLOWW + 2)}{\Phi_{I2C}} + t_R + t_F}$
0b1	非 0	$B = \frac{1}{\frac{(SHIGHW + 2 + DNF) + (SLOWW + 2 + DNF)}{\Phi_{I2C}} + t_R + t_F}$

其中:

- B: 波特率, 单位: Mbps
- Φ_{I2C} : I2C 基准时钟频率, 单位: MHz
- DNF: 数字滤波器滤波能力, 通过寄存器 I2C_FLTR.DNF 设置
- t_R : SCL 线的上升时间, 单位: ns
- t_F : SCL 线的下降时间, 单位: ns



说明

- 在主机模式下，SHIGHW 用于设定 SCL 时钟的高电平宽度。在从机模式下，设定无效。
- 在主机模式下，SLOWW 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 Setup 时间 ($t_{SU,DAT}$)，对于标准模式为 250ns，对于快速模式为 100ns。
- SCL 线的上升时间 (t_R) 和下降时间 (t_F) 取决于总线的总电容量 (C_b) 和上拉电阻 (R_p)，详细内容请参见 I2C 总线规范。

28.5.14 I2C 滤波控制寄存器 (I2C_FLTR)

Offset	Bit Position																															
0x30	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000010																															
Name	Reserved																												Reserved			
Access																											RW	RW		RW		

位/位域	标记	位名	功能描述	读写
31:6	Reserved	保留	-	-
5	ANFEN	模拟滤波功能允许位	● 0b0: 模拟滤波功能禁止 ● 0b1: 模拟滤波功能允许	RW
4	DNFEN	数字滤波功能允许位	● 0b0: 数字滤波功能禁止 ● 0b1: 数字滤波功能允许	RW
3:2	Reserved	保留	-	-
1:0	DNF	数字滤波器滤波能力选择	● 0b00: 滤波能力 1 个 I2C 基准时钟周期 ● 0b01: 滤波能力 2 个 I2C 基准时钟周期 ● 0b10: 滤波能力 3 个 I2C 基准时钟周期 ● 0b11: 滤波能力 4 个 I2C 基准时钟周期	RW

28.5.15 I2C FIFO 状态寄存器 (I2C_FSTR)

Offset	Bit Position																																																											
0x34	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																												
Reset	0x00000000																																																											
Name	Reserved																								RFST		TFST		NACK_TFFLUSH		RFLUSH		TFLUSH		FEN																									
RFLUSH																													TFLUSH		FEN																													
Access																																																	RO		RO		RW		RW		RW		RW	

位/位域	标记	位名	功能描述	读写
31:8	Reserved	保留	-	-
7:6	RFST	接收 FIFO 状态位	0b00: FIFO 空 0b01: FIFO 半满 0b10: FIFO 满 0b11: Reserved	RO
5:4	TFST	发送 FIFO 状态位	0b00: FIFO 空 0b01: FIFO 半空 0b10: FIFO 满 0b11: Reserved	RO
3	NACK_TFFLUSH	NACK 复位发送 FIFO	0b0: 从模式下接收到 NACK 后不执行 TFIFO FLUSH 0b1: 从模式下接收到 NACK 后执行 TFIFO FLUSH	RW
2	RFLUSH	RX FIFO 指针复位	0b0: 写入无效 0b1: 复位 RX FIFO 指针	RW
1	TFLUSH	TX FIFO 指针复位	0b0: 写入无效 0b1: 复位 TX FIFO 指针	RW
0	FEN	FIFO 功能允许位	0b0: FIFO 功能禁止 0b1: FIFO 功能允许	RW

28.5.16 I2C 从机地址接收器寄存器 (I2C_SLVADDR)

Offset	Bit Position																															
0x38	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																				SLVADR											
Access																					RO											

位/位域	标记	位名	功能描述	读写
31:10	Reserved	保留	-	-
9:0	SLVADR	从机地址接收寄存器	SLVADR[6:0]: 7 位从机接收地址 SLVADR[9:0]: 10 位从机接收地址	RO

29 串行外设接口 (SPI)

29.1 简介

本产品搭载 3 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 主要特性

表 29-1 SPI 主要特性

要点	描述
串行通信功能	● 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	● 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	● 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	● 带有 16 字节的数据缓冲器 ● 支持双重缓冲
错误监测	● 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	● 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	● 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	● 接收缓冲器已满 ● 发送缓冲器已空 ● 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	● 可设置模块停止

要点	描述
其他功能	● SPI 初始化功能

说明

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。详情请参见“[过载错误](#)”。

29.3 功能说明

29.3.1 系统框图

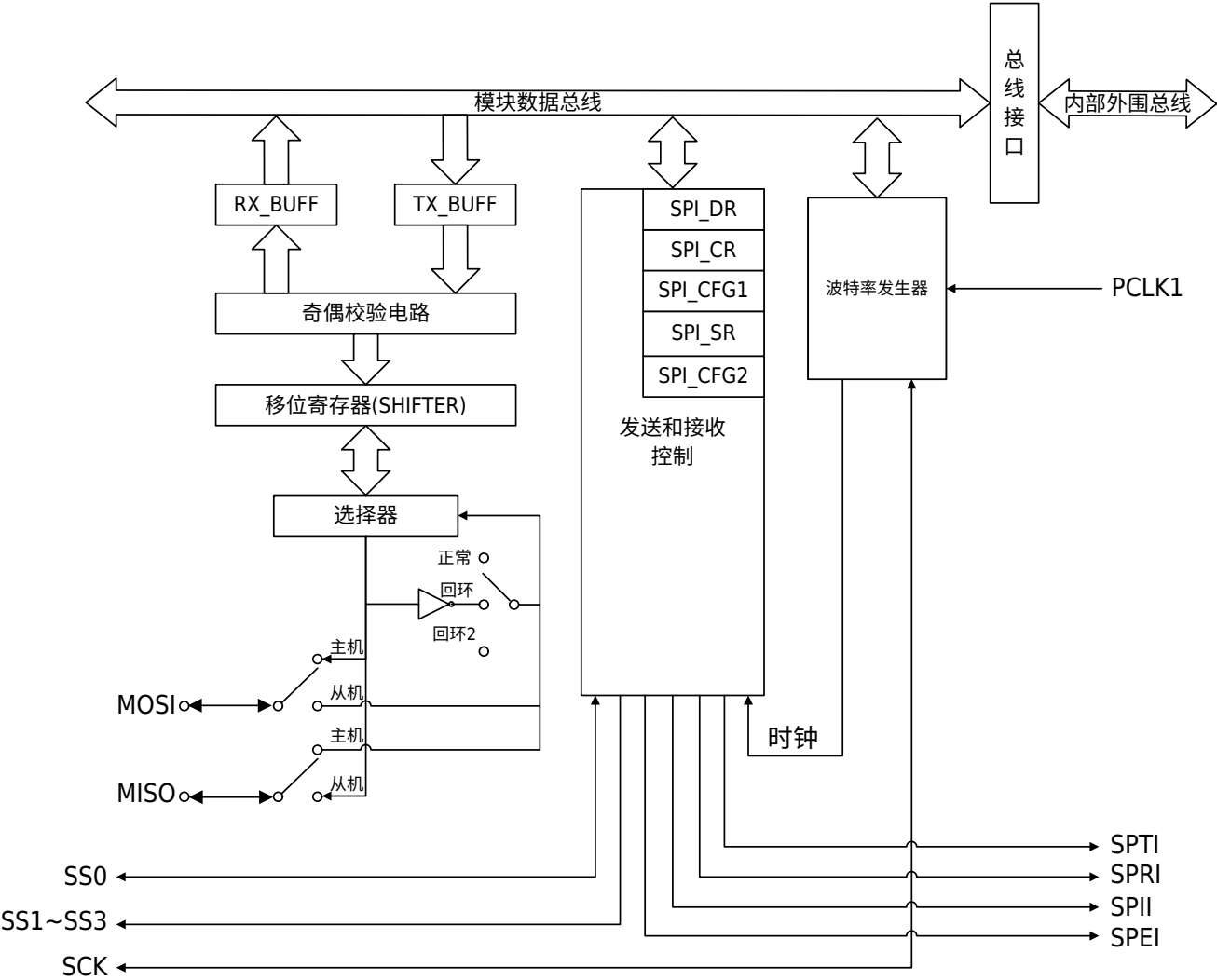


图 29-1 SPI 系统框图

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

管脚名	端口方向	功能
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

说明

管脚输入类型请设定为 CMOS 输入，输出类型请设定为高驱动力模式，相关设定请参见《通用 IO（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

模式		管脚名	管脚状态
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

 **说明**

管脚输入类型请设定为 CMOS 输入，输出类型请设定为高驱动力模式，相关设定请参见《通用 IO (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。

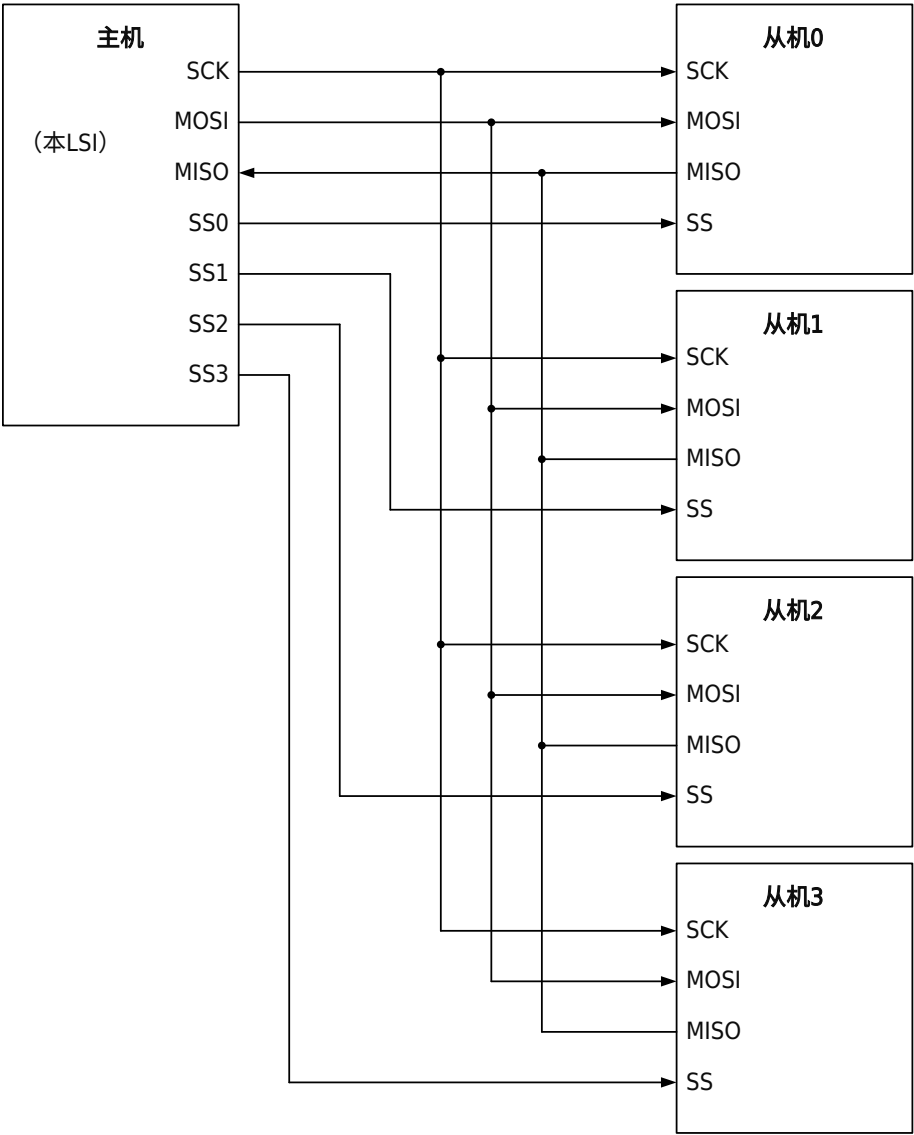


图 29-2 主机模式结构

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。

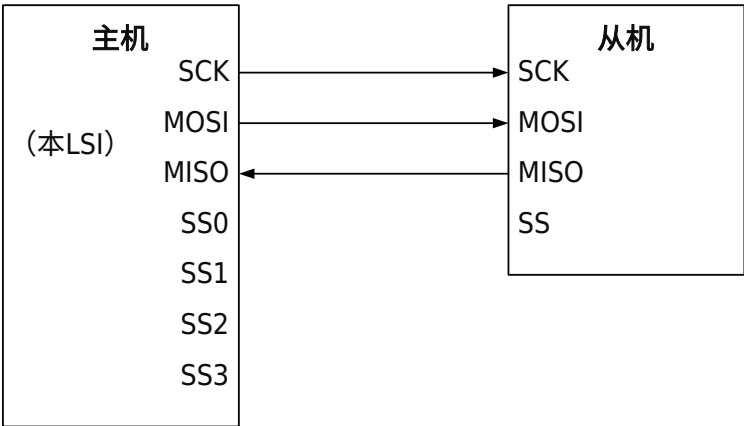


图 29-3 三线式时钟同步运行

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 位的设置。其计算方法如下面公式所示：

波特率 = $\frac{f_{PCLK1}}{2^{N+1} \times (M+1)}$

其中：

- N: SPI_CFG2.MBR 位的设定值，范围是 0~3；
- M: SPI_CFG1.CLKDIV 位的设定值，范围是 0~15。

表 29-5 部分设定值的位速

MBR 位的设定值 (N)	CLKDIV 位的设定 值 (M)	分频比	波特率			
			PCLK1=5MHz	PCLK1=10MHz	PCLK1=20MHz	PCLK1=40MHz
0	0	2	2.50Mbps	5.00Mbps	10.0Mbps	20.0Mbps
0	1	4	1.25Mbps	2.50Mbps	5.00Mbps	10.0Mbps
0	3	8	625kbps	1.25Mbps	2.50Mbps	5.00Mbps
0	7	16	313kbps	625kbps	1.25Mbps	2.50Mbps
0	15	32	156kbps	313kbps	625kbps	1.25Mbps
1	15	64	78kbps	156kbps	313kbps	625kbps
2	15	128	39kbps	78kbps	156kbps	313kbps
3	15	256	20kbps	39kbps	78kbps	156kbps

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 控制寄存器中的奇偶校验许可位（SPI_CR.PAE）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.DSIZE 位设置）的数据作为传送对象进行处理，与 MSB/LSB 移位顺序无关。

SPI_CFG2.DSIZE 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。

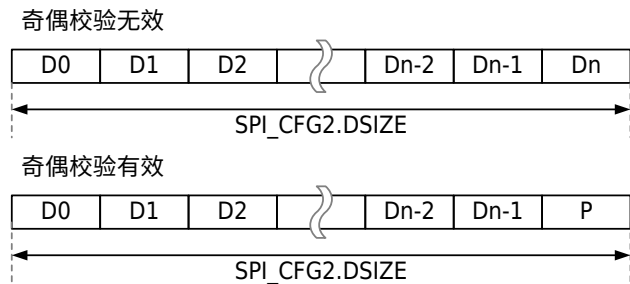


图 29-4 数据格式

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：

● **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。

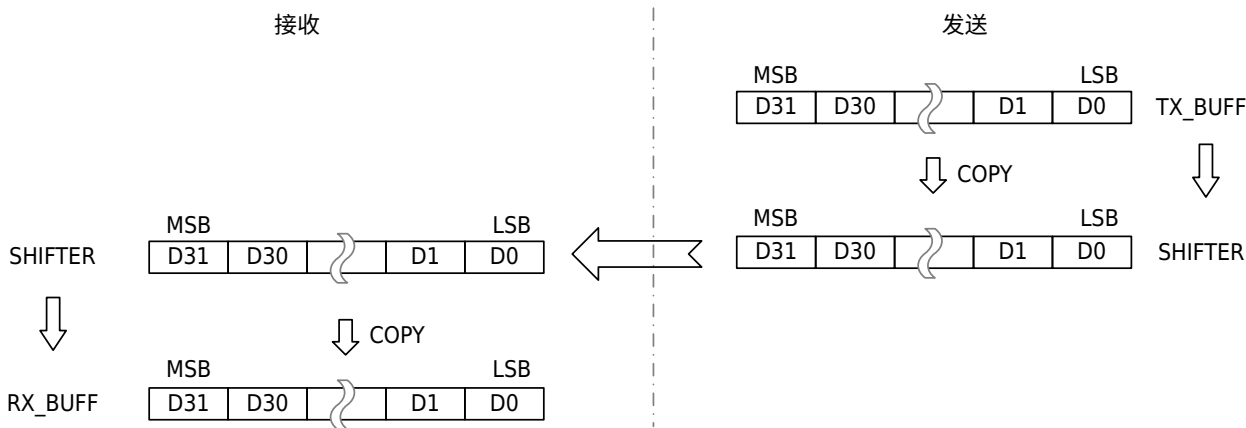


图 29-5 MSB 先传，奇偶校验无效

● **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。

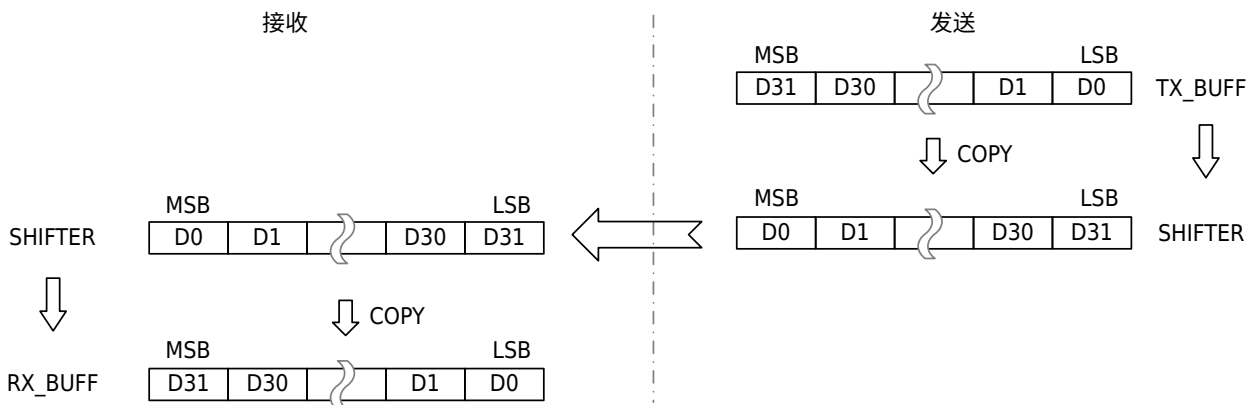


图 29-6 LSB 先传，奇偶校验无效

● MSB 先传，奇偶校验有效

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。

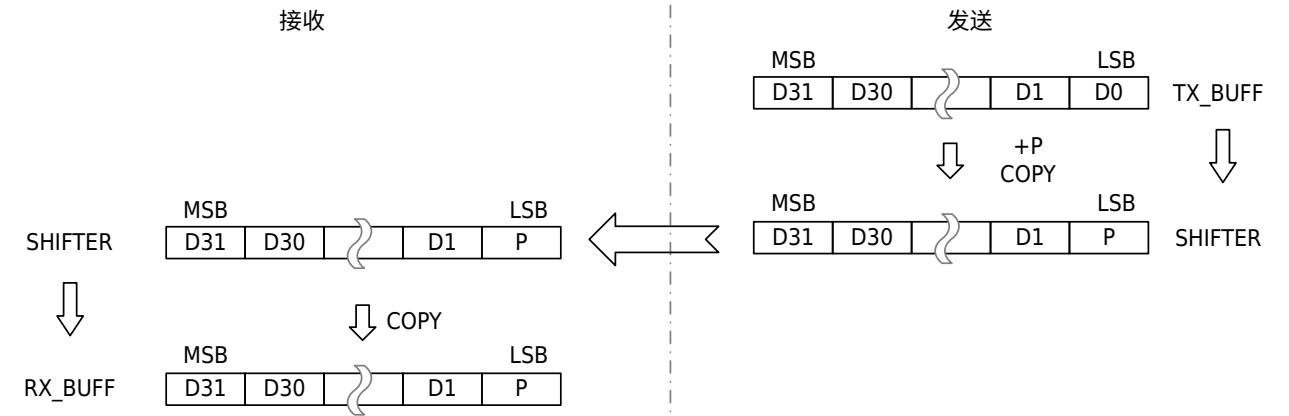


图 29-7 MSB 先传，奇偶校验有效

● LSB 先传，奇偶校验有效

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。

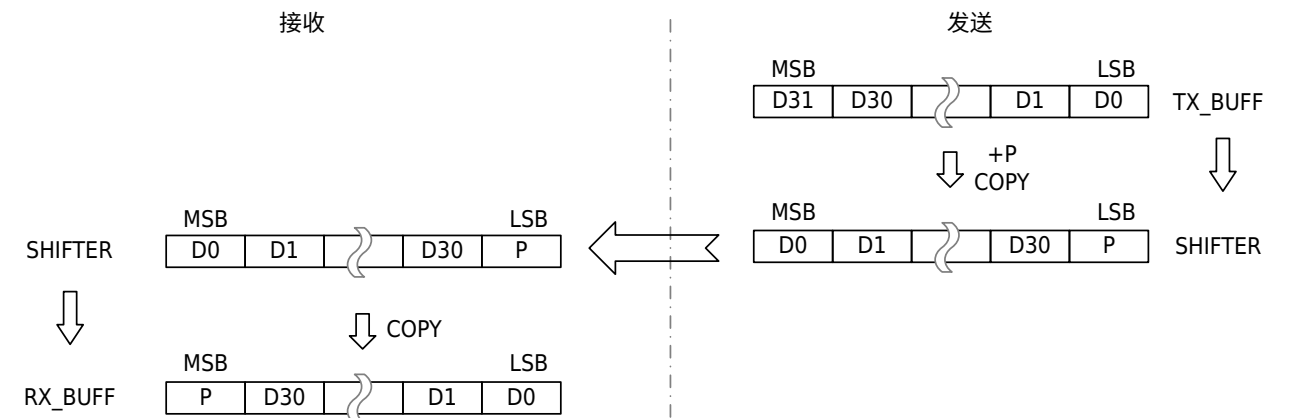


图 29-8 LSB 先传，奇偶校验有效

29.3.3.3 传送格式

CPHA=0 的情况

当 SPI_CFG2.CPHA 位为 0 时，SPI 在 SCK 的奇数边沿进行数据采样，偶数边沿进行数据更新。下图是 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。

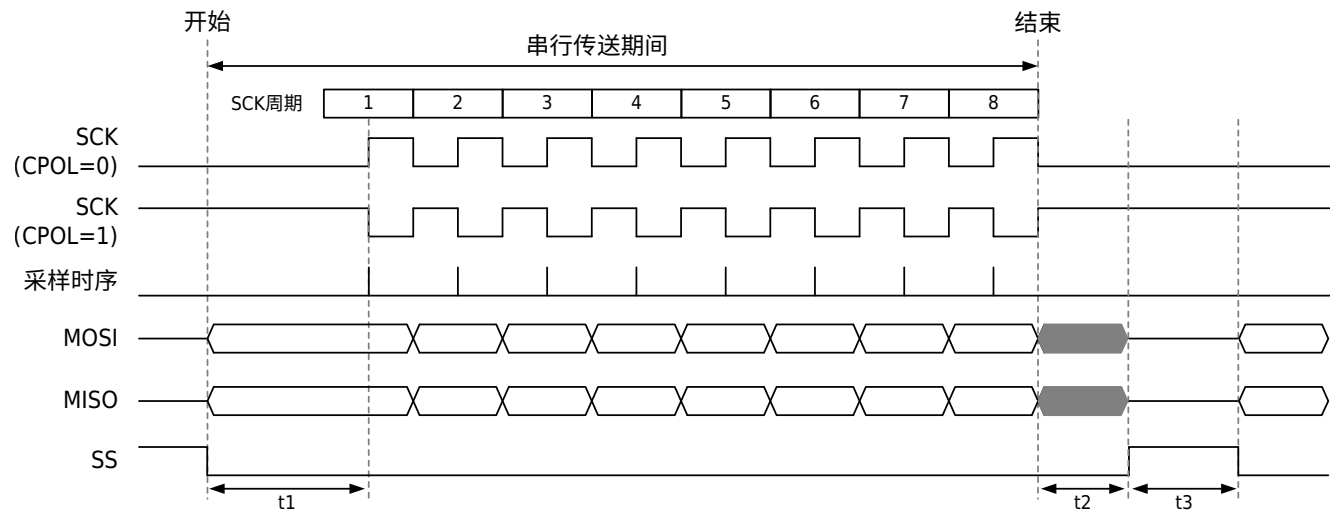


图 29-9 数据传送格式图 (CPHA=0)

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 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 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 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。详细请参见“SPI 运行模式时的主机动作”。

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。下图是 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。

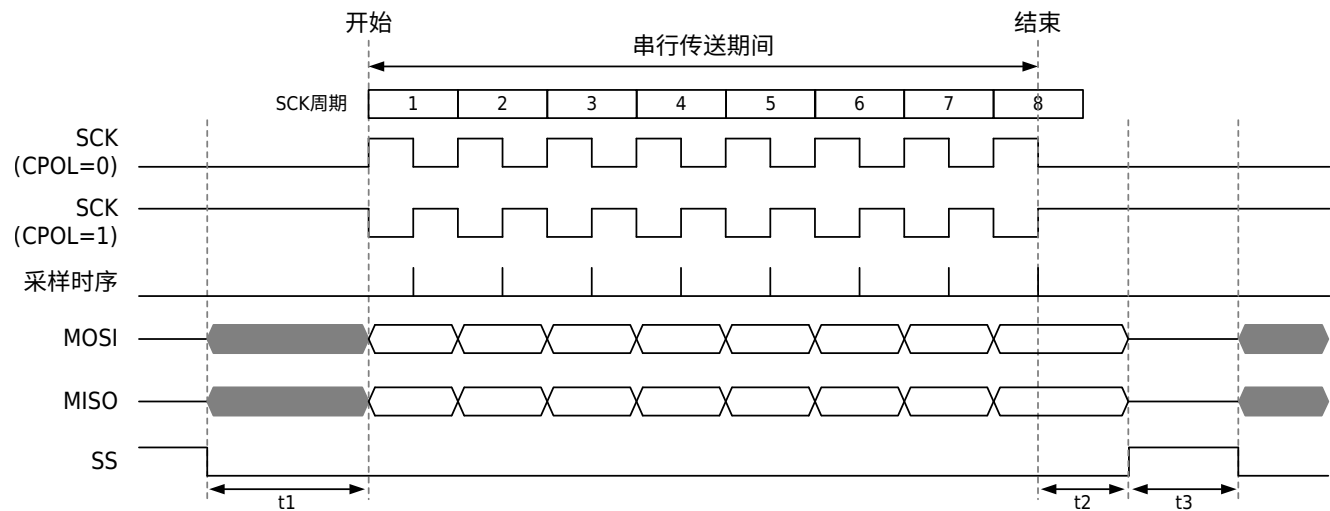


图 29-10 数据传送格式图 (CPHA=1)

t1、t2、t3 和 CPHA 位为 0 的情况相同。

29.3.3.4 通信方式

SPI 支持全双工同步串行通信、只发送串行通信, 可通过 SPI_CR.TXMDS 位进行选择。

全双工同步串行通信方式

当 SPI_CR.TXMDS 位为 0 时, SPI 运行在全双工同步串行通信方式。如下图所示, SPI_CFG1.FTHLV 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。

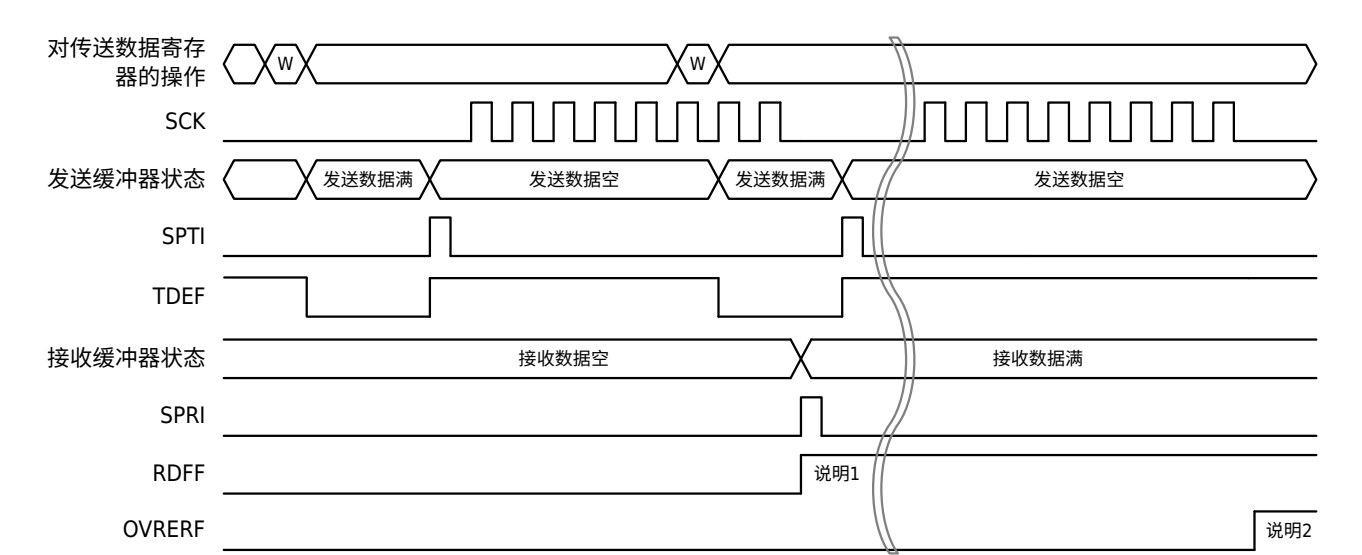


图 29-11 全双工同步串行通信



说明

1. 当本次串行传输结束时，如果接收缓冲器为空，SPI 将会把接收到的数据从移位寄存器复制到接收缓冲器中，接收缓冲器满的标志位 RDFF 被置成 1，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，如果接收缓冲器中还保持着上次收到的数据而没有被系统读取，SPI 会将数据过载标志位 OVRERF 置成 1，本次数据接收无效，接收移位寄存器中的数据将被丢弃。

只发送通信方式

当 SPI_CR.TXMDS 位为 1 时，SPI 运行在只发送通信方式。如下图所示，SPI_CFG1.FTHLV 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。

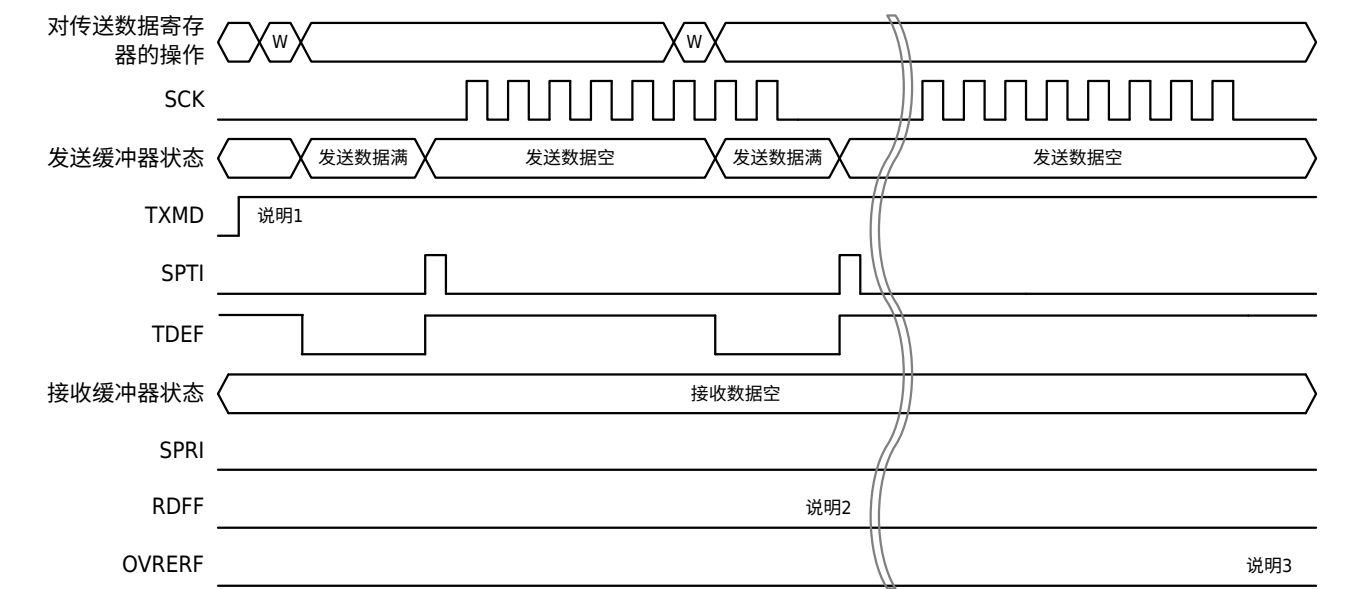


图 29-12 只进行发送通信



说明

1. 在设置进入只发送通信方式之前，请确保接收缓冲器中没有未读取的数据 (RDFF=0) 且未出现数据过载错误 (OVRERF=0)。
2. 在只发送通信方式下，当本次串行传输结束时，即使接收缓冲器为空，也不会有数据被接收，RDFF 将始终保持为 0。
3. 在只发送通信方式下，由于接收缓冲器始终为空，故不会发生数据过载错误，OVRERF 标志位始终保持为 0。

29.3.3.5 通信间隔时间

常规通信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 和 SPI_CFG2.MIDIE 来进行设定，最小间隔时间为 $1 \times SCK + 2 \times PCLK1$ 。如下图所示，SPI_CFG2.CPHA 位为 1 并且 SPI_CFG2.CPOL 位为 0，SPI 进行串行传送。

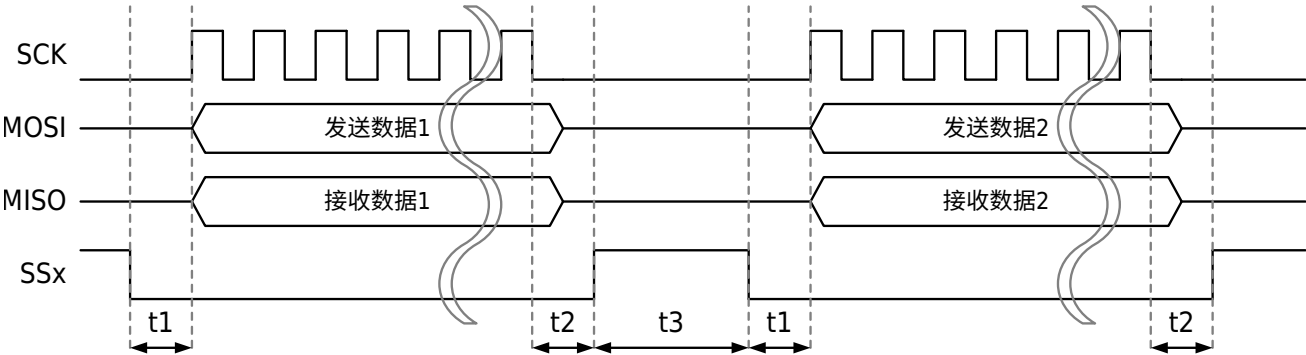


图 29-13 常规通信

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 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 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 和 SPI_CFG2.MIDIE 来控制）。

 **说明**

当通信速率为 PCLK1 的 2 分频（SPI_CFG2.MBR=0b00 且 SPI_CFG1.CLKDIV=0b0000）时，请将 t3 配置为 $2 \times SCK + 2 \times PCLK1$ 及以上，即 SPI_CFG2.MIDIE=1 且 SPI_CFG1.MIDI=0x1~0x7。

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。如下图所示，SPI_CFG2.CPHA 位为 1 并且 SPI_CFG2.CPOL 位为 0，SPI 进行串行传送。

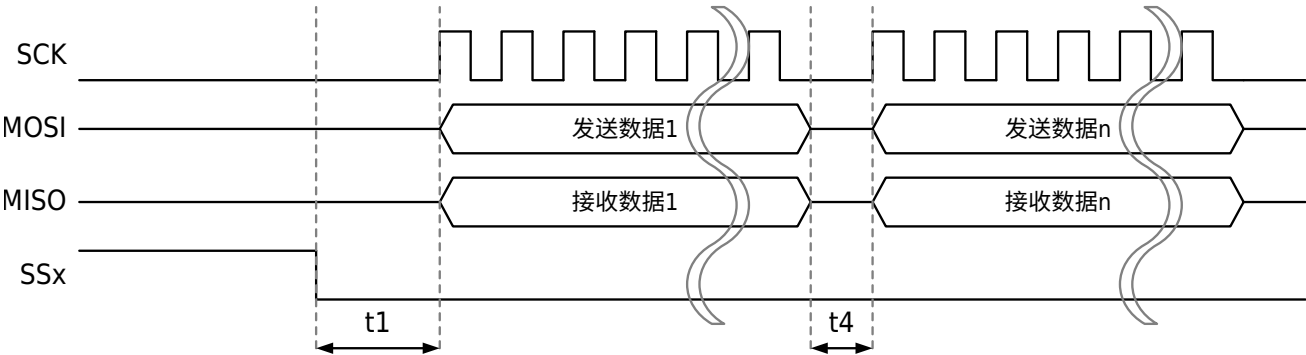


图 29-14 连续通信

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 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 和 SPI_CFG2.MSSIE 进行控制。t1 和 t4 之间的关系定义为 $t4 = t1 - 1 \times SCK$ 。



说明

- 连续通信模式下，主机如果没有新的数据写入发送缓冲器，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器后继续下一帧数据的传输。
- 连续通信模式下，从机如果没有新的数据写入发送缓冲器，则将发生欠载错误，SPI_SR.MODFERF 和 SPI_SR.UDRERF 将置 1，详细描述请参见“欠载错误”。
- 时钟同步模式下的连续通讯，如果使用 CPHA=0 的传送格式，在要保证数据在 CPHA 设置成 0 之前写入。
- 使用 CPHA=0 的传送格式时，从机需要在上一帧数据发送结束前向发送缓冲器写入新的数据，否则无论主机是否开始下一帧数据的通讯，从机都将发生欠载错误。
- 连续通信模式下，也可以通过 MSSI 来调整上一帧数据最后一个 SCK 和下一帧数据第一个 SCK 之间的时间间隔。

29.3.4 运行说明

29.3.4.1 运行模式概述

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_CR.MSTR 和 SPI_CR.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下表所示。

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 ^{*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 ^{*1}	有 ^{*1}	有 ^{*1}	有 ^{*1}
过载错误检测	有 ^{*1}	有 ^{*1}	有 ^{*1}	有 ^{*1}
奇偶校验错误检测	有 ^{*1 *2}	有 ^{*1 *2}	有 ^{*1 *2}	有 ^{*1 *2}
欠载错误检测	无	有	无	有



说明

1. 当 SPI_CR.TXMSDS 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, 具体设定步骤请参见“[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

作为主机时的动作说明

在 SPI 数据发送缓冲器 (TX_BUFF) 为空的状态下 (SPI_SR.TDEF 为 1), 将 SPI_CFG1.FTHLV 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 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见“[传送格式](#)”。

SPI 主机模式的初始化

- 步骤 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;
- 步骤 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;
- 步骤 3. 如需要使用中断, 请设置系统的中断寄存器;
- 步骤 4. 如需要使用 DMA, 请设置 DMA 的相关寄存器;
- 步骤 5. 设定输入输出管脚;
- 步骤 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;
- 步骤 7. 确认 SPI_CR 寄存器的设置;

- 步骤 8. 清除各种标志位；
- 步骤 9. 设置中断许可位；
- 步骤 10. 将 SPI_CR.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

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 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见“[传送格式](#)”。



说明

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 位设置成 0，则不能固定 SS0 输入信号。

SPI 从机模式的初始化

- 步骤 1. 设定通信配置寄存器 1 (SPI_CFG1)，主要包括使用帧数的设定；
- 步骤 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；
- 步骤 3. 如需要使用中断，请设置系统的中断寄存器；
- 步骤 4. 如需要使用 DMA，请设置 DMA 的相关寄存器；
- 步骤 5. 设定输入输出管脚；
- 步骤 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；
- 步骤 7. 确认 SPI_CR 寄存器的设置；
- 步骤 8. 清除各种标志位；
- 步骤 9. 设置中断许可位；
- 步骤 10. 将 SPI_CR.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_CR.SPIMDS 位为 1 时，SPI 处于时钟同步运行模式。在该模式动作时，SPI 只使用 SCK、MOSI 和 MISO 这 3 个管脚进行通信，SSi 管脚被释放可用于普通 I/O 功能。

尽管时钟同步运行模式时不使用 SSi 管脚，但模块内部的运行和 SPI 运行模式是相同的。

SPI 作为主机时的动作说明

在 SPI 数据发送缓冲器 (TX_BUFF) 为空的状态下 (SPI_SR.TDEF 标志位为 1)，将 SPI_CFG1.FTHLV 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 位的设定值。有关 SPI 传送格式的详细内容请参见“[传送格式](#)”。

时钟同步运行模式时主机初始化

- 步骤 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；
- 步骤 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；
- 步骤 3. 如需要使用中断，请设置系统的中断寄存器；
- 步骤 4. 如需要使用 DMA，请设置 DMA 的相关寄存器；
- 步骤 5. 设定输入输出管脚；
- 步骤 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；
- 步骤 7. 确认 SPI_CR 寄存器的设置；
- 步骤 8. 清除各种标志位；
- 步骤 9. 设置中断许可位；
- 步骤 10. 将 SPI_CR.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

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：

- 步骤 1. 进行通信设置，并将 SPI_CR.SPE 位设置为 1；
- 步骤 2. 将 SPI_CFG2.CPHA 设置为 1；
- 步骤 3. 将 SPI_CFG1.CTMDS 设置为 1；
- 步骤 4. 将需要发送的数据写入 SPI_DR；
- 步骤 5. 将 SPI_CFG2.CPHA 设置为 0；
- 步骤 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, 串行传输结束后 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 在串行传输结束后将 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输数据的位长, 从属模式的 SPI 的数据长度取决于 SPI_CFG2.DSIZE 位的设定值。

时钟同步运行模式时从机初始化

- 步骤 1. 设定通信配置寄存器 1 (SPI_CFG1), 主要包括使用帧数的设定;
- 步骤 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;
- 步骤 3. 如需要使用中断, 请设置系统的中断寄存器;
- 步骤 4. 如需要使用 DMA, 请设置 DMA 的相关寄存器;
- 步骤 5. 设定输入输出管脚;
- 步骤 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;
- 步骤 7. 确认 SPI_CR 寄存器的设置;
- 步骤 8. 清除各种标志位;
- 步骤 9. 设置中断许可位;
- 步骤 10. 将 SPI_CR.SPE 位设置成 1, 动作开始。

29.3.4.6 几种 SPI 动作的处理流程

SPI 作为主机时的数据传送处理流程

- 步骤 1. 等待数据发送缓冲器空的中断或通过轮询方式确认数据发送缓冲器处于空状态;
- 步骤 2. 向数据寄存器 SPI_DR 写入要发送的数据;
- 步骤 3. 重复步骤 1、2, 直到最后一个数据发送完成;
- 步骤 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;
- 步骤 5. 硬件产生 SPI 闲置状态中断;
- 步骤 6. 将 SPE 置 0, 停止 SPI 动作, 同时将 IDIE 清零。

数据接收处理流程

- 步骤 1. 等待数据接收缓冲器满的中断或者通过轮询方式确认数据接收缓冲器处于满状态;
- 步骤 2. 通过访问 SPI_DR 从接收缓冲器读取数据;
- 步骤 3. 重复步骤 1、2, 直到最后一个接收数据被读取;
- 步骤 4. 将数据接收缓冲器满的中断允许位 RXIE 清零。

通信错误处理流程

- 步骤 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/OVRERF/UDRERF/PERF) 被置成 1;
- 步骤 2. 确认 SS0 状态, 排除模式故障错误;
- 步骤 3. 将 SPE 清零, 停止 SPI 动作;
- 步骤 4. 通过错误标志位确定通信错误种类, 进行通讯错误处理;
- 步骤 5. 将错误标志位清零;
- 步骤 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。

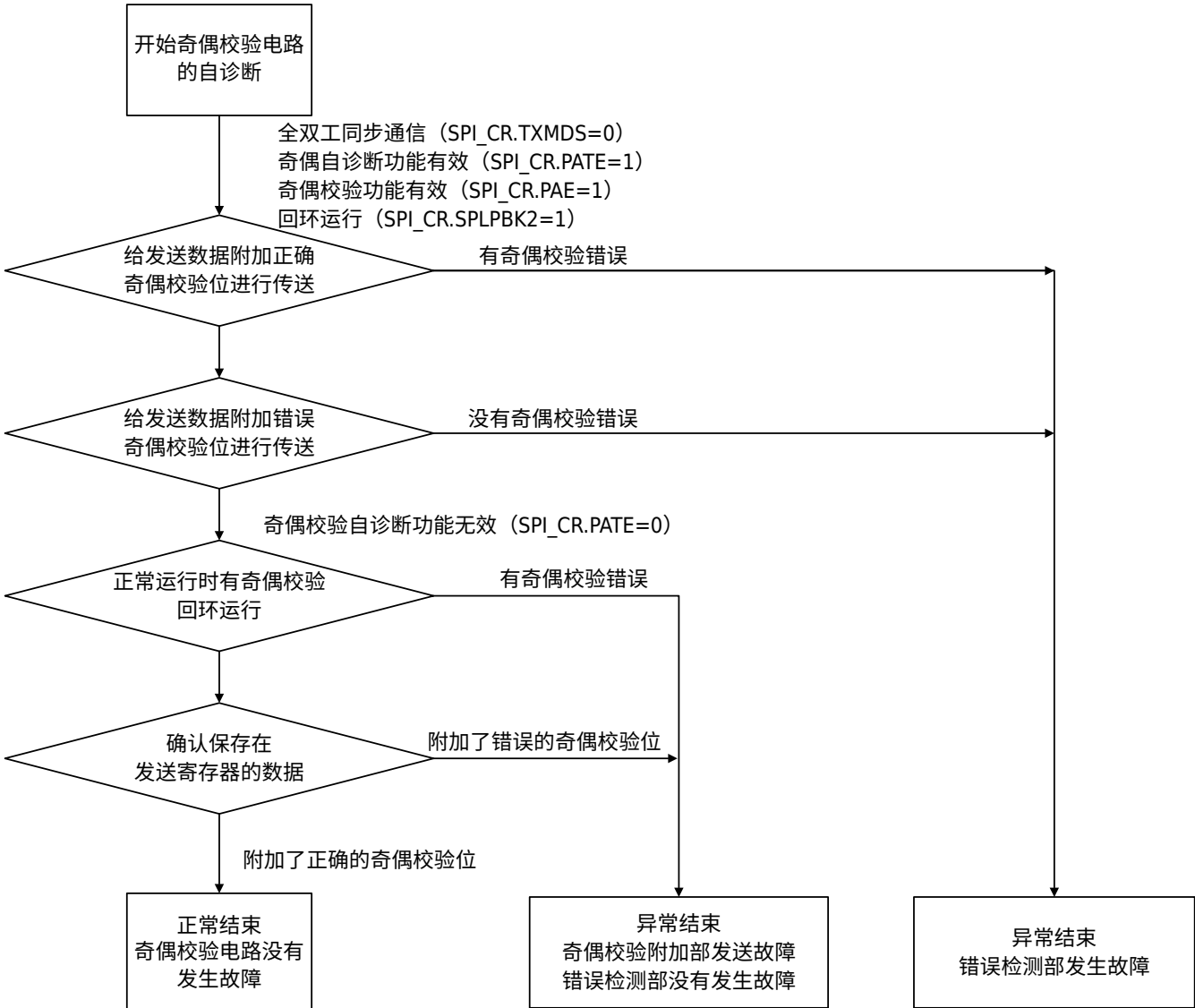


图 29-15 奇偶校验流程

29.3.6 错误检测

29.3.6.1 错误检测概述

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

序号	发生条件	SPI 运行	检测错误
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。

29.3.6.2 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。

29.3.6.3 模式错误

SPI 在主机模式时，SPI_CR.MODFE 请勿设定为 1。在从机模式时，当 SSi 有效电平宽度没有达到传输数据所需时间时，发生模式故障，SPI_SR.MODFERF 置 1，SPI_CR.SPE 置为 0。需要传送时，对 SPI_SR.MODFERF 清零后，再把 SPI_CR.SPE 置 1。

29.3.6.4 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。

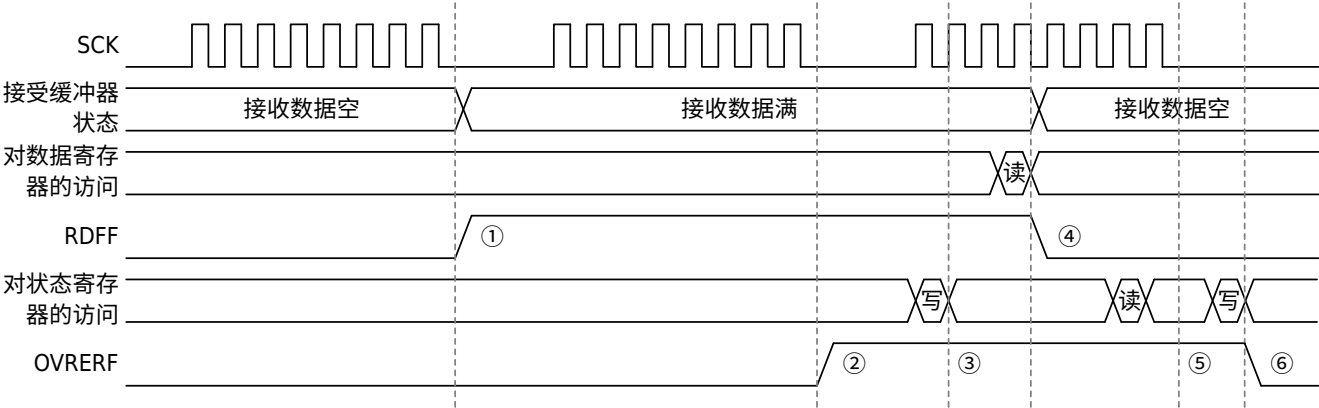


图 29-16 过载错误处理

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器进行读取，读取后接收缓冲器状态变为空，SPI 再重启通信时钟完成最后一位的数据接收。

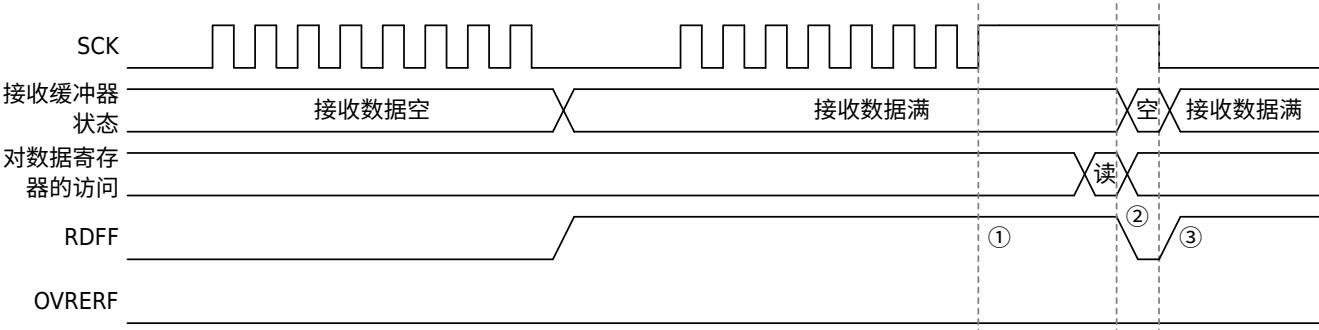


图 29-17 启用时钟自动停止功能时的动作示意图 (CPHA=1)

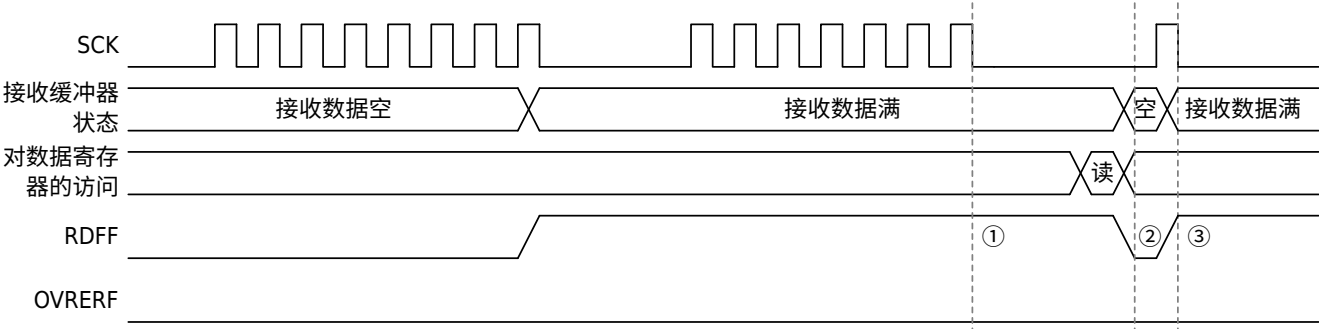


图 29-18 启用时钟自动停止功能时的动作示意图 (CPHA=0)

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 读取接收缓冲器中的数据后，接收缓冲器变为空状态，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。

29.3.6.5 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。

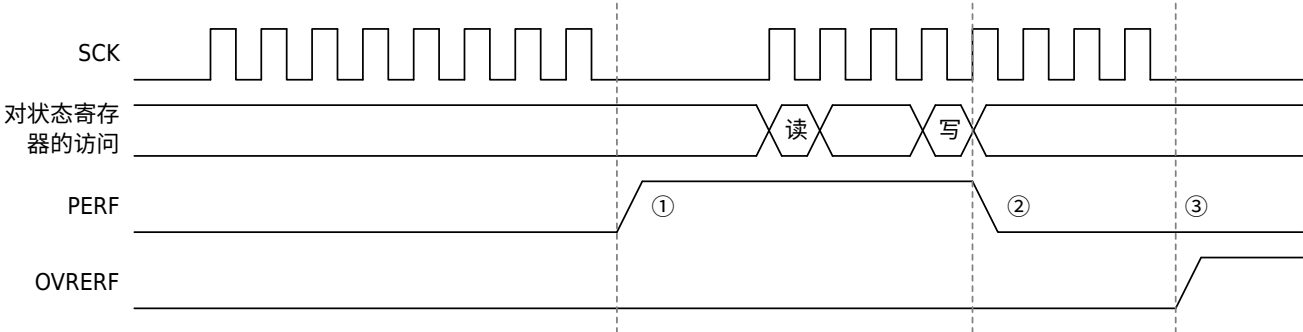


图 29-19 奇偶校验错误

以下说明在图中①~③所示的时序中标志的运行内容：

- ①SPI 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。
- ③SPI 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。如果发生系统复位或使用功能复位寄存器 RMU_FRST1，则对全部 SPI 功能进行初始化。

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。

复位初始化

通过系统复位或使用功能复位寄存器 RMU_FRST1 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。系统复位和功能复位寄存器的内容请参见《复位控制（RMU）》章节相关描述。

29.3.8 中断和事件

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下表所示。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 中断源和触发源说明

中断/内部触发源	略称	中断源	内部触发源	条件
接收缓冲器满	SPI_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	SPI_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）	SPI_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	SPI_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	SPI_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

用户可将上表中可用作触发源的向量写入不同的触发对象寄存器，以实现各种事件触发功能。上述事件触发源对应的向量请参见《中断控制器（INTC）》的“中断事件表”。

29.4 寄存器

29.4.1 寄存器总表

- SPI1 基地址：0x4001C000
- SPI2 基地址：0x4001C400
- SPI3 基地址：0x40020000

表 29-9 串行外设接口（SPI）寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	SPI_DR	SPI 数据寄存器	32
0x04	SPI_CR	SPI 控制寄存器	32
0x0C	SPI_CFG1	SPI 通信配置寄存器 1	32
0x14	SPI_SR	SPI 状态寄存器	32
0x18	SPI_CFG2	SPI 通信配置寄存器 2	32

29.4.2 SPI 数据寄存器（SPI_DR）

Offset	Bit Position																															
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	SPD																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	SPD	串行数据	SPI 数据存储	RW

29.4.3 SPI 控制寄存器 (SPI_CR)

Offset	Bit Position																																
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																PAE	PAOE	PATE	MODFE	IDIE	RXIE	TXIE	EIE	CSUSPE	SPE	SPLPBK2	SPLPBK	MSTR	Reserved	TXMDS	SPIMDS	
Access																	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW	RW

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15	PAE	奇偶校验允许	0b0: 发送数据不附加奇偶校验位, 接收数据不进行奇偶校验 0b1: 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
14	PAOE	奇偶校验模式选择	0b0: 选择偶校验进行发送和接收 0b1: 选择奇校验进行发送和接收	RW
13	PATE	奇偶校验自诊断	0b0: 奇偶校验自诊断功能无效 0b1: 奇偶校验自诊断功能有效	RW
12	MODFE	模式故障错误检测允许	0b0: 禁止模式故障错误检测 0b1: 允许模式故障错误检测	RW
11	IDIE	SPI 空闲中断允许	0b0: 禁止空闲中断请求产生 0b1: 允许空闲中断请求产生	RW
10	RXIE	SPI 接收中断允许	0b0: 禁止 SPI 接收中断请求产生 0b1: 允许 SPI 接收中断请求产生	RW
9	TXIE	SPI 发送中断允许	0b0: 禁止 SPI 发送中断请求产生 0b1: 允许 SPI 发送中断请求产生	RW
8	EIE	SPI 错误中断允许	0b0: 禁止 SPI 错误中断请求产生 0b1: 允许 SPI 错误中断请求产生	RW
7	CSUSPE	通讯自动挂起功能允许	0b0: 通讯自动挂起功能无效 0b1: 通讯自动挂起功能有效  说明 通讯自动挂起功能仅在位速率分频 (由 SPI_CFG1.CLKDIV 和 SPI_CFG2.MBR 组合配置) 不大于 6 时有效。	RW
6	SPE	SPI 功能允许	0b0: SPI 功能无效 0b1: SPI 功能有效	RW
5	SPLPBK2	SPI 回环 2 位	0b0: 正常模式 0b1: 回环模式 (发送数据=接收数据)	RW

位/位域	标记	位名	功能描述	读写
4	SPLPBK	SPI 回环位	● 0b0: 正常模式 ● 0b1: 回环模式 (发送数据的反相=接收数据)	RW
3	MSTR	SPI 主从模式选择	● 0b0: 从机模式 ● 0b1: 主机模式	RW
2	Reserved	保留	-	-
1	TXMDS	通信模式选择	● 0b0: 全双工同步串行通信 ● 0b1: 只进行发送串行通信	RW
0	SPIMDS	SPI 模式选择	● 0b0: SPI 运行 (4 线式) ● 0b1: 时钟同步运行 (3 线式)	RW

29.4.4 SPI 通信配置寄存器 1 (SPI_CFG1)

Offset	Bit Position																																			
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reset	0x00000010																																			
Name	Reserved	MIDI		Reserved	MSSDL		Reserved	MSSI		Reserved					CLKDIV					SS3PV		SS2PV		SS1PV		SS0PV		Reserved	SPRDTD		Reserved		CTMDS		FTHLV	
Access		RW			RW			RW							RW					RW		RW		RW		RW			RW				RW		RW	

位/位域	标记	位名	功能描述	读写
31	Reserved	保留	-	-
30:28	MIDI	主机下次存取数据间隔闲置时间设定位	0b000: 1 个 SCK+2 个 PCLK1 0b001: 2 个 SCK+2 个 PCLK1 0b010: 3 个 SCK+2 个 PCLK1 0b011: 4 个 SCK+2 个 PCLK1 0b100: 5 个 SCK+2 个 PCLK1 0b101: 6 个 SCK+2 个 PCLK1 0b110: 7 个 SCK+2 个 PCLK1 0b111: 8 个 SCK+2 个 PCLK1	RW
27	Reserved	保留	-	-
26:24	MSSDL	主机 SS 无效延迟设定位	0b000: 1 个 SCK 0b001: 2 个 SCK 0b110: 7 个 SCK 0b111: 8 个 SCK	RW
23	Reserved	保留	-	-
22:20	MSSI	主机 SS 闲置时间设定位	0b000: 1 个 SCK 0b001: 2 个 SCK 0b110: 7 个 SCK 0b111: 8 个 SCK	RW
19:16	Reserved	保留	-	-
15:12	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 1001: PCLK1/20 1010: PCLK1/22 1011: PCLK1/24 1100: PCLK1/26	RW

位/位域	标记	位名	功能描述	读写
15:12	CLKDIV	通讯基准时钟选择位	● 1101: PCLK1/28 ● 1110: PCLK1/30 ● 1111: PCLK1/32	RW
11	SS3PV	SS3 信号极性设定	● 0b0: 低电平有效 ● 0b1: 高电平有效	RW
10	SS2PV	SS2 信号极性设定	● 0b0: 低电平有效 ● 0b1: 高电平有效	RW
9	SS1PV	SS1 信号极性设定	● 0b0: 低电平有效 ● 0b1: 高电平有效	RW
8	SS0PV	SS0 信号极性设定	● 0b0: 低电平有效 ● 0b1: 高电平有效	RW
7	<i>Reserved</i>	<i>保留</i>	-	-
6	SPRDTD	数据寄存器读取对象选择	● 0b0: SPI_DR 读接收缓冲器 ● 0b1: SPI_DR 读发送缓冲器 (必须在 TDEF=1 时才可以读取)	RW
5	<i>Reserved</i>	<i>保留</i>	-	-
4	<i>Reserved</i>	<i>保留</i>	读出为 1, 写入时必须写 1	-
3	<i>Reserved</i>	<i>保留</i>	-	-
2	CTMDS	连续通信模式选择	● 0b0: 常规通信模式 ● 0b1: 连续通信模式	RW
1:0	FTHLV	帧数设定位	● 0b00: 1 帧 ● 0b01: 2 帧 ● 0b10: 3 帧 ● 0b11: 4 帧	RW

29.4.5 SPI 状态寄存器 (SPI_SR)

Offset	Bit Position																																						
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
Reset	0x00000020																																						
Name	Reserved																								RO	RDF	Reserved	RO	TDEF	RW	UDRERF	RW	PERF	RW	MODFERF	RO	IDLNF	RW	OVRERF
Access																									RO			RO		RW		RW		RW		RO		RW	

位/位域	标记	位名	功能描述	读写
31:8	Reserved	保留	-	-
7	RDF	接收缓冲器满标志	硬件置位和清零 ● 0b0: 接收缓冲器数据未满 ● 0b1: 接收缓冲器数据已满	RO
6	Reserved	保留	-	-
5	TDEF	发送缓冲器空标志	硬件置位和清零 ● 0b0: 发送缓冲器有数据 ● 0b1: 发送缓冲器无数据	RO
4	UDRERF	欠载错误标志	硬件置位; 读取该位为 1 后写 0 将状态位清零 ● 0b0: 未发生欠载错误 ● 0b1: 欠载错误发生 (MODFERF=1)  说明 当 MODFERF=0, 此位会被初始化。	RW
3	PERF	奇偶校验错误标志	硬件置位; 读取该位为 1 后写 0 将状态位清零 ● 0b0: 未发生奇偶校验错误 ● 0b1: 发生奇偶校验错误	RW
2	MODFERF	模式故障错误标志	硬件置位; 读取该位为 1 后写 0 将状态位清零 ● 0b0: 未发生模式故障错误 ● 0b1: 发生模式故障错误	RW
1	IDLNF	SPI 空闲标志	硬件置位和清零 ● 0b0: SPI 为空闲状态 ● 0b1: SPI 为传送状态	RO
0	OVRERF	过载错误标志	硬件置位; 读取该位为 1 后写 0 将状态位清零 ● 0b0: 未发生过载错误 ● 0b1: 发生过载错误	RW

29.4.6 SPI 通信配置寄存器 2 (SPI_CFG2)

Offset	Bit Position																																	
0x18	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000F1D																																	
Name	Reserved																MSSIE	MSSDLE	MIDIE	LSBF	DSIZE				SSA				Reserved	MBR		CPOL	CPHA	
Access																	RW	RW	RW	RW	RW				RW					RW		RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15	MSSIE	SCK 延迟允许	0b0: SCK 延迟为 1 个 SCK 0b1: SCK 延迟为 MSSIE 的设定值	RW
14	MSSDLE	SS 无效延迟允许	0b0: SS 无效延迟为 1 个 SCK 0b1: SS 无效延迟为 MSSDL 的设定值	RW
13	MIDIE	SPI 下次存取延迟允许	0b0: 下次存取延迟为 1 个 SCK+2 个 PCLK1 0b1: 下次存取延迟为 MIDI 的设定值	RW
12	LSBF	SPI LSB first 位	0b0: MSB first 0b1: LSB first	RW
11:8	DSIZE	SPI 数据长度设定位	0b0000: 4 位 0b0001: 5 位 0b0010: 6 位 0b0011: 7 位 0b0100: 8 位 0b0101: 9 位 0b0110: 10 位 0b0111: 11 位 0b1000: 12 位 0b1001: 13 位 0b1010: 14 位 0b1011: 15 位 0b1100: 16 位 0b1101: 20 位 0b1110: 24 位 0b1111: 32 位	RW
7:5	SSA	SS 信号有效设定位	0b000: SS0 0b001: SS1 0b010: SS2 0b011: SS3 - 其他: 禁止设置	RW
4	Reserved	保留	读出为 1, 写入时必须写 1	-

位/位域	标记	位名	功能描述	读写
3:2	MBR	位速率分频设定位	● 0b00: 选择基准通讯时钟 ● 0b01: 选择基准通讯时钟的 2 分频 ● 0b10: 选择基准通讯时钟的 4 分频 ● 0b11: 选择基准通讯时钟的 8 分频	RW
1	CPOL	SCK 极性设定位	● 0b0: 空闲时的 SCK 为 Low 电平 ● 0b1: 空闲时的 SCK 为 High 电平	RW
0	CPHA	SCK 相位设定位	● 0b0: 在奇数边沿进行数据采样, 在偶数边沿数据发生变化 ● 0b1: 在奇数边沿数据发生变化, 在偶数边沿进行数据采样	RW

30 四线式串行外设接口（QSPI）

30.1 简介

四线式串行外设接口（QSPI）是一个存储器控制模块，主要用于和带 SPI 兼容接口的串行 ROM 进行通信。其对象主要包括有串行闪存，串行 EEPROM 以及串行 FeRAM。

30.2 主要特性

表 30-1 QSPI 主要规格

参数	规格
通道数	1 通道
SPI	支持扩展 SPI，二线式 SPI 和四线式 SPI 等多种协议
	支持 SPI 模式 0 和 SPI 模式 3
	地址线宽度可选择 8 位/16 位/24 位/32 位
时序调整	可通过时序调整以支持各种串行闪存
闪存读取	支持多种读取方式： 1. 标准读/快速读 2. 二线式输出快速读取/二线式输入输出快速读取 3. 四线式输出快速读取/四线式输入输出快速读取
	自由设置指令
	数量可调的虚拟周期
	16 字节的预读取功能
	状态查询功能
	SPI 总线周期延长功能
	XIP 控制功能
直接通信功能	灵活而广泛的支持大量串行闪存软件控制指令，包括擦、写、ID 读取及掉电控制等。
中断源	硬件错误中断
模块停止功能	可通过模块停止降低功耗

30.3 内存映射

30.3.1 内部总线空间

串行闪存及相关的控制寄存器在 AHB 总线空间的位置由总体的地址范围配置来决定。

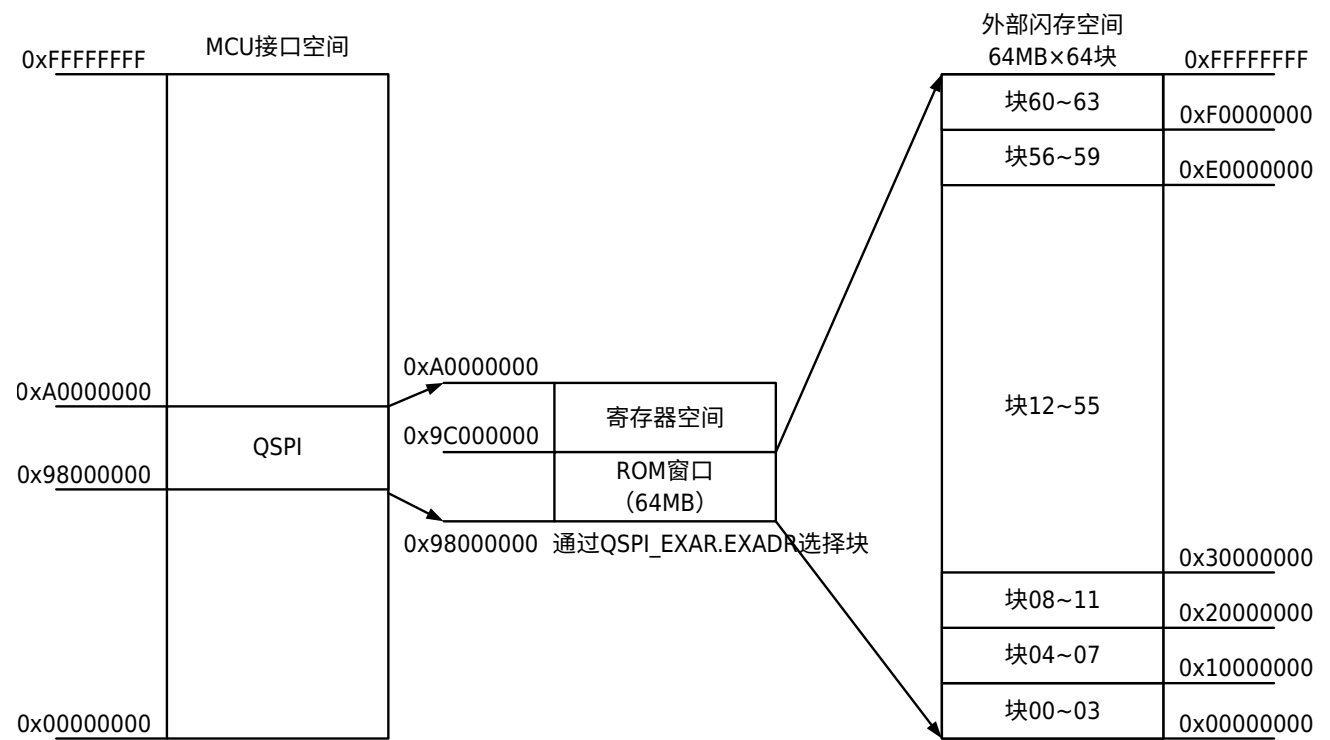


图 30-1 默认区域设定及 AHB 总线空间内存映射关系图

30.3.2 ROM 空间及总线的地址宽度

QSPI 的拥有 32 位地址总线宽度来配合串行闪存。每当对 QSPI 的 ROM 空间进行读访问时，QSPI 总线自动开始工作，将从串行闪存内读到的数据传送过来。

QSPI 不仅可以使用 32 位地址总线宽度，还可以通过设置 QSPI_FCR.AWSL 来选择使用 8 位/16 位/24 位地址总线宽度。

如果选择了 8 位/16 位/24 位的地址总线宽度，那么只有地址与之匹配的低位空间可以被正常访问，也就是说，访问 QSPI 中高位的串行闪存镜像空间将会反复出现低位空间的内容。

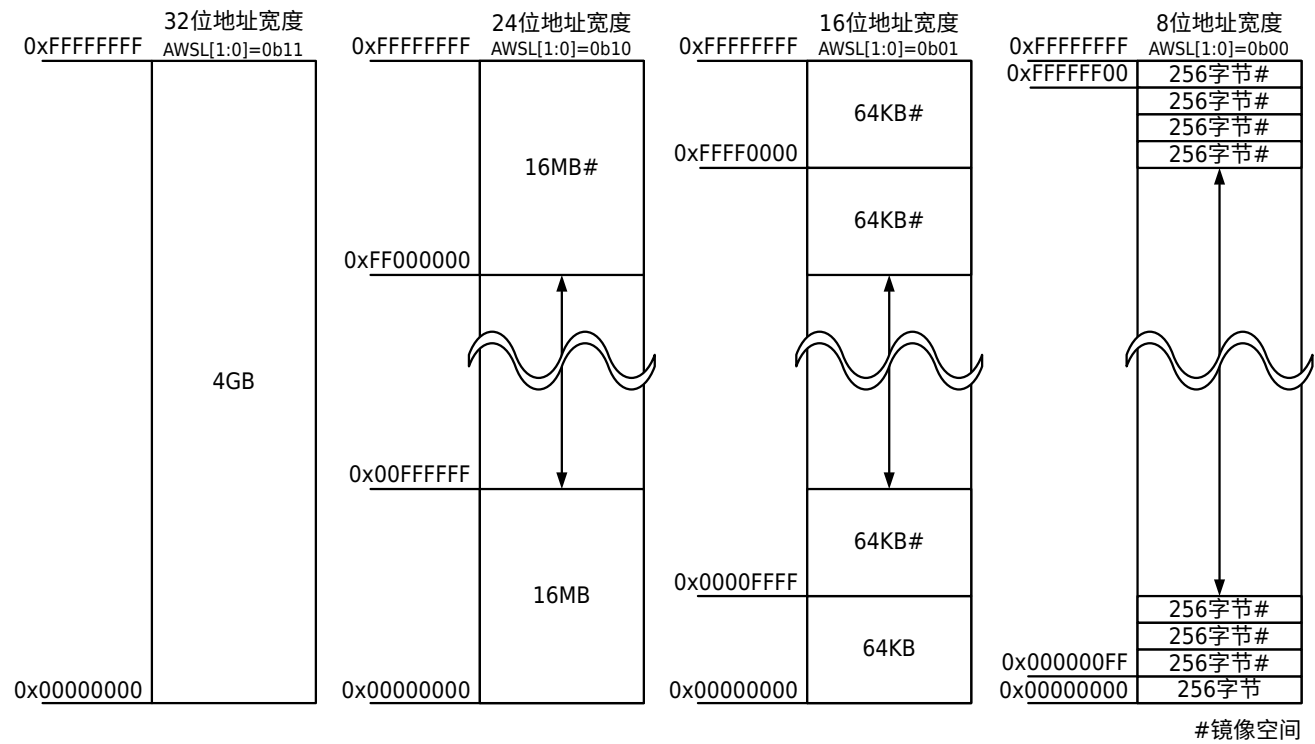


图 30-2 QSPI-ROM 空间内存映像图



说明

- 地址总线宽度可以通过设置 QSPI_FCR.AWSL 来选择使用 8 位/16 位/24 位/32 位。
- 选择 8 位地址宽度时，使用标准读指令仅支持访问低 256 字节空间，不支持访问镜像空间。

30.4 功能说明

30.4.1 基本框图

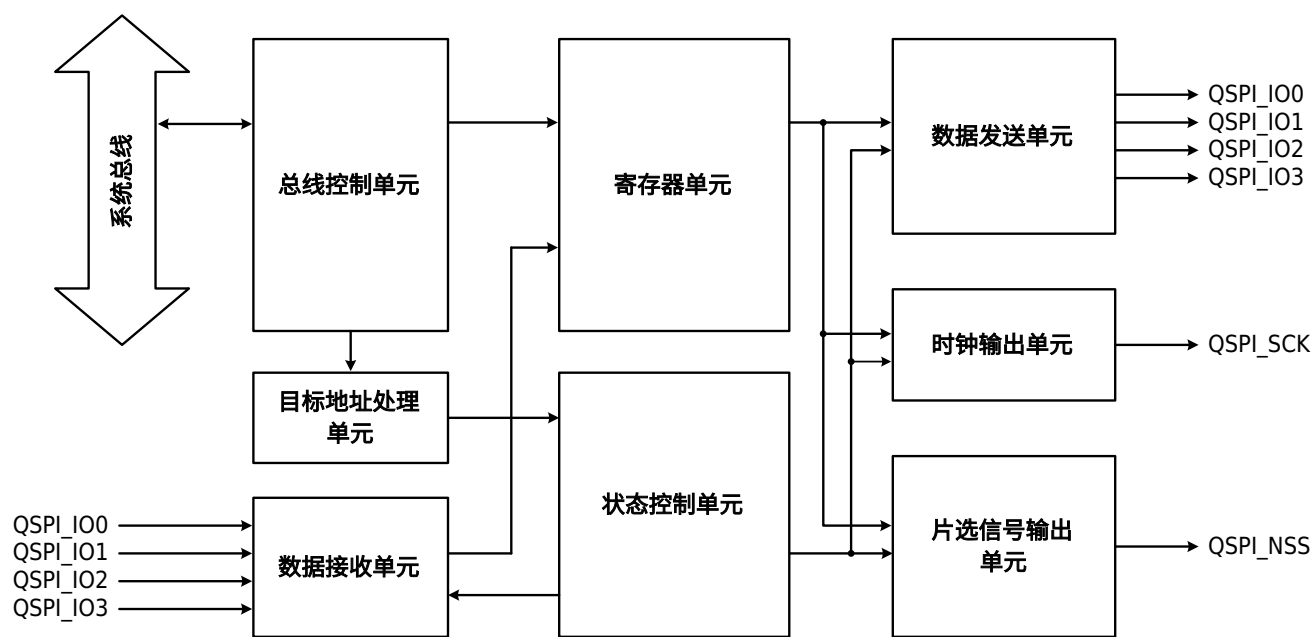


图 30-3 QSPI 的模块构成图

表 30-2 QSPI 管脚

管脚名	输入/输出	功能描述
QSPI_SCK	输出	QSPI 时钟输出管脚
QSPI_NSS	输出	QSPI 从机选择管脚
QSPI_IO0	输入/输出	数据线 0
QSPI_IO1	输入/输出	数据线 1
QSPI_IO2	输入/输出	数据线 2
QSPI_IO3	输入/输出	数据线 3



说明

QSPI_SCK、QSPI_NSS、QSPI_IO0、QSPI_IO1、QSPI_IO2、QSPI_IO3 以下分别简称 QSCK、QSSN、QSIO0、QSIO1、QSIO2、QSIO3。

30.4.2 QSPI 总线

30.4.2.1 SPI 协议

本 QSPI 支持扩展式 SPI、二线式 SPI 和四线式 SPI 三种协议。初始的默认协议是扩展式 SPI 协议。扩展式 SPI 协议只用 QSIO0 管脚单线进行指令输出，之后的地址及数据则根据具体的读取模式指令使用单线式/二线式/四线式输出。

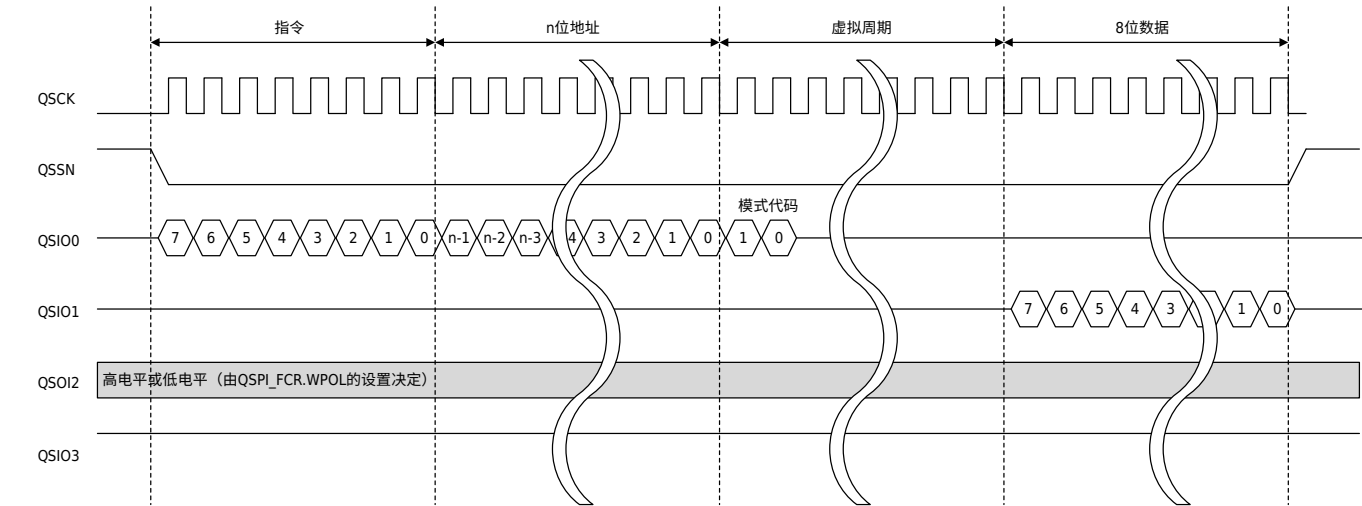


图 30-4 扩展式 SPI 协议动作示意图 1 (快速读模式)

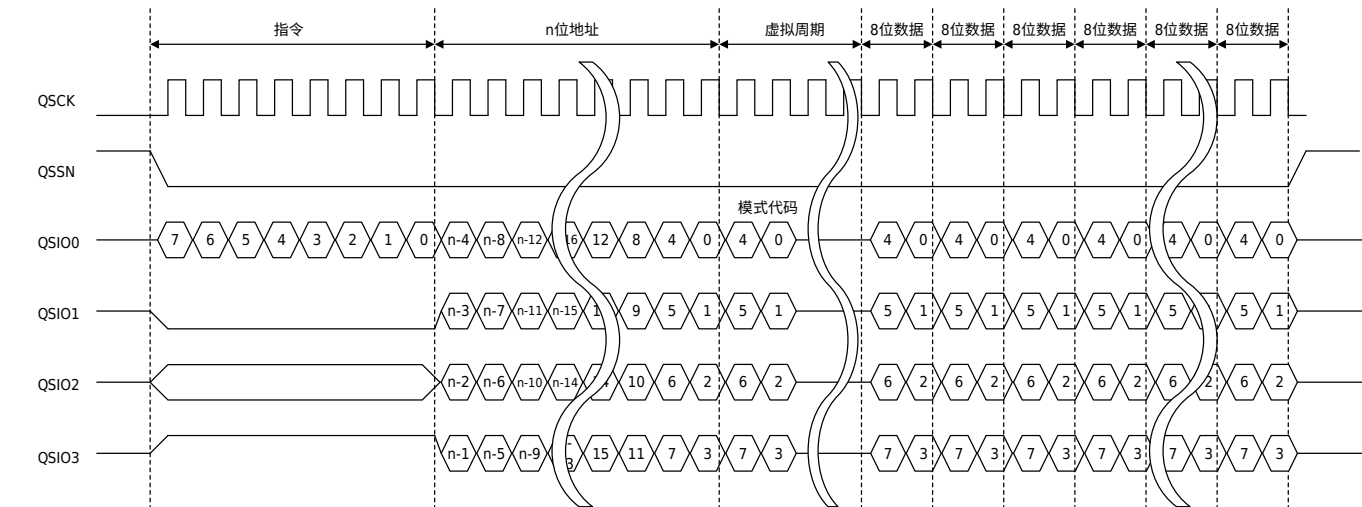


图 30-5 扩展式 SPI 协议动作示意图 2 (四线式输入输出快速读模式)

QSPI 读取模式选择为自定义标准读或自定义快速读时, 可通过设置 QSPI_CR.DPRSL/APRSL/IPRSL 位来分别配置各个阶段的协议。

二线式 SPI 协议使用 QSI00, QSI01 两个管脚实行相应的操作, 包括发出指令、地址、接收数据等等。

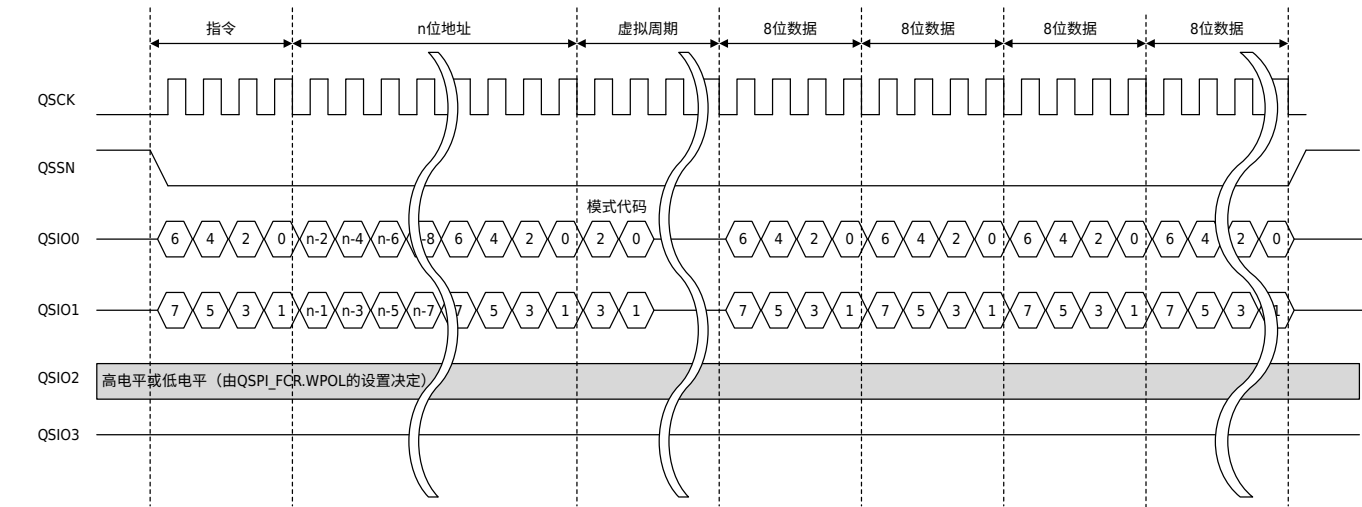


图 30-6 二线式 SPI 协议动作示意图 (快速读模式)

四线式 SPI 协议使用 QSI00, QSI01, QSI02, QSI03 四个管脚实行发出指令, 地址, 接收数据等所有相关操作。

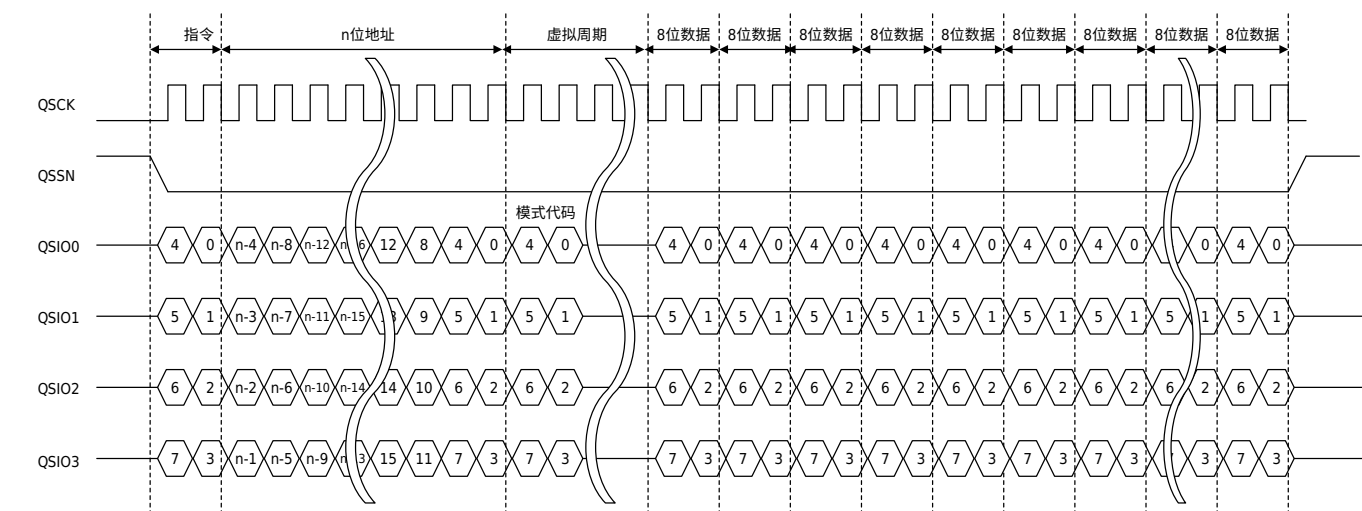


图 30-7 四线式 SPI 协议动作示意图 (快速读模式)

30.4.2.2 SPI 模式

SPI 模式有模式 0 和模式 3 两种, 可通过设置 QSPI_CR.SPIMD3 位来实现模式切换。SPI 模式 0 和模式 3 的区别在于空闲状态时 QSCCK 的电平不同。在 SPI 模式 0 时 QSCCK 的空闲状态电平为低电平, 而模式 3 时空闲状态电平则是高电平。

串行数据在串行时钟的下降沿从 QSPI 输出并在上升沿被读入外部闪存。而外部闪存则在串行时钟的下降沿输出串行数据并在下一个时钟的下降沿被 QSPI 读入。

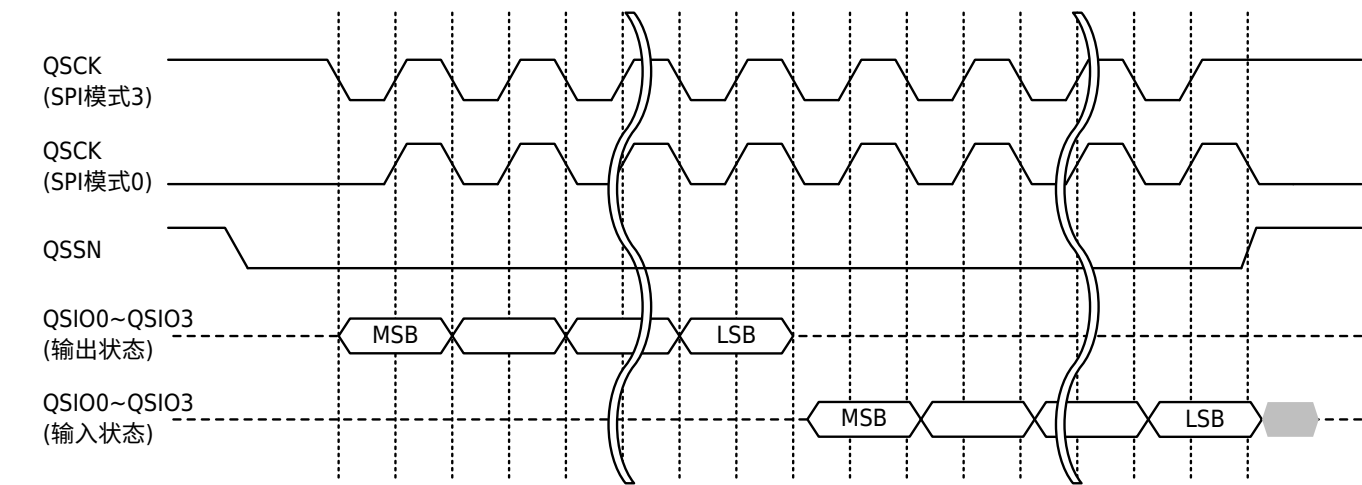


图 30-8 串行接口的基本时序图

30.4.3 QSPI 总线的时序调整

QSPI 总线信号可以通过寄存器进行时序微调, 微调后的时序可更好的应用于各种 QSPI 总线访问, 无论是 ROM 访问模式还是直接通信模式。

30.4.3.1 QSPI 总线基准时钟

QSPI 总线的基准时钟是由 HCLK 经过分频后得到的。通过设置 QSPI_CR.DIV 位可以选择 HCLK 的 2 分频到 64 分频等多种时钟源作为 QSPI 总线的基准时钟。

30.4.3.2 QSPI 通信时钟占空比补正

当基准时钟选择了 HCLK 的偶数倍分频时，QCK 信号的高低电平时间是一致的，如果选择了奇数倍分频，那么 QCK 信号的高电平时间将比低电平多一个 HCLK 周期。

如果希望选择奇数倍分频时的 QCK 也能输出 50%左右占空比的时钟信号，可以将 QSPI_FCR.DUTY 位设置成 1。通过这个设置，QCK 信号的上升沿的输出时间将比调整前晚半个 HCLK 周期，下降沿输出时间保持不变。由此可以得到占空比为 50%的 QCK 信号。当基准时钟选择 HCLK 的偶数倍分频时，DUTY 位请设置为 0。初始状态下 DUTY 位默认为 1。

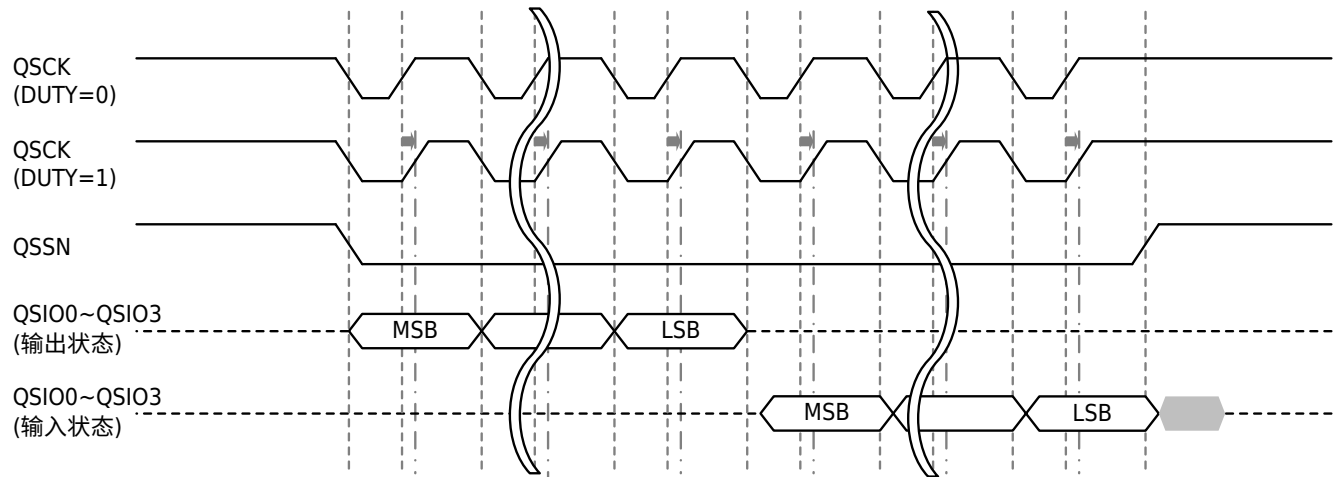


图 30-9 基准时钟选择 HCLK 三分频时输出时钟占空比补正示意图

30.4.3.3 QSSN 信号最小高电平宽度

为满足串行闪存所要求的取消时间要求，相邻的两次 QSPI 总线周期之间 QSSN 信号必须保持足够长时间的高电平状态（即闲置状态）。通过设置 QSPI_CSCR.SSHW[3:0]位可以选择 QSSN 的最小高电平宽度，选择范围为 1 个到 16 个 QSPI 基准时钟周期。

30.4.3.4 QSSN 的建立时间

从 QSSN 信号开始输出低电平（即变为有效状态）到 QCK 信号输出第一个边沿之间的时间称为 QSSN 的建立时间，该时间可以通过寄存器设置进行配置以满足外接串行闪存的要求。设置寄存器 QSPI_FCR.SSNLD 位来选择 QSSN 的建立时间是 0.5 个或是 1.5 个 QSPI 基准时钟周期。该设置同样可用于配置数据管脚从数据输出许可到 QCK 信号输出第一个边沿之间的建立时间，可根据需要合理运用。

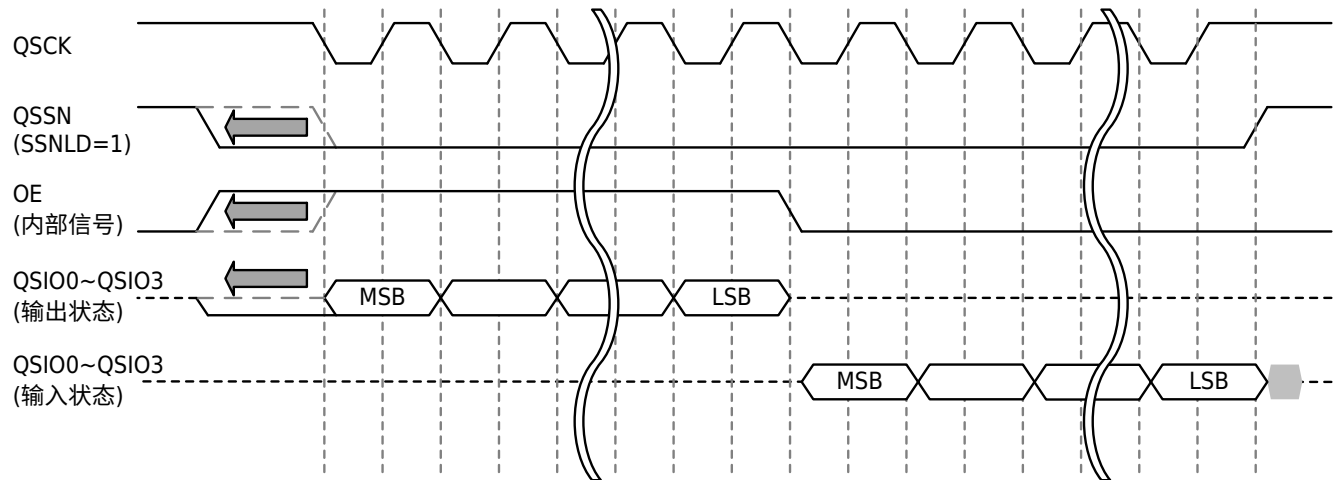


图 30-10 QSSN 建立时间配置示意图

30.4.3.5 QSSN 的保持时间

从 QCK 信号输出最后一个边沿到 QSSN 信号开始输出高电平（即变为闲置状态）之间的时间称为 QSSN 的保持时间，该时间可以通过寄存器设置进行配置以满足外接设备的要求。设置寄存器 QSPI_FCR.SSNHD 位来选择 QSSN 的保持时间是 0.5 个或是 1.5 个 QSPI 基准时钟周期。

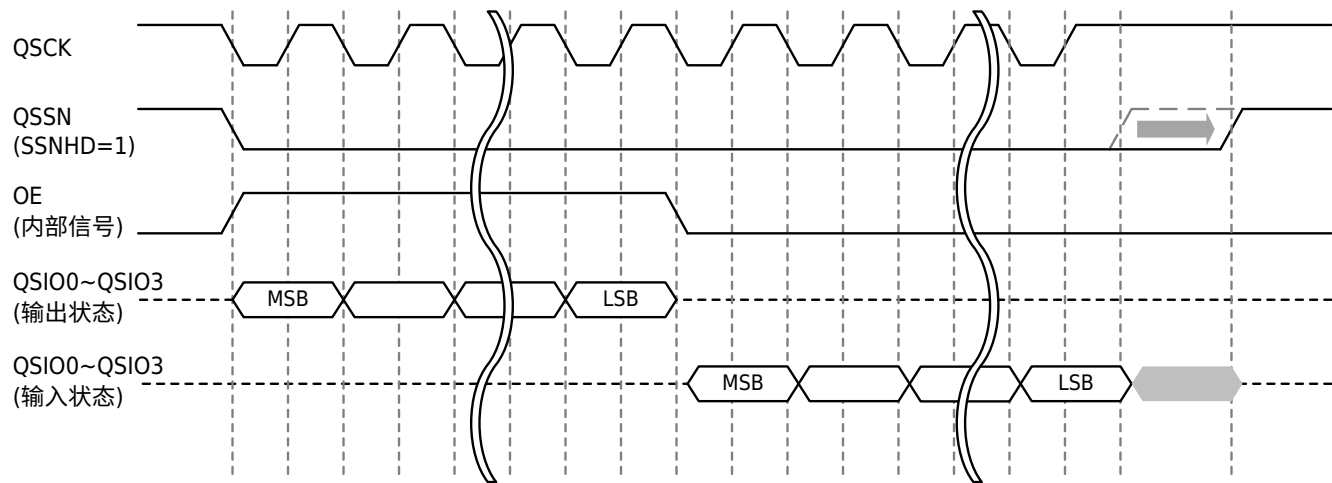


图 30-11 QSSN 保持时间配置示意图

30.4.3.6 串行数据接收延迟

串行闪存的数据和 QCK 的下降沿同步输出，QSPI 在下一个 QCK 的下降沿接收该数据。从串行闪存开始输出数据到该数据被 QSPI 接收之间的这段时间称之为接收延迟。QSPI 在第一个数据接收周期之前加入了一个延迟调整周期。站在串行闪存角度，该周期可视为对手方数据接收周期的增加。该延迟调整周期只会在数据接收动作时产生。

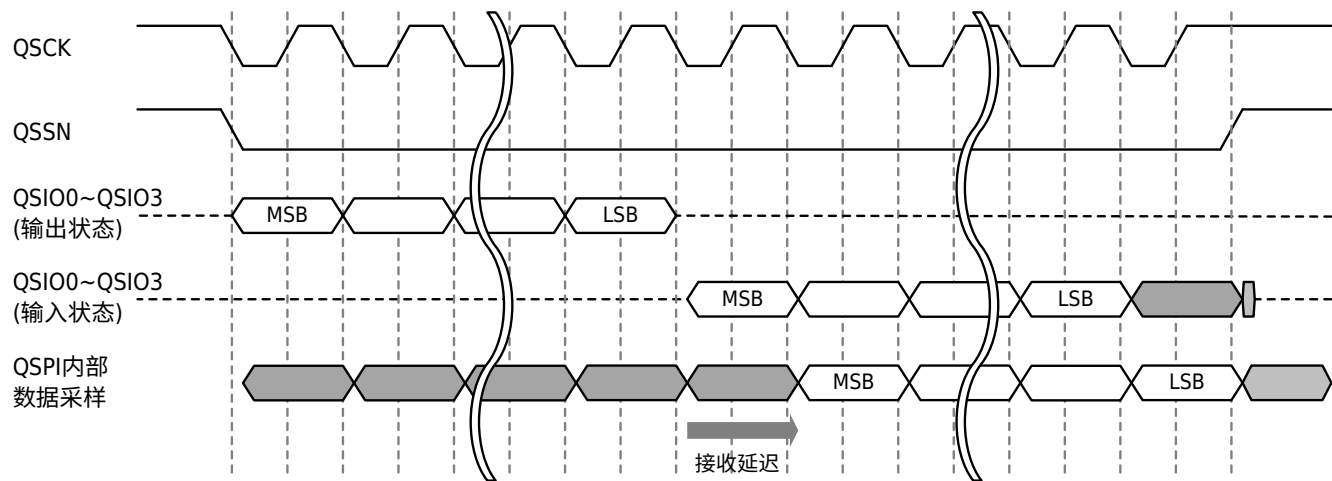


图 30-12 数据接收延迟示意图

30.4.4 用于 ROM 访问的 SPI 指令介绍

30.4.4.1 现有的 QSPI-ROM 指令参考

表 30-3 参考指令一览表

模式名		指令代码	说明
4-Bytes 地址模式	标准读	0x13	-
	快速读	0x0C	-

模式名		指令代码	说明
4-Bytes 地址模式	二线式输出快速读	0x3C	-
	二线式输入输出快速读	0xBC	-
	四线式输出快速读	0x6C	-
	四线式输入输出快速读	0xEC	-
	退出 4-Bytes 模式	0xE9	-
3-Bytes 地址模式	标准读	0x03	-
	快速读	0x0B	-
	二线式输出快速读	0x3B	-
	二线式输入输出快速读	0xBB	-
	四线式输出快速读	0x6B	-
	四线式输入输出快速读	0xEB	-
	进入 4-Bytes 模式	0xB7	-
-	写模式	0x06	-

串行闪存进行“自定义标准读”和“自定义快速读”访问时，指令需通过指令寄存器 QSPI_CCMD 来进行设置。

30.4.4.2 标准读指令

标准读指令是绝大多数串行闪存都支持的一种常用读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，随即 QSPI 就输出该指令的指令代码（0x03/ 0x13），紧接着输出目标地址，该地址的宽度可以通过 QSPI_FCR.AWSL 位来设置。然后就可以接收到数据。QSPI 的初始状态选择的指令就是标准读指令。

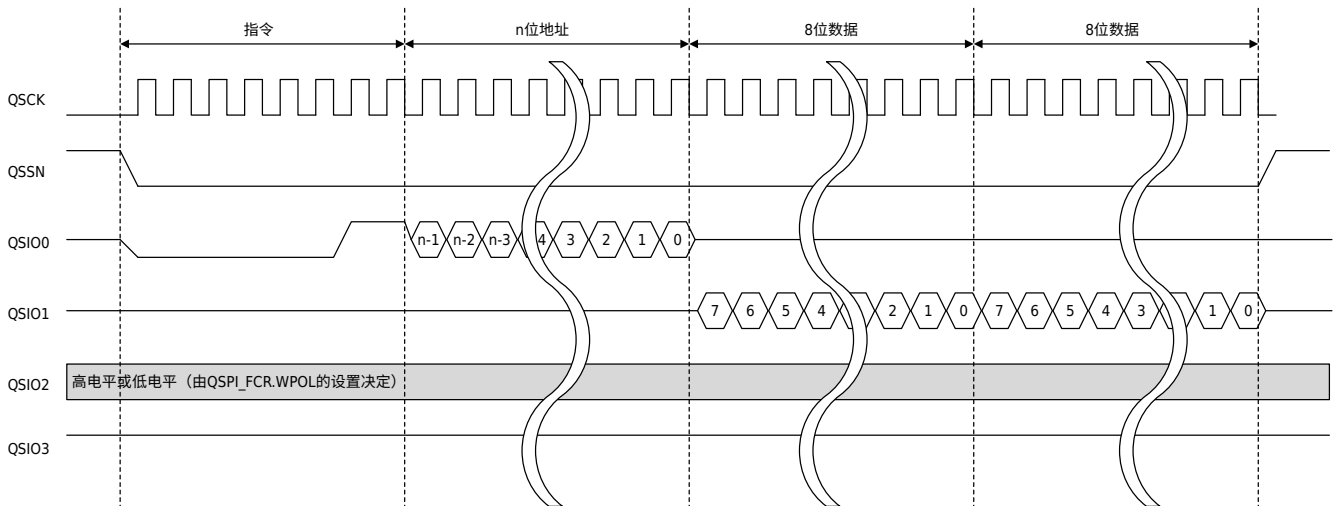


图 30-13 标准读总线周期示意图

30.4.4.3 快速读指令

快速读指令是一种支持更快的通信时钟的读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，随即 QSPI 就输出该指令的指令代码（0x0B/0x0C），紧接着输出目标地址，地址宽度可以通过

QSPI_FCR.AWSL 位来设置。地址输出后是定数量的虚拟周期，其具体数量由 QSPI_FCR.DMCYCN 决定。紧接着就是数据的接收。

虚拟周期最初的两个周期用于决定是否选择 XIP 模式。当选择 XIP 模式时，本次传输所使用的指令将被应用于下一个 SPI 总线周期，在下一个 SPI 总线周期时指令传输部分将会被省略。详细可参见“XIP 控制”。

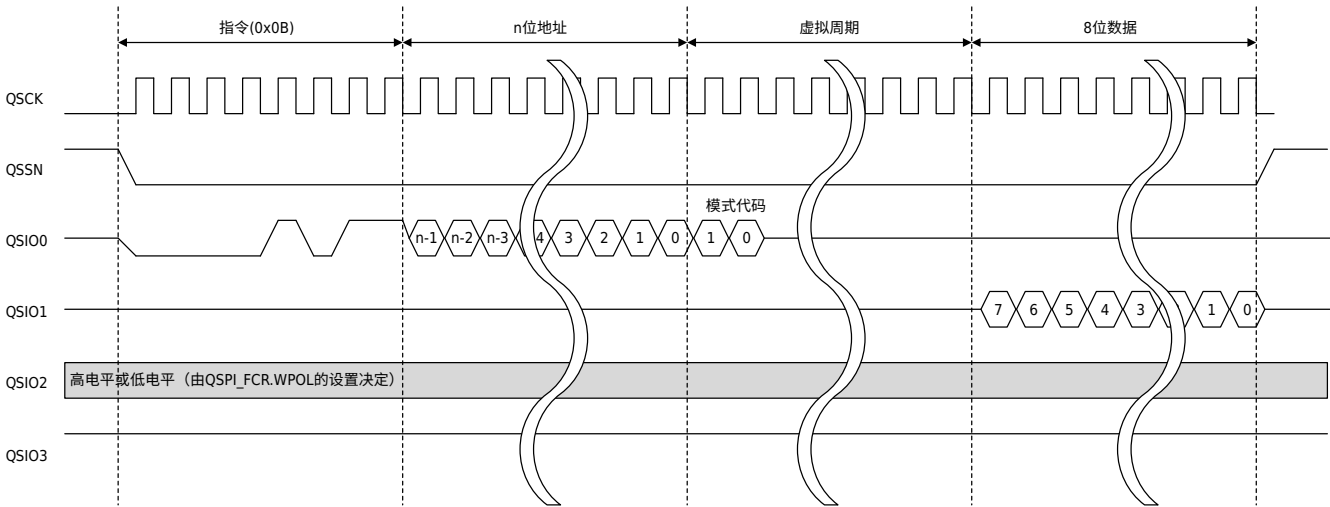


图 30-14 快速读总线周期示意图

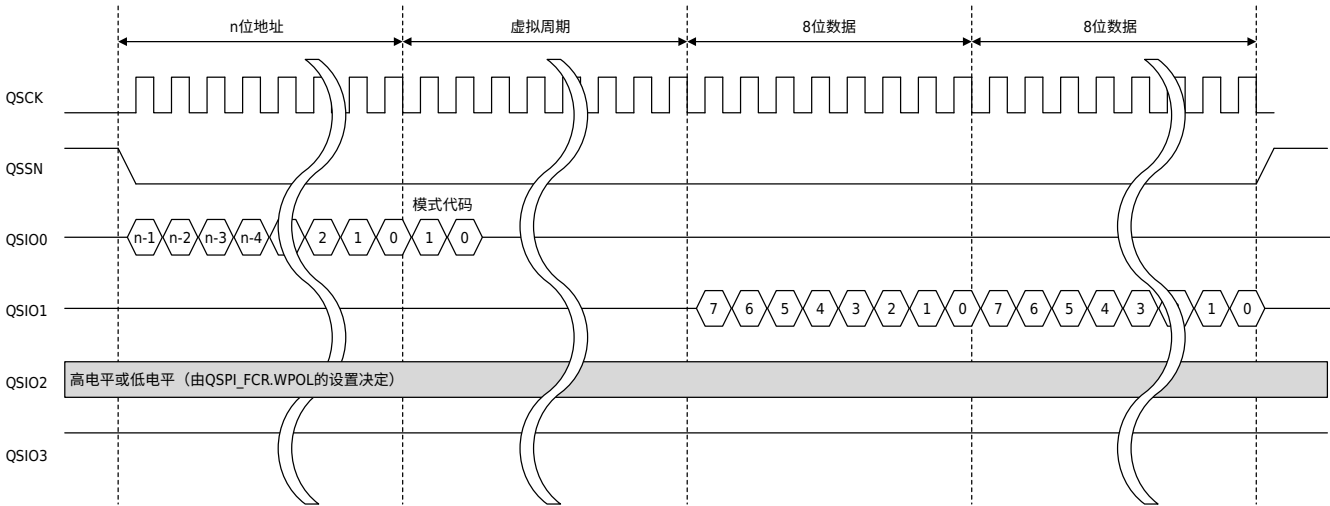


图 30-15 选择 XIP 模式的快速读总线周期示意图

 **说明**
要使用快速读指令请确保使用支持快速读功能的串行闪存。

30.4.4.4 二线式输出快速读指令

二线式输出快速读是一种使用两根信号线进行数据接收的读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSI00 管脚输出该指令的指令代码 (0x3B/0x3C) 和目标地址，地址宽度可以通过 QSPI_FCR.AWSL 位来设置。在这之后是一定数量的虚拟周期，其具体数量由 QSPI_FCR.DMCYCN 决定。然后开始通过 QSI00 和 QSI01 两根管脚进行数据接收。偶位数据在 QSI00 接收，奇位在 QSI01。

虚拟周期最初的两个周期用于决定是否选择 XIP 模式。当选择 XIP 模式时，本次传输所使用的指令将被应用于下一个 QSPI 总线周期，在下一个 QSPI 总线周期时指令传输部分将会被省略。详细可参见“XIP 控制”。

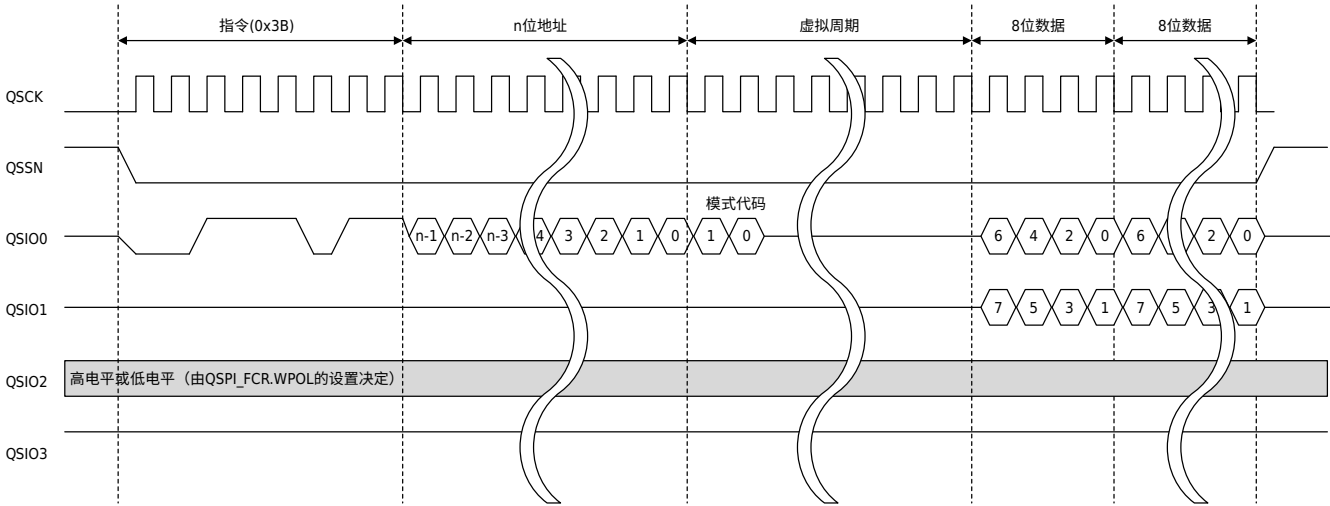


图 30-16 二线式输出快速读总线周期示意图

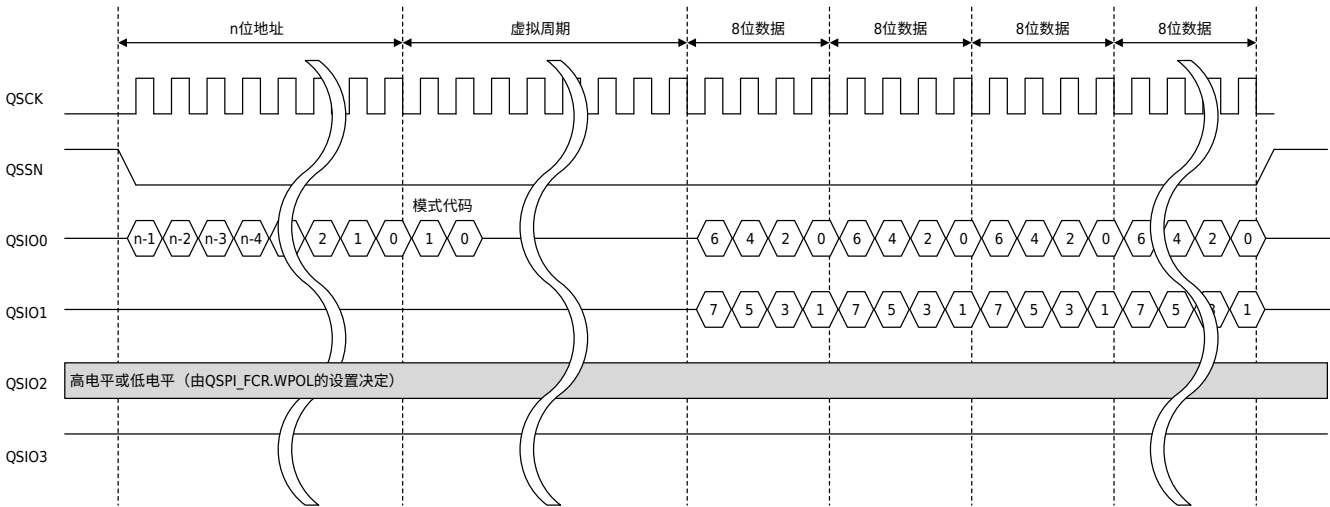


图 30-17 选择 XIP 模式的二线式输出快速读总线周期示意图

说明
要使用二线式输出快速读指令请确保使用支持该功能的串行闪存。

30.4.4.5 二线式输入输出快速读指令

二线式输入输出快速读是一种使用两根信号线进行地址发送和数据接收的读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSI00 管脚输出该指令的指令代码 (0xBB/0xBC)。在这之后 QSPI 从 QSI00 和 QSI01 两个管脚输出目标地址，地址宽度可以通过 QSPI_FCR.AWSL 位来设置。在这之后是一定数量的虚拟周期，其具体数量由 QSPI_FCR.DMCYCN 决定。然后开始通过 QSI00 和 QSI01 两根管脚进行数据接收。偶数位的地址和虚拟周期（包括 XIP 模式选择信息）的传输及数据接收使用 QSI00 管脚，奇数位使用 QSI01 管脚。

虚拟周期最初的两个周期用于决定是否选择 XIP 模式。当选择 XIP 模式时，本次传输所使用的指令将被应用于下一个 QSPI 总线周期，在下一个 QSPI 总线周期时指令传输部分将会被省略。详细可参见“XIP 控制”。

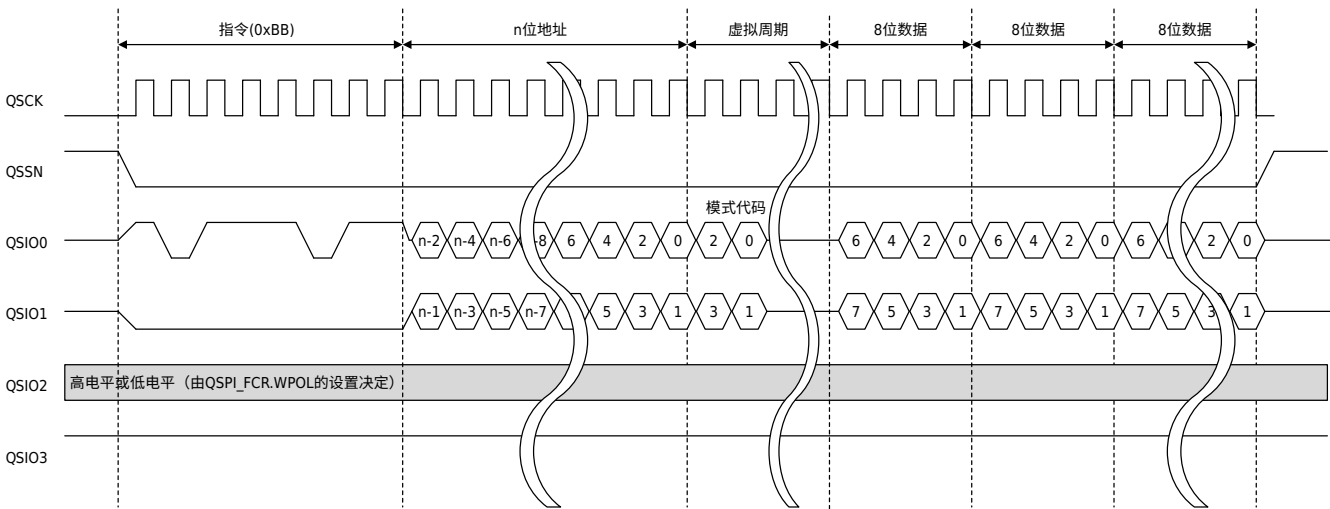


图 30-18 二线式输入输出快速读总线周期示意图

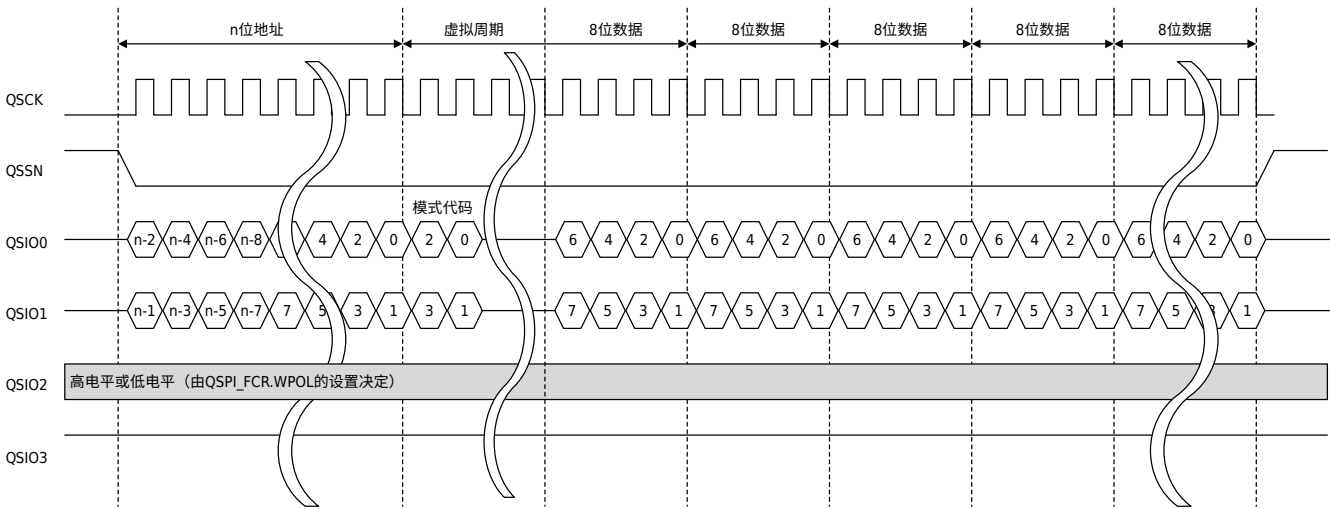


图 30-19 选择 XIP 模式的二线式输入输出快速读总线周期示意图



说明

要使用二线式输入输出快速读指令请确保使用支持该功能的串行闪存。

30.4.4.6 四线式输出快速读指令

四线式输出快速读是一种使用四根信号线进行数据接收的读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSIO0 管脚输出该指令的指令代码 (0x6B/0x6C) 和目标地址，地址宽度可以通过 QSPI_FCR.AWSL 位来设置。在这之后是一定数量的虚拟周期，其具体数量由 QSPI_FCR.DMCYCN 决定。然后开始通过 QSIO0, QSIO1, QSIO2 和 QSIO3 四根管脚进行数据接收。虚拟周期最初的两个周期用于决定是否选择 XIP 模式。当选择 XIP 模式时，本次传输所使用的指令将被应用于下一个 SPI 总线周期，在下一个 SPI 总线周期时指令传输部分将会被省略。详细可参见“XIP 控制”。

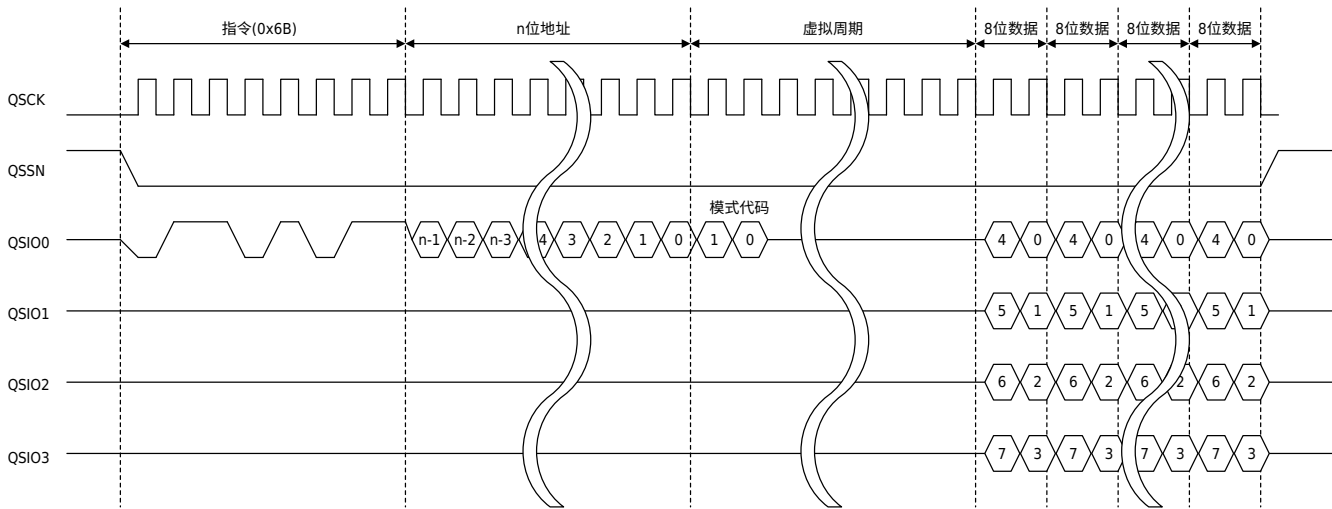


图 30-20 四线式输出快速读总线周期示意图

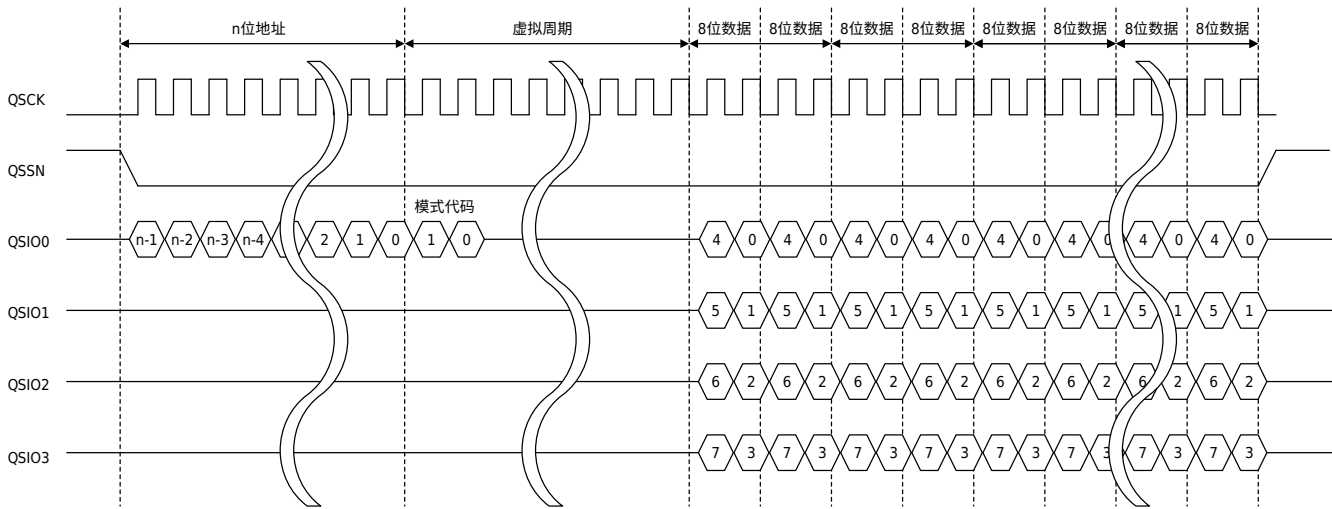


图 30-21 选择 XIP 模式的四线式输出快速读总线周期示意图



说明

要使用四线式输出快速读指令请确保使用支持该功能的串行闪存。

30.4.4.7 四线式输入输出快速读指令

四线式输入输出快速读是一种使用四根信号线进行地址发送和数据接收的读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSI00 管脚输出该指令的指令代码 (0xEB/0xEC)。在这之后 QSPI 从 QSI00, QSI01, QSI02 和 QSI03 四个管脚输出目标地址，地址宽度可以通过 QSPI_FCR.AWSL 位来设置。在这之后是一定数量的虚拟周期，其具体数量由 QSPI_FCR.DMCYN 决定。然后开始通过 QSI00, QSI01, QSI02 和 QSI03 四根管脚进行数据接收。

虚拟周期最初的两个周期用于决定是否选择 XIP 模式。当选择 XIP 模式时，本次传输所使用的指令将被应用于下一个 QSPI 总线周期，在下一个 QSPI 总线周期时指令传输部分将会被省略。详细可参见“XIP 控制”。

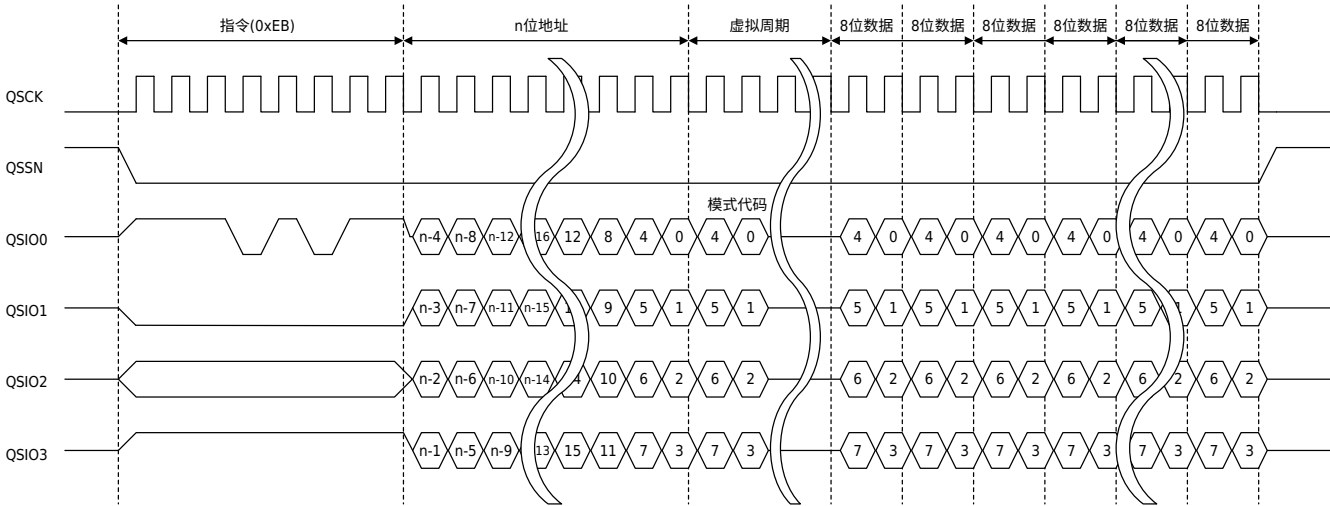


图 30-22 四线式输入输出快速读总线周期示意图

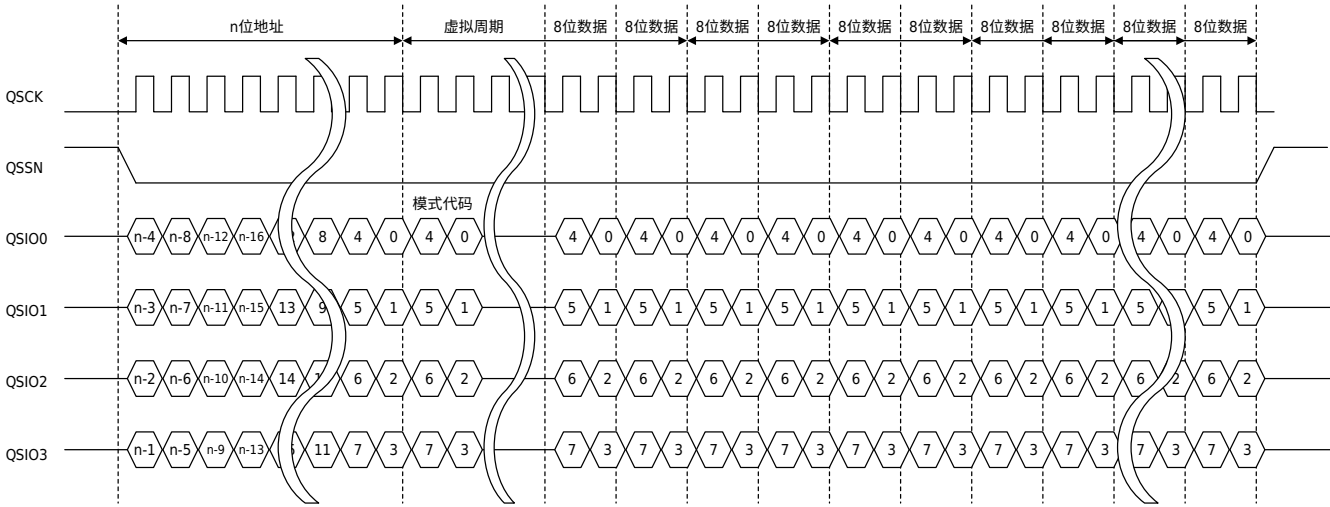


图 30-23 选择 XIP 模式的四线式输入输出快速读总线周期示意图

说明

要使用四线式输入输出快速读指令请确保使用支持该功能的串行闪存。

30.4.4.8 进入 4-Bytes 模式指令

进入 4-Bytes 模式指令可以将串行闪存地址宽度设置成 4 个字节。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSI00 管脚输出该指令的指令代码（0xB7）。

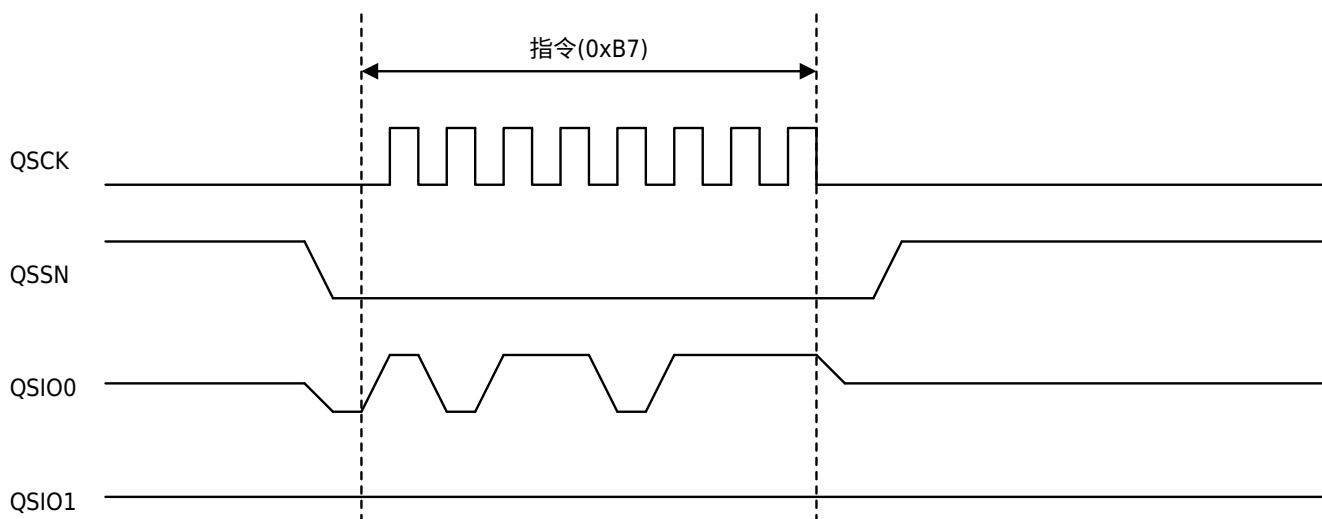


图 30-24 进入 4-Bytes 模式指令总线周期示意图



说明

无论串行闪存是否处于 4-Bytes 或是 3-Bytes 模式，该指令都可发布。

30.4.4.9 退出 4-Bytes 模式指令

退出 4-Bytes 模式指令可以将串行闪存地址宽度设置成 3 个字节。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSI00 管脚输出该指令的指令代码（0xE9）。

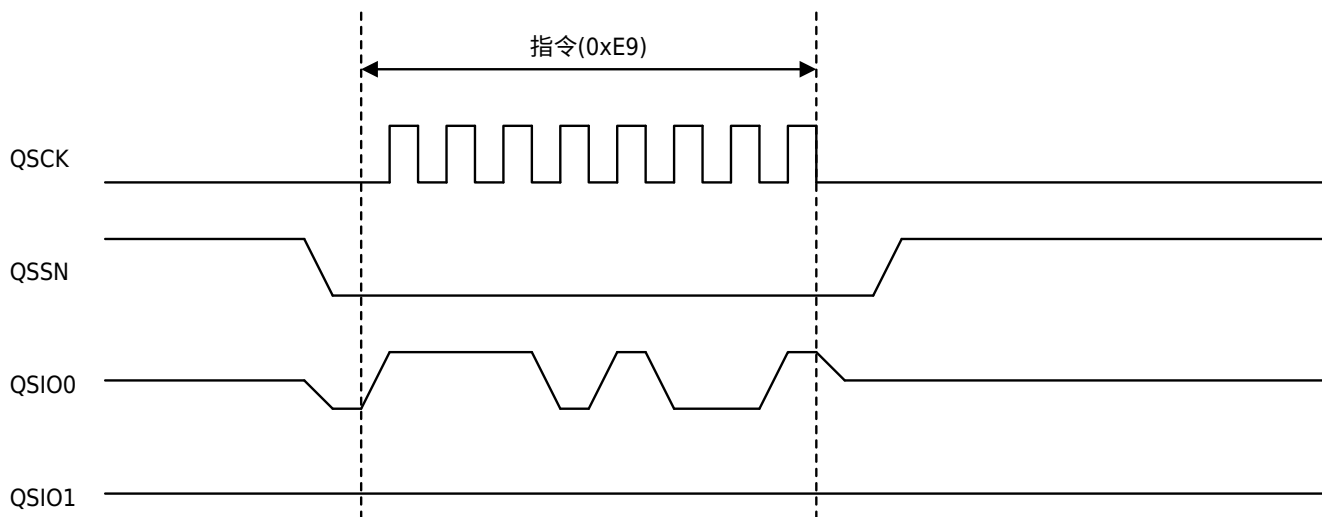


图 30-25 退出 4-Bytes 模式指令总线周期示意图



说明

无论串行闪存是否处于 4-Bytes 或是 3-Bytes 模式，该指令都可发布。

30.4.4.10 写许可指令

写许可指令允许串行闪存进行写操作。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSI00 管脚输出该指令的指令代码（0x06）。

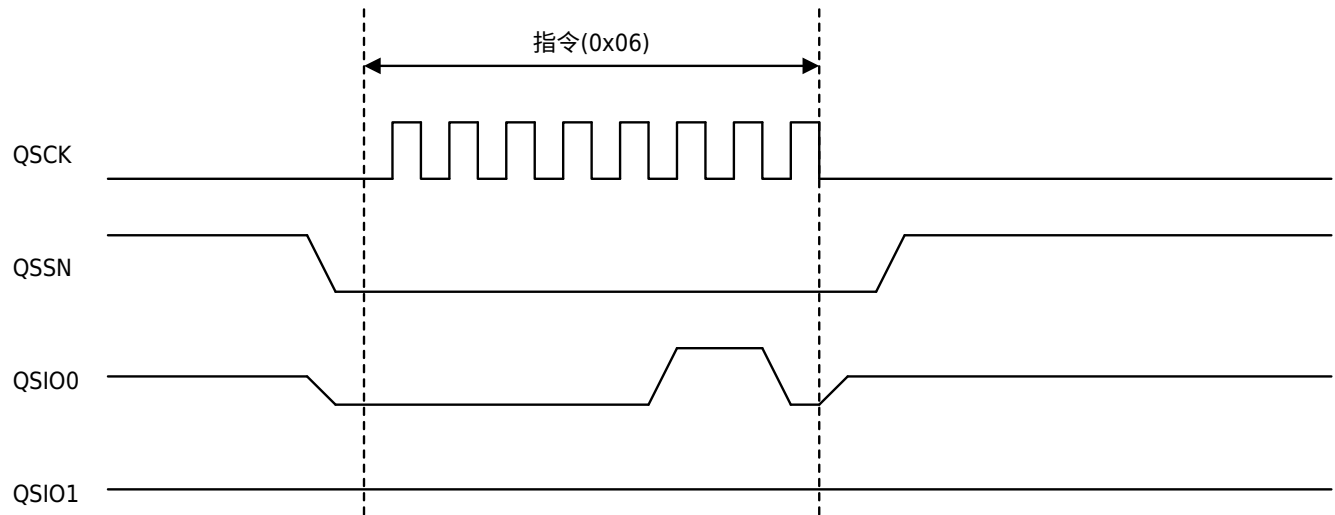


图 30-26 写许可指令总线周期示意图

30.4.5 QSPI 总线周期的安排

30.4.5.1 独立转换的单一闪存读取

对于 ROM 的单一读取指令会独立的从芯片内部总线周期一对一的转换为 QSPI 总线周期。当一个 ROM 的读取总线周期被检测到时，QSSN 信号会置为有效状态，从而启动一个 QSPI 总线周期。当接收完串行闪存的数据后，QSSN 信号变成无效状态，该 QSPI 总线周期宣告完成。

当另一个 ROM 的读取总线周期被检测到时，QSSN 信号会在确保无效保持时间已经超过最小无效保持宽度后再度置为有效状态，新一个 QSPI 总线周期开始。

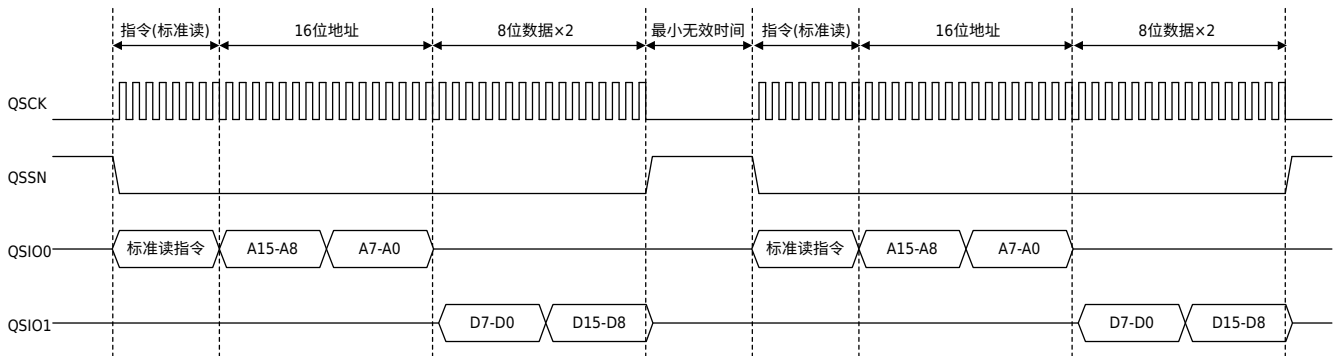


图 30-27 独立转换的单一闪存数据读取操作示意图

30.4.5.2 使用预读取功能进行闪存读取

对于诸如 CPU 指令或是数据块的传输，系统通常是以一个顺次递增的闪存地址顺序进行数据的读取。串行闪存具有连续数据传输能力而不需要再次发送指令代码和地址。但如果 MCU 发布的内部总线周期是独立转换的话，QSPI 总线周期也被分成了各自独立的个体，导致无法有效利用串行闪存的这种连续数据传输的优点。对此，QSPI 提供了预读取功能来进行连续的数据接收。

通过设置 QSPI_CR.PFE 位为 1 激活预读取功能。当该功能有效后，数据会被连续接收并储存到缓冲区而不需要等待另一个闪存读取要求。当 MCU 发出一个闪存读取操作时，QSPI 将会对访问地址进行匹配。如果匹配成功，则将对对应位置的缓冲区的数据传给 MCU，若匹配失败，则缓冲区的数据将会被丢弃并重新发出一个新的 QSPI 总线周期。

预读取的缓冲区最大可存储 16 个字节的数据，除此以外，还有 2 个字节的数据接收缓冲区也可以存储预读取的数据，当所有的缓冲区数据取满后，QSPI 总线周期结束。当缓冲区数据被读取后产生新的缓冲空间，QSPI 会自动开始一个新的 QSPI 总线周期来恢复预读取动作。

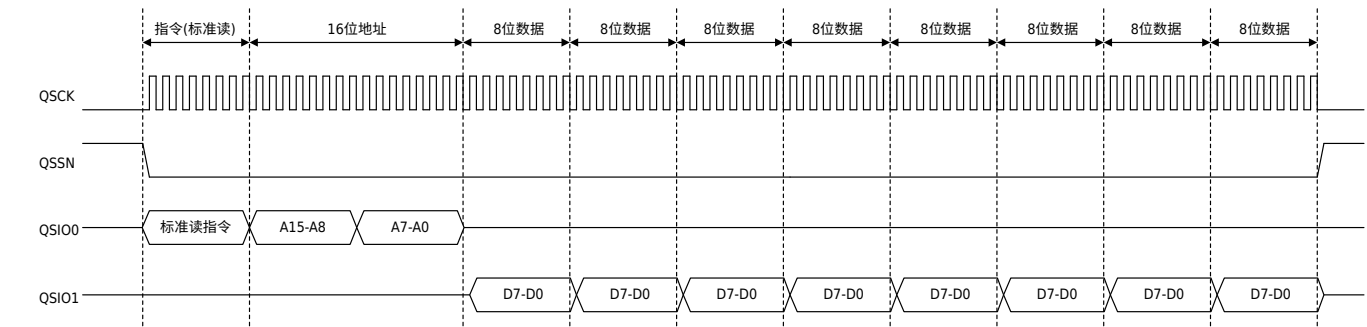


图 30-28 预读取功能有效时数据读取操作示意图

30.4.5.3 预读取的终止

在预读取传送过程中如果发生了对串行闪存其他地址的 ROM 读取总线周期，那么原预读取动作将会被终止并开始一个新的 QSPI 总线周期。通常情况下预读取动作会在当前字节传输完成之后终止，但如果将 QSPI_CR.PFSAE 位设置成 1 的话，QSPI 会即时停止预读取动作而不会等到当前字节传输完成。使用这一功能需要串行闪存器件支持即时停止动作功能。

30.4.5.4 预读取状态监测

从一个低速串行闪存读取数据会增加系统的负荷，这是因为内部总线需要一直处在等待状态直到收信的 QSPI 总线周期完成。QSPI 提供了预读取状态监测功能来降低这种负荷。

在预读取状态寄存器 QSPI_SR 中，PFAN 位显示了当前的预读取工作状态，PFFUL 位表示预读取数据缓冲区已经放满，而 PFNUM 位则显示了目前已经读取到缓冲区的数据的字节数。通过这些状态位可以很方便的通过一条 CPU 指令来确定当前的预读取状况。



说明

当执行一段预读取状态监测程序时，请将该程序代码放置在对象串行闪存区域以外或者启用指令缓存。否则预读取对象将频繁的切换在对象数据区域和指令区域之间，失去了预读取的意义，而监测程序也会因为预读取始终无法完成而进入无限循环状态。

30.4.5.5 使用 QSPI 总线周期延长功能进行闪存读取

如果对 QSPI_CSCR.SSNW 进行设置 0b00 以外的值，QSPI 总线周期将会在接收完数据后处于保持状态等待下一个数据的读取。这时 QSSN 信号将保持低电平的有效状态，而 QSCCK 则处于停止状态。如果下一条闪存读取指令到来时，如果读取对象地址是紧接着当前地址顺序递增的，则 QSPI 会将 QSCCK 重新启动直接进行数据接收。如果是非连续的地址的话，QSSN 信号会被置成高电平的无效状态将原先保持的 QSPI 总线周期结束，然后重新开始一个新的 SPI 总线周期。

这一功能可以避免系统在对连续递增地址进行不连续读取时反复发送指令代码和地址，从而提高读取效率。QSPI 总线周期的延长时间可以通过 QSPI_CSCR.SSNW 设置，当超过所设置的时间仍未发生下一次的读取时，QSSN 会自动置成高电平的无效状态，结束 QSPI 总线周期。如果 QSPI_CSCR.SSNW 设成 0b11 的话，QSSN 会被无限期延长，QSPI 总线周期将始终处在保持状态，但这样做会增加串行闪存的功耗。

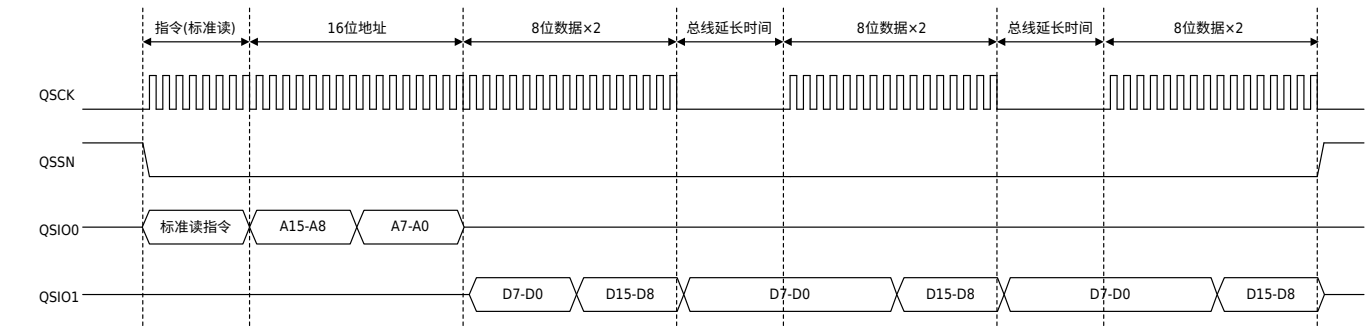


图 30-29 使用 QSPI 总线周期延长功能的数据读取操作示意图

30.4.6 XIP 控制

30.4.6.1 XIP 模式概述

一些串行闪存器件可以通过省略接收读取指令来降低延迟时间。该机能可通过虚拟周期期间发送的模式代码来选择。

在快速度指令时的虚拟周期期间，QSPI 在最初的两个周期通过发送 XIP 模式代码来控制串行闪存的 XIP 模式。不同的串行闪存其 XIP 模式代码也不相同，可通过寄存器 QSPI_XCMD.XIPMC 位进行针对性设置。具体参照下图。

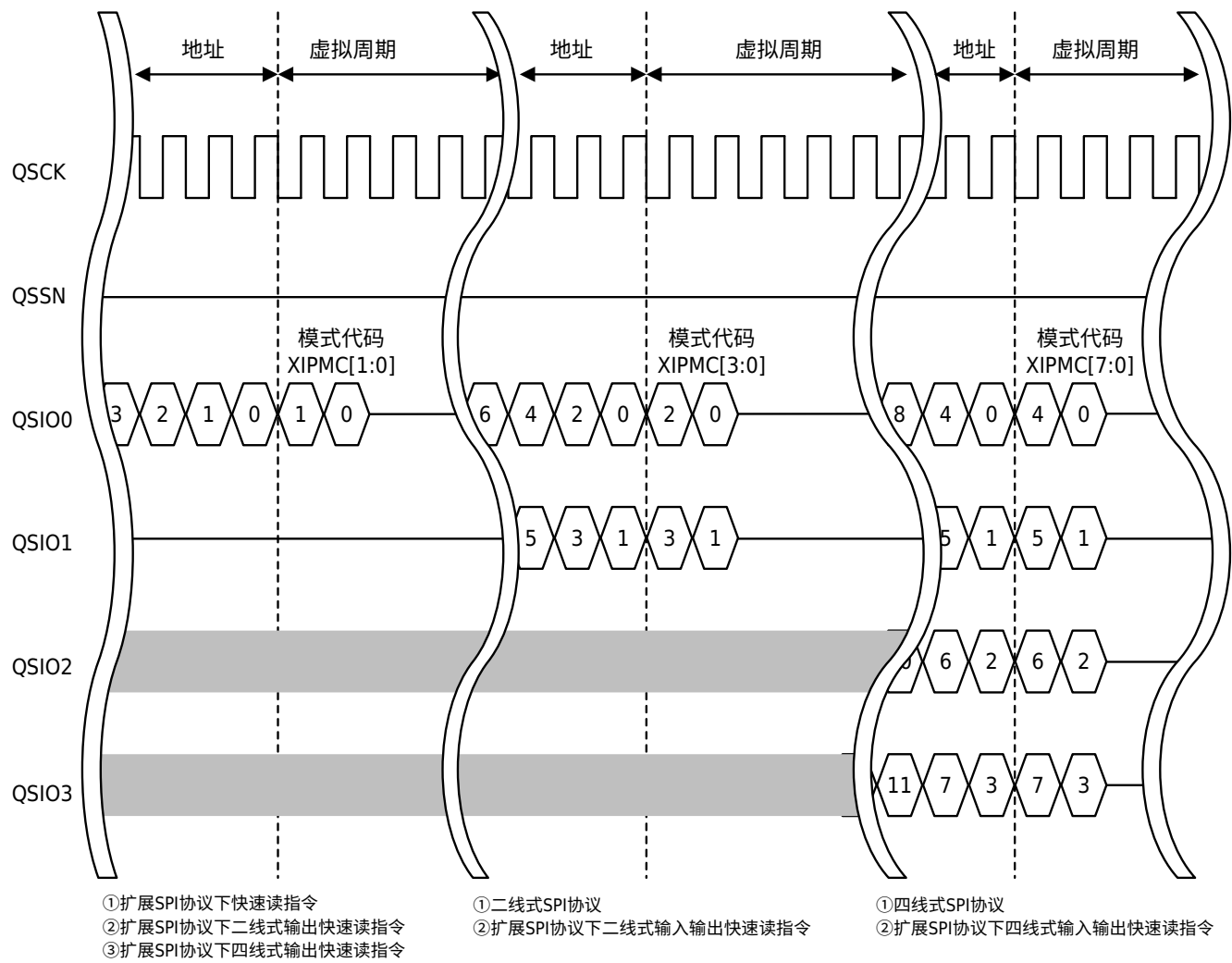


图 30-30 XIP 模式控制示意图

30.4.6.2 XIP 模式的设置

在将对应串行闪存的 XIP 模式代码写入寄存器 QSPI_XCMD.XIPMC 且寄存器 QSPI_CR.XIPE 位设置成 1 的情况下，当下一个快速读指令发生时，所设置的模式代码将在虚拟周期的前两个周期期间传送给对象串行闪存，在模式代码接收完成后，串行闪存及其控制部就启动 XIP 模式。可以通过访问 QSPI_SR.XIPF 位来确认是否已经进入 XIP 模式。

说明

启动串行闪存的 XIP 模式需要在 QSPI_XCMD.XIPMC 中设置相应的模式代码，控制部分的 XIP 模式只需要将 XIPE 位置成 1 就可以，而与 QSPI_XCMD.XIPMC 的值无关。

30.4.6.3 XIP 模式的退出

在将对应串行闪存的退出 XIP 模式的代码写入寄存器 QSPI_XCMD.XIPMC 且寄存器 QSPI_CR.XIPE 位设置成 0 的情况下，当下一个快速读指令发生时，所设置的退出模式代码将在虚拟周期的前两个周期期间传送给对象串行闪存，在退出模式代码接收完成后，串行闪存及其控制部的 XIP 模式被终止。可以通过访问 QSPI_SR.XIPF 位来确认是否已经退出 XIP 模式。

 **说明**

退出串行闪存的 XIP 模式需要在 QSPI_XCMD.XIPMC 中设置相应的退出模式代码，控制部分的 XIP 模式只需要将 XIPE 位清零就可以，而与 QSPI_XCMD.XIPMC 的值无关。

30.4.7 QSI02 和 QSI03 的管脚状态

QSI02 和 QSI03 管脚状态取决于 QSPI_CR 寄存器中 MDSEL[2:0]位所设置的串行读取模式。

表 30-4 QSI02 和 QSI03 的管脚状态

QSPI_CR.MDSEL	QSI02 状态	QSI03 状态	备注
0b000	输出状态，输出电平由 QSPI_FCR.WPOL 位决定，初始输出为低电平	输出高电平	标准读（初始状态）
0b001			快速读
0b010			二线式输出快速读
0b011			二线式输入输出快速读
0b100	作为第三根数据线进行输入或输出动作，空闲状态为 Hi-Z	作为第四根数据线进行输入或输出动作，空闲状态为 Hi-Z	四线式输出快速读
0b101			四线式输入输出快速读
0b110	参照各个阶段具体的协议设置	参照各个阶段具体的协议设置	自定义协议标准读
0b111			自定义协议快速读

 **说明**

- QSI02 管脚也可用作串行闪存的 WP# 功能。
- QSI03 管脚也可用作串行闪存 HOLD# 或 RESET# 功能。

30.4.8 直接通信模式

30.4.8.1 关于直接通信模式

QSPI 可以通过自动将 MCU 的外部 ROM 读取总线周期转换为 QSPI 总线周期来对串行闪存进行读取。但串行闪存还有很多不同的追加功能，诸如 ID 信息读取，擦除，写入及状态信息读取等。这些功能并没有一套标准的指令来进行设置，并且随着串行闪存新功能的迅速增加，硬件层面上的对应变得愈发的困难。针对这种情况，QSPI 提供了直接通信模式，用户可通过软件直接对串行闪存进行控制。由此模式软件可以产生任意所需的 QSPI 总线周期。

30.4.8.2 直接通信模式的设置

将 QSPI_CR.DCOME 位设成 1 可以进入直接通信模式。一旦进入直接通信模式，将无法进行通常的闪存读取操作，如果要进行常规的闪存读取，需要将 DCOME 位清零退出直接通信模式。

 **说明**

如果处于 XIP 模式，则需要先退出 XIP 模式再启动直接通信模式。

30.4.8.3 直接通信模式下 QSPI 总线周期的生成

直接通信模式下一个完整的 QSPI 总线周期从对寄存器 QSPI_DCOM 第一次操作开始直到对 QSPI_CR 寄存器的进行一次写操作后结束。在这期间可以对 QSPI_DCOM.DCOMPRSL 和 QSPI_DCOM.DCOM 进行多次操作。直接通信模式支持多线式动作。

- 当 DCOMPRSL[1:0] 设置为 0b00 时，DCOM[7:0] 的写入转换为一次 QSPI 总线扩展式 SPI 协议的单字节的数据传送，而对 DCOM[7:0] 的读则会转换成一次 QSPI 总线扩展式 SPI 协议的单字节的数据接收。
- 当 DCOMPRSL[1:0] 设置为 0b01 时，DCOM[7:0] 的写入转换为一次 QSPI 总线二线式 SPI 协议的单字节的数据传送，而对 DCOM[7:0] 的读则会转换成一次 QSPI 总线二线式 SPI 协议的单字节的数据接收。
- 当 DCOMPRSL[1:0] 设置为 0b10 时，DCOM[7:0] 的写入转换为一次 QSPI 总线四线式 SPI 协议的单字节的数据传送，而对 DCOM[7:0] 的读则会转换成一次 QSPI 总线四线式 SPI 协议的单字节的数据接收。

从对寄存器 QSPI_DCOM 的第一次操作开始到最后对 QSPI_CR 寄存器的写操作为止，这期间 QSSN 信号始终保持低电平的有效状态。



说明

- 在直接通信模式下对 QSPI_DCOM 以外的 QSPI 寄存器进行写操作，都会退出直接通信模式。使用“QSPI_CR 寄存器写操作”以外的方式退出直接通信模式可能会导致不可预料的情况出现，所以不推荐使用。
- 直接通信模式下可以通过 DCOMPRSL[1:0] 的改写来自由切换一线式、二线式或四线式。如果是直接读操作之前需要切换线式，请注意 DCOMPRSL[1:0] 的写入会带来一次 Dummy 数据传送。
- 对 QSPI_DCOM 寄存器进行写操作时，必须使用 32 位或 16 位操作，同时写入 DCOMPRSL[1:0] 和 DCOM[7:0] 位。对 QSPI_DCOM 寄存器进行读操作时，可以使用 32 位、16 位或 8 位操作。
- 直接通信模式下，仅在最快基准时钟时支持二线式和四线式，即 QSPI_CR.DIV 设置为 0b000000 或 0b000001。

30.4.9 中断

当直接通信模式下检测到对 ROM 的读访问操作时，QSPI_SR.RAER 位被置成 1，这时候 QSPI 将产生总线硬件错误中断。该中断请求将会一直被保留直到 RAER 位被清零。详细信息请参见《中断控制器 (INTC) 》。

30.5 寄存器

30.5.1 寄存器总表

基地址：0x9C000000

表 30-5 四线式串行外设接口 (QSPI) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x0000	QSPI_CR	QSPI 控制寄存器	32
0x0004	QSPI_CSCR	QSPI 片选控制寄存器	32
0x0008	QSPI_FCR	QSPI 格式控制寄存器	32
0x000C	QSPI_SR	QSPI 状态寄存器	32
0x0010	QSPI_DCOM	QSPI 直接通信指令寄存器	32
0x0014	QSPI_CCMD	QSPI 指令代码寄存器	32
0x0018	QSPI_XCMD	QSPI XIP 模式代码寄存器	32
0x0024	QSPI_CLR	QSPI 标志清除寄存器	32
0x0804	QSPI_EXAR	QSPI 外部扩展地址寄存器	32

30.5.2 QSPI 控制寄存器 (QSPI_CR)

Offset	Bit Position															
0x0000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reset	0x003F0000															
Name	Reserved				DIV				Reserved	DPRSL	APRSL	IPRSL	SPIMD3	XIPE	DCOME	PFSAE
Access					RW					RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:22	Reserved	保留	-	-
21:16	DIV	基准时钟选择	串行接口基准时钟周期配置： ● 0b000000：2 个 HCLK 周期 ● 0b000001：2 个 HCLK 周期 ● 0b000010：3 个 HCLK 周期 ● ● 0b111110：63 个 HCLK 周期 ● 0b111111：64 个 HCLK 周期  说明 硬件实际使用值为编程值加 1。	RW
15:14	Reserved	保留	-	-
13:12	DPRSL	数据接收阶段 SPI 协议选择	● 0b00：扩展式 SPI 协议 ● 0b01：二线式 SPI 协议 ● 0b10：四线式 SPI 协议 ● 0b11：禁止设置	RW
11:10	APRSL	地址发送阶段 SPI 协议选择	● 0b00：扩展式 SPI 协议 ● 0b01：二线式 SPI 协议 ● 0b10：四线式 SPI 协议 ● 0b11：禁止设置	RW
9:8	IPRSL	指令发送阶段 SPI 协议选择	● 0b00：扩展式 SPI 协议 ● 0b01：二线式 SPI 协议 ● 0b10：四线式 SPI 协议 ● 0b11：禁止设置	RW
7	SPIMD3	SPI 模式选择	● 0b0：SPI 模式 0 ● 0b1：SPI 模式 3	RW
6	XIPE	XIP 模式许可	● 0b0：XIP 模式禁止 ● 0b1：XIP 模式许可	RW
5	DCOME	直接通信许可	QSPI 总线通信模式选择： ● 0b0：ROM 访问模式 ● 0b1：直接通信模式	RW
4	PFSAE	预读取即时停止许可	选择重置预读取动作的位置： ● 0b0：当前的预读取动作在字节边界中止 ● 0b1：当前的预读取动作即时中止	RW

位/位域	标记	位名	功能描述	读写
3	PFE	预读取许可	0b0: 预读取功能无效 0b1: 预读取功能有效	RW
2:0	MDSEL	QSPI 读取模式选择	0b000: 标准读 0b001: 快速读 0b010: 二线式输出快速读 0b011: 二线式输入输出快速读 0b100: 四线式输出快速读 0b101: 四线式输入输出快速读 0b110: 自定义标准读 0b111: 自定义快速读	RW

30.5.3 QSPI 片选控制寄存器 (QSPI_CSCR)

Offset	Bit Position																															
0x0004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000000F																															
Name	Reserved																										SSNW		SSHW			
Access																											RW		RW			

位/位域	标记	位名	功能描述	读写
31:6	Reserved	保留	-	-
5:4	SSNW	QSSN 有效时间延长设置	QSPI 总线访问后 QSSN 有效时间延长功能选择: 0b00: 不延长 0b01: 延长 32 个 QSCK 周期 0b10: 延长 128 个 QSCK 周期 0b11: 无限延长  说明 直接模式下该位设定无效。	RW
3:0	SSHW	QSSN 最小无效时间设置	QSSN 信号最小无效时间选择: 0b0000: 1 个 QSCK 周期 0b0001: 2 个 QSCK 周期 0b1110: 15 个 QSCK 周期 0b1111: 16 个 QSCK 周期  说明 硬件实际使用值为编程值加 1。	RW

30.5.4 QSPI 格式控制寄存器 (QSPI FCR)

Offset	Bit Position																																							
0x0008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Reset	0x00008033																																							
Name	Reserved																DUTY	Reserved								DMCCYN				Reserved		WPOL	SSNLD	SSNHD	Reserved		4BIC		AWSL	
Access																	RW									RW						RW	RW	RW			RW		RW	

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15	DUTY	QSCK 输出波形占空比补正	0b0：不进行占空比补正 0b1：将 QSCK 的上升沿滞后 0.5 个 HCLK 周期（当 QSCK 选择的频率是 HCLK 的奇数倍时有效）	RW
14:12	Reserved	保留	-	-
11:8	DMCYCN	虚拟周期设置	使用快速读指令时虚拟周期数量选择： 0b0000：3 个 QSCK 周期 0b0001：4 个 QSCK 周期 0b1110：17 个 QSCK 周期 0b1111：18 个 QSCK 周期  说明 硬件实际使用值为编程值加 3。	RW
7	Reserved	保留	-	-
6	WPOL	WP 管脚（QSIO2）输出电平设置	0b0：低电平 0b1：高电平	RW
5	SSNLD	QSSN 信号输出时间延迟设置	0b0：比 QSCK 第一个上升沿提前 0.5 个 QSCK 输出 QSSN 0b1：比 QSCK 第一个上升沿提前 1.5 个 QSCK 输出 QSSN	RW
4	SSNHD	QSSN 信号释放时间延迟设置	0b0：比 QSCK 最后一个上升沿滞后 0.5 个 QSCK 释放 QSSN 0b1：比 QSCK 最后一个上升沿滞后 1.5 个 QSCK 释放 QSSN	RW
3	Reserved	保留	-	-
2	4BIC	4 字节地址读指令代码选择	当地址宽度为 4 字节时的读指令代码选择： 0b0：不使用 4 字节地址读指令代码 0b1：使用 4 字节地址读指令代码	RW
1:0	AWSL	串行接口地址宽度选择	0b00：1 字节 0b01：2 字节 0b10：3 字节 0b11：4 字节	RW

30.5.5 QSPI 状态寄存器 (QSPI_SR)

Offset	Bit Position																															
0x000C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00008000																															
Name	Reserved																RO PFAN	RO PFFUL	Reserved	RO PFNUM				RO RAER	RO XIPF	Reserved				RO BUSY		
Access																	RO	RO		RO				RO	RO					RO		

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15	PFAN	预读取动作状态	0b0: 预读取处于动作状态 0b1: 预读取处于停止状态	RO
14	PFFUL	预读取缓冲区状态	0b0: 预读取缓冲区有剩余空间 0b1: 预读取缓冲区数据已满	RO
13	Reserved	保留	-	-
12:8	PFNUM	预读取缓冲区已存储数据字节数	0b00000: 0 个字节 0b00001: 1 个字节 0b10001: 17 个字节 0b10010: 18 个字节 - 其他: Reserved	RO
7	RAER	ROM 访问错误标志	直接通信模式下发生 ROM 访问的错误标志位: 0b0: 未检测到发生 ROM 访问 0b1: 检测到发生 ROM 访问 清零条件: - QSPI_CLR 的 RAERCLR 位置 1	RO
6	XIPF	XIP 模式标志	0b0: 非 XIP 模式 0b1: XIP 模式	RO
5:1	Reserved	保留	-	-
0	BUSY	总线忙标志	直接通信模式下 QSPI 总线工作状态标志位: 0b0: 总线空闲, 无串行传输进程 0b1: 总线忙, 串行传输进程进行中	RO

30.5.6 QSPI 直接通信指令寄存器 (QSPI_DCOM)

Offset	Bit Position																															
0x0010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved																						DCOMPRSL		DCOM							
Access																							RW		RW							

位/位域	标记	位名	功能描述	读写
31:10	Reserved	保留	-	-
9:8	DCOMPRSL	直接通信模式下 SPI 协议选择	0b00: 扩展式 SPI 协议 0b01: 二线式 SPI 协议 0b10: 四线式 SPI 协议 0b11: 禁止设置	RW
7:0	DCOM	直接通信模式指令	直接通信模式时的接口，通过 QSPI 总线进行直接通信。对该接口的读写访问会被转化成一个相应的 QSPI 总线周期。该接口只在直接通信模式下有效，ROM 访问模式下该接口禁止访问。	RW

30.5.7 QSPI 指令代码寄存器 (QSPI_CCMD)

Offset	Bit Position																															
0x0014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								RW R/C							
Access																																

位/位域	标记	位名	功能描述	读写
31:8	Reserved	保留	-	-
7:0	RIC	替换指令代码	用于替代默认指令的串行闪存指令代码	RW

30.5.8 QSPI XIP 模式代码寄存器 (QSPI_XCMD)

Offset	Bit Position																															
0x0018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000000FF																															
Name	Reserved																								XIPMC							
Access																									RW							

位/位域	标记	位名	功能描述	读写
31:8	Reserved	保留	-	-
7:0	XIPMC	XIP 模式代码	用于串行闪存的模式代码。(设置 XIP 模式)	RW

30.5.9 QSPI 标志清除寄存器 (QSPI_CLR)

Offset	Bit Position																															
0x0024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved																								RAERCLR		Reserved					
Access																									WO							

位/位域	标记	位名	功能描述	读写
31:8	Reserved	保留	-	-
7	RAERCLR	RAER 标志清除	写 1 将 QSPI_SR.RAER 位清零, 写 0 无效	WO
6:0	Reserved	保留	-	-

30.5.10 QSPI 外部扩展地址寄存器 (QSPI_EXAR)

Offset	Bit Position																															
0x0804	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	RW EXADR					Reserved																										
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:26	EXADR	外部扩展地址代码	QSPI 外部地址高 6 位设置, 配合 QSPI 的 ROM 访问窗口地址最大可访问 64MB×64 块的外部 ROM 空间	RW
25:0	Reserved	保留	-	-

30.6 注意事项

30.6.1 QSPI 寄存器的设置

如果在 QSPI 通信过程中动态地对 QSPI 控制寄存器进行设置或更改，可能会导致寄存器还没有完全设置完成就开始新的通信，因此不建议在通信期间动态修改 QSPI 控制寄存器。

30.6.2 模块停止位的设置

QSPI 在系统复位后处于模块停止状态。需要 QSPI 模块工作时，请先将功能时钟控制寄存器 PWC_FCG 中的 QSPI 模块控制位清零。具体描述请参见《电源控制 (PWC)》章节“动作模式与低功耗模式”。

31 外部存储器控制器 (EXMC)

31.1 简介

外部存储器控制器 EXMC (External Memory Controller) 是一个用来访问各种片外存储器，实现数据交换的独立模块。EXMC 通过配置可以把内部的 AMBA 协议接口转换为各种类型的专用片外存储器通信协议接口，包括 SRAM、PSRAM、NOR Flash 等。

31.2 主要特性

支持的特性如下：

- 支持的外部存储器类型：SRAM、PSRAM、NOR Flash
- 接口协议：支持 AHB 协议与各种外部存储器的接口转换
- 接口时序：各种类型存储器的接口时序可编程
- 总线位宽：支持 8 位、16 位 MEM 总线宽度
- 总线复用：SRAM、NOR Flash 和 PSRAM 支持地址线和数据线复用
- 自动分割：AHB 总线位宽和外部存储器位宽不匹配时，支持自动分割及字节选择控制

31.3 功能说明

31.3.1 控制器架构

SMC 外部存储器控制器独立生成对应协议的接口送至端口。

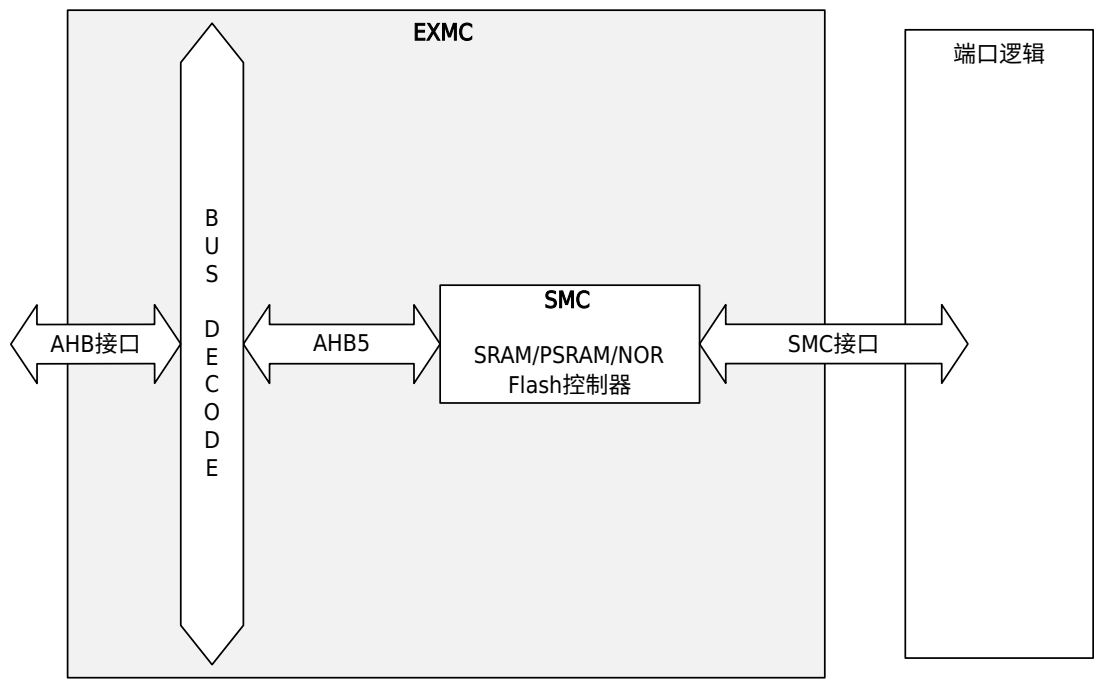


图 31-1 EXMC 架构图

 **说明**
架构图中将 SRAM/PSRAM/NOR Flash 控制器定义为 SMC（Static Memory Controller）。

31.3.2 基本访问规范

EXMC 是 AHB 总线协议和外部存储器协议的转换接口。在 SMC 的数据传输过程中，AHB 数据宽度和存储器数据宽度可能不相同。为了保证数据传输的一致性，读写访问需要遵从以下规范：

- 1. AHB 访问宽度等于存储器宽度，数据正常访问；
- 2. AHB 访问宽度大于存储器宽度，自动将 AHB 访问分割成几个连续的存储器数据宽度的传输；
- 3. AHB 访问宽度小于存储器宽度，通过字节选择控制信号 SMC_BLS[1:0]来访问对应的字节。

AHB 访问宽度和各种数据位宽的存储器之间的具体的访问方式参见下表。

表 31-1 AHB 访问宽度和存储器位宽对应访问方式表

RW	AHB 访问宽度	外部存储器位宽	说明
R	8	8	-
	16		转换成 2 次 EXMC 读操作
	32		转换成 4 次 EXMC 读操作
	8	16	-
	16		-
	32		转换成 2 次 EXMC 读操作

RW	AHB 访问宽度	外部存储器位宽	说明
W	8	8	-
	16		转换成 2 次 EXMC 写操作
	32		转换成 4 次 EXMC 写操作
	8	16	使用字节控制信号 BLS[0]或 BLS[1]
	16		-
	32		转换成 2 次 EXMC 写操作



说明

上表仅适用于 SMC 的数据访问。

31.3.3 地址映射

本产品中定义了总大小为 1GB 的外部存储器访问区间，用于不同类型的外部存储器和内部数据交换，包括 EXMC 和 QSPI 等。该 1GB 的空间按下图的方案分配，SRAM/PSRAM/NOR Flash (SMC) 的数据访问映射 512MB 的地址空间、QSPI 的控制和数据访问映射 128MB 的地址空间。

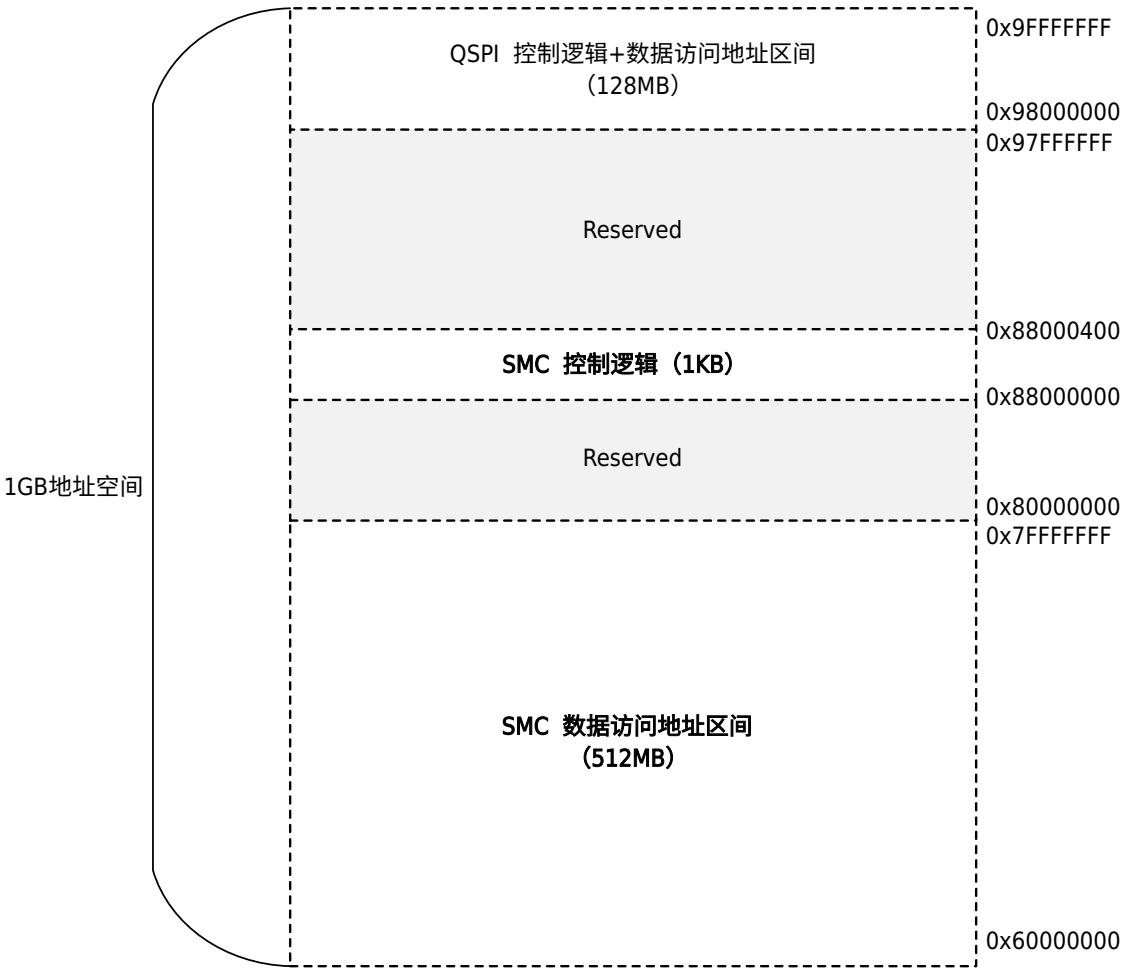


图 31-2 外部空间地址分配



说明

- 1. 地址空间为 0x98000000~0x9FFFFFFF 的访问控制由 QSPI 模块实现，具体请参考《四线式串行外设接口 (QSPI) 》章节相关描述。

SMC 地址空间

SMC 的地址空间可划分为 1 个大小可编程的 Chip，通过片选控制寄存器 SMC_CSCR0 的设定来划分 Chip 的空间大小，对应的 CS (Chip Select) 信号输出至外部端口。下图是 SMC 的 Chip Select 划分，当 CPU 访问的地址空间在该 SMC 的 CS 空间时，其对应的 SMC_CS 信号输出就变为有效电平。

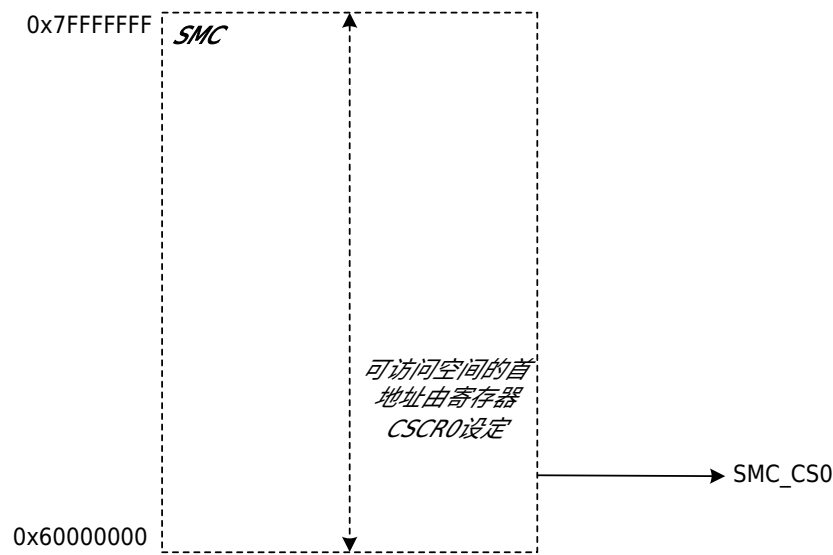


图 31-3 SMC 的地址空间划分

31.3.4 协议接口

SMC 对应的存储器的协议接口信号。EXMC 通过 SMC 实现的接口协议和外部存储器进行数据交换。以下为 SMC 存储器所需要的协议接口说明。

SMC 协议接口

下表为 SRAM/PSRAM/NOR Flash 控制（SMC）访问所需的协议接口。

表 31-2 SMC 协议接口

协议接口名	方向	有效电平	功能描述
SMC_CLK	输出	高电平/低电平	SMC 的时钟输出
SMC_ADD[20:16]	输出	高电平/低电平	SMC 的地址输出
SMC_DATA[15:0]	输入/输出	高电平/低电平	SMC 的地址/数据复用
SMC_WE	输出	低电平	SMC 的写使能
SMC_OE	输出	低电平	SMC 的输出使能
SMC_CS0	输出	低电平	SMC 的片选信号
SMC_BLS[1:0]	输出	低电平	SMC 的字节选通信号
SMC_ADV	输出	低电平	SMC 的地址锁存信号
SMC_CRE	输出	高电平	SMC 的配置寄存器模式信号
SMC_WAIT	输入	低电平	SMC 的输入等待信号
SMC_BAA	输出	低电平	SMC 的地址提示信号

EXMC 端口功能配置

基于上述 SMC 外部存储器的接口信号表，芯片的 EXMC 端口功能配置如下表。

表 31-3 EXMC 端口功能分配

引脚名	方向	有效电平	功能描述
			SMC
EXMC_CLK	输出	高电平/低电平	SMC_CLK
EXMC_ADD20~16	输出	高电平/低电平	SMC_ADD[20:16]
EXMC_DATA15~0	输入/输出	高电平/低电平	SMC_DATA[15:0]
EXMC_WE	输出	低电平	SMC_WE
EXMC_CE5~4	输出	低电平	SMC_BLS[1:0]
EXMC_CE0	输出	低电平	SMC_CS0
EXMC_OE	输出	低电平	SMC_OE
EXMC_BAA	输出	低电平	SMC_BAA
EXMC_ADV	输出	低电平	SMC_ADV
EXMC_ALE	输出	高电平	SMC_CRE
EXMC_RB0	输入	低电平	SMC_WAIT

31.3.5 SMC-SRAM/PSRAM/NOR Flash 控制器

31.3.5.1 SRAM/PSRAM/NOR Flash 介绍

SRAM/NOR Flash 都是具有静态存取功能的存储器，不需要刷新电路即能保存它内部存储的数据。其地址线的宽度决定了存储单元的数量，每一个存储单元上对应存储器存储宽度大小的空间用于存储数据，因此 SRAM/PSRAM/NOR Flash 存储器的容量大小是存储器数据宽度 $\times 2^{\text{地址线宽度}}$ 。

PSRAM 是假静态随机存储器，由于它具有 SRAM 的接口，因此能像 SRAM 一样，与系统内部通过 EXMC 的 SMC 模块实现数据交换。

31.3.5.2 SMC 基本功能

SMC 基本功能特性如下：

- 支持 8 位、16 位外部存储器数据带宽
- AHB 字、半字、字节访问
- 字节选择信号输出
- 地址线、数据线复用
- 可编程的协议时序参数
- 可编程速率的自动刷新动作（PSRAM 时使用）
- 具有 2 个命令 FIFO
- 具有 4 个 32 位的写数据 FIFO
- 具有 4 个 32 位的读数据 FIFO
- 低功耗管理
- 支持 LCD 并行接口，兼容 8080/6800 模式

31.3.5.3 SMC 初始设定

SMC 在上电复位之后的状态图及状态之间的切换如下图所示。
初始复位时 SMC 为 Low Power 状态；通过状态控制寄存器（SMC_STCR0 和 SMC_STCR1）的设定，SMC 可以在 Ready 状态和 Low Power 状态之间进行切换，具体操作请参见“SMC 低功耗管理”。

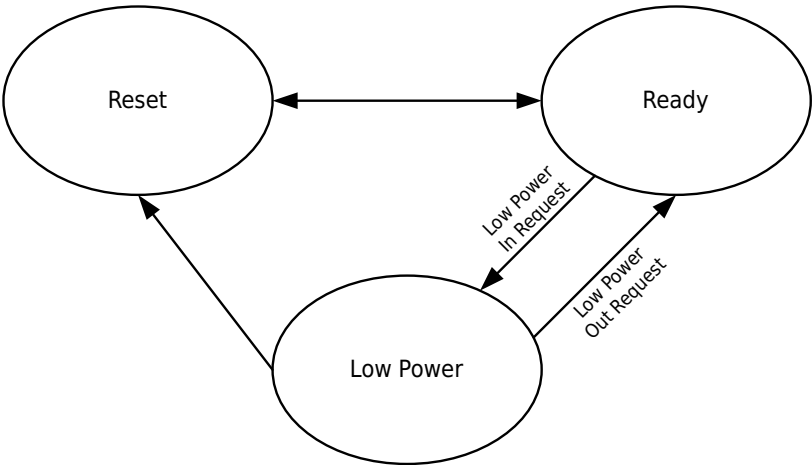


图 31-4 SMC 状态图

SMC 在和外部 SRAM、PSRAM、NOR Flash 等进行通信之前，必须进行初始设定，配置好相关的参数，以保证能进行正确的数据传输。具体的设定顺序参考下图。

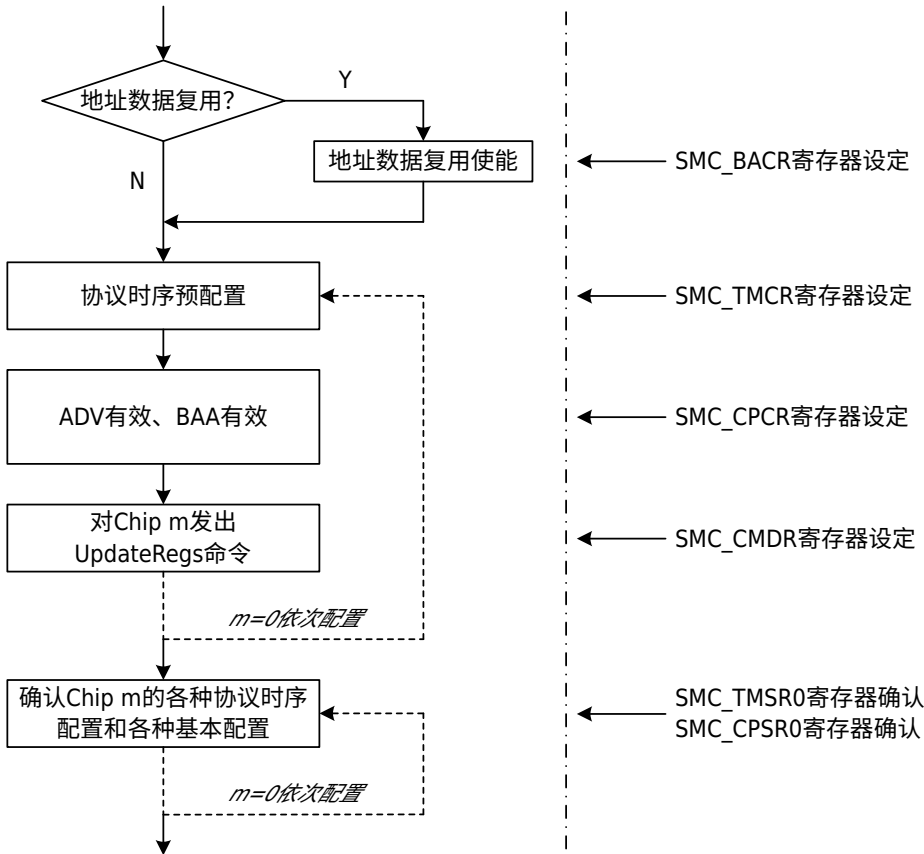


图 31-5 SMC 初始设定流程

31.3.5.4 SMC 访问动作

31.3.5.4.1 访问方式

SMC 对外部存储器的读写，可以选择同步访问方式或者异步访问方式。

同步访问方式是对应同步访问类型的外部存储器，SMC 在读写控制信号等输出时，时钟信号也同时输出，外部存储器根据 SMC 输出的时钟与 SMC 进行数据传输，即 SMC_CLK 有效；异步访问方式是对应异步访问类型的外部存储器，SMC 在读写控制信号等输出时，时钟信号不输出（输出始终为低），外部存储器依据其它控制信号实现数据访问。

在同步访问方式时，SMC 和外部存储器间通过 SMC_WAIT 信号来控制读写动作。在 SMC_WAIT 信号输入低电平时，SMC 处于等待状态，不执行任何操作，等到 SMC_WAIT 变为高电平后 SMC 才正常的进行读写访问。

通过设定 CHIP 配置寄存器（SMC_CPCR）的 RSYN、WSYN 位，可选择同步读写方式或异步读写方式。

 **说明**

若没有引脚配置为 EXMC_RB0 功能（详见《数据手册》“引脚配置及功能-引脚功能说明”），SMC 的读写动作将不受 SMC_WAIT 信号的控制。

31.3.5.4.2 访问时序

- SMC 支持多种类型的存储器的访问，可以把 AHB 的单次读写操作转换成存储器的读写操作。下面列出几种控制器与外部存储器进行数据读写的时序例：
- 1. 单次读动作
 - 2. 地址数据线复用的单次读动作
 - 3. 单次写动作
 - 4. 地址数据线复用异步方式的单次写动作
 - 5. 地址数据线复用同步方式的单次写动作

单次读动作

单次读动作的基本时序图和设定例如下：

表 31-4 单次读动作基本设定例

基本设定	MW	RSYN	WSYN	-	ADVS	BLSS	-表示不关注 <set>表示 用户设定值
	<set>	<set>	-	-	0b0	0b0	
时序设定	T_RC	T_WC	T_CEOE	T_WP	T_TR	T_ADV	
	0b0011	-	0b001	-	0b001	-	

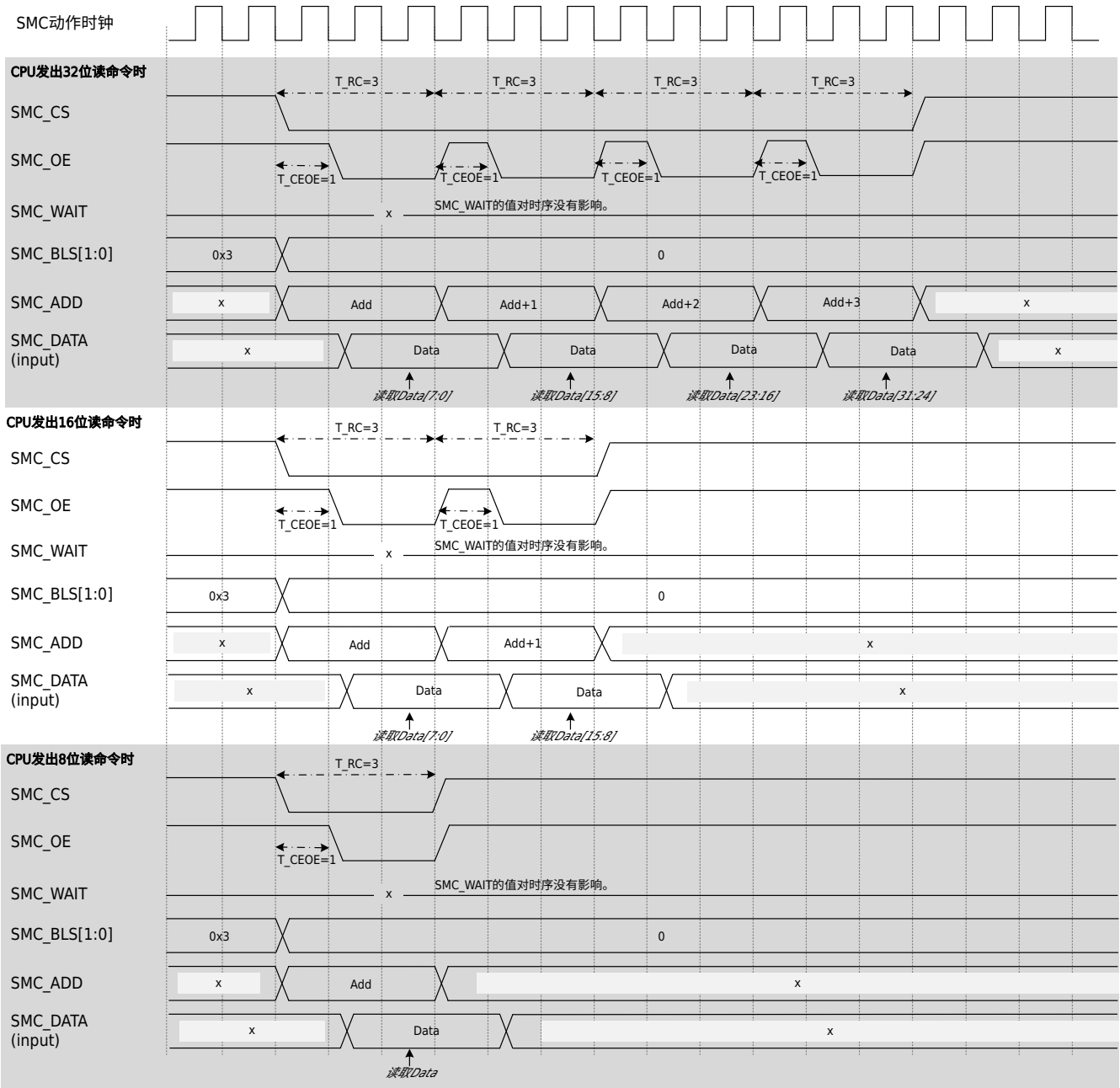


图 31-6 单次读动作基本时序（异步方式（RSYN=0b0）&8 位位宽（MW=0b00））

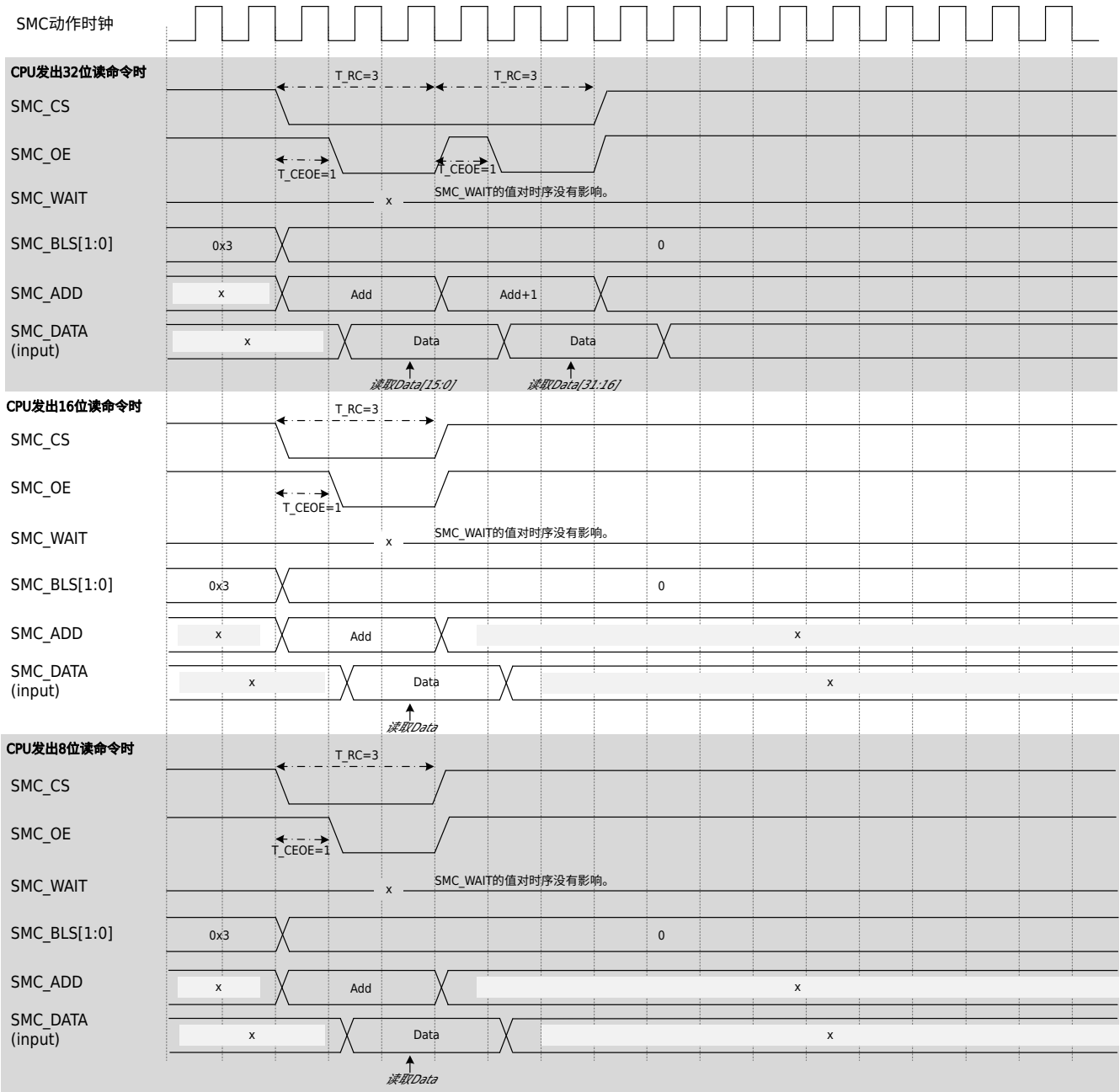


图 31-7 单次读动作基本时序（异步方式（RSYN=0b0）&16 位位宽（MW=0b01））

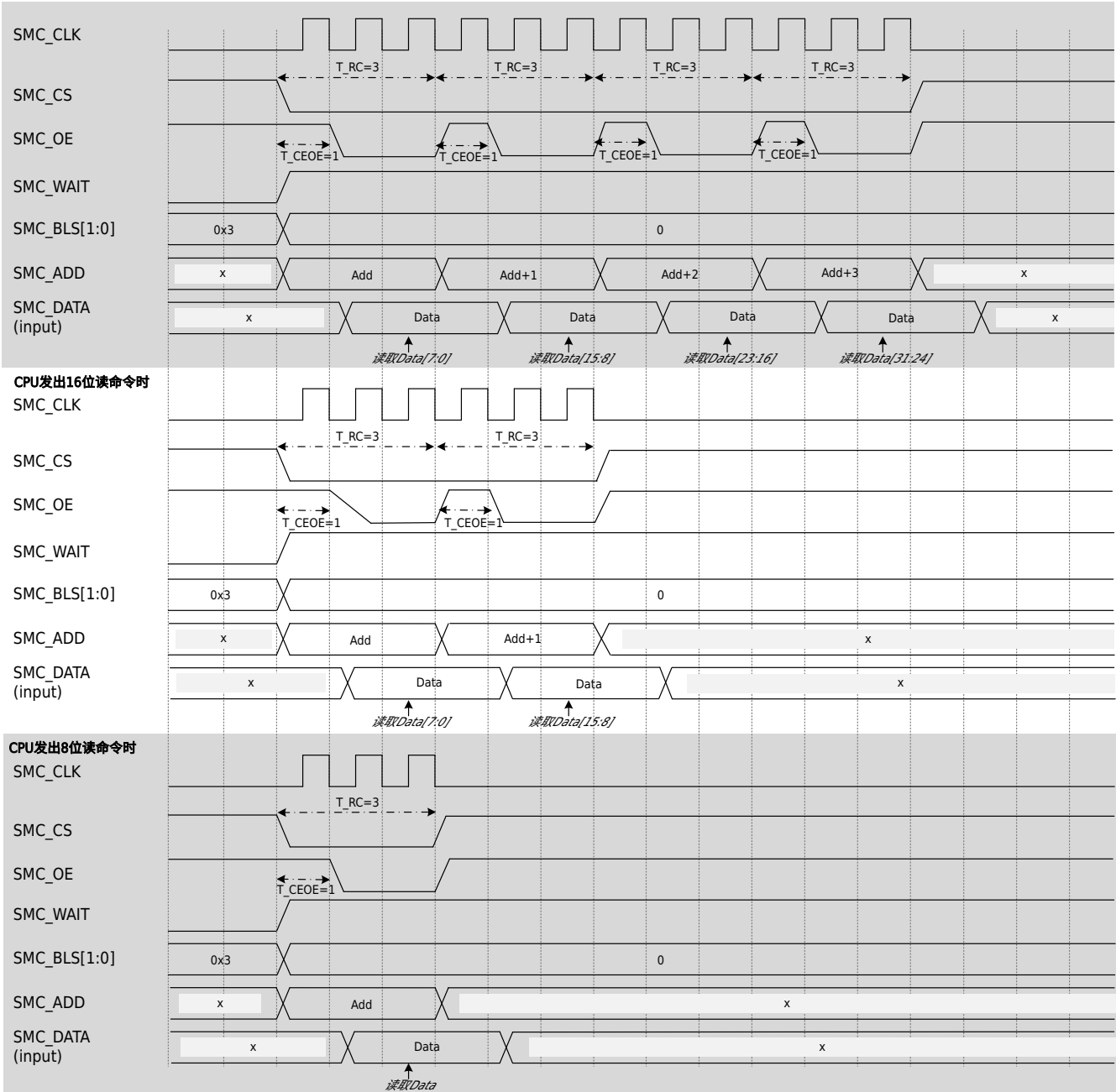


图 31-8 单次读动作基本时序（同步方式（RSYN=0b1）&8 位位宽（MW=0b00））

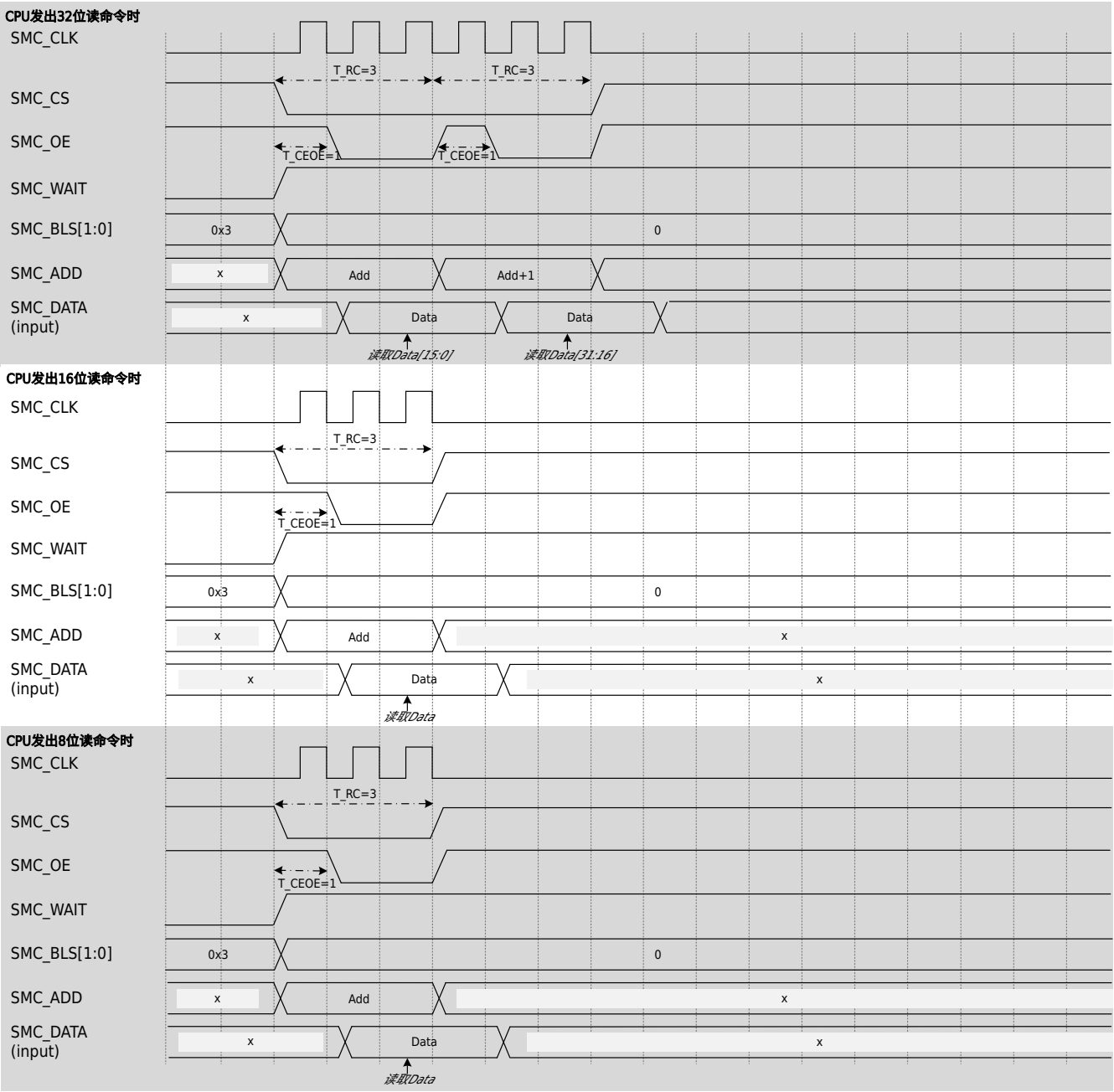


图 31-9 单次读动作基本时序（同步方式（RSYN=0b1）&16 位位宽（MW=0b01））

地址数据线复用的单次读动作

地址数据线复用的单次读动作的基本时序图和设定例如下：

表 31-5 地址数据线复用单次读动作基本设定例

基本设定	MW	RSYN	WSYN	-	ADVS	BLSS	-表示不关 注 <set>表示 用户设定值
	<set>	<set>	-	-	0b1	0b0	
时序设定	T_RC	T_WC	T_CEOE	T_WP	T_TR	T_ADV	
	0b0110	-	0b011	-	0b010	0b001	

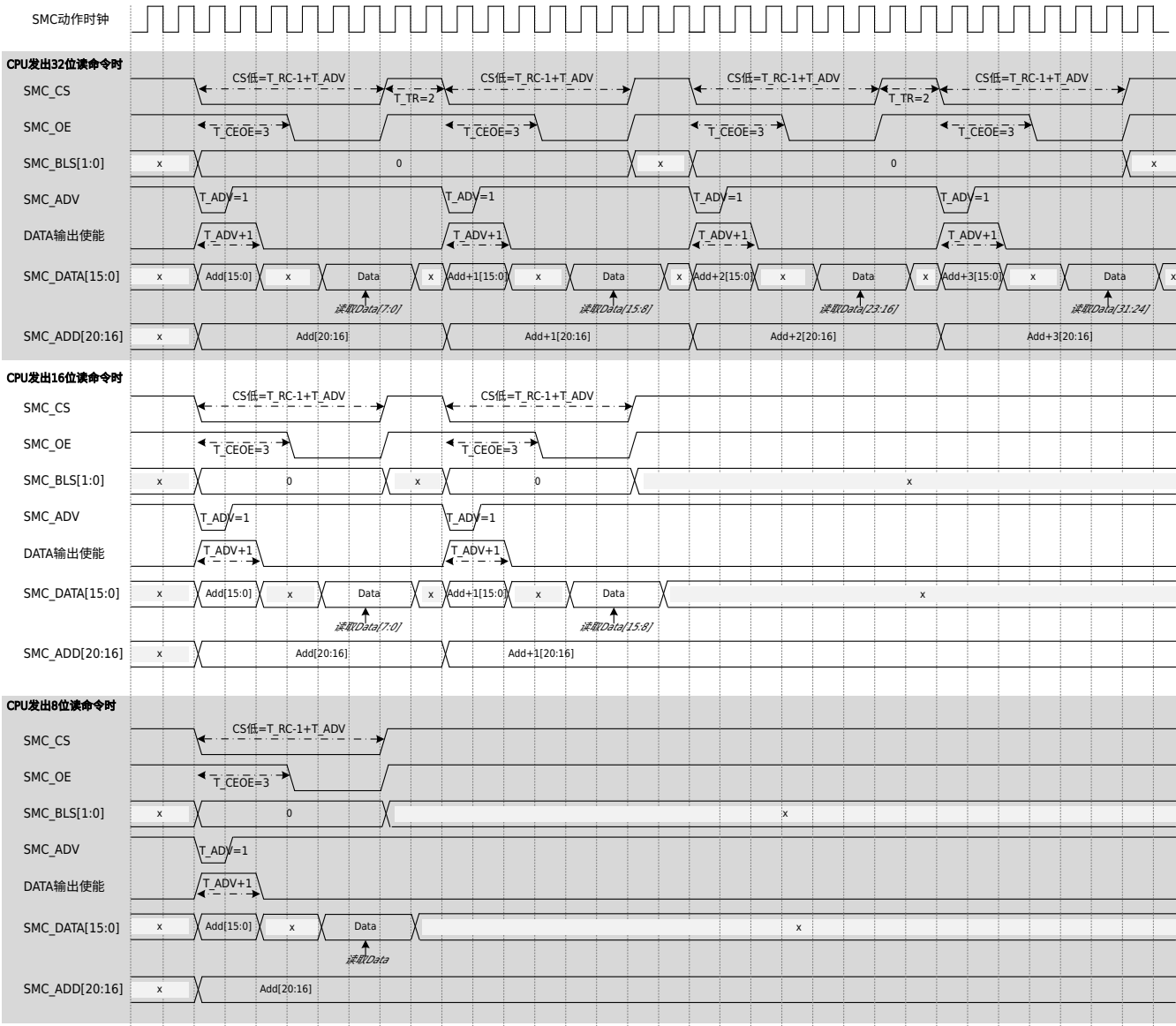


图 31-10 地址数据线复用单次读动作基本时序（8 位位宽（MW=0b00））

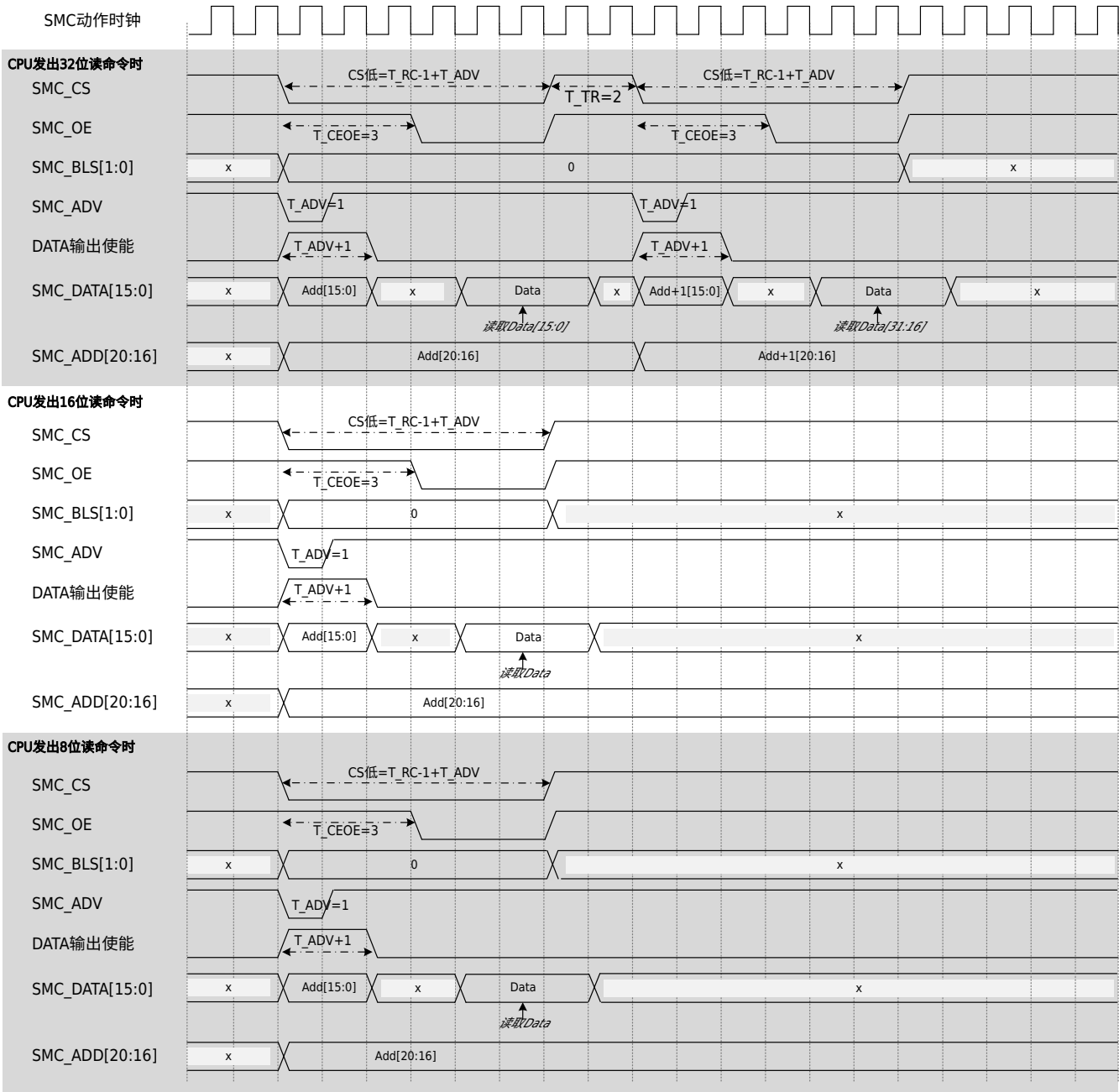


图 31-11 地址数据线复用单次读动作基本时序（16 位位宽（MW=0b01））

单次写动作

单次写动作的基本时序图和设定例如下：

表 31-6 单次写动作基本设定例

基本设定	MW	RSYN	WSYN	-	ADVS	BLSS	-表示不关注 <set>表示 用户设定值
	<set>	-	<set>	-	0b0	<set>	
时序设定	T_RC	T_WC	T_CEOE	T_WP	T_TR	T_ADV	
	-	0b0100	-	0b010	0b001	-	

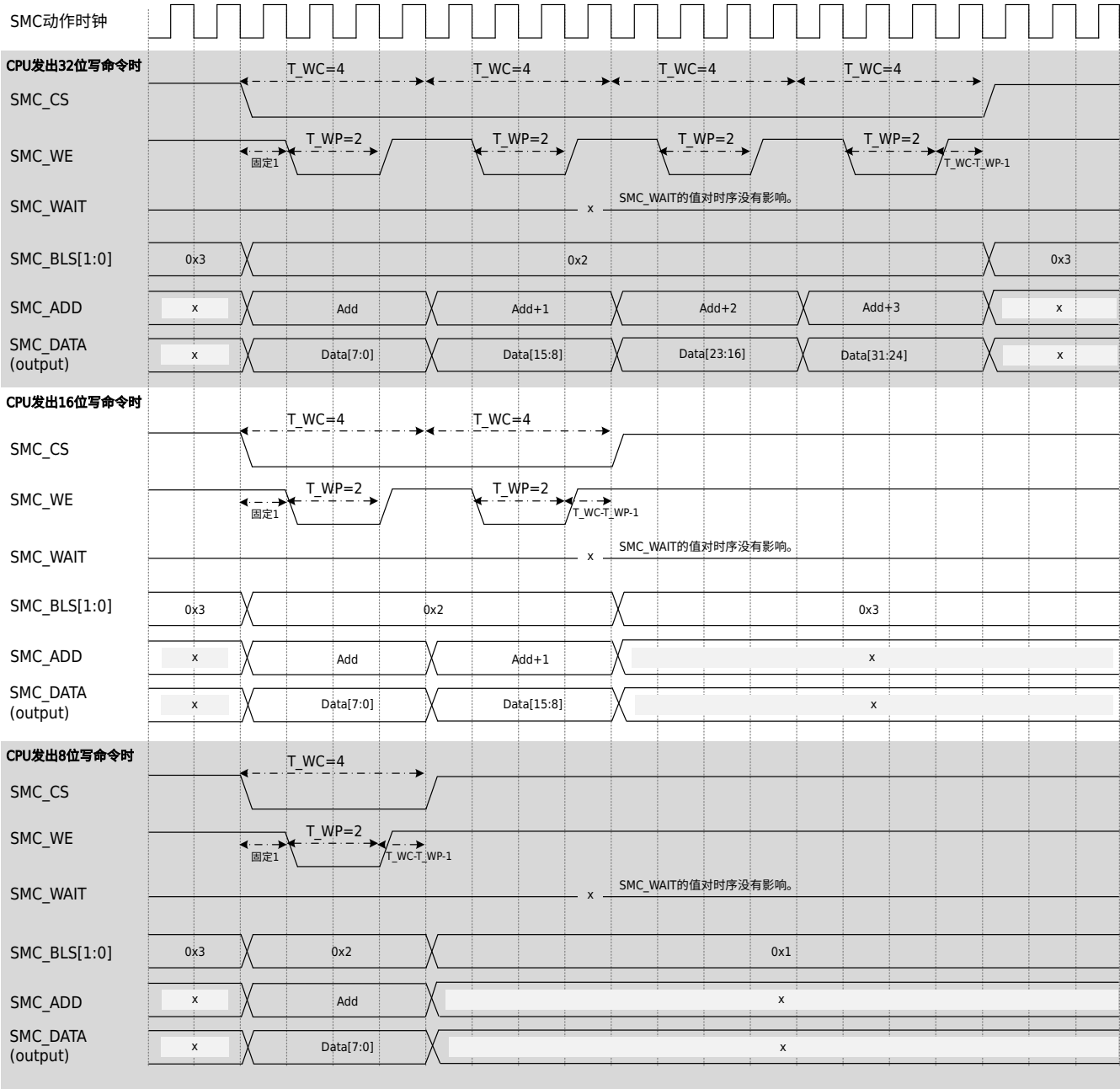


图 31-12 单次写动作基本时序（异步方式（WSYN=0b0）&8 位位宽（MW=0b00）&BLSS=0b0）

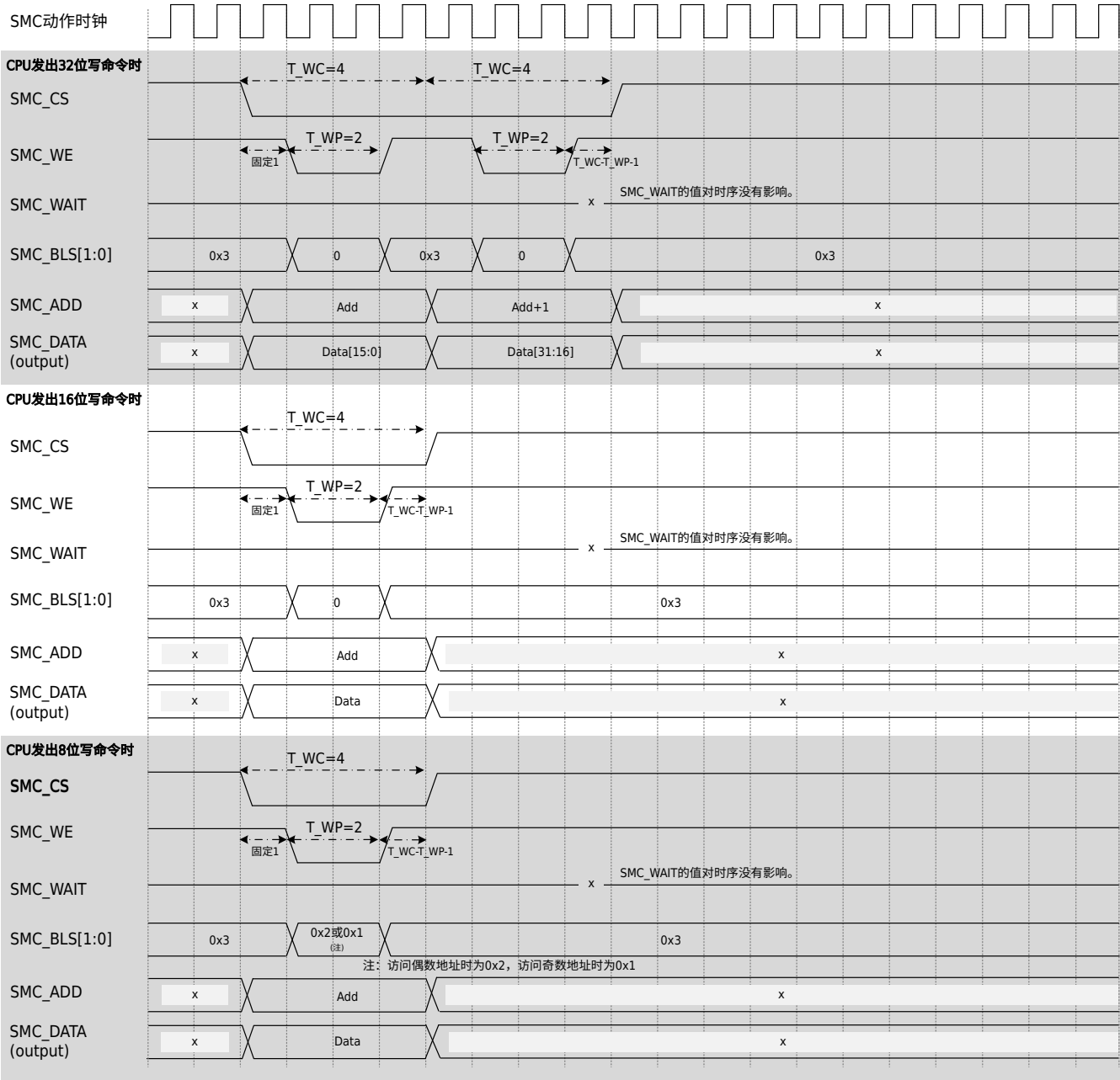


图 31-13 单次写动作基本时序（异步方式（WSYN=0b0）&16 位位宽（MW=0b01）&BLSS=0b1）

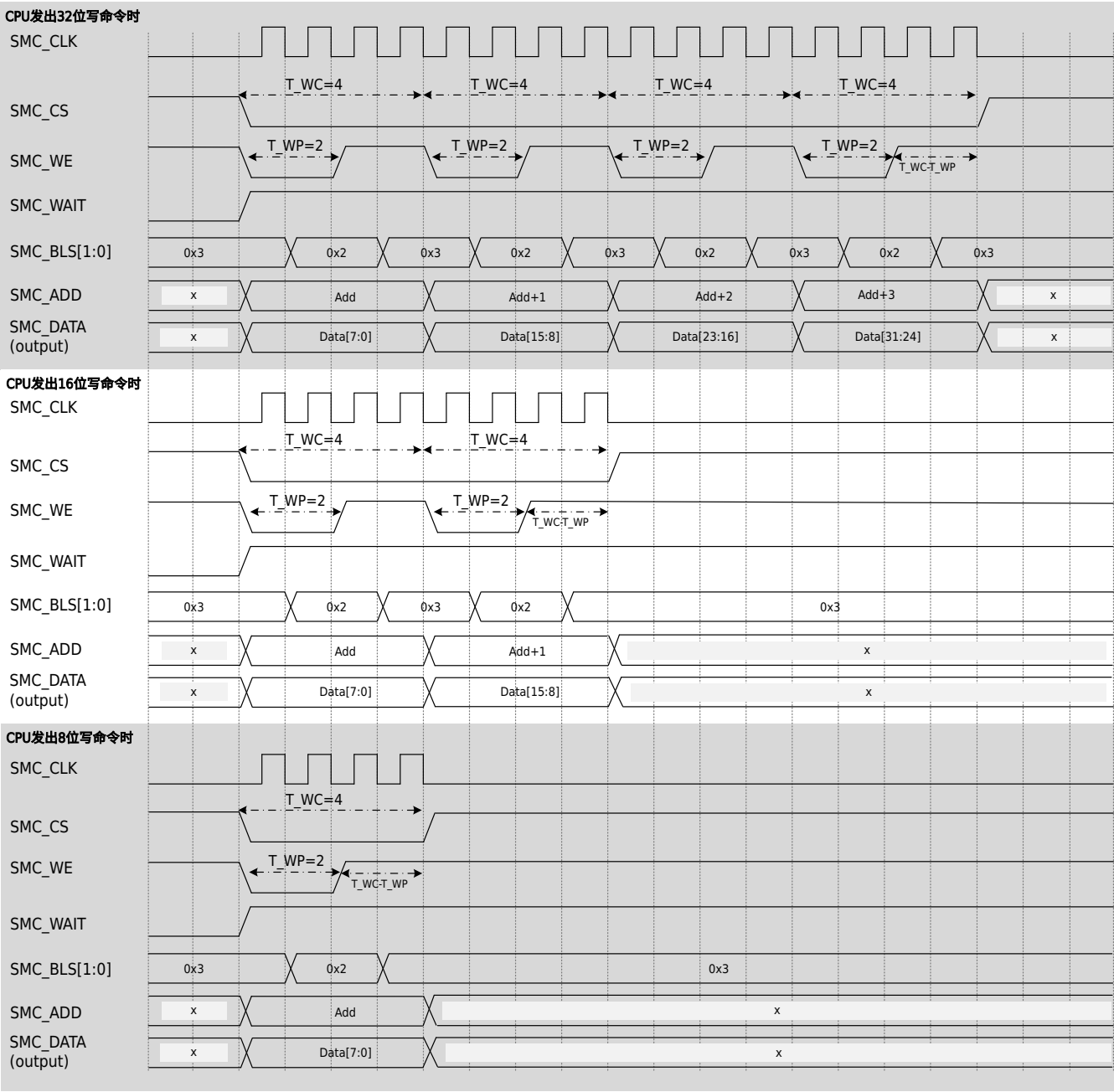


图 31-14 单次写动作基本时序 (同步方式 (WSYN=0b1) &8 位位宽 (MW=0b00) &BLSS=0b1)

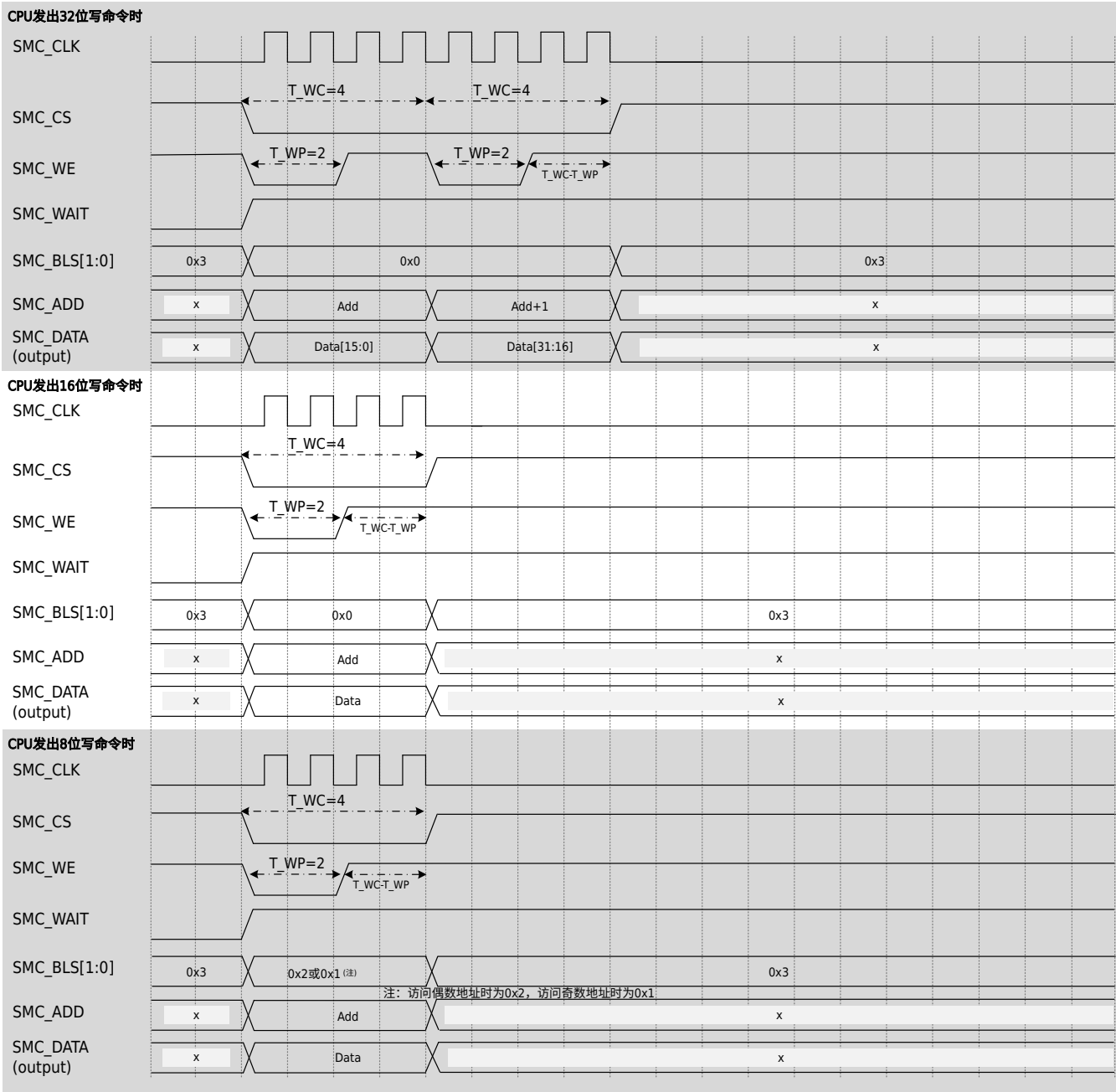


图 31-15 单次写动作基本时序 (同步方式 (WSYN=0b1) &16 位位宽 (MW=0b01) &BLSS=0b0)

地址数据线复用异步方式的单次写动作

地址数据线复用异步方式的单次写动作的基本时序图和设定例如下：

表 31-7 地址数据线复用异步方式单次写动作基本设定例

基本设定	MW	RSYN	WSYN	-	ADVS	BLSS	-表示不关注 <set>表示 用户设定值
	<set>	-	0	-	0b1	0b1	
时序设定	T_RC	T_WC	T_CEOE	T_WP	T_TR	T_ADV	
	-	0b0101	-	0b010	0b010	0b001	

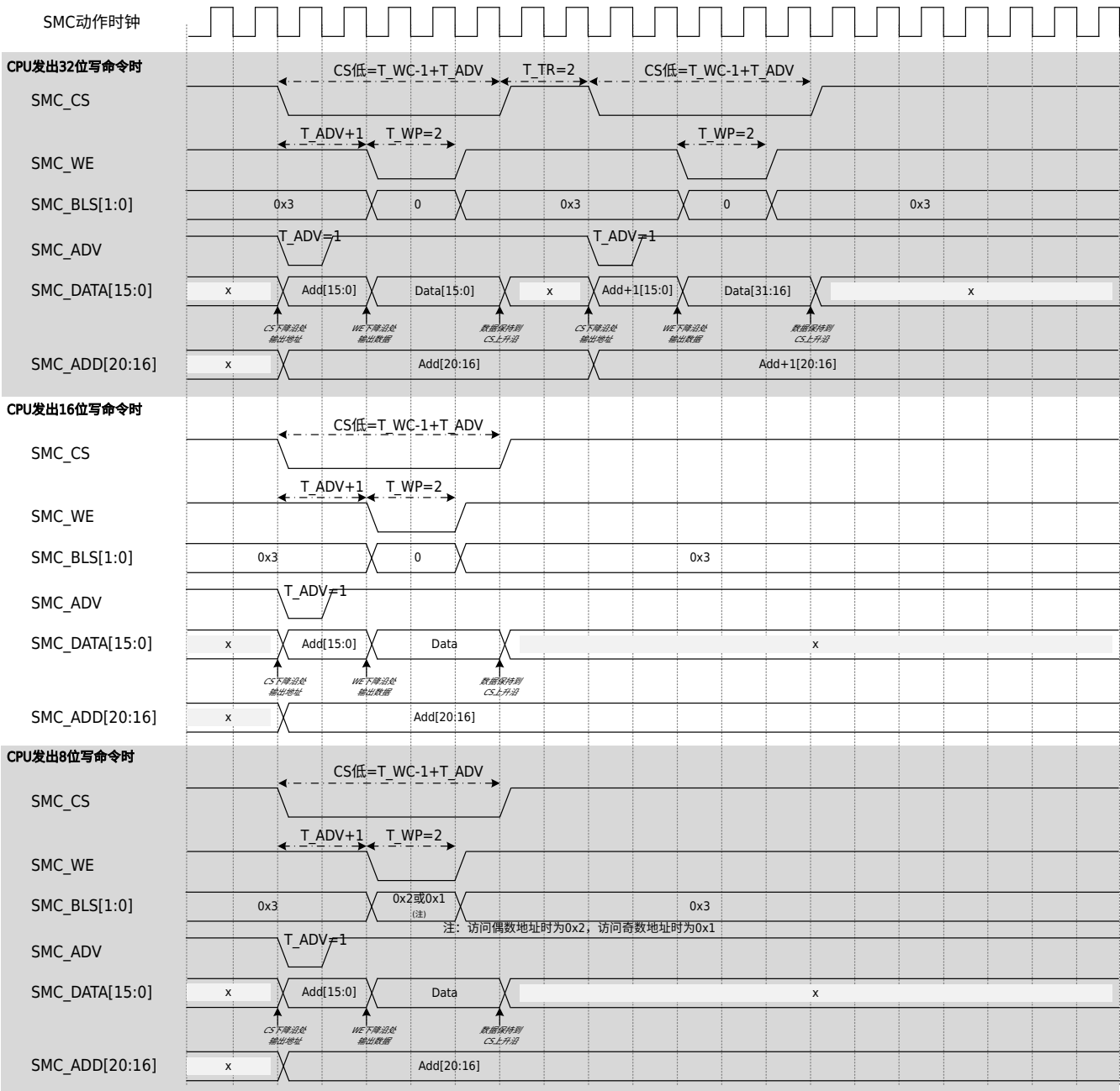


图 31-16 地址数据线复用单次写动作基本时序（异步方式（WSYN=0b0）&16 位位宽（MW=0b01））

地址数据线复用同步方式的单次写动作

地址数据线复用的同步方式的单次写动作的基本时序图和设定例如下：

表 31-8 地址数据线复用同步方式单次写动作基本设定例

基本设定	MW	RSYN	WSYN	-	ADVS	BLSS	-表示不关 注 <set>表示 用户设定值
	<set>	-	1	-	0b1	0b1	
时序设定	T_RC	T_WC	T_CEOE	T_WP	T_TR	T_ADV	
	-	0b0101	-	0b011	0b010	0b001	

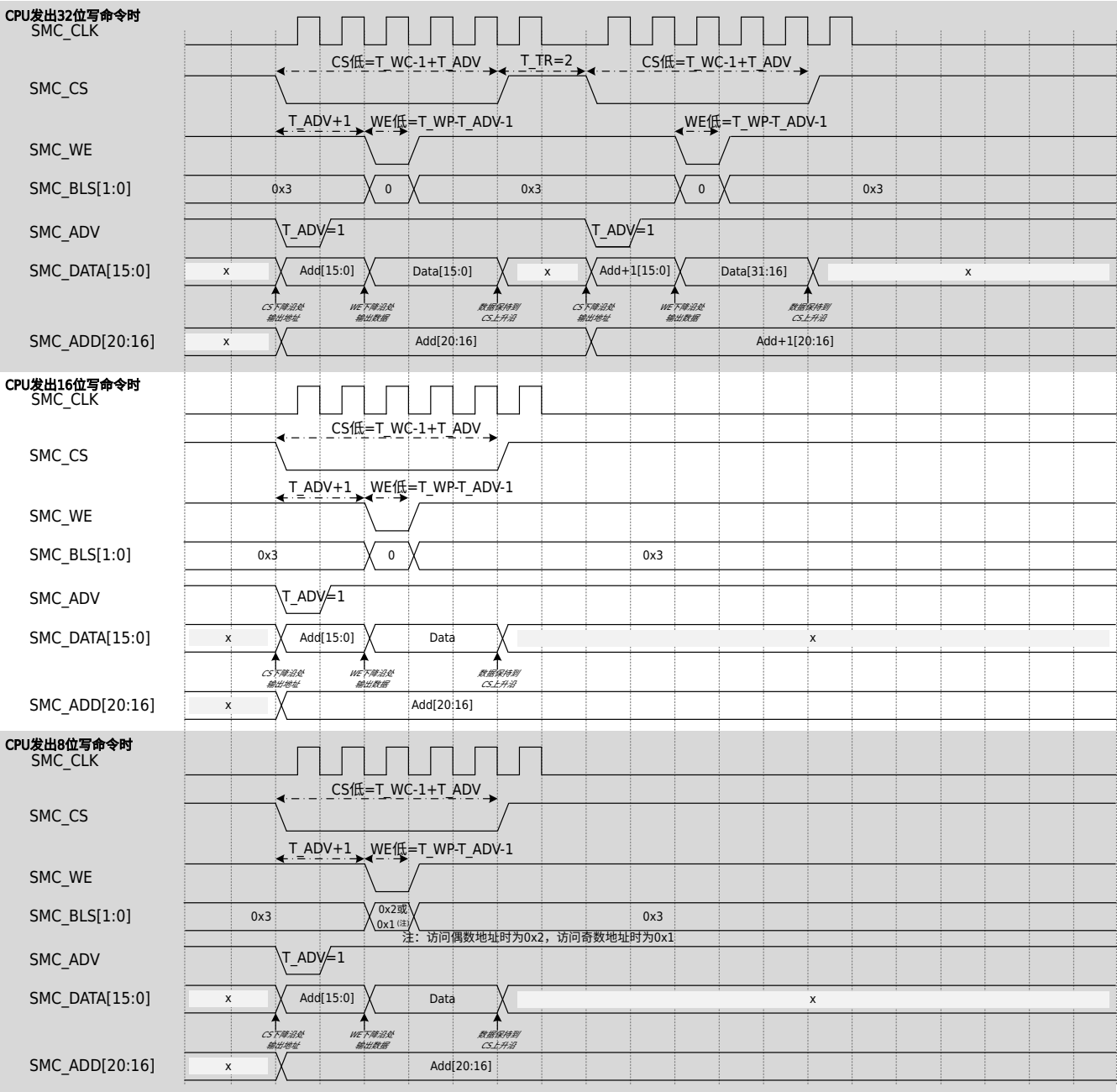


图 31-17 地址数据线复用单次写动作基本时序（同步方式（WSYN=0b1）&16 位位宽（MW=0b01））

31.3.5.4.3 FIFO 管理

SMC 中含有 3 类 FIFO。分别是命令 FIFO、写数据 FIFO、读数据 FIFO。

- 命令 FIFO：2 个命令 FIFO，用于缓存 2 次 SMC 命令及对应的命令地址、CS 等信息。

- 写数据 FIFO：4 个写数据 FIFO，用于缓存 AHB 的写数据。
- 读数据 FIFO：4 个读数据 FIFO，用于缓存从 SRAM/PSRAM/NOR Flash 读入的数据。

31.3.5.5 SMC 低功耗管理

SMC 处于 Low Power 状态时，如果将状态控制寄存器 (SMC_STCR1) 的 LPWOR 位设定为 1，则 SMC 从 Low Power 状态转为 Ready 状态，SMC 开始正常工作；如果对 SMC 长时间无操作，可将状态控制寄存器 (SMC_STCR0) 的 LPWIR 位设定为 1，使 SMC 进入 Low Power 状态，进入 Low Power 状态后，SMC 的内部动作停止，功耗会降低。

31.4 寄存器

31.4.1 寄存器总表

EXMC 公共寄存器基地址: 0x40055400

表 31-9 EXMC 公共寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x0C	SMC_ENAR	SMC 使能寄存器	32

SMC 基地址: 0x88000000

表 31-10 外部存储器控制器 (EXMC) SMC 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x0000	SMC_STSR	状态寄存器	32
0x0010	SMC_CMDR	命令寄存器	32
0x0008	SMC_STCR0	状态控制寄存器 0	32
0x000C	SMC_STCR1	状态控制寄存器 1	32
0x0020	SMC_RFTR	刷新时间寄存器	32
0x0200	SMC_BACR	基本控制寄存器	32
0x0208	SMC_CSCR0	片选控制寄存器	32
0x020C	SMC_CSCR1	片选控制寄存器	32
0x0018	SMC_CPCR	CHIP 配置寄存器	32
0x0104	SMC_CPSR0	CHIP 状态寄存器 0	32
0x0014	SMC_TMCR	时序配置寄存器	32
0x0100	SMC_TMSR0	时序状态寄存器 0	32

31.4.2 EXMC 公共寄存器

31.4.2.1 SMC 使能寄存器 (SMC_ENAR)

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																												SMCEN	Reserved		
Access																													RW			

位/位域	标记	位名	功能描述	读写
31:2	Reserved	保留	-	-
1	SMCEN	SMC 使能	0b0: SMC 模块的功能端口无效 0b1: SMC 模块的功能端口有效	RW
0	Reserved	保留	-	-

31.4.3 SMC-SRAM/PSRAM/NOR Flash 寄存器

31.4.3.1 状态寄存器 (SMC_STSR)

Offset	Bit Position																															
0x0000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000001																															
Name	Reserved																															RO STATUS
Access																																RO

位/位域	标记	位名	功能描述	读写
31:1	Reserved	保留	-	-
0	STATUS	SMC 当前状态	0b0: Ready 0b1: Low Power	RO

31.4.3.2 命令寄存器 (SMC_CMDR)

Offset	Bit Position																																			
0x0010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reset	0xFFFFFFFF																																			
Name	Reserved						CMDCHIP				CMD				CRES				CMDADD																	
Access							WO						WO				WO																			

位/位域	标记	位名	功能描述	读写
31:26	Reserved	保留	-	-
25:23	CMDCHIP	CS 对象设定	0b000: Chip 0 - 其他: 禁止设置	WO
22:21	CMD	命令输入	0b00: 禁止设置 0b01: MdRegConfig 0b10: UpdateRegs 0b11: MdRegConfig & UpdateRegs	WO
20	CRES	CRE 端口设定	0b0: SMC_CRE 输出低电平 0b1: 在发出 MdRegConfig 命令时, SMC_CRE 端口输出高电平	WO
19:0	CMDADD	地址对象设定	当 CMD=0b01 或 0b11 时, CMDADD[19:0]表示要访问的外部 MEM 地址	WO

31.4.3.3 状态控制寄存器 0 (SMC_STCR0)

Offset	Bit Position																																
0x0008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0xFFFFFFFF																																
Name	Reserved																														LPWIR		Reserved
Access																																	

位/位域	标记	位名	功能描述	读写
31:3	Reserved	保留	-	-
2	LPWIR	Low Power 状态 进入请求	写 1 进入 Low Power 状态；写 0 无效	WO
1:0	Reserved	保留	-	-

31.4.3.4 状态控制寄存器 1 (SMC_STCR1)

Offset	Bit Position																																	
0x000C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0xFFFFFFFF																																	
Name	Reserved																														LPWOR		Reserved	
Access																																		

位/位域	标记	位名	功能描述	读写
31:3	Reserved	保留	-	-
2	LPWOR	Low Power 状态 退出请求	写 1 退出 Low Power 状态；写 0 无效	WO
1:0	Reserved	保留	-	-

31.4.3.5 刷新时间寄存器 (SMC_RFTR)

Offset	Bit Position																															
0x0020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																										REFPRD					
Access																											RW					

位/位域	标记	位名	功能描述	读写
31:4	Reserved	保留	-	-
3:0	REFPRD	刷新周期	刷新周期值设定	RW

31.4.3.6 基本控制寄存器 (SMC_BACR)

Offset	Bit Position																																	
0x0200	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000300																																	
Name	Reserved																CKSEL	Reserved								MUXMD	Reserved							
Access																	RW									RW								

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:14	CKSEL	采样时钟选择	0b00: 读取锁存 SMC_DATA、SMC_WAIT 端口数据时, 使用内部 EXCLK 0b01: 读取锁存 SMC_DATA、SMC_WAIT 端口数据时, 使用内部 EXCLK 取反做时钟 0b10: 读取锁存 SMC_DATA、SMC_WAIT 端口数据时, 使用 EXMC_CLK 端口状态反馈做时钟 (仅用于同步模式) 0b11: 禁止设置	RW
13:10	Reserved	保留	-	-
9:8	Reserved	保留	所有位读出的 1, 写入时必须写 1	-
7:5	Reserved	保留	-	-
4	MUXMD	SMC 地址数据复用	0b0: SMC 地址和数据不复用 0b1: SMC 地址和数据复用	RW
3:0	Reserved	保留	-	-

31.4.3.7 片选控制寄存器 (SMC_CSCR0)

Offset	Bit Position																															
0x0208	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved																								ADDMSK0							
Access																									RW							

位/位域	标记	位名	功能描述	读写
31:8	Reserved	保留	所有位读出的 1, 写入时必须写 1	-
7:0	ADDMSK0	CS0 空间屏蔽地址	CS0 空间屏蔽地址设定	RW

31.4.3.8 片选控制寄存器 (SMC_CSCR1)

Offset	Bit Position																															
0x020C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								ADDMAT0							
Access																									RW							

位/位域	标记	位名	功能描述	读写
31:8	Reserved	保留	-	-
7:0	ADDMAT0	CS0 空间匹配地址	CS0 空间匹配地址设定	RW

举例说明，不同 ADDMAT0 和 ADDMSK0 设定值时，SMC 访问空间的对应方式如下表所示。

表 31-29 CS0 设定和访问地址对应关系表

ADDMAT0[7:0]	ADDMSK0[7:0]	对应外部地址	CS0 空间大小
0x70	0xFF	0x70000000~0x70FFFFFF	16MB
	0xFE	0x70000000~0x71FFFFFF	32MB
	0xFC	0x70000000~0x73FFFFFF	64MB
	0xF8	0x70000000~0x77FFFFFF	128MB
	0xF0	0x70000000~0x7FFFFFFF	256MB
	其它设定禁止	-	-
0x68	0xFF	0x68000000~0x68FFFFFF	16MB
	0xFE	0x68000000~0x69FFFFFF	32MB
	0xFC	0x68000000~0x6BFFFFFF	64MB
	0xF8	0x68000000~0x6FFFFFFF	128MB
	其它设定禁止	-	-
0x66	0xFF	0x66000000~0x66FFFFFF	16MB
	0xFE	0x66000000~0x67FFFFFF	32MB
	其它设定禁止	-	-
0x64	0xFF	0x64000000~0x64FFFFFF	16MB
	0xFE	0x64000000~0x65FFFFFF	32MB
	0xFC	0x64000000~0x67FFFFFF	64MB
	其它设定禁止	-	-
0x62	0xFF	0x62000000~0x62FFFFFF	16MB
	0xFE	0x62000000~0x63FFFFFF	32MB
	其它设定禁止	-	-

ADDMAT0[7:0]	ADDMSK0[7:0]	对应外部地址	CS0 空间大小
0x60	0xFF	0x60000000~0x60FFFFFF	16MB
	0xFE	0x60000000~0x61FFFFFF	32MB
	0xFC	0x60000000~0x63FFFFFF	64MB
	0xF8	0x60000000~0x67FFFFFF	128MB
	0xF0	0x60000000~0x6FFFFFFF	256MB
	0xE0	0x60000000~0x7FFFFFFF	512MB
	其它设定禁止	-	-

31.4.3.9 CHIP 配置寄存器 (SMC_CPCR)

Offset	Bit Position																															
0x0018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved																BLSS		ADVS		Reserved		MW		Reserved		WSYN		Reserved		RSYN	
Access																	WO		WO				WO						WO			

位/位域	标记	位名	功能描述	读写
31:13	Reserved	保留	-	-
12	BLSS	BLS 端口设定	0b0: BLS 的端口状态和 CS 端口的状态变化同步 0b1: BLS 的端口状态和 WE 端口的状态变化同步	WO
11	ADVS	ADV 端口设定	0b0: ADV 端口功能无效 0b1: 地址数据线复用功能有效时, ADV 端口功能有效	WO
10	Reserved	保留	-	-
9:8	MW	存储器位宽选择	0b00: 8 位位宽 0b01: 16 位位宽 - 其他: 禁止设置	WO
7:5	Reserved	保留	-	-
4	WSYN	写同步	0b0: 异步写使能 0b1: 同步写使能	WO
3:1	Reserved	保留	-	-
0	RSYN	读同步	0b0: 异步读使能 0b1: 同步读使能	WO

31.4.3.10 CHIP 状态寄存器 0 (SMC_CPSR0)

Offset	Bit Position																																	
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00FF0A00																																	
Name	ADDMAT								ADDMSK								Reserved		BLSS		ADVS		Reserved		MW		Reserved		WSYN		Reserved		RSYN	
Access	RO								RO										RO		RO				RO				RO				RO	

位/位域	标记	位名	功能描述	读写
31:24	ADDMAT	CS 空间匹配地址	CS 空间匹配地址信息	RO
23:16	ADDMSK	CS 空间屏蔽地址	CS 空间屏蔽地址信息	RO
15:13	Reserved	保留	-	-
12	BLSS	SMC_BLS 状态	0b0: BLS 的端口状态和 CS 端口的状态变化同步 0b1: BLS 的端口状态和 WE 端口的状态变化同步	RO
11	ADVS	SMC_ADV 状态	0b0: ADV 端口功能无效 0b1: 地址数据线复用功能有效时, ADV 端口功能有效	RO
10	Reserved	保留	-	-
9:8	MW	存储器位宽状态	0b00: 8 位位宽 0b01: 16 位位宽 - 其他: Reserved	RO
7:5	Reserved	保留	-	-
4	WSYN	写动作方式状态	0b0: 异步写使能 0b1: 同步写使能	RO
3:1	Reserved	保留	-	-
0	RSYN	读动作方式状态	0b0: 异步读使能 0b1: 同步读使能	RO

31.4.3.11 时序配置寄存器 (SMC_TMCR)

Offset	Bit Position																																	
0x0014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0xFFFFFFFF																																	
Name	Reserved				T_ADV				Reserved	T_TR				Reserved				T_WP				Reserved	T_CEOE				T_WC				T_RC			
Access					WO					WO								WO					WO				WO				WO			

位/位域	标记	位名	功能描述	读写
31:27	Reserved	保留	-	-
26:24	T_ADV	T_ADV 周期设置	T_ADV 设定，设置值需要大于等于 1  说明 仅在地址数据复用时有效。	WO
23	Reserved	保留	-	-
22:20	T_TR	T_TR 周期设定	T_TR 设定  说明 仅在地址数据复用时有效，有效值为 1~7；地址数据不复用时写 1。	WO
19:15	Reserved	保留	-	-
14:12	T_WP	T_WP 周期设定	T_WP 设定	WO
11	Reserved	保留	-	-
10:8	T_CEOE	T_CEOE 周期设定	T_CEOE 设定，设定值需要大于 T_ADV	WO
7:4	T_WC	T_WC 周期设定	T_WC 设定	WO
3:0	T_RC	T_RC 周期设定	T_RC 设定	WO

31.4.3.12 时序状态寄存器 0 (SMC_TMSR0)

Offset	Bit Position																																	
0x0100	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x011263CC																																	
Name	Reserved				T_ADV				Reserved	T_TR				Reserved				T_WP				Reserved	T_CEOE				T_WC				T_RC			
Access					RO					RO								RO					RO				RO				RO			

位/位域	标记	位名	功能描述	读写
31:27	Reserved	保留	-	-
26:24	T_ADV	T_ADV 周期状态	T_ADV 周期	RO
23	Reserved	保留	-	-
22:20	T_TR	T_TR 周期状态	T_TR 周期	RO
19:15	Reserved	保留	-	-
14:12	T_WP	T_WP 周期状态	T_WP 周期	RO
11	Reserved	保留	-	-
10:8	T_CEOE	T_CEOE 周期状态	T_CEOE 周期	RO
7:4	T_WC	T_WC 周期状态	T_WC 周期	RO
3:0	T_RC	T_RC 周期状态	T_RC 周期	RO

31.5 注意事项

- 1. SMC 的相关寄存器只能在 Ready 状态下设定。
- 2. 在使用 SMC 模块对外部 Memory 进行读访问时，Memory Device 在接收到控制器发出的读命令后将数据传送给控制器，控制器对传送回的数据进行采样的时钟可以通过 SMC_BACR.CKSEL 进行选择，具体设定请参考寄存器章节。选择不同的采样时钟，可以改善通信路径的时间裕量。

下图是同步模式下的 SMC 访问例。

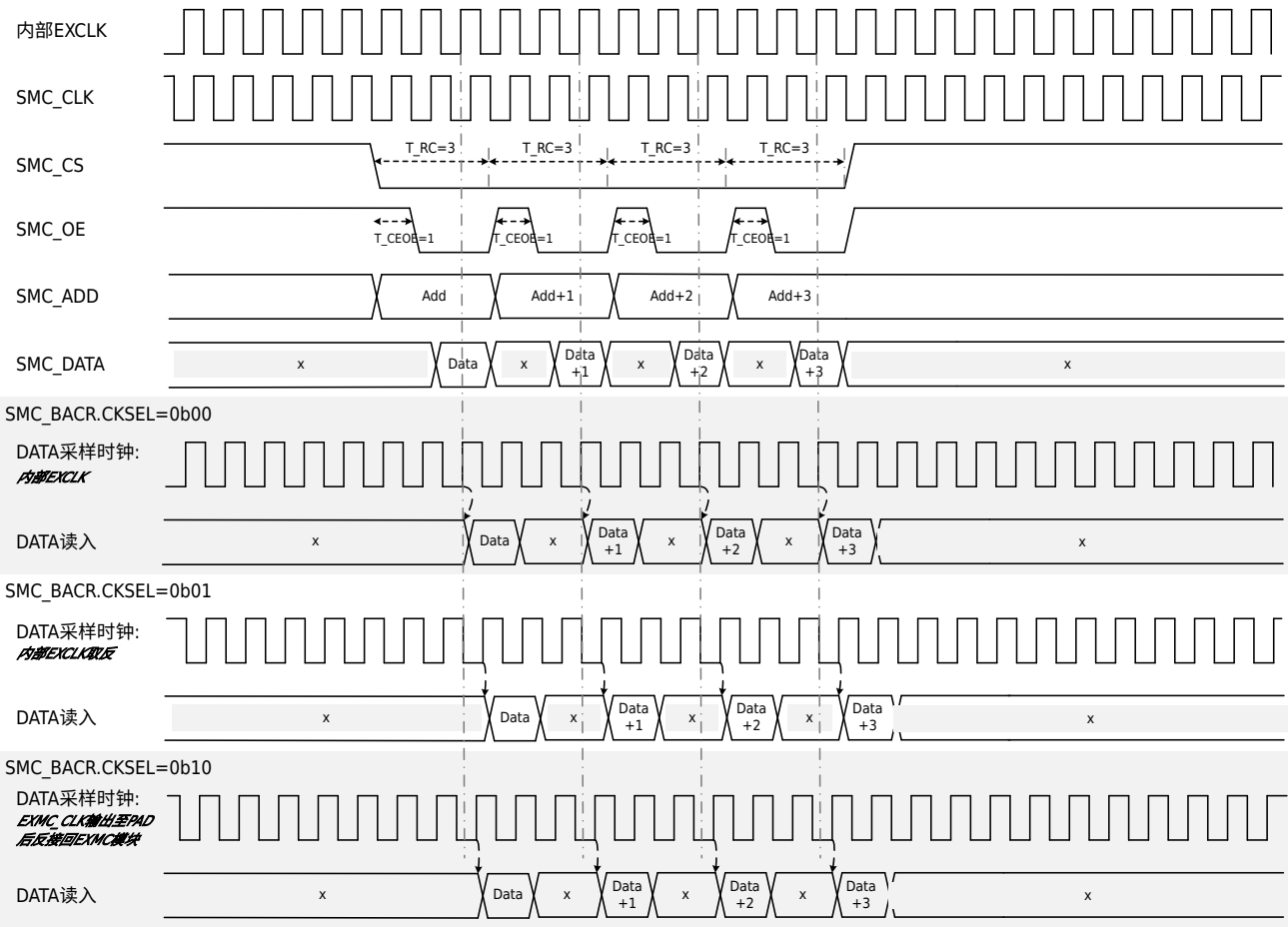


图 31-18 外部反馈时钟访问

32 控制器局域网 (MCAN)

32.1 简介

控制器局域网络 (CAN) 子系统由两个 CAN 模块、一个共享消息 RAM 存储器组成。关于这几个组成部分的基地址, 请参见《存储器映射 (Memory Mapping)》章节。

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。

32.2 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 APB 总线接口时钟

32.3 功能说明

32.3.1 MCAN 系统框图

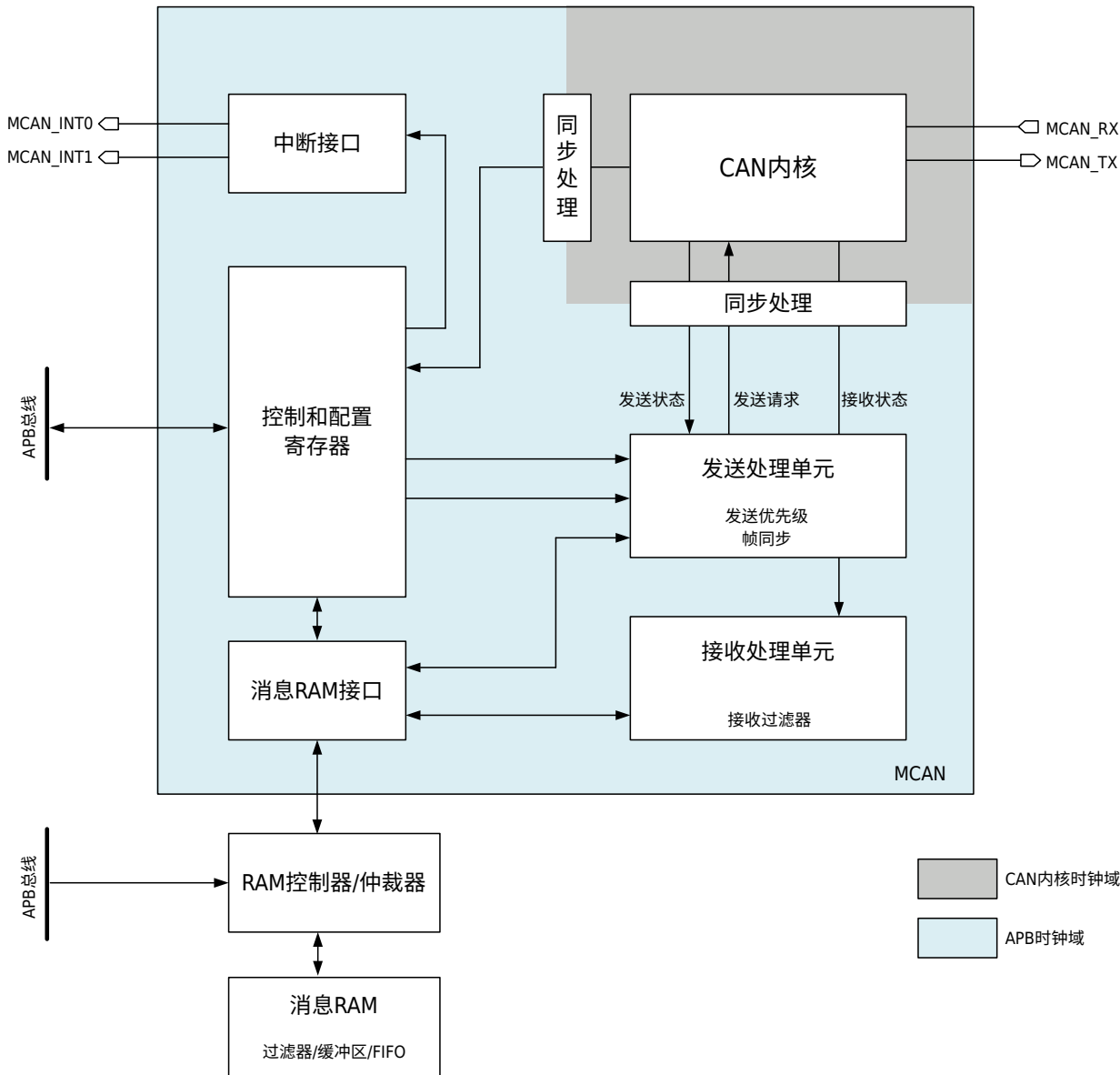


图 32-1 MCAN 系统框图

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 APB 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 APB 时钟域。

APB 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，请参见《时钟控制器（CMU）》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。

APB 接口

将 MCAN 连接至 APB 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。

32.3.2 引脚说明

表 32-1 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

32.3.3 工作模式

32.3.3.1 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 配置寄存器如下表所示。

表 32-2 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

寄存器	说明
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_TDCR	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
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。

32.3.3.2 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。

32.3.3.3 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，则只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 CAN FD 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，如下表所示。

表 32-3 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。

32.3.3.4 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。

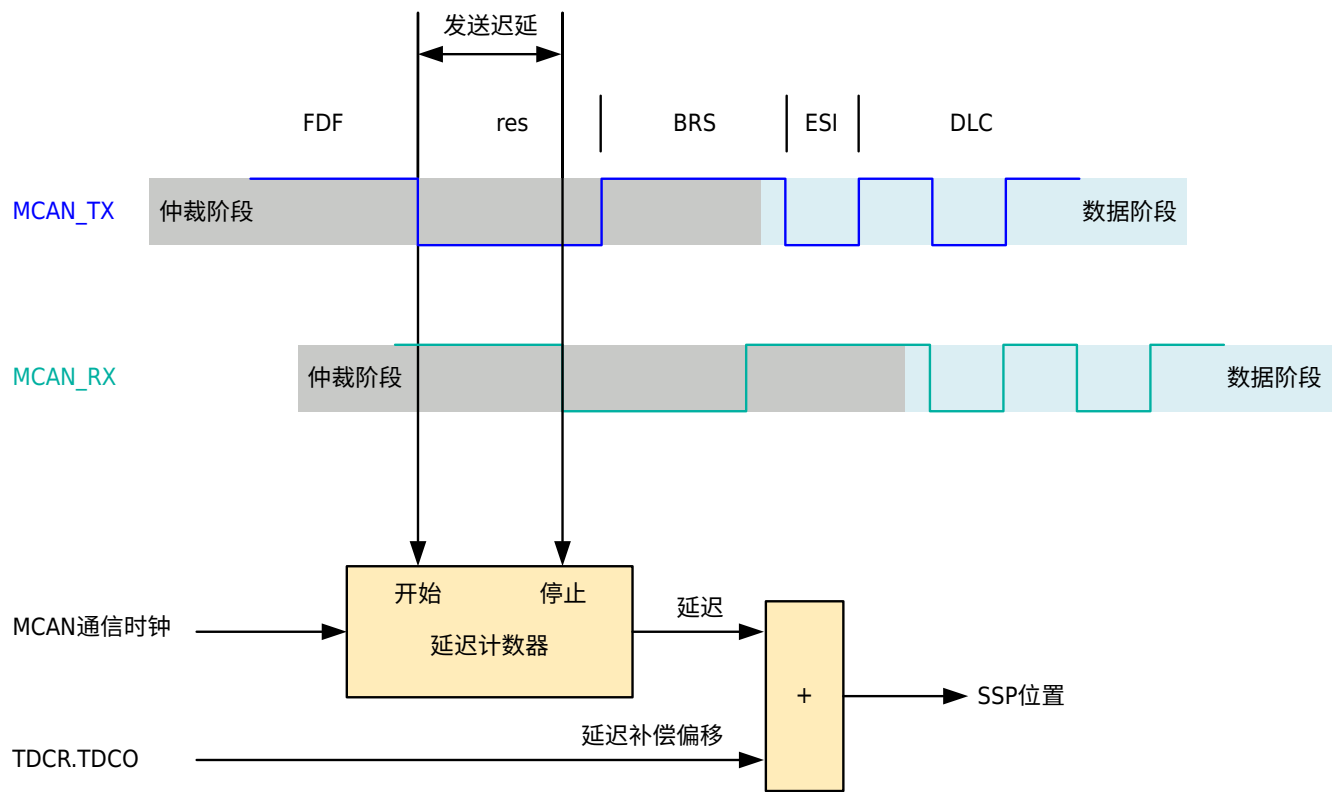


图 32-2 收发器延迟测量

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。

说明
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。

32.3.3.5 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。

说明
受限工作模式不得与回环模式（内部或外部）结合使用。

32.3.3.6 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。下图显示了总线监听模式下 MCAN_TX 和 MCAN_RX 信号与 MCAN 的连接。

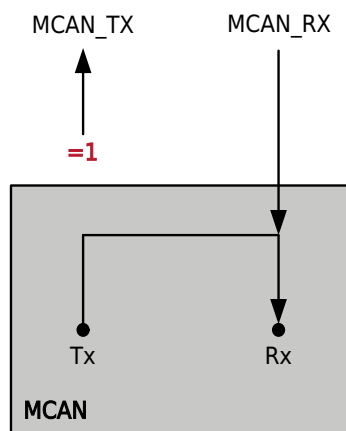


图 32-3 总线监听模式下的引脚控制

32.3.3.7 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）。

32.3.3.8 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。

**说明**

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。

32.3.3.9 测试模式

要使能对 MCAN 测试寄存器 MCAN_TEST (请参见“[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。

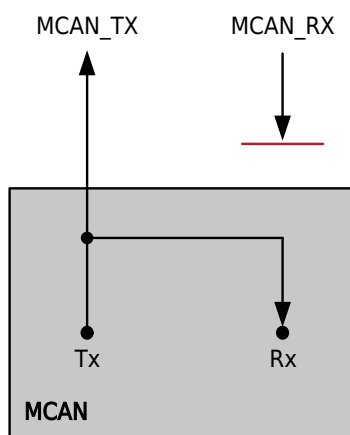
由于 CAN 通信时钟与控制逻辑时钟之间的同步机制, 从配置 MCAN_TEST.TX 到引脚 MCAN_TX 实现指定功能之间, 可能有几个控制逻辑时钟周期的延迟。这一点还适用于通过 MCAN_TEST.RX 读取引脚 MCAN_RX 的情况。

**说明**

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。

32.3.3.10 回环模式**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。下图显示了外部回环模式下 MCAN_TX 和 MCAN_RX 引脚信号与 MCAN 的连接。



外部回环模式

图 32-4 外部回环模式的引脚控制

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。

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。下图显示了内部回环模式下 MCAN_TX 和 MCAN_RX 引脚信号与 MCAN 的连接。

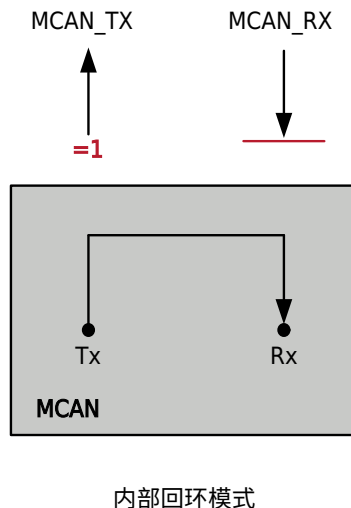


图 32-5 内部回环模式的引脚控制

32.3.4 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。

32.3.5 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。



说明

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。

32.3.6 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。

32.3.6.1 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。



说明

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。

32.3.6.2 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）

32.3.6.3 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。

32.3.6.4 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。

32.3.6.5 标准消息 ID 过滤

下图介绍了标准消息 ID（11 位 ID）的过滤流程。“标准消息 ID 过滤器元素”介绍了标准消息 ID 过滤器元素。

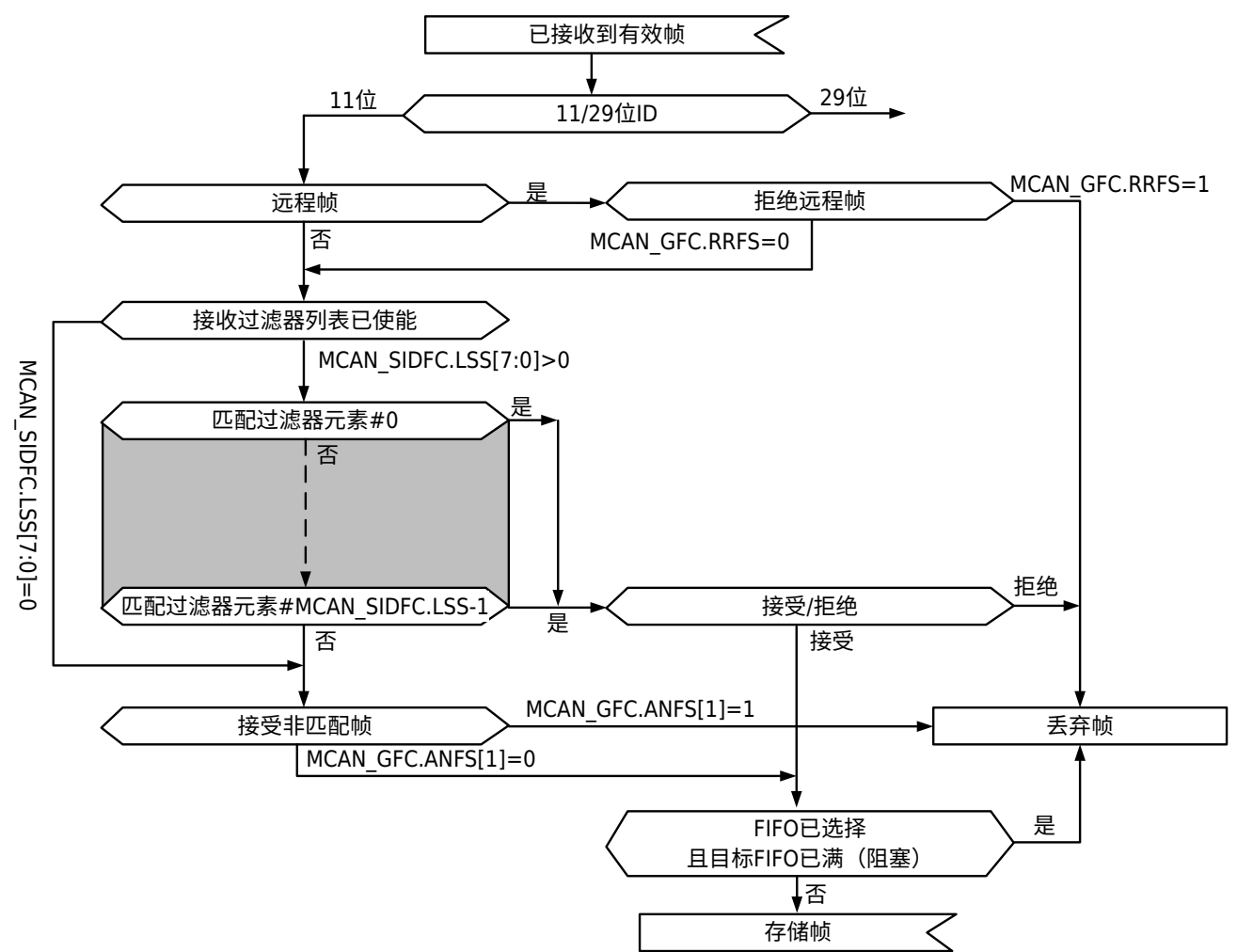


图 32-6 标准消息 ID 过滤器路径

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。

32.3.6.6 扩展消息 ID 过滤

下图介绍了扩展消息 ID (29 位 ID) 的过滤流程。“[扩展消息 ID 过滤器元素](#)”介绍了扩展消息 ID 过滤器元素。

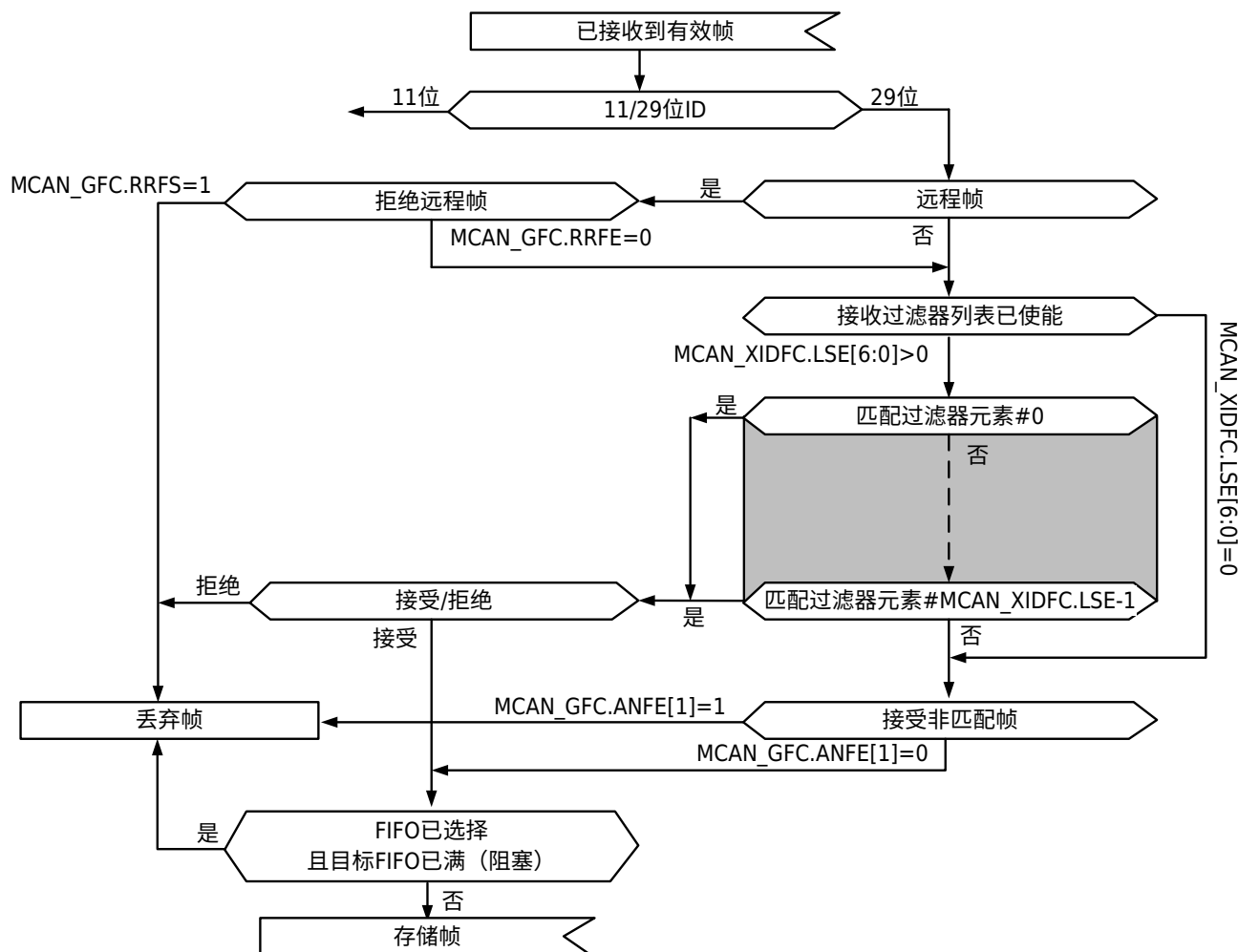


图 32-7 扩展消息 ID 过滤器路径

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。

32.3.6.7 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“[接收过滤](#)”。“[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。

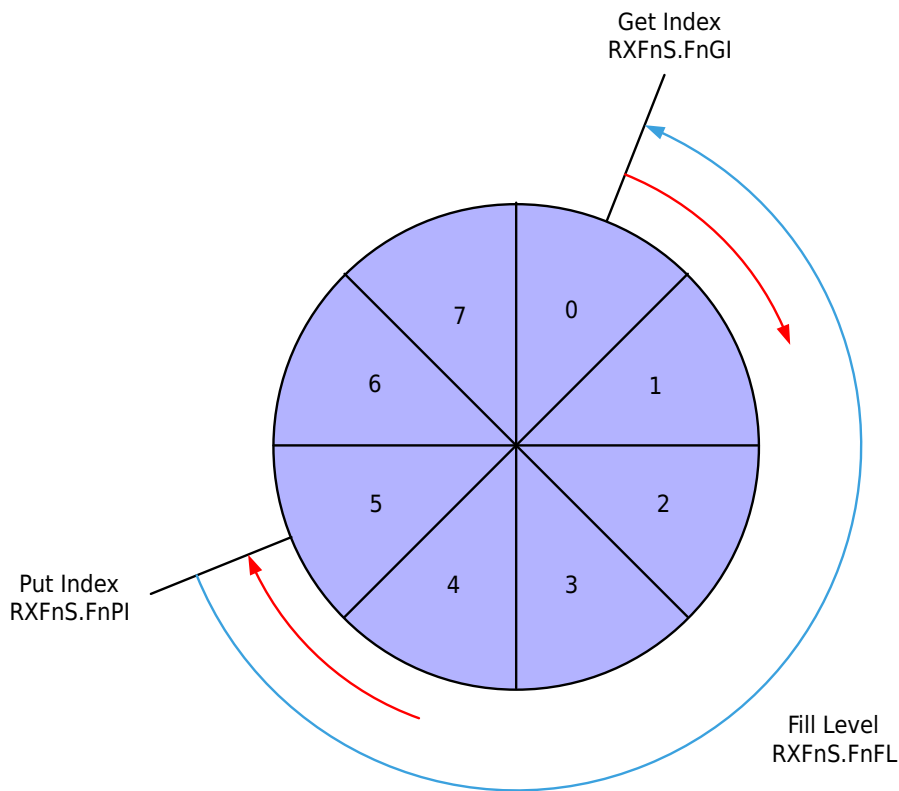


图 32-8 接收 FIFO 状态

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 32-4 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）
0b000	8	4
0b001	12	5
0b010	16	6
0b011	20	7
0b100	24	8
0b101	32	10
0b110	48	14
0b111	64	18

32.3.6.8 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。

32.3.6.9 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。下图显示了读取接收 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。

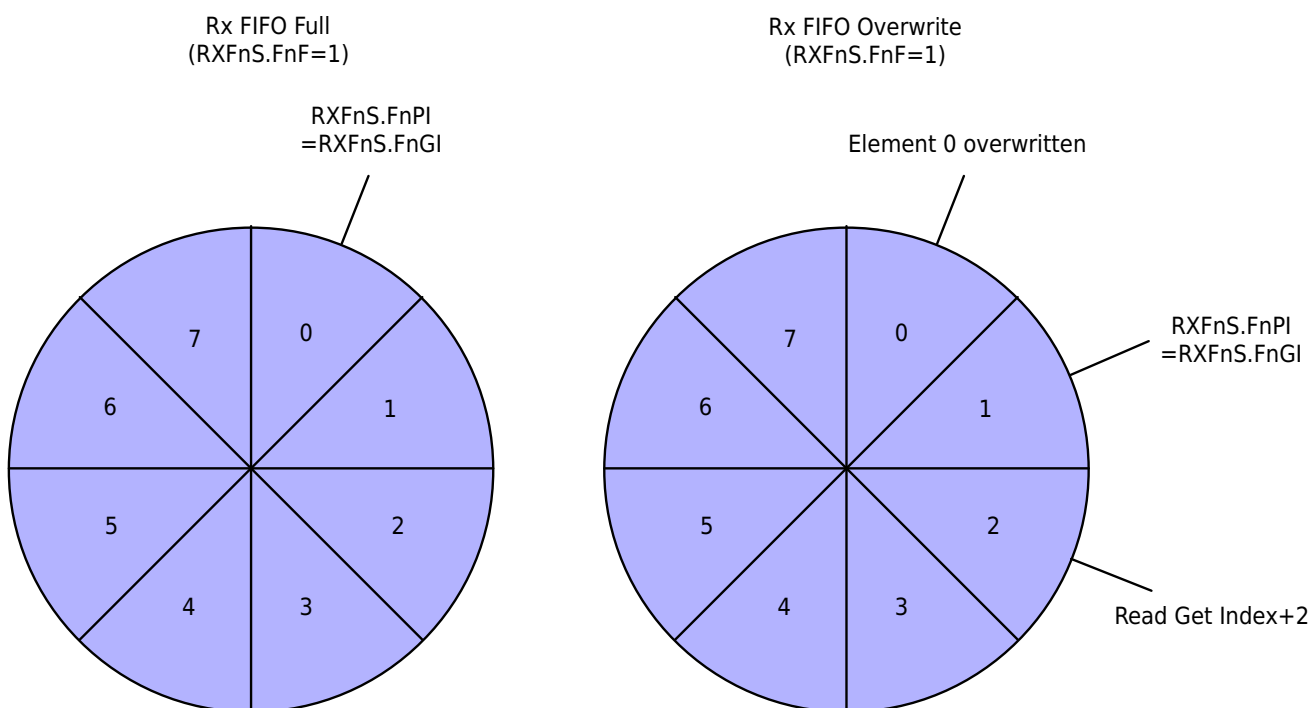


图 32-9 接收 FIFO 溢出处理

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)。

32.3.6.10 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 0b111、SFID2 位为 0b00 (请参见“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 0b111、EFID2 位为 0b00 (请参见“扩展消息 ID 过滤器元素”)。下表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 32-5 接收缓冲区过滤器配置示例

过滤器元素	SFID1[10:0]	SFID2[10:9]	SFID2[5:0]
	EFID1[28:0]	EFID2[10:9]	EFID2[5:0]
0	ID 消息 1	0b00	0b000000
1	ID 消息 2	0b00	0b000001
2	ID 消息 3	0b00	0b000010

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，其他的过滤器元素会继续进行接收过滤。其他消息 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。

32.3.6.11 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

32.3.6.12 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“标准消息 ID 过滤器元素”和“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。下表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 32-6 调试消息过滤器配置示例

过滤器元素	SFID1[10:0]	SFID2[10:9]	SFID2[5:0]
	EFID1[28:0]	EFID2[10:9]	EFID2[5:0]
0	ID 调试消息 A	0b01	0b111101
1	ID 调试消息 B	0b10	0b111110
2	ID 调试消息 C	0b11	0b111111

32.3.7 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）。发送缓冲区的描述请参见“发送缓冲区元素”。下表描述了发送帧的可能配置。

表 32-7 发送帧的可能配置

MCAN_CCCR		发送缓冲区元素		发送帧
BRSE	FDOE	FDF	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
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

 **说明**

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。

32.3.7.1 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。

32.3.7.2 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 32-8 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)
0b000	8	4
0b001	12	5
0b010	16	6
0b011	20	7
0b100	24	8
0b101	32	10
0b110	48	14
0b111	64	18

32.3.7.3 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。

32.3.7.4 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。如果多个发送队列缓冲区配置了相同的消息 ID，则发送顺序取决于存储消息的发送队列缓冲区的编号，而这些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。

32.3.7.5 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。

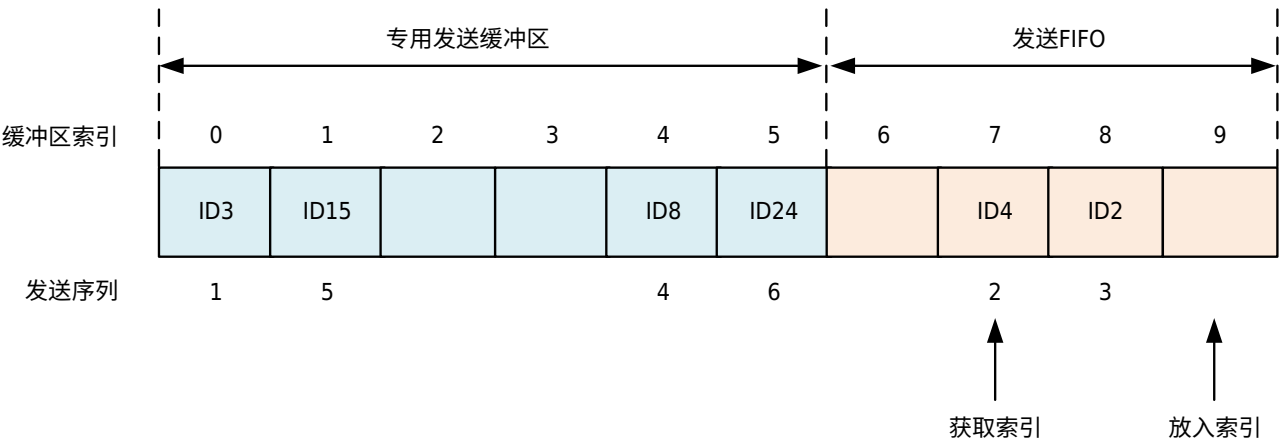


图 32-10 混合配置专用发送缓冲区和发送 FIFO 示例

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_TXFQS.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

32.3.7.6 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。

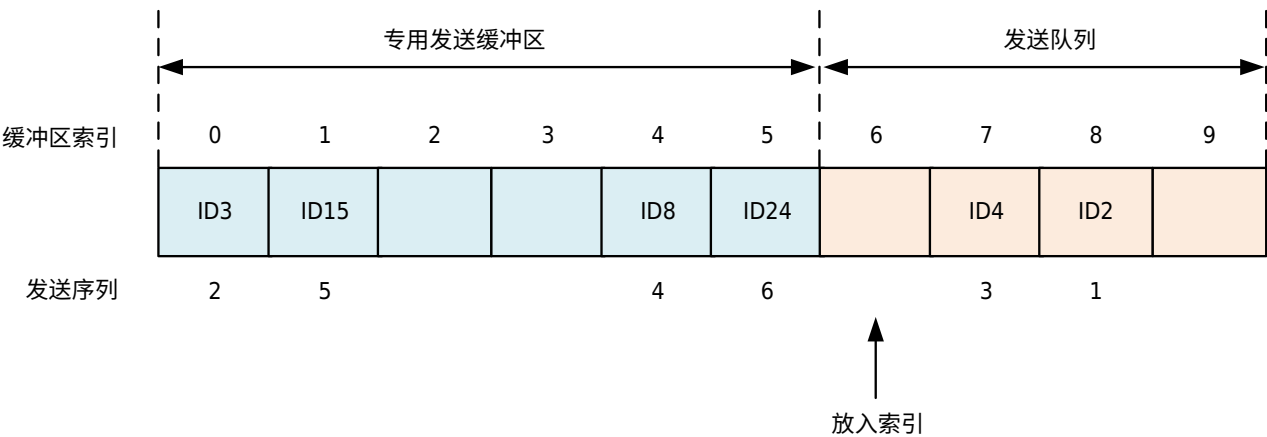


图 32-11 混合配置专用发送缓冲区和发送队列示例

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

32.3.7.7 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。



说明

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。

32.3.7.8 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，“[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。

32.3.8 FIFO 确认处理

接收 FIFO0、接收 FIFO1 和发送事件 FIFO 的获取索引，是通过对应 FIFO 确认索引寄存器进行写操作来控制的，请参见“[MCAN 接收 FIFO0 确认寄存器 \(MCAN_RXF0A\)](#)”、“[MCAN 接收 FIFO1 确认寄存器 \(MCAN_RXF1A\)](#)”和“[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。



说明

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。

32.3.9 消息 RAM

32.3.9.1 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，如下图所示。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。

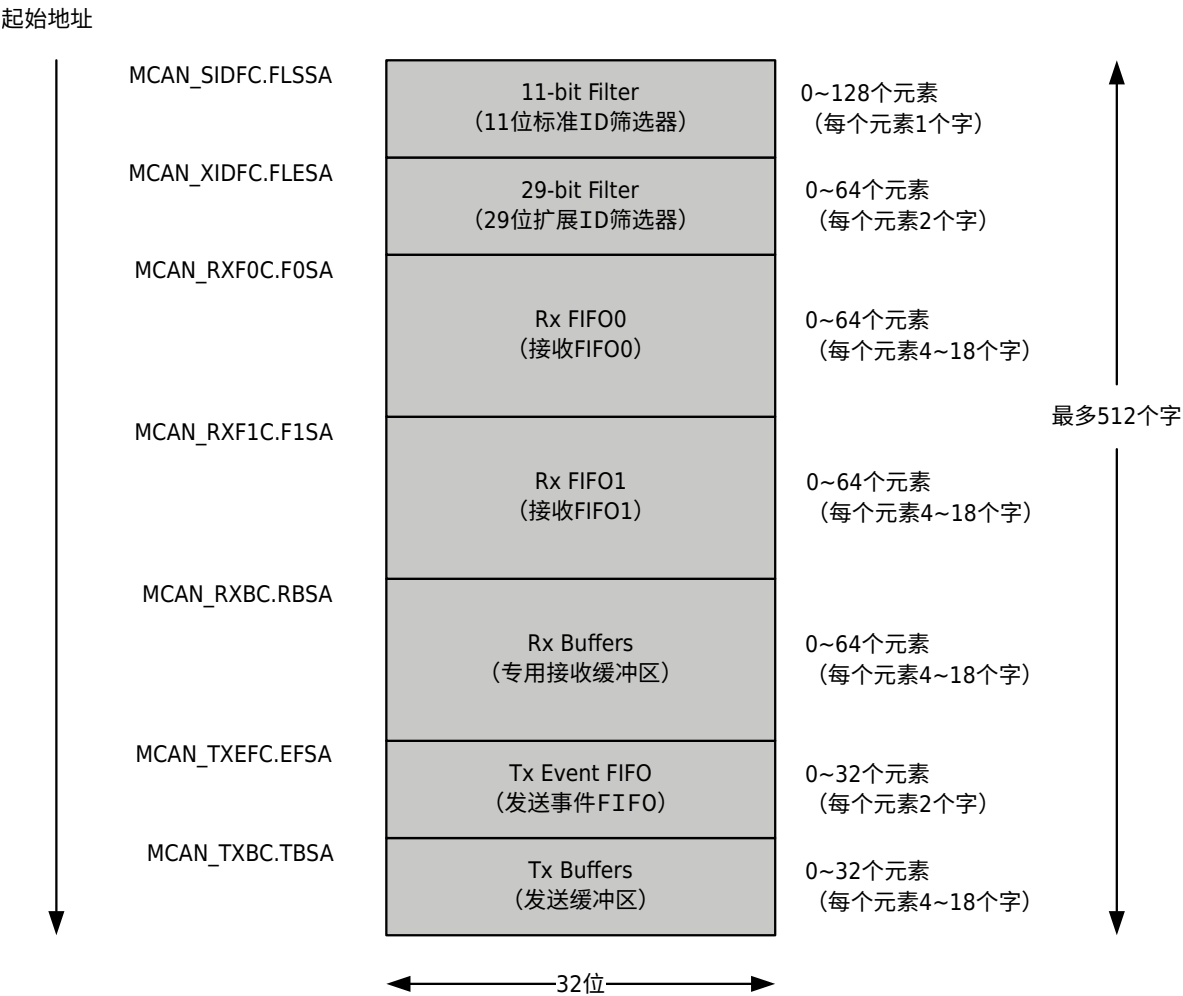


图 32-12 消息 RAM 配置

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。

 **说明**

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。

32.3.9.2 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 元素的结构以及相关说明如下表所示。
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 32-9 接收缓冲区和 FIFO 元素

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	ESI	XTD	RTR	ID																												
R1	ANMF	FIDX							Res.	FDF	BRS	DLC					RXTS															
R2	DB3								DB2							DB1							DB0									
R3	DB7								DB6							DB5							DB4									
.....																																
Rn	DBm								DBm-1							DBm-2							DBm-3									

表 32-10 接收缓冲区和 FIFO 元素说明

位/位域	标记	功能描述
R0[31]	ESI	错误状态指示符（Error State Indicator） ● 0b0：发送节点为错误主动状态 ● 0b1：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。 ● 0b0：11 位标准 ID ● 0b1：29 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于数据帧还是远程帧。 ● 0b0：接收到的帧是数据帧 ● 0b1：接收到的帧是远程帧  说明 CAN FD 没有远程帧。在 CAN FD 帧中（FDF=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。

位/位域	标记	功能描述
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。 ● 0b0: 接收到的帧与过滤器索引 FIDX 指向的过滤器元素匹配 ● 0b1: 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) ● 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) ● 0b0: 经典 CAN 帧格式 ● 0b1: CAN FD 帧格式 (新的 DLC 编码和 CRC)
R1[20]	BRS	比特率切换 (Bit Rate Switch) ● 0b0: 接收帧没有比特率切换 ● 0b1: 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)
...	...	...

位/位域	标记	功能描述
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）

32.3.9.3 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以及相关说明如下表所示。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 32-11 发送缓冲区元素

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T0	ESI	XTD	RTR	ID																												
T1	MM[7:0]								EFC	Res.	FDF	BRS	DLC				MM[15:8]							Res.								
T2	DB3								DB2					DB1							DB0											
T3	DB7								DB6					DB5							DB4											
.....																																
Tn	DBm								DBm-1					DBm-2							DBm-3											

表 32-12 发送缓冲区元素说明

位/位域	标记	功能描述
T0[31]	ESI	错误状态指示符（Error State Indicator） 0b0：CAN FD 格式的 ESI 位仅取决于错误被动标志 0b1：CAN FD 格式的 ESI 位以隐性位发送  说明 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） 0b0：11 位标准 ID 0b1：29 位扩展 ID

位/位域	标记	功能描述
T0[29]	RTR	远程发送请求 (Remote Transmission Request) ● 0b0: 发送数据帧 ● 0b1: 发送远程帧  说明 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) 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) ● 0b0: 不存储发送事件 ● 0b1: 存储发送事件
T1[21]	FDF	FD 格式 (FD Format) ● 0b0: 以经典 CAN 格式发送帧 ● 0b1: 以 CAN FD 格式发送帧
T1[20]	BRS	比特率切换 (Bit Rate Switch) ● 0b0: CAN FD 帧发送时不切换比特率 ● 0b1: CAN FD 帧发送时切换比特率  说明 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。

位/位域	标记	功能描述
T2[31:24]	DB3[7:0]	数据字节 3（Data Byte 3）  说明 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）

32.3.9.4 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 元素的结构以及相关说明如下表所示。

- 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 32-13 发送事件 FIFO 元素

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E0	ESI	XTD	RTR	ID																												
E1A	MM[7:0]								ET		FDF	BRS	DLC			TXTS																
E1B	MM[7:0]								ET		FDF	BRS	DLC			MM[15:8]										Res.						

表 32-14 发送事件 FIFO 元素说明

位/位域	标记	功能描述
E0[31]	ESI	错误状态指示符 (Error State Indicator) ● 0b0: 发送节点为错误主动状态 ● 0b1: 发送节点为错误被动状态
E0[30]	XTD	扩展 ID 标志位 (Extended Identifier) ● 0b0: 11 位标准 ID ● 0b1: 29 位扩展 ID
E0[29]	RTR	远程发送请求 (Remote Transmission Request) ● 0b0: 发送帧为数据帧 ● 0b1: 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) ● 0b00: Reserved ● 0b01: 发送事件 ● 0b10: 取消后仍发送 (在 DAR 模式下发送时始终为该值) ● 0b11: Reserved
E1A/B[21]	FDF	FD 格式 (FD Format) ● 0b0: 经典 CAN 帧格式 ● 0b1: CAN FD 帧格式 (新的 DLC 编码和 CRC)
E1A/B[20]	BRS	比特率切换 (Bit Rate Switch) ● 0b0: CAN FD 帧发送时没有切换比特率 ● 0b1: 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

位/位域	标记	功能描述
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。

32.3.9.5 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 元素的结构以及相关说明如下表所示。

表 32-15 标准消息 ID 过滤器元素

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S0	SFT		SFEC			SFID1											Res.				SFID2											

表 32-16 标准消息 ID 过滤器元素说明

位/位域	标记	功能描述
S0[31:30]	SFT[1:0]	标准过滤器类型 (Standard Filter Type) ● 0b00: 从 SFID1 到 SFID2 的范围过滤器 (SFID2≥SFID1) ● 0b01: SFID1 或 SFID2 定义的双 ID 过滤器 ● 0b10: 经典过滤器: SFID1=ID, SFID2=掩码 ● 0b11: 禁止该过滤器元素  说明 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, 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。 ● 0b000: 禁止该过滤器元素 ● 0b001: 如果匹配, 则存储到接收 FIFO0 中 ● 0b010: 如果匹配, 则存储到接收 FIFO1 中 ● 0b011: 如果匹配, 则拒绝 ID ● 0b100: 如果匹配, 则设置优先级, 不存储

位/位域	标记	功能描述
S0[29:27]	SFEC[2:0]	● 0b101: 如果匹配, 则设置优先级并存储到 FIFO0 中 ● 0b110: 如果匹配, 则设置优先级并存储到 FIFO1 中 ● 0b111: 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]	SFID1[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。 ● 0b00: 将消息存储到接收缓冲区中 ● 0b01: 调试消息 A ● 0b10: 调试消息 B ● 0b11: 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)。

32.3.9.6 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 元素的结构以及相关说明如下表所示。

表 32-17 扩展消息 ID 过滤器元素

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F0	EFEC			EFID1																												
F1	EFT		Res.	EFID2																												

表 32-18 扩展消息 ID 过滤器元素说明

位/位域	标记	功能描述
F0[31:29]	EFEC[2:0]	扩展过滤器元素配置 (Extended Filter Element Configuration) 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。 ● 0b000: 禁止该过滤器元素

位/位域	标记	功能描述
F0[31:29]	EFEC[2:0]	● 0b001: 如果匹配, 则存储到接收 FIFO0 中 ● 0b010: 如果匹配, 则存储到接收 FIFO1 中 ● 0b011: 如果匹配, 则拒绝 ID ● 0b100: 如果匹配, 则设置优先级 ● 0b101: 如果匹配, 则设置优先级并存储到 FIFO0 中 ● 0b110: 如果匹配, 则设置优先级并存储到 FIFO1 中 ● 0b111: 存储到接收缓冲区中或作为调试消息, 忽略 EFT[1:0]的配置 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。
F0[28:0]	EFID1[28:0]	扩展过滤器 ID1 (Extended Filter ID 1) 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。
F1[31:30]	EFT[1:0]	扩展过滤器类型 (Extended Filter Type) ● 0b00: 从 EFID1 到 EFID2 的范围过滤器(EFID2≥EFID1) ● 0b01: EFID1 或 EFID2 定义的双 ID 过滤器 ● 0b10: 经典过滤器: EFID1=ID, EFID2=掩码 ● 0b11: 从 EFID1 到 EFID2 的范围过滤器 (EFID2≥EFID1) , 不使用 MCAN_XIDAM 掩码
F1[28:0]	EFID2[28:0]	扩展过滤器 ID2 (Extended Filter ID 2) 该位域具有不同含义, 具体视 EFEC 的配置而定: ● 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]	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。 ● 0b00: 将消息存储到接收缓冲区中 ● 0b01: 调试消息 A ● 0b10: 调试消息 B ● 0b11: 调试消息 C
	EFID2[5:0]	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) 。

32.4 寄存器

32.4.1 寄存器总表

- MCAN1 基地址: 0x40029000
- MCAN2 基地址: 0x40029400

表 32-19 控制器局域网 (MCAN) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x04	MCAN_ENDN	MCAN 字节序寄存器	32
0x0C	MCAN_DBTP	MCAN 数据位时间和预分频寄存器	32
0x10	MCAN_TEST	MCAN 测试寄存器	32
0x14	MCAN_RWD	MCAN RAM 看门狗寄存器	32
0x18	MCAN_CCCR	MCAN CC 控制寄存器	32
0x1C	MCAN_NBTP	MCAN 标称位时间和预分频寄存器	32
0x20	MCAN_TSCC	MCAN 时间戳计数器配置寄存器	32
0x24	MCAN_TSCV	MCAN 时间戳计数器值寄存器	32
0x28	MCAN_TOCC	MCAN 超时计数器配置寄存器	32
0x2C	MCAN_TOCV	MCAN 超时计数器值寄存器	32
0x40	MCAN_ECR	MCAN 错误计数器寄存器	32
0x44	MCAN_PSR	MCAN 协议状态寄存器	32
0x48	MCAN_TDCR	MCAN 发送延迟补偿寄存器	32
0x50	MCAN_IR	MCAN 中断寄存器	32
0x54	MCAN_IE	MCAN 中断使能寄存器	32
0x58	MCAN_ILS	MCAN 中断线选择寄存器	32
0x5C	MCAN_IIE	MCAN 中断线使能寄存器	32
0x80	MCAN_GFC	MCAN 全局过滤器配置寄存器	32
0x84	MCAN_SIDFC	MCAN 标准 ID 过滤器配置寄存器	32
0x88	MCAN_XIDFC	MCAN 扩展 ID 过滤器配置寄存器	32
0x90	MCAN_XIDAM	MCAN 扩展 ID 与掩码寄存器	32
0x94	MCAN_HPMS	MCAN 高优先级消息状态寄存器	32
0x98	MCAN_NDAT1	MCAN 新数据 1 寄存器	32
0x9C	MCAN_NDAT2	MCAN 新数据 2 寄存器	32
0xA0	MCAN_RXF0C	MCAN 接收 FIFO0 配置寄存器	32

偏移地址	寄存器	描述	位宽
0xA4	MCAN_RXF0S	MCAN 接收 FIFO0 状态寄存器	32
0xA8	MCAN_RXF0A	MCAN 接收 FIFO0 确认寄存器	32
0xAC	MCAN_RXBC	MCAN 接收缓冲区配置寄存器	32
0xB0	MCAN_RXF1C	MCAN 接收 FIFO1 配置寄存器	32
0xB4	MCAN_RXF1S	MCAN 接收 FIFO1 状态寄存器	32
0xB8	MCAN_RXF1A	MCAN 接收 FIFO1 确认寄存器	32
0xBC	MCAN_RXESC	MCAN 接收缓冲区和 FIFO 元素大小配置寄存器	32
0xC0	MCAN_TXBC	MCAN 发送缓冲区配置寄存器	32
0xC4	MCAN_TXFQS	MCAN 发送 FIFO/队列状态寄存器	32
0xC8	MCAN_TXESC	MCAN 发送缓冲区元素大小配置寄存器	32
0xCC	MCAN_TXBRP	MCAN 发送缓冲区请求挂起寄存器	32
0xD0	MCAN_TXBAR	MCAN 发送缓冲区添加请求寄存器	32
0xD4	MCAN_TXBCR	MCAN 发送缓冲区取消请求寄存器	32
0xD8	MCAN_TXBTO	MCAN 发送缓冲区发送已发生寄存器	32
0xDC	MCAN_TXBCF	MCAN 发送缓冲区取消已完成寄存器	32
0xE0	MCAN_TXBTIE	MCAN 发送缓冲区发送中断使能寄存器	32
0xE4	MCAN_TXBCIE	MCAN 发送缓冲区取消已完成中断使能寄存器	32
0xF0	MCAN_TXEFC	MCAN 发送事件 FIFO 配置寄存器	32
0xF4	MCAN_TXEFS	MCAN 发送事件 FIFO 状态寄存器	32
0xF8	MCAN_TXEFA	MCAN 发送事件 FIFO 确认寄存器	32

32.4.2 MCAN 字节序寄存器（MCAN_ENDN）

Offset	Bit Position																															
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x87654321																															
Name	ETV																															
Access	RO																															

位/位域	标记	位名	功能描述	读写
31:0	ETV	字节序测试值	测试值为 0x87654321	RO

32.4.3 MCAN 数据位时间和预分频寄存器 (MCAN_DBTP)

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。

Offset	Bit Position																																
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000A33																																
Name	Reserved								TDC	Reserved				DBRP				Reserved				DTSEG1				DTSEG2				DSJW			
Access									RWA					RWA								RWA				RWA				RWA			

位/位域	标记	位名	功能描述	读写
31:24	Reserved	保留	-	-
23	TDC	发送延迟补偿	0b0: 禁止 0b1: 使能	RWA
22:21	Reserved	保留	-	-
20:16	DBRP	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。预分频有效设定值为 0~31，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
15:13	Reserved	保留	-	-
12:8	DTSEG1	采样点之前的数据时间	有效设定值为 0~31，硬件实际使用值为编程值加 1。	RWA
7:4	DTSEG2	采样点之后的数据时间	有效设定值为 1~15，硬件实际使用值为编程值加 1。	RWA
3:0	DSJW	数据 (重新) 同步跳转宽度	有效设定值为 0~15，硬件实际使用值为编程值加 1。	RWA

 **说明**

1. 当 MCAN 通信时钟为 8MHz 时，复位值 0x00000A33 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。

32.4.4 MCAN 测试寄存器 (MCAN_TEST)

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。

Offset	Bit Position																																	
0x10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x000000X0（Bit7 的复位值由 MCAN_RX 引脚的实际电平决定）																																	
Name	Reserved										SVAL	TXBNS				Reserved	PVAL	TXBNP				RX	TX		LBCK		Reserved							
Access											RO	RO					RO	RO				RO	RWA		RWA									

位/位域	标记	位名	功能描述	读写
31:22	Reserved	保留	-	-
21	SVAL	已开始有效	0b0: TXBNS 的值无效 0b1: TXBNS 的值有效	RO
20:16	TXBNS	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，有效设定值为 0~31。	RO
15:14	Reserved	保留	-	-
13	PVAL	已准备有效	0b0: TXBNP 的值无效 0b1: TXBNP 的值有效	RO
12:8	TXBNP	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，有效设定值为 0~31。	RO
7	RX	接收引脚	监测到的 MCAN_RX 引脚的实际值 0b0: CAN 总线为显性 (MCAN_RX=0) 0b1: CAN 总线为隐性 (MCAN_RX=1)	RO
6:5	TX	发送引脚控制	0b00: 复位值，MCAN_TX 引脚由 CAN 内核控制，在 CAN 位时间结束时更新 0b01: 可以在 MCAN_TX 引脚监测采样点 0b10: MCAN_TX 引脚输出显性电平 (0) 0b11: MCAN_TX 引脚输出隐性电平 (1)	RWA
4	LBCK	回环模式	0b0: 复位值，禁止回环模式 0b1: 使能回环模式 (详见"回环模式")	RWA
3:0	Reserved	保留	-	-

32.4.5 MCAN RAM 看门狗寄存器（MCAN_RWD）

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。

Offset	Bit Position																															
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																WDV								WDC							
Access																	RO								RWA							

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:8	WDV	看门狗计数值	消息 RAM 看门狗计数器的实际值	RO
7:0	WDC	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

32.4.6 MCAN CC 控制寄存器 (MCAN_CCCR)

寄存器各个位的设置条件请参见“[软件初始化](#)”。

Offset	Bit Position																																															
0x18	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
Reset	0x00000001																																															
Name	Reserved																	NISO		TXP		EFBI		PXHD		WMM		Reserved		BRSE		FDOE		TEST		DAR		MON		CSR		CSA		ASM		CCE		INIT
Access																	RWA	NISO	RWA	TXP	RWA	EFBI	RWA	PXHD	RWA	WMM			RWA	BRSE	RWA	FDOE	RWA	TEST	RWA	DAR	RWA	MON	RW	CSR	RO	CSA	RWA	ASM	RWA	CCE	RW	INIT

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15	NISO	非 ISO 模式选择	0b0: 使用 ISO 11898-1:2015 规定的 CAN FD 帧格式 0b1: 使用 Bosch CAN FD 规范 V1.0 规定的 CAN FD 帧格式	RWA
14	TXP	传输暂停	0b0: 禁止传输暂停 0b1: 使能传输暂停  说明 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。	RWA
13	EFBI	总线同步期间的边沿滤波	0b0: 禁止边沿滤波 0b1: 需要两个连续的显性 tq 才能检测硬同步边沿	RWA
12	PXHD	协议异常处理禁止	0b0: 使能协议异常处理 0b1: 禁止协议异常处理  说明 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
11	WMM	宽消息标记	0b0: 使用 8 位消息标记 0b1: 使用 16 位消息标记, 替换发送事件 FIFO 中的 16 位时间戳  说明 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。	RWA
10	Reserved	保留	-	-
9	BRSE	比特率切换	0b0: 禁止发送时进行比特率切换 0b1: 使能发送时进行比特率切换  说明 当 CAN FD 模式无效 (FDOE=0) 时, BRSE 的设置不起作用。	RWA

位/位域	标记	位名	功能描述	读写
8	FDOE	FD 模式使能	0b0: 禁止 FD 模式 0b1: 使能 FD 模式	RWA
7	TEST	测试模式使能	写 1 使能测试模式, 写 0 无效 读取: 0b0: 正常模式, 测试寄存器 MCAN_TEST 保持复位值 0b1: 测试模式, 使能写访问测试寄存器 MCAN_TEST	RWA
6	DAR	禁止自动重发	0b0: 使能允许自动重发未成功发送的消息 0b1: 禁止自动重发	RWA
5	MON	总线监听模式	写 1 使能总线监听模式, 写 0 无效 读取: 0b0: 禁止总线监听模式 0b1: 使能总线监听模式  说明 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。	RWA
4	CSR	时钟停止请求	0b0: 未请求时钟停止 0b1: 已请求时钟停止  说明 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。	RW
3	CSA	时钟停止确认	0b0: 未确认时钟停止请求 0b1: 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	RO
2	ASM	受限操作模式	写 1 使能受限工作模式, 写 0 无效 读取: 0b0: 正常模式 0b1: 受限工作模式  说明 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, 请参见“受限工作模式”。	RWA
1	CCE	配置更改使能	0b0: CPU 对受保护的配置寄存器无写访问权限 0b1: CPU 对受保护的配置寄存器有写访问权限 (当 INIT=1 时)	RWA
0	INIT	初始化	0b0: 正常工作 0b1: 启动初始化  说明 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。	RW

32.4.7 MCAN 标称位时间和预分频寄存器 (MCAN_NBTP)

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。

Offset	Bit Position																																
0x1C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x06000A03																																
Name	NSJW								NBRP								NTSEG1								Reserved	NTSEG2							
Access	RWA								RWA								RWA									RWA							

位/位域	标记	位名	功能描述	读写
31:25	NSJW	标称 (重) 同步 跳转宽度	有效设定值为 0~127，硬件实际使用值为编程值加 1。	RWA
24:16	NBRP	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。预分频有效设定值为 0~511，硬件实际使用值为编程值加 1。	RWA
15:8	NTSEG1	采样点之前的标 称时间段	有效设定值为 1~255，硬件实际使用值为编程值加 1。	RWA
7	Reserved	保留	-	-
6:0	NTSEG2	采样点之后的标 称时间段	有效设定值为 1~127，硬件实际使用值为编程值加 1。	RWA

 说明

当 MCAN 时钟为 8MHz 时，复位值 0x06000A03 会将 MCAN 的比特率配置为 500kbps。

32.4.8 MCAN 时间戳计数器配置寄存器 (MCAN_TSCC)

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["时间戳生成"](#)。

Offset	Bit Position																															
0x20	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved												TCP		Reserved												TSS					
Access													RWA														RWA					

位/位域	标记	位名	功能描述	读写
31:20	Reserved	保留	-	-
19:16	TCP	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
15:2	Reserved	保留	-	-
1:0	TSS	时间戳选择	0b00：时间戳计数器的值一直为 0x0000 0b01：时间戳计数器的值根据 TCP 递增 0b10：禁止设置 0b11：时间戳计数器的值一直为 0x0000（等同于设定值“0b00”）	RWA

32.4.9 MCAN 时间戳计数器值寄存器 (MCAN_TSCV)

Offset	Bit Position																															
0x24	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																TSC															
Access																																

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	TSC	时间戳计数器	在接收到或发送 SOF (Start Of Frame) 时捕获的内部时间戳计数器值。 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。写操作会将计数器清零。  说明 “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。 - 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。	RWC

32.4.10 MCAN 超时计数器配置寄存器 (MCAN_TOCC)

超时计数器的介绍请参见“[超时计数器](#)”。

Offset	Bit Position																																	
0x28	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0xFFFF0000																																	
Name	TOP																Reserved																TOS	ETOC
Access	RW																																RWA	RWA

位/位域	标记	位名	功能描述	读写
31:16	TOP	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
15:3	Reserved	保留	-	-
2:1	TOS	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： ● 0b00：连续模式 ● 0b01：由发送事件 FIFO 控制超时 ● 0b10：由接收 FIFO0 控制超时 ● 0b11：由接收 FIFO1 控制超时	RWA
0	ETOC	超时计数器使能	● 0b0：禁止 ● 0b1：使能	RWA

32.4.11 MCAN 超时计数器值寄存器 (MCAN_TOCV)

Offset	Bit Position																															
0x2C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000FFFF																															
Name	Reserved																TOC															
Access																																

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:0	TOC	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 的设置值决定。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。  说明 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

32.4.12 MCAN 错误计数器寄存器 (MCAN_ECR)

Offset	Bit Position																															
0x40	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved								RC CEL								RP	RO REC								RO TEC						
Access									RC								RO	RO								RO						

位/位域	标记	位名	功能描述	读写
31:24	Reserved	保留	-	-
23:16	CEL	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 下一次增加时中断标志 MCAN_IR.ELO 置位。  说明 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
15	RP	接收错误被动	0b0：接收错误计数器低于错误被动级别 128 0b1：接收错误计数器已达到错误被动级别 128	RO
14:8	REC	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	RO
7:0	TEC	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	RO

 **说明**

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。

32.4.13 MCAN 协议状态寄存器 (MCAN_PSR)

Offset	Bit Position																																
0x44	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000707																																
Name	Reserved								TDCV								Reserved	PXE	RFDF	RBRS	RESI	DLEC				BO	EW	EP	ACT		LEC		
Access									RO									RO	RO	RO	RO	RO				RO	RO	RO	RO	RO		RS	

位/位域	标记	位名	功能描述	读写
31:23	Reserved	保留	-	-
22:16	TDCV	发送延迟补偿值	实际的发送延迟补偿值，有效设定值为 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。	RO
15	Reserved	保留	-	-
14	PXE	协议异常事件	0b0: 自上次读访问后未发生协议异常事件 0b1: 已发生协议异常事件  说明 字节访问时，读取寄存器的字节 0 会将该位清零，读取字节 3/2/1 没有影响。	RO
13	RFDF	收到 CAN FD 消息	该位的置位与接收过滤无关。 0b0: 自从上次被 CPU 清零后未收到 CAN FD 格式的消息 0b1: 收到 FDF 标志为 1 的 CAN FD 格式的消息  说明 字节访问时，读取寄存器的字节 0 会将该位清零，读取字节 3/2/1 没有影响。	RO
12	RBRS	上次收到的 CAN FD 消息的 BRS 标志	该位与 RFDF 一起置位，与接收过滤无关。 0b0: 上次收到的 CAN FD 消息的 BRS 标志未置位 0b1: 上次收到的 CAN FD 消息的 BRS 标志已置位  说明 字节访问时，读取寄存器的字节 0 会将该位清零，读取字节 3/2/1 没有影响。	RO

位/位域	标记	位名	功能描述	读写
11	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。 0b0: 上次收到的 CAN FD 消息的 ESI 标志未置位 0b1: 上次收到的 CAN FD 消息的 ESI 标志已置位  说明 字节访问时，读取寄存器的字节 0 会将该位清零，读取字节 3/2/1 没有影响。	RO
10:8	DLEC	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。  说明 字节访问时，读取寄存器的字节 0 会将该位设置为 0b111，读取字节 3/2/1 没有影响。	RO
7	BO	Bus_Off 状态	0b0: MCAN 未处于 Bus_Off 状态 0b1: MCAN 处于 Bus_Off 状态	RO
6	EW	警告状态	0b0: 两个错误计数器均小于错误警告 (Error_Warning) 上限 96 0b1: 至少有一个错误计数器已达到错误警告 (Error_Warning) 上限 96	RO
5	EP	错误被动	0b0: 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 0b1: MCAN 处于错误被动 (Error_Passive) 状态	RO
4:3	ACT	通信状态	监测 MCAN 的通信状态： 0b00: 同步中，节点正在进行 CAN 通信同步 0b01: 空闲，节点既不是接收节点也不是发送节点 0b10: 接收，节点作为接收节点工作 0b11: 发送，节点作为发送节点工作  说明 协议异常事件会将该位设置为 0b00。	RO
2:0	LEC	最后一次错误码	0b000: 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 0b001: 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 0b010: 格式错误 (Form Error)。已接收帧的固定格式部分有格式错误 0b011: ACK 错误 (ACK Error)。MCAN 发送的消息没有被其他节点确认 0b100: 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) 0b101: 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	RS

位/位域	标记	位名	功能描述	读写
2:0	LEC	最后一次错误码	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） ● 0b110: 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 ● 0b111: 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  说明 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 会将该位设置为 0b111，读取字节 3/2/1 没有影响。	RS

32.4.14 MCAN 发送延迟补偿寄存器 (MCAN_TDCR)

Offset	Bit Position																																
0x48	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																TDCO						Reserved	TDCF									
Access																	RWA							RWA									

位/位域	标记	位名	功能描述	读写
31:15	Reserved	保留	-	-
14:8	TDCO	发送延迟补偿 SSP 偏移	该偏移值指从 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
7	Reserved	保留	-	-
6:0	TDCF	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

32.4.15 MCAN 中断寄存器 (MCAN_IR)

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 无效。**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。

Offset	Bit Position																															
0x50	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved	ARA	PED	PEA	WDI	BO	EW	EP	ELO	Reserved	DRX	TOO	MRAF	TSW	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM	RFIL	RFIF	RFIW	RFIN	RFOL	RFOF	RFOW	RFON		
Access		RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:30	Reserved	保留	-	-
29	ARA	访问保留地址	0b0: 未对保留地址进行访问 0b1: 已对保留地址进行访问	RW
28	PED	数据阶段协议错误 (使用数据位时间)	0b0: 未检测到数据阶段协议错误 0b1: 检测到数据阶段协议错误 (MCAN_PSR.DLEC≠0b000/0b111)	RW
27	PEA	仲裁阶段协议错误 (使用标称位时间)	0b0: 未检测到仲裁阶段协议错误 0b1: 检测到仲裁阶段协议错误 (MCAN_PSR.LEC≠0b000/0b111)	RW
26	WDI	看门狗中断	0b0: 未发生消息 RAM 看门狗事件 0b1: 因 READY 信号缺失而发生消息 RAM 看门狗事件	RW
25	BO	Bus_Off 状态	0b0: Bus_Off 状态未改变 0b1: Bus_Off 状态已改变	RW
24	EW	警告状态	0b0: 错误警告 (Error_Warning) 状态未改变 0b1: 错误警告 (Error_Warning) 状态已改变	RW
23	EP	错误被动	0b0: 错误被动 (Error_Passive) 状态未改变 0b1: 错误被动 (Error_Passive) 状态已改变	RW
22	ELO	错误记录溢出	0b0: CAN 错误记录计数器未溢出 0b1: CAN 错误记录计数器已溢出	RW
21:20	Reserved	保留	-	-
19	DRX	消息存储到专用接收缓冲区	0b0: 接收缓存区未更新 0b1: 至少有一条消息已存储到接收缓冲区  说明 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。	RW
18	TOO	发生超时	0b0: 未发生超时 0b1: 已发生超时	RW

位/位域	标记	位名	功能描述	读写
17	MRAF	消息 RAM 访问失败	0b0: 未发生消息 RAM 访问失败 0b1: 发生了消息 RAM 访问失败 接收处理中置位条件: - 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。 - 无法向消息 RAM 写入消息, 消息存储将终止。  说明 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: - 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。	RW
16	TSW	时间戳回卷	0b0: 时间戳计数器未回卷 0b1: 时间戳计数器已回卷	RW
15	TEFL	发送事件 FIFO 元素丢失	0b0: 发送事件 FIFO 元素未丢失 0b1: 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	RW
14	TEFF	发送事件 FIFO 已满	0b0: 发送事件 FIFO 未滿 0b1: 发送事件 FIFO 已滿	RW
13	TEFW	发送事件 FIFO 到达水位线	0b0: 发送事件 FIFO 的填充级别低于水位线 0b1: 发送事件 FIFO 的填充级别达到水位线	RW
12	TEFN	发送事件 FIFO 新条目	0b0: 发送事件 FIFO 无变化 0b1: 发送处理单元写入了发送事件 FIFO 元素	RW
11	TFE	发送 FIFO 为空	0b0: 发送 FIFO 非空 0b1: 发送 FIFO 为空	RW
10	TCF	发送取消已完成	0b0: 发送取消未完成 0b1: 发送取消已完成	RW
9	TC	发送完成	0b0: 发送未完成 0b1: 发送已完成	RW
8	HPM	高优先级消息	0b0: 未收到高优先级消息 0b1: 已收到高优先级消息	RW
7	RF1L	接收 FIFO1 消息丢失	0b0: 接收 FIFO1 消息未丢失 0b1: 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	RW
6	RF1F	接收 FIFO1 已滿	0b0: 接收 FIFO1 未滿 0b1: 接收 FIFO1 已滿	RW
5	RF1W	接收 FIFO1 到达水位线	0b0: 接收 FIFO1 的填充级别低于水位线 0b1: 接收 FIFO1 的填充级别达到水位线	RW

位/位域	标记	位名	功能描述	读写
4	RF1N	接收 FIFO1 新消息	0b0: 没有新消息写入接收 FIFO1 0b1: 有新消息写入接收 FIFO1	RW
3	RF0L	接收 FIFO0 消息丢失	0b0: 接收 FIFO0 消息未丢失 0b1: 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	RW
2	RF0F	接收 FIFO0 已满	0b0: 接收 FIFO0 未滿 0b1: 接收 FIFO0 已滿	RW
1	RF0W	接收 FIFO0 到达水位线	0b0: 接收 FIFO0 的填充级别低于水位线 0b1: 接收 FIFO0 的填充级别达到水位线	RW
0	RF0N	接收 FIFO0 新消息	0b0: 没有新消息写入接收 FIFO0 0b1: 有新消息写入接收 FIFO0	RW

32.4.16 MCAN 中断使能寄存器 (MCAN_IE)

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。

Offset	Bit Position																																																										
0x54	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																											
Reset	0x00000000																																																										
Name	Reserved										Reserved																																																
Access		RW	ARAE	RW	PEDE	RW	PEAE	RW	WDIE	RW	BOE	RW	EWE	RW	EPE	RW	ELOE			RW	DRXE	RW	TOOE	RW	MRAFE	RW	TSWE	RW	TEFLE	RW	TEFFE	RW	TEFWE	RW	TEFNE	RW	TTEE	RW	TCFE	RW	TCE	RW	HPME	RW	RF1LE	RW	RF1FE	RW	RF1WE	RW	RF1NE	RW	RF0LE	RW	RF0FE	RW	RF0WE	RW	RF0NE

位/位域	标记	位名	功能描述	读写
31:30	Reserved	保留	-	-
29	ARAE	访问保留地址中断使能	0b0: 禁止中断 0b1: 使能中断	RW
28	PEDE	数据阶段协议错误中断使能	0b0: 禁止中断 0b1: 使能中断	RW
27	PEAE	仲裁阶段协议错误中断使能	0b0: 禁止中断 0b1: 使能中断	RW
26	WDIE	看门狗中断使能	0b0: 禁止中断 0b1: 使能中断	RW
25	BOE	Bus_Off 状态中断使能	0b0: 禁止中断 0b1: 使能中断	RW
24	EWE	警告状态中断使能	0b0: 禁止中断 0b1: 使能中断	RW
23	EPE	错误被动中断使能	0b0: 禁止中断 0b1: 使能中断	RW
22	ELOE	错误记录溢出中断使能	0b0: 禁止中断 0b1: 使能中断	RW
21:20	Reserved	保留	-	-
19	DRXE	消息存储到专用接收缓冲区中断使能	0b0: 禁止中断 0b1: 使能中断	RW
18	TOOE	发生超时中断使能	0b0: 禁止中断 0b1: 使能中断	RW
17	MRAFE	消息 RAM 访问失败中断使能	0b0: 禁止中断 0b1: 使能中断	RW
16	TSWE	时间戳回卷中断使能	0b0: 禁止中断 0b1: 使能中断	RW
15	TEFLE	发送事件 FIFO 元素丢失中断使能	0b0: 禁止中断 0b1: 使能中断	RW

位/位域	标记	位名	功能描述	读写
14	TEFFE	发送事件 FIFO 已满中断使能	● 0b0: 禁止中断 ● 0b1: 使能中断	RW
13	TEFWE	发送事件 FIFO 到达水位线中断使能	● 0b0: 禁止中断 ● 0b1: 使能中断	RW
12	TEFNE	发送事件 FIFO 新条目中断使能	● 0b0: 禁止中断 ● 0b1: 使能中断	RW
11	TFEE	发送 FIFO 为空中断使能	● 0b0: 禁止中断 ● 0b1: 使能中断	RW
10	TCFE	发送取消已完成中断使能	● 0b0: 禁止中断 ● 0b1: 使能中断	RW
9	TCE	发送完成中断使能	● 0b0: 禁止中断 ● 0b1: 使能中断	RW
8	HPME	高优先级消息中断使能	● 0b0: 禁止中断 ● 0b1: 使能中断	RW
7	RF1LE	接收 FIFO1 消息丢失中断使能	● 0b0: 禁止中断 ● 0b1: 使能中断	RW
6	RF1FE	接收 FIFO1 已满中断使能	● 0b0: 禁止中断 ● 0b1: 使能中断	RW
5	RF1WE	接收 FIFO1 到达水位线中断使能	● 0b0: 禁止中断 ● 0b1: 使能中断	RW
4	RF1NE	接收 FIFO1 新消息中断使能	● 0b0: 禁止中断 ● 0b1: 使能中断	RW
3	RF0LE	接收 FIFO0 消息丢失中断使能	● 0b0: 禁止中断 ● 0b1: 使能中断	RW
2	RF0FE	接收 FIFO0 已满中断使能	● 0b0: 禁止中断 ● 0b1: 使能中断	RW
1	RF0WE	接收 FIFO0 到达水位线中断使能	● 0b0: 禁止中断 ● 0b1: 使能中断	RW
0	RF0NE	接收 FIFO0 新消息中断使能	● 0b0: 禁止中断 ● 0b1: 使能中断	RW

32.4.17 MCAN 中断线选择寄存器 (MCAN_ILS)

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。

Offset	Bit Position																																
0x58	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved	ARAL	PEDL	PEAL	WDIL	BOL	EWL	EPL	ELOL	Reserved	DRXL	TOOL	MRAFL	TSWL	TEFLL	TEFVL	TEFNL	TEFEL	TCFL	TCL	HPML	RFILL	RFIFL	RFIWL	RFINL	RFOLL	RF0FL	RF0WL	RF0NL				
Access		RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:30	Reserved	保留	-	-
29	ARAL	访问保留地址中断线	0b0: 选择中断线 0 0b1: 选择中断线 1	RW
28	PEDL	数据阶段协议错误中断线	0b0: 选择中断线 0 0b1: 选择中断线 1	RW
27	PEAL	仲裁阶段协议错误中断线	0b0: 选择中断线 0 0b1: 选择中断线 1	RW
26	WDIL	看门狗中断线	0b0: 选择中断线 0 0b1: 选择中断线 1	RW
25	BOL	Bus_Off 状态中断线	0b0: 选择中断线 0 0b1: 选择中断线 1	RW
24	EWL	警告状态中断线	0b0: 选择中断线 0 0b1: 选择中断线 1	RW
23	EPL	错误被动中断线	0b0: 选择中断线 0 0b1: 选择中断线 1	RW
22	ELOL	错误记录溢出中断线	0b0: 选择中断线 0 0b1: 选择中断线 1	RW
21:20	Reserved	保留	-	-
19	DRXL	消息存储到专用接收缓冲区中断线	0b0: 选择中断线 0 0b1: 选择中断线 1	RW
18	TOOL	发生超时中断线	0b0: 选择中断线 0 0b1: 选择中断线 1	RW
17	MRAFL	消息 RAM 访问失败中断线	0b0: 选择中断线 0 0b1: 选择中断线 1	RW
16	TSWL	时间戳回卷中断线	0b0: 选择中断线 0 0b1: 选择中断线 1	RW
15	TEFLL	发送事件 FIFO 元素丢失中断线	0b0: 选择中断线 0 0b1: 选择中断线 1	RW

位/位域	标记	位名	功能描述	读写
14	TEFFL	发送事件 FIFO 已满中断线	● 0b0: 选择中断线 0 ● 0b1: 选择中断线 1	RW
13	TEFWL	发送事件 FIFO 到达水位线中断线	● 0b0: 选择中断线 0 ● 0b1: 选择中断线 1	RW
12	TEFNL	发送事件 FIFO 新条目中断线	● 0b0: 选择中断线 0 ● 0b1: 选择中断线 1	RW
11	TFEL	发送 FIFO 为空中断线	● 0b0: 选择中断线 0 ● 0b1: 选择中断线 1	RW
10	TCFL	发送取消已完成中断线	● 0b0: 选择中断线 0 ● 0b1: 选择中断线 1	RW
9	TCL	发送完成中断线	● 0b0: 选择中断线 0 ● 0b1: 选择中断线 1	RW
8	HPML	高优先级消息中断线	● 0b0: 选择中断线 0 ● 0b1: 选择中断线 1	RW
7	RF1LL	接收 FIFO1 消息丢失中断线	● 0b0: 选择中断线 0 ● 0b1: 选择中断线 1	RW
6	RF1FL	接收 FIFO1 已满中断线	● 0b0: 选择中断线 0 ● 0b1: 选择中断线 1	RW
5	RF1WL	接收 FIFO1 到达水位线中断线	● 0b0: 选择中断线 0 ● 0b1: 选择中断线 1	RW
4	RF1NL	接收 FIFO1 新消息中断线	● 0b0: 选择中断线 0 ● 0b1: 选择中断线 1	RW
3	RF0LL	接收 FIFO0 消息丢失中断线	● 0b0: 选择中断线 0 ● 0b1: 选择中断线 1	RW
2	RF0FL	接收 FIFO0 已满中断线	● 0b0: 选择中断线 0 ● 0b1: 选择中断线 1	RW
1	RF0WL	接收 FIFO0 到达水位线中断线	● 0b0: 选择中断线 0 ● 0b1: 选择中断线 1	RW
0	RF0NL	接收 FIFO0 新消息中断线	● 0b0: 选择中断线 0 ● 0b1: 选择中断线 1	RW

32.4.18 MCAN 中断线使能寄存器 (MCAN_ILE)

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 分别使能或禁止。

Offset	Bit Position																																	
0x5C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved																																EINT1	EINT0
Access																																	RW	RW

位/位域	标记	位名	功能描述	读写
31:2	Reserved	保留	-	-
1	EINT1	中断线 1 使能	● 0b0：禁止 ● 0b1：使能	RW
0	EINT0	中断线 0 使能	● 0b0：禁止 ● 0b1：使能	RW

32.4.19 MCAN 全局过滤器配置寄存器 (MCAN_GFC)

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，请参见“[标准消息 ID 过滤](#)”和“[扩展消息 ID 过滤](#)”。

Offset	Bit Position																																			
0x80	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reset	0x00000000																																			
Name	Reserved																												ANFS		ANFE		RRFS		RRFE	
Access																																				

位/位域	标记	位名	功能描述	读写
31:6	Reserved	保留	-	-
5:4	ANFS	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： 0b00：在接收 FIFO0 中接受不匹配的标准帧 0b01：在接收 FIFO1 中接受不匹配的标准帧 0b10：拒绝不匹配的标准帧 0b11：拒绝不匹配的标准帧	RWA
3:2	ANFE	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： 0b00：在接收 FIFO0 中接受不匹配的扩展帧 0b01：在接收 FIFO1 中接受不匹配的扩展帧 0b10：拒绝不匹配的扩展帧 0b11：拒绝不匹配的扩展帧	RWA
1	RRFS	拒绝远程标准帧	0b0：过滤 11 位标准 ID 的远程帧 0b1：拒绝所有 11 位标准 ID 的远程帧	RWA
0	RRFE	拒绝远程扩展帧	0b0：过滤 29 位扩展 ID 的远程帧 0b1：拒绝所有 29 位扩展 ID 的远程帧	RWA

32.4.20 MCAN 标准 ID 过滤器配置寄存器 (MCAN_SIDFC)

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请参见“[标准消息 ID 过滤](#)”。

Offset	Bit Position																															
0x84	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved								LSS								FLSSA[15:2]								Reserved							
Access									RWA								RWA															

位/位域	标记	位名	功能描述	读写
31:24	Reserved	保留	-	-
23:16	LSS	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
15:2	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, 请参见“ 消息 RAM 配置 ”)	RWA
1:0	Reserved	保留	-	-

32.4.21 MCAN 扩展 ID 过滤器配置寄存器 (MCAN_XIDFC)

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请参见“[扩展消息 ID 过滤](#)”。

Offset	Bit Position																																
0x88	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved									LSE								FLESA[15:2]															Reserved
Access										RWA								RWA															

位/位域	标记	位名	功能描述	读写
31:23	Reserved	保留	-	-
22:16	LSE	标准 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
15:2	FLESA[15:2]	扩展 ID 过滤器列表起始地址	设置扩展 ID 过滤器列表的起始地址 (32 位字地址, 请参见“ 消息 RAM 配置 ”)	RWA
1:0	Reserved	保留	-	-

32.4.22 MCAN 扩展 ID 与掩码寄存器 (MCAN_XIDAM)

Offset	Bit Position																															
0x90	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1FFFFFFF																															
Name	Reserved		EIDM																													
Access			RWA																													

位/位域	标记	位名	功能描述	读写
31:29	Reserved	保留	-	-
28:0	EIDM	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

32.4.23 MCAN 高优先级消息状态寄存器 (MCAN_HPMS)

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。

Offset	Bit Position																															
0x94	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																FIDX							MSI		BIDX						
																FLST																
Access																RO	RO						RO	RO		RO						

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15	FLST	过滤器列表	指示匹配的过滤器列表： 0b0：标准过滤器列表 0b1：扩展过滤器列表	RO
14:8	FIDX	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	RO
7:6	MSI	消息存储器指示符	0b00：未选择 FIFO 0b01：FIFO 消息丢失 0b10：消息存储在 FIFO0 中 0b11：消息存储在 FIFO1 中	RO
5:0	BIDX	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	RO

32.4.24 MCAN 新数据 1 寄存器 (MCAN_NDAT1)

Offset	Bit Position																															
0x98	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:0	NDn (n=31~0)	新数据	对应保存接收缓冲区 31~0 的新数据标志： ● 0b0：接收缓冲区未更新 ● 0b1：接收缓冲区通过新消息更新 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 0 无效。硬复位会将寄存器清零。  说明 每个 Bit 对应一个缓冲区。	RW

32.4.25 MCAN 新数据 2 寄存器 (MCAN_NDAT2)

Offset	Bit Position																															
0x9C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:0	NDn (n=63~32)	新数据	对应保存接收缓冲区 63~32 的新数据标志： ● 0b0：接收缓冲区未更新 ● 0b1：接收缓冲区通过新消息更新 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 0 无效。硬复位会将寄存器清零。  说明 每个 Bit 对应一个缓冲区。	RW

32.4.26 MCAN 接收 FIFO0 配置寄存器 (MCAN_RXF0C)

Offset	Bit Position																															
0xA0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	F0OM	F0WM							Reserved	F0S							F0SA[15:2]															Reserved
Access	RWA	RWA								RWA							RWA															

位/位域	标记	位名	功能描述	读写
31	F0OM	接收 FIFO0 工作模式	0b0: 接收 FIFO0 为阻塞模式 0b1: 接收 FIFO0 为覆盖模式	RWA
30:24	F0WM	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
23	Reserved	保留	-	-
22:16	F0S	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
15:2	F0SA[15:2]	接收 FIFO0 起始地址	消息 RAM 中接收 FIFO0 的起始地址 (32 位地址, 请参见“ 消息 RAM 配置 ”)	RWA
1:0	Reserved	保留	-	-

32.4.27 MCAN 接收 FIFO0 状态寄存器 (MCAN_RXF0S)

Offset	Bit Position																															
0xA4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved						RF0L	F0F	Reserved	F0PI						Reserved	F0GI						Reserved	F0FL								
Access							RO	RO		RO							RO							RO								

位/位域	标记	位名	功能描述	读写
31:26	Reserved	保留	-	-
25	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： 0b0：接收 FIFO0 无消息丢失 0b1：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  说明 当 MCAN_RXF0C.F0OM=1 时，覆盖最早的消息，不会置位该标志位。	RO
24	F0F	接收 FIFO0 满	0b0：接收 FIFO0 未满 0b1：接收 FIFO0 已满	RO
23:22	Reserved	保留	-	-
21:16	F0PI	接收 FIFO0 放入索引	接收 FIFO0 写入索引指针，有效范围是 0~63	RO
15:14	Reserved	保留	-	-
13:8	F0GI	接收 FIFO0 获取索引	接收 FIFO0 读取索引指针，有效范围是 0~63	RO
7	Reserved	保留	-	-
6:0	F0FL	接收 FIFO0 填充级别	接收 FIFO0 中存储的元素数量，有效范围是 0~64	RO

32.4.28 MCAN 接收 FIFO0 确认寄存器 (MCAN_RXF0A)

Offset	Bit Position																															
0xA8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								F0AI							
Access																																

位/位域	标记	位名	功能描述	读写
31:6	Reserved	保留	-	-
5:0	F0AI	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

32.4.29 MCAN 接收缓冲区配置寄存器 (MCAN_RXBC)

Offset	Bit Position																																
0xAC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																RBSA[15:2]																Reserved
Access																	RWA																

位/位域	标记	位名	功能描述	读写
31:16	Reserved	保留	-	-
15:2	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
1:0	Reserved	保留	-	-

32.4.30 MCAN 接收 FIFO1 配置寄存器 (MCAN_RXF1C)

Offset	Bit Position																															
0xB0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	F1OM	F1WM							Reserved	F1S							F1SA[15:2]															Reserved
Access	RWA	RWA								RWA							RWA															

位/位域	标记	位名	功能描述	读写
31	F1OM	接收 FIFO1 工作模式	0b0: 接收 FIFO1 为阻塞模式 0b1: 接收 FIFO1 为覆盖模式	RWA
30:24	F1WM	接收 FIFO1 水位线	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
23	Reserved	保留	-	-
22:16	F1S	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
15:2	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, 请参见“ 消息 RAM 配置 ”)	RWA
1:0	Reserved	保留	-	-

32.4.31 MCAN 接收 FIFO1 状态寄存器 (MCAN_RXF1S)

Offset	Bit Position																															
0xB4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	DMS	Reserved				RF1L	F1F	Reserved				F1PI				Reserved				F1GI				Reserved				F1FL				
Access	RO					RO	RO					RO								RO								RO				

位/位域	标记	位名	功能描述	读写
31:30	DMS	调试消息状态	0b00: 空闲状态, 等待接收调试消息 0b01: 已接收调试消息 A 0b10: 已接收调试消息 A、B 0b11: 已接收调试消息 A、B、C	RO
29:26	Reserved	保留	-	-
25	RF1L	接收 FIFO1 消息丢失	该位是中断标志 MCAN_IR.RF1L 的副本。当 MCAN_IR.RF1L 被复位时, 该位也被复位: 0b0: 接收 FIFO1 无消息丢失 0b1: 接收 FIFO1 有消息丢失; 对空间大小为 0 的接收 FIFO1 进行写操作时, 该位也会置位  说明 当 MCAN_RXF1C.F1OM=1 时, 覆盖最早的消息, 不会置位该标志位。	RO
24	F1F	接收 FIFO1 满	0b0: 接收 FIFO1 未滿 0b1: 接收 FIFO1 已滿	RO
23:22	Reserved	保留	-	-
21:16	F1PI	接收 FIFO1 放入索引	接收 FIFO1 写入索引指针, 有效范围是 0~63	RO
15:14	Reserved	保留	-	-
13:8	F1GI	接收 FIFO1 获取索引	接收 FIFO1 读取索引指针, 有效范围是 0~63	RO
7	Reserved	保留	-	-
6:0	F1FL	接收 FIFO1 填充级别	接收 FIFO1 中存储的元素数量, 有效范围是 0~64	RO

32.4.32 MCAN 接收 FIFO1 确认寄存器 (MCAN_RXF1A)

Offset	Bit Position																															
0xB8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																										F1AI					
Access																																

位/位域	标记	位名	功能描述	读写
31:6	Reserved	保留	-	-
5:0	F1AI	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

32.4.33 MCAN 接收缓冲区和 FIFO 元素大小配置寄存器 (MCAN_RXESC)

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。

Offset	Bit Position																															
0xBC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																						RWA RBDS		Reserved	RWA F1DS		Reserved	RWA F0DS			
Access																							RWA			RWA			RWA			

位/位域	标记	位名	功能描述	读写
31:11	Reserved	保留	-	-
10:8	RBDS	接收缓冲区数据场大小	● 0b000: 8 字节 ● 0b001: 12 字节 ● 0b010: 16 字节 ● 0b011: 20 字节 ● 0b100: 24 字节 ● 0b101: 32 字节 ● 0b110: 48 字节 ● 0b111: 64 字节	RWA
7	Reserved	保留	-	-
6:4	F1DS	接收 FIFO1 数据场大小	● 0b000: 8 字节 ● 0b001: 12 字节 ● 0b010: 16 字节 ● 0b011: 20 字节 ● 0b100: 24 字节 ● 0b101: 32 字节 ● 0b110: 48 字节 ● 0b111: 64 字节	RWA
3	Reserved	保留	-	-
2:0	F0DS	接收 FIFO0 数据场大小	● 0b000: 8 字节 ● 0b001: 12 字节 ● 0b010: 16 字节 ● 0b011: 20 字节 ● 0b100: 24 字节 ● 0b101: 32 字节 ● 0b110: 48 字节 ● 0b111: 64 字节	RWA

32.4.34 MCAN 发送缓冲区配置寄存器 (MCAN_TXBC)

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。

Offset	Bit Position																															
0xC0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved	TFQM	TFQS					Reserved	NDTB					TBSA[15:2]										Reserved								
Access		RWA	RWA						RWA					RWA																		

位/位域	标记	位名	功能描述	读写
31	Reserved	保留	-	-
30	TFQM	发送 FIFO/队列模式	● 0b0：发送 FIFO 模式 ● 0b1：发送队列模式	RWA
29:24	TFQS	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	RWA
23:22	Reserved	保留	-	-
21:16	NDTB	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	RWA
15:2	TBSA[15:2]	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 位地址，请参见“ 消息 RAM 配置 ”）	RWA
1:0	Reserved	保留	-	-

 **说明**

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。

32.4.35 MCAN 发送 FIFO/队列状态寄存器（MCAN_TXFQS）

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。

Offset	Bit Position																																
0xC4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved										TFQF	TFQPI				Reserved				TFGI				Reserved		TFFL							
Access											RO	RO								RO						RO							

位/位域	标记	位名	功能描述	读写
31:22	Reserved	保留	-	-
21	TFQF	发送 FIFO/队列满	0b0：发送 FIFO 或队列未满 0b1：发送 FIFO 或队列已满	RO
20:16	TFQPI	发送 FIFO/队列放入索引	发送 FIFO 或队列的放入索引指针，有效范围是 0~31	RO
15:13	Reserved	保留	-	-
12:8	TFGI	发送 FIFO 获取索引	发送 FIFO 的读取索引指针，有效范围是 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	RO
7:6	Reserved	保留	-	-
5:0	TFFL	发送 FIFO 空闲级别	从 TFGI 开始的连续空闲的发送 FIFO 元素数量，有效范围是 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	RO

 **说明**

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。

32.4.36 MCAN 发送缓冲区元素大小配置寄存器 (MCAN_TXESC)

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。

Offset	Bit Position																															
0xC8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															TBDS
Access																																

位/位域	标记	位名	功能描述	读写
31:3	Reserved	保留	-	-
2:0	TBDS	发送缓冲区数据场大小	0b000: 8 字节 0b001: 12 字节 0b010: 16 字节 0b011: 20 字节 0b100: 24 字节 0b101: 32 字节 0b110: 48 字节 0b111: 64 字节	RW

 **说明**
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。

32.4.37 MCAN 发送缓冲区请求挂起寄存器 (MCAN_TXBRP)

Offset	Bit Position																															
0xCC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
Access	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

位/位域	标记	位名	功能描述	读写
31:0	TRPn (n=31~0)	发送请求挂起标志	对应发送缓冲区 31~0 的发送请求挂起位： ● 0b0：无发送请求挂起 ● 0b1：发送请求挂起 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。 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（参见“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）。 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。 请求取消后，以下情况通过寄存器 MCAN_TXBCF 指示取消已完成： ● 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 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。  说明 每个 Bit 对应一个缓冲区。 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。	RO

32.4.38 MCAN 发送缓冲区添加请求寄存器 (MCAN_TXBAR)

Offset	Bit Position																															
0xD0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:0	ARn (n=31~0)	添加请求	对应发送缓冲区 31~0 的添加请求位： 写 1 置位相应的添加请求位，写 0 无效 读取： 0b0：未添加发送请求 0b1：已添加发送请求 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。  说明 每个 Bit 对应一个缓冲区。 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。	RW

32.4.39 MCAN 发送缓冲区取消请求寄存器 (MCAN_TXBCR)

Offset	Bit Position																															
0xD4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:0	CRn (n=31~0)	取消请求	对应发送缓冲区 31~0 的取消请求位： 写 1 置位相应的取消请求位，写 0 无效 读取： 0b0：无取消请求挂起 0b1：有取消请求挂起 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。  说明 每个 Bit 对应一个缓冲区。	RW

32.4.40 MCAN 发送缓冲区发送已发生寄存器 (MCAN_TXBTO)

Offset	Bit Position																																
0xD8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0	
Access	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

位/位域	标记	位名	功能描述	读写
31:0	TO _n (n=31~0)	发送已发生	对应发送缓冲区 31~0 的发送已发生标志位： ● 0b0：未进行发送 ● 0b1：已进行发送 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。  说明 每个 Bit 对应一个缓冲区。	RO

32.4.41 MCAN 发送缓冲区取消已完成寄存器 (MCAN_TXBCF)

Offset	Bit Position																																
0xDC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0	
Access	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

位/位域	标记	位名	功能描述	读写
31:0	CF _n (n=31~0)	取消已完成	对应发送缓冲区 31~0 的取消已完成标志位： ● 0b0：无发送缓冲区取消 ● 0b1：发送缓冲区已取消 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。  说明 每个 Bit 对应一个缓冲区。	RO

32.4.42 MCAN 发送缓冲区发送中断使能寄存器 (MCAN_TXBTIE)

Offset	Bit Position																															
0xE0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:0	TIE _n (n=31~0)	发送中断使能	对应发送缓冲区 31~0 的发送中断使能位： 0b0：禁止发送中断 0b1：使能发送中断  说明 每个 Bit 对应一个缓冲区。	RW

32.4.43 MCAN 发送缓冲区取消已完成中断使能寄存器 (MCAN_TXBCIE)

Offset	Bit Position																															
0xE4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:0	CFIE _n (n=31~0)	取消已完成中断使能	对应发送缓冲区 31~0 的取消已完成中断使能位： 0b0：禁止取消已完成中断 0b1：使能取消已完成中断  说明 每个 Bit 对应一个缓冲区。	RW

32.4.44 MCAN 发送事件 FIFO 配置寄存器 (MCAN_TXEFC)

Offset	Bit Position																															
0xF0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved	EFWM						Reserved	EFS						EFSA[15:2]														Reserved			
Access		RWA							RWA						RWA																	

位/位域	标记	位名	功能描述	读写
31:30	Reserved	保留	-	-
29:24	EFWM	发送事件 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
23:22	Reserved	保留	-	-
21:16	EFS	发送事件 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
15:2	EFSA[15:2]	发送事件 FIFO 起始地址	消息 RAM 中发送事件 FIFO 的起始地址 (32 位地址, 请参见“ 消息 RAM 配置 ”)	RWA
1:0	Reserved	保留	-	-

32.4.45 MCAN 发送事件 FIFO 状态寄存器 (MCAN_TXEFS)

Offset	Bit Position																																						
0xF4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
Reset	0x00000000																																						
Name	Reserved						TEFL		EFF		Reserved					EFPI					Reserved					EFGI					Reserved					EFFL			
Access							RO		TEFL		RO		EFF			RO					EFPI			RO					EFGI			RO					EFFL		

位/位域	标记	位名	功能描述	读写
31:26	Reserved	保留	-	-
25	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： 0b0：发送事件 FIFO 无元素丢失 0b1：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	RO
24	EFF	发送事件 FIFO 已满	0：发送事件 FIFO 未滿 1：发送事件 FIFO 已滿	RO
23:21	Reserved	保留	-	-
20:16	EFPI	发送事件 FIFO 放入索引	发送事件 FIFO 写入索引指针，有效范围是 0~31	RO
15:13	Reserved	保留	-	-
12:8	EFGI	发送事件 FIFO 获取索引	发送事件 FIFO 读取索引指针，有效范围是 0~31	RO
7:6	Reserved	保留	-	-
5:0	EFFL	发送事件 FIFO 填充级别	发送事件 FIFO 中存储的元素个数，有效范围是 0~32	RO

32.4.46 MCAN 发送事件 FIFO 确认寄存器 (MCAN_TXEFA)

Offset	Bit Position																															
0xF8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								EFAI							
Access																									RW							

位/位域	标记	位名	功能描述	读写
31:5	Reserved	保留	-	-
4:0	EFAI	发送事件 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

32.5 注意事项

32.5.1 CAN 总线抗干扰措施

CAN 总线广泛应用于汽车、工业控制等行业，如果 CAN 应用现场电磁环境比较恶劣，存在电路不平衡性、空间电磁场、电网进线等因素，会导致 CAN 总线因辐射、传导干扰而产生大量通信噪声，致使总线错误帧增加、重发频繁，正确数据不能及时到达等情况，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

32.5.2 CAN 控制器噪声制约

在 CAN 总线网络中应确保通信的位时间满足标准协议的要求，若引入不满足位时间宽度的噪声干扰，可能引起 MCAN 控制器异常动作。

33 加密协处理模块 (CPM)

33.1 简介

加密协处理模块 (CPM) 包括 AES 加解密算法处理器, HASH 安全散列算法, TRNG 真随机数发生器三个子模块。

- AES 加解密算法处理器遵循美国国家标准技术研究所 (NIST) 在 2000 年 10 月 2 日正式宣布的新的数据加密标准, 分组长度固定为 128 位, 而密钥长度支持 128/192/256 位。
- HASH 安全散列算法是 SHA-2 版本的 SHA-256 (Secure Hash Algorithm), 符合美国国家标准和技术局发布的国家标准“FIPS PUB 180-3”, 可以对长度不超过 2^{64} 位的消息产生 256 位的消息摘要输出。
- TRNG 真随机数发生器是以连续模拟噪声为基础的随机数发生器, 提供 64-bit 随机数。

33.2 功能说明

33.2.1 加解密算法处理器 (AES)

33.2.1.1 算法简介

AES (The Advanced Encryption Standard) 是美国国家标准技术研究所 (NIST) 在 2000 年 10 月 2 日正式宣布的新的数据加密标准。

AES 的分组长度固定为 128 位，而密钥长度支持 128、192 和 256 位。对于加密来说，其输入是一个明文分組和一个密钥，输出是一个密文分組；对解密而言，输入是一个密文分組和一个密钥，而输出是一个明文分組。此过程如下所示：

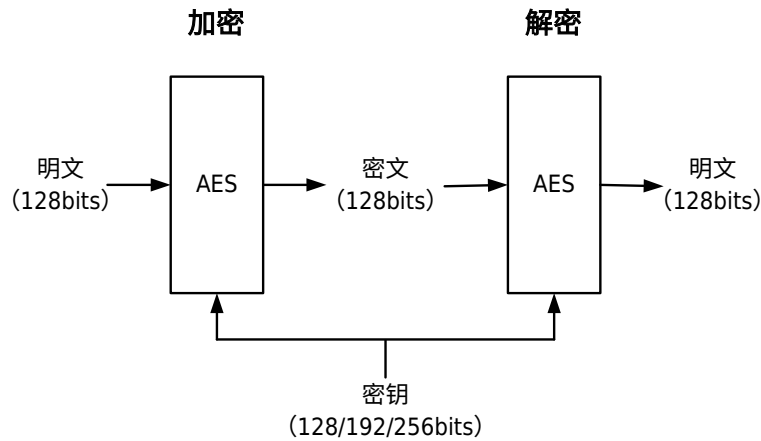


图 33-1 AES 的加解密示意图

AES 算法处理的基本单位是字节，128 位信息被分成 16 个字节，按顺序复制到一个 4×4 的矩阵中，称为状态 (state)，AES 的所有变换都是基于状态矩阵的变换，该矩阵上保存着计算的中间结果。

AES 是一个密钥迭代分組密码，包含了轮变换对状态的重复作用。AES 的轮变换由四个操作组成：SubBytes、ShiftRows、MixColumns、AddRoundKey。其中，SubBytes 包括求每个字节在 GF (2⁸) 中的模逆元和一个仿射变换；ShiftRows 是一个字节换位，它将状态中的行按照不同的偏移量进行循环移位；MixColumns 对状态各列进行线性变换；AddRoundKey，状态中的各字节与轮密钥进行逐位异或操作。AES 的加密流程如下所示：

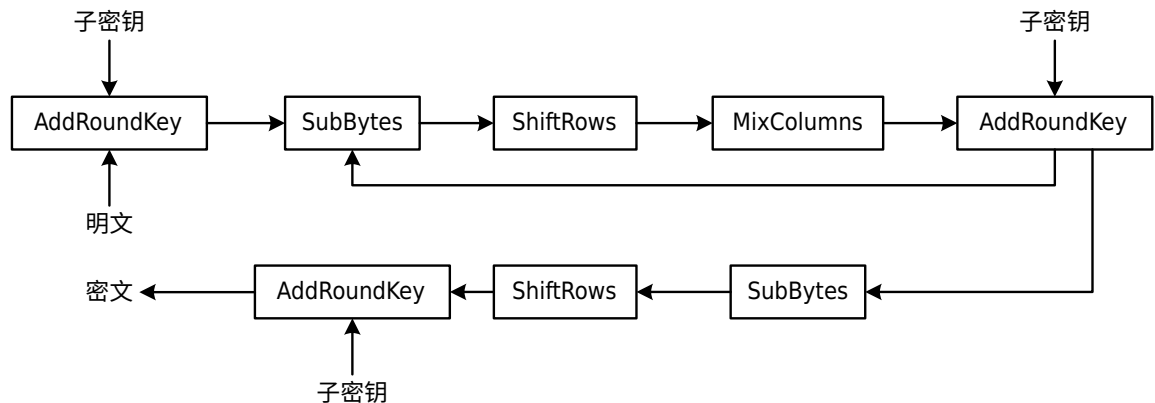


图 33-2 AES 的加密流程图

图中所用子密钥需要由初始密钥扩展而来，且密钥的扩展过程和加密过程是同步进行的。由于明文固定为 128 位，加密过程运行的轮数就取决于密钥的长度。比如，密钥为 128 位时，运行轮数为 10 轮；密钥为 192 位时，运行轮数为 12 轮；密钥为 256 位时，运行轮数为 14 轮。除了最后一轮缺少 MixColumns 变换，其余各轮均进行完整的轮变换操作。

解密流程与加密流程有所区别，首先必须完成所有密钥的扩展，解密过程从扩展的最后一轮子密钥往回使用；然后是轮变换的四个操作变成了相应的逆运算：InvSubBytes、InvShiftRows、InvMixColumns、AddRoundKey。InvSubBytes 中的模逆运算仍然保持，但仿射变换改为逆变换；InvShiftRows 和 InvMixColumns 变成相应的逆变换；AddRoundKey 保持不变。

直接解密流程的轮变换对四个操作的调用顺序为：InvShiftRows、InvSubBytes、AddRoundKey、InvMixColumns，与加密流程的调用顺序不一致，但使用的密钥与加密流程一致；等价解密流程的轮变换对四个操作的调用顺序为：InvSubBytes、InvShiftRows、InvMixColumns、AddRoundKey，与加密流程的调用顺序完全一致，只是每一轮的子密钥需要进行 InvMixColumns 运算。

关于详细的算法表述，可以参见标准《FIPS PUB 197》。

33.2.1.2 主要特性

- 执行 AES 算法标准的加密流程和解密流程，其执行结果完全符合《FIPS PUB 197》对算法原理的描述
- 支持 128、192 和 256 位密钥

33.2.1.3 运行时间说明

本模块从启动一次运算（AES_CR.START 写入 1）到该次运算结束（AES_CR.START 恢复到 0）所需时间如下表所示：

表 33-1 AES 加解密运行时间

操作	128 位密钥	192 位密钥	256 位密钥
加密	220 cycles	260 cycles	300 cycles
解密	290 cycles	332 cycles	398 cycles

33.2.1.4 操作示例

本模块共有两个功能：加密、解密。对两个功能的操作有一些共同的特点，下面先介绍其共同点，再分别介绍每个功能的标准操作流程。

33.2.1.4.1 加解密操作共同点

1. 在 AES 加解密过程中，数据寄存器会改变，如果下次运算的被操作数据就是本次运算的结果，就无需重新写入数据。
2. 支持 128、192 和 256 位密钥，128 位密钥写入偏移地址 0x20~0x2C，192 位密钥写入偏移地址 0x20~0x34，256 位密钥写入偏移地址 0x20~0x3C。
3. 判断模块运算结束的方法：不断读取 AES_CR.START，如果其值变为 0，则表示运算结束。

33.2.1.4.2 加密操作流程

操作步骤

- 步骤 1. 将待加密的 128 位数据写入数据寄存器（AES_DR）中。
- 步骤 2. 将加密密钥写入密钥寄存器（AES_KR）中。
- 步骤 3. 根据密钥长度设置 AES_CR.KEYSIZE。
- 步骤 4. 将 AES_CR.MODE 设置为 0，启动加密模式。
- 步骤 5. 向控制寄存器中的 AES_CR.START 写入 1，启动模块进行运算。
步骤 3~5 可以同时进行。
- 步骤 6. 查询 AES_CR.START，等待 AES_CR.START 的值恢复为 0，模块运算结束。
- 步骤 7. 读取数据寄存器（AES_DR），获得 128 位密文。

33.2.1.4.3 解密操作流程

操作步骤

- 步骤 1. 将待解密的 128 位数据写入数据寄存器 (AES_DR) 中。
- 步骤 2. 将解密密钥写入密钥寄存器 (AES_KR) 中。
- 步骤 3. 根据密钥长度设置 AES_CR.KEYSIZE。
- 步骤 4. 将 AES_CR.MODE 设置为 1，启动解密模式。
- 步骤 5. 向控制寄存器中的 AES_CR.START 写入 1，启动模块进行运算。
步骤 3~5 可以同时进行。
- 步骤 6. 查询 AES_CR.START，等待 AES_CR.START 的值恢复为 0，模块运算结束。
- 步骤 7. 读取数据寄存器 (AES_DR)，获得 128 位明文。

33.2.1.4.4 数据示例

128 位密钥

128 位明文: 0xFFEEDDCCBBAA99887766554433221100
128 位密钥: 0x0F0E0D0C0B0A09080706050403020100
128 位密文: 0x5AC5B47080B7CDD830047B6AD8E0C469

表 33-2 128 位操作寄存器示例

加密前			
寄存器	值 (密钥)	寄存器	值 (明文)
AES_KR0	0x03020100	AES_DR0	0x33221100
AES_KR1	0x07060504	AES_DR1	0x77665544
AES_KR2	0x0B0A0908	AES_DR2	0xBBAA9988
AES_KR3	0x0F0E0D0C	AES_DR3	0xFFEEDDCC
加密后			
寄存器	值 (密钥)	寄存器	值 (密文)
AES_KR0	0x03020100	AES_DR0	0xD8E0C469
AES_KR1	0x07060504	AES_DR1	0x30047B6A
AES_KR2	0x0B0A0908	AES_DR2	0x80B7CDD8
AES_KR3	0x0F0E0D0C	AES_DR3	0x5AC5B470

192 位密钥

128 位明文：0xFFEEDDCCBBAA99887766554433221100

192 位密钥：0x17161514131211100F0E0D0C0B0A09080706050403020100

128 位密文：0x91710DECA070AF6EE0DF4C86A47CA9DD

表 33-3 192 位操作寄存器示例

加密前			
寄存器	值（密钥）	寄存器	值（明文）
AES_KR0	0x03020100	AES_DR0	0x33221100
AES_KR1	0x07060504	AES_DR1	0x77665544
AES_KR2	0x0B0A0908	AES_DR2	0xBBAA9988
AES_KR3	0x0F0E0D0C	AES_DR3	0xFFEEDDCC
AES_KR4	0x13121110		
AES_KR5	0x17161514		
加密后			
寄存器	值（密钥）	寄存器	值（密文）
AES_KR0	0x03020100	AES_DR0	0xA47CA9DD
AES_KR1	0x07060504	AES_DR1	0xE0DF4C86
AES_KR2	0x0B0A0908	AES_DR2	0xA070AF6E
AES_KR3	0x0F0E0D0C	AES_DR3	0x91710DEC
AES_KR4	0x13121110		
AES_KR5	0x17161514		

256 位密钥

128 位明文：0xFFEEDDCCBBAA99887766554433221100

256 位密钥：0x1F1E1D1C1B1A191817161514131211100F0E0D0C0B0A09080706050403020100

128 位密文：0x8960494B9049FCAEBF456751CAB7A28E

表 33-4 256 位操作寄存器示例

加密前			
寄存器	值（密钥）	寄存器	值（明文）
AES_KR0	0x03020100	AES_DR0	0x33221100
AES_KR1	0x07060504	AES_DR1	0x77665544
AES_KR2	0x0B0A0908	AES_DR2	0xBBAA9988
AES_KR3	0x0F0E0D0C	AES_DR3	0xFFEEDDCC
AES_KR4	0x13121110		
AES_KR5	0x17161514		
AES_KR6	0x1B1A1918		
AES_KR7	0x1F1E1D1C		
加密后			
寄存器	值（密钥）	寄存器	值（密文）
AES_KR0	0x03020100	AES_DR0	0xCAB7A28E
AES_KR1	0x07060504	AES_DR1	0xBF456751
AES_KR2	0x0B0A0908	AES_DR2	0x9049FCAE
AES_KR3	0x0F0E0D0C	AES_DR3	0x8960494B
AES_KR4	0x13121110		
AES_KR5	0x17161514		
AES_KR6	0x1B1A1918		
AES_KR7	0x1F1E1D1C		

33.2.2 安全散列算法（HASH）

33.2.2.1 算法简介

安全散列算法的步骤如下：

首先填充消息使其长度恰好为一个比 512 的倍数仅小 64 位的数。填充方法是附一个 1 在消息后面，后接所要求的多个 0，然后在其后附上 64 位的消息长度（填充前），使消息长度恰好是 512 位的整数倍。

其次将 A、B、C、D、E、F、G、H 8 个 32 位变量，用十六进制初始化。然后开始算法的主循环，一次处理 512 位消息，循环次数是消息中 512 位分组的数目。

主循环一共进行 64 次操作，此操作称为压缩函数。每次操作都包含移位、循环移位、逻辑运算、模 232 加等，运算的过程如下图所示，最后的输出由 A、B、C、D、E、F、G、H 级联而成。其中 W_t 为由 512 位消息得到的第 t 步所用的临时值， K_t 为第 t 步所用的常数值， t ($0 \leq t \leq 63$) 是 64 步循环中的一步。

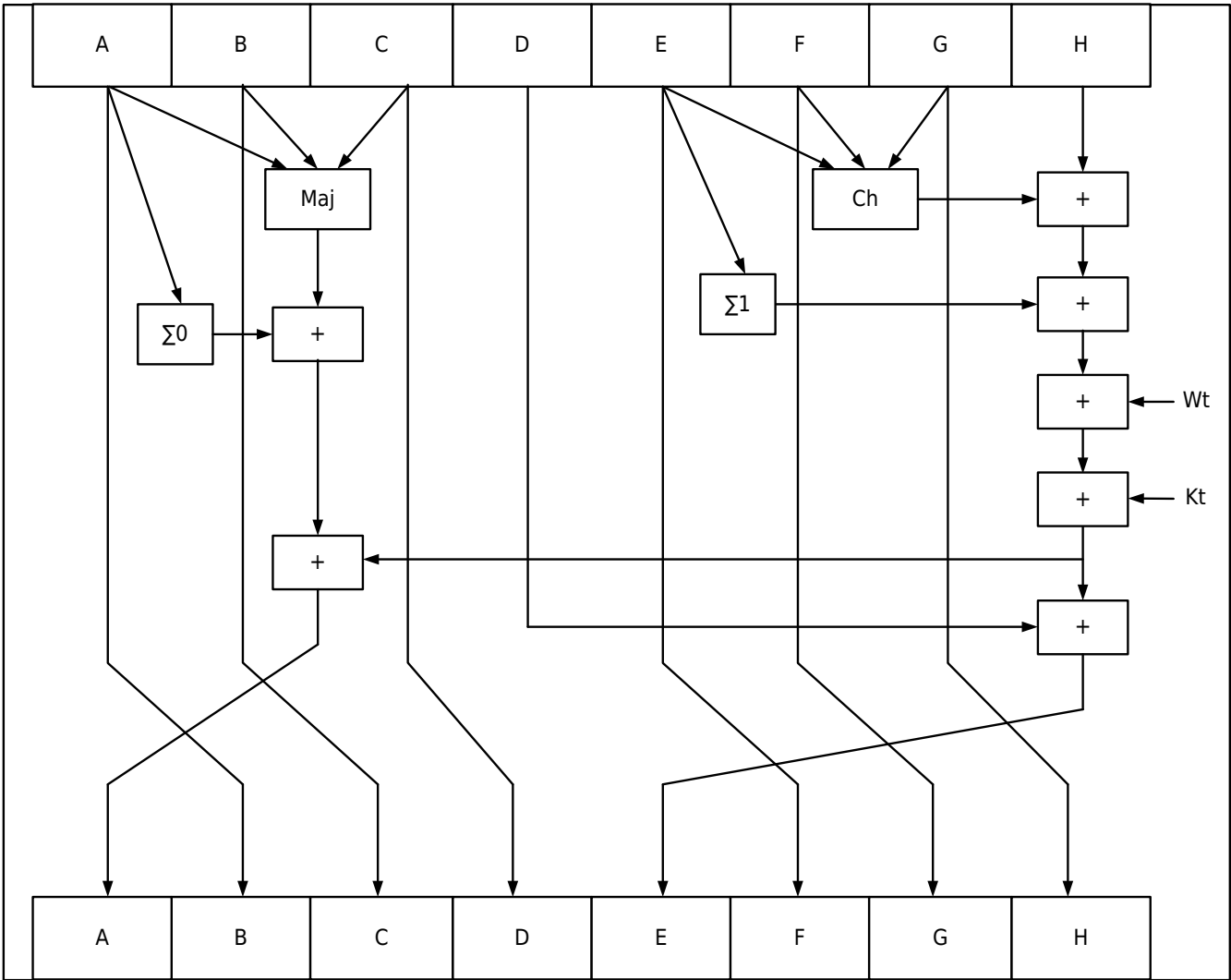


图 33-3 HASH 算法流程图

33.2.2.2 操作流程

HASH 模块的操作流程如下：

- 软件操作流程：
 - 步骤 1.** 软件将原始数据按照算法规则进行填充，并且将填充后的消息按 512 位进行分组。
 - 步骤 2.** 将数据写入数据寄存器 HASH_DR 中。
 - 步骤 3.** 如果本次运算为消息分组的第一组数据，设置 HASH_CR.FST_GRP=1。
 - 步骤 4.** 设置 HASH_CR.START=1，启动本模块进行运算。



说明

步骤 3 和 4 可以同时进行。

- 步骤 5.** 判断本模块本次运算是否完成，通过以下方法：
不断读取 HASH_CR.START 直到读到该位为 0 即表示运算完成。
- 步骤 6.** 如果本次运算不是消息分组的最后一组数据，返回到步骤 2。
- 步骤 7.** 如果本次运算为消息分组的最后一组数据，读取摘要寄存器 HASH_HR 获得本次运算的结果。如需再次进行运算则返回到步骤 1。

33.2.2.3 消息填充

SHA-256 的填充分组处理步骤如下：

步骤 1. 原始消息分组

将原始消息以 512-bit 的大小分为 L 组。令原始消息总的 bit 个数为 I。如果 $I \% 512 < 448$ ，那么分组数 L 为 $I / 512$ ；如果 $I \% 512 \geq 448$ ，那么分组数 L 为 $I / 512 + 1$ 。

步骤 2. 添加长度

a. 添加填充位：

在消息分组的第 $I / 512$ 组的最后添加填充位：一个 1 和若干个 0，0 的个数可以为零。如果 $I \% 512 < 448$ ，填充使得数据位的长度满足长度为 $448 \bmod 512$ （后 64-bit 留做原始消息长度的表示）；如果 $I \% 512 \geq 448$ ，用一个 1 和若干个 0 把第 $I / 512$ 组的 512-bit 的数据块填满，并将第 L ($L = I / 512 + 1$) 组的前 448 位填 0。

b. 添加原始消息长度：

一个 64-bit 块，表示原始消息长度，为 64-bit 无符号整数。将原始消息长度添加在第 L 个分组的最后 64-bit 处。

举例说明填充分组的过程如下：

● 示例 1：

原始消息为字符串“abcde”，用二进制位串表示其 ASCII 码为：“01100001 01100010 01100011 01100100 01100101”，添加长度的步骤如下：

1. 添加“1”。填充后的消息为“01100001 01100010 01100011 01100100 01100101 1”。
2. 添加“0”。因为原始消息长度为 40-bit，所以要添加的 0 的个数为 $512 - 64 - 40 - 1 = 407$ 。

填充后的消息变为（十六进制）：

```
61626364 65800000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000
```

3. 添加原始消息长度。原始消息长度 40-bit，用两个 32-bit 字表达为（十六进制）：00000000 00000028。

填充后的消息变为（十六进制）：

```
61626364 65800000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000028
```

● 示例 2：

原始消息为字符串“abcdbcdecdefdefgefghfghighijhijkijklmklmnlmnomnopnopq”。每个字符都可通过其 ASCII 码转换为 8-bit，所以消息的长度为 $l = 56 * 8 = 448$ 。

1. 添加“1”和“0”。填充后的消息（十六进制）为第一个消息块：

```
61626364 62636465 63646566 64656667
65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F
6D6E6F70 6E6F7071 80000000 00000000
```

2. 添加原始消息长度。原始消息长度 448-bit，用两个 32-bit 字表达为（十六进制）：00000000 000001C0。
- 填充后的消息（十六进制）为第二个消息块：

```
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 000001C0
```

33.2.3 真随机数发生器 (TRNG)

33.2.3.1 模块框图

TRNG 模块提供 1 个真随机数发生器，产生 1 个 64 位随机数。

TRNG 的系统框图如下图所示。其中随机数发生器是一个模拟随机数振荡器电路，用于得到随机性的噪声；算法模块捕捉随机噪声并把结果保存至数据模块、并通过总线输出；控制模块控制 TRNG 的模式和启动。

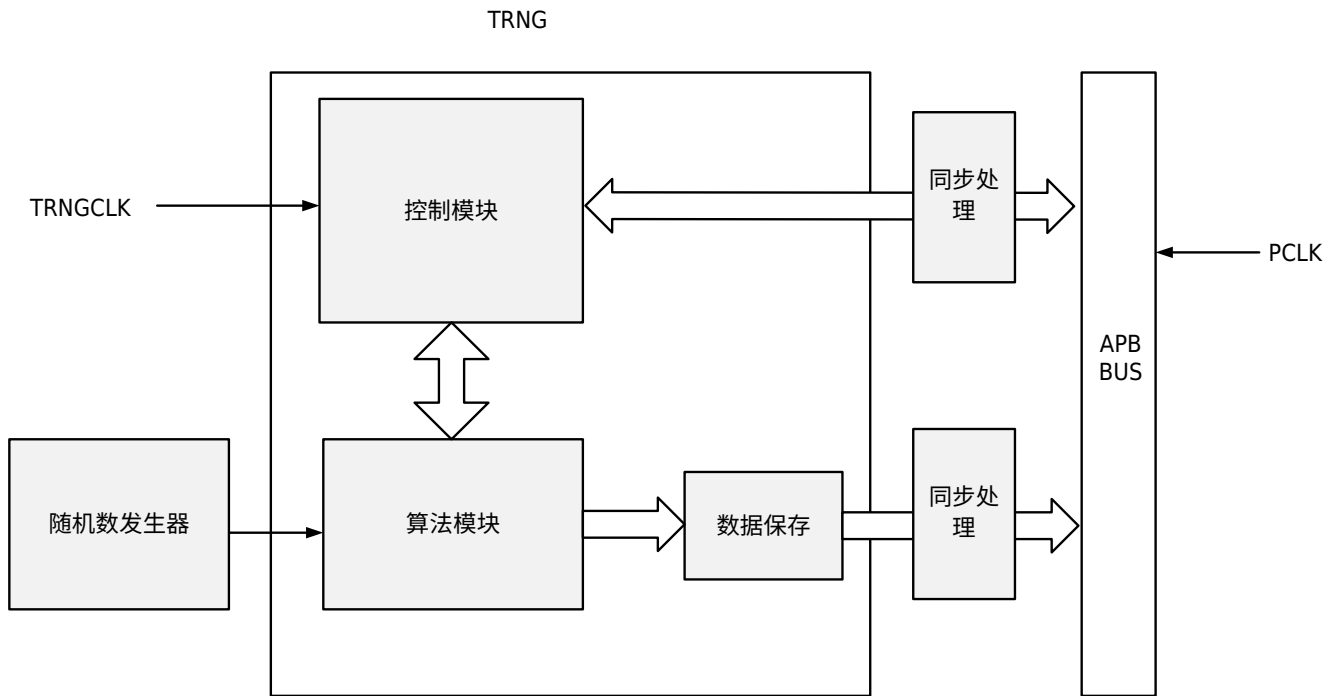


图 33-4 TRNG 系统框图

33.2.3.2 操作流程

真随机数生成流程如下：

- 步骤 1. 打开随机数发生器电路（设定 TRNG_CR.EN 位为 1）。
- 步骤 2. 配置随机数生成模式（设定 TRNG_MR）。
- 步骤 3. 启动随机数生成（设定 TRNG_CR.RUN 位为 1）。
- 步骤 4. 读取随机数（读取 TRNG_DR）。
- 步骤 5. 关闭随机数发生器电路（设定 TRNG_CR.EN 位为 0）。

33.2.3.3 中断和事件

当随机数生成结束后，寄存器位 TRNG_CR.RUN 硬件清零并产生随机数生成完中断请求（TRNG_END）。随机数生成结束同时也产生事件输出，可以触发其他模块联动。

33.3 寄存器

33.3.1 寄存器总表

AES 基地址：0x40008000

表 33-5 加解密算法处理器 (AES) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	AES_CR	控制寄存器	32
0x10+0x4*n	AES_DRn	数据寄存器 n (n=0~3)	32
0x20+0x4*n	AES_KRn	密钥寄存器 n (n=0~7)	32

HASH 基地址：0x40008400

表 33-6 安全散列算法 (HASH) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x0000	HASH_CR	HASH 控制寄存器	32
0x0010+0x4*(7-n)	HASH_HRn	HASH 摘要寄存器 n (n=0~7)	32
0x0040+0x4*(15-n)	HASH_DRn	HASH 数据寄存器 n (n=0~15)	32

TRNG 基地址：0x40042000

表 33-7 真随机数发生器 (TRNG) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x0000	TRNG_CR	TRNG 控制寄存器	32
0x0004	TRNG_MR	TRNG 模式寄存器	32
0x000C+0x4*n	TRNG_DRn	TRNG 数据寄存器 n (n=0~1)	32

33.3.2 AES 寄存器

33.3.2.1 控制寄存器（AES_CR）

Offset	Bit Position																															
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																										KEYSIZE		Reserved	MODE	START	
Access																											RW			RW	RW	

位/位域	标记	位名	功能描述	读写
31:5	Reserved	保留	-	-
4:3	KEYSIZE	密钥长度	● 0b00：密钥长度为 128 位 ● 0b01：密钥长度为 192 位 ● 0b10：密钥长度为 256 位 ● 0b11：密钥长度为 128 位	RW
2	Reserved	保留	-	-
1	MODE	运算模式	● 0b0：加密运算 ● 0b1：解密运算	RW
0	START	运算启动控制	● 0b0：运算结束或未启动 ● 0b1：启动运算	RW

 说明

- AES_CR.START 位的操作方法是：软件对本位写入 1 后，本模块将启动运行，本次运行结束后本模块硬件会自动将本位清 0，软件查询到本位为 0 即表示本次运行完成。
- 对本寄存器的写入操作只能在本模块不处于运算状态时（即 AES_CR.START=0 时）才能进行，否则硬件将自动忽略写操作。读操作则不受此限制。

33.3.2.2 数据寄存器（AES_DRn）（n=0~3）

偏移地址（Offset）：0x10+0x4*n

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	DATA																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	DATA	明文/密文数据	存放 AES 算法的 128 比特明文/密文	RW

 **说明**

- 数据寄存器由四个 32 位的寄存器组成 128 位数据，用于在模块运算前存放需要被加密的明文或者需要被解密的密文，并且运算完成后存放加密后的密文或者解密后的明文。

加密运算		解密运算	
运算前	运算后	运算前	运算后
128 位明文	128 位密文	128 位密文	128 位明文

四个 32 位寄存器连接在一起组成一个 128 位的数据，读写操作时需要分别对四个寄存器进行操作。数据寄存器对应的操作顺序如下（以数据 0xFFEEDDCCBBAA99887766554433221100 为例）：

- ▶ AES_DR0 寄存器填入数据：0x33221100
- ▶ AES_DR1 寄存器填入数据：0x77665544
- ▶ AES_DR2 寄存器填入数据：0xBBAA9988
- ▶ AES_DR3 寄存器填入数据：0xFFEEDDCC

- 对于本寄存器的写入只能在本模块没有处于运算状态时（即 AES_CR.START=0 时）才能进行，否则硬件将自动忽略对本寄存器的写操作。
- 对于本寄存器的读取只能在本模块没有处于运算状态时（即 AES_CR.START=0 时）才能进行，否则对本寄存器的读取将得到全 0。

33.3.2.3 密钥寄存器（AES_KRn）（n=0~7）

偏移地址（Offset）：0x20+0x4*n

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	KEY																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	KEY	AES 密钥	存放 AES 算法的 128（KR0~KR3）、192（KR0~KR5）、256（KR0~KR7）位密钥	RW



说明

- 密钥寄存器由 8 个 32 位的寄存器组成，存放输入的初始密钥。写操作时需要分别对 8 个 32 位的寄存器进行操作。对应的操作顺序分别参考下述举例。
128 位密钥举例：0x0F0E0D0C_0B0A0908_07060504_03020100
 - ▶ AES_KR0 寄存器填入数据：0x03020100
 - ▶ AES_KR1 寄存器填入数据：0x07060504
 - ▶ AES_KR2 寄存器填入数据：0x0B0A0908
 - ▶ AES_KR3 寄存器填入数据：0x0F0E0D0C192 位密钥举例：0x17161514_13121110_0F0E0D0C_0B0A0908_07060504_03020100
 - ▶ AES_KR0 寄存器填入数据：0x03020100
 - ▶ AES_KR1 寄存器填入数据：0x07060504
 - ▶ AES_KR2 寄存器填入数据：0x0B0A0908
 - ▶ AES_KR3 寄存器填入数据：0x0F0E0D0C
 - ▶ AES_KR4 寄存器填入数据：0x13121110
 - ▶ AES_KR5 寄存器填入数据：0x17161514256 位密钥举例：
0x1F1E1D1C_1B1A1918_17161514_13121110_0F0E0D0C_0B0A0908_07060504_03020100
 - ▶ AES_KR0 寄存器填入数据：0x03020100
 - ▶ AES_KR1 寄存器填入数据：0x07060504
 - ▶ AES_KR2 寄存器填入数据：0x0B0A0908
 - ▶ AES_KR3 寄存器填入数据：0x0F0E0D0C
 - ▶ AES_KR4 寄存器填入数据：0x13121110
 - ▶ AES_KR5 寄存器填入数据：0x17161514
 - ▶ AES_KR6 寄存器填入数据：0x1B1A1918
 - ▶ AES_KR7 寄存器填入数据：0x1F1E1D1C
- 对于本寄存器的写入只能在本模块没有处于运算状态时（即 AES_CR.START=0 时）才能进行，否则硬件将自动忽略对本寄存器的写操作。
- 对于本寄存器的读取只能在本模块没有处于运算状态时（即 AES_CR.START=0 时）才能进行，否则对本寄存器的读取将得到全 0。

33.3.3 HASH 寄存器

33.3.3.1 HASH 控制寄存器（HASH_CR）

Offset	Bit Position																															
0x0000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																														FST_GRP	START
Access																															RW	RW

位/位域	标记	位名	功能描述	读写
31:2	Reserved	保留	-	-
1	FST_GRP	密钥或消息分组的第一组	● 0b0：密钥或消息非第一组 ● 0b1：密钥或消息第一组 当第一组密钥或消息转换完成后，该位自动清“0”	RW
0	START	启动	0b0：本模块运算结束或未被启动 0b1：启动本模块进行运算	RW



说明

- START 位的操作方法是：软件对该位写入 1 后，本模块将启动运行；本次运行结束后硬件会自动将该位清 0；软件查询到该位为 0 即表示本次运行完成。
- 对该寄存器的写入操作只能在本模块非处于运算状态时（即 START 位为 0 时）才能进行，否则硬件将自动忽略写操作。读操作则不受此限制。

33.3.3.2 HASH 摘要寄存器 n（HASH_HRn）（n=0~7）

8 个 HASH 摘要寄存器 HASH_HR0~HASH_HR7，对应 256 位 HASH 摘要。HASH_HR0 对应摘要 Bit31~Bit0，HASH_HR1 对应摘要 Bit63~Bit32，依次类推。

偏移地址（Offset）：0x0010+0x4*(7-n)

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	HASH																															
Access	RO																															

位/位域	标记	位名	功能描述	读写
31:0	HASH	摘要值	在模块运算完成后通过读取本寄存器得到消息摘要	RO

说明

- 本寄存器由 8 个 32 位寄存器拼接而成。访问时依次对 8 个 32 位寄存器进行操作，低地址对应的 32 位寄存器存放消息摘要的高字。
- 对于本寄存器的读取只能在本模块非处于运算状态时（HASH_CR.START=0）才能进行，否则对本寄存器的读取将得到全 0。

33.3.3.3 HASH 数据寄存器 n (HASH_DRn) (n=0~15)

16 个 HASH 数据寄存器 HASH_DR0~HASH_DR15, 对应 512 位 HASH 数据。HASH_DR0 对应数据 Bit31~Bit0, HASH_DR1 对应数据 Bit63~Bit32, 依次类推。

偏移地址 (Offset) : 0x0040+0x4*(15-n)

Offset	Bit Position																																					
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	DATA																																					
Access	RW																																					

位/位域	标记	位名	功能描述	读写
31:0	DATA	数据寄存器	用于在模块运算前写入消息	RW

 **说明**

- 本寄存器由 16 个 32 位寄存器拼接而成。访问时依次对 16 个 32 位寄存器进行操作, 低地址对应的 32 位寄存器存放数据的高字。
- 对于本寄存器的写入只能在本模块非处于运算状态时 (HASH_CR.START=0) 才能进行, 否则硬件将自动忽略对本寄存器的写操作。
- 对本寄存器的读取将总是得到全 0。

33.3.4 TRNG 寄存器

33.3.4.1 TRNG 控制寄存器 (TRNG_CR)

Offset	Bit Position																																	
0x0000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved																																RW	RUN
Access																																	RW	RW

位/位域	标记	位名	功能描述	读写
31:2	Reserved	保留	-	-
1	RUN	随机数运算开始	● 0b0: 随机数运算停止 ● 0b1: 随机数运算开始  说明 软件写 1, 产生新的 64 位随机数; 运行完毕后, 硬件清零。	RW
0	EN	模拟振荡器使能	● 0b0: 关闭模拟随机数发生器电路 ● 0b1: 打开模拟随机数发生器电路	RW

33.3.4.2 TRNG 模式寄存器 (TRNG_MR)

Offset	Bit Position																															
0x0004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000012																															
Name	Reserved																										CNT		Reserved	LOAD		
Access																											RW			RW		

位/位域	标记	位名	功能描述	读写
31:5	Reserved	保留	-	-
4:2	CNT	移位次数控制位	捕捉随机噪声时，移位次数控制位 0b000~0b010：禁止设置 0b011：移位 32 次 0b100：移位 64 次 0b101：移位 128 次 0b110：移位 256 次 0b111：禁止设置	RW
1	Reserved	保留	读出为 1，写入时必须写 1	-
0	LOAD	装载控制位	随机数生成前，数据寄存器是否从随机数发生器装载新的初始值 0b0：不装载新的初始值 0b1：装载新的初始值	RW

33.3.4.3 TRNG 数据寄存器 n (TRNG_DRn) (n=0~1)

2 个 TRNG 数据寄存器 TRNG_DR0~TRNG_DR1，对应 64 位 TRNG 数据。TRNG_DR0 对应数据 Bit31~Bit0，TRNG_DR1 对应数据 Bit63~Bit32。

偏移地址 (Offset)：0x000C+0x4*n

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	DATA																															
Access	RO																															

位/位域	标记	位名	功能描述	读写
31:0	DATA	随机数	随机数	RO



说明

TRNG_DR 寄存器由 2 个 32 位寄存器拼接而成。

34 数据计算单元 (DCU)

34.1 简介

数据计算单元 (Data Computing Unit) 是一个不借助于 CPU 的简单处理数据的模块。每个 DCU 单元具有 3 个数据寄存器，能够进行 2 个数据的加减运算和比较大小以及窗口比较功能。还可以通过定时器触发为数模转换模块 (DAC) 提供连续变化的数字量以产生三角波和锯齿波输出。本产品搭载 4 个 DCU 单元。

34.2 主要特性

- 可进行 4 种数据处理：2 个数据的加法，减法，比较以及 3 个数据窗口比较
- 加法和减法对 DATA0 和 DATA1 寄存器中的数据进行运算，结果存放在 DATA0 中；同时自动将结果进行一次减半操作，减半的结果存放到 DATA1
- 加法和减法可以选择写寄存器后计算或由其他外围电路事件触发进行运算
- 比较模式可以对 DATA0 和 DATA1 寄存器之间，以及 DATA0 和 DATA2 寄存器之间的 2 个数据进行比较，可以分别选择当满足大于、小于、等于时产生中断和标志位
- 比较模式可以用来进行窗口比较，即分别设置 DATA1 和 DATA2 为窗口的上下限，根据 DATA0 和 DATA1 及 DATA0 和 DATA2 的比较结果判断 DATA0 在窗口内或窗口外
- 能够由其他外围电路事件触发进行运算，并根据运算结果产生多种中断和事件信号。当其他具有硬件触发启动功能的外围电路选择 DCU 作为触发源时，由 DCU 产生事件信号来启动外围电路开始动作
- 在三角波输出模式下，通过定时器定时触发产生连续递增和递减的数字量给数模转换模块 (DAC)，后者能够将这些数字量通过数模转换转换为三角波输出
- 在递增锯齿波输出模式下，通过定时器定时触发产生连续递增的数字量给数模转换模块 (DAC)，后者能够将这些数字量通过数模转换转换为递增的锯齿波输出
- 在递减锯齿波输出模式下，通过定时器定时触发产生连续递减的数字量给数模转换模块 (DAC)，后者能够将这些数字量通过数模转换转换为递减的锯齿波输出

34.3 功能说明

34.3.1 加法模式

加法模式计算 DATA0 和 DATA1 的和，其中 DATA0 作为被加数，DATA1 作为加数。每次写 DATA1 寄存器进行一次 DATA0+DATA1 的加法运算，加法运算结果存入 DATA0，加法运算结果除以 2 存入 DATA2。当 DATA0+DATA1 结果大于 0xFF (8-bit 模式) 或 0xFFFF (16-bit 模式) 或 0xFFFFFFFF (32-bit 模式) 时产生中断和标志位。

加法模式应用举例：

- 步骤 1.** 控制寄存器 DCU_CTL 选择加法模式，数据宽度为 16-bit；
- 步骤 2.** DCU_INTEVTSEL 选择中断产生条件；
- 步骤 3.** 分别在 DATA0 和 DATA1 中写入 0xFF00 和 0x55，此时计算结果为 0xFF55，结果保存在 DATA0 中，0xFF55/2 保存在 DATA2；
- 步骤 4.** DATA1 中继续写入 0xFF，此时计算结果 (0xFF55+0xFF>0xFFFF) 溢出，产生结果标志。读取标志寄存器 DCU_FLAG 得到结果标志；
- 步骤 5.** 写标志复位寄存器 DCU_FLAGCLR 清除标志位。

34.3.2 减法模式

减法模式计算 DATA0 和 DATA1 的差，其中 DATA0 作为被减数，DATA1 作为减数。每次写 DATA1 寄存器进行一次 DATA0-DATA1 的减法运算，减法运算结果存入 DATA0，减法运算结果除以 2 存入 DATA2。当 DATA0-DATA1 结果小于 0x0 (8-bit、16-bit、32-bit 模式) 时产生中断和标志位。

34.3.3 硬件触发启动模式

DCU 能够根据外围电路产生的事件触发启动运算。使用硬件触发启动模式时，需要先将功能时钟控制 0 寄存器 (FCG0) 的外围电路触发功能使能位置为有效。每个 DCU 单元可独立选择由其他外围电路送来的触发启动信号，选择启动信号时，在触发源选择寄存器 (DCU_TRGSEL) 中写入要选择的外围电路启动源的编号，寄存器详情请参见《自动运行系统 (AOS) 》。当该外围电路事件发生时，会将事件信号输入 DCU 并触发 DCU 启动运算。

硬件触发启动模式包括触发加模式和触发减模式：

- 在触发加模式时，每发生一次事件触发，DCU 将启动并进行一次 DATA0+DATA1 的加法运算，加法运算结果存入 DATA0，加法运算结果除以 2 存入 DATA2。当 DATA0+DATA1 的结果大于 0xFF (8-bit 模式) 或 0xFFFF (16-bit 模式) 或 0xFFFFFFFF (32-bit 模式) 时置位标志位并产生中断。
- 在触发减模式时，每发生一次事件触发，DCU 将启动并进行一次 DATA0-DATA1 的减法运算，减法运算结果存入 DATA0，减法运算结果除以 2 存入 DATA2。当 DATA0-DATA1 的结果小于 0x0 (8-bit、16-bit、32-bit 模式) 时置位标志位并产生中断。

硬件触发启动模式应用举例：

- 步骤 1. 控制寄存器 DCU_CTL 选择触发加模式，数据宽度为 8-bit；
- 步骤 2. 中断条件选择寄存器 DCU_INTEVTSEL 选择运算条件；
- 步骤 3. 分别在 DATA0 和 DATA1 中写入 0x00 和 0x56；
- 步骤 4. 在触发源选择寄存器 DCU_TRGSEL 中写入事件编号；
- 步骤 5. 使选中的外围电路动作并产生事件，DCU 由该事件触发并进行一次加法运算，结果为 0x56 并保存在 DATA0 中，0x56 除以 2 保存在 DATA2；
- 步骤 6. 在连续 3 次触发后，计算结果溢出，产生结果标志。读取标志寄存器 DCU_FLAG 得到结果标志；
- 步骤 7. 写标志复位寄存器 DCU_FLAGCLR 清除标志位。

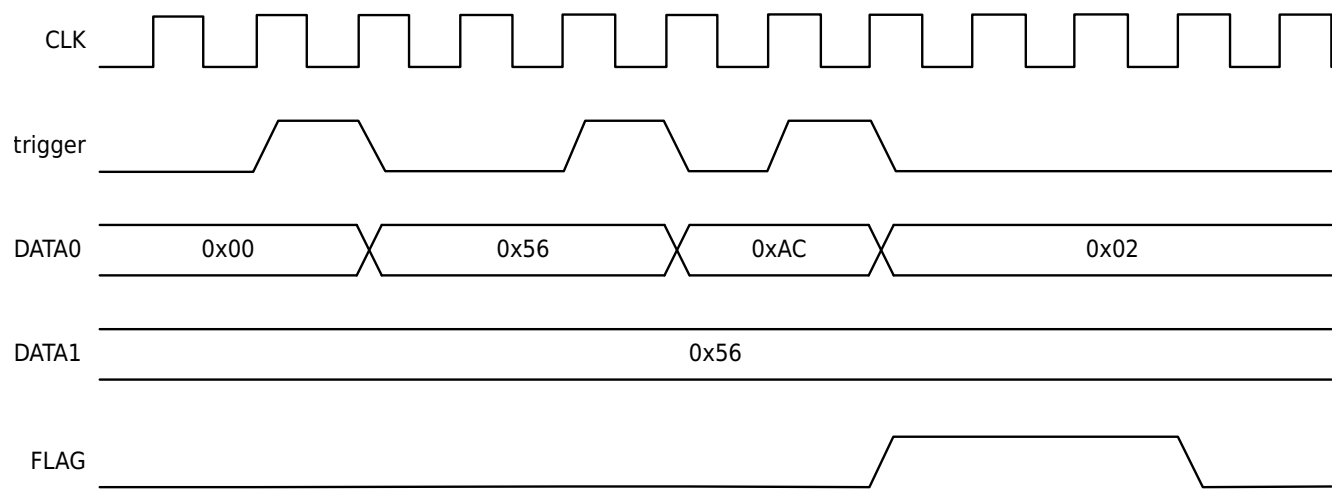


图 34-1 硬件触发启动示意图

34.3.4 比较模式

比较 DATA0 和 DATA1 以及 DATA0 和 DATA2 的大小，可选择当 DATA0 大于 DATA1，DATA0 小于 DATA1，DATA0 等于 DATA1 时以及当 DATA0 大于 DATA2，DATA0 小于 DATA2，DATA0 等于 DATA2 时产生标志位并产生中断。比较模式下可以选择数据开始比较的条件，写 DATA0 后比较或写任何数据寄存器后比较。

比较模式应用举例：

- 步骤 1. 控制寄存器 DCU_CTL 选择比较模式，数据宽度为 16-bit。在写入 DATA0 后开始比较；
- 步骤 2. 中断条件选择寄存器 DCU_INTEVTSEL 选择 DATA0>DATA1 时产生标志；

- 步骤 3. 在 DATA1 和 DATA0 中分别写入 0xB BBBB 和 0xA AAAA，此时由于不满足 DATA0>DATA1 因此不产生标志；
- 步骤 4. 在 DATA1 写入 0x8888，此时虽然满足 DATA0>DATA1，但由于设置为写入 DATA0 后开始比较，因此也不产生标志；
- 步骤 5. DATA0 写入 0x9999 后满足标志产生条件，产生标志；
- 步骤 6. 写标志复位寄存器 DCU_FLAGCLR 清除标志位。

34.3.5 三角波输出模式

三角波输出模式是 DCU 通过定时器作为触发源，为数模转换模块 (DAC) 提供一个连续变化的数字量，DAC 将这些数字量转化为三角波输出。三角波输出模式通过 DCU 控制寄存器 (DCU_CTL) 设定，设定成三角波输出模式后，数据寄存器 DATA0~2 的功能将发生变化，数据寄存器 DATA0 将作为输出给 DAC 的数字量的数据寄存器并复位。DATA0 变为输出数据寄存器，DATA.DO (Bit0~Bit11) 为有效位。DATA1 变为振幅设定寄存器，DATA1.UPL (Bit16~Bit27) 和 DATA1.LWL (Bit0~Bit11) 为有效位。DATA2 变为步长设定寄存器，DATA2.STEP (Bit0~Bit11) 为有效位。

使用三角波输出模式步骤如下：

- 步骤 1. 将 DCU 控制寄存器 (DCU_CTL) 的 MODE 设为 0b1000；
- 步骤 2. 设定三角波振幅的上限和下限。上限通过 DATA1.UPL[11:0]位设定，下限通过 DATA1.LWL[11:0]设定，DATA1.LWL 的值自动加载到 DATA0.DO 并输出给 DAC；
- 步骤 3. 设定三角波递增和递减的步长值，通过 DATA2.STEP[11:0]设定；
- 步骤 4. 通过触发源选择寄存器 (DCU_TRGSEL) 选择合适的定时器事件或软件触发作为触发源；
- 步骤 5. 配置选定的定时器，使其在设定的时间间隔上产生触发事件；
- 步骤 6. 启动定时器，每次产生触发事件时，DATA0.DO 将根据 DATA2.STEP 设定的步长值递增；
- 步骤 7. 当 DATA0.DO 递增至 DATA1.UPL 设定的上限时，将自动根据 DATA2.STEP 设定的步长值开始递减。当递减至 DATA1.LWL 设定的下限后重新开始递增；
- 步骤 8. 如果需要在三角波到达峰点或谷点时发生中断，请先将 DCU 控制寄存器 (DCU_CTL) 的中断使能位 (INTEN) 置为无效，再设置 DATA0~DATA2 的值，在启动定时器开始产生触发事件前清除一次 DCU 标志位后，将 DCU 控制寄存器中的中断使能位 (INTEN) 置为有效后，再启动定时器；
- 步骤 9. 将 DCU 控制寄存器 (DCU_CTL) 的 MODE 设为 0b0000 以退出三角波输出模式。

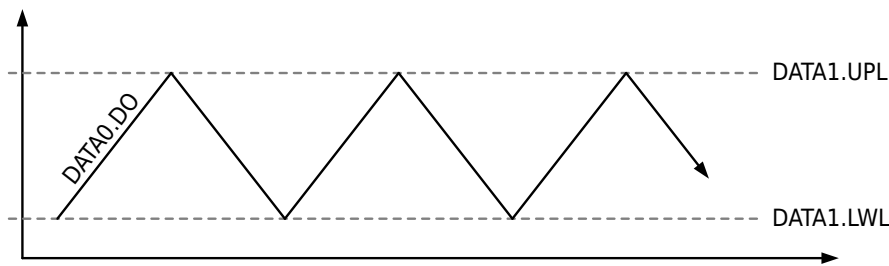


图 34-2 三角波输出模式波形

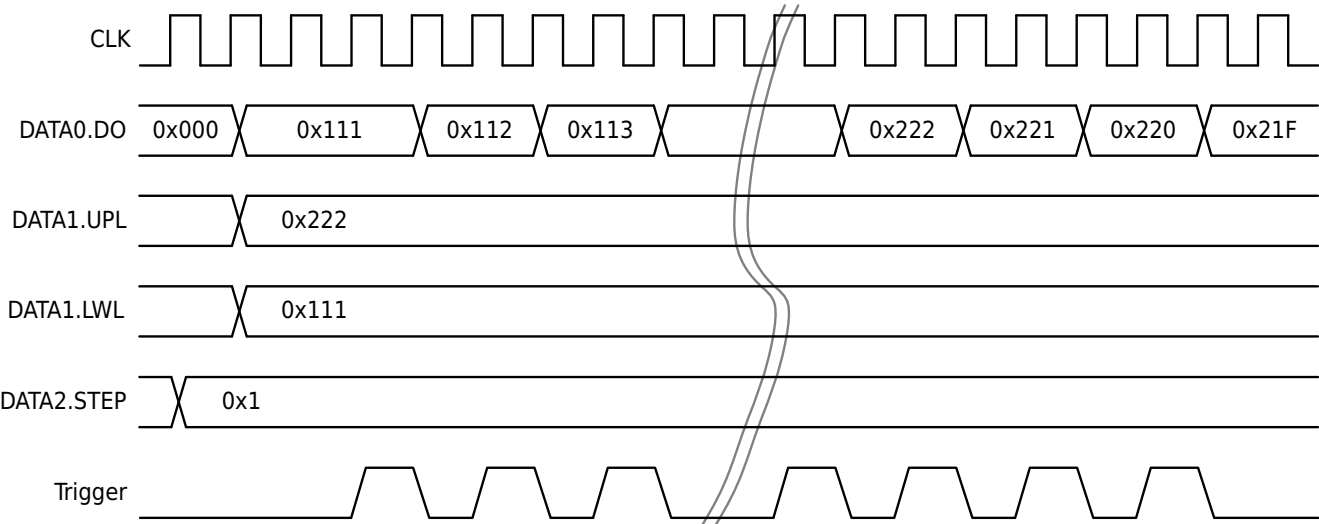


图 34-3 三角波输出模式

34.3.6 递增锯齿波输出模式

递增锯齿波输出模式是 DCU 通过定时器作为触发源，为数模转换模块（DAC）提供一个连续变化的数字量，DAC 将这些数字量转化为锯齿波输出。递增锯齿波输出模式通过 DCU 控制寄存器（DCU_CTL）设定，设定成递增锯齿波输出模式后，数据寄存器 DATA0~2 的功能将发生变化，数据寄存器 DATA0 将作为输出给 DAC 的数字量的数据寄存器并复位。DATA0 变为输出数据寄存器，DATA.DO（Bit0~Bit11）为有效位。DATA1 变为振幅设定寄存器，DATA1.UPL（Bit16~Bit27）和 DATA1.LWL（Bit0~Bit11）为有效位。DATA2 变为步长设定寄存器，DATA2.STEP（Bit0~Bit11）为有效位。

使用递增锯齿波输出模式步骤如下：

- 步骤 1. 将 DCU 控制寄存器（DCU_CTL）的 MODE 设为 0b1001；
- 步骤 2. 设定锯齿波振幅的上限和下限。上限通过 DATA1.UPL[11:0]位设定，下限通过 DATA1.LWL[11:0]设定，DATA1.LWL 的值自动加载到 DATA0.DO 并输出给 DAC；
- 步骤 3. 设定锯齿波递增的步长值，通过 DATA2.STEP[11:0]设定；
- 步骤 4. 通过触发源选择寄存器（DCU_TRGSEL）选择合适的定时器事件或软件触发作为触发源；
- 步骤 5. 配置选定的定时器，使其在设定的时间间隔上产生触发事件；
- 步骤 6. 启动定时器，每次产生触发事件时，DATA0.DO 将根据 DATA2.STEP 设定的步长值递增；
- 步骤 7. 当 DATA0.DO 递增至 DATA1.UPL 设定的上限时，将自动重新加载 DATA1.LWL 到 DATA0.DO，使其从 DATA1.LWL 设定的下限重新开始递增；
- 步骤 8. 如果需要在锯齿波重载时发生中断，请先将 DCU 控制寄存器（DCU_CTL）的中断使能位（INTEN）置为无效，再设置 DATA0~DATA2 的值，在启动定时器开始产生触发事件前清除一次 DCU 标志位后，将 DCU 控制寄存器中的中断使能位（INTEN）置为有效后，再启动定时器；
- 步骤 9. 将 DCU 控制寄存器（DCU_CTL）的 MODE 设为 0b0000 以退出锯齿波输出模式。

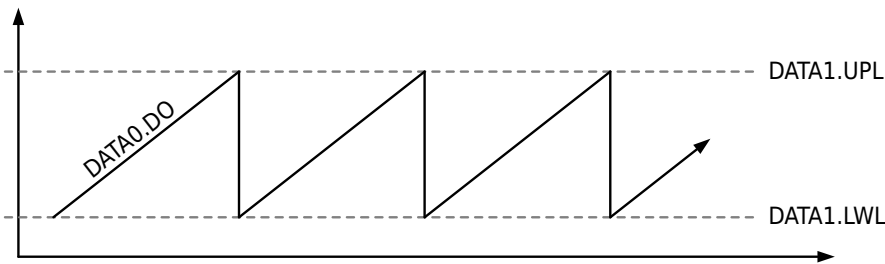


图 34-4 递增锯齿波输出模式波形

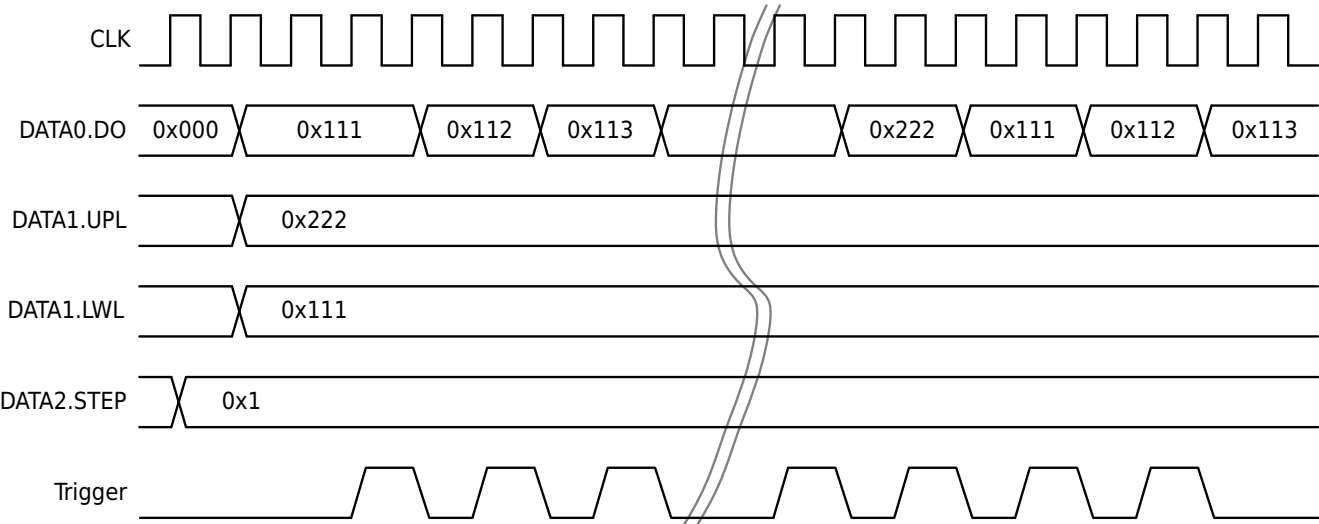


图 34-5 递增锯齿波输出模式

34.3.7 递减锯齿波输出模式

递减锯齿波输出模式是 DCU 通过定时器作为触发源，为数模转换模块（DAC）提供一个连续变化的数字量，DAC 将这些数字量转化为锯齿波输出。递减锯齿波输出模式通过 DCU 控制寄存器（DCU_CTL）设定，设定成递减锯齿波输出模式后，数据寄存器 DATA0~2 的功能将发生变化，数据寄存器 DATA0 将作为输出给 DAC 的数字量的数据寄存器并复位。DATA0 变为输出数据寄存器，DATA.DO（Bit0~Bit11）为有效位。DATA1 变为振幅设定寄存器，DATA1.UPL（Bit16~Bit27）和 DATA1.LWL（Bit0~Bit11）为有效位。DATA2 变为步长设定寄存器，DATA2.STEP（Bit0~Bit11）为有效位。

使用递减锯齿波输出模式步骤如下：

- 步骤 1. 将 DCU 控制寄存器（DCU_CTL）的 MODE 设为 0b1010；
- 步骤 2. 设定锯齿波振幅的上限和下限。上限通过 DATA1.UPL[11:0]位设定，下限通过 DATA1.LWL[11:0]设定；
- 步骤 3. 设定锯齿波递减的步长值，通过 DATA2.STEP[11:0]设定；
- 步骤 4. 通过触发源选择寄存器（DCU_TRGSEL）选择合适的定时器事件或软件触发作为触发源；
- 步骤 5. 配置选定的定时器，使其在设定的时间间隔上产生触发事件；
- 步骤 6. 启动定时器，第一次产生触发事件时，DATA1.UPL 的值加载到 DATA0.DO 并输出给 DAC，以后每次产生触发事件时，DATA0.DO 将根据 DATA2.STEP 设定的步长值递减；
- 步骤 7. 当 DATA0.DO 递减至 DATA1.LWL 设定时，将自动重新加载 DATA1.UPL 到 DATA0.DO，使其从 DATA1.UPL 设定的上限重新开始递减；
- 步骤 8. 如果需要在锯齿波重载时发生中断，请先将 DCU 控制寄存器（DCU_CTL）的中断使能位（INTEN）置为无效，再设置 DATA0~DATA2 的值，在启动定时器开始产生触发事件前清除一次 DCU 标志位后，将 DCU 控制寄存器中的中断使能位（INTEN）置为有效后，再启动定时器；
- 步骤 9. 将 DCU 控制寄存器（DCU_CTL）的 MODE 设为 0b0000 以退出锯齿波输出模式。

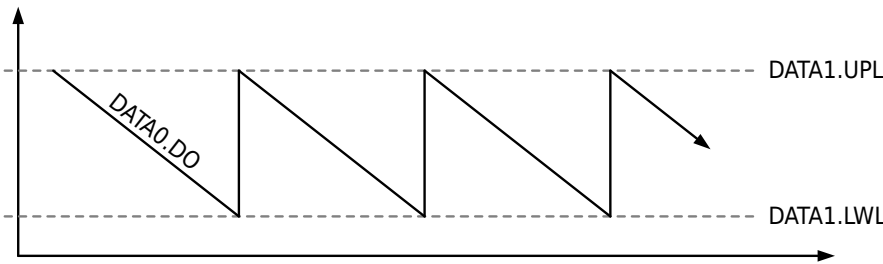


图 34-6 递减锯齿波输出模式波形

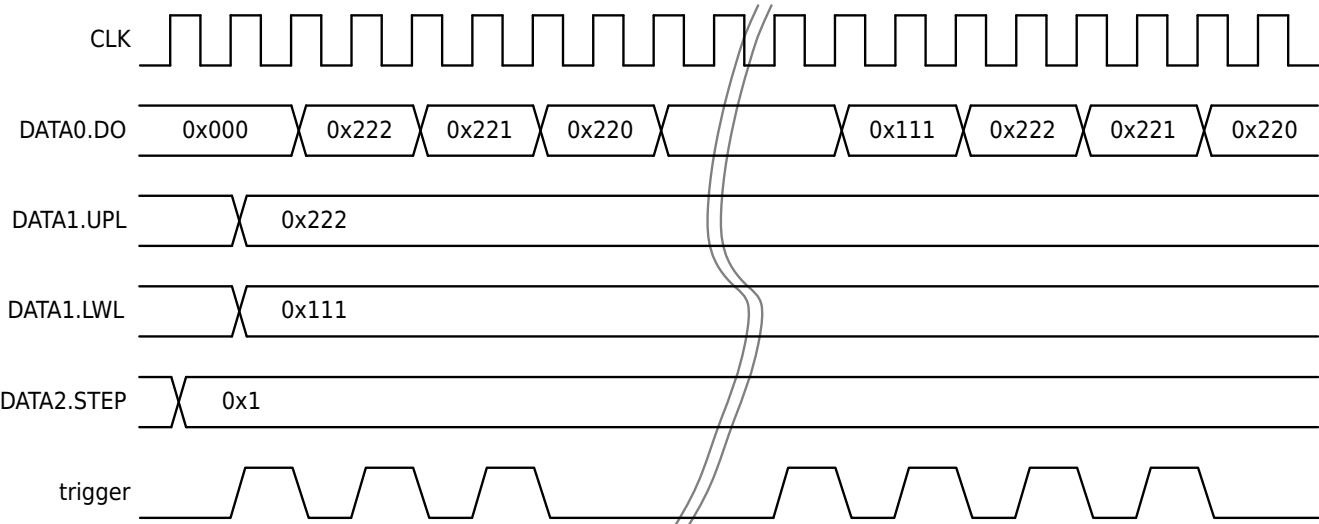


图 34-7 递减锯齿波输出模式

34.3.8 中断和事件

DCU 具有多种中断和用于触发启动其他外围电路的事件输出供用户选择。中断和事件输出的控制由中断和事件寄存器 (DCU_INTEVTSEL) 控制，当需要输出事件信号时，用户需要将中断和事件寄存器 (DCU_INTEVTSEL) 的相应控制位置为有效。每个 DCU 单元各输出一个 DCU 事件信号，在事件列表中分别为 DCU1~DCU4。当需要对应事件在发生时产生中断，用户需要将中断和事件寄存器 (DCU_INTEVTSEL) 的相应控制位置为有效的同时，将控制寄存器 DCU_CTL.INTEN 位置 1。每个 DCU 单元各输出一个 DCU 中断信号，在中断列表中分别为 DCU1~DCU4。

34.4 寄存器

34.4.1 寄存器总表

- DCU1 基地址：0x40056000
- DCU2 基地址：0x40056400
- DCU3 基地址：0x40056800
- DCU4 基地址：0x40056C00

表 34-1 数据计算单元（DCU）寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	DCU_CTL	DCU 控制寄存器	32
0x04	DCU_FLAG	DCU 标志寄存器	32
0x08+0x04*x	DCU_DATAx	DCU 数据寄存器 x（x=0~2）	32
0x14	DCU_FLAGCLR	DCU 标志复位寄存器	32
0x18	DCU_INTEVTSEL	DCU 中断和事件寄存器	32

34.4.2 DCU 控制寄存器 (DCU_CTL)

该寄存器用来配置 DCU 的动作模式。

Offset	Bit Position																															
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x80000000																															
Name	INTEN	Reserved																					COMPTRG	Reserved	DATASIZE	MODE						
Access																							RW		RW							

位/位域	标记	位名	功能描述	读写
31	INTEN	中断使能	0b0: 禁止 0b1: 允许	RW
30:9	Reserved	保留	-	-
8	COMPTRG	比较模式触发比较的时机	0b0: 写入 DATA0 后比较 0b1: 写入 DATA0 或 DATA1 或 DATA2 后比较	RW
7:6	Reserved	保留	-	-
5:4	DATASIZE	加/减法模式，比较模式数据大小	0b00: 8-bit 0b01: 16-bit 0b10: 32-bit 0b11: 禁止设置	RW
3:0	MODE	动作模式	0b0000: DCU 无效 0b0001: 加法模式，运算在 DATA1 寄存器写入数据后进行 0b0010: 减法模式，运算在 DATA1 寄存器写入数据后进行 0b0011: 硬件触发加法模式，由其他外围电路触发启动进行加法运算 0b0100: 硬件触发减法模式，由其他外围电路触发启动进行减法运算 0b0101: 比较模式 0b1000: 三角波输出模式 0b1001: 递增锯齿波输出模式 0b1010: 递减锯齿波输出模式 - 其他: 禁止设置	RW

34.4.3 DCU 标志寄存器 (DCU_FLAG)

该寄存器产生 DCU 的结果标识。

Offset	Bit Position																																													
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
Reset	0x00000000																																													
Name	Reserved																				RO	FLAG_TOP	RO	FLAG_BTM	RO	FLAG_RLD	Reserved		RO	FLAG_GT1	RO	FLAG_EQ1	RO	FLAG_LS1	RO	FLAG_GT2	RO	FLAG_EQ2	RO	FLAG_LS2	RO	FLAG_OP				
Access																					RO		RO		RO				RO		RO		RO		RO		RO		RO		RO		RO		RO	

位/位域	标记	位名	功能描述	读写
31:12	Reserved	保留	-	-
11	FLAG_TOP	三角波峰点	三角波输出模式下，当 DATA0[11:0]的值达到最大时置位，DCU_FLAGCLR.CLR_TOP 位写 1 时清除	RO
10	FLAG_BTM	三角波谷点	三角波输出模式下，当 DATA0[11:0]的值达到最小时置位，DCU_FLAGCLR.CLR_TOP 位写 1 时清除	RO
9	FLAG_RLD	锯齿波重载	递增锯齿波输出模式下，当 DATA0[11:0]的值达到最大时置位；递减锯齿波输出模式下，当 DATA0[11:0]的值达到最小时置位。 DCU_FLAGCLR.CLR_RLD 位写 1 时清除	RO
8:7	Reserved	保留	-	-
6	FLAG_GT1	大于标志位 1	比较模式下，当 DATA0>DATA1 时置位，DCU_FLAGCLR.CLR_GT1 位写 1 时清除	RO
5	FLAG_EQ1	等于标志位 1	比较模式下，当 DATA0=DATA1 时置位，DCU_FLAGCLR.CLR_EQ1 位写 1 时清除	RO
4	FLAG_LS1	小于标志位 1	比较模式下，当 DATA0<DATA1 时置位，DCU_FLAGCLR.CLR_LS1 位写 1 时清除	RO
3	FLAG_GT2	大于标志位 2	比较模式下，当 DATA0>DATA2 时置位，DCU_FLAGCLR.CLR_GT2 位写 1 时清除	RO
2	FLAG_EQ2	等于标志位 2	比较模式下，当 DATA0=DATA2 时置位，DCU_FLAGCLR.CLR_EQ2 位写 1 时清除	RO
1	FLAG_LS2	小于标志位 2	比较模式下，当 DATA0<DATA2 时置位，DCU_FLAGCLR.CLR_LS2 位写 1 时清除	RO
0	FLAG_OP	运算标志位	加法/减法和触发加/触发减模式时，当加法产生上溢或减法产生下溢时置位，DCU_FLAGCLR.CLR_OP 位写 1 时清除	RO

34.4.4 DCU 数据寄存器 (DCU_DATAx) (x=0~2)

各个数据寄存器在各模式下的功能如下：

模式功能	DATA0	DATA1	DATA2
加法模式	被加数/存放结果	加数	存放减半结果
触发加模式	被加数/存放结果	加数	存放减半结果
减法模式	被减数/存放结果	减数	存放减半结果
触发减模式	被减数/存放结果	减数	存放减半结果
比较模式	被比较对象	比较对象 1	比较对象 2
比较模式（窗口比较）	被比较对象	窗口上限	窗口下限
三角波输出模式	输出数据寄存器	振幅设置寄存器	步长寄存器
递增锯齿波输出模式	输出数据寄存器	振幅设置寄存器	步长寄存器
递减锯齿波输出模式	输出数据寄存器	振幅设置寄存器	步长寄存器

偏移地址 (Offset) : 0x08+0x04*x

加/减法模式、触发加/减法模式、比较模式

该寄存器用来存放运算数据。

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	DAT																															
Access	RW																															

位/位域	标记	位名	功能描述	读写
31:0	DAT	运算数据	存放运算数据，实际使用的位数根据 DCU_CTL.DATASIZE 设置： ● 当 DCU_CTL.DATASIZE=0b00 时 DATA[7:0]为有效数据； ● 当 DCU_CTL.DATASIZE=0b01 时 DATA[15:0]为有效数据； ● 当 DCU_CTL.DATASIZE=0b10 时 DATA[31:0]为有效数据。	RW

三角波输出模式

寄存器 DCU_DATA0 作为输出数据寄存器，Bit0~Bit11 为有效位。

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															
Access																																
																	RW DO															

位/位域	标记	位名	功能描述	读写
31:12	Reserved	保留	-	-
11:0	DO	输出数据	输出到 DAC 的数据	RW

寄存器 DCU_DATA1 作为振幅设定寄存器，Bit16~Bit27 及 Bit0~Bit11 为有效位。

Offset	Bit Position																																					
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	Reserved				UPL												Reserved				LWL																	
Access					RW																RW																	

位/位域	标记	位名	功能描述	读写
31:28	Reserved	保留	-	-
27:16	UPL	振幅上限	设定三角波振幅的上限	RW
15:12	Reserved	保留	-	-
11:0	LWL	振幅下限	设定三角波振幅的下限	RW

寄存器 DCU_DATA2 作为步长寄存器，Bit0~Bit11 为有效位。

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																STEP															
Access																																

位/位域	标记	位名	功能描述	读写
31:12	Reserved	保留	-	-
11:0	STEP	步长设置	设定三角波递增和递减的步长值	RW

递增/递减锯齿波输出模式

寄存器 DCU_DATA0 作为输出数据寄存器，Bit0~Bit11 为有效位。

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																DO															
Access																																

位/位域	标记	位名	功能描述	读写
31:12	Reserved	保留	-	-
11:0	DO	输出数据	输出到 DAC 的数据	RW

寄存器 DCU_DATA1 作为振幅设定寄存器，Bit16~Bit27 及 Bit0~Bit11 为有效位。

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved				UPL												Reserved				LWL											
Access					RW																RW											

位/位域	标记	位名	功能描述	读写
31:28	Reserved	保留	-	-
27:16	UPL	振幅上限	设定锯齿波振幅的上限	RW
15:12	Reserved	保留	-	-
11:0	LWL	振幅下限	设定锯齿波振幅的下限	RW

寄存器 DCU_DATA2 作为步长寄存器，Bit0~Bit11 为有效位。

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																STEP															
Access																																

位/位域	标记	位名	功能描述	读写
31:12	Reserved	保留	-	-
11:0	STEP	步长设置	设定锯齿波递增或递减的步长值	RW

34.4.5 DCU 标志复位寄存器 (DCU_FLAGCLR)

该寄存器用于清除 DCU 的结果标识。

Offset	Bit Position																																
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																				CLR_TOP	CLR_BTM	CLR_RLD	Reserved			CLR_GT1	CLR_EQ1	CLR_LS1	CLR_GT2	CLR_EQ2	CLR_LS2	CLR_OP
Access																					WO	WO	WO					WO	WO	WO	WO	WO	WO

位/位域	标记	位名	功能描述	读写
31:12	Reserved	保留	-	-
11	CLR_TOP	清除三角波峰点标志位	写 1 清除 DCU_FLAG.FLAG_TOP 位，写 0 无效	WO
10	CLR_BTM	清除三角波谷点标志位	写 1 清除 DCU_FLAG.FLAG_BTM 位，写 0 无效	WO
9	CLR_RLD	清除锯齿波重载标志位	写 1 清除 DCU_FLAG.FLAG_RLD 位，写 0 无效	WO
8:7	Reserved	保留	-	-
6	CLR_GT1	清除大于标志位 1	写 1 清除 DCU_FLAG.FLAG_GT1 位，写 0 无效	WO
5	CLR_EQ1	清除等于标志位 1	写 1 清除 DCU_FLAG.FLAG_EQ1 位，写 0 无效	WO
4	CLR_LS1	清除小于标志位 1	写 1 清除 DCU_FLAG.FLAG_LS1 位，写 0 无效	WO
3	CLR_GT2	清除大于标志位 2	写 1 清除 DCU_FLAG.FLAG_GT2 位，写 0 无效	WO
2	CLR_EQ2	清除等于标志位 2	写 1 清除 DCU_FLAG.FLAG_EQ2 位，写 0 无效	WO
1	CLR_LS2	清除小于标志位 2	写 1 清除 DCU_FLAG.FLAG_LS2 位，写 0 无效	WO
0	CLR_OP	清除运算标志位	写 1 清除 DCU_FLAG.FLAG_OP 位，写 0 无效	WO

34.4.6 DCU 中断和事件寄存器 (DCU_INTEVTSEL)

该寄存器能够选择 DCU 在何种条件下产生中断和输出事件信号。

Offset	Bit Position																															
0x18	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																				SEL_TOP	SEL_BTM	Reserved	SEL_WIN	SEL_GT1	SEL_EQ1	SEL_LS1	SEL_GT2	SEL_EQ2	SEL_LS2	SEL_OP	
Access																					RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:12	Reserved	保留	-	-
11	SEL_TOP	三角波峰点条件选择	0b0: 在三角波和输出模式下当 DATA0[11:0]达到最大时不产生中断和输出事件信号 0b1: 在三角波输出模式下当 DATA0[11:0]达到最大时产生中断和输出事件信号	RW
10	SEL_BTM	三角波谷点条件选择	0b0: 在三角波输出模式下当 DATA0[11:0]达到最小值时不产生中断和输出事件信号 0b1: 在三角波输出模式下当 DATA0[11:0]达到最小值时产生中断和输出事件信号	RW
9	Reserved	保留	-	-
8:7	SEL_WIN	窗口比较条件选择	比较模式下满足 SEL_WIN 设定的窗口比较条件时产生中断和输出事件信号，在 SEL_WIN 设定有效时，满足其他比较条件时不会产生中断和输出事件信号： 0b00: 不产生窗口比较中断和输出事件信号，此设定下由该寄存器的 Bit1~Bit6 选择其他中断和事件信号产生条件 0b01: 当 DATA0 数据在窗口内时产生中断和输出事件信号，即 $DATA2 \leq DATA0 \leq DATA1$ 0b10: 当 DATA0 数据在窗口外时产生中断和输出事件信号，即 $DATA0 > DATA1$ 或 $DATA0 < DATA2$ 0b11: 比较模式下不产生任何中断和事件信号	RW
6	SEL_GT1	大于条件选择 1	0b0: 比较模式下并且 SEL_WIN=0b00 时，当 $DATA0 > DATA1$ 时不产生中断和输出事件信号 0b1: 比较模式下并且 SEL_WIN=0b00 时，当 $DATA0 > DATA1$ 时产生中断和输出事件信号  说明 SEL_WIN≠0b00 时该位设置无效。	RW
5	SEL_EQ1	等于条件选择 1	0b0: 比较模式下并且 SEL_WIN=0b00 时，当 $DATA0 = DATA1$ 时不产生中断和输出事件信号 0b1: 比较模式下并且 SEL_WIN=0b00 时，当 $DATA0 = DATA1$ 时产生中断和输出事件信号  说明 SEL_WIN≠0b00 时该位设置无效。	RW

位/位域	标记	位名	功能描述	读写
4	SEL_LS1	小于条件选择 1	0b0: 比较模式下并且 SEL_WIN=0b00 时, 当 DATA0<DATA1 时不产生中断和输出事件信号 0b1: 比较模式下并且 SEL_WIN=0b00 时, 当 DATA0<DATA1 时产生中断和输出事件信号  说明 SEL_WIN≠0b00 时该位设置无效。	RW
3	SEL_GT2	大于条件选择 2	0b0: 比较模式下并且 SEL_WIN=0b00 时, 当 DATA0>DATA2 时不产生中断和输出事件信号 0b1: 比较模式下并且 SEL_WIN=0b00 时, 当 DATA0>DATA2 时产生中断和输出事件信号  说明 SEL_WIN≠0b00 时该位设置无效。	RW
2	SEL_EQ2	等于条件选择 2	0b0: 比较模式下并且 SEL_WIN=0b00 时, 当 DATA0=DATA2 时不产生中断和输出事件信号 0b1: 比较模式下并且 SEL_WIN=0b00 时, 当 DATA0=DATA2 时产生中断和输出事件信号  说明 SEL_WIN≠0b00 时该位设置无效。	RW
1	SEL_LS2	小于条件选择 2	0b0: 比较模式下并且 SEL_WIN=0b00 时, 当 DATA0<DATA2 时不产生中断和输出事件信号 0b1: 比较模式下并且 SEL_WIN=0b00 时, 当 DATA0<DATA2 时产生中断和输出事件信号  说明 SEL_WIN≠0b00 时该位设置无效。	RW
0	SEL_OP	运算条件选择	0b0: 加法和减法模式下当运算结果上溢或下溢时不产生中断和输出事件信号 0b1: 加法和减法模式下当运算结果上溢或下溢时产生中断和输出事件信号	RW

**说明**

锯齿波输出模式下产生中断和输出事件信号的条件选择位没有配置。

34.5 注意事项

当选择三角波输出模式或者递增锯齿波输出模式时, 请勿使用第一个周期输出波形。

35 CRC 运算 (CRC)

35.1 简介

循环冗余校验 (CRC) 计算单元将数据流或数据块作为输入，在生成多项式的控制下生成一个输出数。该输出数常用于验证数据传输或存储的正确性和完整性。本模块支持计算 CRC 值和检验 CRC 值。

35.2 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：8-bit，16-bit，32-bit
- 两种工作模式：CRC 编码模式、CRC 校验模式
- CRC-16 多项式： $x^{16}+x^{12}+x^5+1$
- CRC-32 多项式： $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

35.3 功能说明

35.3.1 工作模式

本模块支持两种工作模式：CRC 编码模式、CRC 校验模式。

CRC 编码模式是指向 CRC 模块输入一定数量的原始数据，获取 CRC 模块生成的输出值 (CRC_RESULT)。

CRC 校验模式是指向 CRC 模块输入一定数量的已编码的数据序列 (原始数据+CRC 校验值)，验证原始数据与 CRC 校验值是否匹配 (CRC_CR.FLAG)。

35.3.2 编码方式

本模块支持两种编码方式 CRC-16、CRC-32，其计算结果分别为 16 比特和 32 比特，可通过 CRC_CR.CR 配置需要使用的编码方式。

CRC-16 编码

该模式下的 CRC 算法模型为 CRC-16-X25：

- 多项式为： $x^{16}+x^{12}+x^5+1$ ；
- 初始值为：0xFFFFu；
- 结果异或值为：0xFFFFu；
- 数据输入反转：True；
- 数据输出反转：True。

CRC-32 编码

该模式下的 CRC 算法模型为 CRC-32：

- 多项式为： $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$ ；
- 初始值为：0xFFFFFFFFu；
- 结果异或值为：0xFFFFFFFFu；
- 数据输入反转：True；
- 数据输出反转：True。

35.3.3 写入位宽

本模块支持三种写入位宽：8-bit，16-bit，32-bit。不同位宽的写入需要符合“位宽一致、先低后高”的原则，即“每次写入数据，都必须写入到与本次有效数据位宽相等的寄存器中，并且较低位数据的写入先于较高位数据”。

下方展示了同一序列数据采用三种位宽进行写入的方法，其输出结果相同。

- 8-bit 位宽写入：0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77
- 16-bit 位宽写入：0x1100, 0x3322, 0x5544, 0x7766
- 32-bit 位宽写入：0x33221100, 0x77665544

35.4 操作示例

35.4.1 CRC-16 编码模式

操作步骤

- 步骤 1. 向 CRC_CR.CR 写入 0b0，选择 CRC-16。
- 步骤 2. 向 CRC_RESLT[15:0]写入 0xFFFF，初始化 CRC 计算。
- 步骤 3. 将待编码的原始数据依次写入 CRC_DAT 寄存器，写入位宽可选择 8-bit、16-bit、32-bit。
- 步骤 4. 读取 CRC_RESLT[15:0]以获取 CRC-16 的计算结果。

35.4.2 CRC-16 检验模式

操作步骤

- 步骤 1. 向 CRC_CR.CR 写入 0b0，选择 CRC-16。
- 步骤 2. 向 CRC_RESLT[15:0]写入 0xFFFF，初始化 CRC 计算。
- 步骤 3. 将已编码的数据序列（原始数据+CRC 校验值）依次写入 CRC_DAT 寄存器，写入位宽可选择 8-bit、16-bit、32-bit。
- 步骤 4. 根据 CRC_CR.FLAG 的值判定已编码的数据序列是否被篡改。

35.4.3 CRC-32 编码模式

操作步骤

- 步骤 1. 向 CRC_CR.CR 写入 0b1，选择 CRC-32。
- 步骤 2. 向 CRC_RESLT[31:0]写入 0xFFFFFFFF，初始化 CRC 计算。
- 步骤 3. 将待编码的原始数据依次写入 CRC_DAT 寄存器，写入位宽可选择 8-bit、16-bit、32-bit。
- 步骤 4. 读取 CRC_RESLT[31:0]以获取 CRC-32 的计算结果。

35.4.4 CRC-32 检验模式

操作步骤

- 步骤 1. 向 CRC_CR.CR 写入 0b1，选择 CRC-32。
- 步骤 2. 向 CRC_RESLT[31:0]写入 0xFFFFFFFF，初始化 CRC 计算。
- 步骤 3. 将已编码的数据序列（原始数据+CRC 校验值）依次写入 CRC_DAT 寄存器，写入位宽可选择 8-bit、16-bit、32-bit。
- 步骤 4. 根据 CRC_CR.FLAG 的值判定已编码的数据序列是否被篡改。

35.5 寄存器

35.5.1 寄存器总表

基地址：0x40008C00

表 35-1 CRC 运算（CRC）寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	CRC_CR	控制寄存器	32
0x04	CRC_RESULT	结果寄存器	32
0x80~0xFF	CRC_DAT	数据寄存器	32

35.5.2 控制寄存器 (CRC_CR)

Offset	Bit Position																															
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000001																															
Name	Reserved																														FLAG	
Access																															RO	RW

位/位域	标记	位名	功能描述	读写
31:2	Reserved	保留	-	-
1	FLAG	CRC 校验结果	0b0: 当前 CRC 校验错误 0b1: 当前 CRC 校验正确	RO
0	CR	CRC 编码方式选择	0b0: CRC-16 编码 0b1: CRC-32 编码	RW

35.5.3 结果寄存器 (CRC_RESULT)

Offset	Bit Position																															
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	RESULT																															
Access																																

位/位域	标记	位名	功能描述	读写
31:0	RESULT	CRC 计算结果	读取 RESULT[15:0]以获取 CRC-16 的计算结果 读取 RESULT[31:0]以获取 CRC-32 的计算结果 向 RESULT[15:0]写入 0xFFFF 以初始化 CRC-16 计算 向 RESULT[31:0]写入 0xFFFFFFFF 以初始化 CRC-32 计算	RW

35.5.4 数据寄存器（CRC_DAT）

Offset	Bit Position																															
0x80~0xFF	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	CRC_DAT																															
Access	WO																															

位/位域	标记	位名	功能描述	读写
31:0	CRC_DAT	需要运算的数据	用于写入需要运算的数据，支持 3 种写入位宽： ● 8-bit 写入方式：*((uint8_t*)0x40008C80) = 0xXX ● 16-bit 写入方式：*((uint16_t*)0x40008C80) = 0xFFFF ● 32-bit 写入方式：*((uint32_t*)0x40008C80) = 0xFFFFFFFF 说明 本寄存器的偏移地址是一个范围（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。该寄存器读出恒为 0。	WO

36 调试控制器 (DBGC)

36.1 简介

本 MCU 的内核是 Cortex-M4，该内核包含用于高级调试功能的硬件。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。

提供调试接口：

- 串行调试跟踪接口 SWD
- 并行调试跟踪接口 JTAG

36.2 功能说明

36.2.1 功能框图

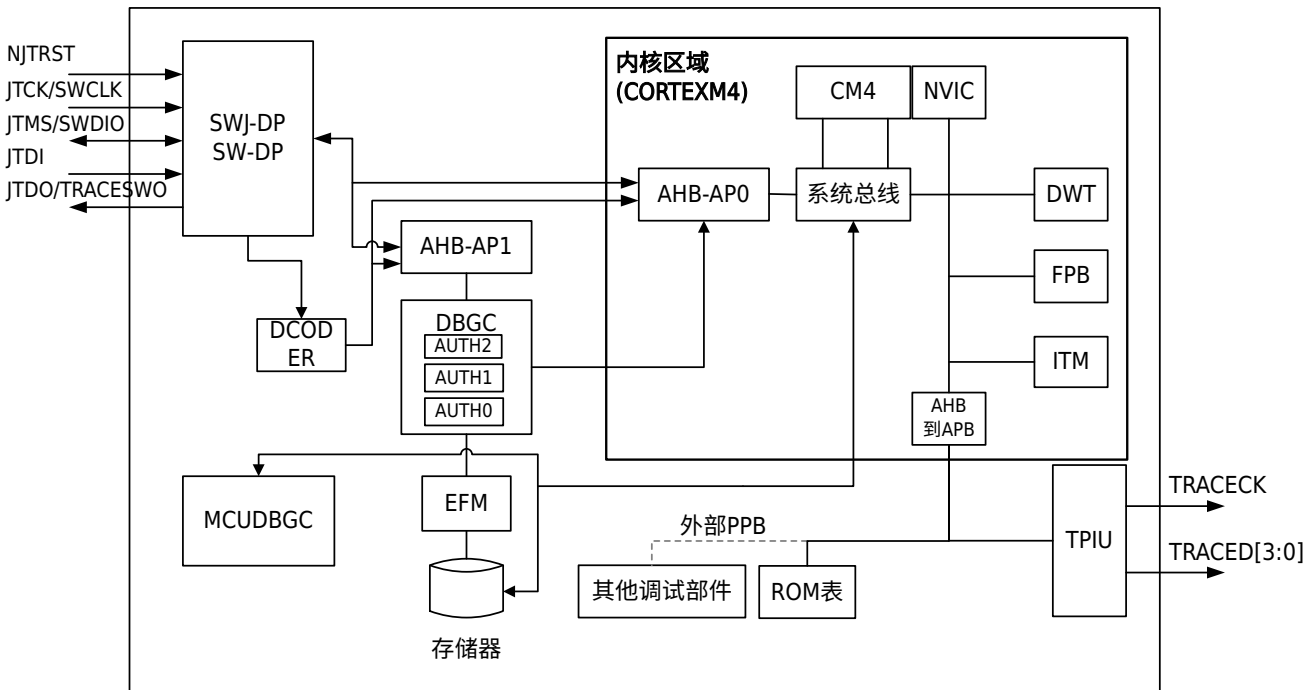


图 36-1 调试控制系统框图

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



说明

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》。

36.2.2 参考文档

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

36.2.3 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。

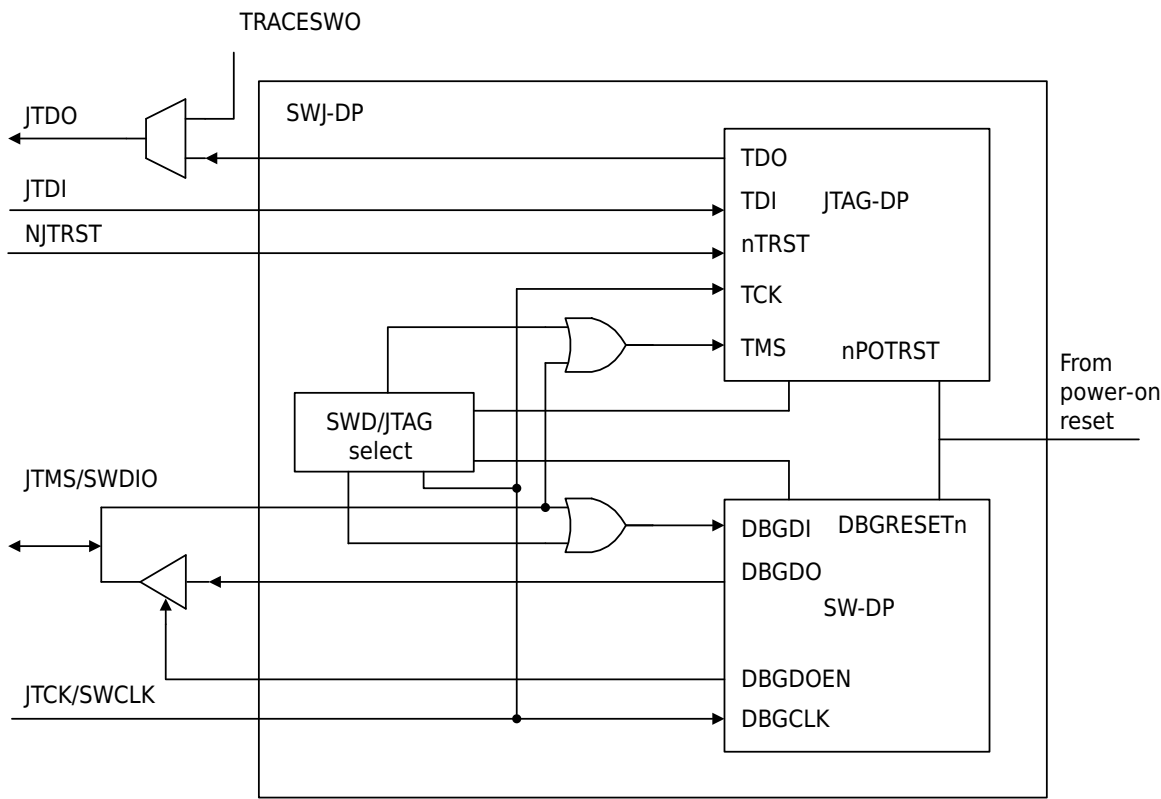


图 36-2 调试控制系统

36.2.3.1 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

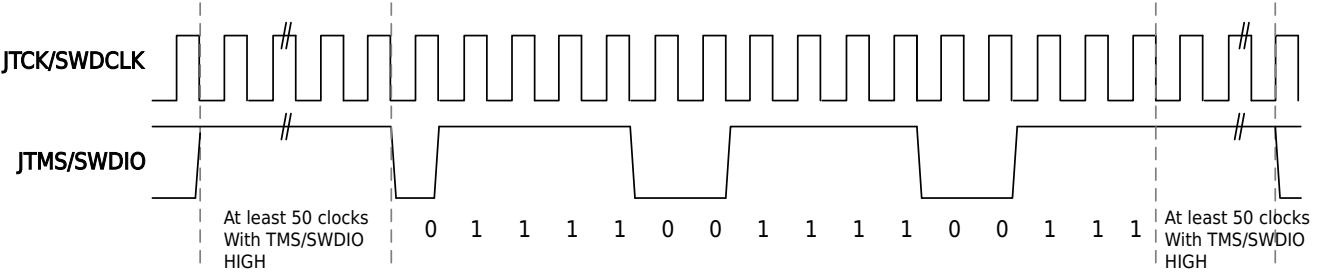


图 36-3 JTAG-DP 到 SW-DP 切换时序

36.2.4 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。

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 36-1 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	-	-

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，若要使用跟踪输出，请配置寄存器 MCUTRACECTL）。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 36-2 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

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 <100kΩ 的弱上拉，建议采用外部 10kΩ 的外部上拉。

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他 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）。

36.2.5 SW 调试端口-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 10kΩ）。

36.2.6 TPIU（跟踪端口接口单元）

36.2.6.1 简介与框图

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。

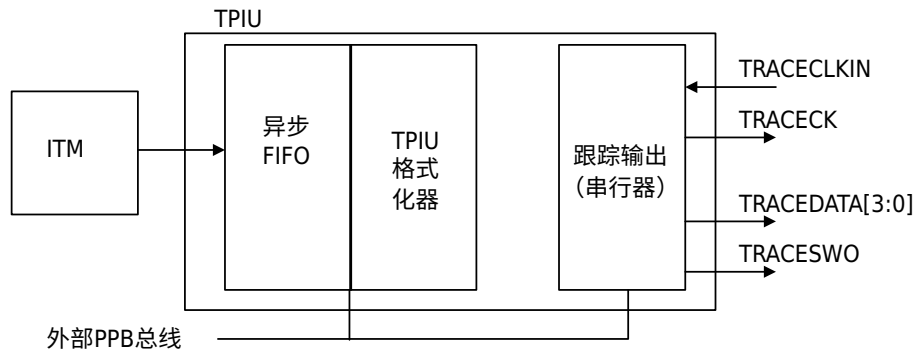


图 36-4 TPIU 框图

36.2.6.2 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 引脚分配原则

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]进行配置。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					
跟踪模式	TRACE_IOEN	TRACE_MODE	JTDO/TRACESWO	TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]
无跟踪（默认状态）	0b0	0bXX	释放 ⁽¹⁾	释放	释放	释放	释放	释放
异步跟踪	0b1	0b00	TRACESWO	释放	释放	释放	释放	释放
同步跟踪 1 位	0b1	0b01	释放 ⁽¹⁾	TRACECK	TRACED[0]	释放	释放	释放
同步跟踪 2 位	0b1	0b10	释放 ⁽¹⁾	TRACECK	TRACED[0]	TRACED[1]	释放	释放
同步跟踪 4 位	0b1	0b11	释放 ⁽¹⁾	TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]

 **说明**

1. 使用串行模式时，释放此引脚。但使用 JTAG 时，此引脚分配给 TDO。

36.2.6.3 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。

36.2.6.4 TPIU 寄存器使用说明

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）。

36.2.6.5 TPIU 配置示例

- 1. 将调试异常和监视控制寄存器（DEMCR）中的位 TRCENA 置 1；
- 2. 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）；
- 3. 将 0x102 写入 TPIU 格式化器和刷新控制寄存器（默认值）；
- 4. 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）；
- 5. 将 0x00 写入 MCUTRACECTL.TRACE_IOEN，为异步模式分配 TRACE I/O；
- 6. 此时发送 TPIU 同步数据包（FF_FF_FF_7F）；
- 7. 配置 ITM 并对 ITM 激励寄存器进行写操作以输出值。

36.3 寄存器

36.3.1 寄存器总表

基地址：0xE0042000

表 36-6 调试控制器 (DBGC) 寄存器偏移地址

偏移地址	寄存器	描述	位宽	访问主机
0x000C	CHIPID	芯片专属标志寄存器	32	调试 IDE
0x001C	MCUDBGSTAT	DBG 状态寄存器	32	CPU/调试 IDE ⁽¹⁾
0x0020	MCUSTPCTL	外设调试暂停寄存器	32	CPU/调试 IDE ⁽¹⁾
0x0024	MCUTRACECTL	调试组件配置寄存器	32	CPU/调试 IDE ⁽¹⁾
0x0028	MCUSTPCTL2	外设调试暂停寄存器 2	32	CPU/调试 IDE ⁽¹⁾

 说明

1. 寄存器位于 PPB 区域，只能在特权模式下 CPU 才能进行访问。

36.3.2 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。

Offset	Bit Position																															
0x000C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x58480448																															
Name	CHIPID																															
Access	RO																															

位/位域	标记	位名	功能描述	读写
31:0	CHIPID	芯片专属标志	CHIPID[31:16]: XH 的对应 ASCII 码, 0x5848 CHIPID[15:0]: 产品型号, 0x0448	RO

36.3.3 DBG 状态寄存器 (MCUDBGSTAT)

DBG 调试上电状态确认寄存器。

Offset	Bit Position																															
0x001C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																														CDBGPWRUPACK	CDBGPWRUPREQ
Access																															RW	RW

位/位域	标记	位名	功能描述	读写
31:2	Reserved	保留	-	-
1	CDBGPWRUPACK	调试器上电反馈	0b0: 无反馈 0b1: 调试上电反馈	RW
0	CDBGPWRUPREQ	调试器上电请求	0b0: 无上电请求 0b1: 上电请求	RW

36.3.4 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。

Offset	Bit Position																																	
0x0020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x0000003B																																	
Name	Reserved															M16STP	M15STP	M14STP	M13STP	M12STP	Reserved						M07STP	M06STP	Reserved			RTCSTP	WDTSTP	SWDSTP
Access																RW	RW	RW	RW	RW							RW	RW				RW	RW	RW

位/位域	标记	位名	功能描述	读写
31:17	Reserved	保留	-	-
16	M16STP	Timer6_2 计数暂停信号	0b0: 即使内核停止，计数器仍然计数 0b1: 内核停止时，计数器暂停计数	RW
15	M15STP	Timer6_1 计数暂停信号	0b0: 即使内核停止，计数器仍然计数 0b1: 内核停止时，计数器暂停计数	RW
14	M14STP	Timer4_3 计数暂停信号	0b0: 即使内核停止，计数器仍然计数 0b1: 内核停止时，计数器暂停计数	RW
13	M13STP	Timer4_2 计数暂停信号	0b0: 即使内核停止，计数器仍然计数 0b1: 内核停止时，计数器暂停计数	RW
12	M12STP	Timer4_1 计数暂停信号	0b0: 即使内核停止，计数器仍然计数 0b1: 内核停止时，计数器暂停计数	RW
11:8	Reserved	保留	-	-
7	M07STP	Timer0_2 计数暂停信号	0b0: 即使内核停止，计数器仍然计数 0b1: 内核停止时，计数器暂停计数	RW
6	M06STP	Timer0_1 计数暂停信号	0b0: 即使内核停止，计数器仍然计数 0b1: 内核停止时，计数器暂停计数	RW
5:3	Reserved	保留	所有位读出的 1，写入时必须写 1	-
2	RTCSTP	RTC 计数暂停信号	0b0: 即使内核停止，计数器仍然计数 0b1: 内核停止时，计数器暂停计数	RW
1	WDTSTP	WDT 计数暂停信号	0b0: 即使内核停止，计数器仍然计数 0b1: 内核停止时，计数器暂停计数	RW
0	SWDTSTP	SWDT 计数暂停信号	0b0: 即使内核停止，计数器仍然计数 0b1: 内核停止时，计数器暂停计数	RW

36.3.5 调试组件配置寄存器 (MCUTRACECTL)

通过该寄存器配置 TRACE 输出引脚。

Offset	Bit Position																															
0x0024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																TRACE_IOEN		TRACE_MODE													
Access																																

位/位域	标记	位名	功能描述	读写
31:3	Reserved	保留	-	-
2	TRACE_IOEN	TRACE 引脚输出控制	● 0b0: 同步跟踪引脚输出禁止 ● 0b1: 同步跟踪引脚输出许可	RW
1:0	TRACE_MODE	TRACED 输出引脚控制	● 0b00: 异步跟踪 ● 0b01: 同步跟踪 1 位 TRACED[0] ● 0b10: 同步跟踪 2 位 TRACED[1:0] ● 0b11: 同步跟踪 4 位 TRACED[3:0]	RW

36.3.6 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。

Offset	Bit Position																																					
0x0028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	Reserved																												M36STP		M35STP		M34STP		M33STP		M32STP	
Access																													RW	RW	RW	RW	RW					

位/位域	标记	位名	功能描述	读写
31:5	Reserved	保留	-	-
4	M36STP	TimerA_5 计数 暂停信号	● 0b0：即使内核停止，计数器仍然计数 ● 0b1：内核停止时，计数器暂停计数	RW
3	M35STP	TimerA_4 计数 暂停信号	● 0b0：即使内核停止，计数器仍然计数 ● 0b1：内核停止时，计数器暂停计数	RW
2	M34STP	TimerA_3 计数 暂停信号	● 0b0：即使内核停止，计数器仍然计数 ● 0b1：内核停止时，计数器暂停计数	RW
1	M33STP	TimerA_2 计数 暂停信号	● 0b0：即使内核停止，计数器仍然计数 ● 0b1：内核停止时，计数器暂停计数	RW
0	M32STP	TimerA_1 计数 暂停信号	● 0b0：即使内核停止，计数器仍然计数 ● 0b1：内核停止时，计数器暂停计数	RW

A 术语表

术语	全称	说明
Byte	-	字节，8 位数据
Half Word	-	半字，16 位数据
Word	-	字，32 位数据
Double Word	-	双字，64 位数据
IAP	-	指可以在用户程序运行期间对微控制器的 Flash 进行重新编程（在应用中编程）
ICP	-	指可以在器件安装于用户应用电路板上时使用 JTAG 协议、SWD 协议或自举程序对微控制器的 Flash 进行编程（在线编程）
AHB	Advanced High-performance Bus	高级高性能总线
APB	Advanced Peripheral Bus	低速外设总线
DMA	Direct Memory Access	直接存储器访问
TIM/TMR	-	定时器

修订记录

文档版本日期	修改说明
Rev1.31 2026/02/12	1. 中断控制器（INTC）：全篇描述“WDT 刷新中断”统一修改为“WDT 刷新错误中断”； 2. 高级控制定时器（Timer6）：“时钟源选择”章节描述形式优化； 3. 通用定时器（TimerA）：“时钟源选择”章节描述形式优化；“缓存控制寄存器（TMRA_BCONRm）”章节功能描述形式优化，与寄存器变量 m 匹配； 4. 实时时钟（RTC）：“控制寄存器 3（RTC_CR3）”章节 RCKSEL 位新增说明； 5. 串行外设接口（SPI）：“SPI 控制寄存器（SPI_CR）”章节 CSUSPE 位新增说明；
Rev1.30 2025/11/27	1. 复位控制（RMU）：“复位条件”章节 SRAM 复位寄存器新增“SRAM0_ECCERRADDR、SRAMB_ECCERRADDR”，复位源由“掉电模式 3 唤醒复位”修改为“掉电唤醒复位”；“功能复位控制寄存器 0（RMU_FRST0）”章节 Bit13 位名由“KEY”修改为“KEYSCAN”； 2. 时钟控制器（CMU）：“CMU XTAL32 控制寄存器（CMU_XTAL32CR）”章节删除软件设定 XTL32 振荡和停止的等待约束； 3. 电源控制（PWC）：“内部电压采样和检测功能”章节“内部基准电压约为 1.10V”修改为“内部基准电压约为 1.15V”；“功能时钟控制 0（PWC_FCG0）”章节 Bit13 位名由“KEY”修改为“KEYSCAN”； 4. 嵌入式 Flash（EFM）：“D-BUS 读保护功能”章节移到“读接口”子章节；“连续编程功能”章节步骤 9 由“修改连续编程模式为非连续编程模式”修改为“修改连续编程模式为只读模式来退出连续编程”； 5. 通用 IO（GPIO）：“简介”章节删除 POD/PMOS 描述；“功能框图”章节补充下拉开关，修改上拉下拉不经过输出状态锁存器；“功能选择寄存器（PFSRxy）”章节说明 2 拆分成两条，细化描述； 6. 中断控制器（INTC）：“功能框图”章节框图优化；不可屏蔽中断相关章节“检测主发振器停止中断”描述统一修改为“XTAL 振荡故障中断”；“不可屏蔽中断”章节说明描述由“除非用 RESET 来复位”修改为“除非复位”；INTC_SWIER 寄存器名称由“软件中断寄存器”修改为“软件中断事件寄存器”，同步修改功能描述； 7. 自动运行系统（AOS）：“AOS 源事件和目标”章节 AOS 目标列表 ADC_1/2/3 对应序号和动作调整，A 和 B 序列与触发寄存器没有对应关系； 8. 内部时钟校准器（CTC）：“时钟校准状态寄存器（CTC_STR）”章节 Bit3 位新增描述“CTC_CR1.CTCEN=1 时置位，在软件将 CTCEN 清零或者发生校准错误时清零”； 9. DMA 控制器（DMA）：“传输请求状态寄存器（DMA_REQSTAT）”章节 RCFGREQ 位由 Bit16 修改为 Bit15；“注意事项”章节描述优化； 10. 电压比较器（CMP）：“比较器滤波和中断寄存器（CMP_FIR）”章节 Bit[2:0]功能描述新增“0b111：使用噪声滤波器，通过 PCLK0/64 进行采样”； 11. 模数转换模块（ADC）：“模拟看门狗窗口 x 阈值寄存器（ADC_AWDxDRy）（x=0~1，y=0~1）”章节功能描述新增说明“AWD0DR0、AWD0DR1、AWD1DR0、AWD1DR1 未使用的位需补 0”； 12. 数模转换器（DAC）：“模拟输出控制寄存器（DAC_DAOCR）”章节 Bit15 和 Bit14 功能描述取值修正为正确实现； 13. 高级控制定时器（Timer6）：“脉宽测量”、“周期测量”章节支持硬件触发的端口新增描述“TMR6_<t>_PWMA~B 端口”；

文档版本日期	修改说明
Rev1.30 2025/11/27	14. 通用控制定时器 (Timer4)：“PWM 状态控制寄存器 (TMR4_PSCR)”章节 Bit9 位功能描述取值修正为正确实现；“中断和事件”章节计数周期匹配中断及事件小节描述由“锯齿波递增计数至上溢点、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”修改为“锯齿波计数至上溢点或下溢点、三角波计数至谷点或峰点时”，重载计数匹配中断及事件小节描述由“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”修改为“TMR4_PFSRn 作为重载定时器的周期计数值，使能计数时从寄存器 TMR4_PFSRn 重载为计数器初始值，执行递减操作，完成一个周期后，产生重载计数中断请求”； 15. 紧急刹车模块 (EMB)：“EMB 通道功能框图”章节 EMB 通道功能框图中寄存器和外部端口描述形式统一； 16. 通用定时器 (TimerA)：“基本动作-比较输出”章节比较输出动作图形描述由“PWMA/B”修改为“PWM1/2”；“基本动作-捕获输入”章节捕获输入动作图形描述由“CCONR1.HICP2=1、CCONR2.HICP1=1”修改为“CCONR1.HICP1=1、CCONR2.HICP0=1”，“PWMA/B”修改为“PWM1/2”；“端口控制寄存器 (TMRA_PCONRm)”章节 FORC 位取值 0b0x 描述由“禁止设置”修改为“不使用强制端口状态设定”； 17. 通用定时器 (Timer0)：“软件同步启动”章节说明描述由“当计数时钟选择 PCLK1 的分频时钟时”修改为“当模块内部计数时钟选择 PCLK 的分频时钟时”；“注意事项”章节新增描述“在写 BCONR.CSTA时，请勿同时改写 BCONR.SYNSA (BCONR.SYNSA须保持当前值)，否则 BCONR.CSTA无法写入”，新增描述“用 Timer0 事件触发其他外设动作时，Timer0 对应通道的基准值寄存器设置须大于 5”； 18. 实时时钟 (RTC)：“控制寄存器 1 (RTC_CR1)”章节 Bit7 位功能描述优化，新增配置约束；“闹钟寄存器 (RTC_ALMWEK)”章节描述由“十进制 0~6 表示周日~周六。请写入十进制 0~6 的 BCD 码，写入超出范围的值将不会发生闹钟匹配”修改为“请设定正确的闹钟匹配值，否则不会发生闹钟匹配”； 19. 看门狗计数器 (WDT/SWDT)：“控制寄存器 (SWDT_CR、WDT_CR)”章节 Bit[7:4]位功能描述“SWDTCLK”修改为“SWDTLRC”； 20. 通用同步异步收发器 (USART)：“UART 接收 TIMEOUT 功能”章节描述新增“Timer0 计数方式需设置为异步计数方式，异步计数时钟源设置为 XTAL32 (Timer0 计数时钟由 UART 波特率生成器产生的时钟 tick 提供，可以不存在 XTAL32 晶振)”；“状态寄存器 (USART_SR)”章节 Bit7 功能描述优化；“控制寄存器 2 (USART_CR2)”章节 Bit[12:11]位功能描述新增说明“UART 模式使用接收 TIMEOUT 功能时，该位必须设置为 0b01，可以不使用 USARTn_CK 管脚”； 21. 集成电路总线 (I2C)：“I2C 控制寄存器 4 (I2C_CR4)”章节新增 Bit[19:16]； 22. 外部存储器控制器 (EXMC)：“时序配置寄存器 (SMC_TMCR)”章节 Bit[22:20]位新增说明“仅在地址数据复用时有效，有效值为 1~7；地址数据不复用时写 1”；SMC“访问时序”章节删除 BAA 功能，相关寄存器位和描述移除；
Rev1.20 2025/03/28	1. 整体手册风格按照结构化开发流程刷新。 2. 复位控制 (RMU)：“功能说明-复位时序-可编程电压检测 1 复位、可编程电压检测 2 复位”章节描述由“RMU_RSTF0.PVD1F”、“RMU_RSTF0.PVD2F”修改为“RMU_RSTF0.PVD1RF”、“RMU_RSTF0.PVD2RF”；“复位条件”章节描述“PWC_WKTC”修改为“PWC_WKTCR”；“功能复位控制寄存器 1 (RMU_FRST1)”章节复位值由“0xFFFFFFFF”修改为“0xFFBFFFFFF”；

文档版本日期	修改说明
Rev1.20 2025/03/28	3. 时钟控制器 (CMU)：“时钟输入模式”图中引脚由“XTAL_OUT”修改为“XTAL_OUT/XTAL_EXT”；“CMU HRC 控制寄存器 (CMU_HRCCR)”章节复位值描述由“ICG1.HRCSTP”修改为“ICG1.HRCSTOP”；“CMU ADC/DAC 时钟配置寄存器 (CMU_PERICKSEL)”章节 Bit[3:0]位功能说明新增描述“当 TRNG 动作时，该位必须设置为 0b0000”；“FCM 标志寄存器 (FCM_SR)”章节保留位由“31:8”修改为“31:3”； 4. 电源控制 (PWC)：“动作模式与低功耗模式”章节原描述重新整合成“模式概述”章节，优化描述形式使得表格与内容配合更符合阅读习惯；“停止模式”章节描述由“INT_WUPENR”修改为“INT_WKEN”；“PWC_FCG0 保护控制 (PWC_FCG0PC)”章节 Bit[31:16]位描述由“写入 0xA5A5 的同时改变 PRT0 位的值”修改为“在写入 PRT0 时，必须同时向 FCG0PCWE 写入 0xA5A5；读出恒为 0”，新增说明“向本寄存器写入 0xA5A50001 可以将 PRT0 置 1；写入 0xA5A50000 可以将 PRT0 清 0；禁止写入其他值修改 PRT0”；“PVD 滤波控制寄存器 (PWC_PVDFCR)”章节 Bit[6:5]位功能描述由“PVD2DNFE”修改为“PVD2NFDIS”，Bit[2:1]位功能描述由“PVD1DNFE”修改为“PVD1NFDIS”；“PVD 检测状态寄存器 (PWC_PVDDSR)”章节 Bit5 和 Bit1 位描述增加描述“写 1 无效”； 5. 初始化配置 (ICG)：“初始化配置寄存器 1 (ICG1)”章节 Bit[7:1]位功能描述删除“读出时所有位为 1，写入时所有位写 1”； 6. 嵌入式 Flash (EFM)：“单次编程无回读功能”、“单编程回读功能”、“擦除功能”和“全擦除功能”章节步骤 6 描述由“等待 Flash 操作处于空闲状态。(FM_FSR.RDY=1)”修改为“等待 Flash 操作结束并处于空闲状态。(EFM_FSR.OPTEND=1, EFM_FSR.RDY=1)”。 7. 内置 SRAM (SRAM)：“SRAM 校验控制保护寄存器 (SRAMC_CKPR)”章节 Bit[7:1]位功能描述由“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”修改为“当 CKPRC 需要置 1 或者清零时，必须同时向 CKPRKW 写入 0x3B”，增加说明“向本寄存器写入 0x00000077 可以将 CKPRC 置 1；写入 0x00000076 可以将 CKPRC 清零”； 8. 通用 IO (GPIO)：“寄存器总表”章节删除说明“32KHz 副振荡器复用端口 PCRC14 和 PCRC15 的复位值为 0x8100”；“写保护寄存器 (PWPR)”章节语言描述优化；“通用控制寄存器 (PCRxy)”章节 Bit8 位功能描述由“与寄存器 PIDRx 中 POUTy 功能一致”修改为“与寄存器 PIDRx.PINy 功能一致”；“32bit 访问”章节整合到“寄存器”章节； 9. 中断控制器 (INTC)：“事件输入”章节图中描述由“中断请求 NVIC IRQ0~32”修改为“中断请求 NVIC IRQ0~31”，事件输入 RXEV 通路上增加“/32”；“噪声滤波器”章节数字滤波器工作示意图优化；“寄存器总表”章节同类寄存器合并，偏移地址使用公式形式呈现；“外部中断控制寄存器 (INTC_EIRQCRx)”章节 Bit15 位功能描述由“滤波宽度由 INTC_NOCCR.NOCSEL 决定”修改为“滤波宽度由 NOCSEL 决定”；“中断事件使能寄存器 (INTC_INTENx)”章节位功能描述形式优化； 10. 自动运行系统 (AOS)：“寄存器总表”章节偏移地址使用公式形式呈现，ADC1 转换开始片内触发源选择寄存器的偏移地址由“0x74, 0x78”修改为“0x78+0x04*x”；“软件触发事件寄存器 (INTSFTTRG)”章节 Bit0 位功能描述形式优化；“PLU 控制寄存器 (AOS_PLUX_CR)”章节新增说明“当使用 PLU 运算结果作为 AOS 触发源时，请将 PLINySEL (y=A,B,C,D) 设为 0b00 或 0b10，且 PLMODE 设为 0b00”； 11. 存储保护单元 (MPU)：“寄存器总表”章节写保护描述修改为寄存器上标加上说明的形式；“区域范围描述寄存器 (MPU_RGDn)”章节 Bit[31:5]位功能描述形式优化

文档版本日期	修改说明
Rev1.20 2025/03/28	并补充举例，Bit[4:0]位功能描述取值格式由二进制优化为十进制数；“写保护寄存器（MPU_WP）”章节 Bit[15:1]位名由“WKEY[15:1]”修改为“WKEY”，功能描述由“必须同时向 WKEY 和 PUWE 写入 0x96A5”修改为“必须同时向 WKEY 写入 0x4B52”；“MSP 保护控制寄存器（MPU_MSPPCTL）”章节 Bit[15:2]位名由“MSPPSIZE[15:2]”修改为“MSPPSIZE”；“PSP 保护控制寄存器（MPU_PSPPCTL）”章节 Bit[15:2]位名由“PSPPSIZE[15:2]”修改为“PSPPSIZE”； 12. 键盘扫描控制模块（KEYSCAN）：“主要特性”、“按键识别功能”章节描述由“EIFR.EIFR”修改为“INTC_EIFR.EIF”；“扫描控制寄存器（KEYSCAN_SCR）”章节功能描述整体优化，使得更清晰； 13. DMA 控制器（DMA）：整个章节中涉及到的重载功能相关描述由“重载功能”修改为“重复传输功能”，“地址重载”修改为“地址重复跳转”；新增“软件触发寄存器（DMA_SWREQ）”，“启动 DMA”章节新增描述“DMA 可以通过软件写软件触发寄存器 DMA_SWREQ.SWREQ 启动 DMA 并开始传输”，“通道重置”章节新增描述“或通过软件写软件触发寄存器 DMA_SWREQ.SWRCFGREQ 触发通道重置功能”和“或软件写寄存器产生重置请求”；“寄存器”章节的子章节所有寄存器位功能描述补充通道相关描述，使之更清晰；“通道重置控制寄存器（DMA_RCFGCTL）”章节 Bit1 位功能描述由“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”修改为“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 Bit21-Bit16 全部无效”；“重复区域大小寄存器（DMA_RPTx）”章节说明描述由“地址重载功能”修改为“地址重复跳转”；“源地址不连续传输控制寄存器（DMA_SNSEQCTLx）”、“源地址不连续传输控制寄存器 B（DMA_SNSEQCTLBx）”、“目标地址不连续传输控制寄存器（DMA_DNSEQCTLx）”章节 Bit[19:0]位功能描述的举例删除冗余的信息“DMA_CHCTLx.HSIZE=0bxx”；“重复区域计数器监视寄存器（DMA_MONRPTx）”章节说明语言描述形式优化；“注意事项”章节语言描述形式优化； 14. 电压比较器（CMP）：“输入选择”章节表格写作形式优化，按照正负端输入重新整理数据；“窗口输出”章节描述由“CMP_MDR 寄存器的 BWEN 位为 1”修改为“CMP_OCR 寄存器的 BWEN 位为 1”； 15. 模数转换模块（ADC）：优化扩展通道内部模拟信号的描述，内部模拟通道、内部模拟输入、内部模拟输出都改为“内部模拟信号”；“功能框图”章节图形描述优化；“通道选择寄存器 A（ADC_CHSELRA）”、“通道选择寄存器 B（ADC_CHSELRB）”和“平均通道选择寄存器（ADC_AVCHSELR）”章节功能描述由“不存在通道的对应位为保留位，读出时为 0，写入时写 0”修改为“ADC1 支持通道 0~15，ADC2 支持通道 0~7，ADC3 支持通道 0~11。不支持的通道需要设置为 0”；“采样周期寄存器（ADC_SSTRx）”章节功能描述优化；“中断状态寄存器（ADC_ISR）”章节功能描述补充取值信息；“模拟看门狗状态寄存器（ADC_AWDSR）”章节功能描述补充取值信息； 16. 数模转换器（DAC）：“DAC 转换”章节描述由“改写 DADR 的值将触发一次新的 DAC 转换”修改为“改写 DAC_DADR1 的值将触发一次新的 DAC 转换”；“ADC 转换优先模式”章节描述由“DAADPCR.ADPSLn（n=0~2）用于选择 ADC 转换优先的 ADC 通道”修改为“DAADPCR.ADPSLn（n=1~3）用于选择 ADC 转换优先的 ADC 通道”，描述“确认 ADC 处于停止状态，设置 DACR.ADPSL”修改为“确认 ADC 处于停止状态，设置 DAC_DAADPCR.ADCSL1”；“数据寄存器（DAC_DADRY）”章节描述由“对同一单元的 DADR 进行 32 位操作可实现双通道同步转换”修改为“对 DAC_DADR1 进行 32 位操作可实现该单元双通道转换数据同时设置”；“控制寄存器

文档版本日期	修改说明
Rev1.20 2025/03/28	(DAC_DACR) ”章节 Bit12 位功能描述由“选择 DADR 作为通道 2 的转换数据”修改为“选择 DAC_DADR2 作为通道 2 的转换数据”，Bit11 位功能描述由“选择 DADR 作为通道 1 的转换数据”修改为“选择 DAC_DADR1 作为通道 1 的转换数据”； 17. 高级控制定时器 (Timer6)： “波形模式”、“比较输出”和“捕获输入”章节合并到“基本动作”章节；所有章节描述中“TIM6_xxx”修改为“TMR6_xxx”；“三角波计数方向”章节描述“设定计数方向无效”修改为“计数过程中设定计数方向无效”；“数字滤波”章节图“捕获输入端口的滤波功能”优化；“单元间多相 PWM 输出”章节优化语言描述，删除 x 相描述；“寄存器总表”章节新增说明“Timer6 为定时器名称，Timer6_<t> (<t>: 1~2) 为该定时器单元”和“触发选择寄存器 (TMR6_TRGSEL0~1) 是 2 个独立于单元的寄存器，为 2 个单元 Timer6 所共有”；“中断控制寄存器 (TMR6_ICONR) ”章节位功能描述形式优化，提取中断条件；“公共软件同步停止控制寄存器 (TMR6_SSTPR) ”、“公共软件同步清零控制寄存器 (TMR6_SCLRR) ”和“公共软件同步刷新控制寄存器 (TMR6_SUPDR) ”章节位功能描述优化，读写由“RW”修改为“WO”； 18. 通用控制定时器 (Timer4)： 所有章节描述中“TIM4_xxx”修改为“TMR4_xxx”；“通用比较寄存器缓存功能”章节描述由“端口输出根据新的 OCCR 数据执行操作”修改为“端口输出控制根据新的 OCCR 数据执行操作”；“通用控制状态寄存器 (TMR4_OCSRn) ”章节 Bit3 和 Bit2 位功能描述形式优化，补充读写取值描述；“通用模式控制寄存器 (TMR4_OCMRm) ”章节描述形式优化；“重载控制状态寄存器 (TMR4_RCSR) ”章节位功能描述形式优化；“PWM 滤波控制寄存器 (TMR4_PFSRn) ”章节描述由“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”修改为“16 位滤波计数器用作 16 位重载计数器，可以周期性产生中断输出”； 19. 紧急刹车模块 (EMB)： “外部端口输入电平变化时控制 PWM 信号输出”章节 EMB 端口分配表的端口名称由“PORTx”修改为“EMB_INx”；“PWM 输出端口电平发生同相 (同高或同低) 时停止 PWM 信号输出”章节描述由“EMB_STAT.PWMF”修改为“EMB_STAT.PWMSF”，“EMB_STATCLR.PWMFCLR”修改为“EMB_STATCLR.PWMSFCLR”；“根据电压比较器比较结果停止 PWM 信号输出”章节描述由“EMB_RLSSEL.CMPRSEly”修改为“EMB_RLSSEL.CMPRSEL”；“EMB 状态寄存器 (EMB_STAT) ”章节 Bit6 位功能描述形式优化； 20. 通用定时器 (TimerA)： 所有章节描述中“TIMA_xxx”修改为“TMRA_xxx”；“数字滤波”章节图“时钟输入端口的滤波功能”优化；“中断控制寄存器 (TMRA_ICONR) ”、“事件控制寄存器 (TMRA_ECONR) ”、“状态标志寄存器 (TMRA_STFLR) ”章节位功能描述形式优化；“控制状态寄存器 H (TMRA_BCSTRH) ”章节 Bit0 位功能描述新增说明“溢出停止后再次启动定时器需初始化 TMRA_CNTER 值”； 21. 实时时钟 (RTC)： 调整部分章节的顺序，归类到“操作示例”章节；“日计数寄存器 (RTC_DAY) ”章节增加描述“请写入十进制 1~28、1~29、1~30 或 1~31 的 BCD 码，写入超出范围的值将被忽略”；“时闹钟寄存器 (RTC_ALMHOUR) ”章节描述由“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”修改为“请根据控制位 RTC_CR1.AMPM 的值，设定正确的十进制 0~23 或者 01~12，21~32 的 BCD 码，写入超出范围的值将不会发生闹钟匹配”；“周闹钟寄存器 (RTC_ALMWEEK) ”章节描述由“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”修改为“十进制 0~6 表示周日~周六。请写入十进制 0~6 的 BCD 码，写入超出范围的值将不会发生闹钟匹配”；

文档版本日期	修改说明
Rev1.20 2025/03/28	22. 看门狗计数器 (WDT/SWDT)：“标志位”章节新增描述“用户对标志位清零后，需再次读取标志位查询清零操作是否成功”，同时新增标志位清零步骤； 23. 通用同步异步收发器 (USART)：“接收器”章节 UART 接收器的容差表格描述形式优化，合并为一个；“UART 功能-RX 线唤醒停止模式功能”、“UART 功能-中断和事件”章节寄存器名称由“INT_WUPEN.RXWUEN”修改为“INTC_WKEN.RXWKEN”；“LIN 接收”章节寄存器名称由“USARTn_CR2.WKUP”修改为“USARTn_CR2.WKUPE”；“波特率寄存器 (USART_BRR)”章节公式中%表现形式优化；“LIN 接收”章节“唤醒信号帧的检测”增加描述“检测到唤醒信号帧后需要清除 USARTn_SR.WKUP 标志，并读取 USARTn_RDR.RDR 寄存器”，接收数据设定步骤的步骤 6 描述由“清除 USARTn_SR.PE/FE/LBD 标志”修改为“清除 USARTn_SR.FE/LBD 标志”，步骤 7 描述由“并确认接收到的数据为 0x55”修改为“接收数据若为 0x55，说明波特率误差在可接受范围内；若接收数据不为 0x55，用户校准波特率为必须步骤，用户根据 USARTn_LBMC 寄存器来重新校准波特率”；“状态寄存器 (USART_SR)”章节 Bit1 位功能描述新增说明“在 LIN 模式接收到间隔段时，会产生该标志位，不代表帧错误；在接收到同步段时，若波特率误差较大，也会产生该标志位”；“控制寄存器 1 (USART_CR1)”章节 Bit9 位功能描述新增说明“智能卡模式时该位必须保持为 0”，Bit5 位功能描述由“接收中断”修改为“接收中断、错误中断”；“控制寄存器 2 (USART_CR2)”章节 Bit17 和 Bit16 位功能描述形式优化合并为一个表格信息，新增修改说明为“该位只在 LINEN=1 时有效”； 24. 集成电路总线 (I2C)：“开始条件”、“地址传送”、“数据传送”、“停止条件”、“重新开始条件”章节合并为“数据传输”；“数据传输”章节“数据传送”中图“7 位地址格式的主机接收数据的时序图 (例)”优化移除寻址阶段后的 I2C_DRR 读操作；“握手”章节从机发送模式描述寄存器由“I2CDT”修改为“I2C_DTR”，“移位寄存器 (I2C_DSR 寄存器)”修改为“内部移位寄存器”；“从机地址匹配”、“广播地址匹配”、“SMBus 主机地址匹配”、“SMBus 报警响应地址匹配”、“SMBus 默认地址匹配”章节描述由“在 SCL 时钟的第 9 个时钟的下降沿”修改为“在 SCL 时钟的第 9 个时钟的上升沿”；“SMBus 超时测量”章节描述由“在总线忙的状态下，能通过检测到 SCL 线的低电平或者高电平被固定了一定的时间以上，并且检测到总线的异常状态”修改为“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 先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”；“复位”章节描述“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”修改为“一种是通信复位：I2C 内部的全部寄存器 (I2C_FSTR[3:0]除外) 及内部状态复位；另一种内部状态复位：I2C_SR (包括 BUSY 位)，内部移位寄存器和内部状态机进行复位”；“可编程数字滤波”章节图中描述由“PCLK3”修改为“PCLK0”；“I2C 控制寄存器 1 (I2C_CR1)”章节 Bit15 位功能描述由“通信复位：I2C 内部的全部寄存器及内部状态复位”修改为“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”，Bit6 位名由“ENGCG”修改为“GCEN”，同步修改“广播地址匹配”章节寄存器名称；“I2C 从机地址寄存器 0 (I2C_SLR0)”章节 Bit[23:16]位描述由“MSLADDR0[9:0]”修改为“MSLADDR0[7:0]”；“I2C 从机地址寄存器 1 (I2C_SLR1)”章节 Bit[23:16]位描述由“MSLADDR1[9:0]”修改为“MSLADDR1[7:0]”；删除无法操作的“数据移位寄存器 (I2C_DSR)”，全文描述替换为“内部移位寄存器”；“I2C 数据发送寄存器 (I2C_DTR)”、“I2C 数据接收寄存器 (I2C_DRR)”章节语言描述形式优化；“I2C

文档版本日期	修改说明
Rev1.20 2025/03/28	时钟控制寄存器 (I2C_CCR) ”章节 Bit[18:16]位名由“FREQ”修改为“CKDIV”，波特率计算公式描述形式优化； 25. 串行外设接口 (SPI)：整体章节有关缓冲区/区域的描述统一修改为“发送缓冲器”或者“接收缓冲器”；“数据格式”章节数据格式图中描述由“SPCMDm.SPB[3:0]”修改为“SPI_CFG2.DSIZE[3:0]”；“奇偶校验位自诊断”章节图中的寄存器描述错误，修改为正确的寄存器位名；“SPI 的初始化”章节描述由“系统复位”修改为“系统复位或使用功能复位寄存器 RMU_FRST1”；“中断和事件”章节描述形式优化；“SPI 通信配置寄存器 2 (SPI_CFG2) ”章节 Bit4 位默认为 1； 26. 四线式串行外设接口 (QSPI)：“内部总线空间”和“QSPI 外部扩展地址寄存器 (QSPI_EXAR) ”章节描述由“64MB×63 块”修改为“64MB×64 块”；“ROM 空间及总线的地址宽度”章节说明新增“选择 8 位地址宽度时，使用标准读指令仅支持访问低 256 字节空间，不支持访问镜像空间”；“QSSN 的建立时间”章节描述由“第一个上升沿”修改为“第一个边沿”；“QSSN 的保持时间”章节描述由“最后一个上升沿”修改为“最后一个边沿”；“现有的 QSPI-ROM 指令参考”章节退出 4-Bytes 模式和进入 4-Bytes 模式的指令代码替换为正确的指令，删除重复的 3-Bytes 地址模式的标准读指令；“直接通信模式下 QSPI 总线周期的生成”章节说明描述由“在直接通信模式下是无法对 QSPI_CR 和 QSPI_DCOM 以外的寄存器进行写操作的。对其他寄存器的写操作将会退出直接通信模式。使用这种方式退出可能导致不可预料的情况出现所以不推荐使用”修改为“在直接通信模式下对 QSPI_DCOM 以外的 QSPI 寄存器进行写操作，都会退出直接通信模式。使用“QSPI_CR 寄存器写操作”以外的方式退出直接通信模式可能会导致不可预料的情况出现，所以不推荐使用”，新增说明“对 QSPI_DCOM 寄存器进行写操作时，必须使用 32 位或 16 位操作，同时写入 DCOMPRSL[1:0]和 DCOM[7:0]位。对 QSPI_DCOM 寄存器进行读操作时，可以使用 32 位、16 位或 8 位操作”；“QSPI 片选控制寄存器 (QSPI_CSCR) ”章节 Bit[5:4]位功能描述新增说明“直接模式下该位设定无效”；寄存器“QSPI_SR2”名称修改为“QSPI_CLR”；“注意事项-QSPI 寄存器的设置”和“模块停止位的设置”章节语言描述优化； 27. 外部存储器控制器 (EXMC)：“SMC 基本功能”新增描述“支持 LCD 并行接口，兼容 8080/6800 模式”； 28. 控制器局域网 (MCAN)：“超时计数器”章节说明描述由“在仲裁场 (Arbitration field) 和数据场 (Data field) ”修改为“在仲裁阶段 (Arbitration phase) 和数据阶段 (Data phase) ”；“发送队列”章节描述由“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”修改为“如果多个发送队列缓冲区配置了相同的消息 ID，则发送顺序取决于存储消息的发送队列缓冲区的编号”；“MCAN 数据位时间和预分频寄存器 (MCAN_DBTP) ”章节 Bit[7:4]位功能描述由“有效值为 0 到 15”修改为“有效设定值为 1~15”；“MCAN 协议状态寄存器 (MCAN_PSR) ”章节 Bit14 位功能描述删除“访问类型为 RX：进行读访问时复位”；“MCAN 接收 FIFO1 状态寄存器 (MCAN_RXF1S) ”章节 Bit25 位增加描述“该位是中断标志 MCAN_IR.RF1L 的副本。当 MCAN_IR.RF1L 被复位时，该位也被复位”；“MCAN 发送缓冲区配置寄存器 (MCAN_TXBC) ”章节 Bit[15:2]位名由“TBSA[13:0]”修改为“TBSA[15:2]”；“MCAN 发送事件 FIFO 状态寄存器 (MCAN_TXEFS) ”章节 Bit[5:0]位功能描述由“接收 FIFO0 中存储的元素个数”修改为“发送事件 FIFO 中存储的元素个数”； 29. 加密协处理模块 (CPM)：“加解密算法处理器 (AES) ”章节“加密操作流程”、“解密操作流程”、“数据示例”和“操作注意事项”整合到操作示例章节；“加解密算法处

文档版本日期	修改说明
Rev1.20 2025/03/28	理器（AES）-操作示例-数据示例”章节 192 位密钥和 256 位密钥的对应密文修改为正确的值；“密钥寄存器（AES_KRx）（x = 0~7）”章节说明描述形式优化；“HASH 摘要寄存器 n（HASH_HRn）（n=0~7）”章节读写由“RW”修改为“RO”，删除描述“硬件将自动忽略对本寄存器的写操作”； 30. 数据计算单元（DCU）：“硬件触发启动模式”章节描述由“在触发加模式时，每发生一次事件触发，DCU 将启动并进行一次（DATA0 + DATA1）/2 的运算，DATA0 + DATA1 的结果存入 DATA0，而（DATA0 + DATA1）/2 的结果存入 DATA2”修改为“在触发加模式时，每发生一次事件触发，DCU 将启动并进行一次 DATA0+DATA1 的加法运算，加法运算结果存入 DATA0，加法运算结果除以 2 存入 DATA2”，“DCU_INTSEL”修改为“DCU_INTEVTSEL”；“递增锯齿波输出模式”章节步骤 2 增加描述“DATA1.LWL 的值自动加载到 DATA0.DO 并输出给 DAC”，步骤 6 删除描述“第一次产生触发事件时，DATA1.LWL 的值自动加载到 DATA0.DO 并输出给 DAC，以后”；“递减锯齿波输出模式”章节步骤 2 的描述“DATA1.UPL 的值加载到 DATA0.DO 并输出给 DAC”移动到步骤 6 中，修改为“启动定时器，第一次产生触发事件时，DATA1.UPL 的值加载到 DATA0.DO 并输出给 DAC，以后 xxx”；“DCU 中断和事件寄存器（DCU_INTEVTSEL）”章节 Bit[8:0]位功能描述中“INT_WIN”修改为“SEL_WIN”； 31. CRC 运算（CRC）：整体章节描述形式优化，重新按照协议整理； 32. 调试控制器（DBGC）：“引脚排列和调试端口引脚”章节描述由“除非明确编程，否则不分配跟踪输出”修改为“若要使用跟踪输出，请配置寄存器 MCUTRACECTL”；
Rev1.00 2024/05/27	第一次正式发布。