



HC32F460 系列

32 位 ARM[®] Cortex[®]-M4 微控制器

用户手册

声 明

- 华大半导体有限公司（以下简称：“HDSC”）保留随时更改、更正、增强、修改华大半导体产品和/或本文档的权利，恕不另行通知。用户可在下单前获取最新相关信息。HDSC 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 HDSC 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。
- HDSC 在此确认未以明示或暗示方式授予任何知识产权许可。
- HDSC 产品的转售，若其条款与此处规定不同，HDSC 对此类产品的任何保修承诺无效。
- 任何带有“®”或“™”标识的图形或字样是 HDSC 的商标。所有其他在 HDSC 产品上显示的产品或服务名称均为其各自所有者的财产。
- 本通知中的信息取代并替换先前版本中的信息。

©2021 华大半导体有限公司 - 保留所有权利

目 录

声 明	2
目 录	3
简介 (Overview)	43
1 存储器映射 (Memory Mapping)	44
1.1 存储器映射.....	44
1.2 外部空间映射.....	50
1.3 位段空间.....	50
1.4 地址重映射.....	51
1.5 重映射寄存器.....	52
1.5.1 访问保护寄存器 (MMF_REMPRT)	52
1.5.2 重映射控制寄存器 (MMF_REMCRx) (x=0, 1)	53
2 总线架构 (BUS)	54
2.1 概述.....	54
2.2 总线架构.....	55
2.3 总线功能.....	56
3 复位控制 (RMU)	57
3.1 简介.....	57
3.2 复位方式和复位标志位.....	58
3.3 复位时序.....	60
3.3.1 上电复位.....	60
3.3.2 NRST 引脚复位.....	61
3.3.3 欠压复位.....	61
3.3.4 可编程电压检测 1 复位、可编程电压检测 2 复位.....	62
3.3.5 看门狗复位、专用看门狗复位.....	63
3.3.6 掉电唤醒复位.....	64
3.3.7 软件复位.....	64
3.3.8 MPU 错误复位	65
3.3.9 RAM 奇偶校验复位.....	65
3.3.10 RAMECC 复位	66
3.3.11 时钟频率异常复位.....	66
3.3.12 外部高速振荡器异常停振复位.....	67
3.3.13 复位方式的判断.....	67
3.3.14 各个模块的复位条件.....	68
3.4 寄存器说明.....	69
3.4.1 复位标志寄存器 0 (RMU_RSTF0)	69
4 时钟控制器 (CMU)	71
4.1 简介.....	71
4.2 系统框图.....	72
4.2.1 系统框图.....	72
4.2.2 时钟频率测量框图	73

4.3	时钟源规格.....	74
4.4	工作时钟规格.....	76
4.5	晶振电路.....	78
4.5.1	外部高速振荡器.....	78
4.5.1.1	振荡器模式.....	78
4.5.1.2	时钟输入模式.....	79
4.5.2	外部高速振荡器故障检测	80
4.5.2.1	检测到 XTAL 故障检测动作	81
4.5.2.2	检测到 XTAL 振荡故障产生中断复位	82
4.5.3	外部低速振荡器	83
4.6	内部 RC 时钟	84
4.6.1	HRC 时钟.....	84
4.6.2	MRC 时钟	84
4.6.3	LRC 时钟	85
4.6.4	SWDTRC 时钟	85
4.7	PLL 时钟	86
4.8	时钟切换步骤.....	86
4.8.1	时钟源切换.....	87
4.8.2	时钟分频切换	88
4.9	时钟输出功能.....	89
4.10	时钟频率测量.....	90
4.10.1	时钟频率测量.....	90
4.10.2	数字滤波功能	91
4.10.3	中断/复位功能.....	91
4.11	寄存器说明	92
4.11.1	CMU_XTAL 配置寄存器(CMU_XTALCFGR)	94
4.11.2	CMU_XTAL 安定配置寄存器(CMU_XTALSTBCR)	94
4.11.3	CMU_XTAL 控制寄存器(CMU_XTALCR)	95
4.11.4	CMU_XTAL 振荡故障控制寄存器(CMU_XTALSTDSCR).....	96
4.11.5	CMU_XTAL 振荡故障状态寄存器(CMU_XTALSTDSSR)	97
4.11.6	CMU_XTAL32 配置寄存器(CMU_XTAL32CFGR)	97
4.11.7	CMU_XTAL32 滤波寄存器(CMU_XTAL32NFR).....	98
4.11.8	CMU_XTAL32 控制寄存器(CMU_XTAL32CR)	98
4.11.9	CMU_HRC 校准寄存器(CMU_HRCTRM)	99
4.11.10	CMU_HRC 控制寄存器(CMU_HRCCR).....	100
4.11.11	CMU_MRC 校准寄存器(CMU_MRCTRL)	101
4.11.12	CMU_MRC 控制寄存器(CMU_MRCCR)	102
4.11.13	CMU_LRC 校准寄存器(CMU_LRCTRL)	103
4.11.14	CMU_LRC 控制寄存器(CMU_LRCCR)	104
4.11.15	CMU_MPLL 配置寄存器(CMU_PLLCFGR)	105
4.11.16	CMU_MPLL 控制寄存器(CMU_PLLCR)	107
4.11.17	CMU_UPLL 配置寄存器(CMU_UPLLCFGR).....	108

4.11.18	CMU UPLL 控制存器(CMU_UPLLCR).....	110
4.11.19	CMU 时钟源稳定状态器(CMU_OSCSTBSR)	111
4.11.20	CMU 系统时钟源切换寄存器(CMU_CKSWR).....	112
4.11.21	CMU 时钟分频配置寄存器(CMU_SCFGR).....	113
4.11.22	CMU USBFS 时钟配置存器(CMU_UFSCKCFG).....	116
4.11.23	CMU AD/TRNG 时钟配置存器(CMU_PERICKSEL).....	117
4.11.24	CMU I2S 时钟配置存器(CMU_I2SCKSEL).....	118
4.11.25	CMU 调试时钟配置存器(CMU_TPIUCKCFG).....	119
4.11.26	CMU MCO1 配置存器(CMU_MCO1CFG).....	120
4.11.27	CMU MCO2 配置存器(CMU_MCO2CFG).....	121
4.11.28	FCM 下限比较值寄存器(FCM_LVR)	122
4.11.29	FCM 上限比较值寄存器(FCM_UVR)	122
4.11.30	FCM 计数器值寄存器(FCM_CNTR)	123
4.11.31	FCM 开始停止寄存器(FCM_STR)	123
4.11.32	FCM 测量对象控制寄存器(FCM_MCCR)	124
4.11.33	FCM 测量基准控制寄存器(FCM_RCCR)	125
4.11.34	FCM 中断复位控制寄存器(FCM_RIER).....	126
4.11.35	FCM 标志寄存器(FCM_SR).....	127
4.11.36	FCM 标志位清除寄存器(FCM_CLR).....	127
5	电源控制 (PWC)	128
5.1	简介.....	128
5.2	电源分布.....	129
5.3	电源电压检测单元(PVD)说明	131
5.3.1	上电复位/掉电复位动作说明	131
5.3.2	欠压复位(BOR)说明	132
5.3.3	可编程电压检测 1(PVD1)、可编程电压检测 2(PVD2).....	133
5.3.4	PVD1、 PVD2 中断/复位框图	134
5.3.5	输入/输出引脚.....	134
5.3.6	PVD1 中断和复位	135
5.3.7	PVD2 中断和复位	137
5.3.8	内部电压采样和检测功能	138
5.4	动作模式与低功耗模式.....	139
5.4.1	运行模式.....	143
5.4.2	睡眠模式.....	146
5.4.3	停止模式.....	147
5.4.4	掉电模式.....	149
5.5	降低功耗的方法.....	154
5.5.1	降低系统时钟速度	154
5.5.2	关闭不使用的时钟源	154
5.5.3	功能时钟停止	155
5.5.4	关闭不使用的 RAM	155
5.6	寄存器保护功能.....	156

5.7	寄存器说明.....	157
5.7.1	电源模式控制寄存器 0 (PWC_PWRC0)	159
5.7.2	电源模式控制寄存器 1 (PWC_PWRC1)	160
5.7.3	电源模式控制寄存器 2 (PWC_PWRC2)	161
5.7.4	电源模式控制寄存器 3 (PWC_PWRC3)	161
5.7.5	掉电唤醒使能寄存器 0(PWC_PDWKE0).....	162
5.7.6	掉电唤醒使能寄存器 1(PWC_PDWKE1).....	163
5.7.7	掉电唤醒使能寄存器 2(PWC_PDWKE2).....	164
5.7.8	掉电唤醒事件边沿选择寄存器(PWC_PDWKES).....	165
5.7.9	掉电唤醒标志寄存器 0(PWC_PDWKF0).....	166
5.7.10	掉电唤醒标志寄存器 1(PWC_PDWKF1).....	167
5.7.11	电源监测控制寄存器 (PWC_PWCMR).....	168
5.7.12	模式切换控制寄存器(PWC_MDSWCR).....	168
5.7.13	功能时钟控制 0(PWC_FCG0).....	169
5.7.14	功能时钟控制 1(PWC_FCG1).....	171
5.7.15	功能时钟控制 2(PWC_FCG2).....	173
5.7.16	功能时钟控制 3(PWC_FCG3).....	175
5.7.17	PWC_FCG0 保护控制(PWC_FCG0PC).....	177
5.7.18	功能保护控制寄存器(PWC_FPRC)	178
5.7.19	STOP 模式控制寄存器(PWC_STPMCR)	179
5.7.20	RAM 功耗控制寄存器 0(PWC_RAMPC0).....	180
5.7.21	RAM 运行条件寄存器(PWC_RAMOPM).....	182
5.7.22	XTAL32 用电流源控制 (PWC_XTAL32CS)	182
5.7.23	唤醒计时器控制寄存器(PWC_WKTCR)	183
5.7.24	PVD 控制寄存器 0(PWC_PVDCR0).....	183
5.7.25	PVD 控制寄存器 1(PWC_PVDCR1).....	184
5.7.26	PVD 滤波控制寄存器(PWC_PVDFCR)	185
5.7.27	PVD 电平控制寄存器(PWC_PVDLCR)	186
5.7.28	PVD 中断控制寄存器(PWC_PVDICR)	187
5.7.29	PVD 检测状态寄存器(PWC_PVDDSR)	188
6	初始化配置 (ICG)	189
6.1	简介.....	189
6.2	寄存器说明.....	190
6.2.1	初始化配置寄存器 0(ICG0).....	190
6.2.2	初始化配置寄存器 1(ICG1).....	193
6.2.3	初始化配置寄存器 n(ICGn)n=2~7	194
7	嵌入式 FLASH (EFM)	195
7.1	简介.....	195
7.2	主要特性.....	195
7.3	嵌入式 FLASH	196
7.4	读接口.....	198
7.4.1	CPU 时钟和 FLASH 读取时间之间的关系.....	198

7.4.2	FLASH 低功耗读	198
7.5	FLASH 读加速缓存.....	201
7.6	FLASH 编程和擦除操作	201
7.6.1	单次编程无回读模式.....	202
7.6.2	单编程回读模式.....	202
7.6.3	连续编程操作	202
7.6.4	擦除操作	203
7.6.5	总线保持功能	204
7.6.6	FLASH 擦除、编程窗口保护	204
7.6.7	中断	204
7.7	一次性可编程字节.....	206
7.8	引导交换.....	207
7.9	寄存器说明.....	208
7.9.1	访问保护寄存器 (EFM_FAPRT)	209
7.9.2	FLASH 停止寄存器 (EFM_FSTP)	209
7.9.3	读模式寄存器 (EFM_FRMC)	210
7.9.4	擦写模式寄存器 (EFM_FWMC)	211
7.9.5	状态寄存器 (EFM_FSR)	212
7.9.6	状态清除寄存器 (EFM_FSCLR)	213
7.9.7	中断许可寄存器 (EFM_FITE)	214
7.9.8	引导交换状态寄存器 (EFM_FSWP)	215
7.9.9	FLASH 窗口保护起始地址寄存器 (EFM_FPMTSW)	215
7.9.10	FLASH 窗口保护结束地址寄存器 (EFM_FPMTEW)	216
7.9.11	UNIQUE ID 寄存器 (EFM_UQID1)	216
7.9.12	UNIQUE ID 寄存器 (EFM_UQID2)	217
7.9.13	UNIQUE ID 寄存器 (EFM_UQID3)	217
7.10	注意事项.....	218
8	内置 SRAM (SRAM)	219
8.1	简介.....	219
8.2	寄存器说明.....	221
8.2.1	SRAM 等待控制寄存器 (SRAM_WTCR)	222
8.2.2	SRAM 等待保护寄存器 (SRAM_WTPR)	224
8.2.3	SRAM 校验控制寄存器 (SRAM_CKCR)	225
8.2.4	SRAM 校验保护寄存器 (SRAM_CKPR)	226
8.2.5	SRAM 校验状态寄存器 (SRAM_CKSR)	227
9	通用 IO (GPIO)	228
9.1	简介.....	228
9.2	端口功能概要.....	229
9.3	动作说明.....	230
9.3.1	通用输入输出 GPIO 功能.....	230
9.3.2	周边功能.....	231
9.3.3	双周边功能.....	231

9.3.4	Event Port 输入输出功能	231
9.3.5	外部中断 EIRQ 输入功能.....	232
9.3.6	模拟功能.....	232
9.3.7	通用控制.....	232
9.4	寄存器说明.....	234
9.4.1	通用输入寄存器 (PIDRx)	236
9.4.2	通用输出数据寄存器 (PODRx)	236
9.4.3	通用输出许可寄存器 (POERx)	237
9.4.4	通用输出置位寄存器 (POSRx)	237
9.4.5	通用输出复位寄存器 (PORRx)	238
9.4.6	通用输出翻转寄存器 (POTRx)	238
9.4.7	特殊控制寄存器 (PSPCR)	239
9.4.8	公共控制寄存器 (PCCR)	239
9.4.9	输入控制寄存器 (PINAER)	240
9.4.10	写保护寄存器 (PWPR)	240
9.4.11	通用控制寄存器 (PCRxy)	241
9.4.12	功能选择寄存器 (PFSRxy)	243
9.4.13	Event Port 方向选择寄存器 (PEVNTDIRm)	244
9.4.14	Event Port 输入数据寄存器 (PEVNTIDRm)	245
9.4.15	Event Port 输出数据寄存器 (PEVNTODRm)	245
9.4.16	Event Port 输出数据复位寄存器 (PEVNTORRm)	246
9.4.17	Event Port 输出数据置位寄存器 (PEVNTOSRm)	246
9.4.18	Event Port 上升沿输入许可寄存器 (PEVNTRISRm)	247
9.4.19	Event Port 下降沿输入许可寄存器 (PEVNTFALRm)	248
9.4.20	Event Port 输入滤波控制寄存器 (PEVNTNFCR)	249
9.4.21	32bit 访问.....	251
9.5	注意事项.....	252
10	中断控制器 (INTC)	253
10.1	简介.....	253
10.2	INTC 系统框图	255
10.2.1	系统框图.....	255
10.3	向量表.....	256
10.3.1	中断向量表.....	256
10.3.2	中断事件请求序号	264
10.4	功能说明.....	283
10.4.1	不可屏蔽中断	283
10.4.2	外部管脚中断事件请求	284
10.4.3	中断源选择	284
10.4.4	软件中断	284
10.4.5	中断/事件选择	285
10.4.6	WFE 唤醒事件管理	285
10.4.7	数字滤波器	286

10.4.8	低功耗模式返回	288
10.4.8.1	休眠模式返回.....	288
10.4.8.2	停止模式返回.....	288
10.4.8.3	掉电模式返回.....	288
10.4.8.4	不可屏蔽中断和 WFI 指令	288
10.4.9	内部触发事件	289
10.5	寄存器说明.....	290
10.5.1	NMI 管脚不可屏蔽中断控制寄存器 (INT_NMICR)	296
10.5.2	不可屏蔽中断使能寄存器 (INT_NMIENR)	297
10.5.3	不可屏蔽中断标志寄存器 (INT_NMIFR)	298
10.5.4	不可屏蔽中断标志清除寄存器 (INT_NMICFR)	299
10.5.5	外部管脚中断控制寄存器 (INT_EIRQCRx) (x=0~15).....	300
10.5.6	外部管脚中断标志寄存器 (INT_EIFR)	301
10.5.7	外部管脚中断标志清除寄存器 (EICFR)	301
10.5.8	中断/事件选择寄存器 (INT_SEL0~31).....	302
10.5.9	中断选择寄存器 (INT_SEL32~127)	303
10.5.10	向量共享中断选择寄存器 (INT_VSSEL128~143)	304
10.5.11	停止模式唤醒事件使能寄存器 (INT_WUPEN)	305
10.5.12	软件中断/事件寄存器 (INT_SWIER)	306
10.5.13	事件使能寄存器 (INT_EVTER)	306
10.5.14	中断使能寄存器 (INT_IER).....	307
10.6	使用注意事项	308
11	自动运行系统 (AOS)	309
11.1	简介	309
11.1.1	功能概览	309
11.1.2	模块示意图	310
11.2	功能描述	311
11.2.1	AOS 源事件列表	311
11.2.2	AOS 目标列表	311
11.3	动作说明	312
11.3.1	专用触发源	312
11.3.2	公共触发源	312
11.4	寄存器说明	313
11.4.1	外设触发事件寄存器(INTSFTTRG)	314
11.4.2	DCU 触发源选择寄存器(DCU_TRGSELx)(x=1~4).....	315
11.4.3	DMA1 传输启动触发源选择寄存器(DMA1_TRGSELx)(x=0~3).....	316
11.4.4	DMA2 传输启动触发源选择寄存器(DMA2_TRGSELx)(x=0~3).....	317
11.4.5	DMA 通道重置触发源选择寄存器(DMA_TRGSELRC).....	318
11.4.6	Timer6 硬件触发事件选择寄存器(TMR6_HTSSRx)(x=0~1).....	319
11.4.7	Timer0 硬件触发事件选择寄存器(TMR0_HTSRR).....	320
11.4.8	Event Port 触发源选择寄存器(PEVNTTRGSR12, PEVNTTRGSR34)	321
11.4.9	TimerA 内部触发事件选择寄存器 0(TMRA_HTSSR0)	322

11.4.10	TimerA 内部触发事件选择寄存器 1(TMRA_HTSSR1)	323
11.4.11	OTS 触发源选择寄存器(OTS_TRG)	324
11.4.12	A/D1 转换开始片内触发源选择寄存器 ADC1_ITRGSELRx(x=0,1)	325
11.4.13	A/D2 转换开始片内触发源选择寄存器 ADC2_ITRGSELRx(x=0,1)	326
11.4.14	公共触发源选择寄存器 1(AOS_COMTRG1)	327
11.4.15	公共触发源选择寄存器 2(AOS_COMTRG2)	327
12	键盘扫描控制模块 (KEYSCAN)	328
12.1	简介	328
12.2	KEYSCAN 系统框图	329
12.3	管脚说明	329
12.4	功能说明	330
12.4.1	按键识别功能	330
12.4.2	键盘扫描功能	330
12.4.3	使用上的注意事项	331
12.5	寄存器说明	331
12.5.1	KEYSCAN 扫描控制寄存器 (KEYSCAN_SCR)	332
12.5.2	KEYSCAN 扫描使能寄存器 (KEYSCAN_SER)	334
12.5.3	KEYSCAN 扫描状态寄存器 (KEYSCAN_SSR)	335
13	存储保护单元 (MPU)	336
13.1	简介	336
13.2	功能描述	337
13.2.1	区域范围设置	337
13.2.2	权限设置	337
13.2.3	MPU 动作选择	337
13.2.4	启动 MPU	337
13.3	应用举例	338
13.3.1	只允许部分空间访问	338
13.3.2	只禁止部分空间访问	338
13.4	寄存器说明	339
13.4.1	区域范围描述寄存器 MPU_RGDn (n=0~15)	340
13.4.2	区域控制寄存器 MPU_RGCRn (n=0~15)	341
13.4.3	控制寄存器 MPU_CR	342
13.4.4	状态标志寄存器 MPU_SR	344
13.4.5	标志清除寄存器 MPU_ECLR	345
13.4.6	写保护寄存器 MPU_WP	346
13.4.7	IP 访问保护寄存器 MPU_IPPR	347
14	DMA 控制器 (DMA)	349
14.1	简介	349
14.2	模块示意图	350
14.3	功能描述	351
14.3.1	使能 DMA 控制器	351
14.3.2	通道选择和通道优先级	351

14.3.3	启动 DMA.....	351
14.3.4	数据块.....	351
14.3.5	传输地址控制.....	352
14.3.6	传输次数.....	352
14.3.7	中断和事件信号输出.....	352
14.3.8	连锁传输.....	353
14.3.9	不连续地址传输.....	355
14.3.10	通道重置.....	356
14.3.11	传输提前终止.....	358
14.4	应用举例.....	359
14.4.1	存储器到存储器的传输.....	359
14.4.2	存储器到外围电路的传输.....	361
14.4.3	存储器到存储器的连锁传输.....	363
14.5	寄存器说明.....	365
14.5.1	DMA 使能寄存器(DMA_EN)	367
14.5.2	中断状态寄存器 0(DMA_INTSTAT0)	367
14.5.3	中断状态寄存器 0(DMA_INTSTAT1)	368
14.5.4	中断屏蔽寄存器(DMA_INTMASK0).....	368
14.5.5	中断屏蔽寄存器(DMA_INTMASK1).....	369
14.5.6	中断复位寄存器(DMA_INTCLR0).....	369
14.5.7	中断复位寄存器(DMA_INTCLR1).....	370
14.5.8	通道使能寄存器(DMA_CHEN)	370
14.5.9	通道重置控制寄存器(DMA_RCFGCTL)	371
14.5.10	传输请求状态寄存器(DMA_REQSTAT)	373
14.5.11	通道状态观测寄存器(DMA_CHSTAT)	374
14.5.12	传输源地址寄存器(DMA_SARx) (x=0~3)	374
14.5.13	传输目标地址寄存器(DMA_DARx) (x=0~3).....	375
14.5.14	数据控制寄存器(DMA_DTCTLx) (x=0~3)	375
14.5.15	重复区域大小寄存器(DMA_RPTx) (x=0~3)	376
14.5.16	重复区域大小寄存器 B(DMA_RPTBx) (x=0~3).....	377
14.5.17	源设备不连续地址传输控制寄存器(DMA_SNSEQCTLx) (x=0~3)	378
14.5.18	源设备不连续地址传输控制寄存器 B(DMA_SNSEQCTLBx) (x=0~3)	379
14.5.19	目标设备不连续地址传输控制寄存器(DMA_DNSEQCTLx) (x=0~3)	380
14.5.20	目标设备不连续地址传输控制寄存器 B(DMA_DNSEQCTLBx) (x=0~3)	381
14.5.21	链指针寄存器(DMA_LLPx) (x=0~3)	382
14.5.22	通道控制寄存器(DMA_CHxCTL) (x=0~3)	383
14.5.23	通道监视寄存器(DMA_MONSARx, DMA_MONDARx, DMA_MONDTCTLx, DMA_MONRPTx, DMA_MONSSEQCTLx, DMA_MONDNSEQCTLx) (x=0~3)	384
14.6	使用注意事项.....	385
15	电压比较器 (CMP)	386
15.1	简介.....	386
15.2	功能说明.....	388

15.2.1	电压比较	388
15.2.2	数字滤波	389
15.2.3	中断和事件	389
15.2.4	扫描比较模式	390
15.2.5	8bit-DAC 设定	391
15.3	注意事项	392
15.3.1	模块停止功能	392
15.3.2	模块停止时的动作	392
15.3.3	停止低功耗模式下的动作	392
15.3.4	掉电低功耗模式下的动作	392
15.4	寄存器说明	393
15.4.1	CMP 控制寄存器 (CMP_CTRL)	394
15.4.2	CMP 电压选择寄存器 (CMP_VLTSEL)	396
15.4.3	CMP 结果监视寄存器 (CMP_OUTMON)	398
15.4.4	CMP 稳定时间寄存器 (CMP_CVSSTB)	399
15.4.5	CMP 比较电压扫描周期寄存器 (CMP_CVSPRD)	399
15.4.6	CMP 用 8bit-DAC 控制寄存器(CMP_DACR)	400
15.4.7	CMP 用 8bit-DAC 数据寄存器(CMP_DADR1,CMP_DADR2)	400
15.4.8	CMP 内部基准电压 AD 转换寄存器(CMP_RVADC)	401
16	模数转换模块 (ADC)	402
16.1	简介	402
16.2	ADC 系统框图	404
16.3	功能说明	407
16.3.1	ADC 时钟	407
16.3.2	通道选择	407
16.3.3	触发源选择	409
16.3.4	序列 A 单次扫描模式	409
16.3.5	序列 A 连续扫描模式	410
16.3.6	双序列扫描模式	411
16.3.7	模拟看门狗功能	413
16.3.8	模拟输入的采样时间和转换时间	414
16.3.9	A/D 数据寄存器自动清除功能	416
16.3.10	转换数据平均计算功能	416
16.3.11	可编程增益放大器 PGA	417
16.3.12	多 ADC 协同工作模式	417
16.3.13	中断和事件信号输出	423
16.4	寄存器说明	424
16.4.1	A/D 启动寄存器 ADC_STR	426
16.4.2	A/D 控制寄存器 0 ADC_CR0	427
16.4.3	A/D 控制寄存器 1 ADC_CR1	428
16.4.4	A/D 转换开始触发寄存器 ADC_TRGSR	429
16.4.5	A/D 通道选择寄存器 A ADC_CHSELRA0	430

16.4.6	A/D 通道选择寄存器 A 1 ADC_CHSELRA1	430
16.4.7	A/D 通道选择寄存器 B ADC_CHSELRB0	431
16.4.8	A/D 通道选择寄存器 B1 ADC_CHSELRB1	431
16.4.9	A/D 平均通道选择寄存器 ADC_AVCHSEL0	432
16.4.10	A/D 平均通道选择寄存器 1 ADC_AVCHSEL1.....	432
16.4.11	A/D 采样状态寄存器 ADC_SSTR	433
16.4.12	A/D 通道映射控制寄存器 ADC_CHMUXR	434
16.4.13	A/D 中断状态寄存器 ADC_ISR.....	435
16.4.14	A/D 中断许可寄存器 ADC_ICR	435
16.4.15	A/D 协同模式控制寄存器 ADC_SYNCCR.....	436
16.4.16	A/D 数据寄存器 ADC_DR	438
16.4.17	模拟看门狗控制寄存器 ADC_AWDCR	439
16.4.18	模拟看门狗阀值寄存器 ADC_AWDDR0、ADC_AWDDR1.....	440
16.4.19	模拟看门狗比较通道选择寄存器 ADC_AWDCHSR0	441
16.4.20	模拟看门狗比较通道选择寄存器 1 ADC_AWDCHSR1	441
16.4.21	模拟看门狗状态寄存器 ADC_AWDSR0.....	442
16.4.22	模拟看门狗状态寄存器 1 ADC_AWDSR1	442
16.4.23	A/D 可编程增益放大器控制寄存器 ADC_PGACR.....	443
16.4.24	A/D 可编程增益倍数寄存器 ADC_PGAGSR	443
16.4.25	A/D 可编程增益放大器输入选择寄存器 ADC_PGAINSR0.....	444
16.4.26	A/D 可编程增益放大器输入选择寄存器 1 ADC_PGAINSR1	444
16.5	使用注意事项	445
16.5.1	数据寄存器读取时注意事项	445
16.5.2	扫描完成中断处理注意事项	445
16.5.3	模块停止与低功耗设定的注意事项	445
16.5.4	A/D 转换模拟通道输入的引脚设定	445
16.5.5	噪声控制	445
17	温度传感器 (OTS)	446
17.1	简介	446
17.2	使用说明	447
17.3	寄存器说明	449
17.3.1	OTS 控制寄存器 (OTS_CTL)	449
17.3.2	OTS 数据寄存器 1 (OTS_DR1)	450
17.3.3	OTS 数据寄存器 2 (OTS_DR2)	450
17.3.4	OTS 误差补偿寄存器 (OTS_ECR)	450
18	高级控制定时器 (Timer6)	451
18.1	简介	451
18.2	基本框图	451
18.3	功能说明	453
18.3.1	基本动作	453
18.3.1.1	波形模式	453
18.3.1.2	比较输出	453

18.3.1.3 捕获输入.....	454
18.3.2 时钟源选择.....	455
18.3.3 计数方向.....	455
18.3.3.1 锯齿波计数方向.....	456
18.3.3.2 三角波计数方向.....	456
18.3.4 数字滤波.....	456
18.3.5 软件同步.....	457
18.3.5.1 软件同步启动.....	457
18.3.5.2 软件同步停止.....	457
18.3.5.3 软件同步清零.....	457
18.3.6 硬件同步.....	458
18.3.6.1 硬件同步启动.....	458
18.3.6.2 硬件同步停止.....	458
18.3.6.3 硬件同步清零.....	458
18.3.6.4 硬件同步捕获输入.....	459
18.3.6.5 硬件同步计数.....	459
18.3.7 脉宽测量.....	460
18.3.8 周期测量.....	460
18.3.9 缓存功能.....	460
18.3.9.1 单缓存动作.....	460
18.3.9.2 双缓存动作.....	461
18.3.9.3 缓存传送时间点.....	462
通用周期基准值缓存传送时间点.....	462
通用比较基准值缓存传送时间点.....	462
捕获输入值缓存传送时间点.....	464
专用比较基准值缓存传送时间点.....	464
死区时间基准值缓存传送时间点.....	464
清零动作时缓存传送.....	465
18.3.10 通用 PWM 输出	465
18.3.10.1 独立 PWM 输出	465
18.3.10.2 互补 PWM 输出	465
软件设定 GCMR 互补 PWM 输出	465
硬件设定 GCMR 互补 PWM 输出	466
18.3.10.3 多相 PWM 输出	468
18.3.11 正交编码计数	470
18.3.11.1 位置计数模式.....	470
基本计数.....	470
相位差计数.....	470
方向计数.....	471
18.3.11.2 公转计数模式.....	472
Z 相计数	472
位置溢出计数	472

混合计数.....	473
18.3.11.3 Z 相动作屏蔽	473
18.3.12 周期间隔响应	475
18.3.13 EMB 控制	476
18.3.14 典型应用例.....	476
18.3.14.1 基本计数及中断动作.....	476
18.3.14.2 比较输出及中断动作.....	477
18.3.14.3 捕获输入及中断动作.....	477
18.3.14.4 缓存传送动作（周期基准值）	478
18.3.14.5 缓存传送动作（通用比较基准值）	478
18.3.14.6 缓存传送动作（专用比较基准值）	478
18.3.14.7 缓存传送动作（死区基准值）	479
18.3.14.8 同步启动动作（软件方式）	479
18.3.14.9 同步启动动作（硬件方式）	479
18.3.14.10 正交编码计数动作（2 相）	480
18.3.14.11 正交编码计数动作（3 相）	480
18.3.14.12 单路 PWM 输出	481
18.3.14.13 互补 PWM 输出（软件死区）	481
18.3.14.14 互补 PWM 输出（硬件死区）	482
18.3.14.15 EMB 监控及中断动作	482
18.3.15 功能汇总表.....	483
18.4 中断及事件说明.....	485
18.4.1 中断输出	485
18.4.1.1 计数比较匹配中断.....	485
18.4.1.2 计数周期匹配中断.....	485
18.4.1.3 死区时间错误中断.....	485
18.4.2 事件输出	486
18.5 寄存器说明.....	487
18.5.1 通用计数值寄存器（TMR6_CNTER）	489
18.5.2 通用周期基准值寄存器（TMR6_PERAR-PERCR）	489
18.5.3 通用比较基准值寄存器（TMR6_GCMAR-GCMFR）	489
18.5.4 专用比较基准值寄存器（TMR6_SCMAR-SCMFR）	490
18.5.5 死区时间基准值寄存器（TMR6_DTU<D>AR）	490
18.5.6 通用控制寄存器（TMR6_GCONR）	491
18.5.7 中断控制寄存器（TMR6_ICONR）	493
18.5.8 端口控制寄存器（TMR6_PCONR）	495
18.5.9 缓存控制寄存器（TMR6_BCONR）	497
18.5.10 死区控制寄存器（TMR6_DCONR）	499
18.5.11 滤波控制寄存器（TMR6_FCONR）	500
18.5.12 有效周期寄存器（TMR6_VPERR）	501
18.5.13 状态标志寄存器（TMR6_STFLR）	502
18.5.14 硬件启动事件选择寄存器（TMR6_HSTAR）	504

18.5.15	硬件停止事件选择寄存器 (TMR6_HSTPR)	506
18.5.16	硬件清零事件选择寄存器 (TMR6_HCLRR)	508
18.5.17	硬件捕获事件选择寄存器 (TMR6_HCPAR)	510
18.5.18	硬件捕获事件选择寄存器 (TMR6_HCPBR)	512
18.5.19	硬件递加事件选择寄存器 (TMR6_HCUPR)	514
18.5.20	硬件递减事件选择寄存器 (TMR6_HCDOR)	516
18.5.21	软件同步启动控制寄存器 (TMR6_SSTAR)	518
18.5.22	软件同步停止控制寄存器 (TMR6_SSTPR)	519
18.5.23	软件同步清零控制寄存器 (TMR6_SCLRR)	519
19	通用控制定时器 (Timer4)	520
19.1	简介	520
19.2	基本框图	520
19.3	功能说明	522
19.3.1	基本动作	522
19.3.1.1	波形模式	522
19.3.1.2	计数动作	522
19.3.1.3	比较输出	524
19.3.2	缓存功能	525
19.3.2.1	周期基准寄存器缓存功能	525
19.3.2.2	通用比较寄存器缓存功能	527
19.3.2.3	专用比较寄存器缓存功能	529
19.3.3	通用 PWM 输出	531
19.3.3.1	独立 PWM 输出	531
19.3.3.2	扩展 PWM 输出	532
19.3.3.3	互补 PWM 输出	533
19.3.4	周期间隔响应	536
19.3.5	EMB 控制	540
19.4	中断及事件说明	541
19.4.1	计数比较匹配中断	541
19.4.2	计数周期匹配中断	541
19.4.3	重载计数匹配中断	541
19.4.4	专用比较匹配事件	541
19.5	寄存器说明	543
19.5.1	计数值寄存器 (TMR4_CNTR)	546
19.5.2	周期基准寄存器 (TMR4_CPSR)	546
19.5.3	控制状态寄存器 (TMR4_CCSR)	547
19.5.4	有效周期寄存器 (TMR4_CVPR)	549
19.5.5	通用比较基准寄存器 (TMR4_OCCRm)	549
19.5.6	通用控制状态寄存器 (TMR4_OCSRn)	550
19.5.7	通用扩展控制寄存器 (TMR4_OCERn)	551
19.5.8	通用模式控制寄存器 (TMR4_OCMRm)	553
19.5.9	专用比较基准寄存器 (TMR4_SCCRm)	558

19.5.10	专用控制状态寄存器 (TMR4_SCSRm)	559
19.5.11	专用模式控制寄存器 (TMR4_SCMRm)	561
19.5.12	PWM 基本控制寄存器 (TMR4_POCRn)	562
19.5.13	PWM 滤波控制寄存器 (TMR4_PFSRn)	563
19.5.14	PWM 死区控制寄存器 (TMR4_PDARn)	563
19.5.15	重载控制状态寄存器 (TMR4_RCSR)	564
19.5.16	EMB 控制状态寄存器 (TMR4_ECSR)	566
19.5.17	EMB 扩展控制寄存器 (TMR4_ECER)	566
20	紧急刹车模块 (EMB)	567
20.1	简介.....	567
20.2	功能描述.....	568
20.2.1	概述.....	568
20.2.2	外部端口输入电平变化时停止 PWM 信号输出.....	568
20.2.3	PWM 输出端口电平发生同相 (同高或同低) 时停止 PWM 信号输出	568
20.2.4	根据电压比较器比较结果停止 PWM 信号输出.....	569
20.2.5	外部振荡器停止振荡时停止 PWM 信号输出.....	570
20.2.6	写寄存器软件控制 PWM 信号输出.....	570
20.3	寄存器说明.....	571
20.3.1	EMB 控制寄存器 0(EMB_CTL0).....	572
20.3.2	EMB 控制寄存器 1~3(EMB_CTL1~3)	573
20.3.3	EMB 反馈电平选择寄存器 0(EMB_PWMLV0).....	574
20.3.4	EMB 反馈电平选择寄存器 1~3(EMB_PWMLV1~3)	575
20.3.5	EMB 软件输出使能控制寄存器(EMB_SOEx)(x=0~3).....	575
20.3.6	EMB 状态寄存器(EMB_STATx) (x=0~3).....	576
20.3.7	EMB 状态复位寄存器(EMB_STATCLRx) (x=0~3)	577
20.3.8	EMB 中断许可寄存器(EMB_INTENx)(x=0~3)	578
21	通用定时器 (TimerA)	579
21.1	简介.....	579
21.2	基本框图.....	579
21.3	功能说明.....	581
21.3.1	基本动作	581
21.3.1.1	波形模式.....	581
21.3.1.2	比较输出.....	581
21.3.1.3	捕获输入.....	582
21.3.2	时钟源选择.....	583
21.3.3	同步启动.....	584
21.3.4	数字滤波.....	585
21.3.5	缓存功能.....	585
21.3.6	级联计数.....	586
21.3.7	PWM 输出	587
21.3.7.1	通用 PWM 输出	587
21.3.7.2	单脉冲 PWM 输出	588

21.3.8 正交编码计数	588
21.3.8.1 位置计数模式.....	589
21.3.8.2 公转计数模式.....	590
21.4 中断及事件说明.....	593
21.4.1 比较匹配中断及事件.....	593
21.4.2 周期匹配中断及事件.....	593
21.5 寄存器说明.....	594
21.5.1 通用计数值寄存器 (TMRA_CNTER)	596
21.5.2 周期基准值寄存器 (TMRA_PERAR)	596
21.5.3 比较基准值寄存器 (TMRA_CMPAR1~8)	596
21.5.4 控制状态寄存器 (TMRA_BCSTR)	597
21.5.5 中断控制寄存器 (TMRA_ICONR)	598
21.5.6 事件控制寄存器 (TMRA_ECONR)	599
21.5.7 滤波控制寄存器 (TMRA_FCONR)	600
21.5.8 状态标志寄存器 (TMRA_STFLR)	601
21.5.9 缓存控制寄存器 (TMRA_BCONR1~4)	603
21.5.10 捕获控制寄存器 (TMRA_CCONR1~8)	604
21.5.11 端口控制寄存器 (TMRA_PCONR1~8)	606
21.5.12 硬件触发事件选择寄存器 (TMRA_HCONR)	608
21.5.13 硬件递加事件选择寄存器 (TMRA_HCUPR)	610
21.5.14 硬件递减事件选择寄存器 (TMRA_HCDOR)	612
22 通用定时器 (Timer0)	614
22.1 简介.....	614
22.2 基本框图.....	614
22.3 功能说明.....	615
22.3.1 时钟源选择.....	615
22.3.1.1 同步计数时钟源.....	615
22.3.1.2 异步计数时钟源.....	615
22.3.2 基本计数动作	615
22.3.3 硬件触发动作	616
22.4 中断及事件说明.....	617
22.4.1 中断输出	617
22.4.2 事件输出	617
22.5 寄存器说明.....	618
22.5.1 计数值寄存器 (TMR0_CNTAR)	619
22.5.2 基准值寄存器 (TMR0_CMPPAR)	619
22.5.3 基本控制寄存器 (TMR0_BCONR)	620
22.5.4 状态标志寄存器 (TMR0_STFLR)	623
22.6 使用注意事项.....	624
23 实时时钟 (RTC)	625
23.1 简介.....	625
23.2 基本框图.....	626

23.3 功能说明.....	627
23.3.1 上电设定	627
23.3.2 RTC 计数开始设定	627
23.3.3 系统低功耗模式切换.....	627
23.3.4 读出计数寄存器.....	628
23.3.5 写入计数寄存器.....	628
23.3.6 闹钟设定.....	628
23.3.7 时钟误差补偿.....	628
23.3.8 1Hz 输出	629
23.4 中断说明.....	630
23.4.1 闹钟中断	630
23.4.2 定周期中断.....	630
23.5 寄存器说明	631
23.5.1 控制寄存器 0 (RTC_CR0)	632
23.5.2 控制寄存器 1 (RTC_CR1)	633
23.5.3 控制寄存器 2 (RTC_CR2)	634
23.5.4 控制寄存器 3 (RTC_CR3)	635
23.5.5 秒计数寄存器 (RTC_SEC)	635
23.5.6 分计数寄存器 (RTC_MIN)	636
23.5.7 时计数寄存器 (RTC_HOUR)	637
23.5.8 日计数寄存器 (RTC_DAY)	639
23.5.9 周计数寄存器 (RTC_WEEK)	640
23.5.10 月计数寄存器 (RTC_MON)	641
23.5.11 年计数寄存器 (RTC_YEAR)	641
23.5.12 分闹钟寄存器 (RTC_ALMMIN)	642
23.5.13 时闹钟寄存器 (RTC_ALMHOUR)	642
23.5.14 时闹钟寄存器 (RTC_ALMEEK)	643
23.5.15 时钟误差补偿寄存器 (RTC_ERRCRH、RTC_ERRCRL)	644
24 看门狗计数器 (WDT/SWDT)	646
24.1 简介.....	646
24.2 功能说明.....	647
24.2.1 启动看门狗.....	647
24.2.2 硬件启动方式.....	648
24.2.3 软件启动方式.....	649
24.2.4 刷新动作.....	650
24.2.5 标志位.....	651
24.2.6 中断复位.....	652
24.2.7 计数下溢.....	653
24.2.8 刷新错误.....	654
24.3 寄存器说明	655
24.3.1 控制寄存器 (WDT_CR)	656
24.3.2 状态寄存器 (SWDT_SR、WDT_SR)	658

24.3.3	刷新寄存器（SWDT_RR、WDT_RR）	659
24.4	使用注意事项	659
25	通用同步异步收发器（USART）	660
25.1	简介	660
25.2	USART 系统框图	662
25.3	管脚说明	662
25.4	功能说明	663
25.4.1	UART	663
25.4.1.1	时钟	663
25.4.1.2	数据格式	664
25.4.1.3	调制解调器操作	664
25.4.1.4	发送器	665
25.4.1.5	接收器	667
25.4.1.6	UART 接收 TIMEOUT 功能	671
25.4.1.7	RX 线唤醒停止模式功能	672
25.4.1.8	UART 中断和事件	673
25.4.2	多处理器通信	674
25.4.2.1	功能简介	674
25.4.2.2	数据格式	674
25.4.2.3	动作说明	675
25.4.2.4	中断和事件	678
25.4.3	智能卡	679
25.4.3.1	连接示意图	679
25.4.3.2	时钟	679
25.4.3.3	数据格式	680
25.4.3.4	智能卡的初始化设定步骤	681
25.4.3.5	智能卡模式动作说明	681
25.4.3.6	中断和事件	683
25.4.4	时钟同步模式	684
25.4.4.1	时钟	684
25.4.4.2	数据格式	684
25.4.4.3	调制解调器操作	685
25.4.4.4	发送器	685
25.4.4.5	接收器	687
25.4.4.6	同时发送接收数据	689
25.4.4.7	时钟同步模式中断和事件	690
25.4.5	数字滤波功能	691
25.5	寄存器说明	692
25.5.1	状态寄存器（USART_SR）	693
25.5.2	数据寄存器（USART_DR）	696
25.5.3	波特率寄存器（USART_BRR）	697
25.5.4	控制寄存器 1（USART_CR1）	699

25.5.5	控制寄存器 2 (USART_CR2)	702
25.5.6	控制寄存器 3 (USART_CR3)	703
25.5.7	预分频寄存器 (USART_PR)	704
25.6	使用注意事项.....	705
25.6.1	UART 注意事项	705
25.6.2	时钟同步模式注意事项	705
25.6.3	其他注意事项	705
26	集成电路总线 (I2C)	706
26.1	简介	706
26.2	I2C 系统框图.....	707
26.2.1	系统框图	707
26.2.2	结构图	708
26.3	动作说明.....	709
26.3.1	I2C 协议	709
26.3.1.1	开始条件	709
26.3.1.2	地址传送	710
26.3.1.3	数据传送	711
26.3.1.4	停止条件	715
26.3.1.5	重新开始条件	715
26.3.1.6	SCL 时钟同步	716
26.3.1.7	仲裁	716
26.3.1.8	握手	717
26.3.2	地址匹配	719
26.3.2.1	从机地址匹配	719
26.3.2.2	广播地址匹配	722
26.3.2.3	SMBus 主机地址匹配	723
26.3.2.4	SMBus 报警响应地址匹配	724
26.3.2.5	SMBus 默认地址匹配	725
26.3.3	SMBus 动作	725
26.3.3.1	SMBus 超时测量	725
26.3.3.2	数据包错误码 (PEC)	727
26.3.4	复位	727
26.3.5	中断和事件信号输出	728
26.3.6	可编程数字滤波	730
26.4	应用软件设定 I2C 初始化流程	730
26.5	寄存器说明	731
26.5.1	I2C 控制寄存器 1(I2C_CR1)	732
26.5.2	I2C 控制寄存器 1(I2C_CR2)	734
26.5.3	I2C 控制寄存器 1(I2C_CR3)	736
26.5.4	I2C 控制寄存器 1(I2C_CR4)	737
26.5.5	I2C 从机地址寄存器 0(I2C_SLR0)	738
26.5.6	I2C 从机地址寄存器 1(I2C_SLR1)	739

26.5.7	I2C SCL 电平超时控制寄存器(I2C_SLTR).....	740
26.5.8	I2C 状态寄存器(I2C_SR).....	741
26.5.9	I2C 状态清零寄存器(I2C_CLR).....	746
26.5.10	I2C 数据发送寄存器(I2C_DTR)	747
26.5.11	I2C 数据接收寄存器(I2C_DR)	747
26.5.12	I2C 数据移位寄存器(I2C_DSR).....	748
26.5.13	I2C 时钟控制寄存器(I2C_CCR)	749
26.5.14	I2C 滤波控制寄存器(I2C_FLTR)	751
27	串行外设接口 (SPI)	752
27.1	简介.....	752
27.2	SPI 系统框图.....	754
27.3	管脚说明.....	755
27.4	SPI 动作系统说明.....	756
27.4.1	主机模式的管脚状态.....	756
27.4.2	从机模式的管脚状态.....	757
27.4.3	SPI 系统连接实例	758
27.5	数据通信说明.....	760
27.5.1	波特率.....	760
27.5.2	数据格式.....	761
27.5.3	传送格式.....	763
27.5.4	通信方式.....	765
27.6	运行说明.....	767
27.6.1	运行模式概要	767
27.6.2	SPI 运行模式时的主机动作	769
27.6.3	SPI 运行模式时的从机动作	770
27.6.4	时钟同步运行模式中的主机动作	771
27.6.5	时钟同步运行模式中的从机动作	772
27.6.6	几种 SPI 动作的处理流程	774
27.7	奇偶校验位自诊断.....	775
27.8	错误检测.....	776
27.8.1	欠载错误.....	777
27.8.2	模式错误.....	777
27.8.3	过载错误.....	777
27.8.4	奇偶校验错误	779
27.9	SPI 的初始化.....	781
27.9.1	清除 SPE 位进行初始化	781
27.9.2	系统复位初始化	781
27.10	中断源.....	782
27.11	可供使用的事件触发源.....	783
27.12	寄存器说明.....	784
27.12.1	SPI 数据寄存器 (SPI_DR)	785
27.12.2	SPI 控制寄存器 (SPI_CR1)	786

27.12.3	SPI 通信配置寄存器 1 (SPI_CFG1)	788
27.12.4	SPI 状态寄存器 (SPI_SR)	790
27.12.5	SPI 通信配置寄存器 2 (SPI_CFG2)	791
28	四线式串行外设接口 (QSPI)	793
28.1	简介.....	793
28.2	内存映射.....	795
28.2.1	内部总线空间	795
28.2.2	ROM 空间及总线的地址宽度	796
28.3	QSPI 总线.....	797
28.3.1	SPI 协议	797
28.3.2	SPI 模式	799
28.4	QSPI 总线的时序调整.....	800
28.4.1	QSPI 总线基准时钟	800
28.4.2	SPI 总线基准时钟	802
28.4.3	QSSN 信号最小高电平宽度.....	802
28.4.4	QSSN 的建立时间.....	802
28.4.5	QSSN 的保持时间.....	803
28.4.6	串行数据接收延迟	803
28.5	用于 ROM 访问的 SPI 指令介绍.....	805
28.5.1	现有的 QSPI-ROM 指令参考	805
28.5.2	标准读指令	806
28.5.3	快速读指令	806
28.5.4	二线式输出快速读指令	808
28.5.5	二线式输入输出快速读指令	809
28.5.6	四线式输出快速读指令	811
28.5.7	四线式输入输出快速读指令	813
28.5.8	进入 4-Byte 模式指令	815
28.5.9	退出 4-Byte 模式指令	816
28.5.10	写许可指令	817
28.6	QSPI 总线周期的安排	818
28.6.1	独立转换的单一闪存读取	818
28.6.2	使用预读取功能进行闪存读取	818
28.6.3	预读取的终止	819
28.6.4	预读取状态监测	819
28.6.5	使用 QSPI 总线周期延长功能进行闪存读取	820
28.7	XIP 控制	821
28.7.1	XIP 模式的设置	822
28.7.2	XIP 模式的退出	822
28.8	QSIO2 和 QSIO3 的管脚状态	823
28.9	直接通信模式	824
28.9.1	关于直接通信模式	824
28.9.2	直接通信模式的设置	824

28.9.3	直接通信模式下 QSPI 总线周期的生成.....	824
28.10	中断.....	825
28.11	使用上的注意事项.....	825
28.11.1	QSPI 寄存器的设置顺序	825
28.11.2	模块停止信号的设置.....	825
28.12	寄存器说明.....	826
28.12.1	QSPI 控制寄存器 (QSCR)	827
28.12.2	QSPI 片选控制寄存器 (QSCSCR)	830
28.12.3	QSPI 格式控制寄存器 (QSFCR)	831
28.12.4	QSPI 状态寄存器 (QSSR)	833
28.12.5	QSPI 指令代码寄存器 (QSCCMD)	835
28.12.6	QSPI 直接通信指令寄存器 (QSDCOM)	835
28.12.7	QSPI XIP 模式代码寄存器 (QSXCMD)	836
28.12.8	QSPI 系统配置寄存器 (QSSR2)	836
28.12.9	QSPI 外部扩展地址寄存器 (QSEXAR)	837
29	集成电路内置音频总线模块 (I2S)	838
29.1	简介.....	838
29.2	I2S 系统框图.....	839
29.3	管脚说明.....	840
29.4	功能说明.....	841
29.4.1	I2S 一般说明	841
29.4.2	通信方式.....	841
29.4.3	支持的音频协议	842
29.4.3.1	I2S Philips 标准.....	842
29.4.3.2	MSB 对齐标准	843
29.4.3.3	LSB 对齐标准	844
29.4.3.4	PCM 标准	846
29.4.4	时钟发生器.....	848
29.4.5	I2S 主模式	850
29.4.6	I2S 从模式	851
29.4.7	I2S 中断	852
29.4.8	使用上的注意事项	854
29.4.8.1	作为主机使用时的注意事项.....	854
29.4.8.2	作为从机使用时的注意事项.....	856
29.5	寄存器说明.....	857
29.5.1	I2S 控制寄存器 (I2S_CTRL)	858
29.5.2	I2S 状态寄存器 (I2S_SR)	860
29.5.3	I2S 错误状态寄存器 (I2S_ER)	861
29.5.4	I2S 配置寄存器 (I2S_CFGR)	862
29.5.5	I2S 发送缓冲 FIFO 数据寄存器 (I2S_TXBUF)	863
29.5.6	I2S 接收缓冲 FIFO 数据寄存器 (I2S_RXBUF)	864
29.5.7	I2S 分频寄存器 (I2S_PR)	865

30 控制器局域网 (CAN)	866
30.1 简介.....	866
30.2 CAN 系统框图.....	867
30.3 管脚说明.....	867
30.4 功能说明.....	868
30.4.1 波特率设定.....	868
30.4.2 发送缓冲器.....	869
30.4.3 接收缓冲器.....	870
30.4.4 接收筛选寄存器组.....	870
30.4.5 数据发送.....	871
30.4.6 单次数据发送.....	872
30.4.7 取消数据发送.....	872
30.4.8 数据接收.....	873
30.4.9 错误处理.....	873
30.4.10 节点关闭.....	874
30.4.11 仲裁失败位置捕捉.....	874
30.4.12 回环模式.....	874
30.4.13 静默模式.....	875
30.4.14 软件复位功能.....	876
30.4.15 向上兼容 CAN-FD 功能.....	878
30.4.16 时间触发 TTCAN.....	878
30.4.16.1 TTCAN 模式下的 TBUF 行为.....	878
30.4.16.2 TTCAN 功能描述.....	878
30.4.16.3 TTCAN 时序	879
30.4.16.4 TTCAN 触发方式	879
30.4.16.5 TTCAN 触发看门时间	880
30.4.17 中断.....	881
30.5 寄存器说明.....	882
30.5.1 CAN 接收 BUF 寄存器 (CAN_RBUF)	884
30.5.2 CAN 发送 BUF 寄存器 (CAN_TBUF)	886
30.5.3 CAN 配置和状态寄存器 (CAN_CFG_STAT)	888
30.5.4 CAN 命令寄存器 (CAN_TCMD)	889
30.5.5 CAN 发送控制寄存器 (CAN_TCTRL)	891
30.5.6 CAN 接收控制寄存器 (CAN_RCTRL)	893
30.5.7 CAN 接收和发送中断使能寄存器 (CAN_RTIE)	894
30.5.8 CAN 接收和发送中断状态寄存器 (CAN_RTIF)	895
30.5.9 CAN 错误中断使能和标志寄存器 (CAN_ERRINT)	897
30.5.10 CAN 位时序寄存器 (CAN_BT)	898
30.5.11 CAN 错误和仲裁失败捕捉寄存器 (CAN_EALCAP)	899
30.5.12 CAN 警告限定寄存器 (CAN_LIMIT)	899
30.5.13 CAN 接收错误计数器寄存器 (CAN_RECNT)	900
30.5.14 CAN 发送错误计数器寄存器 (CAN_TECNT)	900

30.5.15	CAN 筛选器组控制寄存器 (CAN_ACFCTRL)	901
30.5.16	CAN 筛选器组使能寄存器 (CAN_ACFEN)	902
30.5.17	CAN 筛选器组 code 和 mask 寄存器 (CAN_ACF)	903
30.5.18	TTCAN TB slot 指针寄存器 (CAN_TBSLOT)	904
30.5.19	TTCAN 时间触发配置寄存器 (CAN_TTCFG)	905
30.5.20	TTCAN 参考消息寄存器 (CAN_REF_MSG)	906
30.5.21	TTCAN 触发配置寄存器 (CAN_TRG_CFG)	907
30.5.22	TTCAN 触发时间寄存器 (CAN_TT_TRIG)	908
30.5.23	TTCAN 触发看门时间寄存器 (CAN_TT_WTRIG)	908
30.6	使用注意事项	909
30.6.1	CAN 总线抗干扰措施	909
30.6.2	CAN 控制器噪声制约	909
31	USB2.0 全速模块 (USBFS)	910
31.1	USBFS 简介	910
31.2	USBFS 主要特性	910
31.2.1	通用特性	910
31.2.2	主机模式特性	911
31.2.3	设备模式特性	911
31.3	USBFS 系统框图	912
31.4	USBFS 管脚说明	913
31.5	USBFS 功能说明	914
31.5.1	USBFS 时钟以及工作模式	914
31.5.2	USBFS 模式决定	914
31.5.3	USBFS 主机功能	915
31.5.3.1	主机功能简介	915
31.5.3.2	主机端口供电	915
31.5.3.3	主机检测设备连接和断开	916
31.5.3.4	主机枚举	916
31.5.3.5	主机挂起	916
31.5.3.6	主机通道	916
31.5.3.7	主机调度器	918
31.5.4	USBFS 设备功能	920
31.5.4.1	设备功能简介	920
31.5.4.2	设备供电状态	920
31.5.4.3	设备默认状态	921
31.5.4.4	设备挂起状态	921
31.5.4.5	设备软断开	921
31.5.4.6	设备端点	921
31.5.5	USBFS SOF 脉冲管脚输出功能	925
31.5.5.1	主机 SOF	925
31.5.5.2	设备 SOF	925
31.5.6	USBFS 功耗控制	926

31.5.7	USBFS 动态更新 USBFS_HFIR 寄存器.....	927
31.5.8	USBFS 数据 FIFO	927
31.5.9	USBFS 主机 FIFO 架构	928
31.5.9.1	主机 RxFIFO	928
31.5.9.2	主机 TxFIFO	929
31.5.10	USBFS 设备 FIFO 架构	930
31.5.10.1	设备 RxFIFO	930
31.5.10.2	设备 TxFIFO	931
31.5.11	USBFS FIFO RAM 分配	932
31.5.11.1	主机模式.....	932
31.5.11.2	设备模式.....	932
31.5.12	USBFS 系统性能.....	934
31.5.13	USBFS 中断和事件.....	935
31.6	USBFS 编程模型	937
31.6.1	USBFS 模块初始化.....	937
31.6.2	USBFS 主机初始化.....	937
31.6.3	USBFS 设备初始化.....	938
31.6.4	USBFS DMA 模式	939
31.6.5	USBFS 主机编程模型	939
31.6.5.1	通道初始化.....	939
31.6.5.2	通道停止.....	940
31.6.6	USBFS 设备编程模型.....	940
31.6.6.1	USB 复位时的端点端点初始化.....	940
31.6.6.2	USB 复位时的端点初始化.....	941
31.6.6.3	收到 SetAddress 命令时的端点初始化.....	941
31.6.6.4	收到 SetConfiguration/SetInterface 命令时的端点初始化	941
31.6.6.5	端点激活.....	942
31.6.6.6	端点停用.....	942
31.6.7	USBFS 操作模型.....	943
31.6.7.1	SETUP 和 OUT 数据传输	943
31.6.7.2	IN 数据传输	953
31.7	寄存器说明.....	963
31.7.1	USBFS 系统控制寄存器.....	967
31.7.1.1	USBFS 系统控制寄存器(USBFS_SYCTLREG).....	967
31.7.2	USBFS 全局寄存器.....	968
31.7.2.1	USBFS_VBUS 控制寄存器(USBFS_GVBUSCFG).....	968
31.7.2.2	USBFS_AHB 控制寄存器(USBFS_GAHBCFG)	969
31.7.2.3	USBFS_USB 配置寄存器 (USBFS_GUSBCFG)	971
31.7.2.4	USBFS 复位寄存器 (USBFS_GRSTCTL).....	973
31.7.2.5	USBFS 全局中断状态寄存器 (USBFS_GINTSTS)	976
31.7.2.6	USBFS 全局中断屏蔽寄存器 (USBFS_GINTMSK).....	981
31.7.2.7	USBFS 接收状态调试读取/USBFS 状态读取和出栈寄存器	

(USBFS_GRXSTSR/USBFS_GRXSTSP).....	985
31.7.2.8 USBFS 接收 FIFO 大小寄存器 (USBFS_GRXFISIZ)	987
31.7.2.9 USBFS 主机非周期性发送 FIFO 大小寄存器(USBFS_HNPTXFSIZ)/端点 0 发送 FIFO 大小(USBFS_DIEPTXF0)	988
31.7.2.10 USBFS 非周期性发送 FIFO/队列状态寄存器 (USBFS_HNPTXSTS)	989
31.7.2.11 USBFS 模块 ID 寄存器(USBFS_CID).....	991
31.7.2.12 USBFS 主机周期性发送 FIFO 大小寄存器(USBFS_HPTXFSIZ).....	992
31.7.2.13 USBFS 设备 IN 端点发送 FIFO 大小寄存器 (USBFS_DIEPTXF _x) (x = 1..5) ..	993
31.7.3 USBFS 主机模式寄存器.....	994
31.7.3.1 USBFS 主机配置寄存器 (USBFS_HCFG).....	994
31.7.3.2 USBFS 主机帧时间间隔寄存器 (USBFS_HFIR).....	995
31.7.3.3 USBFS 主机帧编号/帧剩余时间寄存器 (USBFS_HFNUM)	996
31.7.3.4 USBFS 主机周期性发送 FIFO/队列状态寄存器(USBFS_HPTXSTS).....	997
31.7.3.5 USBFS 主机全体通道中断寄存器 (USBFS_HAINT)	999
31.7.3.6 USBFS 主机全体通道中断屏蔽寄存器(USBFS_HAINTMSK).....	1000
31.7.3.7 USBFS 主机端口控制和状态寄存器 (USBFS_HPRT).....	1001
31.7.3.8 USBFS 主机通道 x 特性寄存器 (USBFS_HCCHAR _x) (x = 0..11)	1004
31.7.3.9 USBFS 主机通道 x 中断寄存器 (USBFS_HCINT _x) (x = 0..11)	1006
31.7.3.10 USBFS 主机通道 x 中断屏蔽寄存器 (USBFS_HCINTMSK _x) (x = 0..11)	1008
31.7.3.11 USBFS 主机通道 x 传输大小寄存器 (USBFS_HCTSIZ _x) (x = 0..11)	1010
31.7.3.12 USBFS 主机通道 xDMA 地址寄存器 (USBFS_HCDMAX) (x = 0..11)	1011
31.7.4 USBFS 设备模式寄存器.....	1012
31.7.4.1 USBFS 设备配置寄存器(USBFS_DCFG).....	1012
31.7.4.2 USBFS 设备控制寄存器(USBFS_DCTL)	1014
31.7.4.3 USBFS 设备状态寄存器(USBFS_DSTS)	1016
31.7.4.4 USBFS 设备 IN 端点通用中断屏蔽寄存器(USBFS_DIEPMSK)	1017
31.7.4.5 USBFS 设备 OUT 端点通用中断屏蔽寄存器(USBFS_DOEPMSK).....	1019
31.7.4.6 USBFS 设备全体端点中断寄存器 (USBFS_DAINT)	1020
31.7.4.7 USBFS 设备全体端点中断屏蔽寄存器 (USBFS_DAINTMSK).....	1021
31.7.4.8 USBFS 设备 IN 端点 FIFO 空中断屏蔽寄存器(USBFS_DIEPEMPMSK)	1022
31.7.4.9 USBFS 设备控制 IN 端点 0 控制寄存器(USBFS_DIEPCTL0)	1023
31.7.4.10 USBFS 设备 IN 端点 x 控制寄存器(USBFS_DIEPCTL _x)(x=1..5)	1025
31.7.4.11 USBFS 设备 IN 端点 x 中断寄存器 (USBFS_DIEPINT _x) (x=0..5)	1028
31.7.4.12 USBFS 设备 IN 端点 0 传输大小寄存器(USBFS_DIEPTSIZ0)	1030
31.7.4.13 USBFS 设备 IN 端点 x 传输大小寄存器(USBFS_DIEPTSIZ _x)(x=1..5).....	1031
31.7.4.14 USBFS 设备 IN 端点 x DMA 地址寄存器 (USBFS_DIEPDMAx)(x=0..5).....	1032
31.7.4.15 USBFS 设备 IN 端点发送 FIFO 状态寄存器 (USBFS_DTXFSTSx)(x=0..5)	1033
31.7.4.16 USBFS 设备控制 OUT 端点 0 控制寄存器(USBFS_DOEPCTL0).....	1034
31.7.4.17 USBFS 设备 OUT 端点 x 控制寄存器(USBFS_DOEPCTL _x)(x=1..5).....	1036
31.7.4.18 USBFS 设备 OUT 端点 x 中断寄存器 (USBFS_DOEPINT _x) (x=0..5).....	1039
31.7.4.19 USBFS 设备 OUT 端点 0 传输大小寄存器(USBFS_DOEPTSIZ0).....	1041
31.7.4.20 USBFS 设备 OUT 端点 x 传输大小寄存器(USBFS_DOEPTSIZ _x)(x=1..5).....	1042

31.7.4.21 USBFS 设备 OUT 端点 x DMA 地址寄存器 (USBFS_DOEPDMAx)(x=0..5)	1043
31.7.5 USBFS 时钟门控控制寄存器.....	1044
31.7.5.1 USBFS 时钟门控控制寄存器(USBFS_GCCTL).....	1044
32 加密协处理模块 (CPM)	1045
32.1 简介.....	1045
32.2 加解密算法处理器 (AES)	1046
32.2.1 算法简介	1046
32.2.2 加密操作流程	1047
32.2.3 解密操作流程	1047
32.2.4 加密解密时间	1048
32.2.5 操作注意事项	1048
32.2.6 寄存器说明	1049
32.2.6.1 AES 控制寄存器 (AES_CR)	1050
32.2.6.2 AES 数据寄存器 (AES_DR)	1051
32.2.6.3 AES 密钥寄存器 (AES_KR)	1052
32.3 安全散列算法 (HASH)	1053
32.3.1 算法简介	1053
32.3.2 操作流程	1054
32.3.3 消息填充	1055
32.3.4 寄存器说明	1057
32.3.4.1 HASH 控制寄存器 (HASH_CR)	1057
32.3.4.2 HASH 摘要寄存器 (HASH_HR)	1058
32.3.4.3 HASH 数据寄存器 (HASH_DR)	1059
32.4 真随机数发生器 (TRNG)	1060
32.4.1 模块框图	1060
32.4.2 操作流程	1060
32.4.3 中断及事件输出	1060
32.4.4 操作注意事项	1061
32.4.5 寄存器说明	1061
32.4.5.1 TRNG 控制寄存器 (TRNG_CR)	1061
32.4.5.2 TRNG 模式寄存器 (TRNG_MR)	1062
32.4.5.3 TRNG 数据寄存器 (TRNG_DR)	1062
33 数据计算单元 (DCU)	1063
33.1 简介	1063
33.2 功能描述	1064
33.2.1 加法模式	1064
33.2.2 减法模式	1064
33.2.3 硬件触发启动模式	1064
33.2.4 比较模式	1065
33.2.5 中断和事件信号输出	1065
33.3 应用举例	1066
33.3.1 加法模式	1066

33.3.2	触发加模式.....	1066
33.3.3	比较模式.....	1066
33.4	寄存器说明.....	1067
33.4.1	DCU 控制寄存器(DCU_CTL)	1069
33.4.2	DCU 标志寄存器(DCU_FLAG)	1070
33.4.3	DCU 数据寄存器(DCU_DATAx) (x=0,1,2)	1071
33.4.4	DCU 标志复位寄存器(DCU_FLAGCLR)	1072
33.4.5	DCU 中断条件选择寄存器(DCU_INTEVTSEL)	1073
34	CRC 运算 (CRC)	1074
34.1	简介.....	1074
34.2	功能框图.....	1074
34.3	功能说明.....	1075
34.3.1	CRC 编码产生	1075
34.3.2	CRC 校验.....	1075
34.3.3	对 XOROUT,REFOUT,REFIN 不全为 1 的 CRC 校验	1076
34.4	寄存器说明.....	1077
34.4.1	控制寄存器 (CRC_CR)	1078
34.4.2	结果寄存器 (CRC_RESLT)	1079
34.4.3	标志寄存器 (CRC_FLG)	1080
34.4.4	数据寄存器 (CRC_DAT)	1080
35	SDIO 控制器 (SDIOC)	1081
35.1	简介.....	1081
35.2	功能描述.....	1082
35.2.1	端口分配.....	1082
35.2.2	基本访问方式	1083
35.2.3	数据传输.....	1083
35.2.4	SD 时钟	1083
35.2.5	中断和 DMA 启动请求.....	1084
35.2.5.1	SD 中断	1084
35.2.5.2	SDIO 中断	1084
35.2.5.3	DMA 请求	1084
35.2.5.4	卡的插入(insert)和移除(removal)	1085
35.2.6	主机和设备初始化	1085
35.2.6.1	主机初始化.....	1085
35.2.6.2	SD 卡初始化	1086
35.2.6.3	MMC 卡初始化.....	1086
35.2.6.4	SDIO 初始化	1086
35.2.7	SD/MMC 单数据块(single block)读写	1087
35.2.8	SD/MMC 多数据块(multi block)读写	1088
35.2.9	传输终止(abort), 挂起(suspend)和恢复(resume).....	1089
35.2.10	读等待(read wait).....	1090
35.2.11	唤醒(wakeup).....	1091

35.3	寄存器说明.....	1093
35.3.1	数据块长度寄存器(BLKSIZE)	1095
35.3.2	数据块计数器(BLKCNT)	1095
35.3.3	参数寄存器 0(ARG0).....	1096
35.3.4	参数寄存器 1(ARG1).....	1096
35.3.5	传输模式寄存器(TRANSMODE).....	1097
35.3.6	命令寄存器(CMD)	1098
35.3.7	应答寄存器 0(RESP0).....	1099
35.3.8	应答寄存器 1(RESP1).....	1099
35.3.9	应答寄存器 2(RESP2).....	1100
35.3.10	应答寄存器 3(RESP3).....	1100
35.3.11	应答寄存器 4(RESP4).....	1101
35.3.12	应答寄存器 5(RESP5).....	1101
35.3.13	应答寄存器 6(RESP6).....	1102
35.3.14	应答寄存器 7(RESP7).....	1102
35.3.15	数据缓冲寄存器 0(BUF0).....	1103
35.3.16	数据缓冲寄存器 1(BUF1).....	1103
35.3.17	主机状态寄存器(PSTAT)	1104
35.3.18	主机控制寄存器(HOSTCON).....	1105
35.3.19	电源控制寄存器(PWRCON)	1105
35.3.20	数据块间隙控制寄存器(BLKGPCON).....	1106
35.3.21	时钟控制寄存器(CLKCON).....	1107
35.3.22	超时控制寄存器(TOUTCON)	1108
35.3.23	软件复位寄存器(SFTRST)	1109
35.3.24	普通中断状态寄存器(NORINTST).....	1110
35.3.25	错误中断状态寄存器(ERRINTST)	1111
35.3.26	普通中断状态使能寄存器(NORINTSTEN)	1112
35.3.27	错误中断状态使能寄存器(ERRINTSTEN)	1113
35.3.28	普通中断信号使能寄存器(NORINTSGEN).....	1114
35.3.29	错误中断信号使能寄存器(ERRINTSGEN).....	1115
35.3.30	自动命令错误状态寄存器(ATCERRST).....	1116
35.3.31	强制自动命令错误状态控制寄存器(FEA).....	1117
35.3.32	强制错误状态控制寄存器(FEE)	1118
35.3.33	MMC 模式使能寄存器(MMCER).....	1119
36	调试控制器 (DBG)	1120
36.1	简介.....	1120
36.2	DBG 系统框图.....	1121
36.3	SWJ-DP 调试端口 (SWD 和 JTAG)	1122
36.3.1	JTAG-DP 或 SW-DP 的切换机制.....	1123
36.4	引脚排列和调试端口引脚.....	1124
36.4.1	SWJ 调试端口引脚	1124
36.4.2	灵活的 SWJ-DP 引脚分配	1124

36.4.3	JTAG 引脚上的内部上拉.....	1125
36.4.4	使用串行接口以及释放未使用的调试引脚以作 GPIO	1125
36.5	寄存器.....	1126
36.5.1	DBG 状态寄存器 (MCUDBGSTAT)	1127
36.5.2	外设调试暂停寄存器 (MCUSTPCTL)	1128
36.5.3	调试组件配置寄存器 (MCUTRACECTL)	1130
36.6	SW 调试端口.....	1131
36.6.1	SW 协议简介.....	1131
36.7	TPIU (跟踪端口接口单元)	1132
36.7.1	简介	1132
36.7.2	TRACE 引脚分配	1132
36.7.3	MCU 内部 TRACECLKIN 连接.....	1134
36.7.4	TPIU 寄存器	1135
36.7.5	TPIU 配置示例	1135
	版本信息 & 联系方式	1136

表目录

表 1-1 存储器映射	49
表 1-2 QSPI 地址空间分配	50
表 1-3 目标地址配置示例	51
表 1-4 寄存器列表	52
表 3-1 复位方式及产生条件	58
表 3-2 复位方式及复位标志	59
表 3-3 RMU 寄存器一览	69
表 4-1 各个内部时钟的规格	77
表 5-1 BOR 配置	132
表 5-2 PVD1/PVD2 特性	133
表 5-3 运行模式	139
表 5-4 低功耗模式	140
表 5-5 低功耗模式的运行条件及各模块在低功耗模式下的状态	142
表 5-6 运行模式说明	143
表 5-7 掉电模式子模式	150
表 5-8 RAM 模块与 RAM 掉电控制位	155
表 5-9 寄存器保护列表	156
表 5-10 寄存器一览表	158
表 6-1 寄存器一览表	189
表 7-1 CPU 时钟频率和 FLASH 读等待周期对照表	200
表 7-2 FLASH 实际读周期数	201
表 7-3 OTP 地址构成	206
表 7-4 寄存器一览表	208
表 8-1 SRAM 读写访问的等待周期设定和 CPU 时钟频率的关系	219
表 8-2 SRAM 空间分配	220
表 9-1 PORT 寄存器一览 1	234
表 9-2 PORT 寄存器一览 2	235
表 9-3 32bit 访问时 PORT 寄存器一览	251
表 10-1 中断向量表	263
表 10-2 中断事件请求序号与选择	282
表 12-1 KEYSCAN 管脚说明	329
表 12-2 KEYSCAN 寄存器一览表	331
表 14-1 通道重置说明	356
表 15-1 CMP 详细规格	386
表 15-2 CMP 寄存器一览表	393
表 16-1 各 ADC 单元规格	405
表 16-2 转换内部通道时的寄存器设定方法	408
表 16-3 序列 A 和 B 的各种竞争	412
表 16-4 AD 转换时间	415
表 16-5 ADC 寄存器一览 1/2	425

表 17-1 OTS 寄存器一览表.....	449
表 18-1 Timer6 的基本功能及特性.....	451
表 18-2 Timer6 端口列表	452
表 18-3 不同模式下的功能对比表	484
表 18-4 寄存器列表	488
表 19-1 Timer4 的基本功能及特性.....	520
表 19-2 Timer4 端口列表	521
表 19-3 寄存器列表	545
表 21-1 TimerA 的基本功能及特性.....	579
表 21-2 TimerA 端口列表.....	580
表 21-3 寄存器列表	595
表 22-1 寄存器列表	618
表 23-1 RTC 的基本规格.....	625
表 23-2 寄存器列表	631
表 24-1 看门狗计数器的基本特性	646
表 24-2 寄存器列表	655
表 25-1 USART 管脚说明	662
表 11-2 DIV_Fraction 为 0 时 UART 接收器的容差	668
表 11-3 DIV_Fraction 不为 0 时 UART 接收器的容差	668
表 25-2 UART 中断/事件表	673
表 25-3 多处理器模式中断/事件表	678
表 25-4 智能卡模式中断/事件表	683
表 25-5 时钟同步模式中断/事件表	690
表 25-6 USART 寄存器一览表	692
表 25-7 波特率计算公式（小数波特率无效 FBME=0）	697
表 25-8 波特率计算公式（小数波特率有效 FBME=1）	698
表 26-1 输入 / 输出引脚.....	708
表 26-2 寄存器一览表	731
表 27-1 SPI 的特性要点	753
表 27-2 管脚说明	755
表 27-3 主机模式时 SPI 管脚状态说明	756
表 27-4 从机模式时 SPI 管脚状态说明	757
表 27-5 部分设定值下位速	760
表 27-6 SPI 模式和寄存器设定关系	768
表 27-7 错误检测对应表	776
表 27-8 SPI 中断源说明	782
表 28-1 QSPI 主要规格	793
表 28-2 QSPI 管脚	794
表 28-3 QSPI 总线基准时钟选择一览表.....	801
表 28-4 参考指令一览表	805
表 28-5 QIO2 和 QIO3 的管脚状态	823
表 28-6 QSPI 寄存器一览表	826

表 29-1 I2S 主要特性.....	838
表 29-2 I2S 管脚说明.....	840
表 29-3 音频频率精度(针对 VCO 输入频率=1MHz).....	849
表 29-4 I2S 中断请求.....	853
表 29-5 I2S 寄存器一览表.....	857
表 30-1 CAN 管脚说明.....	867
表 30-2 CAN 位时间设定规则.....	869
表 30-3 软件复位范围表.....	877
表 30-4 CAN 中断表.....	881
表 30-5 CAN 寄存器一览表.....	882
表 30-6 CAN 寄存器 BYTE/HALFWORD/WORD 访问排布表	883
表 30-7 标准格式 CAN 接收邮箱格式.....	884
表 30-8 扩展格式 CAN 接收邮箱格式.....	885
表 30-9 标准格式 CAN 发送邮箱格式.....	886
表 30-10 扩展格式 CAN 发送邮箱格式.....	887
表 31-1 USBFS 管脚说明	913
表 31-2 USBFS 中断事件表.....	936
表 11-3 USBFS 系统控制寄存器一览表	963
表 11-4 USBFS 系统控制寄存器一览表	966
表 34-1 CRC 寄存器列表	1077
表 36-1 SWJ 调试端口引脚.....	1124
表 36-2 灵活的 SWJ-DP 引脚分配	1124

图目录

图 2-1 总线架构	55
图 3-1 上电复位	60
图 3-2 NRST 复位时序	61
图 3-3 欠压复位	61
图 3-4 可编程电压检测 1 复位	62
图 3-5 可编程电压检测 2 复位	63
图 3-6 看门狗和专用看门狗复位	64
图 3-7 掉电唤醒复位	64
图 3-8 软件复位	65
图 3-9 MPU 错误复位	65
图 3-10 RAM 奇偶校验复位	65
图 3-11 RAMECC 复位	66
图 3-12 时钟频率异常复位	66
图 3-13 外部高速振荡异常复位	67
图 4-1 时钟系统框图	72
图 4-2 时钟频率测量框图	73
图 4-3 外部高速振荡器连接事例	78
图 4-4 外部时钟输入的连接事例图	79
图 4-5 外部高速振荡器故障检测例	80
图 4-6 系统时钟选择 XTAL，检测到 XTAL 振荡故障例	81
图 4-7 外部低速振荡器连接事例	83
图 4-8 时钟频率测量时序图	90
图 5-1 电源构成图	130
图 5-2 上电复位、掉电复位波形	131
图 5-3 欠压复位波形	132
图 5-4 PVD1 中断/复位框图	134
图 5-5 中断/复位框图	134
图 5-6 电源监视 1 中断时序图	135
图 5-7 电源监视 1 复位时序图	136
图 5-8 电源监视 2 中断运行时序图	137
图 5-9 电源监视 2 复位运行时序图	138
图 5-11 内部电压采样示意图	138
图 5-10 PTWK _n 结构框图	152
图 7-1 FLASH 地址结构（512KB 产品）	196
图 7-2 FLASH 地址结构（256KB 产品）	197
图 7-3 启动扇区交换功能 1	207
图 7-4 启动交换功能 2	207
图 9-1 端口基本结构示意图	229
图 10-1 中断系统框图	255
图 10-2 中断事件选择	285

图 10-3 数字滤波器工作示意图	287
图 12-1 KEYS CAN 系统框图	329
图 12-2 键盘扫描功能示意图	330
图 14-1 DMA 结构图	350
图 14-2 连锁传输示意图	354
图 14-3 不连续地址传输示意图	355
图 14-4 不连续式重置示意图	357
图 14-5 应用例 1：存储器到存储器传输	360
图 14-6 应用例 2：存储器到外围电路的传输	362
图 15-1 CMP, 8bitDAC 功能连接图	387
图 15-2 CMP 工作示意图	388
图 15-3 扫描模式动作示意图	390
图 16-1 ADC 框图	404
图 16-2 内部模拟通道选择	407
图 16-3 序列 A 单次扫描模式	409
图 16-4 连续扫描	410
图 16-5 双序列扫描模式（序列 A 从被中断通道重新启动）	412
图 16-6 双序列扫描模式（序列 A 从第一个通道重新启动）	413
图 16-7 模拟看门狗保护区域（比较条件）	414
图 16-8 A/D 转换时间	415
图 16-9 平均功能有效时的转换动作	416
图 16-10 PGA 工作示意图	417
图 16-11 单次并行触发模式（三 ADC）	418
图 16-12 单次延迟触发模式（三 ADC）	420
图 16-13 循环并行触发模式（三 ADC）	421
图 16-14 循环延迟触发模式（两 ADC）	422
图 16-15 循环延迟触发模式（三 ADC）	422
图 16-16 ADC 中断和事件输出时序	423
图 17-1 OTS 功能框图	446
图 18-1 Timer6 基本框图	452
图 18-2 锯齿波波形（递加计数）	453
图 18-3 三角波波形	453
图 18-4 比较输出动作	454
图 18-5 捕获输入动作	455
图 18-6 捕获输入端口的滤波功能	456
图 18-7 软件同步动作	457
图 18-8 硬件同步动作	459
图 18-9 单缓存方式比较输出时序	461
图 18-10 双缓存方式捕获输入时序	462
图 18-11 锯齿波模式时计数缓存动作	463
图 18-12 三角波 A 模式时计数缓存动作	463
图 18-13 三角波 B 模式时计数缓存动作	464

图 18-14 TIM6_<> PWMA 输出 PWM 波	465
图 18-15 三角波 A 模式时软件设定 GCMBR 互补 PWM 波输出.....	466
图 18-16 三角波 B 模式时硬件设定 GCMBR 互补 PWM 波输出（对称死区）	467
图 18-17 6 相 PWM 波.....	468
图 18-18 三角波 A 模式时带死区时间三相互补 PWM 波输出.....	469
图 18-19 位置模式-基本计数.....	470
图 18-20 位置计数模式-相位差计数（1 倍计数）	471
图 18-21 位置计数模式-相位差计数（2 倍计数）	471
图 18-22 位置计数模式-相位差计数（4 倍计数）	471
图 18-23 位置计数模式-方向计数.....	472
图 18-24 公转计数模式-Z 相计数	472
图 18-25 公转计数模式-位置溢出计数.....	473
图 18-26 公转计数模式-混合计数	473
图 18-27 公转计数模式-混合计数 Z 相屏蔽动作例 1.....	474
图 18-28 公转计数模式-混合计数 Z 相屏蔽动作例 2.....	474
图 18-29 周期间隔有效请求信号动作	475
图 18-30 锯齿波模式时中断&事件输出例	486
图 19-1 Timer4 基本框图	521
图 19-2 Timer4 锯齿波波形.....	522
图 19-3 Timer4 三角波波形.....	522
图 19-4 Timer4 锯齿波模式计数动作.....	522
图 19-5 Timer4 三角波模式计数动作.....	523
图 19-6 锯齿波模式波形输出例	524
图 19-7 三角波模式波形输出例	524
图 19-8 缓存无效时修改锯齿波计数周期	525
图 19-9 缓存使能时修改锯齿波计数周期	526
图 19-10 缓存使能时修改三角波计数周期	526
图 19-11 OCCR 缓冲数据传输（周期间隔响应链接禁止时）	527
图 19-12 OCCR 缓冲数据传输（周期间隔响应链接使能）	528
图 19-13 输出比较缓冲数据传输（OCMR 缓冲使能）	529
图 19-14 SCCR 缓冲传输操作（周期间隔响应链接传输禁止时）	530
图 19-15 SCCR 缓冲传输操作（周期间隔响应链接传输使能时）	531
图 19-16 锯齿波独立 PWM 输出例.....	532
图 19-17 三角波独立 PWM 输出例.....	532
图 19-18 锯齿波扩展 PWM 输出.....	533
图 19-19 软件实现互补 PWM 输出.....	533
图 19-20 死区定时器模式下的互补 PWM 输出	534
图 19-21 脉宽异常时的死区定时器模式下的波形输出	535
图 19-22 死区定时器滤波模式下的互补 PWM 输出	536
图 19-23 周期间隔响应时序图	537
图 19-24 专用事件输出信号周期间隔响应输出	538
图 19-25 延时启动模式下专用事件输出信号的输出时序	542

图 21-1 TimerA 基本框图.....	580
图 21-2 锯齿波波形（递加计数）.....	581
图 21-3 三角波波形.....	581
图 21-4 比较输出动作.....	582
图 21-5 捕获输入动作.....	583
图 21-6 软件同步动作.....	584
图 21-7 时钟输入端口的滤波功能.....	585
图 21-8 锯齿波模式时缓存动作.....	586
图 21-9 32 位级联计数动作.....	587
图 21-10 通用 PWM 输出例.....	587
图 21-11 位置模式-基本计数.....	589
图 21-12 位置计数模式-相位差计数（1 倍计数）.....	589
图 21-13 位置计数模式-相位差计数（2 倍计数）.....	590
图 21-14 位置计数模式-相位差计数（4 倍计数）.....	590
图 21-15 位置计数模式-方向计数.....	590
图 21-16 公转计数模式-Z 相计数.....	591
图 21-17 公转计数模式-位置溢出计数.....	591
图 21-18 公转计数模式-混合计数.....	592
图 22-1 Timer0 基本框图.....	614
图 22-2 Timer0 计数时序图.....	616
图 23-1 RTC 的基本框图.....	626
图 24-1 硬件启动例.....	648
图 24-2 软件启动例.....	649
图 24-3 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）.....	650
图 24-4 计数器下溢动作例.....	653
图 24-5 计数器刷新动作例.....	654
图 25-1 USART 系统框图.....	662
图 25-2 USART 数据格式.....	664
图 25-3 USART 发送数据图例 1.....	666
图 25-4 USART 发送数据图例 2.....	666
图 25-5 USART 内部同步和采样时序.....	667
图 25-6 USART 接收数据图例 1.....	669
图 25-7 USART 接收数据图例 2.....	670
图 25-8 多处理器通信图例.....	674
图 25-9 多处理器模式数据格式.....	674
图 25-10 多处理器模式发送数据图例.....	676
图 25-11 多处理器模式接收数据图例 1.....	677
图 25-12 多处理器模式接收数据图例 2.....	678
图 25-13 智能卡连接示意图.....	679
图 25-14 智能卡模式同步时序和采样时序图.....	680
图 25-15 智能卡模式同步时序和采样时序图.....	680
图 25-16 智能卡模式发送数据图例.....	682

图 25-17 智能卡模式接收数据图例	682
图 25-18 时钟同步模式数据格式	685
图 25-19 时钟同步模式发送数据图例 1	687
图 25-20 时钟同步模式发送数据图例 2	687
图 25-21 时钟同步模式接收数据图例 1	688
图 25-22 时钟同步模式接收数据图例 2	689
图 26-1 I ² C 系统框图	707
图 26-2 I ² C 总线的结构例	708
图 26-3 I ² C 总线的时序图	709
图 26-4 I ² C 总线的数据格式	710
图 26-5 7 位地址格式的主机发送数据时序图（例）	712
图 26-6 7 位地址格式的主机接收数据的时序图（例）	713
图 26-7 7 位地址格式的从机发送模式时序图（例）	714
图 26-8 7 位地址格式从机接收模式时序图（例）	715
图 26-9 SCL 同步时序	716
图 26-10 从机发送时序图（1）	717
图 26-11 从机发送时序图（2）	718
图 26-12 选择 7 位地址格式时的时序	720
图 26-13 选择 10 位地址格式时的时序	721
图 26-14 数字滤波电路框图	730
图 27-1 系统框图	754
图 27-2 主机模式结构	758
图 27-3 三线式时钟同步运行	759
图 27-4 数据格式	761
图 27-5 数据传送格式图（CPHA=0）	763
图 27-6 数据传送格式（CPHA=1）	764
图 27-7 全双工同步串行通信	765
图 27-8 只进行发送通信	766
图 27-9 奇偶校验流程	775
图 27-10 过载错误处理	777
图 27-11 启用时钟自动停止功能时的动作示意图（CPHA=1）	779
图 27-12 启用时钟自动停止功能时的动作示意图（CPHA=0）	779
图 27-13 奇偶校验错误	780
图 28-1 QSPI 的模块构成图	794
图 28-2 默认区域设定及 AHB 总线空间内存映射关系图	795
图 28-3 QSPI-ROM 空间内存映像图	796
图 28-4 扩展式 SPI 协议动作示意图 1（快速读模式）	797
图 28-5 扩展式 SPI 协议动作示意图 2（四线式输入输出快速读模式）	797
图 28-6 二线式 SPI 协议动作示意图（快速读模式）	798
图 28-7 四线式 SPI 协议动作示意图（快速读模式）	798
图 28-8 串行接口的基本时序图	799
图 28-9 基准时钟选择 HCLK 三分频时输出时钟占空比补正示意图	802

图 28-10 QSSL 建立时间配置示意图	803
图 28-11 QSSN 保持时间配置示意图	803
图 28-12 数据接收延迟示意图	804
图 28-13 标准读总线周期示意图	806
图 28-14 快速读总线周期示意图	807
图 28-15 选择 XIP 模式的快速读总线周期示意图	807
图 28-16 二线式输出快速读总线周期示意图	808
图 28-17 选择 XIP 模式的二线式输出快速读总线周期示意图	809
图 28-18 二线式输入输出快速读总线周期示意图	810
图 28-19 选择 XIP 模式的二线式输入输出快速读总线周期示意图	810
图 28-20 四线式输出快速读总线周期示意图	811
图 28-21 选择 XIP 模式的四线式输出快速读总线周期示意图	812
图 28-22 四线式输入输出快速读总线周期示意图	813
图 28-23 选择 XIP 模式的四线式输入输出快速读总线周期示意图	814
图 28-24 进入 4-Byte 模式指令总线周期示意图	815
图 28-25 退出 4-Byte 模式指令总线周期示意图	816
图 28-26 写许可指令总线周期示意图	817
图 28-27 独立转换的单一闪存数据读取操作示意图	818
图 28-28 预读取功能有效时数据读取操作示意图	819
图 28-29 使用 QSPI 总线周期延长功能的数据读取操作示意图	820
图 28-30 XIP 模式控制示意图	821
图 29-1 I2S 系统框图	839
图 29-2 I2S Philips 协议波形 (16/32 位全精度)	842
图 29-3 I2S Philips 协议波形 (16 位数据封装在 32 位帧中)	843
图 29-4 I2S Philips 协议波形 (24 位数据封装在 32 位帧中)	843
图 29-5 I2S MSB 协议波形 (16/32 位全精度)	843
图 29-6 I2S MSB 协议波形 (16 位数据封装在 32 位帧中)	844
图 29-7 I2S MSB 协议波形 (24 位数据封装在 32 位帧中)	844
图 29-8 I2S LSB 协议波形 (16/32 位全精度)	845
图 29-9 I2S LSB 协议波形 (16 位数据封装在 32 位帧中)	845
图 29-10 I2S LSB 协议波形 (24 位数据封装在 32 位帧中)	845
图 29-11 I2S PCM 协议波形 (16/32 位全精度)	846
图 29-12 I2S PCM 协议波形 (16 位数据封装在 32 位帧中)	846
图 29-13 I2S PCM 协议波形 (24 位数据封装在 32 位帧中)	847
图 29-14 音频采样频率定义	848
图 29-15 时钟发生器架构	848
图 29-16 主机只接收暂时停止接收	855
图 29-17 PCM 短帧主机发送暂停后重新发送方式一	856
图 29-18 PCM 短帧主机发送暂停后重新发送方式二	856
图 30-1 CAN 系统框图	867
图 30-2 CAN 位时间定义图	868
图 30-3 CAN TBUF 寄存器写发送缓冲器和示意图	870

图 30-4 CAN RBUF 寄存器读接收缓冲器示意图	870
图 30-5 CAN ACF 寄存器访问筛选器组示意图	871
图 30-6 CAN LBMI 和 LBME 示意图	875
图 31-1 USBFS 系统框图	912
图 31-2 USBFS 主机模式系统构建图	915
图 31-3 USBFS 设备模式系统构建图	920
图 31-4 USBFS 动态更新 USBFS_HFIR 寄存器示意图	927
图 31-5 USBFS 主机模式下 FIFO 架构示意图	928
图 31-6 USBFS 设备模式下 FIFO 架构示意图	930
图 32-1 AES 加密流程示意图	1046
图 32-2 HASH 算法流程图	1053
图 32-3 TRNG 系统框图	1060
图 34-1 CRC 模块框图	1074
图 36-1 调试控制系统	1121
图 36-2 调试控制系统	1122
图 36-3 JTAG-DP 到 SW-DP 切换时序	1123
图 36-4 TPIU 框图	1132

简介 (Overview)

HC32F460 系列是基于 ARM® Cortex®-M4 32-bit RISC CPU，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 DMA 专用 MPU 单元，保障系统运行的安全性。

HC32F460 系列集成了高速片上存储器，包括最大 512KB 的 Flash，最大 192KB 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, USB 专用 DMA 等。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。

HC32F460 系列集成了丰富的外设功能。包括 2 个独立的 12bit 2.5MSPS ADC，1 个增益可调 PGA，3 个电压比较器 (CMP)，3 个多功能 16bit PWM Timer (Timer6) 支持 6 路互补 PWM 输出，3 个电机 PWM Timer(Timer4)支持 18 路互补 PWM 输出，6 个 16bit 通用 Timer(TimerA) 支持 3 路 3 相正交编码输入及 48 路 Duty 独立可设 PWM 输出，11 个串行通信接口 (I2C/UART/SPI)，1 个 QSPI 接口，1 路 CAN，4 个 I2S 支持音频 PLL，2 个 SDIO，1 个 USB FS Controller 带片上 FS PHY 支持 Device/Host。

HC32F460 系列支持宽电压范围 (1.8-3.6V)，宽温度范围 (-40-105°C) 和各种低功耗模式。Run 模式和 Sleep 模式下可切换超高速模式 ($\leq 200\text{MHz}$)、高速模式 ($\leq 168\text{MHz}$) 和超低速模式 ($\leq 8\text{MHz}$)。支持低功耗模式的快速唤醒，STOP 模式唤醒最快至 2us，Power Down 模式唤醒最快至 20us。

典型应用

HC32F460 系列提供 48pin、64pin、100pin 的 LQFP 封装，48pin、60pin 的 QFN 封装，100pin 的 VFBGA 封装，适用于高性能电机变频控制、智能硬件、IoT 连接模块等领域。

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”。

1 存储器映射 (Memory Mapping)

1.1 存储器映射

本 MCU 支持 4GB 的线性地址空间，地址从 0x0000_0000 到 0xFFFF_FFFF。存储器映射的详细信息，请参见表 1-1。

存储器分类		开始地址	结束地址	空间大小	模块 ^{*3}	保护 ^{*4}	说明
系统		0xE010_0000	0xFFFF_FFFF	511MB	Reserved		自定义空间
	私有外设 外部总线	0xE00F_F000 0xE004_2400 0xE004_2000 0xE004_1000 0xE004_0000	0xE00F_FFFF 0xE00F_EFFF 0xE004_23FF 0xE004_1FFF 0xE004_0FFF	4KB 755KB 1KB 4KB 4KB	ROMTABLE DBG ETM TPIU		调试控制寄存器 区域 该 MCU 无 ETM
	私有外设 内部总线	0xE000_F000 0xE000_E000 0xE000_3000 0xE000_2000 0xE000_1000 0xE000_0000	0xE003_FFFF 0xE000_EFFF 0xE000_DFFF 0xE000_2FFF 0xE000_1FFF 0xE000_0FFF	196KB 4KB 44KB 4KB 4KB 4KB	SCS FPB DWT ITM		系统控制空间 NVIC/MPU 等
外部设备	-	0xA000_0000	0xDFFF_FFFF	1024MB	Reserved		
外部 RAM	AHB5 时钟： HCLK	0x9800_0000 0x6000_0000	0x9FFF_FFFF 0x97FF_FFFF	128MB 896MB	QSPI Reserved		

存储器分类		开始地址	结束地址	空间大小	模块 ^{*3}	保护 ^{*4}	说明
外设	-	0x4400_0000	0x5FFF_FFFF	448MB	Reserved		
		0x4200_0000	0x43FF_FFFF	32MB	PeriBitBand		CPU 以外主机 Reserved
		0x4010_0000	0x41FF_FFFF	31MB	Reserved		
	AHB3 时钟： PCLK1	0x400C_0000	0x400F_FFFF	256KB	USBFS		
		0x4007_0800	0x400B_FFFF	318KB	BLANK		
		0x4007_0400	0x4007_07FF	1KB	CAN		
	AHB2 时 钟 : PCLK1	0x4007_0000	0x4007_03FF	1KB	SDIOC_2		
		0x4006_FC00	0x4006_FFFF	1KB	SDIOC_1		
		0x4006_0000	0x4006_FBFF	63KB	BLANK		
	AHB1 时钟： HCLK	0x4005_5800	0x4005_FFFF	42KB	BLANK		
		0x4005_5400	0x4005_57FF	1KB	PERIC		
		0x4005_4800	0x4005_53FF	3KB	BLANK		
		0x4005_4000	0x4005_47FF	2KB	SYSC	带保护	
		0x4005_3800	0x4005_3FFF	2KB	GPIO		
		0x4005_3400	0x4005_37FF	1KB	DMA_2		
		0x4005_3000	0x4005_33FF	1KB	DMA_1		
		0x4005_2C00	0x4005_2FFF	1KB	DCU_4		
		0x4005_2800	0x4005_2BFF	1KB	DCU_3		
		0x4005_2400	0x4005_27FF	1KB	DCU_2		
		0x4005_2000	0x4005_23FF	1KB	DCU_1		
	外设	0x4005_1000	0x4005_1FFF	4KB	INTC	带保护	
		0x4005_0C00	0x4005_0FFF	1KB	KEYSCAN		

存储器分类		开始地址	结束地址	空间大小	模块 ^{*3}	保护 ^{*4}	说明
		0x4005_0800	0x4005_0BFF	1KB	RAMIF	带保护	
		0x4005_0400	0x4005_07FF	1KB	BLANK		
		0x4005_0000	0x4005_03FF	1KB	DMPU	带保护	
	APB4	0x4004_EC00	0x4004_FFFF	5KB	BLANK		
时钟:	0x4004_E800	0x4004_EBFF	1KB	I2C_3			
PCLK3	0x4004_E400	0x4004_E7FF	1KB	I2C_2			
	0x4004_E000	0x4004_E3FF	1KB	I2C_1			
	0x4004_C400	0x4004_DFFF	7KB	BLANK			
	0x4004_C000	0x4004_C3FF	1KB	RTC	带保护		
	0x4004_A800	0x4004_BFFF	6KB	BLANK			
	0x4004_A400	0x4004_A7FF	1KB	OTS			
	0x4004_A000	0x4004_A3FF	1KB	CMP			
	0x4004_9800	0x4004_9FFF	2KB	BLANK			
	0x4004_9400	0x4004_97FF	1KB	SWDT	带保护		
	0x4004_9000	0x4004_93FF	1KB	WDT	带保护		
	0x4004_8800	0x4004_8FFF	2KB	BLANK			
	0x4004_8400	0x4004_87FF	1KB	FCM			
	0x4004_8000	0x4004_83FF	1KB	MSTP	带保护		
	APB3	0x4004_1400	0x4004_7FFF	27KB	BLANK		
时钟:	0x4004_1000	0x4004_13FF	1KB	TRNG	带保护		
PCLK4	0x4004_0800	0x4004_0FFF	2KB	BLANK			
	0x4004_0400	0x4004_07FF	1KB	ADC_2			
	0x4004_0000	0x4004_03FF	1KB	ADC_1			
	APB2	0x4002_5000	0x4003_FFFF	108KB	BLANK		
时钟:	0x4002_4C00	0x4002_4FFF	1KB	Timer4_3			
PCLK1	0x4002_4800	0x4002_4BFF	1KB	Timer4_2			

存储器分类		开始地址	结束地址	空间大小	模块 ^{*3}	保护 ^{*4}	说明
外设	0x4002_0000 ~ 0x4002_4400	0x4002_4400	0x4002_47FF	1KB	Timer0_2		
		0x4002_4000	0x4002_43FF	1KB	Timer0_1		
		0x4002_2800	0x4002_3FFF	6KB	BLANK		
		0x4002_2400	0x4002_27FF	1KB	I2S_4		
		0x4002_2000	0x4002_23FF	1KB	I2S_3		
		0x4002_1800	0x4002_1FFF	2KB	BLANK		
		0x4002_1400	0x4002_17FF	1KB	USART_4		
		0x4002_1000	0x4002_13FF	1KB	USART_3		
		0x4002_0800	0x4002_0FFF	2KB	BLANK		
		0x4002_0400	0x4002_07FF	1KB	SPI_4		
		0x4002_0000	0x4002_03FF	1KB	SPI_3		
	APB1 时钟: PCLK1	0x4001_E800	0x4001_FFFF	6KB	BLANK		
		0x4001_E400	0x4001_E7FF	1KB	I2S_2		
		0x4001_E000	0x4001_E3FF	1KB	I2S_1		
		0x4001_D800	0x4001_DFFF	2KB	BLANK		
		0x4001_D400	0x4001_D7FF	1KB	USART_2		
		0x4001_D000	0x4001_D3FF	1KB	USART_1		
		0x4001_C800	0x4001_CFFF	2KB	BLANK		
		0x4001_C400	0x4001_C43F	1KB	SPI_2		
		0x4001_C000	0x4001_C3FF	1KB	SPI_1		计数时
		0x4001_8000	0x4001_BFFF	16KB	Timer6		钟:PCLK0
		0x4001_7C00	0x4001_7FFF	1KB	EMB		
		0x4001_7400	0x4001_7BFF	2KB	BLANK		
		0x4001_7000	0x4001_73FF	1KB	Timer4_1		
		0x4001_6800	0x4001_6FFF	2KB	BLANK		

存储器分类		开始地址	结束地址	空间大小	模块 ^{*3}	保护 ^{*4}	说明
SRAM	时钟： AHB3 PCLK1	0x4001_6400	0x4001_67FF	1KB	TimerA_6		
		0x4001_6000	0x4001_63FF	1KB	TimerA_5		
		0x4001_5C00	0x4001_5FFF	1KB	TimerA_4		
		0x4001_5800	0x4001_5BFF	1KB	TimerA_3		
		0x4001_5400	0x4001_57FF	1KB	TimerA_2		
		0x4001_5000	0x4001_53FF	1KB	TimerA_1		
		0x4001_0C00	0x4001_4FFF	17KB	BLANK		
		0x4001_0800	0x4001_0BFF	1KB	AOS		内部触发事件寄存器区域
		0x4001_0400	0x4001_07FF	1KB	EFM	带保护	
		0x4001_0000	0x4001_03FF	1KB	BLANK		
		0x4000_9000	0x4000_FFFF	28KB	BLANK		
	时钟： HCLK	0x4000_8C00	0x4000_8FFF	1KB	CRC	带保护	
		0x4000_8800	0x4000_8BFF	1KB	BLANK		
	时钟： HCLK	0x4000_8400	0x4000_87FF	1KB	HASH256	带保护	
		0x4000_8000	0x4000_83FF	1KB	AES128	带保护	
	-	0x4000_0000	0x4000_7FFF	32KB	Reserved		
SRAM	时钟： HCLK	0x2400_0000	0x3FFF_FFFF	448MB	Reserved		
		0x2200_0000	0x23FF_FFFF	32MB	SRAMBitBand		CPU 以外主机 Reserved
		0x2010_0000	0x21FF_FFFF	31MB	Reserved		
	时钟： HCLK	0x200F_1000	0x200F_FFFF	60KB	BLANK		
		0x200F_0000	0x200F_0FFF	4KB	Ret_SRAM		
		0x2002_7000	0x200E_FFFF	804KB	BLANK		
		0x2002_0000	0x2002_6FFF	28KB	SRAM3		ECCRAM
		0x2001_0000	0x2001_FFFF	64KB	SRAM2		

存储器分类		开始地址	结束地址	空间大小	模块 ^{*3}	保护 ^{*4}	说明
		0x2000_0000	0x2000_FFFF	64KB	SRAM1		
CODE	SRAM 时钟： HCLK	0x1FFF_8000	0x1FFF_FFFF	32KB	SRAMH		
	-	0x0320_0000	0x1FFF_7FFF	478.97MB	BLANK		
	REMAP 时钟： HCLK	0x0208_0000	0x020F_FFFF	512KB	REMAP1		地址重映射区域 1
		0x0200_0000	0x0207_FFFF	512KB	REMAP0		地址重映射区域 0
	-	0x0008_0000	0x01FF_FFFF	31.5MB	BLANK		
	ROM 时 钟 : HCLK	0x0000_0000	0x0007_FFFF	512KB	Embedded Flash ^{*5}		

表 1-1 存储器映射

*1 请参考 ARM Cortex-M4 说明书“存储器系统”。

*2 总线说明请参考总线章节。

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

*4 带保护功能的模块，在保护功能有效时只支持CPU特权模式访问。具体寄存器及说明参考 DMPU 章节。

*5 在 256KB 产品中，ROM 地址为 0x0000_0000~0x0003_FFFF。

1.2 外部空间映射

QSPI 空间分为 2 段空间，QSPI I/O 寄存器空间 64MB 和外部 QSPI 设备空间 64MB。分配关系请参考表 1-2。

QSPI	0x9800_0000	0x9FFF_FFFF	128MB	QSPI I/O 寄存器	0x9C00_0000	0x9FFF_FFFF	64MB
				外部 QSPI 设备	0x9800_0000	0x9BFF_FFFF	64MB

表 1-2 QSPI 地址空间分配

1.3 位段空间

Cortex™-M4F 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。

在本 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™-M4F 访问，对于其它总线主接口（如 DMA）无效。

1.4 地址重映射

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

重映射地址 0：

0x0200_0000~0x0208_0000（视重映射空间而定 MMF_REMCR0/1.RMSIZE[4:0]）

重映射地址 1：

0x0208_0000~0x0210_0000（视重映射空间而定 MMF_REMCR0/1.RMSIZE[4:0]）

重映射功能有效时，地址对应如表 1-3 所示。

寄存器设定	重映射地址 (CPU地址—CPUADDR[31:0])	源地址		
		高3位地址	中地址	低地址
RMSIZE[4:0]=01110情形 (重映射空间：16K)	0x0200_0000~0x0200_3FFF	全0	RMTADDR[16:2]	CPUADDR[13:0]
RMSIZE[4:0]=01111情形 (重映射空间：32K)	0x0200_0000~0x0200_7FFF	全0	RMTADDR[16:3]	CPUADDR[14:0]
RMSIZE[4:0]=10000情形 (重映射空间：64K)	0x0208_0000~0x0208_FFFF	全0	RMTADDR[16:4]	CPUADDR[15:0]
RMSIZE[4:0]=10001情形 (重映射空间：128K)	0x0208_0000~0x0209_FFFF	全0	RMTADDR[16:5]	CPUADDR[16:0]

表 1-3 目标地址配置示例

举例，使用重映射地址 0 功能，设定源地址为主闪存 FLASH 地址 0x0000_8000，重映射空间 32K，寄存器 MMF_REMCR0 需设定为 0x8000_800F。

使用重映射地址 1 功能，设定源地址为高速 SRAM 地址 0x1FFF_8000，重映射空间为 16K，寄存器 MMF_REMCR1 需设定为 0x9FFF_800E。

注意：源地址的起始地址应设为重映射空间的整数倍。

1.5 重映射寄存器

重映射模块共有三个寄存器。地址空间如下：

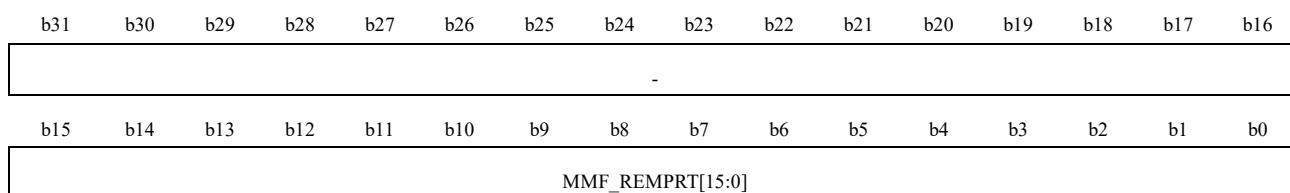
寄存器地址：0x4001_0500

寄存器名	符号	偏移地址	位宽	复位值
访问保护寄存器	MMF_REMPRT	0x0000	32	0x0000_0000
重映射控制寄存器0	MMF_REMCR0	0x0004	32	0x0000_0000
重映射控制寄存器1	MMF_REMCR1	0x0008	32	0x0000_0000

表 1-4 寄存器列表

1.5.1 访问保护寄存器 (MMF_REMPRT)

复位值：0x0000_0000



位	标记	位名	功能	读写
b31~16	Reserved	-	读出时为“0”，写入时写“0”	R
b15~0	MMF_REMPRT[15:0]	保护寄存器	寄存器MMF_REMCR0和MMF_REMCR1写保护： 先对MMF_REMPRT[15:0]写入0x0123接着写入0x3210解除保护； 寄存器MMF_REMCR0和MMF_REMCR1写保护状态时，读寄存器 为0 寄存器MMF_REMCR0和MMF_REMCR1解除写保护状态时，读寄 存器为1	R/W

1.5.2 重映射控制寄存器 (MMF_REMCRx) (x=0, 1)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31	EN	重映射有效位	0: 重映射无效 1: 重映射有效	R/W
b30~29	Reserved	-	读出时为“0”, 写入时写“0”	R
b28~12	RMTADDR[16:0]	源地址	有效位数和RMSIZE[4:0]设定有关。 设定可参考表 1-3。	R/W
b11~b5	Reserved	-	读出时为“0”, 写入时写“0”	R
b4~b0	RMSIZE[4:0]	重映射空间	设定重映射空间 00000~01011: 保留, 设定禁止 01100: 4KB 01101: 8KB 01110: 16KB 01111: 32KB 10000: 64KB 10001: 128KB 10010: 256KB 10011: 512KB (256KB 产品设定禁止) 10100~11111: 保留, 设定禁止	R/W

2 总线架构 (BUS)

2.1 概述

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

- 主机总线
 - Cortex-M4F 内核 CPU-I 总线, CPU-D 总线, CPU-S 总线
 - 系统 DMA_1 总线, 系统 DMA_2 总线
 - USBFS_DMA 总线
- 从机总线
 - Flash ICODE 总线
 - Flash DCODE 总线
 - Flash MCODE 总线(CPU 以外其他主机访问 Flash 的总线)
 - 高速 SRAMH(SRAMH 32KB) 总线
 - 系统 SRAMA (SRAM1 64KB) 总线
 - 系统 SRAMB(SRAM2 64KB, SRAM3 28KB, Ret_SRAM 4KB) 总线
 - APB1 外设总线(EMB/Timers/SPI/USART/I2S)
 - APB2 外设总线(Timers/SPI/USART/I2S)
 - APB3 外设总线(ADC/PGA/TRNG)
 - APB4 外设总线(FCM/WDT/CMP/OTS/RTC/WKTM/I2C)
 - AHB1 外设总线(KEYSCAN/INTC/DCU(GPIO/SYSC)
 - AHB2 外设总线(CAN/SDIOC)
 - AHB3 外设总线(AES/HASH/CRC/USBFS)
 - AHB4 外设总线(SDIOC)
 - AHB5 外设总线(QSPI)

借助总线矩阵，可以实现主机总线到从机总线高效率的并发访问。

2.2 总线架构

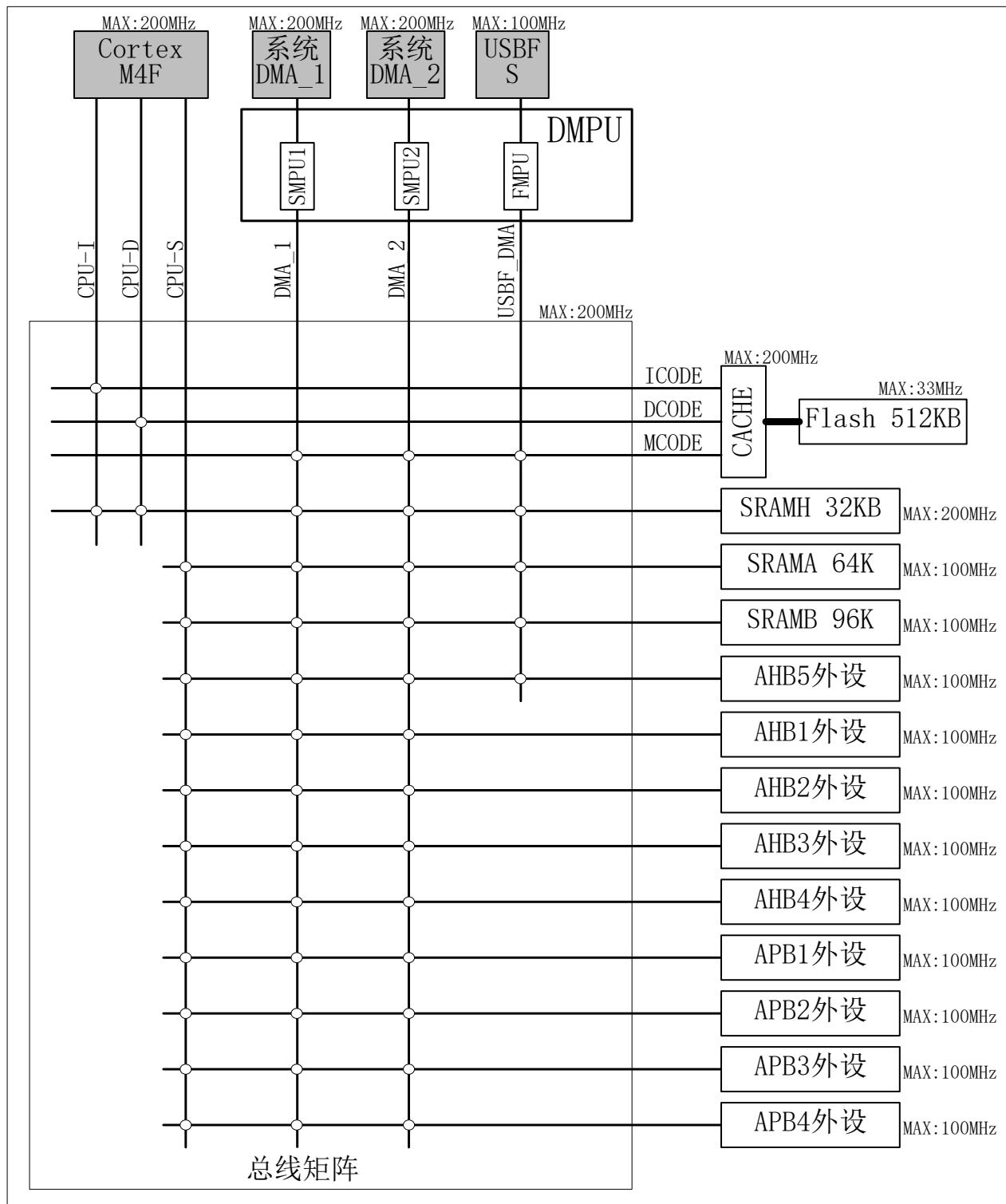


图 2-1 总线架构

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
M4F 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
M4F 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
M4F 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 SRAMA/SRAMB/所有外设以及 AHB5 外部扩展空间。
- DMA_1 总线，DMA_2 总线
系统 DMA_1/系统 DMA_2 专用总线，DMA_1/DMA_2 通过此总线访问数据存储器和外设，访问对象为 Flash/SRAMH/SRAMA/SRAMB/所有外设以及 AHB5 外部扩展空间。
- USBFS_DMA 总线
USBFS 的 DMA 专用总线，USBFS 通过此总线访问所有存储器空间。访问对象为 Flash/SRAMH/SRAMA/SRAMB/AHB5 外部扩展空间。

2.3 总线功能

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

通过总线矩阵，不同主机总线的访问目标不冲突时，各访问能够同时进行。例如 CPU-I 访问 Flash，CPU-D 访问 SRAMH，CPU-S 访问 APB 外设，DMA_1 访问 SRAMA，DMA_2 访问 SRAMB，USBFS_DMA 访问 AHB5 的外部扩展空间，这些访问可以同时进行。

3 复位控制 (RMU)

3.1 简介

芯片配置了 14 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)

3.2 复位方式和复位标志位

复位方式及产生条件如表 3-1 所示。

复位方式	产生条件
上电复位	VCC 上电
NRST 引脚复位	NRST 引脚输入低电平
欠压复位	VCC 电压降至低于 VBOR 电压
可编程电压检测 1 复位	VCC 电压降至低于 PVD1 电压
可编程电压检测 2 复位	VCC 电压降到低于 PVD2 电压
看门狗复位	看门狗定时器产生刷新错误或者溢出错误
专用看门狗复位	专用看门狗发生刷新错误或者溢出错误
掉电唤醒复位	通过设置掉电模式产生的复位，内核在掉电唤醒事件发生后从复位状态唤醒
软件复位	设置复位寄存器位（ARM 寄存器 AIRCR.SYSRESETREQ 位）
MPU 错误复位	MPU 访问错误产生的复位
RAM 奇偶校验复位	RAM 发生奇偶校验错误时产生的复位
RAM ECC 错误复位	RAM 发生 ECC 错误时产生的复位
时钟频率异常复位	时钟频率监视功能（FCM）检测到时钟周期错误时
外部高速振荡器异常停振复位	外部高速振荡器异常停振时产生的复位

表 3-1 复位方式及产生条件

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

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 奇偶校验错误复位	RAM ECC 错误复位	时钟频率异常复位	外部高速振荡器异常停振复位
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)	X	—	—	—	—	—	—	—	√	—	—	—	—	—
软件复位标志 (RMU_RSTF0.SWRF)	X	X	X	—	—	—	—	—	X	√	—	—	—	—
MPU 错误复位(RMU_RSTF0.MPUERF)	X	X	X	—	—	—	—	—	X	—	√	—	—	—
RAM 奇偶错误复位(RMU_RSTF0.RAPERF)	X	X	X	—	—	—	—	—	X	—	—	√	—	—
RAM ECC 复位(RMU_RSTF0.RAECRF)	X	X	X	—	—	—	—	—	X	—	—	—	√	—
时钟频率异常复位 (RMU_RSTF0.CKFERF)	X	X	X	—	—	—	—	—	X	—	—	—	—	√
外部高速振荡器异常停振复位 (RMU_RSTF0.XTALERF)	X	X	X	—	—	—	—	—	X	—	—	—	—	√

√：置位 X：清零 —：不变

表 3-2 复位方式及复位标志

3.3 复位时序

3.3.1 上电复位

上电复位是上电复位电路引起的内部复位，时序如图 3-1。在将 NRST 引脚置为高电平的状态下接通电源，就产生上电复位。VCC 电压高于上电复位电压 V_{POR} 后经过一定的时间 (T_{RSTPOR}) 之后芯片内部复位解除，CPU 开始执行代码。产生上电复位时，上电复位标志 RMU_RSTF0.PORF 被置位。上电复位的详细说明请参照【5.3.1 上电复位/掉电复位动作说明】。

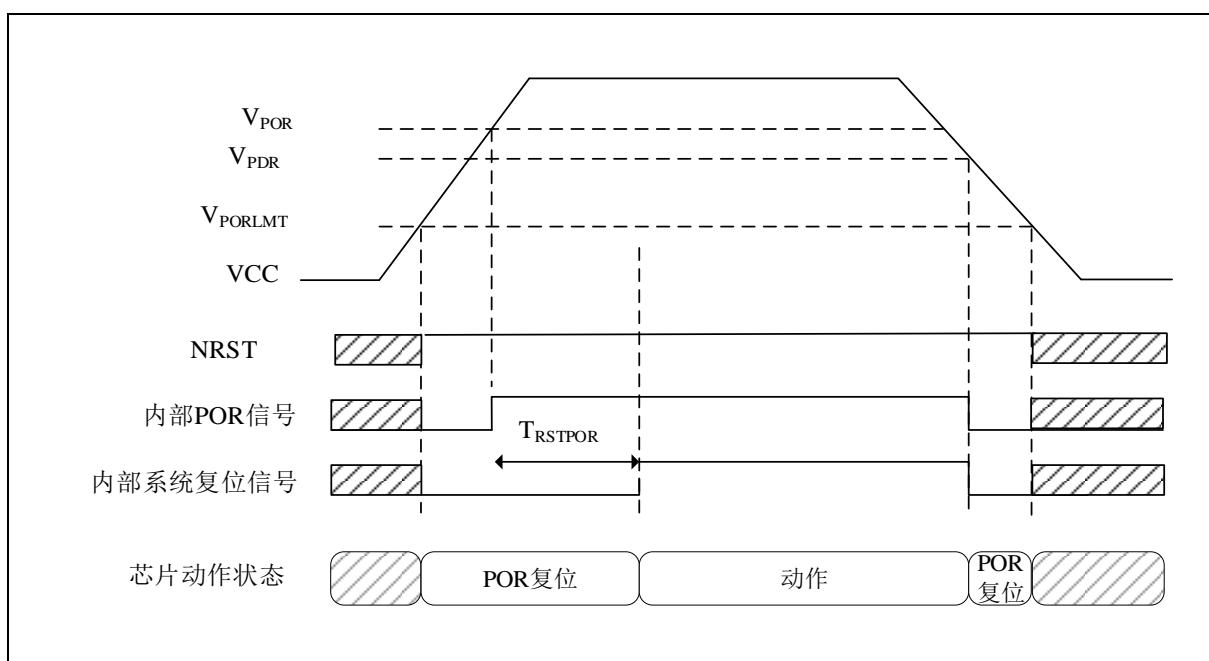


图 3-1 上电复位

3.3.2 NRST 引脚复位

引脚复位是 NRST 引脚被驱动为低电平引起的复位，复位时序如图 3-2。NRST 管脚维持 T_{NRST} 宽度以上的低电平后，经过一定的内部复位时间 (T_{INRST})，解除内部复位。产生 NRST 引脚复位时，引脚复位标志 RMU_RSTF0.PINRF 被置位。

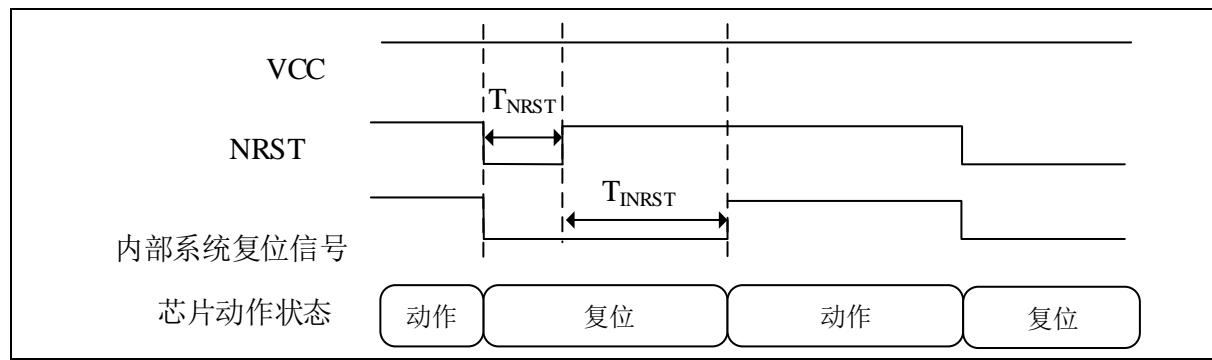


图 3-2 NRST 复位时序

3.3.3 欠压复位

欠压复位是电压监测电路引起的内部复位,时序如图 3-3。欠压通过 ICG 寄存器设置为复位使能后，如果 VCC 电压低于监测电压 V_{BOR} ，RMU_RSTF0.VBORF 被置位。当 VCC 电压高于监测电压 V_{BOR} 经过 T_{RSTBOR} 的复位时间后解除复位。

欠压的复位设定请参考 【5.3.2 欠压复位(BOR)说明】。

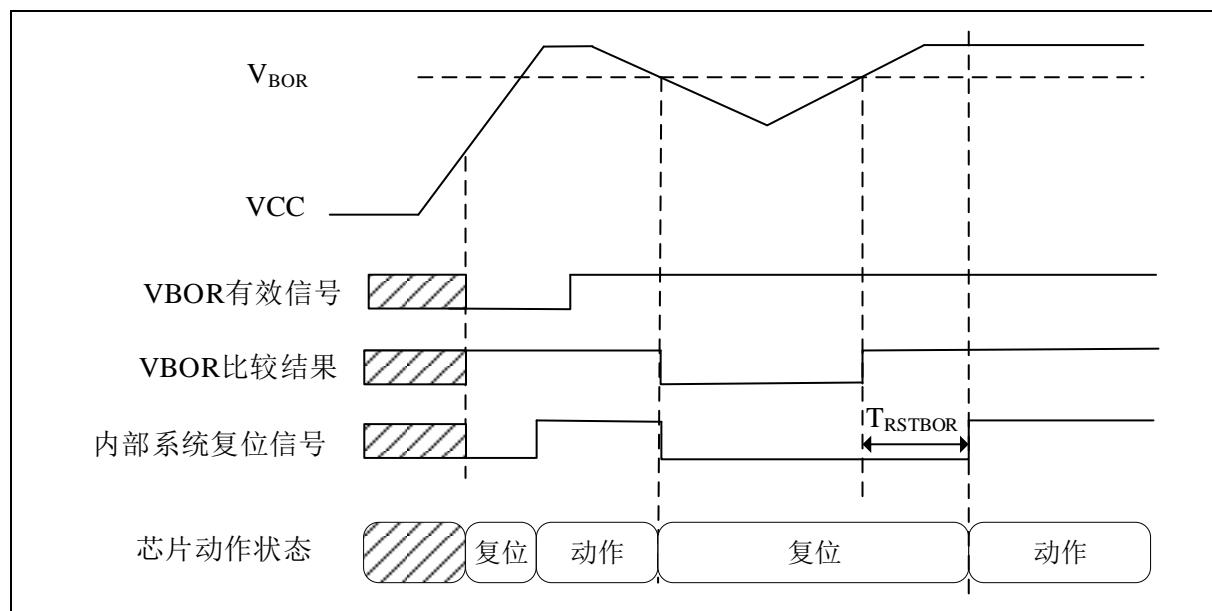


图 3-3 欠压复位

3.3.4 可编程电压检测 1 复位、可编程电压检测 2 复位

可编程电压检测 1 和可编程电压检测 2 复位时电压监测电路引起的复位。

在可编程电压检测 1 有效并设置为复位使能后，如果 VCC 低于可编程电压检测 1 的监测电压，产生可编程电压检测 1 复位，RMU_RSTF0.PVD1F 被置位。当 VCC 电压高于可编程电压检测 1 的监测电压后，经过 PVD1 的复位时间(T_{IPVD1})后解除复位。

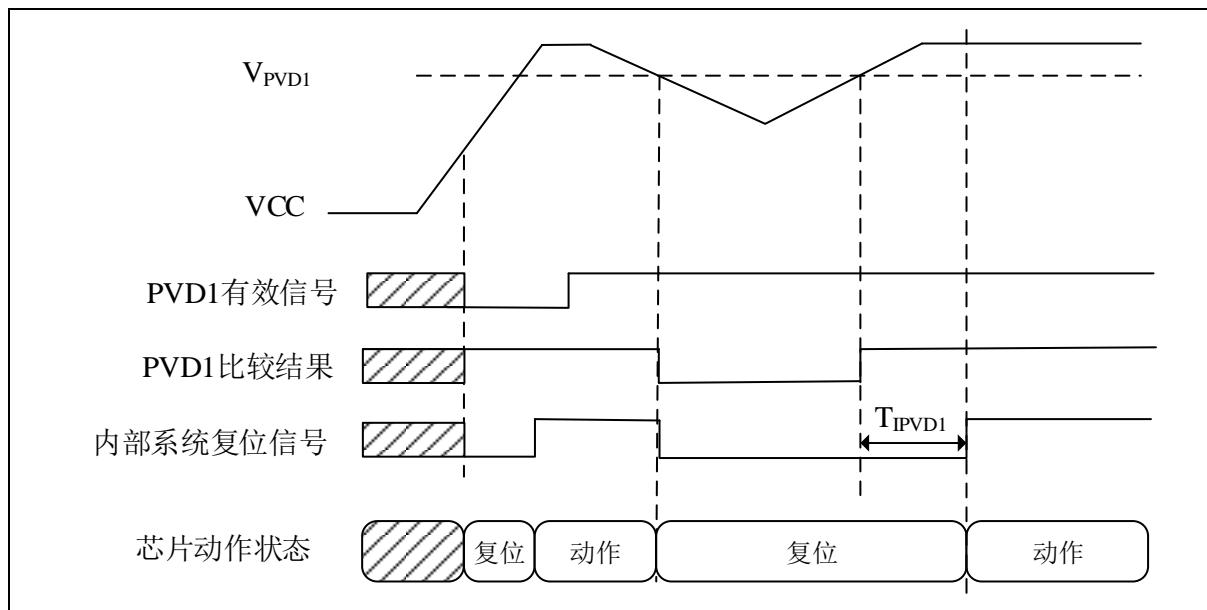


图 3-4 可编程电压检测 1 复位

在可编程电压检测 2 有效并设置为复位使能后，如果 VCC 低于可编程电压检测 2 的监测电压，产生可编程电压检测 2 复位，RMU_RSTF0.PVD2F 被置位。当 VCC 电压高于可编程电压检测 2 的监测电压后，经过 PVD2 的复位时间(T_{IPVD2})后解除复位。

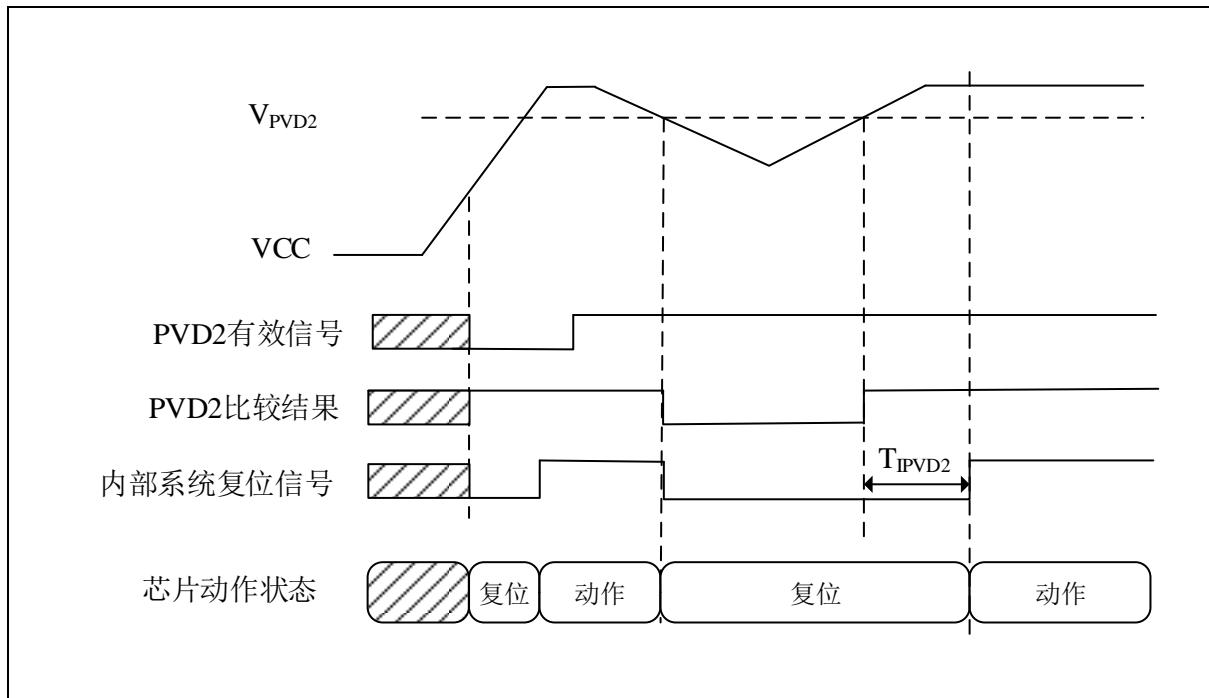


图 3-5 可编程电压检测 2 复位

可编程电压检测 1、可编程电压检测 2 的复位设定请参考【5.3 电源电压检测单元(PVD)说明】。

3.3.5 看门狗复位、专用看门狗复位

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

设定看门狗复位有效后，在看门狗定时器产生下溢或者不在刷新允许期间进行写操作时，产生看门狗复位。看门狗复位将 RMU_RSTF0.WDRF 置位。产生看门狗复位后，经过内部复位时间 T_{RIPT} 之后，芯片解除复位。

设定专用看门狗复位有效后，在专用看门狗定时器产生下溢或者不在刷新允许期间进行写操作时，产生看门狗复位。专用看门狗复位将 RMU_RSTF0.SWDRF 置位。产生专用看门狗复位后，经过内部复位时间 T_{RIPT} 之后，芯片解除复位。

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

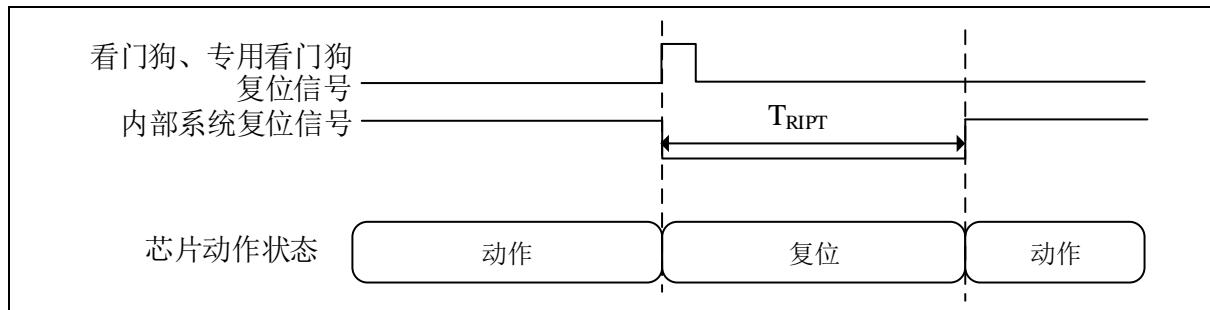
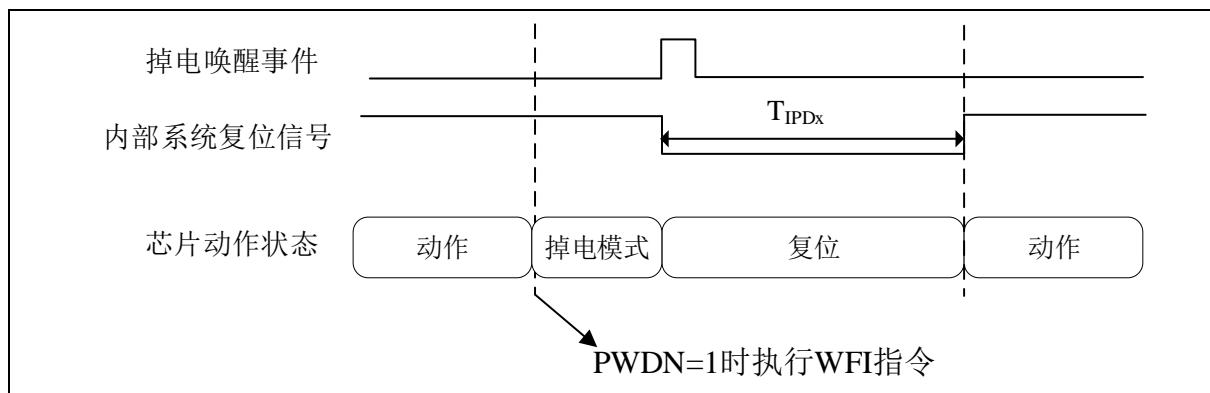


图 3-6 看门狗和专用看门狗复位

3.3.6 掉电唤醒复位

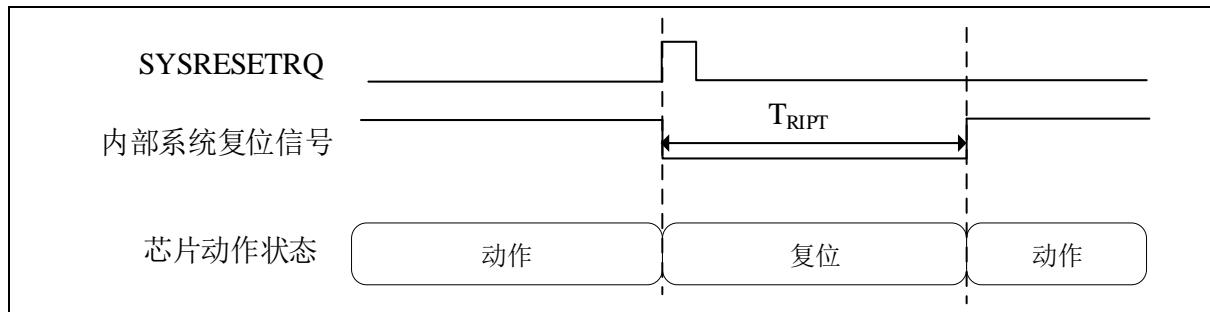
掉电唤醒复位是芯片在设置 PWC_PWRC0.PWDN 为 1 时执行 WFI 命令，进入掉电模式后，通过掉电模式唤醒事件解除掉电模式时产生的内部复位，时序如图 3-7。在解除掉电模式并经过返回时间($T_{IPDx}, x=1,2,3,4$)后，解除掉电唤醒复位。返回时间根据设定的具体掉电模式有所不同，在掉电模式 1 时最小，掉电模式 3 时最大。

有关掉电唤醒复位的详细内容，请参照【5.4.4 掉电模式】。



3.3.7 软件复位

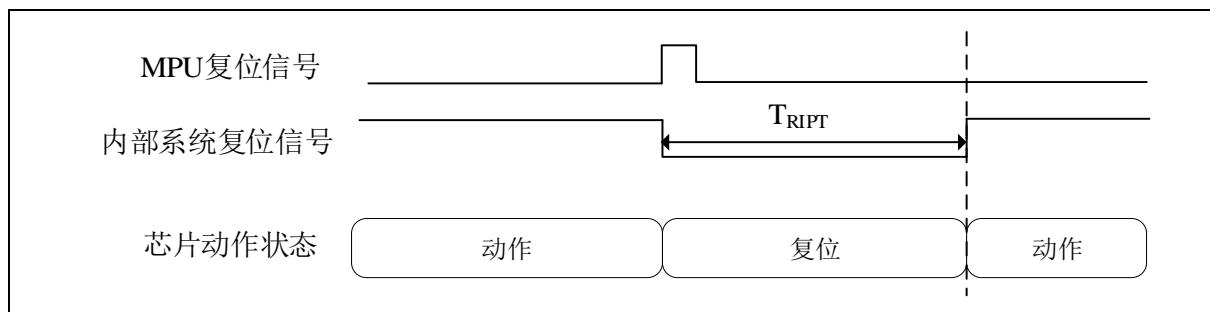
通过写 ARM 寄存器 AIRCR 的 SYSRESETREQ 位产生软件复位。产生软件复位时，RMU_RSTF0.SWRF 位被置位。经过内部复位时间 T_{RIPT} 之后，芯片解除复位。



3.3.8 MPU 错误复位

MPU 错误复位将 RMU_RSTF0.MPUERF 置位，时序如图 3-9。经过内部复位时间 T_{RIPT} 之后，芯片解除复位。

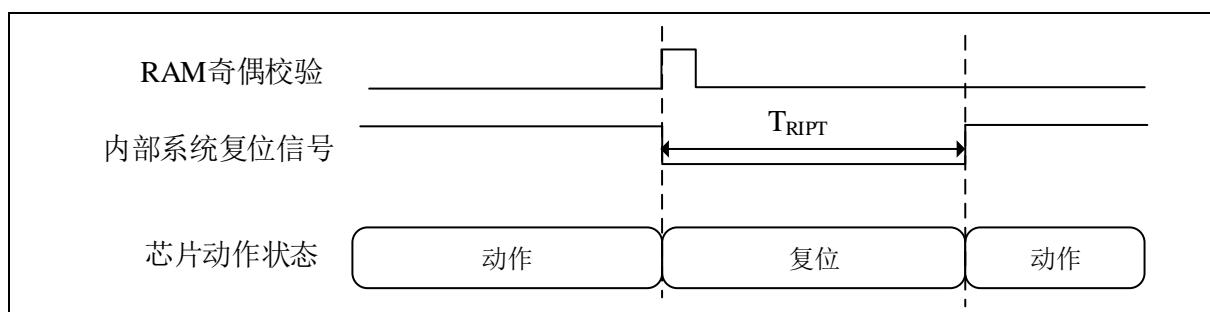
有关 MPU 错误复位的设定，请参照【13 存储保护单元 (MPU)】。



3.3.9 RAM 奇偶校验复位

RAM 奇偶校验发生错误时，产生 RAM 奇偶校验复位，时序如图 3-10。RAM 奇偶校验错误将 RMU_RSTF0.RAPERF 置位。经过内部复位时间 T_{RIPT} 之后，芯片解除复位。

RAM 奇偶校验错误复位的设定，请参照【内置 SRAM (SRAM)】。



3.3.10 RAMECC 复位

RAMECC 校验发生错误时，产生 RAMECC 复位，时序如图 3-11。RAMECC 复位将 RMU_RSTF0.RAECRF 置位。经过内部复位时间 T_{RIPT} 之后，芯片解除复位。

RAMECC 复位的设定，请参照【内置 SRAM（SRAM）】。

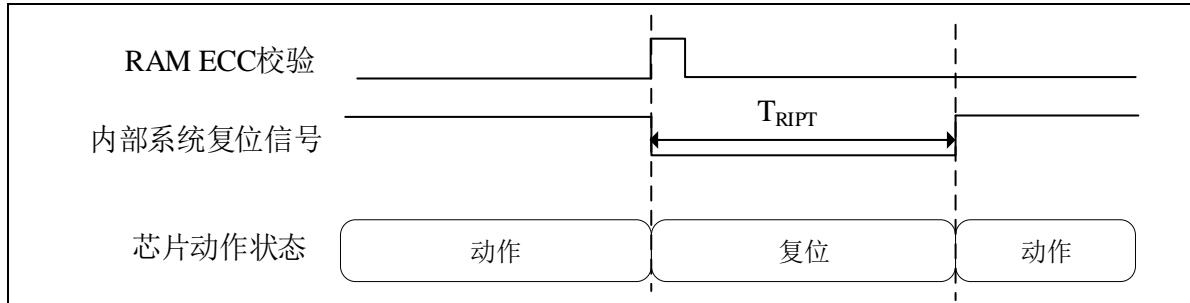


图 3-11 RAMECC 复位

3.3.11 时钟频率异常复位

芯片内置的 FCM 模块在监测到时钟频率发生异常时，如果设置为复位有效就会产生时钟频率异常复位，时序如图 3-12。时钟频率异常复位将 RMU_RSTF0.CKFERF 置位。经过内部复位时间 T_{RIPT} 之后，芯片解除复位。

时钟频率异常复位的设定，请参照【4.10 时钟频率测量】。

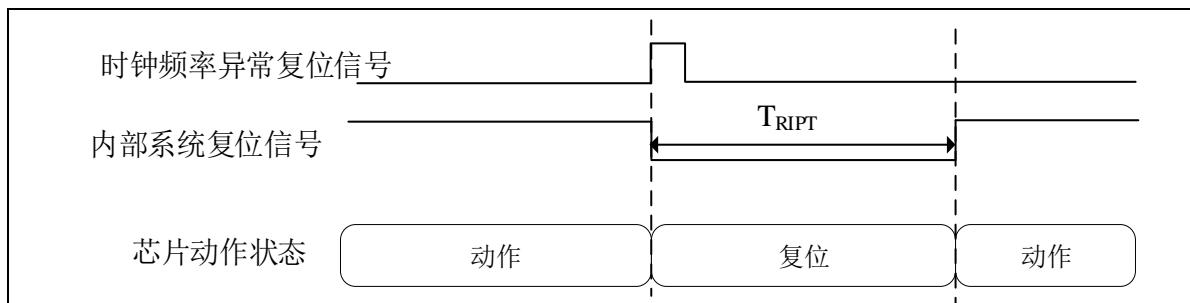


图 3-12 时钟频率异常复位

3.3.12 外部高速振荡器异常停振复位

当芯片的振荡停止检测模块有效并且复位使能后，如果发生外部高速振荡器异常停振时，产生外部高速振荡器异常停振复位，RMU_RSTF0.XTALERF 被置位。经过内部复位时间 T_{RIPT} 之后，芯片解除复位。

外部高速振荡器异常停振复位的设定，请参照【4.5.2 外部高速振荡器故障检测】。

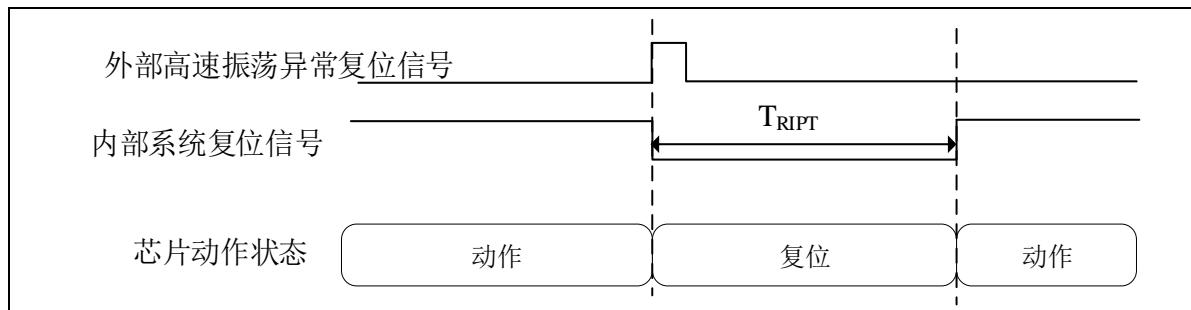


图 3-13 外部高速振荡异常复位

3.3.13 复位方式的判断

根据 RMU_RSTF0 的复位标志可以判断复位方式。当两个或者两个以上复位同时产生时，可能会产生多个复位标志。RMU_RSTF0 中的 MULTIRF 位为 1 时表示发生多个复位。在读 RMU_RSTF0 之后，通过置位 CLRF 位能够将所有的复位标志清 0。在置位将 RMU_RSTF0 清零后，至少需要等待 6 个 CPU 时钟周期后，才能再次读取 RMU_RSTF0 寄存器。

3.3.14 各个模块的复位条件

模块	寄存器	复位源
调试控制器(DBGC)	MCUSTPCTL 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
电源控制(PWC) 时钟控制(CMU)	PWC_PWRC0 PWC_PWRC1 PWC_PDWKE0 PWC_PDWKE1 PWC_PDWKE2 PWC_PDWKES CMU_XTALCFG	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_PDWKF0 PWC_PDWKF1	掉电唤醒复位以外的所有复位源
	PWC_PVDLCSR PWC_PVDCR1 PWC_PVDFCR PWC_PVDCR0 PWC_PWRC2 PWC_PWCMR	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
	PWC_PVDICR[0] PWC_PVDDSR.PVD1DETFLG PWC_PVDICR[4] PWC_PVDDSR.PVD2DETFLG	1. 上电复位 2. 引脚复位 3. 欠压复位 4. 看门狗复位 5. 专用看门狗复位 6. 掉电唤醒复位
	上记以外	所有复位源
	以上模块以外的寄存器	所有复位源

表中所有复位源是指本章简介中描述的 14 种复位源。

3.4 寄存器说明

寄存器一览如表 3-3 所示。

BASE ADDR: 0x400540C0

寄存器名	符号	偏移地址	位宽	复位值
复位状态寄存器	RMU_RSTF0	0x00	16	根据不同的复位方式复位值不同

表 3-3 RMU 寄存器一览

3.4.1 复位标志寄存器 0 (RMU_RSTF0)

复位值: 0xXXXX (根据复位方式不同, 复位值不同)

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CLRF	MULTIRF	XTALERF	CKFERF	RAECRF	RAPERF	MPUERF	SWRF	PDRF	SWDRF	WDRF	PVD2RF	PVD1RF	BORF	PINRF	PORF

位	标记	位名	功能	读写
b15	CLRF	清除复位标志	软件置1, 用于清除复位标志位。 读出时为0。置位动作必须在读取RMU_RSTF0之后进行。 0: 无操作 1: 清零复位标志	R/W
b14	MULTIRF	2个以上复位发生标志位	发生两个及两个以上复位时, 由硬件置位。 通过置位CLRF清零。 0: 未发生两个及两个以上复位 1: 发生两个及两个以上复位时	R/W
b13	XTALERF	外部高速振荡器异常停振复位标志	发生外部高速振荡器异常停振复位时, 由硬件置位。 通过置位CLRF清零。 0: 未发生外部高速振荡器异常停振复位 1: 发生外部高速振荡器异常停振复位	R/W
b12	CKFERF	时钟频率异常复位标志	发生时钟频率异常复位时, 由硬件置位。 通过置位CLRF清零。 0: 未发生时钟频率异常复位 1: 发生时钟频率异常复位	R/W
b11	RAECRF	RAMECC复位标志	发生RAMECC复位时, 由硬件置位。 通过置位CLRF清零。 0: 未发生RAMECC复位 1: 发生RAMECC复位	R/W
b10	RAPERF	RAM奇偶错误复位标志	发生RAM奇偶校验错误复位时, 由硬件置位。通过置位CLRF清零。 0: 未发生RAM奇偶校验错误复位 1: 发生RAM奇偶校验错误复位	R/W

b9	MPUERF	MPU错误复位标志	发生MPU错误复位时，由硬件置位。 通过置位CLRF清零。 0：未发生MPU错误复位 1：发生MPU错误复位	R/W
b8	SWRF	软件复位标志	发生软件复位时，由硬件置位。 通过置位CLRF清零。 0：未发生软件复位 1：发生软件复位	R/W
b7	PDRF	掉电模式复位	发生掉电模式复位时，由硬件置位。 通过写入CLRF清零。 0：未发生掉电模式复位 1：发生掉电模式复位	R/W
b6	SWDRF	专用看门狗复位标志	发生专用看门狗复位时，由硬件置位。 通过写入CLRF清零。 0：未发生专用看门狗复位 1：发生专用看门狗复位	R/W
b5	WDRF	看门狗复位标志	发生看门狗复位时，由硬件置位。 通过写入CLRF清零。 0：未发生看门狗复位 1：发生看门狗复位	R/W
b4	PVD2RF	可编程电压检测2复位标志	发生可编程电压检测2复位时，由硬件置位。 通过写入CLRF清零。 0：未发生可编程电压检测2复位 1：发生可编程电压检测2复位	R/W
b3	PVD1RF	可编程电压检测1复位标志	发生可编程电压检测1复位时，由硬件置位。 通过写入CLRF清零。 0：未发生可编程电压检测1复位 1：发生可编程电压检测1复位	R/W
b2	BORF	欠压复位标志	发生欠压复位时，由硬件置位。 通过写入CLRF清零。 0：未发生欠压复位 1：发生欠压复位	R/W
b1	PINRF	NRST引脚复位标志	发生引脚复位时，由硬件置位。 通过写入CLRF清零。 0：未发生NRST复位 1：发生NRST复位	R/W
b0	PORF	上电复位标志	发生上电复位时，由硬件置位。 通过写入CLRF清零。 0：未发生上电复位 1：发生上电复位	R/W

4 时钟控制器（CMU）

4.1 简介

时钟控制单元提供了一系列频率的时钟功能，包括：一个外部高速振荡器、一个外部低速振荡器、两个 PLL 时钟、一个内部高速振荡器、一个内部中速振荡器、一个内部低速振荡器、一个 SWDT 专用内部低速振荡器、时钟预分频器、时钟多路复用和时钟门控电路。

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

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

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

系统时钟的最大运行时钟频率可以达到 200MHz。SWDT 有独立的时钟源：SWDT 专用内部低速振荡器 (SWDTLRC)。实时时钟 (RTC) 使用外部低速振荡器或者内部低速振荡器作为时钟源。USB-FS 的 48MHz 时钟可以选择系统时钟，MPLL，UPLL 作为时钟源。

对于每一个时钟源，在未使用时都可以单独打开和关闭，以降低功耗。

4.2 系统框图

4.2.1 系统框图

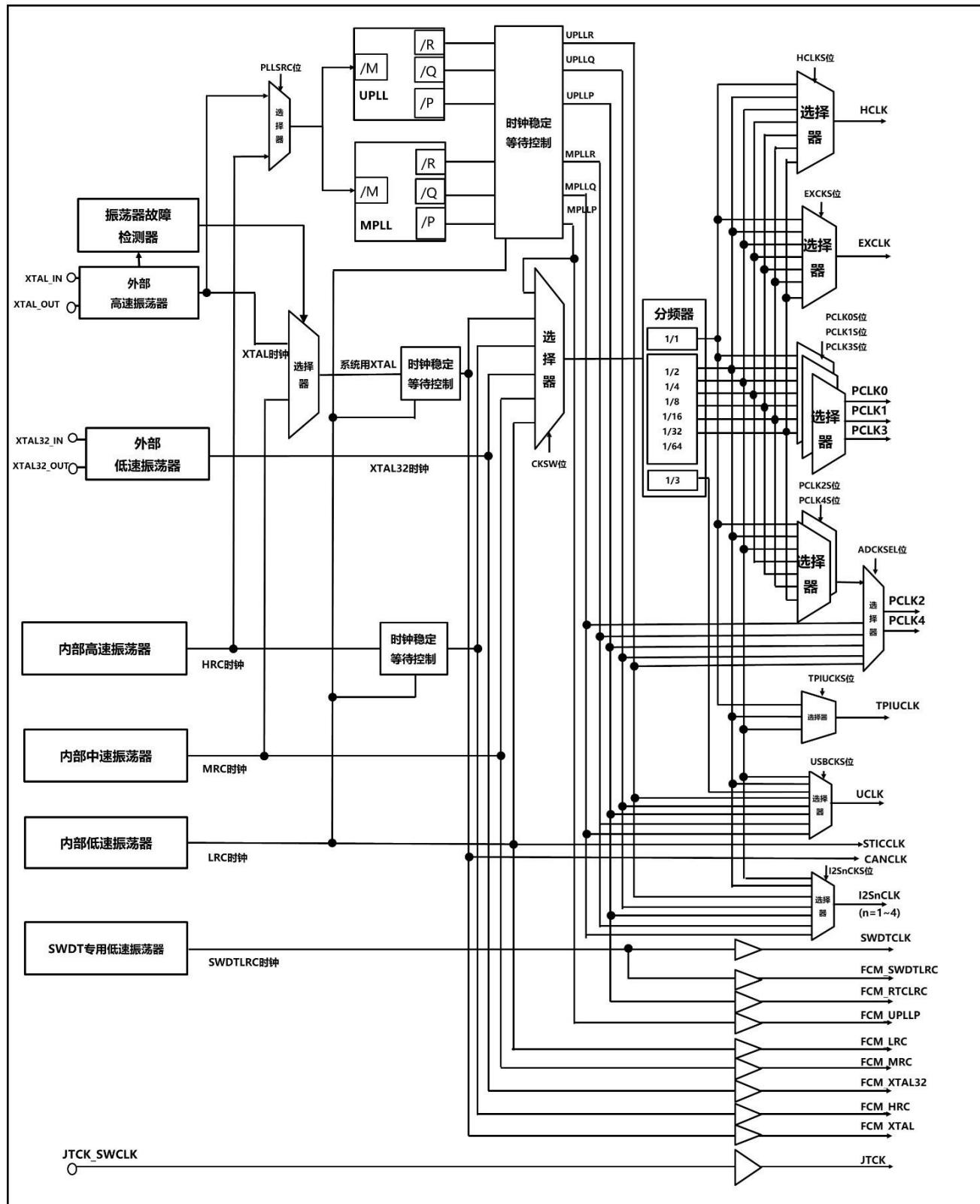


图 4-1 时钟系统框图

4.2.2 时钟频率测量框图

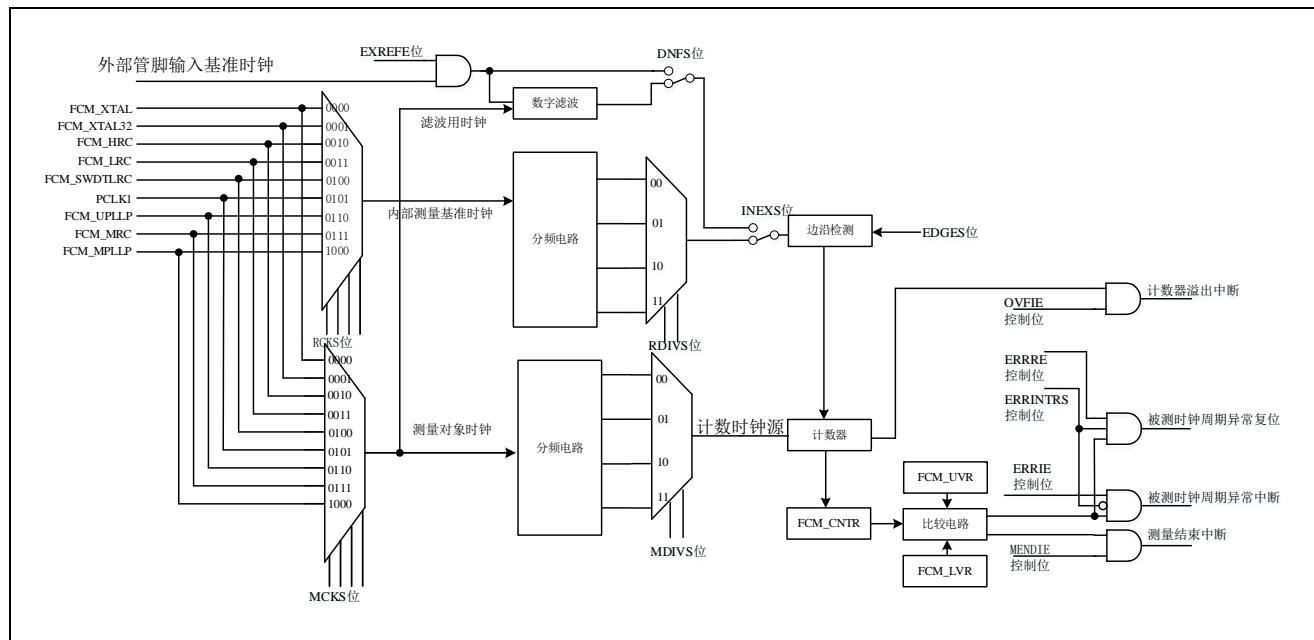


图 4-2 时钟频率测量框图

4.3 时钟源规格

各时钟源的主要特性如下表所示。

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
MPLL 时钟（MPLL）	输入时钟：外部高速振荡器或者内部高速振荡器 MPLL 输入时钟分频：1~24 任意分频可选 PFD 输入频率=输入时钟/MPLL 输入时钟分频，频率范围 1MHz~25MHz MPLL 倍频系数：20~480 倍 VCO 振荡频率：240MHz~480MHz MPLLQ 输出分频比： 2~16 任意分频 MPLLP 输出分频比： 2~16 任意分频 MPLLR 输出分频比： 2~16 任意分频 MPLLP 输出频率=（输入时钟/MPLL 输入时钟分频）*MPLL 倍频 系数/MPLLP 输出分频比 MPLLQ 输出频率=（输入时钟/MPLL 输入时钟分频）*MPLL 倍频 系数/MPLLQ 输出分频比 MPLLR 输出频率=（输入时钟/MPLL 输入时钟分频）*MPLL 倍频 系数/MPLLR 输出分频比
UPLL 时钟（UPLL）	输入时钟：外部高速振荡器或者内部高速振荡器 UPLL 输入时钟分频：1~24 任意分频可选 PFD 输入频率=输入时钟/UPLL 输入时钟分频，频率范围 1MHz~25MHz UPLL 倍频系数：20~480 倍 VCO 振荡频率：240MHz~480MHz UPLL P 输出分频比： 2~16 任意分频 UPLL Q 输出分频比： 2~16 任意分频

	<p>UPLL R 输出分频比：2~16 任意分频</p> <p>UPLL P 输出频率=（输入时钟/UPLL 输入时钟分频）*UPLL 倍频系数/UPLL P 输出分频比</p> <p>UPLL Q 输出频率=（输入时钟/UPLL 输入时钟分频）*UPLL 倍频系数/UPLL Q 输出分频比</p> <p>UPLL R 输出频率=（输入时钟/UPLL 输入时钟分频）*UPLL 倍频系数/UPLL R 输出分频比</p>
内部高速振荡器 (HRC)	<p>频率：16MHz 或者 20MHz</p> <p>用户可写寄存器对频率微调</p>
内部中速振荡器 (MRC)	<p>频率：8MHz</p> <p>用户可写寄存器对频率微调</p>
内部低速振荡器 (LRC)	<p>频率：32.768KHz</p> <p>用户可写寄存器对频率微调</p> <p>可用作 RTC 的计数时钟，唤醒定时器 WKT M 的计数时钟，XTAL32 的备份时钟</p>
SWDT 专用内部低速振荡器 (SWDTRC)	<p>频率：10KHz</p>

4.4 工作时钟规格

时钟	作用范围	规格
HCLK	CPU、DMA(n=1、2)、EFM(主闪存)、 SRAM1、SRAM2、SRAM3、SRAMH、 Ret-SRAM、MPU、GPIO、DCU、 INTC, QSPI	最高频率 200MHz 可选时钟源的分频: 1, 2, 4, 8, 16, 32, 64
PCLK0	Timer6 计数器用时钟	最高频率 200MHz 可选时钟源的分频: 1, 2, 4, 8, 16, 32, 64
PCLK1	USARTn (n=1~4)、SPIn(n=1~4)、 USBFS (控制逻辑)、Timer0n(n=1、2)、 TimerAn(n=1~6) 、 Timer4n(n=1~3)、 Timer6 (控制逻辑)、EMB、CRC、 HASH、AES、I2Sn(n=1~4)控制逻辑	最高频率 100MHz 可选时钟源的分频: 1, 2, 4, 8, 16, 32, 64
PCLK2	ADC 变换时钟	最高频率 60MHz 可选时钟源的分频: 1, 2, 4, 8, 16, 32, 64 可独立选择时钟源: UPLLP, UPLLQ, UPLL, MPLLP, MPLLQ, MPLLR
PCLK3	RTC (控制逻辑)、I2Cn(n=1、2、3)、 CMP、WDT、SWDT (控制逻辑)	最高频率 50MHz 可选时钟源的分频: 1, 2, 4, 8, 16, 32, 64
PCLK4	ADC (控制逻辑)、TRNG	最高频率 100MHz 可选时钟源的分频: 1, 2, 4, 8, 16, 32, 64 可独立选择时钟源: UPLLP, UPLLQ, UPLL, MPLLP, MPLLQ, MPLLR
EXCLK	SDION(n=1、2) 、 CAN	最高频率 100MHz 可选时钟源的分频: 1, 2, 4, 8, 16, 32, 64
UCLK	USBFS 通信用时钟	频率 48MHz 时钟源可选系统时钟分频 2, 3, 4。 可独立选择时钟源:

		UPLL, UPLLQ, UPLL, MPLL, MPLLQ, MPLLR
CANCLK	CAN 通信时钟	频率范围 4~25MHz
STICCLK	CPU 的 SysTickTimer 计数器用时钟, 时钟源为 LRC	可配置为时钟源 LRC 或系统时钟
SWDTCLK	SWDT 计数器用时钟	频率 10KHz
JTCK	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 可选时钟源的分频: 1, 2, 4
I2SnCLK (n=1~4)	I2Sn(n=1~4)	最高频率 200MHz 可独立选择时钟源: UPLL, UPLLQ, UPLL, MPLL, MPLLQ, MPLLR

表 4-1 各个内部时钟的规格

注意:

各时钟之间需遵守下列规则:

- HCLK 频率>=PCLK1 频率、HCLK 频率>=PCLK3 频率、HCLK 频率>=PCLK4 频率
- HCLK 频率:EXCLK 频率=2:1, 4:1, 8:1, 16:1, 32:1
- PCLK0 频率>=PCLK1 频率、PCLK0 频率>=PCLK3 频率
- HCLK 频率:PCLK0 频率=N:1, 1:N
- PCLK2 频率:PCK4 频率=1:4, 1:2, 1:1, 2:1, 4:1, 8:1

4.5 晶振电路

4.5.1 外部高速振荡器

4.5.1.1 振荡器模式

外部高速振荡器可为系统时钟提供更为精确时钟源。频率范围 4~25MHz。

XTAL 通过 CMU_XTALCR 的 XTALSTP 位打开和关闭。

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

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

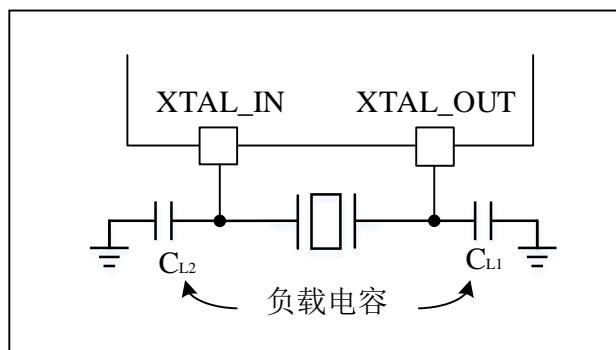


图 4-3 外部高速振荡器连接事例

4.5.1.2 时钟输入模式

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

外部时钟输入的连接例如下图所示。

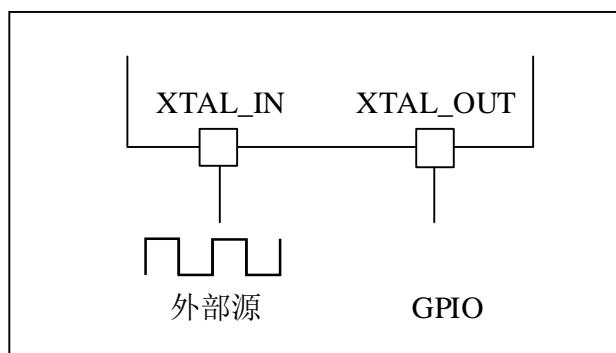


图 4-4 外部时钟输入的连接事例图

4.5.2 外部高速振荡器故障检测

振荡器故障检测是检测外部高速振荡器（XTAL）振荡是否正常。

通过寄存器 CMU_XTALSTDCR 的 XTALSTDE 位打开或关闭。

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

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

因为振荡器故障检测是检测外部因素导致的振荡器异常振荡，所以要通过软件使外部高速振荡器停止振荡或者转移到停止模式和掉电模式时，将振荡器振荡故障检测功能无效。如果外部高速振荡器发生故障，动作波形如下图所示。操作流程参照【检测到 XTAL 故障检测动作】。

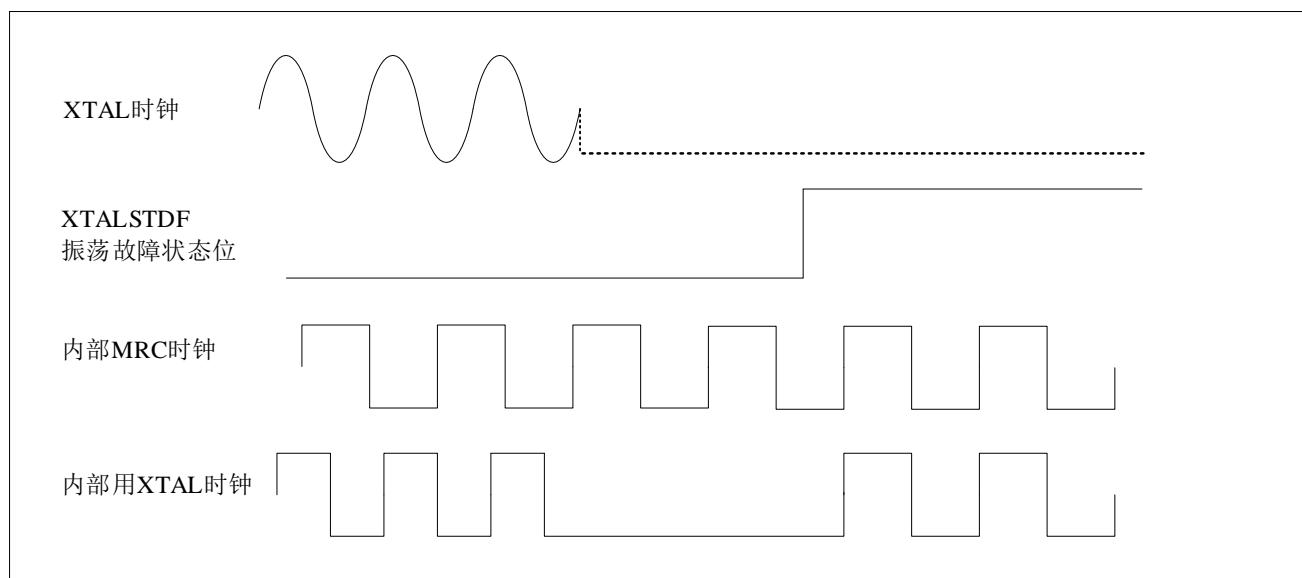


图 4-5 外部高速振荡器故障检测例

4.5.2.1 检测到 XTAL 故障检测动作

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

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

系统时钟选择成 XTAL，检测到 XTAL 故障时，动作例如下图所示。

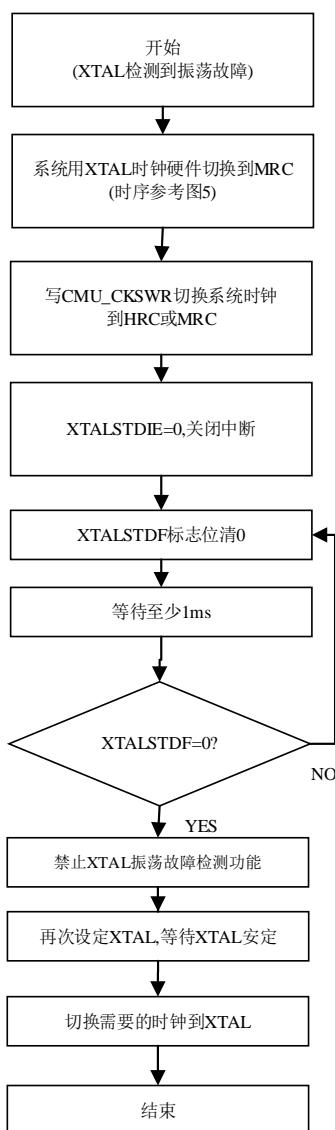


图 4-6 系统时钟选择 XTAL，检测到 XTAL 振荡故障例

4.5.2.2 检测到 XTAL 振荡故障产生中断复位

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

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

4.5.3 外部低速振荡器

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

XTAL32 通过 CMU_XTAL32CR 的 XTAL32STP 位打开和关闭。

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

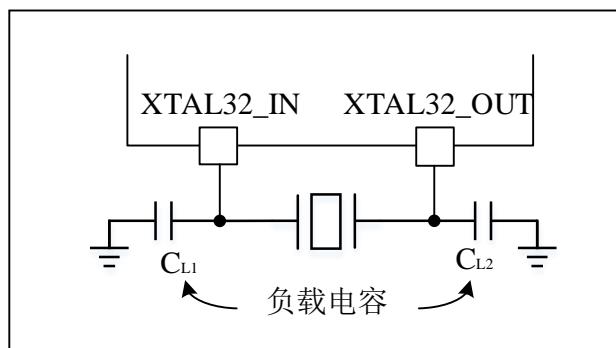


图 4-7 外部低速振荡器连接事例

XTAL32 初次上电的初始化流程如下所示：

1. CMU_XTAL32CR.XTAL32STP 位写 1，停止 XTAL32
2. 通过 CMU_XTAL32FGR 设定匹配的 XTAL32 驱动能力
3. 通过 CMU_XTAL32FGR 设定滤波功能
4. CMU_XTAL32CR.XTAL32STP 位写 0，XTAL32 振荡
5. 软件等待 XTAL32 稳定，稳定时间参考电器特性章节。

如果不使用外部低速振荡器，将 CMU_XTAL32CR 的 XTAL32STP 位设 1，关闭外部低速振荡器。

4.6 内部 RC 时钟

4.6.1 HRC 时钟

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

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

频率校准

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

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

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

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

4.6.2 MRC 时钟

MRC 时钟信号由内部 8MHz 中速振荡器生成，可直接用作系统时钟。

MRC 振荡器的优点是启动速度快，启动后不需要等稳定即可使用。

频率校准

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

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

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

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

4.6.3 LRC 时钟

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

LRC 振荡器的启动速度快，启动后不需要等稳定即可使用。

频率校准

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

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

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

4.6.4 SWDTRC 时钟

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

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

4.7 PLL 时钟

HC32F46xx 器件具有两个 PLL:

- MPLL 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 USBFS、TRNG、ADC 和 I2S 时钟。
- UPLL 三个输出亦可用于生成 USBFS、TRNG、ADC 和 I2S 时钟。

UPLL 使用与 MPLL 相同的输入时钟源，可选择 HRC 或 XTAL 振荡器作为时钟源，由 CMU_PLLCFGR.PLLSRC 位配置。在 HRC 或 XTAL 振荡器稳定后，再对 PLL 进行配置。

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

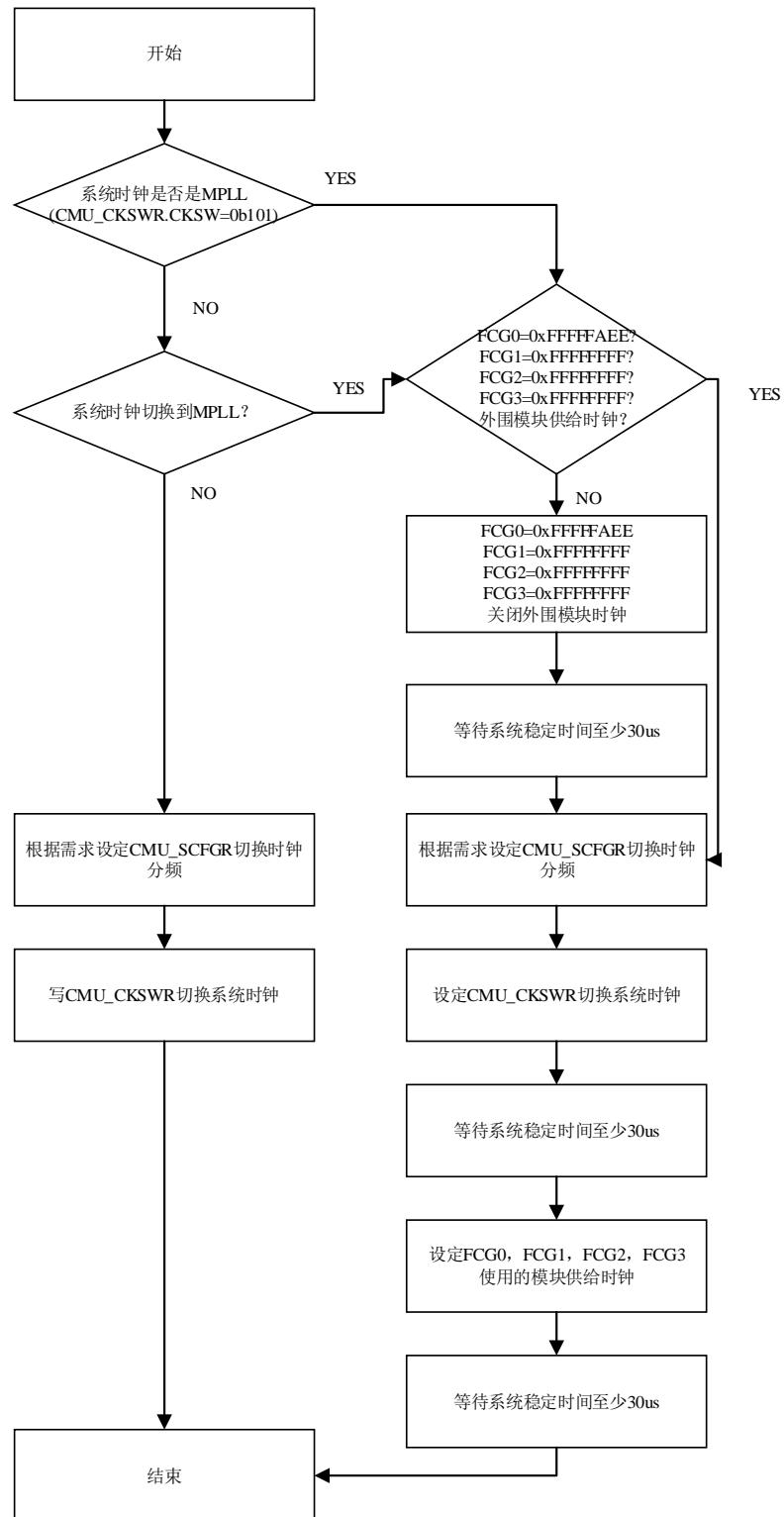
当进入掉电和停止模式后，两个 PLL 将由硬件禁止。

4.8 时钟切换步骤

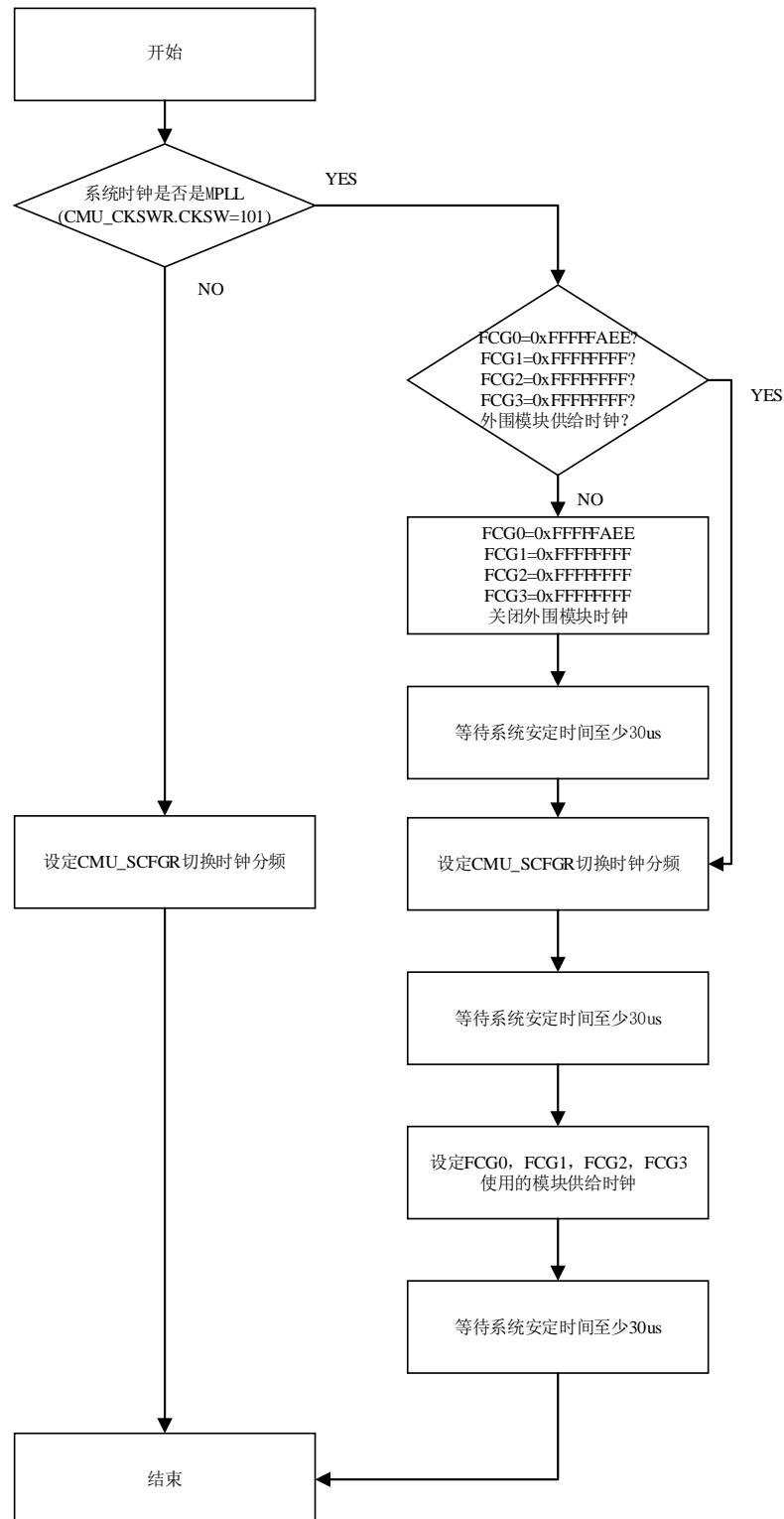
在系统复位后，默认系统时钟为 MRC。通过设定寄存器 CMU_CKSW 切换时钟源，切换步骤参照时钟源切换。只有在目标时钟源已稳定的状态下，才可以从一个时钟源切换到另一个时钟源。

时钟切换时需要正确配置 FLASH/SRAM 的等待周期，防止系统时钟频率大于 FLASH/SRAM 的最大动作频率。参照【CPU 时钟和 FLASH 读取时间之间的关系】、【内置 SRAM (SRAM)】配置。

4.8.1 时钟源切换



4.8.2 时钟分频切换



4.9 时钟输出功能

共有两路时钟输出：

- **MCO_1**

用户可通过可配置的预分配器（从 1 到 128）向 MCO_1 引脚输出不同的时钟源：

- HRC 时钟
- MRC 时钟
- LRC 时钟
- XTAL 时钟
- XTAL32 时钟
- MPPLL/MPLLQ 时钟
- UPLL/PULLQ 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（从 1 到 128）向 MCO_2 引脚输出不同的时钟源：

- HRC 时钟
- MRC 时钟
- LRC 时钟
- XTAL 时钟
- XTAL32 时钟
- MPPLL/MPLLQ 时钟
- UPLL/PULLQ 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 100 MHz（最大 I/O 速度）。

4.10 时钟频率测量

4.10.1 时钟频率测量

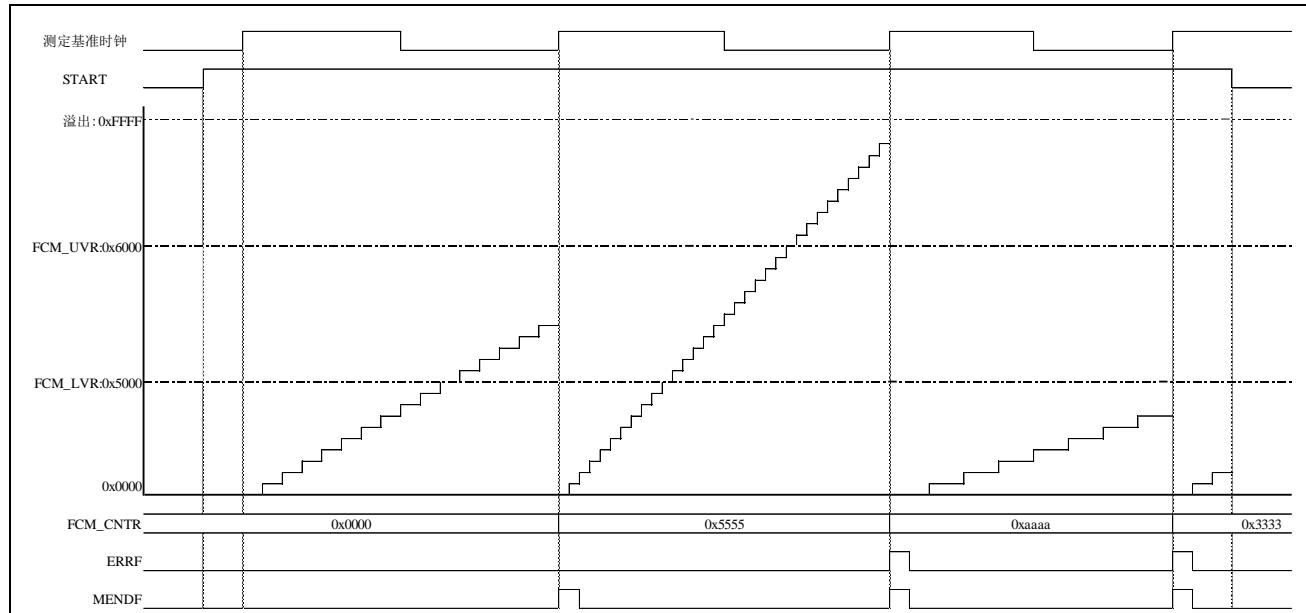


图 4-8 时钟频率测量时序图

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

4.10.2 数字滤波功能

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

数字滤波功能可以设定数字滤波功能有效无效以及采样时钟。

4.10.3 中断/复位功能

时钟频率测量电路有三种中断请求。分别是：

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

时钟频率测量电路有一种复位请求：

- 1) 频率异常复位

4.11 寄存器说明

基准地址 1: 0x4004_8400

寄存器名	符号	偏移地址	位宽	复位值
FCM下限比较值寄存器	FCM_LVR	0x00	32	0x0000_0000
FCM上限比较值寄存器	FCM_UVR	0x04	32	0x0000_0000
FCM计数器值寄存器	FCM_CNTR	0x08	32	0x0000_0000
FCM开始停止寄存器	FCM_STR	0x0C	32	0x0000_0000
FCM测量对象控制寄存器	FCM_MCCR	0x10	32	0x0000_0000
FCM测量基准控制寄存器	FCM_RCCR	0x14	32	0x0000_0000
FCM中断复位控制寄存器	FCM_RIER	0x18	32	0x0000_0000
FCM标志寄存器	FCM_SR	0x1C	32	0x0000_0000
FCM标志位清除寄存器	FCM_CLR	0x20	32	0x0000_0000

基准地址 2: 0x40054000

寄存器名	符号	偏移地址	位宽	复位值
CMU_XTAL配置寄存器	CMU_XTALCFG	0x410	8	0x80
CMU_XTAL安定配置寄存器	CMU_XTALSTBCR	0x0A2	8	0x05
CMU_XTAL控制寄存器	CMU_XTALCR	0x032	8	0x01
CMU_XTAL振荡故障控制寄存器	CMU_XTALSTDRCR	0x040	8	0x00
CMU_XTAL振荡故障状态寄存器	CMU_XTALSTDSR	0x041	8	0x00
CMU_HRC校准寄存器	CMU_HRCTRM	0x062	8	0x00
CMU_HRC控制寄存器	CMU_HRCCR	0x036	8	由ICG1.HRCSTP 值决定
CMU_MRC校准寄存器	CMU_MRCTR	0x061	8	0x00
CMU_MRC控制寄存器	CMU_MRCCR	0x038	8	0x80
CMU_MPLL配置寄存器	CMU_PLLCFG	0x100	32	0x1110_1300
CMU_MPLL控制寄存器	CMU_PLLCR	0x02A	8	0x01
CMU_UPLL配置寄存器	CMU_UPLLCFG	0x104	32	0x1110_1300
CMU_UPLL控制寄存器	CMU_UPLLCR	0x02E	8	0x01
CMU_时钟源安定状态寄存器	CMU_OSCSTBSR	0x03C	8	0x00
CMU_系统时钟源切换寄存器	CMU_CKSWR	0x026	8	0x01
CMU_时钟分频配置寄存器	CMU_SCFGR	0x020	32	0x0000_0000
CMU_USBFS时钟配置寄存器	CMU_UFSCKCFG	0x024	8	0x40
CMU_AD/TRNG时钟配置寄存器	CMU_PERICKSEL	0x010	16	0x0000
CMU_I2S时钟配置寄存器	CMU_I2SCKSEL	0x012	16	0xBBB
CMU_调试时钟配置寄存器	CMU_TPIUCKCFG	0x03F	8	0x00
CMU_MCO1时钟输出配置寄存器	CMU_MCO1CFG	0x03D	8	0x00
CMU_MCO2时钟输出配置寄存器	CMU_MCO2CFG	0x03E	8	0x00
CMU_XTAL32控制寄存器	CMU_XTAL32CR	0x420	8	0x00
CMU_XTAL32配置寄存器	CMU_XTAL32CFG	0x421	8	0x00
CMU_XTAL32滤波寄存器	CMU_XTAL32NFR	0x425	8	0x00
CMU_LRC控制寄存器	CMU_LRCCR	0x427	8	0x00
CMU_LRC校准寄存器	CMU_LRCTR	0x429	8	0x00

4.11.1 CMU XTAL 配置寄存器(CMU_XTALCFG)R

复位值: 0x80

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

位	标记	位名	功能	读写
b7	SUPDRV	XTAL超高速驱动允许	0: 禁止超高速驱动 1: 允许超高速驱动 超高速驱动允许时, XTAL稳定后, 无视此位设定, 禁止超高速驱动, 降低功耗。	R/W
b6	XTALMS	XTAL模式选择位	0: 振荡器模式 1: 外部时钟输入模式	R/W
b5~b4	XTALDRV[1:0]	XTAL驱动能力选择	00: 高驱动能力(建议20~25MHz晶振) 01: 中驱动能力(建议16~20MHz晶振) 10: 小驱动能力(建议8~16MHz晶振) 11: 超小驱动能力(建议4~8MHz晶振)	R/W
b3~b0	Reserved	-	读出时为“0”, 写入时写“0”	R/W

4.11.2 CMU XTAL 安定配置寄存器(CMU_XTALSTBCR)R

复位值: 0x05

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

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

4.11.3 CMU XTAL 控制寄存器(CMU_XTALCR)

复位值: 0x01

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

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

注意:

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

4.11.4 CMU XTAL 振荡故障控制寄存器(CMU_XTALSTDRCR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
XTALSTDE	-	-	-	-	XTALSTDRI	XTALSTDRE	XTALSTDIE

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

注意:

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

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

复位值: 0x00

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

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

4.11.6 CMU XTAL32 配置寄存器(CMU_XTAL32CFGR)

复位值: 0x00

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

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

4.11.7 CMU XTAL32 滤波寄存器(CMU_XTAL32NFR)

复位值: 0x00

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

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

4.11.8 CMU XTAL32 控制寄存器(CMU_XTAL32CR)

复位值: 0x00

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

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

注意:

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

4.11.9 CMU HRC 校准寄存器(CMU_HRCTRM)

复位值: 0x00

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

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

4.11.10 CMU HRC 控制寄存器(CMU_HRCCR)

复位值：由 ICG1.HRCSTP 值决定

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

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

注意：

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

4.11.11 CMU MRC 校准寄存器(CMU_MRCTRIM)

复位值: 0x00

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

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

注意:

- 频率校准需在 MRC 频率保证范围内。

4.11.12 CMU MRC 控制寄存器(CMU_MRCCR)

复位值: 0x80

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

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

注意:

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

4.11.13 CMU LRC 校准寄存器(CMU_LRCTRM)

复位值: 0x00

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

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

注意:

- 频率校准需在 LRC 频率保证范围内。

4.11.14 CMU LRC 控制寄存器(CMU_LRCCR)

复位值: 0x00

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

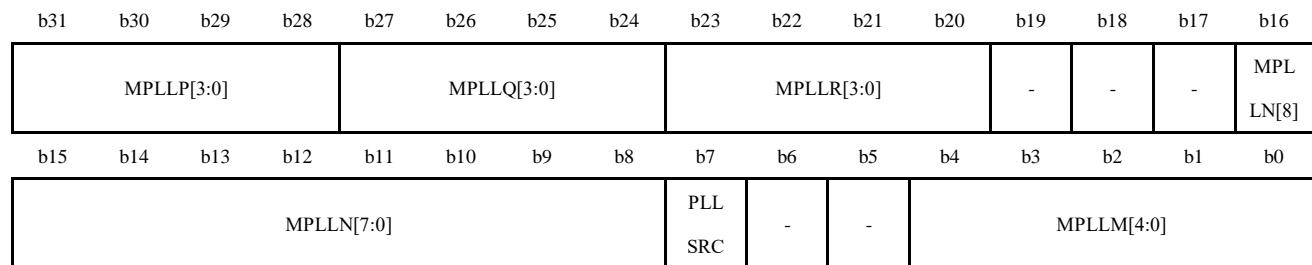
位	标记	位名	功能	读写
b31~b1	Reserved	-	读出时为“0”，写入时写“0”	R/W
b0	LCRSTP	LRC振荡器开启停止 位	0: LRC振荡器振荡 1: LRC振荡器停止	R/W

注意:

- LRC 选作系统时钟源时，禁止 LCRSTP 写“1”停止 LRC 时钟。
- 软件设定 LRC 动作开始，等待 5 个 LRC 周期后，才可以进入停止模式、掉电模式或者停止 LRC。
- 软件设定 LRC 停止，等待 5 个 LRC 周期后，才可以进入停止模式、掉电模式或者再次启动 LRC。
- 等待 XTAL 振荡器、HRC、MPLL、UPLL 时钟稳定时，LCRSTP 位设定无视，LRC 强制振荡。
- RTC 选择 LRC 做时钟源时，LRC 无视此寄存器位，LRC 振荡。RTC 未初始化时，LRC 有振荡的可能。

4.11.15 CMU MPLL 配置寄存器(CMU_PLLCFGR)

复位值: 0x1110_1300



位	标记	位名	功能	读写
b31-b28	MPLL P[3:0]	系统时钟用MPLL分频系数	用于MPLL P时钟的频率，在MPLL停止条件下写MPLL P。 MPLL输出时钟频率=MPLL的VCO频率/MPLL P 0000: 禁止设定 0001: 2分频 0010: 3分频 0011: 4分频 1101: 14分频 1110: 15分频 1111: 16分频	R/W
b27-b24	MPLL Q[3:0]	系统时钟用MPLL分频系数	用于MPLL Q时钟的频率，在MPLL停止条件下写MPLL Q。 MPLL输出时钟频率=MPLL的VCO频率/MPLL Q 0000: 禁止设定 0001: 2分频 0010: 3分频 0011: 4分频 1101: 14分频 1110: 15分频 1111: 16分频	R/W
b23-b20	MPLL R[3:0]	系统时钟用MPLL分频系数	用于MPLL R时钟的频率，在MPLL停止条件下写MPLL R。 MPLL输出时钟频率=MPLL的VCO频率/MPLL R 0000: 禁止设定 0001: 2分频 0010: 3分频 0011: 4分频 1101: 14分频 1110: 15分频 1111: 16分频	R/W

b19-b17	-	-	读出时为“0”，写入时写“0”	R/W
b16-b8	MPLLN[8:0]	MPLL倍频系数	用于控制MPLL的VCO的倍频系数，在MPLL停止条件下写MPLLN。确保MPLL的VCO频率介于240MHz到480MHz之间。 MPLL的VCO频率=MPLL的PFD输入频率 * MPLLN 000010011: 20 000010100: 21 000010101: 22 000010110: 23 111011101: 478 111011110: 479 111011111: 480	R/W
b7	PLLSRC	MPLL/UPLL输入时钟源选择	0: 选择外部高速振荡器作为MPLL/UPLL的输入时钟 1: 选择内部高速振荡器作为MPLL/UPLL的输入时钟	R/W
b6-b5	-	-	读出时为“0”，写入时写“0”	R/W
b4-b0	MPLL M[4:0]	MPLL输入时钟分频系数	用于在MPLL的VCO之前对MPLL输入时钟进行分频。在MPLL停止条件下写MPLL M。确保MPLL的PFD输入频率介于1MHz到25MHz之间。 00000: 1分频 00001: 2分频 00010: 3分频 10111: 24分频 其他禁止	R/W

4.11.16 CMU MPLL 控制存器(CMU_PLLCR)

复位值: 0x01

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

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

注意:

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

4.11.17 CMU UPLL 配置寄存器(CMU_UPLLCFGR)

复位值: 0x1110_1300

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
UPLL[3:0]		UPLLQ[3:0]		UPLL[3:0]		UPLL[3:0]		UPLL[3:0]		-		-		-		UPL LN[8]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
UPLL[7:0]		-		-		-		UPLL[4:0]								

位	标记	位名	功能	读写
b31-b28	UPLL[3:0]	系统时钟用UPLL分频系数	用于控制UPLL时钟的频率，在UPLL停止条件下写UPLL。 UPLL输出时钟频率=UPLL的VCO频率/UPLL 0000: 禁止设定 0001: 2分频 0010: 3分频 0011: 4分频 1101: 14分频 1110: 15分频 1111: 16分频	R/W
b27-b24	UPLLQ[3:0]	系统时钟用UPLL分频系数	用于控制UPLLQ时钟的频率，在UPLL停止条件下写UPLLQ。 UPLL输出时钟频率=UPLL的VCO频率/UPLLQ 0000: 禁止设定 0001: 2分频 0010: 3分频 0011: 4分频 1101: 14分频 1110: 15分频 1111: 16分频	R/W
b23-b20	UPLL[3:0]	系统时钟用UPLL分频系数	用于控制UPLL时钟的频率，在UPLL停止条件下写UPLL。 UPLL输出时钟频率=UPLL的VCO频率/UPLL 0000: 禁止设定 0001: 2分频 0010: 3分频 0011: 4分频 1101: 14分频 1110: 15分频 1111: 16分频	R/W
b19-b17	-	-	读出时为“0”，写入时写“0”	R/W

b16-b8	UPLL[N][8:0]	UPLL倍频系数	用于控制UPLL的VCO的倍频系数，在UPLL停止条件下写UPLL[N]。确保UPLL的VCO频率介于240MHz到480MHz之间。 UPLL的VCO频率=UPLL的PFD输入频率 * UPLL[N] 000010011: 20 000010100: 21 000010101: 22 000010110: 23 111011101: 478 111011110: 479 111011111: 480 其他禁止设定	R/W
b7-b5	-	-	读出时为“0”，写入时写“0”	R/W
b4-b0	UPLLM[4:0]	UPLL输入时钟分频系数	用于在UPLL的VCO之前对UPLL输入时钟进行分频。在UPLL停止条件下写UPLLM。确保UPLL的PFD输入频率介于1MHz到25MHz之间。 00000: 禁止设定 00001: 2分频 00010: 3分频 10111: 24分频	R/W

4.11.18 CMU UPLL 控制存器(CMU_UPLLCSR)

复位值: 0x01

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

位	标记	位名	功能	读写
b7~b1	Reserved	-	读出时为“0”，写入时写“0”	R/W
b0	UPLLOFF	UPLL使能	用于开始停止UPLL。 0: UPLL动作开始 1: UPLL停止	R/W

注意:

- UPLL 选作 I2S/TRNG/ADC/USBFS 的时钟源时，禁止 UPLLOFF 写“1”停止 UPLL 时钟。
- 软件设定 UPLL 动作开始，通过 UPLLSTBF 位确认 UPLL 稳定后，才可以进入停止模式，掉电模式或者软件设定停止 UPLL。
- 软件设定 UPLL 停止，通过 UPLLSTBF 位确认 UPLL 停止后，才可以进入停止模式，掉电模式或者再次启动 UPLL。
- UPLL 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 UPLL 动作开始。UPLL 选择 HRC 作时钟源时，通过 HRCSTBF 位确认 HRC 稳定后，才可以设定 UPLL 动作开始。

4.11.19 CMU 时钟源稳定状态器(CMU_OSCSTBSR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	UPLLSTBF	MPLLSTBF	-	XTALSTBF	-	-	HRCSTBF

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

4.11.20 CMU 系统时钟源切换寄存器(CMU_CKSWR)

复位值: 0x01

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

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

注:

1, 切换的目标时钟源, 需保证在时钟稳定状态。
 2, 流程参照【时钟源切换】章节
 3, PWC_STPMCR.CKSMRC位为1时, 停止模式唤醒后, 此寄存器
 初始化, 系统时钟源选择MRC时钟。

4.11.21 CMU 时钟分频配置寄存器(CMU_SCFGR)

复位值: 0x0000_0000

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

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

器初始化， PCLK4为系统时钟的1分频。				
b15	Reserved	-	读出时为“0”，写入时写“0”	R/W
b14~12	PCLK3S[2:0]	PCLK3时钟分频选择位	000: 系统时钟的1分频 001: 系统时钟的2分频 010: 系统时钟的4分频 011: 系统时钟的8分频 100: 系统时钟的16分频 101: 系统时钟的32分频 110: 系统时钟的64分频 111: 禁止设定	R/W
			注：PWC_STPMCR.CKSMRC位为1时，停止模式唤醒后，此寄存器初始化，PCLK3为系统时钟的1分频。	
b11	Reserved	-	读出时为“0”，写入时写“0”	R/W
b10~8	PCLK2S[2:0]	PCLK2时钟分频选择位	000: 系统时钟的1分频 001: 系统时钟的2分频 010: 系统时钟的4分频 011: 系统时钟的8分频 100: 系统时钟的16分频 101: 系统时钟的32分频 110: 系统时钟的64分频 111: 禁止设定	R/W
			注：PWC_STPMCR.CKSMRC位为1时，停止模式唤醒后，此寄存器初始化，PCLK2为系统时钟的1分频。	
b7	Reserved	-	读出时为“0”，写入时写“0”	R/W
b6~4	PCLK1S[2:0]	PCLK1时钟分频选择位	000: 系统时钟的1分频 001: 系统时钟的2分频 010: 系统时钟的4分频 011: 系统时钟的8分频 100: 系统时钟的16分频 101: 系统时钟的32分频 110: 系统时钟的64分频 111: 禁止设定	R/W
			注：PWC_STPMCR.CKSMRC位为1时，停止模式唤醒后，此寄存器初始化，PCLK1为系统时钟的1分频。	
b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2~0	PCLK0S[2:0]	PCLK0时钟分频选择位	000: 系统时钟的1分频 001: 系统时钟的2分频 010: 系统时钟的4分频 011: 系统时钟的8分频 100: 系统时钟的16分频 101: 系统时钟的32分频 110: 系统时钟的64分频	R/W

111: 禁止设定

注: PWC_STPMCR.CKSMRC位为1时, 停止模式唤醒后, 此寄存器初始化, PCLK0为系统时钟的1分频。

4.11.22 CMU USBFS 时钟配置存器(CMU_USBCKCFGR)

复位值: 0x40

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

位	标记	位名	功能	读写
b7~b4	USBCKS[3:0]	USB-FS的48MHz时钟源选择	0010: 系统时钟2分频 0011: 系统时钟3分频 0100: 系统时钟4分频 1000: MPLL/P 1001: MPLL/Q 1010: MPLL/R 1011: UPLL/P 1100: UPLL/Q 1101: UPLL/R 其他禁止设定 注: 1. 切换的目标时钟源为MPLL/UPLL时, 需保证在MPLL/UPLL时钟处于稳定状态。 2. 系统时钟选择MPLL时, 需要将USB, CAN, QSPI, SPI, 通用定时器, FCM, ADC, DAC設定成模块停止状态, 再写CMU_SCFGR寄存器切换时钟分频。写完CMU_USBCKCFGR寄存器后, 软件等待系统稳定时间30us。 3. PWC_STPMCR.CKSMRC位为1时, 停止模式唤醒后, 此寄存器初始化, USBCLK为系统时钟的4分频。	R/W
b3~b0	Reserved	-	读出时为“0”, 写入时写“0”	R/W

4.11.23 CMU AD/TRNG 时钟配置存器(CMU_PERICKSEL)

复位值: 0x0000

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

位	标记	位名	功能	读写
b15~b4	Reserved	-	读出时为“0”，写入时写“0”	R/W
b3~b0	PERICKSEL[3:0]	AD/TRNG时钟源选择	0000: CMU_SCFG设的PCLK2/PCLK4 1000: MPLL/P 1001: MPLL/Q 1010: MPLL/R 1011: UPLL/P 1100: UPLL/Q 1101: UPLL/R 除此以外禁止设定。	R/W

注意:

- 切换的目标时钟源为 MPLL/UPLL 时，需保证在 MPLL/UPLL 时钟稳定状态。

4.11.24 CMU I2S 时钟配置存器(CMU_I2SCKSEL)

复位值: 0xB BBBB

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
I2S4CKSELc				I2S3CKSEL[3:0]				I2S2CKSEL[3:0]				I2S1CKSEL[3:0]			

位	标记	位名	功能	读写
b15~b12	I2S4CKSEL	I2S时钟源选择	0000: CMU_SCFG设定了PCLK3 1000: MPLL/P 1001: MPLL/Q 1010: MPLL/R 1011: UPLL/P 1100: UPLL/Q 1101: UPLL/R 除此以外设定禁止。 注: 上记以外设定无时钟	R/W
b11~b8	I2S3CKSEL	I2S时钟源选择	0000: CMU_SCFG设定了PCLK3 1000: MPLL/P 1001: MPLL/Q 1010: MPLL/R 1011: UPLL/P 1100: UPLL/Q 1101: UPLL/R 除此以外设定禁止。 注: 上记以外设定无时钟	R/W
b7~b4	I2S2CKSEL	I2S时钟源选择	0000: CMU_SCFG设定了PCLK3 1000: MPLL/P 1001: MPLL/Q 1010: MPLL/R 1011: UPLL/P 1100: UPLL/Q 1101: UPLL/R 除此以外设定禁止。 注: 上记以外设定无时钟	R/W
b3~b0	I2S1CKSEL	I2S时钟源选择	0000: CMU_SCFG设定了PCLK3 1000: MPLL/P 1001: MPLL/Q 1010: MPLL/R 1011: UPLL/P 1100: UPLL/Q 1101: UPLL/R	R/W

除此以外设定禁止。

注：上记以外设定无时钟

注意：

- 切换的目标时钟源为 MPPLL/UPLL 时，需保证在 MPPLL/UPLL 发振稳定状态。
- 目标时钟源选择 MPPLL/UPLL 时，配置 MPPLL/UPLL 的详细方法参考时钟控制器（CMU）。

4.11.25 CMU 调试时钟配置寄存器(CMU_TPIUCKCFGR)

复位值：0x00

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

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

4.11.26 CMU MCO1 配置存储器(CMU_MCO1CFGR)

复位值: 0x00

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

位	标记	位名	功能	读写
b7	MCO1EN	MCO_1输出许可	0: 禁止MCO_1输出 1: 允许MCO_1输出	R/W
b6~b4	MCO1DIV[2:0]	MCO_1分频选择	000: 1分频 001: 2分频 010: 4分频 011: 8分频 100: 16分频 101: 32分频 110: 64分频 111: 128分频	R/W
b3~b0	MCO1SEL[3:0]	MCO_1时钟源选择	0000: HRC时钟 0001: MRC时钟 0010: LRC时钟 0011: XTAL时钟 0100: XTAL32时钟 0110: MPLLP 0111: UPLL 1000: MPLLQ 1001: UPLLQ 1011: 系统时钟 其他禁止设定。	R/W

4.11.27 CMU MCO2 配置存器(CMU_MCO2CFGR)

复位值: 0x00

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

位	标记	位名	功能	读写
b7	MCO2EN	MCO_2输出许可	0: 禁止MCO_2输出 1: 允许MCO_2输出	R/W
b6~b4	MCO2DIV[2:0]	MCO_2分频选择	000: 1分频 001: 2分频 010: 4分频 011: 8分频 100: 16分频 101: 32分频 110: 64分频 111: 128分频	R/W
b3~b0	MCO2SEL[3:0]	MCO_2时钟源选择	0000: HRC时钟 0001: MRC时钟 0010: LRC时钟 0011: XTAL时钟 0100: XTAL32时钟 0110: MPLLP 0111: UPLL 1000: MPLLQ 1001: UPLLQ 1011: 系统时钟 其他禁止设定。	R/W

4.11.28 FCM 下限比较值寄存器(FCM_LVR)

复位值: 0x0000_0000

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

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

4.11.29 FCM 上限比较值寄存器(FCM_UVR)

复位值: 0x0000_0000

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

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

4.11.30 FCM 计数器值寄存器(FCM_CNTR)

复位值: 0x0000_0000

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

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

4.11.31 FCM 开始停止寄存器(FCM_STR)

复位值: 0x0000_0000

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

-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STA
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	-----

-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RT
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----

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

4.11.32 FCM 测量对象控制寄存器(FCM_MCCR)

复位值: 0x0000_0000

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

-	-	-	-	-	-	-	-	MCKS[3:0]	-	-	MDIVS[1:0]				
---	---	---	---	---	---	---	---	-----------	---	---	------------	--	--	--	--

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”，写入时写“0”	R/W
b7~b4	MCKS[3:0]	测量对象时钟选择位	0000: XTAL 0001: XTAL32 0010: HRC 0011: LRC 0100: SWDTRC 0101: PCLK1 0110: UPLL 0111: MRC 1000: MPLLP 1001: RTCLRC 其他: 设定禁止	R/W
b3~b2	Reserved	-	读出时为“0”，写入时写“0”	R/W
b1~b0	MDIVS[1:0]	测量对象分频选择	00: 不分频 01: 4分频 10: 8分频 11: 32分频	R/W

4.11.33 FCM 测量基准控制寄存器(FCM_RCCR)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”，写入时写“0”	R/W
b15	EXREFE	外部管脚输入参考时钟FCMREF允许位	0: 禁止外部管脚输入参考时钟FCMREF 1: 允许外部管脚输入参考时钟FCMREF	R/W
b14	Reserved	-	读出时为“0”，写入时写“0”	R/W
b13~b12	EDGES[1:0]	测量基准边沿选择位	00: 上升沿 01: 下降沿 10: 上升和下降沿 11: 禁止设定	R/W
b11~b10	Reserved	-	读出时为“0”，写入时写“0”	R/W
b9~b8	DNFS[1:0]	数字滤波器功能选择位	00: 无滤波功能 01: MCKS位选择的时钟作为滤波时钟 10: MCKS位选择的时钟的4分频作为滤波时钟 11: MCKS位选择的时钟的16分频作为滤波时钟	R/W
b7	INEXS	测量基准，内部时钟和端子选择位	0: 外部管脚输入参考时钟FCMREF 1: RCKS选择位选择的时钟	R/W
b6~b3	RCKS[3:0]	测量基准时钟选择位	0000: XTAL 0001: XTAL32 0010: HRC 0011: LRC 0100: SWDTRC 0101: PCLK1 0110: UPLL 0111: MRC 1000: MPLLP 1001: RTCLRC 其他: 设定禁止	R/W
b2	Reserved	-	读出时为“0”，写入时写“0”	R/W
b1~b0	RDIVS[1:0]	测量基准分频选择	00: 32分频 01: 128分频 10: 1024分频 11: 8192分频	R/W

4.11.34 FCM 中断复位控制寄存器(FCM_RIER)

复位值: 0x0000_0000

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

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

4.11.35 FCM 标志寄存器(FCM_SR)

复位值: 0x0000_0000

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

-	-	-	-	-	-	-	-	-	-	-	-	-	OVF	MEN DF	ERR F
---	---	---	---	---	---	---	---	---	---	---	---	---	-----	--------	-------

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

4.11.36 FCM 标志位清除寄存器(FCM_CLR)

复位值: 0x0000_0000

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

-	-	-	-	-	-	-	-	-	-	-	-	-	OVF CLR	MEN DFC LR	ERR FCLR
---	---	---	---	---	---	---	---	---	---	---	---	---	---------	------------	----------

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

5 电源控制 (PWC)

5.1 简介

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

芯片的工作电压(VCC)为 1.8V 到 3.6V。电压调节器(LDO)为 VDD 域和 VDDR 域供电，VDDR 电压调压器(RLDO)在掉电模式时为 VDDR 域供电。芯片通过功耗控制逻辑(PWCL)提供了超高速、高速、超低速等三种运行模式，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后可以通过 RLDO 维持电源，保证实时时钟模块(RTC)、唤醒定时器(WKTM)能够继续动作，保持 4KB 的低功耗 SRAM(Ret-SRAM)的数据。模拟模块配备了专用供电引脚，提高了模拟性能。

5.2 电源分布

图 5-1 是芯片的电源分布图。芯片由 VCC 域、VDD 电源域、AVCC 电源域、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)、VDDR 域调压器(RLDO)和振荡器电路构成。振荡器电路包括外部高速振荡器(XTAL)、外部低速振荡器(XTAL32)、内部低速振荡器(LRC)等。

VDD 域由 CPU、数字外设等数字逻辑、RAM、FLASH 等构成，通过 LDO 产生的 VDD 供电。在 VDD 域中 RAM 被分为 4 组，可以通过寄存器控制独立断电。

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

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

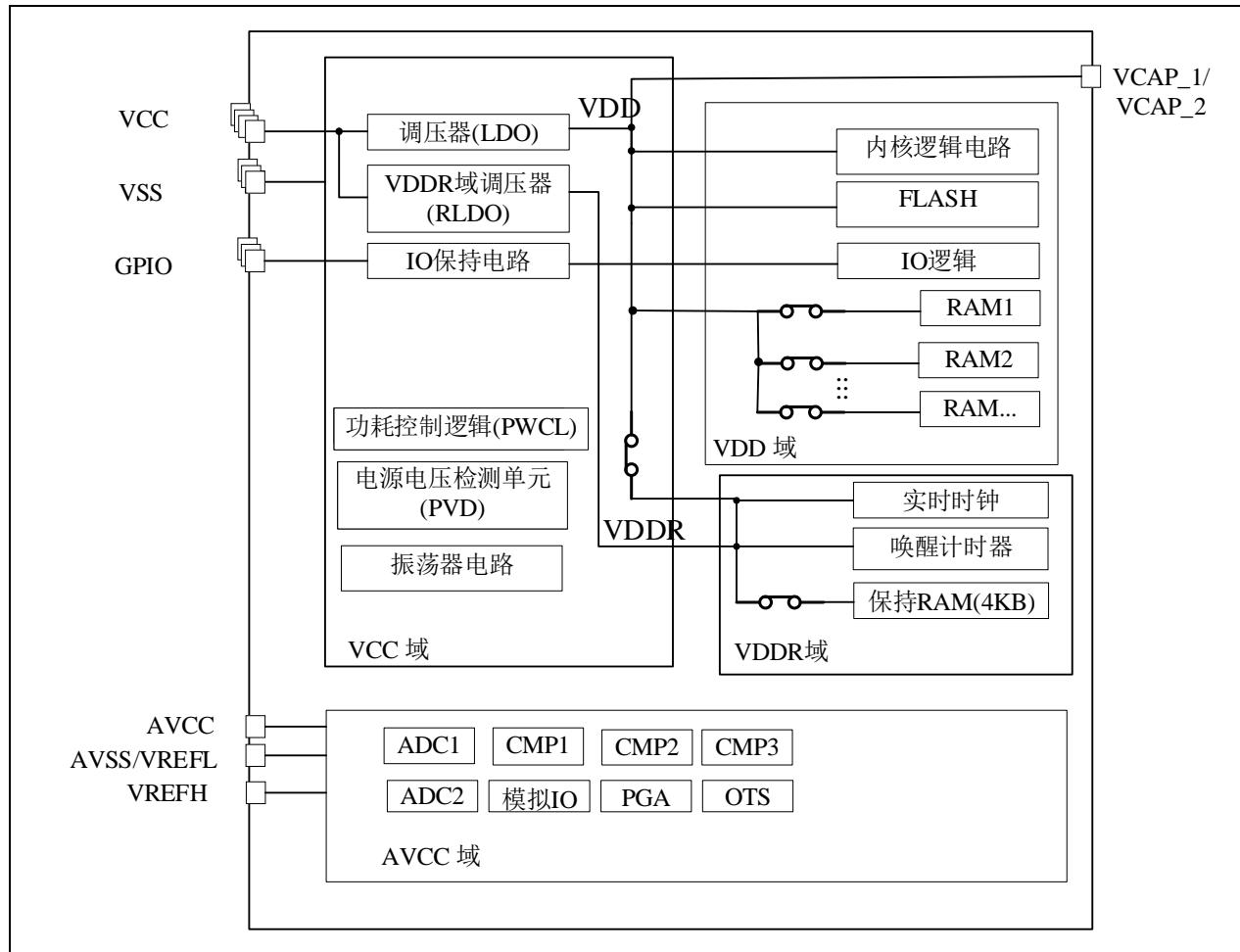


图 5-1 电源构成图

5.3 电源电压检测单元(PVD)说明

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

5.3.1 上电复位/掉电复位动作说明

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

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

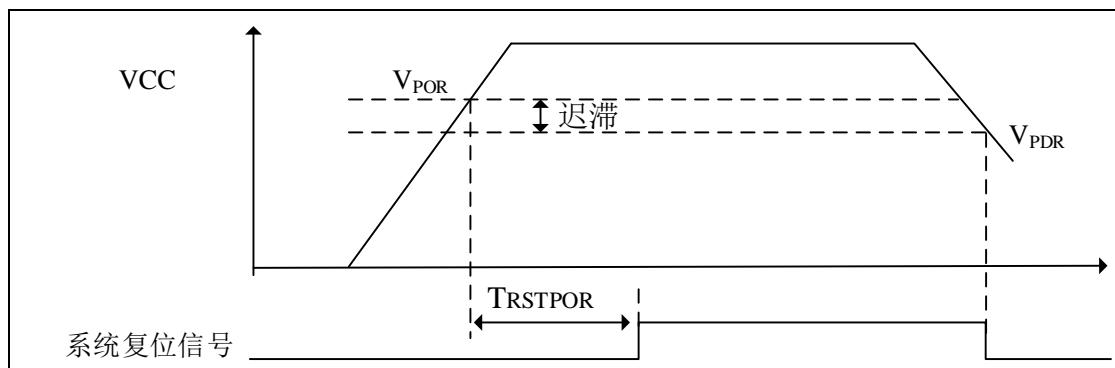


图 5-2 上电复位、掉电复位波形

5.3.2 欠压复位(BOR)说明

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

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

BORDIS	BOR_lev	说明
1	XX	BOR无效
0	00	BOR有效、选择BOR阈值0(V_{BOR0})
0	01	BOR有效、选择BOR阈值1(V_{BOR1})
0	10	BOR有效、选择BOR阈值2(V_{BOR2})
0	11	BOR有效、选择BOR阈值3(V_{BOR3})

表 5-1 BOR 配置

BOR 阈值的电气特性，请参考电气特性。

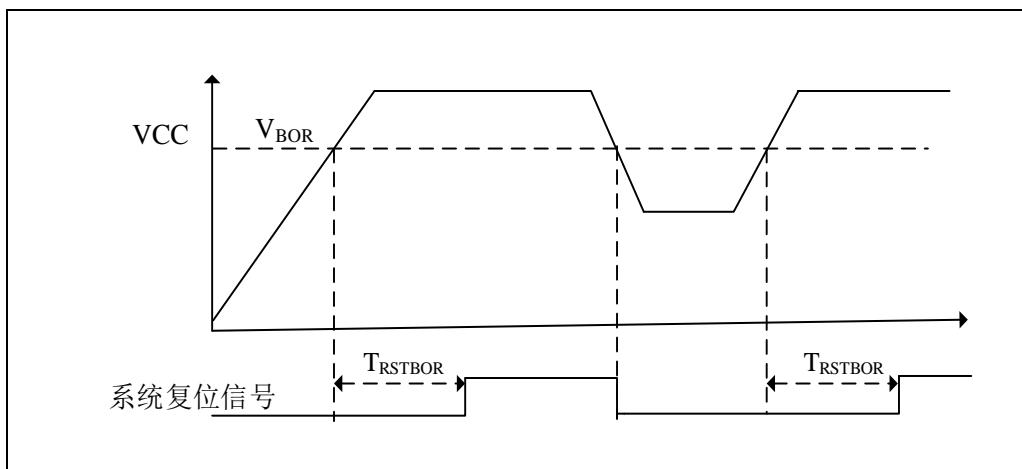


图 5-3 欠压复位波形

5.3.3 可编程电压检测 1(PVD1)、可编程电压检测 2(PVD2)

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

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

可编程电压检测主要特性如表 5-2 所示。

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)	1. PWC_PVDLCR.PVD2LVL[2:0]设置为111以外的值时，VCC下降/上升过程中是否经过阈值电压点(VPVD2) 2. PWC_PVDLCR.PVD2LVL[2:0]=111时,外部输入电压的下降/上升过程是否经过阈值电压点(VPVD2)
检测电压点	由PVD1LVL[2:0]配置	由PVD2LVL[2:0]配置
复位	复位: VCC<VPVD1; 复位解除: VCC> VPVD1经过一定复位处理时间。	复位: VCC<VPVD2; 复位解除: VCC> VPVD2经过一定复位处理时间。
中断	配置成电压检测1中断或者非可屏蔽中断 VCC下降经过阈值电压点(VPVD1)	配置成电压检测2中断或者非可屏蔽中断 VCC下降经过阈值电压点(VPVD2)
滤波功能	数字滤波	数字滤波
外围电路触发功能	VCC下降经过阈值电压点(VPVD1)	VCC下降经过阈值电压点(VPVD2)

表 5-2 PVD1/PVD2 特性

5.3.4 PVD1、PWD2 中断/复位框图

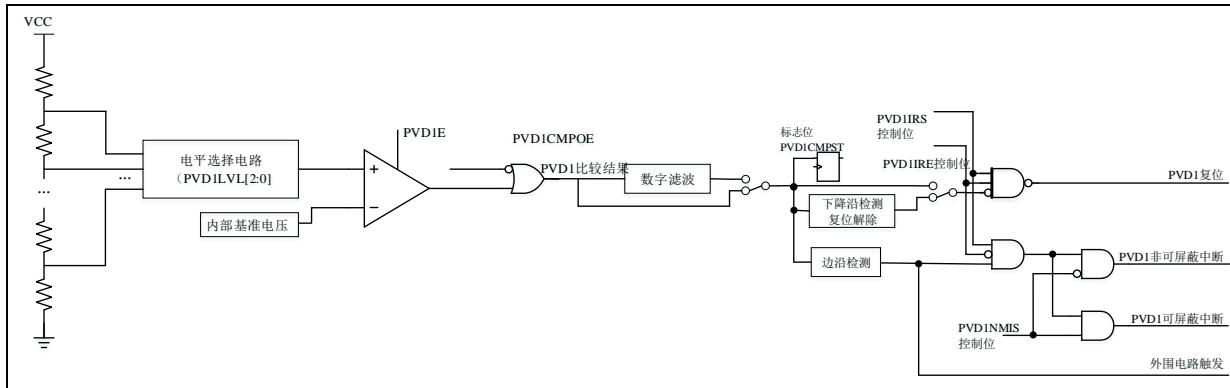


图 5-4 PVD1 中断/复位框图

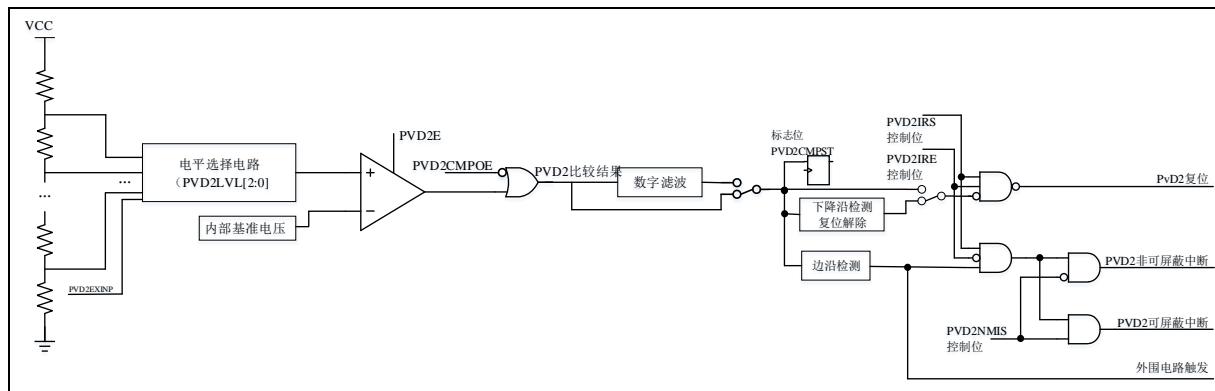


图 5-5 中断/复位框图

5.3.5 输入/输出引脚

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

5.3.6 PVD1 中断和复位

在停止模式或者掉电模式中使用 PVD1 电路时, 请遵守如下注意事项。

1. 停止模式

1) 必须将数字滤波器无效。

2. 掉电模式

1) 必须将数字滤波器无效。

2) PVD1IRS 置 0, 选择 PVD1 产生中断;选择复位功能时, 不能进入掉电模式。

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

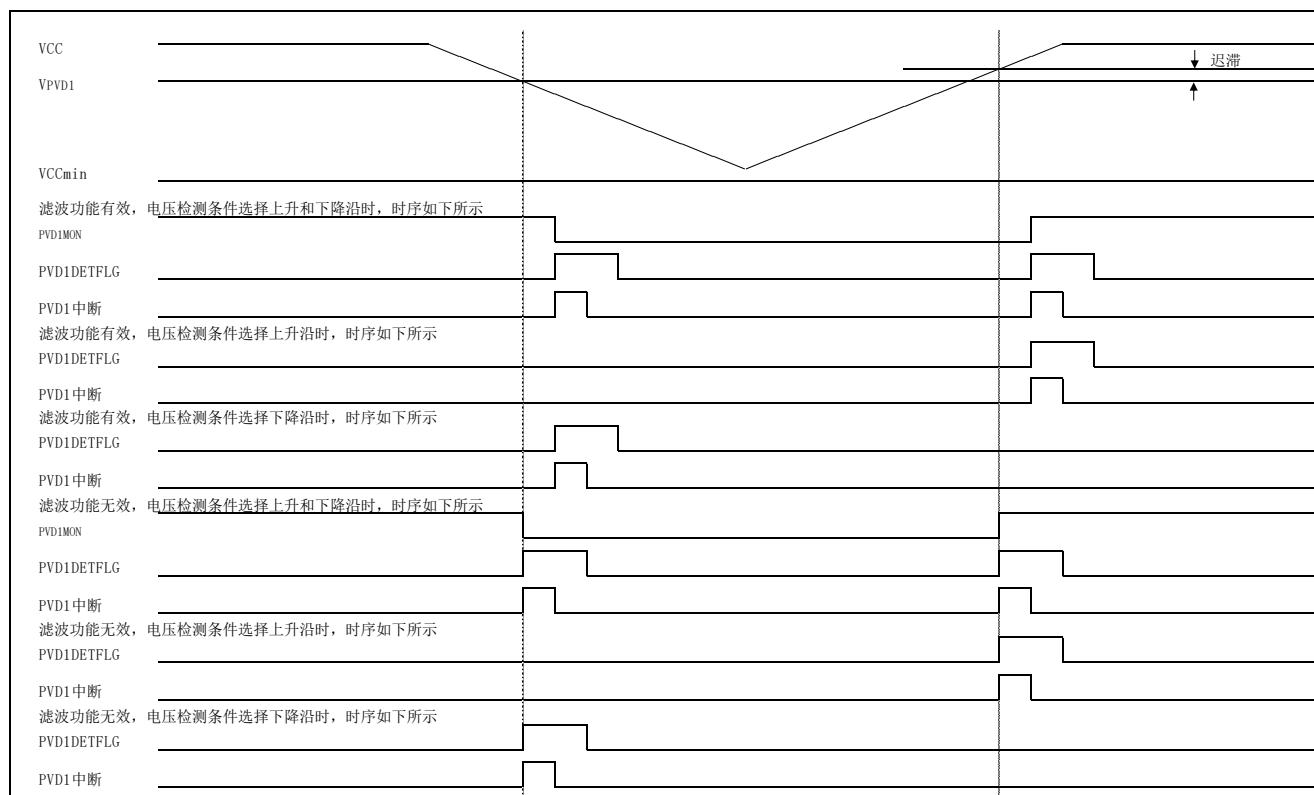


图 5-6 电源监视 1 中断时序图

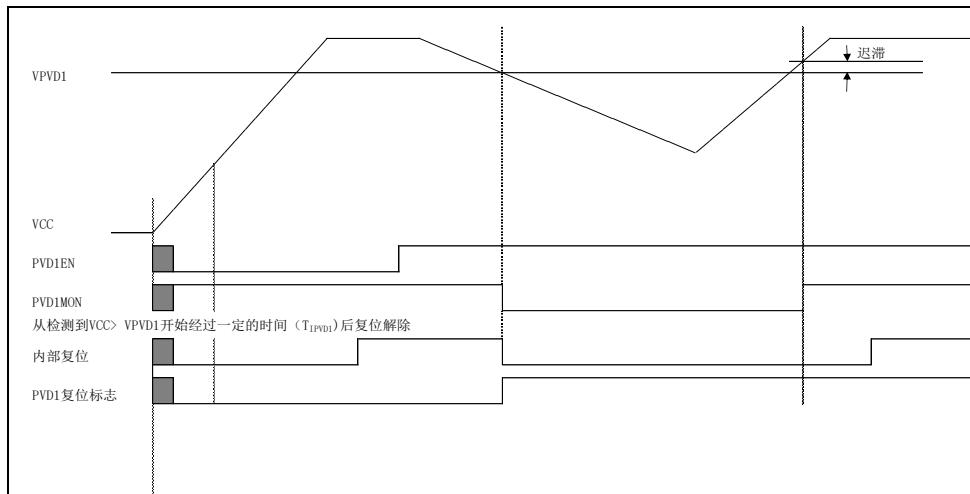


图 5-7 电源监视 1 复位时序图

5.3.7 PVD2 中断和复位

在停止模式或者掉电模式中使用 PVD2 电路时, 请遵守如下注意事项:

1. 停止模式

1) 必须将数字滤波器无效。

2. 掉电模式

1) 必须将数字滤波器无效。

2) 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。

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

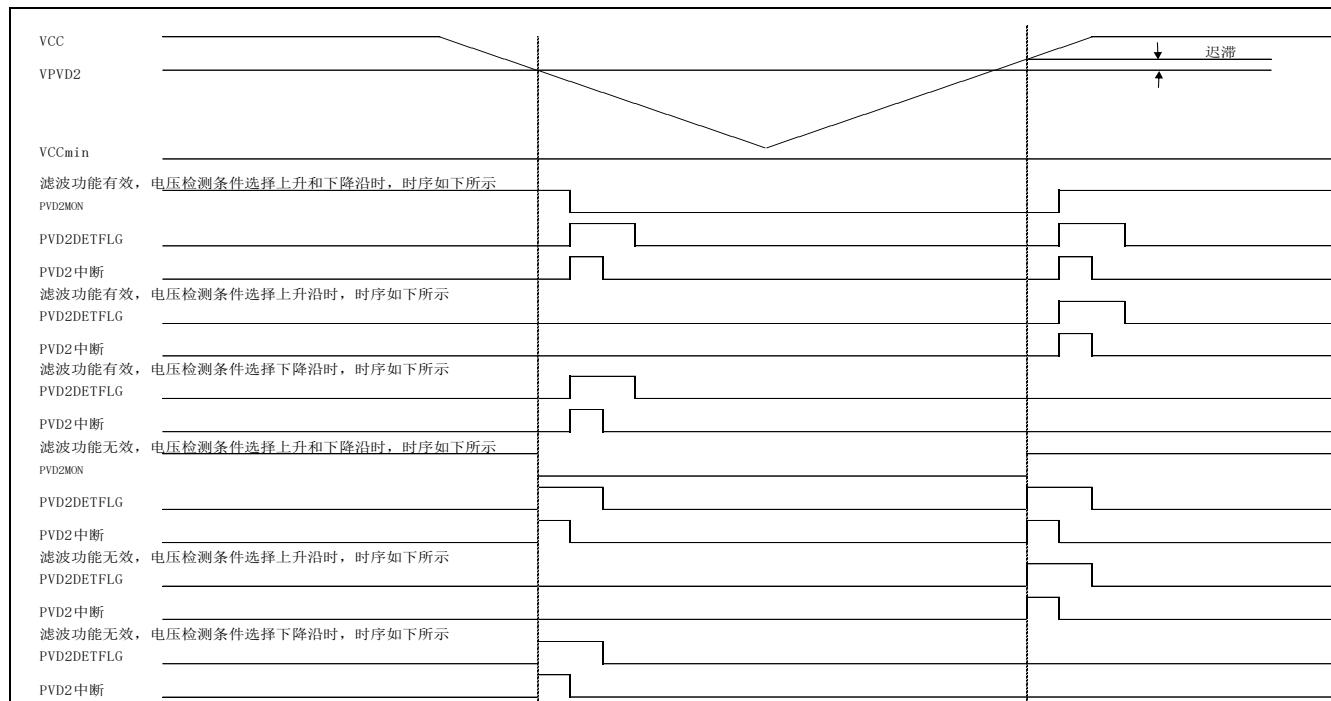


图 5-8 电源监视 2 中断运行时序图

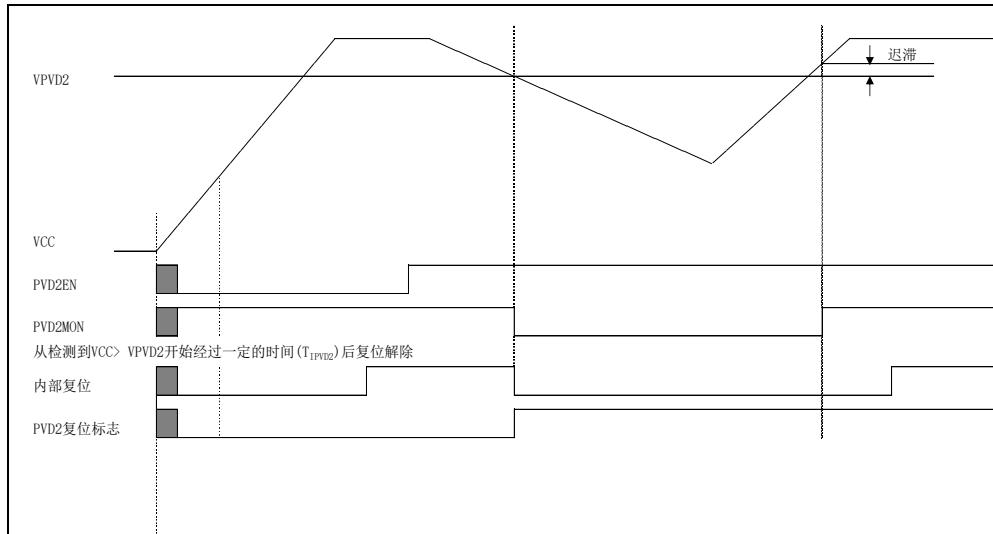


图 5-9 电源监视 2 复位运行时序图

5.3.8 内部电压采样和检测功能

芯片内部电压采样和检测功能是指基准电压测量功能。基准电压测量通路，通过 ADC 测量基准电压。内部基准电压约为 1.15V。

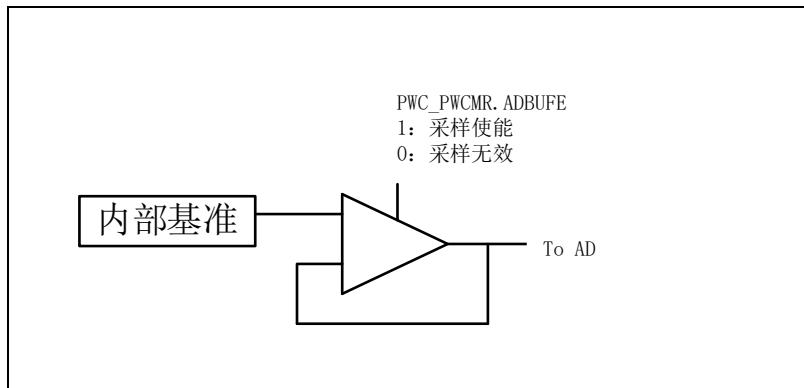


图 5-10 内部电压采样示意图

- 基准电压测量通路

使用基准电压测量通路时，需要按照下述步骤选中基准电压测量通路。

1. PWC_PWCMR.ADBUFE=1，使能内部电压测量功能
2. 根据 ADC 章节的表 16-2 配置寄存器，使 ADC 选通内部电压测量通道
3. 等待 50uS 后使用 ADC 测量内部基准电压

5.4 动作模式与低功耗模式

系统复位或者上电复位之后，芯片的所有电源域都处于供电状态，芯片进入高速运行模式。在运行模式下，CPU 通过 HCLK 提供时钟，并执行程序代码。芯片提供了超高速运行模式、高速运行模式、超低速运行模式等三种运行模式。芯片可以配置的运行模式如表 5-3 所示。

为了节省 CPU 不需要运行时的功耗，系统提供了睡眠模式、停止模式、掉电模式等三种低功耗模式。芯片可以配置的低功耗模式如表 5-4 所示。在睡眠模式中，芯片的 CortexTM-M4F 内核停止动作，外设保持运行状态；在停止模式中，芯片的外设和 CPU 都停止动作；在掉电模式中，VDD 域的电源被关闭，VDD 域的外设都停止动作。位于 VDDR 域中的实时时钟、唤醒定时器在低功耗模式中可以动作，保持 SRAM 可以保持数据；在不需要使用 VDDR 域中的实时时钟、唤醒定时器和保持 SRAM 时，可以设置关闭 VDDR 域的调压器 RLDO，在进入掉电模式后可以进一步降低功耗。

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

低功耗模式的运行条件及各模块在低功耗模式下的状态如表 5-5 所示。

运行模式	说明
超高速运行模式 (Ultra high Speed Run Mode)	主频200MHz以下
高速运行模式 (High Speed Run Mode)	主频168MHz以下
超低速运行模式 (Ultra low Speed Run Mode)	主频8MHz以下

表 5-3 运行模式

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

Mode)	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/RSTF0外芯片彻底复位。
	掉电模式4(PDMD4)	VDD域掉电之外，VDDR域掉电，上电复位电路进入低功耗模式，电压检测单元(PVD)无效

表 5-4 低功耗模式

项目	睡眠模式	停止模式	掉电模式
进入	PWC_STPMCR.STOP=0 PWC_PWRC0.PWDN=0, WFI	PWC_STPMCR.STOP=1 PWC_PWRC0.PWDN=0, WFI	PWC_STPMCR.STOP=1 PWC_PWRC0.PWDN=1, WFI
解除	任意中断或者复位	停止模式下可以使用的中断或复位	掉电模式下可以使用的唤醒事件或者复位
外部高速振荡器	工作可设	停止	停止
外部低速振荡器	工作可设	工作可设	工作可设
内部高速振荡器	工作可设	停止	掉电
内部中速振荡器	工作可设	停止	掉电
内部低速振荡器	工作可设	工作可设	工作可设
WDT专用时钟振荡器	工作可设	工作可设	掉电
PLLU	工作可设	停止	掉电
PLLM	工作可设	停止	掉电
CPU	停止(保持)	停止(保持)	掉电
RAM	工作可设 可以设置工作，掉电	停止(保持) 根据进入待机前的设定， 可以维持掉电或者睡眠	掉电
FLASH	工作可设	停止(保持)	掉电，内容保持
DMA	工作可设	停止(保持)	掉电
调压器	工作 驱动可以调整	工作 驱动可以调整	停止

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
USB-FS	工作可设	停止(保持)	掉电
Timer0	工作可设	工作可设	掉电
Ret-SRAM	工作可设 可以设置工作，掉电，睡眠	停止(保持) 可以设置掉电，睡眠	掉电模式1/2下停止(保持) 可以设置掉电，睡眠 掉电模式3/4掉电
WKTM	工作可设	工作可设	掉电模式1/2工作可设 掉电模式3/4下掉电
其他周边外设模块	工作可设	停止(保持)	掉电
AD	工作可设	停止	掉电
DA	工作可设	工作可设	掉电
PGA	工作可设	工作可设	掉电
CMP	工作可设	工作可设	掉电
PA0-PA10, PB0- PB2, PB5-PB10, PB12-PB15, PC0- PC13, PD0- PD15, PE0-PE15,	工作可设	保持	保持或者高阻

项目	睡眠模式	停止模式	掉电模式
PH2			
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
NRST复位管脚	芯片外面通过电路上拉到VCC	芯片外面通过电阻上拉到VCC	芯片外面通过电阻上拉到VCC
PA11-PA12	工作可设	保持； 由于该管脚的电平被拉高时会产生冗余电流，进入STOP模式时禁止上拉。	保持或者高阻； 由于该管脚的电平被拉高时会产生冗余电流，进入掉电模式时禁止上拉。
PB11/MD	工作可设	保持；	保持； 芯片外面通过电阻上拉到VCC
PA13-PA15， PB3,PB4	工作可设； 作为JTAG功能时，内置上拉电路有效	保持； 作为JTAG功能时，内置上拉电路有效	保持； 作为JTAG功能时，内置上拉电路有效

表 5-5 低功耗模式的运行条件及各模块在低功耗模式下的状态

5.4.1 运行模式

芯片有超高速、高速、超低速等三种运行模式。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。如表 5-6 所示，根据 PWC_PWRC2 的 DVS 和 DDAS 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 EFM_FRMC.LVM=1、PWC_RAMOPM=0x9062。

运行模式	频率范围	寄存器设定	
		PWC_PWRC2.DVS	PWC_PWRC2.DDAS
超高速运行模式	≤200MHz	00	1111
高速运行模式	≤168MHz以下	11	1111
超低速运行模式*	≤8MHz以下	10	1000

*：超低速模式只支持 Ta=-40° C~85° C

表 5-6 运行模式说明

超高速运行模式、高速运行模式、超低速运行模式之间的切换需要遵循如下流程 1~流程 6。

1. 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=1、RAM 动作模式寄存器 PWC_RAMOPM 设置成 0x9062
- 4) 确认 FRMC.LVM=1，PWC_RAMOPM=0x9062
- 5) 设置 PWC_PWRC2.DDAS[3:0]为 1000; PWC_PWRC2.DVS[1:0]为 10
- 6) 写 PWC_MDSWCR=0x10
- 7) 等待 TSWMD1(30uS)
- 8) 芯片在超低速模式下动作

2. 超低速模式向高速模式切换

- 1) 设置 PWC_PWRC2. DDAS[3:0]为 1111; PWC_PWRC2. DVS[1:0]根据系统频率要求设置为 11
- 2) 写 PWC_MDSWCR=0x10
- 3) 等待 TSWMD2(30uS)
- 4) 设置 FLASH 的 FRMC.LVM=0、RAM 动作模式寄存器 PWC_RAMOPM 设置成 0x8043
- 5) 确认 FRMC.LVM=0, PWC_RAMOPM=0x8043
- 6) 芯片在高速模式下动作

3. 高速模式向超高速模式切换

- 1) 设置 PWC_PWRC2. DDAS[3:0]为 1111; PWC_PWRC2. DVS[1:0]设置为 00
- 2) 写 PWC_MDSWCR =0x10
- 3) 等待 T SWMD2 (30uS)
- 4) 芯片在超高速模式下动作

4. 超高速模式向高速模式切换

- 1) 设置在高速速模式下使用的时钟源，确保时钟源符合高速模式下的频率要求
- 2) 设置 PWC_PWRC2. DDAS[3:0]为 1111; PWC_PWRC2. DVS[1:0]设置为 11
- 3) 写 PWC_MDSWCR =0x10
- 4) 等待 TSWMD2(30uS)
- 5) 芯片在高速模式下动作

5. 超低速模式向超高速模式切换

- 1) 设置 PWC_PWRC2. DDAS[3:0]为 1111; PWC_PWRC2. DVS[1:0]设置为 00
- 2) 写 PWC_MDSWCR =0x10
- 3) 等待 TSWMD2(30uS)
- 4) 设置 FLASH 的 FRMC.LVM=0、RAM 动作模式寄存器 PWC_RAMOPM 设置成 0x8043

- 5) 确认 FRMC.LVM=0, PWC_RAMOPM=0x8043
- 6) 芯片在超高速模式下动作

6. 超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=1、RAM 动作模式寄存器 PWC_RAMOPM 设置成 0x9062
- 4) 确认 FRMC.LVM=1, RAMOPT=0x9062
- 5) 设置 PWC_PWRC2.DDAS[3:0]为 1000; PWC_PWRC2.DVS[1:0]为 10
- 6) 写 PWC_MDSWCR =0x10
- 7) 等待 TSWMD1(30uS)
- 8) 芯片在超低速模式下动作

5.4.2 睡眠模式

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

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

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

- 进入睡眠模式

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

- 退出睡眠模式

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

5.4.3 停止模式

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

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

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

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,DAOE0,DAOE1 清“0”。要禁止 ADC，需要 ADC_STR STRT 位清“0”后将寄存器 PWC_FCG3 中 ADC 对应位写 1，使 ADC 进入模块停止状态后，再执行 WFI 指令进入停止模式。

STOP 模式唤醒时，通过 PWC_STPMCR 寄存器的位 CKSMRC 和 FLNWT 来选择唤醒后的时钟及是否需要等待 FLASH 稳定。CKSMRC 用于控制唤醒后的时钟源，CKSMRC=1 时，唤醒后的系统时钟源选择为 MRC；CKSMRC=0 时，唤醒后的系统时钟维持进入 STOP 之前的时钟源不变。FLNWT 用于控制唤醒后是否等待 FLASH 稳定，FLNWT=0 时，唤醒时需要等待 FLASH 稳定；FLNWT=1 时，唤醒是不需要等待 FLASH 稳定。FLNWT 必须在程序运行在 RAM 上时才能置位，否则芯片从 STOP 唤醒后的动作不可保证。在程序运行在 RAM 上进入 STOP 模式时，选择 CKSMRC=1，FLNWT=1 将可以在最短的时间内唤醒系统。

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

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

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

芯片在超低速模式下需要设置 PWC_PWRC1.STPDAS=11 后再进入 STOP 模式时；如果 PWC_PWRC1.STPDAS=00 时进入 STOP 模式，在 STOP 模式中芯片会消耗更多的电流。

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

- 进入停止模式

在 PWC_STPMCR.STOP=1,PWC_PWRC0.PWDN=0 时执行 WFI 指令即可进入停止模式。表 5-5 给出了芯片的外设和时钟源在停止模式下的状态。

- 解除停止模式

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

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

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

5.4.4 掉电模式

在掉电模式中，VDD 域的所有模块的电源被切断，功耗可以达到最低。

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

当电压监测 1 和电压监测 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，并且振荡停止监测功能无效，否则芯片会进入睡眠模式而非掉电模式。

芯片的 VCAP_1/VCAP_2 管脚使用的电容如下：1) 同时存在 VCAP_1 和 VCAP_2 管脚的芯片，每个管脚可以使用 0.047uF 或者 0.1uF 电容（总容量为 0.094uF 或者 0.2uF）。2) 只有 VCAP_1 管脚的芯片，可以使用 0.1uF 或者 0.22uF 电容。从掉电模式唤醒时，内核电压建立过程中需要给 VCAP_1/VCAP_2 充电。一方面，较小的 VCAP_1/VCAP_2 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/VCAP_2 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/VCAP_2 总容量必须与 PWC_PWRC3.PDTS 位的赋值相匹配。 VCAP_1/VCAP_2 的总容量为 0.2uF 或者 0.22uF 时，需要在进入掉电模式之前确保 PWC_PWRC3.PDTS 位清零。VCAP_1/VCAP_2 的总容量为 0.094uF 或者 0.1uF 时，需要在进入掉电模式之前确保 PWC_PWRC3.PDTS 位置位。

通过设置 PWC_PWRC0.PDMDS[1:0]，可以进一步降低掉电模式的功耗。掉电模式的子模式如表 5-7 所示。掉电模式 1 中电压监测电路可以使用，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 域参考电压电路、电压监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，电压监测电路停止工作，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、电压监测电路的稳定时间。掉电模式 3 中，VCC 域参考电压电路、电压监测电路、上电复位检测电路都停止工作，唤醒时需要等待这些电路的稳定，因此在实现最低功耗的同时，唤醒时间比掉电模式 2 和掉电模式 1 长。掉电模式 4 与掉电模式

3 中停止工作的电路相同，因此掉电模式 4 与掉电模式 3 具有相同的功耗。具体功耗数值和唤醒时间，请参考【电气特性-低功耗模式】。

VDDR 域在掉电模式模式 1 和掉电模式 2 下可以工作，所以实时时钟模块、唤醒计时器可以继续运行，并可以用于唤醒掉电模式。Ret-SRAM 在掉电模式下依然可以保持数据。

如果在掉电模式下实时时钟、唤醒计时器、Ret-SRAM 都不需要使用，可以置位 PWC_PWRC0.VVDRSD 关闭低功耗调压器进一步降低功耗。在掉电模式 3 和掉电模式 4 下，VDDR 域也处于掉电状态。

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

表 5-7 掉电模式子模式

功耗和唤醒时间的关系： Ipd1>Ipd2>Ipd3=Ipd4, Tpd1<Tpd2<Tpd4<Tpd3

- 进入掉电模式

在 PWC_STPMCR.STOP=1,PWC_PWRC0.PWDN=1 时执行 WFI 指令即可进入掉电模式。

- 解除掉电模式

掉电模式可以通过掉电模式唤醒事件或者复位解除。能够用于唤醒掉电模式的复位有管脚复位、上电复位和电压监测 0 复位。能够用于掉电模式唤醒的事件包括：

NMI唤醒事件、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、电压监测1唤醒事件、电压监测2唤醒事件、唤醒定时器唤醒事件

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

在掉电模式 3 中 POR、PDR, BOR、PWD1、PWD2 电路都处于无效状态，从掉电模式 3 唤醒后，除 PWC_PDWKF0/PWC_PDWKF1/RSTF0 外的所有寄存器都被复位，

芯片按照类似上电复位的方式工作；复位标志位可以通过 RSTF0.PDRF 查询。从掉电模式 3 唤醒后，VDDR 域的 RTC/WKTM 被复位，Ret-SRAM 的数据无法保证。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 RSTF0.PDRF 查询。从掉电模式 4 唤醒时，VDDR 域的 RTC/WKTM 被复位，Ret-SRAM 的数据无法保证。

掉电唤醒事件通过掉电唤醒使能寄存器(PWC_PDWKE0-PDWKE3)、掉电唤醒事件边沿选择寄存器(PWC_PDWKES)进行控制。当发生掉电唤醒事件时，该事件相对应的掉电唤醒标志(PWC_PDWKF0-PWC_PDWKF1)被置位。掉电唤醒后，如果不清除掉电唤醒标志，芯片不能再次进入掉电模式。掉电唤醒事件的边沿可以通过 PWC_PDWKES 选择。

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

掉电模式	不被复位的寄存器
掉电模式1	PWC_PWRC0 PWC_PWRC1 PWC_PWRC3
掉电模式2	PWC_PDWKE0 PWC_PDWKE1
掉电模式4	PWC_PDWKE2 PWC_PDWKES PWC_PDWKF0 PWC_PDWKF1 PWC_PWCMR PWC_XTAL32CS PWC_PVDCR0 PWC_PVDCR1 PWC_PVDFCR PWC_PVDLCSR PWC_PVDICR PWC_PVDDSR
掉电模式3	PWC_PDWKF0 PWC_PDWKF1 RSTF0

- 解除掉电模式后的管脚状态

在掉电模式中，芯片管脚将保持进入掉电模式之前的状态或者通过寄存器设定为高阻状态。如果 PWC_PWRC0.IORTN[1:0]=10 或者 11，管脚状态在掉电模式中为高阻

状态，掉电模式解除后被初始化。如果 PWC_PWRC0.IORTN[1:0]=00，管脚状态在掉电模式中保持掉电模式前的状态，唤醒后管脚初始成高阻状态。如果 PWC_PWRC0.IORTN[1:0]=01，芯片管脚将保持进入掉电模式之前的状态，唤醒后即使对外设或者管脚的寄存器进行设定，芯片管脚的状态也不会改变。通过软件将 PWC_PWRC0.IORTN 清零后，管脚状态才受外设或者管脚的寄存器设定所控制。

- WKTM 管脚用于掉电唤醒

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、64Hz 内部时钟信号作为时钟源，其中 64Hz 内部时钟信号在 RTC 使用外部低速振荡器作为时钟源，且 RTC 动作时有效。该计数器是加计数的计数器，在 WKTC0.WKTCE 置位后，计数器开始计数，当计数值等于 WKTCMP[11:0]设置值后，计数停止并产生唤醒事件，使芯片从掉电模式唤醒。再次使用 WKTM 时需要将 WKTC0.WKTCE 复位后再次置位。

- PTWK 掉电模式唤醒事件

PTWK 用于芯片的掉电模式唤醒，通过软件设定，每个 PTWK 事件可以从 4 个管脚选择，可以选择上升或者下降触发边沿，具有独立的标志位。PTWK_n 的结构框图如图 5-11 PTWK_n 结构框图所示。

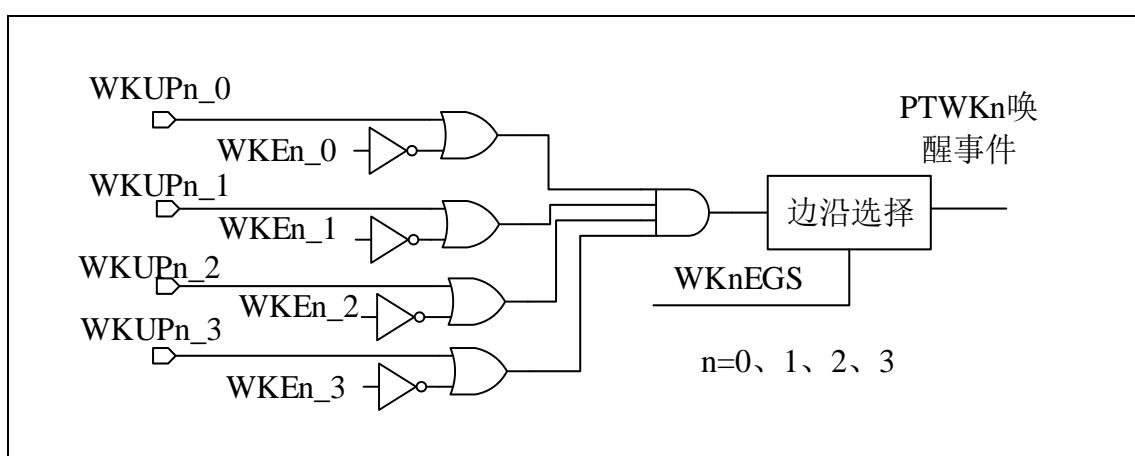


图 5-11 PTWK_n 结构框图

- VDDR 域在掉电模式时的动作说明

VDDR 域在芯片进入掉电模式 1 或者掉电模式 2 后通过 RLDO 继续供电，因此 RTC /WKTM/Ret-SRAM 都能继续动作或者保持数据。掉电唤醒后 VDDR 域由 LDO 供电。在掉电模式 3 或者掉电模式 4 下 VDDR 域停止供电。

5.5 降低功耗的方法

可以通过下列方法优化运行模式下的功耗。

1. 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=0/1/2/3) 关闭不需要使用的功能
5. 关闭 RAM 的电源

5.5.1 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。有关详细信息，请参考【时钟控制器(CMU)】。

5.5.2 关闭不使用的时钟源

芯片的系统时钟有 6 个时钟源：

- 外部高速振荡器(XTAL)
- 外部低速振荡器(XTAL32)
- MPLL 时钟(MPLL)
- 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。除此之外，芯片还配置了一个 I2S 使用的 UPLL 时钟源。

对于每一个时钟源，在未使用时都可以单独关闭，降低系统功耗。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 位切断 HRC 的电源，进一步降低功耗；在 UPLL 和 MPLL 都被关闭后，可以通过置位 PWC_PWRC1.VPLLSD 位切断 UPLL 和 MPLL 的电源，进一步降低功耗。

有关详细信息，请参考【时钟控制器(CMU)】。

5.5.3 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。

5.5.4 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。表 5-8 是 RAM 模块与掉电控制位的对应关系，通过设置 PWC_RAMPC0 寄存器的相应 RAMPDCn(n=0-8)、PW_PWRC0.RETRAMSD 位可以使对应的 RAM 掉电。

RAM模块	说明	掉电控制位
SRAM1	0x2000_0000~0x2000_FFFF 地址空间用RAM	PWC_RAMPC0.RAMPDC0
SRAM2	0x2001_0000~0x2001_FFFF 地址空间用RAM	PWC_RAMPC0.RAMPDC1
SRAM3	0x2002_0000~0x2002_6FFF 地址空间用RAM	PWC_RAMPC0.RAMPDC2
SRAMH	0x1FFF_8000~0x1FFF_FFFF 地址空间用RAM	PWC_RAMPC0.RAMPDC3
USBFS	USBFS FIFO用RAM	PWC_RAMPC0.RAMPDC4
SDIO0RAM	SDIO0用RAM	PWC_RAMPC0.RAMPDC5
SDIO1RAM	SDIO1用RAM	PWC_RAMPC0.RAMPDC6
CANRAM	CAN用RAM	PWC_RAMPC0.RAMPDC7
CACHERAM	Cache用RAM	PWC_RAMPC0.RAMPDC8
Ret-SRAM	0x200F_0000~0x200F_0FFF 地址空间用RAM	PWC_PWRC0.RETRAMSD

表 5-8 RAM 模块与 RAM 掉电控制位

5.6 寄存器保护功能

寄存器保护功能用于将寄存器的写操作无效，以保护寄存器被意外改写。表 5-9 是寄存器保护位和被保护寄存器的列表。

保护寄存器位	被保护寄存器
PWC_FPRC. FPRCB0	CMU_XTALCFGR, CMU_XTALSTBCR, CMU_XTALCR, CMU_XTALSTDRCR, CMU_XTALSTDTSR, CMU_HRCTRM, CMU_HRCCR, CMU_MRCTRM, CMU_MRCCR, CMU_PLLCFGR, CMU_PLLCR, CMU_UPLLCFGR, CMU_UPLLCR, CMU_OSCSTBSR, CMU_CKSWR, CMU_SCFGR, CMU_UFSCKCFG, CMU_TPIUCKCFG, CMU_MCO1CFG, CMU_MCO2CFG, CMU_XTAL32CR, CMU_XTALC32CFG, CMU_XTAL32NFR, CMU_LRCCR, CMU_LRCTRM, PWC_XTAL32CS
PWC_FPRC. FPRCB1	PWC_PWRC0, PWC_PWRC1, PWC_PWRC2, PWC_PWRC3, PWC_PDWKE0, PWC_PDWKE1, PWC_PDWKE2, PWC_PDWKES, PWC_PDWKF0, PWC_PDWKF1, PWC_PWCMR, CMU_PERICKSEL, CMU_I2SCKSEL, PWC_MDSWCR, PWC_STPMCR, PWC_RAMPC0, PWC_RAMOPM, RMU_RSTF0
PWC_FPRC. FPRCB3	PWC_PVDCR0, PWC_PVDCR1, PWC_PVDFCR, PWC_PVDLCR, PWC_PVDICR, PWC_PVDDSR

保护寄存器位	被保护寄存器
PWC_FCG0PRC.B0	PWC_FCG0

表 5-9 寄存器保护列表

5.7 寄存器说明

寄存器一览表如表 5-10 所示。

BASE ADDR:0x4005_4400

寄存器名	符号	偏移地址	位宽	复位值
电源模式控制寄存器0	PWC_PWRC0	0x00	8	0x00
电源模式控制寄存器1	PWC_PWRC1	0x01	8	0x00
电源模式控制寄存器2	PWC_PWRC2	0x02	8	0xFF
电源模式控制寄存器3	PWC_PWRC3	0x03	8	0x07
掉电唤醒使能寄存器0	PWC_PDWKE0	0x04	8	0x00
掉电唤醒使能寄存器1	PWC_PDWKE1	0x05	8	0x00
掉电唤醒使能寄存器2	PWC_PDWKE2	0x06	8	0x00
掉电唤醒事件边沿选择寄存器	PWC_PDWKES	0x07	8	0x00
掉电唤醒标志寄存器0	PWC_PDWKF0	0x08	8	0x00
掉电唤醒标志寄存器1	PWC_PDWKF1	0x09	8	0x00
电源监测寄存器	PWC_PWCMR	0x0A	8	0x00
模式切换控制寄存器	PWC_MDSWCR	0x0F	8	0x00
PVD控制寄存器0	PWC_PVDCR0	0x12	8	0x00
PVD控制寄存器1	PWC_PVDCR1	0x13	8	0x00
PVD滤波控制寄存器	PWC_PVDFCR	0x14	8	0x11
PVD电平控制寄存器	PWC_PVDLCR	0x15	8	0x00
XTAL32电流控制寄存器	PWC_XTAL32CS	0x2B	8	0x02

BASE ADDR:0x4005_4000

寄存器名	符号	偏移地址	位宽	复位值
STOP模式唤醒控制寄存器	PWC_STPMCR	0x0C	16	0x4000
RAM功耗控制寄存器0	PWC_RAMPC0	0x14	32	0x0000_0000
RAM运行条件寄存器	PWC_RAMOPM	0x18	16	0x8043
PVD中断控制寄存器	PWC_PVDICR	0xE0	8	0x00
PVD检测状态寄存器	PWC_PVDDSR	0xE1	8	0x11
功能保护控制寄存器	PWC_FPRC	0x3FE	16	0x0000

BASE ADDR:0x4004_C400				
寄存器名	符号	偏移地址	位宽	复位值
唤醒计时器控制寄存器	PWC_WKTCR	0x00	16	0x0000
BASE ADDR:0x4004_8000				
寄存器名	符号	偏移地址	位宽	复位值
功能时钟控制0	PWC_FCG0	0x00	32	0xFFFF_FAEE
功能时钟控制1	PWC_FCG1	0x04	32	0xFFFF_FFFF
功能时钟控制2	PWC_FCG2	0x08	32	0xFFFF_FFFF
功能时钟控制3	PWC_FCG3	0x0C	32	0xFFFF_FFFF
PWC_FCG0保护控制	PWC_FCG0PC	0x10	32	0x0000_0000

表 5-10 寄存器一览表

5.7.1 电源模式控制寄存器 0 (PWC_PWRC0)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
PWDN	-	IORTN[1:0]		RETRAMSD	VVDRSD		PDMDS[1:0]
<hr/>							
位	标记	位名	功能			读写	
b7	PWDN	掉电模式控制位	0: 掉电模式无效 1: 掉电模式使能			R/W	
b6	Reserved	-	读出时为“0”,写入时写“0”			R/W	
b5-b4	IORTN[1:0]	掉电模式下IO保持控制	00: 掉电模式中IO保持状态, 掉电唤醒后硬件解除IO保持状态 01: 掉电模式中IO保持状态, 掉电唤醒后将IORTN[1:0]设为00b, 解除IO保持状态 1x: 掉电模式中及掉电唤醒后IO为高阻			R/W	
b3	RETRAMSD	保持RAM掉电控制	0: Ret-SRAM不掉电 1: Ret-SRAM 掉电			R/W	
b2	VVDRSD	VDR LDO控制	0: 使用RLDO 1: 关闭RLDO 当RLDO被关闭并且芯片进入掉电模式时RTC/唤醒定时器/保持RAM的电源被关闭, 因此不能使用。			R/W	
b1-b0	PDMDS[1:0]	掉电模式选择控制	PDMDS[1:0] 00: 掉电模式1 01: 掉电模式2 10: 掉电模式3 11: 掉电模式4			R/W	

5.7.2 电源模式控制寄存器 1 (PWC_PWRC1)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
STPDAS[1:0]	-	-	-	-	VHRCSD	VPLLSD	
<hr/>							
位	标记	位名	功能			读写	
b7-b6	STPDAS[1:0]	STOP模式LDO驱动选择	00:超高速模式、高速模式进入STOP模式时设定的驱动能力 11:超低速模式进入STOP模式时设定的驱动能力 0001: 禁止设定。			R/W	
b5-b2	Reserved	-	读出时为“0000”，写入时写“0000”			R/W	
b1	VHRCSD	HRC电源关闭	0: HRC电源使能 1: HRC电源关闭 在HRC不使用时，置位VHRCSD后关闭HRC用电源，进一步降低功耗。VHRCSD清零后，需要等待25uS后再启动HRC模块			R/W	
b0	VPLLSD	PLL电源关闭	0: PLL电源使能 1: PLL电源关闭 在UPLL和MPLL都关闭并等待50us后，置位VPLLSD后关闭PLL用电源，进一步降低功耗。VPLLSD清零后，需要等待25uS后再启动PLL模块。			R/W	

5.7.3 电源模式控制寄存器 2 (PWC_PWRC2)

复位值:0xFF

b7	b6	b5	b4	b3	b2	b1	b0
-	-	DVS[1:0]		DDAS[3:0]			

位	标记	位名	功能	读写
b7	Reserved	-	读出时为“1”,写入时写“1”	R/W
b6	Reserved	-	读出时为“1”,写入时写“1”	R/W
b5-b4	DVS[1:0]	动作模式下电压选择	00: 选择超高速动作模式电压 01: 设定禁止 10: 选择超低速动作电压 11: 选择高速动作模式电压	R/W
b3-b0	DDAS[3:0]	电源驱动选择	1111: 超高速运行模式、高速运行模式驱动能力选择 1110: 设定禁止 1001: 设定禁止 1000: 超低速运行模式驱动能力选择 其他: 设定禁止	R/W

5.7.4 电源模式控制寄存器 3 (PWC_PWRC3)

复位值: 0x07

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

位	标记	位名	功能	读写
b7-b3	Reserved	-	预留位, 写入时写“00000”。	R/W
b2	PDTS	掉电唤醒时间控制位	0: VCAP_1/VCAP_2总电容是2个0.1uF时或者1个0.22uF时 1: VCAP_1/VCAP_2总电容是2个0.047uF时或者1个0.1uF时	R/W
b1	Reserved	-	读出时为“1”,写入时写“1”	R/W
b0	Reserved	-	读出时为“1”,写入时写“1”	R/W

5.7.5 掉电唤醒使能寄存器 0(PWC_PDWKE0)

复位值: 0x00

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

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

5.7.6 掉电唤醒使能寄存器 1(PWC_PDWKE1)

复位值: 0x00

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

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

5.7.7 掉电唤醒使能寄存器 2(PWC_PDWKE2)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
WKTMWKE	-	RTCALMWKE	RTCPRDWKE	-	NMIWKE	PVD2WKE	PVD1WKE

位	标记	位名	功能	读写
b7	WKTMWKE	WKT M 唤醒事件使能	0: WKT M 唤醒事件无效 1: WKT M 唤醒事件使能	R/W
b6	Reserved	-	读出时为“0”，写入时写“0”	R/W
b5	RTCALMWKE	RTC闹钟唤醒事件使能	0: RTC闹钟唤醒事件无效 1: RTC闹钟唤醒事件使能	R/W
b4	RTCPRDWKE	RTC周期唤醒事件使能	0: RTC周期唤醒事件无效 1: RTC周期唤醒事件使能	R/W
b3	Reserved	-	预留位，写入时写“0”。	R/W
b2	NMIWKE	NMI唤醒事件使能	0: NMI唤醒事件无效 1: NMI唤醒事件使能	R/W
b1	PVD2WKE	PVD2唤醒事件使能	0: PVD2唤醒事件无效 1: PVD2唤醒事件使能	R/W
b0	PVD1WKE	PVD1唤醒事件使能	0: PVD1唤醒事件无效 1: PVD1唤醒事件使能	R/W

5.7.8 掉电唤醒事件边沿选择寄存器(PWC_PDWKES)

复位值: 0x00

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

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

5.7.9 掉电唤醒标志寄存器 0(PWC_PDWKF0)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	NMIWKF	PVD2WKF	PVD1WKF	PTWK3F	PTWK2F	PTWK1F	PTWK0F

位	标记	位名	功能	读写
b7	Reserved	-	读出时为“1”,写入时写“1”	R/W
b6	NMIWKF	NMI唤醒标志位	0: 未发生NMI管脚唤醒事件 1: 发生NMI管脚唤醒事件 掉电唤醒后, 需要写零清除本位。	R/W
b5	PVD2WKF	PVD2唤醒标志位	0: 未发生PVD2唤醒事件 1: 发生PVD2唤醒事件 掉电唤醒后, 需要写零清除本位。	R/W
b4	PVD1WKF	PVD1唤醒标志位	0: 未发生PVD1唤醒事件 1: 发生PVD1唤醒事件 掉电唤醒后, 需要写零清除本位。	R/W
b3	PTWK3F	PTWK3唤醒标志位	0: 未发生PTWK3唤醒事件 1: 发生PTWK3唤醒事件 掉电唤醒后, 需要写零清除本位。	R/W
b2	PTWK2F	PTWK2唤醒标志位	0: 未发生PTWK2唤醒事件 1: 发生PTWK2唤醒事件 掉电唤醒后, 需要写零清除本位。	R/W
b1	PTWK1F	PTWK1唤醒标志位	0: 未发生PTWK1唤醒事件 1: 发生PTWK1唤醒事件 掉电唤醒后, 需要写零清除本位。	R/W
b0	PTWK0F	PTWK0唤醒标志位	0: 未发生PTWK0唤醒事件 1: 发生PTWK0唤醒事件 掉电唤醒后, 需要写零清除本位。	R/W

5.7.10 掉电唤醒标志寄存器 1(PWC_PDWKF1)

复位值: 0x00

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

位	标记	位名	功能	读写
b7	WKTMWKF	WKTM唤醒标志位	0: 未发生WKTM唤醒事件 1: 发生WKTM唤醒事件 掉电唤醒后, 需要写零清除本位。	R/W
b6	Reserved	-	预留位, 写入时写“0”。	R/W
b5	RTCALMWKF	RTC闹钟唤醒标志位	0: 未发生RTC闹钟唤醒事件 1: 发生RTC闹钟唤醒事件 掉电唤醒后, 需要写零清除本位。	R/W
b4	RTCPRDWKF	RTC周期唤醒标志位	0: 未发生RTC周期唤醒事件 1: 发生RTC周期唤醒事件 掉电唤醒后, 需要写零清除本位。	R/W
b3	Reserved	-	预留位, 写入时写“0”。	R/W
b2	Reserved	-	预留位, 写入时写“0”。	R/W
b1	Reserved	-	预留位, 写入时写“0”。	R/W
b0	Reserved	-	预留位, 写入时写“0”。	R/W

5.7.11 电源监测控制寄存器 (PWC_PWCMR)

复位值:0x00

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

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

5.7.12 模式切换控制寄存器(PWC_MDSWCR)

复位值： 0x00

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

位	标记	位名	功能	读写
b7-b0	MDSWC[7:0]	模式切换使能	在进行动作模式切换时，设置好PWRC2后，需要将MDSWC[7:0]设置成0x10后才能生效。 只能写入0x10，禁止写入其他值，读出为0x00。	R/W

5.7.13 功能时钟控制 0(PWC_FCG0)

复位值:0xFFFF_FAEE

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
KEY	-	-	-	DCU4	DCU3	DCU2	DCU1	CRC	TRNG	HASH	AES	-	-	AOS	FCM
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DMA2	DMA1	-	-	-	SRAMRET	-	SRAM3	-	-	-	SRAM12	-	-	-	SRAMH

位	标记	位名	功能	读写
b31	KEY	KEYSCAN功能控制	0: 键盘扫描控制模块KEYSCAN功能使能 1: 键盘扫描控制模块KEYSCAN功能无效	R/W
b30	Reserved	-	读出时为“1”, 写入时写“1”	R/W
b29	Reserved	-	读出时为“1”, 写入时写“1”	R/W
b28	Reserved	-	读出时为“1”, 写入时写“1”	R/W
b27	DCU4	DCU4功能控制	0: 数字计算单元DCU3功能使能 1: 数字计算单元DCU3功能无效	R/W
b26	DCU3	DCU3功能控制	0: 数字计算单元DCU2功能使能 1: 数字计算单元DCU2功能无效	R/W
b25	DCU2	DCU2功能控制	0: 数字计算单元DCU1功能使能 1: 数字计算单元DCU1功能无效	R/W
b24	DCU1	DCU1功能控制	0: 数字计算单元DCU0功能使能 1: 数字计算单元DCU0功能无效	R/W
b23	CRC	CRC功能控制	0: CRC功能使能 1: CRC功能无效	R/W
b22	TRNG	TRNG功能控制	0: 加密协处理模块CPM中的真随机发生器TRNG功能使能 1: 加密协处理模块CPM中的真随机发生器TRNG功能无效	R/W
b21	HASH	HASH功能控制	0: 加密协处理模块CPM中的安全散列算法模块HASH功能使能 1: 加密协处理模块CPM中的安全散列算法模块功能无效	R/W
b20	AES	AES功能控制	0: 加密协处理模块CPM中的加解密算法处理器AES功能使能 1: 加密协处理模块CPM中的加解密算法处理器AES功能无效	R/W
b19	Reserved	-	读出时为“1”, 写入时写“1”	R/W
b18	Reserved	-	读出时为“1”, 写入时写“1”	R/W

b17	AOS	外设电路触发功能控制	0:外设电路触发功能使能 1:外设电路触发功能无效	R/W
b16	FCM	FCM功能控制	0:时钟控制器CMU中的时钟频率测量模块FCM功能使能 1: 时钟控制器CMU中的时钟频率测量模块FCM功能无效	R/W
b15	DMA2	DMA2功能控制	0:DMA控制器DMA2功能使能 1: DMA控制器DMA2功能无效	R/W
b14	DMA1	DMA1功能控制	0: DMA控制器DMA1功能使能 1: DMA控制器DMA1功能无效	R/W
b13	Reserved	-	读出时为“1”,写入时写“1”	R/W
b12	Reserved	-	读出时为“1”,写入时写“1”	R/W
b11	Reserved	-	读出时为“1”,写入时写“1”	R/W
b10	SRAMRET	Ret_SRAM功能控制	0: 内置SRAM中的Ret_SRAM功能使能 1: 内置SRAM中的Ret_SRAM功能无效	R/W
b9	Reserved	-	读出时为“1”,写入时写“1”	R/W
b8	SRAM3	ECCRAM功能控制	0: 内置SRAM中的SRAM3功能使能 1: 内置SRAM中的SRAM3功能无效	R/W
b7	Reserved	-	读出时为“1”,写入时写“1”	R/W
b6	Reserved	-	读出时为“1”,写入时写“1”	R/W
b5	Reserved	-	读出时为“1”,写入时写“1”	R/W
b4	SRAM12	SRAM1/SRAM2功能控制	0: 内置SRAM中的SRAM1,SRAM2功能使能 1: 内置SRAM中的SRAM1,SRAM2功能无效	R/W
b3	Reserved	-	读出时为“1”,写入时写“1”	R/W
b2	Reserved	-	读出时为“1”,写入时写“1”	R/W
b1	Reserved	-	读出时为“1”,写入时写“1”	R/W
b0	SRAMH	RAMHS功能控制	0: 内置SRAM中的SRAMH功能使能 1: 内置SRAM中的SRAMH功能无效	R/W

5.7.14 功能时钟控制 1(PWC_FCG1)

复位值: 0xFFFF_FFFF

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	USART4	USART3	USART2	USART1
b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	SPI4	SPI3	SPI2	SPI1
b15	b14	b13	b12	b11	b10	b9	b8
I2S4	I2S3	I2S2	I2S1	SDIOC2	SDIOC1	-	USBFS
b7	b6	b5	b4	b3	b2	b1	b0
-	I2C3	I2C2	I2C1	QSPI	-	-	CAN

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“1”,写入时写“1”	R/W
b30	Reserved	-	读出时为“1”,写入时写“1”	R/W
b29	Reserved	-	读出时为“1”,写入时写“1”	R/W
b28	Reserved	-	读出时为“1”,写入时写“1”	R/W
b27	USART4	USART4功能控制	0:通用同步异步收发器USART4功能使能 1: 通用同步异步收发器USART4功能无效	R/W
b26	USART3	USART3功能控制	0: 通用同步异步收发器USART3功能使能 1: 通用同步异步收发器USART3功能无效	R/W
b25	USART2	USART2功能控制	0: 通用同步异步收发器USART2功能使能 1: 通用同步异步收发器USART2功能无效	R/W
b24	USART1	USART1功能控制	0: 通用同步异步收发器USART1功能使能 1: 通用同步异步收发器USART1功能无效	R/W
b23	Reserved	-	读出时为“1”,写入时写“1”	R/W
b22	Reserved	-	读出时为“1”,写入时写“1”	R/W
b21	Reserved	-	读出时为“1”,写入时写“1”	R/W
b20	Reserved	-	读出时为“1”,写入时写“1”	R/W
b19	SPI4	SPI4功能控制	0: 串行外设接口SPI4功能使能 1: 串行外设接口SPI4功能无效	R/W
b18	SPI3	SPI3功能控制	0: 串行外设接口SPI3功能使能 1: 串行外设接口SPI3功能无效	R/W
b17	SPI2	SPI2功能控制	0: 串行外设接口SPI2功能使能 1: 串行外设接口SPI2功能无效	R/W
b16	SPI1	SPI1功能控制	0: 串行外设接口SPI1功能使能	R/W

			1: 串行外设接口SPI1功能无效	
b15	I2S4	I2S4功能控制	0: 集成电路内置音频总线模块I2S4功能使能 1: 集成电路内置音频总线模块I2S4功能无效	R/W
b14	I2S3	I2S3功能控制	0: 集成电路内置音频总线模块I2S3功能使能 1: 集成电路内置音频总线模块I2S3功能无效	R/W
b13	I2S2	I2S2功能控制	0: 集成电路内置音频总线模块I2S2功能使能 1: 集成电路内置音频总线模块I2S2功能无效	R/W
b12	I2S1	I2S1功能控制	0: 集成电路内置音频总线模块I2S1功能使能 1: 集成电路内置音频总线模块I2S1功能无效	R/W
b11	SDIOC2	SDIOC2功能控制	0: SDIO控制器SDIOC2功能使能 1: SDIO控制器SDIOC2功能无效	R/W
b10	SDIOC1	SDIOC1功能控制	0: SDIO控制器SDIOC1功能使能 1: SDIO控制器SDIOC1功能无效	R/W
b9	Reserved	-	读出时为“1”,写入时写“1”	R/W
b8	USBFS	USBFS功能控制	0: USB2.0全速模块USBFS功能使能 1: USB2.0全速模块USBFS功能无效	R/W
b7	Reserved	-	读出时为“1”,写入时写“1”	R/W
b6	I2C3	I2C3功能控制	0: 集成电路总线I2C3功能使能 1: 集成电路总线I2C3功能无效	R/W
b5	I2C2	I2C2功能控制	0: 集成电路总线I2C2功能使能 1: 集成电路总线I2C2功能无效	R/W
b4	I2C1	I2C1功能控制	0: 集成电路总线I2C1功能使能 1: 集成电路总线I2C1功能无效	R/W
b3	QSPI	QSPI功能控制	0: 四线式串行外设接口QSPI功能使能 1: 四线式串行外设接口QSPI功能无效	R/W
b2	Reserved	-	读出时为“1”,写入时写“1”	R/W
b1	Reserved	-	读出时为“1”,写入时写“1”	R/W
b0	CAN	CAN功能控制	0: 控制器局域网CAN功能使能 1: 控制器局域网CAN功能无效	R/W

5.7.15 功能时钟控制 2(PWC_FCG2)

复位值: 0xFFFF_FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	TIME	TIME	TIME
EMB	-	-	-	-	TIME	R6_3	R6_2	R6_1							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EMB	-	-	-	-	R4_3	R4_2	R4_1	RA_6	RA_5	RA_4	RA_3	RA_2	RA_1	R0_2	R0_1

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“1”,写入时写“1”	R/W
b30	Reserved	-	读出时为“1”,写入时写“1”	R/W
b29	Reserved	-	读出时为“1”,写入时写“1”	R/W
b28	Reserved	-	读出时为“1”,写入时写“1”	R/W
b27	Reserved	-	读出时为“1”,写入时写“1”	R/W
b26	Reserved	-	读出时为“1”,写入时写“1”	R/W
b25	Reserved	-	读出时为“1”,写入时写“1”	R/W
b24	Reserved	-	读出时为“1”,写入时写“1”	R/W
b23	Reserved	-	读出时为“1”,写入时写“1”	R/W
b22	Reserved	-	读出时为“1”,写入时写“1”	R/W
b21	Reserved	-	读出时为“1”,写入时写“1”	R/W
b20	Reserved	-	读出时为“1”,写入时写“1”	R/W
b19	Reserved	-	读出时为“1”,写入时写“1”	R/W
b18	TIMER6_3	TIMER6_3功能控制	0:TIMER6_3功能使能 1:TIMER6_3功能无效	R/W
b17	TIMER6_2	TIMER6_2功能控制	0:TIMER6_2功能使能 1:TIMER6_2功能无效	R/W
b16	TIMER6_1	TIMER6_1功能控制	0:TIMER6_1功能使能 1:TIMER6_1功能无效	R/W
b15	EMB	EMB功能控制	0: 紧急刹车模块EMB功能使能 1: 紧急刹车模块EMB功能无效	R/W
b14	Reserved	-	读出时为“1”,写入时写“1”	R/W
b13	Reserved	-	读出时为“1”,写入时写“1”	R/W
b12	Reserved	-	读出时为“1”,写入时写“1”	R/W
b11	Reserved	-	读出时为“1”,写入时写“1”	R/W
b10	TIMER4_3	TIMER4_3功能控制	0:TIMER4_3功能使能	R/W

			1:TIMER4_3功能无效	
b9	TIMER4_2	TIMER4_2功能控制	0:TIMER4_2功能使能 1:TIMER4_2功能无效	R/W
b8	TIMER4_1	TIMER4_1功能控制	0:TIMER4_1功能使能 1:TIMER4_1功能无效	R/W
b7	TIMERA_6	TIMERA_6功能控制	0:TIMERA_6功能使能 1:TIMERA_6功能无效	R/W
b6	TIMERA_5	TIMERA_5功能控制	0:TIMERA_5功能使能 1:TIMERA_5功能无效	R/W
b5	TIMERA_4	TIMERA_4功能控制	0:TIMERA_4功能使能 1:TIMERA_4功能无效	R/W
b4	TIMERA_3	TIMERA_3功能控制	0:TIMERA_3功能使能 1:TIMERA_3功能无效	R/W
b3	TIMERA_2	TIMERA_2功能控制	0:TIMERA_2功能使能 1:TIMERA_2功能无效	R/W
b2	TIMERA_1	TIMERA_1功能控制	0:TIMERA_1功能使能 1:TIMERA_1功能无效	R/W
b1	TIMER0_2	TIMER0_2功能控制	0:TIMER0_2功能使能 1:TIMER0_2功能无效	R/W
b0	TIMER0_1	TIMER0_1功能控制	0:TIMER0_1功能使能 1:TIMER0_1功能无效	R/W

5.7.16 功能时钟控制 3(PWC_FCG3)

复位值: 0xFFFF_FFFF

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	-	-	-	-
b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	OTS	-	-	-	CMP
b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	ADC2	ADC1

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“1”,写入时写“1”	R/W
b30	Reserved	-	读出时为“1”,写入时写“1”	R/W
b29	Reserved	-	读出时为“1”,写入时写“1”	R/W
b28	Reserved	-	读出时为“1”,写入时写“1”	R/W
b27	Reserved	-	读出时为“1”,写入时写“1”	R/W
b26	Reserved	-	读出时为“1”,写入时写“1”	R/W
b25	Reserved	-	读出时为“1”,写入时写“1”	R/W
b24	Reserved	-	读出时为“1”,写入时写“1”	R/W
b23	Reserved	-	读出时为“1”,写入时写“1”	R/W
b22	Reserved	-	读出时为“1”,写入时写“1”	R/W
b21	Reserved	-	读出时为“1”,写入时写“1”	R/W
b20	Reserved	-	读出时为“1”,写入时写“1”	R/W
b19	Reserved	-	读出时为“1”,写入时写“1”	R/W
b18	Reserved	-	读出时为“1”,写入时写“1”	R/W
b17	Reserved	-	读出时为“1”,写入时写“1”	R/W
b16	Reserved	-	读出时为“1”,写入时写“1”	R/W
b15	Reserved	-	读出时为“1”,写入时写“1”	R/W
b14	Reserved	-	读出时为“1”,写入时写“1”	R/W
b13	Reserved	-	读出时为“1”,写入时写“1”	R/W
b12	OTS	OTS功能控制	0: 温度传感器OTS功能有效 1: 温度传感器OTS功能无效	R/W
b11	Reserved	-	读出时为“1”,写入时写“1”	R/W
b10	Reserved	-	读出时为“1”,写入时写“1”	R/W

b9	Reserved	-	读出时为“1”,写入时写“1”	R/W
b8	CMP	CMP功能控制	0: 电压比较器CMP功能使能 1: 电压比较器CMP功能无效	R/W
b7	Reserved	-	读出时为“1”,写入时写“1”	R/W
b6	Reserved	-	读出时为“1”,写入时写“1”	R/W
b5	Reserved	-	读出时为“1”,写入时写“1”	R/W
b4	Reserved	-	读出时为“1”,写入时写“1”	R/W
b3	Reserved	-	读出时为“1”,写入时写“1”	R/W
b2	Reserved	-	读出时为“1”,写入时写“1”	R/W
b1	ADC2	ADC2功能控制	0: 模数转换模块ADC2功能使能 1: 模数转换模块ADC2功能无效	R/W
b0	ADC1	ADC1功能控制	0: 模数转换模块ADC1功能使能 1: 模数转换模块ADC1功能无效	R/W

5.7.17 PWC_FCG0 保护控制(PWC_FCG0PC)

复位值:0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
FCG0PCWE[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PRT0

位	标记	位名	功能	读写
b31~b16	FCG0PCWE[15:0]	PWC_FCG0PC写使能	写入0xA5A5的同时改变PRT0位的值	R/W
b15-b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	PRT0	保护位	PWC_FCG0写使能控制位 0:PWC_FCG0写无效 1:PWC_FCG0写使能	R/W

5.7.18 功能保护控制寄存器(PWC_FPRC)

复位值:0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
								-	-	-	-	FPRCB3	FPRCB2	FPRCB1	FPRCB0

位	标记	位名	功能	读写
b15~b8	PWC_FPRCWE	PWC_FPRC寄存器写使能	写入0xA5h的同时能够更新PWC_FPRC值，否则对低8位写入值无效。读出时为0x00。	R/W
b7	Reserved	-	预留位，写入时写“0”。	R/W
b6	Reserved	-	预留位，写入时写“0”。	R/W
b5	Reserved	-	预留位，写入时写“0”。	R/W
b4	Reserved	-	预留位，写入时写“0”。	R/W
b3	FPRCB3	FPRC位3	保护寄存器位，保护对象参考表 5-9 0:写保护 1:写使能	R/W
b2	Reserved	-	预留位，写入时写“0”。	R/W
b1	FPRCB1	FPRC位1	保护寄存器位，保护对象参考表 5-9 0:写保护 1:写使能	R/W
b0	FPRCB0	FPRC位0	保护寄存器位，保护对象参考表 5-9 0:写保护 1:写使能	R/W

5.7.19 STOP 模式控制寄存器(PWC_STPMCR)

复位值: 0x4000

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

位	标记	位名	功能	读写
b15	STOP	STOP模式选择位	0: STOP模式无效 1: STOP模式有效	R/W
b14	Reserved	-	读出时为“1”,写入时写“1”	R/W
b13-b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	CKSMRC	时钟切换到MRC选项	0: 维持进入STOP模式之前的系统时钟及分频 1: STOP模式唤醒时系统时钟切换到MRC、SCKCFG寄存器被初始化	R/W
b0	FLNWT	FLASH稳定等待控制	0: STOP模式唤醒时等待FLASH稳定 1: STOP模式唤醒时不等待FLASH稳定	R/W-

5.7.20 RAM 功耗控制寄存器 0(PWC_RAMPC0)

复位值: 0x0000_0000

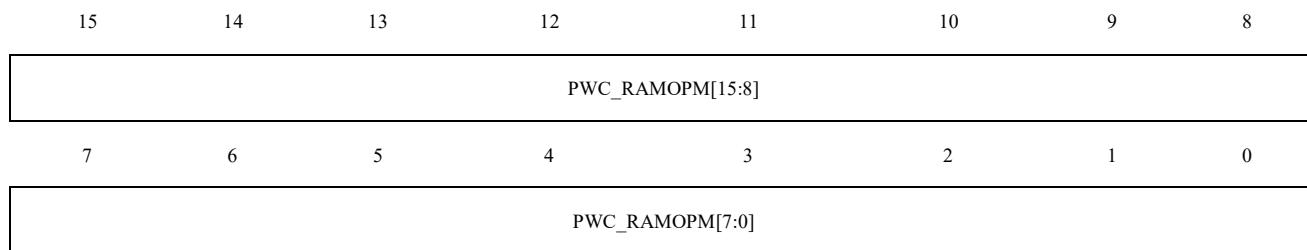
-	-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16	
-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	
-	-	-	-	-	-	-	-	RAMPDC8
7	6	5	4	3	2	1	0	
RAMPDC7	RAMPDC6	RAMPDC5	RAMPDC4	RAMPDC3	RAMPDC2	RAMPDC1	RAMPDC0	

位	标记	位名	功能	读写
b32-b23	Reserved	-	读出时为“0”,写入时写“0”	R/W
b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	Reserved	-	读出时为“0”,写入时写“0”	R/W
b22	Reserved	-	读出时为“0”,写入时写“0”	R/W
b21	Reserved	-	读出时为“0”,写入时写“0”	R/W
b20	Reserved	-	读出时为“0”,写入时写“0”	R/W
b19	Reserved	-	读出时为“0”,写入时写“0”	R/W
b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17	Reserved	-	读出时为“0”,写入时写“0”	R/W
b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15-b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8	RAMPDC8	RAM掉电控制位8	0: cache用RAM动作 1: cache用RAM掉电	R/W
b7	RAMPDC7	RAM掉电控制位7	0: can用RAM动作 1: can用RAM掉电	R/W
b6	RAMPDC6	RAM掉电控制位6	0: sdio1用RAM动作 1: sdio1用RAM掉电	R/W
b5	RAMPDC5	RAM掉电控制位5	0: sdio0用RAM动作 1: sdio0用RAM掉电	R/W
b4	RAMPDC4	RAM掉电控制位4	0: usbfs用RAM动作 1: usbfs用RAM掉电	R/W
b3	RAMPDC3	RAM掉电控制位3	0: 0x1FFF_8000~0x1FFF_FFFF空间RAM (SRAMH) 动作 1: 0x1FFF_8000~0x1FFF_FFFF空间RAM (SRAMH) 掉电	R/W

b2	RAMPDC2	RAM掉电控制位2	0: 0x2002_0000~0x2002_6FFF空间RAM (SRAM3) 动作 1: 0x2002_0000~0x2002_6FFF空间RAM (SRAM3) 掉电	R/W
b1	RAMPDC1	RAM掉电控制位1	0: 0x2001_0000~0x2001_FFFF空间RAM (SRAM2) 动作 1: 0x2001_0000~0x2001_FFFF空间RAM (SRAM2) 掉电	R/W
b0	RAMPDC0	RAM掉电控制位0	0: 0x2000_0000~0x2000_FFFF空间RAM (SRAM1) 动作 1: 0x2000_0000~0x2000_FFFF空间RAM (SRAM1) 掉电	R/W

5.7.21 RAM 运行条件寄存器(PWC_RAMOPM)

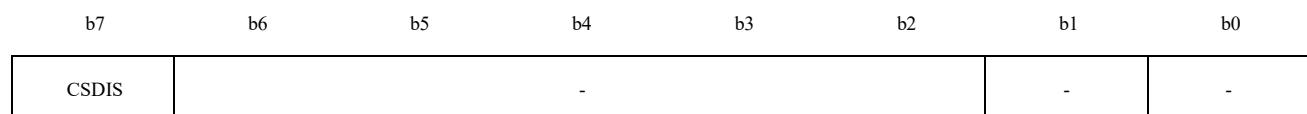
复位值：0x8043



位	标记	位名	功能	读写
b15-b0	PWC_RAMOPM[15:0]	RAM动作模式选择为	芯片工作在超高速/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	R/W

5.7.22 XTAL32 用电流源控制 (PWC_XTAL32CS)

复位值：0x02



位	标记	位名	功能	读写
7	CSDIS	电流源无效控制	0: 电流源有效 1: 电流源无效 在XTAL32/RTC/WTKM/Ret-SRAM等都不需要使用时，可以置位CSDIS，以降低功耗。	R/W
b6-b2	Reserved	-	读出时为“0”，写入时写“0”	R/W
b1	Reserved	-	读出时为“1”，写入时写“1”	R/W
b0	Reserved	-	读出时为“0”，写入时写“0”	R/W

5.7.23 唤醒计时器控制寄存器(PWC_WKTCR)

复位值: 0x0000

b15	b14-b13	b12	b11	-	b0
WKTCE	WKCKS[1:0]	WKOVF		WKTMCMP[11:0]	

位	标记	位名	功能	读写
b15	WKTCE	WKTMCMP使能	0: WKTMCMP停止 1: WKTMCMP计数	R/W
b14-b13	WKCKS[1:0]	WKTMCMP时钟选择	00:64Hz时钟 01: XTAL32 10: LRC 11: 预留	R/W
b12	WKOVF	WKTMCMP计数结果标志	0:计数器与WKTMCMP值不一致 1: 计数器与WKTMCMP值一致	R/W
b11-b0	WKTMCMP[11:0]	WKTMCMP比较位	WKTMCMP的比较值	R/W

5.7.24 PVD 控制寄存器 0(PWC_PVDCR0)

复位值: 0x00

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

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

5.7.25 PVD 控制寄存器 1(PWC_PVDCR1)

复位值: 0x00

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

位	标记	位名	功能	读写
b7	Reserved	-	读出时为“0”，写入时写“0”	R/W
b6	PVD2CMPOE	PVD2比较结果输出使能	0: 禁止输出比较器2的比较结果 1: 允许输出比较器2的比较结果	R/W
b5	PVD2IRS	PVD2中断复位选择	0: VCC或外部输入电压在下降经过VPVD2时产生PVD2中断 1: VCC或外部输入电压在下降经过VPVD2产生PVD复位 注: 当PVD1IRS位为“1”或者PVD2IRS位为“1”时，不能进入掉电模式，要进入掉电模式时，必须将PVD1IRS位置“0”并且将PVD2IRS位置“0”	R/W
b4	PVD2IRE	PVD2中断复位使能	0: 禁止 1: 允许 注: 请在PVD2EN位为“1”且PVD2CMPOE位为“1”状态下将PVD2IRE位写“1”	R/W
b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2	PVD1CMPOE	PVD1比较结果输出使能	0: 禁止输出比较器1的比较结果 1: 允许输出比较器1的比较结果	R/W
b1	PVD1IRS	PVD1中断复位选择	0: VCC在下降经过VPVD1时产生PVD1中断 1: VCC在下降经过VPVD1产生PVD1复位 注1: 当PVD1IRS位为“1”或者PVD2IRS位为“1”时，不能进入掉电模式，要进入掉电模式时，必须将PVD1IRS位置“0”并且将PVD2IRS位置“0”	R/W
b0	PVD1IRE	PVD1中断复位使能	0: 禁止 1: 允许 注: 请在PVD1EN位为“1”且PVD1CMPOE位为“1”状态下将PVD1IRE位写“1”	R/W

5.7.26 PVD 滤波控制寄存器(PWC_PVDFCR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	PVD2NFCKS[1:0]	PVD2NFDIS	-	PVD1NFCKS[1:0]	PVD1NFDIS		
<hr/>							
位	标记	位名	功能				读写
b7	Reserved	-	读出时为“0”,写入时写“0”				R/W
b6~b5	PVD2NFCKS	PVD2数字滤波采样时钟选择	00: 0.25 LRC周期 01: 0.5 LRC周期 10: LRC的1分频 11: LRC的2分频				R/W
			注: 只能在PVD2NFDIS位为“1”时改写该位				
b4	PVD2NFDIS	PVD2数字滤波器屏蔽	0: 数字滤波器有效 1: 数字滤波器无效				R/W
b3	Reserved	-	读出时为“0”,写入时写“0”				R/W
b2~b1	PVD1NFCKS	PVD1数字滤波采样时钟选择	00: 0.25 LRC周期 01: 0.5 LRC周期 10: LRC的1分频 11: LRC的2分频				R/W
			注: 只能在PVD1NFDIS位为“1”时改写该位				
b0	PVD1NFDIS	PVD1数字滤波器屏蔽	0: 数字滤波器有效 1: 数字滤波器无效				R/W

5.7.27 PVD 电平控制寄存器(PWC_PVDLCR)

复位值：0x00

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

位	标记	位名	功能	读写
b7	Reserved	-	读出时为“0”，写入时写“0”	R/W
b6~b4	PVD2LVL	PVD2阀值电压选择	000: 2.1V 001: 2.3V 010: 2.5V 011: 2.6V 100: 2.7V 101: 2.8V 110: 2.9V 111: 1.1V(仅在PWC_PVDCR0.EXVCCINEN=1时有效，其它情况请不要设定该值) 注：上记阈值是芯片工作在高速模式、超低速模式、停止模式时的阈值，芯片工作在超高速模式时的阈值请参考电气特性。	R/W
b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2~b0	PVD1LVL	PVD1阀值电压选择	000: 2.0V 001: 2.1V 010: 2.3V 011: 2.5V 100: 2.6V 101: 2.7V 110: 2.8V 111: 2.9V 注：上记阈值是芯片工作在高速模式、超低速模式、停止模式时的阈值，芯片工作在超高速模式时的阈值请参考电气特性。	R/W

5.7.28 PVD 中断控制寄存器(PWC_PVDICR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	PVD2NMIS	-	-	-	-	PVD1NMIS

位	标记	位名	功能	读写
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6~b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4	PVD2NMIS	PVD2中断类型选择	0: PVD2中断作为非可屏蔽中断 1: PVD2中断作为可屏蔽中断	R/W
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	PVD1NMIS	PVD1中断类型选择	0: PVD1中断作为非可屏蔽中断 1: PVD1中断作为可屏蔽中断	R/W

5.7.29 PVD 检测状态寄存器(PWC_PVDDSR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	PVD2DETFLG	PVD2MON	-	-	PVD1DETFLG	PVD1MON

位	标记	位名	功能	读写
b7~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5	PVD2DETFLG	PVD2检测标志位	0: PVD2未检测到VCC经过VPVD2 1: PVD2检测到VCC经过VPVD2 读出后对位4写0能够清除本位。注: 当PVD2EN位 为“1”且PVD2CMPOE位为“1”时,此标志位有效	R/W
b4	PVD2MON	PVD2监视及检测标志位 清除	0: VCC ≤ VPVD2或者外部输入比较电压≤ PVD2内部 基准电压 1: PVD2无效时或者VCC>VPVD2 或者外部输入比 较电压> PVD2内部基准电压 对本位写0清除PVD2DETFLG位。	R/W
b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	PVD1DETFLG	PVD1检测标志位	0: PVD1未检测到VCC经过VPVD1 1: PVD1检测到VCC经过VPVD1 读出后对位0写0能够清除本位。注: 当PVD1EN位为 “1”且PVD1CMPOE位为“1”时,此标志位有效	R/W
b0	PVD1MON	PVD1监视及检测标志位 清除	0: VCC ≤ VPVD1 1: PVD1无效时或者VCC>VPVD1 对本位写0清除PVD1DETFLG位。	R/W

6 初始化配置 (ICG)

6.1 简介

芯片复位解除后，硬件电路会读取 FLASH 地址 0x0000_0400~0x0000_041F，把数据加载到初始化配置寄存器。地址 0x0000_0408~0x0000_041F 为预留功能，请写入全 1 保证芯片正常动作。用户可以通过编程或擦除扇区 0 来修改初始化配置(Initial Config)寄存器。寄存器复位值由 FLASH 数据确定。

初始化配置寄存器地址一览表如下：

ICG_BASE_ADDR: 0x0000_0400

寄存器名	符号	偏移地址	位宽	复位值
初始化配置寄存器0	ICG0	0x000	32	不定
初始化配置寄存器1	ICG1	0x004	32	不定
初始化配置寄存器2	ICG2	0x008	32	不定
初始化配置寄存器3	ICG3	0x00C	32	不定
初始化配置寄存器4	ICG4	0x010	32	不定
初始化配置寄存器5	ICG5	0x014	32	不定
初始化配置寄存器6	ICG6	0x018	32	不定
初始化配置寄存器7	ICG7	0x01C	32	不定

表 6-1 寄存器一览表

6.2 寄存器说明

6.2.1 初始化配置寄存器 0(ICG0)

复位值：不定

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

位	标记	位名	功能	读写
b31~b29	Reserved	-	功能预留位	R
b28	WDTSLPOFF	WDT在sleep模式下计数停止	0: WDT在sleep模式下计数不停止 1: WDT在sleep模式下计数停止	R
b27~b24	WDTWDPT[3:0]	刷新允许区域计数值百 分比	WDT计数值刷新允许区间 0000: 0%~100% 0001: 0%~25% 0010: 25%~50% 0011: 0%~50% 0100: 50%~75% 0101: 0%~25%,50%~75% 0110: 25%~75% 0111: 0%~75% 1000: 75%~100% 1001: 0%~25%,75%~100% 1010: 25%~50%,75%~100% 1011: 0%~50%,75%~100% 1100: 50%~100% 1101: 0%~25%,50%~100% 1110: 25%~100% 1111: 0%~100%	R

b23~b20	WDTCKS[3:0]	WDT计数时钟	0010: PCLK3/4 0110: PCLK3/64 0111: PCLK3/128 1000: PCLK3/256 1001: PCLK3/512 1010: PCLK3/1024 1011: PCLK3/2048 1101: PCLK3/8192 其它值: 预留	R
b19~b18	WDTPERI[1:0]	WDT计数溢出周期	00: 256 周期 01: 4096 周期 10: 16384 周期 11: 65536 周期	R
b17	WDTITS	WDT中断选择	0: 中断请求 1: 复位请求	R
b16	WDTAUTS	WDT自动启动	0: 复位后, WDT自动启动 (硬件启动) 1: 复位后, WDT停止状态	R
b15~b13	Reserved	-	功能预留位	R
b12	SWDTSLPOFF	SWDT在Sleep,Stop模式下计数停止	0: SWDT在sleep,stop模式下计数不停止 1: SWDT在sleep,stop模式下计数停止	R
b11~b8	SWDTWDPT[3:0]	刷新允许区域计数值百分比	SWDT计数值刷新允许区间 0000: 0%~100% 0001: 0%~25% 0010: 25%~50% 0011: 0%~50% 0100: 50%~75% 0101: 0%~25%,50%~75% 0110: 25%~75% 0111: 0%~75% 1000: 75%~100% 1001: 0%~25%,75%~100% 1010: 25%~50%,75%~100% 1011: 0%~50%,75%~100% 1100: 50%~100% 1101: 0%~25%,50%~100% 1110: 25%~100% 1111: 0%~100%	R

b7~b4	SWDTCKS[3:0]	SWDT计数时钟	0000: SWDTCLK 0100: SWDTCLK/16 0101: SWDTCLK/32 0110: SWDTCLK/64 0111: SWDTCLK/128 1000: SWDTCLK/256 1011: SWDTCLK/2048 其它值: 预留	R
b3~b2	SWDTPERI[1:0]	SWDT计数溢出周期	00: 256 周期 01: 4096 周期 10: 16384 周期 11: 65536 周期	R
b1	SWDTITS	SWDT中断选择	0: 中断请求 1: 复位请求	R
b0	SWDTAUTS	SWDT自动启动	0: 复位后, SWDT自动启动 (硬件启动) 1: 复位后, SWDT停止状态	R

6.2.2 初始化配置寄存器 1(ICG1)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
NMII CGE NA	NFE N	NMI ENR	NMI TRG	SMPCLK[1:0]	-	-	-	-	-	-	-	BOR DIS	BOR_lev[1:0]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	HRC STOP	-	-	-	-	-	-	-	HRC FREQ SEL

位	标记	位名	功能	读写
b31	NMIICGENA	NMI管脚ICG设定使能	0: NMI管脚ICG设定使能 1: NMI管脚ICG设定禁止	R
b30	NFEN	NMI数字滤波器使能	0: 禁止数字滤波器功能 1: 许可数字滤波器功能	R
b29	NMIENR	NMI管脚中断选择	0: 禁止NMI管脚中断 1: 许可NMI管脚中断	R
b28	NMITRG	NMI管脚边沿触发	0: 下降沿 1: 上升沿	R
b27~b26	SMPCLK[1:0]	滤波采样时钟选择	0 0: PCLK3 0 1: PCLK3/8 1 0: PCLK3/32 1 1:: PCLK3/64	R
b25~b19	Reserved	-	功能预留位	R
b18	BORDIS	BOR动作选择	0: 复位后允许BOR动作 1: 复位后禁止BOR动作	R
b17~b16	BOR_lev[1:0]	BOR阈值电压选择	00: 1.9v 01: 2.0v 10: 2.1v 11: 2.3v	R
b15~b9	Reserved	-	功能预留位	R
b8	HRCSTOP	HRC振荡停止位	0: HRC振荡 1: HRC停止	R
b7~b1	Reserved	-	读出时为“1”,写入时写“1”	R
b0	HRCFREQSEL	HRC频率选择	0: 20MHz 1: 16MHz	R

6.2.3 初始化配置寄存器 n(ICGn)n=2~7

复位值：不定

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

位	标记	位名	功能	读写
b31~b0	ICGn[31:0]	-	功能预留位 需要用户设定全1以确保芯片动作正常	R

7 嵌入式 FLASH (EFM)

7.1 简介

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

7.2 主要特性

- 最大 512KByte FLASH 空间
- ICODE 总线 16Byte 预取值
- ICODE 和 DCODE 总线上的共享 64 个缓存(1KByte)
- 提供 960Byte 一次性编程区域 (OTP)
- 支持低功耗读操作
- 支持引导交换功能
- 支持安全保护及数据加密^{*1}

^{*1:} 关于 FLASH 安全保护及数据加密的具体规格，请咨询销售窗口

7.3 嵌入式 FLASH

FLASH 具有以下主要特性：

- 容量高达 512 KBytes(其中有 32Bytes 为功能保留位)
分为 64 个扇区，每个扇区为 8KBytes。
- OTP(One Time Program)区域共 1020Bytes，分为 960Bytes 的数据区和 60Bytes 的锁存区。
- 128 位宽数据读取。
- 编程单位为 4Bytes，擦除单位为 8KBytes。

在 512KB 产品中，FLASH 地址结构如下图所示。

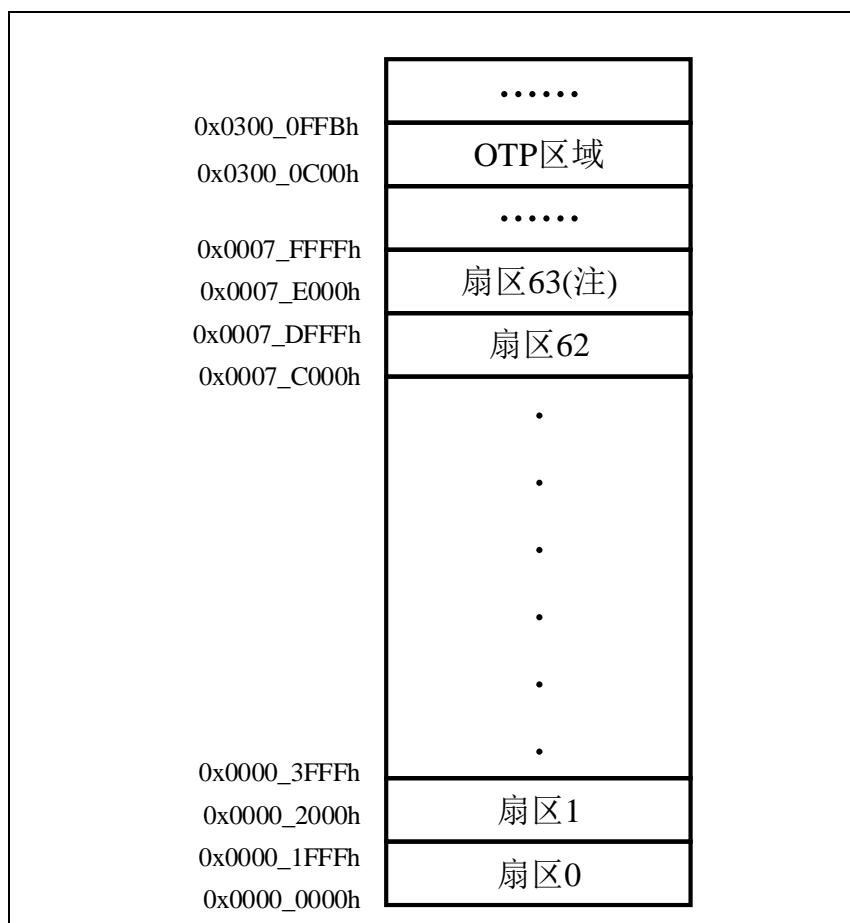


图 7-1 FLASH 地址结构 (512KB 产品)

注意：

- 扇区 63 中地址 0x0007_FFE0~0x0007_FFFF 共 32Bytes 为功能保留地址；对这 32Bytes 地址进行编程、扇区擦除、全擦除，FLASH 数据不会改变，对这些地址读，读到数据为全 1。

在 256KB 产品中，FLASH 地址结构如下图所示。

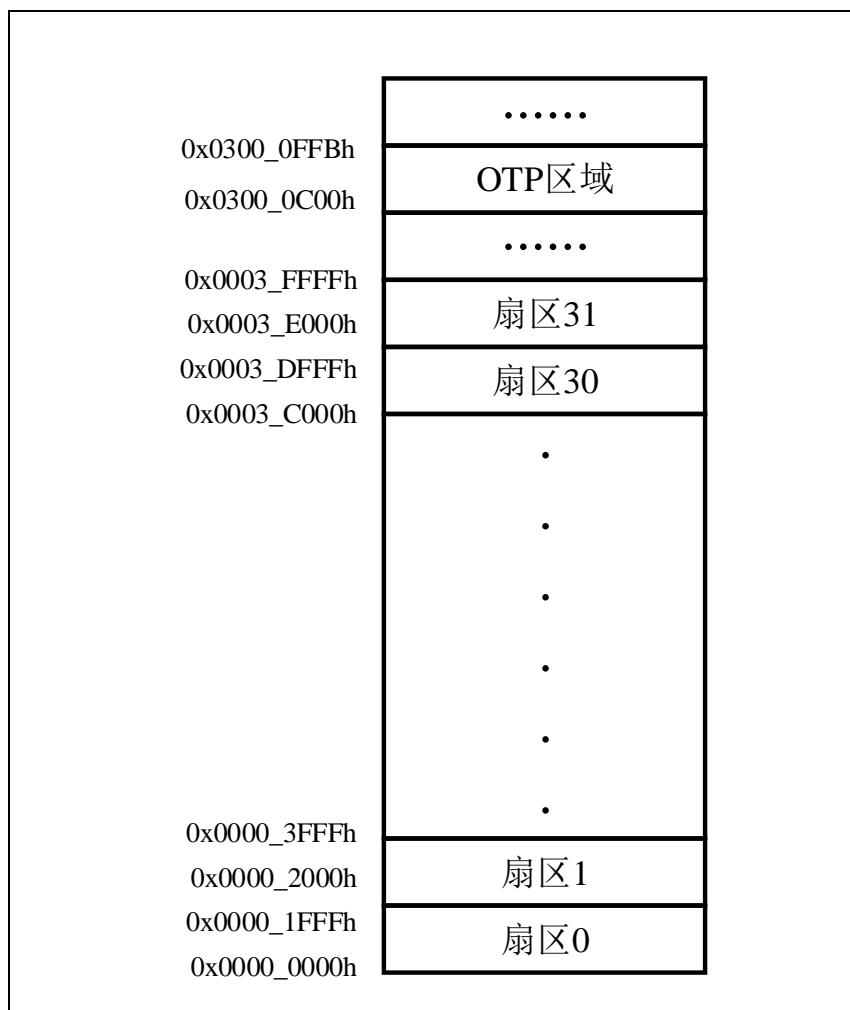


图 7-2 FLASH 地址结构 (256KB 产品)

7.4 读接口

7.4.1 CPU 时钟和 FLASH 读取时间之间的关系

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

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

CPU 频率提高步骤：

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

CPU 频率降低步骤：

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

7.4.2 FLASH 低功耗读

当 CPU 时钟频率低于 2MHz 时，用户可设定寄存器位 EFM_FRMC.SLPMD 进入低功耗读模式，以降低芯片的电流。进入低功耗读模式后，对 FLASH 的编程，擦除操作会被忽视，此模式对 FLASH 进行擦写(编程和擦除，下同)，状态位 EFM_FSR.COLERR 置位。

进入超低功耗读步骤：

1. 根据表 7-1 把新的读等待周期设定值(FLWT[3:0])按超低功耗读模式的推荐值写入寄存器 EFM_FRMC。

2. 置位寄存器 EFM_FRMC.SLPMD。

退出超低功耗读步骤:

1. 寄存器位 EFM_FRMC.SLPMD 清零。
2. 根据表 7-1 把新的读插入等待周期设定值(FLWT[3:0])按普通读模式的推荐值写入寄存器 EFM_FRMC。

CPU时钟频率(HCLK)	FRMC寄存器位FLWT[3:0]设定	
	普通读模式(SLPMD=0)	超低功耗读模式(SLPMD=1)
168MHz<F _{HCLK} ≤200MHz	FLWT[3:0]=4'b0101 插入5个等待读周期	不支持
132MHz<F _{HCLK} ≤168MHz	FLWT[3:0]=4'b0100 插入4个等待读周期	不支持
99MHz<F _{HCLK} ≤132MHz	FLWT[3:0]=4'b0011 插入3个等待读周期	不支持
66MHz<F _{HCLK} ≤99MHz	FLWT[3:0]=4'b0010 插入2个等待读周期	不支持
33MHz<F _{HCLK} ≤66MHz	FLWT[3:0]=4'b0001 插入1个等待读周期	不支持
2MHz<F _{HCLK} ≤33MHz	FLWT[3:0]=4'b0000 无等待读	不支持
1.876MHz<F _{HCLK} ≤2MHz	同上	FLWT[3:0]=4'b1111 插入15个等待读周期
1.752MHz<F _{HCLK} ≤1.876MHz	同上	FLWT[3:0]=4'b1110 插入14个等待读周期
1.628MHz<F _{HCLK} ≤1.752MHz	同上	FLWT[3:0]=4'b1101 插入13个等待读周期
1.504MHz<F _{HCLK} ≤1.628MHz	同上	FLWT[3:0]=4'b1100 插入12个等待读周期
1.38MHz<F _{HCLK} ≤1.504MHz	同上	FLWT[3:0]=4'b1011 插入11个等待读周期
1.256MHz<F _{HCLK} ≤1.38MHz	同上	FLWT[3:0]=4'b1010 插入10个等待读周期

1.132MHz < F _{HCLK} ≤ 1.256MHz	同上	FLWT[3:0]=4'b1001 插入9个等待读周期
1.008MHz < F _{HCLK} ≤ 1.256MHz	同上	FLWT[3:0]=4'b1000 插入8个等待读周期
884KHz < F _{HCLK} ≤ 1.008MHz	同上	FLWT[3:0]=4'b0111 插入7个等待读周期
760KHz < F _{HCLK} ≤ 884KHz	同上	FLWT[3:0]=4'b0110 插入6个等待读周期
636KHz < F _{HCLK} ≤ 760KHz	同上	FLWT[3:0]=4'b0101 插入5个等待读周期
512KHz < F _{HCLK} ≤ 636KHz	同上	FLWT[3:0]=4'b0100 插入4个等待读周期
388KHz < F _{HCLK} ≤ 512KHz	同上	FLWT[3:0]=4'b0011 插入3个等待读周期
264KHz < F _{HCLK} ≤ 388KHz	同上	FLWT[3:0]=4'b0010 插入2个等待读周期
140KHz < F _{HCLK} ≤ 264KHz	同上	FLWT[3:0]=4'b0001 插入1个等待读周期
F _{HCLK} ≤ 140KHz	同上	FLWT[3:0]=4'b0000 无等待读

表 7-1 CPU 时钟频率和 FLASH 读等待周期对照表

7.5 FLASH 读加速缓存

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

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

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

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

EFM_FRMC. FLWT[3:0]设定	缓存不使能(EFM_FRMC.CACHE=0)		缓存使能(EFM_FRMC.CACHE=1)	
	缓冲, 缓存命中	缓冲, 缓存不命中	缓冲, 缓存命中	缓冲, 缓存不命中
0	1	1	1	1
1	1	2	1	2
2	1	3	1	3
3	1	4	1	5
4	1	5	1	6
N(N>4)	1	N+1	1	N+2

表 7-2 FLASH 实际读周期数

7.6 FLASH 编程和擦除操作

FLASH 支持编程，扇区擦除，全擦除操作。

FLASH 编程单位是 4Bytes，编程地址末位必须以 4 对齐(末位地址为：0x0, 0x4, 0x8, 0xC)，重复编程不能确保编程的正确性。FLASH 扇区擦除单位为 8KBytes。在 FLASH

擦除编程前, 请把缓存使能无效。以下分别介绍, 编程和擦除操作的设定步骤。

7.6.1 单次编程无回读模式

单编程无回读模式设定步骤如下:

- 1) 解除 FLASH 的寄存器写保护。(EFM_FAPRT 先写 0x0123, 再写 0x3210)
- 2) 设定编程, 擦写模式许可。(EFM_FWMC.PEMODE=1)
- 3) 设定单次编程模式。(EFM_FWMC.PEMODE[2:0]=001)
- 4) 对编程地址写入 32 位数据。
- 5) 等待 FLASH 处于空闲状态。(EFM_FSR.RDY=1)
- 6) 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。
- 7) 清除编程结束标志位。(EFM_FSR.OPTEND)

7.6.2 单编程回读模式

单编程回读模式是指编程结束后自读取编程地址并和写入数据对比, 输出判断一致标志位 EFM_FSR.PGMISMTCH。

单编程回读模式设定步骤如下:

- 1) 解除 FLASH 的寄存器写保护。(EFM_FAPRT 先写 0x0123, 再写 0x3210)
- 2) 设定编程, 擦写模式许可。(EFM_FWMC.PEMODE=1)
- 3) 设定单次编程回读模式。(EFM_FWMC.PEMODE[2:0]=010)
- 4) 对编程地址写入 32 位数据。
- 5) 等待 FLASH 处于空闲状态。(EFM_FSR.RDY=1)
- 6) 判断编程自读取结果标志位。(EFM_FSR.PGMISMTCH)
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。
- 7) 清除编程结束标志位。(EFM_FSR.OPTEND)

7.6.3 连续编程操作

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, 连续编程写入间隔需小于 16us。连续编程

操作设定步骤如下：

- 1) 解除 FLASH 的寄存器写保护。(EFM_FAPRT 先写 0x0123，再写 0x3210)
- 2) 设定编程,擦写模式许可。(EFM_FWMC.PEMODE=1)
- 3) 设定连续编程模式。(EFM_FWMC.PEMOD[2:0]=011)
- 4) 把步骤 8) 9) 10) 11) 12) 13) 14) 传送到 FLASH 以外区域执行。
- 5) 跳转到传送到步骤 4) 目的地址。
- 6) 读取编程地址并判断是否和写入值一致。
- 7) 一致，表示编程成功，不一致，表示该 FLASH 地址已遭破坏，永久废弃。
- 8) 对编程地址写 32 位数据。
- 9) 等待操作结束标志位(EFM_FSR.OPTEND)置位。
- 10) 清除操作结束标志位，直至读到标志位 EFM_FSR.OPTEND 为 0。
- 11) 重复 8), 9), 10)直至所有数据写完。
- 12) 修改擦写模式控制寄存器为非连续编程模式。
- 13) 等待 FLASH 处于空闲状态。(EFM_FSR.RDY=1)
- 14) 跳转返回主程序。

注意：

- 在 FLASH 连续编程期间，如果发生对 FLASH 的读操作，将会读到不定值。读操作会使读冲突位 EFM_FSR.COLERR 置位，需设定 EFM_FSCLR 寄存器清除。

7.6.4 擦除操作

EFM 提供了扇区擦除和全擦除两种擦除方式。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；对 FLASH 进行全擦除操作后，整个 FLASH 区域所有地址(OTP 除外)数据刷新为全 1。扇区擦除和全擦除操作设定步骤如下：

- 1) 解除 FLASH 的寄存器写保护。(EFM_FAPRT 先写 0x0123，再写 0x3210)
- 2) 设定编程，擦写模式许可。(EFM_FWMC.PEMODE=1)
- 3) 设定擦除模式。(扇区擦除 EFM_FWMC.PEMOD[2:0]=100，全擦除 EFM_FWMC.PEMOD[2:0]=101)
- 4) 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。

- 5) 全擦除时对任意 FLASH 地址(地址需以 4 对齐)写入 32 位任意值。
- 6) 等待 FLASH 处于空闲状态。(EFM_FSR.RDY=1)
- 7) 清除擦除结束标志位。(EFM_FSR.OPTEND)

7.6.5 总线保持功能

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

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

7.6.6 FLASH 擦除、编程窗口保护

对 FLASH 提供窗口保护功能，只有在允许区域 FLASH 才能被扇区擦除，编程，否则发生擦写错误中断。编程、扇区擦除操作时，硬件电路将预先判断是否在允许区域内，全擦除模式不受窗口保护限制。窗口的起始位置和结束位置由寄存器 EFM_FPMTEW，EFM_FPMTSW 设定。

具体保护功能如下：

- 寄存器 EFM_FPMTEW=寄存器 EFM_FPMTSW 整个 FLASH 区域都可擦除，编程。
- 寄存器 EFM_FPMTEW>寄存器 EFM_FPMTSW 可擦除，编程区域在两者之间。
- 寄存器 EFM_FPMTEW<寄存器 EFM_FPMTSW 整个 FLASH 区域都不可擦除，编程。

7.6.7 中断

EFM 模块共有 3 个中断，分别是 PE（编程/擦除）错误中断，读写冲突中断和操作结束中断。

当 FLASH 发生 PE 错误中断置位后，不能对 FLASH 进行编程/擦除/全擦除。必须对标志位清零后才能再次对 FLASH 进行编程/擦除/全擦除。

1. PE 错误中断 EFM_PEERR:

置位:

- 编程地址非以 4 对齐或者数据大小非 32 位(PGSZERR=1)。
- 对 FLASH 窗口保护内地址执行编程，扇区擦除操作(PEPRTERR=1)。
- 未设定擦写模式时，执行对 FLASH 写操作(PEWERR=1)。
- 单编程回读模式时，编程地址自读值与写入值不一致(PGMISMTCH=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 连续编程模式发生 FLASH 读操作。
- FLASH 停止模式时发生 FLASH 读写操作。
- FLASH 编程，擦除未结束前，发生 FLASH 读写操作。
- 在低功耗读模式发生对 FLASH 写操作。

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。
- 擦除模式：扇区擦除，全擦除结束。

清零:

寄存器 EFM_FSCLR 对应清除位置 1，状态位清零。

7.7 一次性可编程字节

下表所示为 OTP 区域一次性可编程地址构成，分为 960Bytes 的数据区和 60Bytes 的锁存区。

块名	OTP数据块地址	OTP锁定地址
块0	0x0300_0C00~0x0300_0C3F	0x0300_0FC0~0x0300_0FC3
块1	0x0300_0C40~0x0300_0C7F	0x0300_0FC4~0x0300_0FC7
块2	0x0300_0C80~0x0300_0CBF	0x0300_0FC8~0x0300_0FCB
块3	0x0300_0CC0~0x0300_0cff	0x0300_0FCC~0x0300_0FCF
块4	0x0300_0D00~0x0300_0D3F	0x0300_0FD0~0x0300_0FD3
块5	0x0300_0D40~0x0300_0D7F	0x0300_0FD4~0x0300_0FD7
块6	0x0300_0D80~0x0300_0DBF	0x0300_0FD8~0x0300_0FDB
块7	0x0300_0DC0~0x0300_0dff	0x0300_0FDC~0x0300_0FDF
块8	0x0300_0E00~0x0300_0E3F	0x0300_0FE0~0x0300_0FE3
块9	0x0300_0E40~0x0300_0E7F	0x0300_0FE4~0x0300_0FE7
块10	0x0300_0E80~0x0300_0EBF	0x0300_0FE8~0x0300_0FEB
块11	0x0300_0EC0~0x0300_0EFF	0x0300_0FEC~0x0300_0FEF
块12	0x0300_0F00~0x0300_0F3F	0x0300_0FF0~0x0300_0FF3
块13	0x0300_0F40~0x0300_0F7F	0x0300_0FF4~0x0300_0FF7
块14	0x0300_0F80~0x0300_0FBF	0x0300_0FF8~0x0300_0FFB

表 7-3 OTP 地址构成

OTP 区域分为 15 个 64 字节的数据块，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；当锁存地址数据为全 0 时，对应的 OTP 区域数据不可编程。所有 OTP 数据块和锁存地址均无法擦除。

7.8 引导交换

用户如要升级引导程序，需要对扇区 0(0x0000_0000~0x0000_1FFF)进行擦写，如擦写时遇到不可期的意外(如复位，掉电)时，有可能会导致整个芯片不能正常启动。针对 512KB 产品，EFM 对此提供了启动扇区交换功能(256KB 产品无此功能)。在对扇区 0 进行擦除前，预先把新的引导程序写入扇区 1(0x0000_2000~0x0000_3FFF)，然后对 EFM 地址 0x0007_FFDC 进行编程数据 0xFFFF_4321。MCU 复位后 CPU 就从扇区 1 启动新的引导程序，此时再对扇区 0 进行擦除，重新编程新的用户程序。

启动交换操作请参考图 7-3。

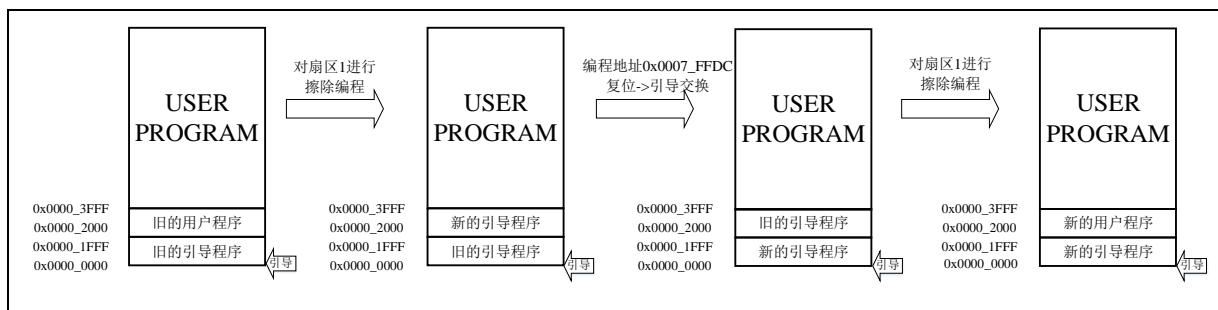


图 7-3 启动扇区交换功能 1

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 0x0007_FFDC 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)，需要对扇区 63 进行扇区擦除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，然后对扇区 63 进行擦除。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。操作流程如图 7-4。

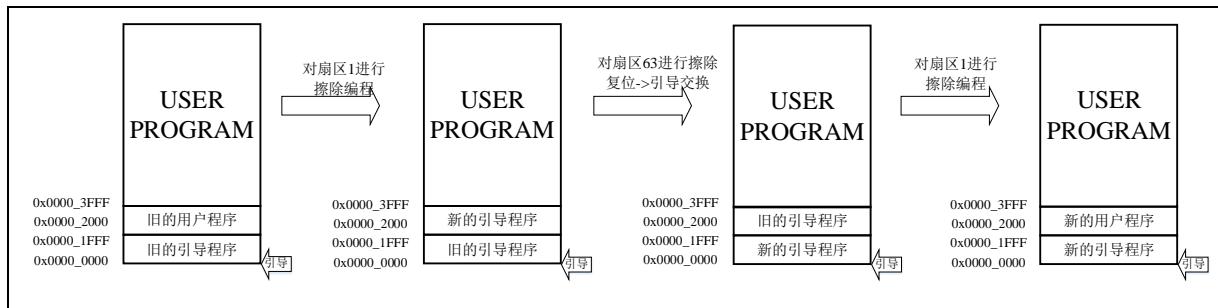


图 7-4 启动交换功能 2

7.9 寄存器说明

EFM_BASE_ADDR: 0x4001_0400

寄存器说明	寄存器名	偏移量	位宽	复位值
FLASH访问保护寄存器	EFM_FAPRT	0x0000	32	0x0000_0000
FLASH停止寄存器	EFM_FSTP	0x0004	32	0x0000_0000
FLASH读模式寄存器	EFM_FRMC	0x0008	32	0x0000_0000
FLASH擦写模式寄存器	EFM_FWMC	0x000C	32	0x0000_0000
FLASH状态寄存器	EFM_FSR	0x0010	32	0x0000_0100
FLASH状态清除寄存器	EFM_FSCLR	0x0014	32	0x0000_0000
FLASH中断许可寄存器	EFM_FITE	0x0018	32	0x0000_0000
FLASH引导交换状态寄存器	EFM_FSWP	0x001C	32	不定
FLASH改写允许区域起始地址	EFM_FPMTSW	0x0020	32	0x0000_0000
FLASH改写允许区域结束地址	EFM_FPMTEW	0x0024	32	0x0000_0000
FLASHuniqueID寄存器	EFM_UQID1	0x0050	32	不定
FLASHuniqueID寄存器	EFM_UQID2	0x0054	32	不定
FLASHuniqueID寄存器	EFM_UQID3	0x0058	32	不定

表 7-4 寄存器一览表

7.9.1 访问保护寄存器 (EFM_FAPRT)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31-b16	Reserved	-	读出时为“0”，写入时写“0”	R
b15-b0	FAPRT[15:0]	EFM寄存器写保护	访问EFM寄存器保护寄存器。 解除方法：对FAPRT先写“16位数据0x0123”再写“16位 数据0x3210”。 在解除保护状态下，写入任意数据，EFM寄存器再次 进入保护状态。 EFM寄存器访问保护有效时，该寄存器读出值为 0x0000_0000。 EFM寄存器访问保护无效时，该寄存器读出值为 0x0000_0001。	R/W

7.9.2 FLASH 停止寄存器 (EFM_FSTP)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31-b1	Reserved	-	读出时为“0”，写入时写“0”	R
b0	FSTP	FLASH停止模式控制	0: FLASH活动状态 1: FLASH处于停止模式 当寄存器位由1设为0后，请在确认FSR.RDY位 为1后，进行FLASH访问。	R/W

7.9.3 读模式寄存器 (EFM_FRMC)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b25	Reserved	-	读出时为“0”,写入时写“0”	R
b24	CRST	缓存复位位	0: 缓存数据不复位 1: 复位缓存数据	R/W
b23~b17	Reserved	-	读出时为“0”,写入时写“0”	R
b16	CACHE	缓存许可位	0: 关闭缓存功能 1: 缓存功能使能	R/W
b15~b9	Reserved	-	读出时为“0”,写入时写“0”	R
b8	LVM	超低速运行模式	0: 关闭超低速运行模式 1: 打开超低速运行模式	R/W
b7-b4	FLWT[3:0]	FLASH读插入的等待周期	0000b: 不插入等待周期 0001b: 插入1个等待周期 0010b: 插入2个等待周期 1110b: 插入14个等待周期 1111b: 插入15个等待周期	R/W
b3~b1	Reserved	-	读出时为“0”,写入时写“0”	R
b0	SLPMD	超低功耗读	0: 普通CPU读模式 1: 超低功耗读模式	R/W

7.9.4 擦写模式寄存器 (EFM_FWMC)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b9	Reserved	-	读出时为“0”,写入时写“0”	R
b8	BUSHLDCTL	FLASH擦除, 编程期间总线控制	0: FLASH编程擦除期间, 总线被占用。 1: FLASH编程擦除期间, 总线释放。	R/W
b7	Reserved	-	读出时为“0”,写入时写“0”	R
b6~b4	PEMOD[2:0]	FLASH擦除, 编程模式	000: 只读模式 001: 单编程模式 010: 单编程回读模式 011: 连续编程模式 100: 扇区擦除模式 101: 全擦除模式 110: 只读模式 111: 只读模式 只有在PEMODE=1的情况下, PEMOD[2:0]才能被写入。	R/W
b3~b1	Reserved	-	读出时为“0”,写入时写“0”	R
b0	PEMODE	FLASH擦除, 编程许可模式	0: PEMOD[2:0]改写不许可 1: PEMOD[2:0]改写许可	R/W

7.9.5 状态寄存器 (EFM_FSR)

复位值: 0x0000_0100

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	RDY	-	-	COL ERR	OPT ND	PGM ISMT CH	PGS ZERR	PEPR TERR	PEW ERR

位	标记	位名	功能	读写
b31~b9	Reserved	-	读出时为“0”,写入时写“0”	R
b8	RDY	FLASH忙/空闲状态	0: FLASH忙状态 1: FLASH空闲状态	R
b7~b6	Reserved	-	读出时为“0”,写入时写“0”	R
b5	COLERR	FLASH读写访问错误 标志位	0: FLASH读写访问正常 1: FLASH读写访问错误 使该位置位的FLASH读写访问操作将会被忽视。	R
b4	OPTEND	操作结束标志位	0: 没有对FLASH进行擦写或者FLASH擦写中 1: 对FLASH擦写结束, 置位条件: 编程/擦除/全擦除操作结束	R
b3	PGMISMTC	单编程回读值不一致 标志位	0: 单编程回读值一致 1: 单编程回读值不一致	R
b2	PGSZERR	编程地址和大小不对 齐标志位	0: 编程地址和大小对齐 1: 编程地址和大小不对齐	R
b1	PEPRERR	对保护地址编程/擦除 错误标志位	0: 编程/擦除地址为允许改写区域 1: 对保护窗口地址进行编程/擦除动作	R
b0	PEWERR	擦写模式错误标志位	0: 在擦写许可模式下擦写FLASH 1: 在擦写不许可模式下擦写FLASH	R

7.9.6 状态清除寄存器 (EFM_FSCLR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	COL ERRC LR	OPTE NDC LR	PG MIS MTC HCL R	PGS ZERR CLR	PEPR TERR CLR	PEW ERR CLR

位	标记	位名	功能	读写
b31~b6	Reserved	-	读出时为“0”,写入时写“0”	R
b5	COLERRCLR	清除读写冲突错误 标志位	0: 不发生清除动作 1: 清除读写冲突错误 该位读时, 始终为0。	R/W
b4	OPTENDCLR	清除操作结束标志	0: 不发生清除动作 1: 清除操作结束标志 该位读时, 始终为0。	R/W
b3	PGMISMTCHCLR	清除编程回读不一致标志位	0: 不发生清除动作 1: 清除编程回读不一致标志位 该位读时, 始终为0。	R/W
b2	PGSZERRCLR	清除编程地址和大小不对齐标志位	0: 不发生清除动作 1: 清除不对齐错误标志位 该位读时, 始终为0。	R/W
b1	PEPRTERRCLR	清除对保护地址编程/擦除错误标志位	0: 不发生清除动作 1: 清除编程/擦除错误 该位读时, 始终为0。	R/W
b0	PEWERRCLR	擦写模式错误标志位	0: 不发生清除动作 1: 清除模式写错误 该位读时, 始终为0。	R/W

7.9.7 中断许可寄存器 (EFM_FITE)

复位值: 0x0000_0000

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

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

7.9.8 引导交换状态寄存器 (EFM_FSWP)

复位值：不定

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

位	标记	位名	功能	读写
b31-b1	Reserved	-	读出时为“0”，写入时写“0”	R
b0	FSWP	扇区0和扇区1地址交换位	0: 扇区0和扇区1地址交换 复位后，CPU从扇区1启动。 1: 扇区0和扇区1地址不交换 复位后，CPU从扇区0启动。 寄存器初始值由FLASH地址 0x0007_FFDC~0x0007_FFDF的决定，其地址数 据为0xFFFF_4321时，复位后，FSWP寄存器位 FSWP初值为0，否则为1。	R

7.9.9 FLASH 窗口保护起始地址寄存器 (EFM_FPMTSW)

复位值：0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FPMTSW[18:16]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FPMTSW[15:0]															

位	标记	位名	功能	读写
b31-b19	Reserved	-	读出时为“0”，写入时写“0”	R
b18-b0	FPMTSW[18: 0]	保护窗口起始地址	FLASH保护窗口起始地址	R/W

7.9.10 FLASH 窗口保护结束地址寄存器 (EFM_FPMTEW)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FPMTEW[18:16]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FPMTEW[15:0]															

位	标记	位名	功能	读写
b31-b19	Reserved	-	读出时为“0”，写入时写“0”	R
b18-b0	FPMTEW[18: 0]	保护窗口结束地址	FLASH保护窗口的结束地址	R/W

7.9.11 UNIQUE ID 寄存器 (EFM_UQID1)

复位值: 不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
UQID1[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UQID1[15:0]															

位	标记	位名	功能	读写
b31-b0	UQID1[31:0]	唯一码	芯片唯一码	R

7.9.12 UNIQUE ID 寄存器 (EFM_UQID2)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16										
UQID2[31:16]																									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0										
UQID2[15:0]																									
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; width: 10%;">位</th><th style="text-align: left; width: 30%;">标记</th><th style="text-align: left; width: 30%;">位名</th><th style="text-align: left; width: 30%;">功能</th><th style="text-align: right; width: 10%;">读写</th></tr> </thead> <tbody> <tr> <td>b31-b0</td><td>UQID2[31:0]</td><td>唯一码</td><td>芯片唯一码</td><td style="text-align: right;">R</td></tr> </tbody> </table>																位	标记	位名	功能	读写	b31-b0	UQID2[31:0]	唯一码	芯片唯一码	R
位	标记	位名	功能	读写																					
b31-b0	UQID2[31:0]	唯一码	芯片唯一码	R																					

7.9.13 UNIQUE ID 寄存器 (EFM_UQID3)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16										
UQID3[31:16]																									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0										
UQID3[15:0]																									
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; width: 10%;">位</th><th style="text-align: left; width: 30%;">标记</th><th style="text-align: left; width: 30%;">位名</th><th style="text-align: left; width: 30%;">功能</th><th style="text-align: right; width: 10%;">读写</th></tr> </thead> <tbody> <tr> <td>b31-b0</td><td>UQID3[31:0]</td><td>唯一码</td><td>芯片唯一码</td><td style="text-align: right;">R</td></tr> </tbody> </table>																位	标记	位名	功能	读写	b31-b0	UQID3[31:0]	唯一码	芯片唯一码	R
位	标记	位名	功能	读写																					
b31-b0	UQID3[31:0]	唯一码	芯片唯一码	R																					

7.10 注意事项

1. 在 FLASH 进行擦写时，发生复位、擦写操作会强制停止，FLASH 数据将得不到保证。用户需在复位解除后对地址擦除后再次进行操作。
2. 擦写操作如果会涉及到缓存中的数据，请在擦写操作结束后复位缓存回路(FRMC.CRST0=1)。
3. 编程，擦除操作之前请把缓存设定为无效(FRMC.CACHE=0)。
4. 一次性可编程区域编程操作不受窗口保护寄存器(FPMTSW,FPMTEW)控制。
5. 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）。
6. 编程、擦除时设定总线释放(FWMC.BUSHLDCTL=1)时，编程、擦除期间如需响应中断，请把中断向量和中断子程序设定到 RAM。

8 内置 SRAM (SRAM)

8.1 简介

本产品带有 4KB 掉电模式保持 SRAM (Ret_SRAM) 和 188KB 系统 SRAM (SRAMH/SRAM1/ SRAM2/SRAM3)。

SRAM 可按照字节、半字（16 位）或全字（32 位）访问。读写操作以 CPU 速度执行，可插入等待周期。读写访问的等待周期设定和 CPU 时钟频率的关系如表 8-1 所示。各个 SRAM 的读写访问的等待周期由 SRAM 等待控制寄存器 (SRAM_WTCR) 设定。

等待周期(CPU访问周期)	访问高速SRAM (SRAMH) 允许的CPU时钟频率范围	访问其它SRAM (SRAM1,2,3,Ret_SRAM) 允许的CPU时钟频率范围
0wait(1个CPU周期访问)	0~200MHz	0~100MHz
1wait(2个CPU周期访问)	0~200MHz	0~200MHz
2wait(3个CPU周期访问)	0~200MHz	0~200MHz
3wait(4个CPU周期访问)	0~200MHz	0~200MHz
4wait(5个CPU周期访问)	0~200MHz	0~200MHz
5wait(6个CPU周期访问)	0~200MHz	0~200MHz
6wait(7个CPU周期访问)	0~200MHz	0~200MHz
7wait(8个CPU周期访问)	0~200MHz	0~200MHz

表 8-1 SRAM 读写访问的等待周期设定和 CPU 时钟频率的关系

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

SRAM3 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/SRAM1/SRAM2/Ret_SRAM 带有奇偶校验 (Even-parity check)，每字节数据带有一位校验位。SRAM 详细定义见表 8-2。

名称	容量	地址范围	校验方式
SRAM1	64KB	0x2000_0000~0x2000_FFFF	Even-parity check
SRAM2	64KB	0x2001_0000~0x2001_FFFF	Even-parity check
SRAM3	28KB	0x2002_0000~0x2002_6FFF	ECC check
Ret_SRAM	4KB	0x200F_0000~0x200F_0FFF	Even-parity check
SRAMH	32KB	0x1FFF_8000~0x1FFF_FFFF	Even-parity check

表 8-2 SRAM 空间分配

注意：

- 在使用 SRAM3 作为堆栈空间时，须将 SRAM3 的等待时间设置为 1wait，即 2 个 CPU 周期以上进行访问。
- 在允许产生 RAM 奇偶校验错误产生 NMI 中断和复位的情况下，当存取数据时，必须对所用 RAM 空间以字为单位进行初始化；当从 SRAMH 空间执行指令时，必须对所用 RAM 空间+3 字的区域以字为单位进行初始化。
- 在允许产生 RAM ECC 校验错误产生 NMI 中断和复位的情况下，当存取数据时，必须对所用 RAM 空间以字为单位进行初始化。
- SRAMH 和 SRAM1 的边界不支持非对齐访问，使用上必须避免对 0x1FFF_FFFD/0x1FFF_FFFE/0x1FFF_FFFF 地址发起 32bit 访问，对 0x1FFF_FFFF 地址发起 16bit 访问。

8.2 寄存器说明

寄存器名	起始地址	复位值
SRAM等待控制寄存器 (SRAM_WTCR)	0x4005_0800	0x0000_0000
SRAM等待保护寄存器 (SRAM_WTPR)	0x4005_0804	0x0000_0000
SRAM校验控制寄存器 (SRAM_CKCR)	0x4005_0808	0x0000_0000
SRAM校验保护寄存器 (SRAM_CKPR)	0x4005_080C	0x0000_0000
SRAM校验状态寄存器 (SRAM_CKSR)	0x4005_0810	0x0000_0000

8.2.1 SRAM 等待控制寄存器 (SRAM_WTCR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Rev	SRAMR_ WWT[2: 0]	Rev	SRAMR_ WWT[2: 0]	Rev	SRAMH_ WWT[2: 0]	Rev	SRAMH_ WWT[2: 0]	Rev	SRAM12_ WWT[2: 0]	Rev	SRAM12_ WWT[2: 0]	Rev	SRAM12_ WWT[2: 0]	Rev	SRAM12_ WWT[2: 0]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Rev	SRAM3_ WWT[2: 0]	Rev	SRAM3_ WWT[2: 0]	Rev	SRAM12_ WWT[2: 0]										

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“0”,写入时写“0”	R/W
b30~b28	SRAMR_ WWT[2: 0]	Ret_SRAM写入周期选择	000b: 1周期写 001b: 2周期写 010b: 3周期写 011b: 4周期写 100b: 5周期写 101b: 6周期写 110b: 7周期写 111b: 8周期写	R/W
b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26~b24	SRAMR_ WWT[2: 0]	Ret_SRAM读取周期选择	000b: 1周期读 001b: 2周期读 010b: 3周期读 011b: 4周期读 100b: 5周期读 101b: 6周期读 110b: 7周期读 111b: 8周期读	R/W
b23	Reserved	-	读出时为“0”,写入时写“0”	R/W
b22~b20	SRAMH_ WWT[2: 0]	SRAMH写入周期选择	000b: 1周期写 001b: 2周期写 010b: 3周期写 011b: 4周期写 100b: 5周期写 101b: 6周期写 110b: 7周期写 111b: 8周期写	R/W
b19	Reserved	-	读出时为“0”,写入时写“0”	R/W
b18~b16	SRAMH_ WWT[2: 0]	SRAMH读取周期选择	000b: 1周期读 001b: 2周期读	R/W

			010b: 3周期读 011b: 4周期读 100b: 5周期读 101b: 6周期读 110b: 7周期读 111b: 8周期读	
b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14~b12	SRAM3_	SRAM3写入周期选择	000b: 1周期写 001b: 2周期写 010b: 3周期写 011b: 4周期写 100b: 5周期写 101b: 6周期写 110b: 7周期写 111b: 8周期写	R/W
	WWT[2: 0]			
b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10~b8	SRAM3_	SRAM3读取周期选择	000b: 1周期读 001b: 2周期读 010b: 3周期读 011b: 4周期读 100b: 5周期读 101b: 6周期读 110b: 7周期读 111b: 8周期读	R/W
	WWT[2: 0]			
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6~b4	SRAM12_	SRAM1和SRAM2写入周期选择	000b: 1周期写 001b: 2周期写 010b: 3周期写 011b: 4周期写 100b: 5周期写 101b: 6周期写 110b: 7周期写 111b: 8周期写	R/W
	WWT[2: 0]			
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~b0	SRAM12_	SRAM1和SRAM2读取周期选择	000b: 1周期读 001b: 2周期读 010b: 3周期读 011b: 4周期读 100b: 5周期读 101b: 6周期读 110b: 7周期读 111b: 8周期读	R/W
	WWT[2: 0]			

8.2.2 SRAM 等待保护寄存器 (SRAM_WTPR)

复位值: 0x0000_0000

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

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

WTPRC: 对 SRAMWTCR 寄存器写操作进行控制。当 WTPRC 被置为 1, 对于 SRAMWTCR 的写入操作是允许的, 如果其被置为 0 时, 不能对 SRAMWTCR 进行写操作。当写入此位时, 必须同时对 WTPRKW[6: 0]写入 0x3B。

8.2.3 SRAM 校验控制寄存器 (SRAM_CKCR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Rev				ECCMOD [1: 0]		Rev				ECC OAD					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Rev															PYOAD
<hr/>															
位	标记	位名	功能	读写											
b31~b26	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b25~b24	ECC	SRAM3的ECC校验允许	00: 禁止ECC校验功能	R/W											
	MOD[1: 0]	位	01: 若1位错误, ECC纠错, 不产生1位错误标志, 不产生中断/复位; 若2位错误, ECC检错, 产生2位错误标志, 产生中断/复位. 10: 若1位错误, ECC纠错, 产生1位错误标志, 不产生中断/复位; 若2位错误, ECC检错, 产生2位错误标志, 产生中断/复位. 11: 若1位错误, ECC纠错, 产生1位错误标志, 产生中断/复位; 若2位错误, ECC检错, 产生2位错误标志, 产生中断/复位.												
b23~b17	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b16	ECCOAD	ECC校验	0: Non-maskable interrupt	R/W											
		出错后操作	1: Reset												
b15~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W											
b0	PYOAD	奇偶校验	0: Non-maskable interrupt	R/W											
		出错后操作	1: Reset												

注意:

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

8.2.4 SRAM 校验保护寄存器 (SRAM_CKPR)

复位值: 0x0000_0000

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

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

CKPRC: 对 SRAMCKCR 寄存器的写进行控制。当 CKPRC 被置为 1, 对于 SRAMCKCR 的写入操作是允许的, 如果其被置为 0 时, 不能对 SRAMCKCR 进行写操作。当写入此位时, 必须同时对 CKPRKW[6: 0]写入 0x3B。

8.2.5 SRAM 校验状态寄存器 (SRAM_CKSR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Rev															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Rev															
SRAMR_ PYERR	SRAMH_ PYERR	SRAM12_ PYERR	SRAM3_ 2ERR	SRAM3_ 1ERR											

位	标记	位名	功能	读写
b31~b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4	SRAMR_PYERR	Ret_SRAM奇偶 校验错误标志	0: 无奇偶校验错误发生 1: 有奇偶校验错误发生	R/W (注1)
b3	SRAMH_PYERR	SRAMH奇偶 校验错误标志	0: 无奇偶校验错误发生 1: 有奇偶校验错误发生	R/W (注1)
b2	SRAM12_PYERR	SRAM1和SRAM2奇偶 校验错误标志	0: 无奇偶校验错误发生 1: 有奇偶校验错误发生	R/W (注1)
b1	SRAM3_2ERR	SRAM3发生ECC 2-bit错误 标志	0: 无2-bit ECC错误发生 1: 有2-bit ECC错误发生	R/W (注1)
b0	SRAM3_1ERR	SRAM3发生ECC 1-bit错误 标志	0: 无1-bit ECC错误发生 1: 有1-bit ECC错误发生	R/W (注1)

注意:

- 写1清0。

9 通用 IO (GPIO)

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 10 个 I/O。
- GPIO (General Purpose Input Output) 通用输入输出。
- NOD/POD (Nmos/Pmos Open Drain) NMOS/PMOS 开漏输出模式。

9.1 简介

主要特性：

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

9.2 端口功能概要

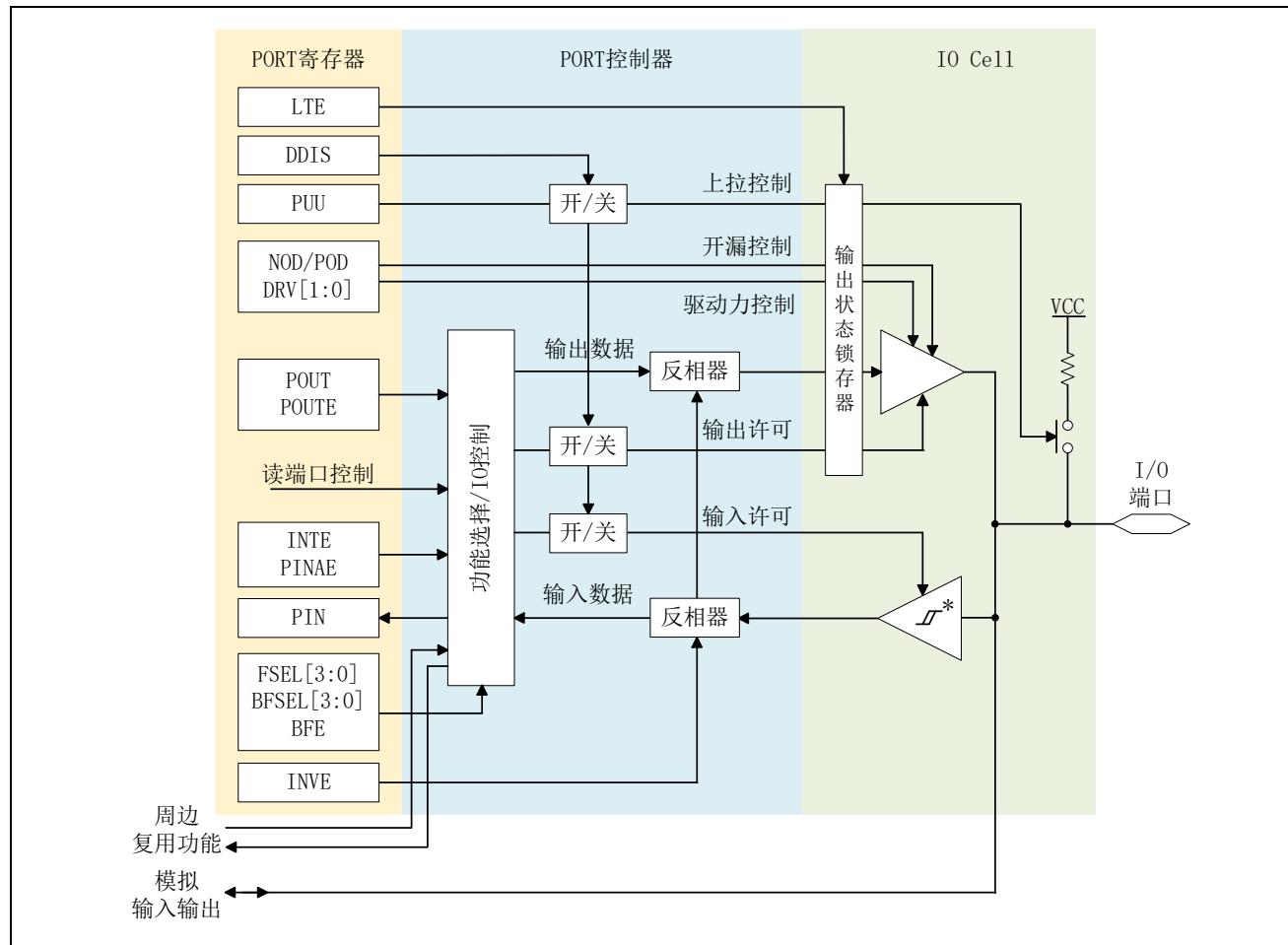


图 9-1 端口基本结构示意图

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

9.3 动作说明

9.3.1 通用输入输出 GPIO 功能

通用输入功能 GPI:

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

为了降低功耗，在 I/O 未选择周边功能时，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。I/O 的这种工作模式会使其输入迟滞特性不能正常发挥，因为每次读操作时输入 MOS 都是从关闭状态的高电平往高或低（根据 I/O 的输入值）电平切换，所以高电平变低电平的阈值 VIL 正常，但低电平变高电平的阈值 VIH 无法发挥。为了让 I/O 正常发挥输入迟滞特性，需要让 MOS 一直处于打开状态。这可以通过设置寄存器 PINAER.PINAE[x]为 1，或者设置 PFSRxy 选择一个周边功能（GPO 除外）来实现。

由于 I/O 输入存在延迟，当系统运行在高速时钟下，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[1:0]，插入若干等待周期。具体参考寄存器 PCCR 说明。

通用输出功能 GPO:

除输入专用的 PB11 端口，其他 I/O 端口都具有通用输出 GPO 功能。通过设置端口功能选择寄存器 PFSRxy.FSEL[5:0]为 0x0 可以有效 GPO 功能。

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

上面的寄存器都是 16 个 PORT 一组一起操作的。为了方便对单个 I/O 的控制，也可通过设置 PCRxy.POUTE 来允许或者禁止 I/O 的输出，PCRxy.POUTE 寄存器位与

POERx.POUTE[y]等价。可通过设置 PCRxy.POUT 来控制 I/O 的输出值，PCRxy.POUT 寄存器位与 PODRx.POUT[y]等价。PCRxy 适合用于控制单个 PORT，POERx/PODRx 适合于控制 16 位整组 PORT。

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

注意：

- 端口 PB11 与 MD 复用，为输入专用端口，无输出功能。

9.3.2 周边功能

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

JTAG/SWD 调试功能，使用寄存器 PSPCR 选择。PSPCR.SPFE[z], z=0~4 为 1 时，对应端口的 PFSRxy.FSEL[5:0]寄存器位无效，即 SPFE 优先级高于 FSEL。PSPCR 寄存器初始值为 0x1F，JTAG/SWD 功能有效。如果要将这些端口设置为 JTAG/SWD 以外的功能，需要先对对应 SPFE[z]位写 0。

9.3.3 双周边功能

有些应用情况，需要将一个端口同时设成两种功能。这种情况可以先由 PFSRxy.FSEL[5:0]选择好一个功能，再通过设置 PFSRxy.BFE 为 1，并设置公共控制寄存器 PCCR.BFSEL[3:0]选择第二个功能。例如：设置 PFSRxy.FSEL[5:0]=0x2，PCCR.BFSEL[3:0]=0x5，PFSRxy.BFE=0x1，则 Pxy 上的功能 2 与功能 5 就将同时有效。禁止在同一端口上同时有效 2 个输出功能。

9.3.4 Event Port 输入输出功能

支持 4 组 Event Port，每组 16 个端口。Event Port1 包含 EVNTP100~EVNTP115，Event Port2 包含 EVNTP200~EVNTP215，以此类推。EVNTPmn (m=1~4, n=0~15) 端口可以作为触发源，根据端口输入生成事件来触发其它周边设备（如 TIMER, ADC, DMA

等) 开始特定的动作。也可以作为被触发对象，接受事件，自动输入或输出。

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

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

使用 Event Port 功能时，需要先将功能时钟控制 0 寄存器(PWC_FCG0)的自动运行系统 AOS 功能使能位设置为有效。

注意：

- 端口 PB11 与 MD 复用，为输入专用端口，故 EVNTP211 无输出功能。

9.3.5 外部中断 EIRQ 输入功能

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。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。

另外，外部非可屏蔽中断 NMI 与 PB11/MD 端口复用。

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

9.3.6 模拟功能

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

9.3.7 通用控制

1. 上/下拉电阻

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

当 I/O 端口选择 I2Cx_SCL/I2Cx_SDA 功能时，将无视寄存器 PUU 的设定，内部上拉功能被强制无效。

PA11, PA12 与 USBFS_DM, USBFS_DP 引脚复用，内置约 $400\text{K}\Omega$ 的下拉电阻，且一直有效。

2. 驱动能力控制

各 I/O 端口都具有高、中、低 3 档驱动能力可调，可根据需要设置寄存器 PFRS_{xy}.DRV[1:0]。

本功能只在端口处于输出状态时才有效。

3. 开漏输出模式

设置 PFRS_{xy}.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。

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

9.4 寄存器说明

BASE_ADDR: 0x4005_3800

寄存器名	符号	偏移地址	位宽	复位值
通用输入数据寄存器	PIDRx	0x00+0x10*n ^{*1}	16/32	0xFFFF
通用输出数据寄存器	PODRx	0x04+0x10*n	16/32	0x0000
通用输出许可寄存器	POERx	0x06+0x10*n	16/32	0x0000
通用输出置位寄存器	POSRx	0x08+0x10*n	16/32	0x0000
通用输出复位寄存器	PORRx	0x0A+0x10*n	16/32	0x0000
通用输出翻转寄存器	POTRx	0x0C+0x10*n	16/32	0x0000
特殊控制寄存器	PSPCR	0x3F4	16/32	0x001F
公共控制寄存器	PCCR	0x3F8	16/32	0x4000
输入控制寄存器	PINAER	0x3FA	16/32	0x0000
写保护寄存器	PWPR	0x3FC	16/32	0x0000
通用控制寄存器	PCRxy	0x400+0x40*n+0x4*y	16/32	0x0X00 ^{*2}
功能选择寄存器	PFSRxy	0x402+0x40*n+0x4*y	16/32	0x0000

表 9-1 PORT 寄存器一览 1

注 *1: 地址计算公式中 x=A~E,H 对应 n=0~4,5

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

BASE_ADDR: 0x4001_0800

寄存器名	符号	偏移地址	位宽	复位值
Event Port方向选择寄存器	PEVNTDIRm	0x100+0x1C*(m-1)	32	0x0000_0000
Event Port输入数据寄存器	PEVNTIDRm	0x104+0x1C*(m-1)	32	0x0000_0000
Event Port输出数据寄存器	PEVNTODRm	0x108+0x1C*(m-1)	32	0x0000_0000
Event Port输出数据复位寄存器	PEVNTORRm	0x10C+0x1C*(m-1)	32	0x0000_0000
Event Port输出数据置位寄存器	PEVNTOSRm	0x110+0x1C*(m-1)	32	0x0000_0000
Event Port上升沿输入许可寄存器	PEVNTRISRm	0x114+0x1C*(m-1)	32	0x0000_0000
Event Port下降沿输入许可寄存器	PEVNTFALRm	0x118+0x1C*(m-1)	32	0x0000_0000
Event Port输入滤波控制寄存器	PEVNTNFCR	0x170	32	0x0000_0000

表 9-2 PORT 寄存器一览 2

注: m=1~4

9.4.1 通用输入寄存器 (PIDRx)

复位值: 0xXXXX

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

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

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

9.4.2 通用输出数据寄存器 (PODRx)

复位值: 0x0000

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

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

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

9.4.3 通用输出许可寄存器 (POERx)

复位值: 0x0000

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

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

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

9.4.4 通用输出置位寄存器 (POSRx)

复位值: 0x0000

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

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

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

9.4.5 通用输出复位寄存器（PORRx）

复位值：0x0000

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

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

此寄存器的读出值始终为 0x0000。

9.4.6 通用输出翻转寄存器（POTRx）

复位值：0x0000

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

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

此寄存器的读出值始终为 0x0000。

9.4.7 特殊控制寄存器 (PSPCR)

复位值: 0x001F

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

位	标记	位名	功能	读写
b15~b6	Reserved	-	读出时为0, 写时请写0	R
b4	SPFE[4]	特殊功能选择	0: NJTRST功能无效 1: NJTRST功能有效	R/W
b3	SPFE[3]	特殊功能选择	0: JTDO_TRACESWO功能无效 1: JTDO_TRACESWO功能有效	R/W
b2	SPFE[2]	特殊功能选择	0: JTDO_SWDIO功能无效 1: JTDO_SWDIO功能有效	R/W
b1	SPFE[1]	特殊功能选择	0: JTMS_SWCLK功能无效 1: JTMS_SWCLK功能有效	R/W
b0	SPFE[0]	特殊功能选择	0: JTCK_SWCLK功能无效 1: JTCK_SWCLK功能有效	R/W

注意:

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

9.4.8 公共控制寄存器 (PCCR)

复位值: 0x4000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0											
RDWT[1:0]	-	-	-	-	-	-	-	-	-	-	-	BFSEL[3:0]														

位	标记	位名	功能	读写
b15-b14	RDWT[1:0]	读端口等待	设置读寄存器PIDRx, PCRxy时插入的等待周期数	R/W
		设定值	等待周期	推荐工作频率
		00	无等待	~42MHz
		01 (默认值)	1周期	42~84MHz
		10	2周期	84~126MHz
		11	3周期	126~200MHz
b13~b4	Reserved	-	读出时为0, 写时请写0	R
b3~b0	BFSEL[3:0]	副功能选择	各端口的功能配置请参考数据手册中的引脚功能表	R/W

9.4.9 输入控制寄存器 (PINAER)

复位值: 0x0000

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

位	标记	位名	功能	读写
b15~b6	Reserved	-	读出时为0, 写时请写0	R
b5~b0	PINAE[5:0]	输入常开	0: 输入MOS常开无效 1: 输入MOS常开有效 PINAE[0]控制PA0~PA15, PINAE[1]控制PB0~PB15, PINAE[2]控制PC0~PC15, PINAE[3]控制PD0~PD15, PINAE[4]控制PE0~PE15, PINAE[5]控制PH0~PH2	R/W

9.4.10 写保护寄存器 (PWPR)

复位值: 0x0000

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

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

9.4.11 通用控制寄存器 (PCRxy)

复位值: b0000_000x_0000_0000 *1

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

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

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

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

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

*1：以下端口通用控制寄存器 PCR 的复位值不是 b0000_000x_0000_0000，请注意。

XTAL32_IN, XTAL32_OUT 复用端口 PC14, PC15 的 PCRC14,PCRC15 寄存器的复位值为 0x8100。

9.4.12 功能选择寄存器 (PFSRxy)

复位值: 0x0000 *1

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

位	标记	位名	功能	读写
b15~b9	Reserved	-	读出时为0, 写时请写0	R
b8	BFE	副功能许可	控制PCCR.BFSEL[3:0]选择的副功能是否有效 0: 副功能禁止 1: 副功能有效	R/W
b7~b6	Reserved	-	读出时为0, 写时请写0	R
b5~b0	FSEL[5:0]	功能选择	各端口的功能配置请参考数据手册中的引脚功能表	R/W

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

注意:

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

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

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为0, 写时请写0	R
b15~b0	PDIR15:0]	方向选择	0: Event Port为输入功能 1: Event Port为输出功能	R/W

注意:

- EVNTP211 功能无输出功能（配置在 PB11 端口上）。

9.4.14 Event Port 输入数据寄存器 (PEVNTIDRm)

复位值: 0x0000_0000

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

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

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

9.4.15 Event Port 输出数据寄存器 (PEVNTODRm)

复位值: 0x0000_0000

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

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

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

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

复位值: 0x0000_0000

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

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

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

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

复位值: 0x0000_0000

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

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

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

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

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为0, 写时请写0	R
b15~b0	RIS[15:0]	上升沿检测许 可	0: EVNTPmn上升沿事件检出无效 1: EVNTPmn上升沿事件检出有效 PEVNTRISRm.RIS[n]对应EVNTPmn	R/W

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

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

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为0, 写时请写0	R
b15~b0	FAL[15:0]	下降沿检测许可	0: EVNTPmn下降沿事件检出无效 1: EVNTPmn下降沿事件检出有效 PEVNTRISRm.FAL[n]对应EVNTPmn	R/W

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

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

复位值: 0x0000_0000

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

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

1: Event Port1 数字滤波有效

9.4.21 32bit 访问

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

地址	b31 ~ b16	b15 ~ b0
0x4005_3800+0x10*n ^{*1}	Reserved	PIDRx
0x4005_3804+0x10*n	POERx	PODRx
0x4005_3808+0x10*n	PORRx	POSRx
0x4005_380C+0x10*n	Reserved	POTRx
0x4005_3BF4	Reserved	PSPCR
0x4005_3BF8	PINAER	PCCR
0x4005_3BFC	Reserved	PWPR
0x4005_3C00+0x40*n+0x04*y	PFSRxy	PCRxy

表 9-3 32bit 访问时 PORT 寄存器一览

注 *1：地址计算公式中 x=A~E,H 对应 n=0~4,5

9.5 注意事项

请不要将同一功能设置到多个端口上。

使用模拟功能时，请将相应端口的数字功能关闭(DDIS=1)。

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

10 中断控制器 (INTC)

10.1 简介

中断控制器 (INTC) 的功能有选择中断事件请求作为中断输入到 NVIC，唤醒 WFI；选择中断事件请求作为事件输入，唤醒 WFE；选择中断事件请求作为低功耗模式（休眠模式和停止模式）的唤醒条件；外部管脚 NMI 和 EIRQ 的中断控制功能；软件中断的中断/事件选择功能。

主要规格：

- 1) NVIC 中断向量：实际使用中断向量数请参考 10.3.1 中断向量表（不包括 CortexTM-M4F 的 16 根中断线），每个中断向量可以根据中断选择寄存器选择对应的外设中断事件请求。更多关于异常和 NVIC 编程的说明，请参考《ARM CortexTM-M4F 技术参考手册》中的第 5 章：异常和第 8 章：嵌套向量中断控制器。
- 2) 可编程优先级：16 个可编程优先级（使用了 4 位中断优先级寄存器）。
- 3) 不可屏蔽中断：除 NMI 管脚作为不可屏蔽中断源以外，可以独立选择多种系统中断事件请求作为不可屏蔽中断，且各中断事件请求配备独立的使能选择、标志以及标志清除寄存器。
- 4) 配备 16 个外部管脚中断。
- 5) 配置多种外设中断事件请求，具体请参考 10.3.2 中断事件请求序号。
- 6) 配备 32 个软件中断事件请求。
- 7) 中断可唤醒系统休眠模式和停止模式。

输入管脚:

管脚名	I/O	说明
NMI	输入	不可屏蔽中断请求管脚
EIRQ0	输入	外部管脚中断事件请求0
EIRQ1	输入	外部管脚中断事件请求1
EIRQ2	输入	外部管脚中断事件请求2
EIRQ3	输入	外部管脚中断事件请求3
EIRQ4	输入	外部管脚中断事件请求4
EIRQ5	输入	外部管脚中断事件请求5
EIRQ6	输入	外部管脚中断事件请求6
EIRQ7	输入	外部管脚中断事件请求7
EIRQ8	输入	外部管脚中断事件请求8
EIRQ9	输入	外部管脚中断事件请求9
EIRQ10	输入	外部管脚中断事件请求10
EIRQ11	输入	外部管脚中断事件请求11
EIRQ12	输入	外部管脚中断事件请求12
EIRQ13	输入	外部管脚中断事件请求13
EIRQ14	输入	外部管脚中断事件请求14
EIRQ15	输入	外部管脚中断事件请求15

10.2 INTC 系统框图

10.2.1 系统框图

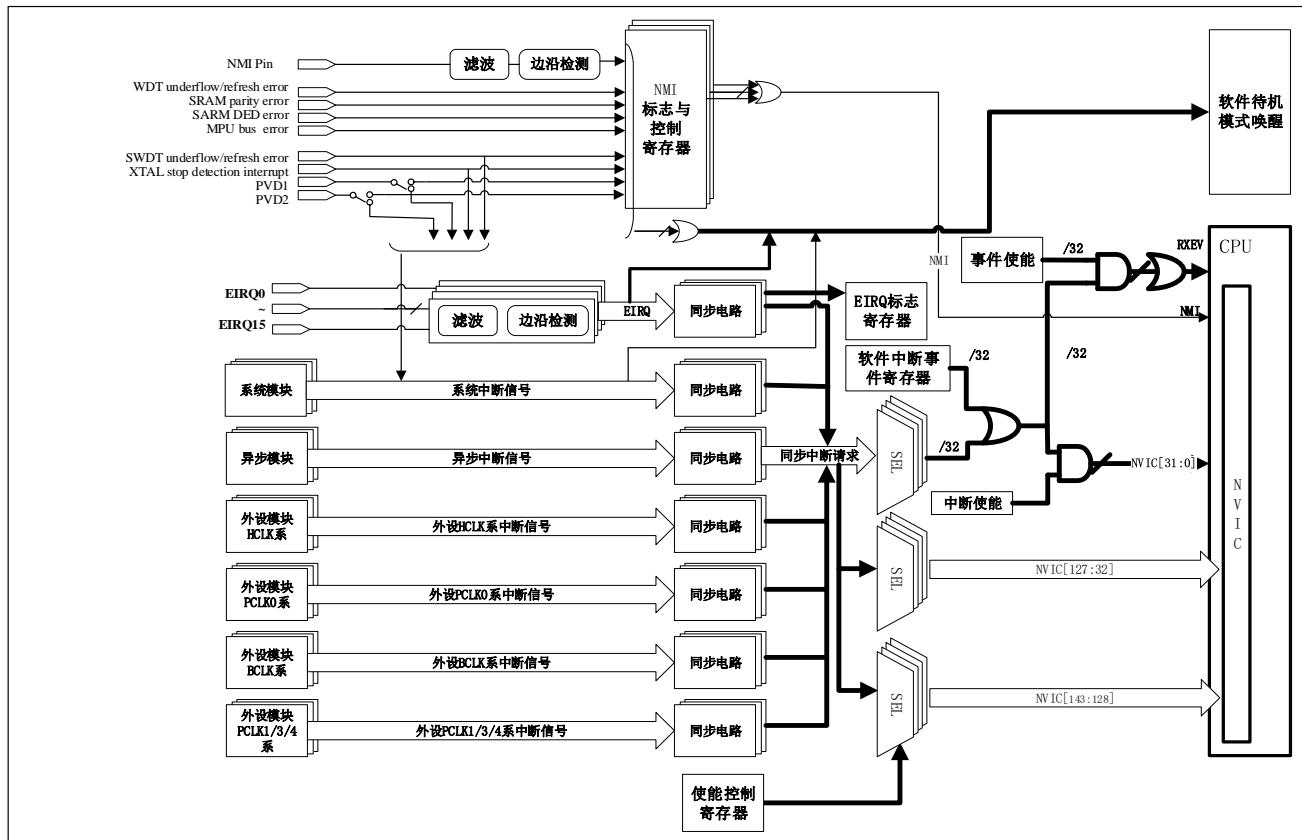


图 10-1 中断系统框图

10.3 向量表

10.3.1 中断向量表

地址	向量序号	IRQ序号	中断源	说明
ARM核中断处理向量				
0x0000_0000	0	-	ARM core	Initial stack pointer
0x0000_0004	1	-	ARM core	Initial Program Counter
0x0000_0008	2	-	ARM core	Non-maskable Interrupt (NMI)
0x0000_000C	3	-	ARM core	Hard Fault
0x0000_0010	4	-	ARM core	MemManage Fault
0x0000_0014	5	-	ARM core	Bus Fault
0x0000_0018	6	-	ARM core	Usage Fault
0x0000_001C	7	-	ARM core	Reserved
0x0000_0020	8	-	ARM core	Reserved
0x0000_0024	9	-	ARM core	Reserved
0x0000_0028	10	-	ARM core	Reserved
0x0000_002C	11	-	ARM core	Supervisor call (SVCall)
0x0000_0030	12	-	ARM core	Debug Monitor
0x0000_0034	13	-	ARM core	Reserved
0x0000_0038	14	-	ARM core	Pendable request for system service(PendableSrvReq)
0x0000_003C	15	-	ARM core	System tick timer (SysTick)
非ARM核中断处理向量				
0x0000_0040	16	0	INT_SEL0	寄存器INT_SEL0选择的中断事件请求/软件中断。
0x0000_0044	17	1	INT_SEL1	寄存器INT_SEL1选择的中断事件请求/软件中断。
0x0000_0048	18	2	INT_SEL2	寄存器INT_SEL2选择的中断事件请求/软件中断。
0x0000_004C	19	3	INT_SEL3	寄存器INT_SEL3选择的中断事件请求/软件中断。

地址	向量序号	IRQ序号	中断源	说明
				中断。
0x0000_0050	20	4	INT_SEL4	寄存器INT_SEL4选择的中断事件请求/软件中断。
0x0000_0054	21	5	INT_SEL5	寄存器INT_SEL5选择的中断事件请求/软件中断。
0x0000_0058	22	6	INT_SEL6	寄存器INT_SEL6选择的中断事件请求/软件中断。
0x0000_005C	23	7	INT_SEL7	寄存器INT_SEL7选择的中断事件请求/软件中断。
0x0000_0060	24	8	INT_SEL8	寄存器INT_SEL8选择的中断事件请求/软件中断。
0x0000_0064	25	9	INT_SEL9	寄存器INT_SEL9选择的中断事件请求/软件中断。
0x0000_0068	26	10	INT_SEL10	寄存器INT_SEL10选择的中断事件请求/软件中断。
0x0000_006C	27	11	INT_SEL11	寄存器INT_SEL11选择的中断事件请求/软件中断。
0x0000_0070	28	12	INT_SEL12	寄存器INT_SEL12选择的中断事件请求/软件中断。
0x0000_0074	29	13	INT_SEL13	寄存器INT_SEL13选择的中断事件请求/软件中断。
0x0000_0078	30	14	INT_SEL14	寄存器INT_SEL14选择的中断事件请求/软件中断。
0x0000_007C	31	15	INT_SEL15	寄存器INT_SEL15选择的中断事件请求/软件中断。
0x0000_0080	32	16	INT_SEL16	寄存器INT_SEL16选择的中断事件请求/软件中断。
0x0000_0084	33	17	INT_SEL17	寄存器INT_SEL17选择的中断事件请求/软件中断。

地址	向量序号	IRQ序号	中断源	说明
0x0000_0088	34	18	INT_SEL18	寄存器INT_SEL18选择的中断事件请求/软件中断。
0x0000_008C	35	19	INT_SEL19	寄存器INT_SEL19选择的中断事件请求/软件中断。
0x0000_0090	36	20	INT_SEL20	寄存器INT_SEL20选择的中断事件请求/软件中断。
0x0000_0094	37	21	INT_SEL21	寄存器INT_SEL21选择的中断事件请求/软件中断。
0x0000_0098	38	22	INT_SEL22	寄存器INT_SEL22选择的中断事件请求/软件中断。
0x0000_009C	39	23	INT_SEL23	寄存器INT_SEL23选择的中断事件请求/软件中断。
0x0000_00A0	40	24	INT_SEL24	寄存器INT_SEL24选择的中断事件请求/软件中断。
0x0000_00A4	41	25	INT_SEL25	寄存器INT_SEL25选择的中断事件请求/软件中断。
0x0000_00A8	42	26	INT_SEL26	寄存器INT_SEL26选择的中断事件请求/软件中断。
0x0000_00AC	43	27	INT_SEL27	寄存器INT_SEL27选择的中断事件请求/软件中断。
0x0000_00B0	44	28	INT_SEL28	寄存器INT_SEL28选择的中断事件请求/软件中断。
0x0000_00B4	45	29	INT_SEL29	寄存器INT_SEL29选择的中断事件请求/软件中断。
0x0000_00B8	46	30	INT_SEL30	寄存器INT_SEL30选择的中断事件请求/软件中断。
0x0000_00BC	47	31	INT_SEL31	寄存器INT_SEL31选择的中断事件请求/软件中断。
0x0000_00C0	48	32	INT_SEL32	寄存器INT_SEL32选择的中断事件请求。

地址	向量序号	IRQ序号	中断源	说明
0x0000_00C4	49	33	INT_SEL33	寄存器INT_SEL33选择的中断事件请求。
0x0000_00C8	50	34	INT_SEL34	寄存器INT_SEL34选择的中断事件请求。
0x0000_00CC	51	35	INT_SEL35	寄存器INT_SEL35选择的中断事件请求。
0x0000_00D0	52	36	INT_SEL36	寄存器INT_SEL36选择的中断事件请求。
0x0000_00D4	53	37	INT_SEL37	寄存器INT_SEL37选择的中断事件请求。
0x0000_00D8	54	38	INT_SEL38	寄存器INT_SEL38选择的中断事件请求。
0x0000_00DC	55	39	INT_SEL39	寄存器INT_SEL39选择的中断事件请求。
0x0000_00E0	56	40	INT_SEL40	寄存器INT_SEL40选择的中断事件请求。
0x0000_00E4	57	41	INT_SEL41	寄存器INT_SEL41选择的中断事件请求。
0x0000_00E8	58	42	INT_SEL42	寄存器INT_SEL42选择的中断事件请求。
0x0000_00EC	59	43	INT_SEL43	寄存器INT_SEL43选择的中断事件请求。
0x0000_00F0	60	44	INT_SEL44	寄存器INT_SEL44选择的中断事件请求。
0x0000_00F4	61	45	INT_SEL45	寄存器INT_SEL45选择的中断事件请求。
0x0000_00F8	62	46	INT_SEL46	寄存器INT_SEL46选择的中断事件请求。
0x0000_00FC	63	47	INT_SEL47	寄存器INT_SEL47选择的中断事件请求。
0x0000_0100	64	48	INT_SEL48	寄存器INT_SEL48选择的中断事件请求。
0x0000_0104	65	49	INT_SEL49	寄存器INT_SEL49选择的中断事件请求。
0x0000_0108	66	50	INT_SEL50	寄存器INT_SEL50选择的中断事件请求。
0x0000_010C	67	51	INT_SEL51	寄存器INT_SEL51选择的中断事件请求。
0x0000_0110	68	52	INT_SEL52	寄存器INT_SEL52选择的中断事件请求。
0x0000_0114	69	53	INT_SEL53	寄存器INT_SEL53选择的中断事件请求。
0x0000_0118	70	54	INT_SEL54	寄存器INT_SEL54选择的中断事件请求。
0x0000_011C	71	55	INT_SEL55	寄存器INT_SEL55选择的中断事件请求。
0x0000_0120	72	56	INT_SEL56	寄存器INT_SEL56选择的中断事件请求。
0x0000_0124	73	57	INT_SEL57	寄存器INT_SEL57选择的中断事件请求。
0x0000_0128	74	58	INT_SEL58	寄存器INT_SEL58选择的中断事件请求。
0x0000_012C	75	59	INT_SEL59	寄存器INT_SEL59选择的中断事件请求。
0x0000_0130	76	60	INT_SEL60	寄存器INT_SEL60选择的中断事件请求。
0x0000_0134	77	61	INT_SEL61	寄存器INT_SEL61选择的中断事件请求。

地址	向量序号	IRQ序号	中断源	说明
0x0000_0138	78	62	INT_SEL62	寄存器INT_SEL62选择的中断事件请求。
0x0000_013C	79	63	INT_SEL63	寄存器INT_SEL63选择的中断事件请求。
0x0000_0140	80	64	INT_SEL64	寄存器INT_SEL64选择的中断事件请求。
0x0000_0144	81	65	INT_SEL65	寄存器INT_SEL65选择的中断事件请求。
0x0000_0148	82	66	INT_SEL66	寄存器INT_SEL66选择的中断事件请求。
0x0000_014C	83	67	INT_SEL67	寄存器INT_SEL67选择的中断事件请求。
0x0000_0150	84	68	INT_SEL68	寄存器INT_SEL68选择的中断事件请求。
0x0000_0154	85	69	INT_SEL69	寄存器INT_SEL69选择的中断事件请求。
0x0000_0158	86	70	INT_SEL70	寄存器INT_SEL70选择的中断事件请求。
0x0000_015C	87	71	INT_SEL71	寄存器INT_SEL71选择的中断事件请求。
0x0000_0160	88	72	INT_SEL72	寄存器INT_SEL72选择的中断事件请求。
0x0000_0164	89	73	INT_SEL73	寄存器INT_SEL73选择的中断事件请求。
0x0000_0168	90	74	INT_SEL74	寄存器INT_SEL74选择的中断事件请求。
0x0000_016C	91	75	INT_SEL75	寄存器INT_SEL75选择的中断事件请求。
0x0000_0170	92	76	INT_SEL76	寄存器INT_SEL76选择的中断事件请求。
0x0000_0174	93	77	INT_SEL77	寄存器INT_SEL77选择的中断事件请求。
0x0000_0178	94	78	INT_SEL78	寄存器INT_SEL78选择的中断事件请求。
0x0000_017C	95	79	INT_SEL79	寄存器INT_SEL79选择的中断事件请求。
0x0000_0180	96	80	INT_SEL80	寄存器INT_SEL80选择的中断事件请求。
0x0000_0184	97	81	INT_SEL81	寄存器INT_SEL81选择的中断事件请求。
0x0000_0188	98	82	INT_SEL82	寄存器INT_SEL82选择的中断事件请求。
0x0000_018C	99	83	INT_SEL83	寄存器INT_SEL83选择的中断事件请求。
0x0000_0190	100	84	INT_SEL84	寄存器INT_SEL84选择的中断事件请求。
0x0000_0194	101	85	INT_SEL85	寄存器INT_SEL85选择的中断事件请求。
0x0000_0198	102	86	INT_SEL86	寄存器INT_SEL86选择的中断事件请求。
0x0000_019C	103	87	INT_SEL87	寄存器INT_SEL87选择的中断事件请求。
0x0000_01A0	104	88	INT_SEL88	寄存器INT_SEL88选择的中断事件请求。
0x0000_01A4	105	89	INT_SEL89	寄存器INT_SEL89选择的中断事件请求。
0x0000_01A8	106	90	INT_SEL90	寄存器INT_SEL90选择的中断事件请求。

地址	向量序号	IRQ序号	中断源	说明
0x0000_01AC	107	91	INT_SEL91	寄存器INT_SEL91选择的中断事件请求。
0x0000_01B0	108	92	INT_SEL92	寄存器INT_SEL92选择的中断事件请求。
0x0000_01B4	109	93	INT_SEL93	寄存器INT_SEL93选择的中断事件请求。
0x0000_01B8	110	94	INT_SEL94	寄存器INT_SEL94选择的中断事件请求。
0x0000_01BC	111	95	INT_SEL95	寄存器INT_SEL95选择的中断事件请求。
0x0000_01C0	112	96	INT_SEL96	寄存器INT_SEL96选择的中断事件请求。
0x0000_01C4	113	97	INT_SEL97	寄存器INT_SEL97选择的中断事件请求。
0x0000_01C8	114	98	INT_SEL98	寄存器INT_SEL98选择的中断事件请求。
0x0000_01CC	115	99	INT_SEL99	寄存器INT_SEL99选择的中断事件请求。
0x0000_01D0	116	100	INT_SEL100	寄存器INT_SEL100选择的中断事件请求。
0x0000_01D4	117	101	INT_SEL101	寄存器INT_SEL101选择的中断事件请求。
0x0000_01D8	118	102	INT_SEL102	寄存器INT_SEL102选择的中断事件请求。
0x0000_01DC	119	103	INT_SEL103	寄存器INT_SEL103选择的中断事件请求。
0x0000_01E0	120	104	INT_SEL104	寄存器INT_SEL104选择的中断事件请求。
0x0000_01E4	121	105	INT_SEL105	寄存器INT_SEL105选择的中断事件请求。
0x0000_01E8	122	106	INT_SEL106	寄存器INT_SEL106选择的中断事件请求。
0x0000_01EC	123	107	INT_SEL107	寄存器INT_SEL107选择的中断事件请求。
0x0000_01F0	124	108	INT_SEL108	寄存器INT_SEL108选择的中断事件请求。
0x0000_01F4	125	109	INT_SEL109	寄存器INT_SEL109选择的中断事件请求。
0x0000_01F8	126	110	INT_SEL110	寄存器INT_SEL110选择的中断事件请求。
0x0000_01FC	127	111	INT_SEL111	寄存器INT_SEL111选择的中断事件请求。
0x0000_0200	128	112	INT_SEL112	寄存器INT_SEL112选择的中断事件请求。
0x0000_0204	129	113	INT_SEL113	寄存器INT_SEL113选择的中断事件请求。
0x0000_0208	130	114	INT_SEL114	寄存器INT_SEL114选择的中断事件请求。
0x0000_020C	131	115	INT_SEL115	寄存器INT_SEL115选择的中断事件请求。
0x0000_0210	132	116	INT_SEL116	寄存器INT_SEL116选择的中断事件请求。
0x0000_0214	133	117	INT_SEL117	寄存器INT_SEL117选择的中断事件请求。
0x0000_0218	134	118	INT_SEL118	寄存器INT_SEL118选择的中断事件请求。
0x0000_021C	135	119	INT_SEL119	寄存器INT_SEL119选择的中断事件请求。

地址	向量序号	IRQ序号	中断源	说明
0x0000_0220	136	120	INT_SEL120	寄存器INT_SEL120选择的中断事件请求。
0x0000_0224	137	121	INT_SEL121	寄存器INT_SEL121选择的中断事件请求。
0x0000_0228	138	122	INT_SEL122	寄存器INT_SEL122选择的中断事件请求。
0x0000_022C	139	123	INT_SEL123	寄存器INT_SEL123选择的中断事件请求。
0x0000_0230	140	124	INT_SEL124	寄存器INT_SEL124选择的中断事件请求。
0x0000_0234	141	125	INT_SEL125	寄存器INT_SEL125选择的中断事件请求。
0x0000_0238	142	126	INT_SEL126	寄存器INT_SEL126选择的中断事件请求。
0x0000_023C	143	127	INT_SEL127	寄存器INT_SEL127选择的中断事件请求。
0x0000_0240	144	128	INT_VSSEL128	寄存器INT_VSSEL128通过使能位选择的中断事件请求共用该向量。
0x0000_0244	145	129	INT_VSSEL129	寄存器INT_VSSEL129通过使能位选择的中断事件请求共用该向量。
0x0000_0248	146	130	INT_VSSEL130	寄存器INT_VSSEL130通过使能位选择的中断事件请求共用该向量。
0x0000_024C	147	131	INT_VSSEL131	寄存器INT_VSSEL131通过使能位选择的中断事件请求共用该向量。
0x0000_0250	148	132	INT_VSSEL132	寄存器INT_VSSEL132通过使能位选择的中断事件请求共用该向量。
0x0000_0254	149	133	INT_VSSEL133	寄存器INT_VSSEL133通过使能位选择的中断事件请求共用该向量。
0x0000_0258	150	134	INT_VSSEL134	寄存器INT_VSSEL134通过使能位选择的中断事件请求共用该向量。
0x0000_025C	151	135	INT_VSSEL135	寄存器INT_VSSEL135通过使能位选择的中断事件请求共用该向量。
0x0000_0260	152	136	INT_VSSEL136	寄存器INT_VSSEL136通过使能位选择的中断事件请求共用该向量。
0x0000_0264	153	137	INT_VSSEL137	寄存器INT_VSSEL137通过使能位选择的中断事件请求共用该向量。
0x0000_0268	154	138	INT_VSSEL138	寄存器INT_VSSEL138通过使能位选择的中断事件请求共用该向量。

地址	向量序号	IRQ序号	中断源	说明
			138	断事件请求共用该向量。
0x0000_026C	155	139	INT_VSSEL139	寄存器INT_VSSEL139通过使能位选择的中断事件请求共用该向量。
0x0000_0270	156	140	INT_VSSEL140	寄存器INT_VSSEL140通过使能位选择的中断事件请求共用该向量。
0x0000_0274	157	141	INT_VSSEL141	寄存器INT_VSSEL141通过使能位选择的中断事件请求共用该向量。
0x0000_0278	158	142	INT_VSSEL142	寄存器INT_VSSEL142通过使能位选择的中断事件请求共用该向量。
0x0000_027C	159	143	INT_VSSEL143	寄存器INT_VSSEL143通过使能位选择的中断事件请求共用该向量。

表 10-1 中断向量表

注意：

- 具体选择的中断事件请求编号请参考寄存器说明章节。

10.3.2 中断事件请求序号

中断事件请求由外设产生，中断事件请求被中断控制器选择为 NVIC 的输入时，中断事件请求被称为中断源；当被选择为事件输入时，被称为事件源。外设中断事件请求也可以作为 MCU 低功耗模式返回的条件。

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	可否选择为内部触发源	对应NVIC向量的中断选择寄存器 ¹		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
0	000h	PORT	PORT_EIRQ0	√	√	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[0]
1	001h		PORT_EIRQ1	√	√	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[1]
2	002h		PORT_EIRQ2	√	√	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[2]
3	003h		PORT_EIRQ3	√	√	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[3]
4	004h		PORT_EIRQ4	√	√	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[4]
5	005h		PORT_EIRQ5	√	√	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[5]
6	006h		PORT_EIRQ6	√	√	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[6]
7	007h		PORT_EIRQ7	√	√	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[7]
8	008h		PORT_EIRQ8	√	√	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[8]
9	009h		PORT_EIRQ9	√	√	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[9]
10	00Ah		PORT_EIRQ10	√	√	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[10]
11	00Bh		PORT_EIRQ11	√	√	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[11]
12	00Ch		PORT_EIRQ12	√	√	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[12]
13	00Dh		PORT_EIRQ13	√	√	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[13]
14	00Eh		PORT_EIRQ14	√	√	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[14]
15	00Fh		PORT_EIRQ15	√	√	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[15]
16	010h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[16]
17	011h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[17]
18	012h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[18]
19	013h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[19]
20	014h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[20]
21	015h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[21]

编号	中断事 件请求 序号	功能	功能名称	是否可 选择为 中断	可否选择 为内部触 发源	对应NVIC向量的中断选择寄存器 ^{*1}		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
22	016h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[22]
23	017h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[23]
24	018h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[24]
25	019h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[25]
26	01Ah	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[26]
27	01Bh	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[27]
28	01Ch	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[28]
29	01Dh	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[29]
30	01Eh	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[30]
31	01Fh	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[31]
32	020h	DMA	DMA1_TC0	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[0]
33	021h		DMA1_TC1	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[1]
34	022h		DMA1_TC2	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[2]
35	023h		DMA1_TC3	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[3]
36	024h		DMA2_TC0	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[4]
37	025h		DMA2_TC1	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[5]
38	026h		DMA2_TC2	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[6]
39	027h		DMA2_TC3	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[7]
40	028h		DMA1_BTC0	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[8]
41	029h		DMA1_BTC1	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[9]
42	02Ah		DMA1_BTC2	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[10]
43	02Bh		DMA1_BTC3	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[11]
44	02Ch		DMA2_BTC0	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[12]
45	02Dh		DMA2_BTC1	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[13]
46	02Eh		DMA2_BTC2	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[14]
47	02Fh		DMA2_BTC3	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[15]
48	030h	DMA	DMA1_ERR	√	-	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[16]
49	031h		DMA2_ERR	√	-	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[17]

编号	中断事 件请求 序号	功能	功能名称	是否可 选择为 中断	可否选择 为内部触 发源	对应NVIC向量的中断选择寄存器 ^{*1}		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
50	032h	EFM	EFM_PEERR	√	-	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[18]
51	033h		EFM_COLERR	√	-	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[19]
52	034h		EFM_OPTEND	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[20]
53	035h	USBFS	USBFS_SOF	-	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[21]
54	036h	QSPI	QSPI_INTR	√	-	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[22]
55	037h	DCU	DCU1	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[23]
56	038h		DCU2	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[24]
57	039h		DCU3	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[25]
58	03Ah		DCU4	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[26]
59	03Bh	-	-	-	-	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[27]
60	03Ch	-	-	-	-	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[28]
61	03Dh	-	-	-	-	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[29]
62	03Eh	-	-	-	-	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[30]
63	03Fh	-	-	-	-	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[31]
64	040h	Timer0_1	TMR01_GCMA	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[0]
65	041h		TMR01_GCMB	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[1]
66	042h	Timer0_2	TMR02_GCMA	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[2]
67	043h		TMR02_GCMB	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[3]
68	044h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[4]
69	045h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[5]
70	046h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[6]
71	047h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[7]
72	048h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[8]
73	049h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[9]
74	04Ah	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[10]
75	04Bh	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[11]
76	04Ch	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[12]
77	04Dh	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[13]

编号	中断事 件请求 序号	功能	功能名称	是否可 选择为 中断	可否选择 为内部触 发源	对应NVIC向量的中断选择寄存器 ^{*1}		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
78	04Eh	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[14]
79	04Fh	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[15]
80	050h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[16]
81	051h	RTC	RTC_ALM	√	√	INT_SEL0~31	INT_SEL44~49	-
82	052h		RTC_PRD	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[18]
83	053h		-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[19]
84	054h	XTAL32	XTAL32_STOP	√	-	INT_SEL0~31	INT_SEL44~49	-
85	055h	XTAL	XTAL_STOP	√	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[21]
86	056h	WKT M	WKT M_PRD	√	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[22]
87	057h	SWDT	SWDT_REFUDF	√	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[23]
88	058h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[24]
89	059h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[25]
90	05Ah	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[26]
91	05Bh	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[27]
92	05Ch	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[28]
93	05Dh	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[29]
94	05Eh	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[30]
95	05Fh	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[31]
96	060h	Timer6_1	TMR61_GCMA	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[0]
97	061h		TMR61_GCMB	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[1]
98	062h		TMR61_GCMC	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[2]
99	063h		TMR61_GCMD	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[3]
100	064h		TMR61_GCME	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[4]
101	065h		TMR61_GCMF	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[5]
102	066h		TMR61_GOVF	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[6]
103	067h		TMR61_GUDF	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[7]
104	068h		TMR61_GDTE	√	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[8]
105	069h		-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[9]

编号	中断事 件请求 序号	功能	功能名称	是否可 选择为 中断	可否选择 为内部触 发源	对应NVIC向量的中断选择寄存器 ^{*1}		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
106	06Ah	-	-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[10]
107	06Bh		TMR61_SCMA	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[11]
108	06Ch		TMR61_SCMB	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[12]
109	06Dh	-	-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[13]
110	06Eh	-	-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[14]
111	06Fh	-	-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[15]
112	070h	Timer6_2	TMR62_GCMA	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[16]
113	071h		TMR62_GCMB	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[17]
114	072h		TMR62_GCMC	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[18]
115	073h		TMR62_GCMD	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[19]
116	074h		TMR62_GCME	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[20]
117	075h		TMR62_GCMF	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[21]
118	076h		TMR62_GOVF	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[22]
119	077h		TMR62_GUDF	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[23]
120	078h		TMR62_GDTE	√	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[24]
121	079h		-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[25]
122	07Ah		-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[26]
123	07Bh		TMR62_SCMA	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[27]
124	07Ch		TMR62_SCMB	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[28]
125	07Dh	-	-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[29]
126	07Eh	-	-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[30]
127	07Fh	-	-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[31]
128	080h	Timer6_3	TMR63_GCMA	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[0]
129	081h		TMR63_GCMB	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[1]
130	082h		TMR63_GCMC	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[2]
131	083h		TMR63_GCMD	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[3]
132	084h		TMR63_GCME	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[4]
133	085h		TMR63_GCMF	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[5]

编号	中断事 件请求 序号	功能	功能名称	是否可 选择为 中断	可否选择 为内部触 发源	对应NVIC向量的中断选择寄存器 ^{*1}		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
134	086h		TMR63_GOVF	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[6]
135	087h		TMR63_GUDF	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[7]
136	088h		TMR63_GDTE	√	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[8]
137	089h		-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[9]
138	08Ah		-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[10]
139	08Bh		TMR63_SCMA	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[11]
140	08Ch		TMR63_SCMB	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[12]
141	08Dh	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[13]
142	08Eh	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[14]
143	08Fh	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[15]
144	090h	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[16]
145	091h	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[17]
146	092h	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[18]
147	093h	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[19]
148	094h	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[20]
149	095h	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[21]
150	096h	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[22]
151	097h	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[23]
152	098h	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[24]
153	099h	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[25]
154	09Ah	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[26]
155	09Bh	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[27]
156	09Ch	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[28]
157	09Dh	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[29]
158	09Eh	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[30]
159	09Fh	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[31]
160	0A0h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[0]
161	0A1h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[1]

编号	中断事 件请求 序号	功能	功能名称	是否可 选择为 中断	可否选择 为内部触 发源	对应NVIC向量的中断选择寄存器 ^{*1}		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
162	0A2h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[2]
163	0A3h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[3]
164	0A4h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[4]
165	0A5h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[5]
166	0A6h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[6]
167	0A7h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[7]
168	0A8h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[8]
169	0A9h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[9]
170	0AAh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[10]
171	0ABh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[11]
172	0ACh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[12]
173	0ADh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[13]
174	0AEh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[14]
175	0AFh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[15]
176	0B0h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[16]
177	0B1h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[17]
178	0B2h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[18]
179	0B3h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[19]
180	0B4h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[20]
181	0B5h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[21]
182	0B6h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[22]
183	0B7h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[23]
184	0B8h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[24]
185	0B9h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[25]
186	0BAh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[26]
187	0BBh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[27]
188	0BCh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[28]
189	0BDh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[29]

编号	中断事 件请求 序号	功能	功能名称	是否可 选择为 中断	可否选择 为内部触 发源	对应NVIC向量的中断选择寄存器 ^{*1}		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
190	0BEh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[30]
191	0BFh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[31]
192	0C0h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[0]
193	0C1h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[1]
194	0C2h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[2]
195	0C3h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[3]
196	0C4h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[4]
197	0C5h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[5]
198	0C6h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[6]
199	0C7h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[7]
200	0C8h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[8]
201	0C9h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[9]
202	0CAh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[10]
203	0CBh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[11]
204	0CCh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[12]
205	0CDh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[13]
206	0CEh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[14]
207	0CFh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[15]
208	0D0h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[16]
209	0D1h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[17]
210	0D2h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[18]
211	0D3h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[19]
212	0D4h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[20]
213	0D5h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[21]
214	0D6h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[22]
215	0D7h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[23]
216	0D8h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[24]
217	0D9h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[25]

编号	中断事 件请求 序号	功能	功能名称	是否可 选择为 中断	可否选择 为内部触 发源	对应NVIC向量的中断选择寄存器 ^{*1}		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
218	0DAh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[26]
219	0DBh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[27]
220	0DCh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[28]
221	0DDh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[29]
222	0DEh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[30]
223	0DFh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[31]
224	0E0h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[0]
225	0E1h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[1]
226	0E2h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[2]
227	0E3h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[3]
228	0E4h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[4]
229	0E5h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[5]
230	0E6h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[6]
231	0E7h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[7]
232	0E8h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[8]
233	0E9h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[9]
234	0EAh	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[10]
235	0EBh	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[11]
236	0Ec h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[12]
237	0EDh	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[13]
238	0EEh	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[14]
239	0EFh	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[15]
240	0F0h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[16]
241	0F1h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[17]
242	0F2h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[18]
243	0F3h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[19]
244	0F4h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[20]
245	0F5h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[21]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	可否选择为内部触发源	对应NVIC向量的中断选择寄存器 ¹		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
246	0F6h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[22]
247	0F7h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[23]
248	0F8h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[24]
249	0F9h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[25]
250	0FAh	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[26]
251	0FBh	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[27]
252	0FCCh	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[28]
253	0FDh	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[29]
254	0FEh	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[30]
255	0FFh	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[31]
256	100h	TimerA_1	TMRA1_OVF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[0]
257	101h		TMRA1_UDF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[1]
258	102h		TMRA1_CMP	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[2]
259	103h	TimerA_2	TMRA2_OVF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[3]
260	104h		TMRA2_UDF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[4]
261	105h		TMRA2_CMP	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[5]
262	106h	TimerA_3	TMRA3_OVF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[6]
263	107h		TMRA3_UDF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[7]
264	108h		TMRA3_CMP	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[8]
265	109h	TimerA_4	TMRA4_OVF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[9]
266	10Ah		TMRA4_UDF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[10]
267	10Bh		TMRA4_CMP	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[11]
268	10Ch	TimerA_5	TMRA5_OVF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[12]
269	10Dh		TMRA5_UDF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[13]
270	10Eh		TMRA5_CMP	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[14]
271	10Fh	-	-	-	-	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[15]
272	110h	TimerA_6	TMRA6_OVF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[16]
273	111h		TMRA6_UDF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[17]

编号	中断事 件请求 序号	功能	功能名称	是否可 选择为 中断	可否选择 为内部触 发源	对应NVIC向量的中断选择寄存器 ^{*1}		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
274	112h		TMRA6_CMP	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[18]
275	113h	USBFS	USBFS_GLB	√	-	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[19]
276	114h	-	-	-	-	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[20]
277	115h	-	-	-	-	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[21]
278	116h	USART1	USART1_REI	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[22]
279	117h		USART1_RI	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[23]
280	118h		USART1_TI	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[24]
281	119h		USART1_TCI	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[25]
282	11Ah		USART1_RTOI	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[26]
283	11Bh	USART2	USART2_REI	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[27]
284	11Ch		USART2_RI	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[28]
285	11Dh		USART2_TI	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[29]
286	11Eh		USART2_TCI	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[30]
287	11Fh		USART2_RTOI	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[31]
288	120h	USART3	USART3_REI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[0]
289	121h		USART3_RI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[1]
290	122h		USART3_TI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[2]
291	123h		USART3_TCI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[3]
292	124h		USART3_RTOI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[4]
293	125h	USART4	USART4_REI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[5]
294	126h		USART4_RI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[6]
295	127h		USART4_TI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[7]
296	128h		USART4_TCI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[8]
297	129h		USART4_RTOI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[9]
298	12Ah	-	-	-	-	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[10]
299	12Bh	SPI1	SPI1_SPRI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[11]
300	12Ch		SPI1_SPTI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[12]
301	12Dh		SPI1_SPII	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[13]

编号	中断事 件请求 序号	功能	功能名称	是否可 选择为 中断	可否选择 为内部触 发源	对应NVIC向量的中断选择寄存器 ^{*1}		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
302	12Eh		SPI1_SPEI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[14]
303	12Fh		SPI1_SPTEND	-	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[15]
304	130h	SPI2	SPI2_SPRI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[16]
305	131h		SPI2_SPTI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[17]
306	132h		SPI2_SPII	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[18]
307	133h		SPI2_SPEI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[19]
308	134h		SPI2_SPTEND	-	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[20]
309	135h	SPI3	SPI3_SPRI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[21]
310	136h		SPI3_SPTI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[22]
311	137h		SPI3_SPII	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[23]
312	138h		SPI3_SPEI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[24]
313	139h		SPI3_SPTEND	-	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[25]
314	13Ah	SPI4	SPI4_SPRI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[26]
315	13Bh		SPI4_SPTI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[27]
316	13Ch		SPI4_SPII	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[28]
317	13Dh		SPI4_SPEI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[29]
318	13Eh		SPI4_SPTEND	-	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[30]
319	13Fh	AOS_STRG	AOS_STRG*2	-	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[31]
320	140h	Timer4_1	TMR41_GCMUH	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[0]
321	141h		TMR41_GCMUL	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[1]
322	142h		TMR41_GCMVH	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[2]
323	143h		TMR41_GCMVL	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[3]
324	144h		TMR41_GCMWH	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[4]
325	145h		TMR41_GCMWL	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[5]
326	146h		TMR41_GOVF	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[6]
327	147h		TMR41_GUDF	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[7]
328	148h		TMR41_RLOU	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[8]
329	149h		TMR41_RLOV	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[9]

编号	中断事 件请求 序号	功能	功能名称	是否可 选择为 中断	可否选择 为内部触 发源	对应NVIC向量的中断选择寄存器 ^{*1}		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
330	14Ah		TMR41_RLOW	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[10]
331	14Bh		-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[11]
332	14Ch	-	-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[12]
333	14Dh	-	-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[13]
334	14Eh	-	-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[14]
335	14Fh	-	-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[15]
336	150h	Timer4_2	TMR42_GCMUH	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[16]
337	151h		TMR42_GCMUL	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[17]
338	152h		TMR42_GCMVH	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[18]
339	153h		TMR42_GCMVL	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[19]
340	154h		TMR42_GCMWH	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[20]
341	155h		TMR42_GCMWL	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[21]
342	156h		TMR42_GOVF	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[22]
343	157h		TMR42_GUDF	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[23]
344	158h		TMR42_RLOU	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[24]
345	159h		TMR42_RLOV	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[25]
346	15Ah		TMR42_RLOW	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[26]
347	15Bh		-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[27]
348	15Ch	-	-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[28]
349	15Dh	-	-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[29]
350	15Eh	-	-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[30]
351	15Fh	-	-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[31]
352	160h	Timer4_3	TMR43_GCMUH	√	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[0]
353	161h		TMR43_GCMUL	√	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[1]
354	162h		TMR43_GCMVH	√	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[2]
355	163h		TMR43_GCMVL	√	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[3]
356	164h		TMR43_GCMWH	√	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[4]
357	165h		TMR43_GCMWL	√	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[5]

编号	中断事 件请求 序号	功能	功能名称	是否可 选择为 中断	可否选择 为内部触 发源	对应NVIC向量的中断选择寄存器 ^{*1}		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
358	166h		TMR43_GOVF	√	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[6]
359	167h		TMR43_GUDF	√	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[7]
360	168h		TMR43_RLOU	√	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[8]
361	169h		TMR43_RLOV	√	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[9]
362	16Ah		TMR43_RLOW	√	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[10]
363	16Bh		-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[11]
364	16Ch		-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[12]
365	16Dh		-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[13]
366	16Eh		-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[14]
367	16Fh		-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[15]
368	170h	Timer4_1 EVT	TMR41_SCMUH	-	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[16]
369	171h		TMR41_SCML	-	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[17]
370	172h		TMR41_SCMVH	-	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[18]
371	173h		TMR41_SCMVL	-	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[19]
372	174h		TMR41_SCMWH	-	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[20]
373	175h		TMR41_SCMWL	-	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[21]
374	176h	Timer4_2 EVT	TMR42_SCMUH	-	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[22]
375	177h		TMR42_SCML	-	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[23]
376	178h		TMR42_SCMVH	-	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[24]
377	179h		TMR42_SCMVL	-	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[25]
378	17Ah		TMR42_SCMWH	-	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[26]
379	17Bh		TMR42_SCMWL	-	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[27]
380	17Ch	-	-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[28]
381	17Dh	-	-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[29]
382	17Eh	-	-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[30]
383	17Fh	-	-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[31]
384	180h	Timer4_3 EVT	TMR43_SCMUH	-	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[0]
385	181h		TMR43_SCML	-	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[1]

编号	中断事 件请求 序号	功能	功能名称	是否可 选择为 中断	可否选择 为内部触 发源	对应NVIC向量的中断选择寄存器 ^{*1}		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
386	182h		TMR43_SCMVH	-	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[2]
387	183h		TMR43_SCMVL	-	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[3]
388	184h		TMR43_SCMWH	-	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[4]
389	185h		TMR43_SCMWL	-	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[5]
390	186h	EMB	EMB_GR0	√	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[6]
391	187h		EMB_GR1	√	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[7]
392	188h		EMB_GR2	√	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[8]
393	189h		EMB_GR3	√	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[9]
394	18Ah	EVENT port	EVENT_PORT1	√	√	INT_SEL0~31	INT_SEL104~109	-
395	18Bh		EVENT_PORT2	√	√	INT_SEL0~31	INT_SEL104~109	-
396	18Ch		EVENT_PORT3	√	√	INT_SEL0~31	INT_SEL104~109	-
397	18Dh		EVENT_PORT4	√	√	INT_SEL0~31	INT_SEL104~109	-
398	18Eh	-	-	-	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[14]
399	18Fh	-	-	-	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[15]
400	190h	I2S1	I2S1_TXIRQOUT	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[16]
401	191h		I2S1_RXIRQOUT	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[17]
402	192h		I2S1_ERRIRQOUT	√	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[18]
403	193h	I2S2	I2S2_TXIRQOUT	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[19]
404	194h		I2S2_RXIRQOUT	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[20]
405	195h		I2S2_ERRIRQOUT	√	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[21]
406	196h	I2S3	I2S3_TXIRQOUT	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[22]
407	197h		I2S3_RXIRQOUT	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[23]
408	198h		I2S3_ERRIRQOUT	√	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[24]
409	199h	I2S4	I2S4_TXIRQOUT	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[25]
410	19Ah		I2S4_RXIRQOUT	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[26]
411	19Bh		I2S4_ERRIRQOUT	√	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[27]
412	19Ch	-	-	-	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[28]
413	19Dh	-	-	-	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[29]

编号	中断事 件请求 序号	功能	功能名称	是否可 选择为 中断	可否选择 为内部触 发源	对应NVIC向量的中断选择寄存器 ¹		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
414	19Eh	-	-	-	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[30]
415	19Fh	-	-	-	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[31]
416	1A0h	ACMP	ACMP1	√	√	INT_SEL0~31	INT_SEL110~115	-
417	1A1h		ACMP2	√	√	INT_SEL0~31	INT_SEL110~115	-
418	1A2h		ACMP3	√	√	INT_SEL0~31	INT_SEL110~115	-
419	1A3h	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[3]
420	1A4h	I2C1	I2C1_RXI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[4]
421	1A5h		I2C1_TXI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[5]
422	1A6h		I2C1_TEI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[6]
423	1A7h		I2C1_EE1	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[7]
424	1A8h	I2C2	I2C2_RXI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[8]
425	1A9h		I2C2_TXI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[9]
426	1AAh		I2C2_TEI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[10]
427	1ABh		I2C2_EE1	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[11]
428	1ACh	I2C3	I2C3_RXI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[12]
429	1ADh		I2C3_TXI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[13]
430	1AEh		I2C3_TEI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[14]
431	1AFh		I2C3_EE1	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[15]
432	1B0h	USART1	USART1_WUPI	√	-	INT_SEL0~31	INT_SEL110~115	-
433	1B1h	PVD	PVD_PVD1	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[17]
434	1B2h		PVD_PVD2	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[18]
435	1B3h	OTS	OTS	√	√	INT_SEL0~31	INT_SEL110~115	-
436	1B4h	FCM	FCMFERRI	√	-	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[20]
437	1B5h		FCMMENDI	√	-	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[21]
438	1B6h		FCMCOVFI	√	-	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[22]
439	1B7h	WDT	WDT_REFUDF	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[23]
440	1B8h	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[24]
441	1B9h	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[25]

编号	中断事 件请求 序号	功能	功能名称	是否可 选择为 中断	可否选择 为内部触 发源	对应NVIC向量的中断选择寄存器 ^{*1}		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
442	1BAh	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[26]
443	1BBh	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[27]
444	1BCh	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[28]
445	1BDh	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[29]
446	1BEh	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[30]
447	1BFh	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[31]
448	1C0h	ADC1	ADC1_EOCA	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[0]
449	1C1h		ADC1_EOCB	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[1]
450	1C2h		ADC1_CHCMP	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[2]
451	1C3h		ADC1_SEQCMP	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[3]
452	1C4h	ADC2	ADC2_EOCA	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[4]
453	1C5h		ADC2_EOCB	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[5]
454	1C6h		ADC2_CHCMP	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[6]
455	1C7h		ADC2_SEQCMP	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[7]
456	1C8h	TRNG	TRNG_END	√	√	INT_SEL0~31	INT_SEL116~121	-
457	1C9h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[9]
458	1CAh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[10]
459	1CBh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[11]
460	1CCh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[12]
461	1CDh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[13]
462	1CEh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[14]
463	1CFh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[15]
464	1D0h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[16]
465	1D1h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[17]
466	1D2h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[18]
467	1D3h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[19]
468	1D4h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[20]
469	1D5h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[21]

编号	中断事 件请求 序号	功能	功能名称	是否可 选择为 中断	可否选择 为内部触 发源	对应NVIC向量的中断选择寄存器 ^{*1}		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
470	1D6h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[22]
471	1D7h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[23]
472	1D8h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[24]
473	1D9h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[25]
474	1DAh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[26]
475	1DBh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[27]
476	1DCh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[28]
477	1DDh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[29]
478	1DEh		-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[30]
479	1DFh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[31]
480	1E0h	SDIOC1	SDIOC1_DMAR	-	√	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[0]
481	1E1h		SDIOC1_DMAW	-	√	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[1]
482	1E2h		SDIOC1_SD	√	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[2]
483	1E3h	SDIOC2	SDIOC2_DMAR	-	√	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[3]
484	1E4h		SDIOC2_DMAW	-	√	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[4]
485	1E5h		SDIOC2_SD	√	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[5]
486	1E6h	CAN	CAN_INT	√	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[6]
487	1E7h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[7]
488	1E8h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[8]
489	1E9h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[9]
490	1EAh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[10]
491	1EBh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[11]
492	1ECb	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[12]
493	1EDh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[13]
494	1EEh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[14]
495	1EFh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[15]
496	1F0h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[16]
497	1F1h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[17]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	可否选择为内部触发源	对应NVIC向量的中断选择寄存器 ^{*1}		
						NVIC向量 0~31	NVIC向量 32~127	NVIC向量 128~143
498	1F2h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[18]
499	1F3h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[19]
500	1F4h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[20]
501	1F5h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[21]
502	1F6h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[22]
503	1F7h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[23]
504	1F8h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[24]
505	1F9h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[25]
506	1FAh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[26]
507	1FBh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[27]
508	1FCh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[28]
509	1FDh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[29]
510	1FEh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[30]
511	1FFh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[31]

表 10-2 中断事件请求序号与选择

*1: 中断选择寄存器选择的中断事件请求序号如果未配置，则该寄存器或该位设定无效。

*2: AOS_STRG 由软件设定外设触发事件寄存器(INTSFTTRG)的 SFTG 位产生。

10.4 功能说明

10.4.1 不可屏蔽中断

不可屏蔽中断源如下：

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

不可屏蔽中断具有最高的优先级。由于不可屏蔽中断可选择多个中断事件请求，各个中断事件请求状态可以通过查询标志寄存器（INT_NMIFR）来确定。请在不可屏蔽中断处理退出前确认所有的标志位都为“0”。不可屏蔽中断默认为禁止状态，可以通过 ICG 或控制寄存器进行设定。

使用寄存器设定时，请按照下面流程设定：

1. 需要使用 NMI 管脚时，先将 INT_NMICR.NFEN 位清“0”，禁止数字滤波器；设定 INT_NMICR 寄存器的 NMITRG 位，选择 NMI 触发边沿；设定 SMPCLK 位选择数字滤波器的采样时钟；设定 NFEN 位，使能数字滤波器。
2. 使用其他不可屏蔽中断事件请求时，请配置相应功能。
3. 对 INT_NMICFR 各寄存器位写“1”，清除 INT_NMIFR 标志寄存器位，防止误动作。
4. 通过设定 INT_NMIENR 选择寄存器来使能不可屏蔽中断事件。

注意：

- 一旦 INT_NMIENR 相应位被设定为“1”后，将不能被更改，除非用 RESET 来复位。ICG 设定只针对外部 NMI 管脚，通过配置 ICG 寄存器 ICG1.NMIIICGENA 位，使能

ICG 设定。设定 ICG1.NMITRG 选择 NMI 触发边沿；设定 ICG1.SMPCLK 位选择数字滤波器采样时钟；设定 ICG1.NFEN 位，使能数字滤波器；设定 ICG1.NMIENR 位，使能 NMI 管脚中断。ICG 设定后，寄存器设定无效。ICG1 的寄存器说明，请参考初始化配置（ICG）章节。

10.4.2 外部管脚中断事件请求

需要使用外部管脚中断事件请求时，请按照如下流程设定：

1. 清除 INT_EIRQCRm.EFEN 位 ($m=0\sim15$)，禁止数字滤波器。
2. 设定 INT_EIRQCRm 的 IRQTRG[1:0]位，选择触发边沿或电平；设定 SMPCLK[1:0]位，选择数字滤波器采样时钟；设定 EFEN 位，使能数字滤波器。

10.4.3 中断源选择

本中断控制器共使用了 144 个中断向量，提供了 3 种中断事件请求选择方式，通过灵活的组合来满足各种中断配置需求。

第一种方式

共 32 个中断向量，所有中断事件请求任选 1 作为中断源，通过中断/事件选择寄存器 INT_SEL0~31 进行选择，通过 INT_IER 寄存器使能，对应 NVIC 的中断向量为 0~31；

第二种方式

共 96 个中断向量，32 选 1 作为中断源，通过中断选择寄存器 INT_SEL32~127 进行选择，对应的中断向量为 32~127；

第三种方式

共 16 个中断向量，32 个外设中断事件请求共享 1 个中断向量，各外设都可以申请中断，通过外设标志位区分，通过中断使能寄存器 INT_VSSEL128~143 进行中断事件请求使能，对应 NVIC 的中断向量为 128~143。选择有外设标志位的中断事件请求，具体参考 10.3.2 中断事件请求序号中 NVIC 向量 128~143 列。

具体中断向量分配请参考 10.3.2 中断事件请求序号章节。

10.4.4 软件中断

软件中断功能可以通过直接写软件中断控制寄存器 INT_SWIER 发生一次中断事件请

求，由 INT_IER 中断使能位控制许可。共配置 32 个软件中断事件请求，对应的中断向量为 0~31，具体请参考 10.3.1 中断向量表章节。

10.4.5 中断/事件选择

中断选择寄存器 INT_SEL0~31 选择的中断与软件中断共用 NVIC 的 0~31 号中断向量，由 INT_IER 中断使能寄存器控制许可；同时这些中断事件请求也可被选择为事件输入来唤醒内核（WFE），通过事件使能寄存器 INT_EVTER 选择，具体框图如下。

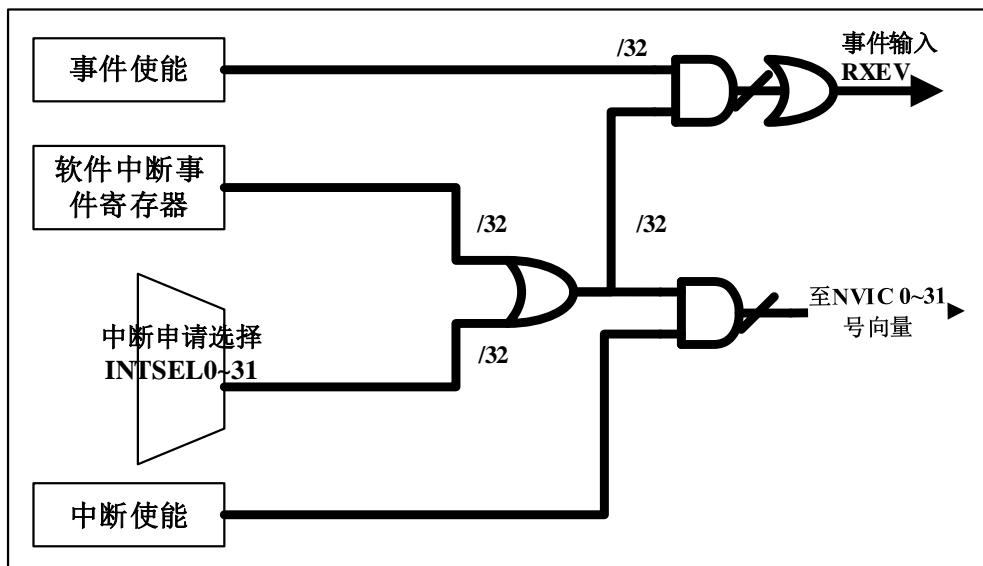


图 10-2 中断事件选择

10.4.6 WFE 唤醒事件管理

该 MCU 能够处理事件来唤醒内核（WFE）。唤醒事件可通过从 NVIC 的中断输入或事件输入两种方式产生。设定流程如下。

- 从 NVIC 的中断输入唤醒 WFE。在外设的控制寄存器使能一个中断，根据选择的中断向量设定 INT_SEL n 和 INT_IER 寄存器，但不在 NVIC 中使能，INT_EVTER 也设定为非使能，同时使能 Cortex™-M4F 系统控制寄存器 SCR 中的 SEVONPEND 位。当 MCU 从 WFE 恢复时，需要清除相应外设的中断标志位和 NVIC 中断标志寄存器。

WFE 停止模式进入与唤醒流程如下：

- 1) 设定停止模式寄存器；
- 2) 设定停止模式唤醒寄存器 INT_WUPEN；

- 3) 设定引脚 EIRQ 输入与 EIRQ 控制寄存器 INT_EIRQCRn;
 - 4) 选择 INT_SELn 选择相应的 EIRQ 中断事件请求序号;
 - 5) INT_IER 寄存器为使能对应的中断事件请求, INT_EVTER 非使能;
 - 6) 设置 SCR 中的 SEVONPEND 位为“1”;
 - 7) 执行如下操作确保系统进入停止模式:

```
__SEV();      设置内部事件寄存器
__WFE();      清除事件寄存器
__WFI();      系统进入停止模式
```
 - 8) 等待所选择的中断事件请求发生, 系统将从停止模式唤醒, 但不进入中断处理子程序。
- 从 NVIC 的事件输入唤醒 WFE。配置一个中断事件请求为事件输入, 通过事件使能寄存器 INT_EVTER 使能。当 CPU 从 WFE 恢复时, 需要清除相应外设的中断标志位。
- WFE 停止模式唤醒流程如下:
- 1) 设定停止模式寄存器;
 - 2) 设定停止模式唤醒寄存器 INT_WUPEN;
 - 3) 设定引脚 EIRQ 输入与 EIRQ 控制寄存器 INT_EIRQCRn;
 - 4) 选择 INT_SELn 选择相应的 EIRQ 中断事件请求序号;
 - 5) INT_IER 寄存器为非使能状态, INT_EVTER 使能对应的中断事件请求;
 - 6) 执行如下操作确保系统进入停止模式:

```
__SEV();      设置内部事件寄存器
__WFE();      清除事件寄存器
__WFI();      系统进入停止模式
```
 - 7) 等待所选择的中断事件请求发生, 系统将从停止模式唤醒, 但不进入中断处理子程序。

10.4.7 数字滤波器

对于 NMI 和 EIRQx (x=0~15) 管脚事件输入, 可选择数字滤波器进行噪声滤波。滤波

器的采样时钟为 PCLK3，小于 3 个滤波周期的输入信号将被滤除。具体工作时序图如下：

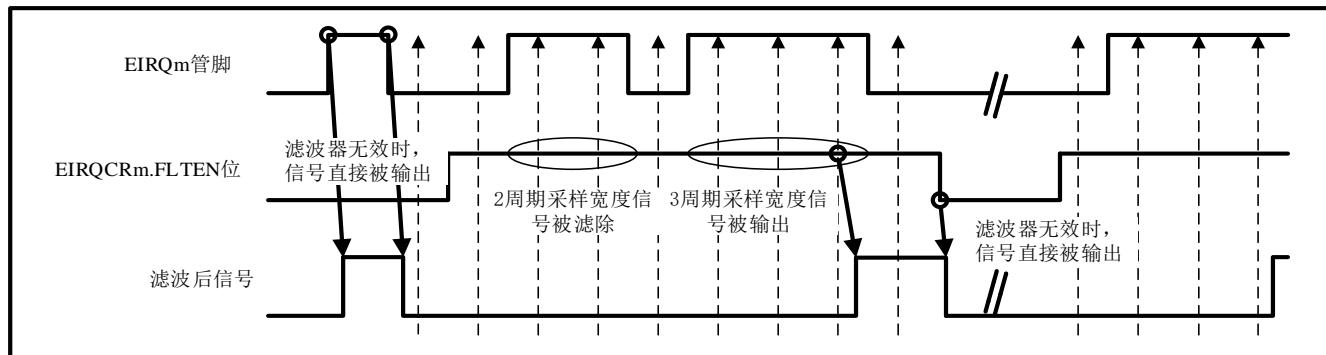


图 10-3 数字滤波器工作示意图

在进入停止模式前，请设定 INT_NMICR.NFEN 和 INT_EIRQCRm.EFEN，禁止使用数字滤波器。从停止模式返回后再使能数字滤波器。设定流程如下：

- 1) 设定系统停止模式寄存器
- 2) 配置停止模式唤醒中断
- 3) 停止数字滤波器
- 4) 执行 WFI，系统进入停止模式

10.4.8 低功耗模式返回

10.4.8.1 休眠模式返回

当选择事件中断源作为休眠模式返回条件时，需要进行如下设定：

- 选择事件为 CPU 中断源
- 使能 NVIC 中的控制寄存器
- 如需要使用不可屏蔽中断，需要设定 INT_NMIENR 使能寄存器

10.4.8.2 停止模式返回

可以选择不可屏蔽事件中断源或在 INT_WUPEN 寄存器中选择的可屏蔽事件源作为停止模式的返回条件。

从停止模式返回需要进行如下设定：

- 选择事件中断源作为停止模式的返回条件
 - A. 对于不可屏蔽中断，通过 INT_NMIENR 使能寄存器设定 NMI 管脚中断，SWDT，PVD1，PVD2 可以唤醒停止模式。
 - B. 对于可屏蔽中断，通过 INT_WUPEN 使能寄存器设定。
- 选择事件为 CPU 中断源
- 使能 NVIC 中的控制寄存器

对于未被选中的 EIRQ 管脚，由于时钟被关闭，将不会被检测。

10.4.8.3 掉电模式返回

掉电模式的返回可由第 5 章电源控制（PWC）章节所表示的条件、RES#管脚复位、上电复位以及低电压检测 0 条件返回。返回后 CPU 进入复位中断处理。详细说明请参考【电源控制（PWC）】章节。

10.4.8.4 不可屏蔽中断和 WFI 指令

在 WFI 指令执行前，请确认不可屏蔽中断标志寄存器 INT_NMIFR 的所有状态位为“0”。

10.4.9 内部触发事件

ADC、Timer、DMA、PORT、DCU 等周边外设除通过配置模块本身的寄存器开始工作以外，也可以通过写外设触发事件寄存器触发其他模块开始工作。使用内部触发事件需要清零 PWC_FCG0.AOS 位，使能外设电路触发功能。具体设定流程请参考各模块章节。

10.5 寄存器说明

下表为 INTC 寄存器列表。

INTC 基地址:0x4005_1000

寄存器名	符号	偏移地址	位宽	复位值
NMI管脚不可屏蔽中断控制寄存器	INT_NMICR	0x0000	32	0x0000_0000
不可屏蔽中断使能寄存器	INT_NMIENR	0x0004	32	0x0000_0000
不可屏蔽中断标志寄存器	INT_NMIFR	0x0008	32	0x0000_0000
不可屏蔽中断标志清除寄存器	INT_NMICFR	0x000C	32	0x0000_0000
外部管脚中断控制寄存器0	INT_EIRQCR0	0x0010	32	0x0000_0000
外部管脚中断控制寄存器1	INT_EIRQCR1	0x0014	32	0x0000_0000
外部管脚中断控制寄存器2	INT_EIRQCR2	0x0018	32	0x0000_0000
外部管脚中断控制寄存器3	INT_EIRQCR3	0x001C	32	0x0000_0000
外部管脚中断控制寄存器4	INT_EIRQCR4	0x0020	32	0x0000_0000
外部管脚中断控制寄存器5	INT_EIRQCR5	0x0024	32	0x0000_0000
外部管脚中断控制寄存器6	INT_EIRQCR6	0x0028	32	0x0000_0000
外部管脚中断控制寄存器7	INT_EIRQCR7	0x002C	32	0x0000_0000
外部管脚中断控制寄存器8	INT_EIRQCR8	0x0030	32	0x0000_0000
外部管脚中断控制寄存器9	INT_EIRQCR9	0x0034	32	0x0000_0000
外部管脚中断控制寄存器10	INT_EIRQCR10	0x0038	32	0x0000_0000
外部管脚中断控制寄存器11	INT_EIRQCR11	0x003C	32	0x0000_0000
外部管脚中断控制寄存器12	INT_EIRQCR12	0x0040	32	0x0000_0000
外部管脚中断控制寄存器13	INT_EIRQCR13	0x0044	32	0x0000_0000
外部管脚中断控制寄存器14	INT_EIRQCR14	0x0048	32	0x0000_0000
外部管脚中断控制寄存器15	INT_EIRQCR15	0x004C	32	0x0000_0000
停止模式唤醒事件使能寄存器	INT_WUPEN	0x0050	32	0x0000_0000
外部管脚中断标志寄存器	INT_EIFR	0x0054	32	0x0000_0000
外部管脚中断标志清除寄存器	INT_EICFR	0x0058	32	0x0000_0000
中断/事件选择寄存器0	INT_SEL0	0x005C	32	0x0000_01FF
中断/事件选择寄存器1	INT_SEL1	0x0060	32	0x0000_01FF

寄存器名	符号	偏移地址	位宽	复位值
中断/事件选择寄存器2	INT_SEL2	0x0064	32	0x0000_01FF
中断/事件选择寄存器3	INT_SEL3	0x0068	32	0x0000_01FF
中断/事件选择寄存器4	INT_SEL4	0x006C	32	0x0000_01FF
中断/事件选择寄存器5	INT_SEL5	0x0070	32	0x0000_01FF
中断/事件选择寄存器6	INT_SEL6	0x0074	32	0x0000_01FF
中断/事件选择寄存器7	INT_SEL7	0x0078	32	0x0000_01FF
中断/事件选择寄存器8	INT_SEL8	0x007C	32	0x0000_01FF
中断/事件选择寄存器9	INT_SEL9	0x0080	32	0x0000_01FF
中断/事件选择寄存器10	INT_SEL10	0x0084	32	0x0000_01FF
中断/事件选择寄存器11	INT_SEL11	0x0088	32	0x0000_01FF
中断/事件选择寄存器12	INT_SEL12	0x008C	32	0x0000_01FF
中断/事件选择寄存器13	INT_SEL13	0x0090	32	0x0000_01FF
中断/事件选择寄存器14	INT_SEL14	0x0094	32	0x0000_01FF
中断/事件选择寄存器15	INT_SEL15	0x0098	32	0x0000_01FF
中断/事件选择寄存器16	INT_SEL16	0x009C	32	0x0000_01FF
中断/事件选择寄存器17	INT_SEL17	0x00A0	32	0x0000_01FF
中断/事件选择寄存器18	INT_SEL18	0x00A4	32	0x0000_01FF
中断/事件选择寄存器19	INT_SEL19	0x00A8	32	0x0000_01FF
中断/事件选择寄存器20	INT_SEL20	0x00AC	32	0x0000_01FF
中断/事件选择寄存器21	INT_SEL21	0x00B0	32	0x0000_01FF
中断/事件选择寄存器22	INT_SEL22	0x00B4	32	0x0000_01FF
中断/事件选择寄存器23	INT_SEL23	0x00B8	32	0x0000_01FF
中断/事件选择寄存器24	INT_SEL24	0x00BC	32	0x0000_01FF
中断/事件选择寄存器25	INT_SEL25	0x00C0	32	0x0000_01FF
中断/事件选择寄存器26	INT_SEL26	0x00C4	32	0x0000_01FF
中断/事件选择寄存器27	INT_SEL27	0x00C8	32	0x0000_01FF
中断/事件选择寄存器28	INT_SEL28	0x00CC	32	0x0000_01FF
中断/事件选择寄存器29	INT_SEL29	0x00D0	32	0x0000_01FF
中断/事件选择寄存器30	INT_SEL30	0x00D4	32	0x0000_01FF
中断/事件选择寄存器31	INT_SEL31	0x00D8	32	0x0000_01FF

寄存器名	符号	偏移地址	位宽	复位值
中断选择寄存器32	INT_SEL32	0x00DC	32	0x0000_01FF
中断选择寄存器33	INT_SEL33	0x00E0	32	0x0000_01FF
中断选择寄存器34	INT_SEL34	0x00E4	32	0x0000_01FF
中断选择寄存器35	INT_SEL35	0x00E8	32	0x0000_01FF
中断选择寄存器36	INT_SEL36	0x00EC	32	0x0000_01FF
中断选择寄存器37	INT_SEL37	0x00F0	32	0x0000_01FF
中断选择寄存器38	INT_SEL38	0x00F4	32	0x0000_01FF
中断选择寄存器39	INT_SEL39	0x00F8	32	0x0000_01FF
中断选择寄存器40	INT_SEL40	0x00FC	32	0x0000_01FF
中断选择寄存器41	INT_SEL41	0x0100	32	0x0000_01FF
中断选择寄存器42	INT_SEL42	0x0104	32	0x0000_01FF
中断选择寄存器43	INT_SEL43	0x0108	32	0x0000_01FF
中断选择寄存器44	INT_SEL44	0x010C	32	0x0000_01FF
中断选择寄存器45	INT_SEL45	0x0110	32	0x0000_01FF
中断选择寄存器46	INT_SEL46	0x0114	32	0x0000_01FF
中断选择寄存器47	INT_SEL47	0x0118	32	0x0000_01FF
中断选择寄存器48	INT_SEL48	0x011C	32	0x0000_01FF
中断选择寄存器49	INT_SEL49	0x0120	32	0x0000_01FF
中断选择寄存器50	INT_SEL50	0x0124	32	0x0000_01FF
中断选择寄存器51	INT_SEL51	0x0128	32	0x0000_01FF
中断选择寄存器52	INT_SEL52	0x012C	32	0x0000_01FF
中断选择寄存器53	INT_SEL53	0x0130	32	0x0000_01FF
中断选择寄存器54	INT_SEL54	0x0134	32	0x0000_01FF
中断选择寄存器55	INT_SEL55	0x0138	32	0x0000_01FF
中断选择寄存器56	INT_SEL56	0x013C	32	0x0000_01FF
中断选择寄存器57	INT_SEL57	0x0140	32	0x0000_01FF
中断选择寄存器58	INT_SEL58	0x0144	32	0x0000_01FF
中断选择寄存器59	INT_SEL59	0x0148	32	0x0000_01FF
中断选择寄存器60	INT_SEL60	0x014C	32	0x0000_01FF
中断选择寄存器61	INT_SEL61	0x0150	32	0x0000_01FF

寄存器名	符号	偏移地址	位宽	复位值
中断选择寄存器62	INT_SEL62	0x0154	32	0x0000_01FF
中断选择寄存器63	INT_SEL63	0x0158	32	0x0000_01FF
中断选择寄存器64	INT_SEL64	0x015C	32	0x0000_01FF
中断选择寄存器65	INT_SEL65	0x0160	32	0x0000_01FF
中断选择寄存器66	INT_SEL66	0x0164	32	0x0000_01FF
中断选择寄存器67	INT_SEL67	0x0168	32	0x0000_01FF
中断选择寄存器68	INT_SEL68	0x016C	32	0x0000_01FF
中断选择寄存器69	INT_SEL69	0x0170	32	0x0000_01FF
中断选择寄存器70	INT_SEL70	0x0174	32	0x0000_01FF
中断选择寄存器71	INT_SEL71	0x0178	32	0x0000_01FF
中断选择寄存器72	INT_SEL72	0x017C	32	0x0000_01FF
中断选择寄存器73	INT_SEL73	0x0180	32	0x0000_01FF
中断选择寄存器74	INT_SEL74	0x0184	32	0x0000_01FF
中断选择寄存器75	INT_SEL75	0x0188	32	0x0000_01FF
中断选择寄存器76	INT_SEL76	0x018C	32	0x0000_01FF
中断选择寄存器77	INT_SEL77	0x0190	32	0x0000_01FF
中断选择寄存器78	INT_SEL78	0x0194	32	0x0000_01FF
中断选择寄存器79	INT_SEL79	0x0198	32	0x0000_01FF
中断选择寄存器80	INT_SEL80	0x019C	32	0x0000_01FF
中断选择寄存器81	INT_SEL81	0x01A0	32	0x0000_01FF
中断选择寄存器82	INT_SEL82	0x01A4	32	0x0000_01FF
中断选择寄存器83	INT_SEL83	0x01A8	32	0x0000_01FF
中断选择寄存器84	INT_SEL84	0x01AC	32	0x0000_01FF
中断选择寄存器85	INT_SEL85	0x01B0	32	0x0000_01FF
中断选择寄存器86	INT_SEL86	0x01B4	32	0x0000_01FF
中断选择寄存器87	INT_SEL87	0x01B8	32	0x0000_01FF
中断选择寄存器88	INT_SEL88	0x01BC	32	0x0000_01FF
中断选择寄存器89	INT_SEL89	0x01C0	32	0x0000_01FF
中断选择寄存器90	INT_SEL90	0x01C4	32	0x0000_01FF
中断选择寄存器91	INT_SEL91	0x01C8	32	0x0000_01FF

寄存器名	符号	偏移地址	位宽	复位值
中断选择寄存器92	INT_SEL92	0x01CC	32	0x0000_01FF
中断选择寄存器93	INT_SEL93	0x01D0	32	0x0000_01FF
中断选择寄存器94	INT_SEL94	0x01D4	32	0x0000_01FF
中断选择寄存器95	INT_SEL95	0x01D8	32	0x0000_01FF
中断选择寄存器96	INT_SEL96	0x01DC	32	0x0000_01FF
中断选择寄存器97	INT_SEL97	0x01E0	32	0x0000_01FF
中断选择寄存器98	INT_SEL98	0x01E4	32	0x0000_01FF
中断选择寄存器99	INT_SEL99	0x01E8	32	0x0000_01FF
中断选择寄存器100	INT_SEL100	0x01EC	32	0x0000_01FF
中断选择寄存器101	INT_SEL101	0x01F0	32	0x0000_01FF
中断选择寄存器102	INT_SEL102	0x01F4	32	0x0000_01FF
中断选择寄存器103	INT_SEL103	0x01F8	32	0x0000_01FF
中断选择寄存器104	INT_SEL104	0x01FC	32	0x0000_01FF
中断选择寄存器105	INT_SEL105	0x0200	32	0x0000_01FF
中断选择寄存器106	INT_SEL106	0x0204	32	0x0000_01FF
中断选择寄存器107	INT_SEL107	0x0208	32	0x0000_01FF
中断选择寄存器108	INT_SEL108	0x020C	32	0x0000_01FF
中断选择寄存器109	INT_SEL109	0x0210	32	0x0000_01FF
中断选择寄存器110	INT_SEL110	0x0214	32	0x0000_01FF
中断选择寄存器111	INT_SEL111	0x0218	32	0x0000_01FF
中断选择寄存器112	INT_SEL112	0x021C	32	0x0000_01FF
中断选择寄存器113	INT_SEL113	0x0220	32	0x0000_01FF
中断选择寄存器114	INT_SEL114	0x0224	32	0x0000_01FF
中断选择寄存器115	INT_SEL115	0x0228	32	0x0000_01FF
中断选择寄存器116	INT_SEL116	0x022C	32	0x0000_01FF
中断选择寄存器117	INT_SEL117	0x0230	32	0x0000_01FF
中断选择寄存器118	INT_SEL118	0x0234	32	0x0000_01FF
中断选择寄存器119	INT_SEL119	0x0238	32	0x0000_01FF
中断选择寄存器120	INT_SEL120	0x023C	32	0x0000_01FF
中断选择寄存器121	INT_SEL121	0x0240	32	0x0000_01FF

寄存器名	符号	偏移地址	位宽	复位值
中断选择寄存器122	INT_SEL122	0x0244	32	0x0000_01FF
中断选择寄存器123	INT_SEL123	0x0248	32	0x0000_01FF
中断选择寄存器124	INT_SEL124	0x024C	32	0x0000_01FF
中断选择寄存器125	INT_SEL125	0x0250	32	0x0000_01FF
中断选择寄存器126	INT_SEL126	0x0254	32	0x0000_01FF
中断选择寄存器127	INT_SEL127	0x0258	32	0x0000_01FF
向量共享中断选择寄存器128	INT_VSSEL128	0x025C	32	0x0000_0000
向量共享中断选择寄存器129	INT_VSSEL129	0x0260	32	0x0000_0000
向量共享中断选择寄存器130	INT_VSSEL130	0x0264	32	0x0000_0000
向量共享中断选择寄存器131	INT_VSSEL131	0x0268	32	0x0000_0000
向量共享中断选择寄存器132	INT_VSSEL132	0x026C	32	0x0000_0000
向量共享中断选择寄存器133	INT_VSSEL133	0x0270	32	0x0000_0000
向量共享中断选择寄存器134	INT_VSSEL134	0x0274	32	0x0000_0000
向量共享中断选择寄存器135	INT_VSSEL135	0x0278	32	0x0000_0000
向量共享中断选择寄存器136	INT_VSSEL136	0x027C	32	0x0000_0000
向量共享中断选择寄存器137	INT_VSSEL137	0x0280	32	0x0000_0000
向量共享中断选择寄存器138	INT_VSSEL138	0x0284	32	0x0000_0000
向量共享中断选择寄存器139	INT_VSSEL139	0x0288	32	0x0000_0000
向量共享中断选择寄存器140	INT_VSSEL140	0x028C	32	0x0000_0000
向量共享中断选择寄存器141	INT_VSSEL141	0x0290	32	0x0000_0000
向量共享中断选择寄存器142	INT_VSSEL142	0x0294	32	0x0000_0000
向量共享中断选择寄存器143	INT_VSSEL143	0x0298	32	0x0000_0000
软件中断事件寄存器	INT_SWIER	0x029C	32	0x0000_0000
事件使能寄存器	INT_EVTER	0x02A0	32	0x0000_0000
中断使能寄存器	INT_IER	0x02A4	32	0xFFFF_FFFF

10.5.1 NMI 管脚不可屏蔽中断控制寄存器 (INT_NMICR)

NMI Interrupt Control Register (INT_NMICR)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	NFEN	NMI数字滤波器使能	0: 禁止数字滤波器功能 1: 许可数字滤波器功能	R/W
b7~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5~b4	SMPCLK[1:0]	滤波器采样时钟选择	0 0: PCLK3 0 1: PCLK3/8 1 0: PCLK3/32 1 1: PCLK3/64	R/W
b3~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	NMITRG	触发边沿选择	0: 下降沿 1: 上升沿	R/W

10.5.2 不可屏蔽中断使能寄存器 (INT_NMIENR)

NMI Interrupt Enable Register (INT_NMIENR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	WDT ENR	BUS MENR	RECC ENR	REPE NR	-	-	XTAL STPE NR	-	PVD2 ENR	PVD1 ENR	SWD TENR	NMIE NR

位	标记	位名	功能	读写
b31~b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	WDTENR	WDT下溢/刷新中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	R/W
b10	BUSMENR	MPU主总线错误中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	R/W
b9	RECCENR	SRAM ECC校验错误中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	R/W
b8	REPENR	SRAM奇偶校验错误中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	R/W
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5	XATLSTPENR	检测主发振器停止中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	R/W
b4	Reserved	-	读出时为“0”,写入时写“0”	R
b3	PVD2ENR	低电压检测PVD2中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	R/W
b2	PVD1ENR	低电压检测PVD1中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	R/W
b1	SWDTENR	SWDT下溢/刷新中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	R/W
b0	NMIENR	NMI管脚中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	R/W

10.5.3 不可屏蔽中断标志寄存器 (INT_NMIFR)

NMI Flag Register (INT_NMIFR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	WDT FR	BUS MFR	REC CFR	REP F R	-	-	XTAL STPFR		PVD 2FR	PVD 1FR	SWD TFR	NMI FR

位	标记	位名	功能	读写
b31~b13	Reserved	-	读出时为“0”,写入时写“0”	R
b12	Reserved	-	读出时为“0”,写入时写“0”	R
b11	WDTFR	WDT下溢/刷新中断标志	0: 没有发生WDT下溢/刷新申请 1: 发生WDT下溢/刷新申请	R
b10	BUSMFR	MPU主总线错误中断标志	0: 没有发生MPU主总线错误申请 1: 发生MPU主总线错误申请	R
b9	RECCFR	SRAM DED校验错误中断标志	0: 没有发生SRAM DED校验错误申请 1: 发生SRAM DED校验错误申请	R
b8	REPFR	SRAM奇偶校验错误中断标志	0: 没有发生SRAM奇偶校验错误申请 1: 发生SRAM奇偶校验错误申请	R
b7	Reserved	-	读出时为“0”,写入时写“0”	R
b6	Reserved	-	读出时为“0”,写入时写“0”	R
b5	XTALSTPFR	检测主发振器停止中断标志	0: 没有发生检测主发振器停止申请 1: 发生检测主发振器停止申请	R
b4	Reserved	-	读出时为“0”,写入时写“0”	R
b3	PVD2FR	低电压检测PVD2中断标志	0: 没有发生低电压检测PVD2申请 1: 发生低电压检测PVD2申请	R
b2	PVD1FR	低电压检测PVD1中断标志	0: 没有发生低电压检测PVD1申请 1: 发生低电压检测PVD1申请	R
b1	SWDTFR	SWDT下溢/刷新中断标志	0: 没有发生SWDT下溢/刷新申请 1: 发生SWDT下溢/刷新申请	R
b0	NMIFR	NMI管脚中断标志	0: 没有发生NMI管脚申请 1: 发生NMI管脚申请	R

10.5.4 不可屏蔽中断标志清除寄存器 (INT_NMICFR)

NMI Clear Flag Register (INT_NMICFR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	WDT CFR	BUS MCFR	RECC CFR	REP CFR	-	-	XTALS TPCFR	-	PVD 2CFR	PVD 1CFR	SWD TCFR	NMI CFR

位	标记	位名	功能	读写
b31~b13	Reserved	-	读出时为“0”,写入时写“0”	R/W[注1]
b12	Reserved	-	读出时为“0”,写入时写“0”	R/W[注1]
b11	WDTCFR	WDT下溢/刷新中断标志清除	0: 无效 1: 清除WDT下溢/刷新标志	R/W[注1]
b10	BUSMCFR	MPU主总线错误中断标志清除	0: 无效 1: 清除MPU主总线错误标志	R/W[注1]
b9	RECCCFR	SRAM DED校验错误中断标志清除	0: 无效 1: 清除SRAM DED校验错误标志	R/W[注1]
b8	REPCFR	SRAM奇偶校验错误中断标志清除	0: 无效 1: 清除SRAM奇偶校验错误标志	R/W[注1]
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W[注1]
b6	Reserved	-	读出时为“0”,写入时写“0”	R/W[注1]
b5	XTALSTPCFR	检测主发振器停止中断标志清除	0: 无效 1: 清除检测主发振器停止标志	R/W[注1]
b4	Reserved	-	读出时为“0”,写入时写“0”	R
b3	PVD2CFR	低电压检测PVD2中断标志清除	0: 无效 1: 清除低电压检测PVD2标志	R/W[注1]
b2	PVD1CFR	低电压检测PVD1中断标志清除	0: 无效 1: 清除低电压检测PVD1标志	R/W[注1]
b1	SWDTCFR	SWDT下溢/刷新中断标志清除	0: 无效 1: 清除SWDT下溢/刷新标志	R/W[注1]
b0	NMICFR	NMI管脚中断标志清除	0: 无效 1: 清除NMI管脚标志	R/W[注1]

[注 1] 只可写“1”，读出时为“0”。

10.5.5 外部管脚中断控制寄存器 (INT_EIRQCRx) (x=0~15)

EIRQ Control Register (INT_EIRQCRx)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	EFEN	EIRQ数字滤波器使能	0: 禁止数字滤波器功能 1: 允许数字滤波器功能	R/W
b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5~b4	EISMPCLK[1:0]	滤波器采样时钟选择	0 0: PCLK3 0 1: PCLK3/8 1 0: PCLK3/32 1 1: PCLK3/64	R/W
b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1~b0	EIRQTRG[1:0]	触发选择	0 0: 下降沿 0 1: 上升沿 1 0: 双边沿 1 1: 低电平	R/W

10.5.6 外部管脚中断标志寄存器 (INT_EIFR)

EIRQ Flag Register (INT_EIFR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIFR1 5	EIFR1 4	EIFR1 3	EIFR1 2	EIFR1 1	EIFR1 0	EIFR9	EIFR8	EIFR7	EIFR6	EIFR5	EIFR4	EIFR3	EIFR2	EIFR1	EIFR0

位	标记	位名	功能	读写
b31~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b0	EIFR	EIFR标志位	0: EIRQ事件未发生, 或写EIFCR位清除位 1: 选择的EIRQ事件发生	R

10.5.7 外部管脚中断标志清除寄存器 (EICFR)

EIRQ Flag Clear Register (INT_EICFR)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b0	EICFR	EIFR清除位	0: 写“0”无效 1: 写“1”清除INT_EIFR寄存器	R/W

10.5.8 中断/事件选择寄存器 (INT_SEL0~31)

Interrupt Source Select Register (INT_SEL0~31)

复位值: 0x0000_01FF

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

位	标记	位名	功能	读写
b31~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8~b0	INTSEL[8:0]	中断事件请求选择	9'h000~9'h1FE: 10.3.2中断事件请求序号所对应的事件	R/W

10.5.9 中断选择寄存器 (INT_SEL32~127)

Interrupt Source Select Register (INT_SEL32~127)

复位值：0x0000_01FF

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

位	标记	位名	功能	读写
b31~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	INTSEL[8:0]	中断事件	分别选择10.3.2中断事件请求序号对应的事件，具体对应关系如下：	R/W
		请求选择	INT_SEL32~INT_SEL37: 选择9'h000~9'h01F对应的中断事件请求，其他选择无效。 INT_SEL38~INT_SEL43: 选择9'h020~9'h03F对应的中断事件请求，其他选择无效。 INT_SEL44~INT_SEL49: 选择9'h040~9'h05F对应的中断事件请求，其他选择无效。 INT_SEL50~INT_SEL55: 选择9'h060~9'h07F对应的中断事件请求，其他选择无效。 INT_SEL56~INT_SEL61: 选择9'h080~9'h09F对应的中断事件请求，其他选择无效。 INT_SEL62~INT_SEL67: 选择9'h0A0~9'h0BF对应的中断事件请求，其他选择无效。 INT_SEL68~INT_SEL73: 选择9'h0C0~9'h0DF对应的中断事件请求，其他选择无效。 INT_SEL74~INT_SEL79: 选择9'h0E0~9'h0FF对应的中断事件请求，其他选择无效。 INT_SEL80~INT_SEL85: 选择9'h100~9'h11F对应的中断事件请求，其他选择无效。 INT_SEL86~INT_SEL91: 选择9'h120~9'h13F对应的中断事件请求，其他选择无效。 INT_SEL92~INT_SEL97: 选择9'h140~9'h15F对应的中断事件请求，其他选择无效。 INT_SEL98~INT_SEL103: 选择9'h160~9'h17F对应的中断事件请求，其他选择无效。 INT_SEL104~INT_SEL109: 选择9'h180~9'h19F对应的中断事件请求，其他选择无效。 INT_SEL110~INT_SEL115: 选择9'h1A0~9'h1BF对应的中断事件请求，其他选择无效。 INT_SEL116~INT_SEL121: 选择9'h1C0~9'h1DF对应的中断事件请求，其他选择无效。 INT_SEL122~INT_SEL127: 选择9'h1E0~9'h1FF对应的中断事件请求，其他选择无效。	

10.5.10 向量共享中断选择寄存器 (INT_VSSEL128~143)

Vector Sharing Interrup Source Select Register (INT_VSSEL128~143)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
VSE															
L31	L30	L29	L28	L27	L26	L25	L24	L23	L22	L21	L20	L19	L18	L17	L16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
VSE															
L15	L14	L13	L12	L11	L10	L9	L8	L7	L6	L5	L4	L3	L2	L1	L0

位	标记	位名	功能	读写
b31~b0	VSEL[31:0]	中断使能	INT_VSSEL128: 每一位分别使能9'h000~9'h01F对应的中断事件请求。 INT_VSSEL129: 每一位分别使能9'h020~9'h03F对应的中断事件请求。 INT_VSSEL130: 每一位分别使能9'h040~9'h05F对应的中断事件请求。 INT_VSSEL131: 每一位分别使能9'h060~9'h07F对应的中断事件请求。 INT_VSSEL132: 每一位分别使能9'h080~9'h09F对应的中断事件请求。 INT_VSSEL133: 每一位分别使能9'h0A0~9'h0BF对应的中断事件请求。 INT_VSSEL134: 每一位分别使能9'h0C0~9'h0DF对应的中断事件请求。 INT_VSSEL135: 每一位分别使能9'h0E0~9'h0FF对应的中断事件请求。 INT_VSSEL136: 每一位分别使能9'h100~9'h11F对应的中断事件请求。 INT_VSSEL137: 每一位分别使能9'h120~9'h13F对应的中断事件请求。 INT_VSSEL138: 每一位分别使能9'h140~9'h15F对应的中断事件请求。 INT_VSSEL139: 每一位分别使能9'h160~9'h17F对应的中断事件请求。 INT_VSSEL140: 每一位分别使能9'h180~9'h19F对应的中断事件请求。 INT_VSSEL141: 每一位分别使能9'h1A0~9'h1BF对应的中断事件请求。 INT_VSSEL142: 每一位分别使能9'h1C0~9'h1DF对应的中断事件请求。 INT_VSSEL143: 每一位分别使能9'h1E0~9'h1FF对应的中断事件请求。 中断事件请求序号请参考表 10-2 中断事件请求序号与选择。	R/W

10.5.11 停止模式唤醒事件使能寄存器 (INT_WUPEN)

Soft-standby Wake Up Enable Register (INT_WUPEN)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	RXW UEEN	-	TMR0 WUEN	RTCP RDW UEEN	RTCA LMW UEEN	WKT MWU EN	CMPI OWUE N	PVD2 WUEN	PVD1 WUEN	SWD TWUE N
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIRQWUEN[15:0]															

位	标记	位名	功能	读写
b31~27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26	Reserved	-	读出时为“0”,写入时写“0”	R/W
b25	RXWUEN	USART1_WUPI停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	TMR0WUEN	TMR01_GCMA停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b22	RTCPRDWUEN	RTC_PRD停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b21	RTCALMWUEN	RTC_ALM停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
b20	WKTMWUEN	WKTM_PRD周期停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
B19	CMPIIWUEN	ACMP1停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
B18	PVD2WUEN	PVD_PVD2停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
B17	PVD1WUEN	PVD_PVD1停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
B16	SWDTWUEN	SWDT_REFUDF停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W
B15~b0	EIRQWUEN[15:0]	EIRQ停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	R/W

10.5.12 软件中断/事件寄存器 (INT_SWIER)

Software Interrupt & Event Register (INT_SWIER)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SWI E31	SWI E30	SWI E29	SWI E28	SWI E27	SWI E26	SWI E25	SWI E24	SWI E23	SWI E22	SWI E21	SWI E20	SWI E19	SWI E18	SWI E17	SWI E16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SWI E15	SWI E14	SWI E13	SWI E12	SWI E11	SWI E10	SWI E9	SWI E8	SWI E7	SWI E6	SWI E5	SWI E4	SWI E3	SWI E2	SWI E1	SWI E0

位	标记	位名	功能	读写
b31~b0	SWIE	软件中断/事件寄存器位	0: 无效 1: 软件中断事件发生 注意: 写“1”后发生软件中断/事件。 写“0”后清除。	R/W

10.5.13 事件使能寄存器 (INT_EVTER)

Event Enable Register (INT_EVTER)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EVT E31	EVT E30	EVT E29	EVT E28	EVT E27	EVT E26	EVT E25	EVT E24	EVT E23	EVT E22	EVT E21	EVT E20	EVT E19	EVT E18	EVT E17	EVT E16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EVT E15	EVT E14	EVT E13	EVT E12	EVT E11	EVT E10	EVT E9	EVT E8	EVT E7	EVT E6	EVT E5	EVT E4	EVT E3	EVT E2	EVT E1	EVT E0

位	标记	位名	功能	读写
b31~b0	EVTE	事件使能寄存器位	0: 事件选择禁止 1: 事件选择许可	R/W

10.5.14 中断使能寄存器 (INT_IER)

Interrupt Enable Register (INT_IER)

复位值: 0xFFFF_FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IER															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
IER															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

位	标记	位名	功能	读写
b31~b0	IER	中断使能寄存器位	寄存器位0~31分别对应NVIC中断向量0~31。当禁止时，各中断向量对应的选择寄存器INTSEL[8:0]选择的中断事件请求将不能被NVIC接收。 0: INTSEL[8:0]选择的中断事件请求与软件中断事件请求被禁止 1: INTSEL[8:0]选择的中断事件请求与软件中断事件请求被许可	R/W

10.6 使用注意事项

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

11 自动运行系统 (AOS)

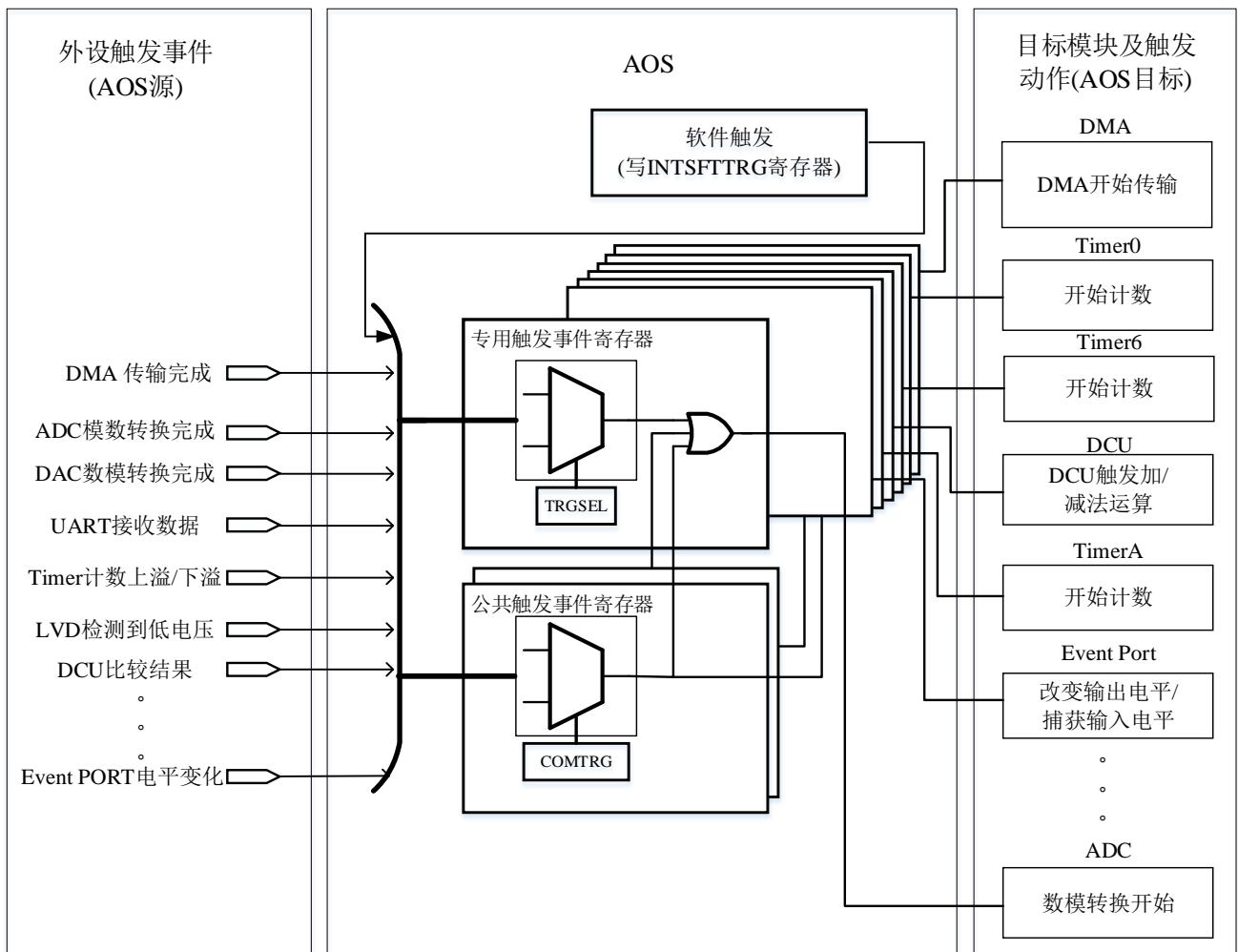
11.1 简介

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

11.1.1 功能概览

- 共 195 种 AOS 源，除特殊限制外，每个 AOS 目标可选其中一个作为触发源，另外可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 选择两个额外的触发源，3 个触发源中的任意一个发生触发事件时均可触发该 AOS 目标。所有 AOS 目标共享这两个公共触发源。
- 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 目标的外设电路动作如下：
 - 4 个 DCU 触发目标，用于触发 DCU1~DCU4
 - 9 个 DMA 触发目标，用于 2 个 4 通道 DMA 开始数据传输以及一个 DMA 事件触发通道重置
 - 2 个高级控制定时器(Timer6) 触发目标
 - 1 个通用定时器 0(Timer0) 触发目标
 - 2 个 Event Port 触发目标，其中 Event Port Group1 和 Event Port Group2 共用一个 AOS 目标，Event Port Group3 和 Event Port Group4 共用一个 AOS 目标
 - 2 个通用定时器 A(TimerA) 触发目标
 - 1 个温度传感器(OTS) 触发目标
 - 2 组每组 2 个 AD 触发目标，用于 AD1~AD2 序列触发

11.1.2 模块示意图



11.2 功能描述

11.2.1 AOS 源事件列表

AOS 源事件编号见【中断控制器 (INTC)】章中中断事件请求序号节表 10-2。表中“是否可选择为事件”列中打“√”的事件可以用作 AOS 源。

11.2.2 AOS 目标列表

模块	动作
DCU1	触发加/减法运算
DCU2	触发加/减法运算
DCU3	触发加/减法运算
DCU4	触发加/减法运算
DMA1	通道 0 开始传输 通道 1 开始传输 通道 2 开始传输 通道 3 开始传输
DMA2	通道 0 开始传输 通道 1 开始传输 通道 2 开始传输 通道 3 开始传输
DMA1&2	事件触发通道重置
Timer6	开始计数
Timer0	开始计数
Event Port	Event Port1&2 触发动作 Event Port3&4 触发动作
TimerA	开始计数/捕获
OTS	开始测温
ADC1	开始模数转换
ADC2	开始模数转换

11.3 动作说明

11.3.1 专用触发源

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

11.3.2 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_COMTRG1,AOS_COMTRG2)。用于实现多个 AOS 源触发同一 AOS 目标的功能。使用时首先在 AOS 目标专用外设触发源选择寄存器中将公共触发源使能位置为有效，再在公共触发源选择寄存器中写入 AOS 源对应的事件编号。当 AOS 源的事件发生时，这个事件将通过 AOS 的公共触发源传递给 AOS 目标，作为 AOS 目标的外设电路根据自身设定开始动作。当同时设置专用触发源和公共触发源后，可以最多同时有 3 个 AOS 源触发同一 AOS 目标，3 个 AOS 源中的任意一个发生触发事件时，都将触发该 AOS 目标。

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

11.4 寄存器说明

寄存器一览

寄存器基地址: 0x4001_0800

缩写	名称	偏移地址
INTSFTTRG	外设触发事件寄存器	0x00
DCU_TRGSELx(x=1~4)	DCU 触发源选择寄存器	0x04,0x08,0x0C,0x10
DMA1_TRGSELx(x=0~3)	DMA1 传输启动触发源选择寄存器	0x14,0x18,0x1C,0x20
DMA2_TRGSELx(x=0~3)	DMA2 传输启动触发源选择寄存器	0x24,0x28,0x2C,0x30
DMA_TRGSELRC	DMA 通道重置触发源选择寄存器	0x34
TMR6_HTSSRx(x=0~1)	Timer6 硬件触发事件选择寄存器	0x38,0x3C
TMR0_HTSRR	Timer0 触发选择寄存器	0x40
PEVNTTRGSR12	Event Port1,2 触发源选择寄存器	0x44
PEVNTTRGSR34	Event Port3,4 触发源选择寄存器	0x48
TMRA_HTSR0	TimerA 内部触发事件选择寄存器 0	0x4C,
TMRA_HTSR1	TimerA 内部触发事件选择寄存器 1	0x50
OTS_TRG	OTS 触发源选择寄存器	0x54
ADC1_ITRGSELRx(x=0,1)	A/D1 启动片内触发源选择寄存器	0x58, 0x5C
ADC2_ITRGSELRx(x=0,1)	A/D2 启动片内触发源选择寄存器	0x60, 0x64
AOS_COMTRG1	公共触发源选择寄存器 1	0x68
AOS_COMTRG2	公共触发源选择寄存器 2	0x6C

11.4.1 外设触发事件寄存器(INTSFTTRG)

寄存器说明：写该寄存器将产生一次触发事件。

偏移地址：0x00

复位值：0x0000_0000

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

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

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

寄存器说明：在 DCU 选择硬件触发启动模式后，将要产生触发的事件的编号写入该寄存器中，当编号对应的外设电路事件发生后，DCU 将由该事件触发启动并进行运算。

当 DCU_TRGSEL1 写入事件编号，该编号事件发生时，会触发 DCU1；当 DCU_TRGSEL2 写入事件编号，该编号事件发生时，会触发 DCU2；当 DCU_TRGSEL3 写入事件编号，该编号事件发生时，会触发 DCU3；当 DCU_TRGSEL4 写入事件编号，该编号事件发生时，会触发 DCU4。

偏移地址：0x04, 0x08, 0x0C, 0x10

复位值：0x0000_01FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止AOS_COMTRG2的公共触发事件触发DCU 1: 允许AOS_COMTRG2的公共触发事件触发DCU	R/W
b30	COMEN[0]	公共触发使能	0: 禁止AOS_COMTRG1的公共触发事件触发DCU 1: 允许AOS_COMTRG1的公共触发事件触发DCU	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号 具体编号请参考中断控制器（INTC）章节。	R/W

11.4.3 DMA1 传输启动触发源选择寄存器(DMA1_TRGSELx)(x=0~3)

寄存器说明：在 DMA1 选择硬件触发启动模式后，将要产生触发的事件的编号写入该寄存器中，当编号对应的外设电路事件发生后，DMA1 将由该事件触发启动并进行传输。

偏移地址：0x14, 0x18, 0x1C, 0x20

复位值: 0x0000_01FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止AOS_COMTRG2的公共触发事件触发DMA1传输 1: 允许AOS_COMTRG2的公共触发事件触发DMA1传输	R/W
b30	COMEN[0]	公共触发使能	0: 禁止AOS_COMTRG1的公共触发事件触发DMA1传输 1: 允许AOS_COMTRG1的公共触发事件触发DMA1传输	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	选择启动对应通道进行传输的事件编号。 具体编号请参考中断控制器（INTC）章节。	R/W

11.4.4 DMA2 传输启动触发源选择寄存器(DMA2_TRGSELx)(x=0~3)

寄存器说明：在 DMA2 选择硬件触发启动模式后，将要产生触发的事件的编号写入该寄存器中，当编号对应的外设电路事件发生后，DMA2 将由该事件触发启动并进行传输。

偏移地址：0x24, 0x28, 0x2C,, 0x30

复位值: 0x0000_01FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止AOS_COMTRG2的公共触发事件触发DMA2传输 1: 允许AOS_COMTRG2的公共触发事件触发DMA2传输	R/W
b30	COMEN[0]	公共触发使能	0: 禁止AOS_COMTRG1的公共触发事件触发DMA2传输 1: 允许AOS_COMTRG1的公共触发事件触发DMA2传输	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	选择启动对应通道进行传输的事件编号。 具体编号请参考中断控制器（INTC）章节。	R/W

11.4.5 DMA 通道重置触发源选择寄存器(DMA_TRGSELRC)

寄存器说明：在 DMA 选择硬件触发启动模式后，将要产生触发的事件的编号写入该寄存器中，当编号对应的外设电路事件发生后，DMA 将由该事件触发进行通道重置。

DMA1 和 DMA2 共用该寄存器

偏移地址：0x34

复位值: 0x0000_01FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止AOS_COMTRG2的公共触发事件触发DMA通道重置 1: 允许AOS_COMTRG2的公共触发事件触发DMA通道重置	R/W
b30	COMEN[0]	公共触发使能	0: 禁止AOS_COMTRG1的公共触发事件触发DMA通道重置 1: 允许AOS_COMTRG1的公共触发事件触发DMA通道重置	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	选择触发通道进行重新设置的事件编号。 具体编号请参考中断控制器（INTC）章节。 DMA_1, DMA_2共用一个重置触发源。	R/W

11.4.6 Timer6 硬件触发事件选择寄存器(TMR6-HTSSRx)(x=0~1)

寄存器说明：在 Timer6 选择硬件触发启动模式后，将要产生触发的事件的编号写入该寄存器中，当编号对应的外设电路事件发生后，Timer6 将由该事件触发启动。

偏移地址：0x38, 0x3C

复位值：0x0000_01FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止AOS_COMTRG2的公共触发事件触发TMR6 1: 允许AOS_COMTRG2的公共触发事件触发TMR6	R/W
b30	COMEN[0]	公共触发使能	0: 禁止AOS_COMTRG1的公共触发事件触发TMR6 1: 允许AOS_COMTRG1的公共触发事件触发TMR6	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	触发源编号写入 具体编号请参考中断控制器（INTC）章节。	R/W

注意：

- 触发选择寄存器（TMR6-HTSSR0~1）是 2 个独立于单元的寄存器，为 3 个单元 Timer6 所共有。

11.4.7 Timer0 硬件触发事件选择寄存器(TMR0-HTSSR)

寄存器说明：在 Timer0 选择硬件触发启动模式后，将要产生触发的事件的编号写入该寄存器中，当编号对应的外设电路事件发生后，Timer0 将由该事件触发启动。

偏移地址：0x40

复位值：0x0000_01FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止AOS_COMTRG2的公共触发事件触发TMR0 1: 允许AOS_COMTRG2的公共触发事件触发TMR0	R/W
b30	COMEN[0]	公共触发使能	0: 禁止AOS_COMTRG1的公共触发事件触发TMR0 1: 允许AOS_COMTRG1的公共触发事件触发TMR0	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	触发源编号写入 具体编号请参考中断控制器（INTC）章节。	R/W

注意：

- 触发选择寄存器 (TMR0-HTSSR) 是一个独立的寄存器，为 2 个单元的 Timer0 所共有。

11.4.8 Event Port 触发源选择寄存器(PEVNTTRGSR12, PEVNTTRGSR34)

寄存器说明：设置相应事件编号，用以触发 Event Port 输出指定电平，或者锁存住 I/O 端口输入状态。PEVNTTRGSR12 设置 Event Port1 和 2 的触发源，PEVNTTRGSR34 设置 Event Port3 和 4 的触发源。

偏移地址：0x44, 0x48

复位值: 0x0000_01FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止AOS_COMTRG2的公共触发事件触发Event Port 1: 允许AOS_COMTRG2的公共触发事件触发Event Port	R/W
b30	COMEN[0]	公共触发使能	0: 禁止AOS_COMTRG1的公共触发事件触发Event Port 1: 允许AOS_COMTRG1的公共触发事件触发Event Port	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	设置相应事件编号，用以触发Event Port输出指定电平，或者锁存住 I/O 端口输入状态。 PEVNTTRGSR12设置Event Port1和2的触发源，PEVNTTRGSR34设置 Event Port3和4的触发源。 具体编号请参考中断控制器（INTC）章节。	R/W

11.4.9 TimerA 内部触发事件选择寄存器 0(TMRA_HTSSR0)

寄存器说明：在 TimerA 选择硬件触发启动模式后，将要产生触发的事件的编号写入该寄存器中，当编号对应的外设电路事件发生后，TimerA 将由该事件触发启动。

偏移地址：0x4C

复位值：0x0000_01FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止AOS_COMTRG2的公共触发事件触发TMRA计数器 1: 允许AOS_COMTRG2的公共触发事件触发TMRA计数器	R/W
b30	COMEN[0]	公共触发使能	0: 禁止AOS_COMTRG1的公共触发事件触发TMRA计数器 1: 允许AOS_COMTRG1的公共触发事件触发TMRA计数器	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	CNTTRGSEL[8:0]	计数器触发事件触发源选择	计数器触发事件触发源编号写入 具体编号请参考中断控制器（INTC）章节。	R/W

注意：

- 内部触发事件选择寄存器（TMRA_HTSSR0~1）是两个独立的寄存器，为 6 个单元的 TimerA 所共有。

11.4.10 TimerA 内部触发事件选择寄存器 1(TMRA_HTSSR1)

寄存器说明：在 TimerA 选择硬件触发启动模式后，将要产生触发的事件的编号写入该寄存器中，当编号对应的外设电路事件发生后，TimerA 将由该事件触发启动。

偏移地址：0x50

复位值: 0x0000_01FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止AOS_COMTRG2的公共触发事件触发TMRA捕获动作 1: 允许AOS_COMTRG2的公共触发事件触发TMRA捕获动作	R/W
b30	COMEN[0]	公共触发使能	0: 禁止AOS_COMTRG1的公共触发事件触发TMRA捕获动作 1: 允许AOS_COMTRG1的公共触发事件触发TMRA捕获动作	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	ICPTRGSEL[8:0]	捕获动作触发事件触发源选择	捕获动作触发事件触发源编号写入 具体编号请参考中断控制器（INTC）章节。	R/W

11.4.11 OTS 触发源选择寄存器(OTS_TRG)

寄存器说明：在 OTS 选择硬件触发启动模式后，将要产生触发的事件的编号写入该寄存器中，当编号对应的外设电路事件发生后，OTS 将由该事件触发启动。

偏移地址：0x54

复位值: 0x0000_01FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止AOS_COMTRG2的公共触发事件触发OTS 1: 允许AOS_COMTRG2的公共触发事件触发OTS	R/W
b30	COMEN[0]	公共触发使能	0: 禁止AOS_COMTRG1的公共触发事件触发OTS 1: 允许AOS_COMTRG1的公共触发事件触发OTS	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	选择硬件触发启动时的触发源编号， 具体编号请参考中断控制器（INTC）章节。	R/W

11.4.12 A/D1 转换开始片内触发源选择寄存器 ADC1_ITRGSELRx(x=0,1)

寄存器说明：在 ADC1 选择硬件触发启动模式后，将要产生触发的事件的编号写入该寄存器中，当编号对应的外设电路事件发生后，ADC1 将由该事件触发启动。

偏移地址：0x58, 0x5C

复位值: 0x0000_01FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止AOS_COMTRG2的公共触发事件触发ADC1 1: 允许AOS_COMTRG2的公共触发事件触发ADC1	R/W
b30	COMEN[0]	公共触发使能	0: 禁止AOS_COMTRG1的公共触发事件触发ADC1 1: 允许AOS_COMTRG1的公共触发事件触发ADC1	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号 具体编号请参考中断控制器（INTC）章节。	R/W

11.4.13 A/D2 转换开始片内触发源选择寄存器 ADC2_ITRGSELRx(x=0,1)

寄存器说明：在 ADC2 选择硬件触发启动模式后，将要产生触发的事件的编号写入该寄存器中，当编号对应的外设电路事件发生后，ADC2 将由该事件触发启动。

偏移地址：0x60, 0x64

复位值: 0x0000_01FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0: 禁止AOS_COMTRG2的公共触发事件触发ADC2 1: 允许AOS_COMTRG2的公共触发事件触发ADC2	R/W
b30	COMEN[0]	公共触发使能	0: 禁止AOS_COMTRG1的公共触发事件触发ADC2 1: 允许AOS_COMTRG1的公共触发事件触发ADC2	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号 具体编号请参考中断控制器（INTC）章节。	R/W

11.4.14 公共触发源选择寄存器 1(AOS_COMTRG1)

寄存器说明：在 AOS_COMTRG1 写入将要产生触发的事件的编号，当编号对应的外设电路事件发生后，如果一个或多个 AOS 目标的专用触发源选择寄存器的 COMEN[1] 位值为 1，则编号对应的外设电路事件将触发这一个或多个 AOS 目标启动。

偏移地址：0x68

复位值：0x0000_01FF

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

位	标记	位名	功能	读写
b31~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	COMTRG[8:0]	触发源选择	写入要选择的外设电路事件的编号 具体编号请参考中断控制器（INTC）章节。	R/W

11.4.15 公共触发源选择寄存器 2(AOS_COMTRG2)

寄存器说明：在 AOS_COMTRG2 写入将要产生触发的事件的编号，当编号对应的外设电路事件发生后，如果一个或多个 AOS 目标的专用触发源选择寄存器的 COMEN[0] 位值为 1，则编号对应的外设电路事件将触发这一个或多个 AOS 目标启动。

偏移地址：0x6C

复位值：0x0000_01FF

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

位	标记	位名	功能	读写
b31~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	COMTRG[8:0]	触发源选择	写入要选择的外设电路事件的编号 具体编号请参考中断控制器（INTC）章节。	R/W

12 键盘扫描控制模块（KEYSCAN）

12.1 简介

本产品搭载键盘控制模块（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=0~7)驱动，而行 KEYIN_n(n=0~15)则作为 EIRQ_n(n=0~15)输入被检测。本模块通过行扫描查询法实现按键识别功能。

KEYSCAN 主要特性：

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

12.2 KEYS defense 系统框图

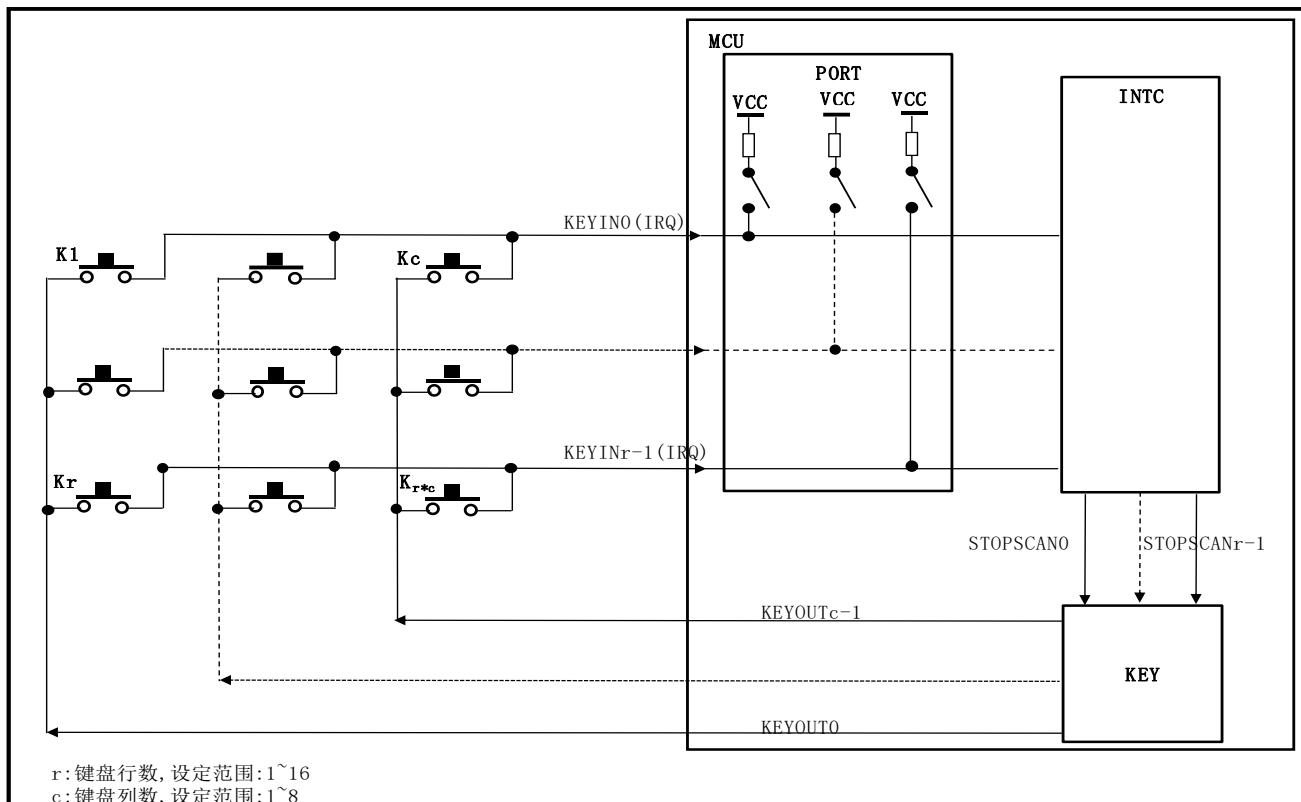


图 12-1 KEYS defense 系统框图

12.3 管脚说明

管脚名	方向	功能描述
KEYINn	输入	键盘行输入信号
KEYOUTm	输出	键盘列输出信号

n:0~15 m:0~7

表 12-1 KEYS defense 管脚说明

12.4 功能说明

本章将对键盘扫描功能和按键识别功能详细说明。

12.4.1 按键识别功能

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

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

12.4.2 键盘扫描功能

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

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

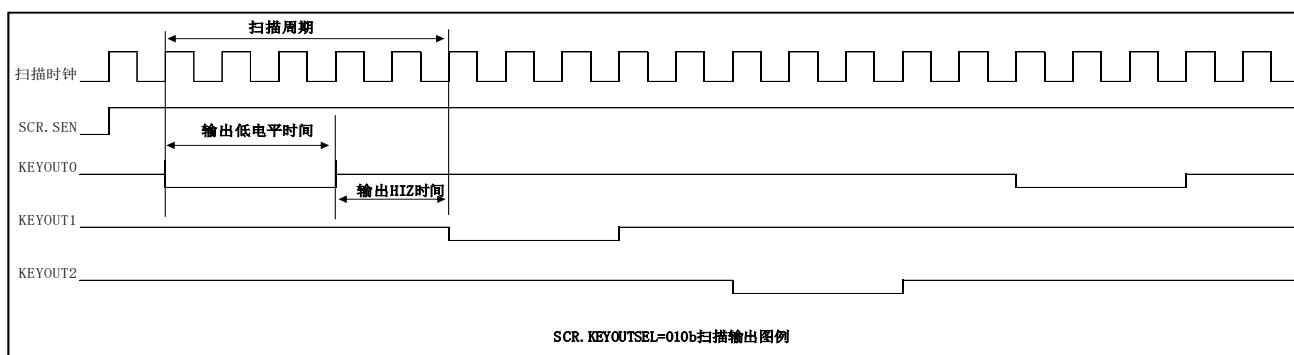


图 12-2 键盘扫描功能示意图

12.4.3 使用上的注意事项

本模块驱动键盘列，而键盘行检测则由中断控制模块(INTC)的外部 EIRQ 功能实现，EIRQ 需要选择下降沿检测，并开启数字滤波功能，设定合适的滤波时间。STOP 模式下如果使用此功能，则需设定好扫描相关参数后，选择内部低速振荡器 LRC 或者外部低速振荡器 XTAL32 时钟作为扫描时钟。如果使用芯片内部上拉电阻，则请参考 PORT 特性，选择合适的扫描时间以及滤波时间。

12.5 寄存器说明

KEYSCAN_BASE_ADDR: 0x4005_0C00

寄存器名	符号	偏移地址	位宽	复位值
KEYSCAN扫描控制寄存器	KEYSCAN_SCR	0x00	32	0x0000_0000
KEYSCAN扫描使能寄存器	KEYSCAN_SER	0x04	32	0x0000_0000
KEYSCAN扫描状态寄存器	KEYSCAN_SSR	0x08	32	0x0000_0000

表 12-2 KEYSCAN 寄存器一览表

12.5.1 KEYS defense control register (KEYSCAN_SCR)

KEYSCAN Scan Control Register

偏移地址: 0x00

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
T_HIZ[2:0]		T_LLEVEL[4:0]				-	-	CKSEL[1:0]		-	KEYOUTSEL[2:0]				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
KEYINSEL[15:0]															

位	标记	位名	功能	读写
b31~b29	T_HIZ[2:0]	输出HIZ时间	KEYOUT输出低电平HIZ时间（扫描时钟数） 扫描周期=输出低电平时间+输出HIZ时间 设定值: HIZ周期数 000b: 4 001b: 8 010b: 16 011b: 32 100b: 64 101b: 256 110b: 512 111b: 1024 注意: SCR.T-HIZ[2: 0]只能在SER.SEN=0时设定有效	R/W
b28~b24	T_LLEVEL[4:0]	输出低电平时间	KEYOUT输出低电平时间（扫描时钟数） 扫描周期=输出低电平时间+输出HIZ时间 输出低电平时间=2的T_LLEVEL次某个扫描时钟数 注意: SCR.T-LLEVEL[4: 0]只能在SER.SEN=0时设定有效,且00000b和00001b设定禁止, 最大可以设定值为11000b	R/W
b23~b22	Reserved	-	读出时为“0”,写入时写“0”	R
b21~b20	CKSEL[1:0]	扫描时钟源选择位	扫描时钟源选择位 00b:系统时钟HCLK 01b:内部低速振荡器LRC 10b:外部低速振荡器XTAL32 11b:设定禁止 注意: SCR.CKSEL[1: 0]只能在SER.SEN=0时设定有效	R/W
b19	Reserved	-	读出时为“0”,写入时写“0”	R

b18~b16	KEYOUTSEL[2:0]	输出选择	KEYOUT输出选择位 设定值： 输出选择 000b: 禁止 001b: KEYOUT0~KEYOUT1 010b: KEYOUT0~KEYOUT2 011b: KEYOUT0~KEYOUT3 100b: KEYOUT0~KEYOUT4 101b: KEYOUT0~KEYOUT5 110b: KEYOUT0~KEYOUT6 111b: KEYOUT0~KEYOUT7	R/W
			注意： SCR.KEYOUTSEL[2: 0]只能在SER.SEN=0时设定有效	
b15~b0	KEYINSEL[15:0]	行输入选择位	行输入选择位，选择的行作为键盘阵列的行使用，并作为 EIRQn (n: 0~15) 被检测 KEYINSEL[n]=0: KEYINSEL[n]不用作键盘阵列的行使用 KEYINSEL[n]=1: KEYINSEL[n]用作键盘阵列的行使用 n: 范围0~15	R/W
			注意： SCR. KEYINSEL[15:0]只能在SER.SEN=0时设定有效	

12.5.2 KEYS defense 扫描使能寄存器 (KEYSCAN_SER)

KEYSCAN Scan Enable Register

偏移地址: 0x04

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b30	Reserved	-	读出时为“0”,写入时写“0”	R
b0	SEN	扫描使能位	扫描使能位 0: 扫描禁止 1: 扫描使能	R/W

12.5.3 KEYS defense status register (KEYSCAN_SSR)

KEYSCAN Scan Status Register

偏移地址: 0x08

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b3	Reserved	-	读出时为“0”，写入时写“0”	R
b2~b0	INDEX[2:0]	当前工作的 SCAN管脚索引位 引位	当前工作的SCAN管脚索引位 000: 当前工作的SCAN管脚为KEYOUT0 001: 当前工作的SCAN管脚为KEYOUT1 010: 当前工作的SCAN管脚为KEYOUT2 011: 当前工作的SCAN管脚为KEYOUT3 100: 当前工作的SCAN管脚为KEYOUT4 101: 当前工作的SCAN管脚为KEYOUT5 110: 当前工作的SCAN管脚为KEYOUT6 111: 当前工作的SCAN管脚为KEYOUT7	R
			注意: SSR.INDEX[2:0]位为只读寄存器，且只有在SER.SEN=1时读取的数据才有意义	

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。

本芯片内置了四个针对主机的 MPU 单元和一个针对 IP 的 MPU 单元。

模块	内容
ARM MPU	CPU的存储保护单元 8区域，详见ARM MPU说明
系统DMA_1 MPU: SMPU1	系统DMA_1的存储保护单元 16区域，8区域系统DMA专用，8区域所有DMA共用
系统DMA_2 MPU: SMPU2	系统DMA_2的存储保护单元 16区域，8区域系统DMA专用，8区域所有DMA共用
USBFS-DMA MPU: FMPU	USBFS-DMA的存储保护单元 8区域，所有DMA共用
IPMPU	针对系统IP和安全相关IP的访问保护单元

其中 ARM MPU 提供 CPU 对全部 4G 地址空间的访问权限控制，介绍略。

SMPU1/SMPU2/FMPU 分别提供系统 DMA_1/系统 DMA_2/USBFS-DMA 对全部 4G 地址空间的读写访问权限控制。对禁止空间发生访问时，可以设置 MPU 动作为无视/总线错误/不可屏蔽中断/复位。

IPMPU 提供非特权模式时对系统 IP 和安全相关 IP 的访问权限控制。

13.2 功能描述

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址和区域大小，可设的范围为 32Byte~4GByte，大小必须为 2^nByte ($n=5\sim32$)，对应的基址低 n 位为 0。

没有被任何区域覆盖的地址空间称为背景区域。

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

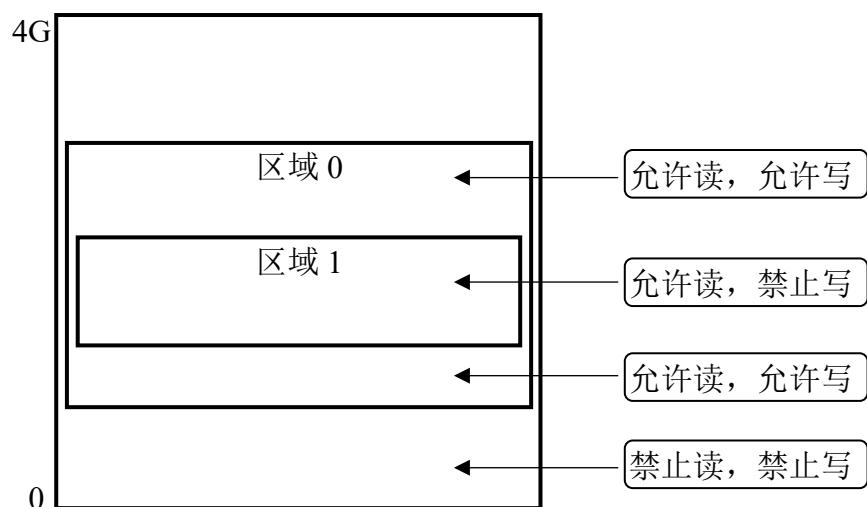
SMPU1/SMPU2/FMPU 可以独立使能。

建议在设置好区域范围/权限设置/动作选择后再使能 MPU。

13.3 应用举例

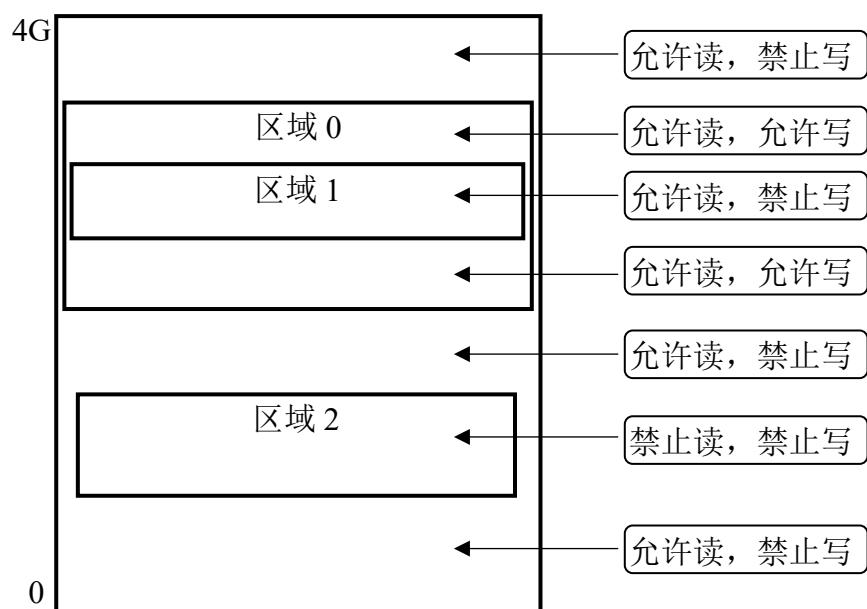
13.3.1 只允许部分空间访问

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。



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。



13.4 寄存器说明

本模块的寄存器只能通过 CPU 来设置。

MPU 基准地址: 0x4005_0000

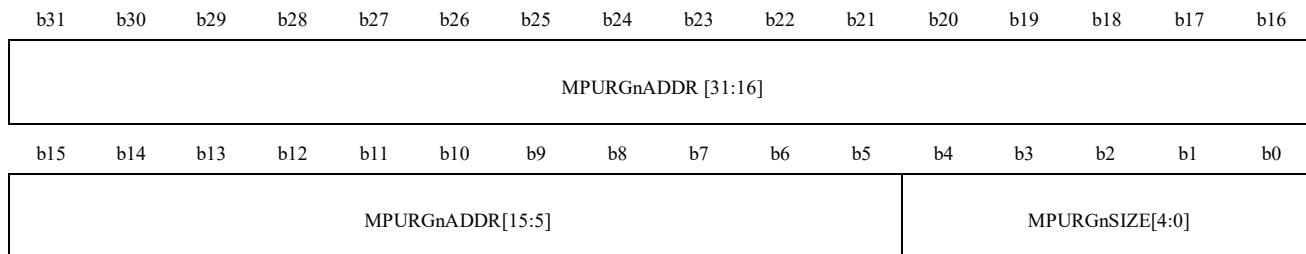
偏移地址	寄存器名	初始值	名称	写保护
+00~+3C	MPU_RGD0~15	0x0000_0000	区域0~15范围描述寄存器	MPUWE
+40~+7C	MPU_RGCR0~15	0x0000_0000	区域0~15控制寄存器	MPUWE
+80	MPU_CR	0x0000_0000	MPU控制寄存器	MPUWE
+84	MPU_SR	0x0000_0000	MPU状态寄存器	无
+88	MPU_ECLR	0x0000_0000	MPU错误标志清除寄存器	无
+8C	MPU_WP	0x0000_0000	MPU写保护寄存器	WKEY

基准地址: 0x40054000

偏移地址	寄存器名	初始值	名称	写保护
+1C	MPU_IPPR	0x0000_0000	IP访问保护寄存器	SYS

13.4.1 区域范围描述寄存器 MPU_RGDn (n=0~15)

复位值: 0x0000_0000



位	标记	位名	功能	读写
b31~b5	MPURGnADDR[31:5]	区域地址	设定区域n的地址, 有效位数与区域大小有关, 低 (MPURGnSIZE+1)位固定为0	R/W
b4~b0	MPURGnSIZE[4:0]	区域大小	设定区域n的大小 00000~00011: 保留, 设定禁止 00100: 32Byte 00101: 64Byte ... 11110: 2GByte 11111: 4GByte	R/W

13.4.2 区域控制寄存器 MPU_RGCRn (n=0~15)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	FRGnE	-	-	-	-	-	FRGnWP	FRGnRP
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S1RGnE	-	-	-	-	-	S1RGnWP	S1RGnRP	S2RGnE	-	-	-	-	-	S2RGnWP	S2RGnRP

位	标记	位名	功能	读写
b31~b24	reserved	-	保留位, 读出为0, 写入时写0	R
b23	FRGnE	FMPU区域n使能	0: FMPU的区域n无效 1: FMPU的区域n有效	R/W
b22~b18	reserved	-	保留位, 读出为0, 写入时写0	R
b17	FRGnWP	FMPU区域n写权	0: 区域n允许USBFS-DMA写 1: 区域n禁止USBFS-DMA写	R/W
b16	FRGnRP	FMPU区域n读权	0: 区域n允许USBFS-DMA读 1: 区域n禁止USBFS-DMA读	R/W
b15	S1RGnE	SMPU1区域n使能	0: SMPU1的区域n无效 1: SMPU1的区域n有效	R/W
b14~b10	reserved	-	保留位, 读出为0, 写入时写0	R
b9	S1RGnWP	SMPU1区域n写权限	0: 区域n允许系统DMA_1写 1: 区域n禁止系统DMA_1写	R/W
b8	S1RGnRP	SMPU1区域n读权限	0: 区域n允许系统DMA_1读 1: 区域n禁止系统DMA_1读	R/W
b7	S2RGnE	SMPU2区域n使能	0: SMPU2的区域n无效 1: SMPU2的区域n有效	R/W
b6~b2	reserved	-	保留位, 读出为0, 写入时写0	R/W
b1	S2RGnWP	SMPU2区域n写权限	0: 区域n允许系统DMA_2写 1: 区域n禁止系统DMA_2写	R/W
b0	S2RGnRP	SMPU2区域n读权限	0: 区域n允许系统DMA_2读 1: 区域n禁止系统DMA_2读	R/W

注意:

- 区域 8~15 的控制寄存器 b31~b16 为保留位。

13.4.3 控制寄存器 MPU_CR

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	FMPUE	-	-	-	FMPUACT[1:0]	FMPUB WP	FMPUB RP	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SMPU1E	-	-	-	SMPU1ACT[1:0]	SMPU1BWP	SMPU1BRP	SMPU2E	-	-	-	SMPU2ACT[1:0]	SMPU2BWP	SMPU2BRP		

位	标记	位名	功能	读写
b31~b24	reserved	-	保留位, 读出为0, 写入时写0	R
b23	FMPUE	FMPU使能	0: FMPU无效 1: FMPU有效	R/W
b22~b20	reserved	-	保留位, 读出为0, 写入时写0	R
b19~b18	FMPUACT[1:0]	FMPU动作选择	设定USBFS-DMA发生被禁止的访问时的动作 00: 无视 (读访问读到0, 写访问忽略) 01: 无视+总线错误 10: 无视+不可屏蔽中断 11: 复位	R/W
b17	FMPUBWP	FMPU背景写权 限设置	0: FMPU背景空间允许USBFS-DMA写 1: FMPU背景空间禁止USBFS-DMA写	R/W
b16	FMPUBRP	FMPU背景读权 限设置	0: FMPU背景空间允许USBFS-DMA读 1: FMPU背景空间禁止USBFS-DMA读	R/W
b15	SMPU1E	SMPU1使能	0: SMPU1无效 1: SMPU1有效	R/W
b14~b12	reserved	-	保留位, 读出为0, 写入时写0	R
b11~b10	SMPU1ACT[1:0]	SMPU1动作选择	设定系统DMAC_1发生被禁止的访问时的动作 00: 无视 (读访问读到0, 写访问忽略) 01: 无视+总线错误 10: 无视+不可屏蔽中断 11: 复位	R/W
b9	SMPU1BWP	SMPU1背景写权 限设置	0: SMPU1背景空间允许系统DMAC_1写 1: SMPU1背景空间禁止系统DMAC_1写	R/W
b8	SMPU1BRP	SMPU1背景读权 限设置	0: SMPU1背景空间允许系统DMAC_1读 1: SMPU1背景空间禁止系统DMAC_1读	R/W
b7	SMPU2E	SMPU2使能	0: SMPU2无效 1: SMPU2有效	R/W
b6~b4	reserved	-	保留位, 读出为0, 写入时写0	R
b3~b2	SMPUACT[1:0]	SMPU动作选择	设定系统DMA_2发生被禁止的访问时的动作	R/W

00: 无视（读访问读到0，写访问忽略）

01: 无视+总线错误

10: 无视+不可屏蔽中断

11: 复位

b1	SMPU2BWP	SMPU2背景写权 限设置	0: SMPU2背景空间允许系统DMA_2写 1: SMPU2背景空间禁止系统DMA_2写	R/W
b0	SMPU2BRP	SMPU2背景读权 限设置	0: SMPU2背景空间允许系统DMA_2读 1: SMPU2背景空间禁止系统DMA_2读	R/W

多个区域设定重叠时，优先级为：设定的禁止>设定的允许。

13.4.4 状态标志寄存器 MPU_SR

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FMPUEAF
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	SMPU1EAF	-	-	-	-	-	-	-	SMPU2EAF

位	标记	位名	功能	读写
b31~b17	reserved	-	保留位, 读出为0, 写入时写0	R
b16	FMPUEAF	FMPU错误标志	0: USBFS-DMA未发生错误访问 1: USBFS-DMA发生了错误访问	R
b15~b9	reserved	-	保留位, 读出为0, 写入时写0	R
b8	SMPU1EAF	SMPU1错误标志	0: 系统DMA_1未发生错误访问 1: 系统DMA_1发生了错误访问	R
b7~b1	reserved	-	保留位, 读出为0, 写入时写0	R
b0	SMPU2EAF	SMPU2错误标志	0: 系统DMA_2未发生错误访问 1: 系统DMA_2发生了错误访问	R

对本寄存器的写操作会被忽略, 清除错误标志请使用 MPUECLR。

13.4.5 标志清除寄存器 MPU_ECLR

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FMPUECLR
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	SMPU1ECLR	-	-	-	-	-	-	-	SMPU2ECLR

位	标记	位名	功能	读写
b31~b17	reserved	-	保留位, 读出为0, 写入时写0	R
b16	FMPUECLR	FMPU错误标志	写入1可以将FMPUEAF清0 清除	W
b15~b9	reserved	-	保留位, 读出为0, 写入时写0	R
b8	SMPU1ECLR	SMPU1错误标志	写入1可以将SMPU1EAF清0 清除	W
b7~b1	reserved	-	保留位, 读出为0, 写入时写0	R
b0	SMPU2ECLR	SMPU2错误标志	写入1可以将SMPU2EAF清0 清除	W

本寄存器读出值固定为 0x0000_0000。

13.4.6 写保护寄存器 MPU_WP

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b16	reserved	-	保留位, 读出为0, 写入时写0	R
b15~b1	WKEY[15:1]	写入码	在写入MPUWE时, 必须同时向WKEY写入 'b1001_0110_1010_010, 读出为0	W
b0	MPUWE	MPU寄存器写允许	0: MPU地址寄存器/控制寄存器不允许写入 1: MPU地址寄存器/控制寄存器允许写入	RW

向本寄存器写入 0x96A5 可以将 MPUWE 置 1, 写入 0x96A4 可以将 MPUWE 清 0, 写入其他值不能改变 MPUWE。

13.4.7 IP 访问保护寄存器 MPU_IPPR

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
BUSER RE	-	MSTP WRP	MSTPR DP	SYSCW RP	SYSCR DP	INTCW RP	INTCR DP	SRAM CWRP	SRAM CRDP	DMPU WRP	DMPU RDP	RTCW RP	RTCRD P	BKSRA MWRP	BKSRA MRDP
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SWDT WRP	SWDT RDP	WDTW RP	WDTR DP	-	-	EFMC WRP	EFMCR DP	CRCW RP	CRCRD P	TRNG WRP	TRNGR DP	HASH WRP	HASHR DP	AESWR P	AESRD P

位	标记	位名	功能	读写
b31	BUSERRE	总线错误允许	0: 忽略对保护IP的访问 1: 发生对保护IP的访问时, 返回总线错误	RW
b30	-	-	保留位, 读出为0, 写入时写0	R
b29	MSTPWWRP	MSTP写保护	0: 允许对寄存器PWC_FCG0/1/2/3、PWC_FCG0PC的写操作 1: 禁止对寄存器PWC_FCG0/1/2/3、PWC_FCG0PC的写操作	RW
b28	MSTPRDWP	MSTP读保护	0: 允许对寄存器PWC_FCG0/1/2/3、PWC_FCG0PC的读操作 1: 禁止对寄存器PWC_FCG0/1/2/3、PWC_FCG0PC的读操作	RW
b27	SYSCWRP	SYSC写保护	0: 允许对RMU/CMU/PWC的写操作 1: 禁止对RMU/CMU/PWC的写操作	RW
b26	SYSCRDP	SYSC读保护	0: 允许对RMU/CMU/PWC的读操作 1: 禁止对RMU/CMU/PWC的读操作	RW
b25	INTCWRP	INTC写保护	0: 允许对INTC的写操作 1: 禁止对INTC的写操作	RW
b24	INTCRDP	INTC读保护	0: 允许对INTC的读操作 1: 禁止对INTC的读操作	RW
b23	SRAMCWRP	SRAMC写保护	0: 允许对【10.内置SRAM】的寄存器写操作 1: 禁止对【10.内置SRAM】的寄存器写操作	RW
b22	SRAMCRDP	SRAMC读保护	0: 允许对【10.内置SRAM】的寄存器读操作 1: 禁止对【10.内置SRAM】的寄存器读操作	RW
b21	DMPUWRP	DMPU写保护	0: 允许对SMPU1/SMPU2/FMPU/IPMPU的写操作 1: 禁止对SMPU1/SMPU2/FMPU/IPMPU的写操作	RW
b20	DMPURDP	DMPU读保护	0: 允许对SMPU1/SMPU2/FMPU/IPMPU的读操作 1: 禁止对SMPU1/SMPU2/FMPU/IPMPU的读操作	RW
b19	RTCWRP	RTC写保护	0: 允许对RTC的写操作 1: 禁止对RTC的写操作	RW
b18	RTCCRDP	RTC读保护	0: 允许对RTC的读操作 1: 禁止对RTC的读操作	RW
b17	BKSRAMWRP	BKSRAM写保护	0: 允许对Ret-SRAM的写操作 1: 禁止对Ret-SRAM的写操作	RW

b16	BKSRAMRDP	BKSRAM读保护	0: 允许对Ret-SRAM的读操作 1: 禁止对Ret-SRAM的读操作	RW
b15	SWDTWRP	SWDT写保护	0: 允许对SWDT的写操作 1: 禁止对SWDT的写操作	RW
b14	SWDTRDP	SWDT读保护	0: 允许对SWDT的读操作 1: 禁止对SWDT的读操作	RW
b13	WDTWRP	WDT写保护	0: 允许对WDT的写操作 1: 禁止对WDT的写操作	RW
b12	WDTRDP	WDT读保护	0: 允许对WDT的读操作 1: 禁止对WDT的读操作	RW
b11~b10	-	-	保留位, 读出为0, 写入时写0	R
b9	EFMCWRP	EFM写保护	0: 允许对【9 嵌入式Flash(EMF)】的寄存器写操作 1: 禁止对【9 嵌入式Flash(EMF)】的寄存器写操作	RW
b8	EFMCRDP	EFM读保护	0: 允许对【9 嵌入式Flash(EMF)】的寄存器读操作 1: 禁止对【9 嵌入式Flash(EMF)】的寄存器读操作	RW
b7	CRCWRP	CRC写保护	0: 允许对CRC的写操作 1: 禁止对CRC的写操作	RW
b6	CRCRDP	CRC读保护	0: 允许对CRC的读操作 1: 禁止对CRC的读操作	RW
b5	TRNGWRP	TRNG写保护	0: 允许对TRNG的写操作 1: 禁止对TRNG的写操作	RW
b4	TRNGRDP	TRNG读保护	0: 允许对TRNG的读操作 1: 禁止对TRNG的读操作	RW
b3	HASHWRP	HASH写保护	0: 允许对HASH的写操作 1: 禁止对HASH的写操作	RW
b2	HASHRDP	HASH读保护	0: 允许对HASH的读操作 1: 禁止对HASH的读操作	RW
b1	AESWRP	AES写保护	0: 允许对AES的写操作 1: 禁止对AES的写操作	RW
b0	AESRDP	AES读保护	0: 允许对AES的读操作 1: 禁止对AES的读操作	RW

特权模式下不受本寄存器影响可以读写访问对象 IP。

14 DMA 控制器 (DMA)

14.1 简介

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

- DMA 总线独立于 CPU 总线，按照 AMBA AHB-Lite 总线协议传输
- 拥有 2 个 DMA 控制单元，共 8 个独立通道，可以独立操作不同的 DMA 传输功能
- 每个通道的启动源通过独立的触发源选择寄存器配置
- 每次请求传输一个数据块
- 数据块最小为 1 个数据，最多可以是 1024 个数据
- 每个数据的宽度可配置为 8bit, 16bit 或 32bit
- 可以配置最多 65535 次传输
- 源地址和目标地址可以独立配置为固定，自增，自减，循环或指定偏移量的跳转
- 可产生 3 种中断：块传输完成中断，传输完成中断，传输错误中断。每种中断都可以配置是否屏蔽。其中块传输完成，传输完成可作为事件输出，可作为其它外围模块的触发源
- 支持连锁传输功能，可实现一次请求传输多个数据块
- 支持外部事件触发通道重置
- 不使用时可设置进入模块停止状态以降低功耗

14.2 模块示意图

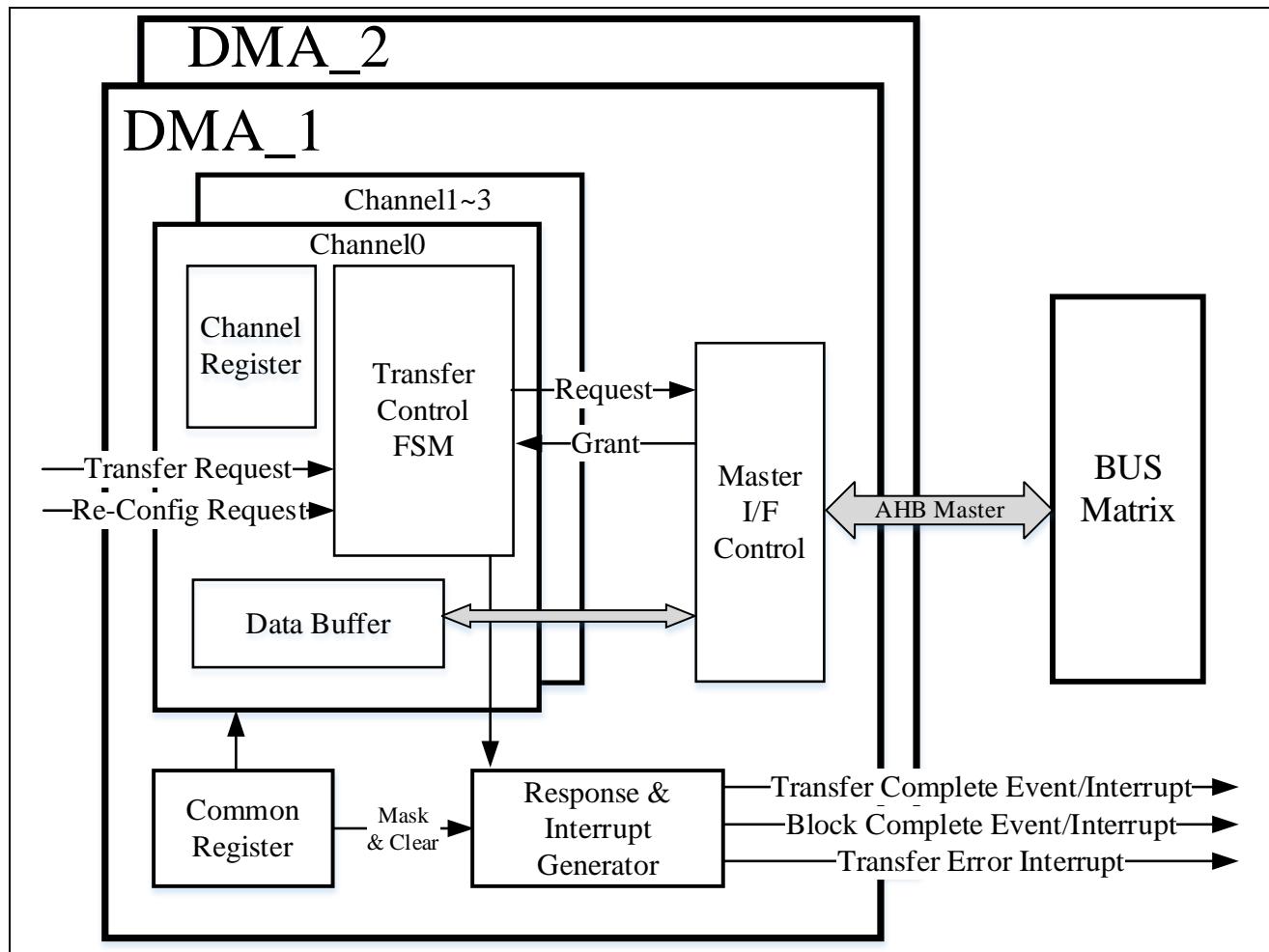


图 14-1 DMA 结构图

14.3 功能描述

14.3.1 使能 DMA 控制器

使用 DMA 时需要先写寄存器将 DMA 控制器使能，使能方法是写 DMA 使能寄存器 DMA_EN.EN 位。

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。EN 写 0 前请先确认寄存器 DMA_CHSTAT.DMAACT 为 0，确保 DMA 已完成所有传输。

14.3.2 通道选择和通道优先级

每个 DMA 控制单元中包含 4 个通道，每个通道可以独立配置传输功能。

4 个通道优先级顺序为：通道 0>通道 1>通道 2>通道 3。

当一个 DMA 单元有多个通道请求传输时将按照优先级顺序执行。但已处于传输中的通道不会被打断，高优先通道需等当前通道传输完成后才会启动。

14.3.3 启动 DMA

DMA 通过外围电路产生的请求启动，这些请求通过触发源选择寄存器 DMA_TRGSELx(x=0~3)进行配置，可以配置通道 x 的启动请求源。当外围电路产生启动请求或软件写寄存器产生启动请求时，并且 DMA 传输使能有效 DMA_EN.EN=1，传输通道处于许可状态 DMA_CHEN.CHEN[x]=1，则启动通道 x 传输。

使用前，需要先将功能时钟控制 0 寄存器(FCG0)的外设电路触发功能及 DMA 功能使能位置为有效。

14.3.4 数据块

DMA 每次启动传输的数据量用块(block)表示，块的大小由数据控制寄存器 DMA_DTCTLx.BLKSIZE 设定，最大可以设置 1024 个数据。每个的数据宽度由 DMA_CHxCTL.HIZE 决定。

14.3.5 传输地址控制

传输的源地址和目标地址可以由寄存器设定为固定、递增、递减、重载或者不连续跳转。

固定: 源地址、目标地址将在传输过程中固定不变。

递增及递减: 源地址、目标地址将在每传输完成 1 个数据后根据 HSIZE 的值向前或向后跳转。例如当 HSIZE 为 8bit 时，地址将每次增加/减少 1，为 16bit 时每次增加/减少 2，为 32bit 时每次增加/减少 4。

重载: 传输指定数量的数据后，源、目标地址将重新返回至最初的地址设定值。地址重载前需要传输的数据量，即重复区域的大小由寄存器 DMA_RPT 设定。

不连续地址传输: 传输指定数量的数据后，源、目标地址将跳过指定偏移量。地址跳转的偏移量，以及跳转前需要传输的数据量，即不连续区域的大小，由寄存器 DMA_SNSEQCTL/DMA_DNSEQCTL 设定。当地址重载与不连续跳转的条件同时满足时，执行地址重载。

14.3.6 传输次数

DMA 传输的总数据块的个数由数据控制寄存器 DMA_DTCTLx 的 CNT 位设定。传输次数最多可设置 65535 次。每传输一个数据块寄存器值减 1，当寄存器值减为 0 时代表本通道全部数据传输完成，自动清除通道传输许可位 DMA_CHEN.CHEN[x]，并产生传输完成中断。如果传输开始时寄存器被设为 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。

14.3.7 中断和事件信号输出

DMA 控制器可以产生以下 3 种中断：

数据块完成中断 DMA_BTxC: 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_CHxCTL.IE 设置中断的有效或无效。另外所有中断还配有独立的 MASK 寄存器，对中断进行屏蔽。上述的 DMA_BTCx, DMA_TCx 中断同时也可以作为事件信号输出，可用作其它外围电路的触发源，事件输出受 MASK 寄存器控制，但不受中断许可位 DMA_CHxCTL.IE 控制

14.3.8 连锁传输

DMA 控制器具有连锁传输功能。连锁传输需要配置以下 8 个寄存器共 8 个 word，称为一个描述符(descriptor)，包含了连锁传输的源地址、目标地址、数据控制信息、地址控制信息、链指针以及传输控制信息。

DMA_SARx

DMA_DARx

DMA_DTCTLx

DMA_RPTx

DMA_SNSEQCTLx

DMA_DNSEQCTLx

DMA_LLPx

DMA_CHxCTL

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_CHxCTLx 的 LLPEN 将连锁传输使能，并将第一个传输的描述符信息写入相应寄存器中。再将后续传输的描述符按顺序初始化在存储器中。需要结束连锁传输时，将通道最后一个描述符中 DMA_CHxCTLx 的 LLPEN 置为无效，DMA 控制器将在传输完成后结束连锁传输。

当一个描述符的最后一次传输结束时，

若 LLPEN=1, LLPRUN=0，根据中断许可的配置，产生 BTC 和 TC 中断，通道许可 CHEN[x]不自动清 0。LLP 指定的下一个描述符从存储器中载入通道配置寄存器，等待下一次传输请求输入，开始新描述符的第一次传输。

若 LLPEN=1, LLPRUN=1，不产生 BTC 和 TC 中断，通道许可 CHEN[x]不自动清 0。LLP 指定的下一个描述符从存储器中载入通道配置寄存器后直接开始新描述符的第一次传输。

若 LL PEN=0，则链传输结束，根据中断许可的配置，产生 BTC 和 TC 中断，通道许可 CHEN[x]自动清 0。

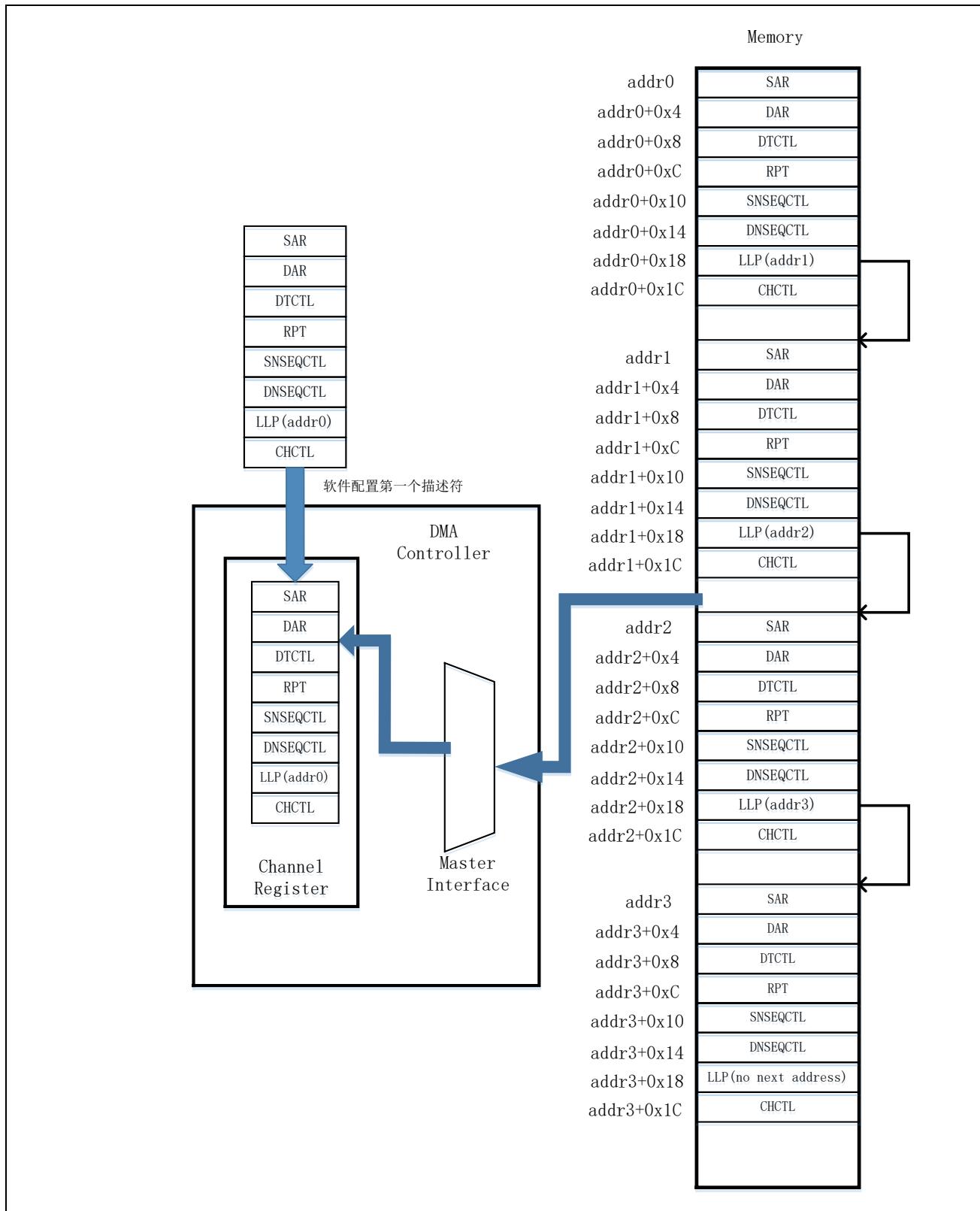


图 14-2 连锁传输示意图

14.3.9 不连续地址传输

使用不连续地址传输能够实现源地址和目标地址在传输完一定量的数据后按照一定的偏移量跳转。跳转的方向按照 DMA_CHxCTL.SINC 和 DMA_CHxCTL.DINC 的设定向前或向后跳转。使用时，首先根据需要将通道控制寄存器 DMA_CHxCTL.SNSEQEN 和 DMA_CHxCTL.DNSEQEN 置 1 使不连续地址传输有效。然后配置源设备和目标设备不连续地址传输控制寄存器 DMA_SNSEQCTLx 和 DMA_DNSEQCTLx。传输过程按照下图方式进行。

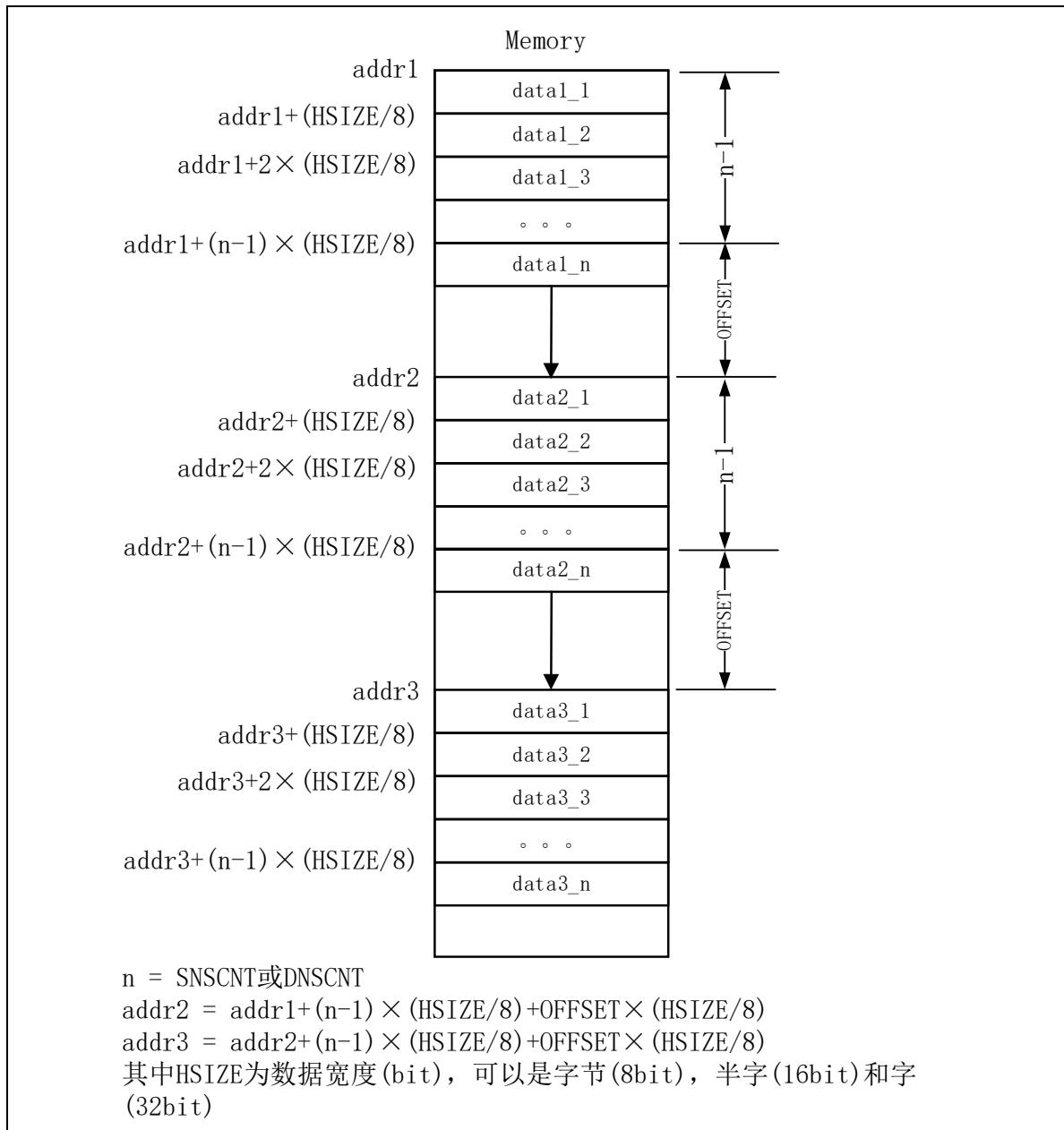


图 14-3 不连续地址传输示意图

14.3.10 通道重置

通道重置功能是指通过外围电路的事件请求来修改通道内部状态寄存器，重新配置下一次数据的传输方式。设置寄存器 DMA_RCFGCTL.RCFGGEN 为 1，允许通道重置。通过触发源选择寄存器 DMA_TRGSELRC 选择重置请求源。当选择的重置请求源输入时，寄存器 DMA_RCFGCTL.RCFGCHS 选择的通道会按指定的方式进行更新。重置请求只更新内部状态，不会启动实际的数据传输。

通道的重置方式有如下三种：链指针式、不连续式、重复式。

选择链指针式重置时，通道的描述符和内部状态全部更新为链指针 LLP 指向的新描述符。之后的传输请求，按新描述符进行传输。

选择不连续式、重载式重置时，通道内部状态按下表所述进行更新。

通道内部状态	重置方式	
	不连续式	重载式
剩余传输次数计数器	更新为正常状态下，下一次地址不连续跳转发生后的数值	更新为正常状态下，下次重载发生后的数值
下次传输的源/目标地址	更新为下一个不连续传输区域的首地址	更新为寄存器 DMA_SARx/DARx 的初始设定值

表 14-1 通道重置说明

注意：

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

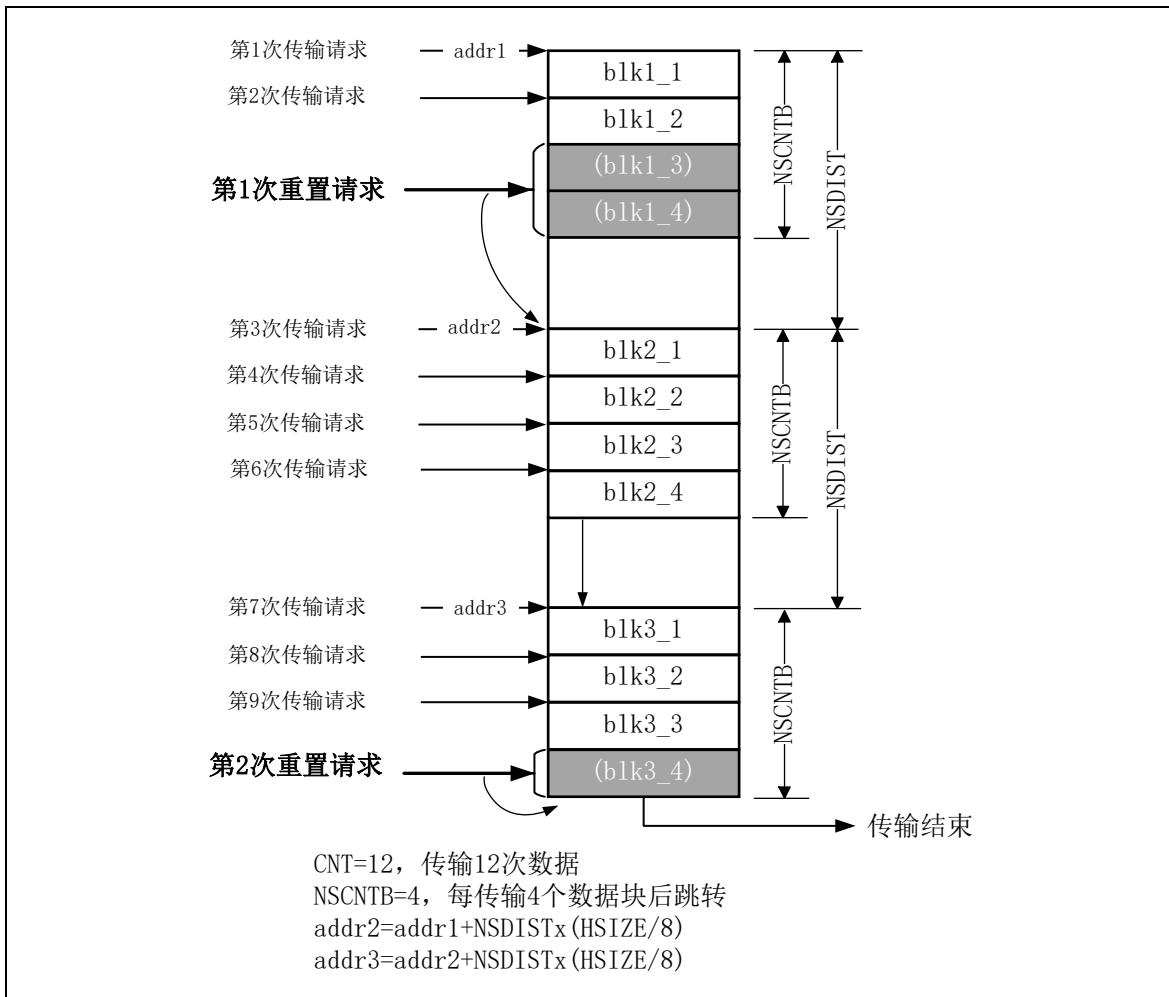


图 14-4 不连续式重置示意图

图 14-4 所示 DMA 动作中，每次传输请求启动一个数据块的传输。第 1 次重置请求发生后，控制器跳过了数据块 blk1_3,blk1_4，传输地址更新为下个不连续区域的首地址即 addr2。第 2 次重置请求发生后，剩余传输次数更新为 0，即所有数据传输完成，通道许可位自动清 0，产生传输完成中断和事件。

14.3.11 传输提前终止

传输过程中通道使能寄存器 DMA_CHEN.CHENx 保持有效，非连锁传输时，数据控制寄存器 DMA_DTCTLx 设定的传输次数完成后自动置为无效，连锁传输时，最后一次连锁传输的传输次数完成后自动置为无效。传输过程中如果软件写 DMA_CHEN.CHENx 为 0，则 DMA 将在完成当次数据读写后终止传输。

注意：

- 软件将 CHENx 位写 0 提前终止传输时，DMA 内部不会保存被终止时的传输状态。在没有重新设置通道配置寄存器（描述符）状态，将 CHENx 写 1 再次允许此通道，传输请求输入后，DMA 将重新传输被终止的数据块，而不是断点续传。

14.4 应用举例

14.4.1 存储器到存储器的传输

目标: 将 22 个数据从 RAM 地址 0x2000_0000 传送到 0x2000_1000, 数据宽度为 32bit。

1. 寄存器设定

- DMA_EN.EN 写 1 使能 DMA 控制器
- 选择一个通道, 例如通道 0, 配置通道寄存器以实现:
 - 写 DMA_SAR0 配置源地址为 SRAM 区域 0x2000_0000
 - 写 DMA_DAR0 配置源地址为 SRAM 区域 0x2000_1000
 - 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
 - 写 DMA_RPT 寄存器配置源地址重复区域大小为 6, 即传输完成 6 个地址后重新载入初始源地址
 - 配置通道控制寄存器 DMA_CH0CTL 以实现:
 - * 源地址和目标地址连锁传输无效
 - * 源地址重载有效, 和目标地址更新方式为自增
 - * 数据宽度为 word(32bit)
 - * 中断使能有效
 - 通道使能位 DMA_CHEN.CHEN0 写 1, 使能通道 0
 - 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
- 写外设事件软件触发寄存器 INTSFTTRG STRG 为 1, 发送第一个软件启动请求, DMA 开始传输数据

2. 传输过程

由于传输的数据块大小为 4，当软件写 INTSFTTRG STRG 为 1 后开始第一次传输，当一个数据块传输完成后，传输次数 DMA_DTCTL0.CNT 减 1，并产生一个块传输完成中断，软件可以在中断子程序中继续写 INTSFTTRG STRG 开始第二次传输。在第二次传输中，由于设置了源地址重复区域大小为 6，因此在传输完 2 个地址后源地址将重新载入初始地址 0x20000000 并继续传输剩余的 2 个地址。第二次传输完成后，传输次数 DMA_DTCTL0.CNT 减 1，并产生一个块传输完成中断，软件可以在中断子程序中继续写 INTSFTTRG STRG 开始第三次传输。第三次传输完成后，传输次数 DMA_DTCTL0.CNT 减为 0，即本次传输全部完成，DMA 产生一个块传输完成中断和一个传输完成中断，同时通道使能位 DMA_CHEN.CHEN0 将被自动清零。

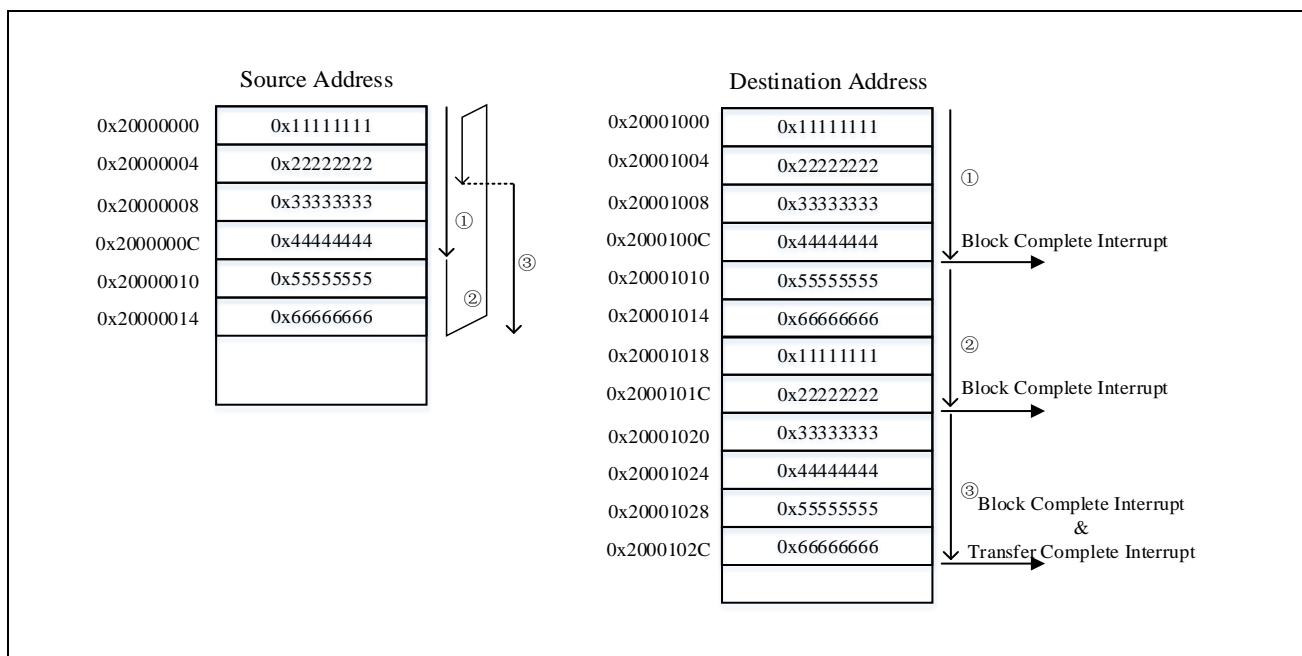


图 14-5 应用例 1：存储器到存储器传输

14.4.2 存储器到外围电路的传输

目标：将 10 个宽度为 half-word 的数据从 RAM 地址 0x2000_0000 传送到通信模块的发送缓冲寄存器中，通信模块每发送完一个数据产生一次传送请求。当最后一个数据发送完成后 DMA 产生一个传输完成中断。

1. 寄存器设定

- DMA_EN.EN 写 1 使能 DMA 控制器
- 配置 DMA_INTMSK 寄存器，屏蔽块传输完成中断，使能传输完成中断
- 选择一个通道，配置通道寄存器，例如选择通道 0
 - 写 DMA_SAR0 配置源地址为 SRAM 区域 0x2000_0000
 - 写 DMA_DAR0 配置源地址为外围电路的寄存器地址 0x4000_0000
 - 写 DMA_DTCTL0 配置数据块的大小为 1，传输次数为 10 次，每个传输请求传送一次，每次 1 个数据。
 - 配置通道控制寄存器 DMA_CH0CTL 以实现：
 - * 源地址和目标地址连锁传输无效
 - * 源地址更新方式为自增，目标地址为固定
 - * 源地址和目标地址数据宽度为半字(16bit)
 - * 中断使能有效
 - 配置触发源控制器 DMA_TRGSEL0，选择通信模块的发送寄存器为空事件作为 DMA 通道 0 的启动请求
 - 通道使能位 DMA_CHEN.CHEN0 写 1，使能通道 0

2. 传输过程

通道使能后，DMA 等待来自通信模块的传输请求。当传输请求产生后，DMA 把 RAM 中的数据传送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。

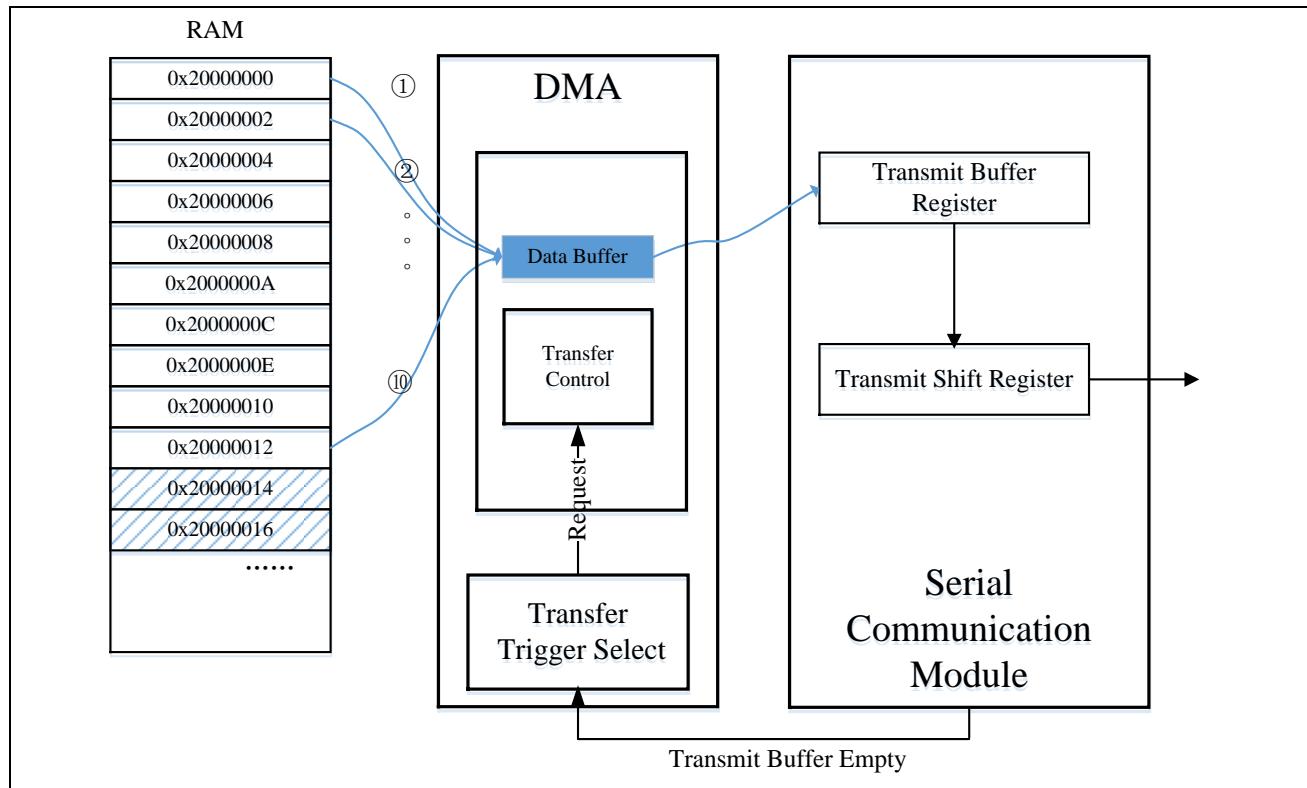


图 14-6 应用例 2：存储器到外围电路的传输

14.4.3 存储器到存储器的连锁传输

1. 寄存器设定

- DMA_EN.EN 写 1 使能 DMA 控制器
- 选择一个通道，配置通道寄存器，例如选择通道 0，配置第一次传输的描述符(descriptor0)
 - 写 DMA_SAR0 配置源地址为 SRAM 区域 0x2000_0000
 - 写 DMA_DAR0 配置目标地址为 SRAM 区域 0x2000_1000
 - 写 DMA_DTCTL0 配置数据块的大小为 10
 - 在链指针寄存器 DMA_LLPO 中写入第二个描述符(descriptor1)的地址 0x20002000
 - 配置通道控制寄存器 DMA_CH0CTL，配置第一个数据块的传输参数以实现：
 - * 连锁传输有效
 - * 连锁传输模式为直接启动下一次传输
 - * 源地址和目标地址更新方式为自增
 - * 数据宽度为字(32bit)
 - * 中断使能无效
- 在 RAM 空间的 0x2000_2000 地址中配置第二次传输的描述符(descriptor1)，包括：
 - 0x2000_2000 中写入 32 位数据 0x2000_0100，此为第二次传输的源地址
 - 0x2000_2004 中写入 32 位数据 0x2000_1100，此为第二次传输的目标地址
 - 0x2000_2008 中配置数据块的大小为 20
 - 0x2000_2018 中写入 32 位数据 0x2000_2100，此为第三次传输的描述符(descriptor2)所在地址
 - 0x2000_021C 中写入第二次传输的控制数据，该控制数据实现：
 - * 连锁传输有效
 - * 连锁传输模式为直接启动下一次传输
 - * 源地址和目标地址更新方式为自增
 - * 数据宽度为半字(16bit)
 - * 中断使能无效

- 在 RAM 空间的 0x2000_2100 地址中配置第三次传输的描述符(descriptor2), 包括:
 - 0x2000_2100 中写入 32 位数据 0x2000_0200, 此为第三次传输的源地址
 - 0x2000_2104 中写入 32 位数据 0x2000_1200, 此为第三次传输的目标地址
 - 0x2000_2108 配置数据块的大小为 40
 - 0x2000_2118 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
 - 0x2000_211C 中写入第三次传输的控制数据实现:
 - * 连锁传输无效
 - * 源地址和目标地址更新方式为自增
 - * 数据宽度为字节(8bit)
 - * 中断使能有效
- 通道使能位 DMA_CHEN.CHEN0 写 1, 使能通道 0
- 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
- 写软件触发寄存器 INTSFTTRG.STRG 为 1, 发送一个启动请求, DMA 开始传输数据

2. 传输过程

软件启动 DMA 开始传输。第一次传输完成后, 由于设置了连锁传输模式为直接启动下一次传输并且中断无效, DMA 将第二次传输的描述符(descriptor1)读入到通道寄存器中。根据描述符配置的参数直接开始第二次传输。第二次传输完成后, 将第三次传输的描述符(descriptor2)读入到通道寄存器中。根据描述符配置的参数开始第三次传输。第三次传输完成后, 根据配置信息该次为连锁传输的最后一次, 且由于中断使能有效, 因此 DMA 将产生一个传输完成中断, 并清除通道使能位 DMA_CHEN.CHEN0。

14.5 寄存器说明

DMA_1 BASE_ADDR:0x4005_3000

DMA_2 BASE_ADDR:0x4005_3400

寄存器名	符号	偏移地址	位宽	复位值
DMA使能寄存器	DMA_EN	0x00	32	0x0000_0000
中断状态寄存器0	DMA_INTSTAT0	0x04	32	0x0000_0000
中断状态寄存器1	DMA_INTSTAT1	0x08	32	0x0000_0000
中断屏蔽寄存器0	DMA_INTMASK0	0x0C	32	0x0000_0000
中断屏蔽寄存器1	DMA_INTMASK1	0x10	32	0x0000_0000
中断复位寄存器0	DMA_INTCLR0	0x14	32	0x0000_0000
中断复位寄存器1	DMA_INTCLR1	0x18	32	0x0000_0000
通道使能寄存器	DMA_CHEN	0x1C	32	0x0000_0000
传输请求状态寄存器	DMA_REQSTAT	0x20	32	0x0000_0000
传输中通道监视寄存器	DMA_CHSTAT	0x24	32	0x0000_0000
通道重置控制寄存器	DMA_RCFGCTL	0x2c	32	0x0000_0000
传输源地址寄存器	DMA_SARx *1	0x40+0x40*x	32	0x0000_0000
传输目标地址寄存器	DMA_DARx	0x44+0x40*x	32	0x0000_0000
数据控制寄存器	DMA_DTCTLx	0x48+0x40*x	32	0x0000_0001
重复区域大小寄存器	DMA_RPTx	0x4C+0x40*x	32	0x0000_0000
重复区域大小寄存器B	DMA_RPTBx			
源设备不连续地址传输控制寄存器	DMA_SNSEQCTLx	0x50+0x40*x	32	0x0000_0000
源设备不连续地址传输控制寄存器B	DMA_SNSEQCTLBx			
目标设备不连续地址传输控制寄存器	DMA_DNSEQCTLx	0x54+0x40*x	32	0x0000_0000
目标设备不连续地址传输控制寄存器B	DMA_DNSEQCTBx			
链指针寄存器	DMA_LLPx	0x58+0x40*x	32	0x0000_0000

通道控制寄存器	DMA_CHxCTL	0x5C+0x40*x	32	0x0000_1000
传输源地址监视寄存器	DMA_MONSARx	0x60+0x40*x	32	0x0000_0000
传输目标地址监视寄存器	DMA_MONDARx	0x64+0x40*x	32	0x0000_0000
数据控制监视寄存器	DMA_MONDTCTLx	0x68+0x40*x	32	0x0000_0001
重复区域计数器监视寄存器	DMA_MONRPTx	0x6C+0x40*x	32	0x0000_0000
源设备不连续传输计数器监视寄存器	DMA_MONSNSEQCTLx	0x70+0x40*x	32	0x0000_0000
目标设备不连续传输计数器监视寄存器	DMA_MONDNSEQCTLx	0x74+0x40*x	32	0x0000_0000

14.5.1 DMA 使能寄存器(DMA_EN)

复位值: 0x0000_0000

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

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

位	标记	位名	功能	读写
b31-b1	Reserved	-	读出时为“0”，写入时写“0”	R/W
b0	EN	DMA使能位	0: DMA无效 1: DMA使能	R/W

14.5.2 中断状态寄存器 0(DMA_INTSTAT0)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	REQERR[3:0]

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

位	标记	位名	功能	读写
b31-b20	Reserved	-	读出时为“0”，写入时写“0”	R/W
b19-b16	REQERR[3:0]	传输请求溢出错 误中断位	0: 该通道没有发生传输请求溢出错误 1: 该通道发生了传输请求溢出错误，即上次请求还处在等待状态 时又来了传输请求	R
b15-b4	Reserved	-	读出时为“0”，写入时写“0”	R/W
b3-b0	TRNERR[3:0]	传输错误中断位	0: 该通道没有发生传输错误 1: 该通道发生了传输错误	R

14.5.3 中断状态寄存器 0(DMA_INTSTAT1)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31-b20	Reserved	-	读出时为“0”,写入时写“0”	R/W
b19-b16	BTC[3:0]	块传输完成中断	该中断在完成一个数据块的传输后发生	R
	位	0:	该通道没有块传输中断发生	
		1:	该通道发生块传输中断	
b15-b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3-b0	TC[3:0]	传输完成中断位	该中断在完成传输次数寄存器DMA_CNTx设置的传输次数后发生 0: 该通道没有传输完成中断发生 1: 该通道发生传输完成中断	R

14.5.4 中断屏蔽寄存器(DMA_INTMASK0)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31-b20	Reserved	-	读出时为“0”,写入时写“0”	R/W
b19-b16	MSKREQERR[3:0]	传输请求溢出中断屏蔽	0: 不屏蔽传输请求溢出中断 1: 屏蔽传输请求溢出中断	R/W
b15-b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3-b0	MSKTRNERR[3:0]	传输错误中断屏蔽	0: 不屏蔽传输错误中断 1: 屏蔽传输错误中断	R/W

14.5.5 中断屏蔽寄存器(DMA_INTMASK1)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	MSKBTC[3:0]

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

位	标记	位名	功能	读写
b31-b20	Reserved	-	读出时为“0”,写入时写“0”	R/W
b19-b16	MSKBTC[3:0]	块传输完成中断 屏蔽	0: 不屏蔽块传输完成中断 1: 屏蔽块传输完成中断	R/W
b15-b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3-b0	MSKTC[3:0]	传输完成中断屏 蔽	0: 不屏蔽传输完成中断 1: 屏蔽传输完成中断	R/W

14.5.6 中断复位寄存器(DMA_INTCLR0)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CLRREQERR[3:0]

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

位	标记	位名	功能	读写
b31-b20	Reserved	-	读出时为“0”,写入时写“0”	R/W
b19-b16	CLRREQERR[3:0]	传输请求溢出错 误中断复位	写0无任何效果, 写1复位传输请求溢出错误中断状态位 读出永远为0	W
b15-b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3-b0	CLRTRNERR[3:0]	传输错误中断复 位	写0无任何效果, 写1复位传输错误中断状态位 读出永远为0	W

14.5.7 中断复位寄存器(DMA_INTCLR1)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CLRBTC[3:0]

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

位	标记	位名	功能	读写
b31-b20	Reserved	-	读出时为“0”，写入时写“0”	R/W
b19-b16	CLRBTC[3:0]	块传输完成中断 复位	写0无任何效果，写1复位块传输完成中断状态位 读出永远为0	W
b15-b4	Reserved	-	读出时为“0”，写入时写“0”	R/W
b3-b0	CLRTC[3:0]	传输完成中断复位	写0无任何效果，写1复位传输完成中断状态位 读出永远为0	W

14.5.8 通道使能寄存器(DMA_CHEN)

复位值: 0x0000_0000

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

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

位	标记	位名	功能	读写
b31-b4	Reserved	-	读出时为“0”，写入时写“0”	R/W
b3-b0	CHEN[3:0]	通道使能位	每个位对应一个通道。使用该通道需将该位置1。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零。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。 0：该通道无效 1：该通道有效	R/W

14.5.9 通道重置控制寄存器(DMA_RCFGCTL)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31-b22	Reserved	-	读出时为“0”，写入时写“0”	R/W
b21-b20	CNTMD[1:0]	剩余传输次数计数器重置方式	00: 保持不变 01: 按源地址方式 10, 11: 按目标地址方式 当选择源地址方式时，且源地址选择不连续式重置，则剩余传输次数计数器更新为DMA_SNSEQCTLBx.SNSCNTB指定传输次数后的状态；源地址选择重复式时，剩余传输次数计数器更新为DMA_RPTBx.SRPTB指定传输次数后的状态。源地址选择保持时，剩余传输次数计数器也保持不变。 当选择目标地址方式时，与源地址方式类似。	R/W
b19-b18	DARMD[1:0]	目标地址重置方式	00: 保持不变 01: 不连续式重置 下次传输的目标地址更新为 addr_base + (DNSDIST x HSIZE(bit)/8) 其中：addr_base表示当前不连续传输区域的首地址 10, 11: 重复式重置 下次传输的目标地址更新为DMA_DARx寄存器初始设置值。 注意：只允许在本通道目标地址不连续传输有效 (DMA_CHxCTL.DNSEQEN=1)状态下设置DARMD[1:0]为01。 只允许在本通道目标地址重载有效(DMA_CHxCTL.DRPTEN=1)状态下设置 DARMD[1]为1。	R/W
b17-b16	SARMD[1:0]	源地址重置方式	00: 保持不变 01: 不连续式重置 下次传输的源地址更新为 addr_base + (SNSDIST x HSIZE(bit)/8) 其中：addr_base表示当前不连续传输区域的首地址 10, 11: 重复式重置 下次传输的源地址更新为DMA_SARx寄存器初始设置值。 注意：只允许在本通道源地址不连续传输有效 (DMA_CHxCTL.SNSEQEN=1)状态下设置SARMD[1:0]为01。 只允许在本通道源地址重载有效(DMA_CHxCTL.SRPTEN=1)状态下设置	R/W

SARMD[1]为1。				
b15-b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11-b8	RCFGCHS[3:0]	重置通道选择	0x0: 通道0 0x1: 通道1 0x2: 通道2 0x3: 通道3 其它: Reserved, 设定禁止	R/W
b7-b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	RCFGPLL	链指针式通道重置	0: 链指针式重置无效 1: 链指针式重置有效 注意: RCFGPLL设为1时, 通道将重新载入存储器中的新描述符, 故本寄存器的bit16-bit21全部无效。	R/W
b0	RCFGEN	通道重置许可	0: 禁止事件触发通道配置寄存器强制更新 1: 允许事件触发通道配置寄存器强制更新	R/W

注意:

- 请在 DMA_EN.EN 为 0 时设置本寄存器, 本寄存器必须在重置通道的第一次传输前设置。

14.5.10 传输请求状态寄存器(DMA_REQSTAT)

复位值: 0x0000_0000

b31	b3	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RCFGREQ

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

位	标记	位名	功能	读写
b31-b17	Reserved	-	读出时为“0”,写入时写“0”	R/W
b16	RCFGREQ	通道重置请求标 志	当外部重置请求输入后置1, 当通道重置启动后, 或者通道重置禁止时清0。 0: 无通道重置请求 1: 有通道重置请求	R
b15-b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3~b0	CHREQ[3:0]	通道传输请求标 志位	每个位对应一个通道。 当外部传输请求输入后对应位置1, 当该通道传输启动后, 或发生传输出错, 或传输传输许可位 (DMAEN或CHEN[x])写0时, 该位清0。 当该位为1状态, 该通道传输请求再次输入, 则发生传输请求溢出 错误, 第二次的请求被忽略。 0: 该通道无传输请求 1: 该通道有传输请求	R

14.5.11 通道状态观测寄存器(DMA_CHSTAT)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31-b20	Reserved	-	读出时为“0”,写入时写“0”	R/W
b19-b16	CHACT[3:0]	传输动作中通道 监视位	每个位对应一个通道。 0: 该通道空闲中 1: 该通道处于动作中	R
b15-b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	RCFGACT	DMA通道重置动作中监视位	0: DMA未处于通道重置动作中 1: DMA处于通道重置动作中	R
b0	DMAACT	DMA动作中监视位	0: DMA未处于传输动作中 1: DMA处于传输动作中	R

14.5.12 传输源地址寄存器(DMA_SARx) (x=0~3)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31-b0	SAR[31:0]	传输源地址	设置传输源地址 注意: – 传输数据宽度为16bit时, 即DMA_CHxctl.HSIZE=01, SAR[0] 无效。传输数据宽度为32bit时, 即 DMA_CHxCTL.HSIZE=1x, SAR[1:0]无效。	R/W

14.5.13 传输目标地址寄存器(DMA_DARx) (x=0~3)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31-b0	DAR[31:0]	传输目标地址	设置传输目标地址	R/W
注意:				
<ul style="list-style-type: none"> - 传输数据宽度为16bit时, 即DMA_CHxCTL.HSIZE=01, DAR[0] 无效。传输数据宽度为32bit时, 即 DMA_CHxCTL.HSIZE=1x, DAR[1:0]无效。 				

14.5.14 数据控制寄存器(DMA_DTCTLx) (x=0~3)

复位值: 0x0000_0001

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CNT[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BLKSIZE[9:0]															

位	标记	位名	功能	读写
b31-b16	CNT[15:0]	传输次数	总的传输次数, 每次请求启动一个数据块的传输, 完成时传输次数 计数器减1, 当减到0时发生传输完成中断。如果设置为0, 表示无限次传输, 每次启动请求传输一个数据块, 完成时传输次数计数器 保持0不变, 不会产生传输完成中断。	R/W
注意:				
<ul style="list-style-type: none"> - 读出时为“0”, 写入时写“0” 				
b15-b10	Reserved	-		R/W
b9-b0	BLKSIZE[9:0]	数据块的大小	设置数据块的大小, 最大可以配置1024个数据。每个数据的宽度由 DMA_CHxCTL寄存器的HSIZE位决定。寄存器值设为1则每次传输 1个数据, 设为0则每次传输1024个数据。	R/W

14.5.15 重复区域大小寄存器(DMA_RPTx) (x=0~3)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DRPT[9:0]

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SRPT[9:0]

位	标记	位名	功能	读写
b31-b26	Reserved	-	读出时为“0”,写入时写“0”	R/W
b25-b16	DRPT[9:0]	目标地址重复区 域大小	设置目标地址重复区域大小 目标设备在每传输DRPT个数据后目标地址重载为DMA_DARx寄存器初始设定值。寄存器设为10则每传输10个数据后地址重载，设为0则每传1024个数据后地址重载。每个数据的宽度由DMA_CHxCTL寄存器的HSIZE位决定。	R/W
b15-b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9-b0	SRPT[9:0]	源地址重复区域 大小	设置源地址重复区域大小 源设备在每传输SRPT个数据后源地址重载为DMA_SARx寄存器初始设定值。寄存器设为10则每传输10个数据后地址重载，设为0则每传1024个数据后地址重载。每个数据的宽度由DMA_CHxCTL寄存器的HSIZE位决定。	R/W

该寄存器配置源地址和目标地址重复区域的大小。使用重复地址传输需要配置 DMA_CHxCTL 寄存器的 SRPTEN/DRPREN 位有效，并且配置 DMA_CHxCTL 寄存器的 SINC/DINC 位使地址更新方式为自增或自减，如果为固定则地址重载功能无效。 DMA_RPTx, DMA_RPTBx 这两个寄存器共用同一个地址，都是用来定义重复区域大小的。要使用哪一个取决于该通道是否启用了重置功能。未启用重置时使用 DMA_RPTx，启用重置时使用 DMA_PRTBx。

14.5.16 重复区域大小寄存器 B(DMA_RPTBx) (x=0~3)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31-b26	Reserved	-	读出时为“0”，写入时写“0”	R/W
b25-b16	DRPTB[9:0]	目标地址重复区 域大小	设置目标地址重复区域大小 目标设备在每传输DRPTB个数据块后目标地址重载为DMA_DARx 寄存器初始设定值。数据块大小由DMA_DTCTLx.BLKSIZE与 DMA_CHxCTL.HSIZE决定。	R/W
b15-b10	Reserved	-	读出时为“0”，写入时写“0”	R/W
b9-b0	SRPTB[9:0]	源地址重复区域 大小	设置源地址重复区域大小 源设备在每传输SRPTB个数据块后源地址重载为DMA_SARx寄存 器初始设定值。数据块大小由DMA_DTCTLx.BLKSIZE与 DMA_CHxCTL.HSIZE决定。	R/W

该寄存器配置源地址和目标地址重复区域的大小。使用重复地址传输需要配置 DMA_CHxCTL 寄存器的 SRPTEN/DRPREN 位有效，并且配置 DMA_CHxCTL 寄存器的 SINC/DINC 位使地址更新方式为自增或自减，如果为固定则重复地址传输功能无效。

DMA_RPTx, DMA_RPTBx 这两个寄存器共用同一个地址，都是用来定义重复区域大小的。要使用哪一个取决于该通道是否启用了重置功能。未启用重置时使用 DMA_RPTx，启用重置时使用 DMA_PRTBx。

14.5.17 源设备不连续地址传输控制寄存器(DMA_SNSEQCTLx) (x=0~3)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SNSCNT[11:0]										SOFFSET[19:16]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SOFFSET[15:0]															

位	标记	位名	功能	读写
b31-b20	SNSCNT[11:0]	源地址跳转的数据量	设置源地址跳转前传输的数据量大小。 源设备在每传输SNSCNT个数据后源地址按SOFFSET指定偏移量跳转。寄存器设为10则每传输10个数据后地址跳转，设为0则每传4096个数据后地址跳转。	R/W
b19-b0	SOFFSET[19:0]	源地址跳转的地址偏移量	当不连续地址传输时，设置源地址跳转的偏移量。偏移量是相对于当前传输地址，即跳转前最后一次传输地址而言的。跳转的方向根据通道控制寄存器DMA_CHxCTL.SINC的值向前或向后跳转。参考图 14-3。当DMA_CHxCTL.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_CHxCTL.HSIZE)设置的bit数和SOFFSET的值进行计算。 地址偏移量=SOFFSET×(HSIZE(bit)/8) 例如，当SOFFSET设为10，HSIZE为字(32bit)时，地址偏移量为10×4=40，若HSIZE为半字(16bit)时，偏移量为10×2=20，若HSIZE为字节(8bit)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	R/W

使用源设备不连续传输需要配置 DMA_CHxCTL 寄存器的 SNSEQEN 位有效，并且配置 DMA_CHxCTL 寄存器的 SINC 位使地址更新方式为自增或自减。

DMA_SNSEQCTLx, DMA_SNSEQCTLBx 这两个寄存器共用同一个地址，都是用来定义不连续传输的。要使用哪一个取决于该通道是否启用了重置功能。未启用重置时使用 DMA_SNSEQCTLx，启用重置时使用 DMA_SNSEQCTLBx.

14.5.18 源设备不连续地址传输控制寄存器 B(DMA_SNSEQCTLBx) (x=0~3)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SNSCNTB[11:0]										SNSDIST[19:16]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SNSDIST[15:0]															

位	标记	位名	功能	读写
b31-b20	SNSCNTB[11:0]	源地址跳转的数据量	设置源地址跳转前传输的数据量大小。 源设备在每传输SNSCNTB个数据块后源地址按SNSDIST指定地址 间距跳转。数据块大小由DMA_DTCTLx.BLKSIZE与 DMA_CHxCTL.HSIZE决定。	R/W
b19-b0	SNSDIST[19:0]	源不连续区域地址间距	当不连续地址传输时，设置源设备两个不连续区域的间距。 跳转的方向根据通道控制寄存器DMA_CHxCTL.SINC的值向前或向 后跳转。参考图 14-4。当DMA_CHxCTL.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_CHxCTL.HSIZE)设置的bit数和 SNSDIST的值进行计算。 地址间距=SNSDIST × (HSIZE(bit)/8) 例如，当SNSDIST设为10，HSIZE为字(32bit)时，地址间距为10× 4=40，若HSIZE为半字(16bit)时，间距为10×2=20，若HSIZE为字 节(8bit)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	R/W

使用源设备不连续传输需要配置 DMA_CHxCTL 寄存器的 SNSEQEN 位有效，并且配置 DMA_CHxCTL 寄存器的 SINC 位使地址更新方式为自增或自减。

DMA_SNSEQCTLx, DMA_SNSEQCTLBx 这两个寄存器共用同一个地址，都是用来定义不连续传输的。要使用哪一个取决于该通道是否启用了重置功能。未启用重置时使用 DMA_SNSEQCTLx，启用重置时使用 DMA_SNSEQCTLBx。

14.5.19 目标设备不连续地址传输控制寄存器(DMA_DNSEQCTLx) (x=0~3)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DNSCNT[11:0]										DOFFSET[19:16]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DOFFSET[15:0]															

位	标记	位名	功能	读写
b31-b20	DNSCNT[11:0]	目标地址跳转的数据量	设置目标地址跳转前传输的数据量大小。 目标设备在每传输DNSCNT个数据后目标地址按DOFFSET指定偏移量跳转。寄存器设为10则每传输10个数据后地址跳转，设为0则每传4096个数据后地址跳转。	R/W
b19-b0	DSOFFSET[19:0]	目标地址跳转的地址偏移量	当不连续地址传输时，设置目标地址跳转的偏移量。偏移量是相对于当前传输地址，即跳转前最后一次传输地址而言的。跳转的方向根据通道控制寄存器DMA_CHxCTL.DINC的值向前或向后跳转。 参考图 14-3。当DMA_CHxCTL.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_CHxCTL.HSIZE)设置的bit数和DOFFSET的值进行计算。 地址偏移量=DOFFSET × (HSIZE(bit)/8) 例如，当DOFFSET设为10，HSIZE为字(32bit)时，地址偏移量为10 × 4=40，若HSIZE为半字(16bit)时，偏移量为10 × 2=20，若HSIZE为字节(8bit)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	R/W

使用目标设备不连续传输需要配置 DMA_CHxCTL 寄存器的 DNSEQEN 位有效，并且配置 DMA_CHxCTL 寄存器的 DINC 位使地址更新方式为自增或自减。

DMA_DNSEQCTLx, DMA_DNSEQCTLBx 这两个寄存器共用同一个地址，都是用来定义不连续传输的。要使用哪一个取决于该通道是否启用了重置功能。未启用重置时使用 DMA_DNSEQCTLx，启用重置时使用 DMA_DNSEQCTLBx。

14.5.20 目标设备不连续地址传输控制寄存器 B(DMA_DNSEQCTLBx) (x=0~3)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DNSCNTB[11:0]										DNSDIST[19:16]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DNSDIST[15:0]															

位	标记	位名	功能	读写
b31-b20	DNSCNTB[11:0]	目标地址跳转的数据量	设置目标地址跳转前传输的数据量大小。 目标设备在每传输DNSCNTB个数据块后目标地址按DNSDIST指定地址间距跳转。数据块大小由DMA_DTCTLx.BLKSIZE与DMA_CHxCTL.HSIZE决定。	R/W
b19-b0	DNSDIST[19:0]	目标不连续区域地址间距	当不连续地址传输时, 设置目标设备两个不连续区域的间距。 跳转的方向根据通道控制寄存器DMA_CHxCTL.DINC的值向前或向后跳转。参考图 14-4。当DMA_CHxCTL.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_CHxCTL.HSIZE)设置的bit数和DNSDIST的值进行计算。 地址间距=DNSDIST×(HSIZE(bit)/8) 例如, 当DNSDIST设为10, HSIZE为字(32bit)时, 地址间距为10×4=40, 若HSIZE为半字(16bit)时, 间距为10×2=20, 若HSIZE为字节(8bit)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	R/W

使用目标设备不连续传输需要配置 DMA_CHxCTL 寄存器的 DNSEQEN 位有效, 并且配置 DMA_CHxCTL 寄存器的 DINC 位使地址更新方式为自增或自减。

DMA_DNSEQCTLx, DMA_DNSEQCTLBx 这两个寄存器共用同一个地址, 都是用来定义不连续传输的。要使用哪一个取决于该通道是否启用了重置功能。未启用重置时使用 DMA_DNSEQCTLx, 启用重置时使用 DMA_DNSEQCTLBx。

14.5.21 链指针寄存器(DMA_LLPtr) (x=0~3)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
LLP[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LLP[15:2]															

位	标记	位名	功能	读写
b31-b2	LLP[31:2]	链指针	连锁传输有效时，设置下一次传输的描述符所在的地址，地址为字对齐，即LLP[1:0]固定为0	R/W
b1-b0	Reserved	-	读出时为“0”，写入时写“0”	R/W

14.5.22 通道控制寄存器(DMA_CHxCTL) (x=0~3)

复位值: 0x0000_1000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	IE	LLP RUN	LLPE N	HSIZE[1:0]	DNSE QEN	SNSE QEN	DRP TEN	SRPT EN	DINC[1:0]	SINC[1:0]			

位	标记	位名	功能	读写
b31-b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	IE	中断使能位	配置该通道是否产生中断。 0: 该通道不产生中断 1: 该通道产生中断	R/W
b11	LLPRUN	链锁传输模式选择	链锁传输有效时, 设置当前传输完成时, 载入链指针指向的新描述符后是否立即开始新描述符对应的传输 0: 不立即传输, 等待下一个传输请求产生后开始传输 1: 新描述符载入后, 立即开始传输	R/W
b10	LLPEN	链锁传输使能	0: 链锁传输无效 1: 链锁传输有效	R/W
b9-b8	HSIZE[1:0]	传输数据的宽度	00: 8bit 01: 16bit 10, 11: 32bit	R/W
b7	DNSEQEN	目标地址不连续传输使能	0: 不允许不连续地址传输 1: 允许不连续地址传输	R/W
b6	SNSEQEN	源地址不连续传输使能	0: 不允许不连续地址传输 1: 允许不连续地址传输	R/W
b5	DRPTEN	目标重复传输功能使能位	设置是否允许目标地址重新载入初始值 0: 不重载 1: 重载	R/W
b4	SRPTEN	源重复传输功能使能位	设置是否允许源地址重新载入初始值 0: 不重载 1: 重载	R/W
b3-b2	DINC[1:0]	目标地址的更新方式	00: 固定 01: 递增 10, 11: 递减	R/W
b1-b0	SINC[1:0]	源地址的更新方式	00: 固定 01: 递增 10, 11: 递减	R/W

14.5.23 通 道 监 视 寄 存 器 (DMA_MONSAR_x , DMA_MONDAR_x, DMA_MONDTCTL_x, DMA_MONRPT_x, DMA_MONSEQCTL_x, DMA_MONDNSEQCTL_x) (x=0~3)

这些监视寄存器与相应的通道配置寄存器对应，寄存器位配置一致，但全部为只读寄存器。

通道配置寄存器在 DMA 传输前后保持不变，而通道监视寄存器会在 DMA 每完成一个请求对应的传输后，即每完成一个数据块传输后更新。更新内容和方式如下：

- DMA_MONSAR_x.SAR[31:0], DMA_MONDAR_x.DAR[31:0]: 按通道配置寄存器设置的固定/递增/递减/重载/不连续跳转方式更新为下一次传输的地址。
- DMA_MONDTCTL_x.CNT[15:0]: 减 1，若已经是 0 则保持为 0。
- DMA_MONRPT_x.SRPT[9:0], DRPT[9:0]: 通道重置无效时，减去块大小，减到 0 时减或该值小于块大小时重载 DMA_RPT_x 设定值。重置有效时，减 1，减到 0 时重载 DMA_RPTB_x 设定值。
- DMA_MONSEQCTL_x.SNSCNT[11:0] , DMA_MONDNSEQCTL_x.DNSCNT[11:0]: 通道重置无效时，减去块大小，减到 0 时或该值小于块大小值时 DMA_SNSEQCTL_x/DMA_DNSEQCTL_x 原始设定值。重置有效时，减 1，减到 0 时重载 DMA_SNSEQCTLB_x/DMA_DNSEQCTLB_x 设定值。

以上之外的监视寄存器位保持与配置寄存器一致。

14.6 使用注意事项

- 1 DMA 的寄存器只支持 32bit 读写，8/16bit 读写操作无效。
- 2 DMA 传输过程中写通道配置寄存器无效。通道配置寄存器包括：DMA_SARx, DMA_DARx, DMA_DTCTLx, DMA_RPTx, DMA_RPTBx, DMA_SNSEQCTLx, DMA_SNSEQCTLBx, DMA_DNSEQCTLx, DMA_DNSEQCTLBx, DMA_LLPx, DMA_CHxCTL (x=0~3)。请确保 DMA 处于空闲状态时写以上寄存器，或者写寄存器之后再将其读出，确认是否写成功。

15 电压比较器 (CMP)

15.1 简介

电压比较器 (Comparator, 以下简称 CMP) 是将两个模拟电压 (正端电压 INP 和负端电压 INM) 进行比较，并输出比较结果的外设模块，共有 3 个独立的比较通道。每个比较通道的正端电压和负端电压均有 4 个输入源，使用时可以各选定一个进行单一比较，也可以选择多个正端电压与同一个负端电压进行扫描比较。比较结果可通过寄存器读取，也可输出到外部管脚，还可以产生中断和事件。

HC32F460 还搭载了 2 个 8 位的数模转换器 (以下简称 8bitDAC)，其模拟输出可作为负端电压输入源供 CMP 使用。

项目	规格
比较通道	3个比较通道: CMP1~3
正端电压输入源	12个输入源: 10个外部模拟输入, 2个内部PGA输出
负端电压输入源	7个输入源: 4个外部模拟输入, 1个内部Vref, 2个内部D/A输出
比较结果	可从寄存器读取或输出到外部管脚VCOUT, 且极性可选
工作模式	单一比较模式和扫描比较模式
数字滤波	7种采样频率可选, 不需要滤波时可关闭
中断和事件	可检测比较结果的有效边沿, 并产生中断和外设触发事件
低功耗控制	模块停止功能可降低系统功耗

表 15-1 CMP 详细规格

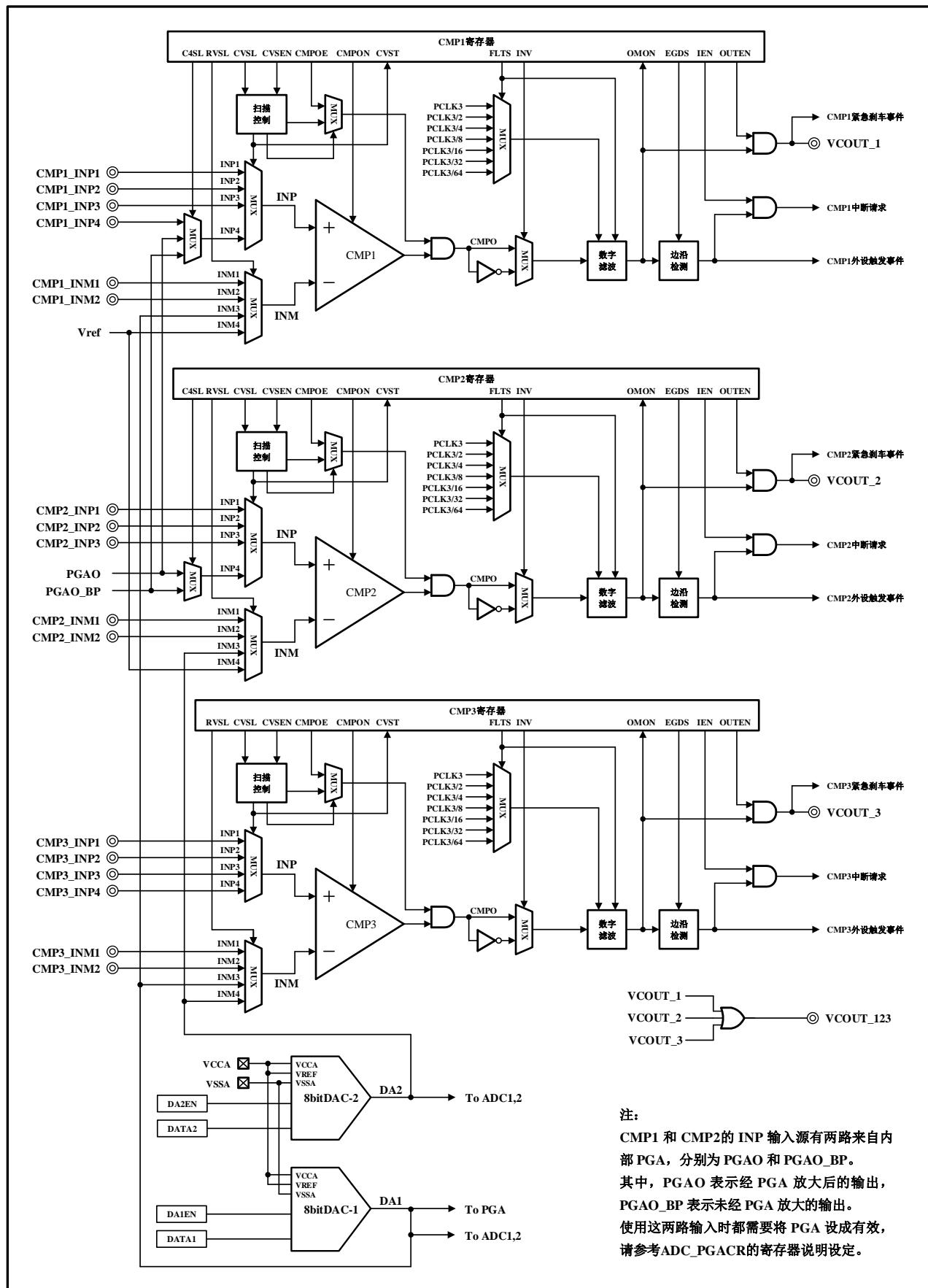


图 15-1 CMP, 8bitDAC 功能连接图

15.2 功能说明

15.2.1 电压比较

CMP 工作在单一比较模式时, 选择任意一个比较通道并为其选定正端电压 INP 和负端 INM, 就可实现这两个电压的比较。如图 15-2 所示为 CMP 的工作示意图。如果设置正极性输出, INP 高于 INM 时 CMP 输出高电平, INP 低于 INM 时 CMP 输出低电平。反之, 如果设置负极性输出, INP 高于 INM 时 CMP 输出低电平, INP 低于 INM 时输出高电平。最终, 比较结果可以从每个比较通道各自的比较结果监视寄存器 CMPMON.OMON 位读取, 也可以输出到外部管脚 VCOUT。

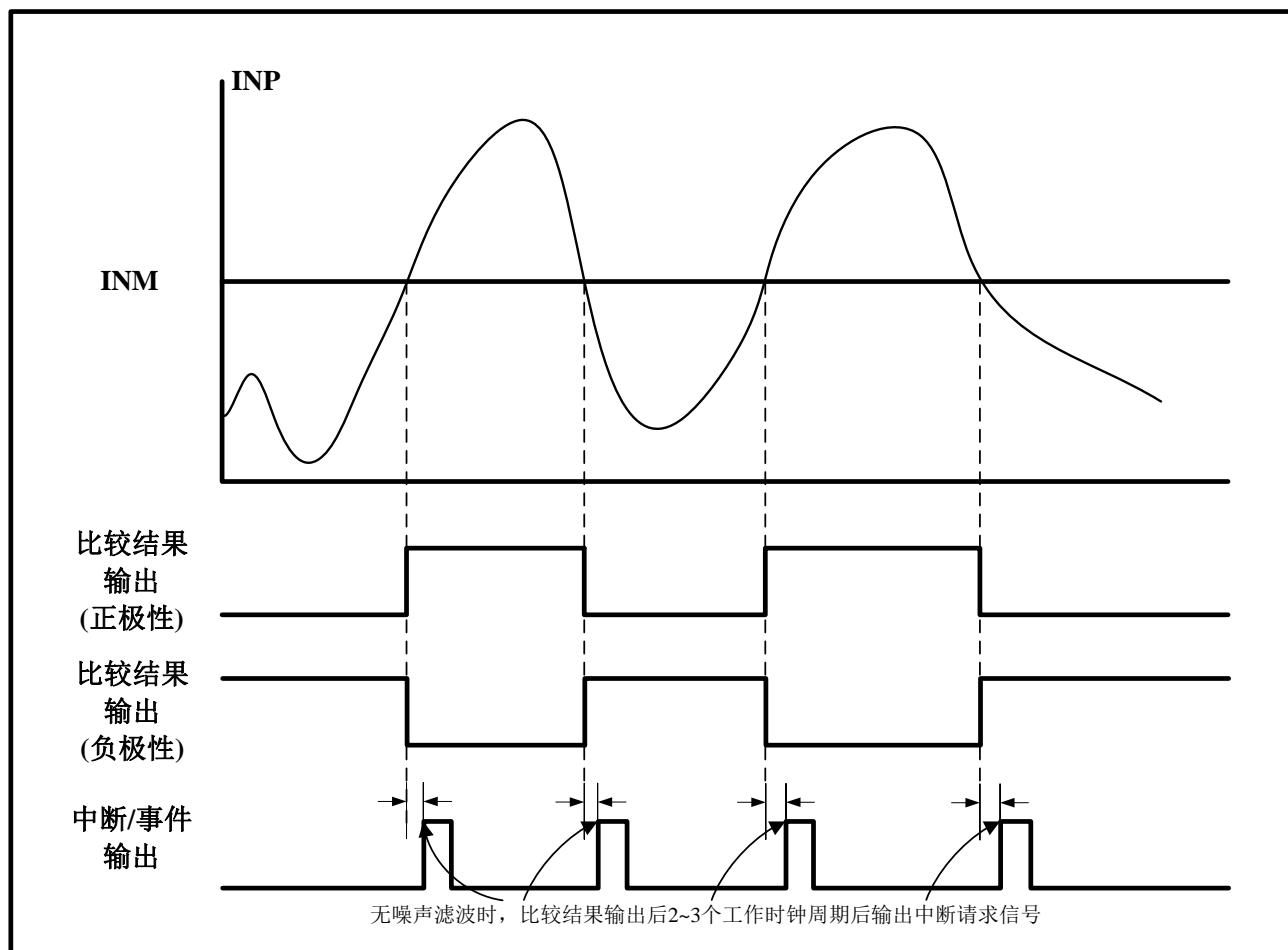


图 15-2 CMP 工作示意图

15.2.2 数字滤波

每个比较通道都有数字滤波电路，可对比较结果进行噪声滤波处理。滤波电路对比较器输出 CMPO 进行三次采样后比较，三次采样结果一致则将采样到的结果作为比较结果输出，三次采样结果不一致则保持比较结果输出不变。采样频率有 7 档可选，具体请参考寄存器 CMP_CTRL.FLTS[2:0]的功能说明。滤波功能也可以关闭，这时比较结果输出与比较器输出 CMPO 完全一致。

15.2.3 中断和事件

比较结果输出发生变化时可以产生中断请求以及触发其他外设启动的事件，产生中断和事件的边沿则可以选择比较结果输出的上升沿、下降沿或双边沿。其中，比较通道 1 (CMP1) 的中断请求还可以唤醒/停止低功耗模式，但必须将中断边沿选择为上升沿并且关闭数字滤波功能。

初次使用比较中断和事件时请参照以下流程进行设定。

- 1) 设定 CMP_VLTSEL，选择 INP 和 INM 输入源。
- 2) 设定 CMP_CTRL.INV，选择比较器输出极性。
- 3) 设定 CMP_CTRL.EDGSL[1:0]，选择中断发生边沿。
- 4) 设定 CMP_CTRL.CMPON，启动比较器并等待 300ns 稳定时间。
- 5) 设定 CMP_CTRL.CMPOE，允许比较器输出。
- 6) 如果使用中断，设定 CMP_CTRL.IEN，允许中断。

如果使用事件，将被启动外设的触发事件选择为 CMP 比较事件。

15.2.4 扫描比较模式

设置寄存器 `CMP_VLTSEL.CVSL[3:0]` 位选择两个或两个以上 INP 作为比较对象时，CMP 即进入扫描比较模式，随后设置寄存器 `CMP_CTRL.CVSEN` 位为 “1” 开始扫描动作。CMP 工作在扫描比较模式时，INM 输入保持不变，INP 输入则会根据设定的扫描周期进行自动切换。如果在扫描过程中监测到比较结果有寄存器 `CMP_CTRL.EDGSL[1:0]` 位选择的有效边沿发生，寄存器 `CMP_CTRL.CVSEN` 位将自动清零并暂停扫描。这时，可以读取寄存器 `CMP_OUTMON.CVST[3:0]` 位来确定当前的 INP 输入。当寄存器 `CMP_CTRL.CVSEN` 位再次被写为 “1” 后，扫描动作继续。

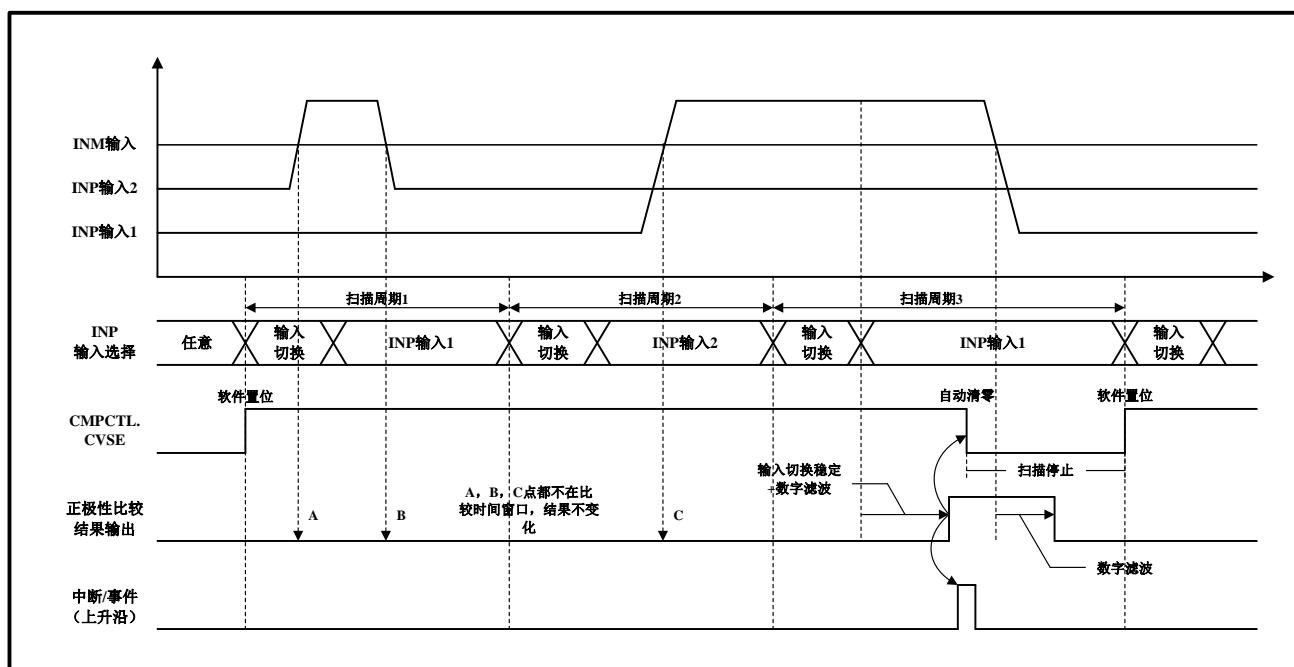


图 15-3 扫描模式动作示意图

上图是扫描两个 INP 输入的动作示意图。在前两个扫描周期，由于 INP 输入 1 和 INP 输入 2 的变化都未发生在各自的比较时间窗口内，比较结果输出保持不变。进入扫描周期 3，INP 输入 1 高于 INM 引起比较器反转，输出经数字滤波后传递到比较结果输出。这时，扫描动作自动停止并产生中断和事件。接下来，CMP 一直保持工作在 INP 输入 1 与 INM 比较的状态，直到软件重启扫描模式。

使用扫描比较模式请参照以下流程进行设定。

- 1) 设定 `CMP_VLTSEL[3:0]`，选择 INP 和 INM 输入源（INP 选择两个或两个以上）。

- 2) 设定 CMP_CVSSTB，设置电压切换稳定时间。
- 3) 设定 CMP_CVSPRD，设置电压扫描周期。
- 4) 设定 CMP_CTRL.INV，选择比较器输出极性。
- 5) 设定 CMP_CTRL.FLTS[2:0]，选择数字滤波采样频率。
- 6) 设定 CMP_CTRL.EDGSL[1:0]，选择中断发生边沿。
- 7) 设定 CMP_CTRL.IEN，允许中断。
- 8) 设定 CMP_CTRL.CMPON，启动比较器并等待 300ns 稳定时间。
- 9) 设定 CMP_CTRL.CMPOE，允许比较器输出。
- 10) 设定 CMP_CTRL.CVSEN，启动扫描。

电压切换稳定时间和电压扫描周期对应的都是 PCLK3 周期数。为了保证动作正确，务必使设定值满足“扫描周期 > 切换稳定时间 + 滤波采样周期 × 4 + 5”的条件。

15.2.5 8bit-DAC 设定

2 个 8bit-DAC 为 CMP 提供了两个内部电压 DA1 和 DA2。工作时的输出电压为：

$$\text{输出电压} = \text{VCCA} \times \text{转换数据} / 255$$

DA1，DA2 的转换数据分别由 CMP_DADR1，CMP_DADR2 来设定。

使用时请参考以下流程进行设定：

- 1) 设定 CMP_DADR1、CMP_DADR2，设置转换数据。
- 2) 设定 CMP_DACR，启动 DA1 和 DA2。
- 3) 等待 DA 转换稳定时间。
- 4) 设定比较器。

15.3 注意事项

15.3.1 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数字部分关闭。CMP 初始为停止状态，设定模块工作时才可以访问 CMP 寄存器。详细说明请参考低功耗章节。

15.3.2 模块停止时的动作

CMP 在工作状态进入到停止状态时，CMP 的模拟部分继续保持工作，功耗等同于工作状态。如需降低功耗，请先将 CMP_CTRL.CMPON 设置为“0”，以停止 CMP 模拟部分的工作。

同样，8bit-DAC 在工作状态进入停止状态时，DA 的模拟部分也继续保持工作，功耗等同于工作状态。如需减少功耗，请先将 DACR 的 DA1EN，DA2EN 位设为“0”，以停止 8bit-DAC 模拟部分的工作。

15.3.3 停止低功耗模式下的动作

HC32F46xx 在 CMP 工作状态进入停止低功耗模式时，CMP 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。在停止低功耗模式下如需进一步降低功耗，请在进入停止低功耗模式前将 CMP_CTRL.CMPON 清“0”，使 CMP 停止工作。

同样，HC32F46xx 在 8bit-DAC 转换时进入停止低功耗模式时，D/A 输出将被保持，功耗等同于进入停止低功耗模式之前的水平。在停止低功耗模式下如需进一步降低功耗，请在进入停止低功耗模式前将 DACR 的 DA1EN，DA2EN 位设为“0”，使 8bit-DAC 停止工作。

15.3.4 掉电低功耗模式下的动作

HC32F46xx 进入掉电低功耗模式时，CMP 和 8bit-DAC 都将停止工作。CMP 比较结果输出 Low，8bit-DAC 输出变为高阻态。

15.4 寄存器说明

CMP1 基准地址: 0x4004_A000

CMP2 基准地址: 0x4004_A010

CMP3 基准地址: 0x4004_A020

寄存器名	符号	偏移地址	位宽	复位值
CMP控制寄存器	CMP_CTRL	0x000	16	0x0000
CMP电压选择寄存器	CMP_VLTSEL	0x002	16	0x0000
CMP结果监视寄存器	CMP_OUTMON	0x004	16	0x0000
CMP稳定时间寄存器	CMP_CVSSTB	0x006	16	0x0005
CMP扫描周期寄存器	CMP_CVSPRD	0x008	16	0x000F
CMP用8bitDAC数据寄存器1	CMP_DADR1	0x100	16	0x0000
CMP用8bitDAC数据寄存器2	CMP_DADR2	0x102	16	0x0000
CMP用8bitDAC控制寄存器	CMP_DACR	0x108	16	0x0000
CMP内部基准电压AD转换寄存器	CMP_RVADC	0x10C	16	0x0000

表 15-2 CMP 寄存器一览表

注意:

- CMP_DADR1,CMP_DADR2,CMP_DACR 为共用寄存器, 只存在于 CMP1 中。

15.4.1 CMP 控制寄存器 (CMP_CTRL)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMP ON	CMP OE	INV	OUT EN	-	-	-	CVS EN	IEN	EDGSL[1:0]	-	-	-	-	-	FLTSL[2:0]

位	标记	位名	功能	读写
b15	CMPON	比较器工作许可	0: 比较器停止 (比较器输出Low) (复位值) 1: 比较器工作 注意: – 比较器工作许可CMPON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。	R/W
b14	CMPOE	比较器输出许可	0: 比较器输出禁止 (比较器输出Low) (复位值) 1: 比较器输出许可	R/W
b13	INV	比较器输出极性选择	0: 比较器正极性输出 (复位值) 1: 比较器负极性输出 (比较器输出被反转) 注意: – 请在比较器输出禁止 (即CMPOE位为 “0”) 时改写INV。改变INV位，可能会引起中断或者外设触发事件，因此请在中断禁止或者外设触发功能无效状态下设定该寄存器。寄存器设定后，请清除相应的中断标志。	R/W
b12	OUTEN	比较结果输出许可	0: 禁止比较结果输出 (比较结果输出Low) (复位值) 1: 允许比较结果输出	R/W
b11~b9	Reserved	-	读出时为“0”,写入时写“0”	R
b8	CVSEN	比较电压扫描许可	0: 比较电压扫描停止 (复位值) 1: 比较电压扫描开始 注意: – 检测到比较结果的有效边沿时，CVSEN将自动清零，如需继续扫描请再次写入“1”。	R/W
b7	IEN	比较中断许可	0: 禁止比较中断 (复位值) 1: 允许比较中断	R/W
b6~b5	EDGSL	比较结果有效边沿	0 0: 不检测比较结果的边沿 (复位值) [1:0] 选择 0 1: 检测比较结果的上升沿 1 0: 检测比较结果的下降沿 1 1: 检测比较结果的上升沿和下降沿	R/W
b4~b3	Reserved	-	读出时为“0”,写入时写“0”	R
b2~b0	FLTSL	滤波采样选择	0 0 0: 不使用噪声滤波 (复位值) [2:0] 0 0 1: 使用PCLK3采样 0 1 0: 使用PCLK3的2分频采样 0 1 1: 使用PCLK3的4分频采样	R/W

1 0 0: 使用PCLK3的8分频采样

1 0 1: 使用PCLK3的16分频采样

1 1 0: 使用PCLK3的32分频采样

1 1 1: 使用PCLK3的64分频采样

注意:

– 请在比较器输出禁止（即CMPOE位为“0”）时改写

FLTSL[2:0]。FLTSL[2:0]从“000b”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FLTSL[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。

15.4.2 CMP 电压选择寄存器 (CMP_VLTSEL)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	C4SL[2:0]			CVSL[3:0]			-	-	-	-	-		RVSL[3:0]		

位	标记	位名	功能	读写
b15	Reserved	-	读出时为“0”,写入时写“0”	R
b14~b12	C4SL[2:0]	INP4输入选择	0 0 0: 无输入（复位值） 0 0 1: 选择内部PGAO输出（只对CMP1和CMP2有效） 0 1 0: 选择内部PGAO_BP输出（只对CMP1和CMP2有效） 1 0 0: 选择外部CMP1_INP4输入（只对CMP1有效）	R/W
			注意: 1. 设定对CMP3无效。 2. 禁止设定其他值。 3. 改写C4SL[2:0]时, 请参照以下步骤: (1) CMP_CTRL.CMPOE位清“0”。 (2) C4SL[2:0]位写入新的值（注意只有1位为“1”）。 (3) 等待输入稳定。 (4) CMP_CTRL.CMPOE位置“1”。 (5) 清除对应的中断标志位。	
b11~b8	CVSL[3:0]	INP输入选择	0 0 0 0: 无输入（复位值） 0 0 0 1: 选择INP1 0 0 1 0: 选择INP2 0 0 1 1: 选择INP1、INP2（扫描模式） 0 1 0 0: 选择INP3 0 1 0 1: 选择INP1, INP3（扫描模式） 0 1 1 0: 选择INP2, INP3（扫描模式） 0 1 1 1: 选择INP1, INP2, INP3（扫描模式） 1 0 0 0: 选择INP4 1 0 0 1: 选择INP1, INP4（扫描模式） 1 0 1 0: 选择INP2, INP4（扫描模式） 1 0 1 1: 选择INP1, INP2, INP4（扫描模式） 1 1 0 0: 选择INP3, INP4（扫描模式） 1 1 0 1: 选择INP1, INP3, INP4（扫描模式） 1 1 1 0: 选择INP2, INP3, INP4（扫描模式） 1 1 1 1: 选择INP1, INP 2, INP 3, INP 4（扫描模式）	R/W
			注意: 1. 具体输入源见“图 15-1”。 2. 选择INP4时请设定C4SL[2:0]选择输入源。 3. 改写CVSL[3:0]时, 请参照以下步骤: (1) CMP_CTRL.CMPOE位清“0”。	

-
- (2) CVSL[3:0]位写入新的值。
 - (3) 等待输入稳定。
 - (4) CMP_CTRL.CMPOE位置“1”。
 - (5) 清除对应的中断标志位。

b7~b4	Reserved	-	读出时为“0”,写入时写“0”	R
b3~b0	RVSL[3:0]	INM输入选择	0 0 0 0: 无输入 (复位值) 0 0 0 1: 选择INM1 0 0 1 0: 选择INM2 0 1 0 0: 选择INM3 1 0 0 0: 选择INM4	R/W

注意:

1. 具体输入源见“图 15-1”。
2. 禁止设定其他值。
3. 改RVSL[3:0]时, 请参照以下步骤:
 - (1) CMP_CTRL.CMPOE位清“0”。
 - (2) RVSL[3:0]位写入新的值 (注意只有1位为“1”)。
 - (3) 等待输入稳定。
 - (4) CMP_CTRL.CMPOE位置“1”。
 - (5) 清除对应的中断标志位。

15.4.3 CMP 结果监视寄存器 (CMP_OUTMON)

复位值: 0x0000

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

位	标记	位名	功能	读写
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R
b11~b8	CVST[3:0]	INP输入状态位	0 0 0 0: 当前INP无输入 0 0 0 1: 当前INP输入选择的是INP1 0 0 1 0: 当前INP输入选择的是INP2 0 1 0 0: 当前INP输入选择的是INP3 1 0 0 0: 当前INP输入选择的是INP4	R
注意:				
1. 本寄存器只读, 写入时写全“0”。 2. 具体输入源见图 15-1。				
b7~b1	Reserved	-	读出时为“0”,写入时写“0”	R
b0	OMON	比较结果监视位	0: 比较结果输出为“0” (复位值) 1: 比较结果输出为“1”	R
注意:				
1. 本寄存器只读, 写入时写全“0”。 2. 在CMP_CTRL.FLTS[2:0]=000b (未使用噪声滤波电路) 的情况下设定比较器工作时, 请采用2次读OMON位状态一致的方法来确定比较状态。				

15.4.4 CMP 稳定时间寄存器 (CMP_CVSSTB)

复位值: 0x0005

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

位	标记	位名	功能	读写
b15~b4	Reserved	-	读出时为“0”, 写入时写“0”	R
b3~b0	STB[3:0]	输入切换稳定时间	设定扫描比较模式下切换INP输入时CMP的输出稳定时间。 复位值为0x5, 设定值可以是0x0 ~ 0xF之间的任意值。 CMP输出稳定时间 = PCLK3周期 × STB设定值 例如, PCLK3为40MHz, STB设定为0x5, 那么 扫描周期 = 25(nS) × 0x5 = 125(nS) 注意: 1. 请参考 数据手册电气特性 中“比较器特性-输入通道切换稳定时间”。 2. 如需改写STB, 请先停止扫描并切换到单一比较模式。	R/W

15.4.5 CMP 比较电压扫描周期寄存器 (CMP_CVSPRD)

复位值: 0x000F

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

位	标记	位名	功能	读写
b15~b8	Reserved	-	读出时为“0”, 写入时写“0”	R
b7~b0	PRD[7:0]	比较电压扫描周期	设定扫描比较模式下切换INP输入的时间间隔, 即扫描周期。 复位值为0x0F, 设定值可以是0x0F ~ 0xFF之间的任意值。 扫描周期 = PCLK3周期 × PRD设定值 例如, PCLK3为40MHz, PRD设定为0x50, 那么 扫描周期 = 25(nS) × 0x50 = 2(uS) 注意: 1. 为了保证动作正确, 设定PRD时请确认 扫描周期 > 切换稳定时间 + 滤波采样周期 × 4 + 5 2. 如需改写PRD, 请先停止扫描并切换到单一比较模式。	R/W

15.4.6 CMP 用 8bit-DAC 控制寄存器(CMP_DACR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	DA2 EN	DA1 EN

位	标记	位名	功能	读写
b15~b2	Reserved	-	读时读出“0”，写时请写“0”。	R
b1	DA2EN	DA2启动位	0: DA2停止（复位值） 1: DA2工作	R/W
b0	DA1EN	DA1启动位	0: DA1停止（复位值） 1: DA1工作	R/W

15.4.7 CMP 用 8bit-DAC 数据寄存器(CMP_DADR1,CMP_DADR2)

复位值: 0x0000

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

位	标记	位名	功能	读写
b15~b8	Reserved	-	读时读出“0”，写时请写“0”。	R
b7~b0	DATA[7:0]	DA转换数据	8'h00 (复位值) ~ 8'hFF之间的任意值	R/W

15.4.8 CMP 内部基准电压 AD 转换寄存器(CMP_RVADC)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
						WPRT[7:0]		-	-	-	VREF SW	-	-	DA2 SW	DA1 SW

位	标记	位名	功能	读写
b15~b8	WPRT[7:0]	写保护位	8'h55: VREFSW, DA2SW, DA1SW位写有效 others: VREFSW, DA2SW, DA1SW位写无效 注意: WPRT在非“8'h55”时下写任何值都将恢复成“8'h00”。	R/W
b7~b5	Reserved	-	读时读出“0”，写时请写“0”。	R
b4	VREFSW	内部Vref AD转换开关	0: 内部Vref AD转换通路断开 1: 内部Vref AD转换通路连接 注意: VREFSW, DA2SW, DA1SW只能一位为“1”。	R/W
b3~b2	Reserved	-	读时读出“0”，写时请写“0”。	R
b1	DA2SW	DA2 AD转换开关	0: DA2 AD转换通路断开 1: DA2 AD转换通路连接 注意: VREFSW, DA2SW, DA1SW只能一位为“1”。	R/W
b0	DA1SW	DA1 AD转换开关	0: DA1 AD转换通路断开 1: DA1 AD转换通路连接 注意: VREFSW, DA2SW, DA1SW只能一位为“1”。	R/W

16 模数转换模块 (ADC)

16.1 简介

本产品的 ADC 是一种采用逐次逼近方式的模拟数字转换器，最高 12 位分辨率，最多支持 17 个模拟输入通道，可以转换来自外部引脚、以及芯片内部的模拟信号。这些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，检测是否超出用户设定的阀值。

ADC 主要特性

- 高性能
 - 可配置 12 位、10 位和 8 位分辨率
 - 数字接口时钟 PCLK4 和 A/D 转换时钟 ADCLK 的频率比可选择：
PCLK4: ADCLK=1: 1, 2: 1, 4: 1, 8: 1, 1: 2, 1: 4
ADCLK 可选与系统时钟 HCLK 异步的 PLL 时钟，此时
PCLK4: ADCLK=1: 1
 - 采样率：2.5MSPS (PCLK4=ADCLK=60MHz, 12 位, 采样 11 周期)
 - 各通道采样时间独立编程
 - 各通道独立数据寄存器
 - 数据寄存器可配置数据对齐方式
 - 连续多次转换平均功能
 - 模拟看门狗，监视转换结果
 - 不使用时可以将 ADC 模块设定成停止状态
- 模拟输入通道
 - 最大 16 个外部模拟输入通道
 - 1 个内部基准电压/8bitDAC 输出的检测通道
- 转换开始条件
 - 软件设置转换开始

- 外部事件触发转换开始
- 外部引脚触发转换开始
- 转换模式
 - 2 个扫描序列 A、B， 可任意指定单个或多个通道
 - 序列 A 单次扫描
 - 序列 A 连续扫描
 - 双序列扫描，序列 A、B 独立选择触发源，序列 B 优先级高于 A
 - 同步模式（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
 - 模拟看门狗通道比较中断和事件 ADC_CHCMP，序列比较中断和事件 ADC_SEQCMP
 - 上述的 4 个事件输出都可启动 DMA

16.2 ADC 系统框图

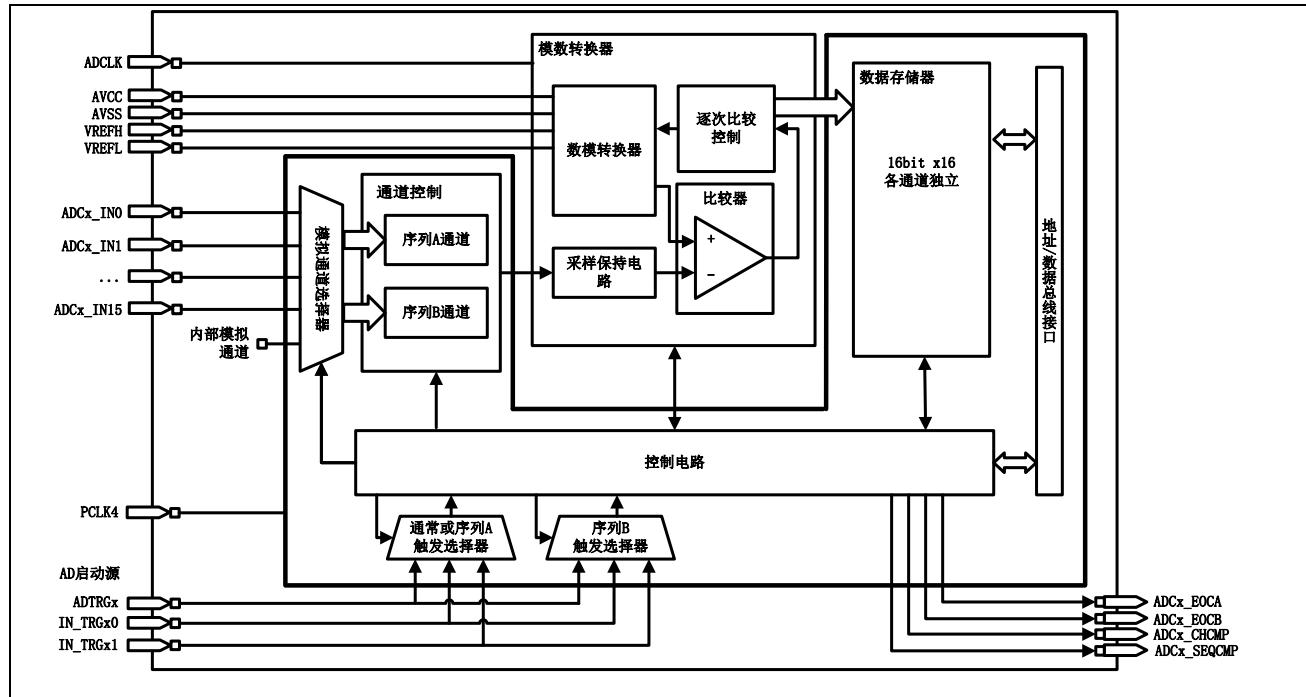


图 16-1 ADC 框图

本芯片搭载了 2 个 ADC 模块单元，每个单元的配置有所不同，具体参考下表：

项目	单元1 (ADC1)		单元2 (ADC2)
电源	AVCC		
	AVSS/VREFL		
基准电压	VREFH *1		
模拟通道 *2	CH0	ADC1_IN0	ADC12_IN4
	CH1	ADC1_IN1	ADC12_IN5
	CH2	ADC1_IN2	ADC12_IN6
	CH3	ADC1_IN3	ADC12_IN7
	CH4	ADC12_IN4	ADC12_IN8
	CH5	ADC12_IN5	ADC12_IN9
	CH6	ADC12_IN6	ADC12_IN10
	CH7	ADC12_IN7	ADC12_IN11
	CH8	ADC12_IN8	内部模拟通道（基准电压/8bitDAC输出）
	CH9	ADC12_IN9	-
	CH10	ADC12_IN10	-
	CH11	ADC12_IN11	-
	CH12	ADC1_IN12	-
	CH13	ADC1_IN13	-
	CH14	ADC1_IN14	-
	CH15	ADC1_IN15	-
	CH16	内部模拟通道（基准电压/8bitDAC输出）	-
PGA	ADC1_IN0~3, ADC12_IN4~7, 8bitDAC_1输出中任意1通道		-
硬件触发源	外部引脚	ADTRG1	ADTRG2
	片内外设	IN_TRG10	IN_TRG20
		IN_TRG11	IN_TRG21

表 16-1 各 ADC 单元规格

注意：

- LQFP100 封装下 VREFH 可用，其他封装下不可用，使用 AVCC 代替 VREFH。
- ADC 模拟通道 CH0~CH15 与实际输入 ADCx_INy 可以设置寄存器自由映射，本表所示为复位后默认的映射关系。

16.3 功能说明

16.3.1 ADC 时钟

ADC 模块需要使用两个时钟：模拟电路时钟 ADCLK，数字接口时钟 PCLK4。

两个时钟都来自时钟控制器中的分频器。ADCLK 等效于 PCLK2，与 PCLK4 是同步关系，PCLK4 与 ADCLK 的频率比率关系为 1: 1, 2: 1, 4: 1, 8: 1, 1: 2, 1: 4。

ADCLK 可以选择与系统时钟 HCLK 异步的 PLL 时钟源，此时 PCLK4 与 ADCLK 相同，为同步同频率关系。ADCLK 的最高频率为 60MHz。

16.3.2 通道选择

ADC 模块有多个通道，可配置为两个序列：序列 A，序列 B 进行转换。序列 A 和 B 配有独立的通道选择寄存器 ADC_CHSELRA, ADC_CHSELRB。寄存器每位代表一个通道，如 bit0 位写 1 表示转换 CH0，写 0 表示不转换 CH0。两个序列可独立选择任意 1 个或多个通道进行转换。例如：ADC_CHSELRA 设置为 0x0055，ADCHSELRB 设置为 0x0002，则序列 A 的触发条件发生时，将依次转换 CH0, CH2, CH4 和 CH6 这 4 个通道。序列 B 的触发条件发生时，将转换 CH1 这一个通道。

对于内部模拟通道，有 3 个可选的模拟量：8bitDAC_1 输出、8bitDAC_2 输出、内部基准电压（约 1.1V）。如下图所示。在选择内部模拟通道作为 ADC 转换对象时，需要先设置内部通道选择寄存器，再设置 ADC 的通道选择寄存器 ADC_CHSELRA/ADC_CHSELRB 选择内部通道，才可以启动 ADC 转换。

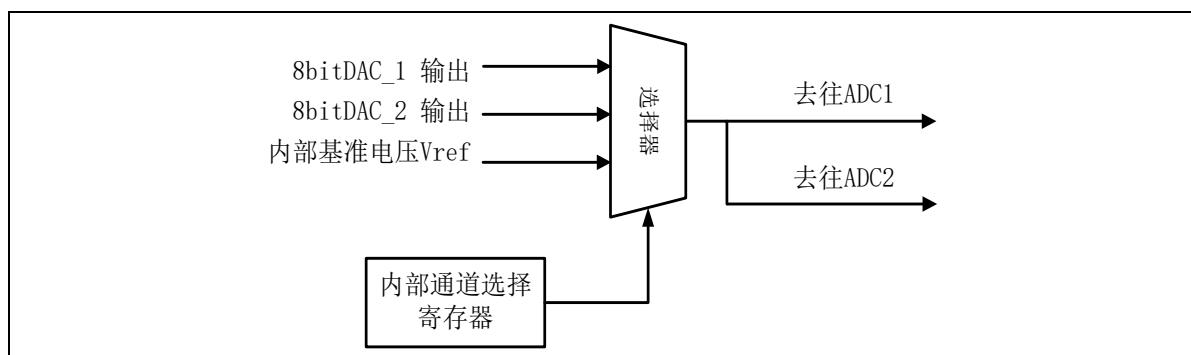


图 16-2 内部模拟通道选择

内部通道选择寄存器的具体设置方法，请参考【电压比较器（CMP）】章节中寄存器 CMP_RVADC 说明以及【电源控制（PWC）】章节中寄存器 PWC_PWCMR 说明。

转换目标	ADC_CHSELR	CMP_RVADC	CMP_DADR	CMP_DACR	PWC_PWCMR
8bitDAC_1 -> ADC1	ADC1_CHSELRA1 或 ADC1_CHSELRB1	写0x0001 *1	写CMP_DADR1指定 8bitDAC_1输出电压值	bit0写1, 允许 8bitDAC_1输出	—
8bitDAC_2 -> ADC1	bit0写1, 选择CH16	写0x0002 *1	写CMP_DADR1指定 8bitDAC_2输出电压值	bit1写1, 允许 8bitDAC_2输出	—
内部基准电压vref -> ADC1		写0x0010 *1	—	—	写0x80 *2
8bitDAC_1 -> ADC2	ADC2_CHSELRA0 或 ADC2_CHSELRB0	写0x0001 *1	写CMP_DADR1指定 8bitDAC_1输出电压值	bit0写1, 允许 8bitDAC_1输出	—
8bitDAC_2 -> ADC2	bit8写1, 选择CH8	写0x0002 *1	写CMP_DADR1指定 8bitDAC_2输出电压值	bit1写1, 允许 8bitDAC_2输出	—
内部基准电压vref -> ADC2		写0x0010 *1	—	—	写0x80 *2

表 16-2 转换内部通道时的寄存器设定方法

注意：

- CMP_RVADC 需要先写 0x5500 之后，再写目标设定值。
- PWC_PWCMR 寄存器需要先打开保护位 PWC_FPRC.FPRCB1。

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。

CH0 表示通道 0，与实际模拟输入通道 ADCx_INy 的对应关系可通过寄存器 ADC_CHMUXR 自由设置。例如：对于 ADC1，CH00MUX 设置为 0x0 表示 CH0 映射为 ADC1_IN0，设置为 0xf 表示 CH0 映射为 ADC1_IN15。对于 ADC2，CH00MUX 设置为 0x0 表示 CH0 映射为 ADC12_IN4，设置为 0xF 为无效映射。同理 CH1~CH15 均可通过对应的 ADC_CHMUXR 寄存器进行通道映射。

16.3.3 触发源选择

序列 A, 序列 B 独立选择触发源。可选择的触发源包括外部端口 ADTRGx, 内部事件 IN_TRGx0, IN_TRGx1。其中, 端口 ADTRGx 下降沿输入有效。IN_TRGx0, IN_TRGx1 由寄存器 ADC_ITRGSELR0,1 设置, 可以选择芯片内部丰富的事件源。此外, 写寄存器 ADC_STR 可生成软件触发信号, 软件触发只能在 ADC 处于待机状态时使用, 并且只适用于序列 A。

16.3.4 序列 A 单次扫描模式

A/D 控制寄存器 ADC_CR0.MS[1:0]设为 00b 选择序列 A 单次扫描模式。

该模式下, 当寄存器 ADC_TRGSR 选择的序列 A 启动条件发生, 或者 ADC_STR.STRT 位写 1 软件触发, ADC 启动, 对序列 A 通道选择寄存器 ADC_CHSELRA 中选择的所有通道依次进行采样和转换, 转换结果存入相应的数据寄存器 ADC_DR 中。ADC 转换过程中 ADC_STR.STRT 保持为 1, 当所有通道转换结束后自动清 0, ADC 进入转换待机状态, 等待下次触发条件的发生。

当所有通道转换结束时, 序列 A 转换结束标志位 ADC_ISR.EOCAF 置 1, 并产生序列 A 转换结束事件 ADC_EOCA, 可以用此事件启动 DMA。若 ADC_ICR.EOCAIEN 为 1, 中断许可的状态下, 同时还产生序列 A 转换结束中断请求。

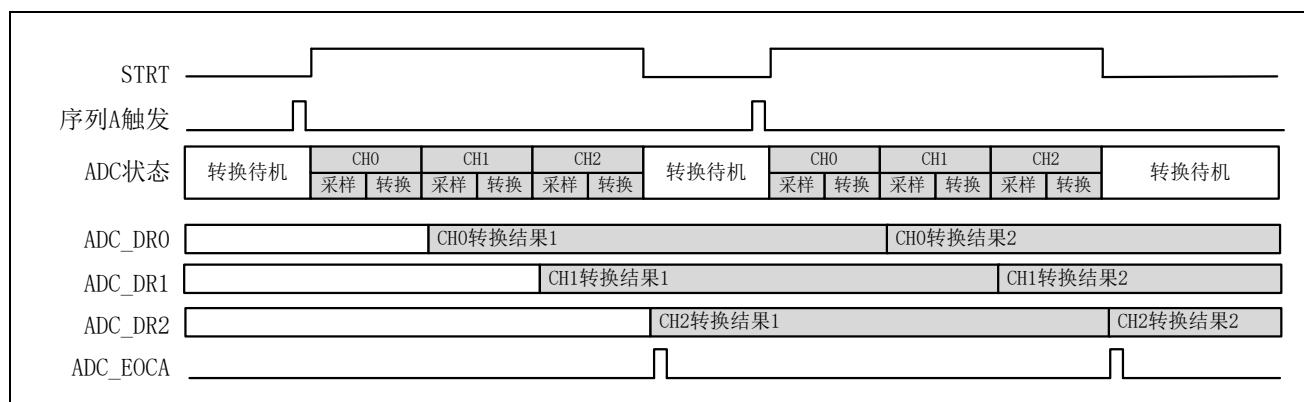


图 16-3 序列 A 单次扫描模式

序列 A 单次扫描模式的软件流程:

1. 确认 ADC_STR.STRT 为 0, ADC 处于转换待机状态。
2. A/D 控制寄存器 ADC_CR0.MS[1:0]设为 00b 选择序列 A 单次扫描模式。
3. 设置序列 A 通道选择寄存器 ADC_CHSELRA。
4. 设置采样时间寄存器 ADC_SSTR。

5. ADC_STR.SRT 写 1 软件触发序列 A，或者设置寄存器 ADC_TRGSR 选择序列 A 触发条件。
6. 查询序列 A 转换结束标志位 EOCAF。
7. 读取各通道数据寄存器 ADC_DR。
8. 写 0 清除 EOCAF 标志位，为下次转换准备。

上面的步骤 6~8 的 CPU 查询方式也可以替换成中断方式，利用 ADC_EOCA 中断来处理转换数据。或者利用 ADC_EOCA 事件启动 DMA 读取数据。

16.3.5 序列 A 连续扫描模式

A/D 控制寄存器 ADCR0.MS[1:0]设为 01b 选择序列 A 连续扫描模式。

序列 A 连续扫描模式与序列 A 单次扫描模式动作类似，不同点在于，连续模式在所用通道转换结束后不是进入转换待机状态，而是重新开始转换序列 A。SRT 位也不会自动清 0。

当需要停止连续扫描时，对 SRT 位写 0，并读 SRT 确认为 0 以判断 ADC 进入转换待机状态。

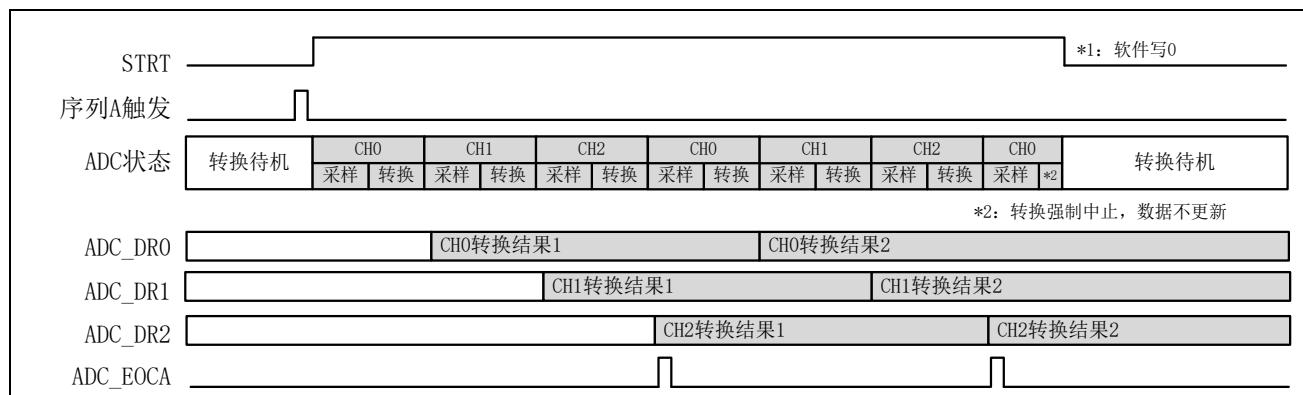


图 16-4 连续扫描

序列 A 连续扫描模式的软件流程：

1. 确认 ADC_STR.SRT 为 0，ADC 处于转换待机状态。
2. A/D 控制寄存器 ADC_CR0.MS[1:0]设为 01b 选择序列 A 连续扫描模式。
3. 设置序列 A 通道选择寄存器 ADC_CHSELRA。
4. 设置采样时间寄存器 ADC_SSTR。
5. ADC_STR.SRT 写 1 软件触发序列 A，或者设置寄存器 ADC_TRGSR 选择序列 A 触发条件。

6. 查询序列 A 转换结束标志位 EOCAF。
7. 读取各通道数据寄存器 ADC_DR。
8. 写 0 清除 EOCAF 标志位，为下次转换准备。
9. 不需要继续转换时，对 STRT 位写 0，并读 STRT 确认为 0 以判断 ADC 进入转换待机状态。

上面的步骤 6~8 的查询方式也可以替换成中断方式，利用 ADC_EOCA 中断来处理转换数据。或者利用 ADC_EOCA 事件启动 DMA 读取数据。

注意：

- 由于是连续转换，每次扫描的间隔比较短，特别是只选择 1 个通道转换时。推荐使用 ADC_EOCA 事件启动 DMA 读取数据，避免查询方式下处理不及时导致数据丢失。

16.3.6 双序列扫描模式

A/D 控制寄存器 ADC_CR0.MS[1:0]设为 10b 或 11b 选择双序列扫描模式，即序列 A 与序列 B 都可由各自选择的触发条件来启动扫描。

当 MS[1:0]=10b 时，序列 A 和 B 等效于两个独立的单次扫描序列。MS[1:0]=11b 序列 A 为连续扫描模式，B 为单次扫描模式。

序列 A 由 ADC_TRGSR.TRGSELA[2:0]选择触发源，由 ADC_CHSELRA 选择转换的通道。序列 B 由 ADC_TRGSR.TRGSELB[2:0]选择触发源，由 ADC_CHSELRB 选择转换的通道。

当序列 A 全部通道转换结束时，序列 A 转换结束标志位 ADC_ISR.EOCAF 置 1，并产生序列 A 转换结束事件 ADC_EOCA，若 ADC_ISCR.EOCAIEN 为 1，中断许可的状态下，同时产生序列 A 转换结束中断请求。当序列 B 全部通道转换结束时，序列 B 转换结束标志位 ADC_ISR.EOCBF 置 1，并产生序列 B 转换结束事件 ADC_EOCB，若 ADC_ISCR.EOCBIEN 为 1，中断许可的状态下，同时产生序列 B 转换结束中断请求。双序列扫描模式下，当序列 A 与序列 B 发生竞争时，序列 B 将会被优先处理，即序列 B 优先级高于序列 A。具体情况请参看下表。

A/D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从第 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		

表 16-3 序列 A 和 B 的各种竞争

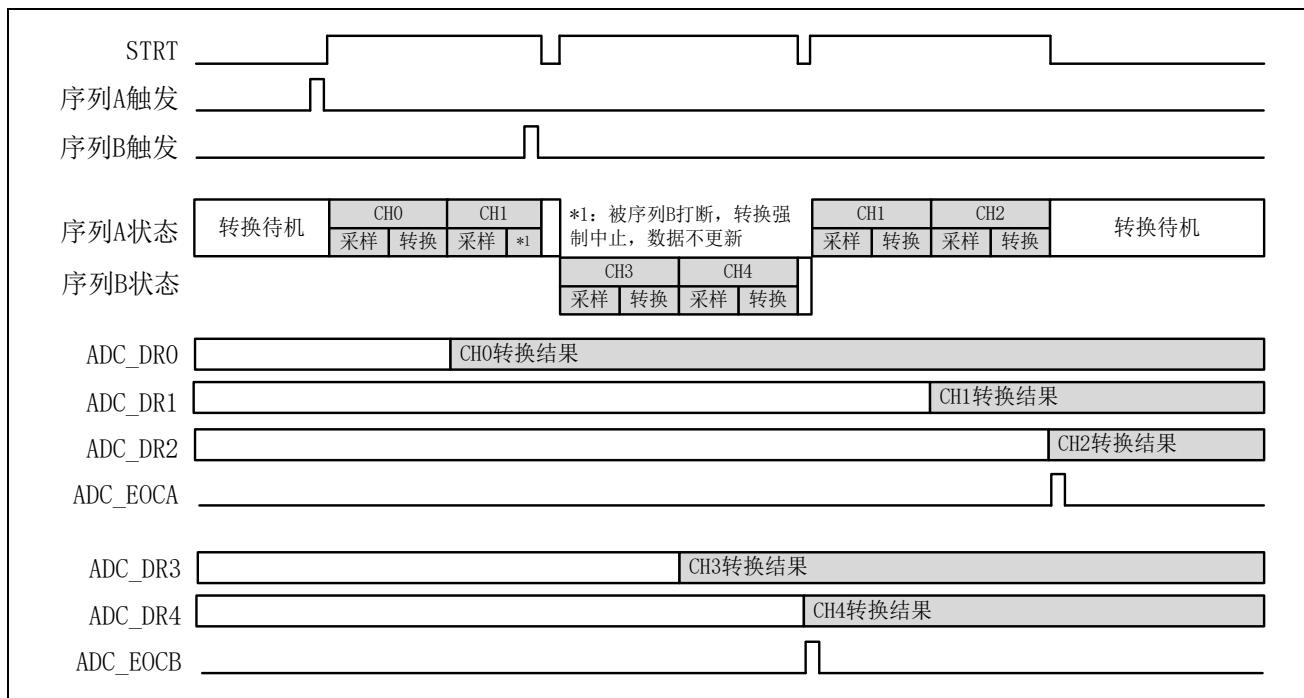


图 16-5 双序列扫描模式 (序列 A 从被中断通道重新启动)

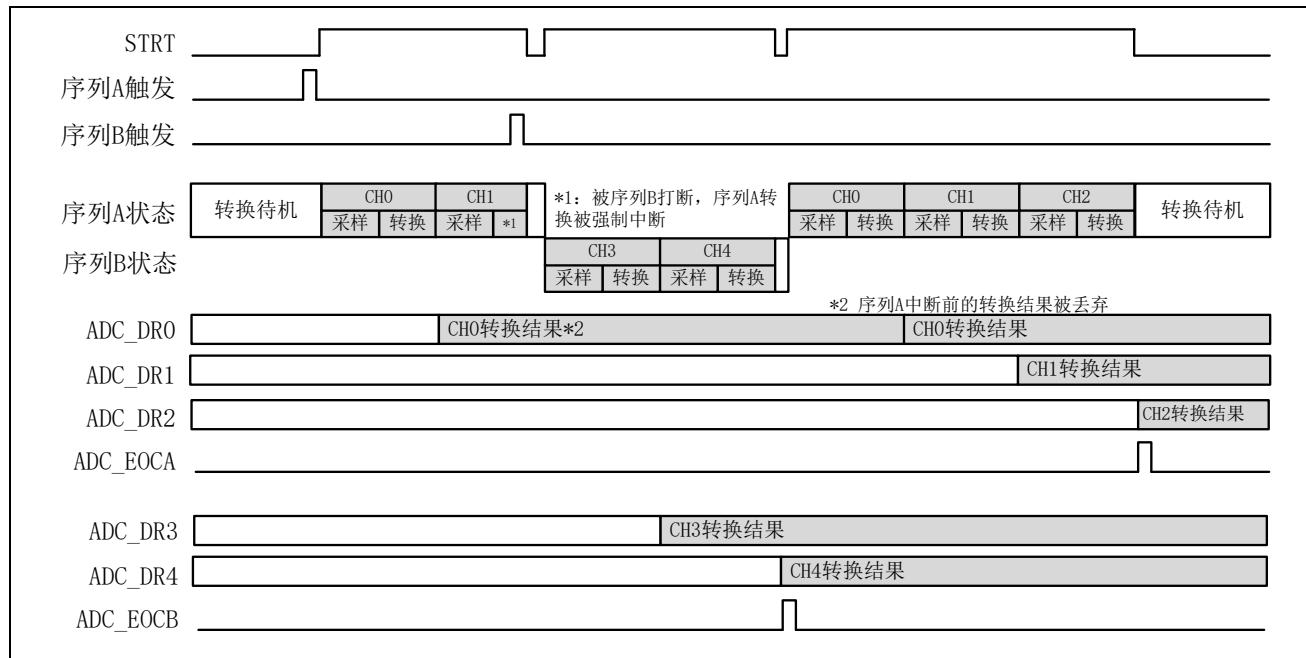


图 16-6 双序列扫描模式（序列 A 从第一个通道重新启动）

双序列扫描模式的软件流程：

1. 确认 `ADC_STR.STRT` 为 0, ADC 处于转换待机状态。
2. A/D 控制寄存器 `ADC_CR0.MS[1:0]` 设为 10b 或 11b 选择双序列扫描模式。
3. 设置寄存器 `ADC_CR1.RSCHSEL` 选择序列 A 被打断后启动方式。
4. 设置序列 A 通道选择寄存器 `ADC_CHSELRA`。
5. 设置序列 B 通道选择寄存器 `ADC_CHSELRB`。
6. 设置采样时间寄存器 `ADC_SSTR`。
7. 设置寄存器 `ADC_TRGSR` 选择序列 A 和 B 触发条件。
8. 通过查询 `EOCAF`, `EOCBF`, 或者 `ADC_EOCA`, `ADC_EOCB` 中断, 或者启动 DMA 在序列 A 或 B 转换结束后处理转换数据。

注意：

- 不要在序列 A 和 B 中选择相同的通道。序列 A 和 B 不要选择相同的触发源。

16.3.7 模拟看门狗功能

模拟看门狗功能是指在通道的 A/D 转换结束时对转换结果进行比较，如下图所示，若转换结果在保护区域内，则生成通道比较中断和事件 `ADC_CHCMP`。并在整个序列扫描结束时，根据各通道比较结果生成序列比较中断和事件 `ADC_SEQCMP`。可以选择对任意单个或多个通道进行比较。多个通道比较有效，任意一个通道比较一致便会产

生序列比较中断和事件。

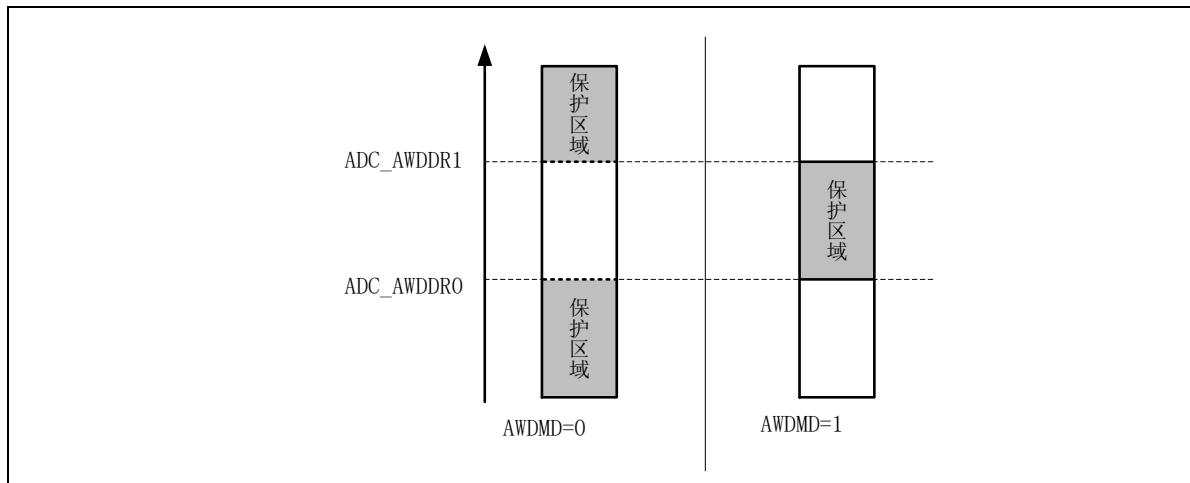


图 16-7 模拟看门狗保护区域（比较条件）

使用模拟看门狗功能的软件流程：

1. 设置阀值寄存器 AD_AWDDR0, ADC_AWDDR1
2. 设置比较通道寄存器 ADC_AWDCHSR, 选择任意单个或多个需要比较的通道
3. 设置 ADC_AWDCR.AWDMD 选择比较条件
4. 设置 ADC_AWDCR.AWDSS[1:0] 选择序列比较中断和事件输出，设置 ADC_AWDCR.AWDIEN 中断许可位。
5. 设置 ADC_AWDCR.AWDEN 允许模拟看门狗功能
6. 根据前文，设置扫描模式，启动 AD 进行转换。
7. 在 ADC_CHCMP/ADC_SEQCMP 中断中，或 A/D 转换结束后，查询比较状态寄存器 ADC_AWDSR，对比较结果进行做相应的处理。

16.3.8 模拟输入的采样时间和转换时间

在单次扫描模式中，A/D 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_SSTRx 调整采样周期数。 t_{CMP} 表示逐次比较时间，12 位精度

13 个 ADCLK, 10 位精度 11 个 ADCLK, 8 位精度 9 个 ADCLK。

一次扫描转换的时间 $t_{SCAN} = t_D + \sum t_{CONV} + t_{ED}$ 。其中 $\sum t_{CONV}$ 表示所有扫描通道的转换时间总和，由于可以独立设置采样时间 t_{SPL} ，各通道的转换时间 t_{CONV} 可以不同。

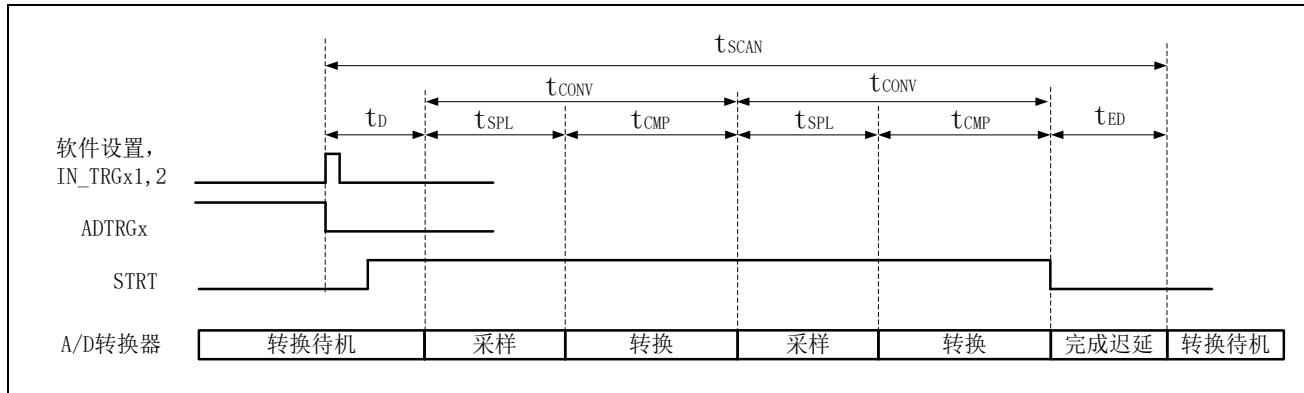


图 16-8 A/D 转换时间

标记	说明	条件							
		同步周边触发	异步周边触发 *注	外部引脚触发	软件触发				
t_D	扫描开始	ADC空闲中，启动转换	1 PCLK4 + 4 ADCLK	3 PCLK4 + 4 ADCLK + 1 PCLK4_SYNC	3 PCLK4 + 4 ADCLK				
	处理时间	序列A转换中被打断，启动序列B转换	2 PCLK4 + 6 ADCLK	4 PCLK4 + 6 ADCLK + 1 PCLK4_SYNC	4 PCLK4 + 6 ADCLK				
t_{CONV}	t_{SPL}	采样时间	ADSSTRx.SST[7:0] x ADCLK						
	t_{CMP}	逐次转换时间	12位分辨率	13 ADCLK					
			10位分辨率	11 ADCLK					
			8位分辨率	9 ADCLK					
t_{ED}	扫描完成处理时间		1 PCLK4 + 3 ADCLK						
t_{TD}	最小连续触发时间间隔		$\sum t_{CONV} + 2 PCLK4 + 5 ADCLK$						

表 16-4 AD 转换时间

注意：

- 异步周边触发是指 ADC 模块选择与系统时钟异步的 PLL 时钟动作时周边触发的情况。此时周边模块时钟与 ADC 模块时钟为异步关系。PCLK4_SYNC 表示 ADC 模块原来的同步时钟，此时的 PCLK4, ADCLK 相同，均为异步的 PLL 时钟。

16.3.9 A/D 数据寄存器自动清除功能

当 ADC_CR0.CLREN 为“1”，A/D 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，A/D 转换结果(0x0222)由于某种原因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)。A/D 转换完成中断处理中将读取未被更新的(0x0111)。为检测 A/D 转换值是否被更新，需要额外将前次转换值存储到 RAM 中，通过对比转换结果来判断。
- 如果 ADC_CR0.CLREN 为“1”，自动清除功能许可的情况下，前次的转换结果(0x0111)被 CPU 或 DMA 读取后，ADC_DR 寄存器将自动被清除为“0x0000”，此后进行 A/D 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 A/D 转换数据是否正确被存储。

16.3.10 转换数据平均计算功能

A/D 转换平均计算功能是指对同一通道进行连续进行 2,4,8,16,32,64,128 或 256 次转换，并将转换结果进行平均后保存到数据寄存器的功能。使用平均计算功能可以去除一定的噪声成分使转换结果更加精确。

寄存器 ADC_CR0.AVCNT[2:0]设置连续转换的次数，寄存器 ADC_AVCHSEL[0]选择任意一个或多个需要平均的通道。

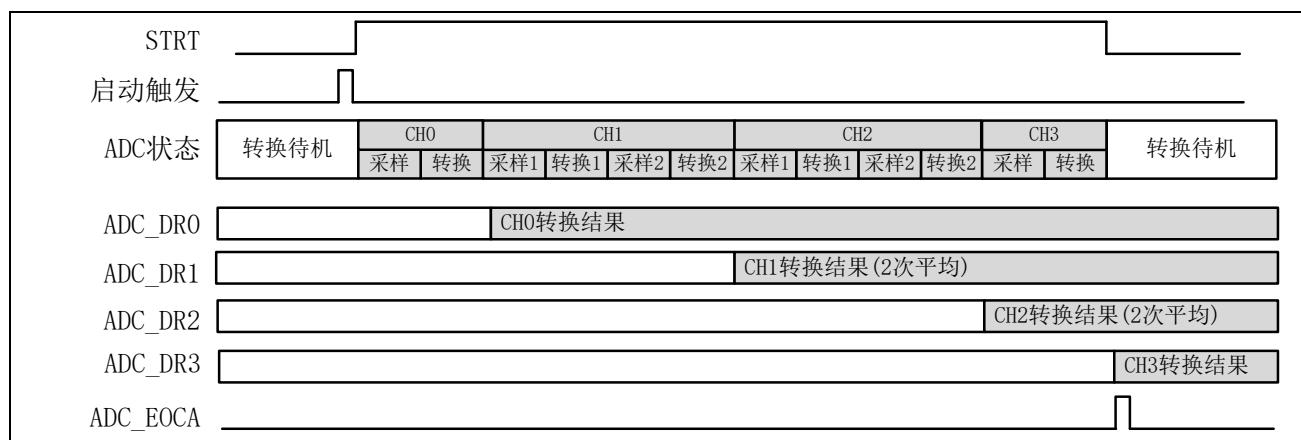


图 16-9 平均功能有效时的转换动作

图 16-9 中，序列 A 单次扫描模式，选择转换 CH0~3 这 4 个通道，其中 CH1、2 设置为 2 次平均模式。在扫描过程中，CH1、2 都会进行连续两次转换，并将平均后的结果保存至对应通道的数据寄存器 ADC_DR1、2 中。

16.3.11 可编程增益放大器 PGA

搭载可编程增益放大器 PGA 时，可以设置寄存器 ADC_PGAINSR 选择 PGA 的输入源，设置寄存器 ADC_PGACR，使 PGA 电路有效，设置寄存器 ADC_PGAGSR 选择增益倍数，增益范围 x2~x32 可选择。此时，模拟输入先经过 PGA 电路进行放大，然后再输入到 ADC 模块进行转换。

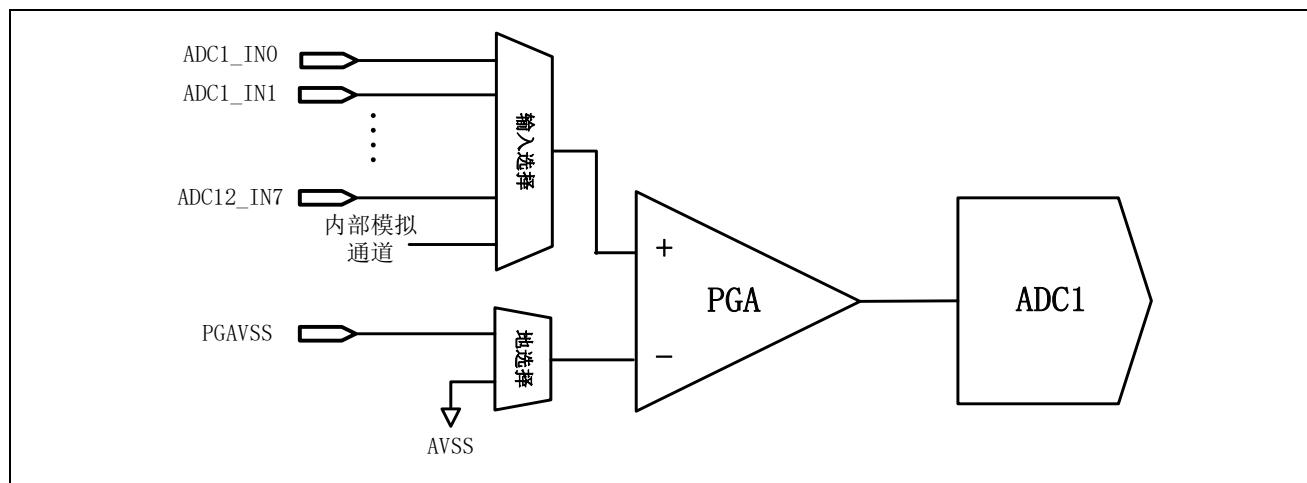


图 16-10 PGA 工作示意图

例如，若需要对 ADC1_IN2 进行放大后转换，可设置通道选择寄存器 ADC_CHSELRA0 为 0x4 选择 CH2 (ADC1_IN2)，设置 PGA 输入源寄存器 ADC_PGAINSR 为 0x4 选择 ADC1_IN2，设置 PGA 有效以及放大倍数然后启动转换。同理，若需要对内部模拟通道进行放大转换，可设置 ADC_CHSELRA1 为 0x1 选择 CH16，设置 ADC_PGAINSR 为 0x100，然后启动转换。

注意：

- PGA 输入源选择内部模拟通道时，模拟量固定为 8bitDAC_1 输出，与 CMP_RVADC 设置无关。
- 只有 ADC1 支持 PGA。

16.3.12 多 ADC 协同工作模式

在具有两个或三个 ADC 模块的芯片上，可以使用 ADC 协同工作模式。

在 ADC 协同工作模式下，通过 ADC1 的触发信号来同步 ADC2 和 ADC3 的转换。即 ADC2 和 ADC3 的序列 A 触发源选择寄存器 ADC_TRGSR.TRGSELA[2:0]设置无效。所有 ADC 模块均由 ADC1 的序列 A 触发源选择寄存器选定的触发源来触发。该模式下 ADC_STR.START 寄存器写 1 不会启动转换，即软件启动无效。

使用协同工作模式时，请禁止序列 B 动作，以免打乱同步。

可以设置 ADC1, ADC2 两个 ADC 模块协同工作，也可以 ADC1, ADC2, ADC3 三个 ADC 模块协同工作。根据产品具体规格，ADC3 可能未搭载。

ADC 可配置成以下四种协同工作模式：

- 单次并行触发模式
- 单次延迟触发模式
- 循环并行触发模式
- 或者循环延迟触发模式

单次并行触发模式

ADC1 的序列 A 触发条件同时触发处于协同工作模式的所有 ADC 模块，且只触发一次。

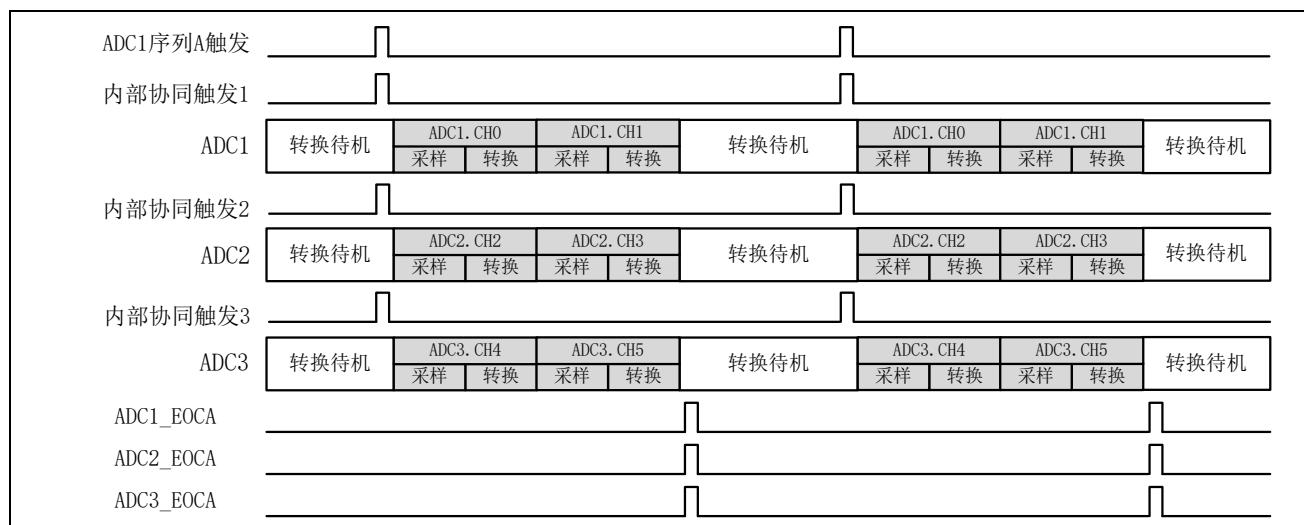


图 16-11 单次并行触发模式 (三 ADC)

注意：

- 禁止多个 ADC 同时对同一个模拟输入进行转换，一个模拟通道同一时间只能给一个 ADC 模块采样，下同。

本模式的软件设定流程如下：

1. 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 0，确认协同工作无效。
2. 设置 ADC1 模块
 - a) 确认 ADC1_STR.STRT 为 0，ADC1 处于转换待机状态。
 - b) 设置控制寄存器 ADC1_CR0.MS[1:0] 为 00b：序列 A 单次扫描模式，或 01b：序列 A 连续扫描模式
 - c) 设置序列 A 通道选择寄存器 ADC1_CHSELRA
 - d) 设置采样时间寄存器 ADC1_SSTR
 - e) 设置序列 A 触发源选择寄存器 ADC1_TRGSR
3. 设置 ADC2 模块
 - a) 确认 ADC2_STR.STRT 为 0，ADC1 处于转换待机状态。
 - b) 设置控制寄存器 ADC2_CR0.MS[1:0]，通道选择寄存器 ADC2_CHSELRA，通道采样时间寄存器 ADC2_SSTR。
- 注意：
 - 为保证 ADC2 与 ADC1 的同步工作，上述寄存器尽量与 ADC1 的寄存器设置相同的值。具体通道无需相同，只要保持通道数，以及对应通道的采样时间一致即可。
4. 设置 ADC3 模块（三 ADC 协同工作时）
 - a) 确认 ADC3_STR.STRT 为 0，ADC2 处于转换待机状态。
 - b) 设置控制寄存器 ADC3_CR0.MS[1:0]，通道选择寄存器 ADC3_CHSELRA，通道采样时间寄存器 ADC3_SSTR。
- 注意：
 - 同 ADC2，为保证 ADC3 与 ADC1 的同步工作，上述寄存器尽量与 ADC1 的寄存器设置相同的值。
5. 设置协同模式控制寄存器 ADC_SYNCCR.SYNCMD[2:0]，写 010b：ADC1, ADC2 两 ADC 协同工作。或者写 011b：ADC1, ADC2, ADC3 三 ADC 协同工作。
6. 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 1，协同工作有效。
7. 等待 ADC1 序列 A 触发源输入，在 ADC1, 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 模块只触发一次。

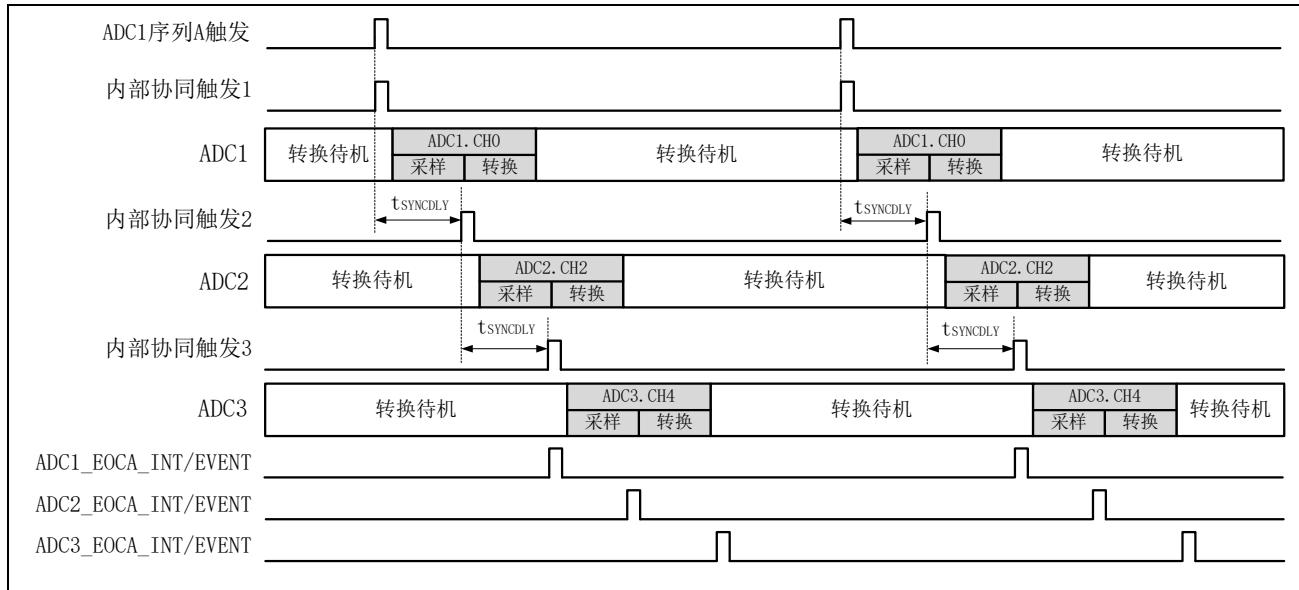


图 16-12 单次延迟触发模式 (三 ADC)

注意:

- ADC1 序列 A 触发第一次输入后, ADC3 协同触发发生前, 再次输入 ADC1 序列 A 触发将被忽略。
- 若各 ADC 单元转换的是同一模拟通道, 需要错开采样时间, 即延迟时间 $t_{SYNCDLY}$ 与通道开采样时间 t_{SPL} 需满足: $t_{SYNCDLY} > t_{SPL}$ 。

本模式的软件设定流程如下:

1. 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 0, 确认协同工作无效。
2. 设置 ADC1, ADC2, ADC3 模块 (参考单次平行模式)
3. 设置协同模式控制寄存器 ADC_SYNCCR.SYNCDLY[7:0], 设置两个 ADC 的启动延迟。
4. 设置协同模式控制寄存器 ADC_SYNCCR.SYNCMD[2:0], 写 000b: ADC1, ADC2 两 ADC 协同工作。或者写 001b: ADC1, ADC2, ADC3 三 ADC 协同工作。
5. 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 1, 协同工作有效。
6. 等待 ADC1 序列 A 触发源输入, 在 ADC1, 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 模块，且之后每经过指定延迟之后会再次同时触发所有 ADC 模块。直至用户主动软件停止 ADC1 模块，或者禁止协同工作模式。

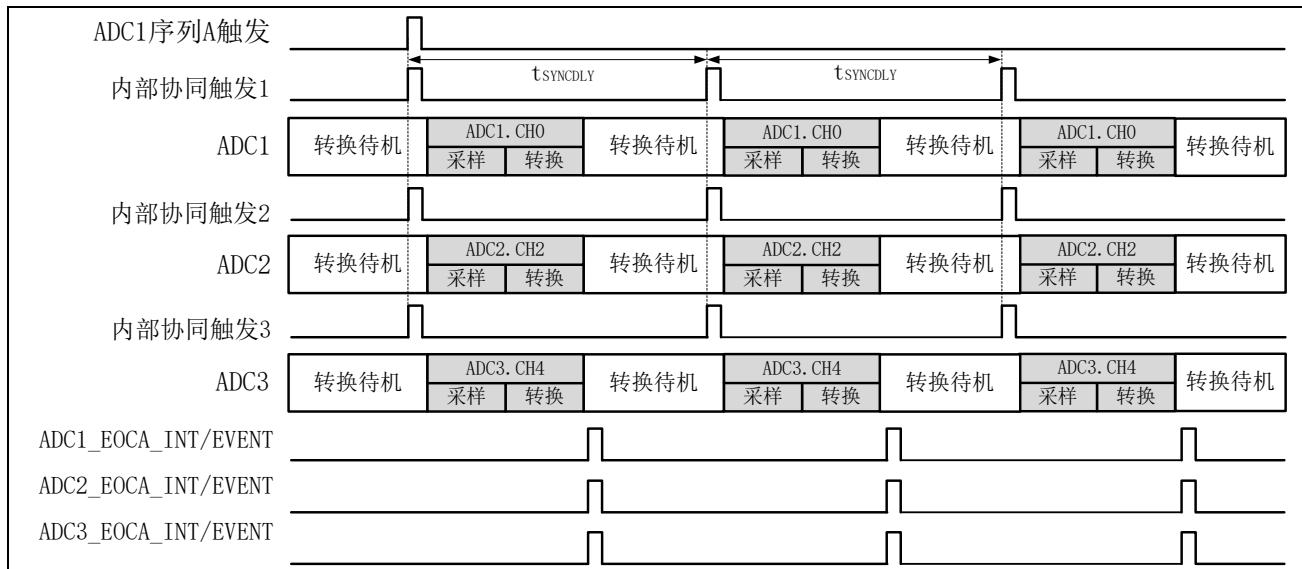


图 16-13 循环并行触发模式 (三 ADC)

注意：

- 延迟时间 tSYNCDLY 与一次扫描转换的时间 tSCAN 需满足：tSYNCDLY>tSCAN。

本模式的软件设定流程如下：

1. 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 0，确认协同工作无效。
2. 设置 ADC1, ADC2, ADC3 模块，参考单次并行模式。ADC_CR0.MS[1:0]设置为 00b: 序列 A 单次扫描模式
3. 设置协同模式控制寄存器 ADC_SYNCCR.SYNCDLY[7:0]，设置每次并行触发的延迟。
4. 设置协同模式控制寄存器 ADC_SYNCCR.SYNCMD[2:0]，写 110b: ADC1, ADC2 两 ADC 协同工作。或者写 111b: ADC1, ADC2, ADC3 三 ADC 协同工作。
5. 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 1，协同工作有效。
6. 等待 ADC1 序列 A 触发源输入，在 ADC1, 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 模块，或者禁止协同工作模式。

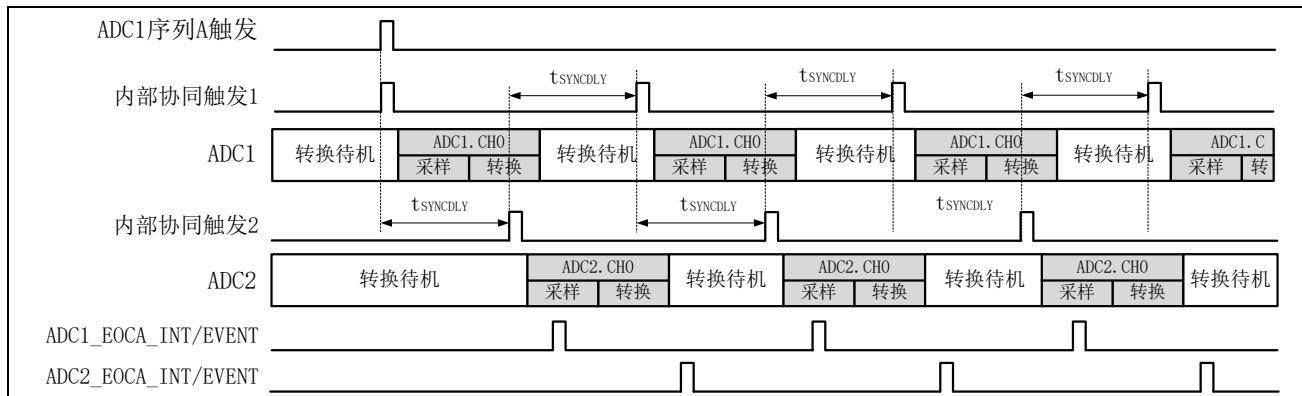


图 16-14 循环延迟触发模式（两 ADC）

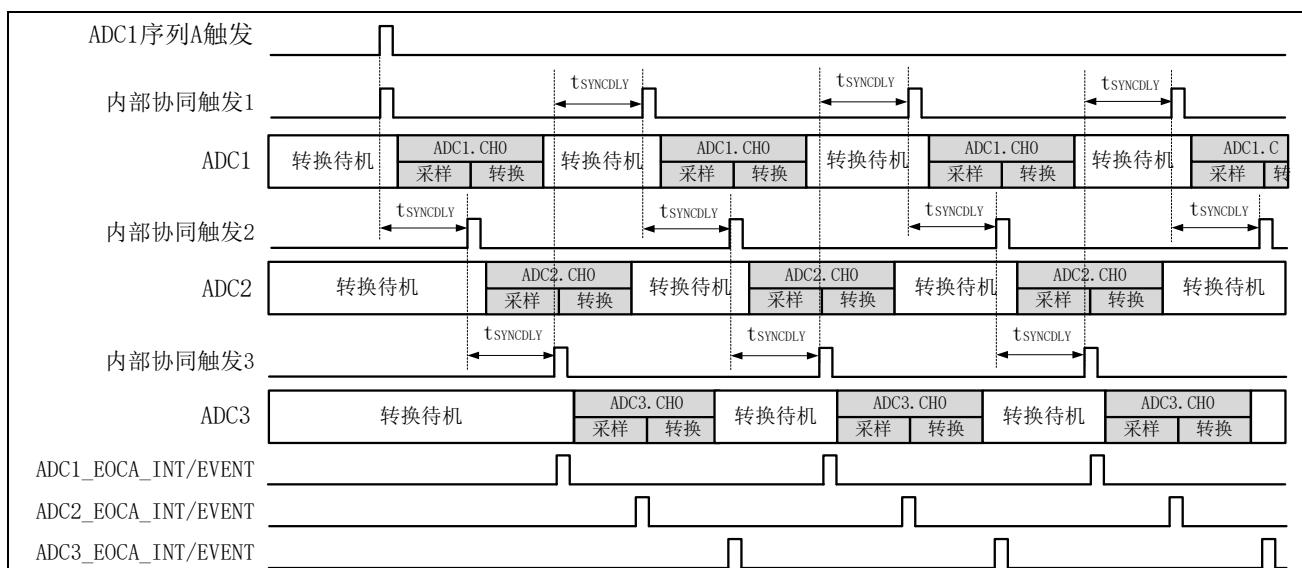


图 16-15 循环延迟触发模式（三 ADC）

注意：

- 两 ADC 协同工作时，延迟时间 $t_{SYNCDLY}$ 与一次扫描转换的时间 t_{SCAN} 需满足：
 $t_{SYNCDLY} > t_{SCAN}/2$ 。三 ADC 协同工作时需满足： $t_{SYNCDLY} > t_{SCAN}/3$ 。同时，如 ADC1, ADC2, ADC3 转换的是同一模拟通道，还需要错开采样时间，即 $t_{SYNCDLY} > t_{SPL}$ 。

本模式的软件设定流程如下：

1. 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 0，确认协同工作无效。
2. 设置 ADC1, ADC2, ADC3 模块，参考循环并行触发模式。

3. 设置协同模式控制寄存器 ADC_SYNCCR.SYNCDLY[7:0], 设置每次触发的延迟。
4. 设置协同模式控制寄存器 ADC_SYNCCR.SYNCMD[2:0], 写 100b: ADC1, ADC2 两 ADC 协同工作。或者写 101b: ADC1, ADC2, ADC3 三 ADC 协同工作。
5. 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 1, 协同工作有效。
6. 等待 ADC1 序列 A 触发源输入, 在 ADC1, ADC2, ADC3 完成转换后处理结果。

16.3.13 中断和事件信号输出

ADC 模块可以产生以下四种事件输出, 每个事件发生时, 若对应的中断许可寄存器设置为有效时, 同时输出中断申请。

1. 序列 A 扫描结束 ADC_EOCA, 对应中断许可寄存器 ADC_ICR.EOCAIEN
2. 序列 B 扫描结束 ADC_EOCB, 对应中断许可寄存器 ADC_ICR.EOCBIEN
3. 通道比较 ADC_CHCMP, 对应中断许可寄存器 ADC_AWDCR.AWDIEN
4. 序列比较 ADC_SEQCMP, 对应中断许可寄存器 ADC_AWDCR.AWDIEN

上述四种事件输出, 可以启动其他片内周边模块, 包括启动 DMA 传送。利用 DMA 传送可以连续读取 A/D 转换结果, 不需要软件干预, 完全由硬件实现, 降低 CPU 的负荷。DMA 的设定请参考 DMA 说明章节。事件信号输出和中断使能位的控制无关, 只要条件发生就会输出。

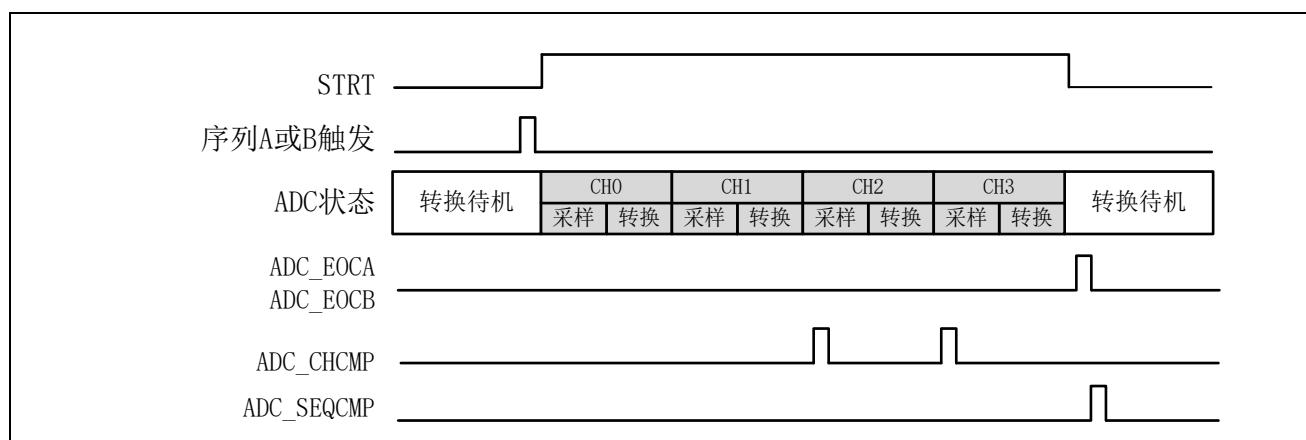


图 16-16 ADC 中断和事件输出时序

上图中, CH1, CH2 看门狗的比较条件满足。通道比较事件在每个通道转换结束时发生, 序列比较事件在序列扫描结束时发生, 比序列扫描事件 ADC_EOCA/B 晚一周期。

16.4 寄存器说明

寄存器一览

单元1 BASE_ADDR: 0x4004_0000

单元2 BASE_ADDR: 0x4004_0400

寄存器名	符号	偏移地址	位宽	复位值
A/D启动寄存器	ADC_STR	0x00	8	0x00
A/D控制寄存器0	ADC_CR0	0x02	16	0x0000
A/D控制寄存器1	ADC_CR1	0x04	16	0x0000
A/D转换开始触发寄存器	ADC_TRGSR	0x0A	16	0x0000
A/D通道选择寄存器A0	ADC_CHSELRA0	0x0C	16	0x0000
A/D通道选择寄存器A1	ADC_CHSELRA1	0x0E	16	0x0000
A/D通道选择寄存器B0	ADC_CHSELRB0	0x10	16	0x0000
A/D通道选择寄存器B1	ADC_CHSELRB1	0x12	16	0x0000
A/D平均通道选择寄存器0	ADC_AVCHSELR0	0x14	16	0x0000
A/D平均通道选择寄存器1	ADC_AVCHSELR1	0x16	16	0x0000
A/D采样周期寄存器	ADC_SSTRx	0x20+x	8	0x0B
	ADC_SSTRL	0x30	8	0x0B
A/D通道映射控制寄存器0	CHMUXR0	0x38	16	0x3210
A/D通道映射控制寄存器1	CHMUXR1	0x3A	16	0x7654
A/D通道映射控制寄存器2	CHMUXR2	0x3C	16	0xBA98
A/D通道映射控制寄存器3	CHMUXR3	0x3E	16	0xFEDC
A/D中断状态寄存器	ADC_ISR	0x46	8	0x00
A/D中断许可寄存器	ADC_ICR	0x47	8	0x03
A/D协同模式控制寄存器	ADC_SYNCCR	0x4C	16	0x0C00
A/D数据寄存器	ADC_DRx	0x50+2*x	16	0x0000
模拟看门狗控制寄存器	ADC_AWDCR	0xA0	16	0x0000
模拟看门狗阀值寄存器	ADC_AWDDR0	0xA4	16	0x0000
	ADC_AWDDR1	0xA6	16	0x0000
模拟看门狗比较通道选择寄存器	ADC_AWDCHSR0	0xAC	16	0x0000
模拟看门狗比较通道选择寄存器1	ADC_AWDCHSR1	0xAE	16	0x0000

模拟看门狗状态寄存器	ADC_AWDSR0	0xB0	16	0x0000
模拟看门狗状态寄存器1	ADC_AWDSR1	0xB2	16	0x0000
A/D可编程增益放大器控制寄存器	ADC_PGACR	0xC0	16	0x0000
A/D可编程增益倍数寄存器	ADC_PGAGSR	0xC2	16	0x0000
A/D可编程增益放大器输入选择寄存器	ADC_PGAINSR0	0xCC	16	0x0000
A/D可编程增益放大器输入选择寄存器1	ADC_PGAINSR1	0xCE	16	0x0000

表 16-5 ADC 寄存器一览 1/2

16.4.1 A/D 启动寄存器 ADC_STR

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	STRT

位	标记	位名	功能	R/W
b7-b1	—	—	读出时为0, 写入时写0	R/W
b0	STRT	AD转换开始	0: 停止转换 1: 开始转换 置“1”条件: (1) 软件设置 (2) 选择的触发条件发生 (3) A/D转换中 清“0”条件: (1) 软件清“0” (2) 转换结束后自动清“0” 注意: - STRT为0 (ADC空闲中) 时写1产生软件触发, 启动序列A - STRT为1 (ADC动作中) 时写1无效。 - STRT为1时写0表示强制停止AD转换。若ADC_TRGSR设置了0x0以外的值且 不希望ADC再启动, 请先将ADC_TRGSR设置为0, 再对STRT写0。 - STRT为0时写0无效。	R/W

16.4.2 A/D 控制寄存器 0 ADC_CR0

复位值: 0x0000

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

位	标记	位名	功能	R/W
b15-b11	—	—	读出时为0, 写入时写0	R/W
b10-b8	AVCNT[2:0]	次数选择	0 0 0: 连续转换2次平均 0 0 1: 连续转换4次平均 0 1 0: 连续转换8次平均 0 1 1: 连续转换16次平均 1 0 0: 连续转换32次平均 1 0 1: 连续转换64次平均 1 1 0: 连续转换128次平均 1 1 1: 连续转换256次平均	R/W
b7	DFMT	数据格式	0: 转换数据右对齐 1: 转换数据左对齐	R/W
b6	CLREN	数据寄存器自动清除	0: 自动清除禁止 1: 自动清除许可 注意: CLREN位设定后, 寄存器ADC_DRx将在CPU、DMA等读取后被自动清除。自动清除功能主要用于检测数据寄存器是否更新。	R/W
b5-b4	ACCSEL[1:0]	分辨率选择	0 0: 12位分辨率 0 1: 10位分辨率 1 0: 8位分辨率 1 1: 设定禁止	R/W
b3-b2	—	—	读出时为0, 写入时写0	R/W
b1-b0	MS[1:0]	模式选择	0 0: 序列A单次扫描模式, 序列B无效 0 1: 序列A连续扫描模式, 序列B无效 1 0: 序列A单次扫描模式, 序列B单次扫描模式 1 1: 序列A连续扫描模式, 序列B单次扫描模式	R/W

注意:

- 请在 ADC_STR.START 为 “0” 时设置本寄存器。

16.4.3 A/D 控制寄存器 1 ADC_CR1

复位值: 0x0000

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

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

注意:

- 请在 ADC_STR.START 为 “0” 时设置本寄存器。

16.4.4 A/D 转换开始触发寄存器 ADC_TRGSR

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
TRGENB	—	—	—	—	TRGSELB[2]	TRGSELB[1]	TRGSELB[0]
b7	b6	b5	b4	b3	b2	b1	b0
TRGENA	—	—	—	—	TRGSEL A[2]	TRGSEL A[1]	TRGSEL A[0]

位	标记	位名	功能	R/W
b15	TRGENB	序列B触发使能	0: 序列B片内或外部引脚触发禁止 1: 序列B片内或外部引脚触发许可 注意: 选择外部引脚触发有效。如果ADTRGx由"High"变为"Low", 检测到下降沿, 则扫描转换开始, 请保持"Low"1.5*PCLK4周期以上。	R/W
b14-b11	—	—	读出时为0, 写入时写0	R/W
b10-b8	TRGSELB[2:0]	序列B触发条件选择	在序列B有效模式下(ADC_CR0.MS[1]=1), 作为序列B的触发条件 000b: ADTRGx 001b: IN_TRGx0 010b: IN_TRGx1 011b: IN_TRGx0 + IN_TRGx1 其它: 未选择 x=1、2 注意: 只在序列B有效模式下有效。其他模式设定无效。 两次触发的间隔必须大于或等于扫描周期tSCAN, 若小于则触发无效。	R/W
b7	TRGENA	序列A触发使能	0: 序列A片内或外部引脚触发禁止 1: 序列A片内或外部引脚触发许可 注意: 选择外部引脚触发有效。如果ADTRGx由"High"变为"Low", 检测到下降沿, 则扫描转换开始, 请保持"Low"1.5*PCLK4周期以上。	R/W
b6-b3	—	—	读出时为0, 写入时写0	R/W
b2-b0	TRGSEL A[2:0]	序列A触发条件选择	序列A的触发条件。 000b: ADTRGx 001b: IN_TRGx0 010b: IN_TRGx1 011b: IN_TRGx0 + IN_TRGx1 其它: 未选择 n=1、2 注意: ADC_STR.START写1软件触发, 无视TRGENA, 或者TRGSEL A[2:0]的设定, 直接开始A/D转换。 两次触发的间隔必须大于或等于扫描周期tSCAN, 若小于则触发无效。	R/W

注意:

- 在 ADC_STR.START 为“0”时设置本寄存器。

16.4.5 A/D 通道选择寄存器 A ADC_CHSELRA0

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
CHSEL[A15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
CHSEL[A7:0]							

位	标记	位名	功能	R/W
b15-b0	CHSEL[A15:0]	转换通道选择	序列A的通道选择，每一位代表一个通道，CHSEL[x]代表通道CHx，可选择任意组合。 0: 未选择对应通道 1: 选择对应通道 不存在通道的对应位为reserved位，读出时为0，写入时写0。 注意：请不要再序列A和序列B中选择相同的通道。	R/W

注意：

- 请在 ADC_STR STRT 为“0”时设置本寄存器。

16.4.6 A/D 通道选择寄存器 A1 ADC_CHSELRA1

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
CHSEL[A31:24]							
b7	b6	b5	b4	b3	b2	b1	b0
CHSEL[A23:16]							

位	标记	位名	功能	R/W
b15-b0	CHSEL[A31:16]	转换通道选择	序列A的通道选择，每一位代表一个通道，CHSEL[x]代表通道CHx，可选择任意组合。 0: 未选择对应通道 1: 选择对应通道 不存在通道的对应位为reserved位，读出时为0，写入时写0。 注意：请不要再序列A和序列B中选择相同的通道。	R/W

注意：

- 请在 ADC_STR STRT 为“0”时设置本寄存器。

16.4.7 A/D 通道选择寄存器 B ADC_CHSELRB0

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
CHSELB[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
CHSELB[7:0]							

位	标记	位名	功能	R/W
b15-b0	CHSELB[15:0]	转换通道选择	序列B的通道选择，每一位代表一个通道，CHSELB[x]代表通道CHx，可选择任意组合。只有在双序列扫描模式是才有效。 0: 未选择对应通道 1: 选择对应通道 不存在通道的对应位为reserved位，读出时为0，写入时写0。 注意：请不要再序列A和序列B中选择相同的通道。	R/W

注意：

- 请在 ADC_STR STRT 为“0”时设置本寄存器。

16.4.8 A/D 通道选择寄存器 B1 ADC_CHSELRB1

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
CHSELB[31:24]							
b7	b6	b5	b4	b3	b2	b1	b0
CHSELB[23:16]							

位	标记	位名	功能	R/W
b15-b0	CHSELB[31:16]	转换通道选择	序列B的通道选择，每一位代表一个通道，CHSELB[x]代表通道CHx，可选择任意组合。只有在双序列扫描模式是才有效。 0: 未选择对应通道 1: 选择对应通道 不存在通道的对应位为reserved位，读出时为0，写入时写0。 注意：请不要再序列A和序列B中选择相同的通道。	R/W

注意：

- 请在 ADC_STR STRT 为“0”时设置本寄存器。

16.4.9 A/D 平均通道选择寄存器 ADC_AVCHSELR0

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
AVCHSEL[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
AVCHSEL[7:0]							

位	标记	位名	功能	R/W
b15-b0	AVCHSEL[15:0]	平均通道选择	每一位代表一个通道，AVCHSEL[x]代表通道CHx，可选择任意组合。 0: 未选择对应通道 1: 选择对应通道 不存在通道的对应位为reserved位，读出时为0，写入时写0。 注意：当AVCHSEL与ADC_CHSELRA或ADC_CHSELB的对应的通道同时被选择时， 则该通道在扫描时将连续执行设定次数A/D转换，并对转换结果进行平均计算后更新入 数据寄存器。如果对应通道AVCHSEL未被设定，该通道将执行普通一次转换。	R/W

注意：

- 请在 ADC_STR.STRT 为“0”时设置本寄存器。

16.4.10 A/D 平均通道选择寄存器 1 ADC_AVCHSELR1

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
AVCHSEL[31:24]							
b7	b6	b5	b4	b3	b2	b1	b0
AVCHSEL[23:16]							

位	标记	位名	功能	R/W
b15-b0	ADAVSEL[31:16]	平均通道选择	每一位代表一个通道，AVCHSEL[x]代表通道CHx，可选择任意组合。 0: 未选择对应通道 1: 选择对应通道 不存在通道的对应位为reserved位，读出时为0，写入时写0。 注意：当AVCHSEL与ADC_CHSELRA或ADC_CHSELB的对应的通道同时被选择时， 则该通道在扫描时将连续执行设定次数A/D转换，并对转换结果进行平均计算后更新入 数据寄存器。如果对应通道AVCHSEL未被设定，该通道将执行普通一次转换。	R/W

注意：

- 请在 ADC_STR.STRT 为“0”时设置本寄存器。

16.4.11 A/D 采样状态寄存器 ADC_SSTR

复位值: 0x0B

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

位	标记	位名	功能	R/W
b7-b0	SST[7:0]	采样周期数	<p>采样周期数可被设为5到255个周期。 通道CH0~15由ADC_SSTRx, x=0~15来设置, 其他通道由ADC_SSTRL来设置。 注意: ADCLK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RAIN太大采样时间不足或者ADCLK频率低时可以设置寄存器进行调整采样时间。采样时间不要少于5周期。</p> $SST \geq (R_{AIN} + R_{ADC}) * C_{ADC} * \ln(2^{N+2}) * f_{ADC} + 1$ <p>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ADCLK频率 (Hz)。具体参考电器特性相关说明。</p>	R/W

注意:

- 请在 ADC_STR.START 为“0”时设置本寄存器。

16.4.12 A/D 通道映射控制寄存器 ADC_CHMUXR

ADC_CHMUXR0 复位值: 0x3210

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH03MUX[3:0]				CH02MUX[3:0]				CH01MUX[3:0]				CH00MUX[3:0]			

ADC_CHMUXR1 复位值: 0x7654

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH07MUX[3:0]				CH06MUX[3:0]				CH05MUX[3:0]				CH04MUX[3:0]			

ADC_CHMUXR2 复位值: 0xBA98

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH11MUX[3:0]				CH10MUX[3:0]				CH09MUX[3:0]				CH08MUX[3:0]			

ADC_CHMUXR3 复位值: 0xFEDC

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH15MUX[3:0]				CH14MUX[3:0]				CH13MUX[3:0]				CH12MUX[3:0]			

位	标记	位名	功能	R/W
CHxMUX[3:0]	通道x映射选择	不存在的通道对应位读出时为0, 写入时写0		R/W
x=0~15		对不同的ADC单元CHx的映射关系如下:		
		设定值	ADC1 映射对象	ADC2 映射对象
		0x0	ADC1_IN0	ADC12_IN4
		0x1	ADC1_IN1	ADC12_IN5
		0x2	ADC1_IN2	ADC12_IN6
		0x3	ADC1_IN3	ADC12_IN7
		0x4	ADC12_IN4	ADC12_IN8
		0x5	ADC12_IN5	ADC12_IN9
		0x6	ADC12_IN6	ADC12_IN10
		0x7	ADC12_IN7	ADC12_IN11
		0x8	ADC12_IN8	内部模拟通道
		0x9	ADC12_IN9	—
		0xa	ADC12_IN10	—
		0xb	ADC12_IN11	—
		0xc	ADC1_IN12	—
		0xd	ADC1_IN13	—
		0xe	ADC1_IN14	—
		0xf	ADC1_IN15	—

注意: 请不要设置到不存在的模拟输入上。

注意:

- 请在 ADC_STR.START 为“0”时设置本寄存器。

16.4.13 A/D 中断状态寄存器 ADC_ISR

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	EOCBF	EOCAF

位	标记	位名	功能	R/W
b7-b2	—	—	读出时为0, 写入时写0	R/W
b1	EOCBF	序列B转换完成	序列B所选通道全部扫描完成后置1 标志 当寄存器置位, 需要清除寄存器时, 请读“1”后写“0”。	R/W
b0	EOCAF	序列A转换完	序列A所选通道全部扫描完成后置1 成标志 当寄存器置位, 需要清除寄存器时, 请读“1”后写“0”。	R/W

16.4.14 A/D 中断许可寄存器 ADC_ICR

复位值: 0x03

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	EOCBIEN	EOCAIEN

位	标记	位名	功能	R/W
b7-b2	—	—	读出时为0, 写入时写0	R/W
b1	EOCBIEN	序列B转换完成	0: 序列B转换完成中断禁止 中断使能 1: 序列B转换完成中断许可	R/W
b0	EOCAIEN	序列A转换完	0: 序列A转换完成中断禁止 成中断使能 1: 序列A转换完成中断许可	R/W

16.4.15 A/D 协同模式控制寄存器 ADC_SYNCCR

复位值: 0x0C00

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

位	标记	位名	功能	R/W
b15-b8	SYNCDLY[7:0]	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} 。 0x1表示 t _{SYNCDLY} = 1*PCLK4, 0xff表示t _{SYNCDLY} = 255*PCLK4 注意：在SYNCEN为“0”时设置本寄存器。请不要写入0x00。 根据各ADC的采样时间，转换时间，设置合理的延迟时间，避免多个ADC同时处于采样状态引起的误差增大，避免ADC还未转换结束就再次发生触发，导致同步失败。推荐设置如下： 单次延迟触发模式： t _{SYNCDLY} > t _{SPL} 两ADC循环延迟触发模式： t _{SYNCDLY} > t _{SPL} , 且 t _{SYNCDLY} > t _{SCAN} /2 三ADC循环延迟触发模式： t _{SYNCDLY} > t _{SPL} , 且 t _{SYNCDLY} > t _{SCAN} /3 单次并行触发模式：本寄存器设置无效。 循环并行触发模式： t _{SYNCDLY} > t _{SCAN}	R/W
b7	—	—	读出时为0, 写入时写0	R/W
b6-b4	SYNCMD[2:0]	同步模式选择	SYNCMD[2] 0: 单次触发 1: 循环触发 SYNCMD[1] 0: 延迟触发模式 1: 并行触发模式 SYNCMD[0] 0: ADC1和ADC2同步工作, ADC3独立工作 1: ADC1, ADC2和ADC3同步工作, 如果无ADC3, 该bit禁止设为1。 注意：在SYNCEN为“0”时设置本寄存器。使用单次触发时，请将需要同步的ADC设置为序列A单次扫描，或序列A连续扫描模式。使用循环触发模式时，请将ADC设置为序列A单次扫描模式。	R/W
b3-b1	—	—	读出时为0, 写入时写0	R/W
b0	SYNCEN	同步模式许可	0: 同步模式无效 1: 同步模式有效 注意： 同步模式只支持序列A。在SYNCEN写1之前，请将参与同步的几个ADC的序列B关闭(ADC_CR0.MS[1]=0)，并给序列A选择相同数目的通道，设置相同的通道采样时间ADC_SSTRx。以避免各ADC扫描时间t _{SCAN} 不一致，引起后续同步失败。 软件对ADC1_STR.START写0强制停止转换时，SYNCEN自动清0。	R/W

注意：

- 本寄存器只在 ADC1 中搭载，ADC2 中无此寄存器。

16.4.16 A/D 数据寄存器 ADC_DR

ADC_DRx (ADC1 x=0~16, ADC2 x=0~8) 通道 x 数据寄存器

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

根据数据对齐方式和转换分辨率，转换结果数据存储方式有所不同。

数据右对齐-12 位分辨率

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

数据右对齐-10 位分辨率

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0										AD[9:0]

数据右对齐-8 位分辨率

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

数据左对齐-12 位分辨率

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

数据左对齐-10 位分辨率

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
															AD[9:0]

数据左对齐-8 位分辨率

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

16.4.17 模拟看门狗控制寄存器 ADC_AWDCR

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
—	—	—	—	—	—	—	AWDIEN
b7	b6	b5	b4	b3	b2	b1	b0
AWDSS[1]	AWDSS[0]	—	AWDMD	—	—	—	AWDEN

位	标记	位名	功能	R/W
b15-b9	—	—	读出时为0, 写入时写0	R/W
b8	AWDIEN	看门狗中断使能	0: 禁止ADC_CHCMP, ADC_SEQCMP中断 1: 允许ADC_CHCMP, ADC_SEQCMP中断 注意: 本寄存器不影响ADC_CHCMP, ADC_SEQCMP事件输出	R/W
b7-b6	AWDSS[1:0]	看门狗序列选择	00: 序列A, 序列B扫描完成时都输出ADC_SEQCMP中断/事件 01: 序列A扫描完成时输出ADC_SEQCMP中断/事件, 序列B不输出 10: 序列B扫描完成时输出ADC_SEQCMP中断/事件, 序列A不输出 11: 同00 注意: 通道看门狗中断/事件ADC_CHCMP不受本寄存器控制, 在每个通道转换结束时根据比较结果正常输出。 ADC_AWDSR各通道比较状态寄存器的置1不受本寄存器控制。	R/W
b5	—	—	读出时为0, 写入时写0	R/W
b4	AWDMD	看门狗比较模式	0: AWDDR0>转换结果, 或转换结果>AWDDR1时, 比较条件满足 1: AWDDR0≤转换结果≤AWDDR1时, 比较条件满足 比较条件满足时, 输出ADC_CHCMP事件, 若中断允许AWDIEN=1则同时输出中断。序 列扫描完成时, 若本序列中存在一个或多个通道满足比较条件, 且AWDSS[1:0]允许本序 列, 则输出ADC_SEQCMP事件, 若中断允许AWDIEN=1则同时输出中断。	R/W
b3-b1	—	—	读出时为0, 写入时写0	R/W
b0	AWDEN	看门狗比较功能使能	0: 看门狗比较功能无效 1: 看门狗比较功能有效	R/W

16.4.18 模拟看门狗阀值寄存器 ADC_AWDDR0、ADC_AWDDR1

复位值：ADC_AWDDR0=0x0000, ADC_AWDDR1=0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8
AWDDR[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
AWDDR[7:0]							

位	标记	位名	功能	R/W
b15-b0	ADDR[15:0]	比较数据	比较数据	R/W

AWDDR0 设置低阀值，AWDDR1 设置高阀值，在 A/D 转换中也可将数据写入寄存器，实现动态的阀值比较功能。

AWDDR0、AWDDR1 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

当多次平均功能有效，只对最后的平均值进行比较。

16.4.19 模拟看门狗比较通道选择寄存器 ADC_AWDCHSR0

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
AWDCH[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
AWDCH[7:0]							

位	标记	位名	功能	R/W
b15-b0	AWDCH[15:0]	看门狗比较功能通道选择	0: 该通道看门狗比较功能非选择 1: 该通道看门狗比较功能选择 每一位各对应一个通道。不存在通道的对应位为reserved位，读出时为0，写入时写0。 注意：只有在序列A或B扫描中选择了相应的通道，即通道选择寄存器ADC_CHSELRA或ADC_CHSELRB的对应位为“1”时才有效。	R/W

注意：

- 请在 ADC_STR.STRT 为“0”，且 ADC_AWDCR.AWDEN 为 “0” 时设置本寄存器。

16.4.20 模拟看门狗比较通道选择寄存器 1 ADC_AWDCHSR1

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
AWDCH[31:24]							
b7	b6	b5	b4	b3	b2	b1	b0
AWDCH[23:16]							

位	标记	位名	功能	R/W
b15-b0	AWDCH[31:16]	看门狗比较功能通道选择	0: 该通道看门狗比较功能非选择 1: 该通道看门狗比较功能选择 每一位各对应一个通道。不存在通道的对应位为reserved位，读出时为0，写入时写0。 注意：只有在序列A或B扫描中选择了相应的通道，即通道选择寄存器ADC_CHSELRA或ADC_CHSELRB的对应位为“1”时才有效。	R/W

注意：

- 请在 ADC_STR.STRT 为“0”，且 ADC_AWDCR.AWDEN 为 “0” 时设置本寄存器。

16.4.21 模拟看门狗状态寄存器 ADC_AWDSR0

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
AWDF[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
AWDF[7:0]							

位	标记	位名	功能	R/W
b15-b0	AWDF[15:0]	看门狗比较状态位	0: 比较条件不成立 1: 比较条件成立 注意: 每一位各对应一个通道。不存在通道的对应位为reserved位，读出时为0，写入时写0。 当寄存器置位，需要清除寄存器时，请读“1”后写“0”。	R/W

16.4.22 模拟看门狗状态寄存器 1 ADC_AWDSR1

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
AWDF[31:24]							
b7	b6	b5	b4	b3	b2	b1	b0
AWDF[23:16]							

位	标记	位名	功能	R/W
b15-b0	AWDF[31:16]	看门狗比较状态位	0: 比较条件不成立 1: 比较条件成立 注意: 每一位各对应一个通道。不存在通道的对应位为reserved位，读出时为0，写入时写0。 当寄存器置位，需要清除寄存器时，请读“1”后写“0”。	R/W

16.4.23 A/D 可编程增益放大器控制寄存器 ADC_PGACR

复位值: 0x0000

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

位	标记	位名	功能	R/W
b15-b12	—	—	读出时为0, 写入时写0	R/W
b3~b0	PGACTL[3:0]	放大器控制	0000: 放大器无效 1110: 放大器有效, 信号按ADC_PGAGSR.GAIN[3:0]设定值放大 注意: 禁止设定上述以外的值。	R/W

16.4.24 A/D 可编程增益倍数寄存器 ADC_PGAGSR

复位值: 0x0000

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

位	标记	位名	功能	R/W
b15-b4	—	—	读出时为0, 写入时写0	R/W
b3-b0	GAIN[3:0]	放大器增益设定	0 0 0 0: × 2.000 0 0 0 1: × 2.133 0 0 1 0: × 2.286 0 0 1 1: × 2.667 0 1 0 0: × 2.909 0 1 0 1: × 3.2 0 1 1 0: × 3.556 0 1 1 1: × 4.000 1 0 0 0: × 4.571 1 0 0 1: × 5.333 1 0 1 0: × 6.4 1 0 1 1: × 8 1 1 0 0: × 10.667 1 1 0 1: × 16 1 1 1 0: × 32 注意: 其他值禁止设定。	R/W

16.4.25 A/D 可编程增益放大器输入选择寄存器 ADC_PGAISNR0

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	
—	—	—	—	—	—	—	—	PGAINSEL[8]
b7	b6	b5	b4	b3	b2	b1	b0	
PGAINSEL[7:0]								

位	标记	位名	功能	R/W
b15-b9	—	—	读出时为0, 写入时写0	R/W
b8~b0	PGAINSEL[8:0]	放大器模拟输入选择	0x000: 输入未选择 0x001: ADC1_IN0 0x002: ADC1_IN1 0x004: ADC1_IN2 0x008: ADC1_IN3 0x010: ADC12_IN4 0x020: ADC12_IN5 0x040: ADC12_IN6 0x080: ADC12_IN7 0x100: 内部模拟通道 (8bitDAC_1输出) 其他: 禁止设置	R/W

16.4.26 A/D 可编程增益放大器输入选择寄存器 1 ADC_PGAISNR1

复位值: 0x0000

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

位	标记	位名	功能	R/W
b15-b1	—	—	读出时为0, 写入时写0	R/W
b0	PGAVSSEN	放大器地切除控制选择	0: 使用外部端口PGAVSS作为PGA负相输入 1: 使用内部的模拟地AVSS作为PGA负相输入	R/W

16.5 使用注意事项

16.5.1 数据寄存器读取时注意事项

A/D 数据寄存器 ADC_DR 请以半字单位访问。请不要以字节单位访问数据寄存器。

16.5.2 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。

16.5.3 模块停止与低功耗设定的注意事项

通过设定寄存器 PWC_FCG3，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 A/D 模块工作时请先设定 PWC_FCG3 寄存器相应位取消停止，并等待 1us 以后再启动 A/D 转换。

在设置模块停止前，请先确认 A/D 处于转换停止中，即 ADC_STR.START 位为“0”。

在设置系统进入停止模式（STOP）前，请先将 ADC 设定为模块停止模式。

具体请参考低功耗说明章节。

16.5.4 A/D 转换模拟通道输入的引脚设定

当芯片引脚设定为 A/D 模拟通道输入时，请先禁用对应引脚的数字功能(PCRxy.DDIS)。

参考 GPIO 章节。

16.5.5 噪声控制

为防止浪涌等异常电压破坏模拟输入引脚，推荐使用数据手册中电气特性章节所示保护电路。

17 温度传感器（OTS）

17.1 简介

温度传感器（On-chip Temperature Sensor，以下简称 OTS）可以获取芯片内部的温度，以支持系统的可靠性操作。OTS 提供了一组与温度相关的数字量，通过计算可以得到温度值。不使用时可以通过模块停止功能将其关闭以减少系统功耗。

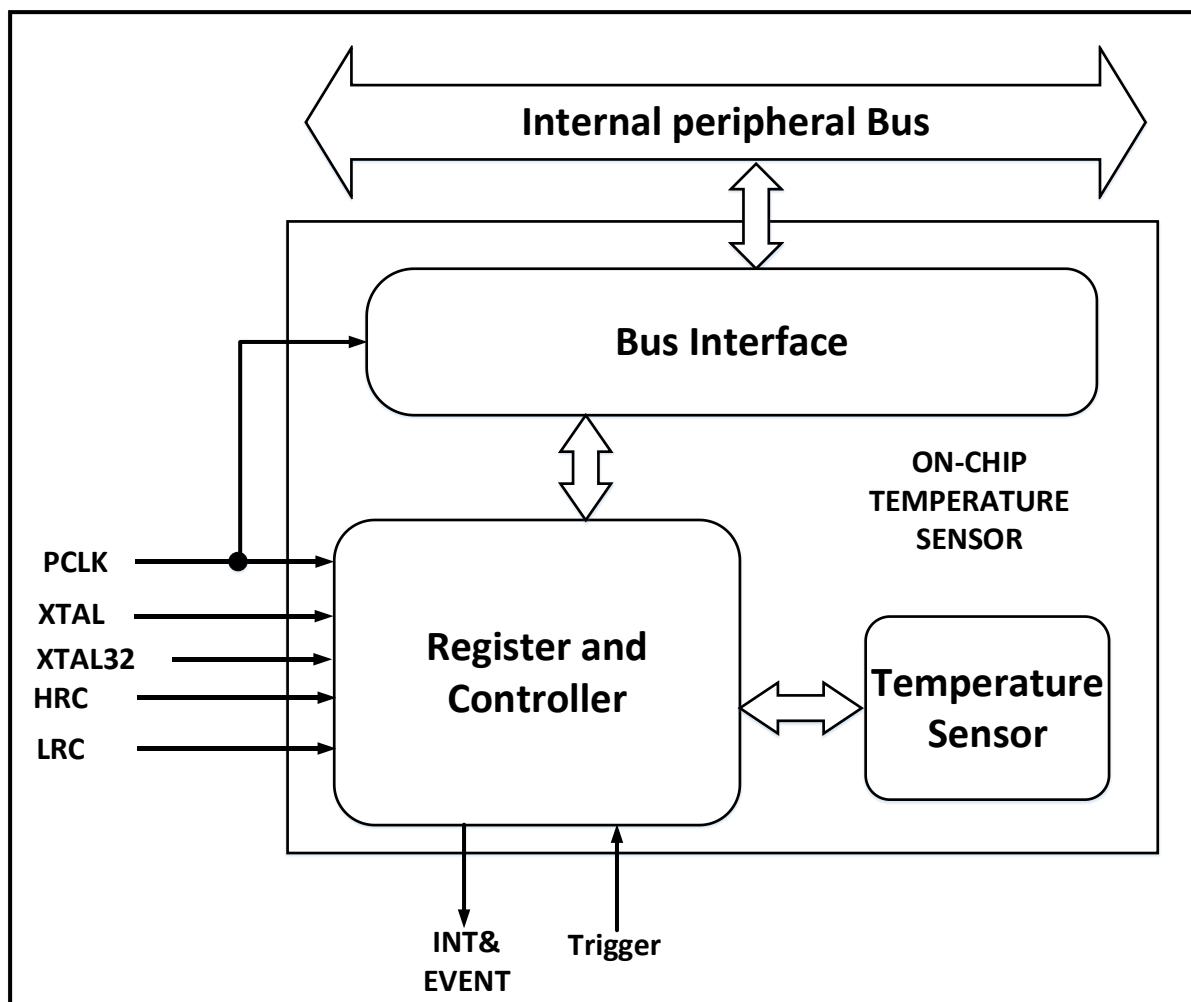


图 17-1 OTS 功能框图

17.2 使用说明

在使用 OTS 获取芯片内部温度之前,请先关闭模块停止功能并启动内部低速时钟 LRC。同时,请根据使用情况选择启动内部高速时钟 HRC、外部高速时钟 XTAL 以及外部低速时钟 XTAL32。

将寄存器 OTS_CTL.OTSST 设置成“1”后测温开始,测温完成后 OTSST 位会自动清零。

因此,确认 OTSST 为“0”后,读取寄存器 OTS_DR1,2 以及 OTS_ECR 中的温度参数,利用以下计算公式求得温度值。

$$T = K \times (1 / D1 - 1 / D2) \times E_{hrc} + M$$

【参数说明】

T: 温度 (°C)

K: 温度斜率 (通过定标实验确定)

D1: 温度参数 1 (从寄存器 OTS_DR1 读取)

D2: 温度参数 2 (从寄存器 OTS_DR2 读取)

E_{hrc}: HRC 频率误差补偿量 (从寄存器 OTS_ECR 读取)

M: 温度偏移量 (通过定标实验确定)

【定标实验】

在两个确定的温度下进行定标实验,计算 K 和 M。

$$K = (T2 - T1) / (A2 - A1)$$

$$M = T1 - K \times A1 = T2 - K \times A2$$

T1: 实验温度 1

T2: 实验温度 2

$$A1: (1 / D1_{T1} - 1 / D2_{T1}) \times E_{hrcT1}$$

D1_{T1}, D1_{T1}, E_{hrcT1} 分别是在温度 T1 下测到的 D1, D2, E_{hrc};

$$A2: (1 / D1_{T2} - 1 / D2_{T2}) \times E_{hrcT2}$$

D1_{T2}, D1_{T2}, E_{hrcT2} 分别是在温度 T2 下测到的 D1, D2, E_{hrc};

寄存器 OTS_CTL.OTSCK 用来选择测温时钟。选择 HRC 动作时，其频率误差可能会影响到最终计算出的温度的精度。为了消除这一误差，请在测温前启动 XTAL32 并在计算温度时使用 Ehrc。选择 XTAL 动作时钟时请将 Ehrc 示为常量 1。

寄存器 OTS_CTL.TSSTP 用来选择测温完成后是否关闭模拟温度传感器。TSSTP 初始值为“0”，表示在一次测温完成后将模拟温度传感器置于开启状态，这样在下一次测温时将自动跳过模拟温度传感器由关闭到开启时的稳定时间。如需在每次测温后关闭模拟温度传感器，请将 TSSTP 设置成“1”。

可以通过其他外设事件触发开始测温，使用时请将触发源的触发目标设置成 OTS。测温完成时也会产生触发其他外设启动的事件，使用时请设定寄存器 OTS_TRG 选择触发目标。使用测温完成中断时请将寄存器 OTS_CTL.OTSIE 设置为“1”。

17.3 寄存器说明

基准地址: 0x4004_A400

寄存器名	符号	偏移地址	位宽	复位值
OTS控制寄存器	OTS_CTL	0x00	16	0x0000
OTS数据寄存器1	OTS_DR1	0x02	16	0x0000
OTS数据寄存器2	OTS_DR2	0x04	16	0x0000
OTS误差补偿寄存器	OTS_ECR	0x06	16	0x0000

表 17-1 OTS 寄存器一览表

17.3.1 OTS 控制寄存器 (OTS_CTL)

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	TSST P	OTS IE	OTS CK	OTS ST

复位值: 0x0000

位	标记	位名	功能	读写
b15~b4	Reserved	-	读时读出0, 写时请写0	R
b3	TSSTP	关闭模拟温度传感 器	选择测温结束后是否自动关闭模拟温度传感器 0: 不关闭模拟温度传感器 1: 关闭模拟温度传感器	R/W
b2	OTSIE	中断使能位	0: 禁止测温结束中断请求 1: 允许测温结束中断请求	R/W
b1	OTSCK	时钟选择位	0: 选择外部高速时钟 (XTAL) 动作 1: 选择内部高速时钟 (HRC) 动作	R/W
b0	OTSST	测温开始位	0: 停止测温 1: 开始测温 置“1”条件: (1) 软件置“1” (2) 硬件触发置“1” 清“0”条件: (1) 软件清“0” (2) 测温结束后自动清“0”	R/W

17.3.2 OTS 数据寄存器 1 (OTS_DR1)

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

复位值: 0x0000

位	标记	位名	功能	读写
b15~b0	TSDC[15:0]	温度数据1	温度数据D1	R
测温完成后自动更新。请确认OTS_CTL.OTSST为“0”后再读取。				

17.3.3 OTS 数据寄存器 2 (OTS_DR2)

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

复位值: 0x0000

位	标记	位名	功能	读写
b15~b0	TSDC[15:0]	温度数据2	温度数据D2	R
测温完成后自动更新。请确认OTS_CTL.OTSST为“0”后再读取。				

17.3.4 OTS 误差补偿寄存器 (OTS_ECR)

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

复位值: 0x0000

位	标记	位名	功能	读写
b15~b0	TSEC[15:0]	误差系数	误差系数Ehrc	R
测温完成后自动更新。请确认OTS_CTL.OTSST为“0”后再读取。				

18 高级控制定时器 (Timer6)

18.1 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，可用于计数产生不同形式的时钟波形，输出以供外部使用。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；单元间可实现软件同步计数和硬件同步计数；各基准值寄存器支持缓存功能；支持 2 相正交编码和 3 相正交编码；支持 EMB 控制。本系列产品中搭载 3 个单元的 Timer6。

18.2 基本框图

Timer6 基本的功能及特性如表 18-1 所示。

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
	计数比较匹配中断
中断输出	计数周期匹配中断
	死区时间错误中断
	事件输出
事件输出	计数比较匹配事件
	计数周期匹配事件

表 18-1 Timer6 的基本功能及特性

Timer6 的基本框图如图 18-1 所示。框图中所示” $<t>$ ”表示单元编号，即” $<t>$ ”为 1~3，本章节后文提到 “ $<t>$ ” 时均指单元编号，不再赘述。

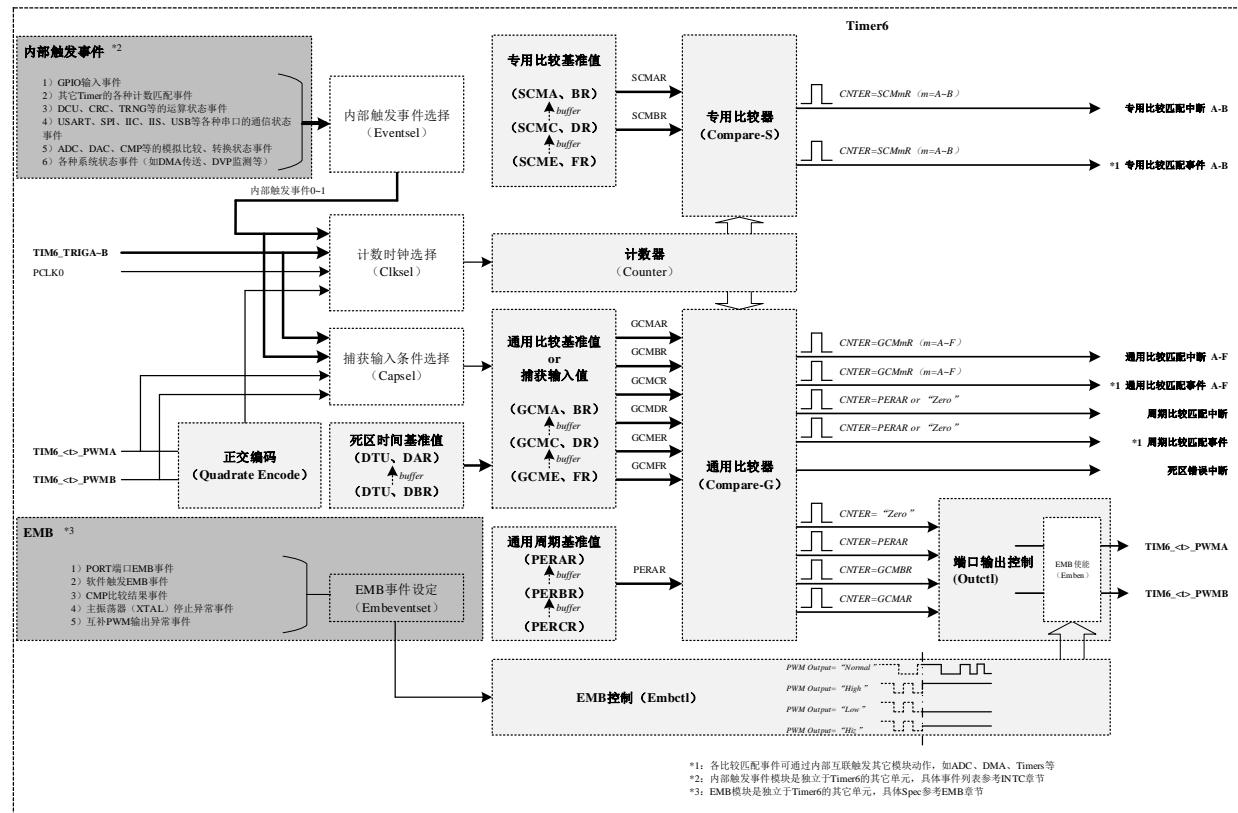


图 18-1 Timer6 基本框图

表 18-2 所示为 Timer6 的输入输出端口列表。

端口名	方向	功能
TIM6_<t>_PWMA	in or out	1) 正交编码计数时钟输入端口或捕获输入端口 或比较输出端口 2) 硬件启动、停止、清零条件输入端口
TIM6_<t>_PWMB		
TIM6_TRIGA	in	1) 硬件计数时钟输入端口或捕获输入端口 2) 硬件启动、停止、清零条件输入端口
TIM6_TRIGB		

表 18-2 Timer6 端口列表

18.3 功能说明

18.3.1 基本动作

18.3.1.1 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。三角波形模式又由于不同的内部计数动作，分为三角波 A 模式、三角波 B 模式。锯齿波和三角波的基本波形如图 18-2 图 18-3 所示。

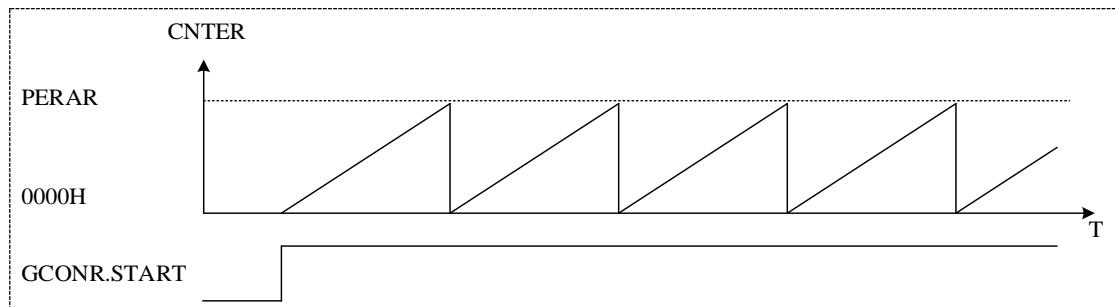


图 18-2 锯齿波波形 (递加计数)

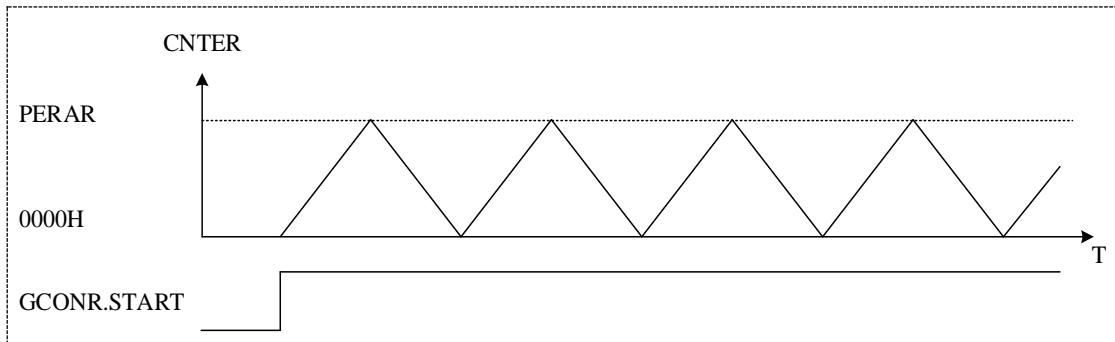


图 18-3 三角波波形

18.3.1.2 比较输出

每个单元的 Timer6 有 2 个比较输出端口 (TIM6_<t>_PWMA、TIM6_<t>_PWMB)，可在计数值与比较基准值比较匹配时输出指定的电平。GCMAR、GCMBR 寄存器分别对应了 TIM6_<t>_PWMA、TIM6_<t>_PWMB 的计数比较基准值。当定时器的计数值和 GCMAR 相等时，TIM6_<t>_PWMA 端口输出指定的电平；当定时器的计数值和 GCMBR 相等时，TIM6_<t>_PWMB 端口输出指定电平。

TIM6_<t>_PWMA、TIM6_<t>_PWMB 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 (PCONR) 的 PCONR.STACA、PCONR.STPCA、PCONR.STASTPSA、

PConR.CMPCA[1:0]、PConR.PERCA[1:0]位设定。图 18-4 为比较输出的动作例。

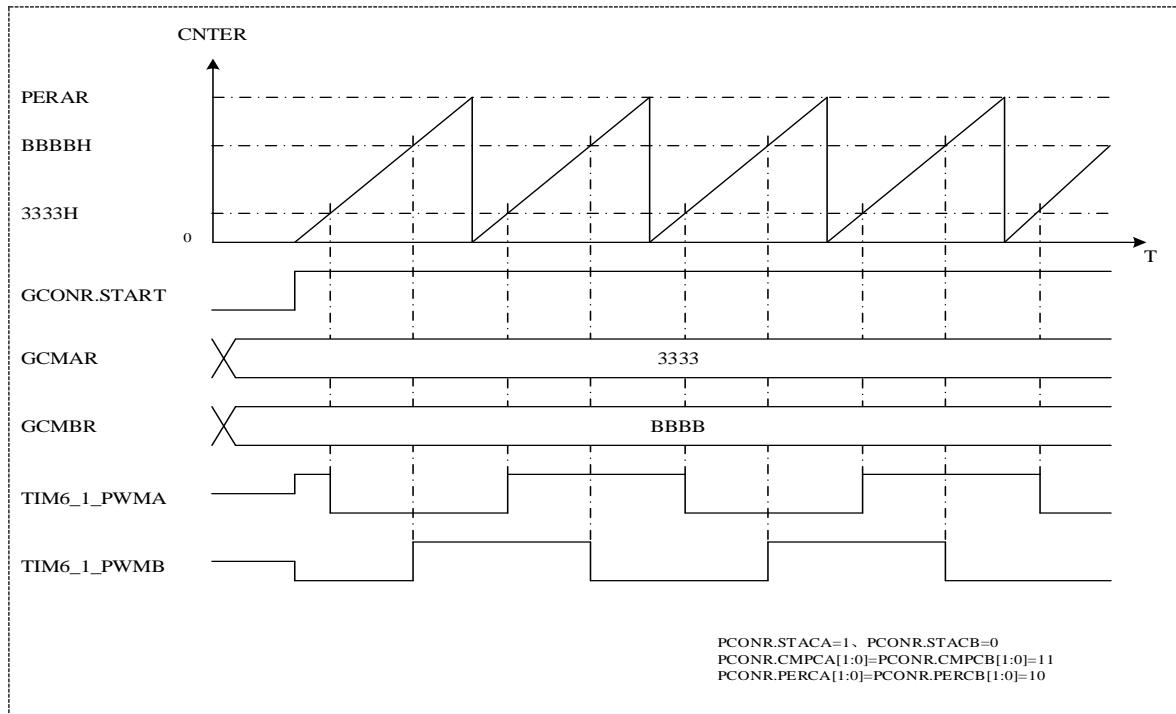


图 18-4 比较输出动作

18.3.1.3 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），用于保存捕获到的计数值。设定端口控制寄存器（PConR）的 PConR.CAPMDA、PConR.CAPMDB 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、TIM6_TRIGA 或 TIM6_TRIGB 端口输入、TIM6_<t>_PWMA 或 TIM6_<t>_PWMB 的端口输入等，具体的条件选择可通过硬件捕获事件选择寄存器（HCPAR、HCPBR）来设定。图 18-5 为捕获输入的动作例。

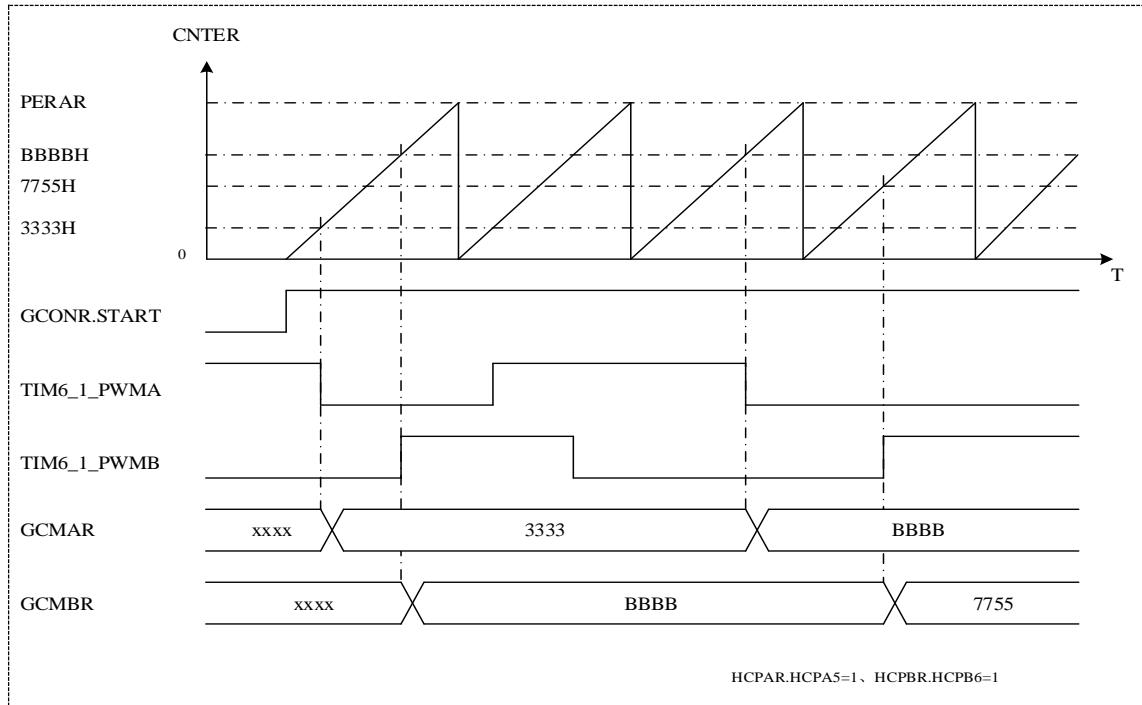


图 18-5 捕获输入动作

18.3.2 时钟源选择

Timer6 的计数时钟可以有以下几种选择:

- PCLK0 及 PCLK0 的 2、4、8、16、64、256、1024 分频 (GCONR.CKDIV[2:0] 设定)
- 内部触发事件触发输入 (HCUPR[17:16] 或 HCDOR[17:16] 设定)
- TIM6_TRIGA-B 的端口输入 (HCUPR[11:8] 或 HCDOR[11:8] 设定)
- TIM6_<t>_PWMA 和 TIM6_<t>_PWMB 的正交编码输入 (HCUPR[7:0] 或 HCDOR[7:0] 设定)

计数时钟源选择 a 时为软件计数模式，计数时钟源选择 b、c、d 时为硬件计数模式。

上述描述可以看到，b、c、d 时钟互相独立，可分别设定有效或无效，并且当选择 b、c、d 时钟时，a 时钟自动无效。

18.3.3 计数方向

Timer6 的定时器计数方向可通过软件方式改变。不同波形模式时，改变计数方向的方法略有不同。

18.3.3.1 锯齿波计数方向

锯齿波模式时，计数方向可在定时器计数中或停止时设定。

在递加计数中时，设定 GCONR.DIR=0（递减计数），则定时器计数到上溢后变为递减计数模式；在递减计数中时，设定 GCONR.DIR=1（递加计数），则定时器计数到下溢后变为递加计数模式。

在计数停止时，设定 GCONR.DIR 位，计数开始后直至上溢或下溢时，GCONR.DIR 的设定反映到计数中。

18.3.3.2 三角波计数方向

三角波模式时，设定计数方向无效，计数至计数峰点或计数谷点时，自动改变计数方向。

18.3.4 数字滤波

Timer6 的 TIM6_<t>_PWMA、TIM6_<t>_PWMB、TIM6_TRIGA~B 端口输入都有数字滤波功能。可通过设定滤波控制寄存器（FCONR）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（FCONR）设定。

在滤波采样基准时钟采样到端口上 3 次一致的电平时，该电平被当作有效电平传送到模块内部；小于 3 次一致的电平会被当作外部干扰滤掉，不传送到模块内部。其动作例如图 18-6 所示。

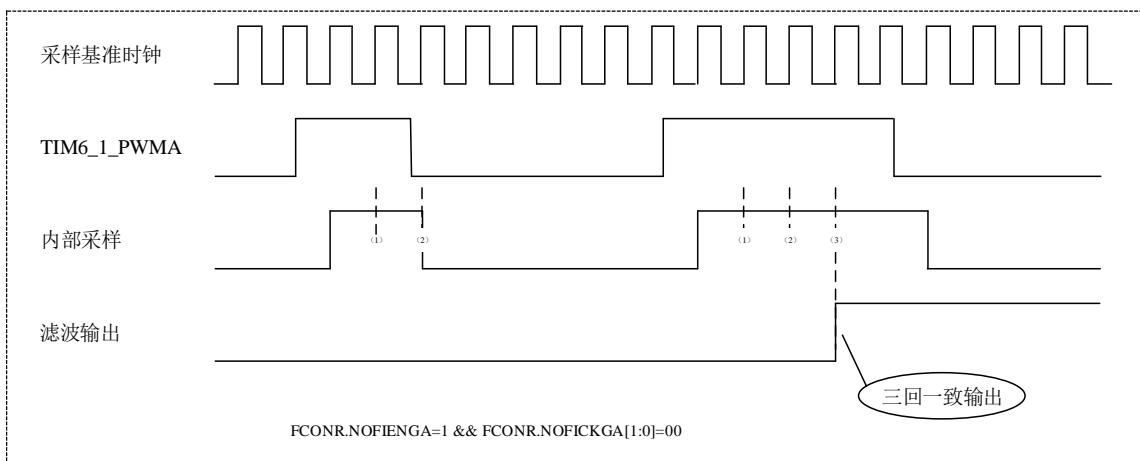


图 18-6 捕获输入端口的滤波功能

TIM6_TRIGA~B 端口是一组单元间共用的端口，该组端口的数字滤波功能由单元 1 的 FCONR 设定，其他单元的 FCONR 对该组端口的数字滤波功能设定无效，因此任意单

元使用该功能时，都需要将功能控制器（PWC_FCG2）中的 TIMER6_1 位置位。

18.3.5 软件同步

18.3.5.1 软件同步启动

各单元可通过设定软件同步启动控制寄存器（SSTAR）的相关位，实现目标单元的同步启动。

18.3.5.2 软件同步停止

各单元可通过设定软件同步停止控制寄存器（SSTPR）的相关位，实现目标单元的同步停止。

18.3.5.3 软件同步清零

各单元可通过设定软件同步清零控制寄存器（SCLRR）的相关位，实现目标单元的同步清零。

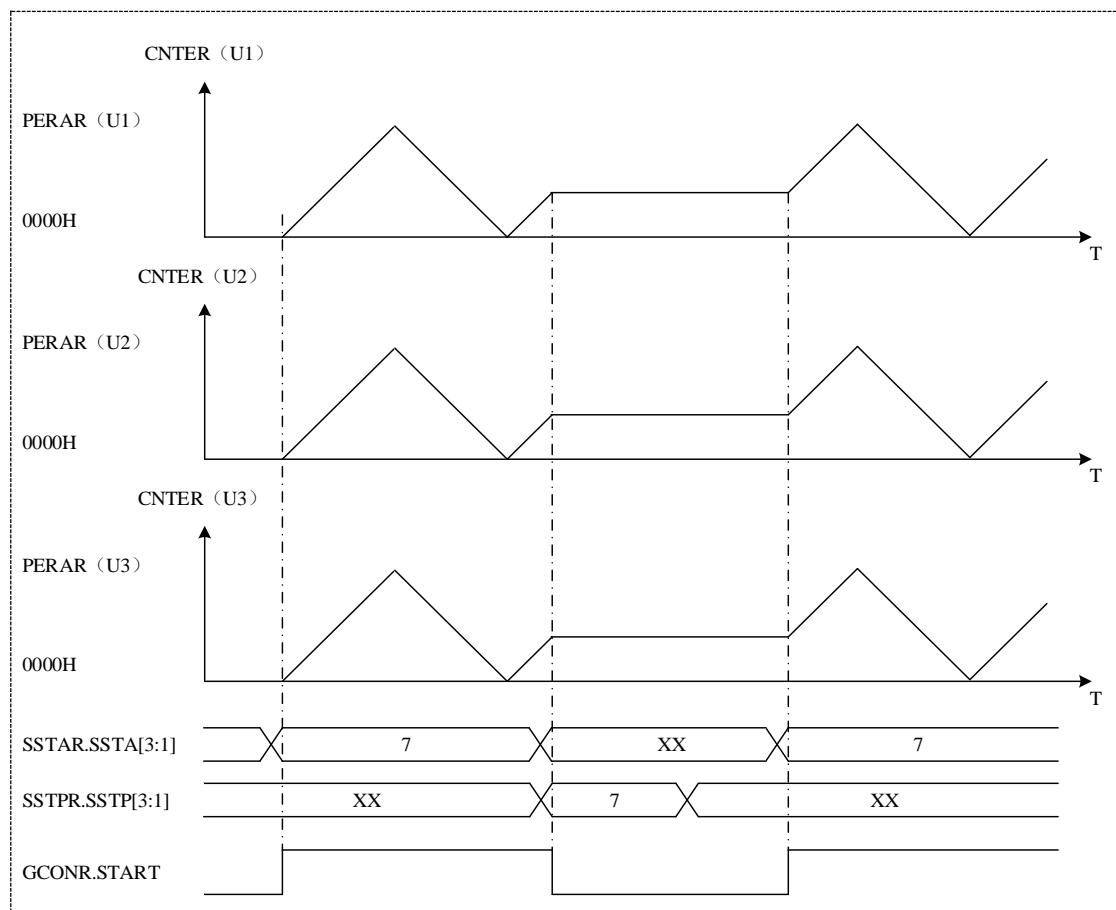


图 18-7 软件同步动作

如图 18-7 所示、若设定 SSTAR.SSTA1=SSTAR.SSTA2=SSTAR.SSTA3=1，即可实现单

元 1~3 的软件同步启动。

软件同步动作相关寄存器（SSTAR、SSTPR、SCLRR）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，在读取 SSTPR 或 SCLRR 时，会读出 0。

18.3.6 硬件同步

每个单元除独立拥有 2 个通用输入端口（TIM6_<t>_PWMA、TIM6_<t>_PWMB）外，还共同拥有 2 个外部通用输入端口（TIM6_TRIGA、TIM6_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。

内部硬件触发事件的事件源可通过硬件触发事件选择寄存器（HTSSR0~1）中对应的编号设定来选择，具体的事件对应关系请参考中断控制器（INTC）章节。使用内部硬件触发功能时，需要先将功能时钟控制寄存器 0（PWC_FCG0）的外围电路触发功能使能位置 1。

18.3.6.1 硬件同步启动

各单元均可选择用硬件方式启动定时器，选择相同硬件启动条件的单元即可在启动条件有效时实现同步启动。具体的硬件启动条件由硬件启动事件选择寄存器（HSTAR）的设定来决定。

18.3.6.2 硬件同步停止

各单元均可选择用硬件方式停止定时器，选择相同硬件停止条件的单元即可在停止条件有效时实现同步停止。具体的硬件停止条件由硬件停止事件选择寄存器（HSTPR）的设定来决定。

18.3.6.3 硬件同步清零

各单元均可选择用硬件方式清零定时器，选择相同硬件清零条件的单元即可在清零条件有效时实现同步清零。具体的硬件清零条件由硬件清零事件选择寄存器（HCLRR）的设定来决定。

18.3.6.4 硬件同步捕获输入

各单元均可选择用硬件方式实现捕获输入功能，选择相同捕获输入功能条件的单元即可在捕获输入功能条件有效时实现同步捕获输入。具体的硬件捕获输入功能条件由硬件捕获事件选择寄存器（HCPAR、HCPBR）的设定来决定。

18.3.6.5 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。

图 18-8 所示，单元 1~3 的硬件同步动作例。

选择硬件同步计数功能时，只是选择了外部输入时钟源，不影响定时器的启动、停止、清零动作。定时器的启动、停止、清零等还需要单独设定。

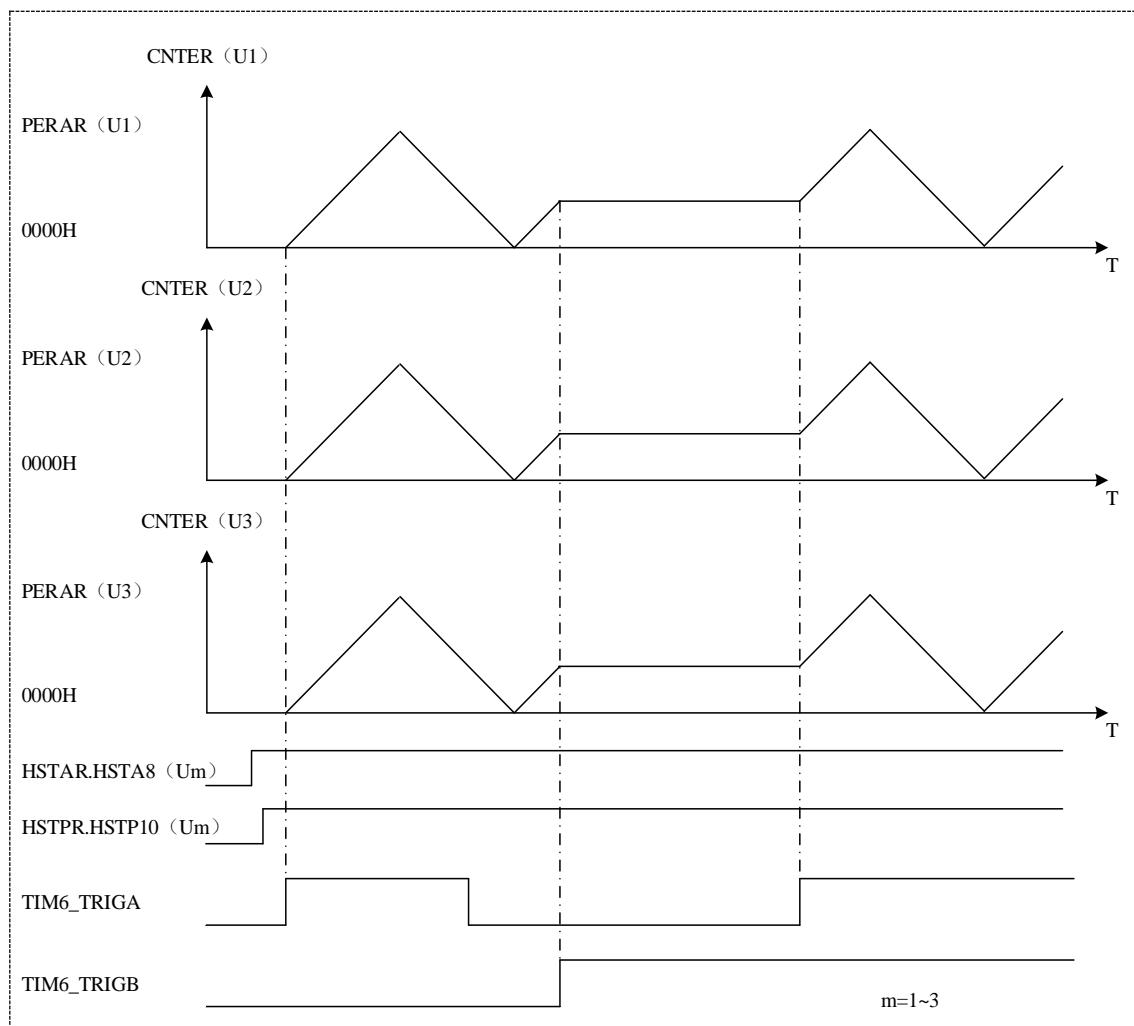


图 18-8 硬件同步动作

18.3.7 脉宽测量

在使用 TIM6_<t>_TRIGA~B 端口的硬件触发相关功能（参考硬件同步章节）时，每个单元都可以实现 2 路独立的脉宽测量功能。

例如，将计数器的硬件启动条件设为 TIM6_<t>_TRIGA 的上升沿，硬件清零条件、停止条件和 GCMAR 寄存器的捕获输入条件均设为 TIM6_<t>_TRIGA 的下降沿，就可以实现连续的脉冲宽度测量。

18.3.8 周期测量

在使用 TIM6_<t>_TRIGA~B 端口的硬件触发相关功能（参考硬件同步章节）时，每个单元都可以实现 2 路独立的周期测量功能。

例如，将计数器的硬件启动条件、硬件清零条件、GCMBR 寄存器的捕获输入条件均设为 TIM6_<t>_TRIGB 的上升沿，就可以实现连续的周期测量。

18.3.9 缓存功能

Timer6 的计数周期值、通用比较基准值、专用比较基准值、死区时间设定值等都具有缓存功能，可在计数期间实现周期变化、占空比变化、死区变化等。计数周期值、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，死区时间设定值有单缓存功能。

18.3.9.1 单缓存动作

单缓存动作是指通过设定缓存控制寄存器（BCONR）、死区控制寄存器（DCONR），在缓存传送时间点，选择发生以下事件：

- a) 通用周期基准值缓存寄存器（PERBR）的值自动传送到通用周期基准值寄存器（PERAR）中
- b) 通用比较基准值缓存寄存器（GCMCR、GCMDR）的值自动传送到通用比较基准值寄存器（GCMAR、GCMBR）中（比较输出时）
- c) 通用比较基准值寄存器（GCMAR、GCMBR）的值自动传送到通用比较基准值缓存寄存器（GCMCR、GCMDR）中（捕获输入时）
- d) 专用比较基准值缓存寄存器（SCMCR、SCMDR）的值自动传送到专用比较基准值

寄存器（SCMAR、SCMBR）中

- e) 死区时间基准值缓存寄存器（DTUBR、DTDBR）的值自动传送到死区时间基准值寄存器（DTUAR、DTDAR）中

如图 18-9 所示，是单元 1 比较输出动作时、通用比较基准值寄存器的单缓存方式的时序图。从图中可以看到，在计数期间改变通用比较基准值寄存器（GCMAR）的值可以调整输出占空比，改变通用周期基准值寄存器（PERAR）的值可以调整输出周期。

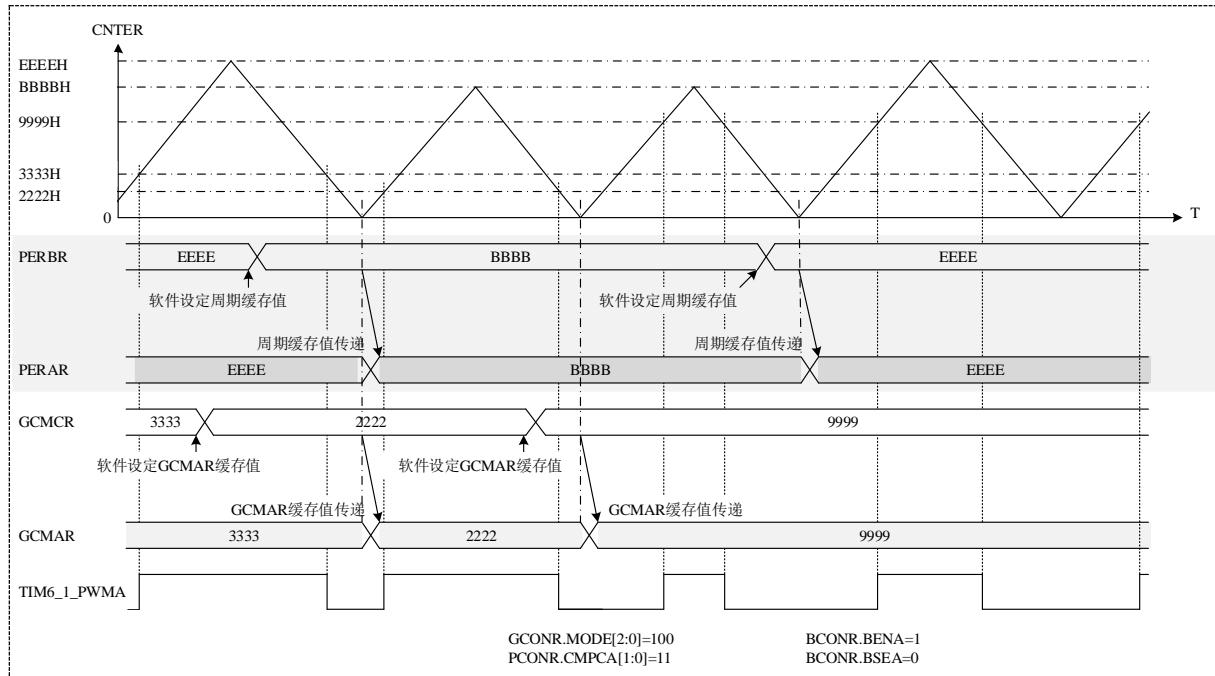


图 18-9 单缓存方式比较输出时序

18.3.9.2 双缓存动作

双缓存动作是指通过设定缓存控制寄存器（BCONR），在缓存传送时间点，选择发生以下事件：

- a) 通用周期基准值缓存寄存器（PERBR）的值自动传送到通用周期基准值寄存器（PERAR）中、通用周期基准值双缓存寄存器（PERCR）的值自动传送到通用周期基准值缓存寄存器（PERBR）中
- b) 通用比较基准值缓存寄存器（GCMCR、GCMDR）的值自动传送到通用比较基准值寄存器（GCMAR、GCMBR）、通用比较基准值双缓存寄存器（GCMER、GCMFR）的值自动传送到通用比较基准值缓存寄存器（GCMCR、GCMDR）中（比较输出时）

- c) 通用比较基准值缓存寄存器 (GCMCR、GCMDR) 的值自动传送到通用比较基准值双缓存寄存器 (GCMER、GCMFR) 中、通用比较基准值寄存器 (GCMAR、GCMBR) 的值自动传送到通用比较基准值缓存寄存器 (GCMCR、GCMDR) 中 (捕获输入时)
- d) 专用比较基准值缓存寄存器 (SCMCR、SCMDR) 的值自动传送到专用比较基准值寄存器 (SCMAR、SCMBR) 中、专用比较基准值双缓存寄存器 (SCMER、SCMFR) 的值自动传送到专用比较基准值缓存寄存器 (SCMCR、SCMDR) 中

图 18-10 所示，是内部触发事件 0 触发捕获输入时、双缓存方式的时序图。

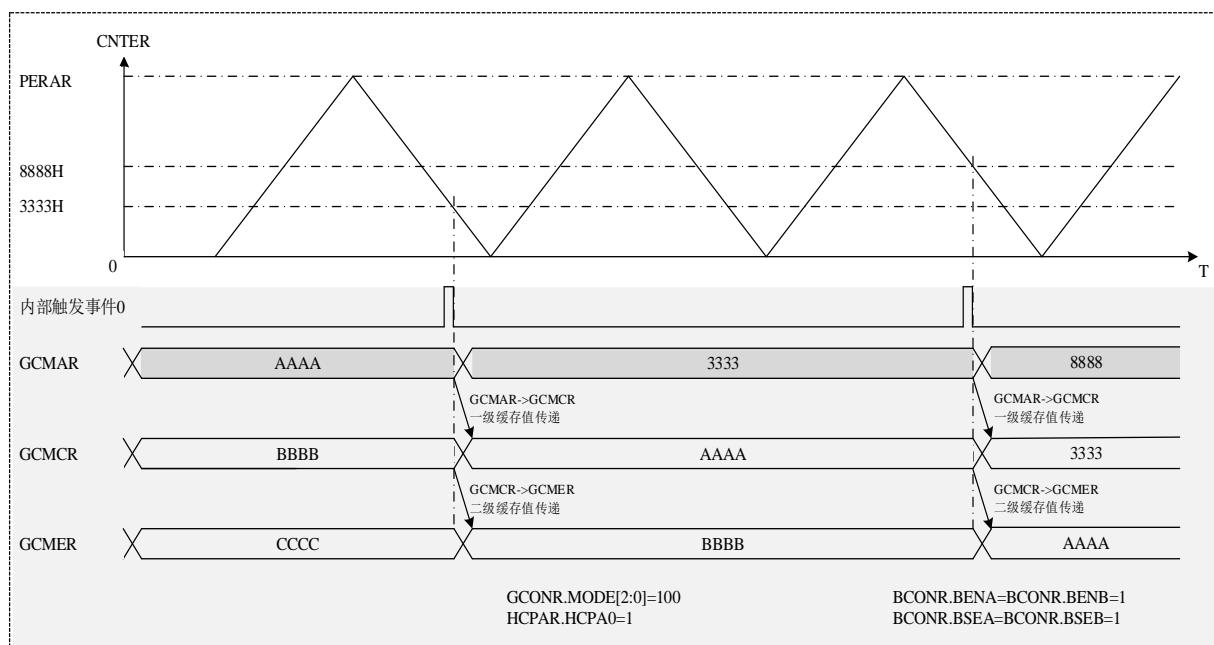


图 18-10 双缓存方式捕获输入时序

18.3.9.3 缓存传送时间点

通用周期基准值缓存传送时间点

周期基准值可选择单缓存功能或双缓存功能 (BCONR.BSEP)。缓存传送时间点为锯齿波时递加计数上溢点或递减计数下溢点、三角波时计数谷点。

通用比较基准值缓存传送时间点

锯齿波模式时，设定 BCONR.BENA=1 或 BCONR.BNEB=1，缓存动作有效。缓存动作可选单缓存功能或双缓存功能。缓存传送发生在上溢点或下溢点，如图 18-11 所示。

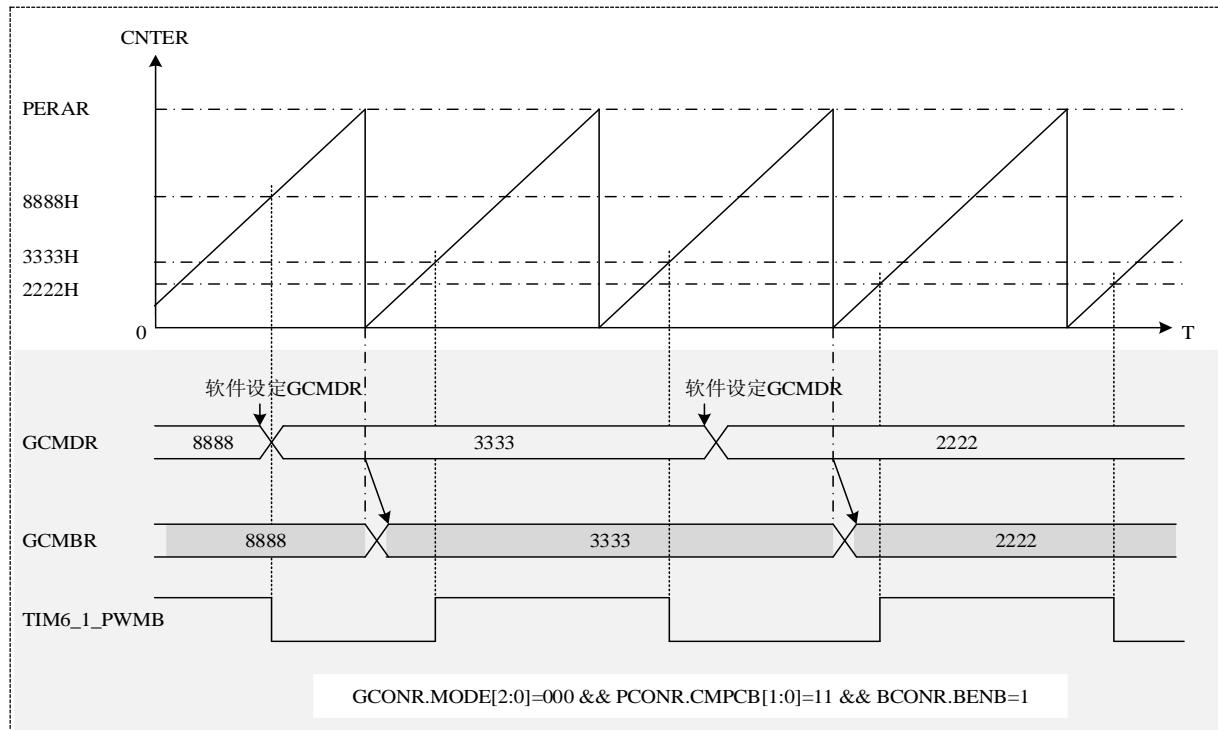


图 18-11 锯齿波模式时计数缓存动作

三角波 A 模式时，设定 BCONR.BENA=1 或 BCONR.BNEB=1，缓存动作有效。缓存动作可选单缓存功能或双缓存功能。缓存传送发生在计数谷点，如图 18-12 所示。

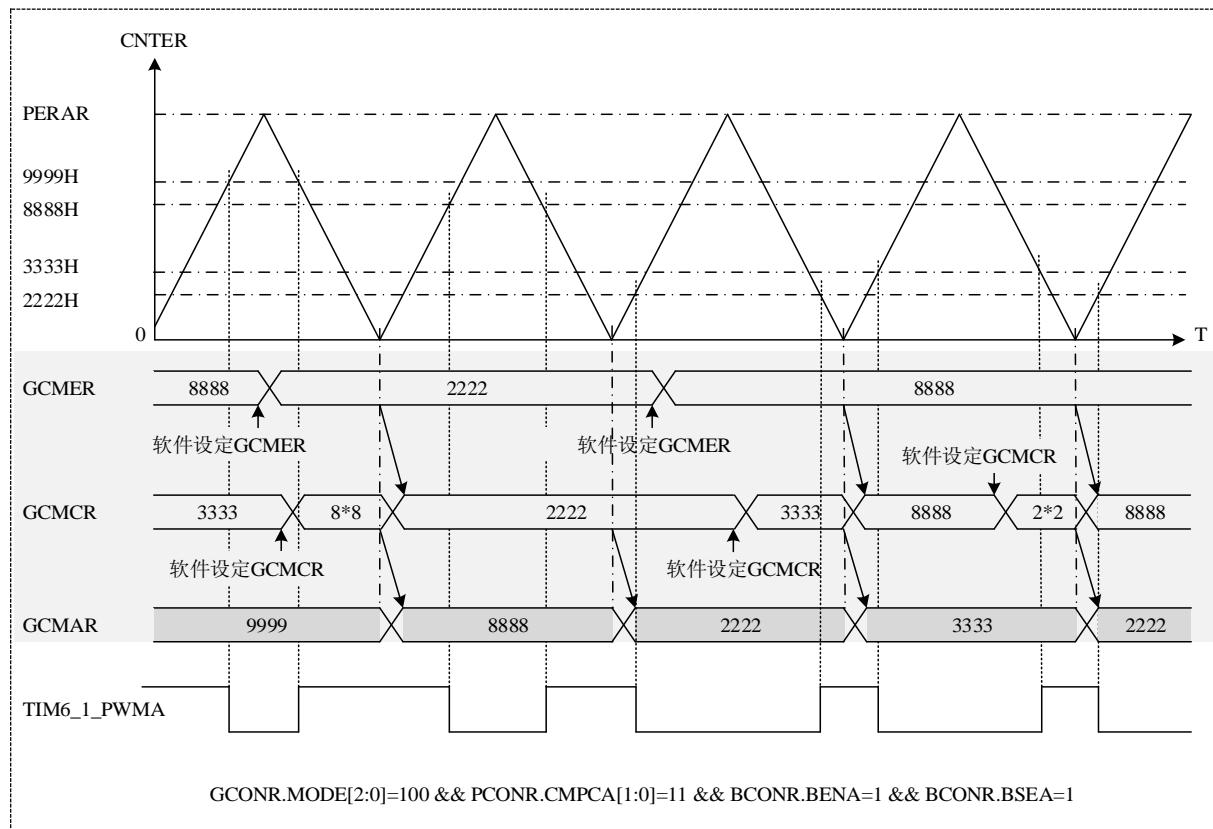


图 18-12 三角波 A 模式时计数缓存动作

三角波 B 模式时，设定 BCONR.BENA=1 或 BCONR.BNEB=1，缓存动作有效。缓存动作可选单缓存功能或双缓存功能。缓存传送发生在计数谷点或计数峰点，如图 18-13 所示。

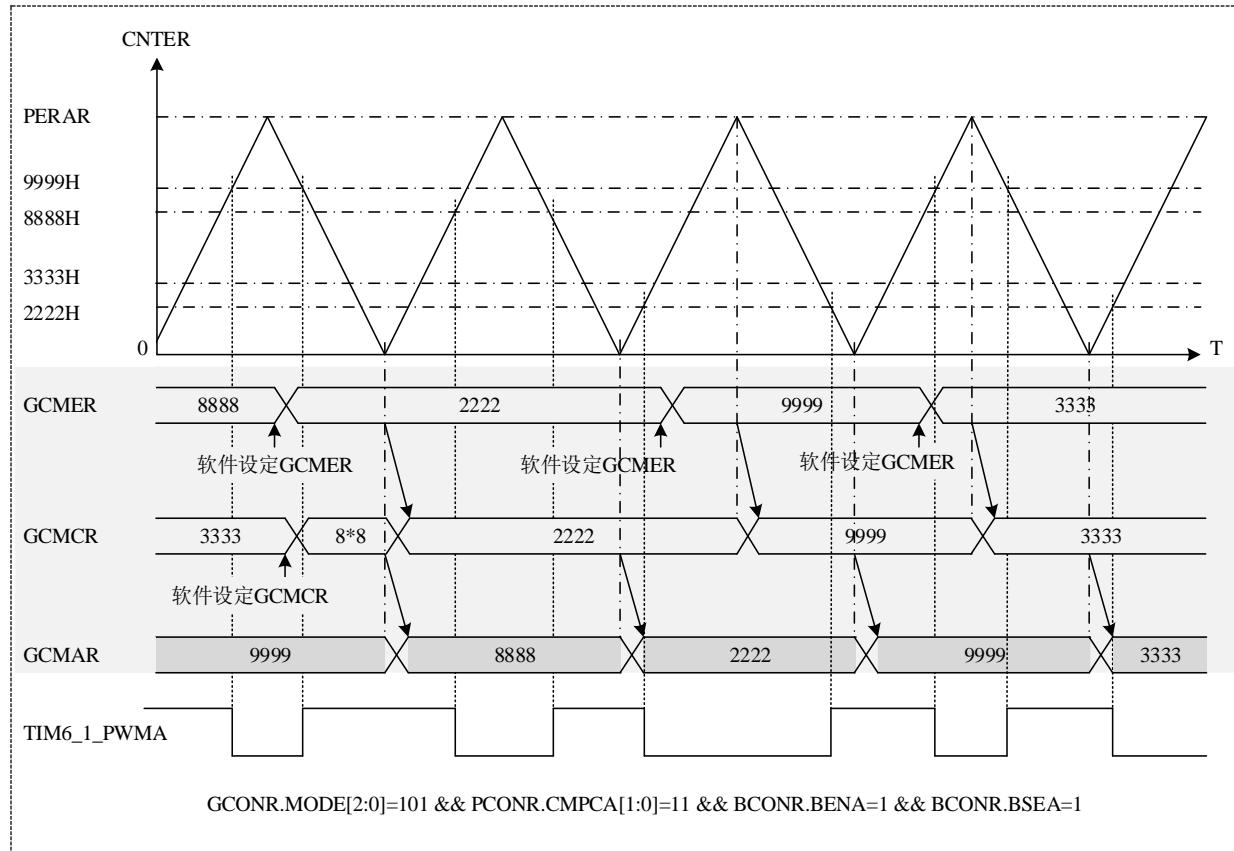


图 18-13 三角波 B 模式时计数缓存动作

单缓存传送还是双缓存传送，由 BCONR.BENA、BCONR.BENB、BCONR.BSEA、BCONR.BSEB 决定。

捕获输入值缓存传送时间点

捕获输入动作可选择单缓存功能或双缓存功能 (BCONR.BSEA 或 BCONR.BSEB)。缓存传送时间点为捕获输入动作时。

专用比较基准值缓存传送时间点

专用比较基准值可选择单缓存功能或双缓存功能 (BCONR.BSESPA 或 BCONR.BSESPB)。缓存传送时间点由缓存控制寄存器 BCONR 的 BCONR.BTRSPA 和 BCONR.BTRSPB 设定。

死区时间基准值缓存传送时间点

死区时间基准值具有单缓存功能。缓存传送时间点为锯齿波时递加计数上溢点或递减计数下溢点、三角波时计数谷点。

清零动作时缓存传送

在锯齿波计数模式或硬件计数模式时，比较输出动作期间若有清零动作产生，通用周期基准值、通用比较基准值、专用比较基准值、死区时间基准值寄存器等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。

18.3.10 通用 PWM 输出

18.3.10.1 独立 PWM 输出

每个单元的 2 个端口 `TIM6_<t>_PWMA`、`TIM6_<t>_PWMB` 能独立的输出 PWM 波。

如图 18-14，`TIM6_<t>_PWMA` 端口输出 PWM 波。

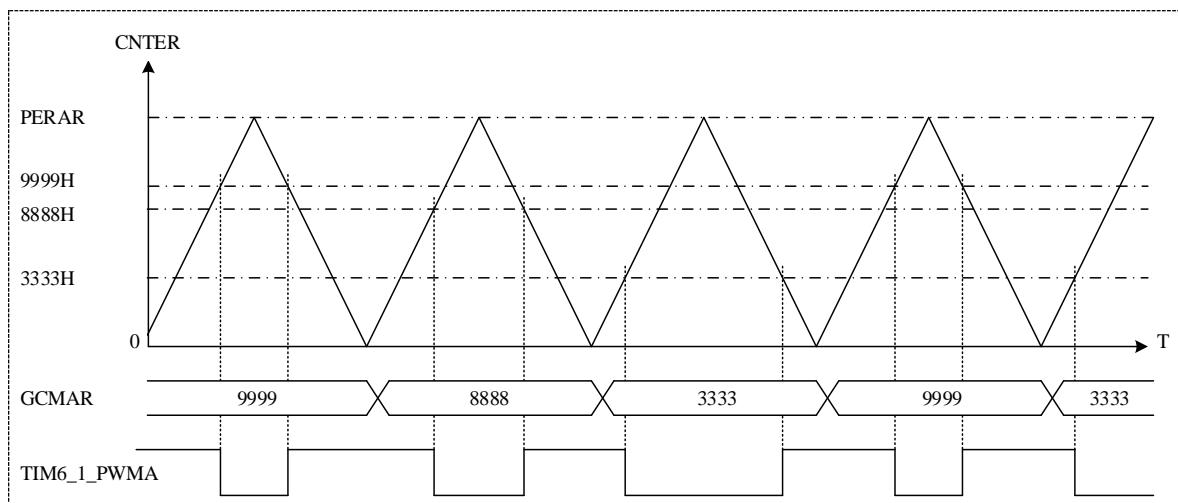


图 18-14 `TIM6_<t>_PWMA` 输出 PWM 波

18.3.10.2 互补 PWM 输出

`TIM6_<t>_PWMA` 端口和 `TIM6_<t>_PWMB` 端口，在不同的模式下可组合输出互补 PWM 波形。

软件设定GCMBR互补PWM输出

软件设定 GCMBR 互补 PWM 输出是指在锯齿波模式和三角波模式下，用于 `TIM6_<t>_PWMB` 端口波形输出的通用比较基准值寄存器 (GCMBR) 直接由 CPU 等写入，与 GCMAR 的值没有直接关系。

图 18-15 为软件设定 GCMBR 互补 PWM 波的输出例。

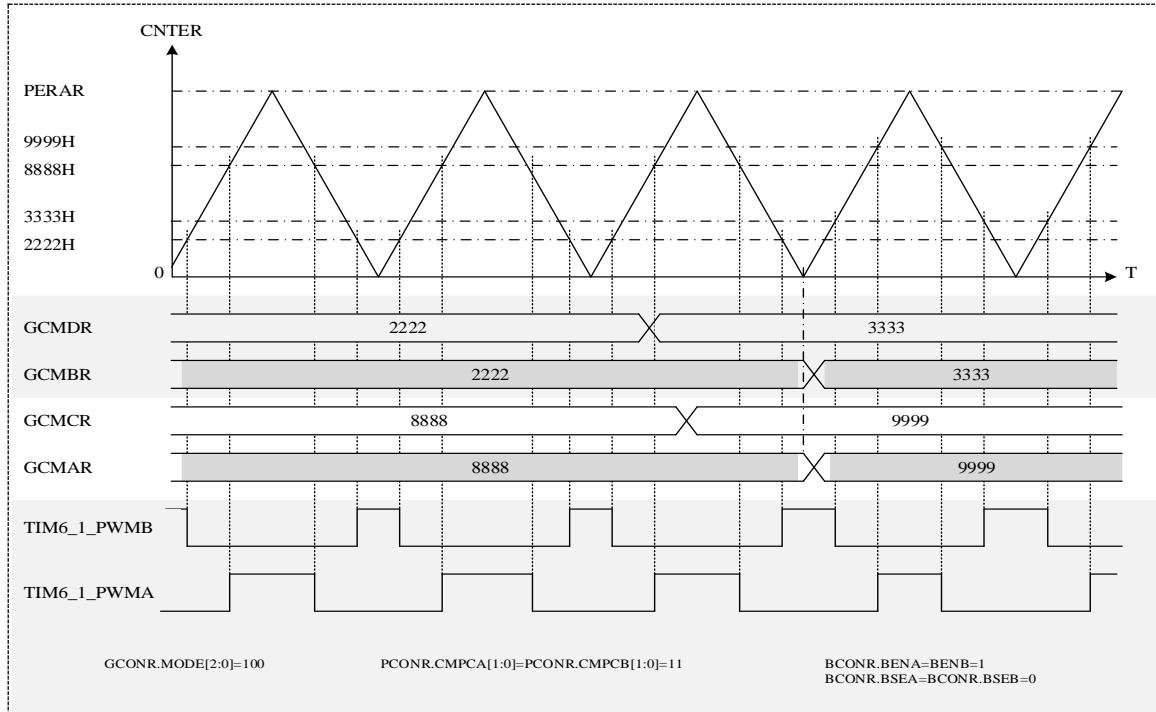


图 18-15 三角波 A 模式时软件设定 GCMBR 互补 PWM 波输出

硬件设定GCMBR互补PWM输出

硬件设定 GCMBR 互补 PWM 输出是指在三角波模式下，用于 $\text{TIM6}_{<\text{t}>}_\text{PWMB}$ 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。

死区时间设定也带有缓存功能。缓存功能有效（DCONR.DTBENU/DTBEND=1）时，在缓存传送时间点（三角波时计数谷点），DTUBR 的值传送给 DTUAR，DTDBR 的值传送给 DTDAR。

图 18-16 为硬件设定 GCMBR 互补 PWM 波输出例。

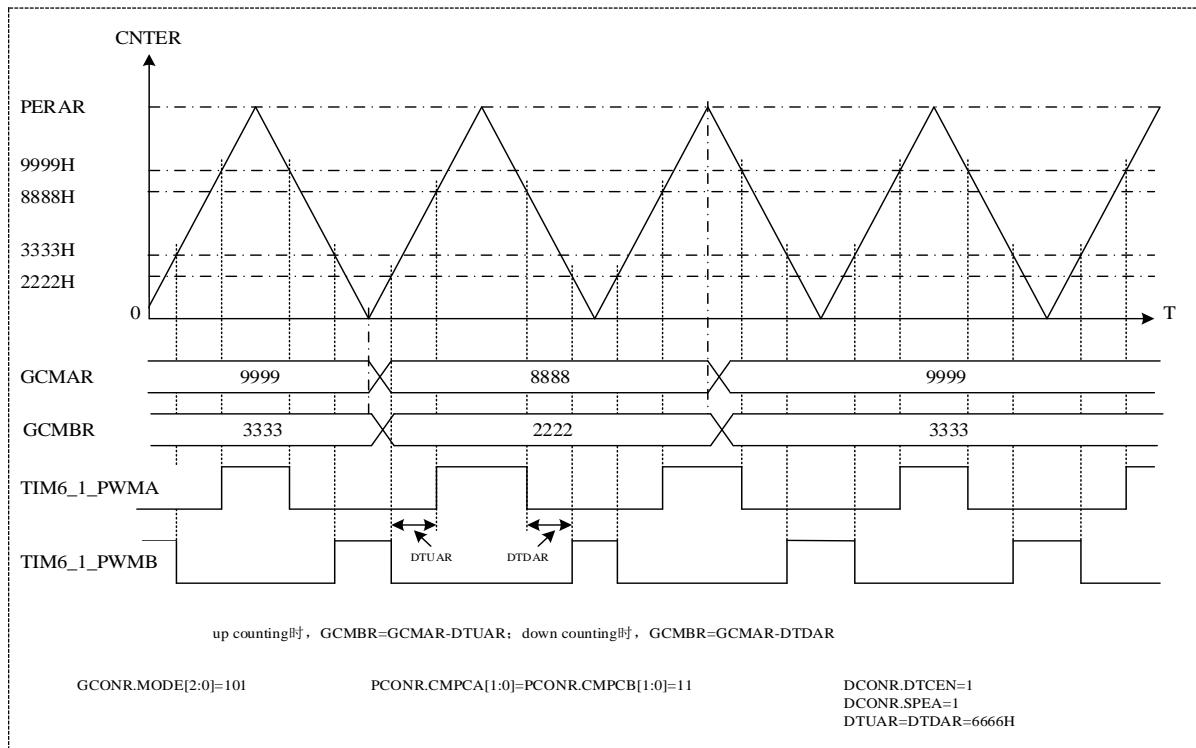


图 18-16 三角波 B 模式时硬件设定 GCMBR 互补 PWM 波输出（对称死区）

18.3.10.3 多相 PWM 输出

每个单元的 TIM6_<t>_PWMA、TIM6_<t>_PWMB 端口都能输出 2 相独立的 PWM 波或一组互补 PWM 波，多个单元间组合，同时结合软件、硬件同步动作就可实现多相 PWM 波输出。如图 18-17，单元 1、单元 2、单元 3 组合输出 6 相 PWM 波；如图 18-18，单元 1、单元 2、单元 3 组合输出 3 相互补 PWM 波。

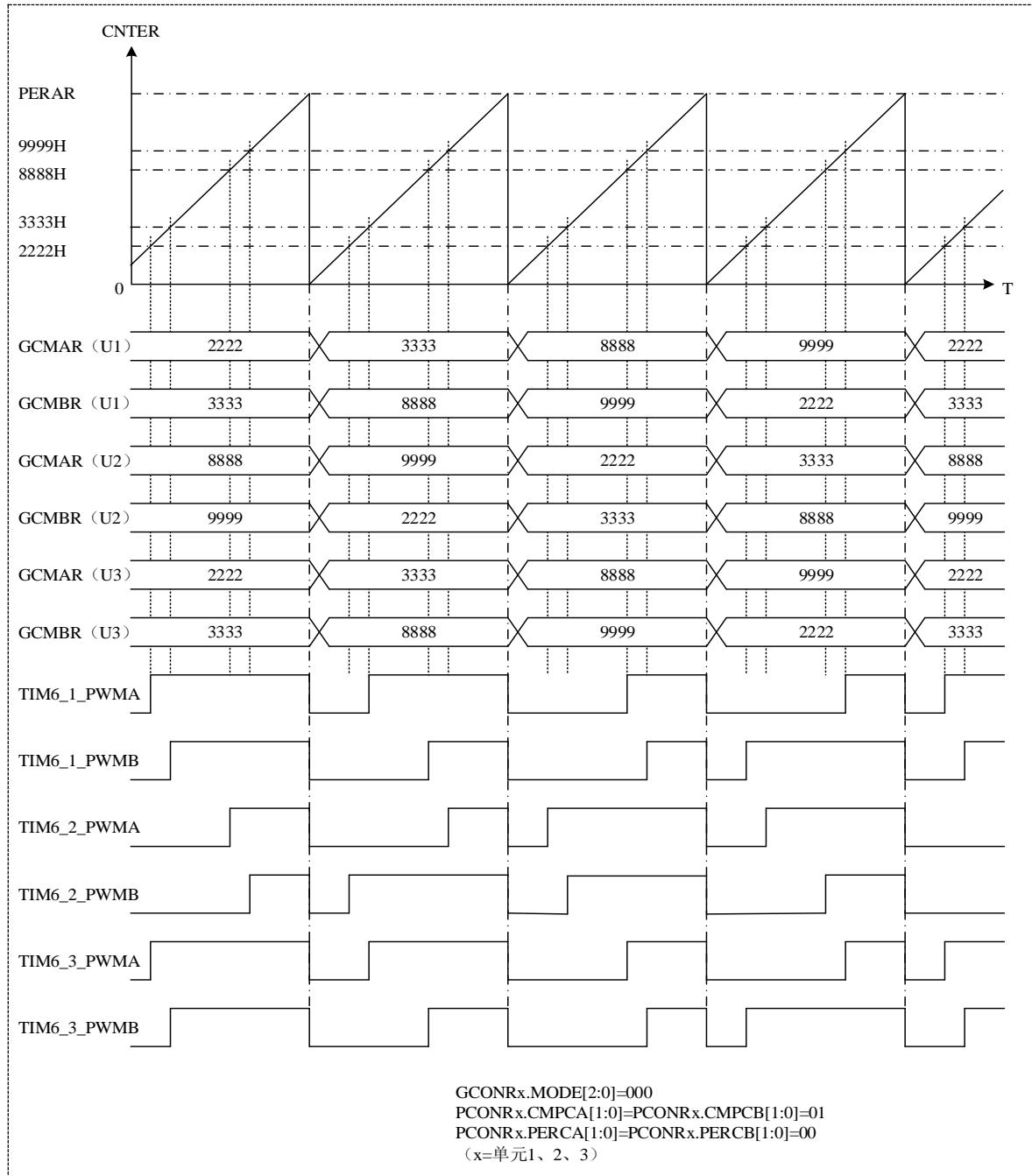


图 18-17 6 相 PWM 波

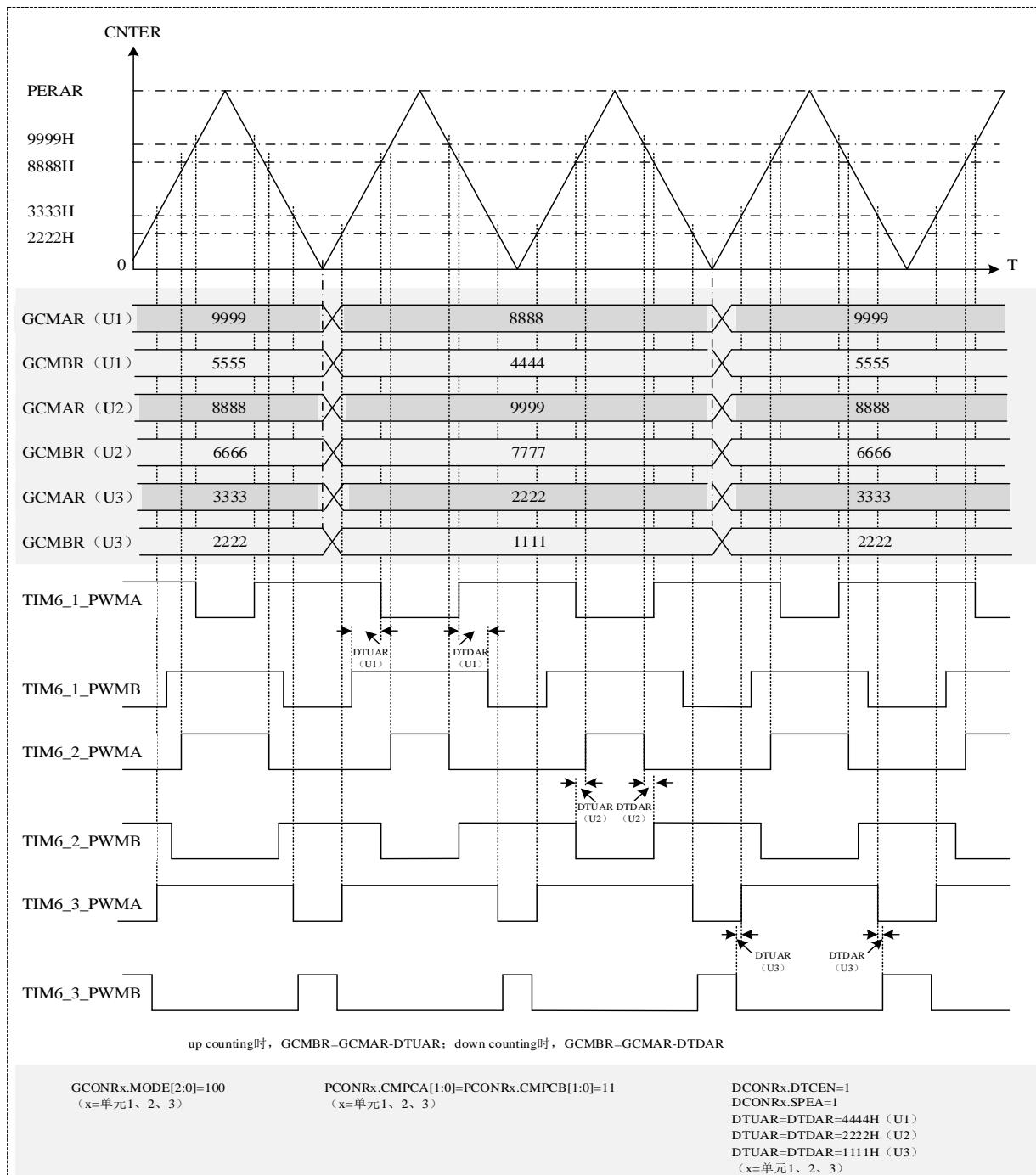


图 18-18 三角波 A 模式时带死区时间三相互补 PWM 波输出

18.3.11 正交编码计数

将 TIM6_<t>_PWMA 输入看作 AIN 输入、TIM6_<t>_PWMB 输入看作 BIN 输入、TIM6_TRIGA-B 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，单元 1、2 组合，单元 1 作为位置计数单元，单元 2 作为公转计数单元，分别实现位置计数和公转计数。单元 3 在公转模式时不使用。

AIN 和 BIN 的计数条件是通过设定硬件递加事件选择寄存器（HCUPR）和硬件递减事件选择寄存器（HCDOR）中 TIM6_<t>_PWMA 和 TIM6_<t>_PWMB 的正交关系实现；ZIN 的输入动作通过设定位置单元的硬件清零事件选择寄存器（HCLRR）实现位置计数单元的位置定时器清零、通过设定公转单元的硬件递加事件选择寄存器（HCUPR）实现公转计数单元的公转定时器计数。

18.3.11.1 位置计数模式

正交编码位置计数模式，是指根据 AIN、BIN 的输入实现基本计数功能、相位差计数功能和方向计数功能。

基本计数

基本计数动作是根据 AIN 或 BIN 端口的输入时钟进行计数，如下图 18-19 所示。

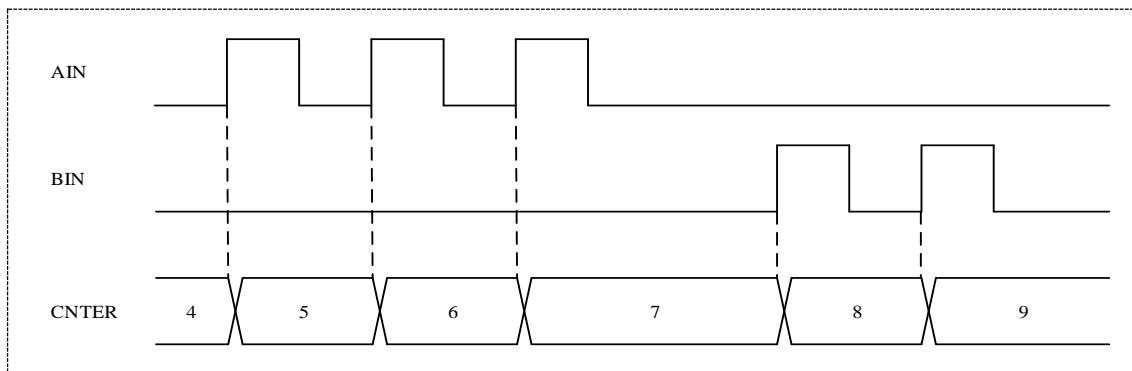


图 18-19 位置模式-基本计数

相位差计数

相位差计数是指根据 AIN 和 BIN 的相位关系进行计数。根据设定的不同，可以实现 1 倍计数、2 倍计数、4 倍计数等，如下图 18-20~图 18-22 所示。

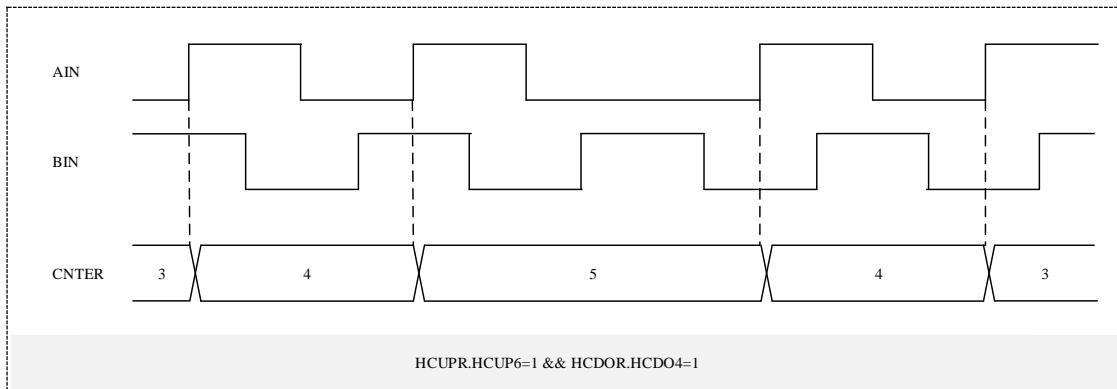


图 18-20 位置计数模式-相位差计数 (1 倍计数)

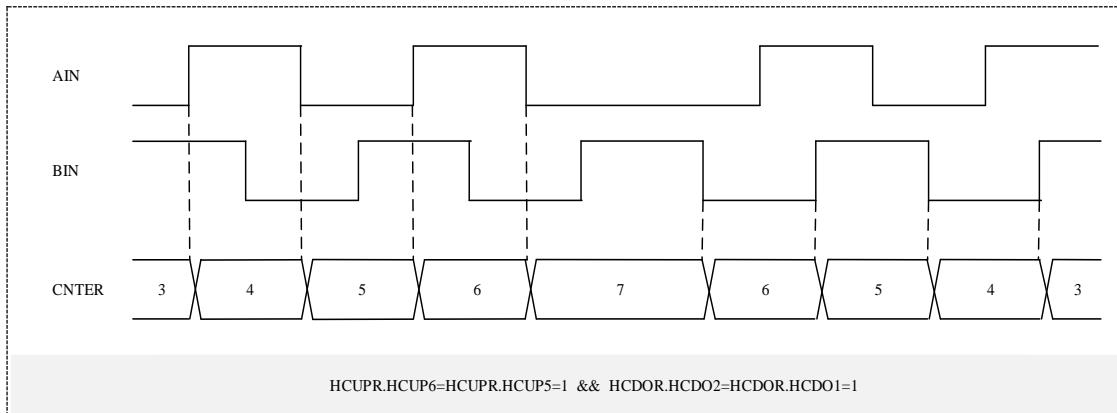


图 18-21 位置计数模式-相位差计数 (2 倍计数)

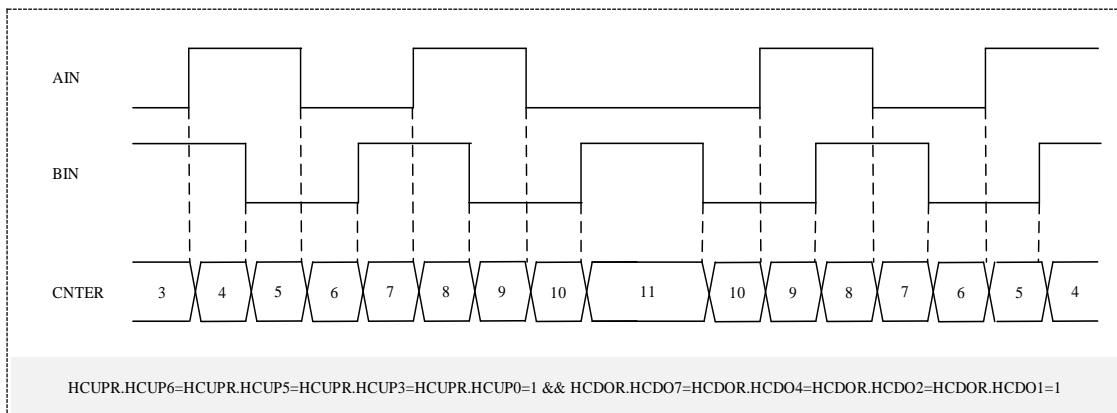


图 18-22 位置计数模式-相位差计数 (4 倍计数)

方向计数

方向计数是指将 AIN 的输入状态设定为方向控制，将 BIN 的输入作为时钟计数，如下图 18-23 所示。

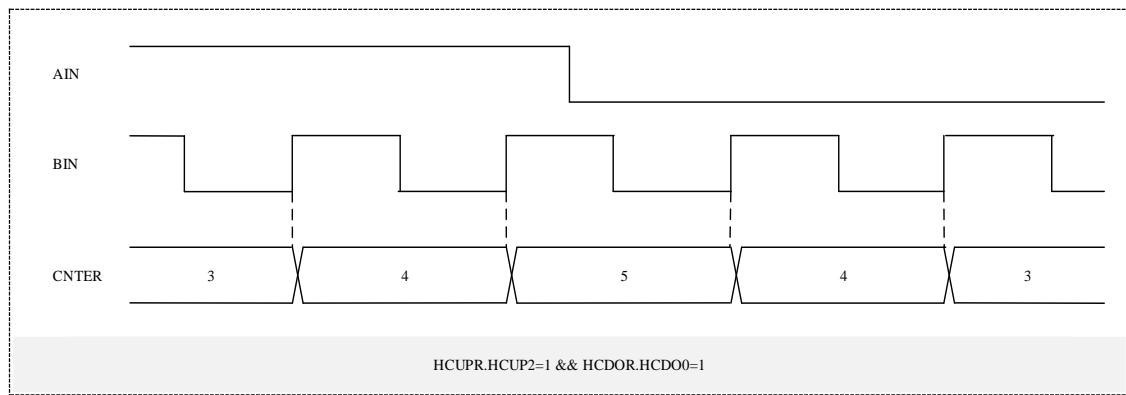


图 18-23 位置计数模式-方向计数

18.3.11.2 公转计数模式

正交编码公转计数模式，是指在 AIN、BIN 计数的基础上，加入 ZIN 的输入事件以实现对公转圈数等的判断。公转计数模式时根据公转定时器的计数方式，可实现 Z 相计数功能、位置溢出计数功能和混合计数功能。

Z相计数

Z 相计数是指根据 ZIN 的输入，公转计数单元进行计数，同时将位置计数单元清零的计数动作。如下图 18-24 所示。

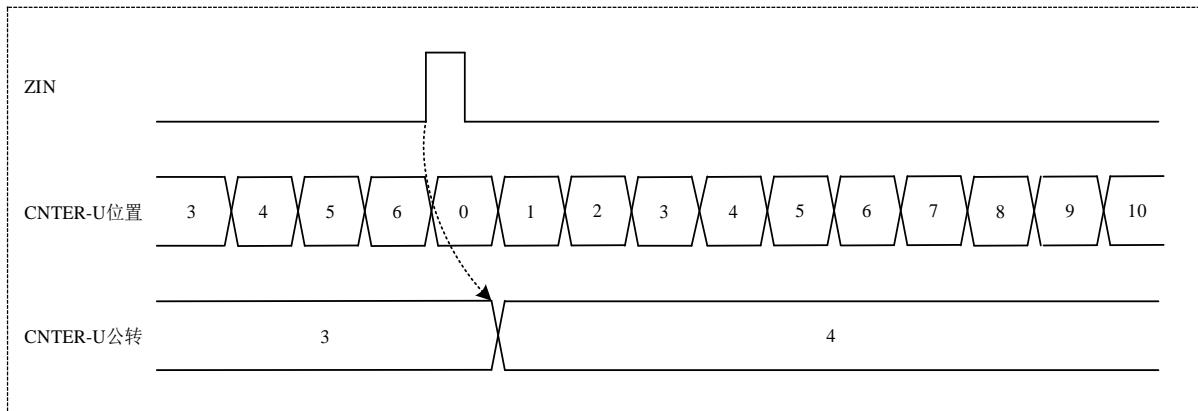


图 18-24 公转计数模式-Z 相计数

位置溢出计数

位置溢出计数是指位置计数单元计数发生上溢或下溢时，产生一个溢出事件，从而触发公转计数单元的定时器进行一次计数（在该计数方式时 ZIN 的输入不进行公转计数单元的计数动作和位置计数单元的清零动作）。

位置计数单元的溢出事件通过内部触发事件接口选通实现公转计数单元计数，即可实现位置溢出计数。公转计数单元的硬件递加（递减）事件选择寄存器（HCUPR 或 HCDOR）的递加（递减）事件选择 Bit16~Bit7 中的 1 位，同时将对应的事件触发选

择寄存器（HTSSR0~1）中的事件编号设定为位置计数单元的上溢或下溢事件。具体事件编号参考中断控制器（INTC）章节。如下图 18-25 所示。

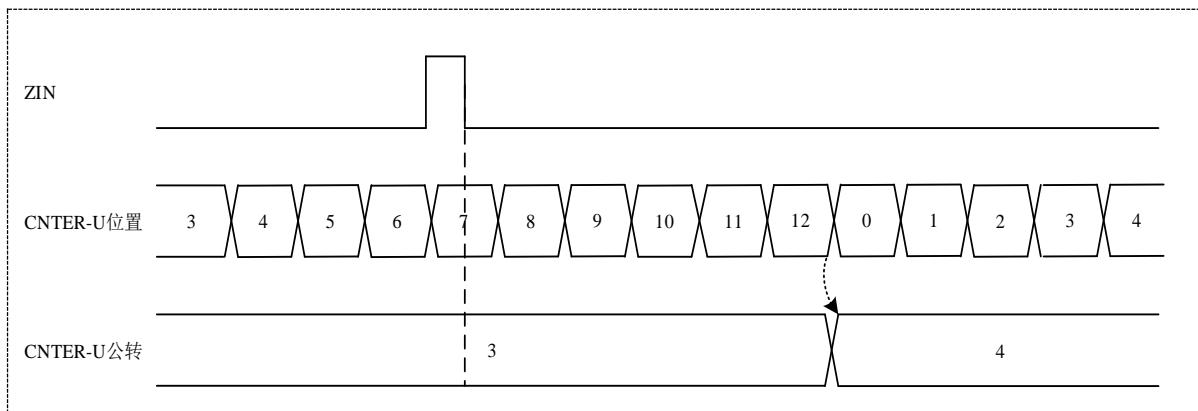


图 18-25 公转计数模式-位置溢出计数

混合计数

混合计数是指上述 Z 相计数和位置溢出计数两种计数方式合并起来的计数动作，其实现方式也是上述两种计数方式的组合。如下图 18-26 所示。

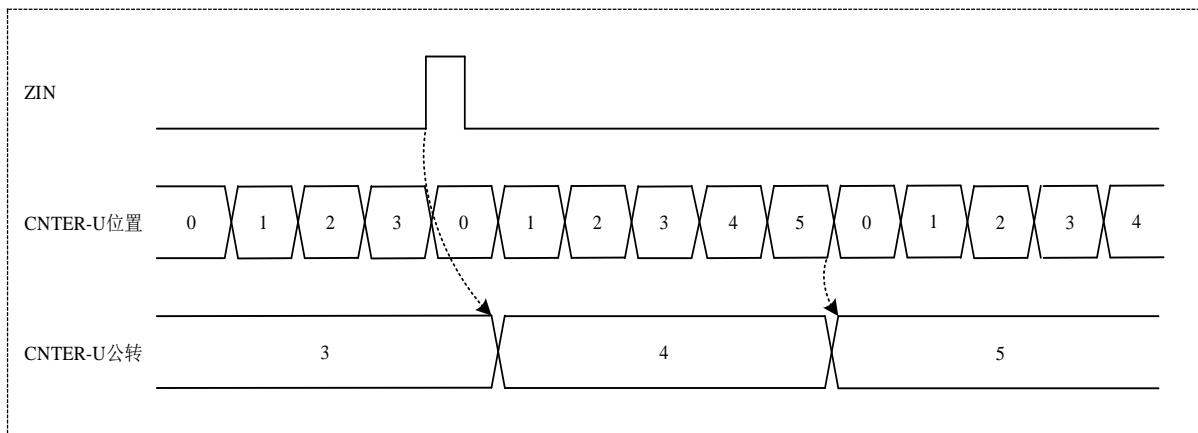


图 18-26 公转计数模式-混合计数

18.3.11.3 Z 相动作屏蔽

在公转计数模式的 Z 相计数功能或混合计数功能时，可以设定在位置定时器的上溢点或下溢点后的几个周期内（GCONR.ZMSKVAL[0:1]设定），将 ZIN 的有效输入屏蔽，不进行公转计数单元的计数和位置计数单元的清零。

位置计数单元的通用控制寄存器（GCONR）的 GCONR.ZMSKPOS 为 1 时，位置计数单元的 Z 相屏蔽功能使能，Z 相屏蔽的周期数由 GCONR.ZMSKVAL 设定；公转计数单元的通用控制寄存器（GCONR）的 GCONR.ZMSKREV 为 1 时，公转计数单元的 Z 相屏蔽功能使能。

图 18-27 是公转计数模式混合计数时，在位置计数单元计数上溢后的 4 个计数周期内有 ZIN 相输入时，ZIN 相输入的动作无效，即公转计数单元不计数、位置计数单元不清零；之后再来的 ZIN 相输入正常动作。

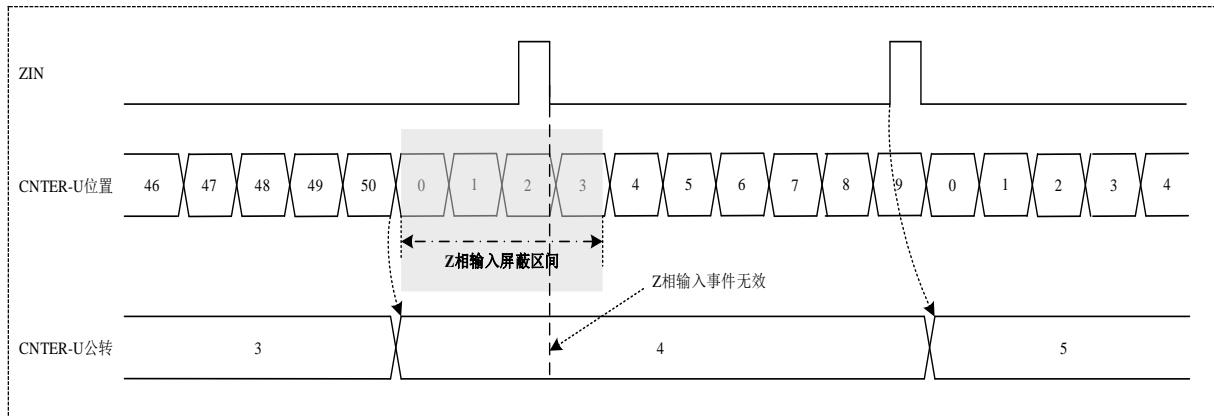


图 18-27 公转计数模式-混合计数 Z 相屏蔽动作例 1

图 18-28 是公转计数模式混合计数时，在位置计数单元计数上溢后的第 3 个周期，计数方向发生变化，此时设定的 4 个周期的屏蔽周期变为无效（实际 ZIN 相屏蔽功能维持了 3 个周期），开始向下计数。在位置计数单元发生计数下溢后，ZIN 相屏蔽功能重新开启，维持 4 个周期后变为无效。在 ZIN 相屏蔽期间，ZIN 相的输入功能无效，即公转计数单元不计数、位置计数单元不清零；之后再来的 ZIN 相输入正常动作。

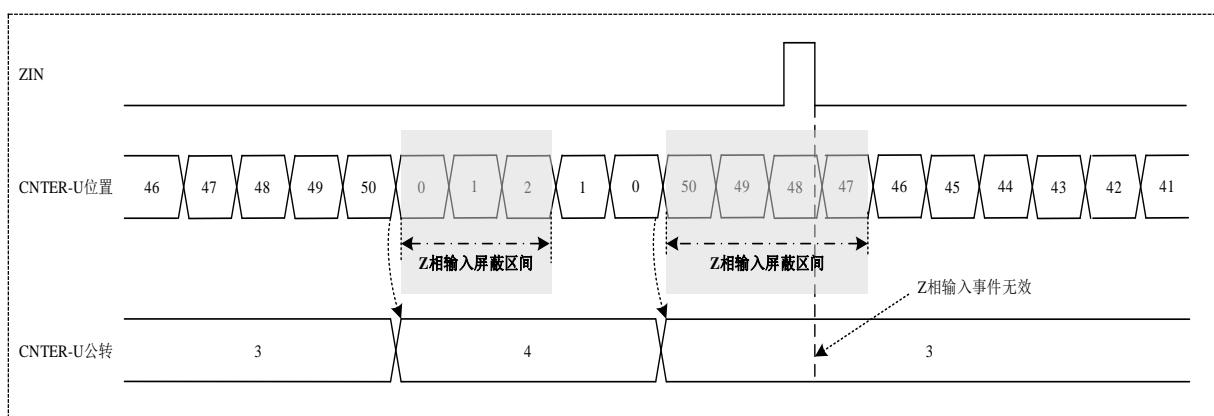


图 18-28 公转计数模式-混合计数 Z 相屏蔽动作例 2

18.3.12 周期间隔响应

Timer6 的 2 个专用比较基准值寄存器 (SCMAR、SCMBR)，在计数比较匹配时可分别输出专用比较匹配中断 A 信号、专用比较匹配中断 B 信号到 INTC 产生对应的中断；同时可分别输出专用比较匹配事件 A 信号、专用比较匹配事件 B 信号，用于和其它模块关联动作，多用于启动 ADC 等。

该中断和事件的请求信号可以每间隔几个周期后产生一次有效的请求信号，即实现周期间隔响应。该功能通过设定有效周期寄存器 (VPERR) 的 VPERR.PCNTE[1:0]位和 VPERR.SPPERIA/B 位使能。设定 VPERR.PCNTS[2:0]位来指定每隔多少个周期请求信号有效一次，其它周期内即使计数值和专用比较基准值寄存器 SCMAR 或 SCMBR 的值相等，也不会输出有效的请求信号。

如果在使用周期间隔响应功能时停止并重新启动定时器，请在停止定时器前配置 VPERR.R.PCNTE[1:0]=00，否则在重新启动后第一次产生周期间隔有效请求信号的时刻可能出现偏差。

该功能有效后，各波形模式下的周期匹配中断和周期匹配事件也只在专用比较匹配中断和事件输出的有效周期里（下图中 STFLR.VPERNUM=0 的周期）输出。图 18-29 所示是周期间隔有效请求信号的动作例。

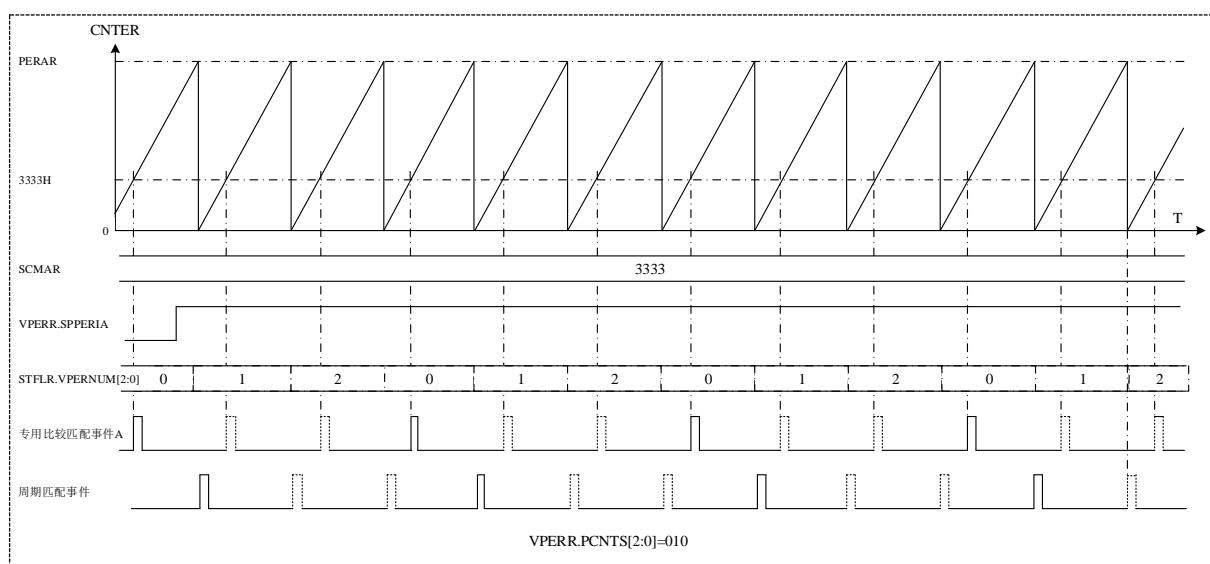


图 18-29 周期间隔有效请求信号动作

18.3.13 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，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 (PCONR.EMBVALA、PCONR.EMBVALB 的设定决定)。例如，若 PCONR.EMBVALA=01 设定时，则在 TIM6_<t>_PWMA 端口正常输出期间，若产生 EMB 事件，则 TIM6_<t>_PWMA 端口上输出变为高阻态。在 EMB 模块选择的异常事件消失、EMB 模块复位对应的事件状态位之后，Timer6 会在紧邻的下一个周期点 (三角波的计数波谷、锯齿波的上溢点或下溢点) 自动释放保护状态，变成正常的 PWM 输出，从而实现 PWM 端口的 Cycle By Cycle 控制。

18.3.14 典型应用例

下面描述几种典型应用情况下，Timer6 相关寄存器的基本设定，供用户参考。

18.3.14.1 基本计数及中断动作

- a) 设定通用周期基准值 (PERAR)
- b) 设定需要的比较基准值，包括通用比较基准值 (GCMAR~GCMFR)、专用比较基准值 (SCMAR~SCMBR) 等
- c) 设定需要的中断使能位，包括计数上溢中断 (ICONR.INTENOVF)、计数下溢中断 (ICONR.INTENUDF)、计数匹配中断 (ICONR.INTENA~F、ICONR.INTENSAU、ICONR.INTENSAD、ICONR.INTENSBU、ICONR.INTENSBD) 等
- d) 设定内部计数时钟分频 (GCONR.CKDIV[2:0])
- e) 设定波形模式 (GCONR.MODE[2:0])
- f) 设定计数方向 (仅在锯齿波模式 GCONR.MODE[2:0]=000 时需要设定)

- g) 启动计数器 (GCONR.START=1)

18.3.14.2 比较输出及中断动作

- a) 设定通用周期基准值 (PERAR)
- b) 设定各通道的比较基准值，包括通用比较基准值 A (GCMAR)、通用比较基准值 B (GCMBR)
- c) 设定需要的中断使能位，包括计数上溢中断 (ICONR.INTENOVF)、计数下溢中断 (ICONR.INTENUDF)、计数匹配中断 (ICONR.INTENA~B) 等
- d) 设定各通道在不同计数状态时的端口输出状态 (参考 PCONR 的 bit7~bit1 以及 bit23~bit17 的相关控制)
- e) 设定内部计数时钟分频 (GCONR.CKDIV[2:0])
- f) 设定波形模式 (GCONR.MODE[2:0])
- g) 设定计数方向 (仅在锯齿波模式 GCONR.MODE[2:0]=000 时需要设定)
- h) 设定各通道比较输出模式 (PCONR.CAPMDA=0、PCONR.CAPMDB=0)
- i) 设定各通道输出使能 (PCONR.OUTENA=1、PCONR.OUTENB=1)
- j) 启动计数器 (GCONR.START=1)

18.3.14.3 捕获输入及中断动作

- a) 设定通用周期基准值 (PERAR)
- b) 设定需要的中断使能位，包括计数上溢中断 (ICONR.INTENOVF)、计数下溢中断 (ICONR.INTENUDF)、捕获输入中断 (ICONR.INTENA~B) 等
- c) 设定各通道的捕获输入外部条件 (参考 HCPAR 或 HCPBR 的全部有效控制位。该有效控制位互相独立，可以同时选择多个均作为某通道的捕获输入条件)
- d) 设定内部计数时钟分频 (GCONR.CKDIV[2:0])
- e) 设定波形模式 (GCONR.MODE[2:0])
- f) 设定计数方向 (仅在锯齿波模式 GCONR.MODE[2:0]=000 时需要设定)
- g) 设定捕获输入模式 (PCONR.CAPMDA=1、PCONR.CAPMDB=1)
- h) 启动计数器 (GCONR.START=1)
- i) 等待捕获输入条件产生，读取对应通道的捕获输入值 (GCMAR 或 GCMBR) 或

等待相应的中断产生

18.3.14.4 缓存传送动作（周期基准值）

- a) 设定需要的通用周期基准值（PERAR、PERBR、PERCR）
- b) 设定单、双缓存传送方式（BCONR.BSEP）
- c) 设定内部计数时钟分频（GCONR.CKDIV[2:0]）
- d) 设定波形模式（GCONR.MODE[2:0]）（不同波形模式时的缓存传送时间点不同）
- e) 设定计数方向（仅在锯齿波模式 GCONR.MODE[2:0]=000 时需要设定）
- f) 设定缓存功能有效（BCONR.BENP=1）
- g) 启动计数器（GCONR.START=1）
- h) 等待对应的缓存传送时间点，发生缓存动作（PERBR->PERAR（BCONR.BSEP=0 时）、PERCR->PERBR->PERAR（BCONR.BSEP=1 时））

18.3.14.5 缓存传送动作（通用比较基准值）

- a) 设定需要的通用比较基准值（GCMAR、GCMCR、GCMER、GCMBR、GCMDR、GCMFR）
- b) 设定各通道单、双缓存传送方式（BCONR.BSEA、BCONR.BSEB）
- c) 设定内部计数时钟分频（GCONR.CKDIV[2:0]）
- d) 设定波形模式（GCONR.MODE[2:0]）（不同波形模式时的缓存传送时间点不同）
- e) 设定计数方向（仅在锯齿波模式 GCONR.MODE[2:0]=000 时需要设定）
- f) 设定各通道缓存功能有效（BCONR.BENA=1、BCONR.BENB=1）
- g) 启动计数器（GCONR.START=1）
- h) 等待各通道所设定的对应缓存传送时间点，发生缓存动作（GCMCR->GCMAR（BCONR.BSEA=0 时）、GCMER->GCMCR->GCMAR（BCONR.BSEA=1 时）、GCMDR->GCMBR（BCONR.BSEB=0 时）、GCMFR->GCMDR->GCMBR（BCONR.BSEB=1 时））

18.3.14.6 缓存传送动作（专用比较基准值）

- a) 设定需要的专用比较基准值（SCMAR、SCMCR、SCMER、SCMBR、SCMDR、SCMFR）

- b) 设定各通道单、双缓存传送方式 (BCONR.BSESPA、BCONR.BSESPB)
- c) 设定各通道缓存传送时间点 (BCONR.BTRSPA[1:0]、BCONR.BTRSPB[1:0])
- d) 设定内部计数时钟分频 (GCONR.CKDIV[2:0])
- e) 设定波形模式 (GCONR.MODE[2:0])
- f) 设定计数方向 (仅在锯齿波模式 GCONR.MODE[2:0]=000 时需要设定)
- g) 设定各通道缓存功能有效 (BCONR.BENSPA=1、BCONR.BENSPB=1)
- h) 启动计数器 (GCONR.START=1)
- i) 等待各通道所设定的对应缓存传送时间点, 发生缓存动作 (SCMCR->SCMAR (BCONR.BSESPA=0 时)、SCMER->SCMCR->SCMAR(BCONR.BSESPA=1 时)、SCMDR->SCMBR (BCONR.BSESPB=0 时)、SCMFR->SCMDR->SCMBR (BCONR.BSESPB=1 时))

18.3.14.7 缓存传送动作 (死区基准值)

- a) 设定需要的死区时间基准值 (DTUAR、DTUBR、DTDAR、DTDBR)
- b) 设定内部计数时钟分频 (GCONR.CKDIV[2:0])
- c) 设定波形模式 (GCONR.MODE[2:0])
- d) 设定计数方向 (仅在锯齿波模式 GCONR.MODE[2:0]=000 时需要设定)
- e) 设定缓存功能有效 (DCONR.DTBENU=1、DCONR.DTBEND=1)
- f) 设定硬件死区功能有效 (DCONR.DTCEN=1)
- g) 启动计数器 (GCONR.START=1)
- h) 等待对应缓存传送时间点, 发生缓存动作(DTUBR->DTUAR、DTDBR->DTDAR)

18.3.14.8 同步启动动作 (软件方式)

- a) 参考[基本计数及中断动作]章节的 a~f 步骤, 对需要同步启动的各个单元做设定
- b) 同步启动计数器 (设定 SSTAR 寄存器的对应位为 1, 每个单元对应一个寄存器位)

18.3.14.9 同步启动动作 (硬件方式)

- a) 设定通用周期基准值 (PERAR)
- b) 设定需要的比较基准值, 包括通用比较基准值 (GCMAR~GCMFR)、专用比较基

准值 (SCMAR~SCMBR) 等

- c) 设定需要的中断使能位，包括计数上溢中断 (ICONR.INTENOVF)、计数下溢中断 (ICONR.INTENUDF)、计数匹配中断 (ICONR.INTENA~F、ICONR.INTENSAU、ICONR.INTENSAD、ICONR.INTENSBU、ICONR.INTENSBD) 等
- d) 设定硬件启动条件 (通过 HSTAR.HSTAx 选择, x=0~1、8~11)
- e) 设定硬件启动使能 (HSTAR.STARTS=1)
- f) 重复上述 a~e 步骤对需要同步启动的各个单元做设定 (需要同步启动的各个单元中，步骤 d 的设定要一致)
- g) 等待设定的触发事件产生，确认各个单元的计数器同步启动

18.3.14.10 正交编码计数动作 (2 相)

- a) 设定通用周期基准值 (PERAR)
- b) 设定需要的比较基准值，包括通用比较基准值 (GCMAR~GCMFR)、专用比较基准值 (SCMAR~SCMBR) 等
- c) 设定需要的中断使能位，包括计数上溢中断 (ICONR.INTENOVF)、计数下溢中断 (ICONR.INTENUDF)、计数匹配中断 (ICONR.INTENA~F、ICONR.INTENSAU、ICONR.INTENSAD、ICONR.INTENSBU、ICONR.INTENSBD) 等
- d) 设定需要的硬件向上计数条件 (通过 HCUPR.HCUPx 选择, x=0~7)
- e) 设定需要的硬件向下计数条件 (通过 HCDOR.HCDOx 选择, x=0~7)
- f) 启动计数器 (GCONR.START=1)
- g) 等待设定的正交编码计数事件产生，确认计数器正常计数

18.3.14.11 正交编码计数动作 (3 相)

- a) 参考[正交编码计数动作 (2 相)]章节的 a~e 步骤，对位置计数单元做设定
- b) 设定位臵计数单元的硬件清零条件 (通过 HCLRR.HCLRx 选择, x=8~11)
- c) 设定位臵计数单元硬件清零使能 (HCLRR.CLEARS=1)
- d) 设定公转计数单元的通用周期基准值 (PERAR)
- e) 设定公转计数单元的比较基准值，包括通用比较基准值 (GCMAR~GCMFR)、专用比较基准值 (SCMAR~SCMBR) 等

- f) 设定公转计数单元需要的中断使能位，包括计数上溢中断 (ICONR.INTENOVF)、计数下溢中断 (ICONR.INTENUDF)、计数匹配中断 (ICONR.INTENA~F、ICONR.INTENSAU、ICONR.INTENSAD、ICONR.INTENSBU、ICONR.INTENSBD) 等
- g) 设定公转计数单元的硬件向上计数条件 1 (ZIN 相输入) (通过 HCUPR.HCUPx 选择， $x=8\sim11$ ，此处的设定事件应和步骤 b 中位置计数单元设定的事件一致)
- h) 设定公转计数单元的硬件向上计数条件 2 (位置计数单元的上溢事件输入) (通过 HCUPR.HCUP16 选择内部硬件触发事件 0)
- i) 设定公转计数单元的硬件向下计数条件 (位置计数单元的下溢事件输入) (通过 HCDOR.HCDO17 选择内部硬件触发事件 1)
- j) 设定 HTSSR0 中的触发源编号为位置计数单元的计数上溢事件 (该上溢事件编号参考 INTC 章节)
- k) 设定 HTSSR1 中的触发源编号为位置计数单元的计数下溢事件 (该下溢事件编号参考 INTC 章节)
- l) 启动公转计数单元计数器 (GCONR.START=1)
- m) 启动位置计数单元计数器 (GCONR.START=1)
- n) 等待设定的 AIN、BIN、ZIN 相计数事件产生，确认计数器正常计数

18.3.14.12 单路 PWM 输出

- a) 参考[比较输出及中断动作]章节 a~j 步骤的设定 (每个单元内部的 2 个 PWM 通道 TIM6_<t>_PWMA 和 TIM6_<t>_PWMB 的输出状态均可独立设定，形成 2 个互不相关的单路 PWM 输出)

18.3.14.13 互补 PWM 输出 (软件死区)

- a) 设定通用周期基准值 (PERAR)
- b) 设定通用比较基准值 A (GCMAR)、通用比较基准值 B (GCMBR)
- c) 设定需要的中断使能位，包括计数上溢中断 (ICONR.INTENOVF)、计数下溢中断 (ICONR.INTENUDF)、计数匹配中断 (ICONR.INTENA~B) 等
- d) 设定在不同计数状态时的端口输出状态 (参考 PCONR 的 bit7~bit1、bit23~bit16)

相关控制，结合 GCMAR 和 GCMBR 的设定值，需要保证 2 个 PWM 输出之间形成互补死区）

- e) 设定内部计数时钟分频 (GCONR.CKDIV[2:0])
- f) 设定波形模式为三角波模式 (GCONR.MODE=100 或 101)
- g) 设定比较输出模式 (PCONR.CAPMDA=0、PCONR.CAPMDB=0)
- h) 设定输出使能 (PCONR.OUTENA=1、PCONR.OUTENB=1)
- i) 启动计数器 (GCONR.START=1)

18.3.14.14 互补 PWM 输出（硬件死区）

- a) 设定通用周期基准值 (PERAR)
- b) 设定通用比较基准值 A (GCMAR)、死区时间基准值 (DTUAR、DTDAR)
- c) 设定需要的中断使能位，包括计数上溢中断 (ICONR.INTENOVF)、计数下溢中断 (ICONR.INTENUFD)、计数匹配中断 (ICONR.INTENA~B)、死区错误中断 (ICONR.INTENDTE) 等
- d) 设定在不同计数状态时的端口输出状态（参考 PCONR 的 bit7~bit1、bit23~bit16 相关控制，结合 GCMAR、DTUAR 和 DTDAR 的设定值，需要保证 2 个 PWM 输出之间形成互补死区）
- e) 设定内部计数时钟分频 (GCONR.CKDIV[2:0])
- f) 设定波形模式为三角波模式 (GCONR.MODE[2:0]=100 或 101)
- g) 设定各通道比较输出模式 (PCONR.CAPMDA=0、PCONR.CAPMDB=0)
- h) 设定各通道输出使能 (PCONR.OUTENA=1、PCONR.OUTENB=1)
- i) 设定硬件死区功能有效 (DCONR.DTCEN=1)
- j) 启动计数器 (GCONR.START=1)

18.3.14.15 EMB 监控及中断动作

- a) 参考[互补 PWM 输出（软件死区）]章节的 a~h 步骤或[互补 PWM 输出（硬件死区）]章节 a~i 步骤，对互补 PWM 输出动作进行设定
- b) 设定 EMB 事件发生时 PWM 端口的状态 (PCONR.EMBVALA、PCONR.EMBVAB)
(根据系统应用的不同，选择相应的保护状态)

- c) 设定 EMB 模块的相关寄存器（包括 EMB 中断许可寄存器（EMB_INTEN0）、EMB 控制寄存器 0（EMB_CTL0）等）
- d) 启动计数器（GCONR.START=1），EMB 模块实时监控系统状态

18.3.15 功能汇总表

Timer6 的锯齿波模式和三角波 A、B 模式下，主要功能的汇总表如下表 18-3 所示。

PWM输出功能			锯齿波	三角波		对应寄存器设定 (X=A、B)	备注
				三角波A	三角波B		
独立 PWM 输出	端口状态 控制	启动时端口输出 设定	支持	支持	支持	PCONR.STACX PCONR.STASTPSX	
		停止时端口输出 设定	支持	支持	支持	PCONR.STACX PCONR.STASTPSX	
		比较匹配时端口 输出设定	支持	支持	支持	PCONR.CMPCX	
		周期匹配时端口 输出设定	支持	支持	支持	PCONR.PERCX	
互补 PWM 输出	缓存 传送	周期 基准值	单缓存	支持	支持	支持 控制位：BCONR.BENP 基准值：PERAR、PERBR	
			双缓存	支持	支持	支持 控制位：BCONR.BSEP 基准值： PERAR、PERBR、PERC	
		比较 基准值	单缓存	支持	支持	支持 控制位：BCONR.BENX 基准值： GCMAR、GCMCR GCMBR、GCMDR	三角波A模式和 三角波B模式的 缓存传送点不同
			双缓存	支持	支持	支持 控制位：BCONR.BSEX 基准值： GCMAR、GCMCR、GCMER GCMBR、GCMDR、GCMFR	
	紧急制动			支持	支持	支持 PCONR.EMBVALX	
	端口状态 控制	启动时端口输出 设定	支持	支持	支持	PCONR.STACX PCONR.STASTPSX	
		停止时端口输出 设定	支持	支持	支持	PCONR.STACX PCONR.STASTPSX	
		比较匹配时端口 输出设定	支持	支持	支持	PCONR.CMPCX	

		周期匹配时端口 输出设定	支持	支持	支持	PCONR.PERCX	
缓存 传送	周期 基准值	单缓存	支持	支持	支持	控制位: BCONR.BENP 基准值: PERAR、PERBR	
		双缓存	支持	支持	支持	控制位: BCONR.BSEP 基准值: PERAR、PERBR、PERC	
	比较 基准值	单缓存	支持	支持	支持	控制位: BCONR.BENX 基准值: GCMAR、GCMCR GCMBR、GCMDR	三角波A模式和 三角波B模式的 缓存传送点不同
		双缓存	支持	支持	支持	控制位: BCONR.BSEX 基准值: GCMAR、GCMCR、GCMER GCMBR、GCMDR、GCMFR	
	死区 基准值	单缓存	支持	支持	支持	控制位: DCONR.DTBENU DCONR.DTBEND 基准值: DTUAR、DTDAR DTUBR、DTDBR	
无死区PWM输出			支持	支持	支持	GCMAR=GCMBR	
带死区PWM输出	软件 方式	支持	支持	支持	支持	GCMAR≠GCMBR	
		不支持	支持	支持	支持	控制位: DCONR.DTCEN 基准值: GCMAR、DTUAR、DTDAR	
紧急制动			支持	支持	支持	PCONR.EMBVALX	

表 18-3 不同模式下的功能对比表

18.4 中断及事件说明

18.4.1 中断输出

Timer6 含有 6 个通用计数比较匹配中断（含 2 个捕获输入中断）、2 个专用计数比较匹配中断、2 个计数周期匹配中断、1 个死区时间错误中断。

18.4.1.1 计数比较匹配中断

通用比较基准值寄存器（GCMAR-GCMFR）共计 6 个，可分别与计数值比较产生比较匹配。计数比较匹配时，状态标志寄存器（STFLR）中的 STFLR.CMAF~STFLR.CMFF 位分别会被置为 1。此时若设定中断控制寄存器（ICONR）的 INTENA~INTENF 中相应位为 1 使能中断，则对应的中断请求（TMR6_U<t>_GCMA~F）也会被触发。

在硬件捕获事件选择寄存器（HCPAR、HCPBR）选择的捕获输入有效条件产生时，捕获输入动作发生。此时若设置中断控制寄存器（ICONR）的 INTENA 或 INTENB 位为 1 使能中断，则对应的中断请求（TMR6_U<t>_GCMA~B）被触发。

2 个专用比较基准值寄存器（SCMAR-SCMBR）也可分别与计数值比较产生比较匹配。计数比较匹配时，状态标志寄存器（STFLR）中的 STFLR.CMSPAF~CMSPBF 位分别会被置为 1。此时若设定中断控制寄存器（ICONR）的 INTENSAU<D>或 INTENSBU<D>中相应位为 1 使能中断，则对应的中断请求（TMR6_U<t>_SCMA~B）也会被触发。

18.4.1.2 计数周期匹配中断

锯齿波递加计数至上溢点、锯齿波递减计数至下溢点、三角波计数至谷点或三角波计数至峰点时，状态标志寄存器（STFLR）的 STFLR.OVFF 或 STFLR.UDFF 位会被置为 1。此时若设定中断控制寄存器（ICONR）的 ICONR.INTENOVF 或 ICONR.INTENUDF 位使能中断，则在对应的时间点可触发计数周期匹配中断（TMR6_U<t>_GOVF 和 TMR6_U<t>_GUDF）。

18.4.1.3 死区时间错误中断

死区时间基准值寄存器（DTU<D>AR）的值加载到通用比较基准值寄存器（GCMBR）中时，若超过周期限制，则会产生死区时间错误，状态标志寄存器（STFLR）的 STFLR.DTEF 位会被置为 1。此时若设置中断控制寄存器（ICONR）的 INTENDTE 位

使能中断，则会在该时刻触发死区时间错误中断 (TMR6_U<t>_GDTE)。

18.4.2 事件输出

在时钟计数过程中，若产生周期匹配事件（锯齿波的上溢和下溢点，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。

下图是单元 1 的通用比较匹配中断 A-F&事件 A-F、专用比较匹配中断 A-B&事件 A-B、周期匹配中断&事件的动作例。

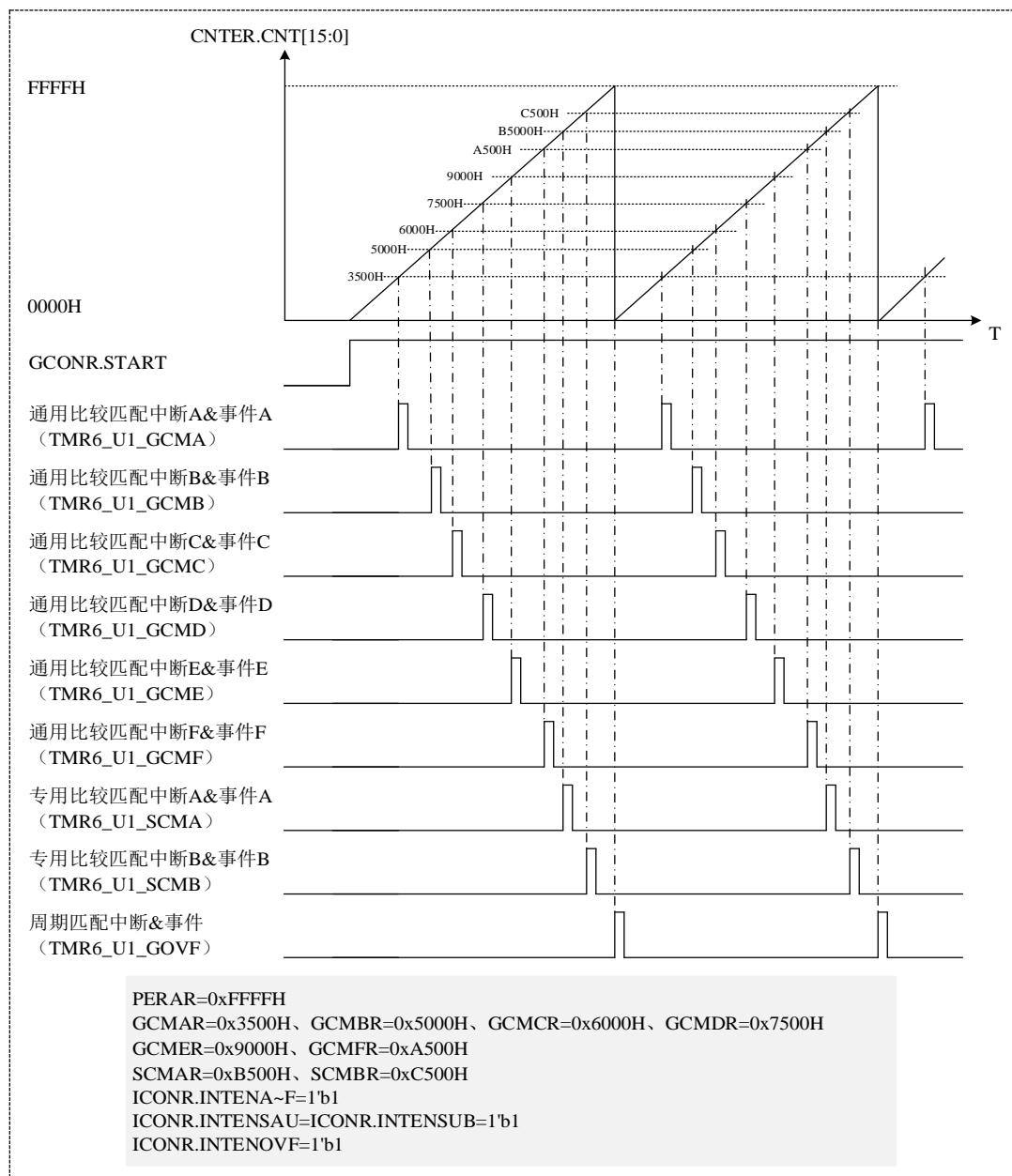


图 18-30 锯齿波模式时中断&事件输出例

18.5 寄存器说明

表 18-4 所示, 为 Timer6 模块的寄存器列表。

BASE ADDR: 0x4001_8000 (U1)、0x4001_8400 (U2)、0x4001_8800 (U3)

寄存器名	符号	偏移量	位宽	复位值
通用计数值寄存器	TMR6_CNTER	0x0000	32	0x0000_0000
通用周期基准值寄存器A	TMR6_PERAR	0x0004	32	0x0000_FFFF
通用周期基准值寄存器B	TMR6_PERBR	0x0008	32	0x0000_FFFF
通用周期基准值寄存器C	TMR6_PERCR	0x000C	32	0x0000_FFFF
通用比较基准值寄存器A	TMR6_GCMAR	0x0010	32	0x0000_FFFF
通用比较基准值寄存器B	TMR6_GCMBR	0x0014	32	0x0000_FFFF
通用比较基准值寄存器C	TMR6_GCMCR	0x0018	32	0x0000_FFFF
通用比较基准值寄存器D	TMR6_GCMDR	0x001C	32	0x0000_FFFF
通用比较基准值寄存器E	TMR6_GCMER	0x0020	32	0x0000_FFFF
通用比较基准值寄存器F	TMR6_GCMFR	0x0024	32	0x0000_FFFF
专用比较基准值寄存器A	TMR6_SCMAR	0x0028	32	0x0000_FFFF
专用比较基准值寄存器B	TMR6_SCMBR	0x002C	32	0x0000_FFFF
专用比较基准值寄存器C	TMR6_SCMCR	0x0030	32	0x0000_FFFF
专用比较基准值寄存器D	TMR6_SCMDR	0x0034	32	0x0000_FFFF
专用比较基准值寄存器E	TMR6_SCMER	0x0038	32	0x0000_FFFF
专用比较基准值寄存器F	TMR6_SCMFR	0x003C	32	0x0000_FFFF
死区时间基准值寄存器UA	TMR6_DTUAR	0x0040	32	0x0000_FFFF
死区时间基准值寄存器DA	TMR6_DTDAR	0x0044	32	0x0000_FFFF
死区时间基准值寄存器UB	TMR6_DTUBR	0x0048	32	0x0000_FFFF
死区时间基准值寄存器DB	TMR6_DTDBR	0x004C	32	0x0000_FFFF
通用控制寄存器	TMR6_GCONR	0x0050	32	0x0000_0100
中断控制寄存器	TMR6_ICONR	0x0054	32	0x0000_0000
端口控制寄存器	TMR6_PCONR	0x0058	32	0x0000_0000

缓存控制寄存器	TMR6_BCONR	0x005C	32	0x0000_0000
死区控制寄存器	TMR6_DCONR	0x0060	32	0x0000_0000
滤波控制寄存器	TMR6_FCONR	0x0068	32	0x0000_0000
有效周期寄存器	TMR6_VPERR	0x006C	32	0x0000_0000
状态标志寄存器	TMR6_STFLR	0x0070	32	0x8000_0000
硬件启动事件选择寄存器	TMR6_HSTAR	0x0074	32	0x0000_0000
硬件停止事件选择寄存器	TMR6_HSTPR	0x0078	32	0x0000_0000
硬件清零事件选择寄存器	TMR6_HCLRR	0x007C	32	0x0000_0000
硬件捕获事件选择寄存器A	TMR6_HCPAR	0x0080	32	0x0000_0000
硬件捕获事件选择寄存器B	TMR6_HCPBR	0x0084	32	0x0000_0000
硬件递加事件选择寄存器	TMR6_HCUPR	0x0088	32	0x0000_0000
硬件递减事件选择寄存器	TMR6_HCDOR	0x008C	32	0x0000_0000
软件同步启动控制寄存器	TMR6_SSTAR	(0x4001_83F4)	32	0x0000_0000
软件同步停止控制寄存器	TMR6_SSTPR	(0x4001_83F8)	32	0x0000_0000
软件同步清零控制寄存器	TMR6_SCLRR	(0x4001_83FC)	32	0x0000_0000

表 18-4 寄存器列表

注意：

- 软件同步寄存器（TMR6_SSTAR、TMR6_SSTPR、TMR6_SCLLR）是 3 个独立于单元的寄存器，为 3 个单元 Timer6 所共有。

18.5.1 通用计数值寄存器 (TMR6_CNTER)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”	R
b15~b0	CNT[15:0]	计数值	当前定时器的计数值	R/W

18.5.2 通用周期基准值寄存器 (TMR6_PERAR-PERCR)

复位值: 0x0000_FFFF

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”	R
b15~b0	PERA-C[15:0]	计数周期值	设定每轮计数的计数周期值及对应缓存值	R/W

18.5.3 通用比较基准值寄存器 (TMR6_GCMAR-GCMFR)

复位值: 0x0000_FFFF

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”	R
b15~b0	GCMA-F[15:0]	计数比较基准值	比较基准值设定，与计数值相等时匹配信号有效	R/W

18.5.4 专用比较基准值寄存器（TMR6_SCMAR-SCMFR）

复位值: 0x0000_FFFF

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”	R
b15~b0	SCMA-F[15:0]	专用比较基准值	设定比较基准值及缓存值	R/W

18.5.5 死区时间基准值寄存器（TMR6_DTU<D>AR）

复位值: 0x0000_FFFF

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”	R
b15~b0	DTU/DA-B[15:0]	死区时间值	死区时间设定值及缓存值	R/W

18.5.6 通用控制寄存器 (TMR6_GCONR)

复位值: 0x0000_0100

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	ZMSK VAL[1:0]	ZMSK POS	ZMSK REV	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	DIR	-	CKDIV[2:0]	CKDIV[2:0]	MODE[2:0]	MODE[2:0]	START		

位	标记	位名	功能	读写
b31~b20	Reserved	-	读出时为“0”,写入时写“0”	R/W
b19~b18	ZMSKVAL[1:0]	Z相输入屏蔽周期数	正交编码Z相输入被屏蔽的计数周期值	R/W
		00: Z相输入屏蔽功能无效		
		01: 位置计数上溢后或下溢后的4个计数周期内的Z相输入被屏蔽		
		10: 位置计数上溢后或下溢后的8个计数周期内的Z相输入被屏蔽		
		11: 位置计数上溢后或下溢后的16个计数周期内的Z相输入被屏蔽		
b17	ZMSKPOS	Z相输入位置定时器选择	0: Z相输入时该单元作为位置定时器,在屏蔽周期期间内位置定时器清零功能正常动作 1: Z相输入时该单元作为位置定时器,在屏蔽周期期间内位置定时器清零功能被屏蔽	R/W
b16	ZMSKREV	Z相输入公转定时器选择	0: Z相输入时该单元作为公转定时器,在屏蔽周期期间内公转定时器计数功能正常动作 1: Z相输入时该单元作为公转定时器,在屏蔽周期期间内公转定时器计数功能被屏蔽	R/W
b15~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8	DIR	计数方向	0: 递减计数 1: 递加计数	R/W
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6~b4	CKDIV[2:0]	计数时钟选择	000: PCLK0 001: PCLK0/2 010: PCLK0/4 011: PCLK0/8 100: PCLK0/16 101: PCLK0/64 110: PCLK0/256 111: PCLK0/1024	R/W
b3~b1	MODE[2:0]	计数模式	000: 锯齿波模式 100: 三角波A模式 101: 三角波B模式 请不要设定其它值	R/W
b0	START	定时器启动	0: 定时器关闭	R/W

1: 定时器启动

注: 该位在软件停止条件或硬件停止条件有效时, 会自动变为0

18.5.7 中断控制寄存器 (TMR6_ICONR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved														INTEN	INTEN
												SBD	SBU	SAD	SAU
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	INTEN DTE	INTEN UDF	INTEN OVF	INTEN F	INTEN E	INTEN D	INTEN C	INTEN B	INTEN A

位	标记	位名	功能	读写
b31~b20	Reserved	-	读出时为“0”,写入时写“0”	R/W
b19	INTENSBD	专用向下计数中断使能B	0: 向下计数期间, SCMBR寄存器和计数值相等时, 该中断无效 1: 向下计数期间, SCMBR寄存器和计数值相等时, 该中断使能 注: 作为专用比较匹配事件输出时, 也由该位使能	R/W
b18	INTENSBU	专用向上计数中断使能B	0: 向上计数期间, SCMBR寄存器和计数值相等时, 该中断无效 1: 向上计数期间, SCMBR寄存器和计数值相等时, 该中断使能 注: 作为专用比较匹配事件输出时, 也由该位使能	R/W
b17	INTENSAD	专用向下计数中断使能A	0: 向下计数期间, SCMAR寄存器和计数值相等时, 该中断无效 1: 向下计数期间, SCMAR寄存器和计数值相等时, 该中断使能 注: 作为专用比较匹配事件输出时, 也由该位使能	R/W
b16	INTENSAU	专用向上计数中断使能A	0: 向上计数期间, SCMAR寄存器和计数值相等时, 该中断无效 1: 向上计数期间, SCMAR寄存器和计数值相等时, 该中断使能 注: 作为专用比较匹配事件输出时, 也由该位使能	R/W
b15~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8	INTENDTE	死区时间错误中断使能	0: 死区时间错误时, 该中断无效 1: 死区时间错误时, 该中断使能	R/W
b7	INTENUDF	下溢中断使能	0: 锯齿波时下溢发生或三角波时计数到谷点, 该中断无效 1: 锯齿波时下溢发生或三角波时计数到谷点, 该中断使能	R/W
b6	INTENOVF	上溢中断使能	0: 锯齿波时上溢发生或三角波时计数到峰点, 该中断无效 1: 锯齿波时上溢发生或三角波时计数到峰点, 该中断使能	R/W
b5	INTENF	计数匹配中断使能F	0: GCMFR寄存器与计数值相等时, 该中断无效 1: GCMFR寄存器与计数值相等时, 该中断使能	R/W
b4	INTENE	计数匹配中断使能E	0: GCMER寄存器与计数值相等时, 该中断无效 1: GCMER寄存器与计数值相等时, 该中断使能	R/W
b3	INTEND	计数匹配中断使能D	0: GCMDR寄存器与计数值相等时, 该中断无效 1: GCMDR寄存器与计数值相等时, 该中断使能	R/W
b2	INTENC	计数匹配中断使能C	0: GCMCR寄存器与计数值相等时, 该中断无效 1: GCMCR寄存器与计数值相等时, 该中断使能	R/W
b1	INTENB	计数匹配中断使能B	0: GCMBR寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断无效	R/W

1: GCMBR寄存器与计数值相等时，或者发生捕获输入事件时，

该中断使能

b0 INTENA 计数匹配中断使能A 0: GCMAR寄存器与计数值相等时，或者发生捕获输入事件时， R/W

该中断无效

1: GCMAR寄存器与计数值相等时，或者发生捕获输入事件时，

该中断使能

18.5.8 端口控制寄存器 (TMR6_PCONR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	EMBVAL B[1:0]	-	-	OUT ENB	PER CB[1:0]	CMP CB[1:0]	STASTP SB	STP CB	STA CB	CAP MDB			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	EMBVAL A[1:0]	-	-	OUT ENA	PER CA[1:0]	CMP CA[1:0]	STASTP SA	STP CA	STA CA	CAP MDA			

位	标记	位名	功能	读写
b31~b29	Reserved	-	读出时为“0”,写入时写“0”	R/W
b28~b27	EMBVALB[1:0]	EMB状态控制B	00: EMB条件成立时, TIM6_<=>_PWMB端口正常输出 01: EMB条件成立时, TIM6_<=>_PWMB端口输出高阻态 10: EMB条件成立时, TIM6_<=>_PWMB端口输出低电平 11: EMB条件成立时, TIM6_<=>_PWMB端口输出高电平	R/W
b26~b25	Reserved	-	读出时为“0”,写入时写“0”	R/W
b24	OUTENB	输出使能B	0: Timer6功能时的TIM6_<=>_PWMB端口输出无效 1: Timer6功能时的TIM6_<=>_PWMB端口输出有效	R/W
b23~b22	PERCB[1:0]	周期值匹配时端口 状态设定B	00: 定时器计数值与周期值相等时, TIM6_<=>_PWMB端口输出设 定为低电平 01: 定时器计数值与周期值相等时, TIM6_<=>_PWMB端口输出设 定为高电平 10: 定时器计数值与周期值相等时, TIM6_<=>_PWMB端口输出保 持先前状态 11: 定时器计数值与周期值相等时, TIM6_<=>_PWMB端口输出设 定为反转电平	R/W
b21~b20	CMPCB[1:0]	比较值匹配时端口 状态设定B	00: 定时器计数值与GCMBR相等时, TIM6_<=>_PWMB端口输出 设定为低电平 01: 定时器计数值与GCMBR相等时, TIM6_<=>_PWMB端口输出 设定为高电平 10: 定时器计数值与GCMBR相等时, TIM6_<=>_PWMB端口输出 保持先前状态 11: 定时器计数值与GCMBR相等时, TIM6_<=>_PWMB端口输出 设定为反转电平	R/W
b19	STASTPSB	计数开始停止端口 状态选择B	0: 计数开始或停止时, TIM6_<=>_PWMB端口输出由STACB、 STPCB决定 1: 计数开始或停止时, TIM6_<=>_PWMB端口输出保持先前状态 注: 此处的计数开始是指初始计数开始或停止再开始; 计数停止是 指初始时停止或计数开始后再停止	R/W
b18	STPCB	计数停止端口状态	0: 计数停止时, TIM6_<=>_PWMB端口输出设定为低电平	R/W

		设定B	1: 计数停止时, TIM6_<>> PWMB端口输出设定为高电平	
b17	STACB	计数开始端口状态	0: 计数开始时, TIM6_<>> PWMB端口输出设定为低电平	R/W
		设定B	1: 计数开始时, TIM6_<>> PWMB端口输出设定为高电平	
b16	CAPMDB	功能模式选择B	0: 比较输出功能	R/W
			1: 捕获输入功能	
b15~b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12~b11	EMBVALA[1:0]	EMB状态控制A	00: EMB条件成立时, TIM6_<>> PWMA端口正常输出 01: EMB条件成立时, TIM6_<>> PWMA端口输出高阻态 10: EMB条件成立时, TIM6_<>> PWMA端口输出低电平 11: EMB条件成立时, TIM6_<>> PWMA端口输出高电平	R/W
b10~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8	OUTENA	输出使能A	0: Timer6功能时的TIM6_<>> PWMA端口输出无效 1: Timer6功能时的TIM6_<>> PWMA端口输出有效	R/W
b7~b6	PERCA[1:0]	周期值匹配时端口 状态设定A	00: 定时器计数值与周期值相等时, TIM6_<>> PWMA端口输出设 定为低电平 01: 定时器计数值与周期值相等时, TIM6_<>> PWMA端口输出设 定为高电平 10: 定时器计数值与周期值相等时, TIM6_<>> PWMA端口输出保 持先前状态 11: 定时器计数值与周期值相等时, TIM6_<>> PWMA端口输出设 定为反转电平	R/W
b5~b4	CMPCA[1:0]	比较值匹配时端口 状态设定A	00: 定时器计数值与GCMAR相等时, TIM6_<>> PWMA端口输出 设定为低电平 01: 定时器计数值与GCMAR相等时, TIM6_<>> PWMA端口输出 设定为高电平 10: 定时器计数值与GCMAR相等时, TIM6_<>> PWMA端口输出 保持先前状态 11: 定时器计数值与GCMAR相等时, TIM6_<>> PWMA端口输出 设定为反转电平	R/W
b3	STASTPSA	计数开始停止端口 状态选择A	0: 计数开始或停止时, TIM6_<>> PWMA端口输出由STACA、 STPCA决定 1: 计数开始或停止时, TIM6_<>> PWMA端口输出保持先前状态 注: 此处的计数开始是指初始计数开始或停止再开始; 计数停止是 指初始时停止或计数开始后再停止	R/W
b2	STPCA	计数停止端口状态 设定A	0: 计数停止时, TIM6_<>> PWMA端口输出设定为低电平 1: 计数停止时, TIM6_<>> PWMA端口输出设定为高电平	R/W
b1	STACA	计数开始端口状态 设定A	0: 计数开始时, TIM6_<>> PWMA端口输出设定为低电平 1: 计数开始时, TIM6_<>> PWMA端口输出设定为高电平	R/W
b0	CAPMDA	功能模式选择A	0: 比较输出功能 1: 捕获输入功能	R/W

18.5.9 缓存控制寄存器 (TMR6_BCONR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	BTRSPB[1:0]	-	-	BSE SPB	BEN SPB	-	-	BTRSPA[1:0]	-	-	BSE SPA	BEN SPA		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	BSE P	BEN P	-	-	-	-	BSE B	BEN B	BSE A	BEN A

位	标记	位名	功能	读写
b31~b30	Reserved	-	读出时为“0”,写入时写“0”	R/W
b29~28	BTRSPB[1:0]	专用比较基准值	锯齿波时	R/W
		缓存传送时间点	00: 不传送	
		B	00以外: 上溢或下溢时传送 三角波时 00: 不传送	
			01: 计数到峰点时缓存传送 10: 计数到谷点时缓存传送 11: 计数到峰点或谷点时缓存传送	
b27-b26	Reserved	-	读出时为“0”,写入时写“0”	R/W
b25	BSESPB	专用比较基准值	0: 单缓存传送 (SCMDR->SCMBR)	R/W
		缓存传送选择B	1: 双缓存传送 (SCMFR->SCMDR->SCMBR)	
b24	BENSPB	专用比较基准值	0: 缓存传送无效	R/W
		缓存传送B	1: 缓存传送使能	
b23~b22	Reserved	-	读出时为“0”,写入时写“0”	R/W
b21~b20	BTRSPA[1:0]	专用比较基准值	锯齿波时	R/W
		缓存传送时间点	00: 不传送	
		A	00以外: 上溢或下溢时传送 三角波时 00: 不传送	
			01: 计数到峰点时缓存传送 10: 计数到谷点时缓存传送 11: 计数到峰点或谷点时缓存传送	
b19~b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17	BSESPA	专用比较基准值	0: 单缓存传送 (SCMCR->SCMAR)	R/W
		缓存传送选择A	1: 双缓存传送 (SCMER->SCMCR->SCMAR)	
b16	BENSPA	专用比较基准值	0: 缓存传送无效	R/W
		缓存传送A	1: 缓存传送使能	
b15~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9	BSEP	周期值缓存传送	0: 单缓存传送 (PERBR->PERAR)	R/W

		选择	1: 双缓存传送 (PERCR->PERBR->PERAR) 注: 传送时间点跟计数模式无关, 只在锯齿波的上溢点、下溢点或 三角波的波谷	
b8	BENP	周期值缓存传送	0: 缓存传送无效 1: 缓存传送使能	R/W
b7~b4	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b3	BSEB	通用比较值缓存 传送选择B	比较输出功能时: 0: 单缓存传送 (GCMRD->GCMBR) 1: 双缓存传送 (GCMFR->GCMRD->GCMBR) 捕获输入功能时: 0: 单缓存传送 (GCMBR->GCMRD) 1: 双缓存传送 (GCMBR->GCMRD->GCMFR)	R/W
b2	BENB	通用比较值缓存 传送B	0: 缓存传送无效 1: 缓存传送使能	R/W
b1	BSEA	通用比较值缓存 传送选择A	比较输出功能时: 0: 单缓存传送 (GCMCR->GCMAR) 1: 双缓存传送 (GCMER->GCMCR->GCMAR) 捕获输入功能时: 0: 单缓存传送 (GCMAR->GCMCR) 1: 双缓存传送 (GCMAR->GCMCR->GCMER)	R/W
b0	BENA	通用比较值缓存 传送A	0: 缓存传送无效 1: 缓存传送使能	R/W

18.5.10 死区控制寄存器 (TMR6_DCONR)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8	SEPA	分离设定	0: DTUAR和DTDAR分别设定 1: DTDAR的值和DTUAR的值自动相等	R/W
b7~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5	DTBEND	死区时间值缓存 传送D	0: 缓存传送无效 1: 缓存传送使能 (DTDBR->DTDAR)	R/W
b4	DTBENU	死区时间值缓存 传送U	0: 缓存传送无效 1: 缓存传送使能 (DTUBR->DTUAR)	R/W
b3~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	DTCEN	死区功能	0: 死区功能无效 1: 死区功能有效	R/W

18.5.11 滤波控制寄存器 (TMR6_FCONR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								NOFI CKTB[1:0]	NOFI ENTB	-	NOFI CKTA[1:0]	NOFI ENTA			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								NOFI CKGB[1:0]	NOFI ENGB	-	NOFI CKGA[1:0]	NOFI ENGA			

位	标记	位名	功能	读写
b31~b23	Reserved	-	读出时为“0”,写入时写“0”	R/W
b22~b21	NOFICKTB[1:0]	滤波采样基准时钟选择TB	00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64	R/W
b20	NOFIENTB	捕获输入端口滤波TB	0: TIM6_TRIGB端口输入滤波功能无效 1: TIM6_TRIGB端口输入滤波功能使能	R/W
b19	Reserved	-	读出时为“0”,写入时写“0”	R/W
b18~b17	NOFICKTA[1:0]	滤波采样基准时钟选择TA	00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64	R/W
b16	NOFIENTA	捕获输入端口滤波TA	0: TIM6_TRIGA端口输入滤波功能无效 1: TIM6_TRIGA端口输入滤波功能使能	R/W
b15~b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6~b5	NOFICKGB[1:0]	滤波采样基准时钟选择GB	00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64	R/W
b4	NOFIENGB	捕获输入端口滤波GB	0: 该单元TIM6_<>_PWMB输入端口滤波功能无效 1: 该单元TIM6_<>_PWMB输入端口滤波功能使能	R/W
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~b1	NOFICKGA[1:0]	滤波采样基准时钟选择GA	00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64	R/W
b0	NOFIENGA	捕获输入端口滤波GA	0: 该单元TIM6_<>_PWMA输入端口滤波功能无效 1: 该单元TIM6_<>_PWMA输入端口滤波功能使能	R/W

18.5.12 有效周期寄存器 (TMR6_VPERR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Reserved										PCNTS[2:0]	PCNTE[1:0]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Reserved					SP PERIB	SP PERIA	Reserved									

位	标记	位名	功能	读写
b31~b21	Reserved	-	读出时为“0”,写入时写“0”	R/W
b20~b18	PCNTS[2:0]	有效周期选择	000: 有效周期选择功能无效 001: 每隔1个周期有效一次 010: 每隔2个周期有效一次 011: 每隔3个周期有效一次 100: 每隔4个周期有效一次 101: 每隔5个周期有效一次 110: 每隔6个周期有效一次 111: 每隔7个周期有效一次	R/W
b17~b16	PCNTE[1:0]	有效周期计数条件选择	00: 有效周期选择功能无效 01: 锯齿波计数上、下溢点或三角波波谷做为计数条件 10: 锯齿波计数上、下溢点或三角波波峰做为计数条件 11: 锯齿波计数上、下溢点或三角波波谷、波峰做为计数条件	R/W
b15~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9	SPPERIB	专用信号有效周期选择B	0: 有效周期选择功能无效 1: 有效周期选择功能使能	R/W
b8	SPPERIA	专用信号有效周期选择A	0: 有效周期选择功能无效 1: 有效周期选择功能使能	R/W
b7~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

18.5.13 状态标志寄存器 (TMR6_STFLR)

复位值: 0x8000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
DIRF	Reserved							VPERNUM[2:0]			Reserved					
-	-	-	CMS BDF	CMS BUF	CMS ADF	CMS AUF	DTE F	UDF F	OVF F	CMF F	CME F	CMD F	CMC F	CMB F	CMA F	b0

位	标记	位名	功能	读写
b31	DIRF	计数方向	0: 递减计数 1: 递加计数	R
b30-b24	Reserved	-	读出时为“0”, 写入时写“0”	R
b23~b21	VPERNUM[2:0]	周期次数	有效周期选择功能使能时, 计数后的周期次数	R
b20~b13	Reserved	-	读出时为“0”, 写入时写“0”	R
b12	CMSBDF	向下计数专用比较基	0: 向下计数时, SCMBR寄存器的值与计数值不相等 1: 向下计数时, SCMBR寄存器的值与计数值相等	R/W
b11	CMSBUF	向上计数专用比较基	0: 向上计数时, SCMBR寄存器的值与计数值不相等 1: 向上计数时, SCMBR寄存器的值与计数值相等	R/W
b10	CMSADF	向下计数专用比较基	0: 向下计数时, SCMAR寄存器的值与计数值不相等 1: 向下计数时, SCMAR寄存器的值与计数值相等	R/W
b09	CMSAUF	向上计数专用比较基	0: 向上计数时, SCMAR寄存器的值与计数值不相等 1: 向上计数时, SCMAR寄存器的值与计数值相等	R/W
b8	DTEF	死区时间错误	0: 未发生死区时间错误 1: 发生死区时间错误	R
b7	UDFF	下溢匹配	0: 未发生锯齿波下溢或三角波计数到谷点 1: 发生锯齿波下溢或三角波计数到谷点	R/W
b6	OVFF	上溢匹配	0: 未发生锯齿波上溢或三角波计数到峰点 1: 发生锯齿波上溢或三角波计数到峰点	R/W
b5	CMFF	计数匹配F	0: GCMFR寄存器的值与计数值不相等 1: GCMFR寄存器的值与计数值相等	R/W
b4	CMEF	计数匹配E	0: GCMER寄存器的值与计数值不相等 1: GCMER寄存器的值与计数值相等	R/W
b3	CMDF	计数匹配D	0: GCMDR寄存器的值与计数值不相等 1: GCMDR寄存器的值与计数值相等	R/W
b2	CMCF	计数匹配C	0: GCMCR寄存器的值与计数值不相等 1: GCMCR寄存器的值与计数值相等	R/W
b1	CMBF	计数匹配B	0: GCMBR寄存器的值与计数值不相等, 且未发生 TIM6_<1>_PWMB捕获完成动作 1: GCMBR寄存器的值与计数值相等, 或发生TIM6_<1>_PWMB 捕获完成动作	R/W

b0	CMAF	计数匹配A	0: GCMAR寄存器的值与计数值不相等，且未发生 TIM6_<t>_PWMA捕获完成动作 1: GCMAR寄存器的值与计数值相等，或发生TIM6_<t>_PWMA 捕获完成动作	R/W
----	------	-------	--	-----

18.5.14 硬件启动事件选择寄存器 (TMR6_HSTAR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
STA RTS	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HSTA 11	HSTA 10	HSTA 9	HSTA 8	HSTA 7	HSTA 6	HSTA 5	HSTA 4	-	-	HSTA 1	HSTA 0

位	标记	位名	功能	读写
b31	STARTS	硬件启动使能	0: 硬件启动无效 1: 硬件启动有效 注: 硬件启动有效时, SSTAR的设定无效	R/W
b30~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	HSTA11	硬件启动条件11	条件: TIM6_TRIGB端口上采样到下降沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W
b10	HSTA10	硬件启动条件10	条件: TIM6_TRIGB端口上采样到上升沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W
b9	HSTA9	硬件启动条件9	条件: TIM6_TRIGA端口上采样到下降沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W
b8	HSTA8	硬件启动条件8	条件: TIM6_TRIGA端口上采样到上升沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W
b7	HSTA7	硬件启动条件7	条件: TIM6_ $\triangleleft\triangleright$ _PWMB端口上采样到下降沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W
b6	HSTA6	硬件启动条件6	条件: TIM6_ $\triangleleft\triangleright$ _PWMB端口上采样到上升沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W
b5	HSTA5	硬件启动条件5	条件: TIM6_ $\triangleleft\triangleright$ _PWMA端口上采样到下降沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W
b4	HSTA4	硬件启动条件4	条件: TIM6_ $\triangleleft\triangleright$ _PWMA端口上采样到上升沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	R/W
b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	HSTA1	硬件启动条件1	条件: 内部硬件触发事件1有效	R/W

0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效				
b0	HSTA0	硬件启动条件0	条件: 内部硬件触发事件0有效	R/W
			0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	

18.5.15 硬件停止事件选择寄存器 (TMR6_HSTPR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
STOPS	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HSTP 11	HSTP 10	HSTP 9	HSTP 8	HSTP 7	HSTP 6	HSTP 5	HSTP 4	-	-	HSTP 1	HSTP 0

位	标记	位名	功能	读写
b31	STOPS	硬件停止使能	0: 硬件停止无效 1: 硬件停止有效 注: 硬件停止有效时, SSTPR的设定无效	R/W
b30~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	HSTP11	硬件停止条件11	条件: TIM6_TRIGB端口上采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b10	HSTP10	硬件停止条件10	条件: TIM6_TRIGB端口上采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b9	HSTP9	硬件停止条件9	条件: TIM6_TRIGA端口上采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b8	HSTP8	硬件停止条件8	条件: TIM6_TRIGA端口上采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b7	HSTP7	硬件停止条件7	条件: TIM6_<>_PWMB端口上采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b6	HSTP6	硬件停止条件6	条件: TIM6_<>_PWMB端口上采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b5	HSTP5	硬件停止条件5	条件: TIM6_<>_PWMA端口上采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b4	HSTP4	硬件停止条件4	条件: TIM6_<>_PWMA端口上采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	HSTP1	硬件停止条件1	条件: 内部硬件触发事件1有效 0: 条件匹配时, 硬件停止无效	R/W

1: 条件匹配时, 硬件停止有效				
b0	HSTP0	硬件停止条件0	条件: 内部硬件触发事件0有效	R/W
			0: 条件匹配时, 硬件停止无效	
			1: 条件匹配时, 硬件停止有效	

18.5.16 硬件清零事件选择寄存器 (TMR6_HCLRR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CLEARS	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HCLE 11	HCLE 10	HCLE 9	HCLE 8	HCLE 7	HCLE 6	HCLE 5	HCLE 4	-	-	HCLE 1	HCLE 0

位	标记	位名	功能	读写
b31	CLEARS	硬件清零使能	0: 硬件清零无效 1: 硬件清零有效 注: 硬件清零有效时, SCLRR的设定无效	R/W
b30~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	HCLE11	硬件清零条件11	条件: TIM6_TRIGB端口上采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b10	HCLE10	硬件清零条件10	条件: TIM6_TRIGB端口上采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b9	HCLE9	硬件清零条件9	条件: TIM6_TRIGA端口上采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b8	HCLE8	硬件清零条件8	条件: TIM6_TRIGA端口上采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b7	HCLE7	硬件清零条件7	条件: TIM6_ $\triangleleft\triangleright$ _PWMB端口上采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b6	HCLE6	硬件清零条件6	条件: TIM6_ $\triangleleft\triangleright$ _PWMB端口上采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b5	HCLE5	硬件清零条件5	条件: TIM6_ $\triangleleft\triangleright$ _PWMA端口上采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b4	HCLE4	硬件清零条件4	条件: TIM6_ $\triangleleft\triangleright$ _PWMA端口上采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	HCLE1	硬件清零条件1	条件: 内部硬件触发事件1有效 0: 条件匹配时, 硬件清零无效	R/W

1: 条件匹配时, 硬件清零有效				
b0	HCLE0	硬件清零条件0	条件: 内部硬件触发事件0有效	R/W
			0: 条件匹配时, 硬件清零无效	
			1: 条件匹配时, 硬件清零有效	

18.5.17 硬件捕获事件选择寄存器 (TMR6_HCPAR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HCP A11	HCP A10	HCP A9	HCP A8	HCP A7	HCP A6	HCP A5	HCP A4	-	-	HCP A1	HCP A0

位	标记	位名	功能	读写
b31~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	HCPA11	硬件捕获A条件11	条件: TIM6_TRIGB端口上采样到下降沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W
b10	HCPA10	硬件捕获A条件10	条件: TIM6_TRIGB端口上采样到上升沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W
b9	HCPA9	硬件捕获A条件9	条件: TIM6_TRIGA端口上采样到下降沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W
b8	HCPA8	硬件捕获A条件8	条件: TIM6_TRIGA端口上采样到上升沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W
b7	HCPA7	硬件捕获A条件7	条件: TIM6_ $\triangleleft\triangleright$ _PWMB端口上采样到下降沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W
b6	HCPA6	硬件捕获A条件6	条件: TIM6_ $\triangleleft\triangleright$ _PWMB端口上采样到上升沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W
b5	HCPA5	硬件捕获A条件5	条件: TIM6_ $\triangleleft\triangleright$ _PWMA端口上采样到下降沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W
b4	HCPA4	硬件捕获A条件4	条件: TIM6_ $\triangleleft\triangleright$ _PWMA端口上采样到上升沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W
b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	HCPA1	硬件捕获A条件1	条件: 内部硬件触发事件1有效 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效	R/W
b0	HCPA0	硬件捕获A条件0	条件: 内部硬件触发事件0有效 0: 条件匹配时, 硬件捕获A无效	R/W

1: 条件匹配时，硬件捕获A有效

18.5.18 硬件捕获事件选择寄存器 (TMR6_HCPBR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HCP B11	HCP B10	HCP B9	HCP B8	HCP B7	HCP B6	HCP B5	HCP B4	-	-	HCP B1	HCP B0

位	标记	位名	功能	读写
b31~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	HCPB11	硬件捕获B条件11	条件: TIM6_TRIGB端口上采样到下降沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W
b10	HCPB10	硬件捕获B条件10	条件: TIM6_TRIGB端口上采样到上升沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W
b9	HCPB9	硬件捕获B条件9	条件: TIM6_TRIGA端口上采样到下降沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W
b8	HCPB8	硬件捕获B条件8	条件: TIM6_TRIGA端口上采样到上升沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W
b7	HCPB7	硬件捕获B条件7	条件: TIM6_<*>_PWMB端口上采样到下降沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W
b6	HCPB6	硬件捕获B条件6	条件: TIM6_<*>_PWMB端口上采样到上升沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W
b5	HCPB5	硬件捕获B条件5	条件: TIM6_<*>_PWMA端口上采样到下降沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W
b4	HCPB4	硬件捕获B条件4	条件: TIM6_<*>_PWMA端口上采样到上升沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W
b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	HCPB1	硬件捕获B条件1	条件: 内部硬件触发事件1有效 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效	R/W
b0	HCPB0	硬件捕获B条件0	条件: 内部硬件触发事件0有效 0: 条件匹配时, 硬件捕获B无效	R/W

1: 条件匹配时，硬件捕获B有效

18.5.19 硬件递加事件选择寄存器 (TMR6_HCUPR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved														HC UP17	HC UP16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HC UP11	HC UP10	HC UP9	HC UP8	HC UP7	HC UP6	HC UP5	HC UP4	HC UP3	HC UP2	HC UP1	HC UP0

位	标记	位名	功能	读写
b31~b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17	HCUP17	硬件递加条件17	条件: 内部硬件触发事件1有效 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b16	HCUP16	硬件递加条件16	条件: 内部硬件触发事件0有效 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	HCUP11	硬件递加条件11	条件: TIM6_TRIGB端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b10	HCUP10	硬件递加条件10	条件: TIM6_TRIGB端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b9	HCUP9	硬件递加条件9	条件: TIM6_TRIGA端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b8	HCUP8	硬件递加条件8	条件: TIM6_TRIGA端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b7	HCUP7	硬件递加条件7	条件: TIM6_<t>_PWMB端口为高电平时, TIM6_<t>_PWMA端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b6	HCUP6	硬件递加条件6	条件: TIM6_<t>_PWMB端口为高电平时, TIM6_<t>_PWMA端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b5	HCUP5	硬件递加条件5	条件: TIM6_<t>_PWMB端口为低电平时, TIM6_<t>_PWMA端口上采样到下降沿	R/W

			0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	
b4	HCUP4	硬件递加条件4	条件: TIM6_<>>_PWMB端口为低电平时, TIM6_<>>_PWMA端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b3	HCUP3	硬件递加条件3	条件: TIM6_<>>_PWMA端口为高电平时, TIM6_<>>_PWMB端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b2	HCUP2	硬件递加条件2	条件: TIM6_<>>_PWMA端口为高电平时, TIM6_<>>_PWMB端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b1	HCUP1	硬件递加条件1	条件: TIM6_<>>_PWMA端口为低电平时, TIM6_<>>_PWMB端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b0	HCUP0	硬件递加条件0	条件: TIM6_<>>_PWMA端口为低电平时, TIM6_<>>_PWMB端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W

18.5.20 硬件递减事件选择寄存器 (TMR6_HCDOR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved														HC DO17	HC DO16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
				HC DO11	HC DO10	HC DO9	HC DO8	HC DO7	HC DO6	HC DO5	HC DO4	HC DO3	HC DO2	HC DO1	HC DO0

位	标记	位名	功能	读写
b31~b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17	HCDO17	硬件递减条件17	条件: 内部硬件触发事件1有效 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b16	HCDO16	硬件递减条件16	条件: 内部硬件触发事件0有效 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	HCDO11	硬件递减条件11	条件: TIM6_TRIGB端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b10	HCDO10	硬件递减条件10	条件: TIM6_TRIGB端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b9	HCDO9	硬件递减条件9	条件: TIM6_TRIGA端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b8	HCDO8	硬件递减条件8	条件: TIM6_TRIGA端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b7	HCDO7	硬件递减条件7	条件: TIM6_<t>_PWMB端口为高电平时, TIM6_<t>_PWMA端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b6	HCDO6	硬件递减条件6	条件: TIM6_<t>_PWMB端口为高电平时, TIM6_<t>_PWMA端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b5	HCDO5	硬件递减条件5	条件: TIM6_<t>_PWMB端口为低电平时, TIM6_<t>_PWMA端口上采样到下降沿	R/W

			0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	
b4	HCDO4	硬件递减条件4	条件: TIM6_<>>_PWMB端口为低电平时, TIM6_<>>_PWMA端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b3	HCDO3	硬件递减条件3	条件: TIM6_<>>_PWMA端口为高电平时, TIM6_<>>_PWMB端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b2	HCDO2	硬件递减条件2	条件: TIM6_<>>_PWMA端口为高电平时, TIM6_<>>_PWMB端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b1	HCDO1	硬件递减条件1	条件: TIM6_<>>_PWMA端口为低电平时, TIM6_<>>_PWMB端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b0	HCDO0	硬件递减条件0	条件: TIM6_<>>_PWMA端口为低电平时, TIM6_<>>_PWMB端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W

18.5.21 软件同步启动控制寄存器 (TMR6_SSTAR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
SSTA3	SSTA2	SSTA1													

位	标记	位名	功能	读写
b31~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	SSTA3	单元3软件启动	0: 软件启动无效 1: 软件启动使能	R/W
b1	SSTA2	单元2软件启动	0: 软件启动无效 1: 软件启动使能	R/W
b0	SSTA1	单元1软件启动	0: 软件启动无效 1: 软件启动使能	R/W

18.5.22 软件同步停止控制寄存器 (TMR6_SSTPR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
SSTP3 SSTP2 SSTP1												SSTP3	SSTP2	SSTP1	

位	标记	位名	功能	读写
b31~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	SSTP3	单元3软件停止	0: 软件停止无效 1: 软件停止使能	R/W
b1	SSTP2	单元2软件停止	0: 软件停止无效 1: 软件停止使能	R/W
b0	SSTP1	单元1软件停止	0: 软件停止无效 1: 软件停止使能	R/W

18.5.23 软件同步清零控制寄存器 (TMR6_SCLRR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SCLE3 SCLE2 SCLE1												SCLE3	SCLE2	SCLE1	

位	标记	位名	功能	读写
b31~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	SCLE3	单元3软件清零	0: 软件清零无效 1: 软件清零使能	R/W
b1	SCLE2	单元2软件清零	0: 软件清零无效 1: 软件清零使能	R/W
b0	SCLE1	单元1软件清零	0: 软件清零无效 1: 软件清零使能	R/W

19 通用控制定时器 (Timer4)

19.1 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。

19.2 基本框图

Timer4 基本的功能及特性如表 19-1 所示。

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
	• 专用事件输出
	• EMB控制
中断类型	计数比较匹配中断
	计数周期匹配中断
	重载计数匹配中断

表 19-1 Timer4 的基本功能及特性

图 19-1 所示，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。

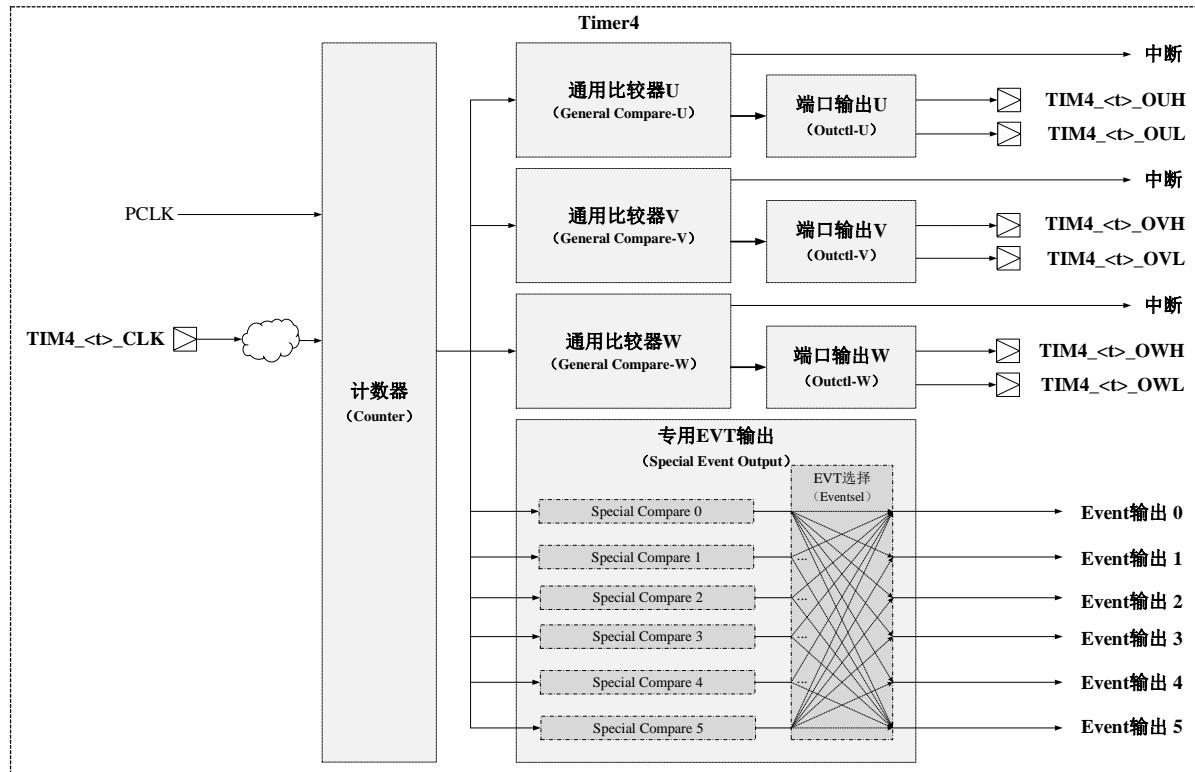


图 19-1 Timer4 基本框图

表 19-2 所示为 Timer4 的输入输出端口列表。

端口名	方向	功能
TIM4_<t>_CLK	in	计数时钟输入端口
TIM4_<t>_OUH		
TIM4_<t>_OUL		
TIM4_<t>_OVH		
TIM4_<t>_OVL		
TIM4_<t>_OWH		PWM输出端口
TIM4_<t>_OWL		

表 19-2 Timer4 端口列表

19.3 功能说明

19.3.1 基本动作

19.3.1.1 波形模式

Timer4 有 2 种基本计数波形模式，锯齿波模式和三角波模式。根据 CCSR.MODE 的设定可分别实现两种波形模式。下图 19-2 和图 19-3 分别是锯齿波和三角波的波形图。

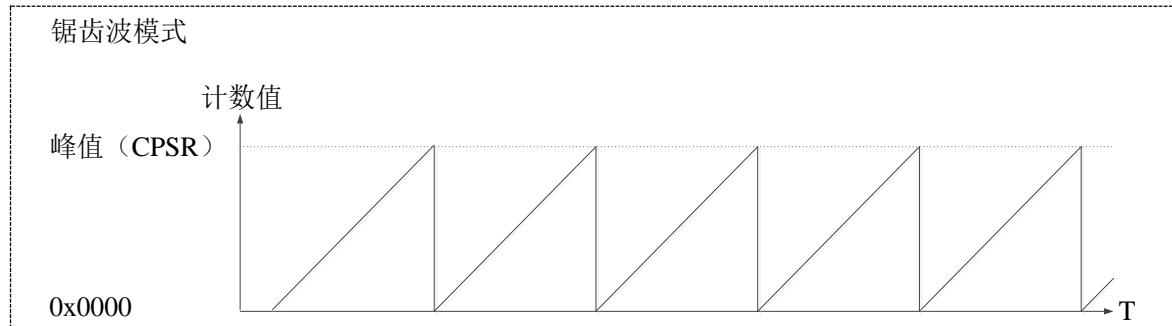


图 19-2 Timer4 锯齿波波形

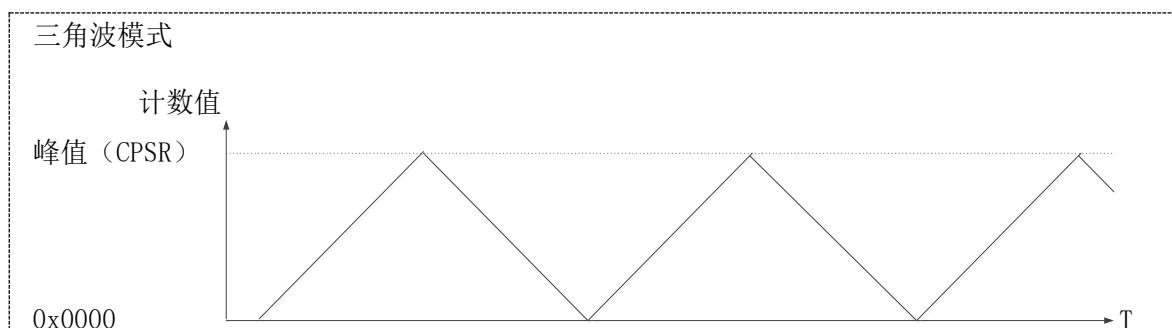


图 19-3 Timer4 三角波波形

19.3.1.2 计数动作

1) 锯齿波计数操作与控制流程如图 19-4 所示。

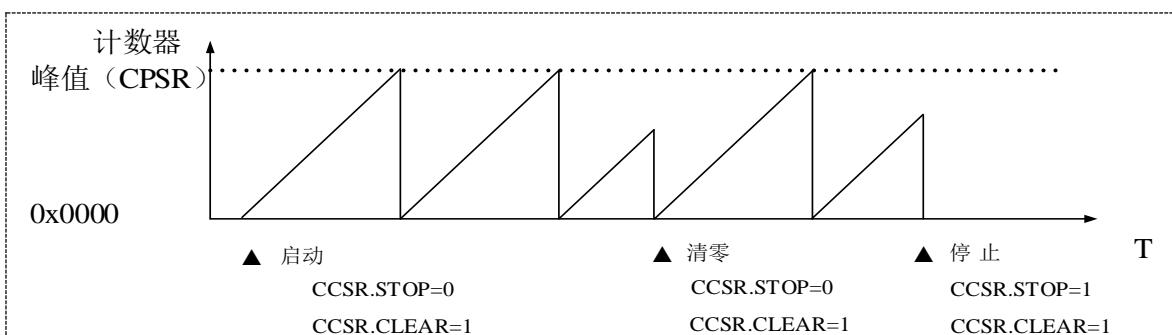


图 19-4 Timer4 锯齿波模式计数动作

a) 设置模式 CCSR.MODE=0。

- b) 设置计数峰值 CPSR 寄存器。
 - c) 写入 CCSR.STOP=0 和 CCSR.CLEAR=1, 计数器计数值 (CNTR) 被初始化到 0x0000 并启动计数操作。计数器值从 0x0000 开始递增计数, 当达到峰值 (CPSR), 计数值归 0x0000, 依次重复此操作。
 - d) 计数周期= (CPSR+1) × 计数时钟周期
 - e) 计数过程中, 写入 CCSR.STOP=0 和 CCSR.CLEAR=1 可以将计数值初始化到 0x0000 并继续进行计数操作。
 - f) 计数过程中, 写入 CCSR.STOP=1 和 CCSR.CLEAR=1 可以将计数值初始化到 0x0000 并停止计数操作。
- 2) 三角波计数操作与控制流程如图 19-5 所示。

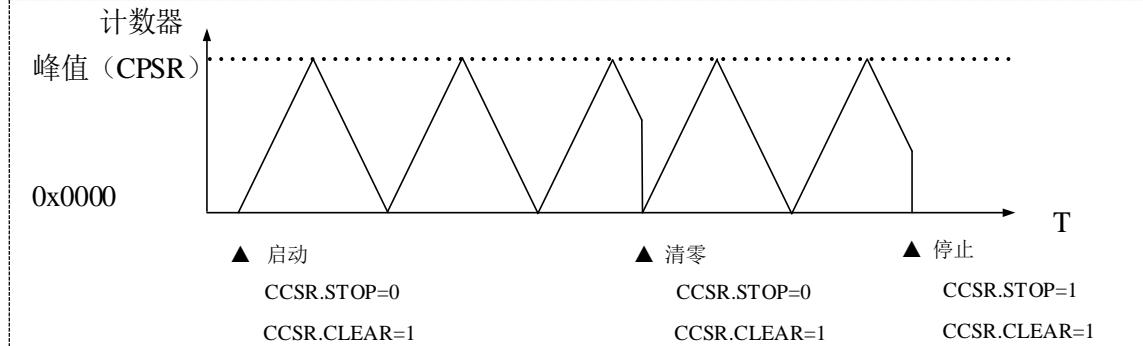


图 19-5 Timer4 三角波模式计数动作

- a) 设置模式 CCSR.MODE=1。
- b) 设置计数峰值 CPSR 寄存器。
- c) 写入 CCSR.STOP=0 和 CCSR.CLEAR=1, 计数器计数值 (CNTR) 被初始化到 0x0000 并启动计数操作。计数器值从 0x0000 开始做递增计数, 在到达计数峰值之前做递增计数; 当达到峰值 (CPSR) 时, 计数器开始做递减计数, 直到计数值归 0x0000; 之后重新进行递增计数操作, 依次重复此操作。
- d) 计数周期= (CPSR) × 2 × 计数时钟周期
- e) 计数过程中, 写入 CCSR.STOP=0 和 CCSR.CLEAR=1 可以将计数值初始化到 0x0000 并重新进行递增计数操作, 之后重复上述操作。
- f) 计数过程中, 写入 CCSR.STOP=1 和 CCSR.CLEAR=1 可以将计数值初始化到 0x0000 并停止计数操作。

19.3.1.3 比较输出

1) 下图 19-6 是锯齿波计数模式下的比较输出模块的波形输出示例。

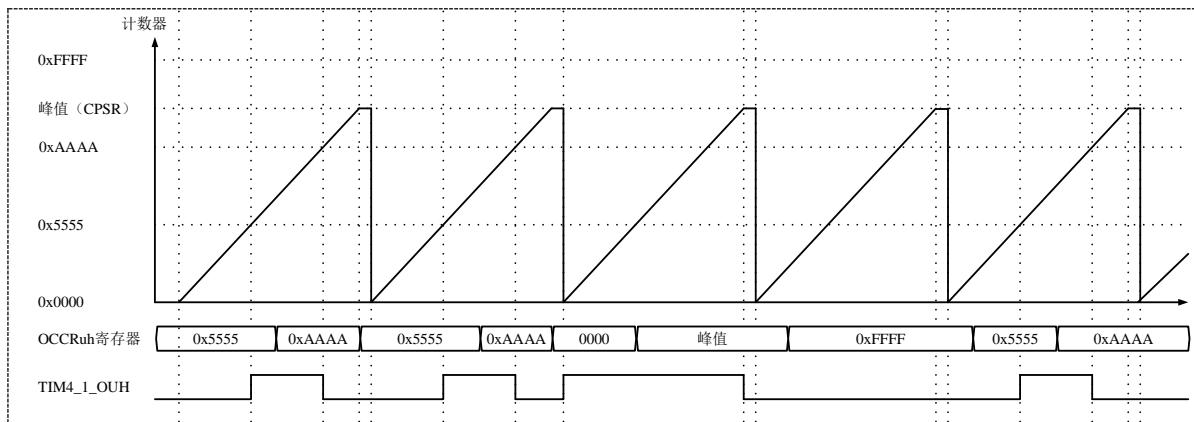


图 19-6 锯齿波模式波形输出例

2) 下图 19-7 是三角波计数模式下的比较输出模块的波形输出示例。

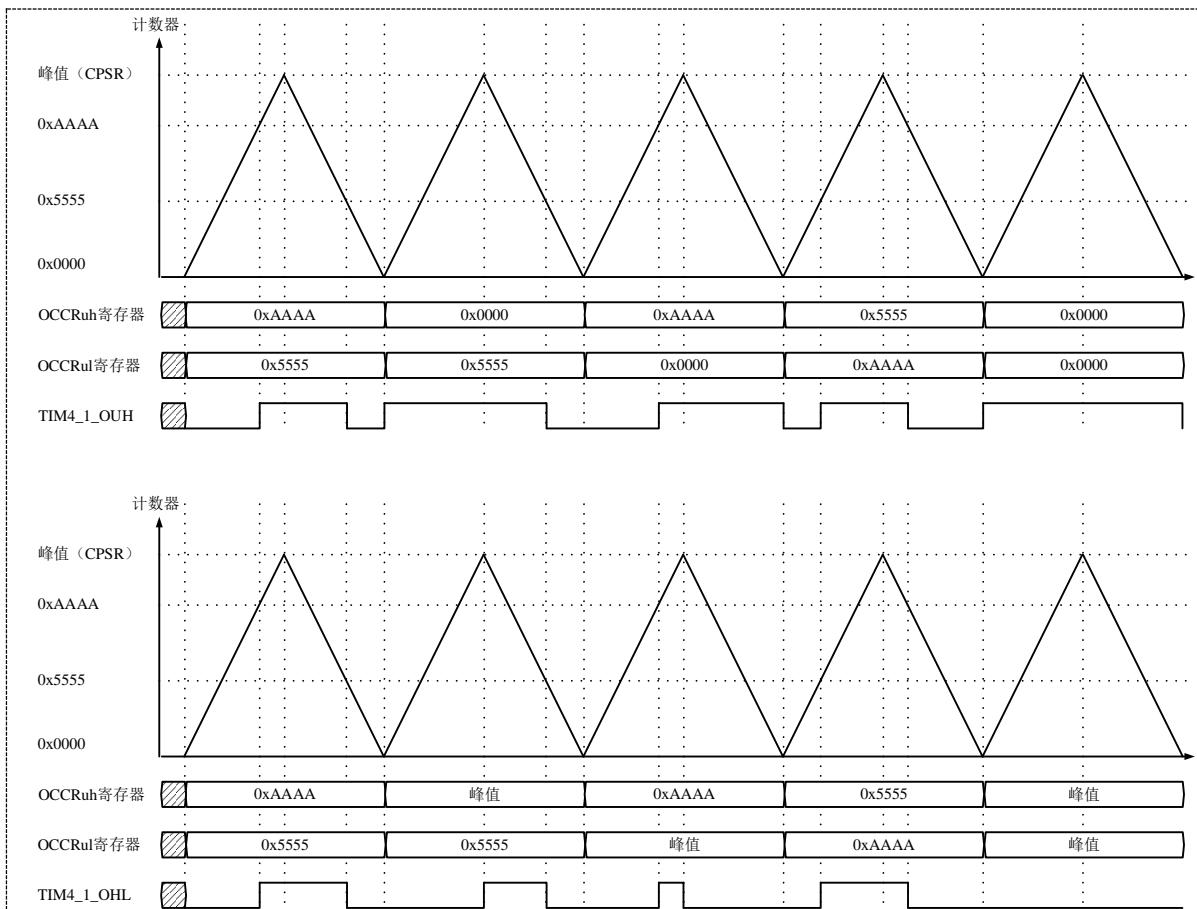


图 19-7 三角波模式波形输出例

19.3.2 缓存功能

Timer4 的周期基准寄存器 (CPSR)、通用比较基准寄存器 (OCCR)、通用模式控制寄存器 (OCMR)、专用比较基准寄存器 (SCCR) 和专用模式控制寄存器 (SCMR) 都具有缓存功能。

19.3.2.1 周期基准寄存器缓存功能

CPSR 具有缓存功能寄存器，写入的计数峰值数据先收纳到缓冲寄存器。在以下条件下，数据从缓冲器寄存器传输到 CPSR 寄存器中。

- a) 缓冲功能禁止时 (CCSR.BUFEN=0)，写入数据立即从缓冲寄存器传输到 CPSR 寄存器。
- b) 缓冲功能使能时 (CCSR.BUFEN=1)，在计数器停止时 (CCSR.STOP=1) 或者计数器的计数值为"0x0000"时，数据从缓冲寄存器传输到 CPSR 寄存器。

注意：

- 从 CPSR 读取数据时，读取的并非 CPSR 缓冲寄存器的值，而是 CPSR 寄存器的值。当缓冲功能使能时，在传输完成之前所读取的值不是最近写入的值，而是上一次写入的 CPSR 值。

图 19-8 所示，当锯齿波模式且禁止缓冲功能时修改计数峰值 CPSR 的操作。

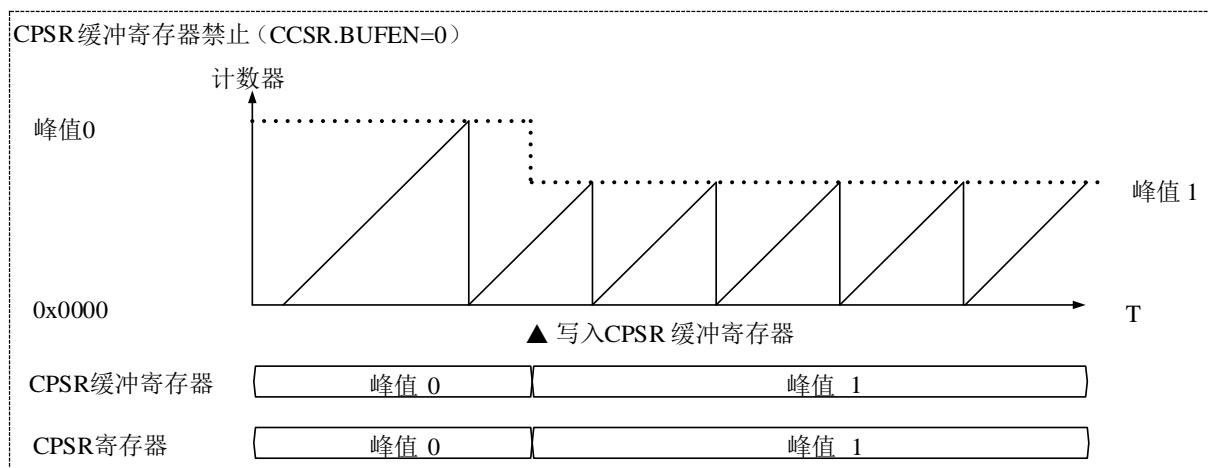


图 19-8 缓存无效时修改锯齿波计数周期

注意：

- 当禁止缓冲功能时，写入数据立即从缓冲寄存器传输到 CPSR 计数器，计数周期在写入操作完成后立即改变。在此情况下，如果写入的值小于计数器当前计数值，

计数器将持续进行递增计数，直到到达"0xFFFF"，因此，需特别注意此种状况。

图 19-9 所示，当锯齿波模式且使能缓冲功能时修改计数峰值 CPSR 的操作。

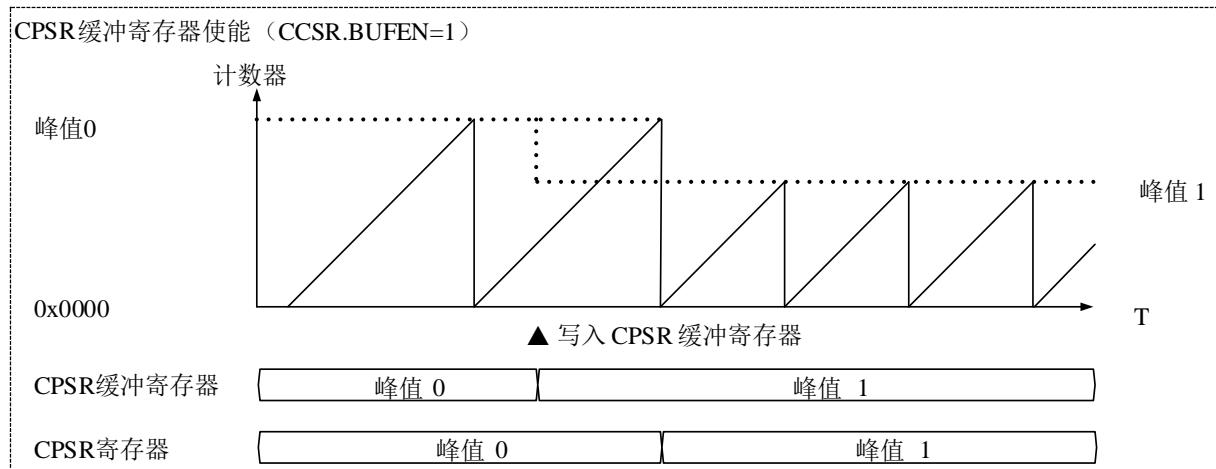


图 19-9 缓存使能时修改锯齿波计数周期

如图所示，当使能缓冲功能时，在计数器停止时或者计数器的计数值为"0x0000"时，写入的数据从缓冲寄存器传输到 CPSR 寄存器。

图 19-10 所示，当三角波模式且使能缓冲功能时修改计数峰值 CPSR 的操作。

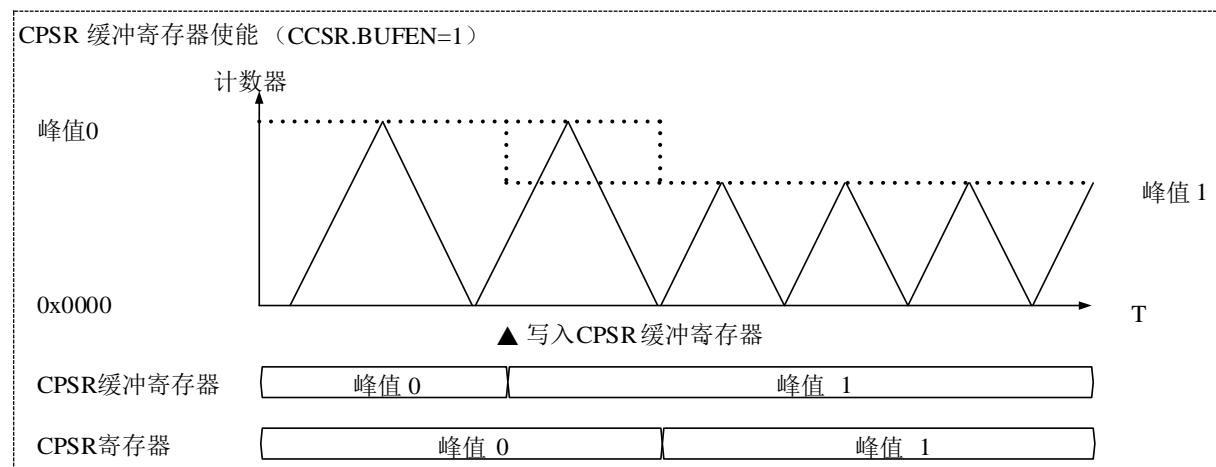


图 19-10 缓存使能时修改三角波计数周期

如图所示，三角波计数模式下，当使能缓冲功能时，在计数器停止时或者计数器的下一次计数值为"0x0000"时，写入的数据从缓冲寄存器传输到 CPSR 寄存器。计数器周期的改变在写入操作完成后的下一个计数器周期开始。

19.3.2.2 通用比较寄存器缓存功能

通用比较基准值寄存器（OCCR）和通用模式控制寄存器（OCMR）均有缓冲寄存器功能，当使能缓冲功能时，在指定的传输时刻传输加载到 OCCR 和 OCMR 寄存器。OCCR 缓冲功能可以用于计数过程中同步改变比较值，OCMR 缓冲功能可以用于计数过程中计数上溢点（锯齿波）、计数谷点或峰点（三角波）同步改变内部 PWM 输出。

- a) 当输出比较与计数器周期间隔响应功能的链接传输禁止时，缓冲值在设定的计数状态时加载到寄存器。此时加载情况与计数器周期间隔计数器无关。

图 19-11 所示，是通用输出比较 OCCR 缓冲功能使能，计数器零值时加载（OCER.CxBUFEN=01），计数器周期间隔响应链接禁止（OCER.LMCx=0）时的波形（ $x=L$ 或 H ）。

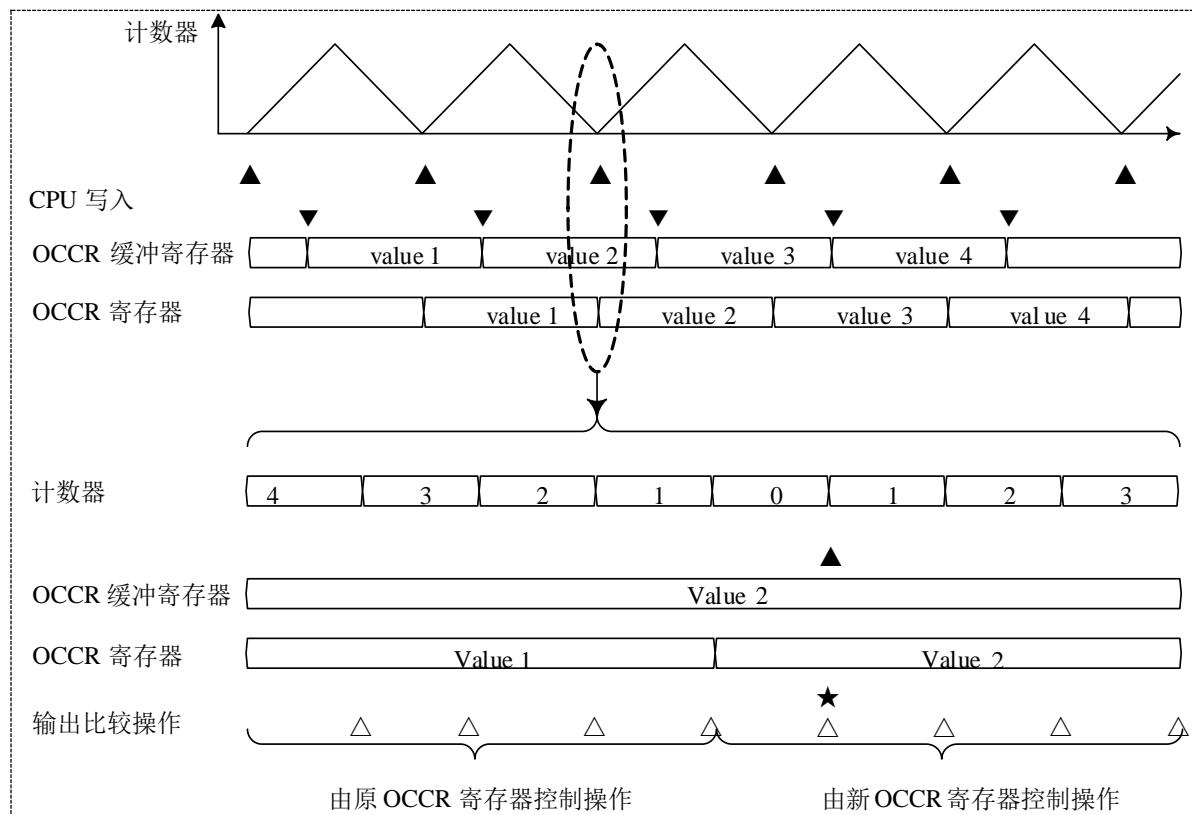


图 19-11 OCCR 缓冲数据传输（周期间隔响应链接禁止时）

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 Δ 时刻，输出比较根据设定的 OCCR 寄存器值与计数值匹配事件，执行改变 PWM 输出和置位 OCSR.OCFx 位 ($x=L$ 或 H)。在时刻★（计数值=0x0000）之后，端口输出根据新的 OCCR 数据执行操作。在时刻★之前根据原 OCCR 数据执行操作。

图中示意了 OCCR 缓冲寄存器在计数谷点的传输操作，OCMR 的缓冲寄存器的缓存传送操作与之类似；同理，在计数峰点的传输操作也类似。新的数据在传输时刻之后即刻生效（新写入数据将控制 PWM 输出和中断标志置位）。

- b) 当计数器周期间隔响应链接使能时，缓冲值在设定的计数状态且周期间隔计数器计数值为 0 时执行缓冲寄存器传输操作。**

图 19-12 所示，是通用输出比较 OCCR 缓冲功能使能、计数器零值时加载 (OCER.CxBUFEN=01)，计数器周期间隔响应链接使能 (OCER.LMCx=1) 时的波形 ($x=L$ 或 H)。

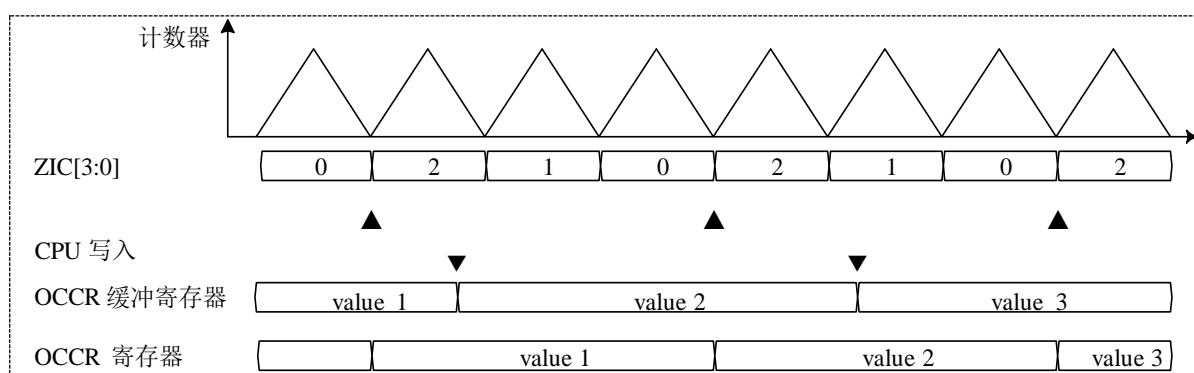


图 19-12 OCCR 缓冲数据传输（周期间隔响应链接使能）

计数器为三角波计数模式，周期间隔计数器 (CVPR.ZIC) 从 2 到 0 做递减计数，下溢中断在标志▲（计数谷点）时刻产生。在▼时刻，CPU 改写 OCCR 寄存器，写入的数据存储在 OCCR 缓冲寄存器中。之后当计数器计数值为 0x0000 且周期间隔计数器 (CVPR.ZIC) 为 0 时，进行数据加载操作，数据从缓冲寄存器到 OCCR 寄存器，同时产生中断标志 IRQZF。

图中示意了 OCCR 缓冲寄存器在计数谷点的传输操作，OCMR 的缓冲寄存器的缓存传送操作与之类似；同理，在计数峰点的传输操作也类似。新的数据在传输时刻之后即刻生效（新写入数据将控制 PWM 输出和中断标志置位）。

使用通道链接操作模式时，同时使能 OCCRuh 和 OCCRul 缓冲功能可以产生各种不同的 PWM 输出波形。图 19-13 示例了在输出比较寄存器 OCCRuh 和 OCCRul 保持不变的情况下，改变 OCMR 寄存器值来产生不同的输出波形 TIM4_<t>_OUL 方式的情形。

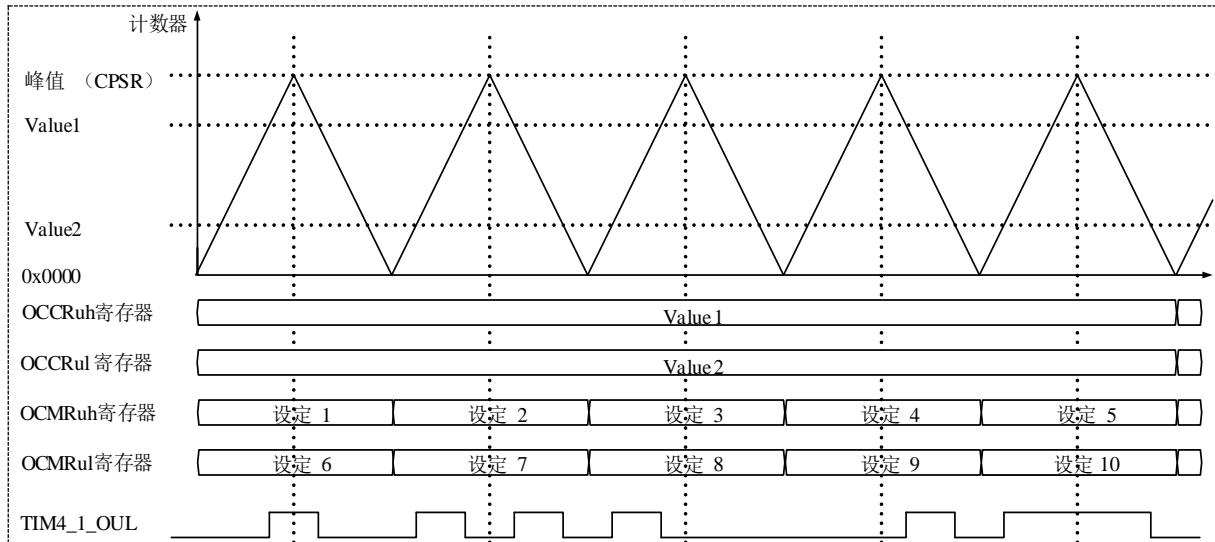


图 19-13 输出比较缓冲数据传输 (OCMR 缓冲使能)

19.3.2.3 专用比较寄存器缓存功能

专用比较基准寄存器 (SCCR) 和在专用模式控制寄存器 (SCMR) 都带有缓冲功能寄存器。当使能缓冲功能时，CPU 写入 SCCR 和 SCMR 缓冲寄存器的值在设定的计数器状态下加载到 SCCR 和 SCMR 寄存器。

- a) 当禁止计数器周期间隔响应链接传输时，缓冲传输操作仅与计数器状态有关，不受计数器周期间隔计数器影响。

图 19-14 所示，是使能寄存器缓冲功能，禁止计数器周期间隔响应链接传输 (SCSR.LMC=0)，计数器零值 (SCSR.BUFEN=01) 时传送到 SCCR 和 SCMR 寄存器。

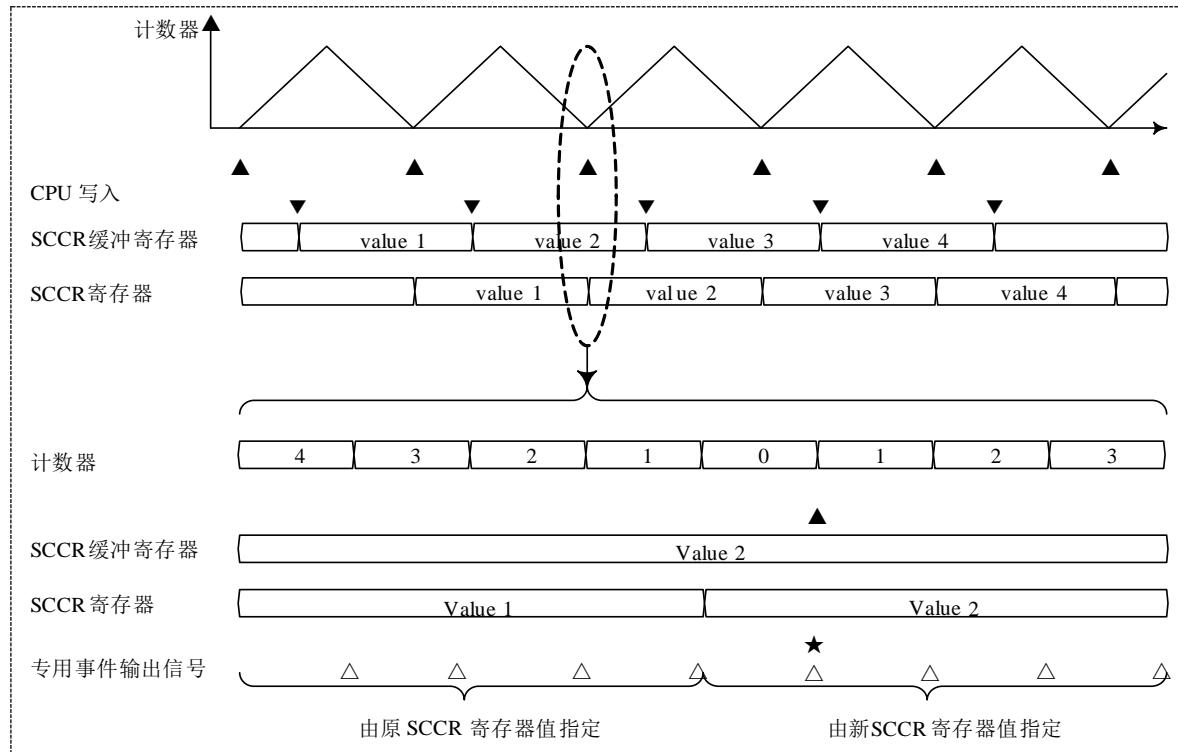


图 19-14 SCCR 缓冲传输操作（周期间隔响应链接传输禁止时）

图中的上半部分是全局示意图，下半部分缓冲寄存器传输操作的局部放大图。

计数器为三角波计数模式，下溢中断在标志▲（计数谷点）时刻产生。在▼时刻，CPU 改写 SCCR 寄存器，写入的数据存储在 SCCR 缓冲寄存器中。之后当计数器计数值为 0x0000 时，进行数据加载操作，数据从缓冲寄存器到 SCCR 寄存器，同时产生中断标志 IRQZF。

在△时刻，根据设定的 SCCR 寄存器值与计数值进行比较操作。在时刻★（计数值 =0x0000）之后，专用事件输出信号根据新的 SCCR 数据执行操作。在时刻★之前根据 SCCR 数据执行操作。

图中示意了 SCCR 缓冲寄存器在计数谷点的传输操作，SCMR 的缓冲寄存器的缓存传送操作与之类似；同理，在计数峰点的传输操作也类似。新的数据在传输时刻之后即刻生效（新写入数据将专用事件输出信号和中断标志置位）。

b) 当计数器周期间隔响应链接使能时，缓冲值在设定的计数状态且周期间隔计数器计数值为 0 时执行缓冲寄存器传输操作。

图 19-15 所示，是使能 SCCR 缓冲功能、计数器零值时加载（SCSR.BUFEN=01），计数器周期间隔响应链接传输使能时（SCSR.LMC=1）的示意图。

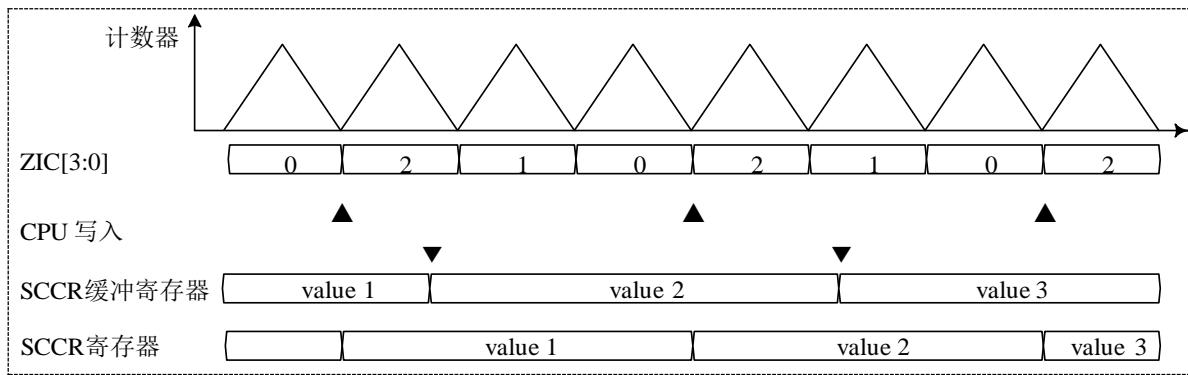


图 19-15 SCCR 缓冲传输操作（周期间隔响应链接传输使能时）

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。

19.3.3 通用 PWM 输出

19.3.3.1 独立 PWM 输出

在直通模式(POCR.PWMMD=00)下,分别设定 OCCRxh、OCCRxl 的基准值和 OCMRxh、OCMRxl 的端口输出状态 ($x=u, v, w$)，可实现不同的 PWM 输出。此时，每个端口的 PWM 输出都独立控制。图 19-16 和图 19-17 分别是单元 1 的锯齿波和三角波下的独立 PWM 输出例。

注意：

- 直通模式是指通用比较基准寄存器 (OCCRxh、OCCRxl) 的值发生比较匹配所产生的内部输出信号 (in_opxh、in_opxl) 直接输出至对应的端口 (TIM4_<t>_OXH、TIM4_<t>_OXL) 上 ($X=U, V, W, x=u, v, w$)。

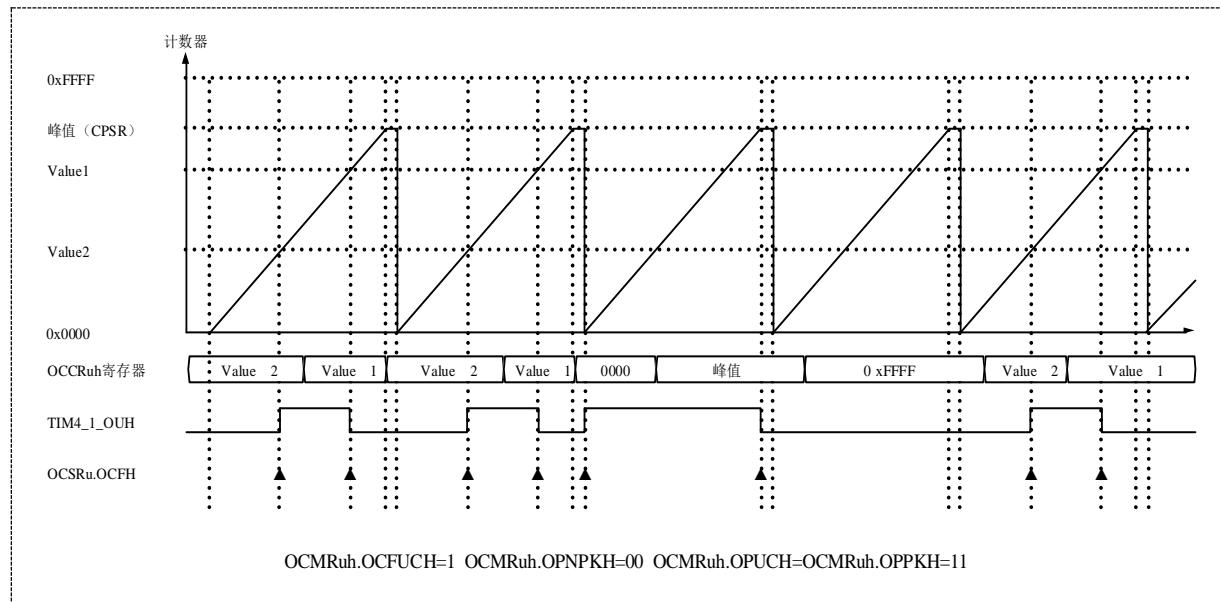


图 19-16 锯齿波独立 PWM 输出例

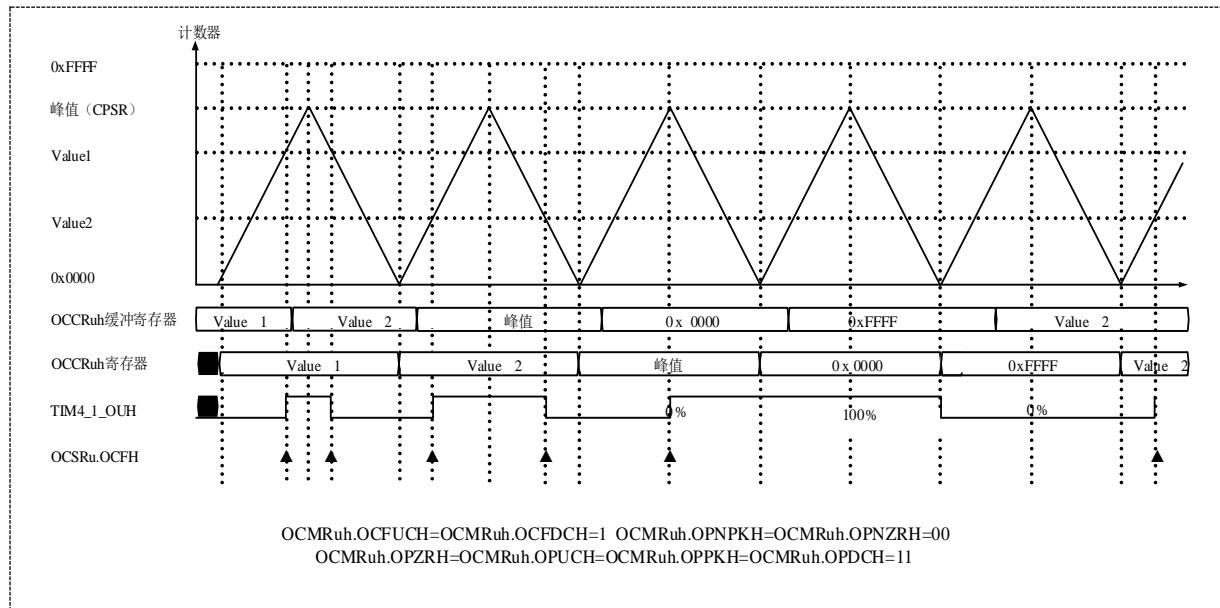


图 19-17 三角波独立 PWM 输出例

19.3.3.2 扩展 PWM 输出

在直通模式(POCR.PWMMD=00)下, TIM4_<t>_OXL 的端口输出状态也可由 OCMRxl 寄存器中的扩展位 (bit32~16) 决定, 该扩展位的设定与 OCCRxl 基准值相关, 从而在 TIM4_<t>_OXL 端口上实现扩展 PWM 输出 (X=U、V、W, x=u、v、w)。图 19-18 所示是该方式下 TIM4_<t>_OUH 和 TIM4_<t>_OUL 端口的 PWM 输出。

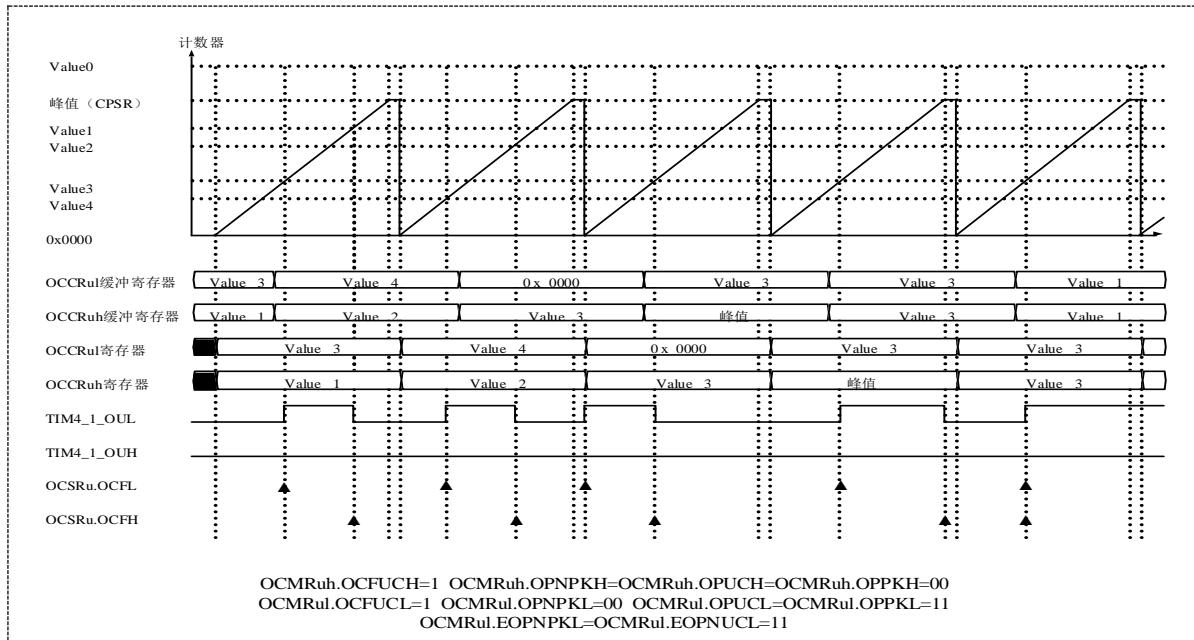


图 19-18 锯齿波扩展 PWM 输出

注意：

- 独立 PWM 输出方式下, TIM4_<t>_OXL 的端口状态由 OCMRx1 寄存器的 bit15~bit4 位决定, 仅与 OCCRxl 基准值有关 (X=U、V、W, x=u、v、w)。

19.3.3.3 互补 PWM 输出

软件设定实现互补 PWM 输出

在直通模式 (POCR.PWMMD=00) 下, 直接设定 OCCRxh 和 OCCRxl (x=u、v、w) 基准值实现一对互补 PWM 波形输出至端口, 3 组端口同样方式设定即可实现 3 相互补 PWM 输出。如图 19-19 所示。

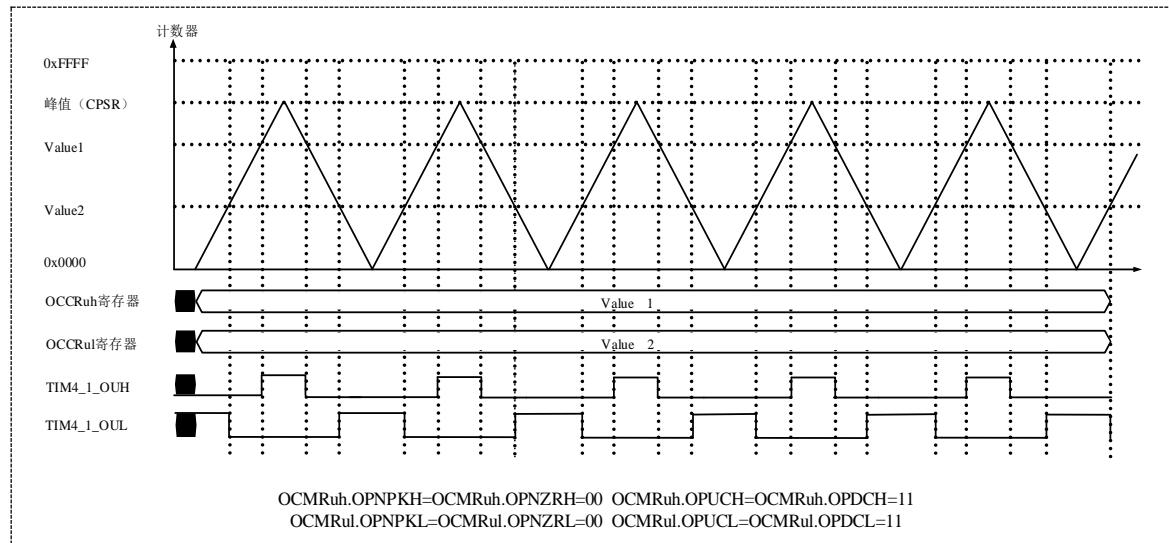


图 19-19 软件实现互补 PWM 输出

硬件设定实现互补 PWM 输出

在死区定时器模式 (POCR.PWMMD=01) 下, 通用比较基准寄存器 (OCCRxl) 的值发生比较匹配产生的内部输出信号 (in_opxl) 和 PWM 死区控制寄存器 (PDAR/PDBR) 的设定值通过时序偏移, 以硬件方式实现互补 PWM 输出。

在该模式下, TIM4_<t>_OXH 端口输出的极性与 in_opxl 相同, TIM4_<t>_OXL 端口输出的极性与 in_opxl 相反 ($X=U、V、W, x=u、v、w$)。

图 19-20 所示是死区定时器模式下的互补 PWM 输出例。

如果检测到 in_opxl 上升沿, TIM4_<t>_OXL 输出变为低电平, 死区计数器加载 PDBRx 寄存器的设定值并启动递减计数, 当计数值变为 0x0000 时, 计数器停止并使 TIM4_<t>_OXH 输出高电平; 如果检测到 in_opxl 下降沿, TIM4_<t>_OXH 输出变为低电平, 死区计数器加载 PDARx 寄存器的设定值并启动递减计数, 当计数值变为 0x0000 时, 计数器停止并使 TIM4_<t>_OXL 输出高电平 ($X=U、V、W, x=u、v、w$)。

通过设置 PWM 死区控制寄存器 PDAR 和 PDBR, 输出上升和下降变化的死区时间可以相应设定。

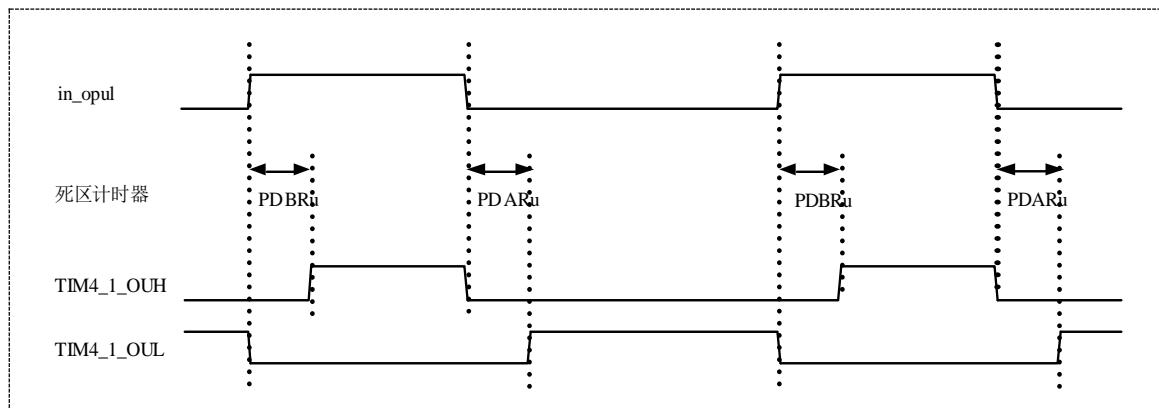


图 19-20 死区定时器模式下的互补 PWM 输出

in_opxl 的高电平脉冲宽度小于 PDBR 设定的死区时间时, 仅 TIM4_<t>_OXL 输出变为低电平。TIM4_<t>_OXL 输出电平变为高的条件是当 in_opxl 下降沿之后经过 PDAR 寄存器设定的死区时间之后。在此情况下, TIM4_<t>_OXH 输出将持续保持为低电平。

in_opxl 的低电平脉冲宽度小于 PDAR 设定的死区时间时, 仅 TIM4_<t>_OXH 输出变为低电平。TIM4_<t>_OXH 输出电平变为高的条件是当 in_opxl 上升沿之后经过 PDBR 寄存器设定的死区时间之后。在此情况下, TIM4_<t>_OXL 输出将持续保持为低电平 ($X=U、V、W, x=u、v、w$)。如图 19-21 所示。

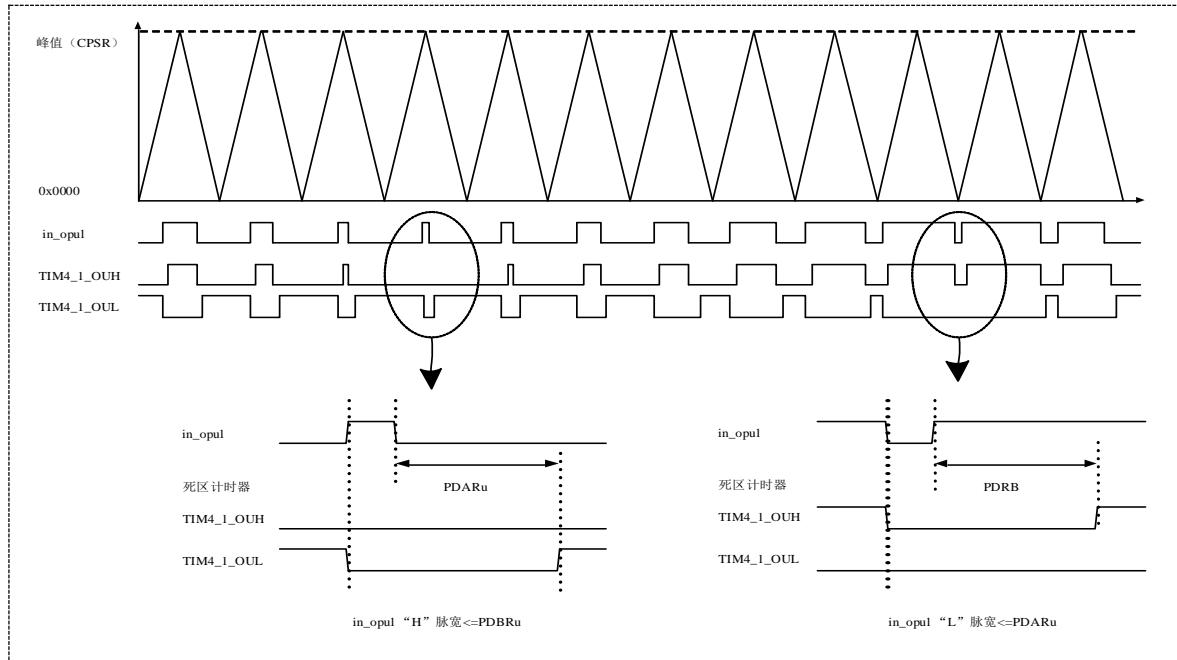


图 19-21 脉宽异常时的死区定时器模式下的波形输出

在上述硬件实现死区输出方式的基础上，还可以对内部的比较匹配信号（in_opxl）进行脉冲宽度的监测，从而实现对 in_opxl 信号的滤波控制。这种 in_opxl 带脉宽滤波的死区输出实现方式叫死区计数器滤波模式（POCR.PWMMD=10）（x=u、v、w）。

在死区计数器滤波模式下，滤波宽度由 PWM 滤波控制寄存器（PFSRn）的设定值决定。当 in_opxl 的脉冲宽度大于寄存器 PFSRn 设定的时间时，滤波计数器将 in_opxl 信号推迟 PFSR 设定的时间后输出，再经过死区定时器模式所描述的方式产生互补 PWM 输出（x=u、v、w）。

如果检测到信号 in_opxl 上升沿，滤波计数器加载 PFSR 寄存器值并启动计量 in_opxl 的高电平宽度，当 in_opxl 的高电平脉冲宽度大于寄存器 PFSR 设定的时间时，经过 PFSR 所设定时间之后， $\text{TIM4}_{\langle t \rangle} \text{ OXL}$ 输出变为低电平，死区计数器加载 PDBR 寄存器的设定值并启动递减计数，当计数值变为 0x0000 时，计数器停止并使 $\text{TIM4}_{\langle t \rangle} \text{ OXH}$ 输出高电平；如果检测到信号 in_opxl 下降沿，滤波计数器加载 PFSR 寄存器值并启动计量 in_opxl 的低电平宽度，当 in_opxl 的低电平脉冲宽度大于寄存器 PFSR 设定的时间时，经过 PFSR 所设定时间之后， $\text{TIM4}_{\langle t \rangle} \text{ OXH}$ 输出变为低电平，死区计数器加载 PDAR 寄存器的设定值并启动递减计数，当计数值变为 0x0000 时，计数器停止并使 $\text{TIM4}_{\langle t \rangle} \text{ OXL}$ 输出高电平。当 in_opxl 的电平脉冲宽度小于寄存器 PFSR 设定的时间时，输出 $\text{TIM4}_{\langle t \rangle} \text{ OXH}$ 和 $\text{TIM4}_{\langle t \rangle} \text{ OXL}$ 将保持不变（X=U、V、W）。

W, x=u、v、w)。

图 19-22 所示，是死区计数器滤波模式下的互补 PWM 输出例。

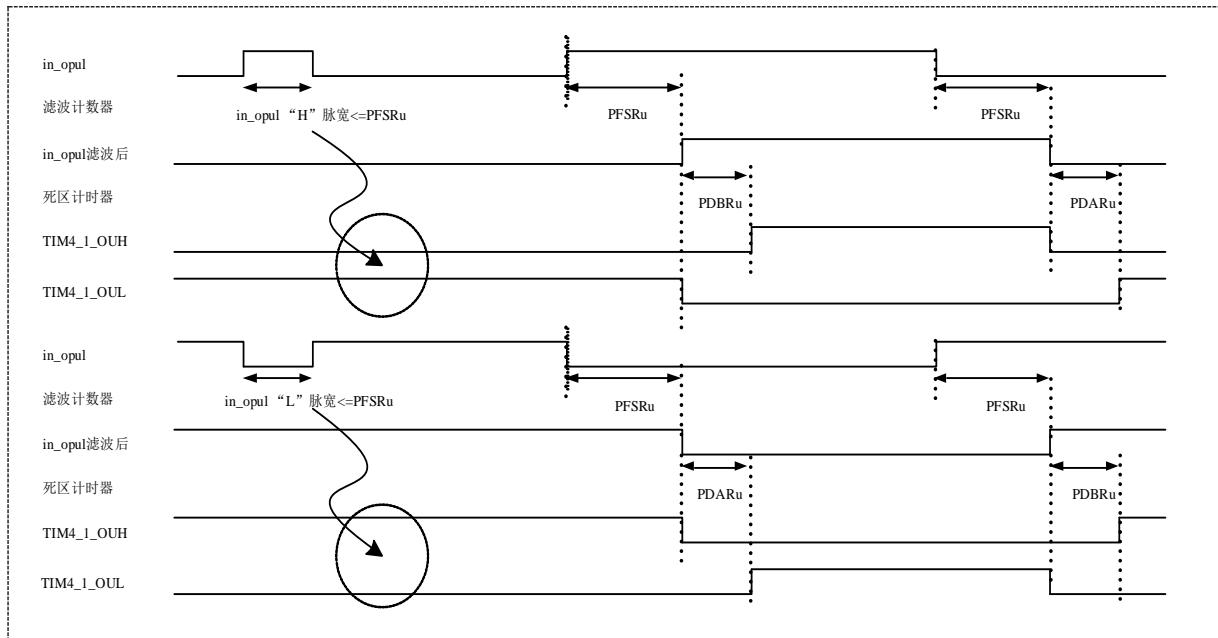


图 19-22 死区定时器滤波模式下的互补 PWM 输出

19.3.4 周期间隔响应

下溢中断屏蔽计数器用于屏蔽下溢标志位 (CCSR.IRQZF) 的被置位次数。下溢中断屏蔽计数器 (CVPR.ZIC[3:0]) 作为递减计数器操作，在开始时载入 CVPR.ZIM[3:0] 所设定的值，当 CVPR.ZIC[3:0] = “0”时，下溢标志位(CCSR.IRQZF)被置位为“1”。

上溢中断屏蔽计数器用于屏蔽上溢标志位 (CCSR.IRQPF) 的被置位次数。上溢中断屏蔽计数器 (CVPR.PIC[3:0]) 作为递减计数器操作，在开始时载入 CVPR.PIM[3:0] 所设定的值，当 CVPR.PIC[3:0] = “0”时，上溢标志位(CCSR.IRQPF)被置位为“1”。

如下图 19-23 所示，是周期间隔响应时的 IRQZF、IRQPF 的置位时序图。

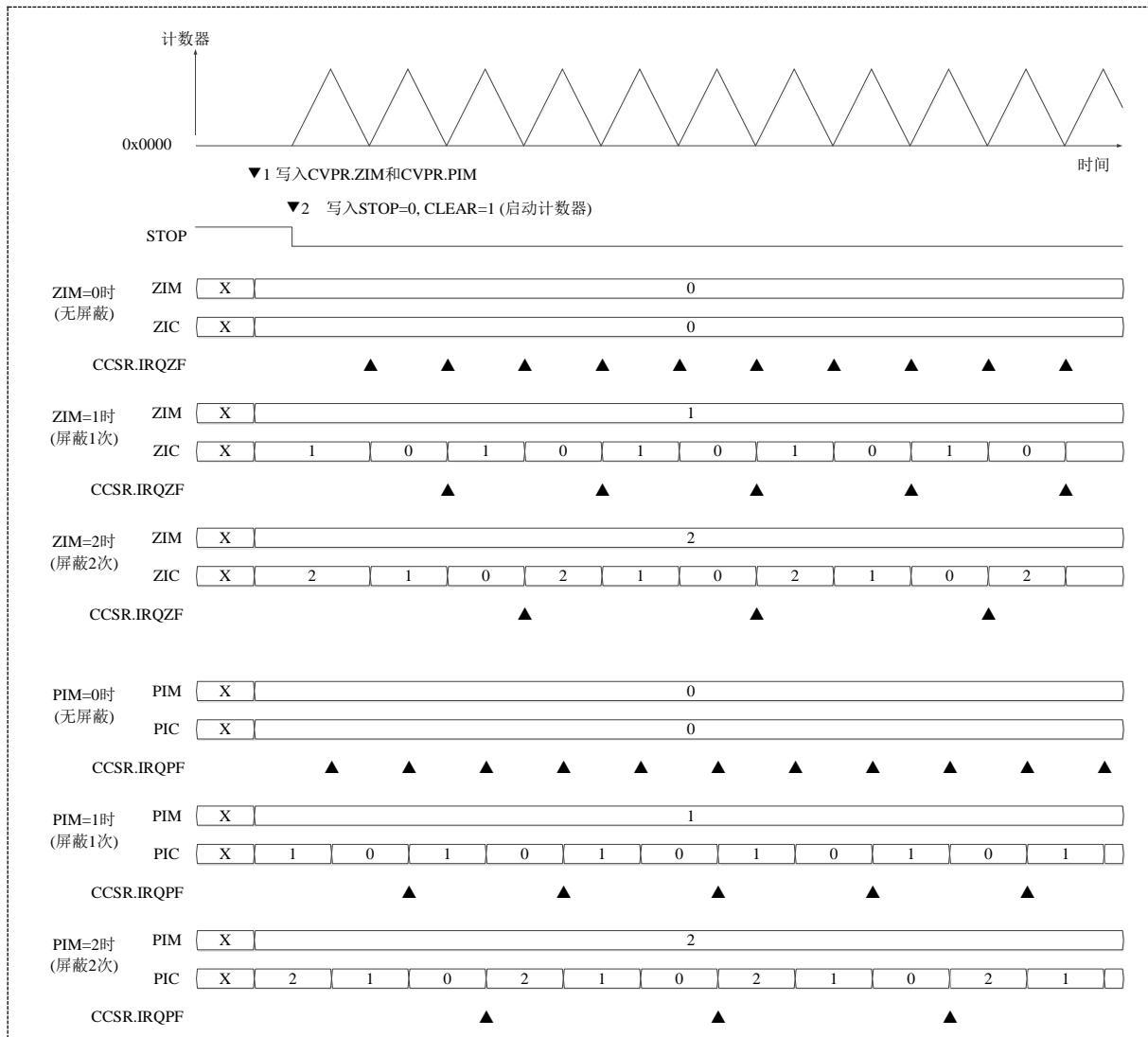


图 19-23 周期间隔响应时序图

▼1 计数器停止时，写入 CVPR.ZIM 和 CVPR.PIM 初始值，初始值立即反映在内部计数器(CVPR.ZIC、CVPR.PIC)。

▼2 初始化并启动计数器(STOP=0 且 CLEAR=1)，计数器在总线复位或者从软件初始化 CLEAR=1 后从零开始进行递增计数，此刻 CCSR.IRQZF 标志不会立即置位，之后每当中断屏蔽计数器的计数值为 0x0000 且计数器的计数值为 0x0000 和 CPSR 时，标志▲为 CCSR.IRQZF 或 CCSR.IRQPF 置位的时刻。

注意：

- 在计数器运行中，写入 CVPR.ZIM 和 CVPR.PIM，该设置值不会立即反应到中断屏蔽计数器(CVPR.ZIC 和 CVPR.PIC)中。如果写入软复位(CLEAR=1)，写入的 CVPR.ZIM 和 CVPR.PIM 值将立即加载为中断屏蔽计数器的初值。

专用比较基准寄存器（SCCRm）的比较匹配事件（专用事件输出）也具有周期间隔响应功能。专用控制状态寄存器 SCSR 设置为比较模式输出时(EVTMS=0)，EVT 事件启动使能为上溢点 (ZEN)、向上计数时(UEN)、下溢点(PEN)和向下计数时(DEN)。在不使用周期间隔功能时，打开使能后，专用事件输出信号在对应的计数匹配事件发生时都会置位。其中，上溢点匹配事件和下溢点匹配事件不支持周期间隔功能。在使用周期间隔功能时，三角波模式下，UEN=1 或者 DEN=1 且计数值达到 SCMRm 寄存器设置的值时，一个有效周期内产生 1 次 EVT 事件；UEN=1 且 DEN=1 且计数值达到 SCMRm 寄存器设置的值时，一个有效周期内会产生 2 次 EVT 事件。如下图 19-24 所示，是专用事件输出信号的周期间隔响应输出图。

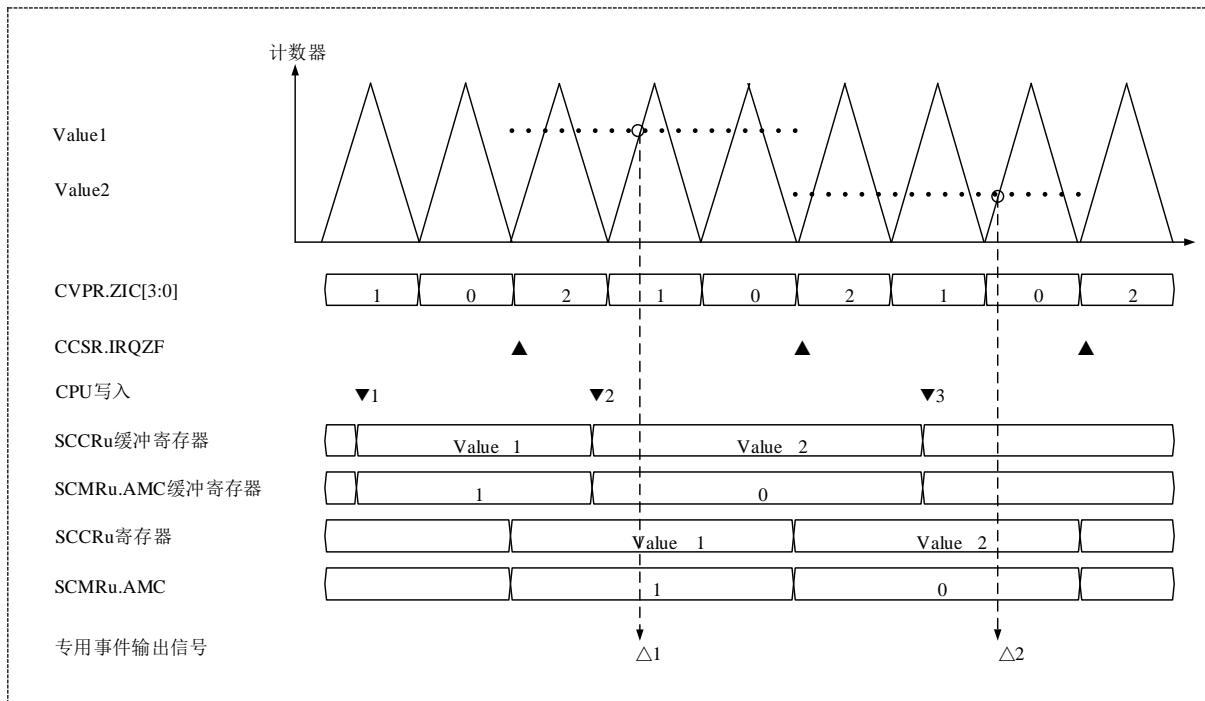


图 19-24 专用事件输出信号周期间隔响应输出

计数器是三角波计数模式，下溢中断屏蔽计数器(CVPR.ZIC)从 2-0 做递减计数。下溢中断在时刻▲产生。

在时刻▼1，向 SCCR_u 缓冲寄存器写入 value 1。同时，MZCE=1、MPCE=0、AMC=0001 写入 SCMR_u 缓冲寄存器。打开 SCSR 寄存器中的向上计数 EVT 使能 (UEN=1)，以及周期间隔响应功能链接使能(LMC=1) 以及设置下溢缓存 (BUFEN=01)，缓冲寄存器 SCCR_u 和 SCMR_u 传送操作在▲时刻执行。因为在向上计数时，计数值=SCCR_u=Value 1 且 MZCE=1 且 AMC=1，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 value 2。同时，MZCE=1、MPCE=0、AMC=0000 写入 SCMRu 缓冲寄存器。之后，缓冲寄存器 SCCRu 和 SCMRu 传送操作在▲时刻执行。因为在向上计数时，MZCE=1 且 AMC=0 且计数值=SCCRu=Value 2，在时刻△2，专用事件输出信号置位。

19.3.5 EMB 控制

每个 Timer4 单元都有一个输出无效事件接口，连接 EMB 模块输出的 EMB 事件。该接口上选通的异常状况事件可从 EMB 设定（参见 EMB 章节）。

各单元内 3 组 PWM 端口在正常输出期间，若监测到从 EMB 过来的 EMB 异常事件，则端口的输出状态可变为预先设定好的状态。该预设端口状态可以为输出高阻态、输出低电平，输出高电平，保持正常输出以及维持之前状态不变。（由 ECER.EMBVAL 和 ECSR.HOLD 设定）。

例如，若 ECER.EMBVAL=01 设定时，在 Timer4 的 PWM 端口正常输出期间产生 EMB 事件，则 PWM 端口上输出变为高阻态。

19.4 中断及事件说明

19.4.1 计数比较匹配中断

通用比较基准值寄存器 (OCCRm) 共计 6 个，可分别与计数值比较产生比较匹配有效信号。计数比较匹配时，通用控制状态寄存器 (OCSRn) 中的 OCSRn.OCFH、OCSRn.OCFL 位分别会被置为 1。此时若设定 OCSRn.OCIEH、OCSRn.OCIEL 使能中断，则对应的中断请求 (TMR4_U<t>_GCMmn, m=U、V、W; n=H、L) 也会被触发。

19.4.2 计数周期匹配中断

锯齿波递加计数至上溢点、三角波计数至谷点或三角波计数至峰点时，控制状态寄存器 (CCSR) 的 CCSR.IRQPF 或 CCSR.IRQZF 位会被置为 1。此时若设置 CCSR.IRQPEN 或 CCSR.IRQZEN 位使能中断，则在对应的时间点可触发计数周期匹配中断 (TMR4_U<t>_GOVF 和 TMR4_U<t>_GUDF)。

19.4.3 重载计数匹配中断

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 位分别会被置为 1。此时若设定 RCSR.RTIDU、RCSR.RTIDV、RCSR.RTIDW 中断屏蔽无效，则对应的重载计数匹配中断请求 (TMR4_U<t>_RLOm, m=U、V、W) 也会被触发。

19.4.4 专用比较匹配事件

Timer4 的 6 个专用比较基准寄存器 (SCCRm) 对应产生 6 个专用事件输出信号可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 (TMR4_U<t>_SCMmn, m=U、V、W; n=H、L)，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）用于触发其它模块。

该事件请求信号的输出可以选择比较启动模式或延时启动模式。在比较启动模式时 (SCSR.EVTMS=0)，产生 SCCR 的计数比较匹配事件后，专用事件输出信号直接输出；在延时启动模式时 (SCSR.EVTMS=1)，产生 OCCRxh 或 OCCRxl(由 SCSR.EVTDS 位选择； $x=u, v, w$) 的计数比较匹配事件后，经过 SCCR 设定的基准周期时间后，专用事件输出信号输出。下图 19-25 所示，是延时启动模式下的专用事件输出信号的请求输出例。

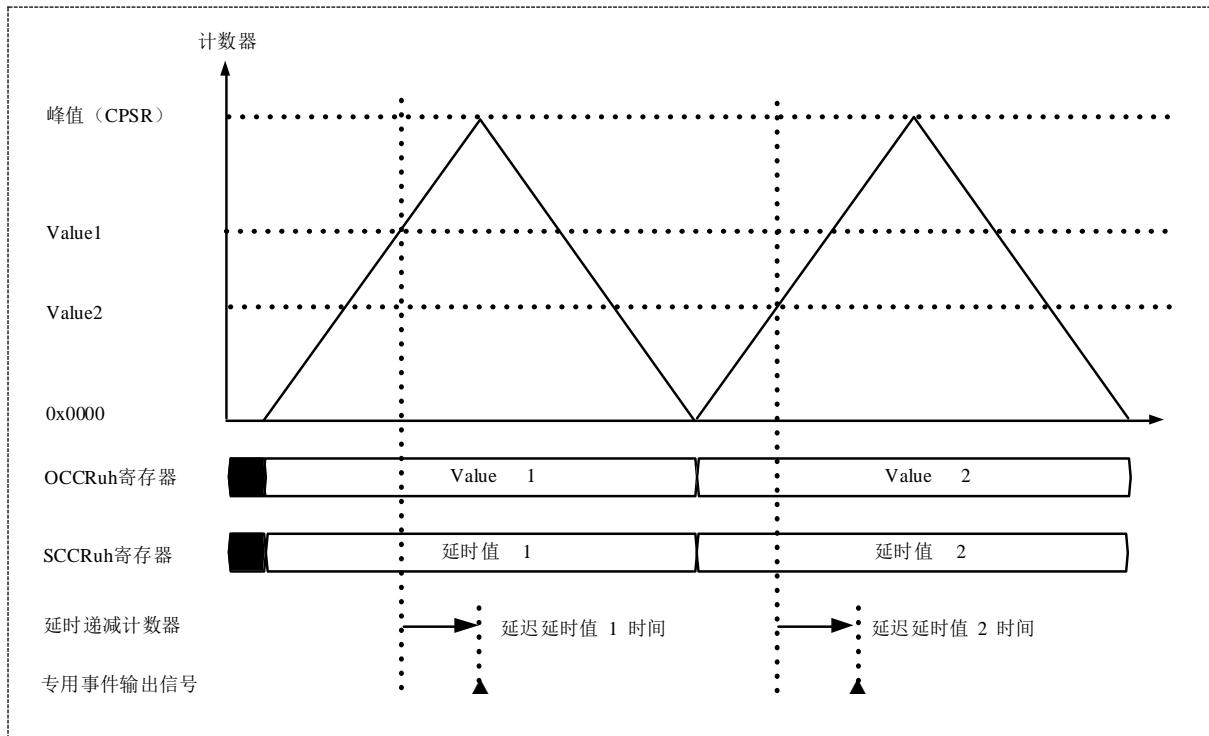


图 19-25 延时启动模式下专用事件输出信号的输出时序

注意：

- 在延迟计数运行中，如果再次发生 OCCR 与计数器匹配事件，延迟计数器重新加载计数值并重新进行递减计数。因此，如果 OCCR 匹配事件时间间隔小于设定的延迟时间 SCCR 时，专用事件输出的请求信号可能一直不会产生。

19.5 寄存器说明

表 19-3 所示，为 Timer4 模块的寄存器列表。

BASE ADDR: 0x4001_7000 (U1)、0x4002_4800 (U2)、0x4002_4C00 (U3)

寄存器名	符号	偏移量	位宽	复位值
计数值寄存器	TMR4_CNTR	0x0046	16	0x0000
周期基准寄存器	TMR4_CPSR	0x0042	16	0xFFFF
控制状态寄存器	TMR4_CCSR	0x0048	16	0x0040
有效周期寄存器	TMR4_CVPR	0x004A	16	0x0000
通用比较基准寄存器UH	TMR4_OCCRuh	0x0002	16	0x0000
通用比较基准寄存器UL	TMR4_OCCRul	0x0006	16	0x0000
通用比较基准寄存器VH	TMR4_OCCRvh	0x000A	16	0x0000
通用比较基准寄存器VL	TMR4_OCCRvl	0x000E	16	0x0000
通用比较基准寄存器WH	TMR4_OCCRwh	0x0012	16	0x0000
通用比较基准寄存器WL	TMR4_OCCRwl	0x0016	16	0x0000
通用控制状态寄存器U	TMR4_OCSRu	0x0018	16	0xFF00
通用控制状态寄存器V	TMR4_OCSRv	0x001C	16	0xFF00
通用控制状态寄存器W	TMR4_OCSRw	0x0020	16	0xFF00
通用扩展控制寄存器U	TMR4_OCERu	0x001A	16	0x0000
通用扩展控制寄存器V	TMR4_OCERv	0x001E	16	0x0000
通用扩展控制寄存器W	TMR4_OCERw	0x0022	16	0x0000
通用模式控制寄存器UH	TMR4_OCMRuh	0x0024	16	0x0000
通用模式控制寄存器UL	TMR4_OCMRul	0x0028	32	0x0000_0000
通用模式控制寄存器VH	TMR4_OCMRvh	0x002C	16	0x0000
通用模式控制寄存器VL	TMR4_OCMRvl	0x0030	32	0x0000_0000
通用模式控制寄存器WH	TMR4_OCMRwh	0x0034	16	0x0000
通用模式控制寄存器WL	TMR4_OCMRwl	0x0038	32	0x0000_0000
专用比较基准寄存器UH	TMR4_SCCRuh	0x00B2	16	0x0000

专用比较基准寄存器UL	TMR4_SCCRul	0x00B6	16	0x0000
专用比较基准寄存器VH	TMR4_SCCRvh	0x00BA	16	0x0000
专用比较基准寄存器VL	TMR4_SCCRvl	0x00BE	16	0x0000
专用比较基准寄存器WH	TMR4_SCCRwh	0x00C2	16	0x0000
专用比较基准寄存器WL	TMR4_SCCRwl	0x00C6	16	0x0000
专用控制状态寄存器UH	TMR4_SCSRuh	0x00C8	16	0x0000
专用控制状态寄存器UL	TMR4_SCSRul	0x00CC	16	0x0000
专用控制状态寄存器VH	TMR4_SCSRvh	0x00D0	16	0x0000
专用控制状态寄存器VL	TMR4_SCSRvl	0x00D4	16	0x0000
专用控制状态寄存器WH	TMR4_SCSRwh	0x00D8	16	0x0000
专用控制状态寄存器WL	TMR4_SCSRwl	0x00DC	16	0x0000
专用模式控制寄存器UH	TMR4_SCMRuh	0x00CA	16	0xFF00
专用模式控制寄存器UL	TMR4_SCMRul	0x00CE	16	0xFF00
专用模式控制寄存器VH	TMR4_SCMRvh	0x00D2	16	0xFF00
专用模式控制寄存器VL	TMR4_SCMRvl	0x00D6	16	0xFF00
专用模式控制寄存器WH	TMR4_SCMRwh	0x00DA	16	0xFF00
专用模式控制寄存器WL	TMR4_SCMRwl	0x00DE	16	0xFF00
PWM基本控制寄存器U	TMR4_POCRu	0x0098	16	0xFF00
PWM基本控制寄存器V	TMR4_POCRv	0x009C	16	0xFF00
PWM基本控制寄存器W	TMR4_POCRw	0x00A0	16	0xFF00
PWM滤波控制寄存器U	TMR4_PFSRu	0x0082	16	0x0000
PWM滤波控制寄存器V	TMR4_PFSRv	0x008A	16	0x0000
PWM滤波控制寄存器W	TMR4_PFSRw	0x0092	16	0x0000
PWM死区控制寄存器AU	TMR4_PDARu	0x0084	16	0x0000
PWM死区控制寄存器BU	TMR4_PDBRu	0x0086	16	0x0000
PWM死区控制寄存器AV	TMR4_PDARv	0x008C	16	0x0000
PWM死区控制寄存器BV	TMR4_PDBRv	0x008E	16	0x0000
PWM死区控制寄存器AW	TMR4_PDARw	0x0094	16	0x0000

PWM死区控制寄存器BW	TMR4_PDBRw	0x0096	16	0x0000
重载控制状态寄存器	TMR4_RCSR	0x00A4	16	0x0000
EMB控制状态寄存器	TMR4_ECSR	0x00F0	16	0x0000
EMB扩展控制寄存器	TMR4_ECER	U1: (0x4005_5408) U2: (0x4005_540C) U3: (0x4005_5410)	32	0x0000_0000

表 19-3 寄存器列表

注意：

- 下述寄存器详细描述中，m=uh、ul、vh、vl、wh、wl，n=u、v、w。m 所指寄存器分别在功能实现时对应端口 TIM4_<t>_OUH、TIM4_<t>_OUL、TIM4_<t>_OVH、TIM4_<t>_OVL、TIM4_<t>_OWH、TIM4_<t>_OWL 的输出控制等；n 所指寄存器分别在功能实现时对应端口 TIM4_<t>_OUx、TIM4_<t>_OVx、TIM4_<t>_OWx 的输出控制等，其中 x=H 或 L，且 H 或 L 的具体控制在这些寄存器中有相对应的对称位。

19.5.1 计数值寄存器 (TMR4_CNTR)

复位值: 0x0000

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

位	标记	位名	功能	读写
b15~b0	CNTR[15:0]	计数器当前值	在计数停止时，通过向该寄存器写入值，可初始化计数器计数值 在计数中时，该位指示当前计数器计数值 注：在计数中时，不能向该寄存器写入值	R/W

19.5.2 周期基准寄存器 (TMR4_CPSR)

复位值: 0xFFFF

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

位	标记	位名	功能	读写
b15~b0	CPSR[15:0]	通用周期基准值	计数器的计数周期值 注：从本地址区域读取数据时，读取的并非缓冲器寄存器的值，而是CPSR寄存器的值	R/W

19.5.3 控制状态寄存器 (TMR4_CCSR)

复位值: 0x0040

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ECK EN	IRQ ZF	IRQ ZEN	-	-	-	IRQ PF	IRQ PEN	BUF EN	STOP	MODE	CLEAR				CKDIV[3:0]

位	标记	位名	功能	读写
b15	ECKEN	时钟源选择	0: 内部PCLK1时钟 1: 外部TIM4_<D>_CLK端口输入时钟 注1: 该位在计数器停止时设定 注2: 使用外部输入时钟进行操作的场合, 计数器操作开始后, 最初的外部输入时钟的边沿, 无论是上升沿还是下降沿都被忽略, 计数操作从之后的边沿开始	R/W
b14	IRQZF	下溢状态	0: 未发生计数下溢 1: 发生计数下溢 注1: 当使用周期间隔响应功能时, 该位的置位条件由CVPR所设定的周期间隔计数器设定 注2: 当计数器被总线复位或者写入CLEAR=“1”, IRQZF位将不会被置位	R/W
b13	IRQZEN	下溢中断使能	0: 禁止IRQZF产生中断到CPU 1: 允许IRQZF产生中断到CPU	R/W
b12~b10	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b9	IRQPF	上溢状态	0: 未发生计数上溢 1: 发生计数上溢 注1: 当使用周期间隔响应功能时, 该位的置位条件由CVPR所设定的周期间隔计数器设定。 注2: 当计数器被总线复位或者写入CLEAR=“1”, IRQZF位将不会被置位	R/W
b8	IRQOPEN	上溢中断使能	0: 禁止IRQPF产生中断到CPU 1: 允许IRQPF产生中断到CPU	R/W
b7	BUFEN	缓存使能	0: 禁止CPSR缓存功能 1: 使能CPSR缓存功能	R/W
b6	STOP	计数器使能	0: 计数器启动 1: 计数器停止	R/W
b5	MODE	波形模式	0: 锯齿波模式 (仅支持向上计数) 1: 三角波模式	R/W
b4	CLEAR	计数器清零	0: 无操作 1: 计数器清零 注: 该位读出时始终为0	R/W
b3~b0	CKDIV	计数时钟分频	该位指示基本计数器的计数时钟分频	R/W

0000: 计数时钟为PCLK1
0001: 计数时钟为PCLK1/2
0010: 计数时钟为PCLK1/4
0011: 计数时钟为PCLK1/8
0100: 计数时钟为PCLK1/16
0101: 计数时钟为PCLK1/32
0110: 计数时钟为PCLK1/64
0111: 计数时钟为PCLK1/128
1000: 计数时钟为PCLK1/256
1001: 计数时钟为PCLK1/512
1010: 计数时钟为PCLK1/1024
请不要设定其它值

19.5.4 有效周期寄存器 (TMR4_CVPR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIC[3:0]				ZIC[3:0]				PIM[3:0]				ZIM[3:0]			

位	标记	位名	功能	读写
b15~b12	PIC[3:0]	上溢中断屏蔽状态	当前需要被屏蔽的上溢中断个数 注1：计数器停止时改写PIM，则PIC立即更新为新的PIM值 注2：如果计数开始后上溢发生前改写PIM，则PIC在上溢时更新为新的PIM值，且仍会触发此次上溢中断。此后每次上溢PIC递减 注3：如果在上溢中断中改写PIM，则PIC在下一次上溢时更新为新的PIM值，且仍会触发下一次上溢中断。此后每次上溢PIC递减	R
b11~b8	ZIC[3:0]	下溢中断屏蔽状态	当前需要被屏蔽的下溢中断个数 注1：计数器停止时改写ZIM，则ZIC立即更新为新的ZIM值 注2：如果计数开始后上溢发生前改写ZIM，则ZIC在上溢时更新为新的ZIM值，且仍会触发此次上溢中断。此后每次上溢ZIC递减 注3：如果在上溢中断中改写ZIM，则PIC在下一次上溢时更新为新的ZIM值，且仍会触发下一次上溢中断。此后每次上溢ZIC递减	R
b7~b4	PIM[3:0]	上溢中断屏蔽设定	设置屏蔽的上溢中断个数	R/W
b3~b0	ZIM[3:0]	下溢中断屏蔽设定	设置屏蔽的下溢中断个数	R/W

19.5.5 通用比较基准寄存器 (TMR4_OCCRm)

复位值: 0x0000

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

位	标记	位名	功能	读写
b15~b0	OCCR[15:0]	通用比较基准值	通用比较基准值 注：从本地址区域读取数据时，读取的并非缓冲器寄存器的值，而是OCCR寄存器的值	R/W

19.5.6 通用控制状态寄存器 (TMR4_OCSRn)

复位值: 0xFF00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								OCFL L	OCFH H	OCIEL L	OCIEH H	OCP L	OCP H	OCE L	OCE H

位	标记	位名	功能	读写
b15~b8	Reserved	-	读出时为“1”，写入时写“1”	R/W
b7	OCFL	计数匹配L	0: 计数器计数值与OCCRxl设定值不相等 1: 计数器计数值与OCCRxl设定值相等 (x=u、v、w) 注: 该位必须在OCEL=1时有效	R/W
b6	OCFH	计数匹配H	0: 计数器计数值与OCCRxh设定值不相等 1: 计数器计数值与OCCRxh设定值相等 (x=u、v、w) 注: 该位必须在OCEH=1时有效	R/W
b5	OCIEL	计数匹配L中断使能	0: OCFL置位时, 不发生中断 1: OCFL置位时, 发生中断	R/W
b4	OCIEH	计数匹配H中断使能	0: OCFH置位时, 不发生中断 1: OCFH置位时, 发生中断	R/W
b3	OCP L	OCEL=0时端口状态	0: OCEL=0时, 此位写“0”则TIM4_<t>_OxL上输出低电平 1: OCEL=0时, 此位写“1”则TIM4_<t>_OxL上输出高电平 (m=U、V、W) 注: OCEL=1时, 写操作无效	R/W
b2	OCP H	OCEH=0时端口状态	0: OCEH=0时, 此位写“0”则TIM4_<t>_OxH上输出低电平 1: OCEH=0时, 此位写“1”则TIM4_<t>_OxH上输出高电平 (m=U、V、W) 注: OCEH=1时, 写操作无效	R/W
b1	OCEL	端口输出使能L	0: TIM4_<t>_OxL输出无效, 端口状态由OCPL决定 1: TIM4_<t>_OxL输出使能, 端口状态由OCMRyl的设定和 OCFL状态决定 (x=U、V、W, y=u、v、w)	R/W
b0	OCEH	端口输出使能H	0: TIM4_<t>_OxH输出无效, 端口状态由OCPH决定 1: TIM4_<t>_OxH输出使能, 端口状态由OCMRyh的设定和 OCFH状态决定 (x=U、V、W, y=u、v、w)	R/W

19.5.7 通用扩展控制寄存器 (TMR4_OCERn)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	MCEC L	MCEC H	LMM L	LMM H	LMC L	LMC H	MLBUF EN[1:0]	MHBUF EN[1:0]	CLBUF EN[1:0]	CHBUF EN[1:0]				

位	标记	位名	功能	读写
b15~b14	Reserved	-	读出时为“0”，写入时写“0”	R/W
b13	MCECL	上溢点匹配使能L	0: 计数值到达上溢点 (CNTR=CPSR) 且OCCRxl > CPSR时，禁止产生比较匹配L 1: 计数值到达上溢点 (CNTR=CPSR) 且OCCRxl > CPSR时，视为发生比较匹配L (x=u、v、w)	R/W
b12	MCECH	上溢点匹配使能H	0: 计数值到达上溢点 (CNTR=CPSR) 且OCCRxl > CPSR时，禁止产生比较匹配H 1: 计数值到达上溢点 (CNTR=CPSR) 且OCCRxl > CPSR时，视为发生比较匹配H (x=u、v、w)	R/W
b11	LMML	周期间隔响应功能链接L	0: 周期间隔响应功能链接无效，OCMRxl的缓存传送由MLBUFEN设定决定 1: 周期间隔响应功能链接有效，OCMRxl的缓存传送在MLBUFEN设定的基础上，还必须满足CVPR.PIC[3:0]=0000 (计数上溢时) 或CVPR.ZIC[3:0]=0000 (计数下溢时) (x=u、v、w)	R/W
b10	LMMH	周期间隔响应功能链接H	0: 周期间隔响应功能链接无效，OCMRxh的缓存传送由MHBUFEN[1:0]设定决定 1: 周期间隔响应功能链接有效，OCMRxh的缓存传送在MHBUFEN[1:0]设定的基础上，还必须满足CVPR.PIC[3:0]=0000 (计数上溢时) 或CVPR.ZIC[3:0]=0000 (计数下溢时) (x=u、v、w)	R/W
b9	LMCL	周期间隔响应功能链接L	0: 周期间隔响应功能链接无效，OCCRxl的缓存传送由CLBUFEN[1:0]设定决定 1: 周期间隔响应功能链接有效，OCCRxl的缓存传送在CLBUFEN[1:0]设定的基础上，还必须满足CVPR.PIC[3:0]=0000 (计数上溢时) 或CVPR.ZIC[3:0]=0000 (计数下溢时) (x=u、v、w)	R/W
b8	LMCH	周期间隔响应功能链接H	0: 周期间隔响应功能链接无效，OCCRxh的缓存传送由CHBUFEN[1:0]设定决定	R/W

1: 周期间隔响应功能链接有效，OCCRxl的缓存传送在
CHBUFEN[1:0]设定的基础上，还必须满足
CVPR.PIC[3:0]=0000（计数上溢时）或CVPR.ZIC[3:0]=0000
(计数下溢时)
(x=u、v、w)

b7~b6	MLBUFEN[1:0]	OCMRxl缓存传送	00: OCMRxl缓存寄存器的值直接写入OCMRxl 01: OCMRxl缓存寄存器的值在计数下溢时写入OCMRxl 10: OCMRxl缓存寄存器的值在计数上溢时写入OCMRxl 11: OCMRxl缓存寄存器的值在计数下溢或上溢时写入 OCMRxl (x=u、v、w)	R/W
b5~b4	MHBUFEN[1:0]	OCMRxh缓存转送	00: OCMRxh缓存寄存器的值直接写入OCMRxh 01: OCMRxh缓存寄存器的值在计数下溢时写入OCMRxh 10: OCMRxh缓存寄存器的值在计数上溢时写入OCMRxh 11: OCMRxh缓存寄存器的值在计数下溢或上溢时写入 OCMRxh (x=u、v、w)	R/W
b3~b2	CLBUFEN[1:0]	OCCRxl缓存传送	00: OCCRxl缓存寄存器的值直接写入OCCRxl 01: OCCRxl缓存寄存器的值在计数下溢时写入OCCRxl 10: OCCRxl缓存寄存器的值在计数上溢时写入OCCRxl 11: OCCRxl缓存寄存器的值在计数下溢或上溢时写入OCCRxl (x=u、v、w)	R/W
b1~b0	CHBUFEN[1:0]	OCCRxh缓存传送	00: OCCRxh缓存寄存器的值直接写入OCCRxh 01: OCCRxh缓存寄存器的值在计数下溢时写入OCCRxh 10: OCCRxh缓存寄存器的值在计数上溢时写入OCCRxh 11: OCCRxh缓存寄存器的值在计数下溢或上溢时写入 OCCRxh (x=u、v、w)	R/W

19.5.8 通用模式控制寄存器 (TMR4_OCMRm)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OPN ZRH[1:0]	OPN PKH[1:0]	OP ZRH[1:0]	OP UCH[1:0]	OP PKH[1:0]	OP DCH[1:0]	OP ZRH	OCF UCH	OCF PKH	OCF DCH						

注: 该寄存器位描述用于 OCMRuh、OCMRvh、OCMRwh 时

位	标记	位名	功能	读写
b15~b14	OPNZRH[1:0]	下溢点端口状态H	条件: 计数下溢 & OCCRxh计数不匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyH端口位保持不变 01: 满足条件时, TIM4_<t>_OyH端口位输出高电平 10: 满足条件时, TIM4_<t>_OyH端口位输出低电平 11: 满足条件时, TIM4_<t>_OyH端口位输出反转 (y=U、V、W)	R/W
b13~b12	OPNPKH[1:0]	上溢点端口状态H	条件: 计数上溢 & OCCRxh计数不匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyH端口位保持不变 01: 满足条件时, TIM4_<t>_OyH端口位输出高电平 10: 满足条件时, TIM4_<t>_OyH端口位输出低电平 11: 满足条件时, TIM4_<t>_OyH端口位输出反转 (y=U、V、W)	R/W
b11~b10	OPZRH[1:0]	下溢点端口状态H	条件: 计数下溢 & OCCRxh计数匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyH端口位保持不变 01: 满足条件时, TIM4_<t>_OyH端口位输出高电平 10: 满足条件时, TIM4_<t>_OyH端口位输出低电平 11: 满足条件时, TIM4_<t>_OyH端口位输出反转 (y=U、V、W)	R/W
b9~b8	OPUCH[1:0]	向上计数端口状态H	条件: 计数器向上计数 & OCCRxh计数匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyH端口位保持不变 01: 满足条件时, TIM4_<t>_OyH端口位输出高电平 10: 满足条件时, TIM4_<t>_OyH端口位输出低电平 11: 满足条件时, TIM4_<t>_OyH端口位输出反转 (y=U、V、W)	R/W
b7~b6	OPPKH[1:0]	上溢点端口状态H	条件: 计数上溢 & OCCRxh计数匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyH端口位保持不变 01: 满足条件时, TIM4_<t>_OyH端口位输出高电平 10: 满足条件时, TIM4_<t>_OyH端口位输出低电平 11: 满足条件时, TIM4_<t>_OyH端口位输出反转 (y=U、V、W)	R/W
b5~b4	OPDCH[1:0]	向下计数端口状态H	条件: 计数器向下计数 & OCCRxh计数匹配 (x=u、v、w) 00: 满足条件时, TIM4_<t>_OyH端口位保持不变	R/W

01: 满足条件时, TIM4_<1>_OyH端口位输出高电平
 10: 满足条件时, TIM4_<1>_OyH端口位输出低电平
 11: 满足条件时, TIM4_<1>_OyH端口位输出反转
 (y=U、V、W)

b3	OCFZRH	下溢点OCFH状态H	条件: 计数下溢 & OCCRxh计数匹配 (x=u、v、w) 0: 满足条件时, OCSR.OCFH位保持不变 1: 满足条件时, OCSR.OCFH位置位	R/W
b2	OCFUCH	向上计数OCFH状态H	条件: 计数器向上计数 & OCCRxh计数匹配 (x=u、v、w) 0: 满足条件时, OCSR.OCFH位保持不变 1: 满足条件时, OCSR.OCFH位置位	R/W
b1	OCFPKH	上溢点OCFH状态H	条件: 计数上溢 & OCCRxh计数匹配 (x=u、v、w) 0: 满足条件时, OCSR.OCFH位保持不变 1: 满足条件时, OCSR.OCFH位置位	R/W
b0	OCFDCH	向下计数OCFH状态H	条件: 计数器向下计数 & OCCRxh计数匹配 (x=u、v、w) 0: 满足条件时, OCSR.OCFH位保持不变 1: 满足条件时, OCSR.OCFH位置位	R/W

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EOPN	EOPN	EOP	EOP	EOP	EOP	EOP	EOP	EOPN							
ZRL[1:0]	PKL[1:0]	ZRL[1:0]	UCL[1:0]	PKL[1:0]	DCL[1:0]	UCL[1:0]	DCL[1:0]	ZRL	UCL	PKL	DCL	ZRL	UCL	PKL	DCL
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OPN	OPN	OP	OP	OP	OP	OP	OP	OP	OP	OP	OP	OPC	OPC	OPC	OPC
ZRL[1:0]	PKL[1:0]	ZRL[1:0]	UCL[1:0]	PKL[1:0]	DCL[1:0]	UCL[1:0]	DCL[1:0]	ZRL	UCL	PKL	DCL	ZRL	UCL	PKL	DCL

注: 该寄存器位描述用于 OCMRul、OCMRvl、OCMRwl 时

位	标记	位名	功能	读写
b31~b30	EOPNZRL[1:0]	扩张下溢点端口 状态L	条件: 计数下溢 & OCCRxl计数不匹配 & OCCRxh计数匹配 (x=u、v、w) 00: 满足条件时, TIM4_<1>_OyL端口位保持不变 01: 满足条件时, TIM4_<1>_OyL端口位输出高电平 10: 满足条件时, TIM4_<1>_OyL端口位输出低电平 11: 满足条件时, TIM4_<1>_OyL端口位输出反转 (y=U、V、W)	R/W
b29~b28	EOPNPKL[1:0]	扩张上溢点端口 状态L	条件: 计数上溢 & OCCRxl计数不匹配 & OCCRxh计数匹配 (x=u、v、w) 00: 满足条件时, TIM4_<1>_OyL端口位保持不变 01: 满足条件时, TIM4_<1>_OyL端口位输出高电平 10: 满足条件时, TIM4_<1>_OyL端口位输出低电平 11: 满足条件时, TIM4_<1>_OyL端口位输出反转 (y=U、V、W)	R/W
b27~b26	EOPZRL[1:0]	扩张下溢点端口	条件: 计数下溢 & OCCRxl计数匹配 & OCCRxh计数匹配 (x=u、v、w)	R/W

		状态L	v、 w)	
			00: 满足条件时, TIM4_<=>_OyL端口位保持不变	
			01: 满足条件时, TIM4_<=>_OyL端口位输出高电平	
			10: 满足条件时, TIM4_<=>_OyL端口位输出低电平	
			11: 满足条件时, TIM4_<=>_OyL端口位输出反转 (y=U、 V、 W)	
b25~b24	EOPUCL[1:0]	扩张向上计数端口状态L	条件: 计数器向上计数 & OCCRxI计数匹配 & OCCRxH计数匹配 (x=u、 v、 w)	R/W
			00: 满足条件时, TIM4_<=>_OyL端口位保持不变	
			01: 满足条件时, TIM4_<=>_OyL端口位输出高电平	
			10: 满足条件时, TIM4_<=>_OyL端口位输出低电平	
			11: 满足条件时, TIM4_<=>_OyL端口位输出反转 (y=U、 V、 W)	
b23~b22	EOPPKL[1:0]	扩张上溢点端口状态L	条件: 计数上溢 & OCCRxI计数匹配 & OCCRxH计数匹配 (x=u、 v、 w)	R/W
			00: 满足条件时, TIM4_<=>_OyL端口位保持不变	
			01: 满足条件时, TIM4_<=>_OyL端口位输出高电平	
			10: 满足条件时, TIM4_<=>_OyL端口位输出低电平	
			11: 满足条件时, TIM4_<=>_OyL端口位输出反转 (y=U、 V、 W)	
b21~b20	EOPDCL[1:0]	扩张向下计数端口状态L	条件: 计数器向下计数 & OCCRxI计数匹配 & OCCRxH计数匹配 (x=u、 v、 w)	R/W
			00: 满足条件时, TIM4_<=>_OyL端口位保持不变	
			01: 满足条件时, TIM4_<=>_OyL端口位输出高电平	
			10: 满足条件时, TIM4_<=>_OyL端口位输出低电平	
			11: 满足条件时, TIM4_<=>_OyL端口位输出反转 (y=U、 V、 W)	
b19~b18	EOPNUCL[1:0]	扩张向上计数端口状态L	条件: 计数器向上计数 & OCCRxI计数不匹配 & OCCRxH计数匹配 (x=u、 v、 w)	R/W
			00: 满足条件时, TIM4_<=>_OyL端口位保持不变	
			01: 满足条件时, TIM4_<=>_OyL端口位输出高电平	
			10: 满足条件时, TIM4_<=>_OyL端口位输出低电平	
			11: 满足条件时, TIM4_<=>_OyL端口位输出反转 (y=U、 V、 W)	
b17~b16	EOPNDCL[1:0]	扩张向下计数端口状态L	条件: 计数器向下计数 & OCCRxI计数不匹配 & OCCRxH计数匹配 (x=u、 v、 w)	R/W
			00: 满足条件时, TIM4_<=>_OyL端口位保持不变	
			01: 满足条件时, TIM4_<=>_OyL端口位输出高电平	
			10: 满足条件时, TIM4_<=>_OyL端口位输出低电平	
			11: 满足条件时, TIM4_<=>_OyL端口位输出反转 (y=U、 V、 W)	
b15~b14	OPNZRL[1:0]	下溢点端口状态	条件: 计数下溢 & OCCRxI计数不匹配 & OCCRxH计数不匹配	R/W

		L	(x=u、v、w)	
		00:	满足条件时, TIM4_<=>_OyL端口位保持不变	
		01:	满足条件时, TIM4_<=>_OyL端口位输出高电平	
		10:	满足条件时, TIM4_<=>_OyL端口位输出低电平	
		11:	满足条件时, TIM4_<=>_OyL端口位输出反转	
			(y=U、V、W)	
b13~b12	OPNPKL[1:0]	上溢点端口状态	条件: 计数上溢 & OCCRxI计数不匹配 & OCCRxH计数不匹配	R/W
		L	(x=u、v、w)	
		00:	满足条件时, TIM4_<=>_OyL端口位保持不变	
		01:	满足条件时, TIM4_<=>_OyL端口位输出高电平	
		10:	满足条件时, TIM4_<=>_OyL端口位输出低电平	
		11:	满足条件时, TIM4_<=>_OyL端口位输出反转	
			(y=U、V、W)	
b11~b10	OPZRL[1:0]	下溢点端口状态	条件: 计数下溢 & OCCRxI计数匹配 & OCCRxH计数不匹配	R/W
		L	(x=u、v、w)	
		00:	满足条件时, TIM4_<=>_OyL端口位保持不变	
		01:	满足条件时, TIM4_<=>_OyL端口位输出高电平	
		10:	满足条件时, TIM4_<=>_OyL端口位输出低电平	
		11:	满足条件时, TIM4_<=>_OyL端口位输出反转	
			(y=U、V、W)	
b9~b8	OPUCL[1:0]	向上计数端口状 态L	条件: 计数器向上计数 & OCCRxI计数匹配 & OCCRxH计数不匹配	R/W
			(x=u、v、w)	
		00:	满足条件时, TIM4_<=>_OyL端口位保持不变	
		01:	满足条件时, TIM4_<=>_OyL端口位输出高电平	
		10:	满足条件时, TIM4_<=>_OyL端口位输出低电平	
		11:	满足条件时, TIM4_<=>_OyL端口位输出反转	
			(y=U、V、W)	
b7~b6	OPPKL[1:0]	上溢点端口状态	条件: 计数上溢 & OCCRxI计数匹配 & OCCRxH计数不匹配	R/W
		L	(x=u、v、w)	
		00:	满足条件时, TIM4_<=>_OyL端口位保持不变	
		01:	满足条件时, TIM4_<=>_OyL端口位输出高电平	
		10:	满足条件时, TIM4_<=>_OyL端口位输出低电平	
		11:	满足条件时, TIM4_<=>_OyL端口位输出反转	
			(y=U、V、W)	
b5~b4	OPDCL[1:0]	向下计数端口状 态L	条件: 计数器向下计数 & OCCRxI计数匹配 & OCCRxH计数不匹配	R/W
			(x=u、v、w)	
		00:	满足条件时, TIM4_<=>_OyL端口位保持不变	
		01:	满足条件时, TIM4_<=>_OyL端口位输出高电平	
		10:	满足条件时, TIM4_<=>_OyL端口位输出低电平	
		11:	满足条件时, TIM4_<=>_OyL端口位输出反转	
			(y=U、V、W)	
b3	OCFZRL	下溢点OCFL状	条件: 计数下溢 & OCCRxI计数匹配 (x=u、v、w)	R/W

		态L	0: 满足条件时, OCSR.OCFL位保持不变 1: 满足条件时, OCSR.OCFL位置位	
b2	OCFUCL	向上计数OCFL	条件: 计数器向上计数 & OCCRxI计数匹配 (x=u、v、w)	R/W
		状态L	0: 满足条件时, OCSR.OCFL位保持不变 1: 满足条件时, OCSR.OCFL位置位	
b1	OCFPKL	上溢点OCFL状	条件: 计数上溢 & OCCRxI计数匹配 (x=u、v、w)	R/W
		态L	0: 满足条件时, OCSR.OCFL位保持不变 1: 满足条件时, OCSR.OCFL位置位	
b0	OCFDCL	向下计数OCFL	条件: 计数器向下计数 & OCCRxI计数匹配 (x=u、v、w)	R/W
		状态L	0: 满足条件时, OCSR.OCFL位保持不变 1: 满足条件时, OCSR.OCFL位置位	

注意:

- 从本地址区域读取数据时, 读取的并非缓冲器寄存器的值, 而是 OCMR 寄存器的值。
- TIM4_<t>_OyL 可以由 OCCRxI 与计数器的计数值(独立操作模式), 或者 OCCRxh 与计数器的计数值且 OCCRxI 与计数器的计数值(链接操作模式)来决定。向寄存器 OCMRxI 的 bit[31:20] 和 bit[15:4]写入相同的 12 位值, 同时将 OCMRxI[19:16]写入 “0000”, 此时, TIM4_<t>_OyL 输出将不受 OCCRxh 的影响, 而仅由 OCCRxI 决定。此模式叫做独立操作模式----通道 yH 由 OCCRxh 决定, 通道 yL 由 OCCRxI 决定配置。如果不满足上述条件的独立操作模式, 就是链接操作模式---通道 yL 输出同时受 OCCRxh 和 OCCRxI 影响。

(x=u、v、w, y=U、V、W)

19.5.9 专用比较基准寄存器 (TMR4_SCCRm)

复位值: 0x0000

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

位	标记	位名	功能	读写
b15~b0	SCCR[15:0]	专用比较基准值	专用比较基准值（比较启动模式）或延迟基准值（延迟启动模式） 注：从本地址区域读取数据时，读取的并非缓冲器寄存器的 值，而是SCCR寄存器的值	R/W

19.5.10 专用控制状态寄存器 (TMR4_SCSRm)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ZEN	UEN	PEN	DEN	-	-	EVT DS	EVT MS	-	-	LMC	EVTOS[2:0]			BUFEN[1:0]	

位	标记	位名	功能	读写
b15	ZEN	下溢点EVT使能	0: 计数下溢时, EVT无操作 1: 计数下溢时: EVTMS=0&SCCR比较匹配&SCMR设定匹配时, EVT启动输出 EVTMS=1&OCCR比较匹配&SCMR设定匹配时, EVT延时模式启动	R/W
b14	UEN	向上计数EVT使能	0: 向上计数时, EVT无操作 1: 向上计数时: EVTMS=0&SCCR比较匹配&SCMR设定匹配时, EVT启动输出 EVTMS=1&OCCR比较匹配&SCMR设定匹配时, EVT延时模式启动	R/W
b13	PEN	上溢点EVT使能	0: 计数上溢时, EVT无操作 1: 计数上溢时: EVTMS=0&SCCR比较匹配&SCMR设定匹配时, EVT启动输出 EVTMS=1&OCCR比较匹配&SCMR设定匹配时, EVT延时模式启动	R/W
b12	DEN	向下计数EVT使能	0: 向下计数时, EVT无操作 1: 向下计数时: EVTMS=0&SCCR比较匹配&SCMR设定匹配时, EVT启动输出 EVTMS=1&OCCR比较匹配&SCMR设定匹配时, EVT延时模式启动	R/W
b11~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9	EVTDS	EVT延时对象选择	0: 在延时启动模式时, OCCRxh作为延时比较匹配对象 1: 在延时启动模式时, OCCRxld作为延时比较匹配对象 (x=u、v、w) 注: 该位在EVTMS=0时无效	R/W
b8	EVTMS	EVT模式选择	0: 比较启动模式 (CNTR和SCCR的比较结果触发) 1: 延时启动模式 (比较匹配事件经SCCR延迟后触发)	R/W
b7~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5	LMC	周期间隔响应功能链接	0: 周期间隔响应功能链接无效, SCCR的缓存传送由BUFEN设定 1: 周期间隔响应功能链接有效, SCCR的缓存传送在BUFEN设定的基础上, 还必须满足CVPR.PIC[3:0]=0000 (计数上溢时) 或 CVPR.ZIC[3:0]=0000 (计数下溢时)	R/W

b4~b2	EVTOS[2:0]	EVT输出选择	000: Special Event 0的EVT输出有效 001: Special Evnet 1的EVT输出有效 010: Special Evnet 2的EVT输出有效 011: Special Evnet 3的EVT输出有效 100: Special Evnet 4的EVT输出有效 101: Special Evnet 5的EVT输出有效 请不要设定其他值	R/W
b1~b0	BUFEN[1:0]	SCCR&SCMR缓存传递	00: SCCR、SCMR缓存寄存器的值直接写入SCCR、SCMR 01: SCCR、SCMR缓存寄存器的值在计数下溢时写入SCCR、SCMR 10: SCCR、SCMR缓存寄存器的值在计数上溢时写入SCCR、SCMR 11: SCCR、SCMR缓存寄存器的值在计数下溢或上溢时写入SCCR、SCMR	R/W

19.5.11 专用模式控制寄存器 (TMR4_SCMRm)

复位值: 0xFF00

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

位	标记	位名	功能	读写
b15~b8	Reserved	-	读出时为“1”，写入时写“1”	R/W
b7	MPCE	周期间隔响应使能	0: 禁止AMC与CVPR.PIC比较 1: 使能AMC与CVPR.PIC比较	R/W
b6	MZCE	周期间隔响应使能	0: 禁止AMC与CVPR.ZIC比较 1: 使能AMC与CVPR.ZIC比较	R/W
b5~b4	Reserved	-	读出时为“0”，写入时写“0”	R/W
b3~b0	AMC[3:0]	专用事件输出周期间隔值	该位设定专用事件输出功能时的周期间隔值，在AMC和CVPR.PIC或CVPR.ZIC相等时，专用事件输出功能有效	R/W

注意:

- 从本地址区域读取数据时，读取的并非缓冲器寄存器的值，而是 SCMR 寄存器的值。

19.5.12 PWM 基本控制寄存器 (TMR4_POCRn)

复位值: 0xFF00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								LVLS[1:0]		PWMMD[1:0]		-	DIVCK[2:0]		

位	标记	位名	功能	读写
b15~b8	Reserved	-	读出时为“1”，写入时写“1”	R/W
b7~b6	LVLS[1:0]	PWM输出极性控制	00: TIM4_<1>_OxH和TIM4_<1>_OxL的输出都不反转 01: TIM4_<1>_OxH和TIM4_<1>_OxL的输出都反转 10: TIM4_<1>_OxH的输出反转，TIM4_<1>_OxL的输出不反转 11: TIM4_<1>_OxH的输出不反转，TIM4_<1>_OxL的输出反转	R/W
b5~b4	PWMMD[1:0]	PWM输出模式	00: 直通模式 01: 死区定时器模式 10: 死区定时器滤波模式 11: 设定禁止	R/W
b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2~b0	DIVCK[2:0]	计数时钟分频	该位指示滤波计数器和死区计数器的计数时钟分频 000: 计数时钟为PCLK1 001: 计数时钟为PCLK1/2 010: 计数时钟为PCLK1/4 011: 计数时钟为PCLK1/8 100: 计数时钟为PCLK1/16 101: 计数时钟为PCLK1/32 110: 计数时钟为PCLK1/64 111: 计数时钟为PCLK1/128	R/W

19.5.13 PWM 滤波控制寄存器 (TMR4_PFSRn)

复位值: 0x0000

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

位	标记	位名	功能	读写
b15~b0	PFSR[15:0]	滤波初始值	滤波计数初始值 注: 当PWM波形输出模式不选死区定时器滤波模式时, 16位滤波计数器用作的16位重载计数器, 此时16位滤波计数器可以周期性产生中断输出, 此功能与PWM波形生成器功能无关。	R/W

19.5.14 PWM 死区控制寄存器 (TMR4_PDARn)

复位值: 0x0000

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

位	标记	位名	功能	读写
b15~b0	PDA/BR[15:0]	死区初始值	死区计数初始值	R/W

19.5.15 重载控制状态寄存器 (TMR4_RCSR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RTS W	RTE W	RTIC W	RTIF W	RTS V	RTE V	RTIC V	RTIF V	RTS U	RTE U	RTIC U	RTIF U	-	RTID W	RTID V	RTID U

位	标记	位名	功能	读写
b15	RTSW	重载计数器停止W	0: 无操作 1: 停止重载计数器W并清除RTIFW 注: 该位在读出时始终为0	R/W
b14	RTEW	重载计数器启动W	0: 重载计数器W停止 1: 重载计数器W启动	R/W
b13	RTICW	清零计数匹配状态W	0: 无操作 1: 清零RTIFW标志位 注: 该位在读出时始终为0	R/W
b12	RTIFW	计数匹配状态W	0: 重载计数器计数值与PFSRw发生未比较匹配 1: 重载计数器计数值与PFSRw发生比较匹配	R
b11	RTSV	重载计数器停止V	0: 无操作 1: 停止重载计数器V并清除RTIFV 注: 该位在读出时始终为0	R/W
b10	RTEV	重载计数器启动V	0: 重载计数器V停止 1: 重载计数器V启动	R/W
b9	RTICV	清零计数匹配状态V	0: 无操作 1: 清零RTIFV标志位 注: 该位在读出时始终为0	R/W
b8	RTIFV	计数匹配状态V	0: 重载计数器计数值与PFSRv发生未比较匹配 1: 重载计数器计数值与PFSRv发生比较匹配	R
b7	RTSU	重载计数器停止U	0: 无操作 1: 停止重载计数器U并清除RTIFU 注: 该位在读出时始终为0	R/W
b6	RTEU	重载计数器启动U	0: 重载计数器U停止 1: 重载计数器U启动	R/W
b5	RTICU	清零计数匹配状态U	0: 无操作 1: 清零RTIFU标志位 注: 该位在读出时始终为0	R/W
b4	RTIFU	计数匹配状态U	0: 重载计数器计数值与PFSRu发生未比较匹配 1: 重载计数器计数值与PFSRu发生比较匹配	R
b3	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b2	RTIDW	重载中断屏蔽W	0: 重载功能有效时, 重载中断W输出有效 1: 重载功能有效时, 重载中断W输出无效	R/W

b1	RTIDV	重载中断屏蔽V	0: 重载功能有效时, 重载中断V输出有效 1: 重载功能有效时, 重载中断V输出无效	R/W
b0	RTIDU	重载中断屏蔽U	0: 重载功能有效时, 重载中断U输出有效 1: 重载功能有效时, 重载中断U输出无效	R/W

19.5.16 EMB 控制状态寄存器 (TMR4_ECSR)

复位值: 0x0000

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

位	标记	位名	功能	读写
b15~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	HOLD	PWM保持	0: 检测到EMB输入事件时, PWM正常输出 1: 检测到EMB输入事件时, 保持当前PWM输出状态, 不再变化	R/W
b6~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

19.5.17 EMB 扩展控制寄存器 (TMR4_ECER)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved														EMBVAL[1:0]	

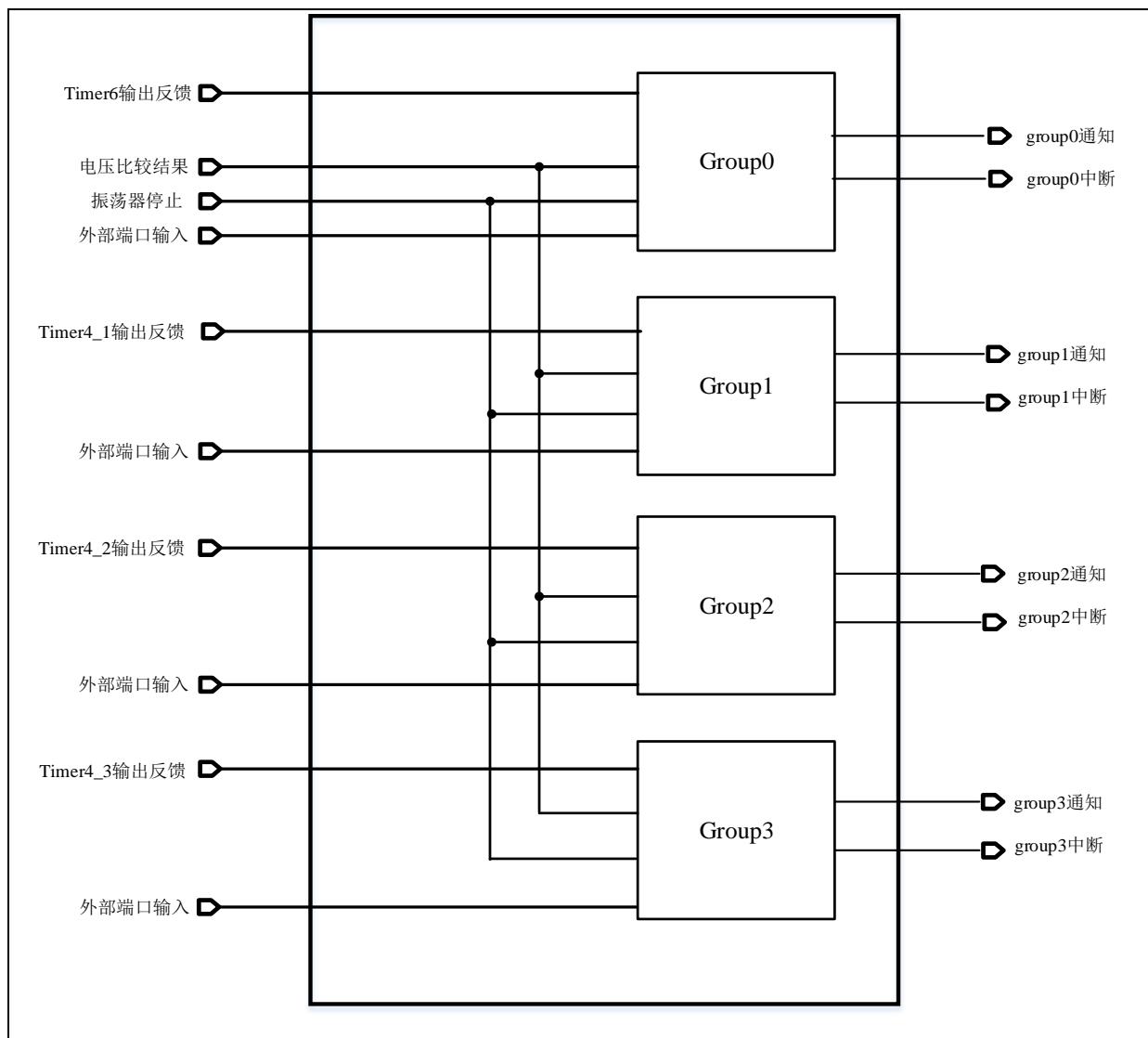
位	标记	位名	功能	读写
b15~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1~b0	EMBVAL[1:0]	EMB状态控制	00: 发生EMB事件时, PWM端口状态由ECSR.HOLD位设定决定 01: 发生EMB事件时, PWM端口输出变为HiZ 10: 发生EMB事件时, PWM端口输出固定为低电平 11: 发生EMB事件时, PWM端口输出固定为高电平	R/W

20 紧急刹车模块（EMB）

20.1 简介

紧急刹车模块是在满足一定条件时通知定时器，以使定时器停止向外部电机输出 PWM 信号的功能模块，下列事件用于产生通知：

- 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
- 外部振荡器停止振荡
- 写寄存器软件控制



20.2 功能描述

20.2.1 概述

EMB 用于向具有 PWM 功能的定时器模块(Timer4, Timer6)在满足一定条件时输出一个通知信号，通知定时器模块关闭当前 PWM 输出。EMB 模块具有 4 个组群(group)，其中 group0 对应 Timer6，由 Timer6 寄存器设定选择使用，group1~3 对应 Timer4 的 3 个单元。

20.2.2 外部端口输入电平变化时停止 PWM 信号输出

EMB 的每个组群分配一个端口用于实现当外部端口输入电平变化时停止 PWM 信号输出。端口分配如下表所示。

功能名	功能	对应定时器模块
EMB_IN1	group 0端口输入控制信号	Timer6
EMB_IN2	group 1端口输入控制信号	Timer4_1
EMB_IN3	group 2端口输入控制信号	Timer4_2
EMB_IN4	group 3端口输入控制信号	Timer4_3

EMB 根据控制寄存器 EMB_CTLx(x=0~3)的设定，可在端口电平为高(INVSEL=0)时或端口电平为低(INVSEL=1)时发送通知信号。同时，每个端口设有数字滤波功能，可根据需要设置滤波强度。当端口上发生符合条件的电平变化时，EMB 将产生通知信号通知 Timer6 和 Timer4，Timer6 和 Timer4 在接到通知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。同时 EMB 还能够产生中断请求。当端口电平转为无效后，用户可通过写 EMB 状态复位寄存器(EMB_STATCLR_x) (x=0~3)来清除通知信号，从而使 Timer6 和 Timer4 恢复输出 PWM 波。

20.2.3 PWM 输出端口电平发生同相（同高或同低）时停止 PWM 信号输出

EMB 能够监控 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 将产生通知信号给 Timer6 和 Timer4 。Timer6 和 Timer4 在接到通知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。同时 EMB 还能够产生中断请求。其中 group0 可用于监控 Timer6 的互补 PWM 输出信号，group1-3 可用于监

控 Timer4_1/ Timer4_2/ Timer4_3 的互补 PWM 输出信号。当端口的同高或同低情况解除，即 EMB 状态寄存器(EMB_STATx) (x=0~3)的 PWMSF 位复位后，用户可通过写 EMB 状态复位寄存器(EMB_STATCLR_x) (x=0~3)来清除通知信号，从而使 Timer6 和 Timer4 恢复输出 PWM 波。

端口名	功能	对应组群	EMB_CTL _x 控制位	EMB_PWMLV _x 控制位
TIM6Am(m=1~3)	Timer6的互补PWM输出信号	group0	PWMSEN[2:0]	PWMLV[2:0]
TIM6Bm(m=1~3)				
TIM4OUH_1	Timer4_1的互补PWM输出信号	group1	PWMSEN[2]	PWMLV[2]
TIM4OUL_1			PWMSEN[1]	PWMLV[1]
TIM4OVH_1			PWMSEN[0]	PWMLV[0]
TIM4OVL_1			PWMSEN[2]	PWMLV[2]
TIM4OWH_1			PWMSEN[1]	PWMLV[1]
TIM4OWL_1			PWMSEN[0]	PWMLV[0]
TIM4OUH_2	Timer4_2的互补PWM输出信号	group2	PWMSEN[2]	PWMLV[2]
TIM4OUL_2			PWMSEN[1]	PWMLV[1]
TIM4OVH_2			PWMSEN[0]	PWMLV[0]
TIM4OVL_2			PWMSEN[2]	PWMLV[2]
TIM4OWH_2			PWMSEN[1]	PWMLV[1]
TIM4OWL_2			PWMSEN[0]	PWMLV[0]
TIM4OUH_3	Timer4_3的互补PWM输出信号	group3	PWMSEN[2]	PWMLV[2]
TIM4OUL_3			PWMSEN[1]	PWMLV[1]
TIM4OVH_3			PWMSEN[0]	PWMLV[0]
TIM4OVL_3			PWMSEN[2]	PWMLV[2]
TIM4OWH_3			PWMSEN[1]	PWMLV[1]
TIM4OWL_3			PWMSEN[0]	PWMLV[0]

20.2.4 根据电压比较器比较结果停止 PWM 信号输出

EMB 的每个组群能够根据电压比较器的比较结果向 Timer6 和 Timer4 发送通知信号，电压比较器输出结果的设定请参考电压比较器章节。当电压比较器比较结果标志位置起时，EMB 将产生通知信号给 Timer6 和 Timer4 。Timer6 和 Timer4 在接到通知后可

根据寄存器设定将输出端口置为高电平，低电平或高阻态。同时 EMB 还能够产生中断请求。当电压比较器标志位复位后，用户可通过写 EMB 状态复位寄存器 (EMB_STATCLR x) ($x=0\sim3$) 来清除通知信号，从而使 Timer6 和 Timer4 恢复输出 PWM 波。

20.2.5 外部振荡器停止振荡时停止 PWM 信号输出

EMB 的每个组群能够在外部振荡器停止震荡时向 Timer6 和 Timer4 发送通知信号，外部振荡器停止震荡的设定请参考电压比较器章节。当外部振荡器停止震荡标志位置起时，EMB 将产生通知信号给 Timer6 和 Timer4。Timer6 和 Timer4 在接到通知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。同时 EMB 还能够产生中断请求。当外部振荡器停止震荡标志位复位后，用户可通过写 EMB 状态复位寄存器 (EMB_STATCLR x) ($x=0\sim3$) 来清除通知信号，从而使 Timer6 和 Timer4 恢复输出 PWM 波。

20.2.6 写寄存器软件控制 PWM 信号输出

EMB 的软件输出使能控制寄存器(EMB_SOEx)($x=0\sim3$)能够允许用户通过软件直接置位和复位的方式向 Timer6 和 Timer4 发送通知信号，软件控制 PWM 输出时不会产生中断请求。

20.3 寄存器说明

寄存器一览

名称	英文缩写	说明	偏移地址
EMB输出控制寄存器	EMB_CTL	控制各个输出使能的条件	0x0
EMB反馈电平选择寄存器	EMB_PWMLV	选择PWM反馈信号的有效电平	0x4
EMB软件输出使能控制寄存器	EMB_SOE	软件产生输出禁止事件	0x8
EMB状态寄存器	EMB_STAT	表示输出使能的状态	0xC
EMB状态复位寄存器	EMB_STATCLR	清除输出使能状态	0x10
EMB中断许可寄存器	EMB_INTEN	控制中断使能	0x14

20.3.1 EMB 控制寄存器 0(EMB_CTL0)

该寄存器为单次写入寄存器，即复位之后仅能够写入一次。

地址: 0x4001_7C00

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
INVS EL	NFE N	NFSEL[1:0]													Reserved
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
									PWM SEN2	PWM SEN1	PWM SENO	OSCS TPEN	Res.	CMPEN[2:0]	PORT INEN

位	标记	位名	功能	读写
b31	INVSEL	端口输入有效电平选择	0: 高电平有效 1: 低电平有效	R/W
b30	NFEN	端口输入数字滤波器使能	0: 滤波器无效 1: 滤波器有效	R/W
b29~b28	NFSEL[1:0]	数字滤波器滤波时钟选择	00: 使用总线时钟滤波 01: 使用总线时钟的8分频滤波 10: 使用总线时钟的32分频滤波 11: 使用总线时钟的128分频滤波	R/W
b27~b9	Reserved		读时读出0, 写时请写0	R
b8~b6	PWMSEN[2:0]	TIM6A/Bm(m=1~3)短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	R/W
b5	OSCSTPEN	b6对应Timer6_1; b7对应Timer6_2; b8对Timer6_3;		
b4	Reserved	振荡器停止输出控制使能	0: 振荡器停止振荡时输出控制无效 1: 振荡器停止振荡时输出控制有效	R/W
b3~b1	CMPEN[2:0]	电压比较器比较结果控制使能	0: 电压比较器比较结果输出控制无效 1: 电压比较器比较结果输出控制有效	R/W
b0	PORTINEN	b1对应CMP1; b2对应CMP2; b3对应CMP3;	端口输入控制使能	R/W
			0: 端口输入控制无效 1: 端口输入控制有效	

20.3.2 EMB 控制寄存器 1~3(EMB_CTL1~3)

该寄存器为单次写入寄存器，即复位之后仅能够写入一次。

地址：0x4001_7C20, 0x4001_7C40, 0x4001_7C60

复位值：0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
INVSEL	NFEN	Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								PWM SEN2	PWM SEN1	PWM SEN0	OSCST PEN	Res.	CMPE[2:0]		PORTIN EN

位	标记	位名	功能	读写
b31	INVSEL	端口输入有效电平选择	0: 高电平有效 1: 低电平有效	R/W
b30	NFEN	端口输入数字滤波器使能	0: 滤波器无效 1: 滤波器有效	R/W
b29~b28	NFSEL[1:0]	数字滤波器滤波时钟选择	00: 使用总线时钟滤波 01: 使用总线时钟的8分频滤波 10: 使用总线时钟的32分频滤波 11: 使用总线时钟的128分频滤波	R/W
b27~b9	Reserved		读时读出0, 写时请写0	R
b8	PWMSEN2	TIM4_x(x=1~3)的U相位短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	R/W
b7	PWMSEN1	TIM4_x(x=1~3)的V相位短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	R/W
b6	PWMSEN0	TIM4_x(x=1~3)的W相位短路输出控制使能	0: 短路时输出控制无效 1: 短路时输出控制有效	R/W
b5	OSCSTPEN	振荡器停止输出控制使能	0: 振荡器停止振荡时输出控制无效 1: 振荡器停止振荡时输出控制有效	R/W
b4	Reserved		读时读出0, 写时请写0	R
b3~b1	CMPE[2:0]	电压比较器比较结果 控制使能 b1对应CMP1; b2对应CMP2; b3对应CMP3;	0: 电压比较器比较结果输出控制无效 1: 电压比较器比较结果输出控制有效	R/W
b0	PORTINEN	端口输入控制使能	0: 端口输入控制无效 1: 端口输入控制有效	R/W

20.3.3 EMB 反馈电平选择寄存器 0(EMB_PWMLV0)

该寄存器为单次写入寄存器，即复位之后仅能够写入一次

地址：0x4001_7C04

复位值：0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
												PWMLV 2	PWMLV 1	PWMLV 0	

位	标记	位名	功能	读写
b31~3	Reserved		读时读出0, 写时请写0	R
		TIM6A/Bm(m=1~3)输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	R/W
b2~b0	PWMLV[2:0]	b0对应Timer6_1; b1对应Timer6_2; b2对Timer6_3;		

20.3.4 EMB 反馈电平选择寄存器 1~3(EMB_PWMLV1~3)

该寄存器为单次写入寄存器，即复位之后仅能够写入一次

地址: 0x4001_7C24, 0x4001_7C44, 0x4001_7C64

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												PWM LV2	PWM LV1	PWM LV0	

位	标记	位名	功能	读写
b31~3	Reserved		读时读出0, 写时请写0	R
b2	PWMLV2	TIM4_x(x=1~3)的U相 位输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	R/W
b1	PWMLV1	TIM4_x(x=1~3)的V相 位输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	R/W
b0	PWMLV0	TIM4_x(x=1~3)的W相 位输出有效电平选择	0: 低电平为有效电平 1: 高电平为有效电平	R/W

20.3.5 EMB 软件输出使能控制寄存器(EMB_SOEx)(x=0~3)

地址: 0x4001_7C08, 0x4001_7C28, 0x4001_7C48, 0x4001_7C68

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~1	Reserved		读时读出0, 写时请写0	R
b0	SOE	软件控制输出	0: PWM正常输出 1: PWM停止输出	R/W

20.3.6 EMB 状态寄存器(EMB_STATx) (x=0~3)

地址: 0x4001_7C0C, 0x4001_7C2C, 0x4001_7C4C, 0x4001_7C6C

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
PWM ST	PORT INST	OSF	CMP F	PWM SF	POR TINF										

位	标记	位名	功能	读写
b31~6	Reserved		读时读出0, 写时请写0	R
b5	PWMST	PWM输出状态	0: 没有发生PWM输出同相 1: 发生PWM输出同相	R
b4	PORTINST	端口输入控制状态	0: 端口输入控制处于无效状态 1: 端口输入控制处于有效状态	R
b3	OSF	振荡器停止震荡停止 PWM输出的状态	0: 当前没有因振荡器停止震荡而停止PWM输出 1: 当前因振荡器停止震荡而停止PWM输出	R
b2	CMPF	电压比较器停止PWM 输出的状态	0: 当前没有因电压比较器比较结果而停止PWM输出 1: 当前因电压比较器比较结果而停止PWM输出	R
b1	PWMSF	PWM输出同相位停止 PWM输出的状态	0: 当前没有因PWM输出反馈同相位而停止PWM输出 1: 当前因PWM输出反馈同相位而停止PWM输出	R
b0	PORTINF	端口输入控制停止 PWM输出的状态	0: 当前没有因端口输入控制而停止PWM输出 1: 当前因端口输入控制而停止PWM输出	R

20.3.7 EMB 状态复位寄存器(EMB_STATCLR_x) (x=0~3)

地址: 0x4001_7C10, 0x4001_7C30, 0x4001_7C50, 0x4001_7C70

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
OSF CLR	CMP FCLR	PWM SFCLR	PORTI NFCLR												

位	标记	位名	功能	读写
b31~4	Reserved		读时读出0, 写时请写0	R
	复位		0: 无任何效果	W
b3	OSFCLR	EMB_STAT.OSF	1: 当CMU.MOSCSTP=0时, 清除EMB_STAT.OSF位并恢复因振荡器停止震荡而停止的PWM输出	
	复位		0: 无任何效果	W
b2	CMPFCLR	EMB_STAT.CMPF	1: 当CMPMONn. OMON=0时, 清除EMB_STAT. CMPF位并恢复因电压比较器比较结果而停止的PWM输出	
	复位		0: 无任何效果	W
b1	PWMSFCLR	EMB_STAT.PWMSF	1: 当EMB_STAT.PWMST=0时, 清除EMB_STAT. PWMSF位并恢复因PWM输出反馈同相位而停止的PWM输出	
	复位		0: 无任何效果	W
b0	PORTINFCLR	EMB_STAT.PORTINF	1: 当EMB_STAT.PORTINST=0时, 清除EMB_STAT. PORTINF位并恢复因端口输入控制而停止的PWM输出	

20.3.8 EMB 中断许可寄存器(EMB_INTENx)(x=0~3)

地址: 0x4001_7C14, 0x4001_7C34, 0x4001_7C54, 0x4001_7C74

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~4	Reserved		读时读出0, 写时请写0	R
b3	OSINTEN	振荡器停止震荡停止 PWM的中断使能	0: 振荡器停止震荡停止PWM时不产生中断 1: 振荡器停止震荡停止PWM时产生中断	R/W
b2	CMPINTEN	电压比较器停止PWM 的中断使能	0: 电压比较器比较结果停止PWM时不产生中断 1: 电压比较器比较结果停止PWM时产生中断	R/W
b1	PWMINTEN	PWM输出同相位停止 PWM的中断使能	0: PWM输出同相位停止PWM时不产生中断 1: PWM输出同相位停止PWM时产生中断	R/W
b0	PORTINTEN	端口输入控制停止 PWM的中断使能	0: 端口输入控制停止PWM时不产生中断 1: 端口输入控制停止PWM时产生中断	R/W

21 通用定时器 (TimerA)

21.1 简介

通用定时器 A (TimerA) 是一个具有 16 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；支持计数器同步启动；比较基准值寄存器支持缓存功能；支持单元间级联实现 32 位计数；支持 2 相正交编码计数和 3 相正交编码计数。本系列产品搭载 6 个单元 TimerA，最大可实现 48 路 PWM 输出。

21.2 基本框图

TimerA 基本的功能及特性如表 21-1 所示。

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

表 21-1 TimerA 的基本功能及特性

TimerA 的基本框图如图 21-1 所示。图中”<t>”为单元编号，即”<t>”为 1~6，本章节后文提到“<t>”时均指单元编号，不再赘述。

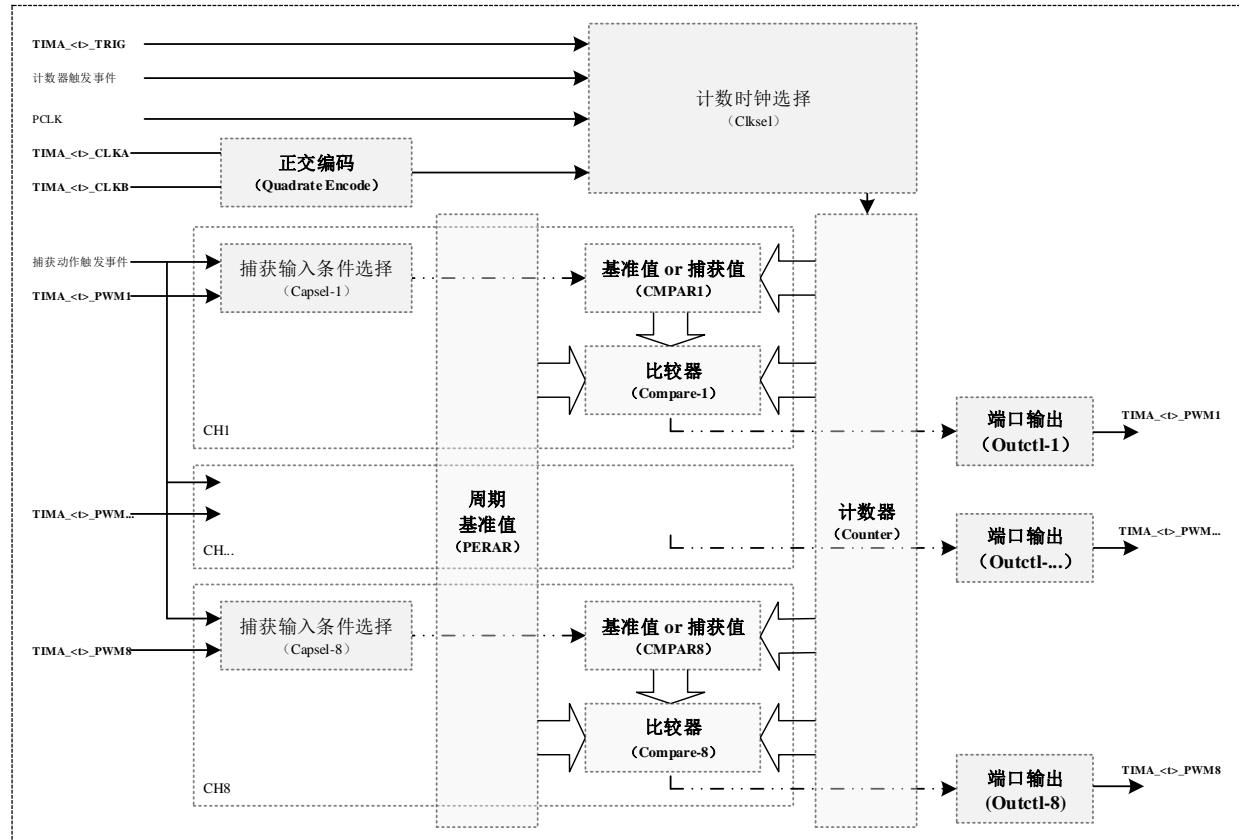


图 21-1 TimerA 基本框图

表 21-2 所示为 TimerA 的输入输出端口列表。

端口名	方向	功能
TIMA_<t>_PWMm	in or out	捕获输入事件端口或PWM输出端口 (m=1~8)
TIMA_<t>_CLKA	in	正交编码计数事件输入端口
TIMA_<t>_CLKB		
TIMA_<t>_TRIG	in	硬件触发启动、停止、清零事件输入端口

表 21-2 TimerA 端口列表

21.3 功能说明

21.3.1 基本动作

21.3.1.1 波形模式

TimerA 有 2 种基本计数波形模式，锯齿波模式和三角波模式。两种波形模式的基本波形如图 21-2、图 21-3 所示。

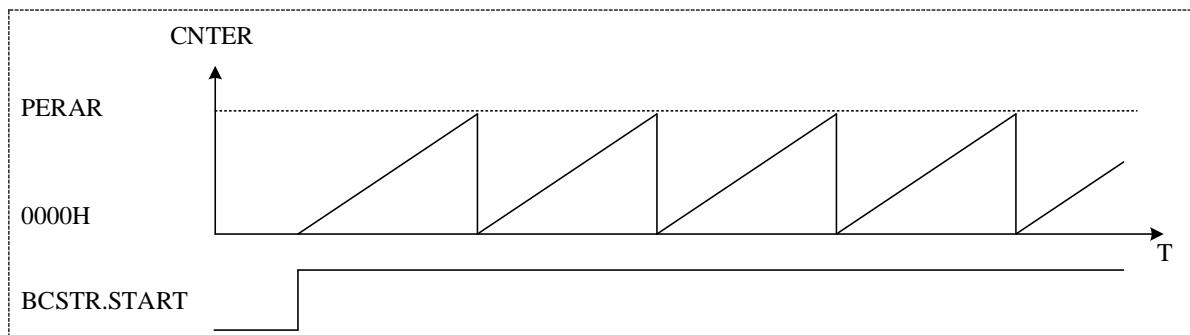


图 21-2 锯齿波波形 (递加计数)

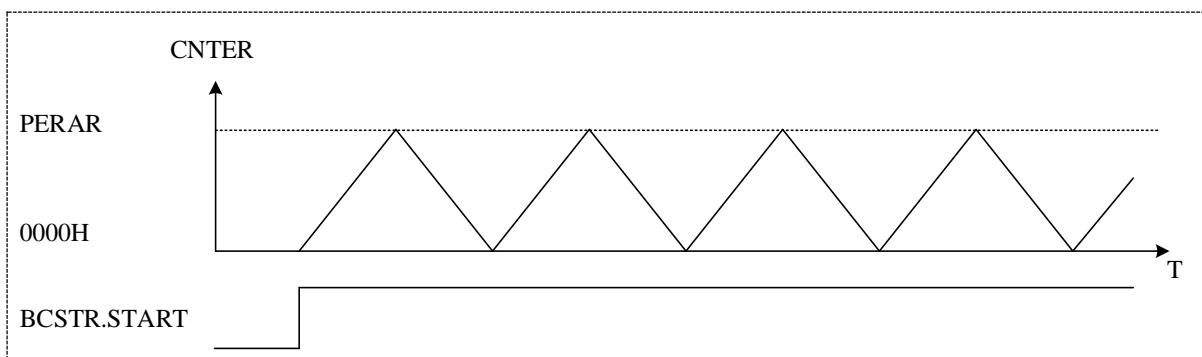


图 21-3 三角波波形

21.3.1.2 比较输出

各个 TimerA 单元内部均含有 8 个通道的比较输出 (TIMA_<t>_PWMn)，可在计数值与比较基准值比较匹配时输出指定的电平。TMRA_CMPARn 寄存器分别对应了 TIMA_<t>_PWMn 输出端口的计数比较基准值。当定时器的计数值和 TMRA_CMPARn 相等时，TIMA_<t>_PWMn 端口输出指定的电平 (n=1~8)。

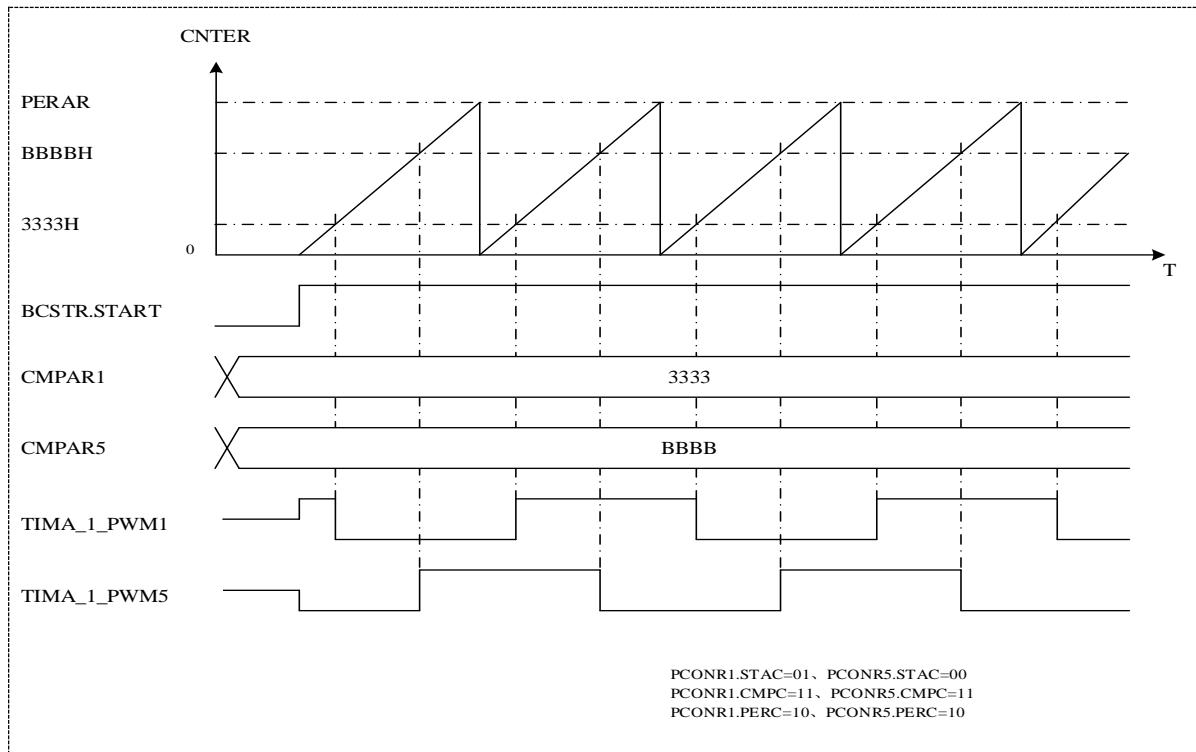


图 21-4 比较输出动作

TIMA_<t>_PWMn 端口的计数开始时的电平、计数停止时的电平、计数比较匹配时的电平、计数周期匹配时的电平等，可通过端口控制寄存器（PCONRn）的 STAC、STPC、CMPC、PERC、FORC 位设定控制（n=1~8）。图 21-4 为单元 1 的比较输出动作例。

21.3.1.3 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（CCONRn）的 CCONR.CAPMD 位为 1，捕获输入功能变为有效。此时选择对应的捕获输入条件且该条件有效时，当前的计数值就被保存到相应的寄存器（CMPARn）中（n=1~8）。

捕获输入条件可以选择内部捕获动作触发事件（通过 HTSSR1 寄存器选择）、TIMA_<t>_PWMn 端口输入等，具体的条件选择可通过捕获控制寄存器（CCONRn）的 HICP 位来设定（n=1~8）。图 21-5 为单元 2 的捕获输入动作例。

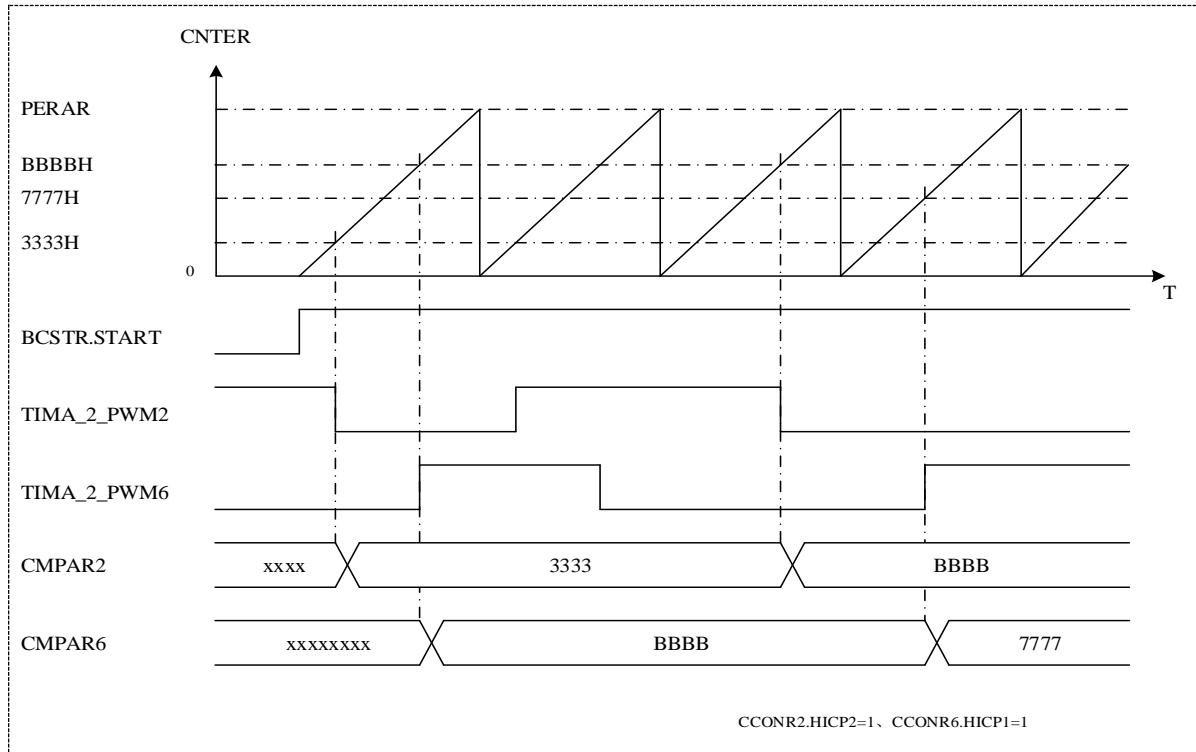


图 21-5 捕获输入动作

21.3.2 时钟源选择

TimerA 的计数时钟可以有以下几种选择：

- a) PCLK1 的 1、2、4、8、16、32、64、128、256、512、1024 分频 (BCSTR.CKDIV[3:0] 设定)
- b) TIMA_<t>_TRIG 端口事件输入 (HCUPR[9:8]或 HCDOR[9:8]设定)
- c) 内部计数器触发事件输入 (HCUPR[10]或 HCDOR[10]设定)
- d) 对称单元的计数上溢或计数下溢事件输入 (HCUPR[12:11]或 HCDOR[12:11]设定)
- e) TIMA_<t>_CLKA、TIMA_<t>_CLKB 的端口正交编码输入 (HCUPR[7:0]或 HCDOR[7:0]设定)

计数时钟源选择 a 时为软件计数模式，计数时钟源选择 b、c、d、e 时为硬件计数模式。

计数时钟选择 d 时多用于三相正交编码计数的公转计数模式（参见[位置溢出计数]和 [混合计数]章节），也可用于级联计数。上述描述可以看到，b、c、d、e 时钟互相独立，可分别设定有效或无效，并且当选择 b、c、d、e 时钟时，a 时钟自动无效。

21.3.3 同步启动

本产品所搭载的 6 个单元的 TimerA，可以实现软件同步启动或硬件同步启动。单元 2~单元 6 可以选择与单元 1 实现同步启动。当单元 2~单元 6 中的 BCSTR.SYNST 位设定为 1 时，对应单元与单元 1 的同步启动功能有效。此时，若软件设定单元 1 的 BCSTR.START 位为 1，被同步单元（单元 2~单元 6）的计数器开始软件同步计数；若硬件设定单元 1 的 HCONR.HSTA1~0 中任意位为 1，且单元 1 的对应硬件事件发生时，被同步单元（单元 2~单元 6）的计数器开始硬件同步计数。在选择硬件同步计数启动功能时，被同步单元（单元 2~单元 6）的 HCONR.HSTA1~0 的对应位也必须设定为有效。

图 21-6 为设定单元 2、单元 4、单元 5 的 BCSTR.SYNST=1 时的软件同步启动例。

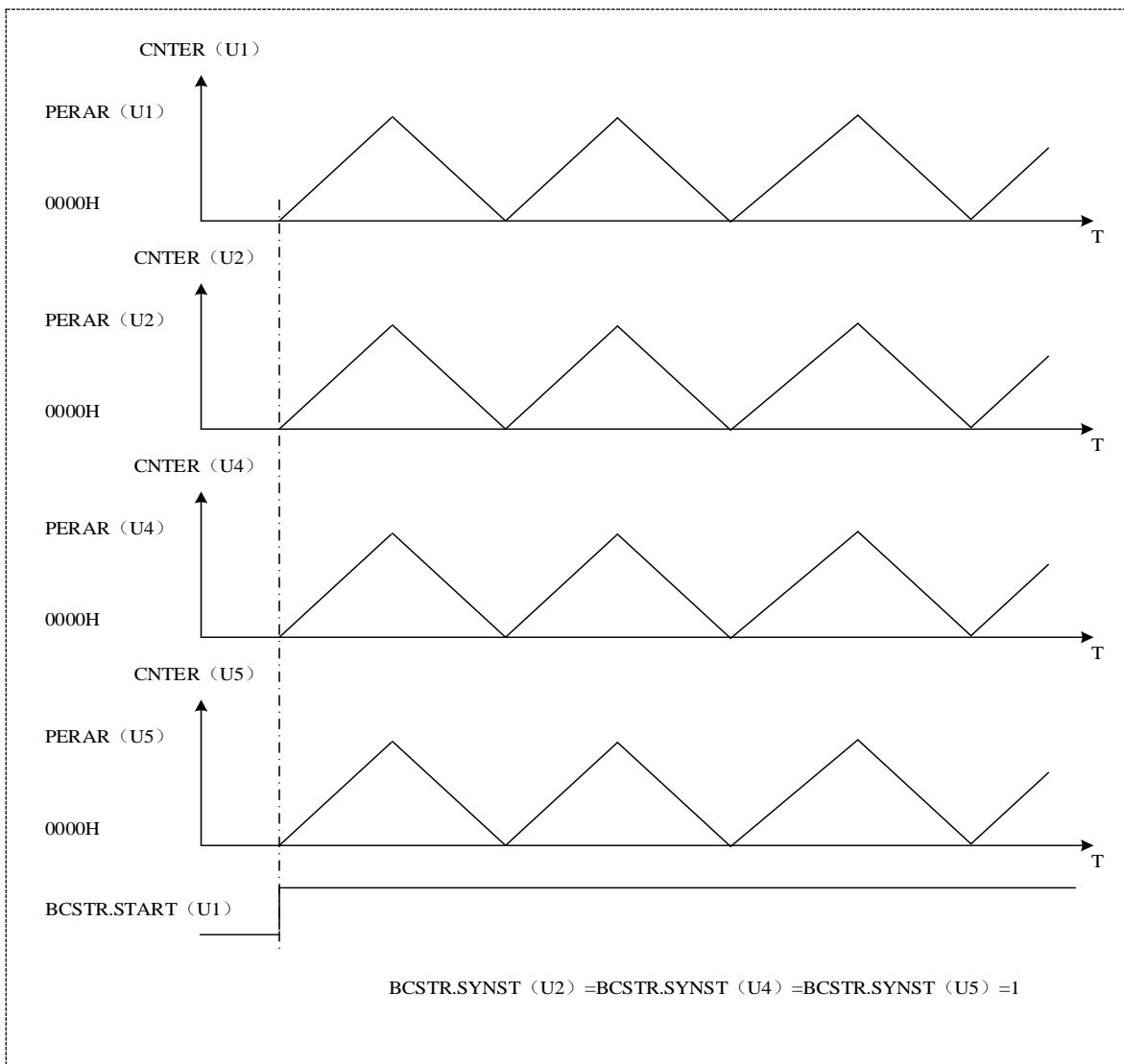


图 21-6 软件同步动作

21.3.4 数字滤波

各个单元的 $\text{TIMA}_{<\text{t}>}_\text{CLKA}$ 、 $\text{TIMA}_{<\text{t}>}_\text{CLKB}$ 、 $\text{TIMA}_{<\text{t}>}_\text{TRIG}$ 、 $\text{TIMA}_{<\text{t}>}_\text{PWMn}$ （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（CCONR n ）的对应位来实现（ $n=1\sim 8$ ）。

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。图 21-7 所示为单元 1 的 CLKA 端口滤波动作例。

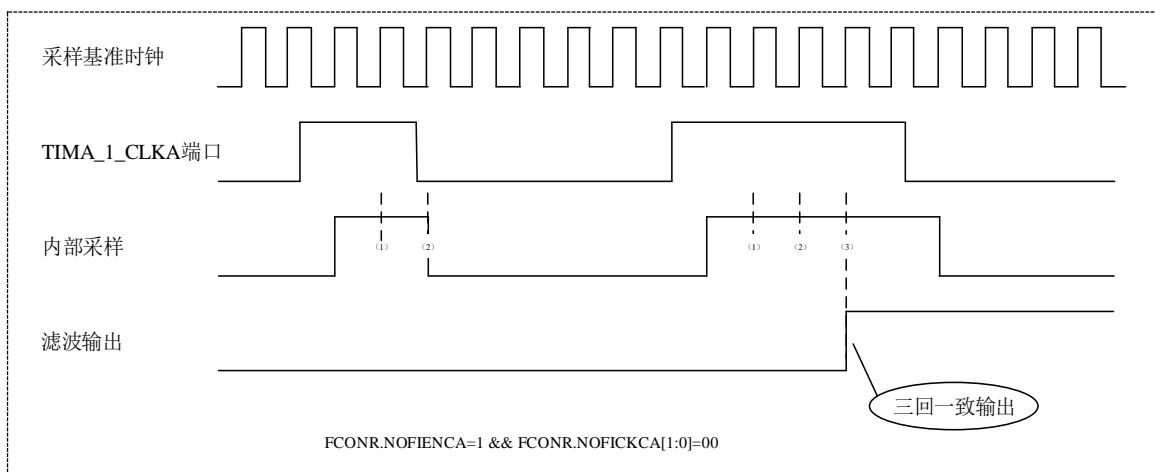


图 21-7 时钟输入端口的滤波功能

21.3.5 缓存功能

TimerA 的共计 8 个比较基准寄存器（CMPAR n ）可以成对实现缓存功能（ $n=1\sim 8$ ）。即 CMPAR2 作为 CMPAR1 的缓存基准值、CMPAR4 作为 CMPAR3 的缓存基准值、CMPAR6 作为 CMPAR5 的缓存基准值、CMPAR8 作为 CMPAR7 的缓存基准值。缓存控制寄存器（BCONR m ）分别实现对四组缓存功能的控制（ $m=1\sim 4$ ）。

当缓存控制寄存器（BCONR m ）的 BEN 位被置位时，缓存功能变为有效（ $m=1\sim 4$ ）。计数器计数到特定时间点时就发生一次缓存传送（ $\text{CMPAR8/6/4/2} \rightarrow \text{CMPAR7/5/3/1}$ ）。该“特定时间点”有以下几种情况：

- a) 硬件计数模式时，计数到上溢点（BCSTR.DIR=1 时）或下溢点（BCSTR.DIR=0 时）
- b) 锯齿波计数模式（BCSTR.MODE=0）时，计数器计数到上溢点（BCSTR.DIR=1 时）

时) 或下溢点 (BCSTR.DIR=0 时)

- c) 三角波计数模式 (BCSTR.MODE=1) 时, 计数到峰点 (BCSTR.DIR=1 && BCONRn.BSE0=1 时) (n=1~4)
- d) 三角波计数模式 (BCSTR.MODE=1) 时, 计数到谷点 (BCSTR.DIR=0 && BCONRn.BSE1=1 时) (n=1~4)
- e) 硬件计数模式或锯齿波计数模式时, 发生清零动作

下图 21-8 所示, 是单元 2 锯齿波模式时的缓存传送示意图。

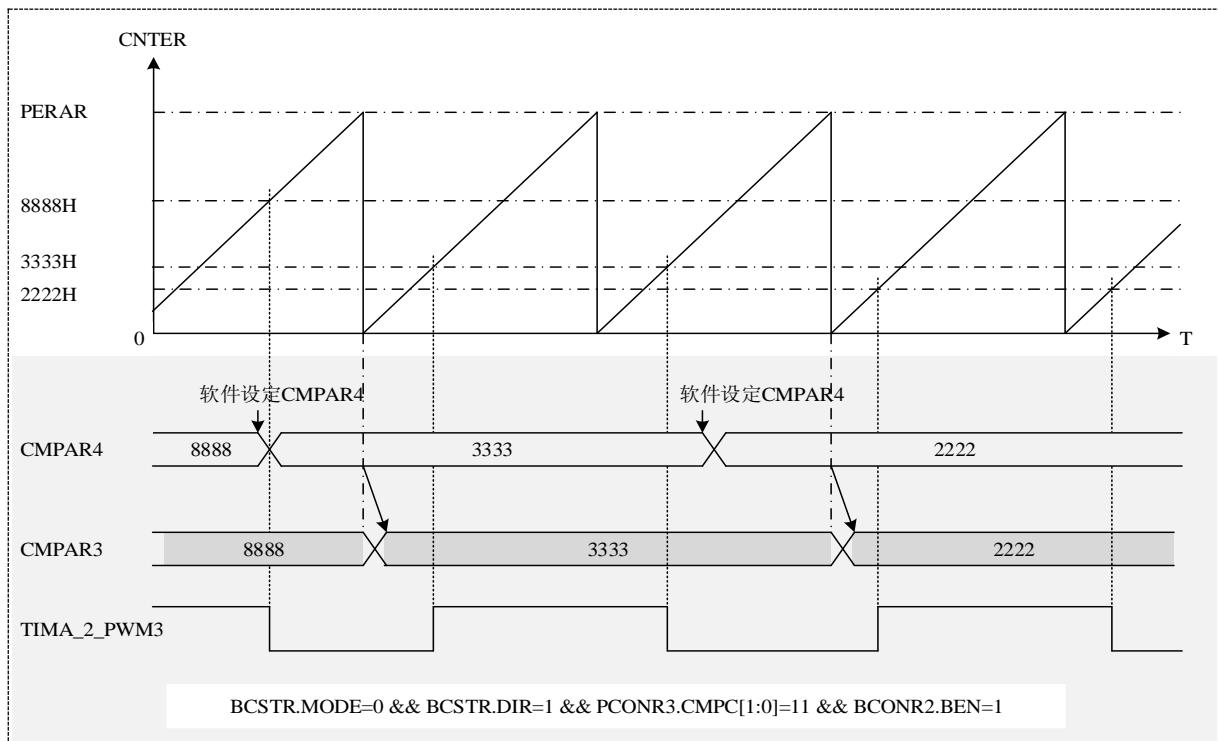


图 21-8 锯齿波模式时缓存动作

21.3.6 级联计数

在计数时钟源选择章节中, 当时钟源选择 d) 时, 本单元的计数时钟就选择为对称单元的计数溢出(计数上溢或计数下溢)事件, 此时两个单元级联合并可实现 32 位计数器。

在级联计数中, 对称单元的 CNTER 为低 16 位计数器, 本单元的 CNTER 为高 16 位计数器。

例如, 在三角波向上计数模式 (BCSTR.MODE=0、BCSTR.DIR=1) 时, 设定单元 1 的计数时钟为 PCLK1, 设定单元 2 的计数时钟源为单元 1 的计数上溢事件 (单元 2 的 TMRA_HCUPR.HCUP11=1), 启动单元 1、2 计数 (先启动单元 2, 再启动单元 1) 就

实现级联计数。单元 1 的 CNTER 位低 16 位计数器，单元 2 的 CNTER 为高 16 位计数器。如图 21-9 所示是单元 1、2 级联计数的示意图。

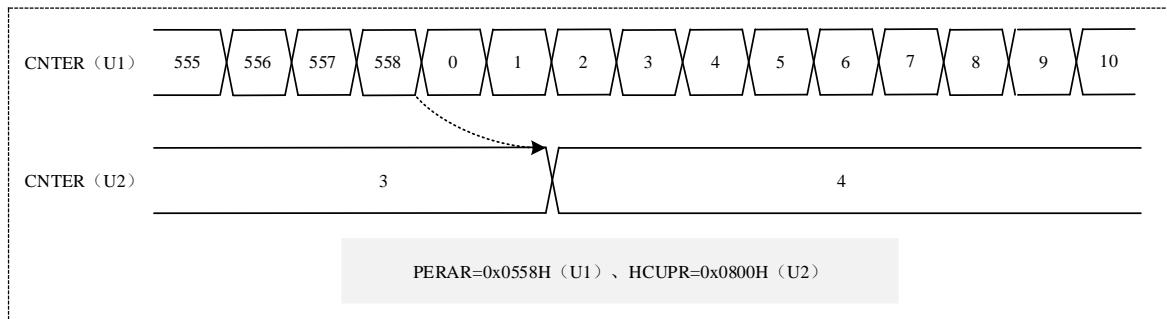


图 21-9 32 位级联计数动作

21.3.7 PWM 输出

21.3.7.1 通用 PWM 输出

TimerA 内部的 8 路输出 $\text{TIMA}_{<t>}\text{PWM}_n$ 都可以通过端口控制寄存器 (PCONR_n) 的相关控制位实现不同的输出波形 ($n=1\sim 8$)。

图 21-10 所示，是单元 1 三角波模式下，通道 1、2、6、7 的 PWM 输出波形例。

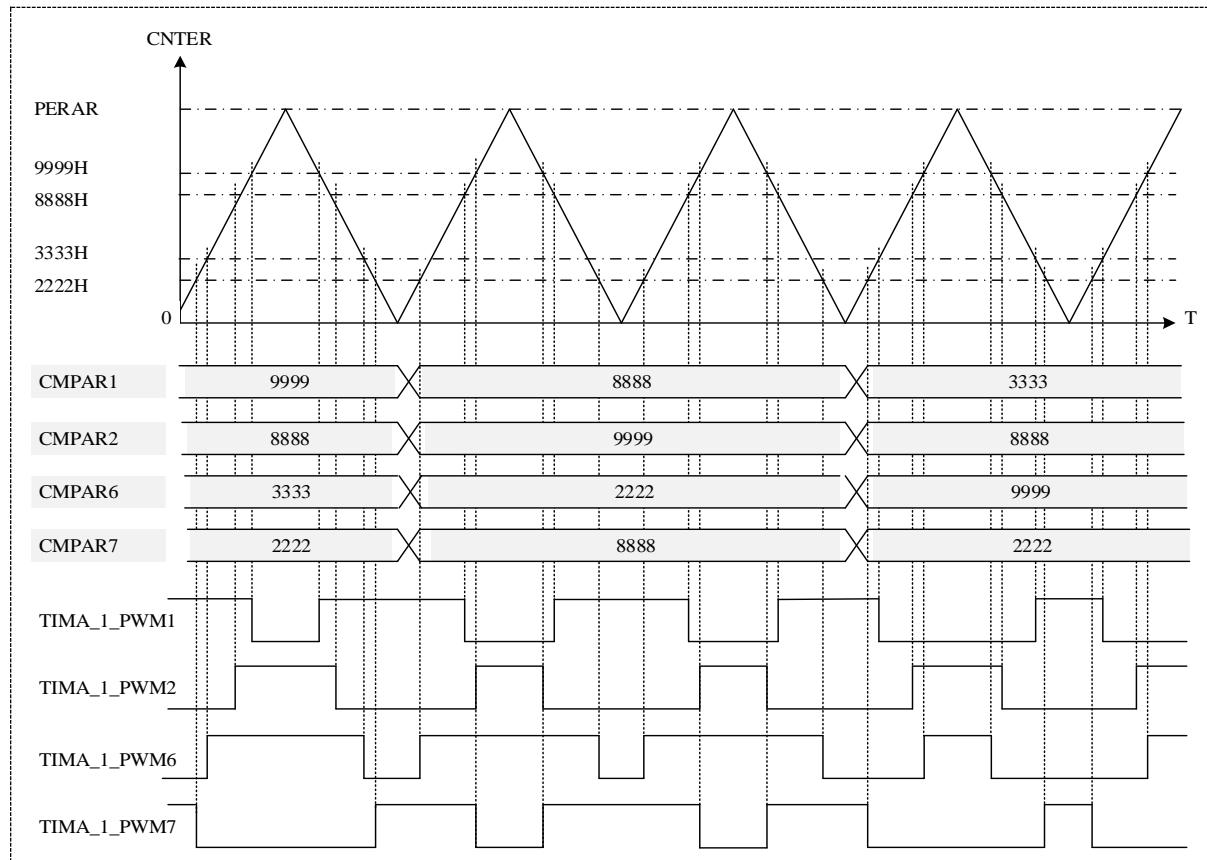


图 21-10 通用 PWM 输出例

21.3.7.2 单脉冲 PWM 输出

锯齿波计数模式时，设定在比较基准值比较匹配时翻转（PCONR.CMPC=11）、在周期基准值比较匹配时翻转（PCONR.PERC=11），就可实现在一个周期内产生单脉冲的 PWM 输出。

图 21-12 所示，是单元 1 三角波模式下，单脉冲 PWM 输出波形例。

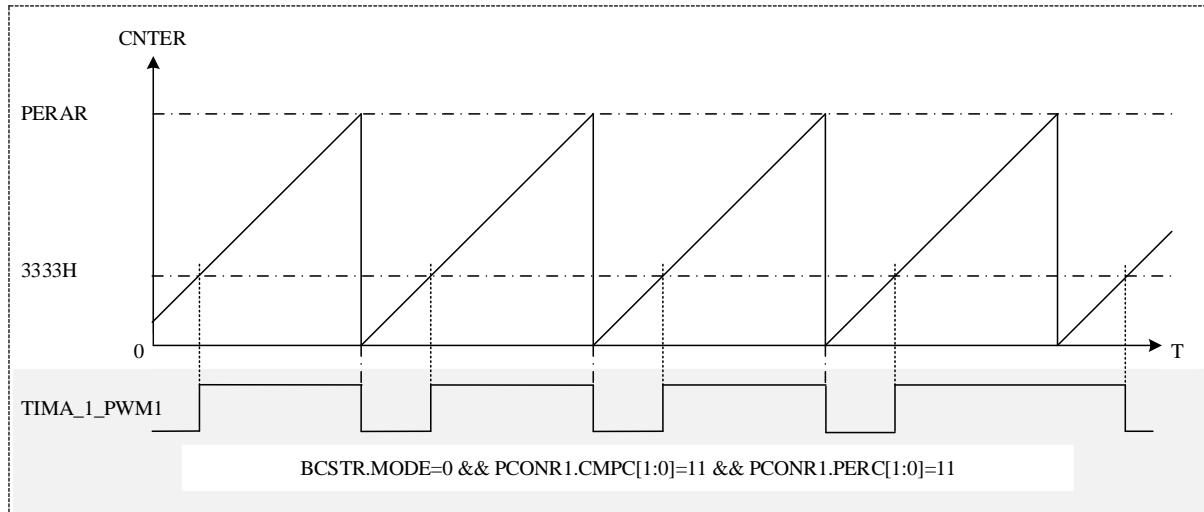


图 21-11 单脉冲 PWM 输出

21.3.8 正交编码计数

将 $\text{TIMA}_{<t>}_{_}\text{CLKA}$ 输入看作 AIN 输入、 $\text{TIMA}_{<t>}_{_}\text{CLKB}$ 输入看作 BIN 输入、 $\text{TIMA}_{<t>}_{_}\text{TRIG}$ 输入看作 ZIN 输入，TimerA 就可以实现三路输入的正交编码计数。每个单元的 AIN、BIN 单独动作可以实现位置计数模式；两个单元的 AIN、BIN、ZIN 组合动作可以实现公转计数模式，其中用于位置计数的单元称之为位置计数单元、用于公转计数的单元称之为公转计数单元。公转计数模式时，每两个单元间互相组合（单元 1、2 组合；单元 3、4 组合；单元 5、6 组合），组合内位置计数单元和公转计数单元可以任意指定。

AIN 和 BIN 的计数条件使能通过设定硬件递加事件选择寄存器（HCUPR）和硬件递减事件选择寄存器（HCDOR）中 $\text{TIMA}_{<t>}_{_}\text{CLKA}$ 和 $\text{TIMA}_{<t>}_{_}\text{CLKB}$ 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。

21.3.8.1 位置计数模式

正交编码位置计数模式，是指根据 AIN、BIN 的输入实现基本计数功能、相位差计数功能和方向计数功能。

基本计数

基本计数动作是根据 AIN 或 BIN 端口的输入时钟进行计数，如下图 21-11 所示。

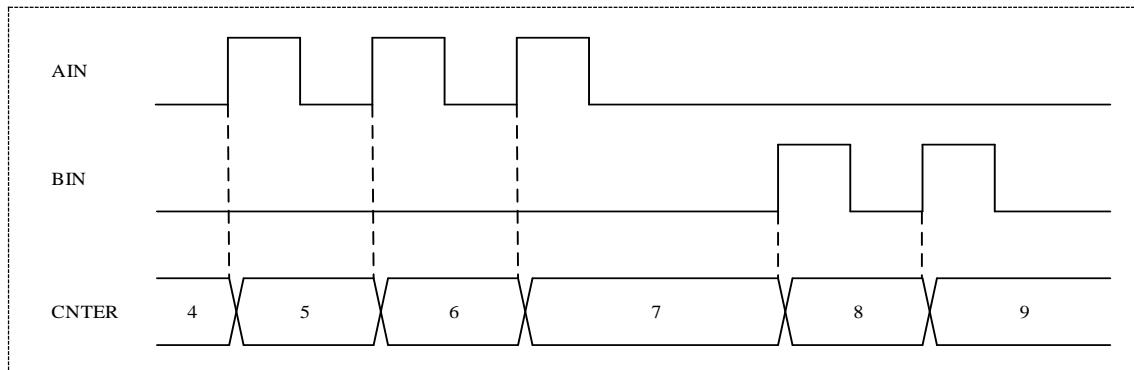


图 21-11 位置模式-基本计数

相位差计数

相位差计数是指根据 AIN 和 BIN 的相位关系进行计数。根据设定的不同，可以实现 1 倍计数、2 倍计数、4 倍计数等，如下图 21-12~图 21-14 所示。

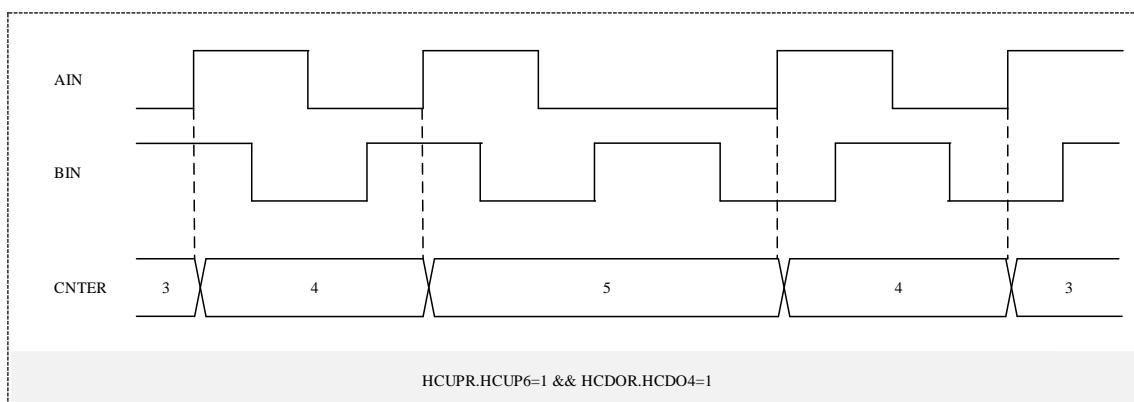


图 21-12 位置计数模式-相位差计数（1 倍计数）

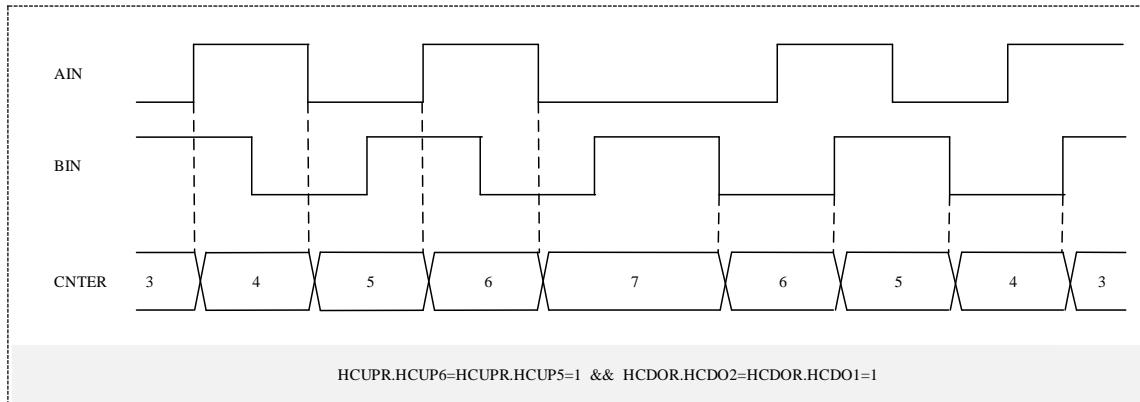


图 21-13 位置计数模式-相位差计数（2 倍计数）

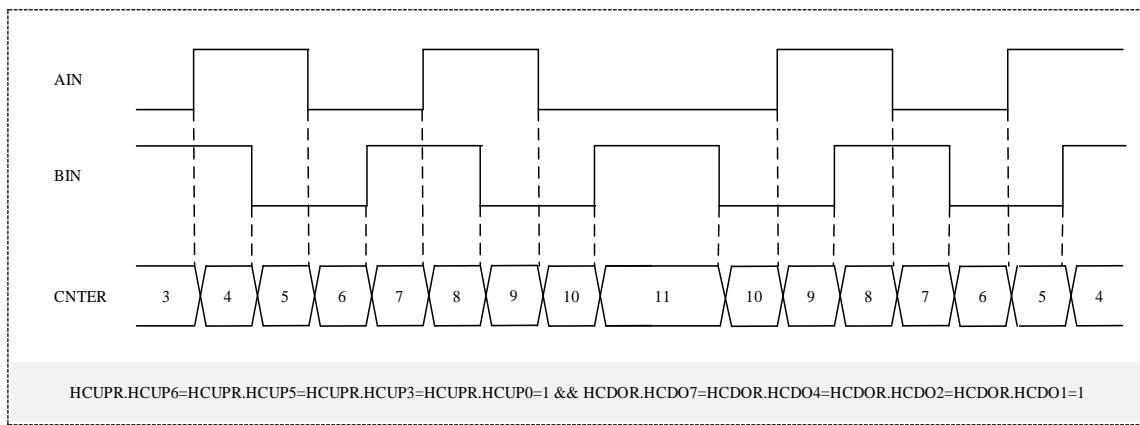


图 21-14 位置计数模式-相位差计数（4 倍计数）

方向计数

方向计数是指将 AIN 的输入状态设定为方向控制，将 BIN 的输入作为时钟计数，如下图 21-15 所示。

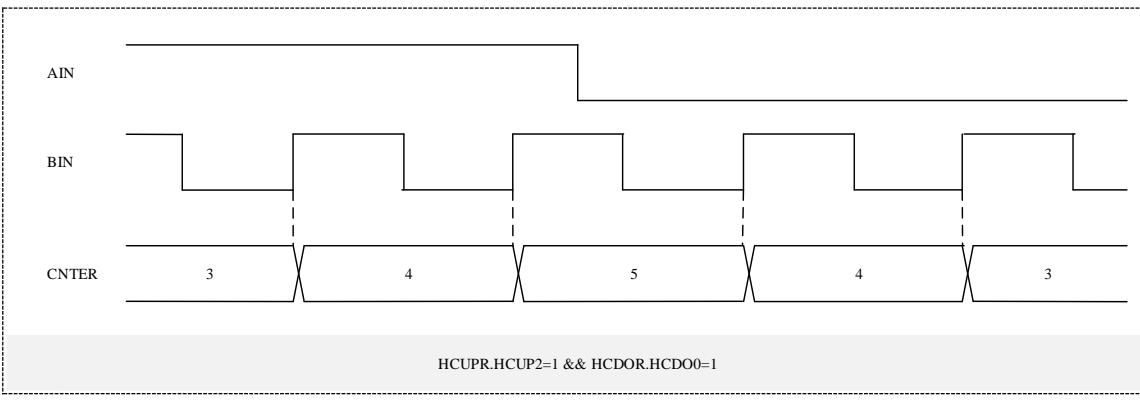


图 21-15 位置计数模式-方向计数

21.3.8.2 公转计数模式

正交编码公转计数模式，是指在 AIN、BIN 计数的基础上，加入 ZIN 的输入事件以实现对公转圈数等的判断。公转计数模式时根据公转定时器的计数方式，可实现 Z 相计

数功能、位置溢出计数功能和混合计数功能。

Z 相计数

Z 相计数是指根据 ZIN 的输入，公转计数单元进行计数，同时将位置计数单元清零的计数动作。如下所示。

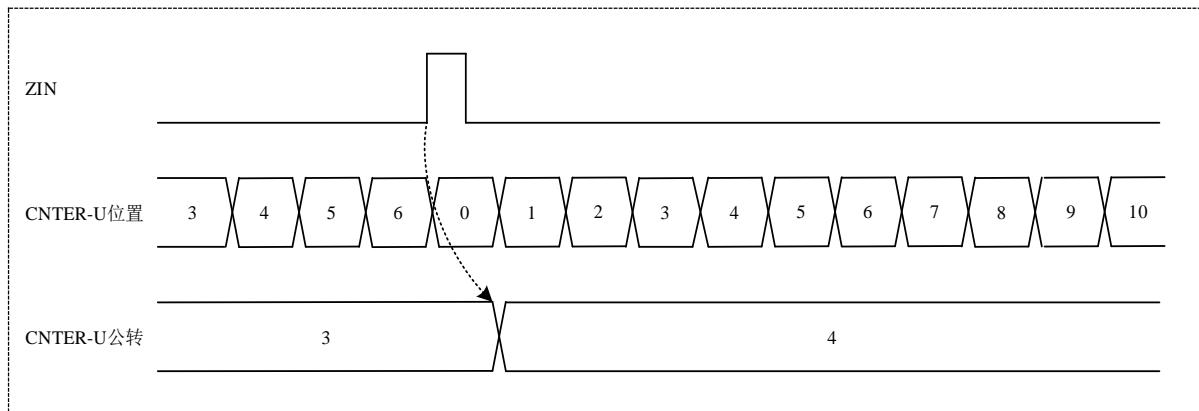


图 21-16 公转计数模式-Z 相计数

位置溢出计数

位置溢出计数是指位置计数单元计数发生上溢或下溢时，产生一个溢出事件，从而触发公转计数单元的定时器进行一次计数（在该计数方式时 ZIN 的输入不进行公转计数单元的计数动作和位置计数单元的清零动作）。

公转计数单元的硬件递加（递减）事件选择寄存器（HCUPR 或 HCDOR）的递加（递减）事件 bit12~11 位使能，位置计数单元的溢出事件就可以触发公转计数单元实现一次计数。如下图 21-17 所示。

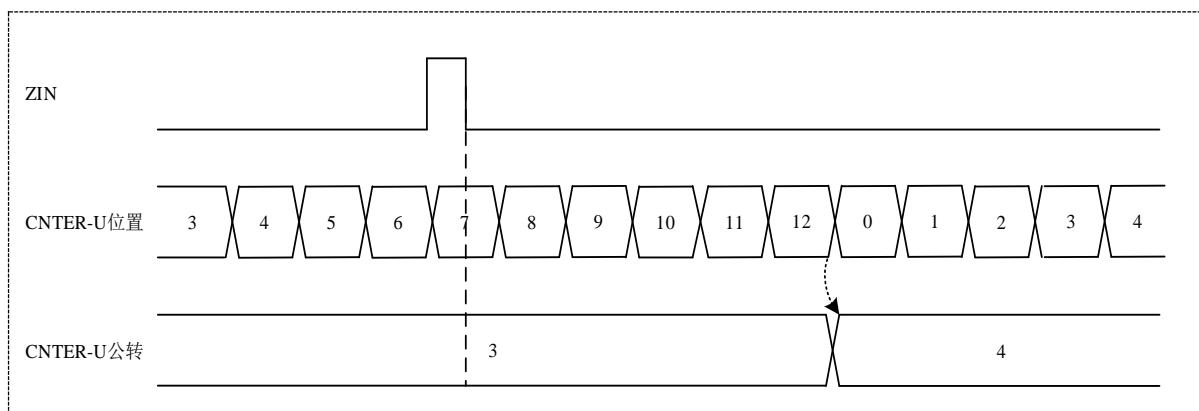


图 21-17 公转计数模式-位置溢出计数

混合计数

混合计数是指上述 Z 相计数和位置溢出计数两种计数方式合并起来的计数动作，其实现方式也是上述两种计数方式的组合。如下图 21-18 所示。

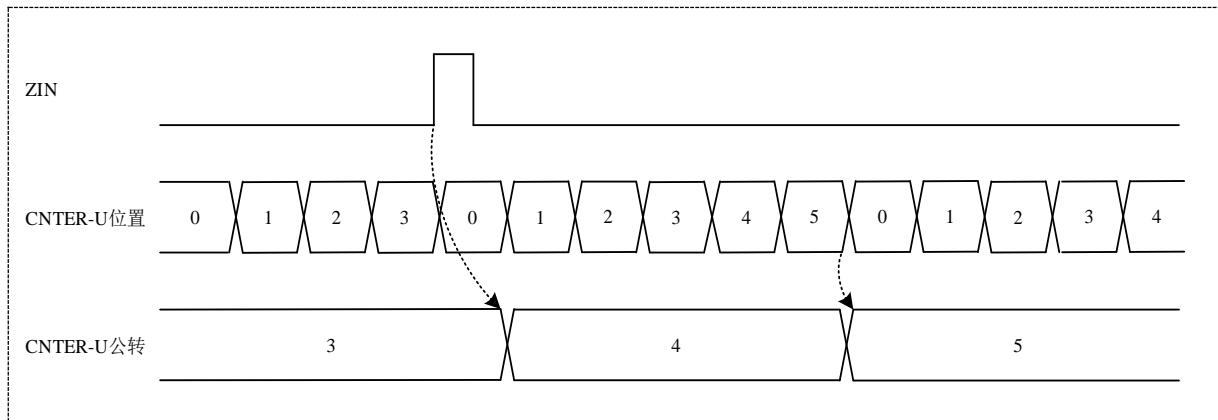


图 21-18 公转计数模式-混合计数

21.4 中断及事件说明

TimerA 含有 3 个中断输出和 3 个事件输出，分别是 1 个比较匹配中断和事件、2 个周期匹配中断和事件。

21.4.1 比较匹配中断及事件

比较基准值寄存器(CMPARn)与计数值比较发生比较匹配时，状态标志寄存器(STFLR)中的对应位(STFLR.CMPFn)会被置为 1。此时，若中断控制寄存器(ICONR)的对应位(ICONR.ITENn)设定为 1，则对应的中断请求(TMRA_U<t>_CMP)会被触发；若事件控制寄存器(ECONR)的对应位(ECONR.ETENn)设定为 1，则对应的事件请求(TMRA_U<t>_CMP)会被触发(n=1~8)。

捕获控制寄存器(CCONRn)选择的捕获输入有效条件产生时，捕获输入动作发生。此时，若中断控制寄存器(ICONR)的对应位(ICONR.ITENn)设定为 1，则对应的中断请求(TMRA_U<t>_CMP)被触发；若事件控制寄存器(ECONR)的对应位(ECONR.ETENn)设定为 1，则对应的事件请求(TMRA_U<t>_CMP)会被触发(n=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 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。

21.4.2 周期匹配中断及事件

锯齿波模式递加计数至上溢点、锯齿波模式递减计数至下溢点、三角波模式计数至谷点或峰点，控制状态寄存器(BCSTR)的 OVFF 或 UDFF 位会被置为 1。此时，若 BCSTR.ITENOVF 或 BCSTR.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_U<t>_OVF 和 TMRA_U<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_U<t>_OVF 和 TMRA_U<t>_UDF)输出用于选择触发别的模块。

21.5 寄存器说明

表 21-3 所示，为 TimerA 模块的寄存器列表。

BASE ADDR:

0x4001_5000 (U1)、0x4001_5400 (U2)、0x4001_5800 (U3)、

0x4001_5C00 (U4)、0x4001_6000 (U5)、0x4001_6400 (U6)

寄存器名	符号	偏移量	位宽	复位值
通用计数值寄存器	TMRA_CNTER	0x0000	16	0x0000
周期基准值寄存器	TMRA_PERAR	0x0004	16	0xFFFF
比较基准值寄存器1	TMRA_CMPAR1	0x0040	16	0xFFFF
比较基准值寄存器2	TMRA_CMPAR2	0x0044	16	0xFFFF
比较基准值寄存器3	TMRA_CMPAR3	0x0048	16	0xFFFF
比较基准值寄存器4	TMRA_CMPAR4	0x004C	16	0xFFFF
比较基准值寄存器5	TMRA_CMPAR5	0x0050	16	0xFFFF
比较基准值寄存器6	TMRA_CMPAR6	0x0054	16	0xFFFF
比较基准值寄存器7	TMRA_CMPAR7	0x0058	16	0xFFFF
比较基准值寄存器8	TMRA_CMPAR8	0x005C	16	0xFFFF
控制状态寄存器	TMRA_BCSTR	0x0080	16	0x0002
中断控制寄存器	TMRA_ICONR	0x0090	16	0x0000
事件控制寄存器	TMRA_ECONR	0x0094	16	0x0000
滤波控制寄存器	TMRA_FCONR	0x0098	16	0x0000
状态标志寄存器	TMRA_STFLR	0x009C	16	0x0000
缓存控制寄存器1	TMRA_BCONR1	0x00C0	16	0x0000
缓存控制寄存器2	TMRA_BCONR2	0x00C8	16	0x0000
缓存控制寄存器3	TMRA_BCONR3	0x00D0	16	0x0000
缓存控制寄存器4	TMRA_BCONR4	0x00D8	16	0x0000
捕获控制寄存器1	TMRA_CCONR1	0x0100	16	0x0000
捕获控制寄存器2	TMRA_CCONR2	0x0104	16	0x0000

捕获控制寄存器3	TMRA_CCONR3	0x0108	16	0x0000
捕获控制寄存器4	TMRA_CCONR4	0x010C	16	0x0000
捕获控制寄存器5	TMRA_CCONR5	0x0110	16	0x0000
捕获控制寄存器6	TMRA_CCONR6	0x0114	16	0x0000
捕获控制寄存器7	TMRA_CCONR7	0x0118	16	0x0000
捕获控制寄存器8	TMRA_CCONR8	0x011C	16	0x0000
端口控制寄存器1	TMRA_PCONR1	0x0140	16	0x0000
端口控制寄存器2	TMRA_PCONR2	0x0144	16	0x0000
端口控制寄存器3	TMRA_PCONR3	0x0148	16	0x0000
端口控制寄存器4	TMRA_PCONR4	0x014C	16	0x0000
端口控制寄存器5	TMRA_PCONR5	0x0150	16	0x0000
端口控制寄存器6	TMRA_PCONR6	0x0154	16	0x0000
端口控制寄存器7	TMRA_PCONR7	0x0158	16	0x0000
端口控制寄存器8	TMRA_PCONR8	0x015C	16	0x0000
硬件触发事件选择寄存器	TMRA_HCONR	0x0084	16	0x0000
硬件递加事件选择寄存器	TMRA_HCUPR	0x0088	16	0x0000
硬件递减事件选择寄存器	TMRA_HCDOR	0x008C	16	0x0000

表 21-3 寄存器列表

21.5.1 通用计数值寄存器（TMRA_CNTER）

复位值: 0x0000

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

位	标记	位名	功能	读写
b15~b0	CNT[15:0]	计数值	当前定时器的计数值	R/W

21.5.2 周期基准值寄存器（TMRA_PERAR）

复位值: 0xFFFF

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

位	标记	位名	功能	读写
b15~b0	PER[15:0]	计数周期值	设定每轮计数的计数周期值	R/W

21.5.3 比较基准值寄存器（TMRA_CMPAR1~8）

复位值: 0xFFFF

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

位	标记	位名	功能	读写
b15~b0	CMP[15:0]	计数比较基准值	设定比较基准值	R/W

21.5.4 控制状态寄存器 (TMRA_BCSTR)

复位值: 0x0002

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UDF F	OVF F	ITEN UDF	ITEN OVF	-	-	-	-	CKDIV[3:0]	SYN ST	MODE	DIR	START			

位	标记	位名	功能	读写
b15	UDFF	下溢标志	0: 向下计数时, 未发生计数下溢 1: 向下计数时, 发生计数下溢	R/W
b14	OVFF	上溢标志	0: 向上计数时, 未发生计数上溢 1: 向上计数时, 发生计数上溢	R/W
b13	ITENUDF	下溢中断使能	0: 计数下溢中断不使能 1: 计数下溢中断使能	R/W
b12	ITENOVF	上溢中断使能	0: 计数上溢中断不使能 1: 计数上溢中断使能	R/W
b11~b8	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b7-b4	CKDIV[3:0]	计数时钟选择	0000: PCLK 0001: PCLK/2 0010: PCLK/4 0011: PCLK/8 0100: PCLK/16 0101: PCLK/32 0110: PCLK/64 0111: PCLK/128 1000: PCLK/256 1001: PCLK/512 1010: PCLK/1024	R/W
b3	SYNST	同步启动使能	0: 与单元1的同步启动功能无效 1: 与单元1的同步启动功能有效 注: 单元1的该位设定无效, 读出时为“0”	R/W
b2	MODE	计数模式	0: 锯齿波模式 1: 三角波模式	R/W
b1	DIR	计数方向	0: 计数器向下计数 1: 计数器向上计数	R/W
b0	START	定时器启动	0: 定时器关闭 1: 定时器启动 注1: 该位在硬件停止条件有效时, 会自动变为0 注2: 单元2~6的同步启动功能有效时, 相应单元的该位在单元1软 件启动后也会被置位	R/W

21.5.5 中断控制寄存器 (TMRA_ICONR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								IT EN8	IT EN7	IT EN6	IT EN5	IT EN4	IT EN3	IT EN2	IT EN1

位	标记	位名	功能	读写
b15~b8	Reserved	-	读出时为“0”，写入时写“0”	R/W
b7	ITEN8	计数匹配中断使能8	0: CMPAR8寄存器与计数值相等时，或者发生捕获输入事件时， 该中断无效 1: CMPAR8寄存器与计数值相等时，或者发生捕获输入事件时， 该中断使能	R/W
b6	ITEN7	计数匹配中断使能7	0: CMPAR7寄存器与计数值相等时，或者发生捕获输入事件时， 该中断无效 1: CMPAR7寄存器与计数值相等时，或者发生捕获输入事件时， 该中断使能	R/W
b5	ITEN6	计数匹配中断使能6	0: CMPAR6寄存器与计数值相等时，或者发生捕获输入事件时， 该中断无效 1: CMPAR6寄存器与计数值相等时，或者发生捕获输入事件时， 该中断使能	R/W
b4	ITEN5	计数匹配中断使能5	0: CMPAR5寄存器与计数值相等时，或者发生捕获输入事件时， 该中断无效 1: CMPAR5寄存器与计数值相等时，或者发生捕获输入事件时， 该中断使能	R/W
b3	ITEN4	计数匹配中断使能4	0: CMPAR4寄存器与计数值相等时，或者发生捕获输入事件时， 该中断无效 1: CMPAR4寄存器与计数值相等时，或者发生捕获输入事件时， 该中断使能	R/W
b2	ITEN3	计数匹配中断使能3	0: CMPAR3寄存器与计数值相等时，或者发生捕获输入事件时， 该中断无效 1: CMPAR3寄存器与计数值相等时，或者发生捕获输入事件时， 该中断使能	R/W
b1	ITEN2	计数匹配中断使能2	0: CMPAR2寄存器与计数值相等时，或者发生捕获输入事件时， 该中断无效 1: CMPAR2寄存器与计数值相等时，或者发生捕获输入事件时， 该中断使能	R/W
b0	ITEN1	计数匹配中断使能1	0: CMPAR1寄存器与计数值相等时，或者发生捕获输入事件时， 该中断无效 1: CMPAR1寄存器与计数值相等时，或者发生捕获输入事件时， 该中断使能	R/W

21.5.6 事件控制寄存器 (TMRA_ECONR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								ET EN8	ET EN7	ET EN6	ET EN5	ET EN4	ET EN3	ET EN2	ET EN1

位	标记	位名	功能	读写
b15~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	ETEN8	计数匹配事件使能8	0: CMPAR8寄存器与计数值相等时, 或者发生捕获输入事件时, 该事件输出无效 1: CMPAR8寄存器与计数值相等时, 或者发生捕获输入事件, 该事件输出使能	R/W
b6	ETEN7	计数匹配事件使能7	0: CMPAR7寄存器与计数值相等时, 或者发生捕获输入事件, 该事件输出无效 1: CMPAR7寄存器与计数值相等时, 或者发生捕获输入事件, 该事件输出使能	R/W
b5	ETEN6	计数匹配事件使能6	0: CMPAR6寄存器与计数值相等时, 或者发生捕获输入事件, 该事件输出无效 1: CMPAR6寄存器与计数值相等时, 或者发生捕获输入事件, 该事件输出使能	R/W
b4	ETEN5	计数匹配事件使能5	0: CMPAR5寄存器与计数值相等时, 或者发生捕获输入事件, 该事件输出无效 1: CMPAR5寄存器与计数值相等时, 或者发生捕获输入事件, 该事件输出使能	R/W
b3	ETEN4	计数匹配事件使能4	0: CMPAR4寄存器与计数值相等时, 或者发生捕获输入事件, 该事件输出无效 1: CMPAR4寄存器与计数值相等时, 或者发生捕获输入事件, 该事件输出使能	R/W
b2	ETEN3	计数匹配事件使能3	0: CMPAR3寄存器与计数值相等时, 或者发生捕获输入事件, 该事件输出无效 1: CMPAR3寄存器与计数值相等时, 或者发生捕获输入事件, 该事件输出使能	R/W
b1	ETEN2	计数匹配事件使能2	0: CMPAR2寄存器与计数值相等时, 或者发生捕获输入事件, 该事件输出无效 1: CMPAR2寄存器与计数值相等时, 或者发生捕获输入事件, 该事件输出使能	R/W
b0	ETEN1	计数匹配事件使能1	0: CMPAR1寄存器与计数值相等时, 或者发生捕获输入事件, 该事件输出无效 1: CMPAR1寄存器与计数值相等时, 或者发生捕获输入事件, 该事件输出使能	R/W

21.5.7 滤波控制寄存器 (TMRA_FCONR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	NOFI CKCB[1:0]	NOFI ENCB	-	NOFI CKCA[1:0]	NOFI ENCA	-	-	-	-	-	-	NOFI CKTG[1:0]	NOFI ENTG		

位	标记	位名	功能	读写
b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14~b13	NOFICKCB[1:0]	滤波采样基准时 钟选择CB	00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64	R/W
b12	NOFIENCB	捕获输入端口滤 波CB	0: TIMA_<*>_CLKB端口输入滤波功能无效 1: TIMA_<*>_CLKB端口输入滤波功能使能	R/W
b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10~b9	NOFICKCA[1:0]	滤波采样基准时 钟选择CA	00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64	R/W
b8	NOFIENCA	捕获输入端口滤 波CA	0: TIMA_<*>_CLKA端口输入滤波功能无效 1: TIMA_<*>_CLKA端口输入滤波功能使能	R/W
b7~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~b1	NOFICKTG[1:0]	滤波采样基准时 钟选择TG	00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64	R/W
b0	NOFIENTG	捕获输入端口滤 波TG	0: TIMA_<*>_TRIG输入端口滤波功能无效 1: TIMA_<*>_TRIG输入端口滤波功能使能	R/W

21.5.8 状态标志寄存器 (TMRA_STFLR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								CMP F8	CMP F7	CMP F6	CMP F5	CMP F4	CMP F3	CMP F2	CMP F1

位	标记	位名	功能	读写
b15-b8	Reserved	-	读出时为“0”，写入时写“0”	R
b7	CMPF8	计数匹配标志8	0: CMPAR8寄存器的值与计数值不相等，且未发生TIMA_<t>_PWM8捕获完成动作 1: CMPAR8寄存器的值与计数值相等，或发生TIMA_<t>_PWM8捕获完成动作	R/W
b6	CMPF7	计数匹配标志7	0: CMPAR7寄存器的值与计数值不相等，且未发生TIMA_<t>_PWM7捕获完成动作 1: CMPAR7寄存器的值与计数值相等，或发生TIMA_<t>_PWM7捕获完成动作	R/W
b5	CMPF6	计数匹配标志6	0: CMPAR6寄存器的值与计数值不相等，且未发生TIMA_<t>_PWM6捕获完成动作 1: CMPAR6寄存器的值与计数值相等，或发生TIMA_<t>_PWM6捕获完成动作	R/W
b4	CMPF5	计数匹配标志5	0: CMPAR5寄存器的值与计数值不相等，且未发生TIMA_<t>_PWM5捕获完成动作 1: CMPAR5寄存器的值与计数值相等，或发生TIMA_<t>_PWM5捕获完成动作	R/W
b3	CMPF4	计数匹配标志4	0: CMPAR4寄存器的值与计数值不相等，且未发生TIMA_<t>_PWM4捕获完成动作 1: CMPAR4寄存器的值与计数值相等，或发生TIMA_<t>_PWM4捕获完成动作	R/W
b2	CMPF3	计数匹配标志3	0: CMPAR3寄存器的值与计数值不相等，且未发生TIMA_<t>_PWM3捕获完成动作 1: CMPAR3寄存器的值与计数值相等，或发生TIMA_<t>_PWM3捕获完成动作	R/W
b1	CMPF2	计数匹配标志2	0: CMPAR2寄存器的值与计数值不相等，且未发生TIMA_<t>_PWM2捕获完成动作 1: CMPAR2寄存器的值与计数值相等，或发生TIMA_<t>_PWM2捕获完成动作	R/W
b0	CMPF1	计数匹配标志1	0: CMPAR1寄存器的值与计数值不相等，且未发生TIMA_<t>_PWM1捕获完成动作 1: CMPAR1寄存器的值与计数值相等，或发生TIMA_<t>_PWM1捕获完成动作	R/W

成动作

21.5.9 缓存控制寄存器 (TMRA_BCONR1~4)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Reserved														BSE1	BSE0	BEN

位	标记	位名	功能	读写
b15~b3	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b2	BSE1	三角波缓存传送 选择1	0: 三角波计数模式计数到谷点时, 缓存值不传送 1: 三角波计数模式计数到谷点时, 缓存值传送, 即: CMMARm -> CMPARn (m=2、4、6、8, n=1、3、5、7)	R/W
b1	BSE0	三角波缓存传送 选择0	0: 三角波计数模式计数到峰点时, 缓存值不传送 1: 三角波计数模式计数到峰点时, 缓存值传送, 即: CMMARm -> CMPARn (m=2、4、6、8, n=1、3、5、7)	R/W
b0	BEN	缓存使能	0: CMPARn基准值的缓存功能无效 1: CMPARn基准值的缓存功能有效 (n=1、3、5、7)	R/W

21.5.10 捕获控制寄存器 (TMRA_CCONR1~8)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	NOFI CKCP[1:0]	NOFI ENCP	-	-	HICP 4	HICP 3	-	HICP 2	HICP 1	HICP 0	-	-	-	CAP MD	

位	标记	位名	功能	读写
b15	Reserved	-	读出时为“0”，写入时写“0”	R/W
b14~b13	NOFICKCP[1:0]	滤波采样基准时钟 选择CP	00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64	R/W
b12	NOFIENCP	捕获输入端口滤波 CP	0: TIMA_<1>_PWMMn端口输入滤波功能无效 1: TIMA_<1>_PWMMn端口输入滤波功能使能 (n=1~8)	R/W
b11~b10	Reserved	-	读出时为“0”，写入时写“0”	R/W
b9	HICP4	捕获输入条件使能4	0: TIMA_<1>_TRIG端口输入采样到下降沿时，通道m+1不发生捕获输入动作 1: TIMA_<1>_TRIG端口输入采样到下降沿时，通道m+1产生捕获输入动作 注：该位只有CCONR3寄存器有效。即，该位有效后且对应事件发生时，在CCONR4.CAPMD=1的条件下、当前计数器值被捕获保存到CMPAR4，且STFLR.CMPF4置位	R/W
b8	HICP3	捕获输入条件使能3	0: TIMA_<1>_TRIG端口输入采样到上升沿时，通道m+1不发生捕获输入动作 1: TIMA_<1>_TRIG端口输入采样到上升沿时，通道m+1产生捕获输入动作 注：该位只有CCONR3寄存器有效。即，该位有效后且对应事件发生时，在CCONR4.CAPMD=1的条件下、当前计数器值被捕获保存到CMPAR4，且STFLR.CMPF4置位	R/W
b7	Reserved	-	读出时为“0”，写入时写“0”	R/W
b6	HICP2	捕获输入条件使能2	0: TMRA_HTSSR1寄存器中指定的事件发生时，不发生捕获输入动作 1: TMRA_HTSSR1寄存器中指定的事件发生时，产生捕获输入动作	R/W
b5	HICP1	捕获输入条件使能1	0: TIMA_<1>_PWMMn端口输入采样到下降沿时，不发生捕获输入动作 1: TIMA_<1>_PWMMn端口输入采样到下降沿时，产生捕获输入动作 (n=1~8)	R/W
b4	HICP0	捕获输入条件使能0	0: TIMA_<1>_PWMMn端口输入采样到上升沿时，不发生捕获输入动作 1: TIMA_<1>_PWMMn端口输入采样到上升沿时，产生捕获输入动作 (n=1~8)	R/W
b3~b1	Reserved	-	读出时为“0”，写入时写“0”	R/W

b0	CAPMD	功能模式选择	0: 比较输出功能 1: 捕获输入功能	R/W
----	-------	--------	------------------------	-----

21.5.11 端口控制寄存器 (TMRA_PCONR1~8)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	OUTEN	-	-	FORC[1:0]	PERC[1:0]	PERC[1:0]	CMPC[1:0]	CMPC[1:0]	STPC[1:0]	STPC[1:0]	STAC[1:0]	STAC[1:0]	

位	标记	位名	功能	读写
b15~b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	OUTEN	输出使能	0: PWM输出功能时的TIMA_<1>_PWMn端口输出无效 1: PWM输出功能时的TIMA_<1>_PWMn端口输出有效 (n=1~8)	R/W
b11~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9~b8	FORC[1:0]	强制端口状态设 定	0x: 设定无效定 10: 下周期开始, TIMA_<1>_PWMn端口输出设定为低电平 11: 下周期开始, TIMA_<1>_PWMn端口输出设定为高电平 注1: 下周期是指硬件计数模式或锯齿波计数到上溢点或下溢点、 三角波计数到谷点 注2: 该寄存器位可用于实现PWM输出占空比0%或100%的控制	R/W
b7~b6	PERC[1:0]	周期值匹配时端 口状态设定	00: 计数值与PERAR相等时, TIMA_<1>_PWMn端口输出设定为低电平 01: 计数值与PERAR相等时, TIMA_<1>_PWMn端口输出设定为高电平 10: 计数值与PERAR相等时, TIMA_<1>_PWMn端口输出保持先前状态 11: 计数值与PERAR相等时, TIMA_<1>_PWMn端口输出设定为反转电平 (n=1~8)	R/W
b5~b4	CMPC[1:0]	比较值匹配时端 口状态设定	00: 计数值与CMPARN相等时, TIMA_<1>_PWMn端口输出设定为低电平 01: 计数值与CMPARN相等时, TIMA_<1>_PWMn端口输出设定为高电平 10: 计数值与CMPARN相等时, TIMA_<1>_PWMn端口输出保持先前状态 11: 计数值与CMPARN相等时, TIMA_<1>_PWMn端口输出设定为反转电平 (n=1~8)	R/W
b3~b2	STPC[1:0]	计数停止时端口 状态设定	00: 计数停止时, TIMA_<1>_PWMn端口输出设定为低电平 01: 计数停止时, TIMA_<1>_PWMn端口输出设定为高电平 10: 计数停止时, TIMA_<1>_PWMn端口输出保持先前状态 11: 计数停止时, TIMA_<1>_PWMn端口输出保持先前状态 (n=1~8)	R/W

b1~b0	STAC[1:0]	计数开始时端口状态设定	00: 计数开始时, TIMA_< \ominus >_PWMn端口输出设定为低电平 01: 计数开始时, TIMA_< \ominus >_PWMn端口输出设定为高电平 10: 计数开始时, TIMA_< \ominus >_PWMn端口输出保持先前状态 11: 计数开始时, TIMA_< \ominus >_PWMn端口输出保持先前状态 (n=1~8)	R/W
注: 该位设定只在不分频 (BCSTR.CKDIV=4'h0) 的情况下有效, 其它分频请设定为2'b10或2'b11				

21.5.12 硬件触发事件选择寄存器 (TMRA_HCONR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HCLE 6	HCLE 5	HCLE 4	HCLE 3	-	HCLE 2	HCLE 1	HCLE 0	-	HSTP 2	HSTP 1	HSTP 0	-	HSTA 2	HSTA 1	HSTA 0

位	标记	位名	功能	读写
b15	HCLE6	硬件清零条件6	条件: TIMA_<=>_PWM3端口输入采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b14	HCLE5	硬件清零条件5	条件: TIMA_<=>_PWM3端口输入采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b13	HCLE4	硬件清零条件4	条件: 本单元为单元m时, 单元n的TRIG端口输入采样到下降沿 (当m=1、3、5时, n=2、4、6; 当m=2、4、6时, n=1、3、5) 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b12	HCLE3	硬件清零条件3	条件: 本单元为单元m时, 单元n的TRIG端口输入采样到上升沿 (当m=1、3、5时, n=2、4、6; 当m=2、4、6时, n=1、3、5) 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b11	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b10	HCLE2	硬件清零条件2	条件: TMRA_HTSSR0寄存器中指定的事件发生 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b9	HCLE1	硬件清零条件1	条件: TIMA_TRIG端口输入采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b8	HCLE0	硬件清零条件0	条件: TIMA_TRIG端口输入采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	R/W
b7	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b6	HSTP2	硬件停止条件2	条件: TMRA_HTSSR0寄存器中指定的事件发生 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b5	HSTP1	硬件停止条件1	条件: TIMA_<=>_TRIG端口输入采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	R/W
b4	HSTP0	硬件停止条件0	条件: TIMA_<=>_TRIG端口输入采样到上升沿 0: 条件匹配时, 硬件停止无效	R/W

			1: 条件匹配时，硬件停止有效	
b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2	HSTA2	硬件启动条件2	条件：TMRA_HTSSR0寄存器中指定的事件发生 0: 条件匹配时，硬件启动无效 1: 条件匹配时，硬件启动有效	R/W
b1	HSTA1	硬件启动条件1	条件： 1) 本单元TIMA_<t>_TRIG端口输入采样到下降沿（同步启动功能无效） 2) TIMA_1_TRIG端口输入采样到下降沿（同步启动功能有效） 0: 条件匹配时，硬件启动无效 1: 条件匹配时，硬件启动有效 注：条件2) 只有单元2~6可选择，单元1无效	R/W
b0	HSTA0	硬件启动条件0	条件： 1) 本单元TIMA_<t>_TRIG端口输入采样到上升沿（同步启动功能无效） 2) TIMA_1_TRIG端口输入采样到上升沿（同步启动功能有效） 0: 条件匹配时，硬件启动无效 1: 条件匹配时，硬件启动有效 注：条件2) 只有单元2~6可选择，单元1无效	R/W

21.5.13 硬件递加事件选择寄存器 (TMRA_HCUPR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	HC UP12	HC UP11	HC UP10	HC UP9	HC UP8	HC UP7	HC UP6	HC UP5	HC UP4	HC UP3	HC UP2	HC UP1	HC UP0

位	标记	位名	功能	读写
b15~b13	Reserved	-	读出时为“0”，写入时写“0”	R/W
b12	HCUP12	硬件递加条件12	条件：本单元为单元m时，单元n发生计数下溢（当m=1、3、5时，n=2、4、6；当m=2、4、6时，n=1、3、5） 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b11	HCUP11	硬件递加条件11	条件：本单元为单元m时，单元n发生计数上溢（当m=1、3、5时，n=2、4、6；当m=2、4、6时，n=1、3、5） 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b10	HCUP10	硬件递加条件10	条件：TMRA_HTSSR0寄存器中指定的事件发生 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b9	HCUP9	硬件递加条件9	条件：TIMA_<1>_TRIG端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b8	HCUP8	硬件递加条件8	条件：TIMA_<1>_TRIG端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b7	HCUP7	硬件递加条件7	条件：TIMA_<1>_CLKB端口为高电平时，TIMA_<1>_CLKA端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b6	HCUP6	硬件递加条件6	条件：TIMA_<1>_CLKB端口为高电平时，TIMA_<1>_CLKA端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b5	HCUP5	硬件递加条件5	条件：TIMA_<1>_CLKB端口为低电平时，TIMA_<1>_CLKA端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b4	HCUP4	硬件递加条件4	条件：TIMA_<1>_CLKB端口为低电平时，TIMA_<1>_CLKA端口上采样到上升沿 0：条件匹配时，硬件递加无效	R/W

			1: 条件匹配时, 硬件递加有效	
b3	HCUP3	硬件递加条件3	条件: TIMA_<1>_CLKA端口为高电平时, TIMA_<1>_CLKB端口上采样 到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b2	HCUP2	硬件递加条件2	条件: TIMA_<1>_CLKA端口为高电平时, TIMA_<1>_CLKB端口上采样 到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b1	HCUP1	硬件递加条件1	条件: TIMA_<1>_CLKA端口为低电平时, TIMA_<1>_CLKB端口上采样 到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W
b0	HCUP0	硬件递加条件0	条件: TIMA_<1>_CLKA端口为低电平时, TIMA_<1>_CLKB端口上采样 到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	R/W

21.5.14 硬件递减事件选择寄存器 (TMRA_HCDOR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	HC DO12	HC DO11	HC DO10	HC DO9	HC DO8	HC DO7	HC DO6	HC DO5	HC DO4	HC DO3	HC DO2	HC DO1	HC DO0

位	标记	位名	功能	读写
b15~b13	Reserved	-	读出时为“0”，写入时写“0”	R/W
b12	HCDO12	硬件递减条件12	条件：本单元为单元m时，单元n发生计数下溢（当m=1、3、5时，n=2、4、6；当m=2、4、6时，n=1、3、5） 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b11	HCDO11	硬件递减条件11	条件：本单元为单元m时，单元n发生计数上溢（当m=1、3、5时，n=2、4、6；当m=2、4、6时，n=1、3、5） 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b10	HCDO10	硬件递减条件10	条件：TMRA_HTSSR0寄存器中指定的事件发生 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b9	HCDO9	硬件递减条件9	条件：TIMA_<1>_TRIG端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b8	HCDO8	硬件递减条件8	条件：TIMA_<1>_TRIG端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b7	HCDO7	硬件递减条件7	条件：TIMA_<1>_CLKB端口为高电平时，TIMA_<1>_CLKA端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b6	HCDO6	硬件递减条件6	条件：TIMA_<1>_CLKB端口为高电平时，TIMA_<1>_CLKA端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b5	HCDO5	硬件递减条件5	条件：TIMA_<1>_CLKB端口为低电平时，TIMA_<1>_CLKA端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b4	HCDO4	硬件递减条件4	条件：TIMA_<1>_CLKB端口为低电平时，TIMA_<1>_CLKA端口上采样到上升沿 0：条件匹配时，硬件递减无效	R/W

			1: 条件匹配时, 硬件递减有效	
b3	HCDO3	硬件递减条件3	条件: TIMA_< t >_CLKA端口为高电平时, TIMA_< t >_CLKB端口上采样 到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b2	HCDO2	硬件递减条件2	条件: TIMA_< t >_CLKA端口为高电平时, TIMA_< t >_CLKB端口上采样 到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b1	HCDO1	硬件递减条件1	条件: TIMA_< t >_CLKA端口为低电平时, TIMA_< t >_CLKB端口上采样 到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W
b0	HCDO0	硬件递减条件0	条件: TIMA_< t >_CLKA端口为低电平时, TIMA_< t >_CLKB端口上采样 到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	R/W

22 通用定时器 (Timer0)

22.1 简介

通用定时器 0(Timer0)是一个可以实现同步计数、异步计数两种方式的基本定时器。

该定时器内含 2 个通道 (CH-A 和 CH-B)，可以在计数期间产生比较匹配事件。该事件可以触发中断，也可作为事件输出来控制其它模块等。本系列产品中搭载 2 个单元的 Timer0。

22.2 基本框图

Timer0 的基本框图如图 22-1 所示。

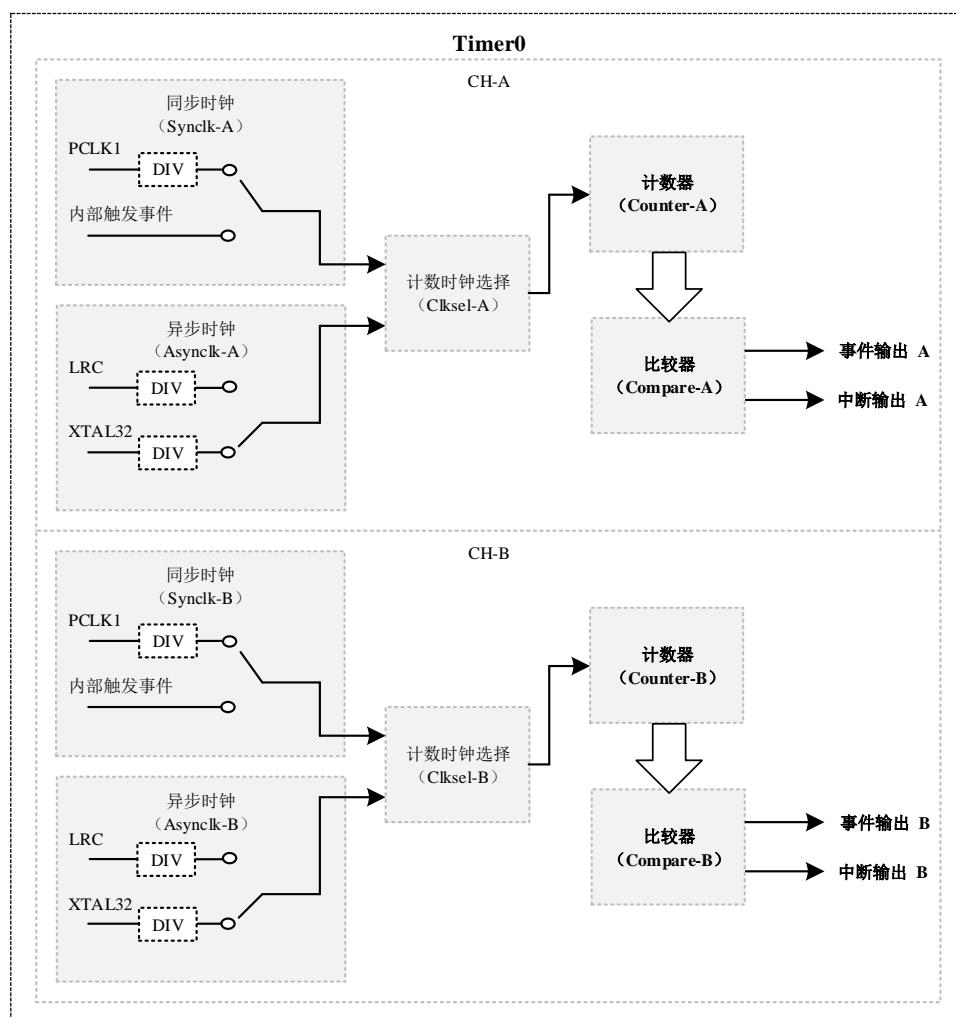


图 22-1 Timer0 基本框图

22.3 功能说明

22.3.1 时钟源选择

Timer0 的计数方式可以选择同步计数方式或异步计数方式。

同步计数方式是指定时器的计数时钟和总线访问时钟（寄存器读写操作时钟）有同步时序关系；异步计数方式是指定时器的计数时钟和总线访问时钟（寄存器读写操作时钟）是非同步的时序关系。异步计数方式对寄存器读操作时，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。

22.3.1.1 同步计数时钟源

同步计数方式时（BCONR.SYNNSA=0），时钟源可以有以下几种选择（BCONR.SYNCLKA设定）：

- a) PCLK1 及 PCLK1 的 2、4、8、16、32、64、128、256、512、1024 分频作为同步计数时钟（BCONR.SYNCLKA=0 & BCONR.CKDIVA [3:0] 设定）
- b) 内部硬件触发事件输入作为同步计数时钟（BCONR.SYNCLKA=1）

22.3.1.2 异步计数时钟源

异步计数方式时（BCONR.SYNNSA=1），时钟源可以有以下几种选择（BCONR.ASYNCLK A设定）：

- a) LRC 时钟源输入及其 2、4、8、16、32、64、128、256、512、1024 分频作为异步计数时钟（BCONR.ASYNCLK A=0 & BCONR.CKDIVA [3:0] 设定）
- b) XTAL32 时钟源输入及其 2、4、8、16、32、64、128、256、512、1024 分频作为异步计数时钟（BCONR.ASYNCLK A=1 & BCONR.CKDIVA [3:0] 设定）

22.3.2 基本计数动作

Timer0 的每个通道可设定基准计数值，在计数值和基准值相等时产生计数比较匹配事件。如图 22-2 所示。

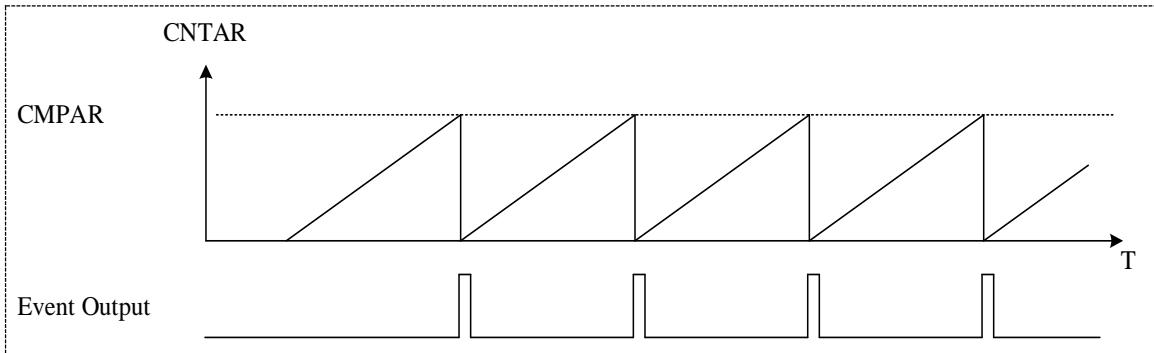


图 22-2 Timer0 计数时序图

22.3.3 硬件触发动作

Timer0 的 2 个通道有一个共用的内部硬件触发源,可以通过基本控制寄存器(BCONR)的相关设定来控制定时器的状态(计数、启动、停止、清零)以及捕获输入动作等。该硬件触发源的源选择通过向硬件触发选择寄存器(HTSSR)中输入对应的编号来实现,具体的事件对应关系请参考中断控制器(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)的外设电路触发功能位(AOS)使能。

22.4 中断及事件说明

22.4.1 中断输出

一个 Timer0 含有 2 个中断，分别是通道 A 和通道 B 的计数比较匹配中断或输入捕获中断。

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.CMAF、STFLR.CMBF 位分别会被置为 1。此时若设定基本控制寄存器(BCONR)的 BCONR.INTENA位使能中断，则对应的中断请求 (TMR0_Um_GCMn, m=1、2; n=A、B) 也会被触发。

在内部硬件触发输入作为捕获输入条件时，可以产生相应的捕获输入动作。此时若设定基本控制寄存器 (BCONR) 的 BCONR.INTENA位使能中断，则对应的中断请求 (TMR0_Um_GCMn, m=1、2; n=A、B) 被触发。

注意：

- 单元 1 的通道 A 的比较匹配中断 (TMR0_U1_GCMA) 仅在异步计数方式 (BCONR.SYNSA=1) 时可用。

22.4.2 事件输出

一个Timer0含有2个事件输出，分别是通道A和通道B的计数比较匹配事件或捕获输入事件。

在计数过程中发生计数比较匹配或捕获输入动作时，会分别产生相应的事件请求 (TMR0_Um_GCMn, m=1、2; n=A、B) 输出信号，可以用于选择触发其它模块。

22.5 寄存器说明

表 22-1 所示，为 Timer0 模块的寄存器列表。

BASE ADDR: 0x4002_4000 (U1)、0x4002_4400 (U2)

寄存器名	符号	偏移量	位宽	复位值
计数值寄存器	TMR0_CNTAR	0x0000	32	0x0000_0000
计数值寄存器	TMR0_CNTBR	0x0004	32	0x0000_0000
基准值寄存器	TMR0_CMPAR	0x0008	32	0x0000_FFFF
基准值寄存器	TMR0_CMPBR	0x000C	32	0x0000_FFFF
基本控制寄存器	TMR0_BCONR	0x0010	32	0x0000_0000
状态标志寄存器	TMR0_STFLR	0x0014	32	0x0000_0000

表 22-1 寄存器列表

22.5.1 计数值寄存器 (TMR0_CNTAR)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”	R
b15~b0	CNTA[15:0]	计数值	当前定时器的计数值	R/W

22.5.2 基准值寄存器 (TMR0_CMPAR)

复位值: 0x0000_FFFF

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”	R
b15~b0	CMPA[15:0]	基准值	设定计数基准值，产生Compare Match事件	R/W

22.5.3 基本控制寄存器 (TMR0_BCONR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
HICP B	HCLE	HSTP	HSTA	-	ASYN CLKB	SYN CLKB	SYN SB		CKDIV B[3:0]	-		INT ENB	CAP MDB	CST B	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HICP A	HCLE	HSTP	HSTA	-	ASYN CLKA	SYN CLKA	SYN SA		CKDIV A[3:0]	-		INT ENA	CAP MDA	CST A	

位	标记	位名	功能	读写
b31	HICPB	硬件触发输入捕获B	条件: 内部硬件触发事件有效 0: 条件匹配时, 捕获输入无效 1: 条件匹配时, 捕获输入有效	R/W
b30	HCLEB	硬件触发清零B	条件: 内部硬件触发事件有效 0: 条件匹配时, 定时器清零无效 1: 条件匹配时, 定时器清零有效	R/W
b29	HSTPB	硬件触发停止B	条件: 内部硬件触发事件有效 0: 条件匹配时, 定时器停止无效 1: 条件匹配时, 定时器停止有效	R/W
b28	HSTAB	硬件触发启动B	条件: 内部硬件触发事件有效 0: 条件匹配时, 定时器启动无效 1: 条件匹配时, 定时器启动有效	R/W
b27	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b26	ASYNCLKB	通道B异步计数时钟 源选择	0: LRC 1: XTAL32 注: 当使能UART对应通道的超时功能时, 时钟源不再是XTAL32, 而是UART波特率生成器产生的时钟tick	R/W
b25	SYNCLKB	通道B同步计数时钟 源选择	0: PCLK1 1: 内部硬件触发事件	R/W
b24	SYNSB	通道B计数方式选择	0: 同步计数方式 1: 异步计数方式	R/W
b23~b20	CKDIVB[3:0]	通道B计数时钟分频 选择	通道B计数时钟分频选择: 0000: 时钟源 0001: 时钟源/2 0010: 时钟源/4 0011: 时钟源/8 0100: 时钟源/16 0101: 时钟源/32 0110: 时钟源/64 0111: 时钟源/128	R/W

1000: 时钟源/256

1001: 时钟源/512

1010: 时钟源/1024

请不要设定其它值

注：被分频的时钟源可以是异步计数时的各种时钟源、同步计数

时的PCLK1

b19	Reserved	-	读出时为“0”,写入时写“0”	R/W
b18	INTENB	计数匹配中断使能B	0: CMPBR寄存器与计数值(CNTBR)相等时, 或者发生捕获输入事件时, 该中断无效 1: CMPBR寄存器与计数值(CNTBR)相等时, 或者发生捕获输入事件时, 该中断使能	R/W
b17	CAPMDB	功能模式选择B	0: 比较输出功能 1: 捕获输入功能	R/W
b16	CSTB	定时器启动	0: 通道B定时器关闭 1: 通道B定时器启动 注: 该位在硬件触发停止条件有效时, 会自动变为0	R/W
b15	HICPA	硬件触发输入捕获A	条件: 内部硬件触发事件有效 0: 条件匹配时, 捕获输入无效 1: 条件匹配时, 捕获输入有效	R/W
b14	HCLEA	硬件触发清零A	条件: 内部硬件触发事件有效 0: 条件匹配时, 定时器清零无效 1: 条件匹配时, 定时器清零有效	R/W
b13	HSTPA	硬件触发停止A	条件: 内部硬件触发事件有效 0: 条件匹配时, 定时器停止无效 1: 条件匹配时, 定时器停止有效	R/W
b12	HSTAA	硬件触发启动A	条件: 内部硬件触发事件有效 0: 条件匹配时, 定时器启动无效 1: 条件匹配时, 定时器启动有效	R/W
b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10	ASYNCLKA	通道A异步计数时钟源选择	0: LRC 1: XTAL32	R/W
b9	SYNCLKA	通道A同步计数时钟源选择	0: PCLK1 1: 内部硬件触发事件	R/W
b8	SYNSA	通道A计数方式选择	0: 同步计数方式 1: 异步计数方式	R/W
b7~b4	CKDIVA[3:0]	通道A计数时钟分频选择	0000: 时钟源 0001: 时钟源/2 0010: 时钟源/4 0011: 时钟源/8 0100: 时钟源/16 0101: 时钟源/32	R/W

0110: 时钟源/64

0111: 时钟源/128

1000: 时钟源/256

1001: 时钟源/512

1010: 时钟源/1024

请不要设定其它值

注：被分频的时钟源可以是异步计数时的各种时钟源、同步计数

时的PCLK1

b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2	INTENA	计数匹配中断使能 A	0: CMPAR寄存器与计数值(CNTBR)相等时，或者发生捕获输入事件时，该中断无效 1: CMPAR寄存器与计数值(CNTBR)相等时，或者发生捕获输入事件时，该中断使能	R/W
b1	CAPMDA	功能模式选择A	0: 比较输出功能 1: 捕获输入功能	R/W
b0	CSTA	定时器启动	0: 通道A定时器关闭 1: 通道A定时器启动 注：该位在硬件触发停止条件有效时，会自动变为0	R/W

注意：

- 该寄存器中提到的内部硬件触发事件 (bit31~bit28 和 bit15~bit12) 及异步计数时的 XTAL32 时钟源 (bit26 和 bit10)，在 USART 模块的 TIMEOUT 功能有效时均由 USART 模块提供输入，具体请参考 USART 章节介绍。

22.5.4 状态标志寄存器 (TMR0_STFLR)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b17	Reserved	-	读出时为“0”,写入时写“0”	R
b16	CMBF	计数匹配B	0: CMPBR寄存器的值与计数值不相等且未发生捕获输入动作 1: CMPBR寄存器的值与计数值相等或发生捕获输入动作	R/W
b15~b1	Reserved	-	读出时为“0”,写入时写“0”	R
b0	CMAF	计数匹配A	0: CMPAR寄存器的值与计数值不相等且未发生捕获输入动作 1: CMPAR寄存器的值与计数值相等或发生捕获输入动作	R/W

22.6 使用注意事项

- 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，对计数值(CNTAR)、基准值 (CMPAR)、启动位 (BCONR.CSTA)、状态位 (STFLR.CMAF) 进行写动作时，Timer0 从接收到写动作后经过 3 个异步计数时钟才将修改值写入对应的寄存器中。

23 实时时钟 (RTC)

23.1 简介

实时时钟 (RTC) 是一个以 BCD 码格式保存时间信息的计数器。记录从 00 年到 99 年间的具体日历时间。支持 12/24 小时两种时制，根据月份和年份自动计算日数 28、29 (闰年)、30 和 31 日。表 23-1 所示是其基本特性。

计数时钟源	外部低速振荡器 (32.768KHz) RTC 内部低速振荡器 (32.768KHz)
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

表 23-1 RTC 的基本规格

23.2 基本框图

RTC 的基本框图如图 23-1 所示。

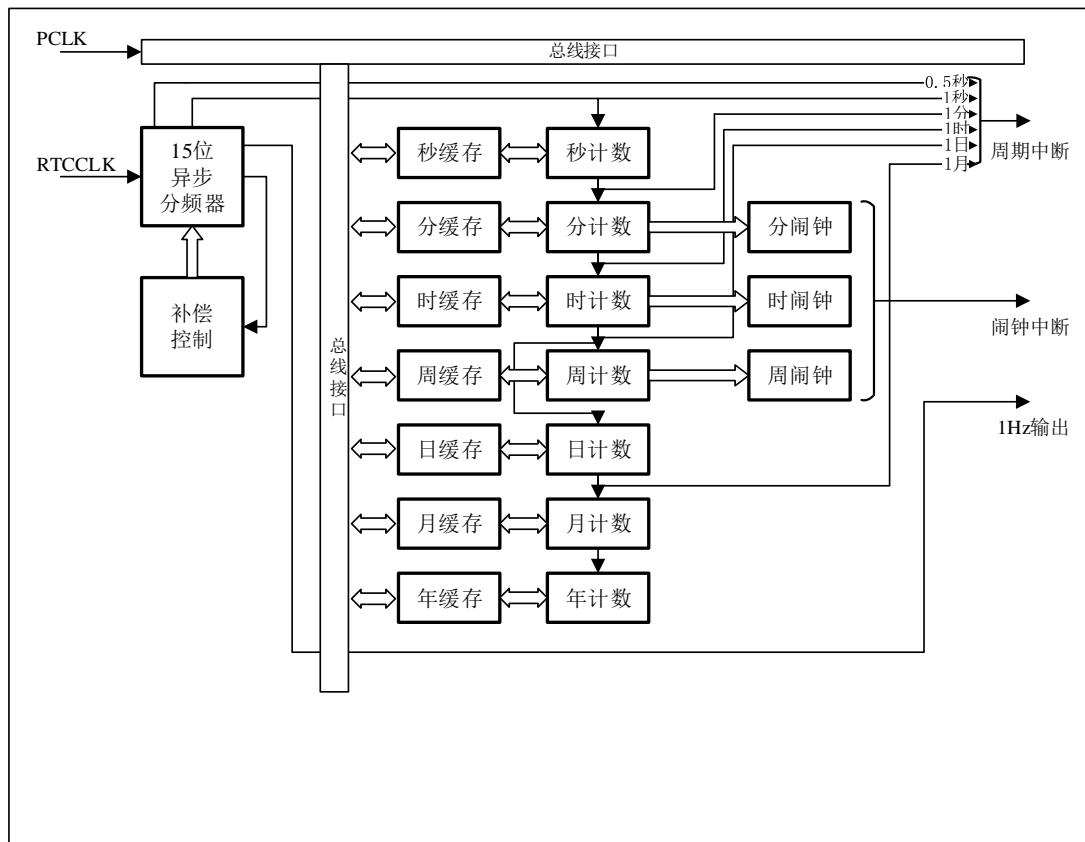


图 23-1 RTC 的基本框图

23.3 功能说明

23.3.1 上电设定

RTC 寄存器通过上电复位和设定控制寄存器 RESET 位复位所有寄存器，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 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。

23.3.2 RTC 计数开始设定

- 1) 上电后，除 RTC_CR0 以外的所有寄存器复位；也可以设定 RTC_CR0.RESET=0，确认 RESET 位为“0”后，设定 RTC_CR0.RESET=1，复位所有寄存器；
- 2) 设定 RTC_CR1.START=0，停止计数；
- 3) 设定系统时钟寄存器，打开副发振器，再设定 RTC_CR3，选择 RTC 计数时钟源；
- 4) 设定 RTC_CR1，设定时制、周期、1Hz 时钟输出；
- 5) 设定秒，分，时，日，周，月，年的日历计数寄存器；
- 6) 需要进行时钟误差补偿时，设定计数时钟误差补偿寄存器 RTC_ERRCRL，RTC_ERRCRH；
- 7) 清除寄存器 RTC_CR1 中的标志寄存器位，并使能中断；
- 8) 设定 RTC_CR1.START=1，计数开始。

23.3.3 系统低功耗模式切换

在 RTC 计数开始后，系统立即切换为低功耗模式时，请执行下列任意一种确认后再进行模式切换。

- 1) 在 RTC_CR1.START=1 设定后，经过 2 个以上的 RTC 计数时钟后再进行模式切换。
- 2) 在 RTC_CR1.START=1 设定后，设定 RTC_CR2.RWREQ=1，查询 RTC_CR2.RWEN=1。设定日历计数寄存器，再设定 RTC_CR2.RWREQ=0，查询 RTC_CR2.RWEN=0，进行模式切换。

23.3.4 读出计数寄存器

- 1) 在 RTC_CR1.START=1 设定后，经过 2 个以上的 RTC 计数时钟后，设定 RTC_CR2.RWREQ=1，进行日历寄存器读请求；
- 2) 查询直到 RTC_CR2.RWEN=1；
- 3) 读出全部或者部分秒，分，时，日，周，月，年计数寄存器值；
- 4) 设定 RTC_CR2.RWREQ=0；
- 5) 查询直到 RTC_CR2.RWEN=0。

23.3.5 写入计数寄存器

- 1) 在 RTC_CR1.START=1 设定后，经过 2 个以上的 RTC 计数时钟后，设定 RTC_CR2.RWREQ=1，进行日历寄存器写请求；
- 2) 查询直到 RTC_CR2.RWEN=1；
- 3) 写入全部或者部分秒，分，时，日，周，月，年计数寄存器值；
- 4) 设定 RTC_CR2.RWREQ=0。注意，须在 1 秒内完成所有写操作；
- 5) 查询直到 RTC_CR2.RWEN=0。

23.3.6 闹钟设定

- 1) 设定 RTC_CR2.ALME=0，闹钟禁止；
- 2) 设定 RTC_CR2.ALMIE=1，闹钟中断许可；
- 3) 分闹钟 RTC_ALMMIN，时闹钟 RTC_ALMHOUR，周闹钟 RTC_ALMEEK 设定；
- 4) 设定 RTC_CR2.ALME=1，闹钟许可；
- 5) 等待发生中断；
- 6) 当 RTC_CR2.ALMF=1 时，进入闹钟中断处理。

23.3.7 时钟误差补偿

由于外部副发振晶振在各种温度条件下工作存在误差，在需要得到高精度的计数结果时，需要对误差进行补偿。补偿方法参照 23.5.15 时钟误差补偿寄存器(RTC_ERRCRH、RTC_ERRCRL)。

23.3.8 1Hz 输出

RTC 可输出 1Hz 时钟，提供三种精度输出方式，第一种，无时钟误差补偿的普通精度 1Hz 输出；第二种，每 32 秒内平均补偿的分布式补偿 1Hz 输出和第三种每秒补偿的均匀式补偿 1Hz 输出。当时钟误差补偿功能有效 RTC_ERRCRH.COMPEN=1 时可选择分布式补偿 1Hz 输出和均匀式补偿 1Hz 输出。其中，

普通精度的 1Hz 输出设定如下：

- 1) 设定 RTC_CR0.RESET=0，确认 RESET 位为“0”后，设定 RTC_CR0.RESET=1，复位日历计数寄存器；
- 2) 设定 RTC_CR1.START=0，计数停止；
- 3) 1Hz 输出引脚设定；
- 4) RTC_CR1.ONEHZOE=1，时钟输出许可；
- 5) 设定 RTC_CR1.START=1，计数开始；
- 6) 等待 2 个计数周期以上；
- 7) 1Hz 输出开始。

分布式补偿 1Hz 输出设定如下：

- 1) 设定 RTC_CR0.RESET=0，确认 RESET 位为“0”后，设定 RTC_CR0.RESET=1，复位日历计数寄存器；
- 2) 设定 RTC_CR1.START=0，计数停止；
- 3) 1Hz 输出引脚设定；
- 4) RTC_CR1.ONEHZOE=1，时钟输出许可；
- 5) 时钟误差补偿寄存器 RTC_ERRCRL.COMP[7:0] 与 RTC_ERRCRH.COMP[8] 补偿数设定；
- 6) 时钟误差补偿寄存器 RTC_ERRCRH.COMPEN=1，误差补偿有效；
- 7) 设定 RTC_CR1.START=1，计数开始；
- 8) 等待 2 个计数周期以上；
- 9) 1Hz 输出开始。

均匀式补偿 1Hz 输出设定如下：

- 1) 设定 RTC_CR0.RESET=0，确认 RESET 位为“0”后，设定 RTC_CR0.RESET=1，

复位日历计数寄存器；

- 2) 设定 RTC_CR1.START=0，计数停止；
- 3) RTC 输出引脚设定；
- 4) RTC_CR1.ONEHZOE=1，时钟输出许可；
- 5) RTC_CR1.ONEHZSEL=1，选择输出均匀式补偿 1Hz 时钟；
- 6) 时钟误差补偿寄存器 RTC_ERRCRL.COMP[7:0] 与 RTC_ERRCRH.COMP[8] 补偿数设定；
- 7) 时钟误差补偿寄存器 RTC_ERRCRH.COMPEN=1，精度补偿有效；
- 8) 设定 RTC_CR1.START=1，计数开始；
- 9) 等待 2 个计数周期以上；
- 10) 1Hz 输出开始。

23.4 中断说明

RTC 支持 2 种中断类型。计时闹钟中断和定周期中断。

23.4.1 闹钟中断

闹钟中断 RTC_ALM，在控制寄存器 2 (RTC_CR2) 的 ALMIE=1 并且控制寄存器 2 (RTC_CR2) 的 ALME=1 时，若当前日历时间与分闹钟寄存器 (RTC_ALMMIN)、时闹钟寄存器 (RTC_ALMHOUR)、周闹钟寄存器 (RTC_ALMWEEK) 相等时，触发闹钟中断。闹钟配置独立的标志寄存器位 RTC_CR2.ALMF，对 RTC_CR1.ALMFCLR 位写“0”清除闹钟标志。闹钟中断只能配置通过 INTC.INT_SELx (x=0~31,44~49) 配置到相应的 NVIC 向量。

23.4.2 定周期中断

定周期中断 RTC_PRD，控制寄存器 2 (RTC_CR2) 的 PRDIE=1 时，选择的周期发生后，触发定周期唤醒中断。定周期中断没有配置独立的标志位，可通过 INTC.INT_SELx (x=0~31,44~49) 选择 RTC_PRD 中断到 NVIC 向量，查询 NVIC 对应标志位。

23.5 寄存器说明

表 23-2 所示，为 RTC 模块的寄存器列表。

寄存器基地址：0x4004_C000

寄存器名	符号	偏移量	位宽	复位值
控制寄存器0	RTC_CR0	0x0000	8	不定
控制寄存器1	RTC_CR1	0x0004	8	0x00
控制寄存器2	RTC_CR2	0x0008	8	0x00
控制寄存器3	RTC_CR3	0x000C	8	0x00
秒计数寄存器	RTC_SEC	0x0010	8	0x00
分计数寄存器	RTC_MIN	0x0014	8	0x00
时计数寄存器	RTC_HOUR	0x0018	8	0x12
周计数寄存器	RTC_WEEK	0x001C	8	0x00
日计数寄存器	RTC_DAY	0x0020	8	0x00
月计数寄存器	RTC_MON	0x0024	8	0x00
年计数寄存器	RTC_YEAR	0x0028	8	0x00
分闹钟寄存器	RTC_ALMMIN	0x002C	8	0x00
时闹钟寄存器	RTC_ALMHOUR	0x0030	8	0x12
周闹钟寄存器	RTC_ALMWEEK	0x0034	8	0x00
时钟误差补偿寄存器	RTC_ERRCRH	0x0038	8	0x00
时钟误差补偿寄存器	RTC_ERRCRL	0x003C	8	0x20

表 23-2 寄存器列表

23.5.1 控制寄存器 0 (RTC_CR0)

复位值: 不定

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

位	标记	位名	功能	读写
b31~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	RESET	RTC日历计数 器复位	写入状态 0: 初始化寄存器无效 1: 初始化寄存器有效 初始化所有的RTC寄存器。 读出状态 0: 正常计数状态或RTC软件复位结束 1: RTC处于复位状态 注意: 请在写入“1”前确认该位为“0”, 否则将无法初始化。	R/W

23.5.2 控制寄存器 1 (RTC_CR1)

复位值: 0x00

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								START	ONEHZSEL	ONEHZOE	ALMFCLR	AMPM	PRDS[2]	PRDS[1]	PRDS[0]

位	标记	位名	功能	读写																																
b31~b8	Reserved	-	读出时为“0”，写入时写“0”	R/W																																
b7	START	RTC计数开始	0: RTC计数停止 1: RTC计数工作	R/W																																
b6	ONEHZSEL	1Hz输出选择	0: 分布式补偿1Hz输出 1: 均匀式补偿1Hz输出 注意: RTC_ERRCRH.COMPEN=1时, 该位设定有效。	R/W																																
b5	ONEHZOE	1Hz输出使能	0: 1Hz输出禁止 1: 1Hz输出许可	R/W																																
b4	ALMFCLR	ALMF标志清除	0: 清除RTC_CR2.ALMF标志寄存器 1: 无效	R/W																																
b3	AMPM	时制选择	0: 12小时时制 1: 24小时时制	R/W																																
b2~0	PRDS[2:0]	周期中断选择	周期选择设定:	R/W																																
			<table border="1" style="width: 100%; text-align: center;"> <tr> <th>PRDS[2]</th><th>PRDS[1]</th><th>PRDS[0]</th><th>周期选择</th></tr> <tr> <td>0</td><td>0</td><td>0</td><td>不选择</td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>每0.5秒周期</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>每1秒周期</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>每1分周期</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>每1时周期</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>每1日周期 (每日00时00分00秒)</td></tr> <tr> <td>1</td><td>1</td><td>X</td><td>每1月周期 (每月1日00时00分00秒)</td></tr> </table>	PRDS[2]	PRDS[1]	PRDS[0]	周期选择	0	0	0	不选择	0	0	1	每0.5秒周期	0	1	0	每1秒周期	0	1	1	每1分周期	1	0	0	每1时周期	1	0	1	每1日周期 (每日00时00分00秒)	1	1	X	每1月周期 (每月1日00时00分00秒)	
PRDS[2]	PRDS[1]	PRDS[0]	周期选择																																	
0	0	0	不选择																																	
0	0	1	每0.5秒周期																																	
0	1	0	每1秒周期																																	
0	1	1	每1分周期																																	
1	0	0	每1时周期																																	
1	0	1	每1日周期 (每日00时00分00秒)																																	
1	1	X	每1月周期 (每月1日00时00分00秒)																																	

注意: 在START=1计数过程中写入周期选择时, 为防止误动作请将周期中断许可关闭。并且在写入后应将相关标志位清除。

23.5.3 控制寄存器 2 (RTC_CR2)

复位值: 0x00

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

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”，写入时写“0”	R/W
b7	ALME	闹钟使能	0：闹钟功能禁止 1：闹钟功能许可	R/W
b6	ALMIE	闹钟中断使能	0：闹钟中断禁止 1：闹钟中断许可	R/W
b5	PRDIE	周期中断使能	0：周期中断禁止 1：周期中断许可	R/W
b4	Reserved	-	读出时为“0”，写入时写“0”	R/W
b3	ALMF	闹钟标志	0：闹钟不匹配 1：闹钟匹配	R
b2	Reserved	-	读出时不定，写入时写“0”	R/W
b1	RWEN	读出/写入允许	0：读出/写入禁止 1：读出/写入允许 注意：日历寄存器读写允许标志。在读出/写入前请确认该位是否为“1”。 “。日历寄存器包括秒，分，时，周，日，月，年计数寄存器。	R/W
b0	RWREQ	读出/写入请求	0：正常计数模式 1：读出/写入请求 注意：在读出/写入日历寄存器时请将该位置“1”，请求读写，由于计数器在连续计数，请在1秒的时间内完成读出/写入操作并将该位清“0”。	R/W

23.5.4 控制寄存器 3 (RTC_CR3)

复位值: 0x00

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

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	RCKSEL	RTC计数时钟选择	0: 选择副发振器 1: 选择LOCO	R/W
b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4	LCREN	低速发振器使能	0: 低速发振器停止 1: 低速发振器工作 注意: 低速发振的工作与停止由时钟控制电路和LCREN任何一个设定值决定。低速发振作为RTC时钟源时请设定LCREN位使能。	R/W
b3~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

23.5.5 秒计数寄存器 (RTC_SEC)

复位值: 0x00

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								-	SECD[2:0]	SECU[3:0]					

位	标记	位名	功能	读写
b31~b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6~b4	SECD[2:0]	秒十位	秒十位计数值	R/W
b3~b0	SECU[3:0]	秒个位	秒个位计数值	R/W

表示 0-59 秒, 采用十进制计数。请写入十进制 0-59 的 BCD 码, 写入错误值时, 写入值将被忽略。

23.5.6 分计数寄存器 (RTC_MIN)

复位值: 不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								-	MIND[2:0]			MINU[3:0]			

位	标记	位名	功能	读写
b31~b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6~b4	MIND[2:0]	分十位	分十位计数值	R/W
b3~b0	MINU[3:0]	分个位	分个位计数值	R/W

表示 0-59 分, 采用十进制计数。请写入十进制 0-59 的 BCD 码, 写入错误值时, 写入值将被忽略。

23.5.7 时计数寄存器 (RTC_HOUR)

复位值: 0x12

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

位	标记	位名	功能	读写
b31~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5~b4	HOURD[1:0]	时十位	时十位计数值	R/W
b3~b0	HOURU[3:0]	时个位	时个位计数值	R/W

24 小时时制时, 表示 0-23 小时。12 小时时制时, b5=0 表示 AM, 则 01~12 表示上午;

b5=1 表示 PM, 则 21~32 表示下午。

请根据控制位 AMPM 的值, 设定正确十进制的 0~23 或者 01~12,21~32 的 BCD 码。写入超出范围的值将被忽略。

具体时间表示参考下表:

24小时制	AMPM=1	12小时制	AMPM=0
时间	寄存器表示	时间	寄存器表示
00时	00H	AM 12时	12H
01时	01H	AM 01时	01H
02时	02H	AM 02时	02H
03时	03H	AM 03时	03H
04时	04H	AM 04时	04H
05时	05H	AM 05时	05H
06时	06H	AM 06时	06H
07时	07H	AM 07时	07H
08时	08H	AM 08时	08H
09时	09H	AM 09时	09H
10时	10H	AM 10时	10H
11时	11H	AM 11时	11H
12时	12H	PM 12时	32H
13时	13H	PM 01时	21H
14时	14H	PM 02时	22H
15时	15H	PM 03时	23H
16时	16H	PM 04时	24H
17时	17H	PM 05时	25H

18时	18H	PM 06时	26H
19时	19H	PM 07时	27H
20时	20H	PM 08时	28H
21时	21H	PM 09时	29H
22时	22H	PM 10时	30H
23时	23H	PM 11时	31H

23.5.8 日计数寄存器 (RTC_DAY)

复位值: 0x00

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

位	标记	位名	功能	读写
b31~b6	Reserved	-	读出时为“0”，写入时写“0”	R/W
b5~b4	DAYD[1:0]	日十位	日十位计数值	R/W
b3~b0	DAYU[3:0]	日个位	日个位计数值	R/W

十进制表示 1~31 日，自动计算闰年和月份。具体表示如下：

月份	日计数表示
2月（普通年）	01~28
2月（闰年）	01~29
4、6、9、11月	01~30
1、3、5、7、8、10、12月	01~31

23.5.9 周计数寄存器 (RTC_WEEK)

复位值: 0x00

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

位	标记	位名	功能	读写
b31~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~b0	WEEK[2:0]	周	周计数值	R/W

十进制 0~6 表示周日~周六。请写入正确的十进制 0~6 的 BCD 码，写入其他值，将被忽略。周计数值对应关系如下：

周	周计数表示
周日	00H
周一	01H
周二	02H
周三	03H
周四	04H
周五	05H
周六	06H

23.5.10 月计数寄存器 (RTC_MON)

复位值: 0x00

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

位	标记	位名	功能	读写
b31~b5	Reserved	-	读出时为“0”，写入时写“0”	R/W
b4~b0	MON[4:0]	月	月计数值	R/W

十进制 1~12 表示 1~12 月。请写入正确的十进制 1~12 的 BCD 码，写入其他值，将被忽略。

23.5.11 年计数寄存器 (RTC_YEAR)

复位值: 0x00

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

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”，写入时写“0”	R/W
b7~b4	YEARD[3:0]	年十位	年十位计数值	R/W
b3~b0	YEARU[3:0]	年个位	年个位计数值	R/W

十进制 0~99 表示 0~99 年。根据月进位计数。自动计算闰年如：00、04、08、...、92、96 等。请写入正确的十进制年计数值，写入错误值将被忽略。

23.5.12 分闹钟寄存器 (RTC_ALMMIN)

复位值: 0x00

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								-	ALMMIND[2:0]			ALMMINU[3:0]			

位	标记	位名	功能	读写
b31~b7	Reserved	-	读出时为“0”，写入时写“0”	R/W
b6~b4	ALMMIND[2:0]	分闹钟十位	分闹钟十位匹配值	R/W
b3~b0	ALMMINU[3:0]	分闹钟个位	分闹钟个位匹配值	R/W

请设定十进制 0~59 的 BCD 码。写入其他值，不会发生闹钟匹配。

23.5.13 时闹钟寄存器 (RTC_ALMHOUR)

复位值: 0x12

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

位	标记	位名	功能	读写
b31~b6	Reserved	-	读出时为“0”，写入时写“0”	R/W
b5~b4	ALMHOURD[1:0]	时闹钟十位	时闹钟十位匹配值	R/W
b3~b0	ALMHOURU[3:0]	时闹钟个位	时闹钟个位匹配值	R/W

请根据时制设定正确的闹钟匹配值，否则不会发生时闹钟匹配。

23.5.14 时闹钟寄存器 (RTC_ALMWEEK)

复位值: 0x00

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

位	标记	位名	功能	读写
b31~b7	Reserved	-	读出时为“0”，写入时写“0”	R/W
b6~b0	ALMWEEK[6:0]	周闹钟	周闹钟匹配值。 b0~b6分别对应周日~周六，对应为置“1”时，代表每周该日闹钟有效。如，b0=1，b5=1代表周日和周五闹钟设定有效。	R/W

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。

23.5.15 时钟误差补偿寄存器（RTC_ERRCRH、RTC_ERRCRL）

复位值: 0x0000_0020

RTC_ERRCRH

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

RTC_ERRCRL

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

RTC_ERRCRH

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”，写入时写“0”	R/W
b7	COMPEN	补偿使能	0: 时钟误差补偿无效 1: 时钟误差补偿有效	R/W
B6~b1	Reserved	-	读出时为“0”，写入时写“0”	R/W
b0	COMP[8]	补偿值	与COMP[7:0]共同设定补偿值	R/W

RTC_ERRCRL

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”，写入时写“0”	R/W
b7~b0	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。具体补偿精度请参考下表：	R/W
补偿值设定				补偿数
COMPEN COMP[8:0]				
1				-275.5ppm
				-274.6ppm
				~
				-30.5ppm
				~
				-0.96ppm

	0	0	0	1	0	0	0	0	0	0ppm
	0	0	0	1	0	0	0	0	1	+0.96ppm
	~	~	~	~	~	~	~	~	~	~
	0	0	1	0	0	0	0	0	0	+30.5ppm
	~	~	~	~	~	~	~	~	~	~
	0	1	1	1	1	1	1	1	0	+212.0ppm
	0	1	1	1	1	1	1	1	1	+212.9ppm
0	X	X	X	X	X	X	X	X	X	无效

补偿计算说明：

当默认状态下直接输出 1Hz 时钟，通过测定该时钟的精度，计算补偿目标值。

假设实际测定值为 0.9999888Hz，则：

$$\text{实际发振频率} = 32768 \times 0.9999888 \approx 32767.63$$

$$\begin{aligned}\text{补偿目标值} &= (\text{实际发振频率} - \text{目标频率}) / \text{目标频率} \times 10^6 \\ &= (32767.96 - 32768) / 32768 \times 10^6 \\ &= -11.29\text{ppm}\end{aligned}$$

设定值计算：

$$\text{COMP}[8 : 0] = \left(\frac{\text{补偿目标值}[\text{ppm}] \times 2^{15}}{10^6} \right)_{\text{取2的补码}} + 0001.00000\text{B}$$

如果补偿目标值为+20.3ppm，计算相应的寄存器值如下：

$$\begin{aligned}\text{COMP}[8:0] &= (20.3 \times 215/106) \text{取2的补码} + 0001.00000\text{B} \\ &= (0.6651904) \text{取2的补码} + 0001.00000\text{B} \\ &= 0000.10101\text{B} + 0001.00000\text{B} \\ &= 0001.10101\text{B}\end{aligned}$$

如果补偿目标值为-20.3ppm，计算相应的寄存器值如下：

$$\begin{aligned}\text{COMP}[8:0] &= (-20.3 \times 215/106) \text{取2的补码} + 0001.00000\text{B} \\ &= (-0.6651904) \text{取2的补码} + 0001.00000\text{B} \\ &= 1111.01011\text{B} + 0001.00000\text{B} \\ &= 0000.01011\text{B}\end{aligned}$$

24 看门狗计数器 (WDT/SWDT)

24.1 简介

看门狗计数器有两个，一种是计数时钟源为专用内部 RC (SWDTLRC:10KHz) 的专用看门狗计数器(SWDT)，另一种是计数时钟源为 PCLK3 的通用看门狗计数器(WDT)。专用看门狗和通用看门狗是 16 位递减计数器，用来监测由于外部干扰或不可预见的逻辑条件造成应用程序背离正常的运行而产生的软件故障。

两个看门狗都支持窗口功能。在计数开始前可预设窗口区间，计数值位于窗口区间时，可刷新计数器，计数重新开始。基本特性如表 24-1。

计数时钟	SWDT: SWDTLRC的1/16/32/64/128/256/2048分频 WDT: PCLK3的4/64/128/256/512/1024/2048/8192分频
最长溢出时间	SWDT:3.72hour(max) WDT:10.7s (PCLK3=50MHz)
计数模式	递减计数
窗口功能	可设定窗口区间，定义刷新动作的允许区间
启动方式	1) 硬件启动 2) 软件启动(SWDT不支持软件启动)
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

表 24-1 看门狗计数器的基本特性

24.2 功能说明

24.2.1 启动看门狗

看门狗计数器的启动方式有两种：硬件启动方式和软件启动方式。(SWDT 只支持硬件启动)

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。

24.2.2 硬件启动方式

ICG0 寄存器的位 16(WDTAUTS)、位 0(SWDTAUTS)为 0 时，为硬件启动方式。选择硬件启动方式时，WDT_CR 寄存器的相关设定信息无效。

硬件启动方式时，在复位期间将 ICG0 寄存器里的 WDT/SWDT 相关设定（计数时钟、窗口设定值、计数周期等）载入到 WDT/SWDT 的模块中，复位之后、计数器按照设定自动开始计数。图 24-1 为硬件启动方式的动作例。

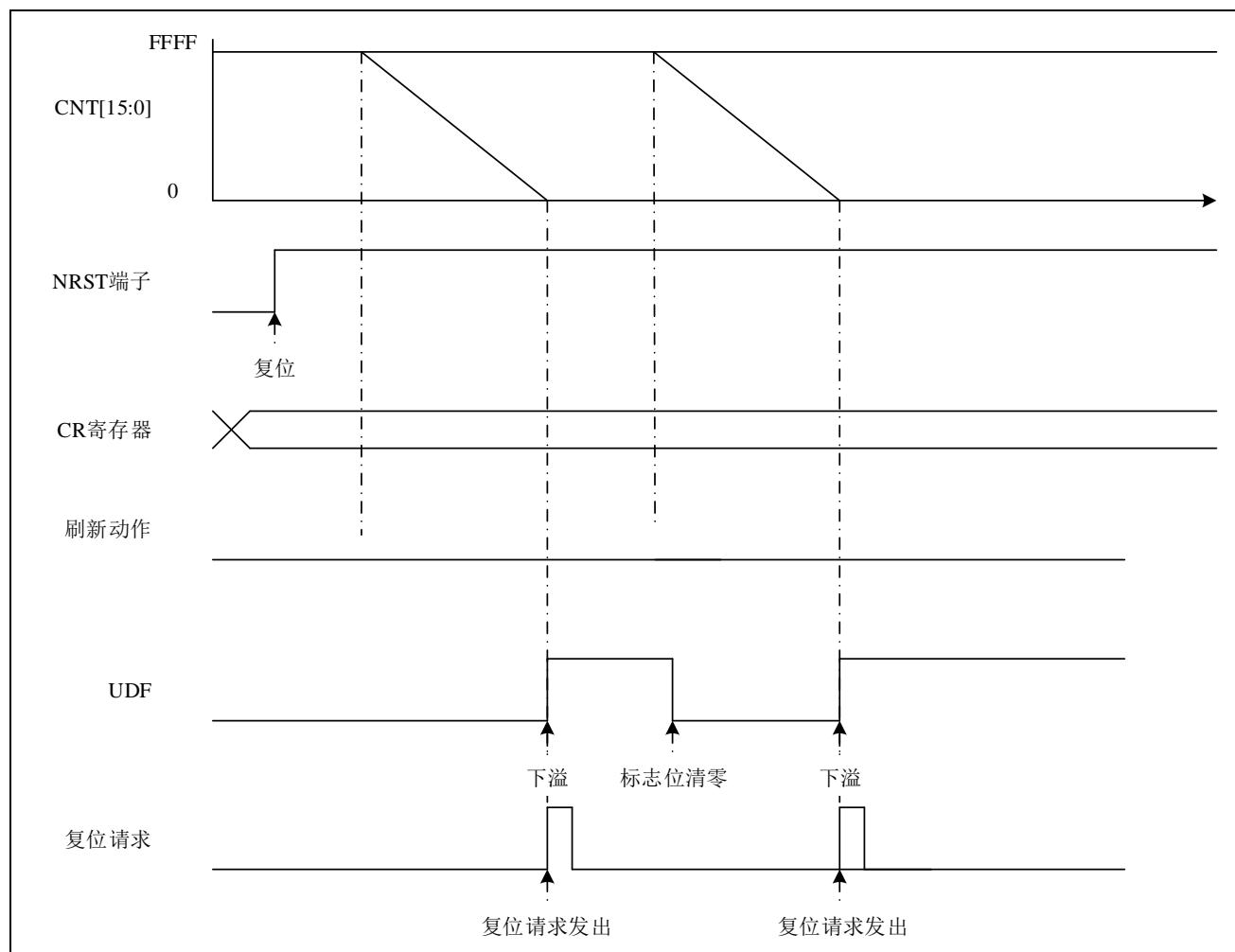


图 24-1 硬件启动例

24.2.3 软件启动方式

ICG0 寄存器的位 16(WDTAUTS)为 1 时，通过设定刷新寄存器的方式启动 WDT 为软件启动方式。复位后，设定 WDT_CR 寄存器中的计数时钟、窗口设定值、计数周期等，然后执行刷新动作，计数器就开始计数。WDT_CR 设定只能 1 次，再次设定写入值无效。图 24-2 为软件启动方式的动作例。

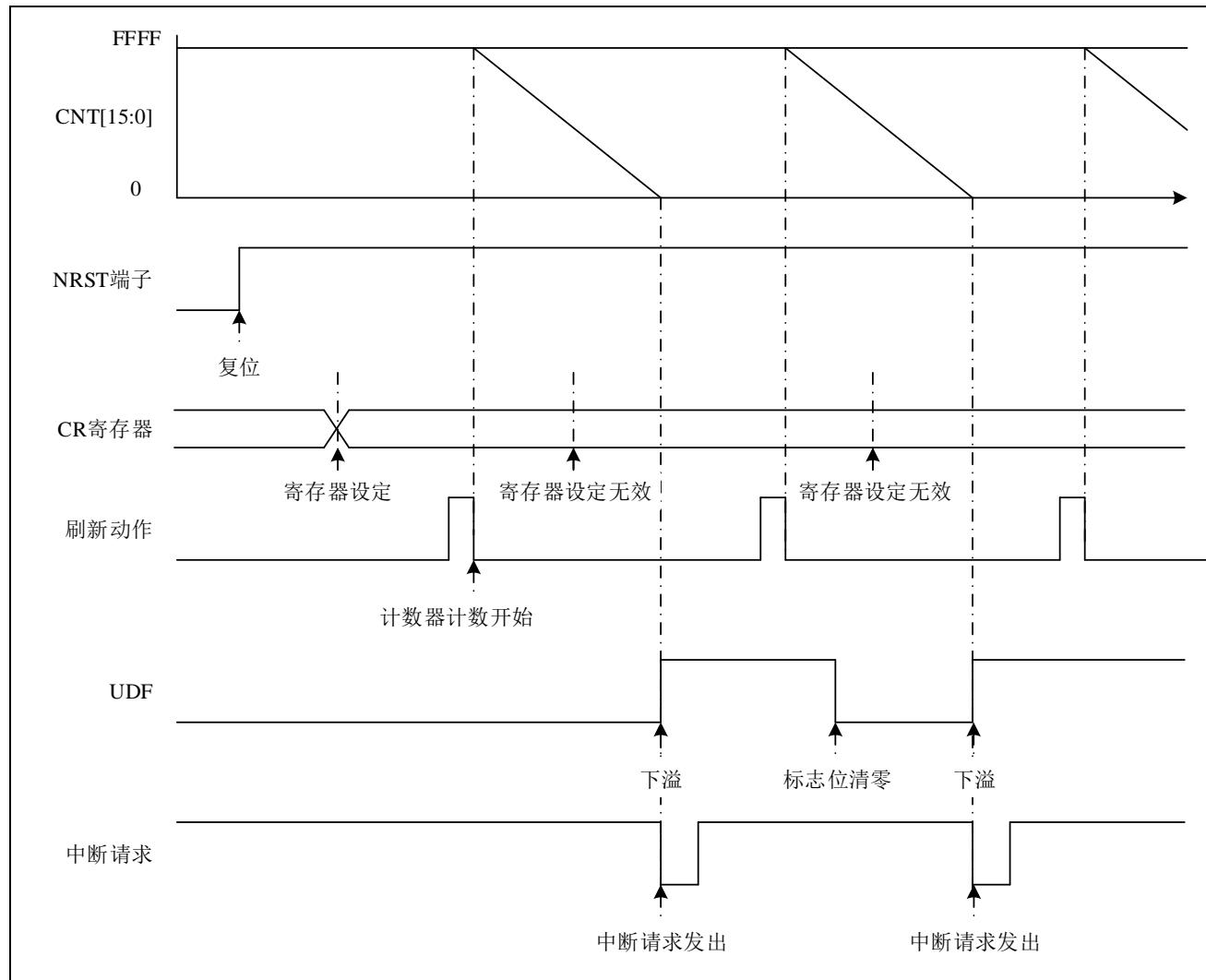


图 24-2 软件启动例

24.2.4 刷新动作

(S)WDT_RR 寄存器中先写 0x0123、再写 0x3210 完成一次刷新动作，WDT/SWDT 的计数器就开始计数(软件启动)或重新开始计数。

(S)WDT_RR 寄存器在写 0x0123、0x3210 之间，若对发生对其他寄存器访问或读取(S)WDT_RR 寄存器等，不影响正常的刷新动作。

如图 24-3 所示动作示例。

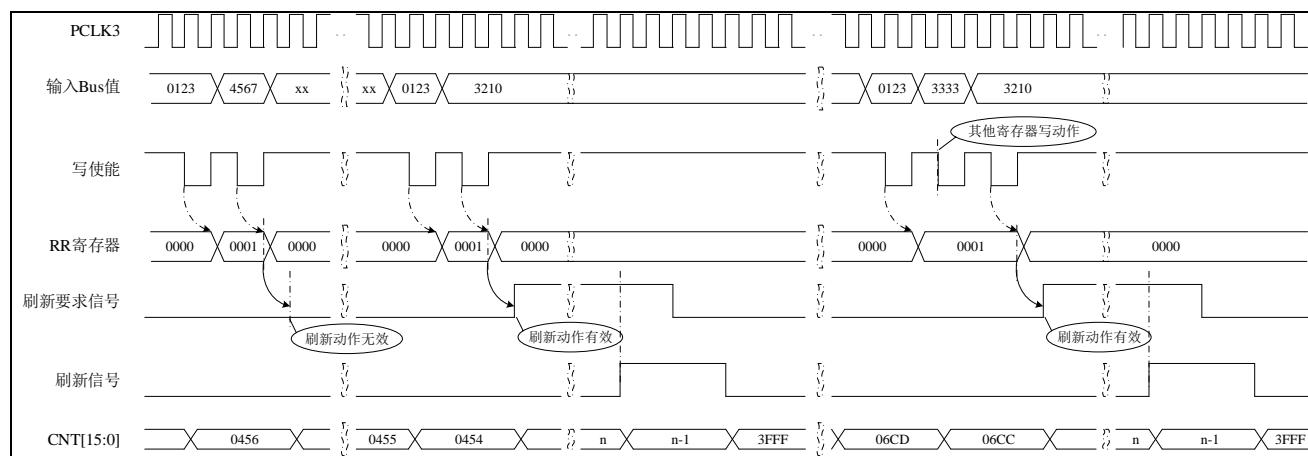


图 24-3 各种刷新动作时序示例（动作确认，刷新要求信号的下降沿等）

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。

24.2.5 标志位

刷新错误标志位和计数下溢标志位在复位请求和中断请求的情况下都会保持。当复位解除或进入中断后，可以通过查询标志位来确认复位或中断原因。

标志位清零：先读“1”再写“0”。

刷新错误或者计数下溢标志位为置位时，硬件启动模式看门狗计数不停止。对于 SWDT 来说，当对标志位写“0”时，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。对于 WDT 来说，当对标志位写“0”时，最多需要经过 5 个 PCLK3 时间后，寄存器位才能被清零。另，在发生刷新错误或者下溢错误的一定时间内，对标志位读“1”清零无效，这段时间为：1 个计数周期+2 个 SWDTLRC(SWDT 模块)；1 个计数周期+2 个 PCLK3 (WDT 模块)。

24.2.6 中断复位

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。通过设定 ICG0 的 SWDTITS 位，来决定产生中断请求还是复位请求。

WDT 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。硬件启动时通过设定 ICG0 的 WDTITS 位，软件启动时通过设定 WDT_CR 寄存器 ITS 位，来决定产生中断请求还是复位请求。

WDT/SWDT 的中断复位产生条件有两种。一种是计数器计数产生下溢；一种是在刷新允许区间之外执行刷新动作，产生刷新错误。

24.2.7 计数下溢

如图 24-4 例，在递减计数到零时产生下溢。

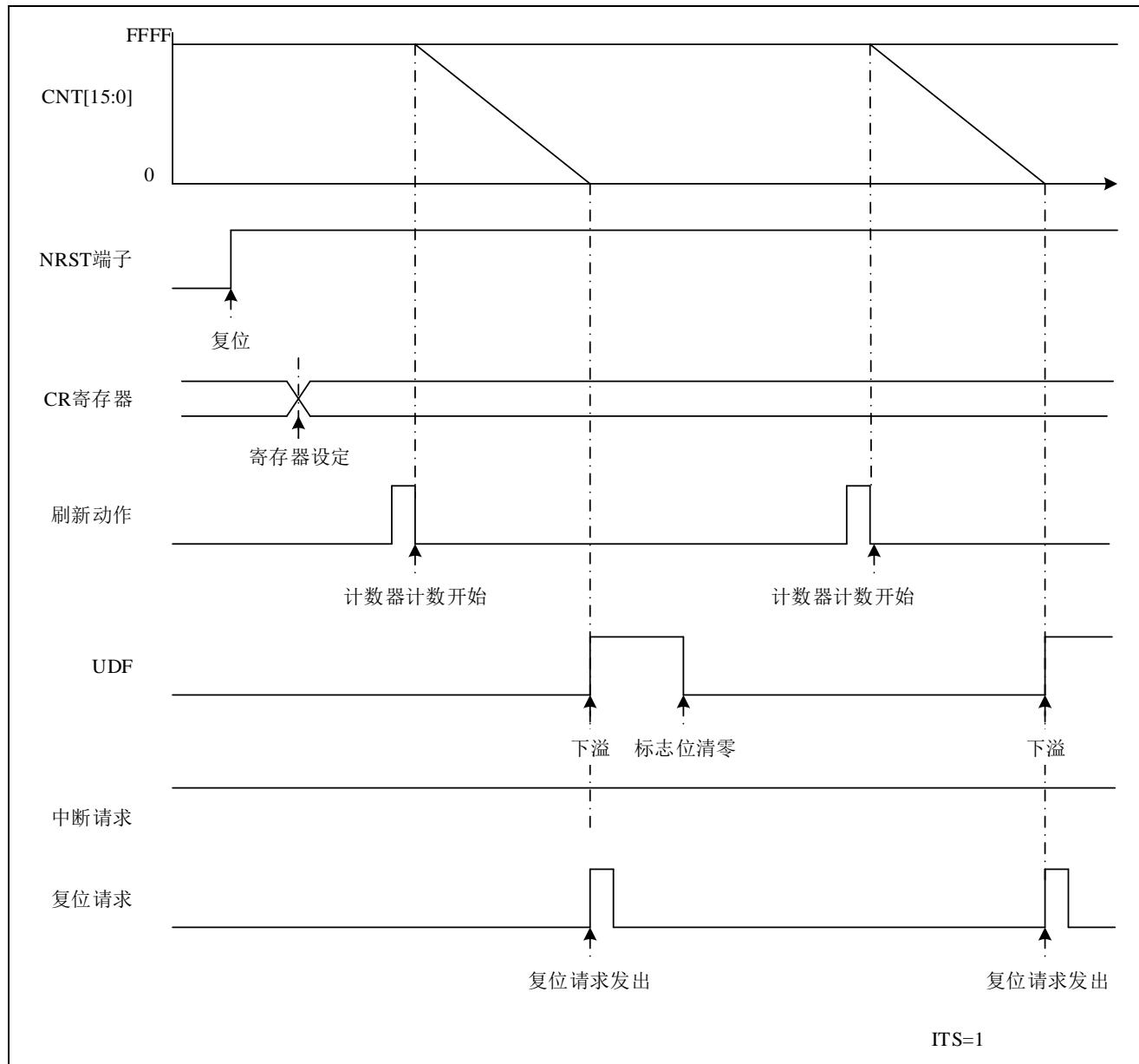


图 24-4 计数器下溢动作例

24.2.8 刷新错误

在设定了窗口区间后，只有在窗口区间内执行刷新动作时计数器才会被刷新、重新开始计数，在窗口区间外执行刷新动作时产生刷新错误。图 24-5 为刷新动作例。

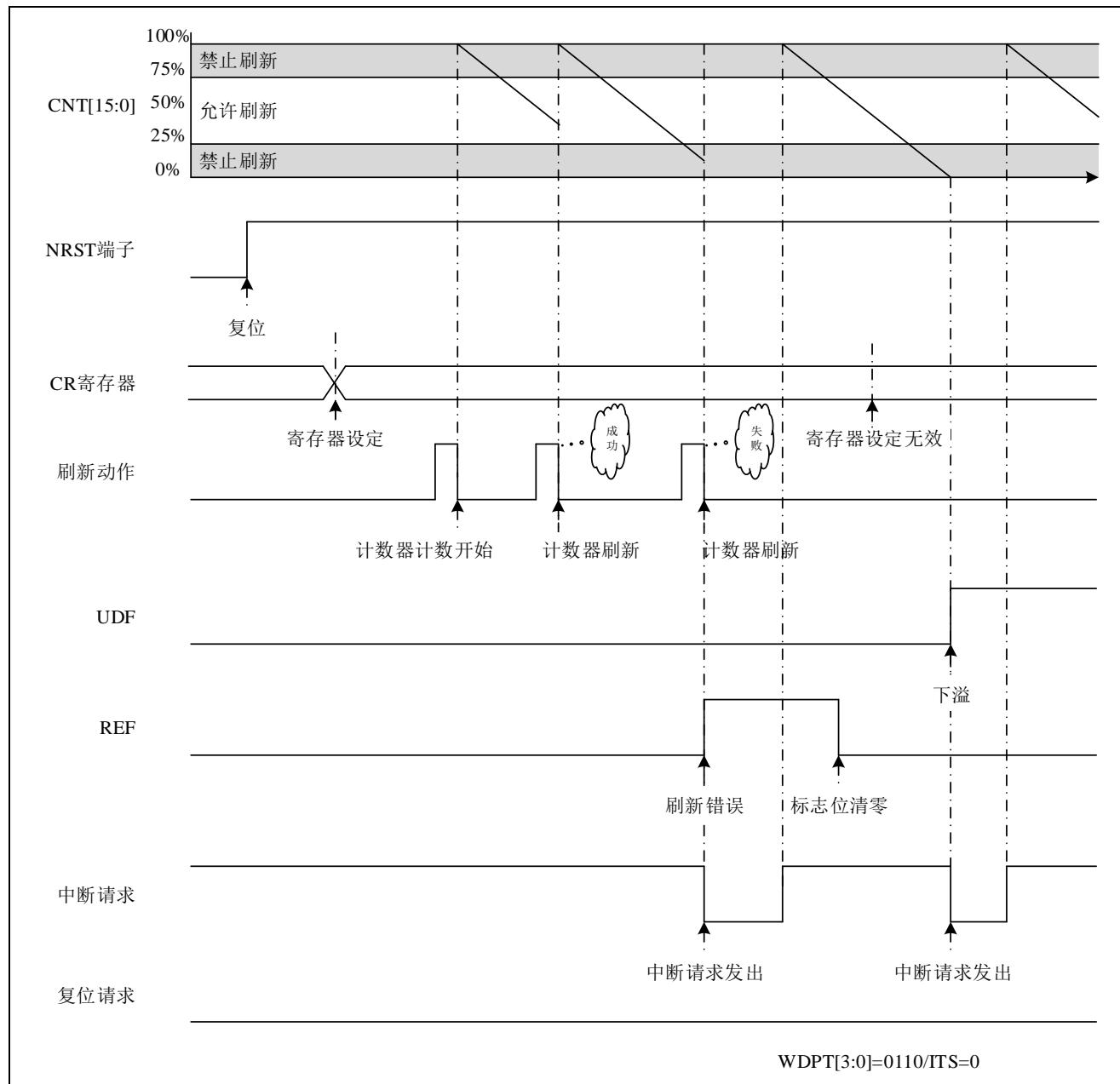


图 24-5 计数器刷新动作例

24.3 寄存器说明

表 24-2 所示，为 SWDT 和 WDT 模块的寄存器列表。

WDT_BASE_ADDR: 0x4004_9000

SWDT_BASE_ADDR: 0x4004_9400

寄存器名	符号	偏移地址	位宽	复位值
SWDT状态寄存器	SWDT_SR	0x04	32	0x0000_0000
SWDT刷新寄存器	SWDT_RR	0x08	32	0x0000_0000
WDT控制寄存器	WDT_CR	0x00	32	0x8001_0FF3
WDT状态寄存器	WDT_SR	0x04	32	0x0000_0000
WDT刷新寄存器	WDT_RR	0x08	32	0x0000_0000

表 24-2 寄存器列表

24.3.1 控制寄存器 (WDT_CR)

复位值: 0x8001_0FF3

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

位	标记	位名	功能	读写
b31	ITS	刷新错误/溢出 中断/复位选择	0: 中断请求 1: 复位请求	R/W
b30~b17	Reserved	-	读出时为“0”,写入时写“0”	R
b16	SLPOFF	WDT在Sleep模式 下计数禁止	0: WDT在sleep模式下计数许可 1: WDT在sleep模式下计数禁止	R/W
b15~b13	Reserved	-	读出时为“0”,写入时写“0”	R
b11~b8	WDPT[3:0]	刷新允许区域计 数值百分比	0000: 0%~100% 0001: 0%~25% 0010: 25%~50% 0011: 0%~50% 0100: 50%~75% 0101: 0%~25%,50%~75% 0110: 25%~75% 0111: 0%~75% 1000: 75%~100% 1001: 0%~25%,75%~100% 1010: 25%~50%,75%~100% 1011: 0%~50%,75%~100% 1100: 50%~100% 1101: 0%~25%,50%~100% 1110: 25%~100% 1111: 0%~100%	R/W
b7~b4	CKS[3:0]	计数时钟	0010: PCLK3/4 0110: PCLK3/64 0111: PCLK3/128 1000: PCLK3/256 1001: PCLK3/512 1010: PCLK3/1024 1011: PCLK3/2048 1101: PCLK3/8192 其余值: 预留功能	R/W

b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R
b1~b0	PERI[1:0]	计数周期	00: 256 cycle 01: 4096 cycle 10: 16384 cycle 11: 65536 cycle	R/W

24.3.2 状态寄存器 (SWDT_SR、WDT_SR)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b18	Reserved	-	读出时为“0”，写入时写“0”	R
b17	REF	刷新错误标志	0: 没有刷新错误 1: 发生刷新错误 对该位读出1后写入0，该位清零。	R/W
b16	UDF	计数下溢标志	0: 没有计数下溢 1: 发生计数下溢 对该位读出1后写入0，该位清零。	R/W
b15~b0	CNT[15:0]	计数值	计数器当前计数值	R

24.3.3 刷新寄存器（SWDT_RR、WDT_RR）

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”，写入时写“0”	R
b15~b0	RF[15:0]	刷新值	依次写入0x0123、0x3210之后，完成刷新动作。 当寄存器写入0x0123后，读出寄存器为0x0000_0001；其余情况读出值都是0x0000_0000。	R/W

24.4 使用注意事项

SWDT 动作时，周边时钟 PCLK3 的动作频率必须大于或等于计数时钟（ICG0.SWDTCKS[3:0]设定 SWDTLRC 分频后的时钟）的 4 倍，即 PCLK3 频率≥计数时钟频率 x4。

25 通用同步异步收发器 (USART)

25.1 简介

本产品搭载通用串行收发器模块 (USART) 4 个单元。通用串行收发器模块 (USART) 能够灵活地与外部设备进行全双工数据交换；本 USART 支持通用异步串行通信接口 (UART)，时钟同步通信接口，智能卡接口 (ISO/IEC7816-3)。支持调制解调器操作 (CTS/RTS 操作)，多处理器操作。和 Timer0 模块配合支持 UART 接收 TIMEOUT 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
- 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

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

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)

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

25.2 USART 系统框图

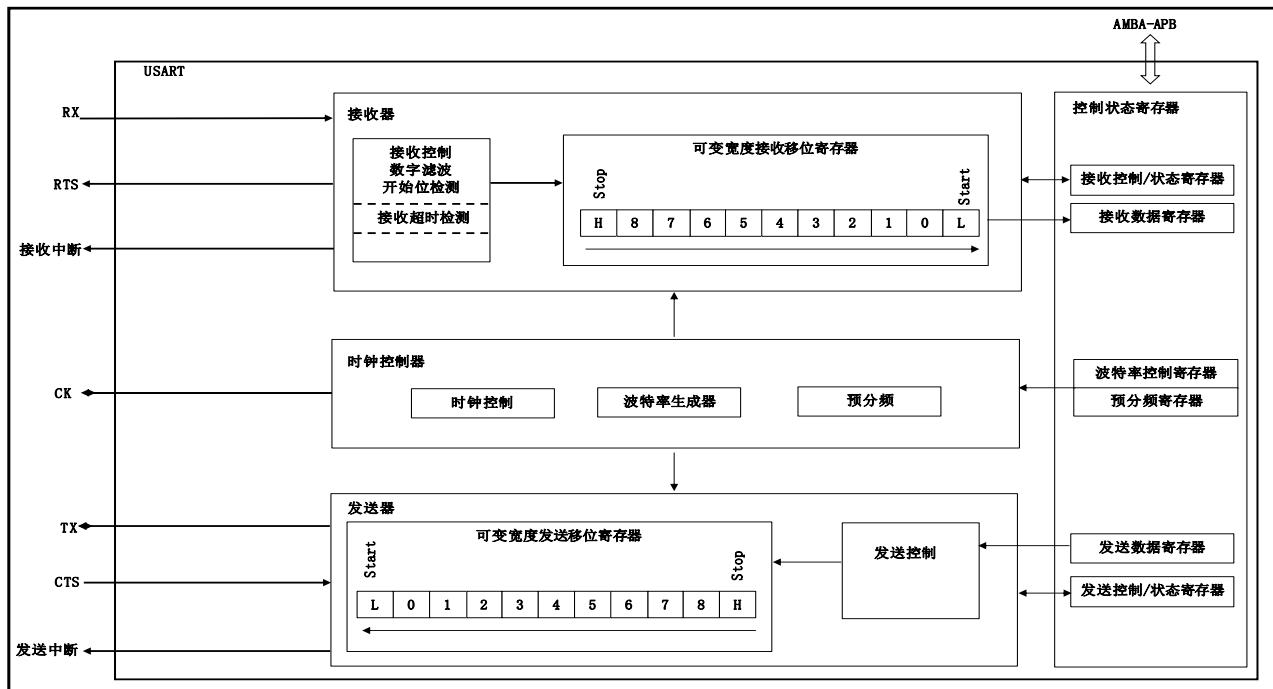


图 25-1 USART 系统框图

25.3 管脚说明

管脚名	方向	功能描述
USARTn_CK	输入输出	时钟
USARTn_TX	输出	发送数据管脚
USARTn_RX	输入	接收数据管脚
USARTn_CTS	输入	调制解调器操作管脚
USARTn_RTS	输出	调制解调器操作管脚

n:1~4

表 25-1 USART 管脚说明

25.4 功能说明

本章将对 UART, 多处理器通信, 智能卡, 时钟同步模式的功能详细说明。

25.4.1 UART

25.4.1.1 时钟

UART 可以选择内部波特率生成器生成的时钟(内部时钟源)或 USARTn_CK 管脚输入的时钟(外部时钟源)作为通信的时钟源。

内部时钟源

USARTn_CR2.CLKC[1:0]位设定为 00b 或者 01b 时选择时钟源为内部时钟源即内部波特率生成器生成的时钟。

USARTn_CR2.CLKC[1:0]=00b 时 USARTn_CK 管脚不作为时钟管脚使用, 可以作为普通 IO 使用。

USARTn_CR2.CLKC[1:0]=01b 时从 USARTn_CK 管脚输出与通信波特率相同频率的时钟。

内部波特率生成器的时钟源由 USARTn_PR.PSC[1:0]位的设定选择为 PCLK, PCLK/4, PCLK/16, PCLK/64。

外部时钟源

USARTn_CR2.CLKC[1:0]位设定为 10b 或者 11b 时选择时钟源为从 USARTn_CK 管脚输入的外部时钟, 输入时钟的频率为波特率的 16 倍(USARTn_CR1.OVER8=0)或者 8 倍(USARTn_CR1.OVER8=1)。

最高波特率

内部时钟源时, 内部波特率生成器生成的波特率计算公式为:

$$B = \frac{C}{8 \times (2 - OVER8) \times (DIV_Integer + 1)}$$

B: 波特率 单位: MBps

C: USARTn_PR.PSC[1:0]位设定的时钟 (PCLK,PCLK/4,PCLK/16,PCLK/64) 单位: MHz

OVER8:USARTn_CR1.OVER8 设定值

DIV_Integer:USARTn_BRR.DIV_Integer 设定值

最高波特率为 PCLK/8(MBps)。

外部时钟源时，外部输入 UART 时钟的最高频率要求为 PCLK(MHz)/4，所以时钟源为外部输入时钟时最高波特率为 PCLK/64(MBps) (USARTn_CR1.OVER8=0 时) 或者 PCLK/32 (MBps) (USARTn_CR1.OVER8=1 时)。

需要注意的是，UART 最高通信波特率除了以上描述的基于 PCLK 的计算方法外，还需要参考电气特性章节规定的最高通信波特率。

25.4.1.2 数据格式

UART 模式时一帧数据是由开始位，数据位，校验位和停止位组成。

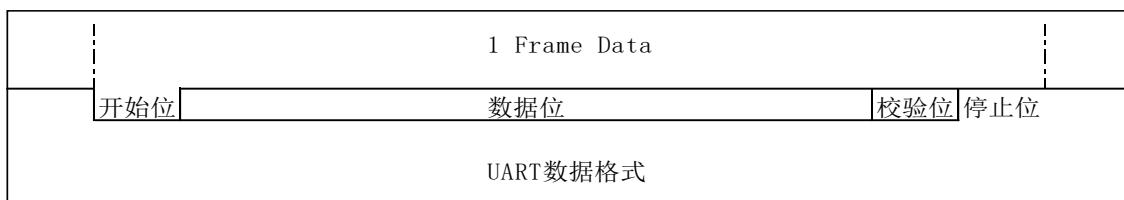


图 25-2 UART 数据格式

开始位

开始位固定有一位的低电平构成。

数据位

数据位可以配置成 8 位或者 9 位。

校验位

校验位可以配置成 1 位偶校验或 1 位奇校验或无校验位。

停止位

停止位固定为高电平，可以配置成 1 位或者 2 位。

25.4.1.3 调制解调器操作

调制解调器操作包括 CTS 功能和 RTS 功能。CTS 功能和 RTS 功能只能二选一，不能同时使用。USARTn_CR3.CTSE=0 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_DR.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

25.4.1.4 发送器

发送器可发送 8 位或 9 位的数据，具体取决于 USARTn_CR1.M 位的设定值。

发送器使能位 (USARTn_CR1.TE) 置 1，写入发送数据后，发送数据在 TX 管脚上串行输出；相应的时钟脉冲可以选择在 USARTn_CK 管脚输出或者不输出。

发送数据的顺序为：开始位->数据位(MSB/LSB)->校验位(有或者无)->停止位。

发送数据寄存器 TDR 和内部的发送移位寄存器组成双缓冲器结构，可以连续发送数据。

通过发送数据寄存器空中断或者 DMA 写入发送数据时，为保证发送的正确性，一次请求只能写入一次数据。

发送数据设定步骤

1. 将 USARTn_CR1 寄存器设定为复位值
2. 设定 UART 所需要使用的管脚
3. 通过 USARTn_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器
5. 设定 USARTn_PR 选择预分频值, USARTn_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. 使能发送器 (USARTn_CR1.TE=1)，如果需要使用发送数据寄存器空中断，则设置 USARTn_CR1.TXEIE=1 (TE 和 TXEIE 位同时写入 1)
7. 等待发送数据寄存器空，写通信数据到 USARTn_DR.TDR，数据传输到发送移位寄存器，发送开始
(CTS 功能有效时，USARTn_CTS 输入为低电平时数据传输到发送移位寄存器，发

送开始)

8. 如果需要连续发送数据时，重复步骤 7
9. 通过确认 USARTn_SR.TC 位确认发送是否完成。连续发送数据并使用发送中断的情况，可通过 TI 中断写入最后一个发送数据，并将 USARTn_CR1.TXEIE 写 0，USARTn_CR1.TCIE 写 1，最后一帧数据发送结束后，产生发送完成中断。

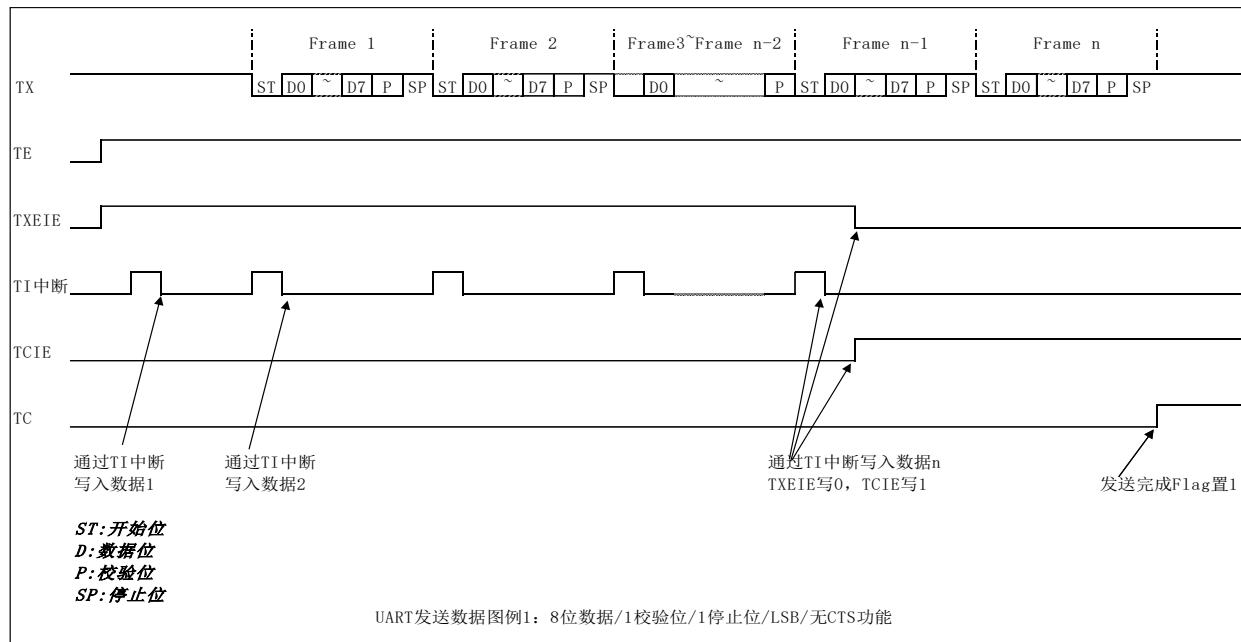


图 25-3 UART 发送数据图例 1

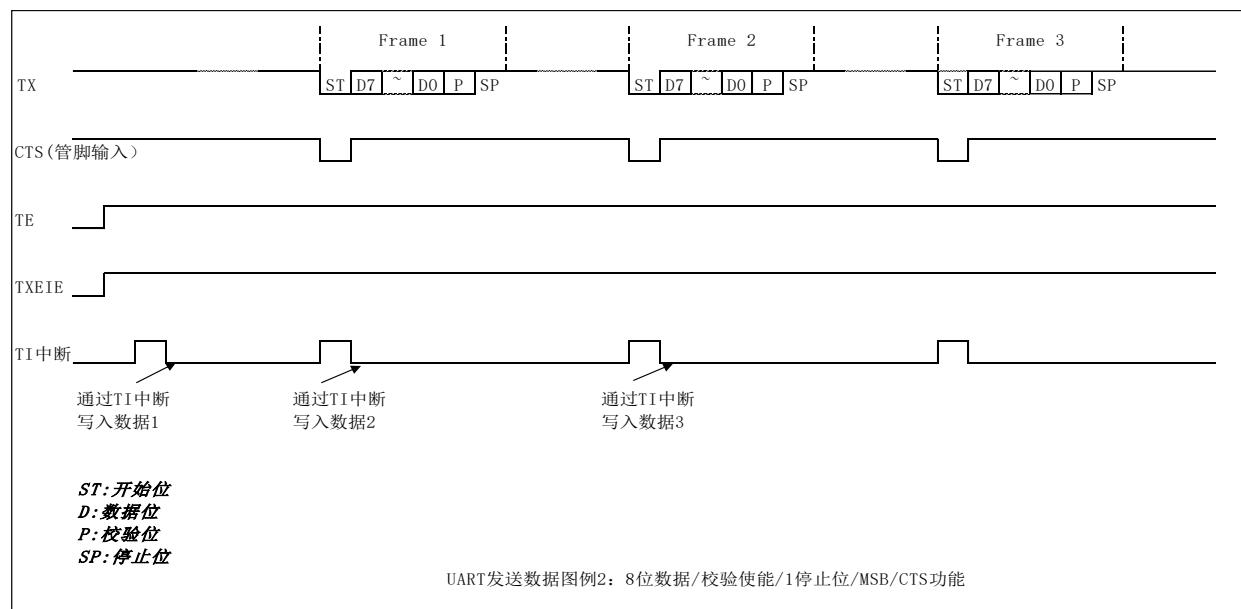


图 25-4 UART 发送数据图例 2

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_DR.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_DR.TDR 寄存器没有更新则 TCI 中断发生。

25.4.1.5 接收器

接收器可接收 8 位或 9 位的数据，具体取决于 USARTn_CR1.M 位的设定值。接收器使能位 (USARTn_CR.RE) 置 1 并检测到开始位后，RX 管脚上数据接收到接收移位寄存器，收满一帧数据，数据从接收移位寄存器传送到接收数据寄存器 USARTn_DR.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_DR.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。

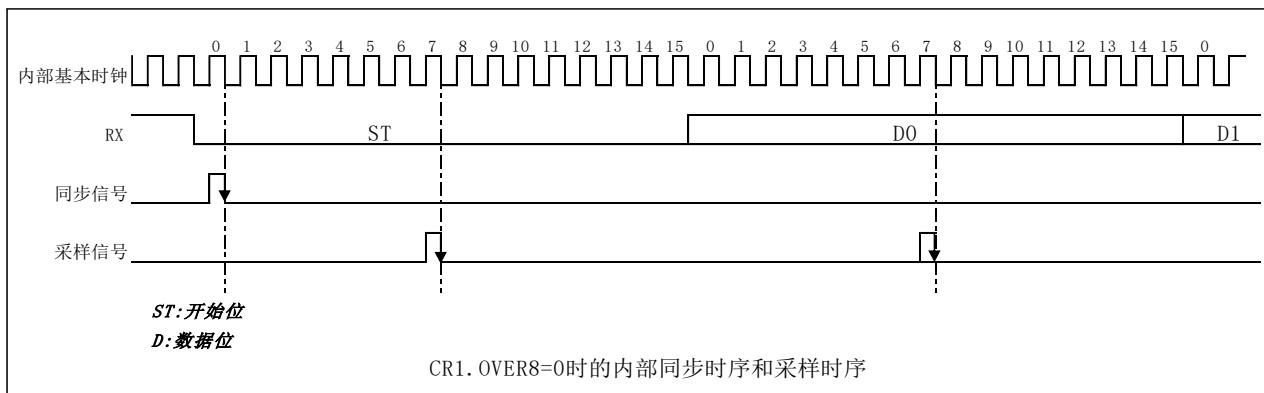


图 25-5 UART 内部同步和采样时序

仅当总时钟系统偏差小于 UART 接收器的容差时，UART 异步接收器才能正常工作。

影响总偏差的因素包括：

- 发送器误差引起的偏差（其中还包括发送器本地振荡器的偏差）
- 接收器的波特率量化引起的误差
- 接收器本地振荡器的偏差
- 传输线路引起的偏差

对于正确接收数据，UART 异步接收器所容许的最大偏差值具体取决于以下选项：

- 数据长度 FL。FL 由 USART_CR1 寄存器中 M 位定义的 8 或 9 数据位和 PCE 位定义的校验使能位决定
- 由 USART_CR1 寄存器中 OVER8 位定义的 8 倍或 16 倍过采样
- 由 USART_CR1 寄存器中 FBME 位定义的是否使用小数波特率

FL	OVER8位=0	OVER8位=1
10	4.375%	3.75%
11	3.97%	3.41%
12	3.646%	3.125%

表 25-2 DIV_Fraction 为 0 时 UART 接收器的容差

FL	OVER8位=0	OVER8位=1
10	3.88%	3%
11	3.53%	2.73%
12	3.23%	2.5%

表 25-3 DIV_Fraction 不为 0 时 UART 接收器的容差

在特殊情况下，当接收的帧包含 10 个位或 11 个位或 12 个位时间的空闲帧时，表 25-2 和表 25-3 中指定的数据可能会略微不同。

接收数据设定步骤

1. 将 USARTn_CR1 寄存器设定为复位值
2. 设定 UART 所需要使用的管脚

3. 通过 USARTn_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器
5. 设定 USARTn_PR 选择预分频值, USARTn_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. 使能接收器(USARTn_CR1.RE=1), 如果需要使用接收中断, 则设置 USARTn_CR1.RIE=1
7. 当检测到开始位后, 接收器将数据接收到接收移位寄存器, 并检查校验位和停止位
 - 1) 校验错误时, 接收到的数据传送到 USARTn_DR.RDR 寄存器中并置位 USARTn_SR.PE 标志
 - 2) 停止位不为高电平时, 发生帧错误, 接收到的数据传送到 USARTn_DR.RDR 寄存器中并置位 USARTn_SR.FE 标志
 - 3) 发生上溢错误时, 数据丢失并置位 USARTn_SR.ORE 标志
 - 4) 无错误发生时, 接收到的数据传送到 USARTn_DR.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

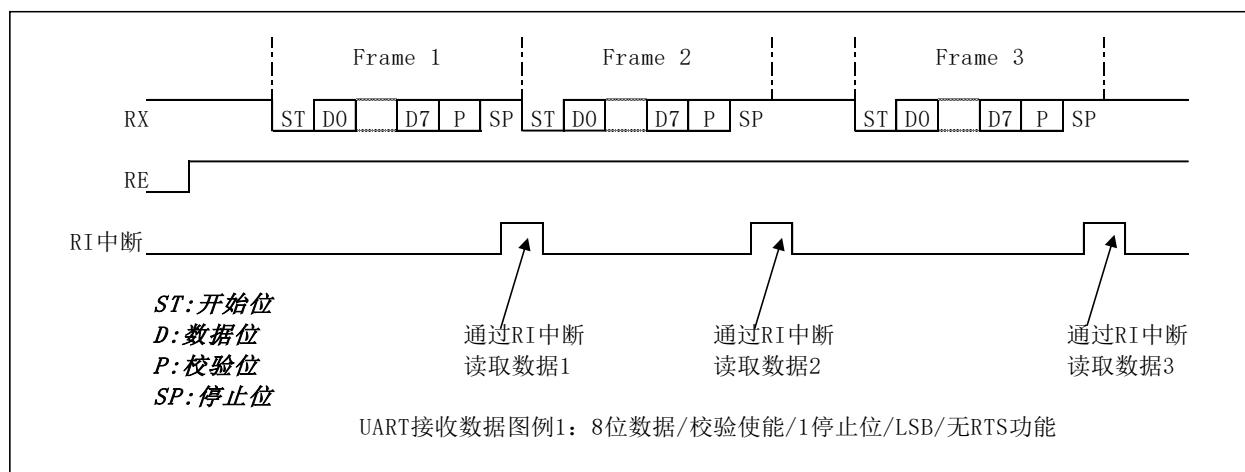


图 25-6 UART 接收数据图例 1

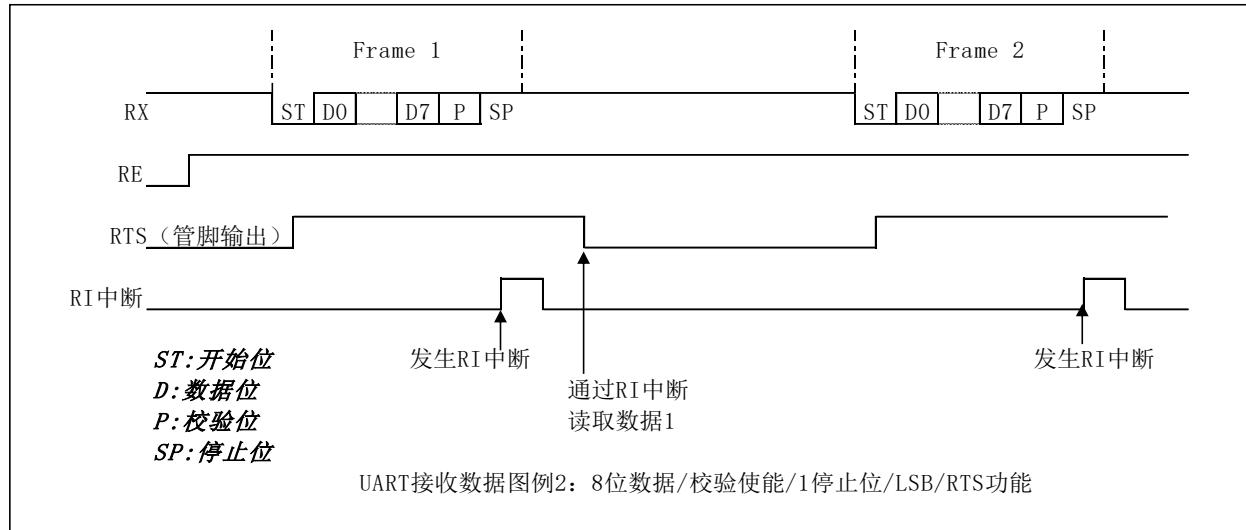


图 25-7 UART 接收数据图例 2

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_DR.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_DR.RDR，RI 中断不发生。

发生帧错误时接收到的数据传送给 USARTn_DR.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 REI。

USARTn_CR.RIE=1，未发生任何接收错误，数据从接收移位寄存器传送到接收数据寄存器时 RI 中断发生。

USARTn_CR.RIE=1，接收过程中发生上溢错误，校验错误或者帧错误时 REI 中断发生。

25.4.1.6 UART 接收 TIMEOUT 功能

UART 接收数据停止位被检测时 TIMEOUT 计数器启动，经过设定的 TIMEOUT 时间（设定单位为接收位数）后未检测到下一帧接收数据时，发生 TIMEOUT，如果此时 CR1.RE=1，则 TIMEOUT 状态位 USARTn_SR.RTOF 置位，如果此时 USARTn_CR1.RE=0，则等待 USARTn_CR1.RE=1 后 TIMEOUT 状态位 USARTn_SR.RTOF 置位。

TIMEOUT 计数器采用 Timer0 模块的计数器，具体对应关系如下：

USART1: Timer0 Unit1 A 通道

USART2: Timer0 Unit1 B 通道

USART3: Timer0 Unit2 A 通道

USART4: Timer0 Unit2 B 通道

TIMEOUT 功能 Timer0 比较计数器值设定

Timer0 为 16 位计数器，计数时钟最大可以选择 1024 分频，TMR0_CMPAR 值设定计算公式如下：

$$\text{CMPA } < \text{B} > \text{R} = \frac{\text{RTB}}{2^{\text{CKDIVA } < \text{B} >}} - 4 \text{ (计数时钟不分频)}$$

$$\text{CMPA } < \text{B} > \text{R} = \frac{\text{RTB}}{2^{\text{CKDIVA } < \text{B} >}} - 2 \text{ (计数时钟 2 分频)}$$

$$\text{CMPA } < \text{B} > \text{R} = \frac{\text{RTB}}{2^{\text{CKDIVA } < \text{B} >}} - 1 \text{ (计数时钟 4 分频及以上)}$$

CMPAR:TMR0_CMPAR 寄存器值

RTB: Receive Timeout Bits

CKDIRA: TMR0.BCONR.CKDIVA位寄存器值

TIMEOUT 功能设定步骤

1. 将 USARTn_CR1 寄存器设定为复位值
2. 设定 UART 所需要使用的管脚
3. 通过 USARTn_CR2.CLK[1:0] 位选择时钟源（如果选择内部时钟源时需设置 CR2.CLKC[0]=1）
4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器

5. 设定 USARTn_PR 选择预分频值, USARTn_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. 清除 USARTn_SR.RTOF 标志, 设置 USARTn_CR1.RTOE=1, 如果需要使用中断, 则设定 USARTn_CR1.RTOIE=1
7. 设置 TMR0.BCONR.CSTA=0
8. 设置 TMR0_CNTAR 为 0, 设置 TMR0_CMPAR 寄存器和 TMR0.BCONR.CKDIVA寄存器决定 TIMETOU 时间, 设置 TMR0.BCONR.HCLEA=1, TMR0.BCONR.HSTAA=1, TMR0.BCONR.ASYNCLKA=1, TMR0.BCONR.SYNSA=1
9. 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
10. 检测到 TIMEOUT 后设置 TMR0.BCONR.CSTA=0, 设置 TMR0.BCONR.CSTA=0, 通过写 USARTn_CR1.CRTOF 清除 USARTn_SR.RTOF 状态位。

25.4.1.7 RX 线唤醒停止模式功能

UART 通信空闲时, 可以让系统进入停止模式以节省电流消耗, 在不改变 UART PORT 设定的情况下, UART 单元 1 可以通过 RX 线来唤醒系统的停止模式。具体步骤如下:

1. UART 通信空闲时, 设定 USART_1_WUPI 中断向量以及 INT_WUPEN.RXWUEN 位使能 UART 接收信号线唤醒停止模式功能。
2. 系统进入停止模式。
3. 系统检测到 RX 线下降沿时, 从停止模式返回, 在 USART_1_WUPI 中断处理程序中关闭该功能。

需要注意的是, 当通信方需要唤醒本系统时, 需要发送一帧唤醒数据 (建议为 0x00), 该数据不会被 UART 接收且不置位相关的标志。并且通信方需要经过系统停止模式唤醒需要的时间后再进行 UART 通信数据的传输。

25.4.1.8 UART 中断和事件

功能名称	标记	使能位（仅中断）	标志位	可否作为事件源
接收错误中断	REI	RIE	ORE,FE,PE	可
接收数据寄存器满中断	RI	RIE	无	可
发送数据寄存器空中断	TI	TXEIE	无	可
发送完成中断	TCI	TCIE	TC	可
TIMEOUT中断	RTOI	RTOIE	RTOF	可
RX线唤醒停止模式中断	WUPI	INT_WUPEN.RXWUEN	-	不可

表 25-4 UART 中断/事件表

25.4.2 多处理器通信

25.4.2.1 功能简介

多处理器通信模式是指多个处理器间共用通信线的一种通信方式，处理器分为发送站和接收站，每个接收站都有自己固有的 ID。发送站发送数据的类型有接收站 ID 和通信数据两种。通过在数据格式中添加 MPB 位来区分当前发送的是接收站的 ID 还是通信数据。MPB 位为 0 时当前帧为通信数据，MPB 位为 1 时当前帧为接收站的 ID。所有接收站都能接收发送站发送的 ID 并与自己的 ID 比较，如果一致，则接收数据，不一致则进入静默模式(既不接收数据也不置位接收相关标志)直到再次接收到 ID。

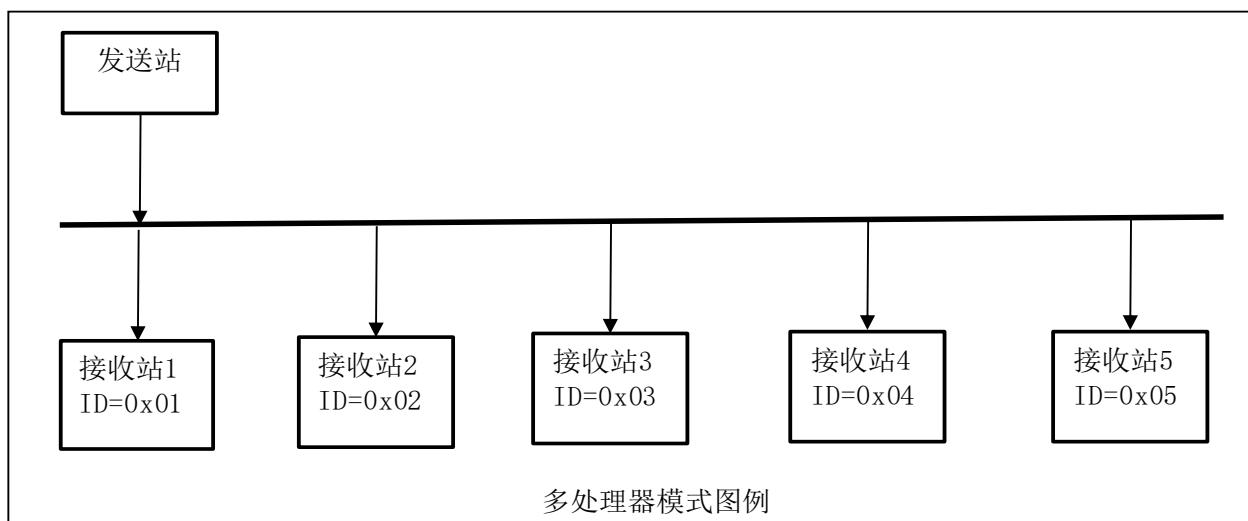


图 25-8 多处理器通信图例

25.4.2.2 数据格式

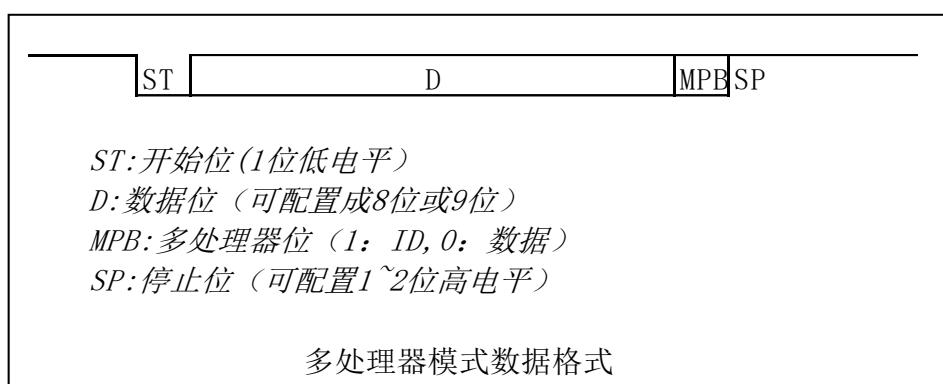


图 25-9 多处理器模式数据格式

25.4.2.3 动作说明

多处理器模式时校验位功能无效，增加了多处理器位功能，其余功能如时钟，中断等与 UART 模式相同。

发送站动作

1. 将 USARTn_CR1 寄存器设定为复位值
2. 设定所需要使用的管脚
3. 通过 USARTn_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器
5. 设定 USARTn_PR 选择预分频值, USARTn_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. 使能发送器 (USARTn_CR1.TE=1)，如果需要使用发送数据寄存器空中断，则设置 USARTn_CR1.TXEIE=1 (TE 和 TXEIE 位同时写入 1)
7. 等待发送数据寄存器空，设定 USARTn_DR.MPID 位为 1(发送 ID),写 ID 值到 USARTn_DR，发送 ID
(CTS 功能有效时，USARTn_CTS 输入为低电平时数据传输到发送移位寄存器，发送开始)
8. 设定 USARTn_DR.MPID 位为 0(发送数据)，写数据到 USARTn_DR，发送数据
(CTS 功能有效时，USARTn_CTS 输入低电平时数据传输到发送移位寄存器，发送开始)
9. 如果需要连续发送数据，重复步骤 8，如果需要改变 ID 后再发送数据重复 7 和 8。
10. 通过确认 USARTn_SR.TC 位确认发送是否完成。连续发送数据并使用发送中断的情况，可通过 TI 中断写入最后一个发送数据，并将 USARTn_CR1.TXEIE 写 0，USARTn_CR1.TCIE 写 1，最后一帧数据发送结束后，产生发送完成中断。

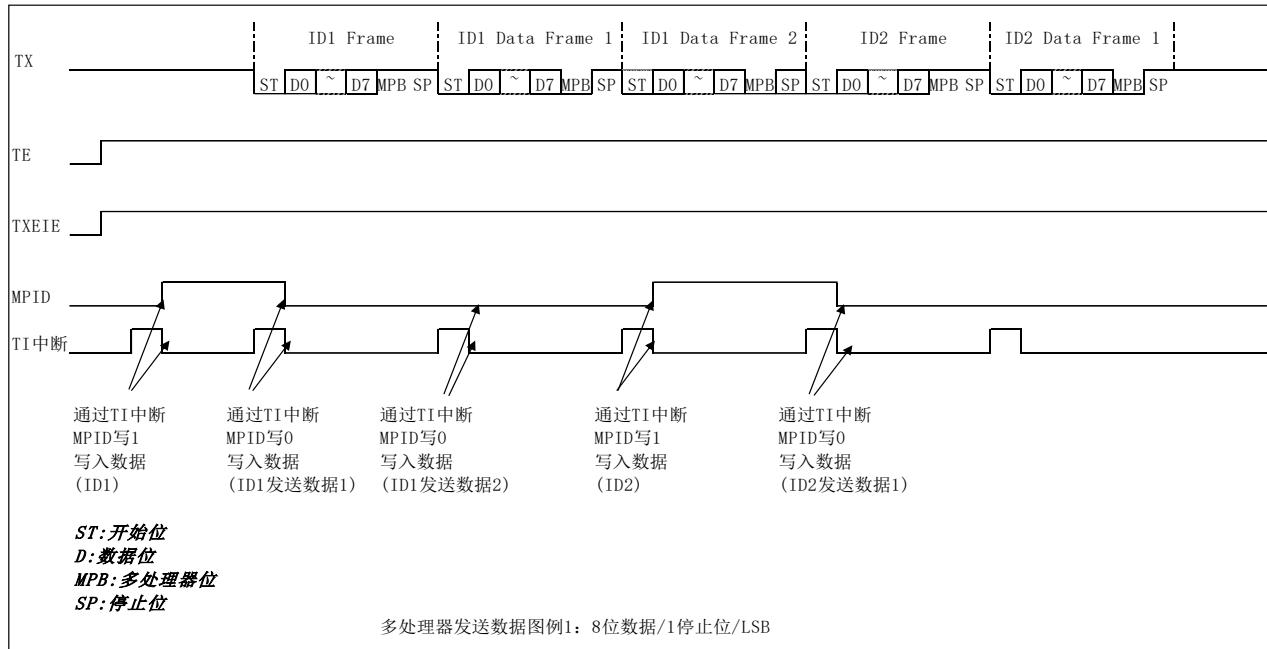


图 25-10 多处理器模式发送数据图例

接收站动作

在多处理器模式时，接收站必须保证能收到每一个 ID 数据，并与自身的 ID 比较，如果一致则接收数据，不一致则进入静默模式(不接收数据，也不置位接收相关标志)，直到接收到下一个 ID 数据。通过 USARTn_CR1.SLME 位来实现这个功能。

USARTn_CR1.SLME=0 时正常接收数据。

USARTn_CR1.SLME=1 时，除非接收到 MPB 位为 1 (ID) 的数据，否则不接收数据，不发生 RI 中断，错误标志 FE, ORE 也不置位。当接收到 MPB 位为 1 的数据时 (ID)，USARTn_CR1.SLME 位自动清零，正常接收数据和发生中断。

动作步骤：

1. 将 USARTn_CR1 寄存器设定为复位值
2. 设定所需要使用的管脚
3. 通过 USARTn_CR1.CLKC[1:0]位选择时钟源
4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器
5. 设定 USARTn_PR 选择预分频值, USARTn_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. USARTn_CR1.RE=1, USARTn_CR1.SLME=1(等待接收 ID)，如果使用接收中断，

则设定 USARTn_CR1.RIE=1

7. 当检测到开始位时，接收器将数据接收到接收移位寄存器，并检查 USARTn_SR.MPB 位
8. 如果 USARTn_SR.MPB=1，USARTn_CR1.SLME 位自动清零，正常接收数据，软件比较接收的 ID 与自身的 ID
 - 1) 如果 ID 一致，则正常接收数据，发生中断，进行错误检测，与 UART 接收数据相同
 - 2) 如果 ID 不一致，软件再次将 USARTn_CR1.SLME 位写 1，重复 8 的动作

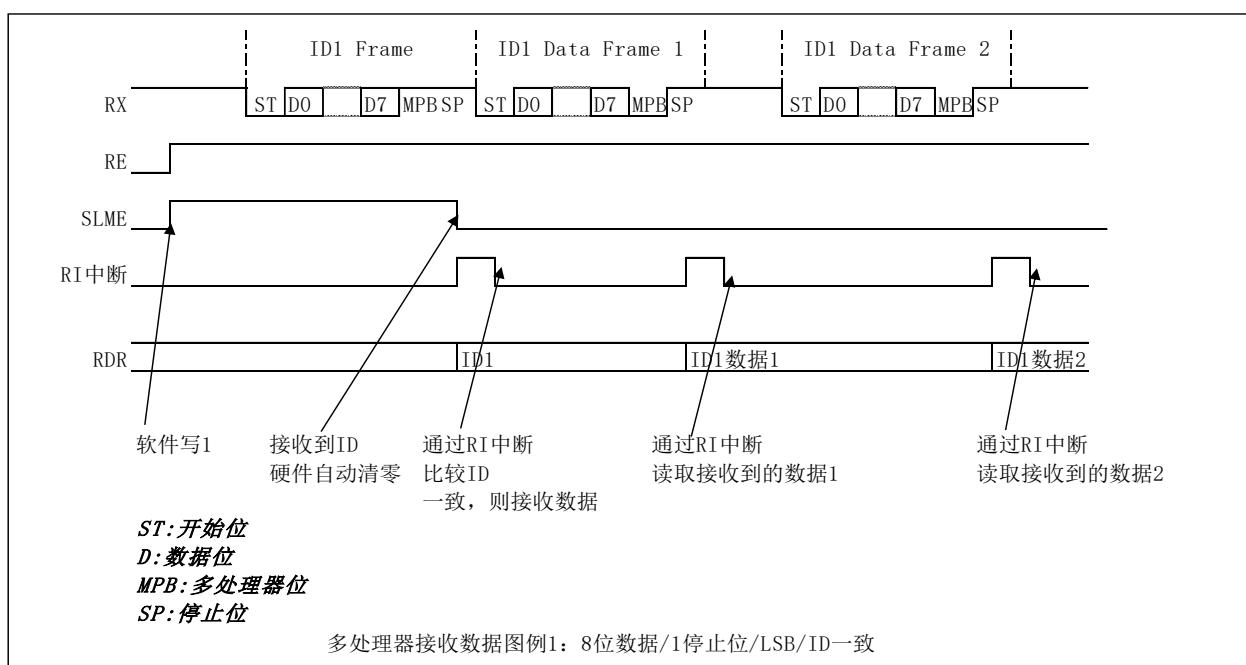


图 25-11 多处理器模式接收数据图例 1

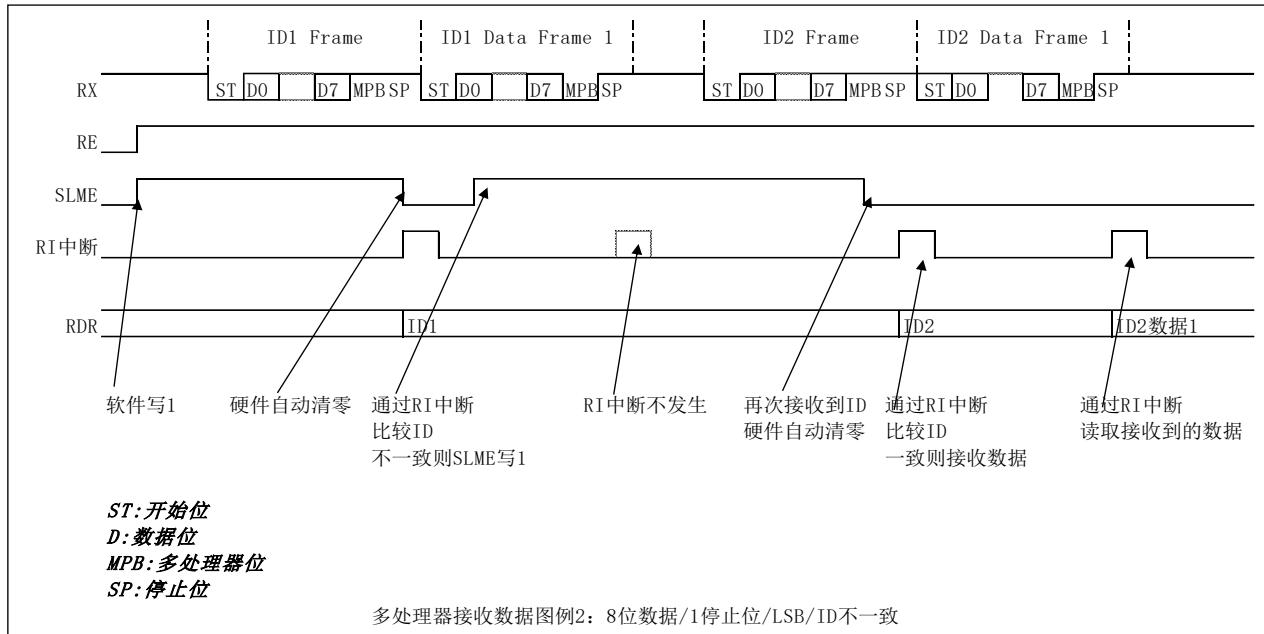


图 25-12 多处理器模式接收数据图例 2

25.4.2.4 中断和事件

多处理器模式除了无校验错误外，中断处理与 UART 模式相同。

功能名称	标记	使能位（仅中断）	标志	可否作为事件源
接收错误中断	REI	RIE	ORE,FE	不可
接收数据寄存器满中断	RI	RIE	RXNE	可
发送数据寄存器空中断	TI	TXEIE	TXE	可
发送完成中断	TCI	TCIE	TC	不可

表 25-5 多处理器模式中断/事件表

25.4.3 智能卡

25.4.3.1 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。

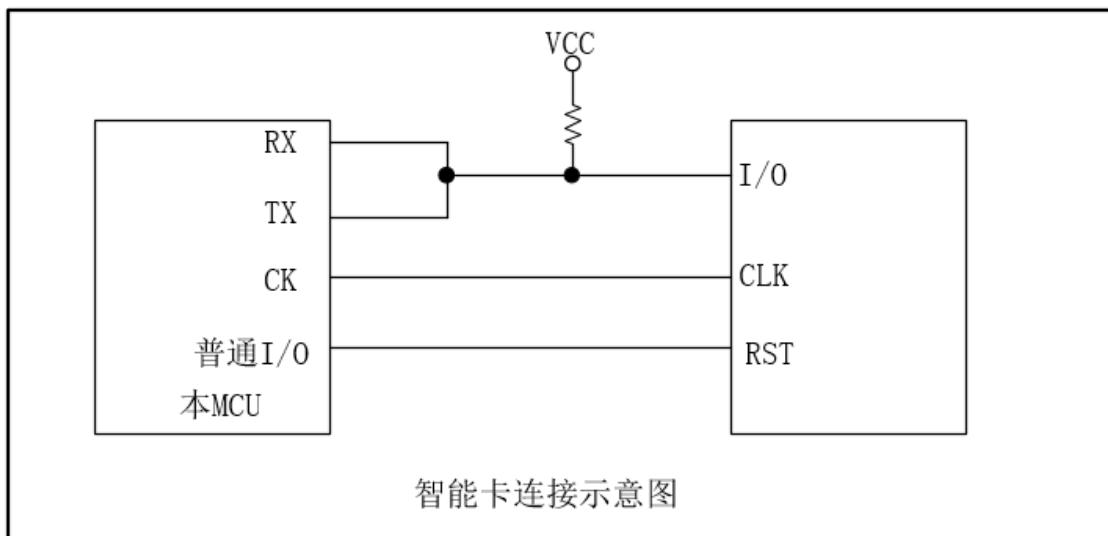


图 25-13 智能卡连接示意图

25.4.3.2 时钟

内部时钟源

智能卡模式时只能使用内部波特率生成器生成的时钟作为时钟源。

一位数据传输的基本时钟数为 USARTn_CR3.BCN[2:0]设定值。

通过设定寄存器 USARTn_CR2.CLKC[1:0]位控制智能卡模式的时钟输出。

最高波特率

内部时钟源时，内部波特率生成器生成的波特率计算公式为：

$$B = \frac{C}{2 \times BCN \times (DIV_Integer + 1)}$$

B: 波特率 单位: MBps

C: USARTn_PR.PSC[1:0]位设定的时钟 (PCLK,PCLK/4,PCLK/16,PCLK/64) 单位:
MHz

DIV_Integer: USARTn_BRR.DIV_Integer 设定值

BCN: USARTn_CR3.BCN 寄存器设定值

当 C 为 PCLK, DIV_Integer=0, BCN=0 时，波特率为最高波特率为 PCLK/64(MBps)。

采样和接收容差

检测到 RX 的下降沿后，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。接收数据将在数据中央被采样。

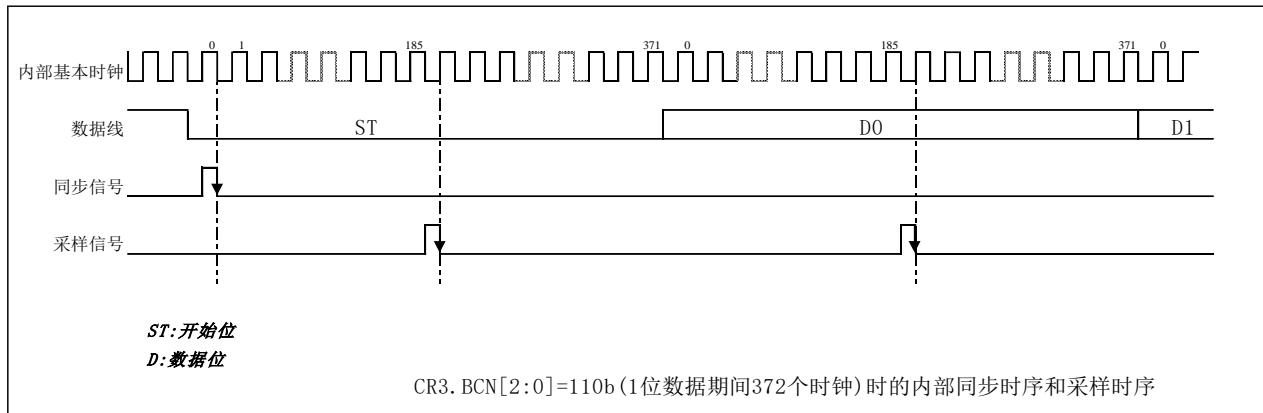


图 25-14 智能卡模式同步时序和采样时序图

接收容差的计算公式如下：

$$RM[\%] = |0.5 \times \left(1 - \frac{1}{BCN}\right) - 9.5CFD| \times 100$$

RM：接收容差

BCN：一位数据传输所需要的时钟数 (USARTn.CR3.BCN[2:0] 设定值)

CFD：时钟频率偏差

25.4.3.3 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。

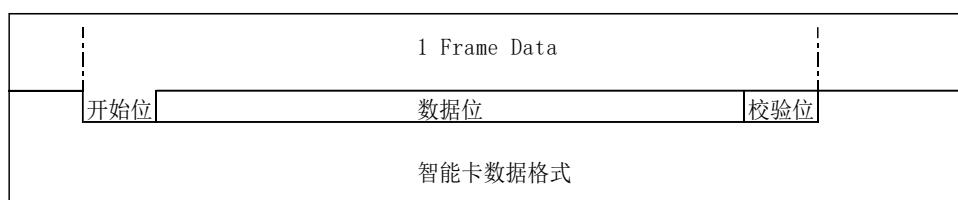


图 25-15 智能卡模式同步时序和采样时序图

起始位

起始位固定有一位低电平构成。

数据位

数据位固定为 8 位数据。

校验位

校验位需配置成 1 位偶校验。

25.4.3.4 智能卡的初始化设定步骤

1. 将 USARTn_CR1 寄存器设定为复位值
2. 设定所需要使用的管脚
3. 状态寄存器确认， USARTn_SR 寄存器设定为复位值
4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器
5. 设定 USARTn_PR 选择预分频值， USARTn_BRR 寄存器设定通信波特率
6. USARTn_CR2.CLKC[1:0]位设定时钟控制
7. USARTn_CR1 寄存器(TE,RE,RIE,TXEIE 位)设定，除了自测外， TE 和 RE 不要同时设定为 1

由发送模式切换到接收模式，或者有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。

25.4.3.5 智能卡模式动作说明

智能卡模式时， TI 中断(发送数据空中断)的标志位为 USARTn_SR.TC 位。

USARTn_SR.TC=1 而且 USARTn_CR1.TXEIE=1 时产生 TI 中断。

功能概述

发送数据时两帧数据间 (从校验位结束到下一帧的起始位开始)有 2etu(Elementary Time Unit)以上的保护时间。

发送数据时如果检测到接收方发送的错误信号则经过 2etu 后自动重发数据。

接收数据发生校验错误时，发送 1etu 的低电平即错误信号，错误信号送出的时序为接收开始经过 10.5etu。

发送说明

1. 一帧数据发送完成后，如果检测到接收方发送的错误信号，则 USARTn_SR.FE 置 1 (如果 USARTn_CR.RIE=1, 则发生错误中断)， USARTn_SR.TC 标志不置 1，数据自动重发。 USARTn_SR.FE 位必须在接受到下一帧校验位之前清零。
2. 一帧数据无错误发送完成后， USARTn_SR.TC 标志置位， USARTn_CR1.TXEIE=1 时发生 TI 中断。再次写入数据，则可实现连续发送数据。

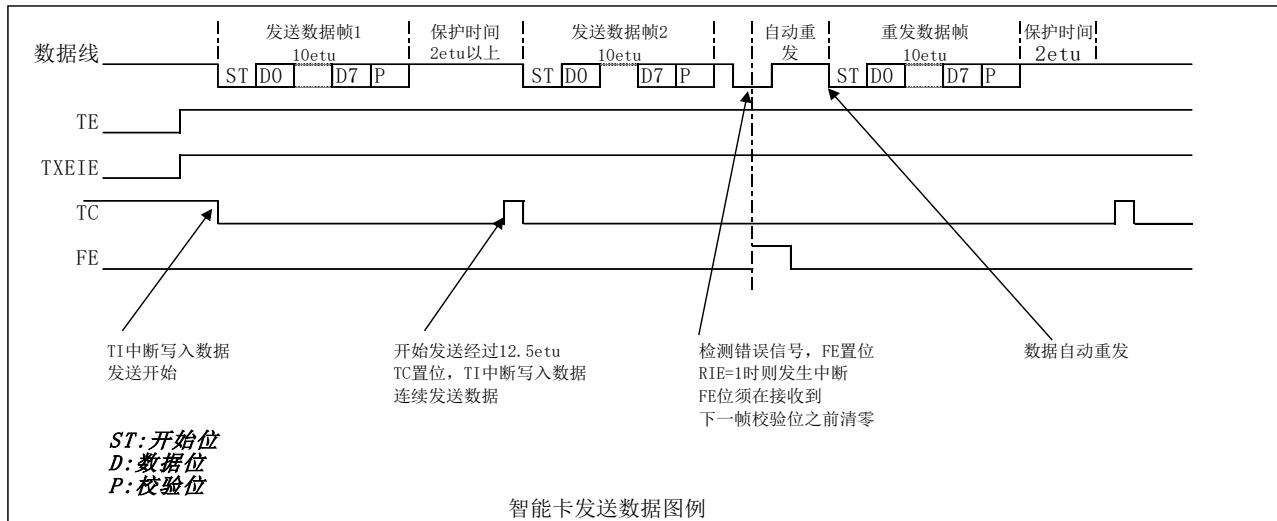


图 25-16 智能卡模式发送数据图例

接收说明

1. 接收数据时，如果检测到校验错误，则 USARTn_SR.PE 置位，中断使能时，发生 REI 中断。USARTn_SR.PE 位需要在接收到下一帧校验位之前清零。
2. 发生校验错误时，会发送 1etu 的低电平，即错误信号，要求发送方重新发送数据。
3. 正常接收数据，可以通过 RI 中断读取接收到的数据，连续接收。
4. 接收数据时，检测上溢错误。

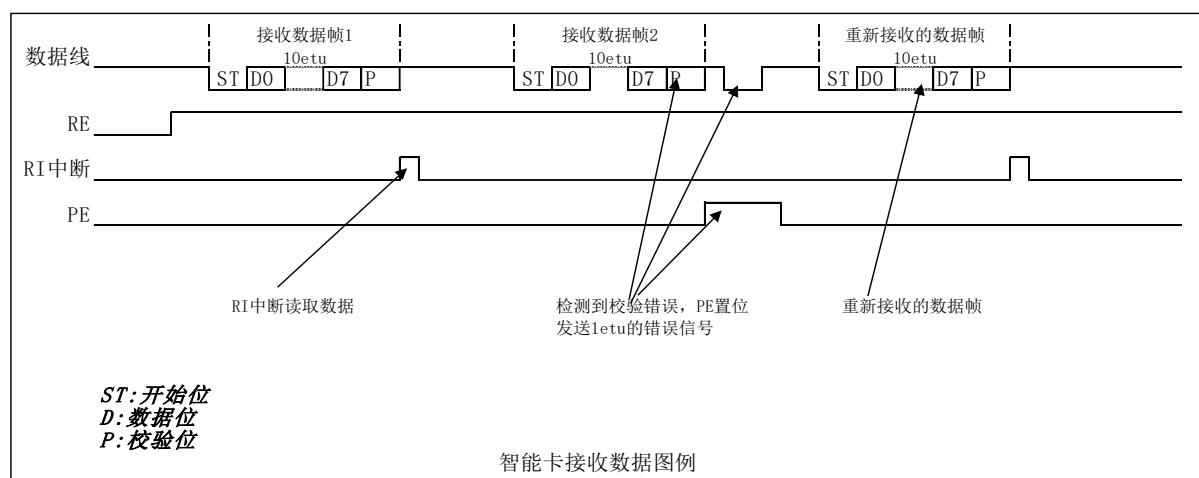


图 25-17 智能卡模式接收数据图例

25.4.3.6 中断和事件

功能名称	标记	使能位（仅中断）	标志位	可否作为事件源
错误中断	REI	RIE	ORE,PE,FE	可
接收数据满中断	RI	RIE	RXNE	可
发送数据空中断	TI	TXEIE	TC	可

表 25-6 智能卡模式中断/事件表

25.4.4 时钟同步模式

25.4.4.1 时钟

时钟同步模式可以选择内部波特率生成器生成的时钟(内部时钟源)或 USARTn_CK 管脚输入的时钟(外部时钟源)作为通信的时钟源。

内部时钟源

同步时钟从 USARTn_CK 管脚输出，一帧数据输出 8 个时钟脉冲，既不发送数据也不接收数据时，时钟输出固定为高电平。

外部时钟源

外部时钟源即从 USARTn_CK 管脚输入时钟作为通信时钟。

最高波特率

内部时钟源时，内部波特率生成器生成的波特率计算公式为：

$$B = \frac{C}{4 \times (\text{DIV_Integer} + 1)}$$

B: 波特率 单位: MBps

C: USARTn_PR.PSC[1:0]位设定的时钟(PCLK,PCLK/4,PCLK/16,PCLK/64) 单位: MHz

DIV_Integer:USARTn_BRR.DIV_Integer 设定值

内部时钟源时，当 C 为 PCLK, DIV_Integer=1 时，最高波特率为 PCLK/8(MBps)。注意同步模式时 DIV_Integer 禁止设置为 0。

外部时钟源时，外部输入时钟的最大频率要求为 PCLK(MHz)/6，所以最高波特率为 PCLK/6(MBps)。

需要注意的是，同步模式最高通信波特率除了以上描述的基于 PCLK 的计算方法外，还需要参考电气特性章节规定的最高通信波特率。

25.4.4.2 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。

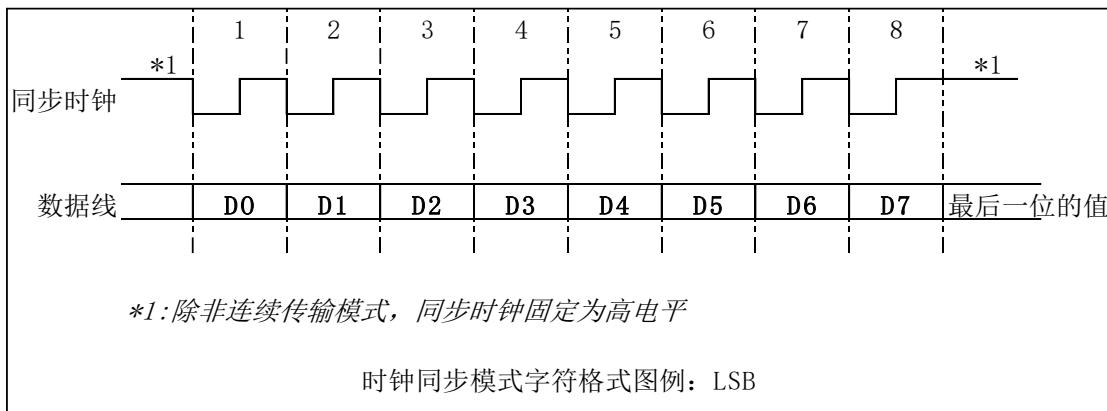


图 25-18 时钟同步模式数据格式

25.4.4.3 调制解调器操作

调制解调器操作包括 CTS 功能和 RTS 功能。CTS 功能和 RTS 功能只能二选一，不能同时使用。USARTn_CR3.CTSE=0 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_DR.RDR 寄存器中没有未读取的数据(USARTn_CR1.RE=1 时)
- USARTn_DR.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。

25.4.4.4 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_DR.TDR 和内部的发送移位寄存器组成双缓冲器结构，可以连续发送数据。

通过发送数据寄存器空中断或者 DMA 写入发送数据时，为保证发送的正确性，一次请求只能写入一次数据。

发送数据设定步骤

1. 将 USARTn_CR1, USARTn_SR1 寄存器设定为复位值
2. 设定所需要使用的管脚
3. 通过 USARTn_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器
5. 设定 USARTn_PR 选择预分频值, USARTn_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. 使能发送器 (USARTn_CR1. TE=1)，如果需要使用发送数据寄存器空中断，则设置 USARTn_CR1. TXEIE=1 (TE 和 TXEIE 位同时写入 1)
7. 等待发送数据寄存器空，写通信数据到 USARTn_DR.TDR，数据传输到发送移位寄存器，发送开始
(CTS 功能有效时，USARTn_CTS 输入为低电平时数据传输到发送移位寄存器，发送开始)
8. 如果需要连续发送数据时，重复步骤 7
9. 通过确认 USARTn_SR.TC 位确认发送是否完成。连续发送数据并使用发送中断的情况，可通过 TI 中断写入最后一个发送数据，并将 USARTn_CR1. TXEIE 写 0，USARTn_CR1. TCIE 写 1，最后一个数据发送结束后，产生发送完成中断。

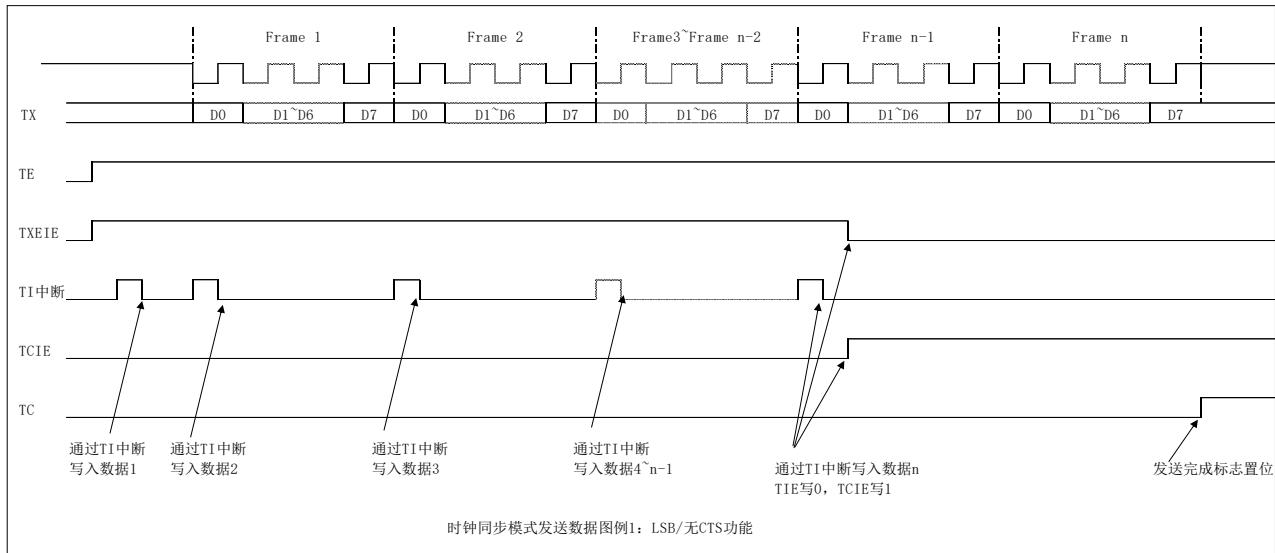


图 25-19 时钟同步模式发送数据图例 1

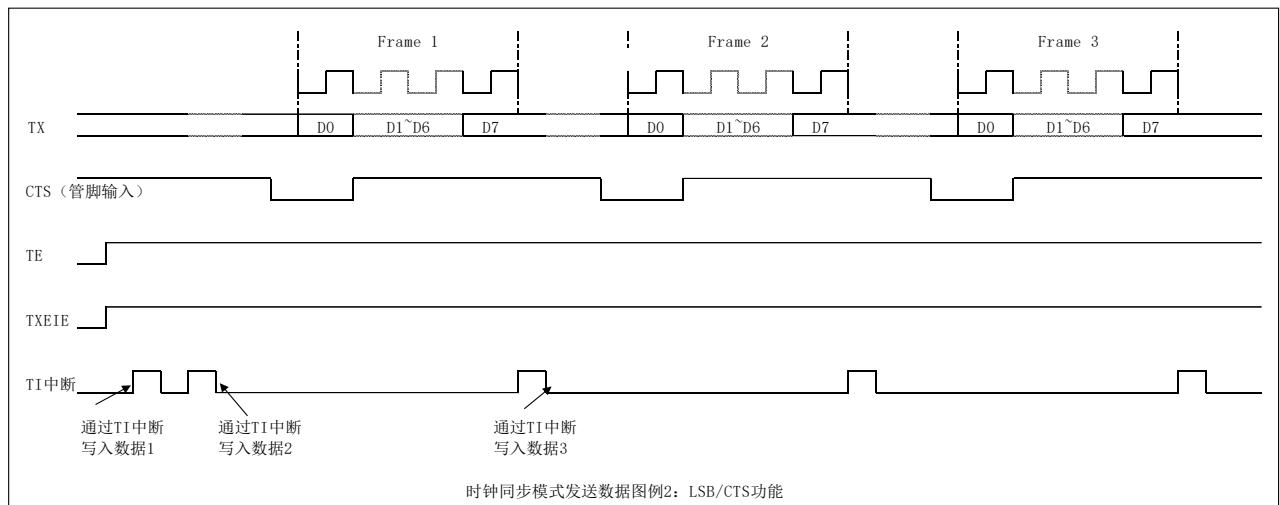


图 25-20 时钟同步模式发送数据图例 2

发送器中断

时钟同步模式发送器支持两种中断，发送数据寄存器空中断 TI 和发送完成中断 TCI。

TXEIE=1，USARTn_DR.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_DR.TDR 寄存器没有更新则 TCI 中断发生。

25.4.4.5 接收器

接收数据设定步骤

1. 将 USARTn_CR1, USARTn_SR 寄存器设定为复位值

2. 设定所需要使用的管脚
 3. 通过 USARTn_CR2.CLKC[1:0]位选择时钟源
 4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器
 5. 设定 USARTn_PR 选择预分频值, USARTn_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
 6. 使能接收器(USARTn_CR1.RE=1), 如果需要使用接收中断, 则设置 USARTn_CR1.RIE=1
(使用 RTS 功能时, RE=1 后 USARTn_RTS 输出低电平)
 7. 同步于输入同步时钟或者内部生成的同步时钟开始接收数据, 接收数据到接收移位寄存器。
 - 1) 发生上溢错误时, 数据丢失并置位 USARTn_SR.ORE 标志
 - 2) 无错误发生时, 接收到的数据传送到 USARTn_DR.RDR 寄存器中, 置位 USARTn_SR.RXNE 标志, 将当前接收到的数据在接收到下一帧数据最后一位前读取后重复步骤 7 可实现连续接收数据功能。
- (使用 RTS 功能时, 数据读取后 USARTn_RTS 输出低电平)

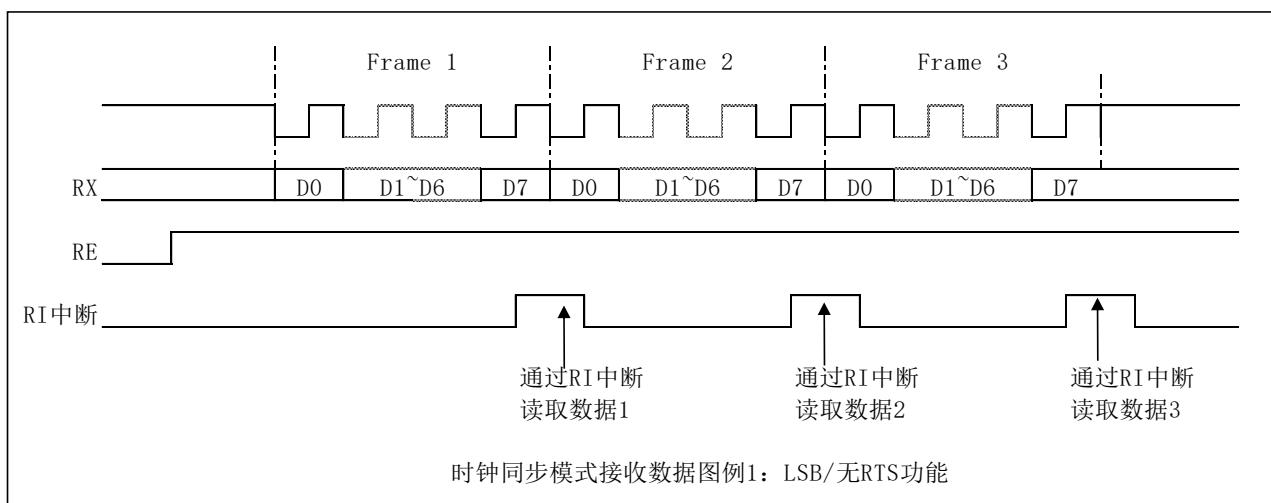


图 25-21 时钟同步模式接收数据图例 1

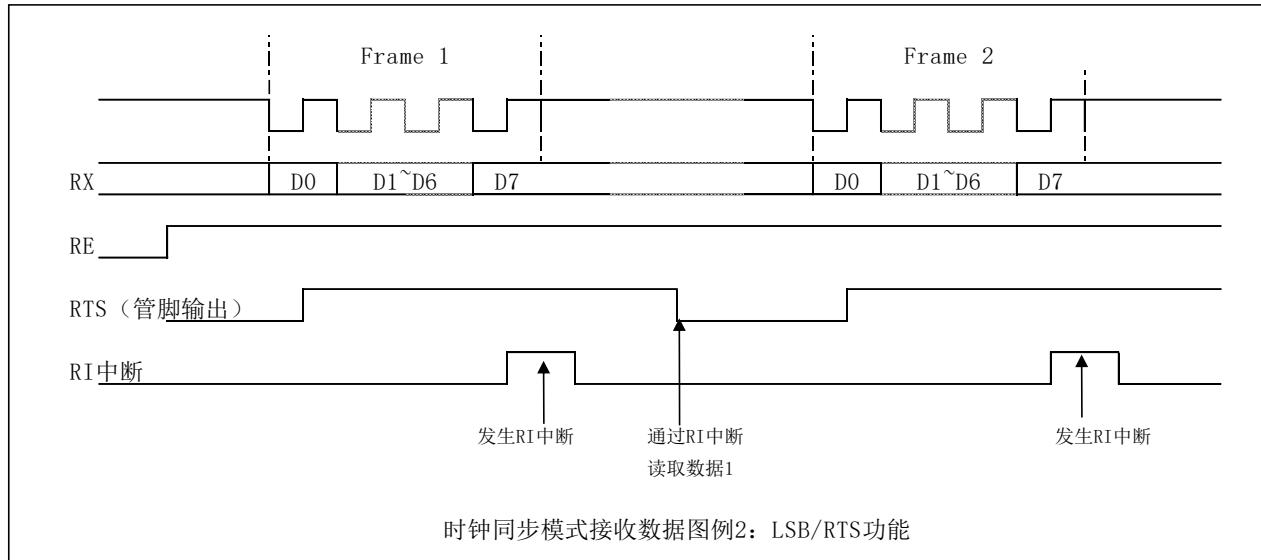


图 25-22 时钟同步模式接收数据图例 2

错误处理

时钟同步模式接收数据时接收错误为上溢错误（USARTn_SR.ORE）。发生接收错误不能再进行数据的接收和发送。可以通过将错误标志清零来重启数据传输，清除方法是写对应的清零寄存器。

上溢错误发生的条件是 USARTn_DR.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 REI。

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 REI 中断发生。

25.4.4.6 同时发送接收数据

USART 时钟同步模式支持全双工动作，同时发送接收数据。同时发送接收数据时需要一个命令把 RE,TE,RIE,TXEIE 写 1，其它设定流程与发送器和接收器相同。

25.4.4.7 时钟同步模式中断和事件

中断名称	标记	使能位(仅中断)	标志位	可否作事件源
接收错误中断	REI	RIE	ORE	不可
接收数据寄存器满中断	RI	RIE	RXNE	可
发送数据寄存器空中断	TI	TXEIE	TXE	可
发送完成中断	TCI	TCIE	TC	不可

表 25-7 时钟同步模式中断/事件表

25.4.5 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。

25.5 寄存器说明

USART1_BASE_ADDR:0x4001_D000

USART2_BASE_ADDR:0x4001_D400

USART3_BASE_ADDR:0x4002_1000

USART4_BASE_ADDR:0x4002_1400

寄存器名	偏移地址	复位值
状态寄存器 (USART_SR)	0x00	0x0000_00C0
数据寄存器 (USART_DR)	0x04	0x0000_01FF
波特率寄存器 (USART_BRR)	0x08	0x0000_FFFF
控制寄存器1 (USART_CR1)	0x0C	0x8000_0000
控制寄存器2 (USART_CR2)	0x10	0x0000_0000
控制寄存器3 (USART_CR3)	0x14	0x0000_0000
预分频寄存器 (USART_PR)	0x18	0x0000_0000

表 25-8 USART 寄存器一览表

25.5.1 状态寄存器 (USART_SR)

USART Status Register

偏移地址: 0x00

复位值: 0x0000_00C0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	MPB
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	RTO F	TXE	TC	RXN E	-	ORE	-	FE	PE

位	标记	位名	功能	读写
b31~b17	Reserved	-	读出时为“0”,写入时写“0”	R
b16	MPB	多处理器位	多处理器位标志 0: 当前接收数据为通信数据 1: 当前接收数据为ID 注意: MPB位只在多处理器模式时有效	R
b15~b9	Reserved	-	读出时为“0”,写入时写“0”	R
b8	RTOF	UART接收TIMEOUT 标志位	UART接收TIMEOUT标志位 0: 无UART接收TIMEOUT 1: 发生UART接收TIMEOUT RTOF置位条件 ● 从检测到最后一帧数据的STOP位起经过设定时间后没有检测到新的接收数据 RTOF清零条件 ● 清零寄存器CR1.RTOF位写入 注意: RTOF由硬件置1,且只有在CR1.RE=1且CR1.RTOE=1时置1。CR1.RE=0时, TIMEOUT功能有效,但RTOF不置1。 RTOF仅在USART_CR1.RTOE=1时读取值才为有效值,其它情况请忽略该位, 使用 UART接收Timeout功能时,设置RTOE=1之前,需软件清除该位。	R
b7	TXE	发送数据寄存器空	发送数据寄存器空标志 TXE位在UART和时钟同步模式时有效。 0: 数据未传输到移位寄存器,发送数据寄存器非空 1: 数据传输到移位寄存器,发送数据寄存器空 注意: TXE位由硬件置1和清0,数据未传输到移位寄存器时硬件将TXE清0,数据传 输到移位寄存器时硬件将TXE置1	R

位	标记	位名	功能	读写
b6	TC	发送完成标志	<p>发送完成标志位</p> <p>0: 发送数据中 1: 发送数据完成</p> <p>UART模式, 时钟同步模式</p> <p>TC置位条件</p> <ul style="list-style-type: none"> ● TE=0发送禁止时 ● 送出一帧数据的最后一位时, 发送数据寄存器的值没有被更新 <p>TC清零条件</p> <ul style="list-style-type: none"> ● TE=1时, 向发送数据寄存器写入发送数据 <p>智能卡模式</p> <p>TC置位条件</p> <ul style="list-style-type: none"> ● TE=0发送禁止时 ● 最后1字节的数据送出后经过特定时间后, FE=0而且发送数据寄存器的值没有被更新。 <p>TC置位的具体时序为: 校验位送出后经过2.5位时间</p> <p>TC清零条件</p> <ul style="list-style-type: none"> ● TE=1时, 向发送数据寄存器写入发送数据 <p>注意: TE位由0变化为1时TC为保持为1</p>	R
b5	RXNE	接收数据寄存器不为空	<p>接收数据寄存器不为空标志</p> <p>0: 未接收到数据 1: 准备好读取接收到的数据</p> <p>注意: RXNE位由硬件置1和清0, 准备好读取接收到的数据时硬件将RXNE置1, 读取接收数据后硬件将RXNE清0</p>	R
b4	Reserved	-	读出时为“0”, 写入时写“0”	R
b3	ORE	接收上溢错误	<p>接收上溢错误标志位</p> <p>0: 无接收上溢错误 1: 发生接收上溢错误</p> <p>ORE置位条件</p> <ul style="list-style-type: none"> ● 接收数据寄存器未被读取的情况下, 又接收到一帧新的数据 <p>ORE清零条件</p> <ul style="list-style-type: none"> ● 清零寄存器CR1.CORE位写入1 <p>注意: RE=0并不能复位ORE位</p> <p>ORE=1之前接收到的数据会被保持, ORE=1时接收到的数据会丢掉</p> <p>ORE=1后不能继续接收数据, 时钟同步模式下也不能发送数据</p>	R
b2	Reserved	-	读出时为“0”, 写入时写“0”	R

位	标记	位名	功能	读写
b1	FE	接收帧错误	<p>接收帧错误标志位</p> <p>0: 无接收帧错误 1: 发生接收帧错误</p> <p>UART模式</p> <p>FE置位条件</p> <ul style="list-style-type: none"> 接收的数据帧的停止位为低电平, 两个停止位的情况只检查第一个停止位 <p>FE清零条件</p> <ul style="list-style-type: none"> 清零寄存器CR1.CFE位写入1 <p>注意: UART模式时, RE=0并不能复位PE位</p> <p>FE=1时收到的数据会保留但是RI中断不会发生, FE=1后不能继续接收数据</p>	R
b0	PE	接收数据校验	<p>智能卡模式</p> <p>FE置位条件</p> <ul style="list-style-type: none"> 采样到低电平错误信号标志 <p>FE清零条件</p> <ul style="list-style-type: none"> 清零寄存器CLR.CFE位写入1 <p>注意: 智能卡模式时, RE=0并不能复位FE位</p>	R
			<p>接收数据校验错误标志</p> <p>0: 无接收数据校验错误 1: 发生接收数据校验错误</p> <p>PE置位条件</p> <ul style="list-style-type: none"> 接收数据发生奇偶校验错误时 <p>PE清零条件</p> <ul style="list-style-type: none"> 清零寄存器CR1.CPE位写入1 <p>注意: RE=0并不能复位PE位</p> <p>PE=1时收到的数据会保留但是RI中断不会发生, PE=1后不能继续接收数据</p>	

25.5.2 数据寄存器（USART_DR）

USART Data Register

偏移地址：0x04

复位值：0x0000_01FF

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

位	标记	位名	功能	读写
b31~b25	Reserved	-	读出时为“0”，写入时写“0”	R
b24~b16	RDR[8:0]	接收数据寄存器	接收数据寄存器 注意：最高位RDR[8]仅在UART模式而且数据长度设定为9位时有效	R
b15~b10	Reserved	-	读出时为“0”，写入时写“0”	R
b9	MPIID	多处理器模式ID位	多处理器模式时，发送通信数据或者发送ID的选择位 0：发送数据 1：发送ID 注意：MPIID位只在多处理器模式时有效，其他模式须设定为复位值	R/W
b8~b0	TDR[8:0]	发送数据寄存器	发送数据寄存器 注意：最高位TDR[8]仅在UART模式而且数据长度设定为9位时有效	R/W

25.5.3 波特率寄存器 (USART_BRR)

USART Bit Rate Register

偏移地址: 0x08

复位值: 0x0000_FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DIV_Integer[7:0]										-	DIV_Fraction[6:0]				

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R
b15~b8	DIV_Integer[7:0]	整数分频寄存器	整数分频寄存器	R/W
		存器	注意: DIV_Integer[7:0]只能在TE=0&RE=0(发送/接收禁止)时设定	
b7	Reserved	-	读出时为“0”,写入时写“0”	R
b6~b0	DIV_Fraction[6:0]	小数分频寄存器	小数分频寄存器	R/W
		存器	注意: DIV_Fraction[6:0]只能在TE=0&RE=0(发送/接收禁止)时设定,而且仅在FBME=1时设定值有效	

模式	波特率计算公式	误差E(%)计算公式
UART模式 多处理器模式	$B = \frac{C}{8 \times (2 - OVER8) \times (DIV_Integer + 1)}$	$E(%) = \left\{ \frac{C}{8 \times (2 - OVER8) \times (DIV_Integer + 1) \times B} - 1 \right\} \times 100$
时钟同步模式	$B = \frac{C}{4 \times (DIV_Integer + 1)}$	-
智能卡模式	$B = \frac{C}{2 \times BCN \times (DIV_Integer + 1)}$	$E(%) = \left\{ \frac{C}{2 \times BCN \times (DIV_Integer + 1) \times B} - 1 \right\} \times 100$

B: 波特率 单位: Mbps

C: PR.PSC[1:0]位设定的时钟 单位: MHz

BCN: CR3.BCN 寄存器设定值

表 25-9 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	$B = \frac{C \times (128 + DIV_Fraction)}{8 \times (2 - OVER8) \times (DIV_Integer + 1) \times 256}$	$E(%) = \left\{ \frac{C \times (128 + DIV_Fraction)}{(8 \times (2 - OVER8) \times (DIV_Integer + 1) \times 256 \times B} - 1 \right\} \times 100$
时钟同步模式	$B = \frac{C \times (128 + DIV_Fraction)}{4 \times (DIV_Integer + 1) \times 256}$	
智能卡模式	$B = \frac{C \times (128 + DIV_Fraction)}{2 \times BCN \times (DIV_Integer + 1) \times 256}$	$E(%) = \left\{ \frac{C \times (128 + DIV_Fraction)}{2 \times BCN \times (DIV_Integer + 1) \times 256 \times B} - 1 \right\} \times 100$

B: 波特率 单位: Mbps

C: PR.PSC[1:0]位设定的时钟 单位: MHz

BCN: CR3.BCN 寄存器设定值

表 25-10 波特率计算公式 (小数波特率有效 FBME=1)

25.5.4 控制寄存器 1 (USART_CR1)

USART Control Register 1

偏移地址: 0x0C

复位值: 0x8000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SBS	NFE	FBM E	ML	-	-	-	MS	-	-	-	CRTO F	COR E	-	CFE	CPE
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OVER 8	-	-	M	-	PCE	PS	-	TXEI E	TCIE	RIE	SLM E	TE	RE	RTOI E	RTO E

位	标记	位名	功能	读写
b31	SBS	UART模式接收数 据开始位检测方 式设定位	UART模式接收数据时, 开始位检测方式设定位 0: 开始位检测方式为RX管脚低电平 1: 开始位检测方式为RX管脚下降沿 注意: 非UART模式时SBS位须保持复位值 SBS位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b30	NFE	数字滤波使能位	数字滤波使能位 0: 禁止数字滤波功能 1: 使能数据滤波功能 注意: 非UART模时NFE位必须保持复位值 NFE位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b29	FBME	小数波特率功能 使能	小数波特率功能使能位 0: 禁止 1: 使能 注意: FBME位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b28	ML	MSB/LSB选择位	UART模式/时钟同步模式/智能卡模式时, MSB/LSB方式选择位 0: LSB方式 1: MSB方式 注意: ML位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b27~b25	Reserved	-	读出时为“0”, 写入时写“0”	R
b24	MS	通信模式选择位	通信模式选择位 0: UART模式 1: 时钟同步模式 注意: MS位只能在TE=0&RE=0(发送/接收禁止)时设定, 智能卡模式MS需写入复位值	R/W
b23~b21	Reserved	-	读出时为“0”, 写入时写“0”	R

位	标记	位名	功能	读写
b20	CRTOF	RTOF清零位	RTOF清零位 0:不清零RTOF标志 1:清零RTOF标志 注意: CRTOF位写1清除CRTOF标志, 读取时返回0	R/W
b19	CORE	ORE标志清零位	ORE标志清零位 0: 不清零ORE标志 1: 清零ORE标志 注意: CORE位写1清除ORE标志, 读取时返回0	R/W
b18	Reserved	-	读出时为“0”, 写入时写“0”	R
b17	CFE	FE标志清零位	FE标志清零位 0:不清零FE标志 1:清零FE标志 注意: CFE位写1清除FE标志, 读取时返回0	R/W
b16	CPE	PE标志清零位	PE标志清零位 0:不清零PE标志 1:清零PE标志 注意: CPE位写1清除FE标志, 读取时返回0	R/W
b15	OVER8	UART过采样模式	UART过采样模式设定, 即一位数据传输期间的基本时钟数 0: 16位 1: 8位 注意: 非UART模式时OVER8位必须保持复位值 OVER8位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b14~b13	Reserved	-	读出时为“0”, 写入时写“0”	R/W
b12	M	数据长度设定位	UART模式时, 发送/接收数据长度设定位 0: 8位 1: 9位 注意: 非UART模式时M位必须保持复位值 M位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b11	Reserved	-	读出时为“0”, 写入时写“0”	R
b10	PCE	校验使能位	UART模式时, 奇偶校验使能位 0: 无校验 1: 校验 注意: 智能卡模式时PCE位必须为1, 时钟同步模式时PCE位必须保持复位值 PCE位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b9	PS	校验位	UART模式时, 奇偶校验选择位 0: 偶校验 1: 奇校验 注意: PS位只能在TE=0&RE=0(发送/接收禁止)时设定, PS位只在PCE=1时有效	R/W
b8	Reserved	-	读出时为“0”, 写入时写“0”	R

位	标记	位名	功能	读写
b7	TXEIE	发送数据寄存器 空中断使能位	发送数据寄存器空中断使能位 0: TI中断请求无效, TI中断不发生 1: TI中断请求有效, TI中断发生 注意: TXEIE位和TE位应同时写入1	R/W
b6	TCIE	发送完成中断使能位	发送完成中求使能位 0: TCI中断请求无效, TCI中断不发生 1: TCI中断请求使能, TCI中断发生	R/W
b5	RIE	接收中断使能位	接收中断使能位 0: 接收中断请求无效, RI和REI中断不发生 1: 接收中断请求有效, RI和REI中断发生	R/W
b4	SLME	静默模式使能位	多处理器操作时, 静默模式使能位 0: 正常模式 1: 静默模式 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。 注意: SLME位只在UART多处理器模式时有效, 其他模式时此位必须保持复位值。	R/W
b3	TE	发送器使能位	发送器使能位 0: 发送器禁止 1: 发送器使能 注意: 时钟同步模式时TE位只能在TE=0&RE=0(发送/接收禁止)时写1。	R/W
b2	RE	接收器使能位	接收器使能位 0: 接收器禁止 1: 接收器使能 注意: 时钟同步模式时RE位只能在TE=0&RE=0(发送/接收禁止)时写1	R/W
b1	RTOIE	UART TIMEOUT 中断使能位	UART TIMEOUT中断使能位 0: UART TIMEOUT中断请求无效, RTOI中断不发生 1: UART TIMEOUT中断请求有效, RTOI中断发生 注意: RTOIE需要和RTOE同时设定为1, 或者在RTOE=1后设定为1	R/W
b0	RTOE	UART TIMEOUT 功能使能位	UART TIMEOUT功能使能位 0: UART TIMEOUT功能禁止 1: UART TIMEOUT功能使能 注意: RTOE=1设定为1前, 需要清除USARTn.SR.RTOF标志, 并停止对应通道的Timer0计数功能	R/W

25.5.5 控制寄存器 2 (USART_CR2)

USART Control Register 2

偏移地址: 0x10

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b14	Reserved	-	读出时为“0”，写入时写“0”	R
b13	STOP	停止位设定位	UART模式时，停止位长度设定位 0: 1停止位 1: 2停止位 注意：非UART模式时STOP位必须保持复位值 STOP位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b12~b11	CLKC[1:0]	时钟控制位	UART模式 00b: 时钟源为内部波特率生成器生成的时钟，时钟不输出到USARTn_CK管脚，USARTn_CK管脚可以当作普通IO使用 01b: 时钟源为内部波特率生成器生成的时钟，时钟输出到USARTn_CK管脚，输出时钟频率和波特率相同 10b or 11b: 时钟源为外部输入时钟，输入时钟的频率为波特率的16倍(OVER8=0)或者8倍(OVER8=1) 时钟同步模式 00b or 01b: 时钟源为内部波特率生成器生成的时钟，输出到USARTn_CK管脚 10b or 11b: 时钟源为外部输入时钟，输入时钟的频率和波特率相同 智能卡模式 00b: 时钟源为内部波特率生成器生成的时钟，时钟不输出到CK管脚，CK管脚可以当作普通IO使用 01b: 时钟源为内部波特率生成器生成的时钟，时钟输出到CK管脚 10b or 11b: 设定禁止 注意：CLKC[1:0]位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b10~b1	Reserved	-	读出时为“0”，写入时写“0”	R
b0	MPE	多处理器功能使能位	UART模式时，多处理器功能使能位 0: 禁止 1: 使能 注意：非UART模式时MPE位必须保持复位值 MP位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W

25.5.6 控制寄存器 3 (USART_CR3)

USART Control Register 3

偏移地址: 0x14

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	BCN[2:0]	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b15	b14
-	-	-	-	-	-	CTS E	-	-	-	SCE N	-	-	-	-	-

位	标记	位名	功能	读写
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R
b23~b21	BCN[2:0]	基本时钟数	智能卡模式时, 设定一位数据传输期间的基本时钟数 BCN[2:0]设定值 一位数据传输期间的基本时钟数 000b 32 001b 64 010b 设定禁止 011b 128 100b 设定禁止 101b 256 110b 372 111b 设定禁止 注意: 非智能卡模式时BCN[2:0]位必须保持复位值 BCN[2:0]位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b20~b10	Reserved	-	读出时为“0”,写入时写“0”	R
b9	CTSE	CTS功能使能位	CTS功能使能位 0: RTS功能 1: CTS功能 注意: CTSE位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b8~b6	Reserved	-	读出时为“0”,写入时写“0”	R
b5	SCEN	智能卡模式使能位	智能卡模式使能位 0: 禁止智能卡模式 1: 使能智能卡模式 注意: 非智能卡模式时SCEN位必须保持复位值 SCEN位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b4~b0	Reserved	-	读出时为“0”,写入时写“0”	R

25.5.7 预分频寄存器 (USART_PR)

USART prescaler register

偏移地址: 0x18

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b2	Reserved	-	读出时为“0”,写入时写“0”	R
b1~b0	PSC[1:0]	预分频器值	内部时钟源时, 预分频器分频值选择位 00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64	R/W
注意: PSC[1:0]位只能在TE=0&RE=0(发送/接收禁止)时设定				

25.6 使用注意事项

25.6.1 UART 注意事项

发送器

UART 模式发送器发送禁止时(USARTn_CR1.TE=0)，则 TX 管脚可以当作普通 IO 使用，可以设定输出值和方向。如果输出 0，则会使接收方产生帧错误，从而中断数据传输。如果输出 1，则使接收方检测不到开始位从而无法开始数据传输。

接收器

UART 模式产生帧错误时，可以软件检测后续 RX 线是否为低电平，从而判断发送方是否想中断传输。

如果接收数据开始位检测方式为低电平检测，则在清除错误标志后继续接收全为低电平数据，接收错误会再次发生。

25.6.2 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_DR.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。

25.6.3 其他注意事项

- 1) 为了防止发送禁止时 TX 通信线 Hi-Z 状态，可以采用以下方法：
 - 通信线上拉
 - 发送数据结束时，USARTn_CR1.TE=0 之前，将 TX 管脚设为普通 IO 输出
 - 发送数据开始前，USARTn_CR1.TE=1 之后，将 IO 设为 TX 功能

26 集成电路总线 (I2C)

26.1 简介

I2C（集成电路总线）用作微控制器和 I2C 串行总线之间的接口。提供多主模式功能，可以控制所有 I2C 总线的协议、仲裁。支持标准模式、快速模式。还支持 SMBus 总线。

I2C 主要特性：

- 1) 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。
- 2) 标准模式最大 100Kbps，快速模式最大 400Kbps。
- 3) 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。
- 4) 可以设定 2 个从机模式地址。可同时设定 7 位地址格式和 10 位地址格式。能检测到广播呼叫地址，SMBus 主机地址，SMBus 设备默认地址，SMBus 报警地址。
- 5) 发送时可以自动判定应答位。接收时可以自动发送应答位。
- 6) 握手功能。
- 7) 仲裁功能。
- 8) 超时功能，可以检测 SCL 时钟长时间停止。
- 9) SCL 输入和 SDA 输入内置数字滤波器，滤波能力可编程。
- 10) 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。

26.2 I2C 系统框图

26.2.1 系统框图

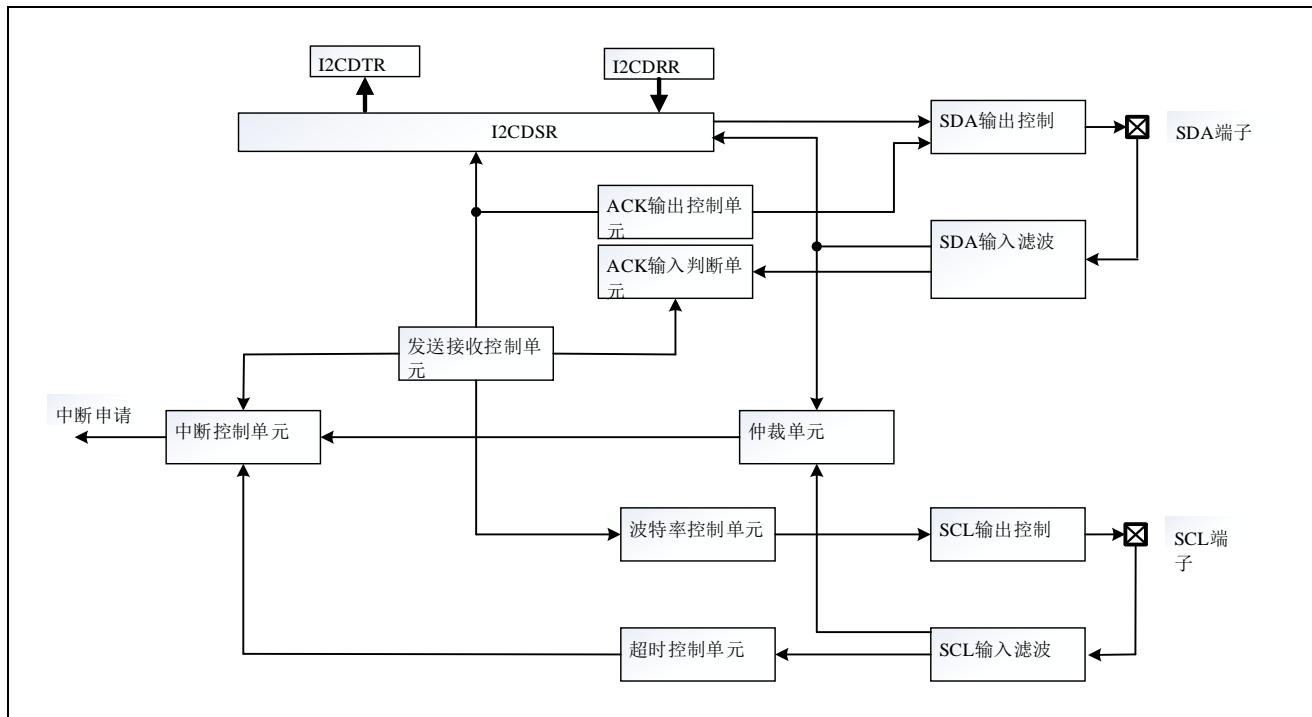


图 26-1 I2C 系统框图

26.2.2 结构图

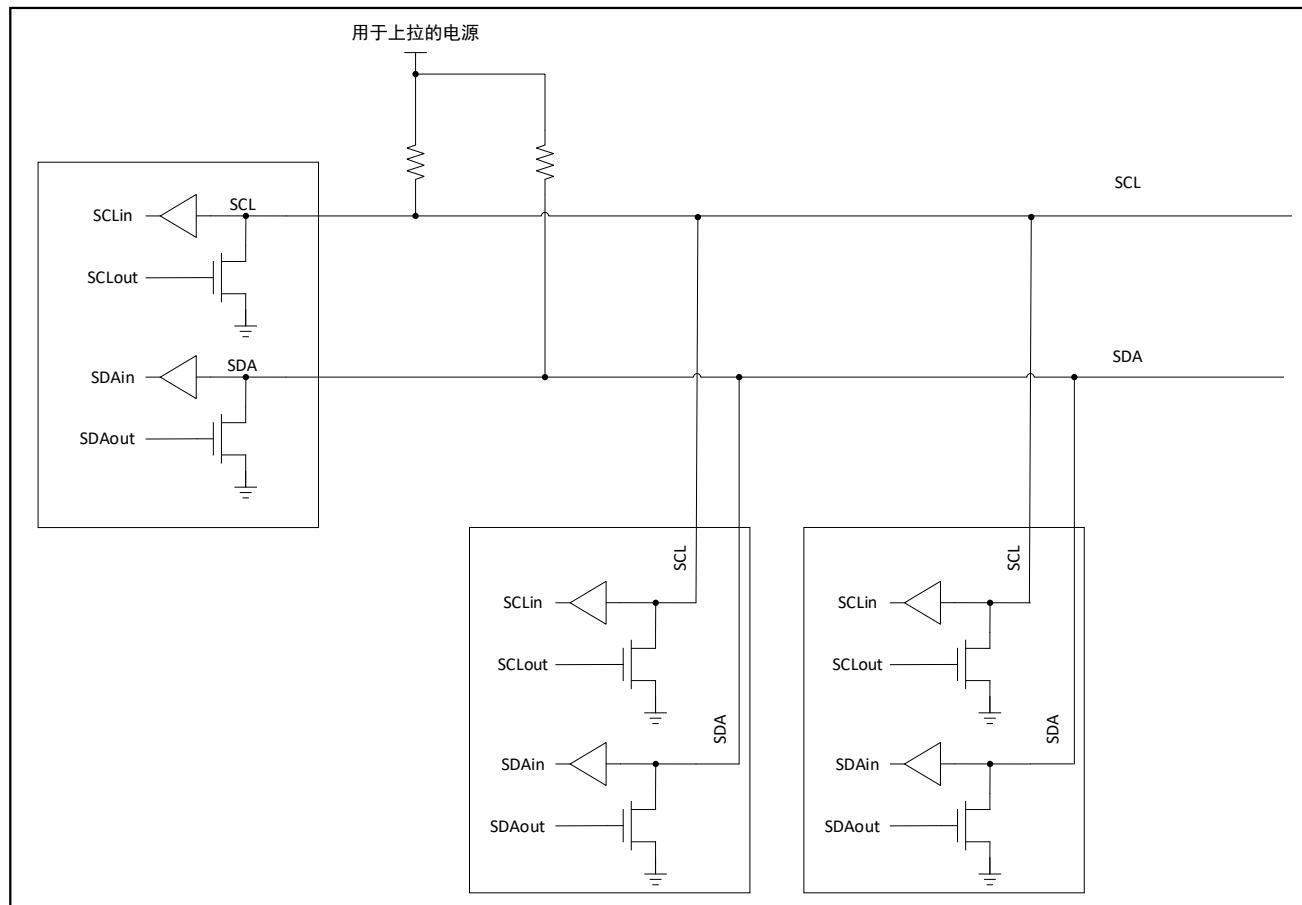


图 26-2 I2C 总线的结构例

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	串行时钟的输入/输出引脚

表 26-1 输入 / 输出引脚

当选择 I2C 总线时，SCL/SDA 输入电平为 CMOS 电平。当选择 SMBus 时，SCL/SDA 输入电平为 TTL 电平。

26.3 动作说明

本节提供了 I²C 模块功能的描述。

26.3.1 I²C 协议

I²C 总线由一根时钟线（SCL），一根数据线（SDA）构成。所有的连接器件必须是漏极开路输出。SCL，SDA 线外接上拉电阻。电阻阻值取决于系统应用。

通常情况下，一个完整的通信过程包括下列 4 部分：

1. 开始条件
2. 地址传送
3. 数据传送
4. 停止条件

下图是 I²C 总线的时序图。

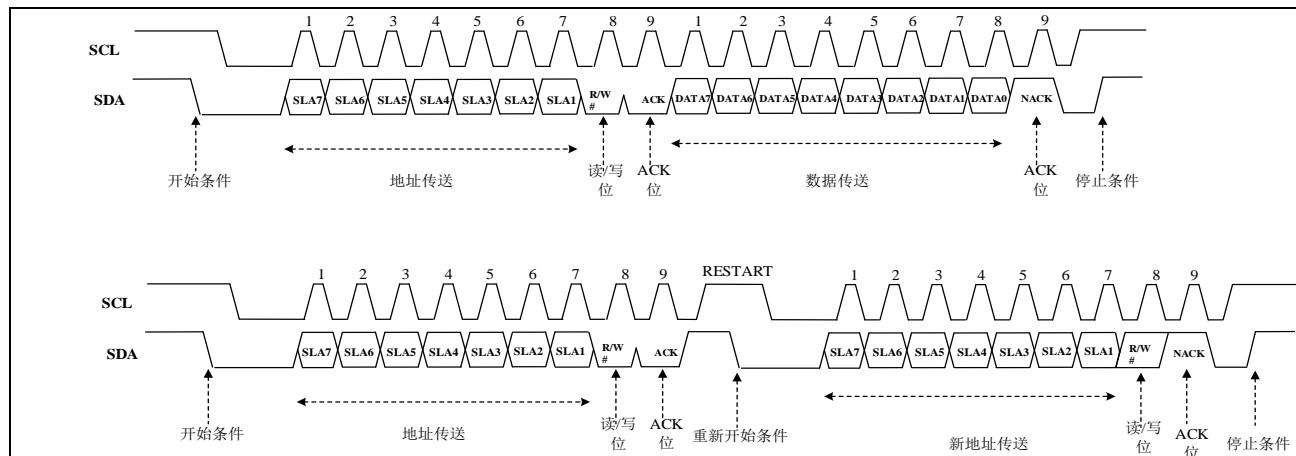


图 26-3 I²C 总线的时序图

26.3.1.1 开始条件

当总线上的主机都不驱动总线，总线进入空闲状态。SCL 和 SDA 都为高电平。总线空闲状态下总线上设备都可以通过发送开始条件启动通信。

在 I₂C_SR.BUSY 标志为“0”（总线空闲）的状态下如果将 START 位置“1”，就发行开始条件。如果检测到开始条件，就自动将 I₂C_SR.BUSY 标志和 I₂C_SR.STARTF 标志置“1”，并且自动将 START 位置“0”。此时，如果在 START 位为“1”的状态下发送的 SDA 信号和 SDA 线的信号状态相同，并且检测到开始条件，就视为通过 START 位

正确地发行了开始条件，在将 I2C_SR.MSL 位和 I2C_SR.TRA 位自动置“1”后变为主控发送模式。另外，I2C_SR.TEMPTYF 因 TRA 位为“1”而自动变为“1”。接下来把从机地址写入 I2C_DTR 寄存器，发送地址。

26.3.1.2 地址传送

开始条件或者重新开始条件后面的帧是地址帧，用于指定主机通信的对象地址。在发送停止条件之前，指定的从机一直有效。

地址帧的高 7 位为从机地址。地址帧第 8 位来决定数据帧传送的方向。

- 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 (11110XX0，其中 XX 表示 10 位地址的高两位)，然后第二帧发送低八位从机地址。

主机接受模式，主机第一帧发送头序列 (11110XX0，其中 XX 表示 10 位地址的高两位)，然后第二帧发送低八位从机地址。接下来会发送一个重新开始条件，然后再发送一帧头序列 (11110XX1，其中 XX 表示 10 位地址的高两位)。

7位地址格式

S	SLA (7位)	R/W #	ACK/NACK	DATA(8位)	AC K	-	DATA(8位)	ACK/NACK	P
---	----------	-------	----------	----------	------	---	----------	----------	---

10位地址格式

S	11110b+SLA (2位)	W#	AC K	SLA (8位)	AC K	DATA(8位)	AC K	-	DATA(8位)	ACK/NACK	P
---	-----------------	----	------	----------	------	----------	------	---	----------	----------	---

S	11110b+SLA (2位)	W#	AC K	SLA (8位)	AC K	Sr	11110b+SLA (2位)	R	AC K	DATA(8位)	AC K	-	DATA(8位)	ACK/NACK	P
---	-----------------	----	------	----------	------	----	-----------------	---	------	----------	------	---	----------	----------	---

图 26-4 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。

26.3.1.3 数据传送

地址匹配一致后，总线上的主机根据 R/W 定义的方向一帧一帧的传送数据。

所有的地址帧后传送的数据都视为数据帧。即使是 10 位地址格式的低 8 位地址也视为数据帧。

数据帧的长度是 8 位。SCL 的低电平 SDA 变化，SCL 的高电平 SDA 保持，每个时钟周期发送一位数据。数据帧后的第 9 个时钟是应答位，是接收方向发送方传送的握手信号。

如果总线上从机接收数据，在第 9 个时钟周期不响应主机，从机必须发送 NACK。如果总线上主机接收数据，第 9 个周期发送 NACK，从机接收到 NACK，从机停止发送数据。

无论主机还是从机发送了 NACK，数据传送终止。主机可以做下列任一动作：

- 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。

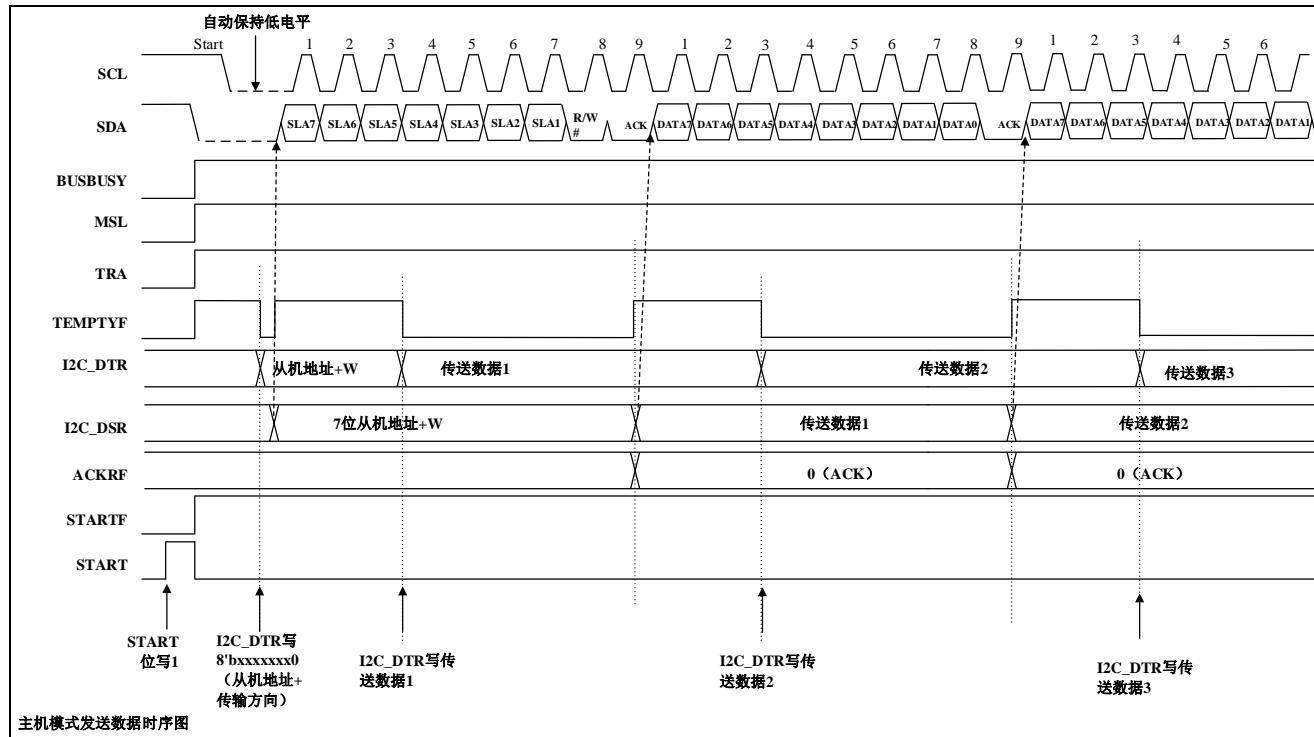


图 26-5 7 位地址格式的主机发送数据时序图（例）

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。

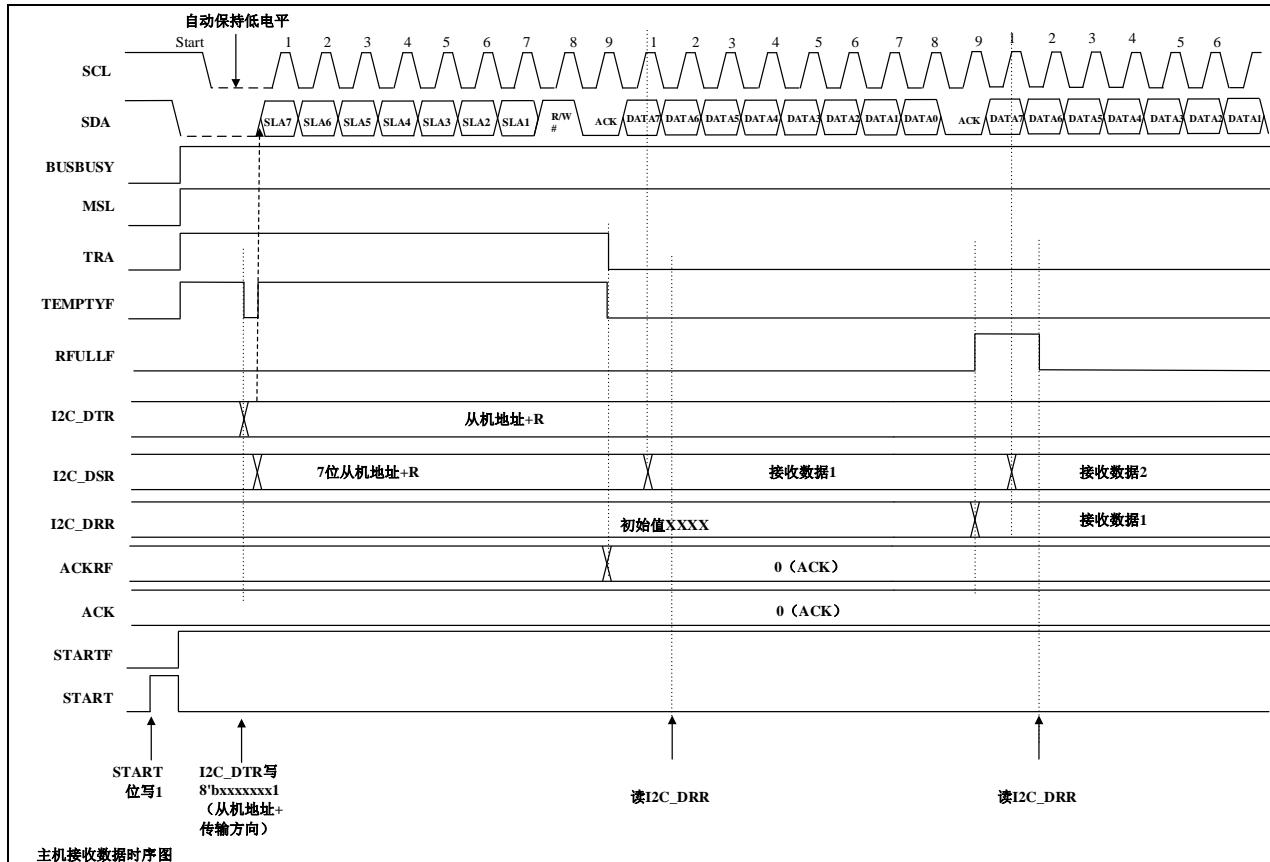


图 26-6 7 位地址格式的主机接收数据的时序图（例）

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。

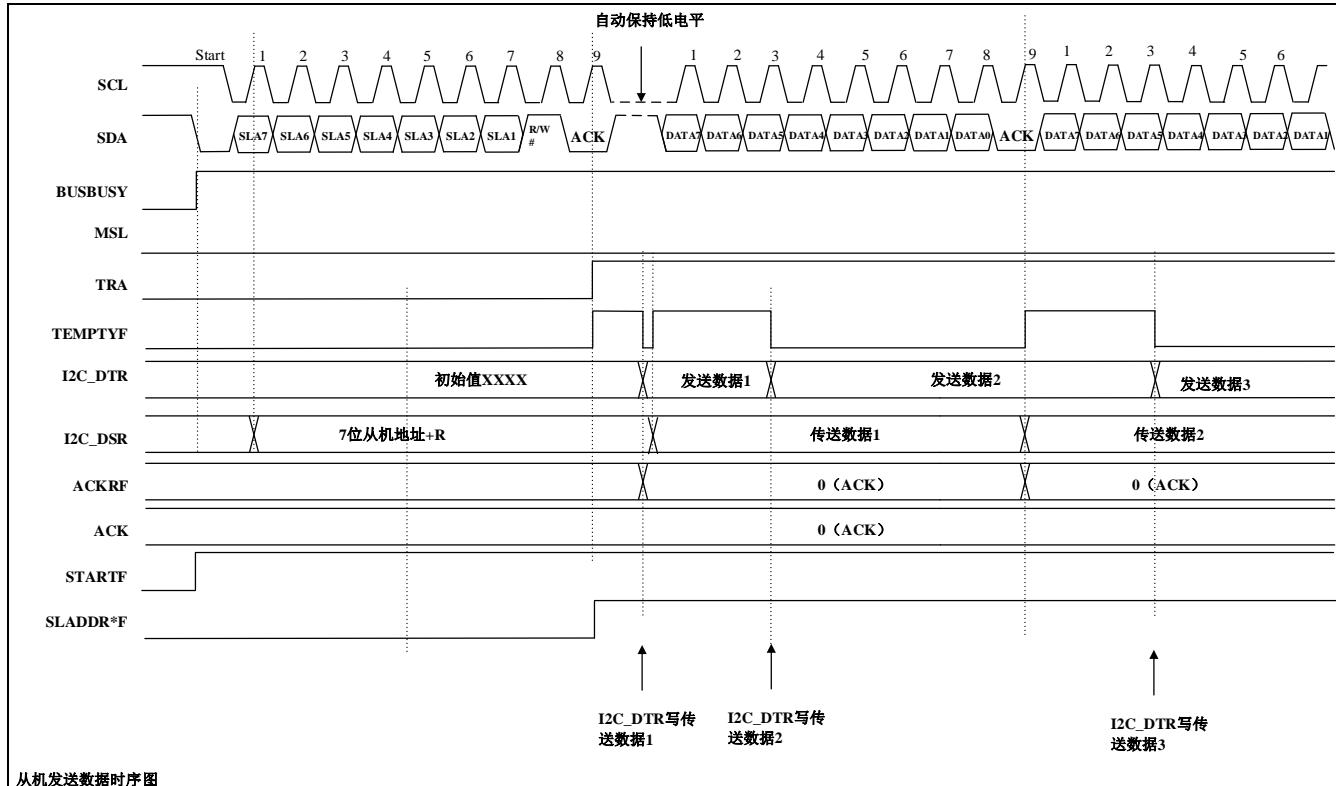
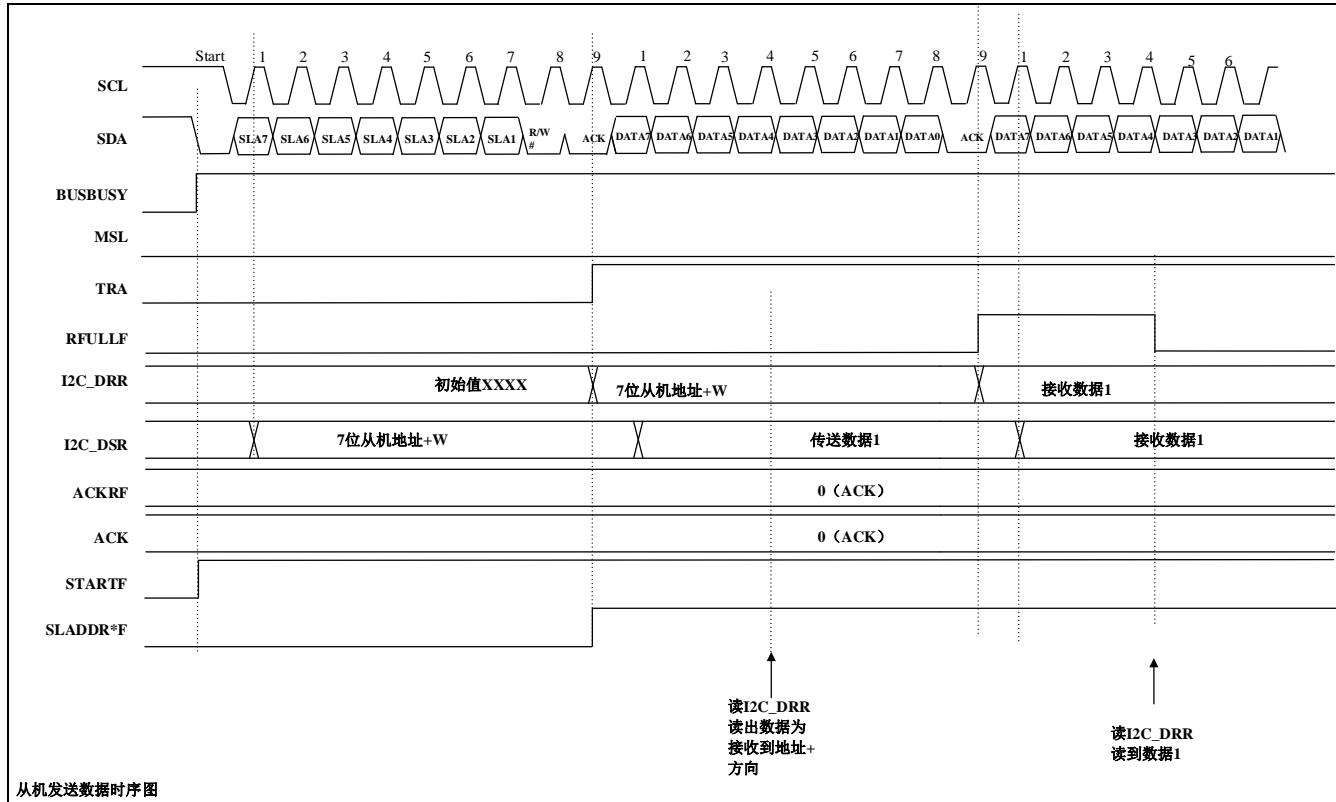


图 26-7 7 位地址格式的从机发送模式时序图（例）

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。



26.3.1.4 停止条件

通过 I2C_CR1.STOP 位发行停止条件。

在 I2C_SR.BUSY 标志为“1”（总线忙）并且 I2C_SR.MSL 位为“1”（主机模式）的状态下将 STOP 位置“1”，就发行停止条件。

26.3.1.5 重新开始条件

通过 I2C_CR1.RSTART 位生成重新开始条件。

在 I2C_SR.BUSY 标志为“1”（总线忙）并且 I2C_SR.MSL 位为“1”（主机模式）的状态下 RSTART 位置“1”，就生成行重新开始条件。

通过重新开始条件，主机可以在不释放 BUS 权的情况下，切换发送/接收模式。也可以在不释放 BUS 权的情况下和另一个从机建立通信。

26.3.1.6 SCL 时钟同步

在多主机模式中使用 I2C 总线时，有可能因和其他主机的竞争而使 SCL 时钟和 SCL 时钟发生冲突。如果 SCL 时钟发生冲突，主机就需要与 SCL 时钟同步，而且需要逐位进行 SCL 时钟的同步。当检测到 SCL 线的上升沿并且在对 I2C_CCR.SHIGHW 寄存器设定的高电平进行计数的过程中，如果因其他主机的 SCL 时钟输出而使 SCL 线下降，就在检测到 SCL 线的下降沿时中止高电平宽度的递增计数，并且在 SCL 线被驱动为低电平的同时开始对 I2C_CCR.SLOWW 设定的低电平宽度进行递增计数，在结束低电平宽度的计数时结束 SCL 线的低电平驱动，并释放 SCL 线。此时，如果其他主机的 SCL 时钟的低电平宽度大于 SLOWW 设定的低电平宽度，就延长 SCL 时钟的低电平宽度。当其他主机结束低电平输出时，释放 SCL 线并且 SCL 时钟上升。因此，在发生 SCL 时钟输出冲突时，SCL 时钟的高电平宽度与短时钟同步，低电平宽度与长时钟同步。

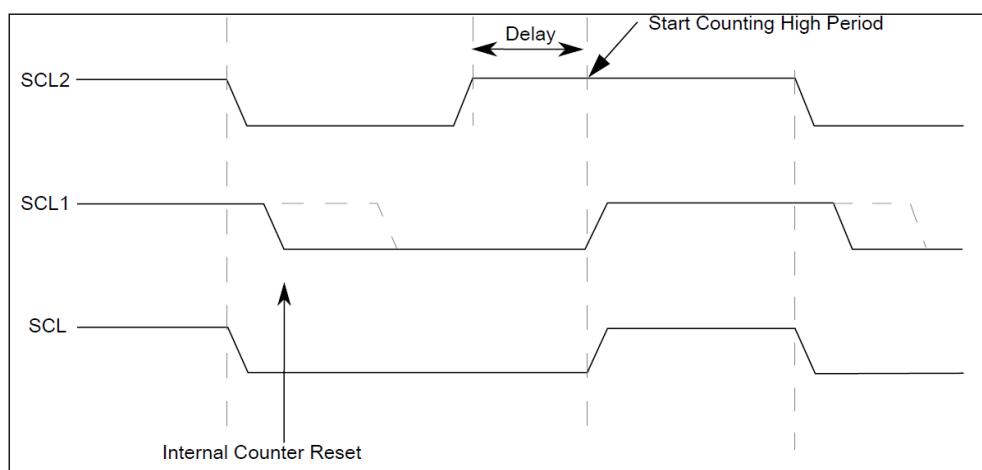


图 26-9 SCL 同步时序

26.3.1.7 仲裁

I2C 总线是一个真正的多主机总线，允许多个主机连接。

如果两个或两个以上的主机试图同时控制总线，SCL 时钟同步过程决定了总线时钟。总线时钟的低周期取决于最长的低电平时钟，高周期取决于最短的高电平时钟。高电平采集到的数据决定了仲裁结果。发送的 SDA 输出为高电平输出（SDA 引脚为高阻抗状态）而检测到 SDA 线为低电平时，就产生仲裁失败。I2C_SR.AROLF 位会硬件置“1”。如果发生主机仲裁失败，就立即转移到从机接收模式。此时，如果包括广播地

址在内的从机地址匹配，就继续从机模式的运行。

26.3.1.8 握手

数据传送过程中通过 SCL 时钟同步机制实现握手。从机在传送完一帧数据后（包含 ACK 位），将 SCL 时钟线维持在低电平。在这种情况下，SCL 时钟的低电平让主机进入等待状态，直到从机释放 SCL 线。

【从机发送模式】

- 1) 在发送模式中 (I2C_SR.TRA 位=1)，如果移位寄存器 (I2C_DSR 寄存器) 为空状态并且未写发送数据 (I2CDT 寄存器)，就在第 9 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。

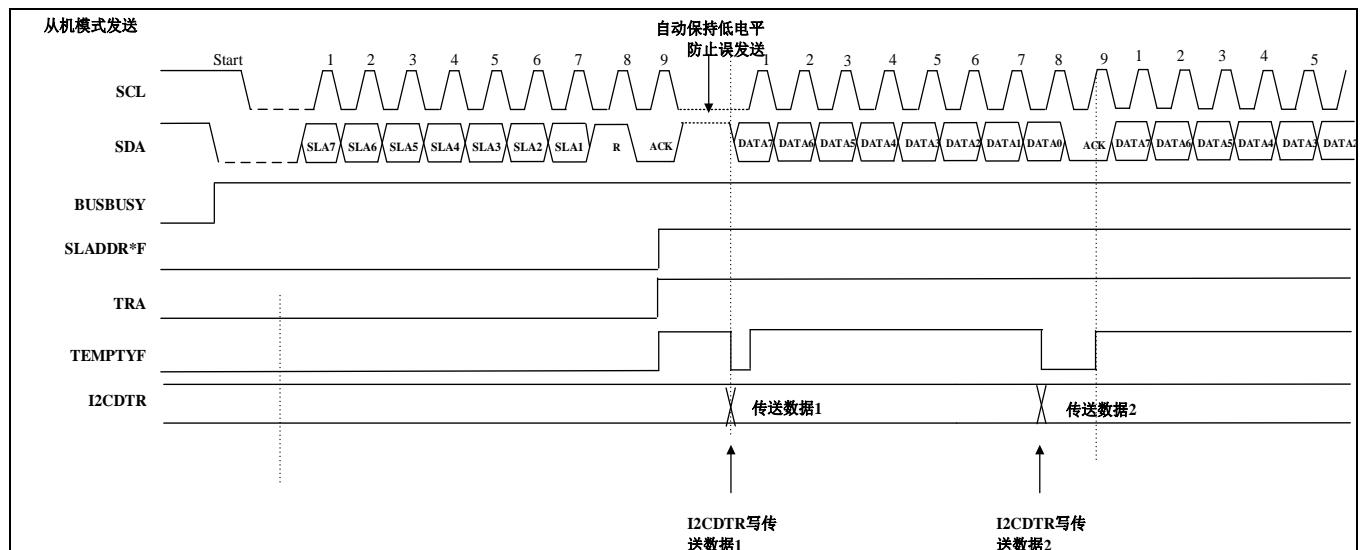


图 26-10 从机发送时序图 (1)

- 2) 在 I2C_SR.NACKF 标志变为“1”或者将最后的发送数据写到 I2C_DTR 寄存器后，在 I2C_SR.TEMPTYF 标志为“1”的状态下等到 I2C_SR.TENDF 标志变为“1”。当 I2C_SR.NACKF 标志或者 TENDF 标志为“1”时，在第 9 个时钟下降后将 SCL 线保持为低电平。此时必须通过读 I2C_DRR 寄存器来结束通信，从而释放 SCL 线。

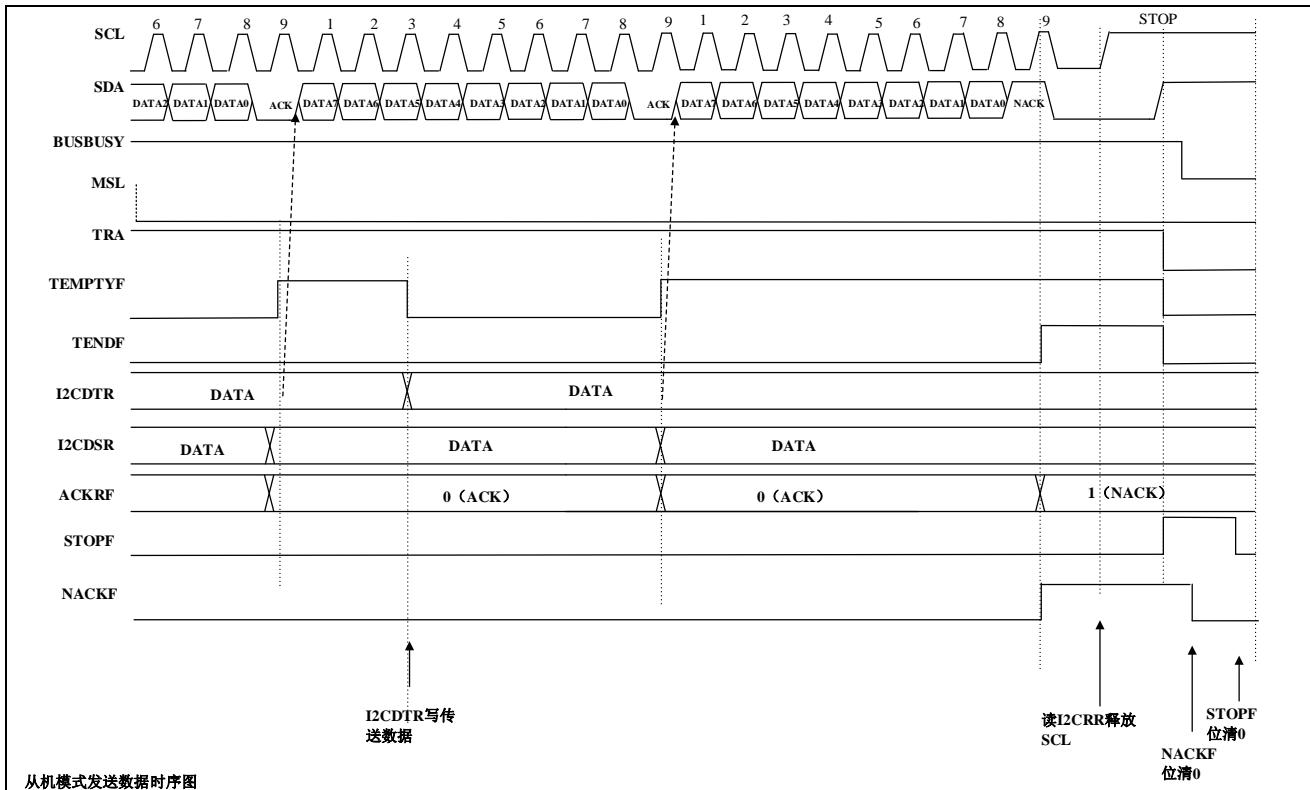
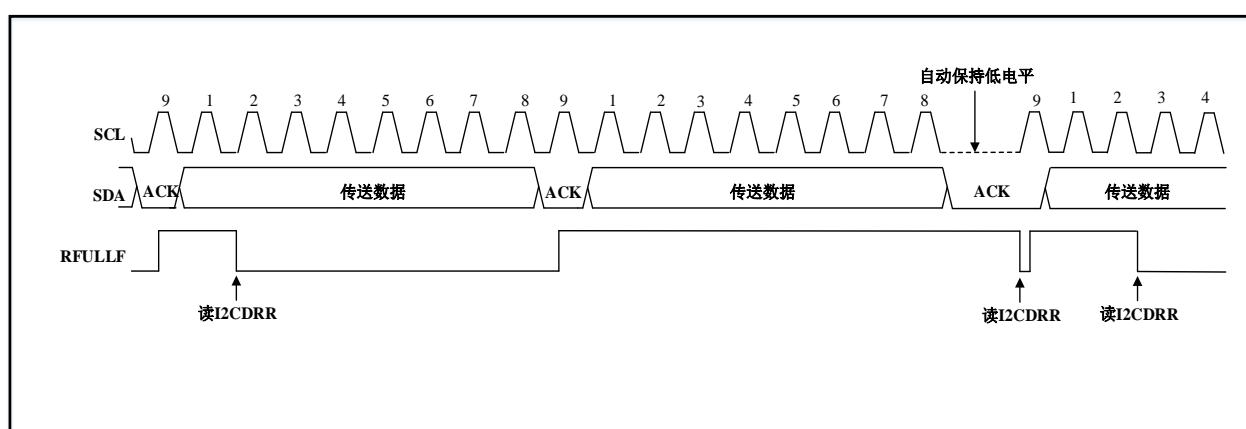


图 26-11 从机发送时序图 (2)

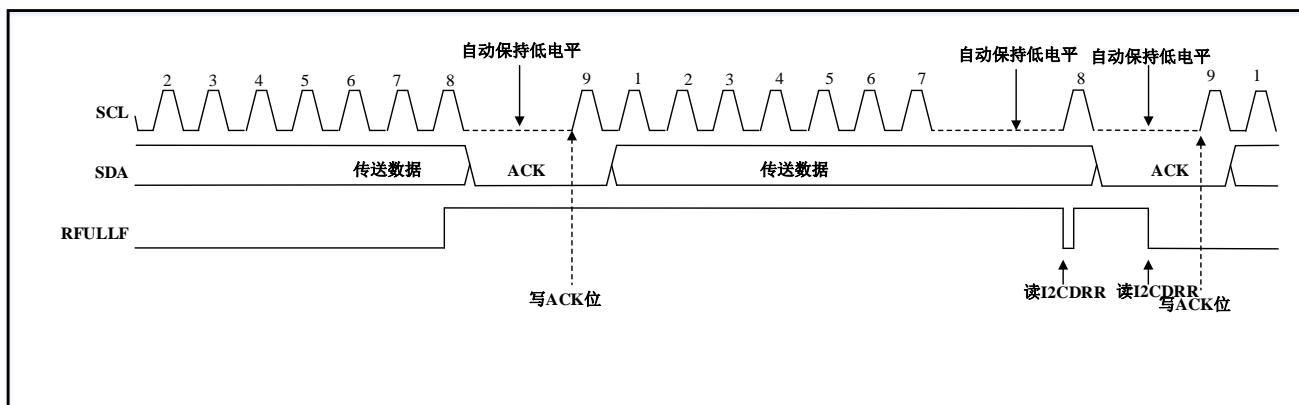
【从机接收模式】

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。



【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。



26.3.2 地址匹配

作为从机，可以设定除广播地址和主机通知地址以外的 2 种地址，从机地址能设定 7 位地址或者 10 位地址格式。

26.3.2.1 从机地址匹配

本 I2C 总线能设定 2 种从机地址，有分别对应的从机地址检测功能。当 SLADDR1EN、SLADDR0EN 为“1”时，能检测到 I2C_SLR1、I2C_SLR0 寄存器设定的从属地址。如果设定的从机地址匹配一致，就在 SCL 时钟的第 9 个时钟的下降沿将对应的 SLADDR1F、SLADDR0F 置“1”，然后根据后续的 R/W# 位将 I2C_SR.RFULLF 标志或者 I2C_SR.TEMPTYF 标志置“1”。由此，就能产生接收数据满中断或者发送数据空中断，并且能通过确认 I2C_SR.SLADDR1F、SLADDR0F 标志判断指定了哪个从机地址。

I2C_SR.SLADDR1F、SLADDR0F 标志变为“1”的时序分别如下图所示。

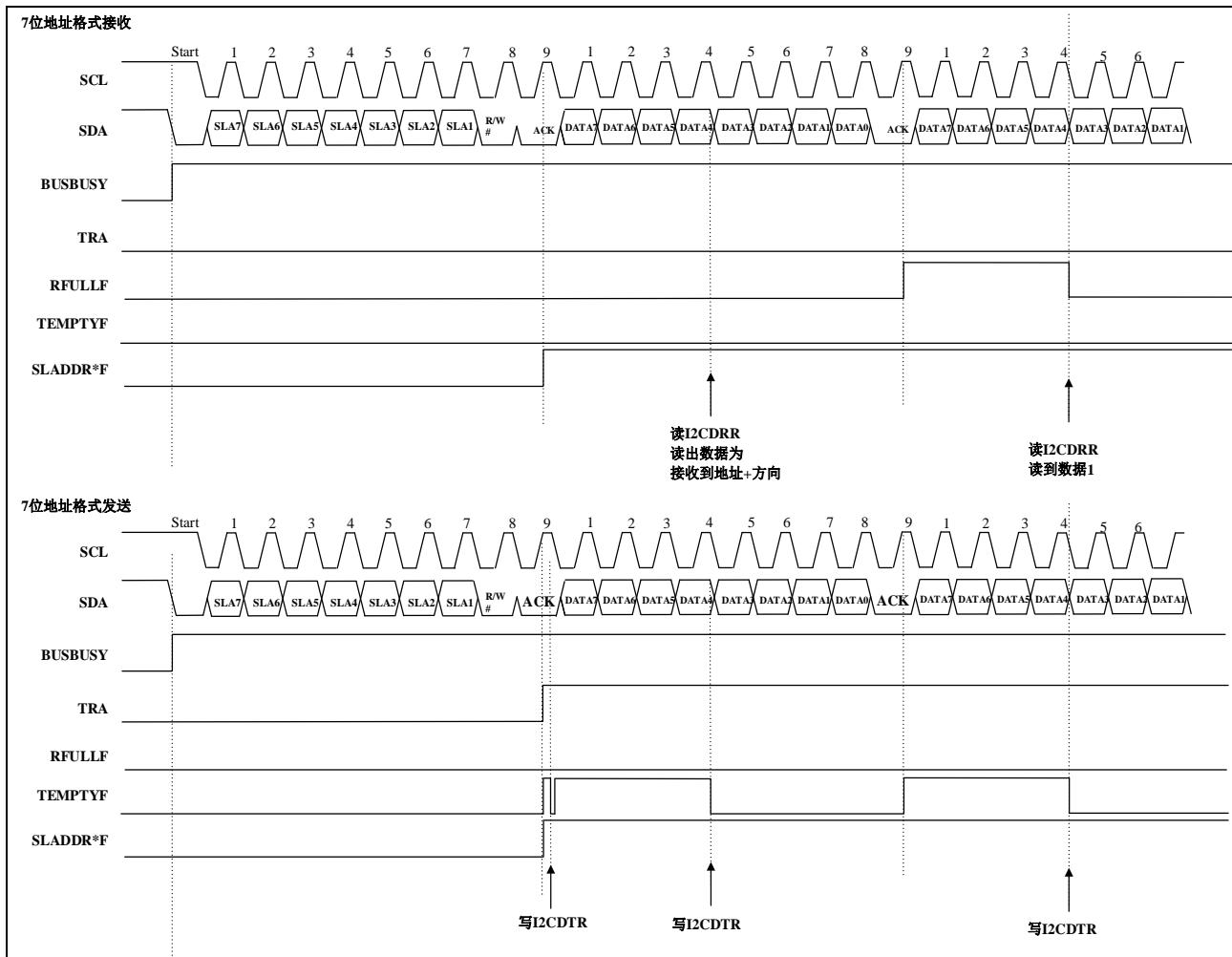


图 26-12 选择 7 位地址格式时的时序

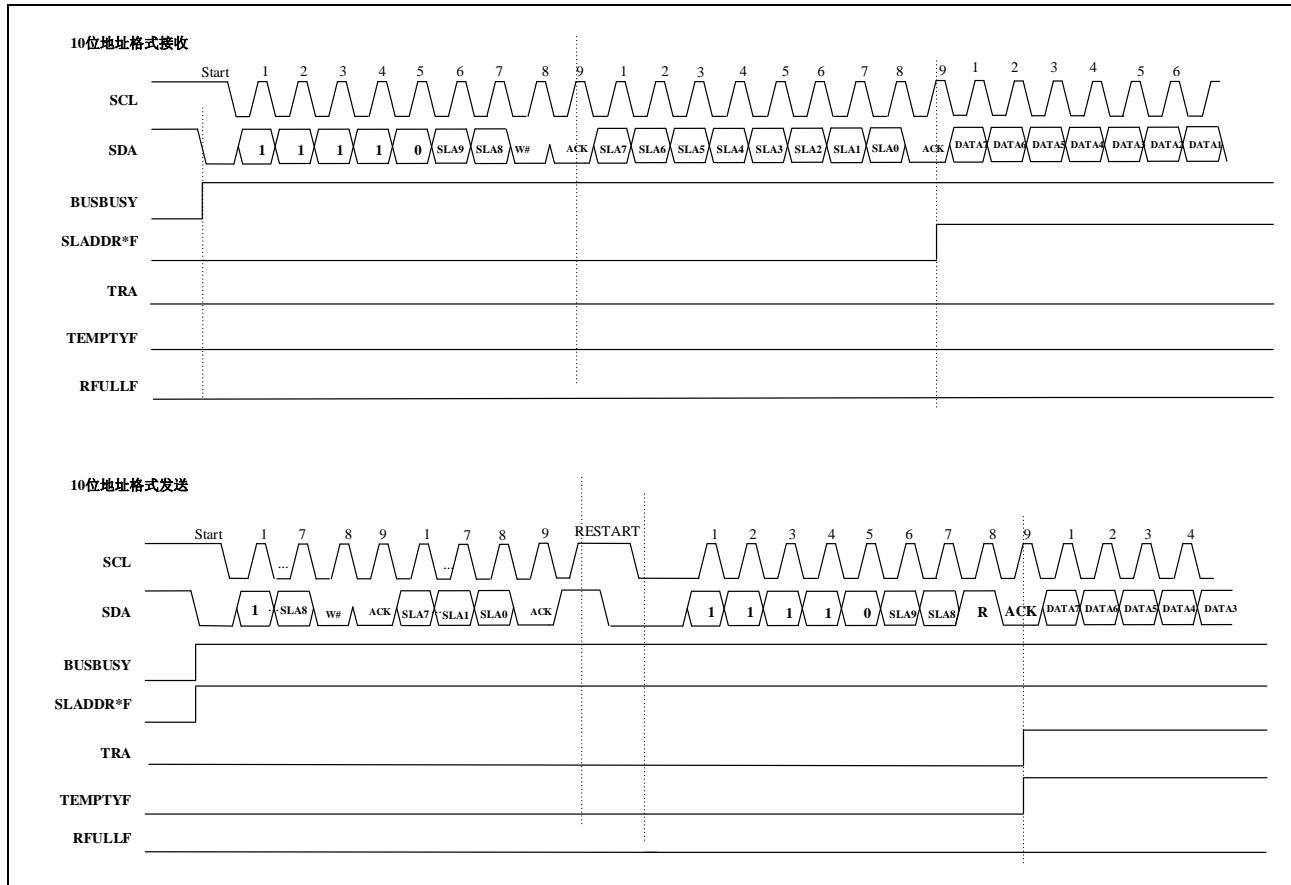


图 26-13 选择 10 位地址格式时的时序

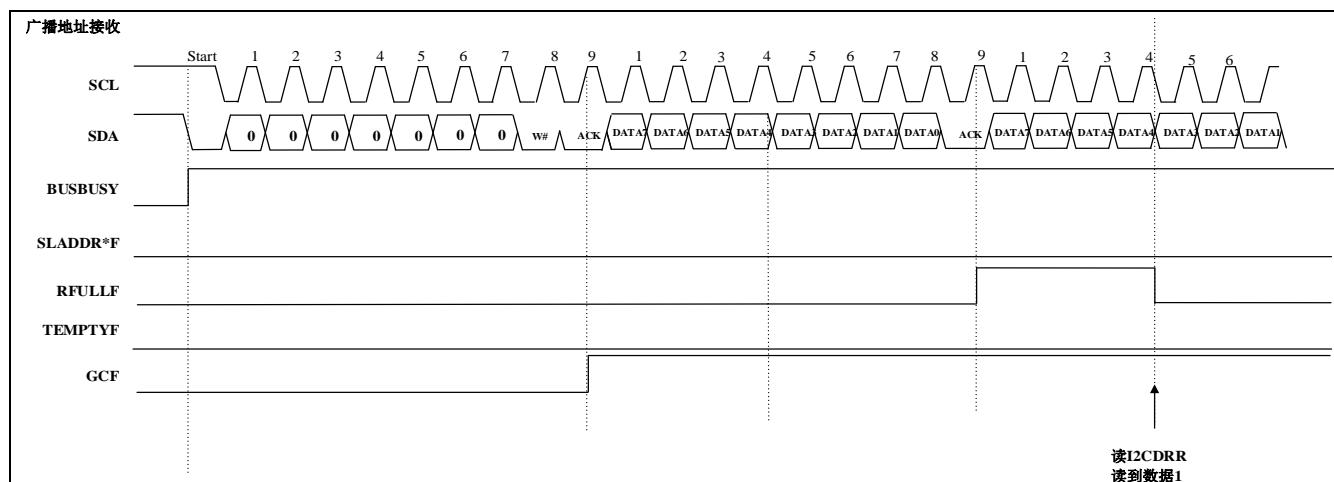
26.3.2.2 广播地址匹配

当 I2C_CR1.GCEN 位为“1”时，能检测广播地址（0000 000b+0[W]）。

但是开始条件或者重新开始条件后的地址为 0000 000b+1[R]（开始字节），就将此地址视为 All “0” 的从机地址而不视为广播地址。

如果匹配到广播地址，就在 SCL 时钟的第 9 个时钟的下降沿将 I2C_SR.GCF 标志置“1”。

广播地址匹配一致后的运行和普通的从机接收运行相同。

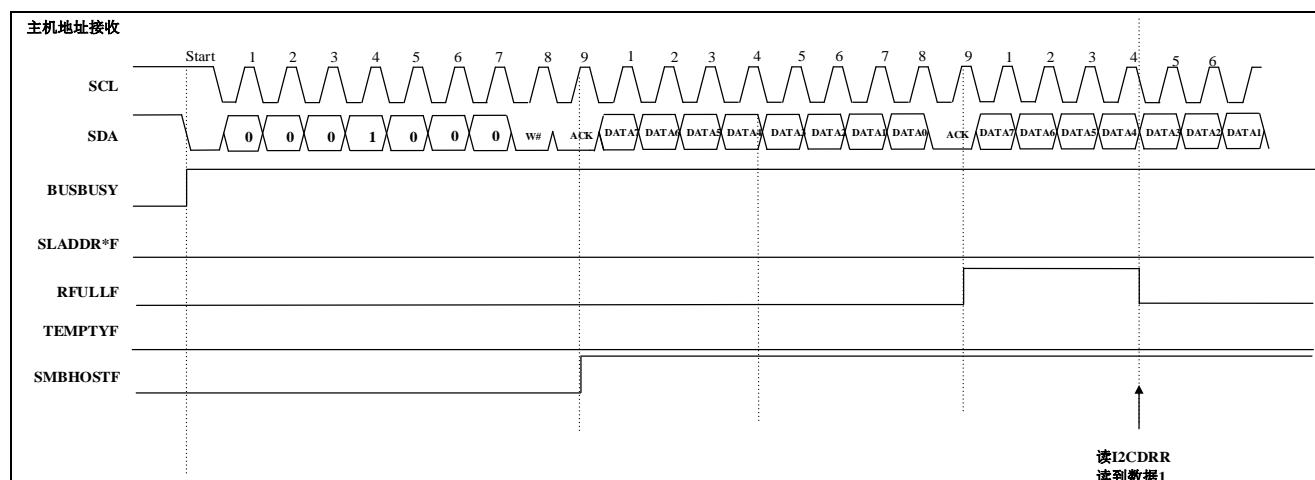


26.3.2.3 SMBus 主机地址匹配

本产品具有 SMBus 运行时的主机地址检测功能。如果在 I2C_CR1.SMBUS 位为“1”时将 I2C_CR1.SMBHOSTEN 位置“1”，就能在从机接收模式（I2C_CR1.MSL 位 TRA 位为“00b”）中检测主机地址（0001 000b）。

如果检测到 SMBUS 主机地址，就在 SCL 时钟的第 9 个时钟的下降沿将 I2C_SR.SMBHOSTF 标志置“1”。

即使接在 SMBUS 主机地址（0001 000b）后面的位是 Rd 位（R/W# 位接收到“1”），也能检测 SMBUS 主机地址。SMBUS 主机地址检测后的运行和普通的从机模式运行相同。

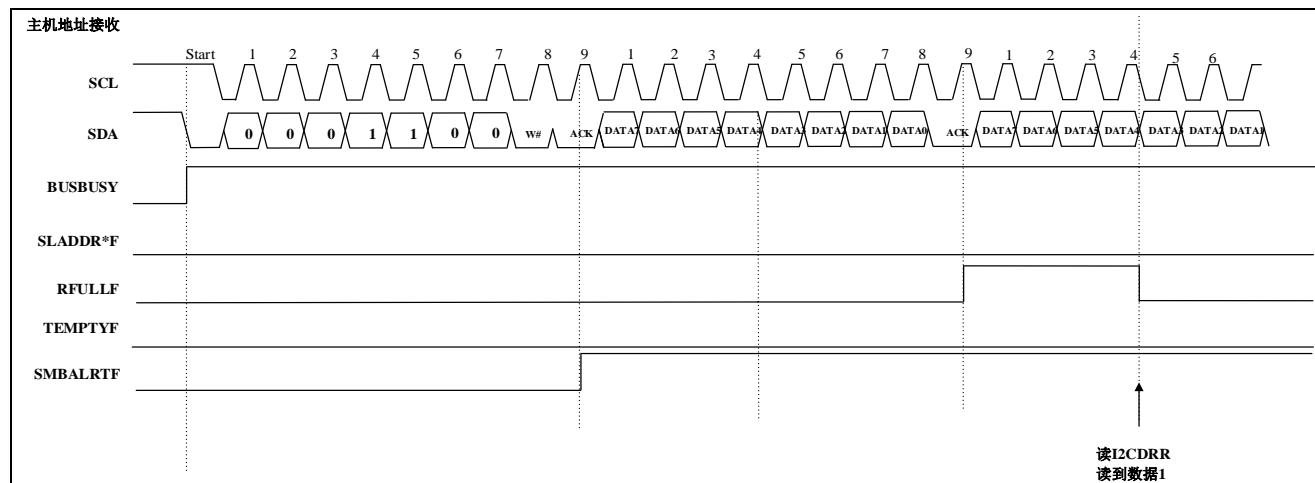


26.3.2.4 SMBus 报警响应地址匹配

本产品具有 SMBus 运行时的报警响应地址检测功能。如果在 I2C_CR1.SMBUS 位为“1”时将 I2C_CR1.SMBARLERTEN 位置“1”，就能在从机接收模式(I2C_CR1.MSL 位 TRA 位为“00b”)中检测 SMBUS 报警响应地址(0001 100b)。

如果检测到 SMBUS 报警响应地址，就在 SCL 时钟的第 9 个时钟的下降沿将 I2C_SR.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。

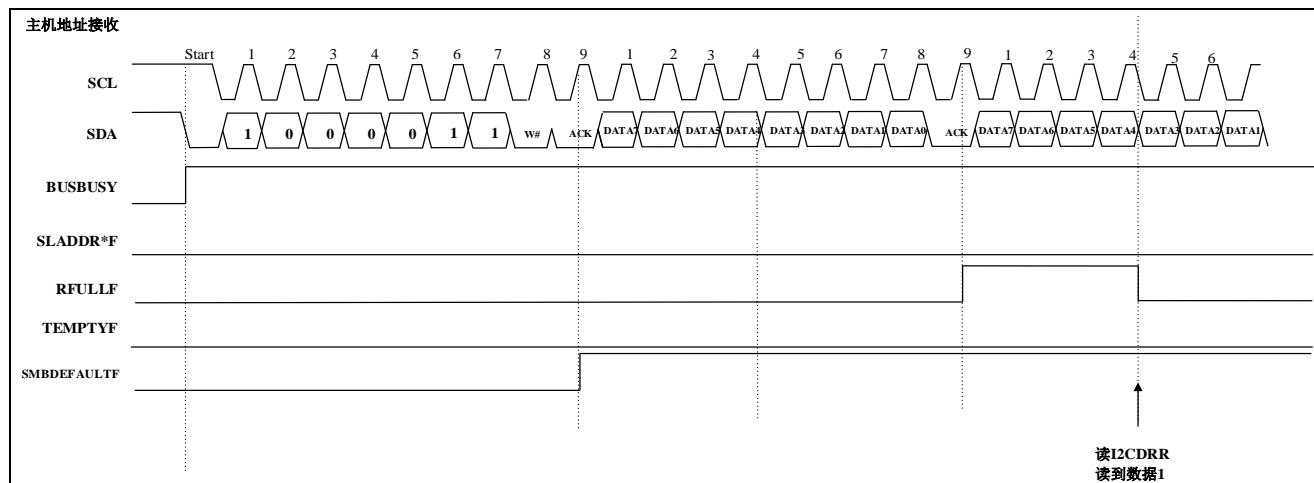


26.3.2.5 SMBus 默认地址匹配

本产品具有 SMBus 运行时的默认地址检测功能。如果在 I2C_CR1.SMBUS 位为“1”时将 I2C_CR1.SMBDEFAULTEN 位置“1”，就能在从机接收模式（I2C_CR1.MSL 位 TRA 位为“00b”）中检测 SMBUS 默认地址（1100 001b）。

如果检测到 SMBUS 默认地址，就在 SCL 时钟的第 9 个时钟的下降沿将 I2C_SR.SMBDEFAULTF 标志置“1”。

SMBUS 默认地址检测后的运行和普通的从机模式运行相同。



26.3.3 SMBus 动作

此 I2C 接口能进行以 SMBus（Ver.2.0）为基准的通信。要进行 SMBus 通信时，必须将 I2C_CR1.SMBUS 位置“1”。必须通过设定 I2C_CR1.CKDIV[2:0] 位、I2CCCR 寄存器，将传送速度设定在 SMBus 规格的 10kbps~ 100kbps 范围内。

26.3.3.1 SMBus 超时测量

1) SCL 电平超时测量

在总线忙的状态下，能通过检测到 SCL 线的低电平或者高电平被固定了一定的时间以上，并且检测到总线的异常状态。

超时检测功能监视 SCL 线的状态，通过内部计数器对高电平或者低电平的时间进行计数。如果 SCL 线有变化（上升/下降），就对内部计数器进行复位，否则就继续进行计数。如果在 SCL 线没有变化的状态下内部计数器计数到 TOUTHIGH/TOUTLOW 设定值，就能检测到超时并且通知总线的异常状态。

对于内部计数器的计数，能通过设定 HTMOUT、LTMOUT 位选择是在 SCL 线的低电平还是在高电平的状态下进行计数，或者在低电平和高电平的状态下都进行计数。如果将 HTMOUT、LTMOUT 位都置“0”，就不进行内部计数。

2) 从机的超时测量

SMBus 通信的从属设备需要测量以下所示的区间（超时间隔：TLOW:SEXT）。

- 开始条件到停止条件的区间

在通过从属设备进行超时测量时，使用开始条件检测中断和停止条件检测中断并且通过芯片定时器，测量从检测到开始条件到检测到停止条件的时间。此超时的测量时间必须在 SMBus 规格的时钟低电平的累积时间[从属设备]TLOW:SEXT：25ms (max) 以内。

如果定时器测量的时间超过 SMBus 规格的时钟 Low 电平检测的超时 TTIMEOUT：25ms (min)，从机就需要释放总线。

3) 主机的超时测量

SMBus 通信的主控设备需要测量以下所示的区间（超时间隔：TLOW:MEXT）。

- 开始条件到应答位的区间
- 应答位到下一个应答位的区间
- 应答位到停止条件的区间

在主机进行超时测量时，使用开始条件检测中断、停止条件检测中断以及发送结束中断或者接收数据满中断，通过芯片定时器测量各区间的时间。此超时的测量时间必须在 SMBus 规格的时钟低电平的累积时间[主机]TLOW:MEXT：10ms (max) 以内，开始条件到停止条件的全部 TLOW:MEXT 的累加结果必须在 TLOW:SEXT：25ms (max) 以内。

如果定时器测量的时间超过 SMBus 规格的时钟低电平的累积时间[主控设备]TLOW:MEXT：10ms (max)，或者各测量时间的累加结果超过 SMBus 规格的时钟低电平检测的超时 TTIMEOUT：25ms (min)，主机就需要中止处理。在主机发送时，必须立即中止发送（写 I2C_DTR 寄存器）。通过发行停止条件中止主机的处理。

26.3.3.2 数据包错误码（PEC）

通信中，利用 CPU 运算 CRC，发送 SMBus 的数据包错误码（PEC）或者检查接收数据。

26.3.4 复位

具有对通信模块进行复位的功能。有 2 种复位，1 种是对包括 ICCR2.BBSY 标志在内的全部寄存器进行初始化的复位，另一种是在保持各种设定值的状态下解除从属地址匹配状态以及对内部计数器进行初始化等的内部复位。

复位后，必须将 I2C_CR1.SWRST 位置“0”。

因为无论进行哪种复位都要解除 SCL 引脚/SDA 引脚的输出状态而变为高阻抗状态，所以也能用于解除总线意外停机状态。

从属模式中的复位会引起与主控设备的不同步，因此尽量避免使用。必须注意：在复位（I2C_CR1.PE 位和 I2C_CR1.SWRST 位为“01b”）过程中不能监视开始条件等的总线状态。

26.3.5 中断和事件信号输出

I2C 具有 4 种中断和用于触发启动其他外围电路的事件输出供用户选择。包括：通信错误的发生(仲裁失败检测、NACK 检测、超时检测、开始条件检测、停止条件检测)、接收结束、发送数据空、发送结束。

中断一览表如下所示。

名称	中断源	中断标志	中断条件
I2C_EEI	通信错误/通信事件	ARLOF	ARLOF=1&ARLOIE=1
		SLADDR0F	SLADDR0F=1& SLADDR0IE=1
		SLADDR1F	SLADDR1F=1& SLADDR1IE=1
		SMBALRTF	SMBALRTF =1& SMBALRTIE=1
		SMBHOSTF	SMBHOSTF =1& SMBHOSTFIE=1
		SMBDEFAULTF	SMBDEFAULTF =1& SMBDEFAULTIE=1
		GENCALLF	GENCALLF =1& GENCALLIE=1
		NACKF	NACKF=1&NACKIE-1
		TMOUTF	TMOUTF=1&TMOUTIE=1
		STARTF	STARTF=1&STARTIE=1
		STOPF	STOPF=1&STOPIE=1
I2C_RXI	接收数据满	RFULLF	RFULLF=1&RFULLIE=1
I2C_TXI	发送数据空	TEMPTYF	TEMPTYF=1&TEMPTYIE=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。

名称	事件源	事件条件
I2C_EEI	通信错误/通信时间	ARLOF=1
		SLADDR0F=1
		SLADDR1F=1
		SMBALRTF=1
		SMBHOSTF=1
		SMBDEFAULTF=1
		GENCALLF=1
		NACKF=1
		TMOUTF=1
		STARTF=1
		STOPF=1
I2C_RXI	接收数据满	RFULLF=1
I2C_TXI	发送数据空	TEMPTYF=1
I2C_TEI	发送结束	TENDF=1

26.3.6 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。

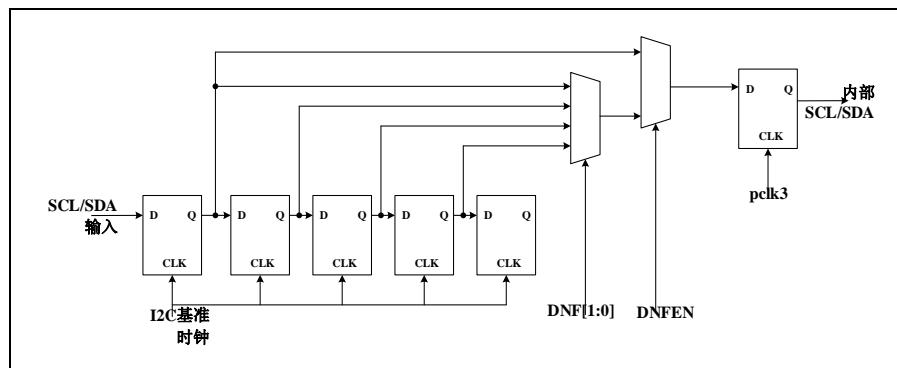


图 26-14 数字滤波电路框图

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。

26.4 应用软件设定 I2C 初始化流程

在开始发送或者接收数据时，必须下图所示的步骤进行初始化。

1. PE 位设定为 0。
2. SWRST 设定为 1，通信复位
3. PE 位设定为 1，内部状态复位
4. 设定从机地址格式和地址
5. 设定波特率
6. 根据需要设定控制寄存器功能及中断
7. SWRST 位设定为 0，解除内部状态复位。
8. 初始化结束。可发送接收数据。

26.5 寄存器说明

I2C1 基准地址: 0x4004_E000

I2C2 基准地址: 0x4004_E400

I2C3 基准地址: 0x4004_E800

寄存器名	符号	偏移地址	位宽	复位值
I2C控制寄存器1	I2C_CR1	0x00	32	0x0000_0040
I2C控制寄存器2	I2C_CR2	0x04	32	0x0000_0000
I2C控制寄存器3	I2C_CR3	0x08	32	0x0000_0006
I2C控制寄存器4	I2C_CR4	0x0C	32	0x0030_0307
I2C从机地址寄存器0	I2C_SLR0	0x10	32	0x0000_1000
I2C从机地址寄存器1	I2C_SLR1	0x14	32	0x0000_0000
I2C状态寄存器	I2C_SLTR	0x18	32	0xFFFF_FFFF
I2C状态寄存器	I2C_SR	0x1C	32	0x0000_0000
I2C状态寄存器	I2C_CLR	0x20	32	0x0000_0000
I2C数据发送寄存器	I2C_DTR	0x24	8	0xFF
I2C数据接收寄存器	I2C_DR	0x28	8	0x00
I2C波特率控制寄存器	I2C_CCR	0x2C	32	0x0000_1F1F
I2C波特率控制寄存器	I2C_FLTR	0x30	32	0x0000_0010

表 26-2 寄存器一览表

26.5.1 I2C 控制寄存器 1(I2C_CR1)

复位值: 0x0000_0040

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

SWR	-	-	-	-	ACK	STOP	STA RT	RES TART	ENG C	-	SMB HOST EN	SMBD EFAU LTEN	SMB ALRT EN	SMB US	PE
ST	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

位	标记	位名	功能	读写									
b31-16	Reserved	-	读出时为“0”,写入时写“0”	R/W									
b15	SWRST	软件复位	0: 解除复位 1: 软件复位 本位与PE位组合, 选择内部状态复位或者通信复位	R/W									
			<table border="1"> <tr> <td>SWRST</td><td>PE</td><td>复位内容</td></tr> <tr> <td>1</td><td>0</td><td>通信复位: I2C内部的全部寄存器及内部状态复位。</td></tr> <tr> <td>1</td><td>1</td><td>内部状态复位: I2C_SR, I2C_DSR寄存器和内部状态机进行复位</td></tr> </table>	SWRST	PE	复位内容	1	0	通信复位: I2C内部的全部寄存器及内部状态复位。	1	1	内部状态复位: I2C_SR, I2C_DSR寄存器和内部状态机进行复位	
SWRST	PE	复位内容											
1	0	通信复位: I2C内部的全部寄存器及内部状态复位。											
1	1	内部状态复位: I2C_SR, I2C_DSR寄存器和内部状态机进行复位											
b14-11	Reserved	-	读出时为“0”,写入时写“0”	R/W									
b10	ACK	发送应答	0: 应答位发送“0”(发送ACK) 1: 应答位发送“1”(发送NACK)	R/W									
b9	STOP	停止条件生成位	0: 不生成停止条件 1: 生成停止条件 此位可软件置1和清0。 硬件清0条件: 检测到停止条件 仲裁失败 检测到开始条件 通信复位	R/W									
b8	START	起始条件生成位	0: 不生成起始条件 1: 生成起始条件 此位可软件置1和清0。 硬件清0条件: 检测到开始条件 仲裁失败时 通信复位	R/W									
b7	RESTART	重复起始条件生成位	0: 不生成重复起始条件	R/W									

1: 生成重复起始条件

此位可软件置1和清0。

硬件清0条件:

1) 检测到开始条件

2) 仲裁失败时

3) 通信复位

b6	ENGC	广播呼叫使能	0: 广播地址检测无效 1: 广播地址检测有效	R/W
b5	Reserved	-	读出时为“0”，写入时写“0”	R/W
b4	SMBHOSTEN	允许匹配SMBUS主机地址位	0: 禁止匹配SMBUS主机地址 1: 允许匹配SMBUS主机地址	R/W
b3	SMBDEFAULTEN	允许匹配SMBUS默认地址位	0: 禁止匹配SMBUS默认地址 1: 允许匹配SMBUS默认地址	R/W
b2	SMBALRTEN	允许匹配SMBUS报警响应地址位	0: 禁止SMBUS报警响应地址 1: 允许SMBUS报警响应地址	R/W
b1	SMBUS	SMBUS/I2C总线模式选择位	0: I2C总线模式 1: SMBUS总线模式	R/W
b0	PE	I2C功能使能	0: I2C功能禁止 1: I2C功能允许 本位与SWRST位组合，选择内部状态复位或者通信复位	R/W

26.5.2 I2C 控制寄存器 1(I2C_CR2)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	SMB ALRT IE	SMB HOST IE	SMB DEFA ULTI	GEN CALL IE	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	TMO UTIE	-	NAC KIE	-	-	ARL OIE	-	TEMP TYIE	RFUL LIE	-	STO PIE	TEN DIE	SLA DDR1 IE	SLA DDR0 IE	STA RTIE

位	标记	位名	功能	读写
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	SMBALRTIE	SMBUS报警响应地址	0: SMBUS报警响应地址匹配一致中断禁止 1: SMBUS报警响应地址匹配一致中断允许	R/W
b22	SMBHOSTIE	SMBUS主机地址匹配	0: SMBUS主机地址匹配一致中断禁止 1: SMBUS主机地址匹配一致中断允许	R/W
b21	SMBDEFAULTIE	SMBUS默认地址匹配	0: SMBUS默认地址匹配一致中断禁止 1: SMBUS默认地址匹配一致中断允许	R/W
b20	GENCALLIE	广播呼叫地址匹配一 致中断允许	0: 广播呼叫地址匹配一致中断禁止 1: 广播呼叫地址匹配一致中断允许	R/W
b19~b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14	TMOUTIE	超时中断允许	0: 超时中断禁止 1: 超时中断允许	R/W
b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	NACKIE	NACK中断允许	0: 接收到NACK中断禁止 1: 接收到NACK中断允许	R/W
b11~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9	ARLOIE	仲裁失败中断允许	0: 仲裁失败中断禁止 1: 仲裁失败中断允许	R/W
b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	TEMPTYIE	发送数据空中断允许 位	0: 发送数据空中断禁止 1: 发送数据空中断允许	R/W
b6	RFULLIE	接收数据满中断允许 位	0: 接收数据满中断禁止 1: 接收数据满中断允许	R/W
b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4	STOPIE	停止条件中断允许	0: 总线检测到停止条件中断禁止 1: 总线检测到停止条件中断允许	R/W
b3	TENDIE	发送一帧数据结束中	0: 发送一帧数据结束中断禁止	R/W

		断允许位	1: 发送一帧数据结束中断允许	
b2	SLADDR1IE	从机地址1匹配一致中 断允许	0: 从机地址1匹配一致中断禁止 1: 从机地址1匹配一致中断允许	R/W
b1	SLADDR0IE	从机地址0匹配一致中 断允许	0: 从机地址0匹配一致中断禁止 1: 从机地址0匹配一致中断允许	R/W
b0	STARTIE	开始条件/重新开始条 件中断允许	0: 总线检测到开始条件中断禁止 1: 总线检测到开始条件中断允许	R/W

26.5.3 I2C 控制寄存器 1(I2C_CR3)

复位值: 0x0000_0006

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

-	-	-	-	-	-	-	FACK EN	-	-	-	-	-	HTM OUT	LTM OUT	TMOU TEN
---	---	---	---	---	---	---	------------	---	---	---	---	---	------------	------------	-------------

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	FACKEN	RFULLF标志位置位时间点选择	0: 在SCL时钟的第9个时钟上升时,此位为“1”。(在第8个时钟的下降沿, SCL线不保持低电平) 1: 在SCL时钟的第8个时钟上升时,此位为“1”。(在第8个时钟的下降沿, SCL线保持低电平) 通过写ACK位来解除保持的低电平。	R/W
b6~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	HTMOUT	高电平超时检测允许	0: 在SCL线为高电平时,禁止超时检测。 1: 在SCL线为高电平时,允许超时检测。	R/W
b1	LTMOUT	低电平超时检测允许	0: 在SCL线为低电平时,禁止超时检测。 1: 在SCL线为低电平时,允许超时检测。	R/W
b0	TMOUTEN	超时功能允许位	0: 检测SCL电平超时功能禁止 1: 检测SCL电平超时功能允许	R/W

26.5.4 I2C 控制寄存器 1(I2C_CR4)

复位值: 0x0030_0307

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	BUS WAIT	-	-	-	-	-	-	-	-	-	-

位	标记	位名	功能	读写
b31~b22	Reserved	-	读出时为“0”,写入时写“0”	R/W
b21~b20	Reserved	-	读出时为“1”,写入时写“1”	R/W
b19~b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10	BUSWAIT	总线等待位	0: I2C_DRR接收满、I2C_DSR为空时, 在第9个时钟和下次传送的第1个时钟之间不保持低电平, 继续接收下一个数据。 1: I2C_DRR接收满、I2C_DSR为空时, 在第9个时钟和下次传送的第1个时钟之间保持低电平, 通过读I2C_DRR寄存器来解除保持的低电平。	R/W
b9~b8	Reserved	-	读出时为“1”,写入时写“1”	R/W
b7~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~b0	Reserved	-	读出时为“1”,写入时写“1”	R/W

26.5.5 I2C 从机地址寄存器 0(I2C_SLR0)

复位值: 0x0000_1000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADD RMO D0	-	-	SLA DDR0 EN	-	-										SLADDR0[9:0]

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”，写入时写“0”	R/W
b15	ADDRMOD0	7位/10位地址格式选择位	0: 选择7位地址格式 1: 选择10位地址格式	R/W
b14~b13	Reserved	-	读出时为“0”，写入时写“0”	R/W
b12	SLADDR0EN	从机地址0有效位	0: 从机地址寄存器0设定值无效 1: 从机地址寄存器0设定值有效	R/W
b11~b10	Reserved	-	读出时为“0”，写入时写“0”	R/W
b9~b8	SLADDR0[9:8]	10位从机地址的高位	设定从机地址。 当ADDRMOD0位为“0”时，此位设定无效。 当ADDRMOD0位为“1”时，此位作为10位从机地址的高两位。	R/W
b7~b0	SLADDR0[7:0]	7位地址/10位地址的低位	设定从机地址。 当ADDRMOD0位为“0”时，SLADDR0[7:1]为7位从机地址。 SLADDR0[0]位无效。 当ADDRMOD0位为“1”时，SLADDR0[7:0]为10位从机地址的低8位地址。	R/W

26.5.6 I2C 从机地址寄存器 1(I2C_SLR1)

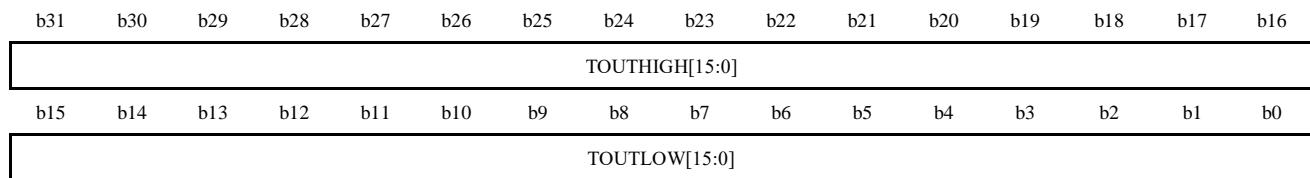
复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADD RMO D1	-	-	SLA DDR1 EN	-	-										SLADDR1[9:0]

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”，写入时写“0”	R/W
b15	ADDRMOD1	7位/10位地址格式选择位	0: 选择7位地址格式 1: 选择10位地址格式	R/W
b14~b13	Reserved	-	读出时为“0”，写入时写“0”	R/W
b12	SLADDR1EN	从机地址1有效位	0: 从机地址寄存器1设定值无效 1: 从机地址寄存器1设定值有效	R/W
b11~b10	Reserved	-	读出时为“0”，写入时写“0”	R/W
b9~b8	SLADDR1[9:8]	10位从机地址的高位	设定从机地址。 当ADDRMOD1位为“0”时，此位设定无效。 当ADDRMOD1位为“1”时，此位作为10位从机地址的高两位。	R/W
b7~b0	SLADDR1[7:0]	7位地址/10位地址的低位	设定从机地址。 当ADDRMOD1位为“0”时，SLADDR1[7:1]为7位从机地址。 SLADDR1[0]位无效。 当ADDRMOD1位为“1”时，SLADDR1[7:0]为10位从机地址的低8位地址。	R/W

26.5.7 I2C SCL 电平超时控制寄存器(I2C_SLTR)

复位值: 0xFFFF_FFFF



位	标记	位名	功能	读写
b31~b16	TOUTHIGH	SCL高电平超时周期	TOUTHIGH设定SCL高电平超时周期。 SCL高电平超时时间=TOUTHIGH×I2C基准时钟周期	R/W
b15~b0	TOUTLOW	SCL低电平超时周期	TOUTLOW设定SCL低电平超时周期。 SCL低电平超时时间=TOUTLOW×I2C基准时钟周期	R/W

26.5.8 I2C 状态寄存器(I2C_SR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	SMB ALR TF	SMB HOS TF	SMBD EFAU LT	GENC ALLF	-	TRA	BUSY	MSL
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	TMO UTF	-	NAC KF	-	ACK RF	ARL OF	-	TEM PTYF	RFU LLF	-	STO PF	TEN DF	SLAD DR1F	SLAD DR0F	STA RTF

位	标记	位名	功能	读写
b31~b21	Reserved	-	读出时为“0”,写入时写“0”	R
b23	SMBALRTF	SMBUS报警响应地址匹配一致标志位	0: 未匹配到SMBUS报警响应地址 1: 检测到主机地址 置“1”条件: 接收到的地址和0001 100b匹配一致 清“0”条件: SMBALRTFCLR写“1” 检测到停止条件 通信复位	R
b22	SMBHOSTF	SMBUS主机地址匹配一致标志位	0: 未匹配到SMBUS主机地址 1: 匹配到SMBUS主机地址 地址匹配一致条件如下: 置“1”条件: 接收到的地址和0001 000b匹配一致 清“1”条件: SMBHOSTFCLR写“1” 检测到停止条件 通信复位	R
b21	SMBDEFAULTF	SMBUS默认地址匹配一致标志位	0: 未匹配到SMBUS默认地址 1: 匹配到SMBUS默认地址 置“1”条件: 接收到的地址和1100 001b匹配一致 清“0”条件: SMBDEFAULTFCLR写“1” 检测到停止条件 通信复位	R
b20	GENCALLF	广播呼叫地址匹配一致标志	0: 未匹配到广播呼叫地址 1: 匹配到广播呼叫地址	R

置“1”条件:

当接收的从机地址与广播呼叫地址（All“0”）匹配一致时

清“0”条件:

GENCALLFCLR写“1”

检测到停止条件

通信复位

b19	Reserved	-	读出时为“0”，写入时写“0”	R															
b18	TRA	发送接收选择位	<p>此位表示选择发送数据还是接收数据。</p> <p>0：接收数据</p> <p>1：发送数据</p> <p>此位可软件置1和清0。</p> <p>硬件置“1”条件</p> <p>检测到开始条件</p> <p>主机模式下，发送的R/W位为0</p> <p>从机模式下，地址匹配且接收的R/W位为1</p> <p>硬件清“0”条件</p> <p>检测到停止条件</p> <p>主机模式下，发送的R/W位为1</p> <p>从机模式下，地址匹配且接收的R/W位为0</p> <p>通信复位</p>	R/W															
b17	BUSY	总线忙标志位	<p>0：空闲状态，总线上无通信</p> <p>1：占有状态，总线正在通信</p> <p>置“1”条件:</p> <p>检测到总线上开始条件</p> <p>清“0”条件:</p> <p>检测到总线停止条件</p> <p>通信复位</p>	R															
b16	MSL	主从机选择位	<p>此位表示主机还是从机。</p> <p>0：从机模式</p> <p>1：主机模式</p> <p>通过和TRA位的组合，表示I2C的运行模式。</p> <table border="1" data-bbox="759 1545 1303 1769"> <tbody> <tr> <th>MSL</th><th>TRA</th><th>I2C运行模式</th></tr> <tr> <td>0</td><td>0</td><td>从机接收模式</td></tr> <tr> <td>0</td><td>1</td><td>从机发送模式</td></tr> <tr> <td>1</td><td>0</td><td>主机接收模式</td></tr> <tr> <td>1</td><td>1</td><td>主机发送模式</td></tr> </tbody> </table> <p>此位可软件置1和清0。</p> <p>硬件置“1”条件</p> <p>START位为1的状态下，检测到开始条件</p> <p>硬件清“0”条件</p> <p>1) 检测到停止条件</p> <p>2) 仲裁失败</p>	MSL	TRA	I2C运行模式	0	0	从机接收模式	0	1	从机发送模式	1	0	主机接收模式	1	1	主机发送模式	R/W
MSL	TRA	I2C运行模式																	
0	0	从机接收模式																	
0	1	从机发送模式																	
1	0	主机接收模式																	
1	1	主机发送模式																	

3) 通信复位				
b15	Reserved	-	读出时为“0”,写入时写“0”	R
14	TMOUTF	超时标志位	0: 未检测到SCL电平超时 1: SCL电平超时 置“1”条件: I2C_SLTR设定的周期内, SCL未翻转 清“1”条件: TMOUTFCLR写“1” 通信复位	R
b13	Reserved	-	读出时为“0”,写入时写“0”	R
b12	NACKF	NACK标志位	0: 未接收到NACK 1: 接收到NACK 置“1”条件: 发送模式下, 接收到NACK 清“0”条件: NACKFCLR写“1” 通信复位	R
b11	Reserved	-	读出时为“0”,写入时写“0”	R
b10	ACKRF	接收应答位	0: 接收到应答位为“0” (接收ACK) 1: 接收到应答位为“1” (接收NACK) 置“1”条件: 发送模式下, 接收到NACK 清“0”条件: 发送模式下, 接收到ACK 通信复位	R
b9	ARLOF	仲裁失败标志位	0: 未发生仲裁失败 1: 仲裁失败 置“1”条件: 仲裁失败 清“0”条件: ARLOFCLR写“1” 通信复位	R
b8	Reserved	-	读出时为“0”,写入时写“0”	R
b7	TEMPTYF	发送数据空标志位	0: I2C_DTR寄存器满 1: I2C_DTR寄存器空 置“1”条件: I2C_DTR数据传送到I2C_DSR TRA位置1 清“0”条件: 写I2C_DTR TRA位清0 通信复位	R

b6	RFULLF	接收数据满标志位	0: I2C_DRR寄存器空 1: I2C_DRR寄存器满 置“1”条件: 接收的数据从I2C_DSR传送到I2C_DRR 清“0”条件: 读I2C_DRR RFULLFCLR写“1” 通信复位	R
b5	Reserved	-	读出时为“0”,写入时写“0”	R
b4	STOPF	停止条件标志位	0: 总线未检测到停止条件 1: 总线检测到停止条件 置“1”条件: 检测到停止条件 清“0”条件: STOPFCLR写“1” 通信复位	R
b3	TENDF	发送数据结束标志位	0: I2C_DSR寄存器发送中 1: I2C_DSR寄存器发送结束 置“1”条件: TEMPTYF=1的条件下, SCL的第9个上升沿此位置“1” 清“0”条件: 检测到停止条件 写I2C_DTR TENDFCLR写“1” 通信复位	R
b2	SLADDR1F	从机地址寄存器1匹配一致标志	0: 未检测到从机地址寄存器1一致地址 1: 检测到从机地址寄存器1一致地址 置“1”条件: 当I2C_SLR1.ADDRMOD1位为“0”时, 接收到的从机地址和I2C_SLR1.SLADDR1[7:1]匹配时。 当I2C_SLR1.ADDRMOD1位为“1”时, 接收10位从机地址的第一个字节地址与11110b+I2C_SLR1.SLADDR1[9:8]匹配一致并且第二个字节地址与I2C_SLR1.SLADDR1[7:0]匹配一致。 清“0”条件: 检测到停止条件 SLADDR1FCLR写“1” 通信复位	R
b1	SLADDR0F	从机地址寄存器0匹配一致标志	0: 未检测到从机地址寄存器0一致地址 1: 检测到从机地址寄存器0一致地址 置“1”条件: 当I2C_SLR0.ADDRMOD0位为“0”时, 接收到的从机地址和I2C_SLR0.SLADDR0[7:1]匹配时。	R

当I2C_SLR0.ADDRMOD0位为“1”时，接收10位从机地址的
第一个字节地址与11110b+I2C_SLR0.SLADDR0[9:8]匹配一致
并且第二个字节地址与I2C_SLR0.SLADDR0[7:0]匹配一致。

清“0”条件：

检测到停止条件

SLADDR0FCLR写“1”

通信复位

b0	STARTF	开始条件/重新开始条件 标志位	0: 总线未检测到开始条件 1: 总线检测到开始条件 置“1”条件 1) 检测到开始条件 清“0”条件 1) 检测到停止条件 2) STARTFCLR写“1” 3) 通信复位	R
----	--------	--------------------	--	---

26.5.9 I2C 状态清零寄存器(I2C_CLR)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	SMB ALRT FCLR	SMB HOST FCLR	SMB DEFA ULTF CLR	GEN CALL FCLR	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	TMO UTFC LR	-	NAC KFCLR	-	-	ARLO FCLR	-	TEM PTYF CLR	RFU LLFC LR	-	STO PFC LR	TEN DFC LR	SLA DDR1 FCLR	SLAD DR0F CLR	STA RTFC LR

位	标记	位名	功能	读写
b31~b24	Reserved	-	写入时写“0”	W
b23	SMBALRTFCLR	SMBUS报警响应地址匹配一致标志清零位	写“1”清除SMBALRTF标志位	W
b22	SMBHOSTFCLR	SMBUS主机地址匹配一致标志清零位	写“1”清除SMBHOSTF标志位	W
b21	SMBDEFAULTFCLR	SMBUS默认地址匹配一致标志清零位	写“1”清除SMBDEFAULTF标志位	W
b20	GENCALLFCLR	广播呼叫地址匹配一致标志位	写“1”清除GENCALLF标志位	W
b19~b15	Reserved	-	写入时写“0”	W
b14	TMOUTFCLR	超时标志位	写“1”清除TMOUTF标志位	W
b13	Reserved	-	写入时写“0”	W
b12	NACKFCLR	NACK标志位	写“1”清除NACKF标志位	W
b11~b10	Reserved	-	写入时写“0”	W
b9	ARLOFCLR	仲裁失败标志位	写“1”清除ARLOF标志位	W
b8	Reserved	-	写入时写“0”	W
b7	TEMPTYFCLR	发送数据空标志位	写“1”清除TEMPTYF标志位	W
b6	RFULLFCLR	接收数据满标志位	写“1”清除RFULLF标志位	W
b5	Reserved	-	写入时写“0”	W
b4	STOPFCLR	停止条件标志位	写“1”清除STOPF标志位	W
b3	TENDFCLR	发送数据结束标志位	写“1”清除TENDF标志位	W
b2	SLADDR1FCLR	从机地址寄存器1匹配一致标志清零位	写“1”清除SLADDR1F标志位	W
b1	SLADDR0FCLR	从机地址寄存器0匹配一致标志清零位	写“1”清除SLADDR0F标志位	W
b0	STARTFCLR	开始条件/重新开始条件标志清零位	写“1”清除STARTF标志位	W

26.5.10 I2C 数据发送寄存器(I2C_DTR)

复位值: 0xFF

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

如果 I2C_DSR 寄存器为空，就将写在 I2C_DTR 寄存器的发送数据传送到 I2C_DSR 寄存器，发送模式时开始发送数据到 SDA 上。

I2C_DSR 寄存器和 I2C_DTR 寄存器是双缓冲结构，在 I2C_DSR 寄存器数据发送过程中，如果预先写 I2C_DTR 寄存器的数据，就能进行连续发送数据。

I2C_DRR 寄存器可读可写。请在发送数据空中断要求发生时，仅写一次 I2C_DTR 寄存器。

26.5.11 I2C 数据接收寄存器(I2C_DRD)

复位值: 0x00

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

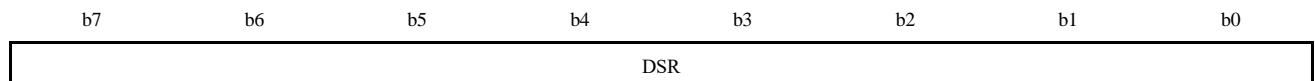
如果接收到 1 帧数据，就能将接收数据从移位寄存器（I2C_DSR）转存到 I2C_DRD 寄存器，进而可以进入到下一个数据接收状态。

I2C_DSR 寄存器和 I2C_DRD 寄存器是双缓冲结构，在 I2C_DSR 寄存器数据接收过程中，如果读取了 I2C_DRD 寄存器的数据，就能进行连续接收数据。

禁止对 I2C_DRD 寄存器写。请在接收数据满中断要求发生时，仅读一次 I2C_DRD 寄存器。

在 I2C_SR.RFULLF 标志位为“1”的状态下，如果不读取 I2C_DRD 寄存器的数据，而立即接收下一个数据，SCL 时钟就在下一次 RFULLF 标志位变“1”的前一个 SCL 时钟自动保持为低电平。

26.5.12 I2C 数据移位寄存器(I2C_DSR)



I2C_DSR 寄存器用于发送和接收数据的移位寄存器。I2C_DSR 寄存器不可读也不可写。

在数据发送时，将发送数据从 I2C_DTR 寄存器传送到 I2C_DSR 寄存器，从 SDA 引脚发送数据。在数据接收时，一旦 1 帧数据接收结束，就将数据从 I2C_DSR 寄存器传送到 I2C_DRR 寄存器。

26.5.13 I2C 时钟控制寄存器(I2C_CCR)

复位值: 0x0000_1F1F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	FREQ[2:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	SHIGHW[4:0]					-	-	-	SLOWW[4:0]				

位	标记	位名	功能	读写
b31~b23	Reserved	-	读出时为“0”,写入时写“0”	R/W
b22~b19	Reserved	-	读出时为“1”,写入时写“1”	R/W
b18-b16	FREQ[2:0]	I2C基准时钟频率设定位	0 0 0: I2C基准时钟频率=PCLK3/1 0 0 1: I2C基准时钟频率=PCLK3/2 0 1 0: I2C基准时钟频率=PCLK3/4 0 1 1: I2C基准时钟频率=PCLK3/8 1 0 0: I2C基准时钟频率=PCLK3/16 1 0 1: I2C基准时钟频率=PCLK3/32 1 1 0: I2C基准时钟频率=PCLK3/64 1 1 1: I2C基准时钟频率=PCLK3/128	R/W
b15~b13	Reserved	-	读出时为“1”,写入时写“1”	R/W
b12~b8	SHIGHW[4:0]	设定SCL高电平宽度位	设定SCL时钟的高电平宽度	R/W
b7~b5	Reserved	-	读出时为“1”,写入时写“1”	R/W
b4~b0	SLOWW[4:0]	设定SCL低电平宽度位	设定SCL时钟的低电平宽度	R/W

SHIGHW 位 (设定 SCL 高电平宽度位)

在主机模式下，SHIGHW 是用于设定 SCL 时钟的高电平宽度。在从机模式下，设定无效。

SLOWW 位 (设定 SCL 低电平宽度位)

SLOWW 是用于设定 SCL 时钟的低电平宽度。在从机模式下，设定值要大于数据准备时间。数据准备时间(tSU:DAT)250ns (~100kbps: 标准模式)100ns (~400kbps: 快速模式)

波特率：

DNFE=0,FREQ=000

波特率= $1/\{[(SHIGHW+3)+(SLOWW+3)]/\Phi I2C+SCL\text{ 上升时间}+SCL\text{ 下降时间}\}$

DNFE=1,FREQ=000

波特率= $1/\{[(SHIGHW+3+\text{滤波能力})+(SLOWW+3+\text{滤波能力})]/\Phi I2C+SCL\text{ 上升时间}+SCL\text{ 下降时间}\}$

DNFE=0,FREQ!=000

波特率= $1/\{[(SHIGHW+2)+(SLOWW+2)]/\Phi I2C+SCL\text{ 上升时间}+SCL\text{ 下降时间}\}$

DNFE=1,FREQ!=000

波特率= $1/\{[(SHIGHW+2+\text{滤波能力})+(SLOWW+2+\text{滤波能力})]/\Phi I2C+SCL\text{ 上升时间}+SCL\text{ 下降时间}\}$

注意：

- SCL 线的上升时间[tr] 和下降时间[tf] 取决于总线的总电容量[Cb] 和上拉电阻[Rp]，详细内容请参照 NXP 公司的 I2C 总线规格书。

26.5.14 I2C 滤波控制寄存器(I2C_FLTR)

复位值: 0x0000_0010

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

-	-	-	-	-	-	-	-	-	-	ANF EN	DNF EN	-	-	DNF[1:0]	
---	---	---	---	---	---	---	---	---	---	-----------	-----------	---	---	----------	--

位	标记	位名	功能	读写
b31~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5	ANFEN	模拟滤波功能允许位	0: 模拟滤波功能禁止 1: 模拟滤波功能允许	R/W
b4	DNFEN	数字滤波功能允许位	0: 数字滤波功能禁止 1: 数字滤波功能允许	R/W
b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1-b0	DNF[1:0]	数字滤波器滤波能力选择	00: 滤波能力1个I2C基准时钟周期 01: 滤波能力2个I2C基准时钟周期 10: 滤波能力3个I2C基准时钟周期 11: 滤波能力4个I2C基准时钟周期	R/W

27 串行外设接口 (SPI)

27.1 简介

本产品搭载 4 个通道的串行外设接口 SPI，支持高速全双工串行同步传输，方便地与外围设备进行数据交换。用户可根据需要进行三线/四线，主机/从机及波特率范围的设置。

SPI 主要特性：

要点	描述
通道数	4通道
串行通信功能	<ul style="list-style-type: none"> 支持4线式SPI模式和3线式时钟同步运行模式 支持全双工和只传送两种通信方式 可调整通信时钟SCK的极性和相位
数据格式	<ul style="list-style-type: none"> 可选择数据移位顺序:MSB开始/LSB开始 可选择数据宽度:4/5/6/7/8/9/10/11/12/13/14/15/16/20/24/32位 单次最多可传送或接收4帧宽度为32位的数据
波特率	<ul style="list-style-type: none"> 主机模式下可通过内置专用波特率发生器对波特率进行调整，波特率范围为PCLK的2分频~256分频 从机模式下允许的最大波特率为PCLK的6分频
数据缓冲	<ul style="list-style-type: none"> 带有16字节的数据缓冲区域 支持双重缓冲
错误监测	<ul style="list-style-type: none"> 模式故障错误监测 数据过载错误监测 数据欠载错误监测 奇偶校验错误监测
片选信号控制	<ul style="list-style-type: none"> 每个通道配置四根片选信号线 可对片选信号和通信时钟的相对时序关系进行调整 可对连续两次通信之间的片选信号无效时间进行调整 极性可调
主机模式下的 传输控制	<ul style="list-style-type: none"> 通过将数据写入数据寄存器启动传输 通信自动挂起功能

中断/AOS源	<ul style="list-style-type: none">接收数据区域已满发送数据区域已空SPI错误（模式/过载/欠载/奇偶校验）SPI闲置传输完成
低功耗控制	可设置模块停止
其他功能	<ul style="list-style-type: none">SPI初始化功能

表 27-1 SPI 的特性要点

注意：

- 在主接收模式使用通信自动挂起功能时，由于通信时钟停止，将不会发生过载错误。详情请参考 27.8.2 过载错误。

27.2 SPI 系统框图

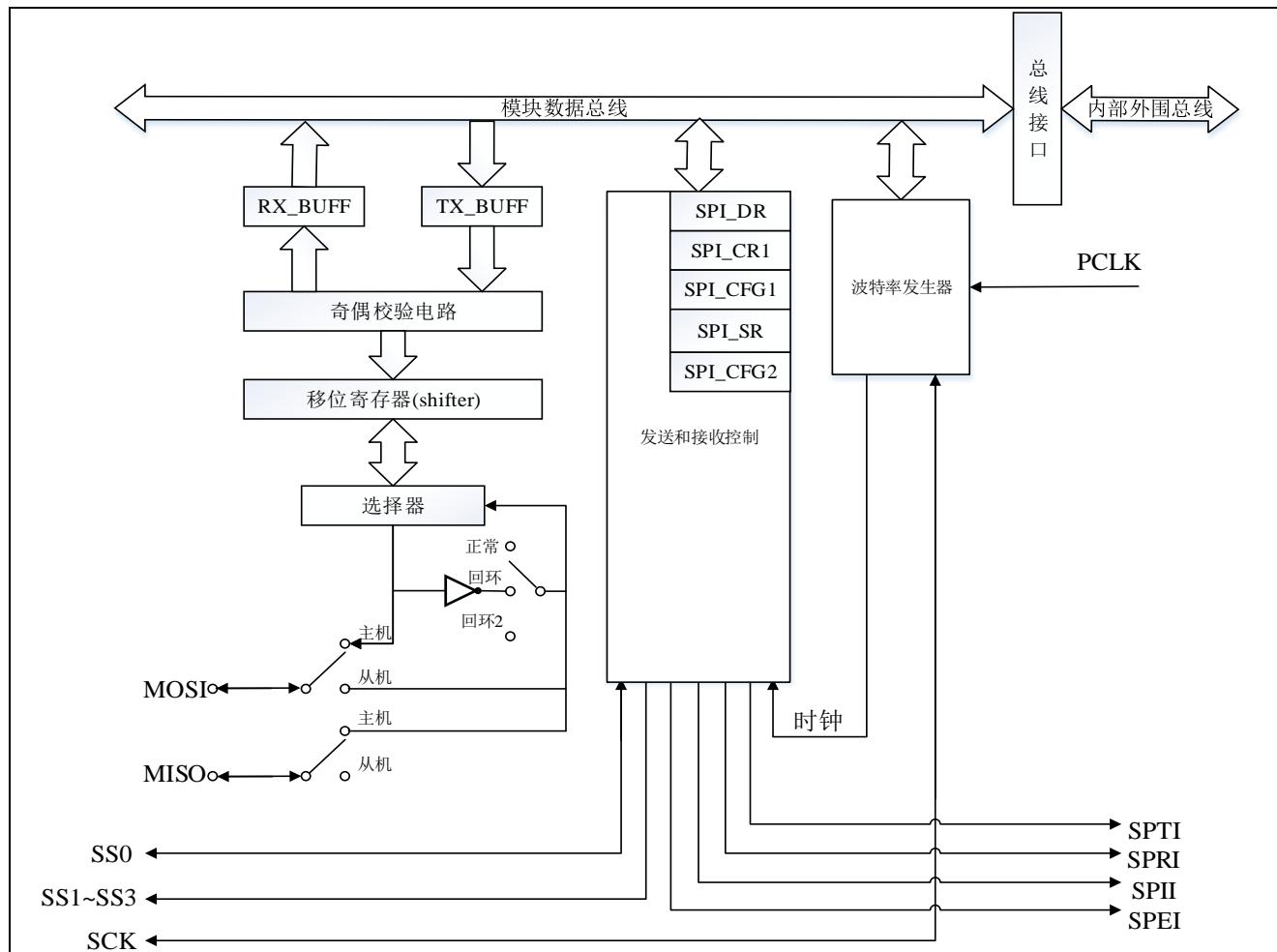


图 27-1 系统框图

27.3 管脚说明

管脚名	端口方向	功能
SCK	输入/输出	通信时钟管脚
MOSI	输入/输出	主机数据传输管脚
MISO	输入/输出	从机数据传输管脚
SS0	输入/输出	从机选择输入/输出管脚
SS1	输出	从机选择输出管脚
SS2	输出	从机选择输出管脚
SS3	输出	从机选择输出管脚

表 27-2 管脚说明

27.4 SPI 动作系统说明

27.4.1 主机模式的管脚状态

SPI 工作在主机模式下时，各个管脚的状态如下表所示。

模式		管脚名	管脚状态 (PFS.ODS=0)	管脚状态 (PFS.ODS=1)
SPI动作 (SPIMDS=0)	主机模式 (MSTR=1、 MODFE=0)	SCK	CMOS输出	OD输出
		SS0~SS3	CMOS输出	OD输出
		MOSI	CMOS输出	OD输出
		MISO	输入	输入
时钟同步运行 (SPIMDS=1)	主机模式 (MSTR=1)	SCK	CMOS输出	OD输出
		SS0~SS3 (不使 用)	Hi-Z (可作为通用 I/O)	Hi-Z (可作为通用 I/O)
		MOSI	CMOS输出	OD输出
		MISO	输入	输入

表 27-3 主机模式时 SPI 管脚状态说明

注意：

- 当 SS0 输入有效电平时，SPI 让出总线控制权，管脚状态为 Hi-Z。

27.4.2 从机模式的管脚状态

SPI 工作在从机模式下时，各个管脚的状态如下表所示。

模式		管脚名	管脚状态(PFS.ODS=0)	管脚状态(PFS.ODS=1)
SPI动作 (SPIMDS=0)	从机模式 (MSTR=0、 MODFE=0)	SCK	输入	输入
		SS0	输入	输入
		SS1~SS3 (不使用)	Hi-Z (可作为通用I/O)	Hi-Z (可作为通用I/O)
		MOSI	输入	输入
		MISO (注2)	CMOS输出/Hi-Z	OD输出/Hi-Z
时钟同步运行 (SPIMDS=1)	从机模式 (MSTR=0)	SCK	输入	输入
		SS0~SS3 (不使用)	Hi-Z (可作为通用I/O)	Hi-Z (可作为通用I/O)
		MOSI	输入	输入
		MISO	CMOS输出	OD输出

表 27-4 从机模式时 SPI 管脚状态说明

27.4.3 SPI 系统连接实例

主机模式

在主机-多从机模式的 SPI 系统结构中，主机驱动 SCK、MOSI 和 SS0 ~ SS3。在 SPI 从机设备 0~3 中，当某一从机的 SS 输入为有效电平时，该从机设备驱动 MISO。

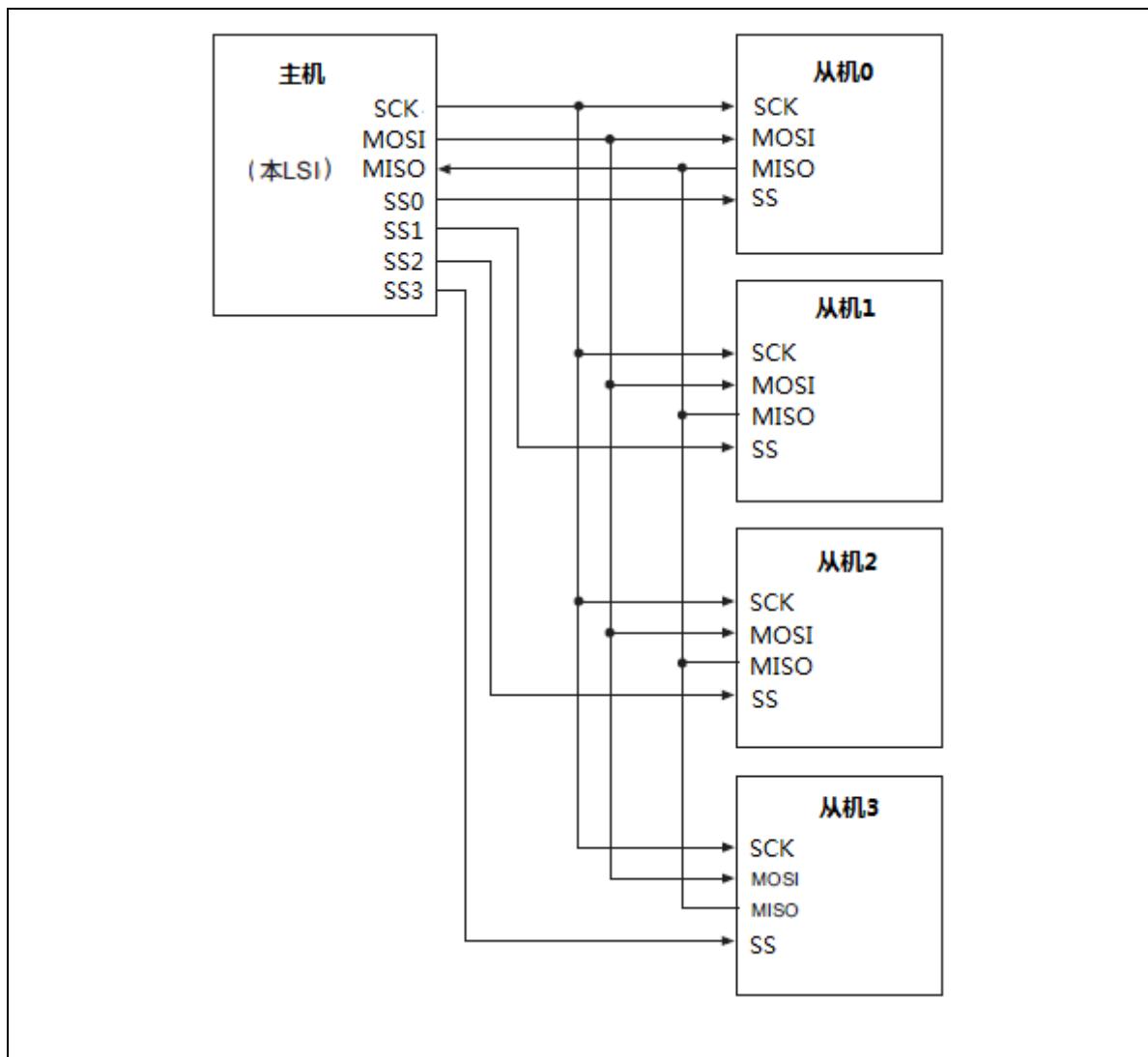


图 27-2 主机模式结构

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。

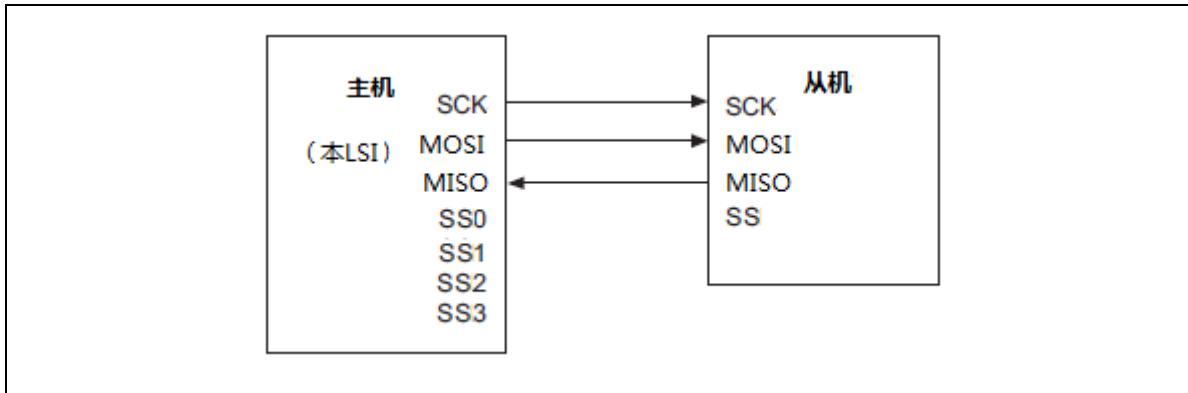


图 27-3 三线式时钟同步运行

27.5 数据通信说明

27.5.1 波特率

主机模式下，SPI 时钟由内部波特率发生器提供；从机模式下，时钟为 SCK 脚输入。波特率取决于 SPI_CFG2.MBR[2:0] 位的设置。其计算方法如下面公式所示，公式中 N 为 MBR[2:0] 位的设定值，范围是 0~7。

$$\text{波特率} = \frac{fpck}{2^{N+1}}$$

MBR[2:0]位的设定值	分频比	波特率			
		PCLK=5MHz	PCLK=10MHz	PCLK=20MHz	PCLK=40MHz
0	2	2.50Mbps	5.00Mbps	10.0Mbps	20.0Mbps
1	4	1.25Mbps	2.50Mbps	5.00Mbps	10.0Mbps
2	8	625kbps	1.25Mbps	2.50Mbps	5.00Mbps
3	16	313kbps	625kbps	1.25Mbps	2.50Mbps
4	32	156kbps	313kbps	625kbps	1.25Mbps
5	64	78kbps	156kbps	313kbps	625kbps
6	128	39kbps	78kbps	156kbps	313kbps
7	256	20kbps	39kbps	78kbps	156kbps

表 27-5 部分设定值下位速

27.5.2 数据格式

SPI 的数据格式取决于 SPI 命令寄存器 SPI_CFG2 和 SPI 控制寄存器 SPI_CR1 中的奇偶校验许可位 PAE 的设定值。SPI 将数据寄存器 SPI_DR 中从 LSB 位开始一定长度（数据长度由寄存器 SPI_CFG2 中的 DSIZE[3:0]位设置）的数据作为传送对象进行处理，与 MSB/LSB 移位顺序无关。

SPI_CFG2.DSIZE[3:0]决定数据的位宽，位宽范围为 4~32 位，SPI_CR1.PAE 决定数据最末位，PAE 为 1 时最末位作为奇偶校验位，为 0 时则是数据本身最低位。如图 28-4 所示。

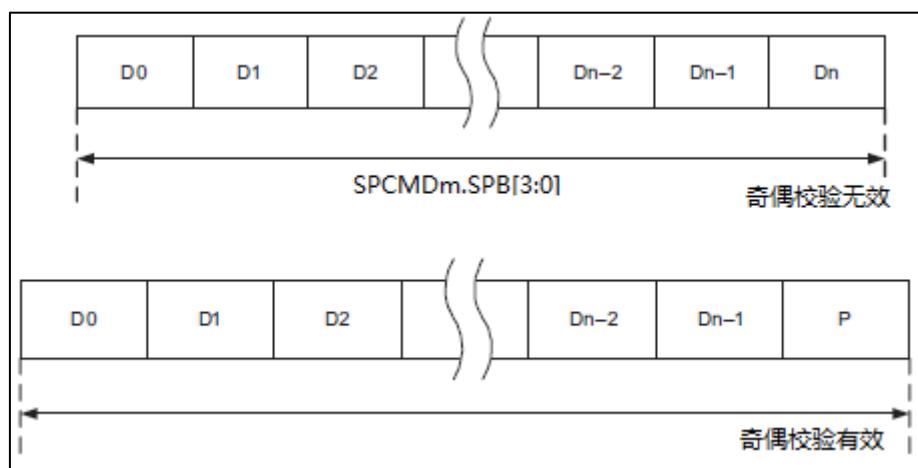


图 27-4 数据格式

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_CR1.PAE 的设置分为 4 种情况：

1) MSB 先传，奇偶校验无效

发送时，数据 d31~d0 按照顺序从 TX_BUFF 复制到 shifter，按照 d31~d0 的顺序从 shifter 的最高位移出；

接收时，数据 d31~d0 从 shifter 的最低位移入，等到数据全部移入后再将数据复制到 RX_BUFF。

2) LSB 先传，奇偶校验无效

发送时，数据 d31~d0 按照 d0~d31 的顺序从 TX_BUFF 复制到 shifter，按照 d0~d31

的顺序从 shifter 的最高位移出；

接收时，数据 d0~d31 从 shifter 的最低位移入，等到数据全部移入后，按照 d31~d0 的顺序从 shifter 复制到 RX_BUFF。

3) MSB 先传，奇偶校验有效时

发送时，先根据 d31~d1 的值计算出奇偶校验位 P 的值，然后用 P 代替 d0，按照 d31~d1，P 的顺序复制到 shifter，按照 d31~P 的顺序从 shifter 的最高位移出；

接收时，数据 d31~P 从 shifter 的最低位移入，在数据复制到 shifter 时，进行奇偶校验。最后再将数据复制到 RX_BUFF。

4) LSB 先传，奇偶校验有效时

发送时，先根据 d30~d0 的值计算出奇偶校验位 P 的值，然后用 P 代替 d31，按照 d0~P 的顺序从 TX_BUFF 复制到 shifter，按照 d0~P 的顺序从 shifter 的最高位移出；

接收时，数据 d0~P 从 shifter 的最低位移入，在数据复制到 shifter 时，进行奇偶校验。数据 d0~P 在复制时重新排列，按照 P~d0 的顺序复制到 RX_BUFF。

27.5.3 传送格式

1) CPHA=0 的情况

当 SPI_CFG2.CPHA 位为“0”时，SPI 在 SCK 的奇数边沿进行数据采样，偶数边沿进行数据更新。图 27-5 是 CPHA=0 时 SPI 的传送时序图。当 SS_i 信号的输入电平变为有效电平时，MOSI/MISO 开始更新传输数据。在 SS_i 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。

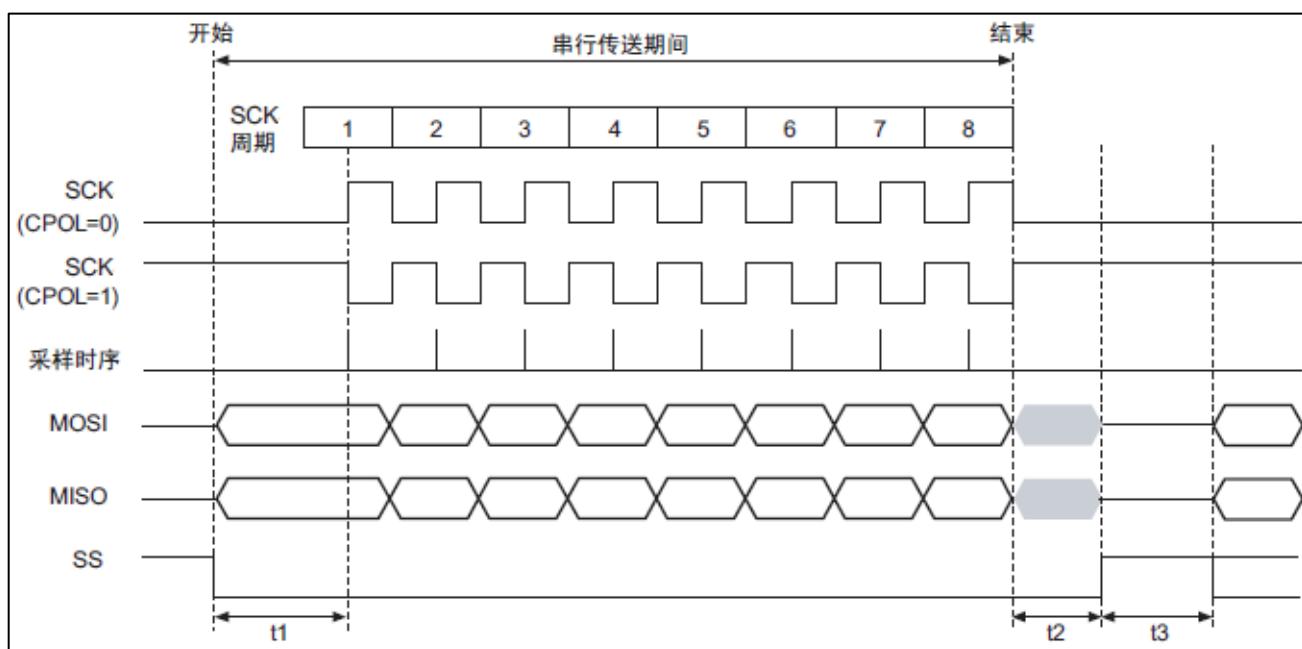


图 27-5 数据传送格式图 (CPHA=0)

上图中 t₁ 表示从 SS_i 信号有效到 SCK 振荡的间隔时间 (SCK 延迟时间)，t₂ 表示从 SCK 振荡停止到 SS_i 信号变成无效的间隔期间 (SS_i 无效延迟时间)，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)。t₁、t₂ 和 t₃ 由 SPI 系统上的主机设备进行控制。详细可参考 27.6.2 章节。

2) CPHA=1 的情况

当 SPI_CFG2.CPHA 位为“1”时，SPI 在 SCK 的奇数边沿进行数据更新，偶数边沿进行数据采样。图 27-6 是 CPHA=1 时 SPI 的传送时序图。MOSI/MISO 在 SS_i 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。

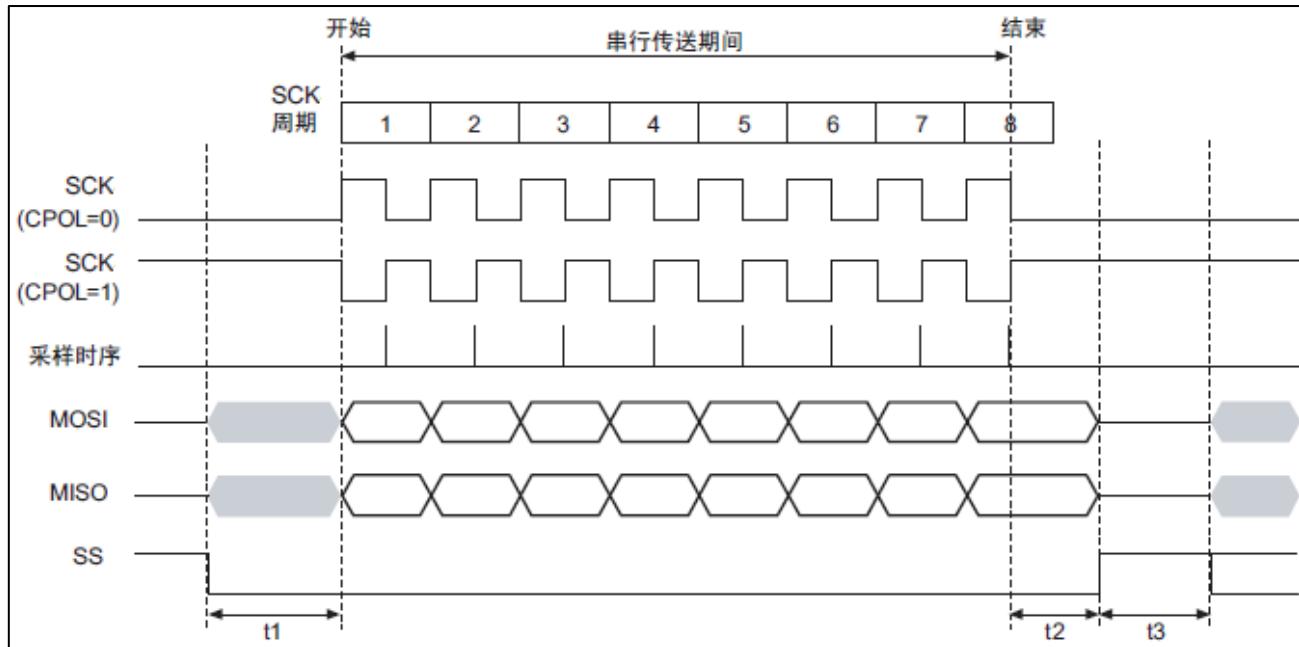


图 27-6 数据传送格式 (CPHA=1)

t1、t2、t3 和 CPHA 位为 0 的情况相同。

27.5.4 通信方式

本 SPI 有全双工同步串行通信和只发送串行通信两种通信方式，可通过 SPI 控制寄存器（SPI_CR1）的 TXMDS 位进行选择。

1) 全双工同步串行通信方式

当 SPI_CR1.TXMDS 位为“0”时，SPI 运行在全双工同步串行通信方式。如图 27-7 所示，SPI_CFG1.FTHLV[1:0] 位为 “00b”，SPI_CFG2.CPHA 位为 “1” 并且 SPI_CFG2.CPOL 位为“0”，SPI 进行 8 位串行传送。

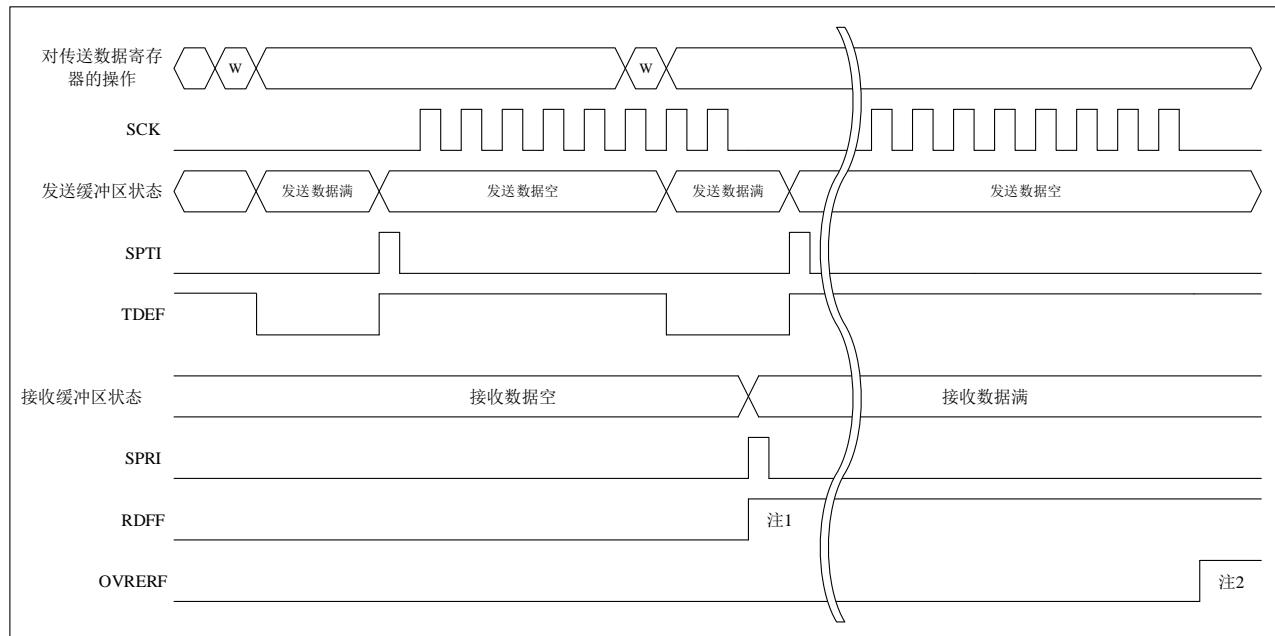


图 27-7 全双工同步串行通信

注：

1. 当本次串行传输结束时，如果接收数据缓冲寄存器为空，SPI 将会把接收到的数据从移位寄存器复制到接收数据缓冲寄存器中，接收数据缓冲寄存器满的标志位被置成 1（RDFF），并产生一个接收数据满的中断请求（SPRI）。
2. 当本次串行传输结束时，如果接收数据缓冲寄存器中还保持着上次收到的数据而没有被系统读取，SPI 会将数据过载标志位置成 1，本次数据接收无效，接收移位寄存器中的数据将被丢弃。

2) 只进行发送通信方式

当 SPI_CR1.TXMDS 位为“1”时，SPI 运行在只发送通信方式。如图 27-8 所示，SPI_CFG1.FTHLV[1:0]位为“00b”，SPI_CFG2.CPHA 位为“1”并且 SPI_CFG2.CPOL 位为“0”，SPI 进行 8 位串行传送。

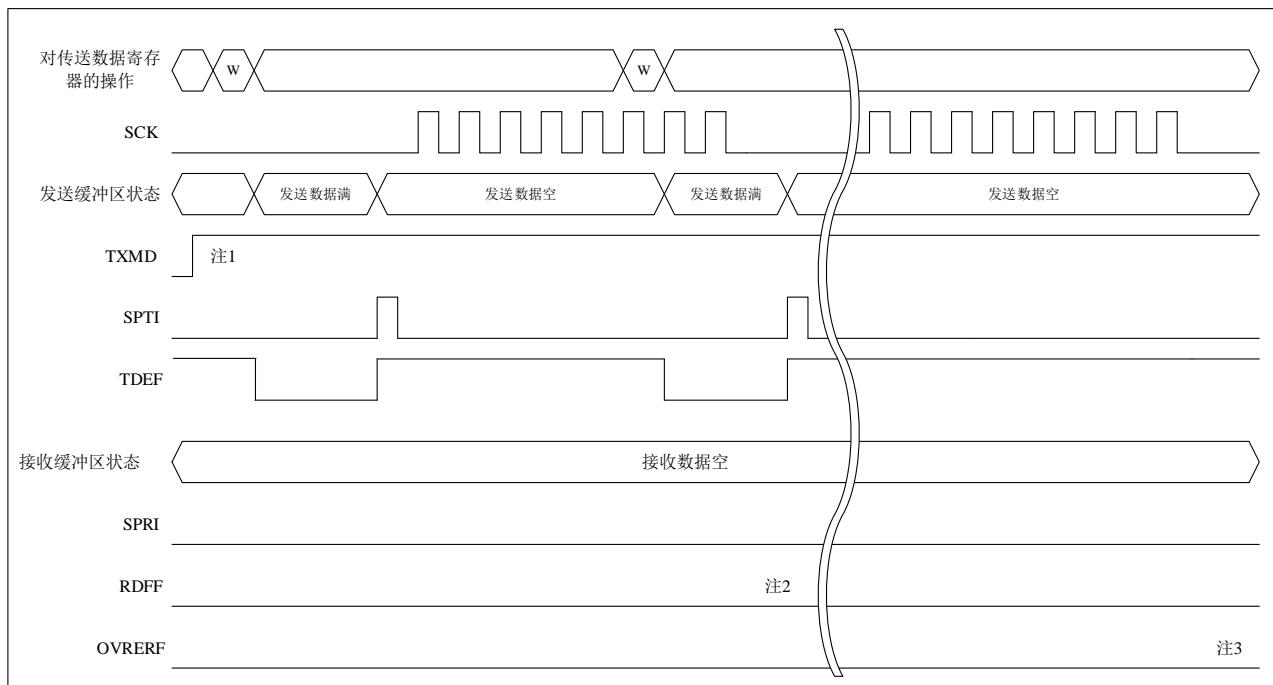


图 27-8 只进行发送通信

注：

1. 在设置进入只发送通信方式之前，请确保接收缓冲寄存器中没有未读取的数据（即 RDFF 为 0）且未出现数据过载错误（即 OVRERF 为 0）。
2. 在只发送通信方式下，当本次串行传输结束时，即使接收数据缓冲寄存器为空，也不会有数据被接受，RDFF 将始终保持 0 的状态。
3. 在只发送通信方式下，由于接收数据缓冲寄存器始终为空，所以不会发生数据过载错误，OVRERF 标志位始终保持 0 的状态。

27.6 运行说明

27.6.1 运行模式概要

本 SPI 支持 4 线式 SPI 模式和 3 线式时钟同步运行模式。每个运行模式下都可作为主机或者从机进行串行通信。设置 SPI 控制寄存器 SPI_CR1 中的 MSTR 和 SPIMDS 位设定 SPI 的模式。SPI 的模式和 SPI_CR1 寄存器设定的关系以及各模式概要如表 27-6 所示。

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	~PCLK/2	~PCLK/6	~PCLK/2	~PCLK/6
时钟源	内部波特率发生器	SCK输入	内部波特率发生器	SCK输入
时钟极性	2种	2种	2种	2种
时钟相位	2种	2种	2种	1种 (CHPA=1)
开始传送位	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB
传送数据长度	4~32位	4~32位	4~32位	4~32位
SCK延迟控制	有	无	有	无
SS无效延迟控制	有	无	有	无
下次存取延迟控制	有	无	有	无
传送启动方法	通过发送缓冲器空 中断请求, 写发送 缓冲器	SS输入有效或 SCK时钟边沿	通过发送缓冲器空 中断请求, 写发送 缓冲器	SCK振荡
发送缓冲器空检测	有	有	有	有

接收缓冲器满检测	有（注1）	有（注1）	有（注1）	有（注1）
过载错误检测	有（注1）	有（注1）	有（注1）	有（注1）
奇偶校验错误检测	有（注1、注2）	有（注1、注2）	有（注1、注2）	有（注1、注2）
欠载错误检测	有	无	有	无

表 27-6 SPI 模式和寄存器设定关系

注：

1. 当 SPI_CR1.TXMDS 位为“1”时，不进行接收缓冲器满的检测、过载错误的检测和奇偶校验错误的检测。
2. 当 SPI_CR1.PAE 位为“0”时，不进行奇偶校验错误的检测。

27.6.2 SPI 运行模式时的主机动作

1) 作为主机时的动作说明

在 SPI 数据发送缓冲寄存器 (TX_BUFF) 为空的状态下 (状态寄存器 SPI_SR 中的 TDEF 标志位为 1)，将格式控制寄存器 SPI_CFG1 的 FTHLV[1:0] 位所设帧数长度的数据写到 SPI 数据寄存器 (SPI_DR) 后，SPI 会将 SPI_DR 数据更新至 TX_BUFF 中。此时若移位寄存器 (shifter) 为空，SPI 将 TX_BUFF 的数据复制到移位寄存器开始串行传送。

当发送数据复制到 shifter 后，SPI 将会 shifter 状态改为满状态；当串行传送结束，改为空状态。shifter 的状态无法进行读取。

当 SPI 发送完最后采样时序所需的 SCK 边沿后，本次串行传送结束，结束时间与 SPI_CFG2.CPHA 位无关。在接收缓冲器 (RX_BUFF) 为空的情况下，串行传送结束后 SPI 将 shifter 中的数据复制到 RX_BUFF 中，可通过数据寄存器 SPI_DR 进行读取。

最后的采样时序取决于传送数据的位长，主控模式的 SPI 数据长度取决于 SPI_CFG2.DSIZE[3:0] 位的设定值，SS 输出引脚的极性取决于 SPI_CFG1 寄存器的设定值。有关 SPI 传送格式的详细内容，请参照 27.5.3 传送格式。

2) SPI 主机模式的初始化

- ① 设置通信配置寄存器 1 (SPI_CFG1)，包括有波特率的设定，使用帧数的设定，各种延迟时间的设定等。
- ② 设置通信配置寄存器 2 (SPI_CFG2)，包括有 SS 电平设定，数据移位顺序设定，各种延迟的允许位的设定，数据格式及时钟极性相位的设定等。
- ③ 如需要使用中断，请设置系统的中断寄存器。
- ④ 如需要使用 DMA，请设置 DMA 的相关寄存器。
- ⑤ 设定输入输出管脚。
- ⑥ 设定 SPI 控制寄存器 SPI_CR1，包括有模式及运行方式的设定，自诊断功能的设定，奇偶校验的设定等。
- ⑦ 确认 SPI_CR1 寄存器的设置。
- ⑧ 清除各种标志位。

- ⑨ 设置中断许可位。
- ⑩ 将控制寄存器 SPI_CR1 的 SPE 位设置成 1，动作开始。

27.6.3 SPI 运行模式时的从机动作

1) SPI 作为从机时的动作说明

当 SPI_CFG2.CPHA 位为 0 时，如果 SPI 检测到 SS0 输入信号变为有效电平，就需要开始向 MISO 输出信号驱动有效数据。因此，在 CPHA 位为 0 时，将 SS0 输入信号电平从无效变为有效视为开始串行传送的触发信号。

当 CPHA 位为“1”时，如果在 SS0 输入信号为有效电平的状态下 SPI 检测到最初的 SCK 边沿，就需要开始向 MISO 输出信号驱动有效数据。因此，在 CPHA 位为“1”时，将 SS0 信号处于有效电平状态下的首个 SCK 边沿视为开始串行传送的触发信号。

如果 SPI 在 shifter 为空的状态下检测到串行传送的开始，就将 shifter 改为满状态，并且不能在串行传送过程中将数据从 TX_BUFF 传送到 shifter。如果在开始串行传送前 shifter 已满，SPI 就保持 shifter 的满状态。

如果 SPI 检测到最后采样时序的 SCK 边沿，则本次串行传输结束，结束时间与 SPI_CFG2.CPHA 位无关。在 RX_BUFF 为空的情况下，串行传送结束后 SPI 将 shifter 的接受数据复制到 RX_BUFF。可通过访问 SPI_DR 读取该数据。SPI 在串行传送结束后将 shifter 改为空状态，该状态与 RX_BUFF 的状态无关。

在串行传送期间，如果 SPI 检测到 SS0 输入信号无效，就发生模式故障错误。

最后的采样时序取决于传送数据的位长，从属模式的 SPI 的数据长度取决于 SPI_CFG2.DSIZE[3:0] 位的设定值，SS0 输入信号的极性取决于 SPI_CFG1.SS0PV 位的设定值。有关 SPI 传送格式的详细内容，请参照 27.5.3 传送格式。

注意：

- 当 SPI_CFG2.CPHA 位为“0”时，将 SS0 输入信号电平从无效变为有效视为开始串行传送的触发信号。如果在单从机模式的结构中 SS0 输入信号被固定为有效状态，此时 SPI 将无法正常开始串行传送。因此，在 SS0 输入信号被固定为有效状态的结构中，要使从机模式的 SPI 正常进行发送和接收，必须将 CPHA 位设置为“1”。如果需要将 CPHA

位设成“0”，则不能固定 SS0 输入信号。

2) SPI 从机模式的初始化

- ① 设定通信配置寄存器 1 (SPI_CFG1)，主要包括使用帧数的设定。
- ② 设置通信配置寄存器 2 (SPI_CFG2)，包括有传输速率，数据格式及时钟极性相位的设定等。
- ③ 如需要使用中断，请设置系统的中断寄存器。
- ④ 如需要使用 DMA，请设置 DMA 的相关寄存器。
- ⑤ 设定输入输出管脚。
- ⑥ 设定 SPI 控制寄存器 SPI_CR1，包括有模式及运行方式的设定，自诊断功能的设定，奇偶校验的设定等。
- ⑦ 确认 SPI_CR1 寄存器的设置。
- ⑧ 清除各种标志位。
- ⑨ 设置中断许可位。
- ⑩ 将控制寄存器 SPI_CR1 的 SPE 位设置成 1，动作开始。

27.6.4 时钟同步运行模式中的主机动作

当 SPI 控制寄存器 SPI_CR1 中的 SPIMDS 位为 1 时，SPI 处于时钟同步运行模式。在该模式动作时，SPI 只使用 SCK、MOSI 和 MISO 这 3 个管脚进行通信，SSI 管脚被释放可用于普通 I/O 功能。

尽管时钟同步运行模式时不使用 SSI 管脚，但模块内部的运行和 SPI 运行模式是相同的。但由于没有了 SSI 的输入，所以检测不到模式故障错误。

1) SPI 作为主机时的动作说明

在 SPI 数据发送缓冲寄存器 (TX_BUFF) 为空的状态下（状态寄存器 SPI_SR 中的 TDEF 标志位为 0），将格式控制寄存器 SPI_CFG1 的 FTHLV[1:0] 位所设帧数长度的数据写到 SPI 数据寄存器(SPI_DR)后，SPI 会将 SPI_DR 数据更新至 TX_BUFF 中。此时若移位寄存器 (shifter) 为空，SPI 将 TX_BUFF 的数据复制到移位寄存器开始串行传送。

当发送数据复制到 shifter 后，SPI 将会 shifter 状态改为满状态；当串行传送结束，

改为空状态。shifter 的状态无法进行读取。

当 SPI 发送完最后采样时序所需的 SCK 边沿后，本次串行传送结束，结束时间与 SPI_CFG2.CPHA 位无关。在接收缓冲器 (RX_BUFF) 为空的情况下，串行传送结束后 SPI 将 shifter 中的数据复制到 RX_BUFF 中，可通过数据寄存器 SPI_DR 进行读取。

最后的采样时序取决于传送数据的位长，主控模式的 SPI 数据长度取决于 SPI_CFG2.DSIZE[3:0]位的设定值，SS 输出引脚的极性取决于 SPI_CFG1 寄存器的设定值。有关 SPI 传送格式的详细内容，请参照 27.5.3 传送格式。

2) 时钟同步运行模式时主机的初始化设置

- ① 设置通信配置寄存器 1 (SPI_CFG1)，包括有波特率的设定，使用帧数的设定，各种迟延时间的设定等
- ② 设置通信配置寄存器 2 (SPI_CFG2)，包括有数据移位顺序设定，各种延迟的允许位的设定，数据格式及时钟极性相位的设定等。
- ③ 如需要使用中断，请设置系统的中断寄存器。
- ④ 如需要使用 DMA，请设置 DMA 的相关寄存器。
- ⑤ 设定输入输出管脚。
- ⑥ 设定 SPI 控制寄存器 SPI_CR1，包括有模式及运行方式的设定，自诊断功能的设定，奇偶校验的设定等。
- ⑦ 确认 SPI_CR1 寄存器的设置。
- ⑧ 清除各种标志位。
- ⑨ 设置中断许可位。
- ⑩ 将控制寄存器 SPI_CR1 的 SPE 位设置成 1，动作开始。

27.6.5 时钟同步运行模式中的从机动作

1) SPI 作为从机时的动作说明

当 SPI_CFG2.CPHA 位为 0 时，需要 SPI 检测到 SS0 输入信号变为有效电平作为开始串行通信的触发信号。由于时钟同步运行模式下不使用 SS0 管脚，因此，在 CPHA

位为 0 时无法进行正常通信。

当 CPHA 位为“1”时，如果在 SS0 输入信号为有效电平的状态下 SPI 检测到最初的 SCK 边沿，就需要开始向 MISO 输出信号驱动有效数据。由于时钟同步运行模式下不适用 SS0 管脚，因此，在 CPHA 位为“1”时，将首个 SCK 边沿视为开始串行传送的触发信号。

如果 SPI 在 shifter 为空的状态下检测到串行传送的开始，就将 shifter 改为满状态，并且不能在串行传送过程中将数据从 TX_BUFF 传送到 shifter。如果在开始串行传送前 shifter 已满，SPI 就保持 shifter 的满状态。

如果 SPI 检测到最后采样时序的 SCK 边沿，则本次串行传输结束。在 RX_BUFF 为空的情况下，串行传送结束后 SPI 将 shifter 的接收数据复制到 RX_BUFF。可通过访问 SPI_DR 读取该数据。SPI 在串行传送结束后将 shifter 改为空状态，该状态与 RX_BUFF 的状态无关。

最后的采样时序取决于传送数据的位长，从属模式的 SPI 的数据长度取决于 SPI_CFG2.DSIZE[3:0] 位的设定值。

2) 时钟同步运行模式时从机的初始化设置

- ① 设定通信配置寄存器 1 (SPI_CFG1)，主要包括使用帧数的设定。
- ② 设置通信配置寄存器 2 (SPI_CFG2)，包括有传输速率，数据格式及时钟极性相位的设定等。
- ③ 如需要使用中断，请设置系统的中断寄存器。
- ④ 如需要使用 DMA，请设置 DMA 的相关寄存器。
- ⑤ 设定输入输出管脚。
- ⑥ 设定 SPI 控制寄存器 SPI_CR1，包括有模式及运行方式的设定，自诊断功能的设定，奇偶校验的设定等。
- ⑦ 确认 SPI_CR1 寄存器的设置。
- ⑧ 清除各种标志位。
- ⑨ 设置中断许可位。
- ⑩ 将控制寄存器 SPI_CR1 的 SPE 位设置成 1，动作开始。

27.6.6 几种 SPI 动作的处理流程

1) SPI 作为主机时的数据传送处理流程

- ① 等待数据发送缓冲寄存器空的中断或通过轮询方式确认数据发送缓冲寄存器处于空状态。
- ② 向数据寄存器 SPI_DR 写入要发送的数据。
- ③ 重复①②步骤直到最后一个数据发送完成。
- ④ 将发送数据寄存器空中断的允许位 TXIE 清零，同时将 SPI 闲置状态中断允许位 IDIE 设为 1。
- ⑤ 发送 SPI 闲置状态中断。
- ⑥ 将 SPE 置 0，停止 SPI 动作，同时将 IDIE 清零。

2) 数据接收处理流程

- ① 等待数据接收缓冲寄存器满的中断或者通过轮询方式确认数据接收缓冲寄存器处于满状态。
- ② 通过访问 SPI_DR 从接收缓冲寄存器读取数据。
- ③ 重复①②步骤直到最后一个接收数据被读取。
- ④ 将数据接收缓冲寄存器满的中断允许位 RXIE 清零。

3) 通信错误处理流程

- ① 等待通信错误中断或者通过轮询方式确认通信错误标志位（MODFERF/OVRERF/UDRERF/PERF）被置成 1。
- ② 确认 SS0 状态，排除模式故障错误。
- ③ 将 SPE 清零，停止 SPI 动作。
- ④ 将所有 SPI 中断允许位清零，屏蔽 SPI 中断。
- ⑤ 通过错误标志位确定通信错误种类，进行通讯错误处理。
- ⑥ 将错误标志位清零。
- ⑦ 启动 SPI，重新开始通信。

27.7 奇偶校验位自诊断

奇偶校验电路由传输数据的奇偶校验位和接收数据的错误检测部分构成。可按照下图所示流程使用自诊断功能对奇偶校验电路进行故障诊断。

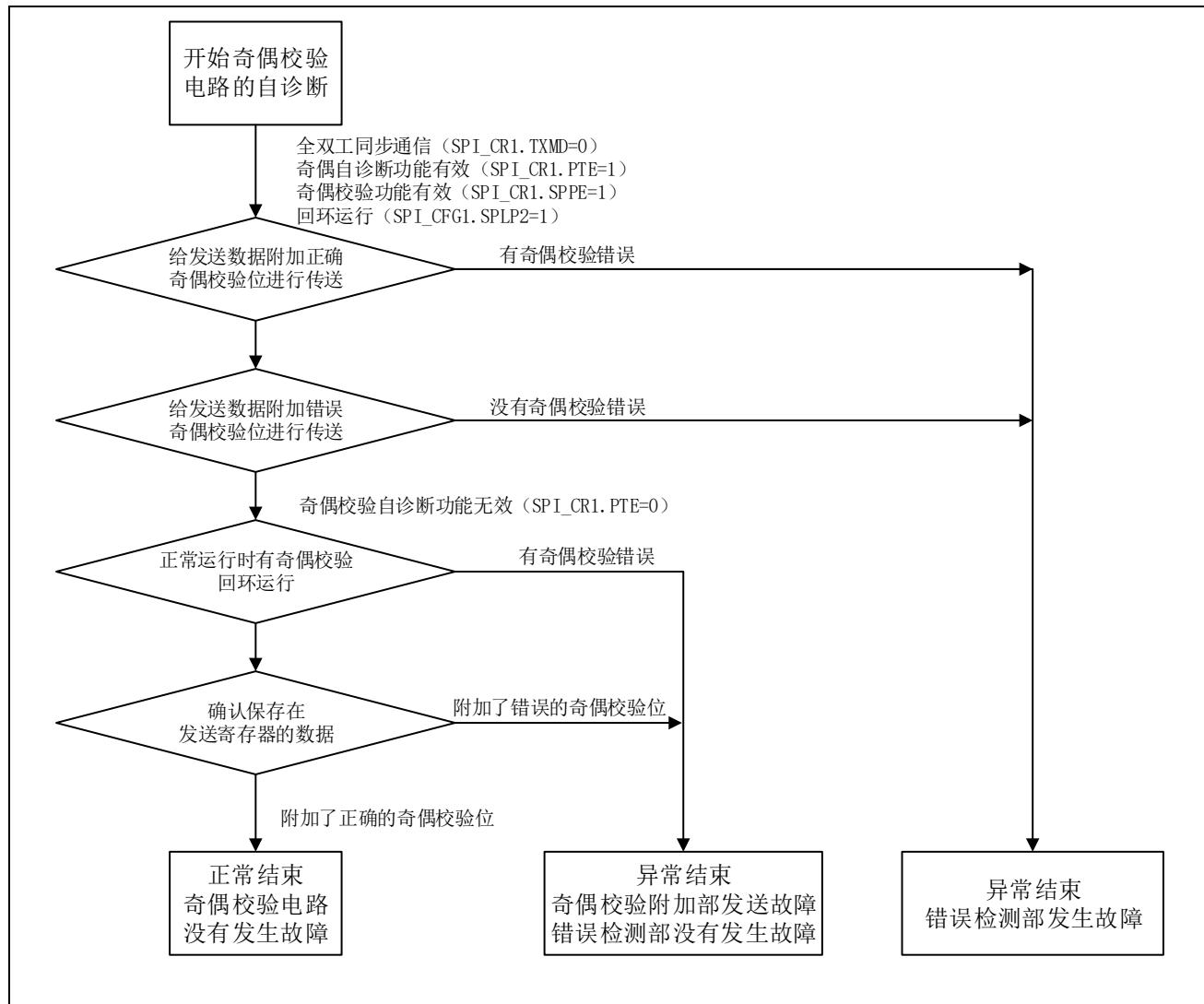


图 27-9 奇偶校验流程

27.8 错误检测

在正常的 SPI 串行传送中，系统通过对 SPI_DR 寄存器写操作进行数据串行发送，通过对 SPI_DR 寄存器的读操作获得串行接收的数据。但由于发送/接收缓冲器的状态以及串行传送开始/结束时的 SPI 的状态，某些情况下有可能会发生异常传送。当异常传送发生时，SPI 会将此次传输检测为欠载错误、过载错误、奇偶校验错误或者模式故障错误。异常传送和 SPI 错误检测的对应关系如下表 27-7 所示。

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	<ul style="list-style-type: none"> 保持发送缓冲器内容 写数据丢失 	无
②	在接收缓冲器为空的状态下读SPI_DR寄存器	输出上次的串行接收数据	无
③	在从机模式中，发送数据未转移到移位寄存器的状态下开始串行传送	<ul style="list-style-type: none"> 中止串行传输 丢失发送和接收数据 停止驱动MISO输出信号 停止SPI功能 	欠载错误
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。	<ul style="list-style-type: none"> 中止传输 丢失发送和接收数据 停止SPI功能 	模式错误
⑤	在接收缓冲器满的状态下结束串行传送	<ul style="list-style-type: none"> 保持接收缓冲器内容 接收数据丢失 	过载错误
⑥	在进行全双工同步串行通信，并且奇偶校验功能有效的情况下，接收到错误的奇偶校验位	奇偶校验错误标志有效	奇偶校验错误

表 27-7 错误检测对应表

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。

27.8.1 欠载错误

MSTR 位为 0 时，SPI 以从机状态运行，如果 SPE 置为 1 时，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_CR1.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_CR1.SPE 置成 1 使能 SPI 功能必须先将 MODFERF 标志清零。

27.8.2 模式错误

SPI 在主机模式时，SPI_CR1.MODFE 请勿设定为 1。在从机模式时，当 SS_i 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_CR1.SPE 置为 0。需要传送时，对 SPI_SR.MODFREF 清零后，再把 SPI_CR1.SPE 置 1。

27.8.3 过载错误

如果在接收缓冲器满的状态下结束串行传送，SPI 发生过载错误，SPI_SR.OVRERF 标志被置成 1。因为在 OVRERF 标志为 1 的状态下 SPI 不会把移位寄存器的数据复制到接收缓冲器，所以接收缓冲器中保存的是发生错误前的接收数据。要将 OVRERF 标志置 0 时，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”。

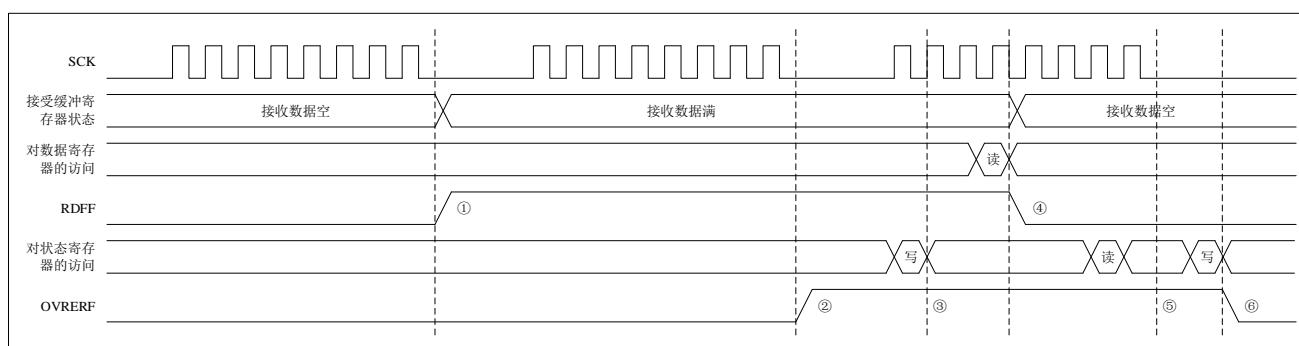


图 27-10 过载错误处理

以下说明在图中的①~⑥所示的时序中标志的运行内容：

- ① 在接受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受缓冲寄存器中，并将 RDFF 标志置 1。
- ② 在接收缓冲器满的状态下结束串行传送，SPI 检测到过载错误并且将 OVRERF 标志置 1。SPI 将不会把移位寄存器的数据复制到接收缓冲器。即使 PAE 位为“1”时，也不检测奇偶校验错误。
- ③ 在没有对 SPI_SR 寄存器进行读取的情况下，无法对 OVRERF 位进行写操作，清零失败。
- ④ 对数据寄存器 SPI_DR 进行读取访问，SPI 就能读到接收缓冲器的数据。RDFF 标志变为 0。即使此时接收缓冲器状态为空，OVRERF 标志也不会变为 0。
- ⑤ 在 OVRERF 标志为“1”的状态（过载错误）下结束串行传送，SPI 将不会把移位寄存器的数据复制到接收缓冲寄存器，也不产生接收缓冲器满中断，RDFF 标志保持为 0。即使 PAE 位为“1”，也不检测奇偶校验错误。在发生过载错误的状态下，如果在没有将接收数据从移位寄存器复制到接收缓冲器的情况下结束串行传送，SPI 就判断为移位寄存器处于空的状态，允许将数据从发送缓冲寄存器传送到移位寄存器。
- ⑥ 在 OVRERF 标志为 1 的状态下读 SPI_SR 寄存器后再给 OVRERF 标志写 0，SPI 就将 OVRERF 标志置 0。

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。

只有在将 OVRERF 标志变为 0 后才可以进行正常的接收运行。

在主机模式下如果启用通信自动挂起功能（将 SPI_CR1.CSUSPE 位设成 1），则 SPI 将在发生过载错误前的最后一个采样周期将通信时钟暂停，此时由于移位寄存器尚未完成最后一位的接收，SPI 保持在正常通信状态，过载错误不会发生。在通信时钟暂停期间可以对接收缓冲寄存器进行读取，读取后接收缓冲寄存器状态变为空，SPI 再重启通信时钟完成最后一位的数据接收。详细动作可参考下图 27-11 和图 27-12。

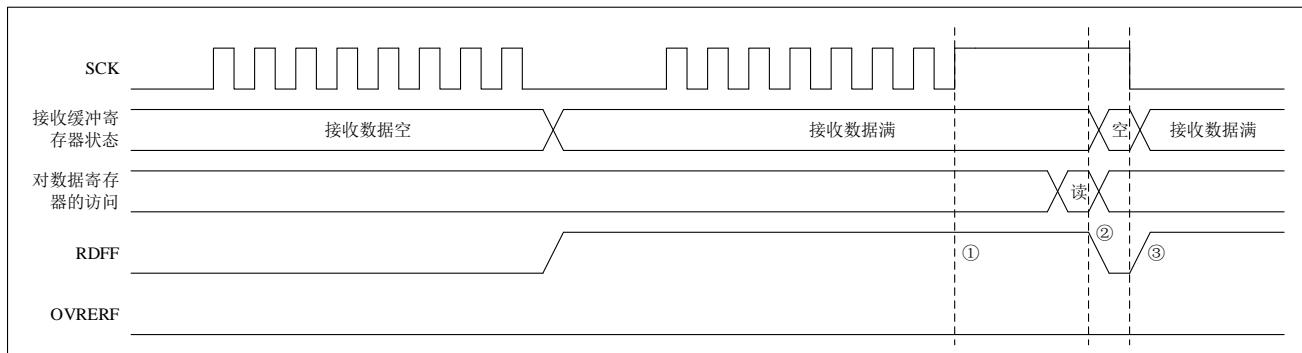


图 27-11 启用时钟自动停止功能时的动作示意图 (CPHA=1)

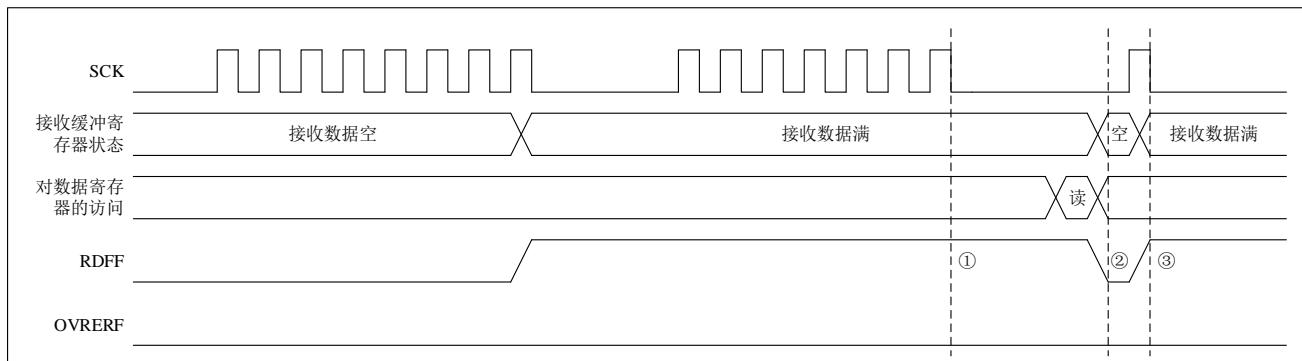


图 27-12 启用时钟自动停止功能时的动作示意图 (CPHA=0)

以下说明在图中的①~③所示的时序中标志的运行内容：

- ① 当处于接收缓冲寄存器满状态时，SPI 在最后一位数据接收前暂停通信时钟。此时将不会发生过载错误。
- ② 当通过访问 SPI_DR 读取接收缓冲寄存器中的数据后，接收缓冲寄存器变为空状态，RDFF 标志为清零，SPI 重启通信时钟完成最后一位的数据通信。
- ③ 最后一位数据通信完成，接收缓冲寄存器再次变为满状态，RDFF 标志置 1，可通过访问 SPI_DR 读取接收到的数据。

27.8.4 奇偶校验错误

在 SPI_CR1.TXMDS 位为“0”并且 SPI_CR1.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.OVRFERF 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 27-13 所示。在图中的例子中，SPI 在 SPI_CR1.TXMDS 位为 0 并且 SPI_CR1.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。

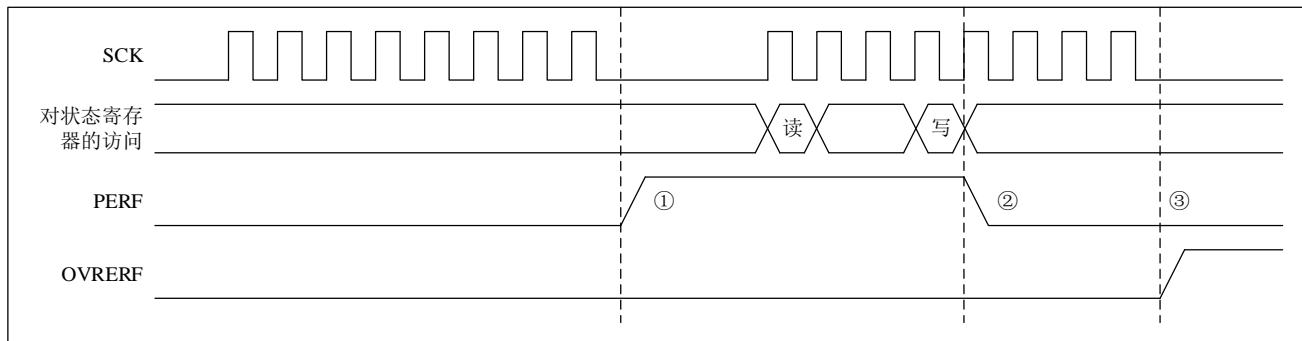


图 27-13 奇偶校验错误

以下说明在图中①~③所示的时序中标志的运行内容：

- ① SPI 未检测到过载错误，串行传送正常结束。SPI 将移位寄存器的数据复制到接收缓冲器。此时，SPI 对接收数据进行奇偶校验。如果检测到奇偶校验错误，则将 PERF 标志置 1。
- ② 在 PERF 标志为 1 的状态下读 SPI_SR 寄存器后给 PERF 标志写 0，将 RERF 标志清零。
- ③ SPI 检测到过载错误，此时 SPI 不会将移位寄存器的数据复制到接收缓冲器，SPI 不会对数据进行奇偶校验，不会发生奇偶校验错误。

可通过直接访问 SPI_SR 寄存器或者经 SPI 错误中断读取 SPI_SR 寄存器等方式监测奇偶校验错误的发生。在进行串行传送时，必须通过访问状态寄存器 SPI_SR 等方法，尽早监测到奇偶校验错误的发生。

27.9 SPI 的初始化

通过写操作或者模式故障错误检测将 SPE 位清零，可以使 SPI 功能无效并且对部分 SPI 功能进行初始化。如果发生系统复位，则对全部 SPI 功能进行初始化。

27.9.1 清除 SPE 位进行初始化

当 SPI_CR1.SPE 位为 0 时，SPI 进行以下的初始化操作：

- 中止正在进行的串行传送。
- 若处于从机状态则停止驱动输出信号（状态变为 Hi-Z）。
- 对 SPI 内部状态进行初始化。
- 清空发送缓冲寄存器，SPI_SR.TDEF 标志置为 1。

通过将 SPE 位清零来进行初始化时，不会对 SPI 的控制位进行初始化。因此，只要重新将 SPE 位置为 1，就能以和初始化前相同的传送模式启动 SPI。

清除 SPE 位不会对错误标志位和序列状态进行初始化。因此，在即使在 SPE 被清零以后，也能通过读取接收缓冲器的数据来确认 SPI 传送时的错误发生状况。

由于清除 SPE 位会清空发送缓冲寄存器，并将 SPI_SR.TDEF 标志置为 1。因此，如果在初始化后将 SPI_CR1.TXIE 位设成 1，就会产生 SPI 发送缓冲寄存器为空的中断。

要避免系统发生此中断，必须在将 SPE 位清零的同时也将 TXIE 位设成 0。

27.9.2 系统复位初始化

通过系统复位进行初始化，将对 SPI 所有的控制位、状态位和数据寄存器进行初始化。

27.10 中断源

SPI 的中断源有接收缓冲器满、发送缓冲器空、模式故障、过载、欠载、奇偶校验错误和 SPI 空闲。其中接收缓冲器满和发送缓冲器空中断可用于启动 DMA 进行数据传送。

过载、欠载和奇偶校验错误的中断被集成成为 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 中断源的具体说明如表 27-8 所示。一旦中断条件成立，就产生相应的中断请求。

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。在使用 DMA 进行发送或者接收时，必须先进行 DMA 的配置，DMA 设为动作许可的状态后再设定 SPI。

中断源	略称	中断条件	启动DMA
接收缓冲器满	SPRI	在SPI_CR1.RXIE位为“1”的状态下接收缓冲器变满时	能
发生缓冲器空	SPTI	在SPI_CR1.TXIE位为“1”的状态下发送缓冲器变空时	能
SPI错误（过载、欠载、奇偶校验错误）	SPEI	在SPI_CR1.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	不能
SPI空闲中断	SPII	在SPI_CR1.IDIE位为“1”的状态下IDLF标志变为“0”时	不能

表 27-8 SPI 中断源说明

27.11 可供使用的事件触发源

SPI 产生的可供使用的事件触发源主要有以下几种：

- 数据发送缓冲寄存器空
- 数据接收缓冲寄存器满
- SPI 通信错误（包括过载，欠载，奇偶校验等错误）
- SPI 处于闲置状态
- SPI 通讯结束

用户可将上述事件触发源对应的向量写入不同的触发对象寄存器实现各种事件触发功能。

上述事件触发源对应的向量请参考【中断控制器（INTC）】。

27.12 寄存器说明

寄存器一览

寄存器基准地址: SPI1_BASE:0x4001_C000; SPI2_BASE:0x4001_C400

SPI3_BASE:0x4002_0000; SPI4_BASE:0x4002_0400

寄存器名	偏移地址	复位值
SPI数据寄存器SPI_DR	0x00	0x0000_0000
SPI控制寄存器SPI_CR1	0x04	0x0000_0000
SPI通信配置寄存器1 SPI_CFG1	0x0C	0x0000_0010
SPI状态寄存器SPI_SR	0x14	0x0000_0020
SPI通信配置寄存器2 SPI_CFG2	0x18	0x0000_0F1D

27.12.1 SPI 数据寄存器 (SPI_DR)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b0	SPD[31:0]	串行数据	SPI数据存储	R/W

27.12.2 SPI 控制寄存器 (SPI_CR1)

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PAE	PAOE	PATE	MODFE	IDIE	RXIE	TXIE	EIE	CSUSPE	SPE	SPLPBK2	SPLPBK	MSTR	-	TXMDS	SPIMDS

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15	PAE	奇偶校验允许	0: 发送数据不附加奇偶校验位, 接收数据不进行奇偶校验 1: 发送数据附加奇偶校验位, 接收数据进行奇偶校验 (SPI_CR1.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_CR1.TXMDS=1)	R/W
b14	PAOE	奇偶校验模式选择	0: 选择偶校验进行发送和接收 1: 选择奇校验进行发送和接收	R/W
b13	PATE	奇偶校验自诊断	0: 奇偶校验自诊断功能无效 1: 奇偶校验自诊断功能有效	R/W
b12	MODFE	模式故障错误检测允许	0: 禁止模式故障错误检测 1: 允许模式故障错误检测	R/W
b11	IDIE	SPI空闲中断允许	0: 禁止空闲中断请求产生 1: 允许空闲中断请求产生	R/W
b10	RXIE	SPI接收中断允许	0: 禁止SPI接收中断请求产生 1: 允许SPI接收中断请求产生	R/W
b9	TXIE	SPI发送中断允许	0: 禁止SPI发送中断请求产生 1: 允许SPI发送中断请求产生	R/W
b8	EIE	SPI错误中断允许	0: 禁止SPI错误中断请求产生 1: 允许SPI错误中断请求产生	R/W
b7	CSUSPE	通讯自动挂起功能允许	0: 通讯自动挂起功能无效 1: 通讯自动挂起功能有效	R/W
b6	SPE	SPI功能允许	0: SPI功能无效 1: SPI功能有效	R/W
b5	SPLPBK2	SPI回环2位	0: 正常模式 1: 回环模式 (发送数据=接收数据)	R/W
b4	SPLPBK	SPI回环位	0: 正常模式 1: 回环模式 (发送数据的反相=接收数据)	R/W
b3	MSTR	SPI主从模式选择	0: 从机模式 1: 主机模式	R/W
b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	TXMDS	通信模式选择	0: 全双工同步串行通信	R/W

			1: 只进行发送串行通信	
b0	SPI MDS	SPI模式选择	0: SPI运行 (4线式) 1: 时钟同步运行 (3线式)	R/W

27.12.3 SPI 通信配置寄存器 1 (SPI_CFG1)

复位值: 0x0000_0010

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	MIDI[2:0]	-		MSSDL[2:0]	-		MSSI[2:0]	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	SS3P V	SS2P V	SS1P V	SS0P V	-	SPR DTD	-	-	-	-	-	FTHLV[1:0]

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“0”,写入时写“0”	R/W
b30~b28	MIDI[2:0]	主机下次存取数据间隔闲置时间设定位	b30~b28 0 0 0: 1 个SCK+2个SCK 0 0 1: 2 个SCK+2个SCK 0 1 0: 3 个SCK+2个SCK 0 1 1: 4 个SCK+2个SCK 1 0 0: 5 个SCK+2个SCK 1 0 1: 6 个SCK+2个SCK 1 1 0: 7 个SCK+2个SCK 1 1 1: 8 个SCK+2个SCK	R/W
b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26~b24	MSSDL[2:0]	主机SS无效延迟设定位	b26~b24 0 0 0: 1 个SCK 0 0 1: 2 个SCK 0 1 0: 3 个SCK 0 1 1: 4 个SCK 1 0 0: 5 个SCK 1 0 1: 6 个SCK 1 1 0: 7 个SCK 1 1 1: 8 个SCK	R/W
b23	Reserved	-	读出时为“0”,写入时写“0”	R/W
b22~b20	MSSI[2:0]	主机SS闲置时间设定位	b22~b20 0 0 0: 1 个SCK 0 0 1: 2 个SCK 0 1 0: 3 个SCK 0 1 1: 4 个SCK 1 0 0: 5 个SCK 1 0 1: 6 个SCK 1 1 0: 7 个SCK 1 1 1: 8 个SCK	R/W
b19~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W

b11	SS3PV	SS3信号 极性设定	0: SS3信号的Low电平有效 1: SS3信号的High电平有效	R/W
b10	SS2PV	SS2信号 极性设定	0: SS2信号的Low电平有效 1: SS2信号的High电平有效	R/W
b9	SS1PV	SS1信号 极性设定	0: SS1信号的Low电平有效 1: SS1信号的High电平有效	R/W
b8	SS0PV	SS0信号 极性设定	0: SS0信号的Low电平有效 1: SS0信号的High电平有效	R/W
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6	SPRD TD	数据寄存器读取对象 选择	0: SPI_DR 读接收缓冲器 1: SPI_DR 读发送缓冲器(必须在TDEF=1时才可以读取)	R/W
b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4	Reserved	-	读出时为“1”,写入时写“1”	R/W
b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1~b0	FTHLV[1:0]	帧数设定位	b1~b0 0 0: 1 帧 0 1: 2 帧 1 0: 3 帧 1 1: 4 帧	R/W

27.12.4 SPI 状态寄存器 (SPI_SR)

复位值: 0x0000_0020

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	RDF F	-	TDE ERF	UDR PERF	MOD FERF	IDLN F	OVR ERF	-

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	RDFF	接收缓冲器满标志	0: SPI_DR寄存器无数据 1: SPI_DR寄存器有数据 硬件置位、清零, 写入时写“1”	R
b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5	TDEF	发送缓冲器空标志	0: 发送缓冲器有数据 1: 发送缓冲器无数据 硬件置位、清零, 写入时写“1”	R
b4	UDRERF	欠载错误标志	0: 模式故障错误发生 (MODFERF=1) 1: 欠载错误发生 (MODFERF=1) 当MODFERF=0, 此位被初始化 硬件置位后, 读取1写0, 状态位清零	R/W
b3	PERF	奇偶校验错误标志	0: 未发生奇偶校验错误 1: 发生奇偶校验错误 硬件置位后, 读取1写0, 状态位清零	R/W
b2	MODFERF	模式故障错误标志	0: 未发送模式故障错误 1: 发生模式故障错误 硬件置位后, 读取1写0, 状态位清零	R/W
b1	IDLNF	SPI空闲标志	0: SPI为空闲状态 1: SPI为传送状态 硬件置位, 清零	R
b0	OVRERF	过载错误标志	0: 未发生过载错误 1: 发生过载错误 硬件置位后, 读取1写0, 状态位清零	R/W

27.12.5 SPI 通信配置寄存器 2 (SPI_CFG2)

复位值: 0x0000_0F1D

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MSSI E	MSS DLE	MIDI E	LSBF	DSIZE[3:0]		SSA[2:0]		MBR[2:0]		CPO L	CPH A				

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15	MSSIE	SCK延迟允许	0: SCK 延迟为1 个SCK 1: SCK 延迟为MSSI的设定值	R/W
b14	MSSDLE	SS无效延迟允许	0: SS 无效延迟为1 个SCK 1: SS 无效延迟为MSSDLE的设定值	R/W
b13	MIDIE	SPI下次存取延迟允许	0: 下次存取延迟为1 个SCK+2个PCLK 1: 下次存取延迟为MIDI的设定值	R/W
b12	LSBF	SPI LSB first 位	0: MSB first 1: LSB first	R/W
b11~b8	DSIZE[3: 0]	SPI数据长度设定位	b11~b8 0 0 0 0: 4位 0 0 0 1: 5位 0 0 1 0: 6位 0 0 1 1: 7位 0 1 0 0: 8位 0 1 0 1: 9位 0 1 1 0: 10位 0 1 1 1: 11位 1 0 0 0: 12位 1 0 0 1: 13位 1 0 1 0: 14位 1 0 1 1: 15位 1 1 0 0: 16位 1 1 0 1: 20位 1 1 1 0: 24位 1 1 1 1: 32位	R/W
b7~b5	SSA[2: 0]	SS信号有效设定位	b7~b5 0 0 0: SS0 0 0 1: SS1 0 1 0: SS2 0 1 1: SS3	R/W

			1 x x: 禁止设定	
b4~b2	MBR[2: 0]	位速率分频设定位	b4~b2 0 0 0: 选择基本位速率的2分频 0 0 1: 选择基本位速率的4分频 0 1 0: 选择基本位速率的8分频 0 1 1: 选择基本位速率的16分频 1 0 0: 选择基本位速率的32分频 1 0 1: 选择基本位速率的64分频 1 1 0: 选择基本位速率的128分频 1 1 1: 选择基本位速率的256分频	R/W
b1	CPOL	SCK极性设定位	0: 空闲时的SCK为Low电平 1: 空闲时的SCK为High电平	R/W
b0	CPHA	SCK相位设定位	0: 在奇数边沿进行数据采样, 在偶数边沿数据发生变化 1: 在奇数边沿数据发生变化, 在偶数边沿进行数据采样	R/W

28 四线式串行外设接口 (QSPI)

28.1 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。

参数	规格
通道数	1通道
SPI	<ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • 支持多种读取方式 <ul style="list-style-type: none"> a. 标准读/快速读 b. 二线式输出快速读取/二线式输入输出快速读取 c. 四线式输出快速读取/四线式输入输出快速读取
闪存读取	<ul style="list-style-type: none"> • 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

表 28-1 QSPI 主要规格

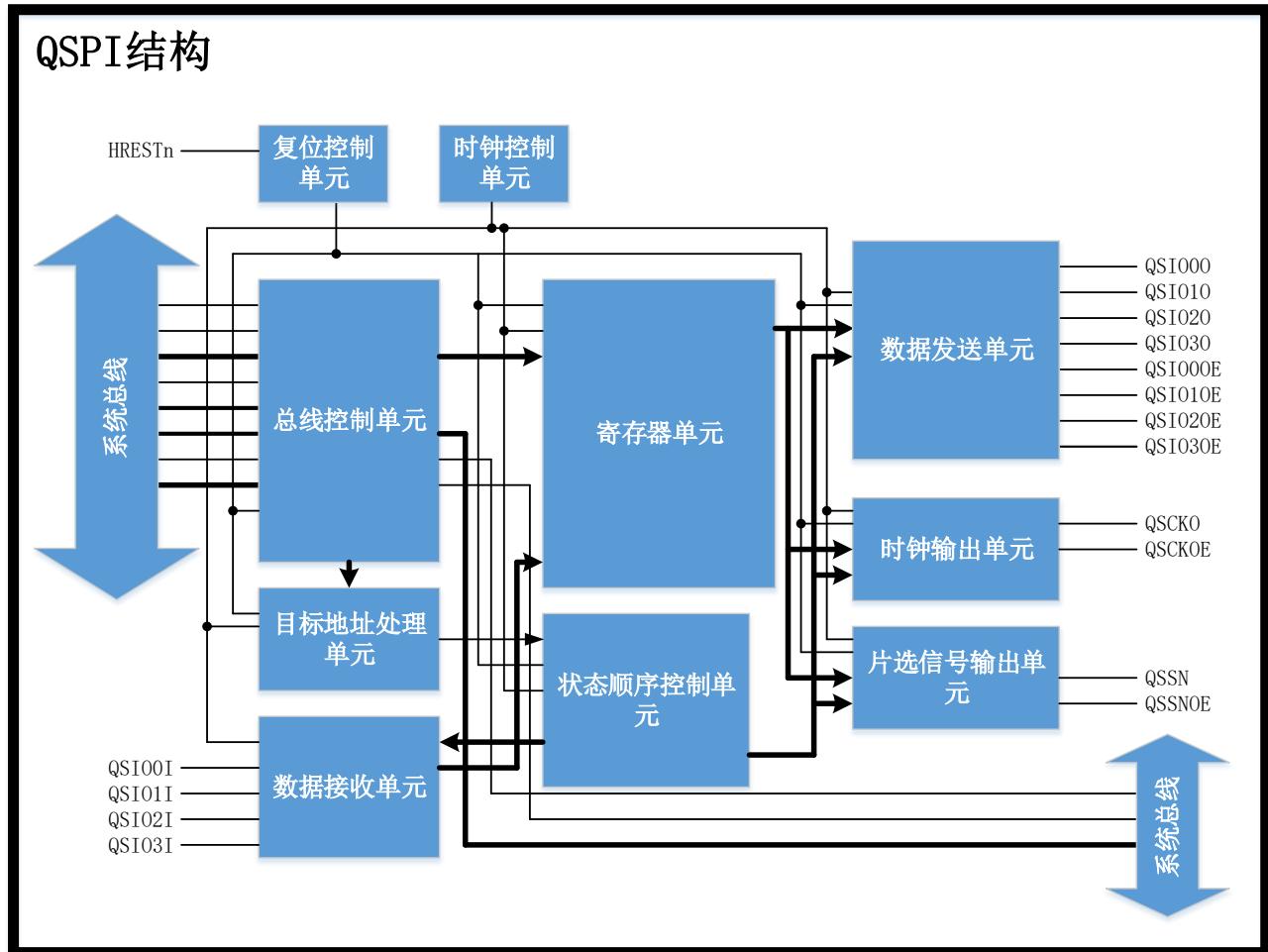


图 28-1 QSPI 的模块构成图

管脚名	输入/输出	功能描述
QSCK	输出	QSPI时钟输出管脚
QSSN	输出	QSPI从机选择管脚
QSI00	输入/输出	数据线0
QSI01	输入/输出	数据线1
QSI02	输入/输出	数据线2
QSI03	输入/输出	数据线3

表 28-2 QSPI 管脚

28.2 内存映射

28.2.1 内部总线空间

串行闪存及相关的控制寄存器在 AHB 总线空间的位置由总体的地址范围配置来决定。

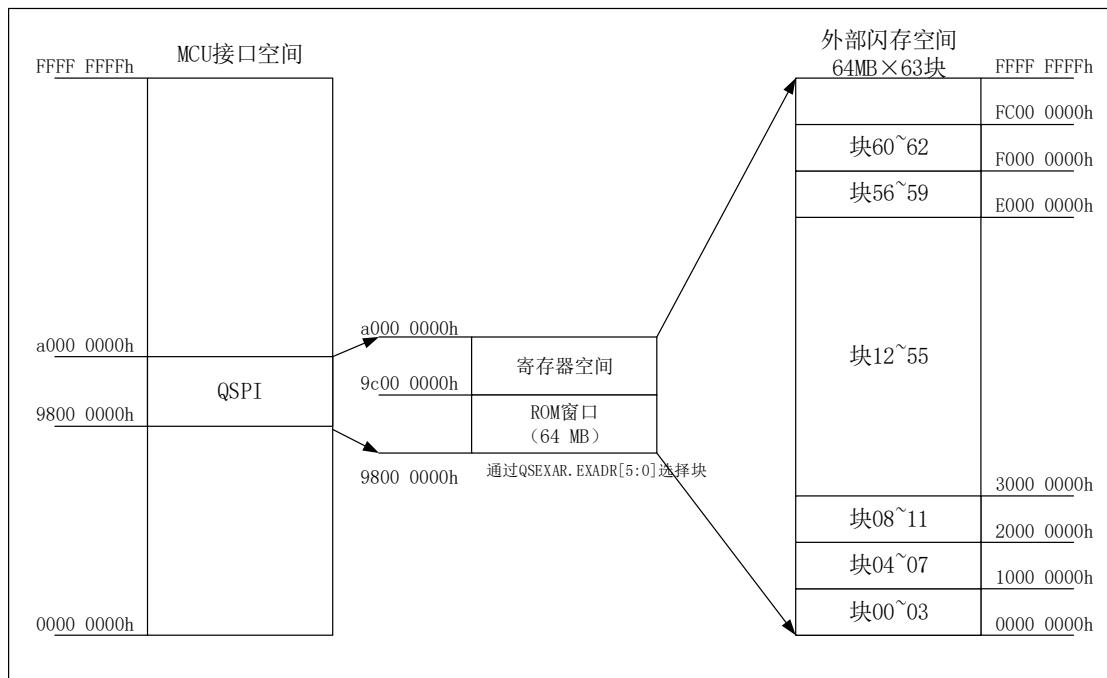


图 28-2 默认区域设定及 AHB 总线空间内存映射关系图

28.2.2 ROM 空间及总线的地址宽度

QSPI 的拥有 32 位地址总线宽度来配合串行闪存。每当对 QSPI 的 ROM 空间进行读访问时，QSPI 总线自动开始工作，将从串行闪存内读到的数据传送过来。

QSPI 不仅仅只能使用 32 位地址总线宽度，还可以通过设置 QSFCR 寄存器内的 AWSL[1:0]来选择使用 8 位/16 位/24 位地址总线宽度。

如果选择了 8 位/16 位/24 位的地址总线宽度，那么只有地址与之匹配的低位空间可以被正常访问，也就是说，访问 QSPI 中高位的串行闪存镜像空间将会反复出现低位空间的内容。

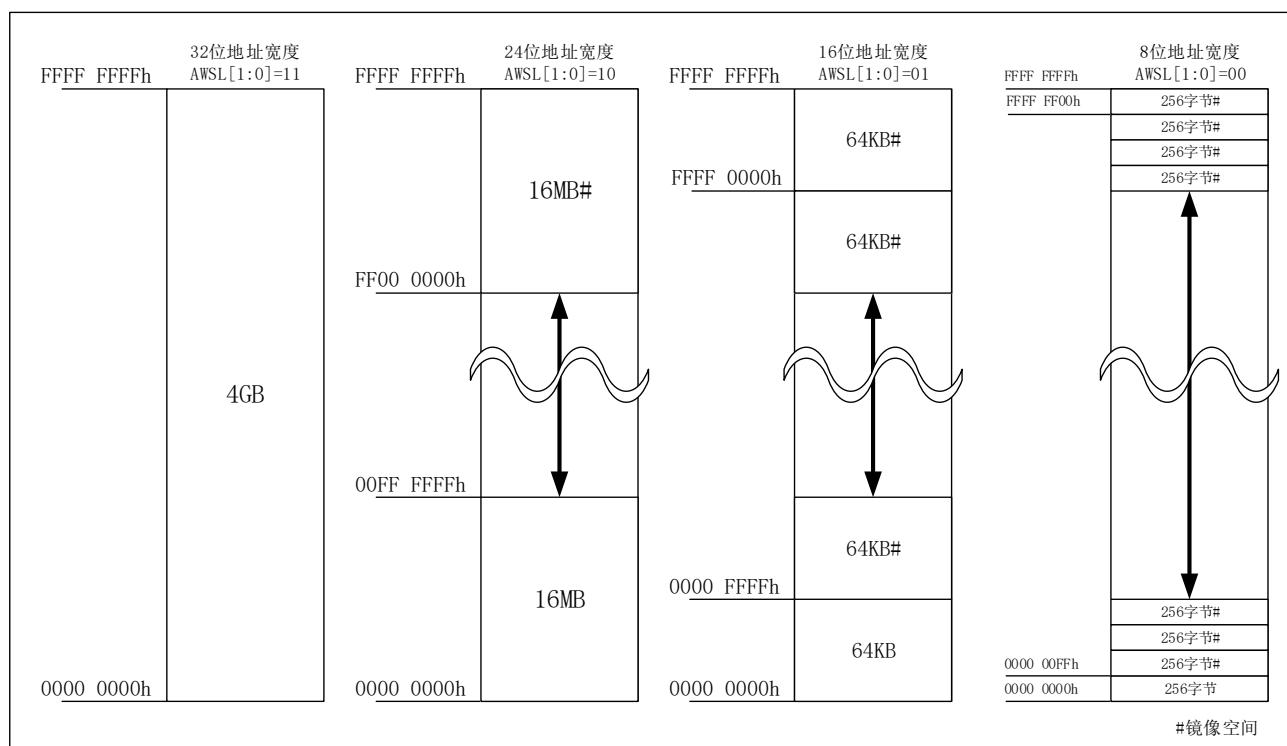


图 28-3 QSPI-ROM 空间内存映像图

注意：

- 地址总线宽度可以通过设置 QSFCR 寄存器内的 AWSL[1:0]来选择使用 8 位/16 位/24 位/32 位。

28.3 QSPI 总线

28.3.1 SPI 协议

本 QSPI 支持扩展式 SPI，二线式 SPI 和四线式 SPI 三种协议。初始的默认协议是扩展式 SPI 协议。可通过设置 QSCR 寄存器中的 DPRSL[1:0]/APRSL[1:0]/IPRSL[1:0]位来分别配置各个阶段的协议。扩展式 SPI 协议只用 QSIO0 管脚单线进行指令输出，之后的地址及数据则根据具体的读取模式指令使用单线式/二线式/四线式输出。

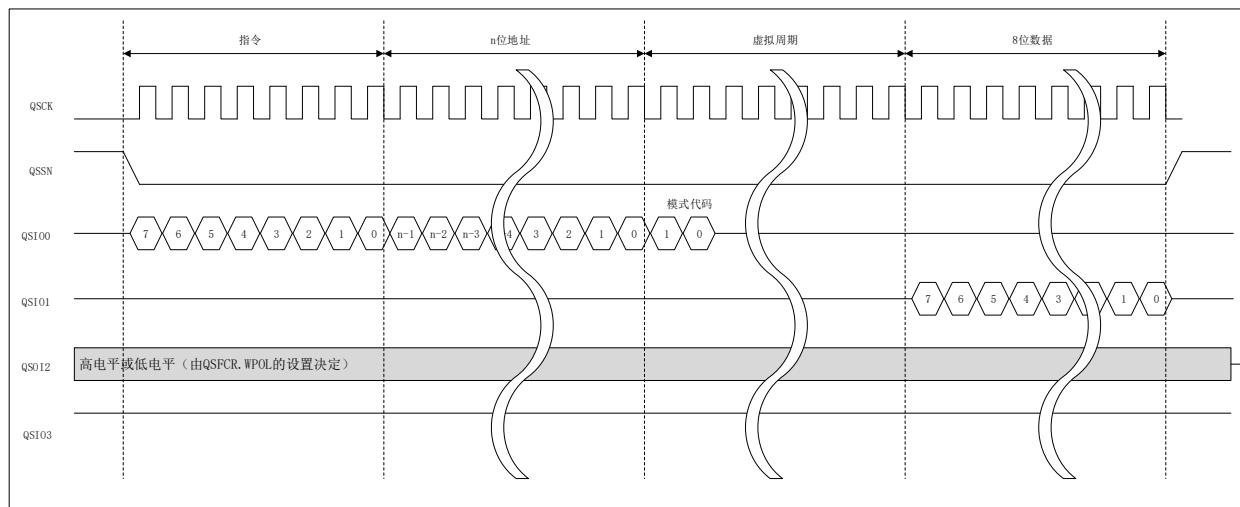


图 28-4 扩展式 SPI 协议动作示意图 1 (快速读模式)

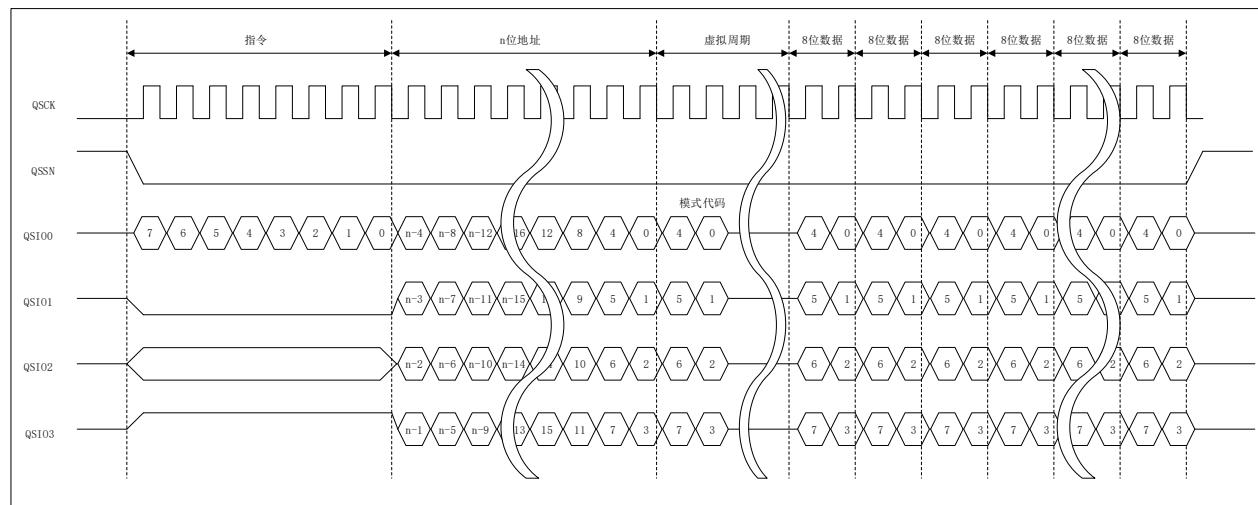


图 28-5 扩展式 SPI 协议动作示意图 2 (四线式输入输出快速读模式)

二线式 SPI 协议使用 QSIO0, QSIO1 两个管脚实行相应的操作，包括发出指令，地址，接收数据等等。

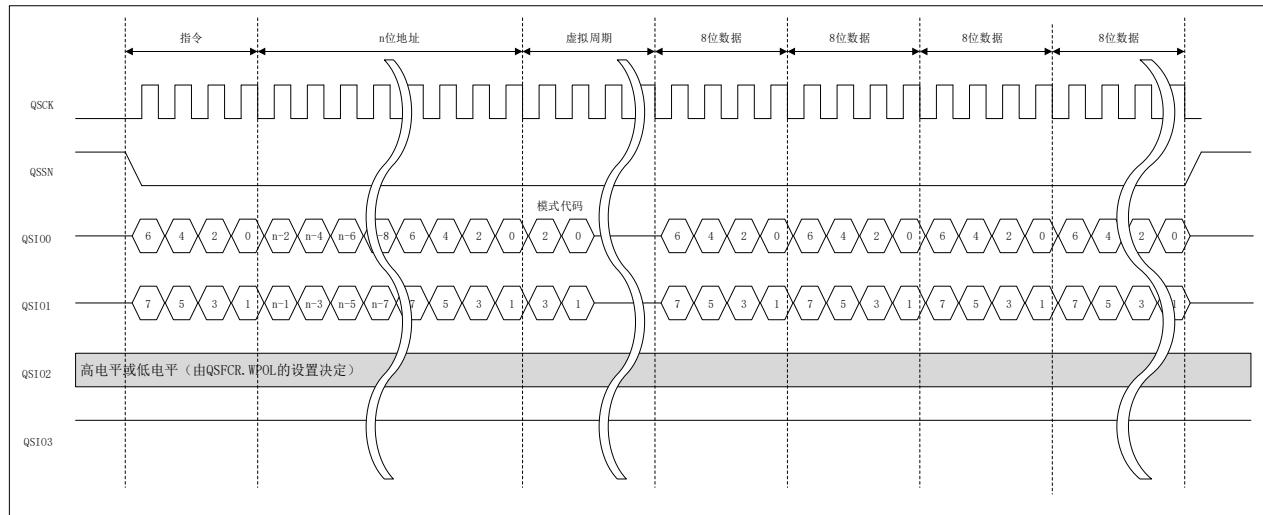


图 28-6 二线式 SPI 协议动作示意图（快速读模式）

四线式 SPI 协议使用 QSIO0, QSIO1, QSIO2, QSIO3 四个管脚实行发出指令, 地址, 接收数据等所有相关操作。

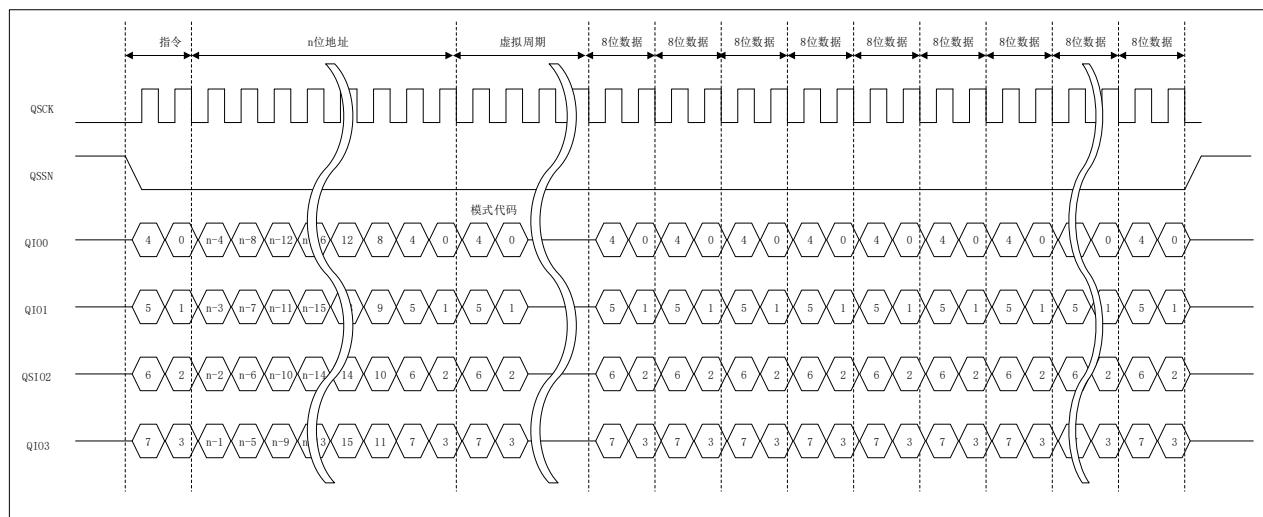


图 28-7 四线式 SPI 协议动作示意图（快速读模式）

28.3.2 SPI 模式

SPI 模式有模式 0 和模式 3 两种，可通过设置 QSCR 寄存器中的 SPIMD3 位来实现模式切换。SPI 模式 0 和模式 3 的区别在于待机状态时 QSCK 的电平不同。在 SPI 模式 0 时 QSCK 的待机电平为低电平，而模式 3 时待机电平则是高电平。

串行数据在串行时钟的下降沿从 QSPI 输出并在上升沿被读入外部闪存。而外部闪存则在串行时钟的下降沿输出串行数据并在下一个时钟的下降沿被 QSPI 读入。

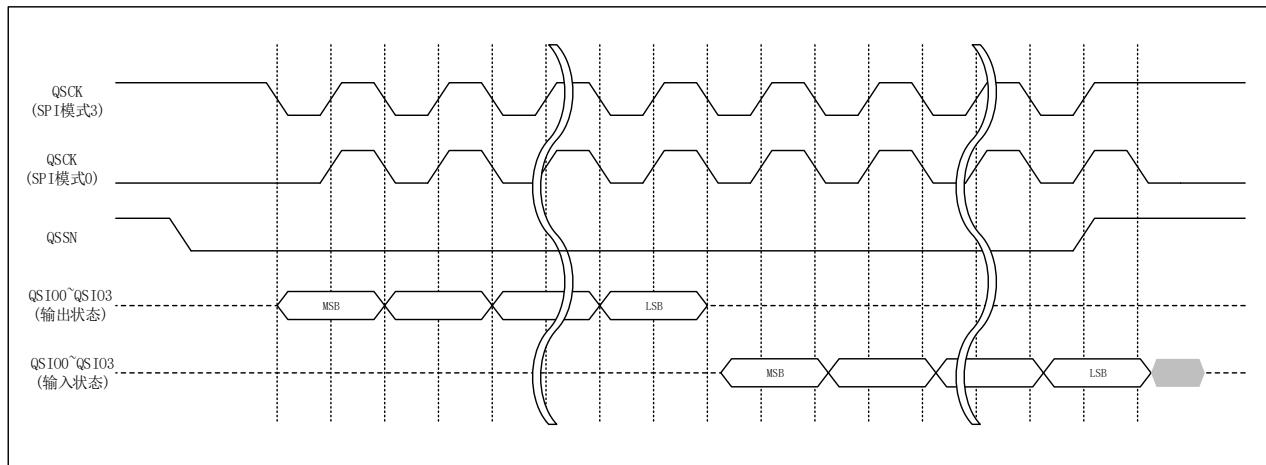


图 28-8 串行接口的基本时序图

28.4 QSPI 总线的时序调整

QSPI总线信号可以通过寄存器进行时序微调，微整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。

28.4.1 QSPI 总线基准时钟

QSPI 总线的基准时钟是由 HCLK 经过分频后得到的。通过设置 QSCR 寄存器的 DIV[5:0]位可以选择 HCLK 的 2 分频到 64 分频等多种时钟源作为 QSPI 总线的基准时钟。

DIV[5:0]	分频比	实际动作频率	DIV[5:0]	分频比	实际动作频率
		(HCLK=200MHz)			(HCLK=200MHz)
6'b0000000	2	100.00	6'b100000	33	6.06
6'b0000001	2	100.00	6'b100001	34	5.88
6'b0000010	3	66.67	6'b100010	35	5.71
6'b0000011	4	50.00	6'b100011	36	5.56
6'b0000100	5	40.00	6'b100100	37	5.41
6'b0000101	6	33.33	6'b100101	38	5.26
6'b0000110	7	28.57	6'b100110	39	5.13
6'b0000111	8	25.00	6'b100111	40	5.00
6'b001000	9	22.22	6'b101000	41	4.88
6'b001001	10	20.00	6'b101001	42	4.76
6'b001010	11	18.18	6'b101010	43	4.65
6'b001011	12	16.67	6'b101011	44	4.55
6'b001100	13	15.38	6'b101100	45	4.44
6'b001101	14	14.29	6'b101101	46	4.35
6'b001110	15	13.33	6'b101110	47	4.26
6'b001111	16	12.50	6'b101111	48	4.17
6'b010000	17	11.76	6'b110000	49	4.08
6'b010001	18	11.11	6'b110001	50	4.00
6'b010010	19	10.53	6'b110010	51	3.92

6'b010011	20	10.00	6'b110011	52	3.85
6'b010100	21	9.52	6'b110100	53	3.77
6'b010101	22	9.09	6'b110101	54	3.70
6'b010110	23	8.70	6'b110110	55	3.64
6'b010111	24	8.33	6'b110111	56	3.57
6'b011000	25	8.00	6'b111000	57	3.51
6'b011001	26	7.69	6'b111001	58	3.45
6'b011010	27	7.41	6'b111010	59	3.39
6'b011011	28	7.14	6'b111011	60	3.33
6'b011100	29	6.90	6'b111100	61	3.28
6'b011101	30	6.67	6'b111101	62	3.23
6'b011110	31	6.45	6'b111110	63	3.17
6'b011111	32	6.25	6'b111111	64	3.13

表 28-3 QSPI 总线基准时钟选择一览表

28.4.2 SPI 总线基准时钟

当基准时钟选择了 HCLK 的偶数倍分频时，QSCK 信号的高低电平时间是一致的，如果选择了奇数倍分频，那么 QSCK 信号的高电平时间将比低电平多一个 HCLK 周期。如果希望选择奇数倍分频时的 QSCK 也能输出 50% 左右占空比的时钟信号，可以将 QSFCR 寄存器中 DUTY 位设置成 1。通过这个设置，QSCK 信号的上升沿的输出时间将比调整前晚半个 HCLK 周期，下降沿输出时间保持不变。由此可以得到占空比为 50% 的 QSCK 信号。当基准时钟选择 HCLK 的偶数倍分频时，DUTY 位请设置为 0。初始状态下 DUTY 位默认为 1。

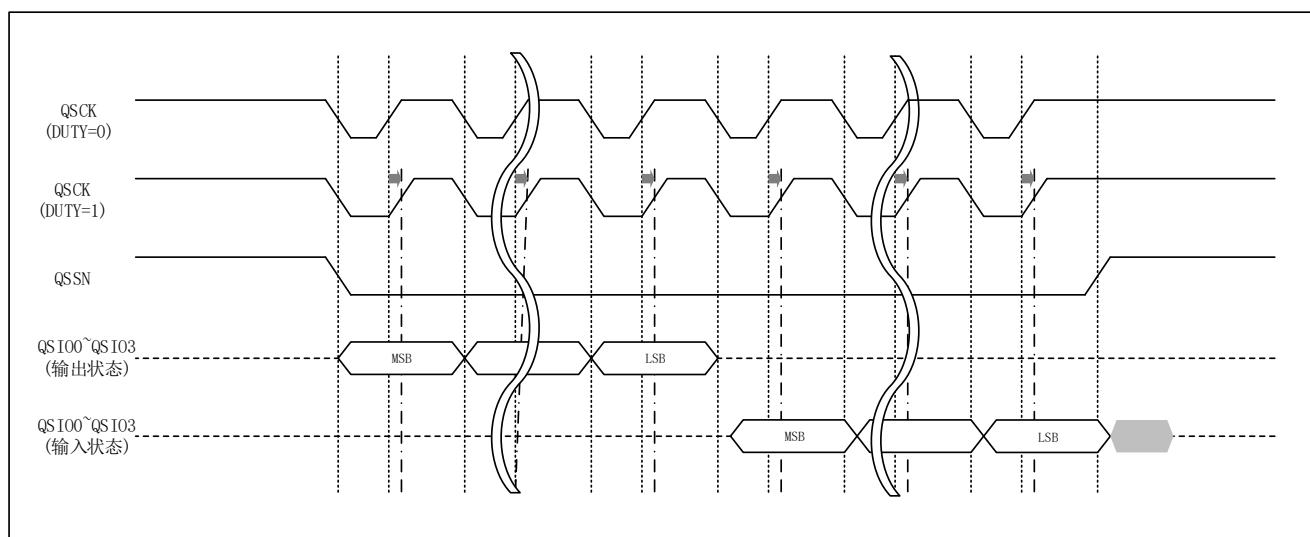


图 28-9 基准时钟选择 HCLK 三分频时输出时钟占空比补正示意图

28.4.3 QSSN 信号最小高电平宽度

为满足串行闪存所要求的取消时间要求，相邻的两次 QSPI 总线周期之间 QSSN 信号必须保持足够长时间的高电平状态（即闲置状态）。通过设置 QSCSCR 寄存器中的 SSHW[3:0]位可以选择 QSSN 的最小高电平宽度，选择范围为 1 个到 16 个 QSPI 基准时钟周期。

28.4.4 QSSN 的建立时间

从 QSSN 信号开始输出低电平（即变为有效状态）到 QSCK 信号输出第一个上升沿之间的时间称为 QSSN 的建立时间，该时间可以通过寄存器设置进行配置以满足外接串行闪存的要求。设置寄存器 QSFCR 的 SSNLD 位来选择 QSSN 的建立时间是 0.5 个或

是 1.5 个 QSPI 基准时钟周期。该设置同样可用于配置数据管脚从数据输出许可到 QCKS 信号输出第一个上升沿之间的建立时间，可根据需要合理运用。

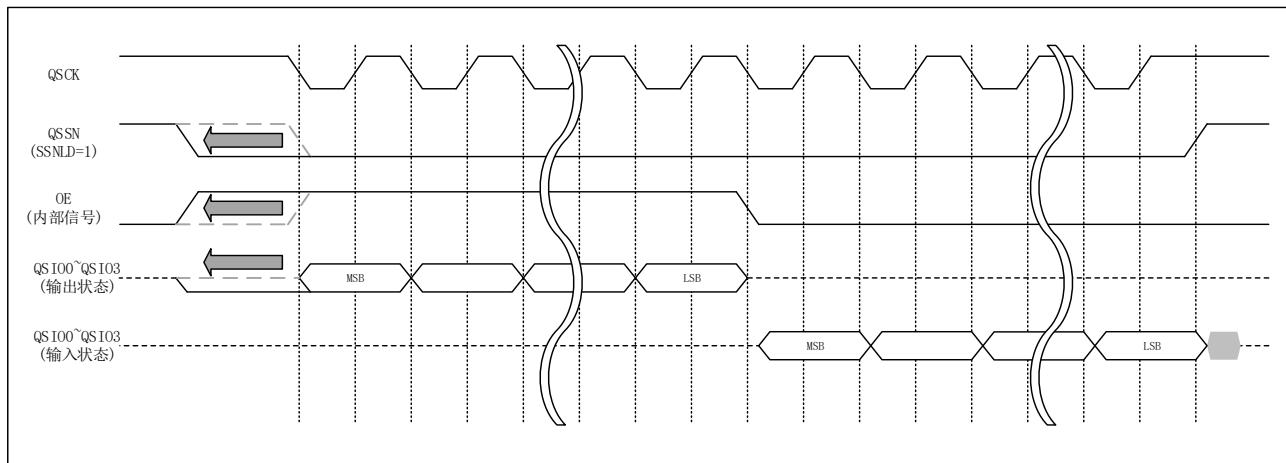


图 28-10 QSSL 建立时间配置示意图

28.4.5 QSSN 的保持时间

从 QSCK 信号输出最后一个上升沿到 QSSN 信号开始输出高电平（即变为闲置状态）之间的时间称为 QSSN 的保持时间，该时间可以通过寄存器设置进行配置以满足外接设备的要求。设置寄存器 QSFCT 的 SSNHD 位来选择 QSSN 的保持时间是 0.5 个或是 1.5 个 QSPI 基准时钟周期。

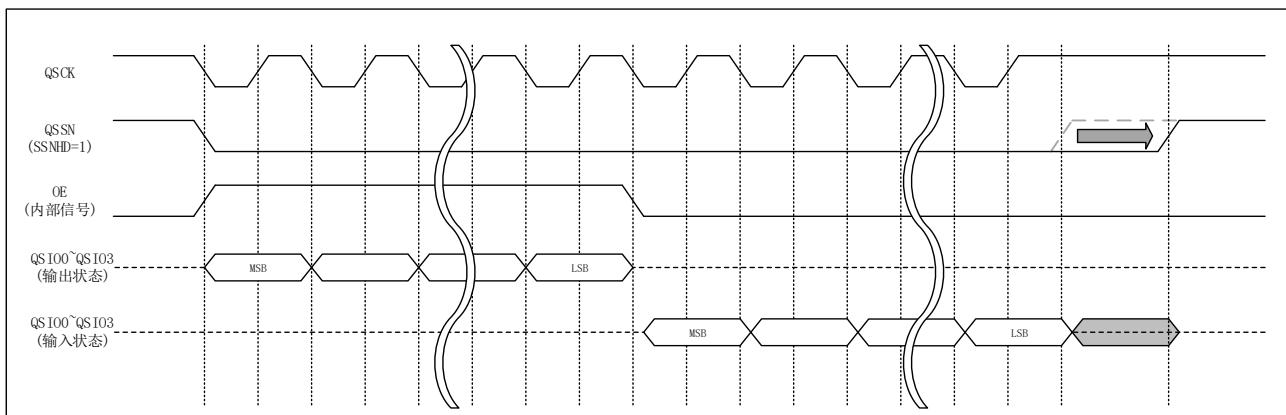


图 28-11 QSSN 保持时间配置示意图

28.4.6 串行数据接收延迟

串行闪存的数据和 QSCK 的下降沿同步输出，QSPI 在下一个 QSCK 的下降沿接收该数据。从串行闪存开始输出数据到该数据被 QSPI 接收之间的这段时间称之为接收延迟。QSPI 在第一个数据接收周期之前加入了一个延迟调整周期。站在串行闪存角度，

该周期可视为对手方数据接收周期的增加。该迟延调整周期只会在数据接收动作时产生。

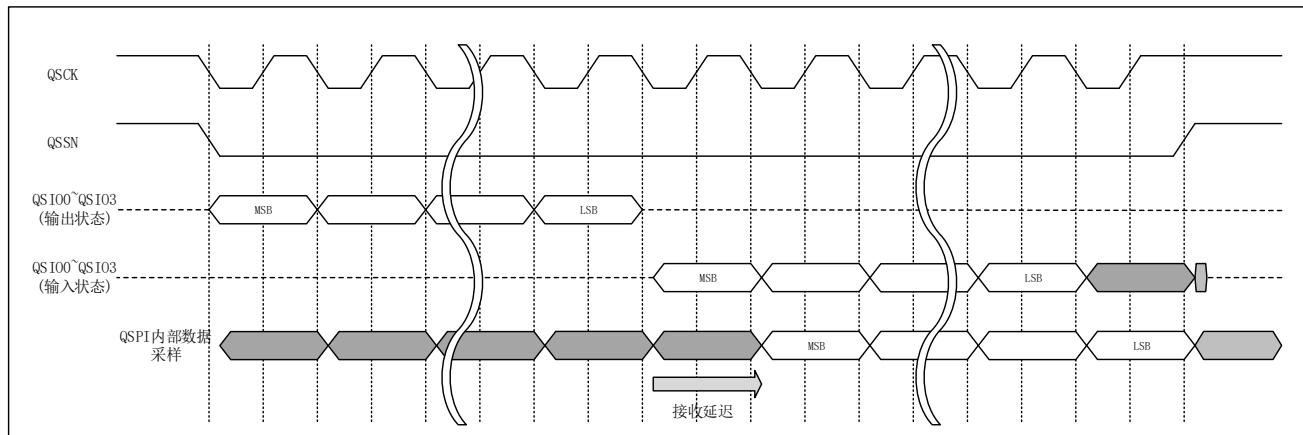


图 28-12 数据接收延迟示意图

28.5 用于 ROM 访问的 SPI 指令介绍

28.5.1 现有的 QSPI-ROM 指令参考

模式名	指令代码	说明	
4-byte指令模式	标准读	8'h13	
	快速读	8'h0C	
	二线式输出快速读	8'h3C	
	二线式输入输出快速读	8'hbC	
	四线式输出快速读	8'h6C	
	四线式输入输出快速读	8'hEC	
	退出4-byte指令模式	8'HB7	
3-byte指令模式	标准读	8'h03	
	标准读	8'h0B	当选择8位地址且A8=1时
	快速读	8'h0B	—
	二线式输出快速读	8'h3B	—
	二线式输入输出快速读	8'hBB	—
	四线式输出快速读	8'h6B	—
	四线式输入输出快速读	8'hEB	—
	进入4-byte指令模式	8'hE9	—
	写模式	8'h06	—

表 28-4 参考指令一览表

对串行闪存进行访问时，指令需通过指令寄存器 QSCCMD 来进行设置。

28.5.2 标准读指令

标准读指令是绝大多数串行闪存都支持的一种常用读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，随即 QSPI 就输出该指令的指令代码（03h/13h）*1，紧接着输出目标地址，该地址的宽度可以通过 QSFCR 寄存器中的 AWSL[1:0]位来设置。然后就可以接收到数据。QSPI 的初始状态选择的指令就是标准读指令。

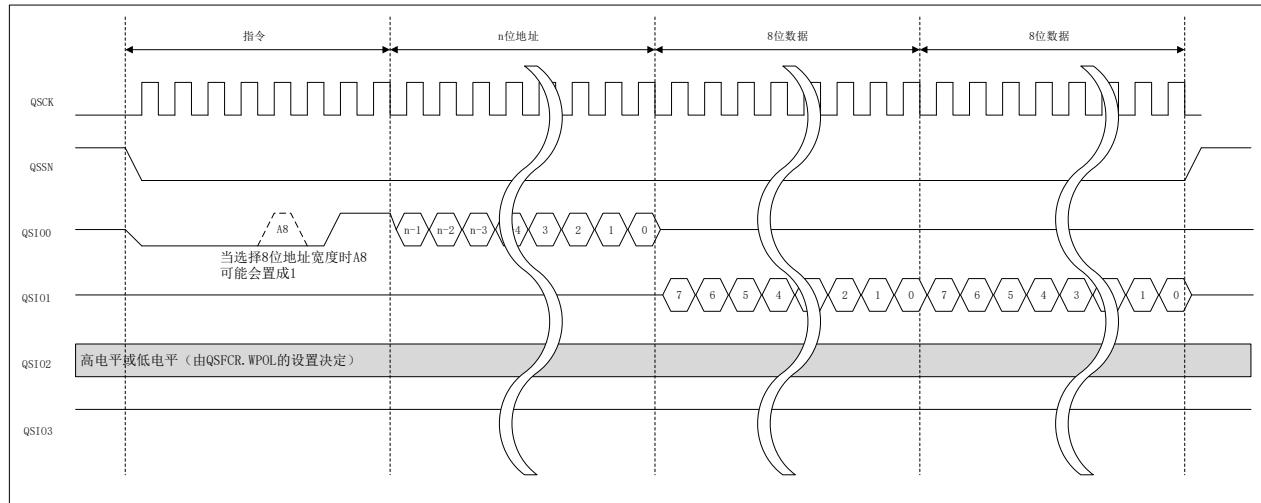


图 28-13 标准读总线周期示意图

28.5.3 快速读指令

快速读指令是一种支持更快的通信时钟的读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，随即 QSPI 就输出该指令的指令代码（0Bh/0Ch），紧接着输出目标地址，地址宽度可以通过 QSFCR 寄存器中的 AWSL[1:0]位来设置。地址输出后是一定数量的虚拟周期，其具体数量由 QSFCR 寄存器中的 DMCYCN[3:0]决定。紧接着就是数据的接收。

虚拟周期最初的两个周期用于决定是否选择 XIP 模式。当选择 XIP 模式时，本次传输所使用的指令将被应用于下一个 SPI 总线周期，在下一个 SPI 总线周期时指令传输部分将会被省略。详细可参考 28.7 XIP 控制。

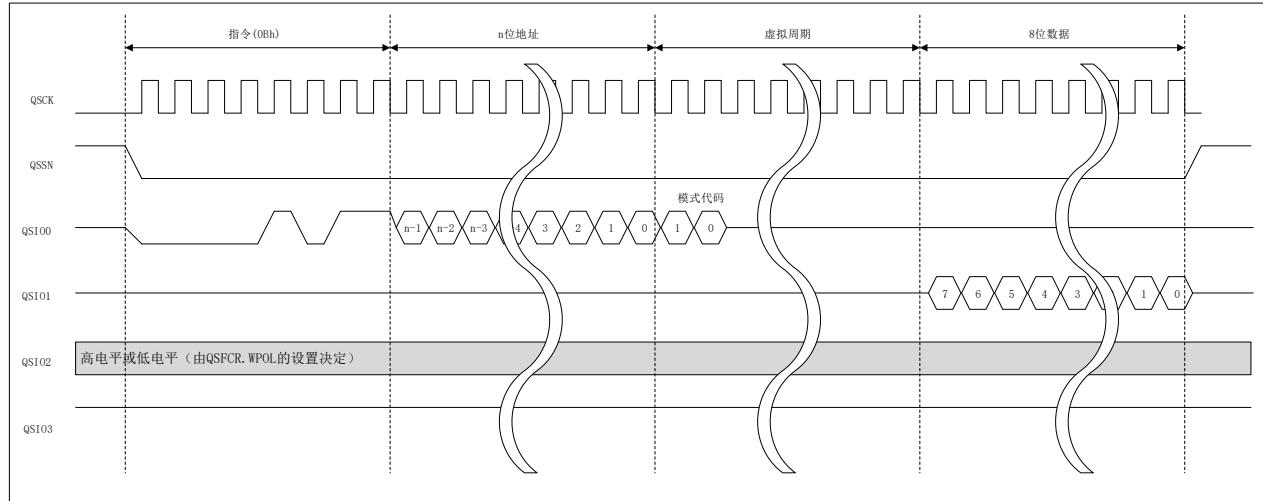


图 28-14 快速读总线周期示意图

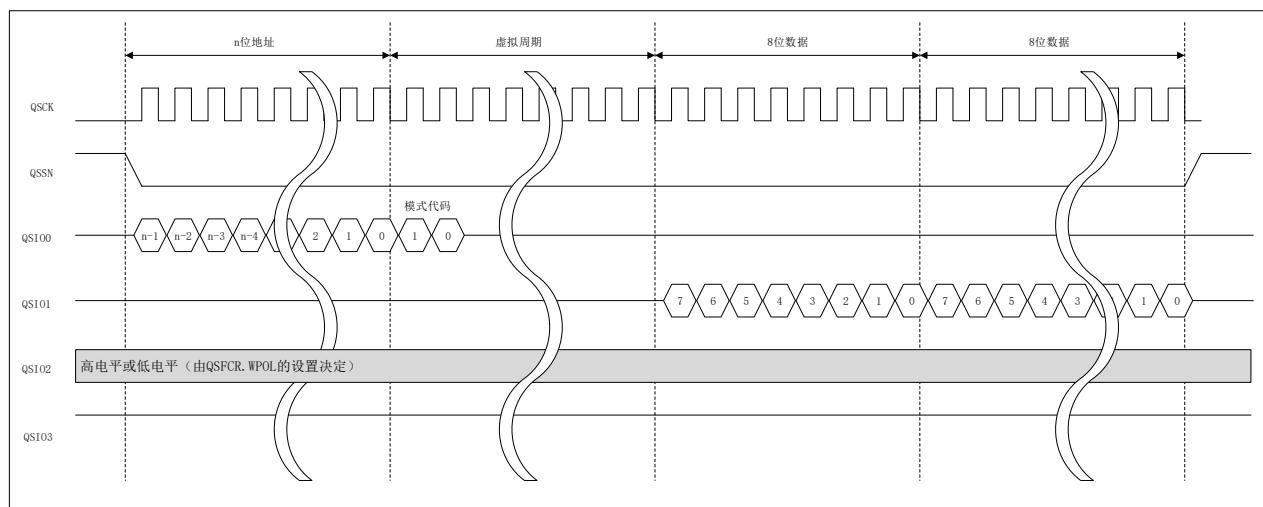


图 28-15 选择 XIP 模式的快速读总线周期示意图

注意：

- 要使用快速读指令请确保使用支持快速读功能的串行闪存。

28.5.4 二线式输出快速读指令

二线式输出快速读是一种使用两根信号线进行数据接收的读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSIO0 管脚输出该指令的指令代码（3Bh/3Ch）和目标地址，地址宽度可以通过 QSFCR 寄存器中的 AWSL[1:0]位来设置。在这之后是一定数量的虚拟周期，其具体数量由 QSFCR 寄存器中的 DMCYCN[3:0]决定。然后开始通过 QSIO0 和 QSIO1 两根管脚进行数据接收。偶位数据在 QSIO0 接收，奇位在 QSIO1。

虚拟周期最初的两个周期用于决定是否选择 XIP 模式。当选择 XIP 模式时，本次传输所使用的指令将被应用于下一个 QSPI 总线周期，在下一个 QSPI 总线周期时指令传输部分将会被省略。详细可参考 28.7 XIP 控制。

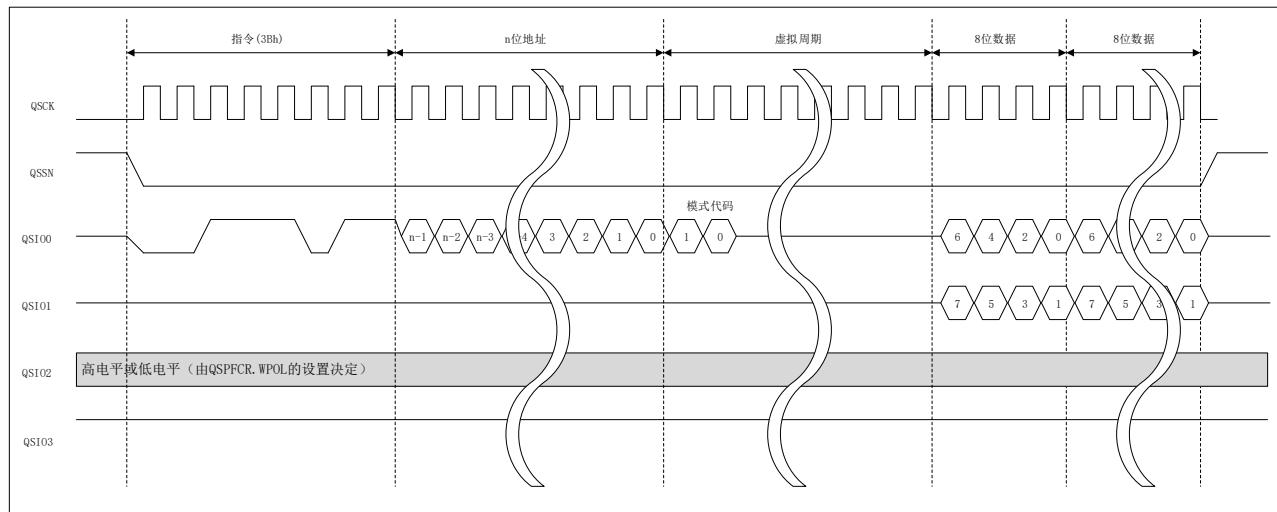


图 28-16 二线式输出快速读总线周期示意图

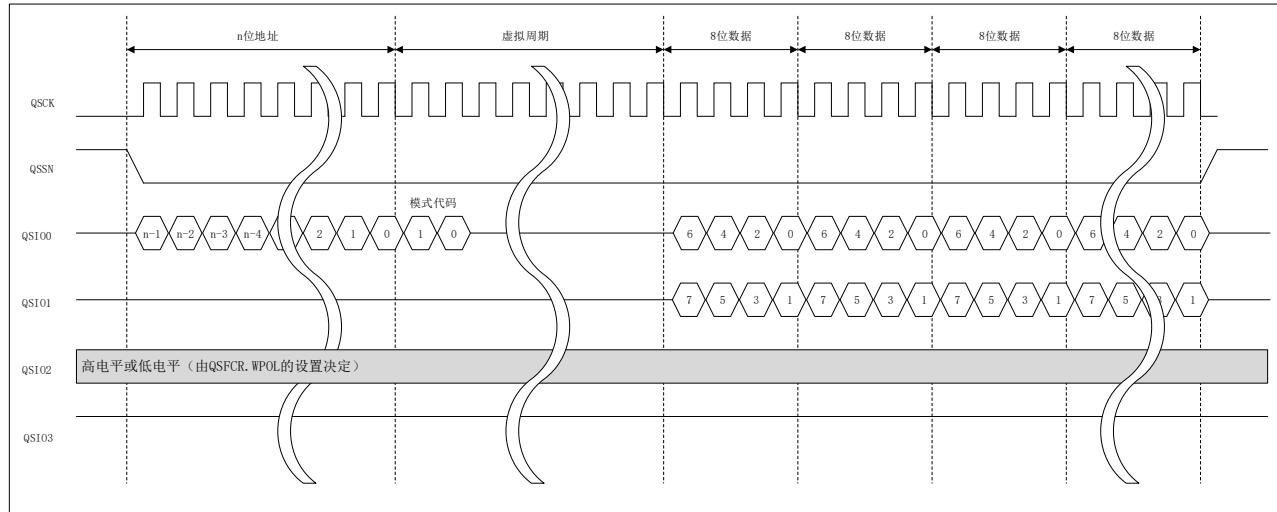


图 28-17 选择 XIP 模式的二线式输出快速读总线周期示意图

注意：

- 要使用二线式输出快速读指令请确保使用支持该功能的串行闪存。

28.5.5 二线式输入输出快速读指令

二线式输入输出快速读是一种使用两根信号线进行地址发送和数据接收的读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSIO0 管脚输出该指令的指令代码（BBh/BCh）。在这之后 QSPI 从 QSIO0 和 QSIO1 两个管脚输出目标地址，地址宽度可以通过 QSFCR 寄存器中的 AWSL[1:0]位来设置。在这之后是一定数量的虚拟周期，其具体数量由 QSFCR 寄存器中的 DMCYCN[3:0]决定。然后开始通过 QSIO0 和 QSIO1 两根管脚进行数据接收。偶数位的地址和虚拟周期（包括 XIP 模式选择信息）的传输及数据接收使用 QSIO0 管脚，奇数位使用 QSIO1 管脚。

虚拟周期最初的两个周期用于决定是否选择 XIP 模式。当选择 XIP 模式时，本次传输所使用的指令将被应用于下一个 QSPI 总线周期，在下一个 QSPI 总线周期时指令传输部分将会被省略。详细可参考 28.7 XIP 控制。

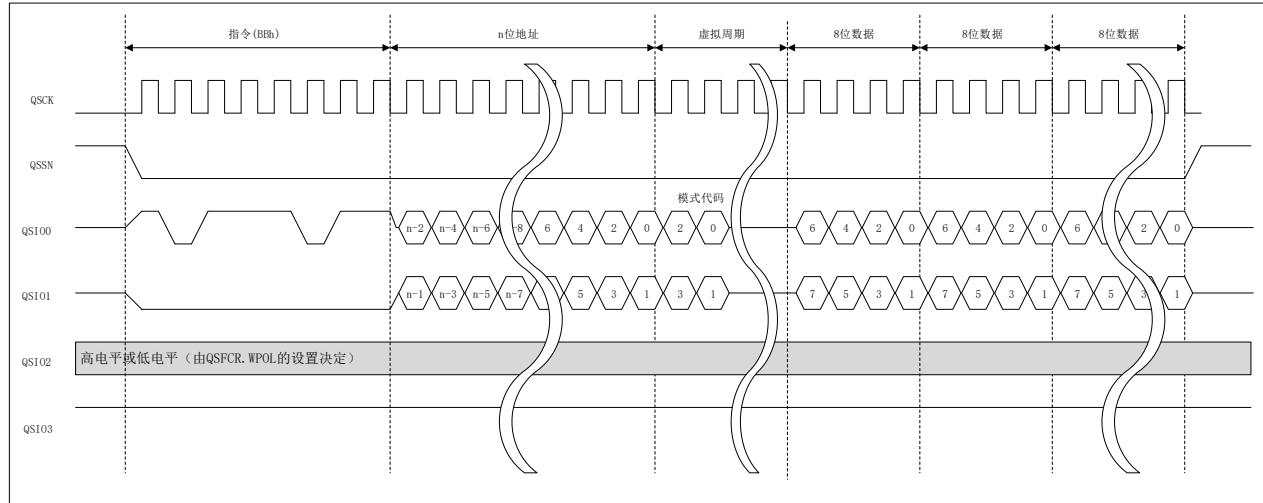


图 28-18 二线式输入输出快速读总线周期示意图

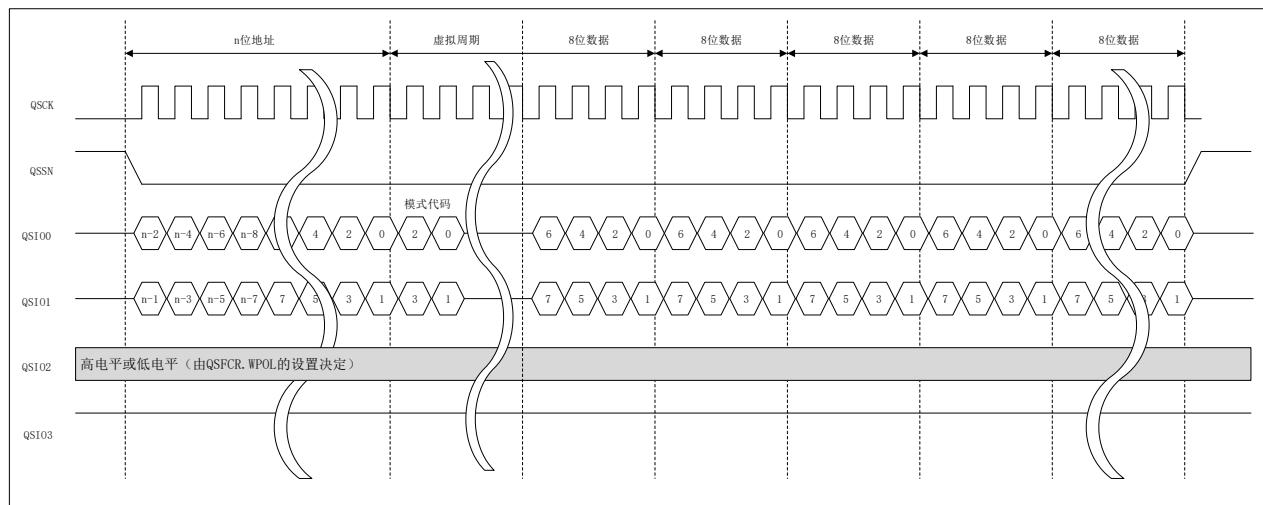


图 28-19 选择 XIP 模式的二线式输入输出快速读总线周期示意图

注意：

- 要使用二线式输入输出快速读指令请确保使用支持该功能的串行闪存。

28.5.6 四线式输出快速读指令

四线式输出快速读是一种使用四根信号线进行数据接收的读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSIO0 管脚输出该指令的指令代码（6Bh/6Ch）和目标地址，地址宽度可以通过 QSFCR 寄存器中的 AWSL[1:0]位来设置。在这之后是一定数量的虚拟周期，其具体数量由 QSFCR 寄存器中的 DMCYCN[3:0]决定。然后开始通过 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。详细可参考 28.7 XIP 控制。

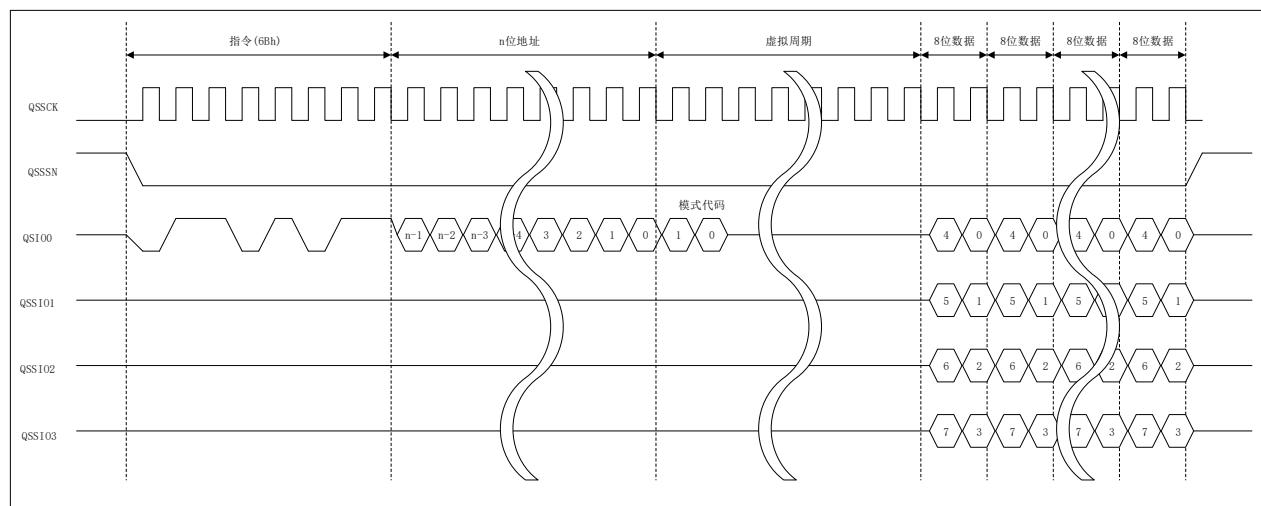


图 28-20 四线式输出快速读总线周期示意图

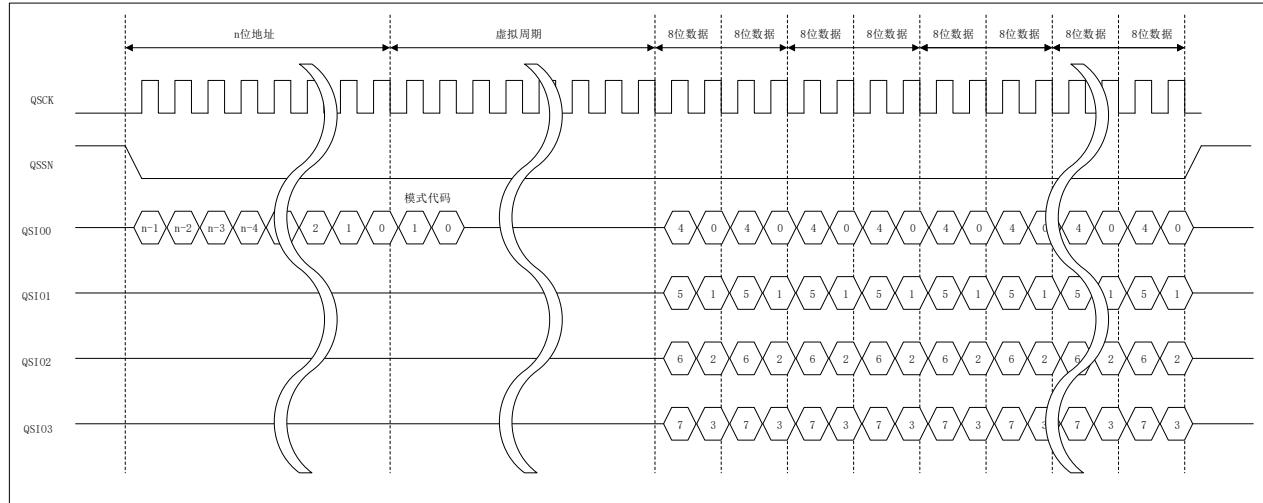


图 28-21 选择 XIP 模式的四线式输出快速读总线周期示意图

注意：

- 要使用四线式输出快速读指令请确保使用支持该功能的串行闪存。

28.5.7 四线式输入输出快速读指令

四线式输入输出快速读是一种使用四根信号线进行地址发送和数据接收的读指令。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSIO0 管脚输出该指令的指令代码（EBh/ECh）。在这之后 QSPI 从 QSIO0, QSIO1, QSIO2 和 QSIO3 四个管脚输出目标地址，地址宽度可以通过 QSFCR 寄存器中的 AWSL[1:0]位来设置。在这之后是一定数量的虚拟周期，其具体数量由 QSFCR 寄存器中的 DMCYCN[3:0]决定。然后开始通过 QSIO0, QSIO1, QSIO2 和 QSIO3 四根管脚进行数据接收。

虚拟周期最初的两个周期用于决定是否选择 XIP 模式。当选择 XIP 模式时，本次传输所使用的指令将被应用于下一个 QSPI 总线周期，在下一个 QSPI 总线周期时指令传输部分将会被省略。详细可参考 28.7 XIP 控制。

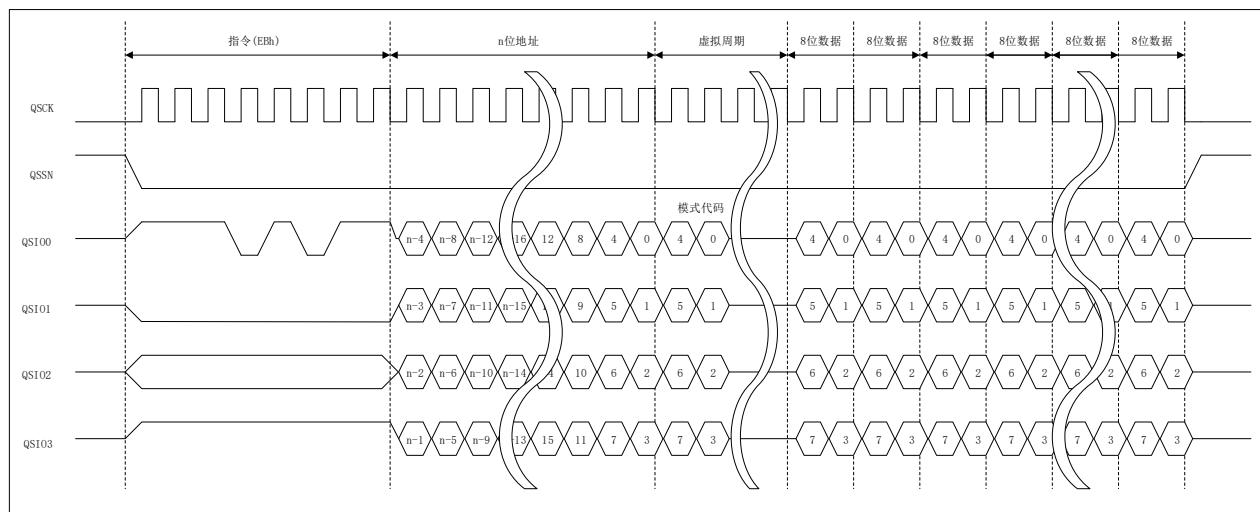


图 28-22 四线式输入输出快速读总线周期示意图

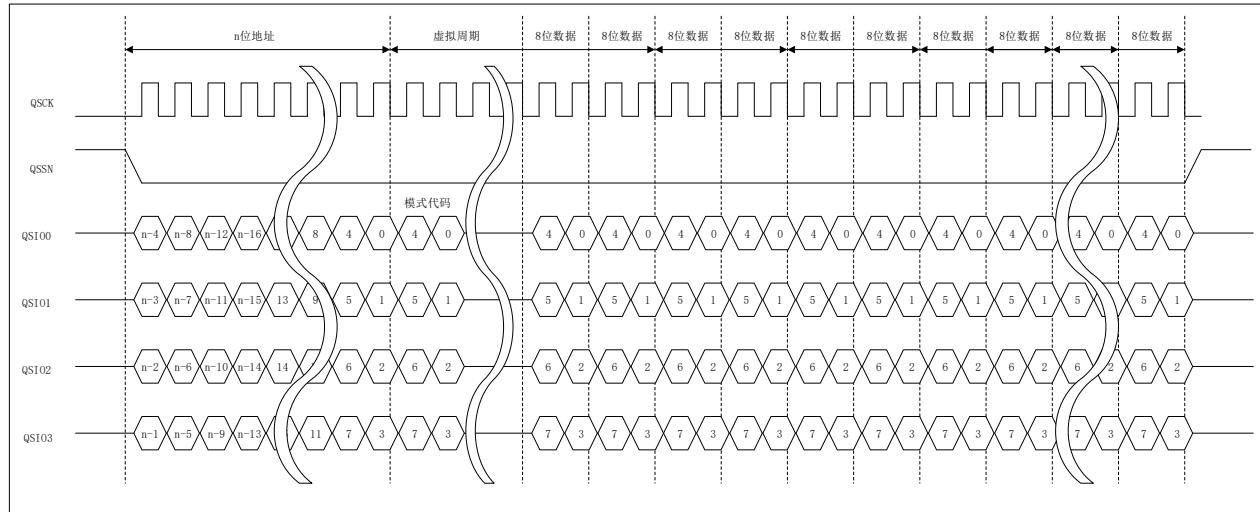


图 28-23 选择 XIP 模式的四线式输入输出快速读总线周期示意图

注意：

- 要使用四线式输入输出快速读指令请确保使用支持该功能的串行闪存。

28.5.8 进入 4-Byte 模式指令

进入 4-Byte 模式指令可以将串行闪存的地址宽度设置成 4 个字节。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSIO0 管脚输出该指令的指令代码（B7h）。

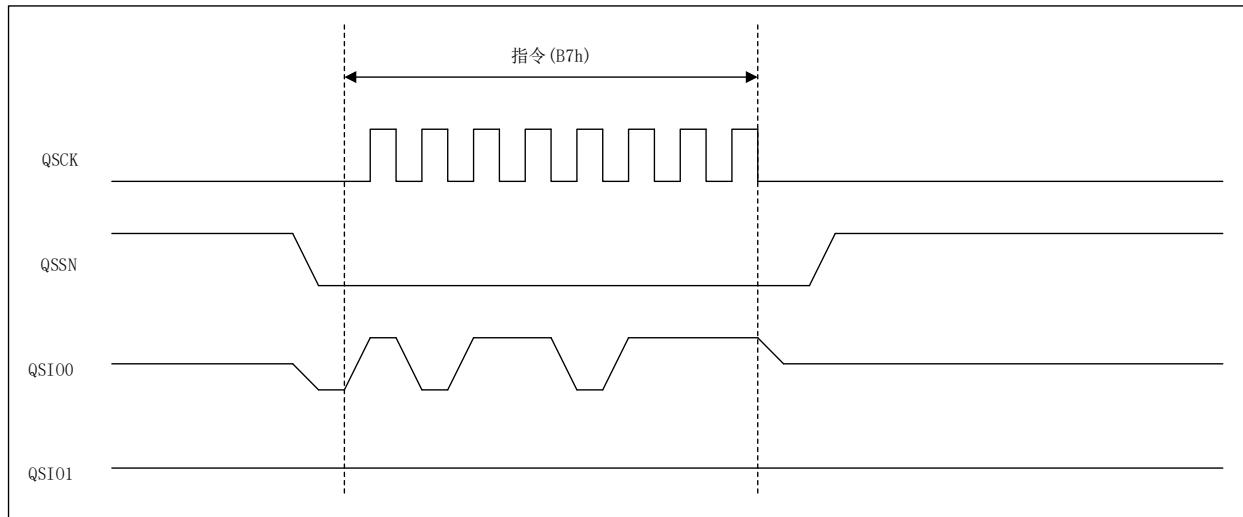


图 28-24 进入 4-Byte 模式指令总线周期示意图

注意：

- 无论串行闪存是否处于 4-Byte 或是 3-Byte 模式，该指令都可发布。

28.5.9 退出 4-Byte 模式指令

退出 4-Byte 模式指令可以将串行闪存的地址宽度设置成 3 个字节。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI 开始从 QSIO0 管脚输出该指令的指令代码（E9h）。

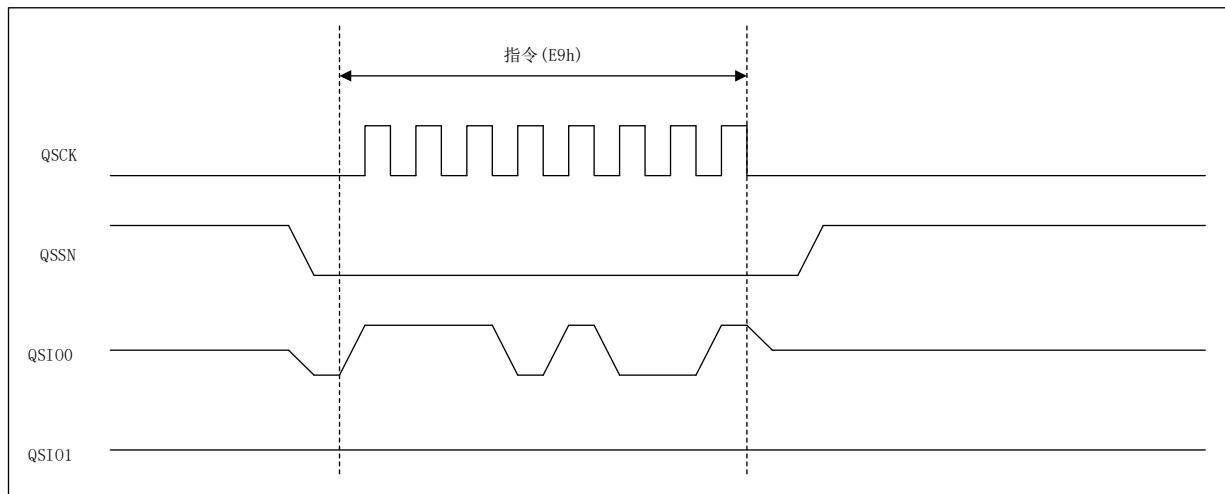


图 28-25 退出 4-Byte 模式指令总线周期示意图

注意：

- 无论串行闪存是否处于 4-Byte 或是 3-Byte 模式，该指令都可发布。

28.5.10 写许可指令

写许可指令允许改变串行闪存的地址宽度。当一个串行总线周期开始的时候，串行闪存选择信号被置为有效状态，QSPI开始从QSIO0管脚输出该指令的指令代码(06h)。

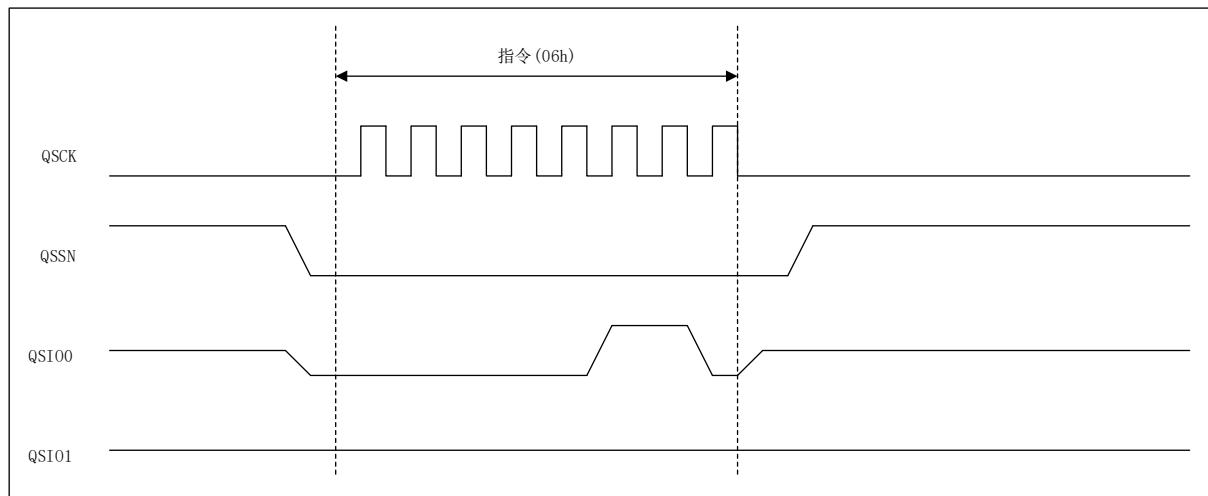


图 28-26 写许可指令总线周期示意图

28.6 QSPI 总线周期的安排

28.6.1 独立转换的单一闪存读取

对于 ROM 的单一读取指令会独立的从芯片内部总线周期一对一的转换为 QSPI 总线周期。当一个 ROM 的读取总线周期被检测到时，QSSN 信号会置为有效状态，从而启动一个 QSPI 总线周期。当接收完串行闪存的数据后，QSSN 信号变成无效状态，该 QSPI 总线周期宣告完成。

当另一个 ROM 的读取总线周期被检测到时，QSSN 信号会在确保无效保持时间已经超过了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。

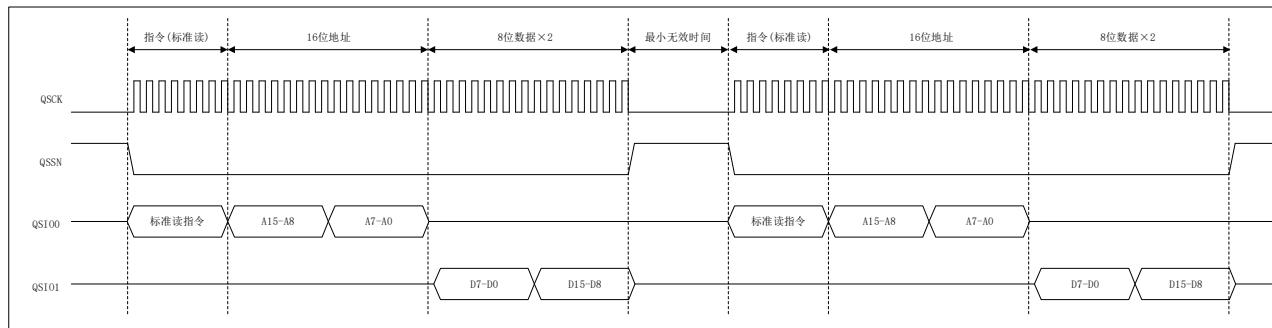


图 28-27 独立转换的单一闪存数据读取操作示意图

28.6.2 使用预读取功能进行闪存读取

对于诸如 CPU 指令或是数据块的传输，系统通常是以一个顺次递增的闪存地址顺序进行数据的读取。串行闪存具有连续数据传输能力而不需要再次发送指令代码和地址。但如果 MCU 发布的内部总线周期是独立转换的话，QSPI 总线周期也被分成了各自独立的个体，导致无法有效利用串行闪存的这种连续数据传输的优点。对此，QSPI 提供了预读取功能来进行连续的数据接收。

通过设置 QSCR 寄存器中的 PFE 位为 1 激活预读取功能。当该功能有效后，数据会被连续接收并储存到缓冲区而不需要等待另一个闪存读取要求。当 MCU 发出一个闪存读取操作时，QSPI 将会对访问地址进行匹配。如果匹配成功，则将对应位置的缓冲区的数据传给 MCU，若匹配失败，则缓冲区的数据将会被丢弃并重新发出一个新的 QSPI 总线周期。

预读取的缓冲区最大可存储 16 个字节的数据，除此以外，还有 2 个字节的数据接收缓

冲区也可以存储预读取的数据，当所有的缓冲区数据取满后，QSPI 总线周期结束。当缓冲区数据被读取后产生新的缓冲空间，QSPI 会自动开始一个新的 QSPI 总线周期来恢复预读取动作。

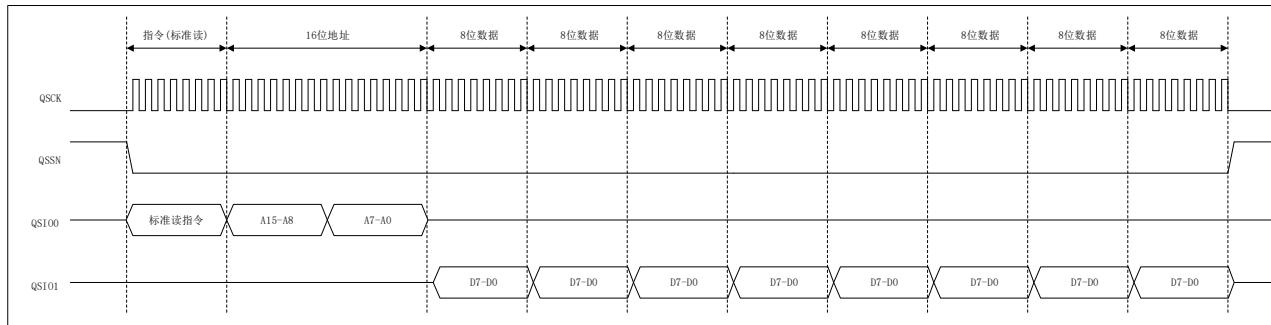


图 28-28 预读取功能有效时数据读取操作示意图

28.6.3 预读取的终止

在预读取传送过程中如果发生了对串行闪存其他地址的 ROM 读取总线周期，那么原预读取动作将会被终止并开始一个新的 QSPI 总线周期。通常情况下预读取动作会在当前字节传输完成之后终止，但如果将 QSCR 寄存器中的 PFSAE 位设置成 1 的话，QSPI 会即时停止预读取动作而不会等到当前字节传输完成。使用这一功能需要串行闪存器件支持即时停止动作功能。

28.6.4 预读取状态监测

从一个低速串行闪存读取数据会增加系统的负荷，这是因为内部总线需要一直处在等待状态直到收信的 QSPI 总线周期完成。QSPI 提供了预读取状态监测功能来降低这种负荷。

在预读取状态寄存器 QSSR 中，PFAN 位显示了当前的预读取工作状态，PFFUL 位表示预读取数据缓冲区已经放满，而 PFNUM[4:0] 则显示了目前已经读取到缓冲区的数据的字节数。通过这些状态位可以很方便的通过一条 CPU 指令来确定当前的预读取状况。

注意：

- 当执行一段预读取状态监测程序时，请将该程序代码放置在对象串行闪存区域以外或者启用指令缓存。否则预读取对象将频繁的切换在对象数据区域和指令区域之间，失去了预读取的意义，而监测程序也会因为预读取始终无法完成而进入无限循环状态。

28.6.5 使用 QSPI 总线周期延长功能进行闪存读取

如果对 QSCSCR 寄存器中的 SSNW[1:0]进行设置 00 以外的值, QSPI 总线周期将会在接收完数据后处于保持状态等待下一个数据的读取。这时 QSSN 信号将保持低电平的有效状态, 而 QSCK 则处于停止状态。如果下一条闪存读取指令到来时, 如果读取对象地址是紧接着当前地址顺序递增的, 则 QSPI 会将 QSCK 重新启动直接进行数据接受。如果是非连续的地址的话, QSSN 信号会被置成高电平的无效状态将原先保持的 QSPI 总线周期结束, 然后重新开始一个新的 SPI 总线周期。

这一功能可以避免系统在对连续递增地址进行不连续读取时反复发送指令代码和地址, 从而提高读取效率。

QSPI 总线周期的延长时间可以通过 SSNW[1:0]设置, 当超过所设置的时间仍未发生下一次的读取时, QSSN 会自动置成高电平的无效状态, 结束 QSPI 总线周期。如果 SSNW[1:0]设成 11 的话, QSSN 会被无限期延长, QSPI 总线周期将始终处在保持状态, 但这样做会增加串行闪存的功耗。

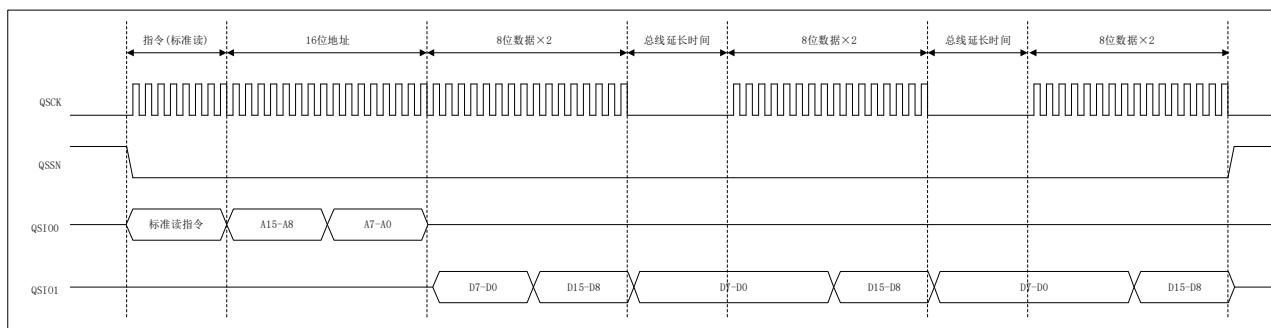


图 28-29 使用 QSPI 总线周期延长功能的数据读取操作示意图

28.7 XIP 控制

一些串行闪存器件可以通过省略接收读取指令来降低延迟时间。该机能可通过虚拟周期期间发送的模式代码来选择。

在快速度指令时的虚拟周期期间，QSPI 在最初的两个周期通过发送 XIP 模式代码来控制串行闪存的 XIP 模式。不同的串行闪存其 XIP 模式代码也不相同，可通过寄存器 QSXCMD 的 XIPMC[7:0]位进行针对性设置。具体参照下图 28-30。

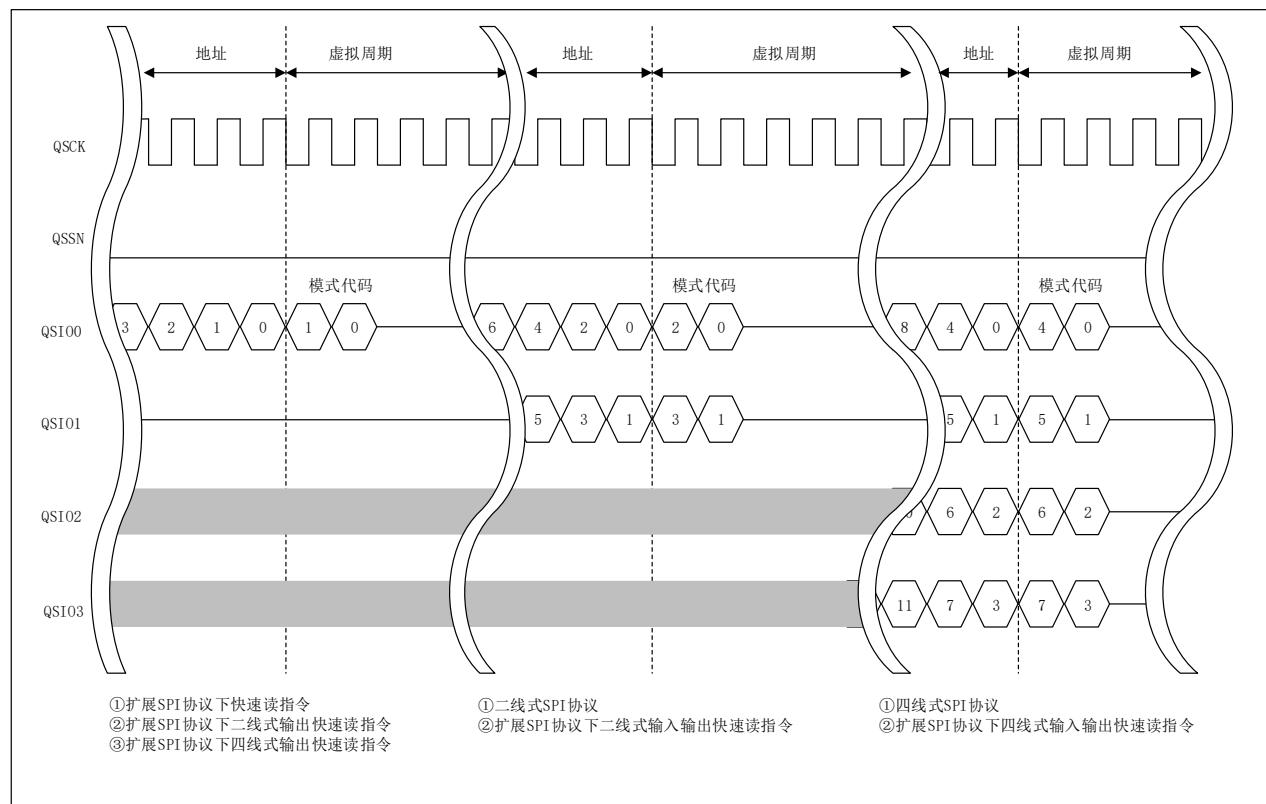


图 28-30 XIP 模式控制示意图

28.7.1 XIP 模式的设置

在将对应串行闪存的 XIP 模式代码写入寄存器 QSXCMD 的 XIPMC[7:0]且寄存器 QSCR 的 XIPE 位设置成 1 的情况下，当下一个快速读指令发生时，所设置的模式代码将在虚拟周期的前两个周期期间传送给对象串行闪存，在模式代码接收完成后，串行闪存及其控制部就启动 XIP 模式。可以通过访问 QSSR 的 XIPF 位来确认是否已经进入 XIP 模式。

注意：

- 启动串行闪存的 XIP 模式需要在 QSXCMD[7:0]中设置相应的模式代码，控制部分的 XIP 模式只需要将 XIPE 位置成 1 就可以，而与 QSXCMD[7:0]的值无关。

28.7.2 XIP 模式的退出

在将对应串行闪存的退出 XIP 模式的代码写入寄存器 QSXCMD 的 XIPMC[7:0]且寄存器 QSCR 的 XIPE 位设置成 0 的情况下，当下一个快速读指令发生时，所设置的退出模式代码将在虚拟周期的前两个周期期间传送给对象串行闪存，在退出模式代码接收完成后，串行闪存及其控制部的 XIP 模式被终止。可以通过访问 QSSR 的 XIPF 位来确认是否已经退出 XIP 模式。

注意：

- 退出串行闪存的 XIP 模式需要在 QSXCMD[7:0]中设置相应的退出模式代码，控制部分的 XIP 模式只需要将 XIPE 位清零就可以，而与 QSXCMD[7:0]的值无关。

28.8 QSIO2 和 QSIO3 的管脚状态

QSIO2 和 QSIO3 管脚状态取决于 QSCR 寄存器中 MDSEL[2:0]位所设置的串行读取模式。

QSCR寄存器 MDSEL[2:0]位	QSIO2状态	QSIO3状态	备注
000			标准读（初始状态）
001			快速读
010			二线式输出快速读
011			二线式输入输出快速读
100	作为第三根数据线进行输入或输出动作，待机状态为Hi-Z	作为第四根数据线进行输入或输出动作，待机状态为Hi-Z	四线式输出快速读
101			四线式输入输出快速读
110	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
111			自定义协议快速读

表 28-5 QIO2 和 QIO3 的管脚状态

注意：

- QSIO2 管脚也可用作串行闪存的 WP#功能。
- QSIO3 管脚也可用作串行闪存 HOLD#或 RESET#功能。

28.9 直接通信模式

28.9.1 关于直接通信模式

QSPI 可以通过自动将 MCU 的外部 ROM 读取总线周期转换为 QSPI 总线周期来对串行闪存进行读取。但串行闪存还有很多不同的追加功能，诸如 ID 信息读取，擦除，写入及状态信息读取等。这些功能并没有一套标准的指令来进行设置，并且随着串行闪存新功能的迅速增加，硬件层面上的对应变得愈发的困难。

针对这种情况，QSPI 提供了直接通信模式，用户可通过软件直接对串行闪存进行控制。由此模式软件可以产生任意所需的 QSPI 总线周期。

28.9.2 直接通信模式的设置

将 QSCR 寄存器的 DCOME 位设成 1 可以进入直接通信模式。一旦进入直接通信模式，将无法进行通常的闪存读取操作，如果要进行常规的闪存读取，需要将 DCOME 位清零退出直接通信模式。

注意：

- 如果处于 XIP 模式，则需要先退出 XIP 模式再启动直接通信模式。

28.9.3 直接通信模式下 QSPI 总线周期的生成

直接通信模式下一个完整的 QSPI 总线周期从对寄存器 QSDCOM 的 DCOM[7:0]第一次操作开始直到对 QSCR 寄存器的进行一次写操作后结束。在这期间可以对 DCOM[7:0]进行多次操作，对 DCOM[7:0]的写会转换为一次 QSPI 总线的单字节的数据传送，而对 DCOM[7:0]的读则会转换成一次 QSPI 总线的单字节的数据接收。

从对寄存器 QSCOM 的 DCOM[7:0]第一次操作开始到最后对 QSCR 寄存器的写操作为止，这期间 QSSN 信号始终保持低电平的有效状态。

直接通信模式不支持多线式动作。

注意：

- 在直接通信模式下是无法对 QSCR 和 QSDCOM 以外的寄存器进行写操作的。对其他寄存器的写操作将会退出直接通信模式。使用这种方式退出可能导致不可预料的情况出现所以不推荐使用。

28.10 中断

当直接通信模式下检测到对 ROM 的读访问操作时，QSSR 寄存器的 RAER 位被置成 1，这时候 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。详细请参考【中断控制器（INTC）】。

28.11 使用上的注意事项

28.11.1 QSPI 寄存器的设置顺序

在系统操作中可以动态的对 QSPI 控制寄存器进行设置或更改。但是不注意寄存器的设置顺序可能会导致 QSPI 总线周期在寄存器还没有完全设置完成时就开始，因此请仔细配置寄存器的设置顺序以避免这类情况的发生。

28.11.2 模块停止信号的设置

QSPI 在系统复位后处于模块停止状态，只有在将模块停止控制寄存器中的 QSPI 模块停止信号清零号才可以对寄存器进行设置。具体请参考 5.5 降低功耗的方法。

28.12 寄存器说明

寄存器基准地址: 0x9C00_0000

寄存器名	偏移地址	复位值
控制寄存器QSCR	0x0000	0x003F_0000
片选控制寄存器QSCSCR	0x0004	0x0000_000F
格式控制寄存器QSFCR	0x0008	0x0000_8033
状态寄存器QSSR	0x000C	0x0000_8000
直接通信指令寄存器QSDCOM	0x0010	-
指令代码寄存器QSCCMD	0x0014	0x0000_0000
XIP模式代码寄存器QSXCMD	0x0018	0x0000_00FF
标志清除寄存器QSSR2 (只写)	0x0024	-
外部地址寄存器QSEXAR	0x0804	0x0000_0000

表 28-6 QSPI 寄存器一览表

28.12.1 QSPI 控制寄存器 (QSCR)

复位值: 0x003F_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DIV[5:0]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	DPRSL[1:0]	APRSL[1:0]	IPRSL[1:0]	SPIMD3	XIPE	DCOME	PFSAE	PFE	MDSEL[2:0]	—	—	—	—	—

位	标记	位名	功能	读写
b31~b21	Reserved	—	读出时为“0”,写入时写“0”	R/W
b20~b16	DIV[5:0]	基准时钟选择位	基准时钟选择位 串行接口基准时钟选择 b5 b4 b3 b2 b1 b0 0 0 0 0 0 0: 2个HCLK周期 0 0 0 0 0 1: 2个HCLK周期* 0 0 0 0 1 0: 3个HCLK周期 0 0 0 0 1 1: 4个HCLK周期* 0 0 0 1 0 0: 5个HCLK周期 0 0 0 1 0 1: 6个HCLK周期* 0 0 0 1 1 0: 7个HCLK周期 0 0 0 1 1 1: 8个HCLK周期* 0 0 1 0 0 0: 9个HCLK周期 0 0 1 0 0 1: 10个HCLK周期* 0 0 1 0 1 0: 11个HCLK周期 0 0 1 0 1 1: 12个HCLK周期* 0 0 1 1 0 0: 13个HCLK周期 0 0 1 1 0 1: 14个HCLK周期* 0 0 1 1 1 0: 15个HCLK周期 0 0 1 1 1 1: 16个HCLK周期* 0 1 0 0 0 0: 17个HCLK周期 0 1 0 0 0 1: 18个HCLK周期 0 1 0 0 1 0: 19个HCLK周期 0 1 0 0 1 1: 20个HCLK周期 0 1 0 1 0 0: 21个HCLK周期 0 1 0 1 0 1: 22个HCLK周期 0 1 0 1 1 0: 23个HCLK周期 0 1 0 1 1 1: 24个HCLK周期 0 1 1 0 0 0: 25个HCLK周期 0 1 1 0 0 1: 26个HCLK周期 0 1 1 0 1 0: 27个HCLK周期 0 1 1 0 1 1: 28个HCLK周期	R/W

			0 1 1 1 0 0: 29个HCLK周期 0 1 1 1 0 1: 30个HCLK周期 0 1 1 1 1 0: 31个HCLK周期 0 1 1 1 1 1: 32个HCLK周期 1 0 0 0 0 0: 33个HCLK周期 1 0 0 0 0 1: 34个HCLK周期* 1 0 0 0 1 0: 35个HCLK周期 1 0 0 0 1 1: 36个HCLK周期* 1 0 0 1 0 0: 37个HCLK周期 1 0 0 1 0 1: 38个HCLK周期* 1 0 0 1 1 0: 39个HCLK周期 1 0 0 1 1 1: 40个HCLK周期* 1 0 1 0 0 0: 41个HCLK周期 1 0 1 0 0 1: 42个HCLK周期* 1 0 1 0 1 0: 43个HCLK周期 1 0 1 0 1 1: 44个HCLK周期* 1 0 1 1 0 0: 45个HCLK周期 1 0 1 1 0 1: 46个HCLK周期* 1 0 1 1 1 0: 47个HCLK周期 1 0 1 1 1 1: 48个HCLK周期* 1 1 0 0 0 0: 49个HCLK周期 1 1 0 0 0 1: 50个HCLK周期 1 1 0 0 1 0: 51个HCLK周期 1 1 0 0 1 1: 52个HCLK周期 1 1 0 1 0 0: 53个HCLK周期 1 1 0 1 0 1: 54个HCLK周期 1 1 0 1 1 0: 55个HCLK周期 1 1 0 1 1 1: 56个HCLK周期 1 1 1 0 0 0: 57个HCLK周期 1 1 1 0 0 1: 58个HCLK周期 1 1 1 0 1 0: 59个HCLK周期 1 1 1 0 1 1: 60个HCLK周期 1 1 1 1 0 0: 61个HCLK周期 1 1 1 1 0 1: 62个HCLK周期 1 1 1 1 1 0: 63个HCLK周期 1 1 1 1 1 1: 64个HCLK周期	
b15~b14	Reserved	—	读出时为“0”,写入时写“0”	R/W
b13~b12	DPRSL[1:0]	数据接收阶段SPI协议选择	数据接收阶段SPI协议选择。 b1 b0 0 0: 扩展式SPI协议 0 1: 二线式SPI协议 1 0: 四线式SPI协议	R/W

			1 1: 设定禁止	
b11~b10	APRSL[1:0]	地址发送阶段SPI协议选择	地址发送阶段SPI协议选择。 b1 b0 0 0: 扩展式SPI协议 0 1: 二线式SPI协议 1 0: 四线式SPI协议 1 1: 设定禁止	R/W
b9~b8	IPRSL[1:0]	指令发送阶段SPI协议选择	指令发送阶段SPI协议选择。 b1 b0 0 0: 扩展式SPI协议 0 1: 二线式SPI协议 1 0: 四线式SPI协议 1 1: 设定禁止	R/W
b7	SPIMD3	SPI模式选择	SPI模式选择 0: SPI模式0 1: SPI模式3	R/W
b6	XIPE	XIP模式许可	0: XIP模式禁止 1: XIP模式许可	R/W
b5	DCOME	直接通信许可	QSPI总线通信模式选择 0: ROM访问模式 1: 直接通信模式	R/W
b4	PFSAE	预读取即时停止许可	选择重置预读取动作的位置 0: 当前的预读取动作在字节边界中止 1: 当前的预读取动作即时中止	R/W
b3	PFE	预读取许可	预读取功能有效/无效选择 0: 预读取功能无效 1: 预读取功能有效	R/W
b2~b0	MDSEL[2:0]	QSPI读取模式选择	串行接口读取模式选择 b2 b1 b0 0 0 0: 标准读 0 0 1: 快速读 0 1 0: 二线式输出快速读 0 1 1: 二线式输入输出快速读 1 0 0: 四线式输出快速读 1 0 1: 四线式输入输出快速读 1 1 0: 自定义标准读 1 1 1: 自定义快速读	R/W

28.12.2 QSPI 片选控制寄存器 (QSCSCR)

复位值: 0x0000_000F

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

位	标记	位名	功能	读写
b31~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5~b4	SSNW[1:0]	QSSN有效时间延长设置	QSPI总线访问后QSSN有效时间延长功能选择 b5 b4 0 0: 不延长QSSN有效时间 0 1: 将QSSN有效时间延长32个QSCK周期 1 0: 将QSSN有效时间延长128个QSCK周期 1 1: 将QSSN有效时间无限延长	R/W
b3~b0	SSHW[3:0]	QSSN最小无效时间设置	QSSN信号最小无效时间选择 b3 b2 b1 b0 0 0 0 0: 1个QSCK周期 0 0 0 1: 2个QSCK周期 0 0 1 0: 3个QSCK周期 0 0 1 1: 4个QSCK周期 0 1 0 0: 5个QSCK周期 0 1 0 1: 6个QSCK周期 0 1 1 0: 7个QSCK周期 0 1 1 1: 8个QSCK周期 1 0 0 0: 9个QSCK周期 1 0 0 1: 10个QSCK周期 1 0 1 0: 11个QSCK周期 1 0 1 1: 12个QSCK周期 1 1 0 0: 13个QSCK周期 1 1 0 1: 14个QSCK周期 1 1 1 0: 15个QSCK周期 1 1 1 1: 16个QSCK周期	R/W

28.12.3 QSPI 格式控制寄存器 (QSFCR)

复位值: 0x0000_8033

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DUTY	—	—	—	DMCYCN[3:0]	—	WPOL	SSNLD	SSNHD	—	Four_BIC	AWSL[1:0]				

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15	DUTY	占空比补正	QSCK输出波形占空比补正 0: 不进行占空比补正 1: 将QSCK的上升沿滞后0.5个HCLK周期 (当QSCK选择的频率是HCLK的奇数倍时有效)	R/W
b14~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11~b8	DMCYCN[3:0]	虚拟周期设置	使用快速读指令时虚拟周期数量选择 b3 b2 b1 b0 0 0 0 0: 3个QSCK周期*1 0 0 0 1: 4个QSCK周期 0 0 1 0: 5个QSCK周期 0 0 1 1: 6个QSCK周期 0 1 0 0: 7个QSCK周期 0 1 0 1: 8个QSCK周期 0 1 1 0: 9个QSCK周期 0 1 1 1: 10个QSCK周期 1 0 0 0: 11个QSCK周期 1 0 0 1: 12个QSCK周期 1 0 1 0: 13个QSCK周期 1 0 1 1: 14个QSCK周期 1 1 0 0: 15个QSCK周期 1 1 0 1: 16个QSCK周期 1 1 1 0: 17个QSCK周期 1 1 1 1: 18个QSCK周期	R/W
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6	WPOL	WP管脚输出电平设置	WP管脚 (QIO2) 电平设置 0: 低电平 1: 高电平	R/W
b5	SSNLD	QSSN信号输出时间延迟设置	QSSN信号输出时序选择 0: 比QSCK第一个上升沿提前0.5个QSCK输出QSSN 1: 比QSCK第一个上升沿提前1.5个QSCK输出QSSN	R/W

b4	SSNHD	QSSN信号释放时间延迟设置	QSSN信号释放时序选择 0: 比QSCK最后一个上升沿滞后0.5个QSCK释放QSSN 1: 比QSCK最后一个上升沿滞后1.5个QSCK释放QSSN	R/W
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	Four_BIC	4字节地址读指令代码选择	当地址宽度为4字节时的读指令代码选择 0: 不使用4字节地址读指令代码 1: 使用4字节地址读指令代码	R/W
b1~b0	AWSL[1:0]	地址宽度选择	串行接口地址宽度选择 b1 b0 0 0: 1字节 0 1: 2字节 1 0: 3字节 1 1: 4字节	R/W

*1 为避免 QIO0 端子在切换输入输出状态时发生冲突，在 QSSMD.QSOEX 位设置成 1 的情况下（处理许可信号延长 1 周期），请选择 4 个 QSPICK 周期以上的虚拟周期。

28.12.4 QSPI 状态寄存器 (QSSR)

复位值: 0x0000_8000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PFAN	PFFUL	—	PFNUM[4:0]				RAER	XIPF	—	—	—	—	—	—	BUSY

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15	PFAN	预读取动作状态	预读取动作状态信号 0: 预读取处于动作状态 1: 预读取处于停止状态	R
b14	PFFUL	预读取缓冲区状态	预读取缓冲区状态信号 0: 预读取缓冲区有剩余空间 1: 预读取缓冲区数据已满	R
b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12~b8	PFNUM[4:0]	预读取缓冲区已存储数据字节数	预读取缓冲区已存储数据字节数显示 b4 b3 b2 b1 b0 0 0 0 0 0: 0个字节 0 0 0 0 1: 1个字节 0 0 0 1 0: 2个字节 0 0 0 1 1: 3个字节 0 0 1 0 0: 4个字节 0 0 1 0 1: 5个字节 0 0 1 1 0: 6个字节 0 0 1 1 1: 7个字节 0 1 0 0 0: 8个字节 0 1 0 0 1: 9个字节 0 1 0 1 0: 10个字节 0 1 0 1 1: 11个字节 0 1 1 0 0: 12个字节 0 1 1 0 1: 13个字节 0 1 1 1 0: 14个字节 0 1 1 1 1: 15个字节 1 0 0 0 0: 16个字节 1 0 0 0 1: 17个字节 1 0 0 1 0: 18个字节 其余设定无效	R
b7	RAER*1	ROM访问错误标	直接通信模式下发生ROM访问的错误标志位	R/W

		志	0: 未检测到发生ROM访问 1: 检测到发生ROM访问	
b6	XIPF	XIP模式标志	XIP模式状态信号 0: 非XIP模式 1: XIP模式	R
b5~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	BUSY	总线忙标志	直接通信模式下QSPI总线工作状态标志位 0: 总线空闲, 无串行传输进程 1: 总线忙, 串行传输进程进行中	R

*1: RAER 需通过 QSSR2 的 RAERCLR 位清零

28.12.5 QSPI 指令代码寄存器 (QSCCMD)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”，写入时写“0”	R/W
b7~b0	RIC[7:0]	替换指令代码	用于替代默认指令的串行闪存指令代码	R/W

28.12.6 QSPI 直接通信指令寄存器 (QSDCOM)

复位值: 无

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

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”，写入时写“0”	R/W
b7~b0	DCOM[7:0]	直接通信模式指令	直接通信模式时的接口，通过QSPI总线进行直接通信。对该接口的读写访问会被转化成一个相应的QSPI总线周期。该接口只在直接通信模式下有效，ROM访问模式下该接口禁止访问。	R/W

28.12.7 QSPI XIP 模式代码寄存器 (QSXCMD)

复位值: 0x0000_00FF

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

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7~b0	XIPMC[7:0]	XIP模式代码	用于串行闪存的模式代码。(设置XIP模式)	R/W

28.12.8 QSPI 系统配置寄存器 (QSSR2)

复位值: 无

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

位	标记	位名	功能	读写
b31~b8	Reserved	-	写入时写“0”	W
b7	RAERCLR	RAER清除	写入1时将QSSR中的RAER位清零	W
b6~b0	Reserved	-	写入时写“0”	W

28.12.9 QSPI 外部扩展地址寄存器 (QSEXAR)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b26	EXADR[5:0]	外部扩展地址代码	QSPI外部地址高6位设置，配合QSPI的ROM访问窗口地址最大可访问64MB×63块的外部ROM空间	R/W
b25~b0	Reserved	-	读出时为“0”，写入时写“0”	R/W

29 集成电路内置音频总线模块 (I2S)

29.1 简介

I2S (Inter_IC Sound Bus), 集成电路内置音频总线, 该总线专责于音频设备之间的数据传输。

功能	主要特性
通信方式	<ul style="list-style-type: none"> 支持全双工和半双工通信 支持主模式或从模式操作
数据格式	<ul style="list-style-type: none"> 可选通道长度: 16/32位 可选传送数据长度: 16/24/32位 数据移位顺序: MSB开始
波特率	<ul style="list-style-type: none"> 8位可编程线性预分频器, 可实现精确的音频采样频率 支持采样频率192k, 96k, 48k, 44.1k, 32k, 22.05k, 16k, 8k 可输出驱动时钟以驱动外部音频元件, 比率固定为$256 \times F_s$(F_s为音频采样频率)
支持I2S协议	<ul style="list-style-type: none"> I2S Philips标准 MSB对齐标准 LSB对齐标准 PCM标准
数据缓冲	<ul style="list-style-type: none"> 带有2字深, 32位宽的输入输出FIFO缓冲区域
时钟源	<ul style="list-style-type: none"> 可使用内部 I2SCLK(UPLL/R/UPLL/Q/UPLL/P/MPLL/R/MPLL/Q/MPLL/P); 也可由 I2S_EXCK引脚上的外部时钟提供
中断	<ul style="list-style-type: none"> 发送缓冲区有效空间达到报警阈值时产生中断 接收缓冲区有效空间达到报警阈值时产生中断 接收数据区域已满仍有写入数据请求, 接收上溢 发送数据区域已空仍有发送请求, 发送下溢 发送数据区域已满仍有写入数据请求, 发送上溢

表 29-1 I2S 主要特性

29.2 I2S 系统框图

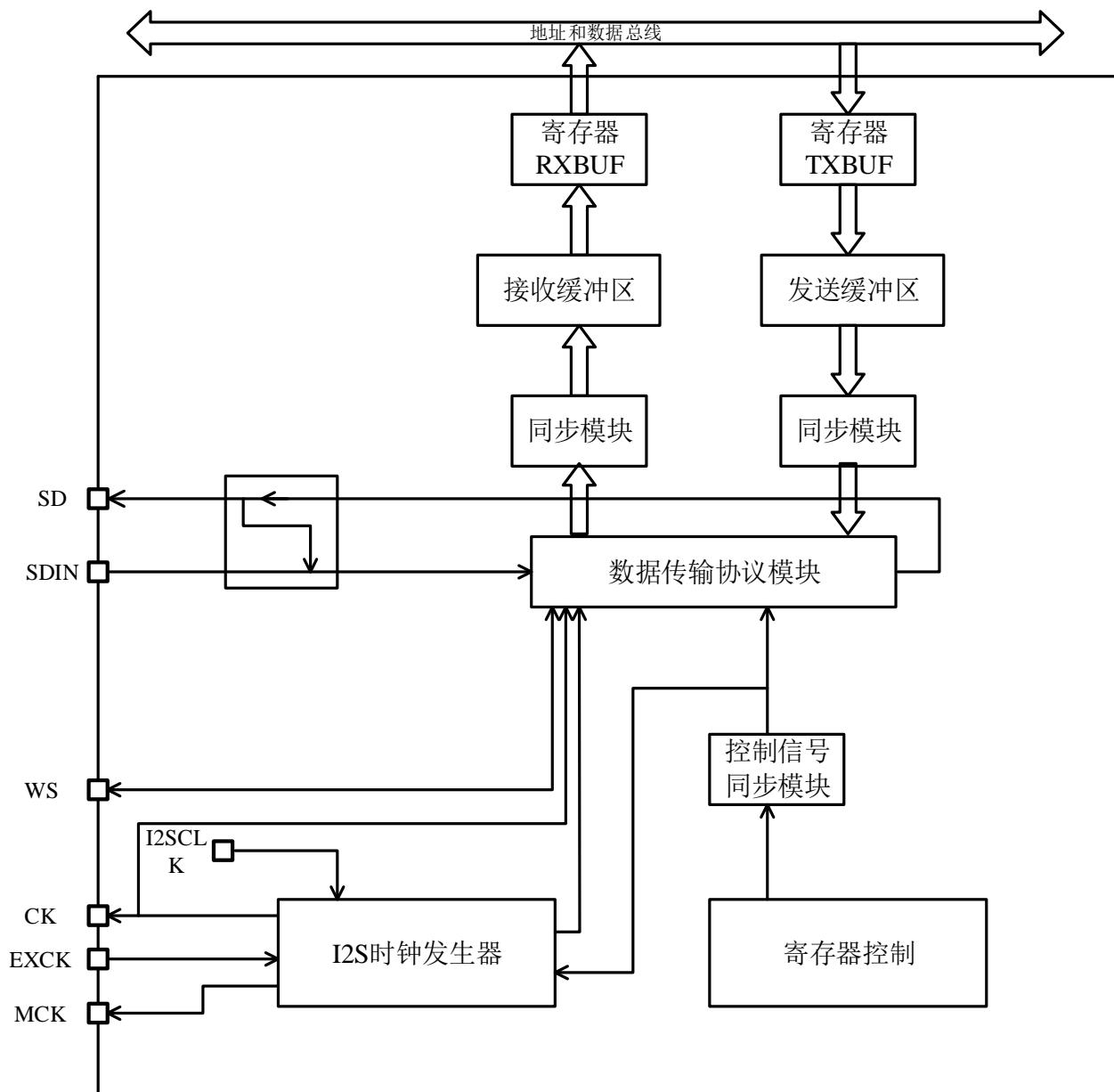


图 29-1 I2S 系统框图

29.3 管脚说明

管脚名	方向	功能描述
I2Sn_CK	输入输出	通讯时钟
I2Sn_WS	输入输出	字选择
I2Sn_SD	输入输出	串行数据
I2Sn_SDIN	输入	全双工音频数据输入
I2Sn_EXCK	输入	外部时钟源管脚
I2Sn_MCK	输出	驱动时钟

n:1~4

表 29-2 I2S 管脚说明

29.4 功能说明

本章将对 I2S 的功能进行详细说明。

29.4.1 I2S 一般说明

I2S 引脚功能

- I2Sn_SD: 串行数据, 用于半双工模式数据输入, 或半/全双工模式数据输出。
- I2Sn_WS: 字选择, 是主模式下的数据控制信号输出以及从模式下的数据控制信号输入。
- I2Sn_CK: 串行时钟, 是主模式下的串行时钟输出以及从模式下的串行时钟输入。
- I2Sn_EXCK: 外部时钟源, 是主模式下时钟发生器选择外部时钟作为分频时钟源。
- I2Sn_SDIN: I2S 全双工模式时用于串行数据输入的引脚。
- I2Sn_MCK: 当 I2S 配置为主模式 (并且 MCKOE 位置 1) 时, 使用驱动时钟 (单独映射) 输出此附加时钟, 该时钟输出频率 $256 \times F_s$, 其中 F_s 为音频信号采样频率。

I2S 在主模式下使用自身的时钟发生器生成通信时钟。此时钟发生器也是驱动时钟输出的源。

29.4.2 通信方式

支持半双工和全双工两种通信方式, 可通过 I2S 控制寄存器 (I2S_CTRL) 的 DUPLEX 位进行选择。

当 I2S_CTRL.DUPLEX 位为 0 时, I2S 运行在半双工通信方式, 只发送时使用 I2Sn_SD 引脚作为输出数据引脚, 只接收时使用 I2Sn_SD 引脚作为输入数据引脚。

当 I2S_CTRL.DUPLEX 位为 1 时, I2S 运行在全双工通信方式, 此时使用 I2Sn_SDIN 引脚作为输入数据引脚, I2Sn_SD 引脚作为输出数据引脚, 以实现全双工通信。

29.4.3 支持的音频协议

有四种数据和帧格式组合，可采用下列格式发送数据：

- 将 16 位数据封装在 16 位帧中
- 将 16 位数据封装在 32 位帧中
- 将 24 位数据封装在 32 位帧中
- 将 32 位数据封装在 32 位帧中

当使用 32 位数据包中的 16 位数据时，前 16 位(MSB)为有效位，16 位 LSB 被强制清零，无需任何软件操作（只需一个读/写操作）。

当使用 32 位数据包中的 24 位数据时，前 24 位(MSB)为有效位，8 位 LSB 被强制清零，无需任何软件操作（只需一个读/写操作）。

对于所有数据格式和通信标准而言，始终会先发送最高有效位（MSB 优先）。

I2S 接口支持四种音频协议，可使用 I2S_CFGR 寄存器中的 I2SSTD[1:0]和 PCMSYNC 位对其进行配置。

29.4.3.1 I2S Philips 标准

使用 WS 信号来指示当前正在发送的数据所属的通道。该信号从当前通道数据的第一个位(MSB)之前的一个时钟开始有效。

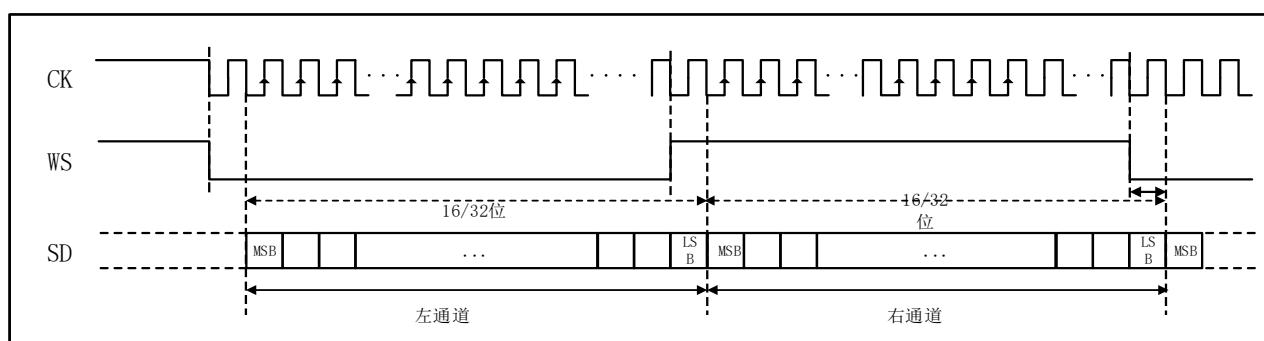


图 29-2 I2S Philips 协议波形 (16/32 位全精度)

16 位装载在 16 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXXXX_3344，SD 输出串行数据 0x3344；接收模式下，SD 输入串行数据 0xEEDD，I2S_RXBUF 寄存器读出数据 0x0000_EEDD。

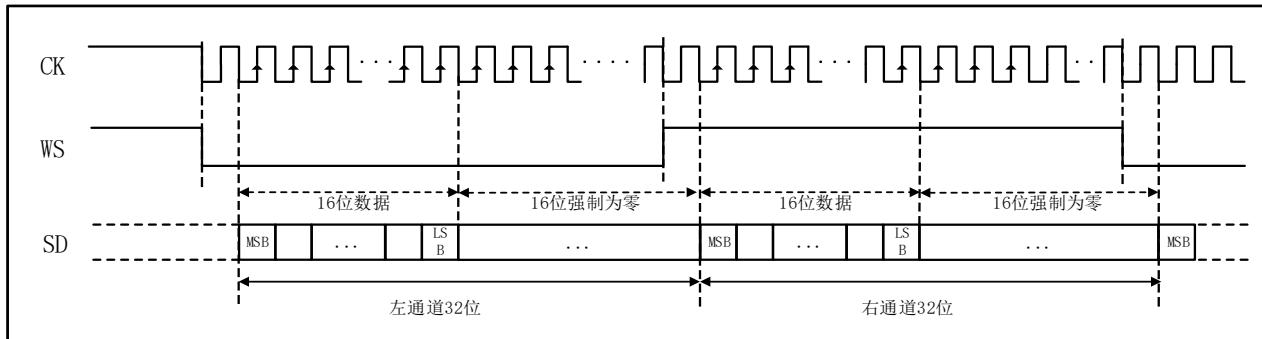


图 29-3 I2S Philips 协议波形 (16 位数据封装在 32 位帧中)

16 位装载在 32 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXXXX_3344，SD 输出串行数据 0x3344_0000；接收模式下，SD 输入串行数据 0xEEDD_XXXX，I2S_RXBUF 寄存器读出数据 0x0000_EEDD。

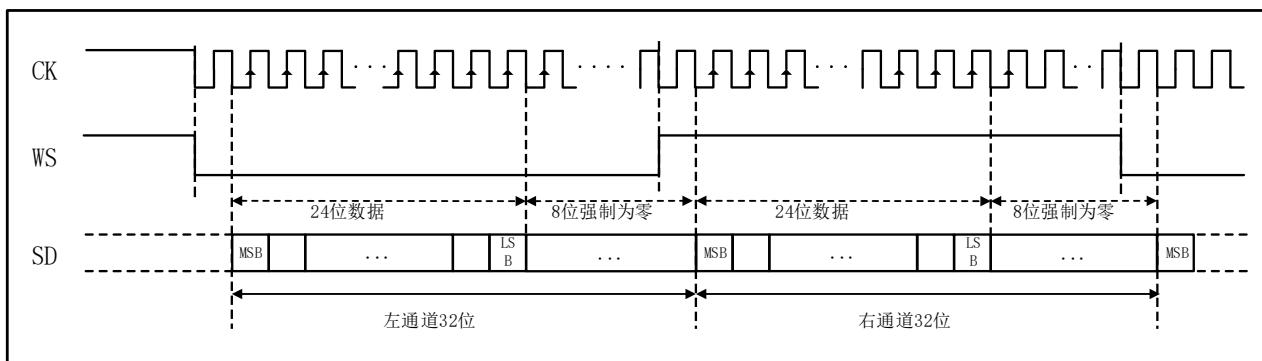


图 29-4 I2S Philips 协议波形 (24 位数据封装在 32 位帧中)

24 位装载在 32 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXX22_3344，SD 输出串行数据 0x2233_4400；接收模式下，SD 输入串行数据 0xEEDD_11XX，I2S_RXBUF 寄存器读出数据 0x00EE_DD11。

29.4.3.2 MSB 对齐标准

此标准同时生成 WS 信号和第一个数据位（即 MSBit）。

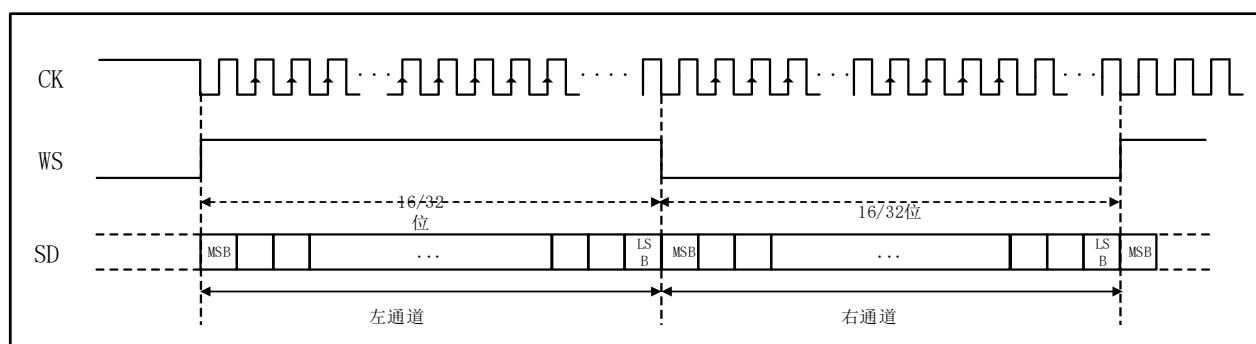


图 29-5 I2S MSB 协议波形 (16/32 位全精度)

发送方在时钟信号的下降沿改变数据；接收方在上升沿读取数据。

16位装载在16位帧，发送模式下，写入到I2S_TXBUF寄存器0xXXXX_3344，SD输出串行数据0x3344；接收模式下，SD输入串行数据0xEEDD，I2S_RXBUF寄存器读出数据0x0000_EEDD。

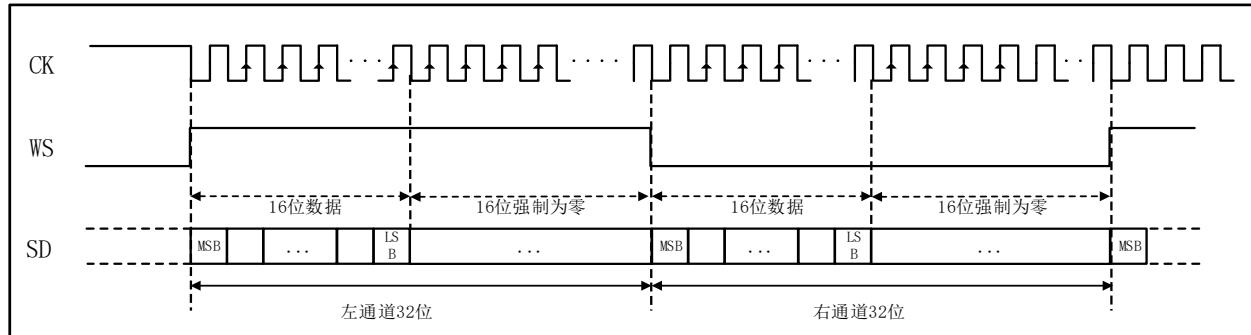


图 29-6 I2S MSB 协议波形（16 位数据封装在 32 位帧中）

16位装载在32位帧，发送模式下，写入到I2S_TXBUF寄存器0xXXXX_3344，SD输出串行数据0x3344_0000；接收模式下，SD输入串行数据0xEEDD_XXXX，I2S_RXBUF寄存器读出数据0x0000_EEDD。

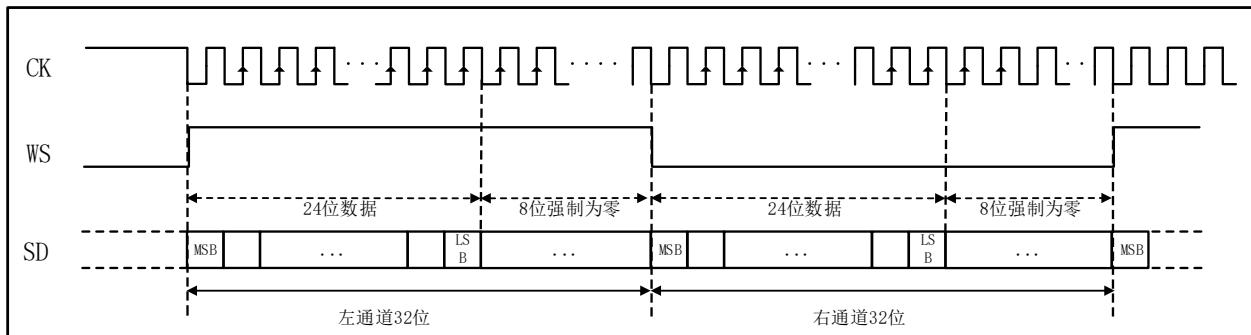


图 29-7 I2S MSB 协议波形（24 位数据封装在 32 位帧中）

24位装载在32位帧，发送模式下，写入到I2S_TXBUF寄存器0xXX22_3344，SD输出串行数据0x2233_4400；接收模式下，SD输入串行数据0xEEDD_11XX，I2S_RXBUF寄存器读出数据0x00EE_DD11。

29.4.3.3 LSB 对齐标准

该标准与MSB对齐标准类似（对于16位和32位全精度帧格式，没有任何不同）。

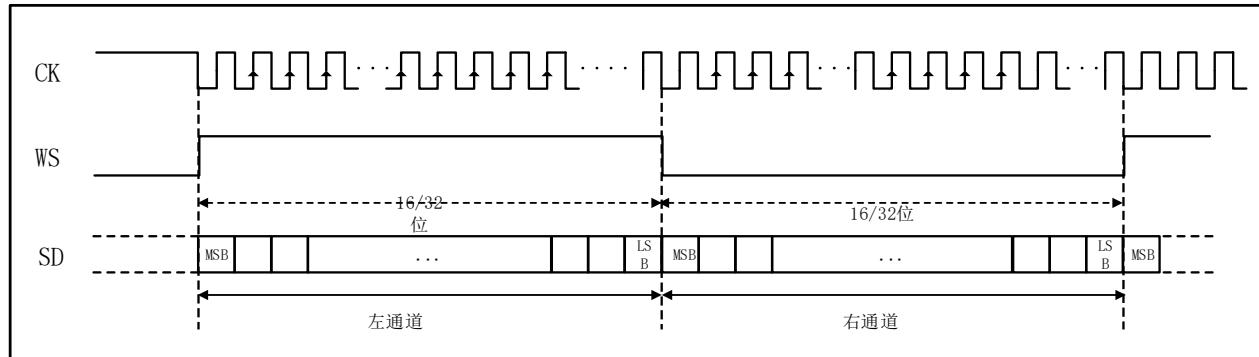


图 29-8 I2S LSB 协议波形 (16/32 位全精度)

16 位装载在 16 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXXXX_3344，SD 输出串行数据 0x3344；接收模式下，SD 输入串行数据 0xEEDD，I2S_RXBUF 寄存器读出数据 0x0000_EEDD。

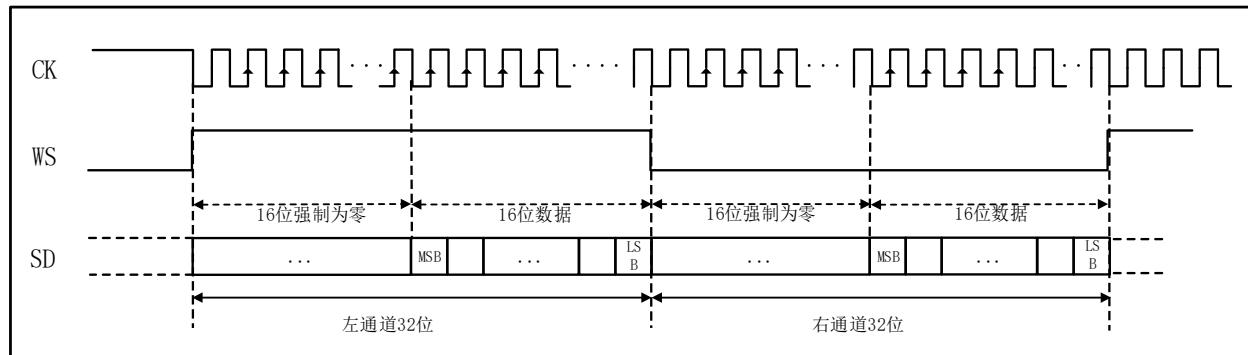


图 29-9 I2S LSB 协议波形 (16 位数据封装在 32 位帧中)

16 位装载在 32 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXXXX_3344，SD 输出串行数据 0x0000_3344；接收模式下，SD 输入串行数据 0xXXXX_1122，I2S_RXBUF 寄存器读出数据 0x0000_1122。

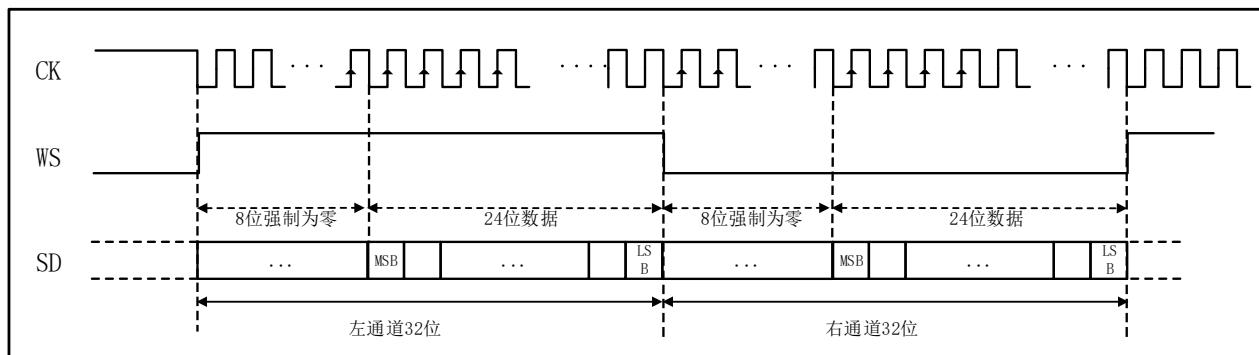


图 29-10 I2S LSB 协议波形 (24 位数据封装在 32 位帧中)

24 位装载在 32 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXX22_3344，SD 输出串行数据 0x0022_3344；接收模式下，SD 输入串行数据 0xXXDD_1122，I2S_RXBUF

寄存器读出数据 0x00DD_1122。

29.4.3.4 PCM 标准

对于 PCM 标准，无需使用通道信息。有两种 PCM 模式（短帧和长帧），并且可使用 I2S_CFGR 中的 PCMSYNC 位来配置。

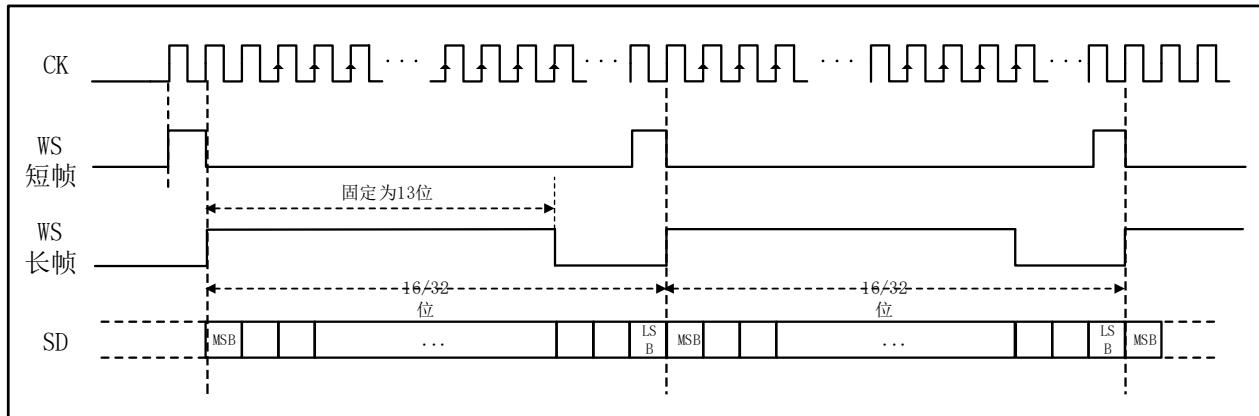


图 29-11 I2S PCM 协议波形 (16/32 位全精度)

对于长帧同步，在主模式下会将 WS 信号持续 13 个周期。

对于短帧同步，WS 同步信号的持续时间仅为一个周期。

16 位装载在 16 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXXXX_3344，SD 输出串行数据 0x3344；接收模式下，SD 输入串行数据 0xEEDD，I2S_RXBUF 寄存器读出数据 0x0000_EEDD。

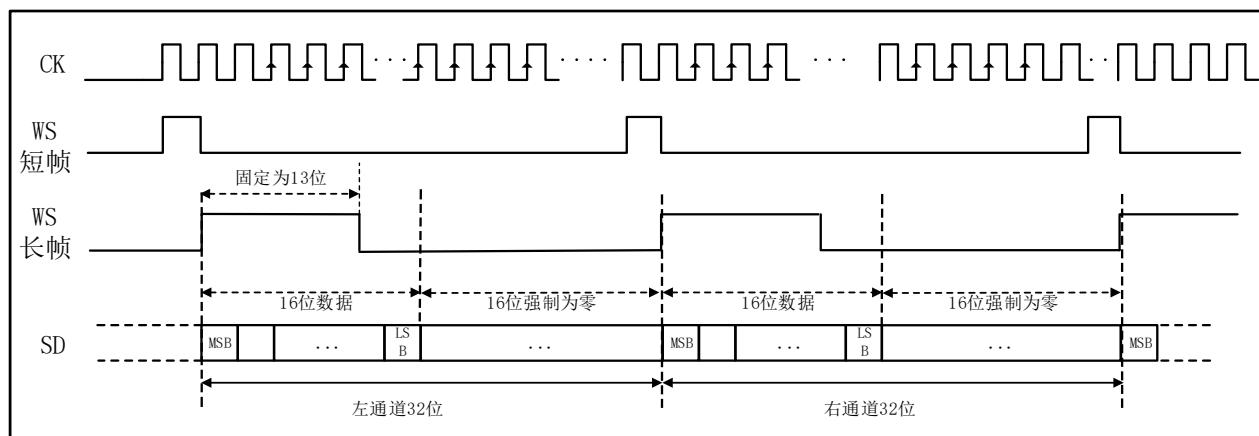


图 29-12 I2S PCM 协议波形 (16 位数据封装在 32 位帧中)

16 位装载在 32 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXXXX_3344，SD 输出串行数据 0x0000_3344；接收模式下，SD 输入串行数据 0xEEDD_XXXX，I2S_RXBUF 寄存器读出数据 0x0000_EEDD。

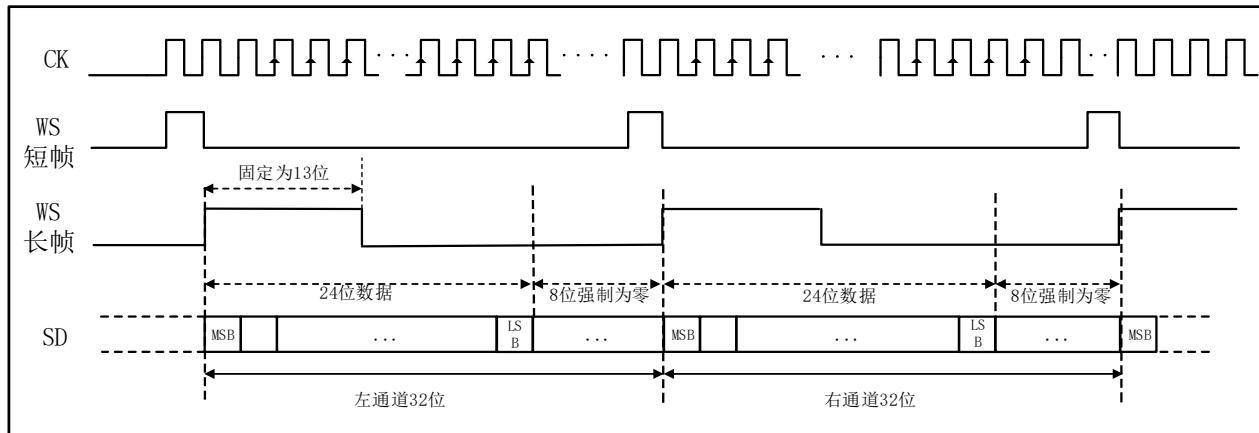


图 29-13 I2S PCM 协议波形（24 位数据封装在 32 位帧中）

24 位装载在 32 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXX22_3344，SD 输出串行数据 0x0022_3344；接收模式下，SD 输入串行数据 0xEEDD_11XX，I2S_RXBUF 寄存器读出数据 0x00EE_DD11。

注意：

- 对于两种模式（主/从模式）和两种同步（短/长同步），即使在从模式下，也需要指定两组连续数据（以及两个同步信号）之间位的个数（I2S_CFGR 寄存器中的 DATLEN 位和 CHLEN 位）。

29.4.4 时钟发生器

I2S 比特率用来确定 I2S 数据线上的数据流和 I2S 时钟信号频率。

I2S 比特率 = 每个通道的位数×通道数×音频采样频率

对于 16 位双通道音频，I2S 比特率的计算公式如下：I2S 比特率 = $16 \times 2 \times F_s$ 。如果数据包为 32 位宽，则 I2S 比特率 = $32 \times 2 \times F_s$ 。

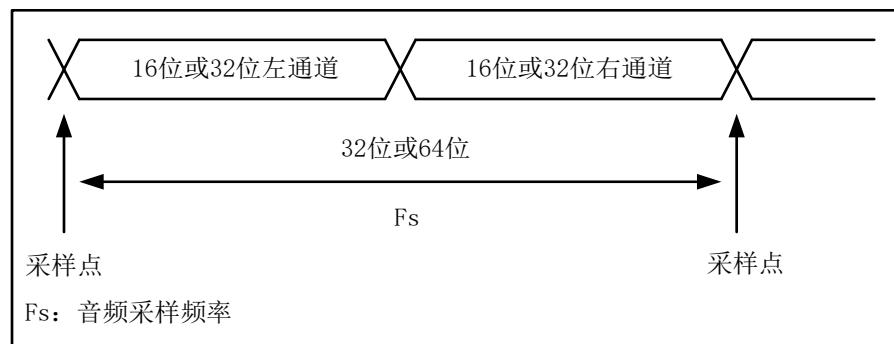


图 29-14 音频采样频率定义

配置主模式时，需要正确地对线性分频器进行设置，以便采用所需的音频频率进行通信。

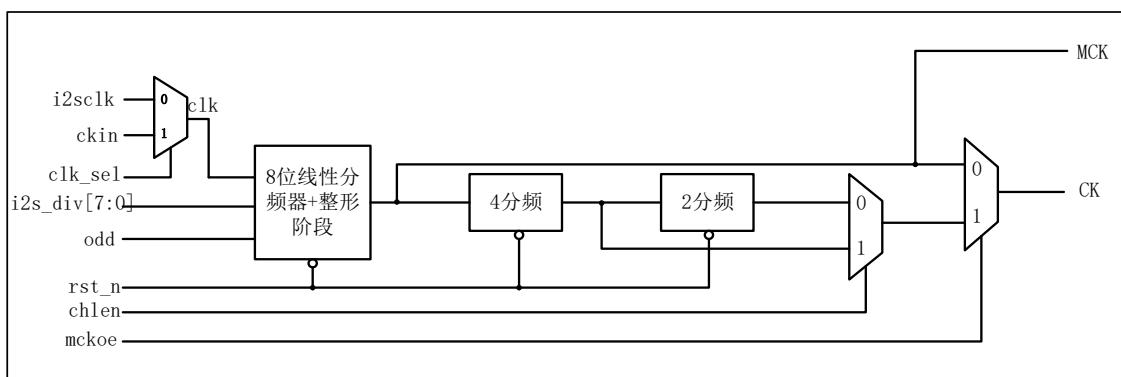


图 29-15 时钟发生器架构

clk 时钟源可为 I2SCLK 输出或外部时钟。

音频采样频率可能是 192kHz, 96kHz 或 48kHz 等。为达到所需频率，需要根据以下公式对现行分频器进行编程：

输出驱动时钟 (mckoe 置 1) 时：

$$F_s = \text{clk} / [(16 \times 2) \times ((2 \times \text{I2SDIV} + \text{ODD}) \times 8)] \quad (\text{通道帧宽度为 16 位时})$$

$$F_s = \text{clk} / [(32 \times 2) \times ((2 \times \text{I2SDIV} + \text{ODD}) \times 4)] \quad (\text{通道帧宽度为 32 位时})$$

关闭驱动时钟 (mckoe 清零) 时：

$$F_s = \text{clk} / [(16 \times 2) \times (2 \times \text{I2SDIV} + \text{ODD})] \quad (\text{通道帧宽度为 16 位时})$$

$F_s = \text{clk}/[(32 \times 2) \times (2 \times \text{I2SDIV} + \text{ODD})]$ (通道帧宽度为 32 位时)

下表提供了针对不同时钟配置的示例精度值。还有其他配置可达到更好的时钟精度。

驱动时钟	目标 fS (Hz)	数据格式	倍频系数	输出分频比	I2SDIV	ODD	实时 fS (Hz)	误差
输出关闭	8000	16位	288	3	187	1	8000	0.0000%
		32位	256	4	62	1	8000	0.0000%
	16000	16位	256	4	62	1	16000	0.0000%
		32位	256	2	62	1	16000	0.0000%
	32000	16位	256	2	62	1	32000	0.0000%
		32位	256	5	12	1	32000	0.0000%
	48000	16位	384	10	12	1	48000	0.0000%
		32位	384	5	12	1	48000	0.0000%
	96000	16位	384	5	12	1	96000	0.0000%
		32位	424	3	11	1	96014.49219	0.0151%
	22050	16位	290	3	68	1	22049.87695	0.0006%
		32位	302	2	53	1	22050.23438	0.0011%
	44100	16位	302	2	53	1	44100.46875	0.0011%
		32位	429	4	19	0	44099.50781	0.0011%
	192000	16位	424	3	11	1	192028.9844	0.0151%
		32位	258	3	3	1	191964.2813	0.0186%
输出使能	8000	无关	256	5	12	1	8000	0.0000%
	16000	无关	426	4	13	0	16000.60059	0.0038%
	32000	无关	426	4	6	1	32001.20117	0.0038%
	48000	无关	258	3	3	1	47991.07031	0.0186%
	96000	无关	344	2	3	1	95982.14063	0.0186%
	22050	无关	429	4	9	1	22049.75291	0.0011%
	44100	无关	271	2	6	0	44108.07422	0.0183%

表 29-3 音频频率精度(针对 VCO 输入频率=1MHz)

注意：

- 倍频系数可为 UPLL_N/MPLL_N，倍频系数为 UPLL_N 时输出分频比可为 UPLL_P/UPLL_Q/UPLL_R，倍频系数为 MPLL_N 时输出分频比可为 MPLL_P/MPLL_Q/MPLL_R。具体设置参考时钟控制器（CMU）的 CMU_UPLL 配置寄存器和 CMU_MPLL 配置寄存器。

29.4.5 I2S 主模式

I2S 可如下配置：

- 发送主器件或接收主器件（使用 I2S 的半双工模式）
- 同时收发的主器件（使用 I2S 的全双工模式）。

I2S 工作在主模式，串行时钟由引脚 CK 输出，字选信号由引脚 WS 产生。可以通过设置寄存器 I2S_CTRL 的 MCKOE 位来选择输出或者不输出驱动时钟(MCK)。

步骤

1. 设定 I2S 所需要使用的管脚。
2. 通过 I2S_CTRL.CLKSEL, I2S_CTRL.I2SPLLSEL 位，选择时钟源。
3. 设置 I2S_PR.I2SDIV[7:0]位和 I2S_CTRL.ODD 位，以定义串行数据波特率，从而达到适当的音频采样频率。
4. 设置 I2S 配置寄存器(I2S_CFGR)，通过 I2SSTD[1:0]和 PCMSYNC 位选择 I2S 标准，通过 DATLEN[1:0]位选择数据长度并通过配置 CHLEN 位选择每个通道的位数。
5. 如需要使用中断，请设置系统的中断寄存器。
6. 如需使用 DMA，请设置 DMA 的相关寄存器。
7. 设定 I2S 控制寄存器(I2S_CTRL)，包括有工作模式设定，通信方式设定，时钟输出许可设定，数据输出许可设定，FIFO 复位设定，发送/接收缓冲阈值设定等，工作模式 WMS 位选择 I2S 主模式。
8. 设置中断许可位。
9. 设置 I2S_CTRL.TXE 和 I2S_CTRL.RXE，动作开始。

注意：

- 使用 TXIRQOUT 中断写通信数据到 TXBUF 中时，若每次中断写入两个数据时，在中

断开始后首先关掉发送中断使能标志位 TXIE，写完数据后，再打开 TXIE。或者每次中断只写入一个数据。

发送序列

I2S_CTRL 寄存器中的 TXE 位置 1，允许发送。将数据写入发送缓冲区后，发送序列随即开始。一个完整帧表示先进行左通道数据发送再进行右通道数据发送。不存在仅发送左通道的部分帧。首位发送期间，数据并行加载到移位寄存器中，然后以串行方式移位并输出到 SD 引脚 (MSB 在前)。有关各种 I2S 标准模式中的写操作的更多详细信息，请参见 29.4.3 支持的音频协议。

接收序列

此工作模式与发送模式基本相同，只有在 I2S_CTRL 寄存器的发送接收设置上不同，将 RXE 位置 1，允许接收。有关各种 I2S 标准模式中的读操作的更多详细信息，请参见 29.4.3 支持的音频协议。如果在先前收到的数据尚未读取时又接收到新数据，将产生溢出错误并将 I2S_ER.RXERR 标志置 1。如果 I2S_CTRL.EIE 位置 1，将产生中断以指示该错误。

29.4.6 I2S 从模式

I2S 可如下配置：

- 发送从器件或接收从器件 (使用 I2S 的半双工模式)
- 同时收发的从器件 (使用 I2S 的全双工模式)。

此工作模式所遵循的规则与 I2S 主模式基本相同。在从模式下，I2S 接口不产生时钟。时钟和 WS 信号从 I2S 接口所连接的外部主器件输入。这样，用户便不需要配置时钟。

步骤

1. 设定 I2S 所需要使用的管脚。
2. 设置 I2S 配置寄存器(I2S_CFGR)，通过 I2SSTD[1:0]和 PCMSYNC 位选择 I2S 标准，通过 DATLEN[1:0]位选择数据长度并通过配置 CHLEN 位选择每个通道的位数。
3. 如需要使用中断，请设置系统的中断寄存器。
4. 如需使用 DMA，请设置 DMA 的相关寄存器。

5. 若要发送数据，应先向 I2S_TXBUF 中预先写入 1~4 个待发送数据。
6. 设定 I2S 控制寄存器(I2S_CTRL)，包括有工作模式设定，通信方式设定，时钟输出许可设定，数据输出许可设定，FIFO 复位设定，发送/接收缓冲阈值设定等，工作模式 WMS 位选择 I2S 从模式。
7. 设置中断许可位。
8. 设置 I2S_CTRL.TXE 和 I2S_CTRL.RXE，动作开始。

发送序列

当 WMS 置 1，TXE 置 1，外部主器件发送时钟并且通过 WS 信号请求传输数据时，发送序列开始。通信开始时，数据从发送缓冲区传输到移位寄存器。首位发送期间，数据从内部总线并行加载到移位寄存器中，然后以串行方式移位并输出到 SD 引脚 (MSB 在前)。每次发送缓冲 FIFO 空间大于设置的阈值时，如果 I2S_CTRL.TXIE 位置 1，将产生中断。有关各种 I2S 标准模式中的写操作的更多详细信息，请参见 29.4.3 支持的音频协议。

为确保连续进行音频数据发送，必须在当前数据发送结束前将下一个要发送数据写入 TX FIFO。若在数据尚未写入 TX FIFO 时下一个数据通信的首个时钟边沿到来，将产生发送下溢，I2S_ER.TXERR 标志将置 1 并可能产生中断。如果 I2S_CTRL.EIE 位置 1，则当 I2S_ER.TXERR 标志变为 1 时，将产生中断。

接收序列

此工作模式与发送模式基本相同，只有在 I2S_CTRL 寄存器的发送接收设置上不同，将 RXE 位置 1，允许接收。有关各种 I2S 标准模式中的读操作的更多详细信息，请参见 29.4.3 支持的音频协议。如果在先前收到的数据尚未读取时又接收到新数据，将产生接收溢出错误并将 RXERR 标志置 1。如果 I2S_CTRL.EIE 位置 1，将产生中断以指示该错误。

29.4.7 I2S 中断

I2S 的中断源有发送缓冲区有效空间大于报警阈值、接收缓冲区有效空间小于报警阈值、接收上溢、发送下溢和发送上溢。发送下溢和发送上溢被集成为 I2S 发送错误中断 TXERR，所以需要通过标志判断实际发生的中断源。I2S 中断源的具体说明如表

29-4 所示。一旦中断条件成立，就产生相应的中断请求。

用户可将上述事件触发源对应的向量写入不同的触发对象寄存器实现各种事件触发功能。

上述事件触发源对应的向量请参考【中断控制器（INTC）】。

表 29-4 为 I2S 中断的列表。

中断事件	事件标志	使能控制位
发送缓冲区有效空间大于报警阈值	TXBA	TXIE
接收缓冲区有效空间小于报警阈值	RXBA	RXIE
接收数据区域已满仍有写入数据请求，接收上溢	RXERR	EIE
发送数据区域已空仍有发送请求，发送下溢	TXERR	EIE
发送数据区域已满仍有写入数据请求，发送上溢	TXERR	EIE

表 29-4 I2S 中断请求

29.4.8 使用上的注意事项

29.4.8.1 作为主机使用时的注意事项

- 1) 当 I2S 作为主机进行数据只发送动作时，如果 I2S_TXBUF 中所有的数据都已发送完成而没有新的数据写入时，I2S 将在最后一个数据发送完成后暂停动作，此时 I2S 将不再产生通讯时钟，发送错误标志位 TXERR 将被置位成 1。此时用户可选择将 I2S_CTRL.TXE 位写成 0 来关闭 I2S，或者向 I2S_TXBUF 写入新的发送数据继续发送动作。继续发送时 WS 将重新从左声道开始（Philips、MSB/LSB 模式）。
如果在发送动作过程中直接将 I2S_CTRL.TXE 位写成 0，则 I2S 将立刻被关闭，当前的数据发送被终止。这种做法会导致从机状态无法把握，在不复位从机的情况下重新开始通讯时造成从机接收数据混乱，因此建议用户在 I2S 处于暂停状态时将 I2S_CTRL.TXE 位写成 0 来关闭 I2S。
- 2) 当 I2S 作为主机进行数据只接收动作时，如果希望暂时停止接收动作，可预先写入两帧 dummy 数据，需要暂停时计数到相应位置将 I2S_CTRL.TXE 置 1，波特率为 8k~96k 时提前 4 个数据 TXE 置 1，波特率为 192k 时提前 5 个数据 TXE 置 1，当这两帧 dummy 数据发送后 I2S 会暂停动作，此时 I2S 将不再产生通讯时钟，发送错误标志位 TXERR 将被置位成 1。此时用户可选择将 I2S_CTRL.TXE 和 I2S_CTRL.RXE 清零来关闭 I2S，或者向 I2S_TXBUF 再写入一帧 dummy 数据重启通讯动作，在暂停后重新开始接收第一个数据时关掉 TXE，重启动作时 WS 将重新从左声道开始（Philips、MSB/LSB 模式）。当通讯时钟重新产生后将 I2S_CTRL.TXE 清零，即可重新回到只接收状态。详细请参照下图。

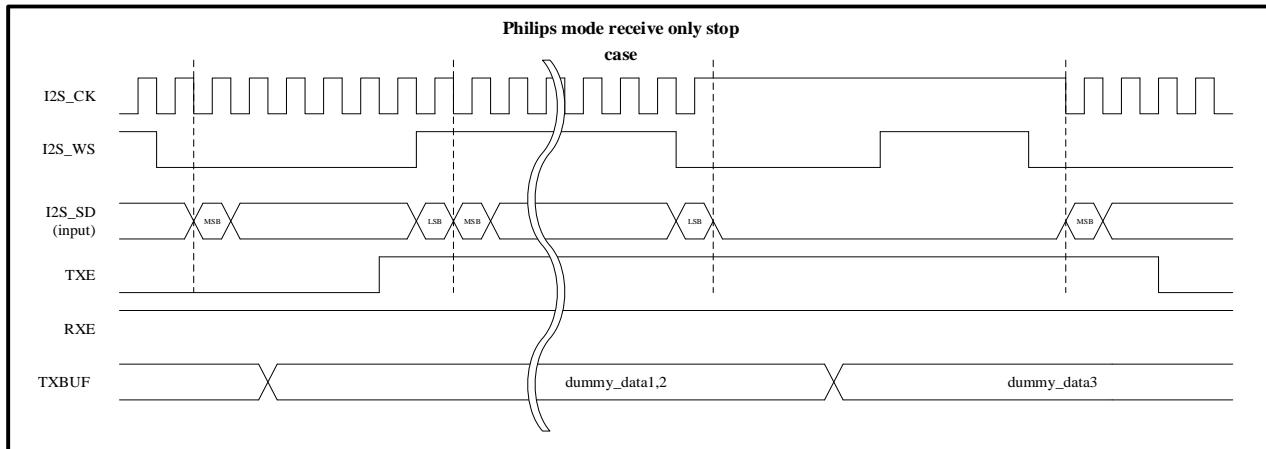


图 29-16 主机只接收暂时停止接收

如果在接收动作过程中直接将 I2S_CTRL.RXE 位写成 0，则 I2S 将立刻被关闭，当前的数据接收被终止。这种做法会导致从机状态无法把握，在不复位从机的情况下重新开始通讯时造成从机发送数据混乱，因此建议用户在 I2S 处于暂停状态时将 I2S_CTRL.TXE 和 I2S_CTRL.RXE 位清零来关闭 I2S。

- 3) 当 I2S 作为主机进行全双工动作时，如果 I2S_TXBUF 中所有的数据都已发送完成而没有新的数据写入时，I2S 将在最后一个数据发送完成后暂停动作，此时 I2S 将不再产生通讯时钟，发送错误标志位 TXERR 将被置位成 1。此时用户可选择将 I2S_CTRL.TXE 和 I2S_CTRL.RXE 清零来关闭 I2S，或者向 I2S_TXBUF 写入新的发送数据继续发送和接收动作。继续发送时 WS 将重新从左声道开始（Philips、MSB/LSB 模式）。

如果在全双工动作过程中直接将 I2S_CTRL.TXE 和 I2S_CTRL.RXE 位写成 0，则 I2S 将立刻被关闭，当前的数据发送及接收被终止。这种做法会导致从机状态无法把握，在不复位从机的情况下重新开始通讯时造成通讯数据混乱，因此建议用户在 I2S 处于暂停状态时将 I2S_CTRL.TXE 和 I2S_CTRL.RXE 位清零来关闭 I2S。

- 4) 当 I2S 作为主机进行 PCM 短帧数据发送动作时，当 I2S 因为 I2S_TXBUF 没有新的数据写入动作暂停后，有两种设置方式可以重新开始发送，具体选择哪一种方式需要根据从机的数据接收规格来确定。

如果从机每次接收数据都要检测 WS 的状态，则 I2S 暂停后需要先将 I2S_CTRL.TXE 置 0，然后向 I2S_TXBUF 写入新的发送数据后再把 I2S_CTRL.TXE 置 1 来重新开始传送。具体动作如下图所示。

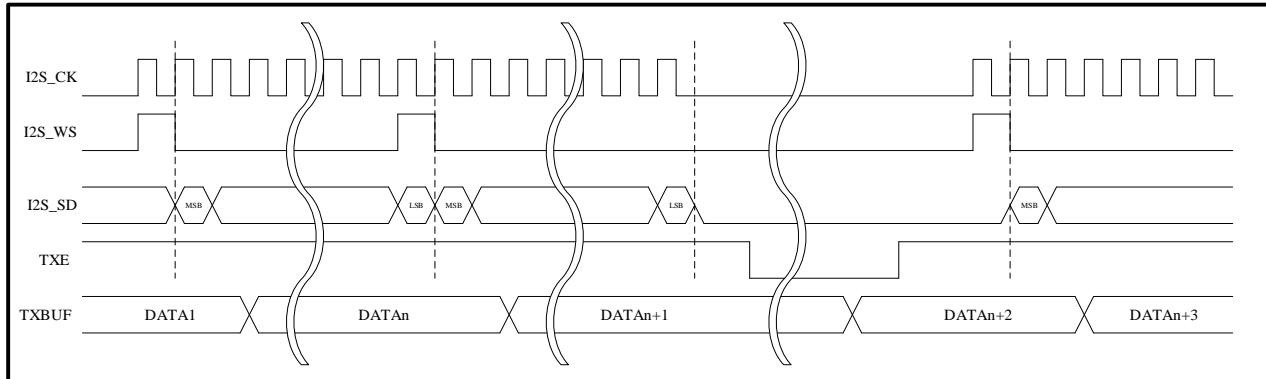


图 29-17 PCM 短帧主机发送暂停后重新发送方式一

如果从机只会在接收第一帧数据的时候才检测 WS 状态，则 I2S 暂停后可直接向 I2S_TXBUF 写入新的发送数据来重新开始传送。具体动作如下图所示。

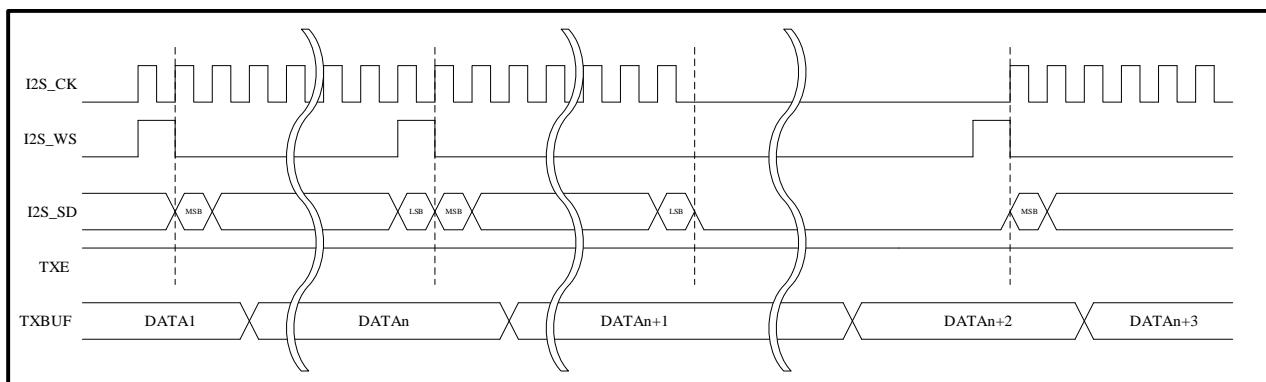


图 29-18 PCM 短帧主机发送暂停后重新发送方式二

29.4.8.2 作为从机使用时的注意事项

- 1) 当 I2S 作为从机动作时，需要确保在所有的寄存器配置都完成之后，最后再打开 I2S_CTRL.TXE 或者 I2S_CTRL.RXE 来启动从机动作。
- 2) 当 I2S 在 Philips、MSB/LSB 模式下启动从机动作时，需要保证启动时 WS 信号处于右声道电平状态，当 I2S 在 PCM 模式下启动从机动作时，需要保证启动时 WS 信号处于低电平状态。
- 3) 当 I2S 作为从机进行数据接收时，每次接收到的数据将会在下一帧数据接收开始后才能被读取，因此在通讯暂停或终止时，I2S 接收到的最后一帧数据将会在下次通讯开始时才能被读到。
- 4) 当 I2S 在 Philips、MSB/LSB 模式下作为从机进行数据接收动作时，每帧左声道数据接收时会检验 WS 是否为左声道电平。当 I2S 在 PCM 模式下作为从机进行数据接收时，每帧数据接收前都会检验 WS 是否按标准通讯协议产生了有效电平。

29.5 寄存器说明

I2S1 基准地址: 0x4001_E000

I2S2 基准地址: 0x4001_E400

I2S3 基准地址: 0x4002_2000

I2S4 基准地址: 0x4002_2400

寄存器名	符号	偏移地址	位宽	复位值
I2S控制寄存器	I2S_CTRL	0x000	32	0x0000_2200
I2S状态寄存器	I2S_SR	0x004	32	0x0000_0014
I2S错误状态寄存器	I2S_ER	0x008	32	0x0000_0000
I2S配置寄存器	I2S_CFGR	0x00C	32	0x0000_0000
I2S发送缓冲FIFO数据寄存器	I2S_TXBUF	0x010	32	0x0000_0000
I2S接收缓冲FIFO数据寄存器	I2S_RXBUF	0x014	32	0x0000_0000
I2S预分频寄存器	I2S_PR	0x018	32	0x0000_0002

表 29-5 I2S 寄存器一览表

注：只支持 32 位写寄存器

CMU_BASE_ADDR2 : 0x40054000

寄存器名	符号	偏移地址	位宽	复位值
CMU I2S时钟配置寄存器	CMU_I2SCKSEL	0x012	16	0xB BBBB

注：该寄存器详见 CMU 章节。I2S 主模式时钟源选择 I2SPPLL 时使用该寄存器配置时钟源，可配置为 UPLL/R/UPLL/Q/UPLL/P/MPLL/R/MPLL/Q/MPLL/P。

29.5.1 I2S 控制寄存器 (I2S_CTRL)

I2S Control Register

偏移地址: 0x000

复位值: 0x0000_2200

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	CLKSEL L	DUPLEX X	CKOE	LRCKOE	SDOE	I2SPLL SEL	CODEC RC	FIFOR
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	RXBIRQWL		-	TXBIRQWL		MCKOE E	ODD	WMS	EIE	RXIE	RXE	TXIE	TXE		

位	标记	位名	功能	读写
b31~b27	Reserved	-	读出时为“0”，写入时写“0”	R
b23	CLKSEL	时钟源选择	0: 选择I2SPLL 1: 选择外部时钟	R/W
b22	DUPLEX	通信方式选择	0: 半双工 1: 全双工	R/W
b21	CKOE	通讯时钟输出许可	0: 输出禁止 1: 输出许可	R/W
b20	LRCKOE	声道时钟输出许可	0: 输出禁止 1: 输出许可	R/W
b19	SDOE	数据输出许可	0: 输出禁止 1: 输出许可	R/W
b18	I2SPLLSEL	I2SPLL输入选择	0: 输入禁止 1: 输入许可	R/W
b17	CODECRC	编解码器复位控制	0: 软件复位 1: 解除复位	R/W
b16	FIFOR	fifo复位	0: 解除复位 1: 软件复位	R/W
b15	Reserved	-	读出时为“0”，写入时写“0”	R
b14~b12	RXBIRQWL[2:0]	接收缓冲中断请求水平	当可用空间少于设定值时中断请求触发 注：只能设为0/1/2，因为fifo空间为2	R/W
b11	Reserved	-	读出时为“0”，写入时写“0”	R
b10~b8	TXBIRQWL[2:0]	发送缓冲中断请求水平	当可用空间多于设定值时中断请求触发 注：只能设为0/1/2，因为fifo空间为2	R/W
b7	MCKOE	驱动时钟输出使能	0: 禁止驱动时钟输出 1: 使能驱动时钟输出 注：只有在I2S主模式时才会使用此位	R/W

b6	ODD	预分频器奇数因子	0: 实际分频值=I2SDIV×2 1: 实际分频值=I2SDIV×2+1 注: 只有在I2S主模式时才会使用此位, 若要设置ODD为1, 应先设置I2S_PR寄存器, 再设置I2S_CTRL寄存器。	R/W
b5	WMS	I2S工作模式选择	0: I2S主机模式 1: I2S从机模式	R/W
b4	EIE	通讯错误中断使能	0: 通讯错误中断无效 1: 通讯错误中断有效	R/W
b3	RXIE	接收中断使能	0: 接收中断无效 1: 接收中断有效	R/W
b2	RXE	接收使能	0: 禁止接收 1: 允许接收	R/W
b1	TXIE	发送中断使能	0: 发送中断无效 1: 发送中断有效	R/W
b0	TXE	发送使能	0: 禁止发送 1: 允许发送	R/W

29.5.2 I2S 状态寄存器 (I2S_SR)

I2S Status Register

偏移地址: 0x004

复位值: 0x0000_0014

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

位	标记	位名	功能	读写
b31~b6	Reserved	-	读出时为“0”,写入时写“0”	R
b5	RXBF	接收缓冲满	0: 接收缓冲区未满 1: 接收缓冲区满	R
b4	RXBE	接收缓冲空	0: 接收缓冲区非空 1: 接收缓冲区空	R
b3	TXBF	发送缓冲满	0: 发送缓冲区未满 1: 发送缓冲区满	R
b2	TXBE	发送缓冲空	0: 发送缓冲区非空 1: 发送缓冲区空	R
b1	RXBA	接收缓冲报警 (与water level有关)	0: 接收缓冲区未报警 1: 接收缓冲区报警	R
b0	TXBA	发送缓冲报警 (与water level有关)	0: 发送缓冲区未报警 1: 发送缓冲区报警	R

29.5.3 I2S 错误状态寄存器 (I2S_ER)

I2S Error Status Register

偏移地址: 0x008

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b2	Reserved	-	读出时为“0”，写入时写“0”	R
b1	RXERR	接收错误	0: 不清除标志位 1: 清除标志位	R/W
b0	TXERR	发送错误	0: 不清除标志位 1: 清除标志位	R/W

当发生发送上溢/下溢时 TXERR=1，发生接收上溢时 RXERR=1。

发生发送上溢/下溢时对 TXERR 位写 1 可清除标志位，发生接收上溢时对 RXERR 位写 1 可清除标志位。

29.5.4 I2S 配置寄存器 (I2S_CFGR)

I2S Configuration Register

偏移地址: 0x00C

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	PCMSY NC	CHLEN	DATLEN[1:0]	I2SSTD[1:0]		

位	标记	位名	功能	读写
b31~b6	Reserved	-	读出时为“0”, 写入时写“0”	R
b5	PCMSYNC	PCM帧同步	PCM帧同步 0: 短帧同步 1: 长帧同步 注: 只有在I2SSTD=11(使用PCM标准), 此位才有意义	R/W
b4	CHLEN	通道长度	单声道一帧数据长度选择 0: 16bit 1: 32bit	R/W
b3~b2	DATLEN[1:0]	传送数据长度选择	传送数据长度选择 00: 16bit 01: 24bit 1X: 32bit	R/W
b1~b0	I2SSTD[1:0]	通讯协议选择	通讯协议选择 00: Philips 协议 01: MSB justified 协议 (左对齐) 10: LSB justified 协议 (右对齐) 11: PCM 协议	R/W

29.5.5 I2S 发送缓冲 FIFO 数据寄存器 (I2S_TXBUF)

I2S Transmit Buffer FIFO Data Register

偏移地址: 0x010

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b0	TXBUF[31:0]	发送数据	存储发送数据	W

注:

- 16 位帧时, TXBUF[15:0]存储一帧左通道或一帧右通道发送数据。
- 32 位帧时, TXBUF[31:0]存储一帧左通道或一帧右通道发送数据。

29.5.6 I2S 接收缓冲 FIFO 数据寄存器 (I2S_RXBUF)

I2S Receive Buffer FIFO Data Register

偏移地址: 0x014

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b0	RXBUF [31:0]	接收数据	存储接收数据	R

注:

- 16 位帧时, RXBUF[15:0]存储一帧左通道或一帧右通道接收数据。
- 32 位帧时, RXBUF[31:0]存储一帧左通道或一帧右通道接收数据。

29.5.7 I2S 分频寄存器 (I2S_PR)

I2S Prescaler Register

偏移地址: 0x018

复位值: 0x0000_0002

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	I2SDIV[[7:0]]

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R
b7~b0	I2SDIV[7:0]	分频因子	I2SDIV[7:0]=0或I2SDIV[7:0]=1为禁用值 参见4.4时钟发生器模块 实际分频值=I2SDIV×2+I2S_CTRL.ODD 00000010: 2分频 00000011: 3分频 00000100: 4分频 11111101: 253分频 11111110: 254分频 11111111: 255分频 注: 只有在I2S主模式时才会使用此位	R/W

30 控制器局域网（CAN）

30.1 简介

CAN (Controller Area Network) 总线是一种可以在无主机情况下实现微处理器或者设备之间相互通信的总线标准。本模块遵循 CAN 总线协议 2.0A 和 2.0B 协议并向上兼容 CAN-FD。CAN 总线控制器可以处理总线上的数据收发，在本产品中，CAN 具有 8 组筛选器。筛选器用于为应用程序选择要接收的消息。

应用程序通过 1 个高优先级的主发送缓冲器 (Primary Transmit Buffer, 以下简称 PTB) 和 4 个辅发送缓冲器 (Secondary Transmit Buffer, 以下简称 STB) 将发送数据送至总线，由发送调度器决定邮箱发送顺序。通过 10 个接收缓冲器 (Receive Buffer, 以下简称 RB) 获取总线数据。4 个 STB 以及 10 个 RB 可以理解为一个 4 级 FIFO 和一个 10 级 FIFO，FIFO 完全由硬件控制。

CAN 总线控制器同时也可以支持时间触发 CAN 通信 (Time-trigger communication)。

CAN 主要特性：

- 完全支持 CAN2.0A/CAN2.0B 协议。
- 向上兼容 CAN-FD 协议。
- 支持最高通信波特率 1Mbit/s
- 支持 1~1/256 的波特率预分频，灵活配置波特率。
- 10 个接收缓冲器
- FIFO 方式
 - 错误或者不被接收的数据不会覆盖存储的消息
 - 1 个高优先主发送缓冲器 PTB
 - 4 个副发送缓冲器 STB
 - FIFO 方式
 - 优先级仲裁方式
 - 8 组独立的筛选器
 - 支持 11 位标准 ID 和 29 位扩展 ID
 - 可编程 ID CODE 位以及 MASK 位

- PTB/STB 均支持单次发送模式
- 支持静默模式
- 支持回环模式
- 支持捕捉传输的错误种类以及定位仲裁失败位置
- 可编程的错误警告值
- 支持 ISO11898-4 规定时间触发 CAN 以及接收时间戳

30.2 CAN 系统框图

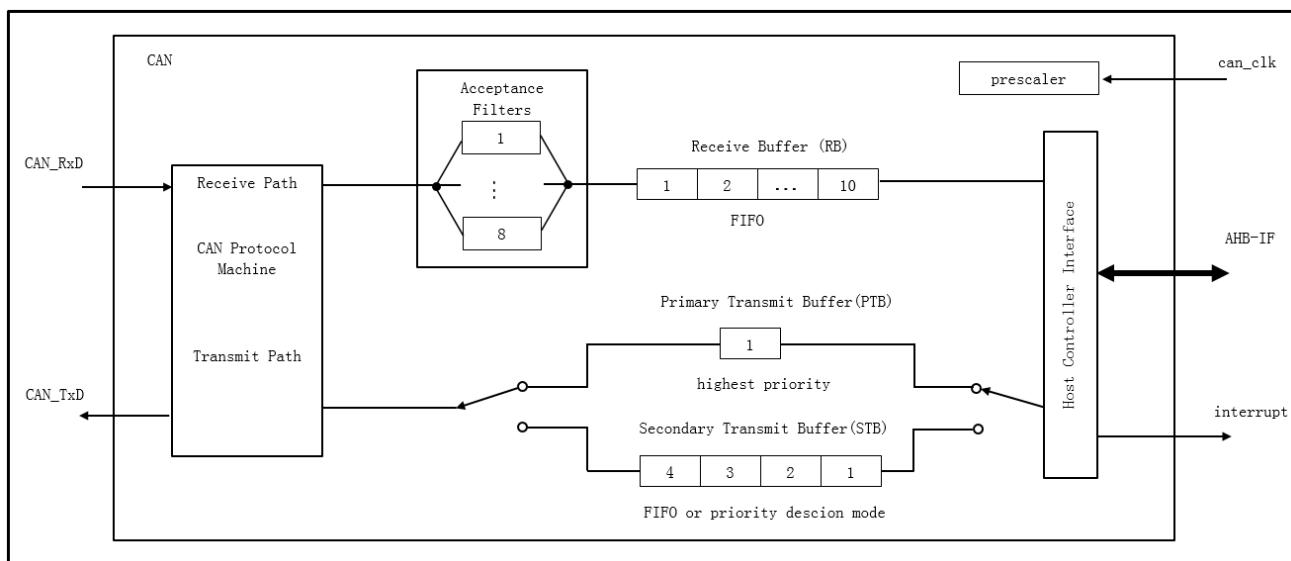


图 30-1 CAN 系统框图

30.3 管脚说明

管脚名	方向	功能描述
CAN_RxD	输入	CAN接收数据信号
CAN_TxD	输出	CAN发送数据信号

表 30-1 CAN 管脚说明

30.4 功能说明

本章将对 CAN 功能详细说明。

30.4.1 波特率设定

CAN 通信使用时钟 can_clk 的时钟源为外部高速振荡器，使用 CAN 模块之前，需要在 CMU 章节设定 CAN 通信时钟。

下图给出 CAN 位时间定义图，虚线上部分为 CAN 协议规定的位时间，虚线下部分为本 CAN 控制器 CAN-CTRL 定义的位时间。其中 segment1 和 segment2 可以通过寄存器 BT 设定。BT 寄存器只能在 CFG_STAT.RESET=1 即 CAN 软件复位时设定。

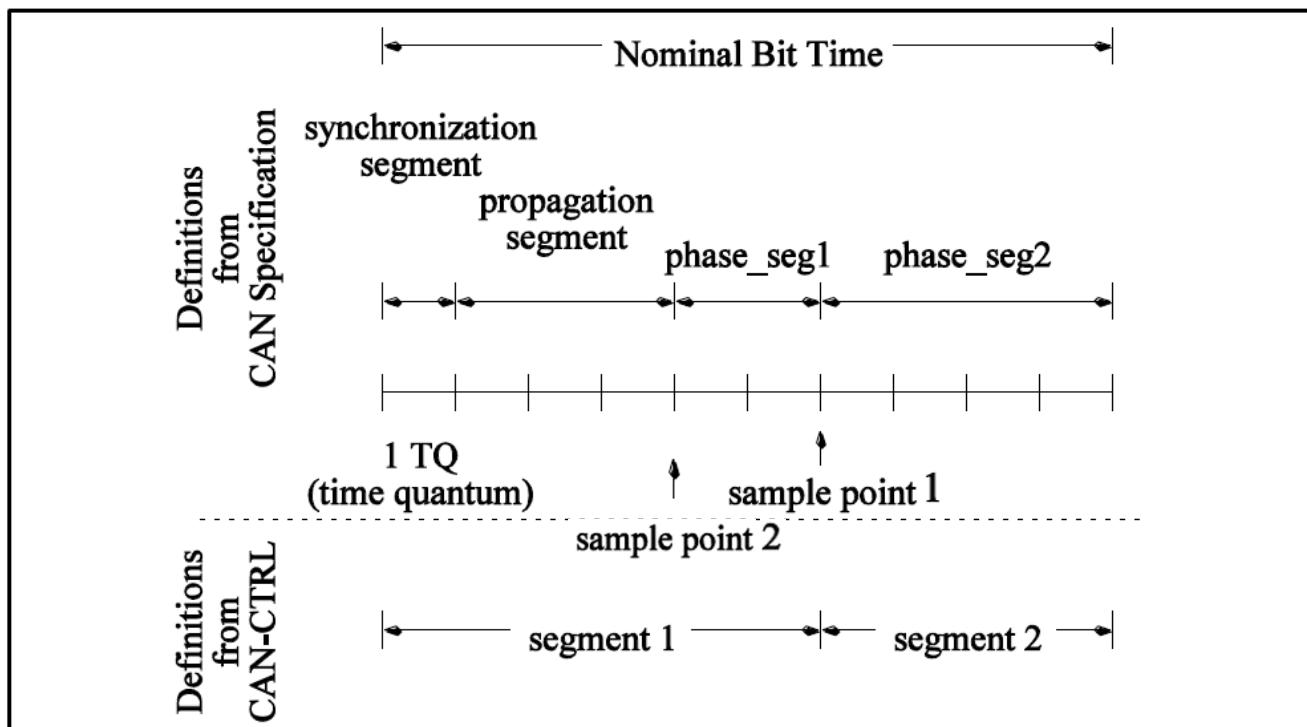


图 30-2 CAN 位时间定义图

TQ 计算方法请参考以下公式，其中 PRESC 通过 BT 寄存器的 PRESC 位设定。 f_{can_clk} 为 CAN 通信时钟频率。

$$TQ = \frac{PRESC+1}{f_{can_clk}}$$

采样点的计算方法请参考下列说明。

- 1) 若 BT 寄存器的 PRESC 位设定 ≥ 1 时，参考图 30-2 的 sample point 1 所示，采样点位于 segment1 和 segment2 的分界点上。

2) 若 BT 寄存器的 PRESC 位设定为 0 时, 参考图 30-2 的 sample point 2 所示, 采样点在 segment1 和 segment2 分界点前面 2 个 TQ 的位置。

建议将 BT 寄存器的 PRESC 位设定值 ≥ 1 。

位时间计算方法请参考以下公式, 其中 SEG_1 和 SEG_2 通过 BT 寄存器的 SEG_1 位和 SEG_2 位设定。

$$BT = t_{SEG1} + t_{SEG2} = ((SEG_1 + 2) + (SEG_2 + 1)) \times TQ$$

位	设定范围	规则
BT寄存器的SEG_1位	0~63	SEG_1 \geq SEG_2 + 1 SEG_2 \geq SJW
BT寄存器的SEG_2位	0~7	
BT寄存器的SJW位	0~7	

表 30-2 CAN 位时间设定规则

30.4.2 发送缓冲器

CAN_CTRL 提供两种发送缓冲器用于发送数据, 主发送数据缓冲器 PTB 和副发送缓冲器 STB。PTB 具有最高的优先级, 但只能缓冲一帧数据。STB 优先级比 PTB 低, 但可以缓冲 4 帧数据, 且 STB 内 4 帧数据可以工作在 FIFO 模式或者优先级仲裁模式。

STB 中的 4 帧数据可以通过 TCMD 寄存器的 TSALL 位设定为 1 全部发送, 在 FIFO 模式下, 最先写入的数据先发送, 在优先级模式下, ID 小的数据先发送。

PTB 中的数据具有最高优先级, 所以 PTB 发送能推迟 STB 发送, 但是已经赢得仲裁并开始发送的 STB 不能够被 PTB 发送推迟。

PTB 和 STB 中的一帧数据需占用 4 个字, 可以通过 TBUF 寄存器进行访问。通过 TCMD 寄存器的 TBSEL 位选择 PTB 或者 STB, TBSEL=0, 选择 PTB, TBSEL=1, 选择 STB。通过 TCTRL 寄存器的 TSNEXT 位选择 STB 中的下一个 SLOT。对应关系如下图所示:

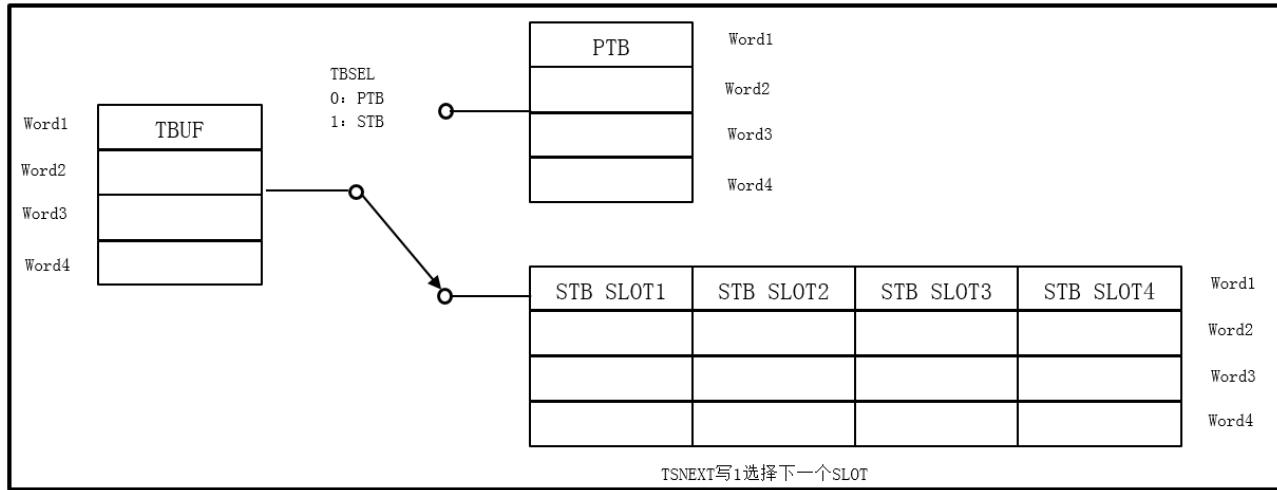


图 30-3 CAN TBUF 寄存器写发送缓冲器和示意图

30.4.3 接收缓冲器

CAN_CTRL 提供 10 个 SLOT 的接收缓冲器用于存储接收到的数据，该 10 个 SLOT 的接收缓冲器工作在 FIFO 模式。每个 RB SLOT 需占用 4 个字，通过 RBUF 寄存器来读取接收到的数据，总是最先读取最早接收到数据，并通过 RCTRL 寄存器的 RREL 设置为 1 释放已经读取的 RB SLOT，并指向下一个 RB SLOT。

通过 RBUF 读取 RB SLOT 示意图如下。



图 30-4 CAN RBUF 寄存器读接收缓冲器示意图

30.4.4 接收筛选寄存器组

CAN_CTRL 提供 8 组 32 位筛选器用于过滤接收到的数据从而降低 CPU 负荷，筛选器可以支持标准格式 11 位 ID 或者扩展格式 29 位 ID。每组筛选器有一个 32 位 ID CODE 寄存器和一个 32 位 ID MASK 寄存器，ID CODE 寄存器用于比较接收到 CAN ID，而 ID MASK 寄存器用于选择比较的 CAN ID 位。对应的 ID MASK 位为 1 时，不比较该位的 ID CODE。

接收到的数据只要通过 8 组筛选器的任意一组，则被接收，接收到的数据存储在 RB

中，否则数据不被接收，也不被存储。

每组筛选器通过 ACFEN 寄存器使能或者禁止。ID CODE 和 ID MASK 通过 ACFCTRL 寄存器的 SELMASK 位设定，SELMASK=0 时，指向 ID CODE，SELMASK=1 时，指向 ID MASK。筛选器通过 ACFCTRL 寄存器的 ACFADR 位选择。ID CODE 和 ID MASK 通过 ACF 寄存器访问且只能在 CFG_STAT.RESET=1 即 CAN 软件复位时设定。ACF 寄存访问筛选寄存器组的方式请参考下图。

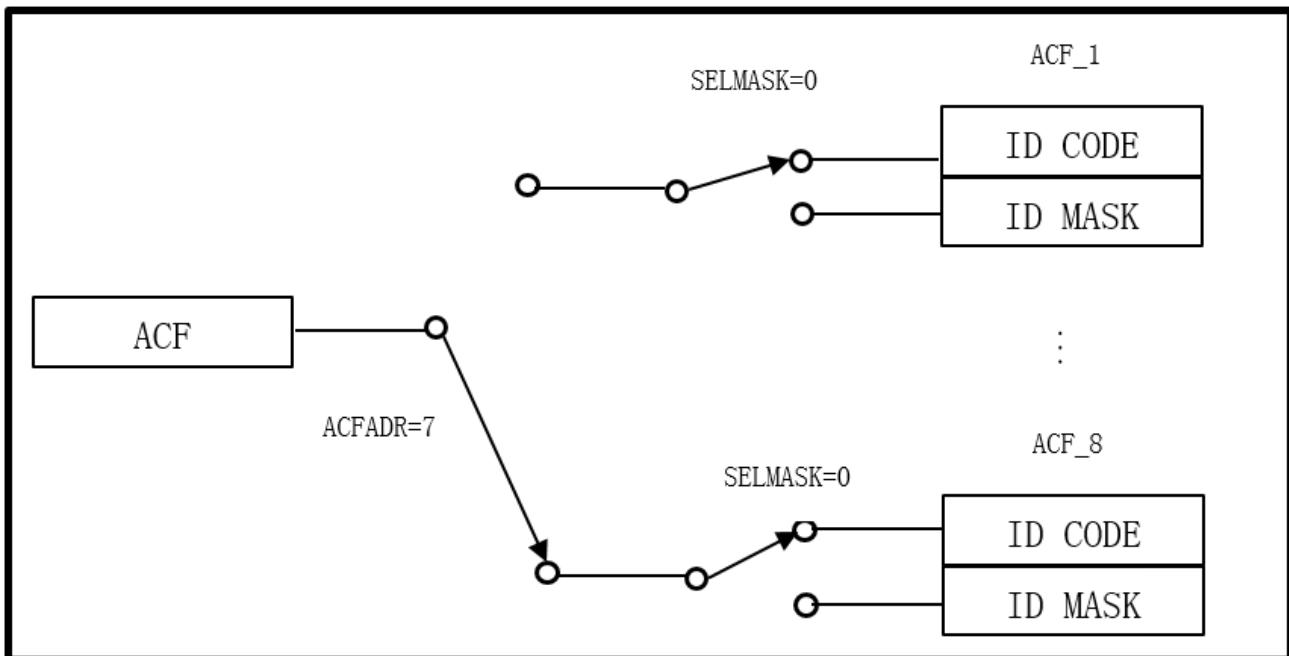


图 30-5 CAN ACF 寄存器访问筛选器组示意图

30.4.5 数据发送

在开始发送前必须保证 PTB 或者 STB 中至少有一帧数据已被装载，PTB 发送过程中 TPE 被锁定，STB 的填充情况可以通过 TSSTAT 位确认。发送数据设定步骤如下：

1. 设定 TBSEL 从 PTB 和 STB 中选择发送 BUF
2. 通过 TBUF 寄存器写需要发送的数据。
3. 如果选择的是 STB，设置 TSNEXT=1 以完成 STB SLOT 的装载。
4. 发送使能
 - PTB 发送使用 TPE
 - STB 发送使用 TSALL 或者 TSONE
5. 发送完成状态确认

- PTB 发送完成使用 TPIF，TPIE 用于使能 TPIF
- STB 采用 TSONE 发送完成时使用 TSIF，TSIE 用于使能 TSIF
- STB 采用 TSALL 发送完成时使用 TSIF，此时需要设定的全部 STB SLOT 数据发送完成后，TSIF 才置位，TSIE 用于使能 TSIF

30.4.6 单次数据发送

不需要使用自动重新发送功能时，可以通过寄存器设定为单次发送模式，CFG_STAT 寄存器的 TPSS 位用于 PTB 的单次发送模式设定，TSSS 位用于 STB 的单次发送模式。数据成功发送时单次发送和正常发送模式时动作相同。但是数据没有成功发送时会出现以下结果：

- TPIF 置位 (TPIE=1)，对应的 BUF SLOT 数据会被清除。
- 有错误发送时，KOER 更新，BEIF 置位 (BEIE=1)。
- 仲裁失败，ALIF 置位 (ALIE=1)。

单次发送模式，不能单独依靠 TPIF 来判断发送完成，需要同 BEIF 和 ALIF 一起判断发送是否完成。

30.4.7 取消数据发送

可以通过 TPA 或者 TSA 取消已请求但还没有被执行的数据发送。取消数据发送会出现以下几种情况：

- 仲裁中
 - 节点仲裁失败，则取消数据发送。
 - 节点仲裁成功，则继续发送。
- 数据发送中
 - 成功发送数据且收到 ACK，对应的标志和状态正常置位。数据发送不取消。
 - 成功发送数据但没有收到 ACK，数据发送取消，错误计数器增加。
 - TSALL=1 设定的发送数据，正在发送的 STB SLOT 数据正常发送，没有开始发送的 STB SLOT 被取消。

取消数据发送的结果有以下两种情况。

- TPA 释放 PTB，且使 TPE=0。

- TSA 释放一个 STB SLOT 或者全部 STB SLOT 取决是 TSONE 还是 TSALL 使能的发送。

30.4.8 数据接收

接收筛选器组可以过滤掉不需要的接收数据，减少中断的发生和 RB 的读取，从而降低 CPU 负荷。接收数据设定步骤如下：

1. 设定筛选器组。
2. 设定 RFIE, RAFIE 和 AFWL。
3. 等待 RFIF 或者 RAIF。
4. 通过 RBUF 从 RB FIFO 中读取最早接收到的数据。
5. 设置 RREL=1，选择下一个 RB SLOT。
6. 重复 4, 5 直到通过 RSTAT 确认 RB 为空。

30.4.9 错误处理

CAN_CTRL 一方面可以自动处理部分错误，比如自动重发数据或者丢弃接收到含有错误的帧，另一方面通过中断将错误向 CPU 报告。

CAN 节点有以下三种错误状态：

- 错误主动：节点检测到错误时自动发送主动错误标志。
- 错误被动：节点检测到错误时自动发送被动错误标志。
- 节点关闭：关闭状态下此节点不再影响整个 CAN 网络。

CAN_CTRL 提供 TECNT 和 RECNT 两个计数器用于计数错误。TECNT 和 RECNT 计数器按照 CAN2.0B 协议规定的规则进行增减。另外提供可编程的 CAN 错误警告 LIMIT 寄存器用于产生错误中断通知 CPU。

CAN 通信过程中有以下 5 种错误类型，错误类型可以通过 EALCAP 寄存器的 KOER 位识别。

- 位错误
- 形式错误
- 填充错误
- 应答错误

- CRC 错误

30.4.10 节点关闭

当发送错误数大于 255 时，CAN 节点自动进入节点关闭状态从而不参与 CAN 通信，直到返回到错误主动状态。可以通过 CFG_STAT 寄存器的 BUSOFF 位确认 CAN 节点关闭状态。BUSOFF 被置位的同时 EIF 中断产生。

CAN 从节点关闭状态恢复到错误主动状态有以下两种方法：

- 上电复位
- 接收到连续 128 个 11 位的隐形位序列（恢复序列）

节点关闭状态下，TECNT 值保持不变，RECNT 用于计数恢复序列。从节点关闭状态恢复后，TECNT 和 RECNT 被复位为 0。

节点关闭标志 BUSOFF 置位的同时，CFG_STAT 寄存器的 RESET 位也被置位。

30.4.11 仲裁失败位置捕捉

CAN_CTRL 能够精确捕捉到仲裁失败位的位置并反映到 ALC 寄存器中。ALC 寄存器中保存着最近一次仲裁失败位的位置，如果节点赢得仲裁，则 ALC 位不更新。

ALC 值定义如下：

SOF 位后，第一个 ID 数据位 ALC 为 0，第二个 ID 数据位 ALC 为 1，依次类推。因为仲裁只发生在仲裁场内，所以 ALC 的最大值为 31。比如一个标准格式远程帧和一个扩展帧仲裁，扩展帧在 IDE 位失败，则 ALC=12。

30.4.12 回环模式

CAN_CTRL 支持以下两种回环模式：

- 内部回环
- 外部回环

两种回环模式都可以接收自己发出的数据帧，主要用于测试用途。

内部回环模式，模块内部将接收数据线连接到发送数据线，并且发送数据不输出。内部回环模式下，节点会生成自应答信号以避免 ACK 错误。

外部回环模式保持和收发器的连接因此发送的数据仍能出现在 CAN 总线上，在收发

器的帮助下，CAN 能收到自己发送的数据。外部回环模式可以通过 RCTRL 寄存器的 SACK 位来决定是否生成自应答信号，SACK=0 时，不生成自应答信号，SACK=1 时，生成自应答信号。

外部回环模式，SACK=0 时，会出现以下两种情况：

- 其它节点也收到本节点发送的数据帧并发送应答信号，该情况下本节点能够成功收发数据。
- 如果没有其它节点返回应答信号，则会产生应答错误，会重新发送数据并增加错误计数器。此时推荐采用单次发送模式。

从回环模式返回到正常模式时，除了清除模式位以外，还需要软件复位 CAN_CTRL。

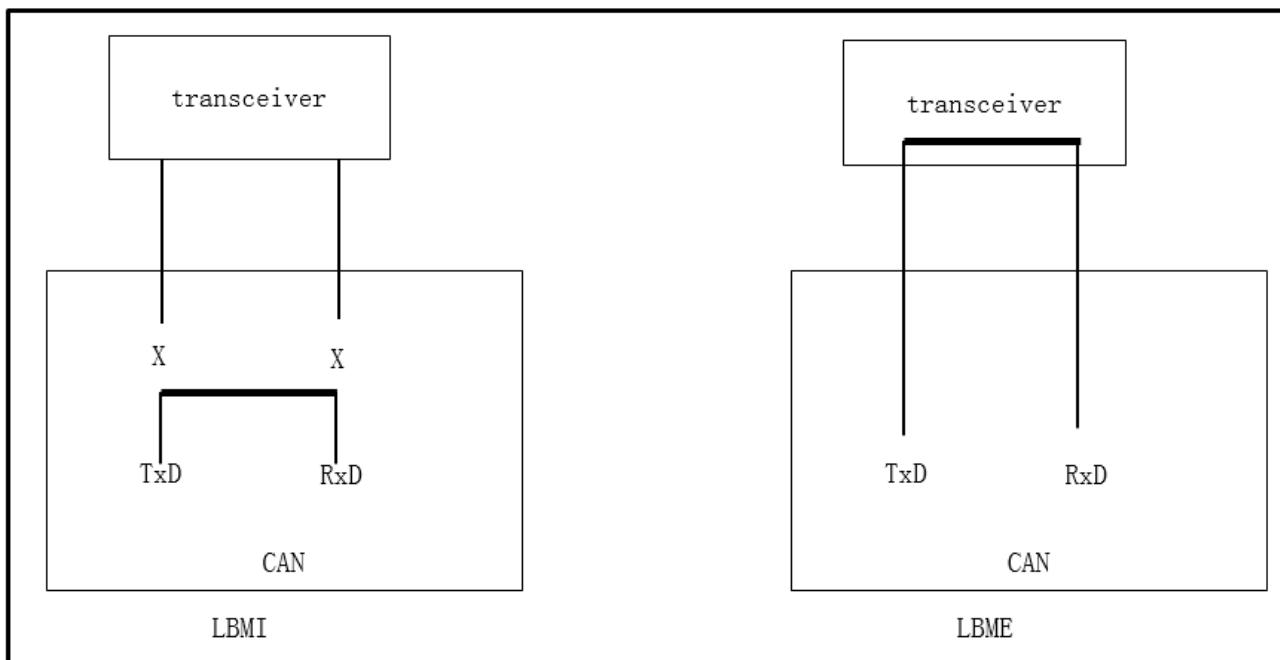


图 30-6 CAN LBMI 和 LBME 示意图

30.4.13 静默模式

静默模式可以用来监控 CAN 网络数据。在静默模式下，可以从 CAN 总线接收数据，不向总线发送任何数据。将 TCMD 寄存器中的 LOM 置 1，使 CAN 总线控制器进入静默模式，将其清 0 可以离开静默模式。

外部回环模式可以和静默模式组合成外部回环静默模式，此时 CAN 可以认为一个安静的接收者，但在有必要的时候可以发送数据。外部回环静默模式下，帧包含自应答信号允许被发送，但是该节点不会产生错误标志和过载帧。

30.4.14 软件复位功能

通过设定寄存器 CFG_STAT 寄存器的 RESET 位为 1，实现软件复位功能的复位范围如下表所示。

寄存器位名	软件复位	备注	寄存器位名	软件复位	备注
ACFADR	否	-	EWL	是	-
ACODE	否	只能在软件复位时可以写	KOER	是	-
AE_x	否	-	LBME	是	-
AFWL	否	-	LBMI	是	-
AIF	是	-	RACTIV	是	接收立即停止，并不生成ACK
ALC	是	-	RAFIE	否	-
ALIE	否	-	RAFIF	是	-
ALIF	是	-	RBALL	是	-
AMASK	否	只能在软件复位时可以写	RBUF	是	所有RB被标记为空，数值不定
BEIE	否	-	REF_ID	否	-
BEIF	是	-	REF_IDE	否	-
BUSOFF	否	通过写1清除	RFIE	否	-
EIE	否	-	RFIF	是	-
EIF	否	-	RIE	否	-
EPASS	否	-	RIF	是	-
EPIE	否	-	ROIE	否	-
EPIF	是	-	ROIF	是	-
EWARN	否	-	ROM	否	-

寄存器位名	软件复位	备注	寄存器位名	软件复位	备注
ROV	是	-	TSNEXT	是	-
RREL	是	-	TSONE	是	-
PRESC	否	只能在软件复位时可以写	TPIE	否	-
RSTAT	是	-	TPIF	是	-
SACK	是	-	TPSS	是	-
SELMASK	否	-	TSFF	是	所有STB SLOT被标记为空
SEG_1	否	只能在软件复位时可以写	TSIE	否	-
SEG_2	否	只能在软件复位时可以写	TSIF	是	-
SJW	否	只能在软件复位时可以写	TSSS	是	-
TACTIVE	是	发送立即停止	TSSTAT	是	所有STB SLOT被标记为空
TBE	是	-	TTEN	是	-
TBF	否	-	TTIF	是	-
TBPTR	否	-	TTIE	否	-
TBSEL	是	-	TTPTR	否	-
TBUF	是	所有的STB被标记为空，指向PTB	TTTBM	否	-
TECNT	否	通过BUSOFF=1清除	TTYPE	否	-
TEIF	是	-	TT_TRIG	否	-
TPA	是	-	TT_WTRIG	否	-
TPE	是	-	T_PRESC	否	-
TSA	是	-	WTIE	否	-
TSALL	是	-	WTIF	是	
TSMODE	否	-			

表 30-3 软件复位范围表

30.4.15 向上兼容 CAN-FD 功能

CAN-CTRL 即使在包含 CAN-FD 网络中收到 CAN-FD 的帧，接收器会自动忽略这些帧，不返回 ACK，等到总线空闲时，再发送或者接收下一个 CAN2.0B 的帧。

30.4.16 时间触发 TTCAN

CAN-CTRL 为 ISO11898-4 规定的时间触发通信方式提供部分(lever 1)硬件支持。本章节从以下 5 个部分介绍 TTCAN 功能。

30.4.16.1 TTCAN 模式下的 TBUF 行为

TTTBM=1

TTTBM=1 时，PTB 和 STB SLOT 一样组成 TB SLOT，通过 TB PTR 寄存器指定发送 BUF，其中 TB PTR=0 时，指向 PTB，TB PTR=1 是指向 STB SLOT1，依次类推。主机可以通过 TPE 和 TPF 寄存器来标记发送 BUF SLOT。此时 TB SEL 和 TS NEXT 寄存器无任何意义从而可以被忽略。

TTTBM=1 时，PTB 不具有任何特殊的属性，和 STB SLOT 一样，传送完成标志也采用 TSIF。

TTCAN 模式时，发送 BUF 没有 FIFO 模式和优先级仲裁模式，同时也只有一个选定的 SLOT 可以发送数据。

TTCAN 模式下，传输开始需要采用时间触发方式，TPE，TS ONE，TS ALL，TPSS 和 TPA 被固定为 0 且被忽略。

TTTBM=0

TTTBM=0 时，组合使用事件驱动通信和接收时间戳功能。在该模式下，PTB 和 STB 的功能和 TTEN=0 时一致，因此 PTB 始终具有最高的优先级，而 STB 可以工作在 FIFO 模式或者仲裁模式。

30.4.16.2 TTCAN 功能描述

上电后，Time Master 需要根据 ISO 11898-4 协议进行初始化。一个 CAN 网络中，最多可以有 8 个潜在的 Time Master。每一个 Time Master 都具有自己的参考消息 ID (ID 最后 3 位)。这些潜在的 Time Master 根据自己的优先级发送各自的参考消息。

TTEN=1 后，16 位的计数器开始工作，当参考消息被成功接收或者 Time Master 成功发送参考消息时，CAN 控制器将 Sync_Mark 拷贝给 Ref_Mark，Ref_Mark 将 cycle time 设置为 0。成功接收参考消息置位 RIF 标志而成功发送参考消息置位 TPIF 标志或者 TSIF 标志。此时主机需要准备下一个动作的触发条件。

触发条件可以是接收触发。该触发仅触发中断可用于检测期待的消息没有被收到。

触发条件也可以是发送触发。该触发开始发送通过 TTPTR 寄存器指定的 TBUF SLOT 里的数据。如果选定的 TBUF SLOT 被标记为空，则不开始发送，但置位中断标志。

30.4.16.3 TTCAN 时序

CAN_CTRL 支持 ISO11898-4 level 1。包含的一个 16 位计数器工作在为 PRESC，SEG_1, SET_2 定义的位时间下。如果 TTEN=1，则有一个额外的预分频器 T_PRESC。一帧数据的 SOF 时，计数器的值为 Sync_Mark。如果该帧数据为参考消息，则将 Sync_Mark 拷贝给 Ref_Mark。cycle time 等于计数器的值减去 Ref_Mark。该时间用作为接收消息的时间戳或者发送消息的触发时间基准。

30.4.16.4 TTCAN 触发方式

通过 TTYPE 寄存器定义 TTCAN 的触发方式，TTPTR 寄存器指定发送 SLOT，而 TT_TRIG 指定触发器的 cycle time。

包含以下五种触发方式：

- 立即触发
- 时间触发
- 单次发送触发
- 发送开始触发
- 发送停止触发

除了立即触发方式外，所有的触发器都使用 TTIF 标志。TTBIM=1 时，只支持时间触发方式。

立即触发

通过写 TT_TRIG 的高位（不在意写入的值），启动触发器。此模式下，TTPTR 选定的 TBUF SLOT 内的数据会立即发送。TTIF 不置位。

时间触发

时间触发方式仅通过置位 TTIF 标志产生中断，并无其他功能。如果一个节点期待在特定的时间窗口内收到期待的数据，则可以使用时间触发方式。如果 TT_TRIG 值小于实际的 cycle time，则 TEIF 置位且无其它动作。

单次发送触发

单次发送触发方式用于在执行时间窗口内发送数据。此时，忽略 TSSS 位。

通过 TEW 位设定 ISO11898-4 规定的最多 16 个 cycle time 的 Tick，设定范围为 1~16。

如果在规定的发送使能时间窗口内数据没有开始发送，则帧被丢弃。对应的发送 BUF SLOT 被标记为空，并且置位 AIF，对应的发送 BUF 内的数据不会被改写，因为可以通过置位 TPF 再次发送。

如果 TT_TRIG 值小于实际的 cycle time，则 TEIF 置位且无其它动作。

发送开始触发

发送开始触发方式用于仲裁时间窗口内，参与仲裁。TSSS 用于决定是否自动重发或者单次发送模式。如果指定的消息没有被成功发送，可以使用发送停止触发来停止该发送。

如果 TT_TRIG 值小于实际的 cycle time，则 TEIF 置位且无其它动作。

发送停止触发

发送停止触发方式用于停止通过发送开始触发方式已经开始的发送。如果发送被停止，则发送帧被舍弃，置位 AIF 并将选定的 TBUF SLOT 标记为空，但 TBUF SLOT 内的数据不会被改写，可以通过置位 TPF 就可以再次发送。

如果 TT_TRIG 值小于实际的 cycle time 则 TEIF 置位且执行停止。

30.4.16.5 TTCAN 触发看门时间

TTCAN 触发看门时间功能类似于看门狗功能，在 TTTBM=1 时使用。用来看门从上次成功接收到参考消息开始的时间。参考消息可以在周期 cycle time 中或者一个事件后被接收，应用程序应该根据具体情况设定合适的看门时间。

如果 cycle count 等于 TT_WTRIG，则置位 WTIF。通过 WTIE 写 0，关闭看门触发。

如果 TT_WTRIG 比实际的 cycle time 小，则 TEIF 置位。

30.4.17 中断

中断标志	描述
RIF	接收中断
ROIF	接收上溢中断
ROIF	接收BUF满中断
RAFIF	接收BUF将满中断
TPIF	PTB发送中断
TSIF	STB发送中断
EIF	错误中断
AIF	取消发送中断
EPIE	错误被动中断
ALIF	仲裁失败中断
BEIF	总线错误中断
WTIF	触发看门中断
TEIF	触发错误中断
TTIF	时间触发中断

表 30-4 CAN 中断表

30.5 寄存器说明

CAN_BASE_ADDR:0x40070400

寄存器名	符号	偏移地址	位宽	复位值
CAN接收BUF寄存器	CAN_RBUF	0x00~0x0F	128	0xFFFF XXXX
CAN发送BUF寄存器	CAN_TBUF	0x50~0x5F	128	0xFFFF XXXX
CAN配置和状态寄存器	CAN_CFG_STAT	0xA0	8	0x80
CAN命令寄存器	CAN_TCMD	0xA1	8	0x00
CAN发送控制寄存器	CAN_TCTRL	0xA2	8	0x90
CAN接收控制寄存器	CAN_RCTRL	0xA3	8	0x00
CAN接收和发送中断使能寄存器	CAN_RTIE	0xA4	8	0xFE
CAN接收和发送中断标志寄存器	CAN_RTIF	0xA5	8	0x00
CAN错误中断使能和标志寄存器	CAN_ERRINT	0xA6	8	0x00
CAN警告限定寄存器	CAN_LIMIT	0xA7	8	0x1B
CAN位时序寄存器	CAN_BT	0xA8	32	0x0102 0203
CAN错误和仲裁失败捕捉寄存器	CAN_EALCAP	0xB0	8	0x00
CAN接收错误计数器寄存器	CAN_RECNT	0xB2	8	0x00
CAN发送错误计数器寄存器	CAN_TE_CNT	0xB3	8	0x00
CAN筛选器组控制寄存器	CAN_ACFCTRL	0xB4	8	0x00
CAN筛选器组使能寄存器	CAN_ACFEN	0xB6	8	0x01
CAN筛选器组code和mask寄存器	CAN_ACF	0xB8	32	0xFFFF XXXX
TTCAN TB slot指针寄存器	CAN_TBSLOT	0xBE	8	0x00
TTCAN 时间触发配置寄存器	CAN_TTCFG	0xBF	8	0x90
TTCAN 参考消息寄存器	CAN_REF_MSG	0xC0	32	0xFFFF XXXX
TTCAN 触发配置寄存器	CAN_TRG_CFG	0xC4	16	0x0000
TTCAN 触发时间寄存器	CAN_TT_TRIG	0xC6	16	0x0000
TTCAN 触发看门时间寄存器	CAN_TT_WTRIG	0xC8	16	0x0000

表 30-5 CAN 寄存器一览表

地址	BYTE访问	HALFWORD访问		WORD访问					
0x00~0x0F	CAN_RBUF	CAN_RBUF		CAN_RBUF					
0x50~0x5F	CAN_TBUF	CAN_TBUF		CAN_TBUF					
0xA0	CAN_CFG_STAT	CAN_TCMD	CAN_CFG_STAT	CAN_RCTRL	CAN_TCTRL	CAN_TCMD	CAN_CFG_STAT		
0xA1	CAN_TCMD	-		-					
0xA2	CAN_TCTRL	CAN_RCTRL	CAN_TCTRL	-					
0xA3	CAN_RCTRL	-		-					
0xA4	CAN_RTIE	CAN_RTIF	CAN_RTIE	CAN_LIMIT	CAN_ERRINT	CAN_RTIF	CAN_RTIE		
0xA5	CAN_RTIF	-		-					
0xA6	CAN_ERRINT	CAN_LIMIT	CAN_ERRINT	-					
0xA7	CAN_LIMIT	-		-					
0xA8	CAN_BT[7:0]	CAN_BT[15:0]		CAN_BT					
0xA9	CAN_BT[15:8]	-		-					
0xAA	CAN_BT[23:16]	-		-					
0xAB	CAN_BT[31:24]	CAN_BT[31:16]		-					
0xB0	-	-		CAN_TECNT	CAN_RECNT	-			
0xB1	-	-		-					
0xB2	CAN_RECNT	CAN_TECNT	CAN_RECNT	-					
0xB3	CAN_TECNT	-		-					
0xB4	CAN_ACFCTRL[7:0]	CAN_ACFCTRL		CAN_ACFEN		CAN_ACFCTRL			
0xB5	CAN_ACFCTRL[15:8]	-		-					
0xB6	CAN_ACFEN[7:0]	CAN_ACFEN		-					
0xB7	CAN_ACFEN[15:8]	-		-					
0xB8	CAN_ACF	CAN_ACF		CAN_ACF					
0xBC	-	-		CAN_TTCFG	CAN_TBSLOT	-			
0xBD	-	-		-					
0xBE	CAN_TBSLOT	CAN_TTCFG	CAN_TBSLOT	-					
0xBF	CAN_TTCFG	-		-					
0xC0	CAN_REF_MSG[7:0]	CAN_REF_MSG[15:0]		CAN_REF_MSG					
0xC1	CAN_REF_MSG[15:8]	-		-					
0xC2	CAN_REF_MSG[23:16]	CAN_REF_MSG[31:16]		-					
0xC3	CAN_REF_MSG[31:24]	-		-					
0xC4	CAN_TRG_CFG[7:0]	CAN_TRG_CFG		CAN_TT_TRIG		CAN_TT_CFG			
0xC5	CAN_TRG_CFG[15:8]	-		-					
0xC6	CAN_TT_TRIG[7:0]	CAN_TT_TRIG		-					
0xC7	CAN_TT_TRIG[15:8]	-		-					
0xC8	CAN_TT_WTRIG[7:0]	CAN_TT_WTRIG		-		CAN_TT_WTRIG			
0xC9	CAN_TT_WTRIG[15:8]	-		-					

表 30-6 CAN 寄存器 BYTE/HALFWOD/WORD 访问排布表

30.5.1 CAN 接收 BUF 寄存器 (CAN_RBUF)

CAN Receive Buffer Registers

偏移地址: 0x50

复位值: 0xFFFF XXXX

RBUF 寄存器指向最早接收到的 CAN 邮箱的 RB SLOT 地址, RBUF 寄存器可以按照任意顺序读取。

KOER 位即为寄存器 EALCAP.KOER, 仅在 RBALL=1 时有意义。

TX 位表示在回环模式下接受到自己发送的邮箱。

CYCLE_TIME 位仅在 TTCAN 模式时有效, 表示 SOF 开始时的 cycle time。

CAN 接收邮箱的数据格式如下:

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
RBUF	ID[7:0]								ID
RBUF+1	-				ID[10:8]				ID
RBUF+2	-								ID
RBUF+3	-								ID
RBUF+4	IDE=0	RTR	0	0	DLC[3:0]				Control
RBUF+5	KOER[2:0]			TX	-				Status
RBUF+6	CYCLE_TIME[7:0]								TTCAN
RBUF+7	CYCLE_TIME[15:8]								TTCAN
RBUF+8	DATA1								Data
RBUF+9	DATA2								Data
RBUF+10	DATA3								Data
RBUF+11	DATA4								Data
RBUF+12	DATA5								Data
RBUF+13	DATA6								Data
RBUF+14	DATA7								Data
RBUF+15	DATA8								Data

表 30-7 标准格式 CAN 接收邮箱格式

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
RBUF	ID[7:0]								ID
RBUF+1	ID[15:8]								ID
RBUF+2	ID[23:16]								ID
RBUF+3	-			ID[28:24]					ID
RBUF+4	IDE=1	RTR	0	0	DLC[3:0]				Control
RBUF+5	KOER[2:0]			TX	-				Status
RBUF+6	CYCLE_TIME[7:0]								TTCAN
RBUF+7	CYCLE_TIME[15:8]								TTCAN
RBUF+8	DATA1								Data
RBUF+9	DATA2								Data
RBUF+10	DATA3								Data
RBUF+11	DATA4								Data
RBUF+12	DATA5								Data
RBUF+13	DATA6								Data
RBUF+14	DATA7								Data
RBUF+15	DATA8								Data

表 30-8 扩展格式 CAN 接收邮箱格式

控制位含义如下：

IDE(IDentifier Extension):

0：标准格式

1：扩展格式

RTR(Remote Transmission Request)

0：数据帧

1：远程帧

DLC(Data Length Code):

数据长度码，设定范围为 0~8，对应数据长度为 0Byte~8Byte

30.5.2 CAN 发送 BUF 寄存器 (CAN_TBUF)

CAN Transmit Buffer Registers

偏移地址: 0x00

复位值: 0xFFFF FFFF

TBUF 寄存器指向下一个空的 CAN 发送 BUF SLOT，TBUF 寄存器可以按照任意顺序读取。通过软件将 TSNEXT 写 1 来标记对应的 TBUF SLOT 已经写入数据，从而指向下一个 TBUF SLOT。

TBUF 只能 WORD 访问。

CAN 发送邮箱的数据格式如下:

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
TBUF	ID[7:0]								ID
TBUF+1	-				ID[10:8]				ID
TBUF+2	-								ID
TBUF+3	-								ID
TBUF+4	IDE=0	RTR	0	0	DLC[3:0]				Control
TBUF+5	-								-
TBUF+6	-								-
TBUF+7	-								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
TBUF+11	DATA4								Data
TBUF+12	DATA5								Data
TBUF+13	DATA6								Data
TBUF+14	DATA7								Data
TBUF+15	DATA8								Data

表 30-9 标准格式 CAN 发送邮箱格式

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
TBUF	ID[7:0]								ID
TBUF+1	ID[15:8]								ID
TBUF+2	ID[23:16]								ID
TBUF+3	-			ID[28:24]					ID
TBUF+4	IDE=0	RTR	0	0	DLC[3:0]				Control
TBUF+5	-								-
TBUF+6	-								-
TBUF+7	-								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
TBUF+11	DATA4								Data
TBUF+12	DATA5								Data
TBUF+13	DATA6								Data
TBUF+14	DATA7								Data
TBUF+15	DATA8								Data

表 30-10 扩展格式 CAN 发送邮箱格式

控制位含义如下：

IDE(IDentifier Extension):

0：标准格式

1：扩展格式

RTR(Remote Transmission Request)

0：数据帧

1：远程帧

DLC(Data Length Code):

数据长度码，设定范围为 0~8，对应数据长度为 0Byte~8Byte

30.5.3 CAN 配置和状态寄存器 (CAN_CFG_STAT)

CAN Configuration and Status Register

偏移地址: 0xA0

复位值: 0x80

b7	b6	b5	b4	b3	b2	b1	b0
RESET	LBME	LBMI	TPSS	TSSS	RACTIVE	TACTIVE	BUSOFF

位	标记	位名	功能	读写
b7	RESET	复位请求	复位请求位 0: 不请求局部复位 1: 请求局部复位 部分寄存器只能在RESET=1时进行写操作，具体请参考软件复位功能，当该节点进入BUS OFF状态时，硬件自动将RESET位置1。请注意，当RESET=0后需要11个CAN bit time该节点才能参与通信。	R/W
b6	LBME	外部回环模式使能位	外部回环模式使能位 0: 禁止外部回环模式 1: 使能外部回环模式 注意：通信中禁止设定该位。	R/W
b5	LBMI	内部回环模式使能位	内部回环模式使能位 0: 禁止内部回环模式 1: 使能内部回环模式 注意：通信中禁止设定该位。	R/W
b4	TPSS	PTB单次传输模式	PTB单次传输模式 0: 禁止PTB单次传输模式 1: 使能PTB单次传输模式	R/W
b3	TSSS	STB单次传输模式	STB单次传输模式 0: 禁止STB单次传输模式 1: 使能STB单次传输模式	R/W
b2	RACTIVE	接收中状态信号	接收中状态信号 0: 非接收中 1: 接收中	R
b1	TACTIVE	发送中状态信号	发送中状态信号 0: 非发送中 1: 发送中	R
b0	BUSOFF	总线关闭状态	总线关闭状态 0: 总线有效状态 1: 总线关闭状态 注意：写1可以清零TECNT和RECNT寄存器，但仅限用于调试用途。	R/W

30.5.4 CAN 命令寄存器 (CAN_TCMD)

CAN Command Register

偏移地址: 0xA1

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
TBSEL	LOM	-	TPE	TPA	TSONE	TSALL	TSA

位	标记	位名	功能	读写
b7	TBSEL	发送BUF选择位	发送BUF选择位 (Transmit Buffer Select) 0: PTB 1: STB 当TTEN=1&TTTBM=1时，TBSEL被复位成复位值。 注意：写TBUF寄存器或者TSNEXT位时，此位需要保持定值。	R/W
b6	LOM	静默模式使能位	静默模式使能位 (Listen Only Mode) 0: 禁止静默模式 1: 使能静默模式 LOM=1&LBME=0时禁止发送。 LOM=1&LBME=1时禁止应答相应接收到的帧以及错误帧，但可以发送数据。 注意：通信中禁止设定该位。	R/W
b5	Reserved	-	必须保持复位值。	R
b4	TPE	PTB发送使能位	PTB发送使能位 (Transmit Primary Enable) 0: 禁止PTB发送 1: 使能PTB发送 此位使能后，PTB中的Mailbox将在下一个可以发送的位置被发送。已经开始的STB发送将继续，但是下一个等待的STB发送会被延迟到PTB发送完成后再进行。 该位写1后将保持为1直到PTB发送完成或者通过TPA取消发送。软件不能通过写0清除该位。 以下情况TPE被硬件复位成复位值： - RESET=1 - BUSOFF=1 - LOM=1&LBME=0 - TTEN=1&TTTBM=1	R/W

位	标记	位名	功能	读写
b3	TPA	PTB发送取消位	PTB发送取消位 (Transmit Primary Abort) 0: 不取消 1: 取消已经通过TPE置1请求但还未开始的PTB发送 该位软件写1但是通过硬件清零。通过写1可以清零TPE位，因此和TPE同时写1。 以下情况TPE被硬件复位成复位值： - RESET=1 - BUSOFF=1 - TTEN=1&TTTBMM=1	R/W
b2	TSONE	发送一帧STB数据	发送一帧STB数据设定位 (Transmit Secondary ONE frame) 设定位 0: 不发送 1: 发送一帧STB数据 FIFO模式中，发送最早写入的数据，优先级模式里发送最高优先级的数据 该位写1后将保持为1直到STB发送完成或者通过TSA取消发送。软件不能通过写0清除该位。 以下情况TSONE被硬件复位成复位值： - RESET=1 - BUFOFF=1 - LOM=1&LBME=0 - TTEN=1&TTTBMM=1	R/W
b1	TSALL	发送所有的STB数	发送所有的STB数据设定位 (Transmit Secondary ALL frame) 据设定位 0: 不发送 1: 发送STB中所有的数据 该位写1后将保持为1直到STB发送完成或者通过TSA取消发送。软件不能通过写0清除该位。 以下情况TSALL被硬件复位成复位值： - RESET=1 - BUFOFF=1 - LOM=1&LBME=0 - TTEN=1&TTTBMM=1	R/W
b0	TSA	STB发送取消位	STB发送取消位 (Transmit Secondary Abort) 0: 不取消 1: 取消已经通过TSONE或者TSALL置1请求但还未开始的STB发送 该位通过软件写1但是通过硬件清零。写1可以清零TSONE或者TSALL位。 以下情况TSA被硬件复位成复位值： - RESET=1 - BUSOFF=1	R/W

30.5.5 CAN 发送控制寄存器 (CAN_TCTRL)

CAN Transmit Control Register

偏移地址: 0xA2

复位值: 0x90

b7	b6	b5	b4	b3	b2	b1	b0
-	TSNEXT	TSMODE	TTTBM	-	-	-	TSSTAT[1:0]

位	标记	位名	功能	读写
b7	Reserved	-	必须保持复位值。	R
b6	TSNEXT	下一个STB	下一个STB (Transmit buffer Secondary NEXT)	R/W
	SLOT	0: 无动作 1: 当前STB SLOT已填充, 指向下一个SLOT	应用程序将TBUF中的数据写完后, 通过置位TSNEXT位标识当前STB SLOT已经被填充, 从而硬件将TBUF指向下一个STB SLOT。 被TSNEXT位标识的STB SLOT中的数据可以通过TSONE或者TSALL位发送。 该位通过应用程序写1, 硬件清零。 所有的STB SLOT被填满后, TSNEXT保持为1直到有STB SLOT被释放。 注意: TTCAN模式时此位固定为0。	
b5	TSMODE	STB发送模式	STB发送模式 (Transmit buffer Secondary operation MODE) 0: FIFO模式 1: 优先级模式 FIFO模式根据数据帧写入的先后顺序发送。 优先级模式根据ID自动判断, ID越小, 优先级越高。 无论何种模式, PTB具有最高的优先级。 注意: TSMODE位只能在STB空时设定。	R/W
b4	TTTBM	TTCAN BUF模式	TTCAN BUF模式 (TTCAN Transmit Buffer Mode) TTEN=0时, TTTBM被忽略。 0: TSMODE决定, PTB和STB 1: 通过TBPTR和TTPTR设定 TTCAN模式时, 只需要接收时间戳功能时, 此位可以设置为0, 通过TSMODE决定使用PTB还是STB。 注意: TSMODE位只能在STB空时设定。	R/W
b3~b2	Reserved	-	必须保持复位值。	R

位	标记	位名	功能	读写
b1~b0	TSSTAT	STB状态	STB状态 (Transmission Secondary STATus bits) TTEN=0 或 TTTBM=0 00: STB空 01: STB小于等于半满 10: STB大于半满 11: STB满 TTEN=1 且 TTTBM=1 00: PTB和STB空 01: PTB和STB非满 10: 保留 11: PTB和STB满	R

30.5.6 CAN 接收控制寄存器 (CAN_RCTRL)

CAN Receive Control Register

偏移地址: 0xA3

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
SACK	ROM	ROV	RREL	RBALL	-		RSTAT[1:0]

位	标记	位名	功能	读写
b7	SACK	自应答	自应答 (Self-ACKnowledge) 0: 无自应答 1: LBME=1时, 使能自应答功能	R/W
b6	ROM	接收BUF上溢模式设定位	接收BUF上溢模式设定位 (Receive buffer Overflow Mode) 0: 最早接收到的数据被覆盖 1: 新接收到的数据不被存储	R/W
b5	ROV	接收BUF上溢标志位	接收BUF上溢标志位 (Receive buffer OVerflow) 0: 无上溢 1: 上溢, 最少有一个数据丢失 通过写RREL为1清零。	R
b4	RREL	释放接收BUF	释放接收BUF (Receive buffer RElease) 0: 不释放 1: 表示该接收BUF已经被读取过, RBUF寄存器指向下一个RB SLOT。	R/W
b3	RBALL	接收BUF数据存储所有数据帧	接收BUF数据存储所有数据帧 (Receive Buffer stores ALL data frames) 0: 正常模式 1: 存储所有的数据包括有错误的数据。	R/W
b2	Reserved	-	必须保持复位值。	R
b1~b0	RSTAT	接收BUF状态	接收BUF状态 (Receive buffer STATus) 00: RBUF空 01: RBUF非空但小于AFWL编程值 10: RBUF大于等于AFWL编程值但未满 11: 满 (上溢时保持此值)	R

30.5.7 CAN 接收和发送中断使能寄存器 (CAN_RTIE)

CAN Receive and Transmit Interrupt Enable Register

偏移地址: 0xA4

复位值: 0xFE

b7	b6	b5	b4	b3	b2	b1	b0
RIE	ROIE	RFIE	RAFIE	TPIE	TSIE	EIE	TSFF

位	标记	位名	功能	读写
b7	RIE	接收中断使能	接收中断使能 (Receive Interrupt Enable) 0: 禁止 1: 使能	R/W
b6	ROIE	接收上溢中断使能	接收上溢中断使能 (Receive Overrun Interrupt Enable) 0: 禁止 1: 使能	R/W
b5	RFIE	接收BUF满中断使能	接收BUF满中断使能 (RB Full Interrupt Enable) 0: 禁止 1: 使能	R/W
b4	RAFIE	接收BUF将满中断使能	接收BUF将满中断使能 (RB Almost Full Interrupt Enable) 0: 禁止 1: 使能	R/W
b3	TPIE	PTB发送中断使能	PTB发送中断使能 (Transmission Primary Interrupt Enable) 0: 禁止 1: 使能	R/W
b2	TSIE	STB发送中断使能	STB发送中断使能 (Transmission Secondary Interrupt Enable) 0: 禁止 1: 使能	R/W
b1	EIE	错误中断使能	错误中断使能 (Error Interrupt Enable) 0: 禁止 1: 使能	R/W
b0	TSFF	发送BUF满标志	TTEN=0 or TTTBM=0: STB满标志 (Transmit Secondary buffer Full Flag) 0: STB SLOT没有被全部填充 1: STB SLOT被全部填充 TTEN=1 and TTTBM=1: TB满标志 (Transmit buffer Full Flag) 0: TBPTR选择的发送BUF没有被全部填充 1: TBPTR选择的发送BUF被全部填充	R

30.5.8 CAN 接收和发送中断状态寄存器 (CAN_RTIF)

CAN Receive and Transmit Interrupt Status Register

偏移地址: 0xA5

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
RIF	ROIF	RFIF	RAFIF	TPIF	TSIF	EIF	AIF
<hr/>							
位	标记	位名	功能			读写	
b7	RIF	接收中断标志	接收中断标志 (Receive Interrupt Flag) 0: 未收到数据帧 1: 接收到有效的数据帧或者远程帧 通过应用程序写1清0。			R/W	
b6	ROIF	接收上溢中断标志	接收上溢中断标志 (Receive Overrun Interrupt Flag) 0: 未发生接收上溢 1: 发生接收上溢 上溢时ROIF和RFIF同时置1。 通过应用程序写1清0。			R/W	
b5	RFIF	接收BUF满中断标志	接收BUF满中断标志 (RB Full Interrupt Flag) 0: RB FIFO未满 1: RB FIFO满 通过应用程序写1清0。			R/W	
b4	RAFIF	接收BUF将满中断标志	接收BUF将满中断标志 (RB Almost Full Interrupt Flag) 0: 被填充的RB SLOT数目小于AFWL设定值 1: 被填充的RB SLOT数目大于等于AFWL设定值 通过应用程序写1清0。			R/W	
b3	TPIF	PTB发送中断标志	PTB发送中断标志 (Transmission Primary Interrupt Flag) 0: 没有PTB发送完成 1: 请求的PTB发送成功完成 通过应用程序写1清0。 注意: TTCAN模式时, TPIF无效, 仅适用TSIF标志			R/W	
b2	TSIF	STB发送中断标志	STB发送中断标志 (Transmission Secondary Interrupt Flag) 0: 没有STB发送完成 1: 请求的STB发送成功完成 通过应用程序写1清0。 注意: TTCAN模式时, TPIF无效, 仅使用TSIF标志			R/W	

位	标记	位名	功能	读写
b1	EIF	错误中断标志	<p>错误中断标志 (Error Interrupt Flag)</p> <p>0: BUSOFF位未发生变化，或者错误计数器的值与ERROR warning limit设定值的相对关系未发生变化。</p> <p>1: BUSOFF位发生变化，或者错误计数器的值与ERROR warning limit设定值的相对关系发生变化。比如错误计数器的值从小于设定值变为大于设定值，或者从大于设定值变为小于设定值。</p> <p>通过应用程序写1清0。</p>	R/W
b0	AIF	取消发送中断标志	<p>取消发送中断标志 (Abort Interrupt Flag)</p> <p>0: 未取消发送数据</p> <p>1: 通过TPA和TSA请求的发送消息被成功取消。</p> <p>通过应用程序写1清0。</p>	R/W

30.5.9 CAN 错误中断使能和标志寄存器 (CAN_ERRINT)

CAN ERROr INTerrupt Enable and Flag Register

偏移地址: 0xA6

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
EWARN	EPASS	EPIE	EPIF	ALIE	ALIF	BEIE	BEIF

位	标记	位名	功能	读写
b7	EWARN	到达设定的 ERROR WARNING LIMIT	到达设定的ERROR WARNING LIMIT (Error WARNING limit reached) 0: RECNT或者TECNT小于EWL设定值 1: RECNT或者TECNT大于等于EWL设定值 通过应用程序写1清0。	R/W
b6	EPASS	错误被动 0: 1: 通过应用程序写1清0。	错误被动 (Error Passive mode active) 0: 节点是主动错误节点 1: 节点是被动错误节点	R
b5	EPIE	错误被动中断使 能 0: 1:	错误被动中断使能 (Error Passive Interrupt Enable) 0: 禁止 1: 使能	R/W
b4	EPIF	错误被动中断标 志 0: 1: 通过应用程序写1清0。	错误被动中断标志 (Error Passive Interrupt Flag) 0: 未发生错误主动到错误被动或者错误被动到错误主动的变化 1: 发生错误主动到错误被动或者错误被动到错误主动的变化	R/W
b3	ALIE	仲裁失败中断使 能 0: 1:	仲裁失败中断使能 (Arbitration Lost Interrupt Enable) 0: 禁止 1: 使能	R/W
b2	ALIF	仲裁失败中断标 志 0: 1: 通过应用程序写1清0。	仲裁失败中断标志 (Arbitration Lost Interrupt Flag) 0: 仲裁成功 1: 仲裁失败	R/W
b1	BEIE	总线错误中断使 能 0: 1:	总线错误中断使能 (Bus Error Interrupt Enable) 0: 禁止 1: 使能	R/W
b0	BEIF	总线错误中断标 志 0: 1: 通过应用程序写1清0。	总线错误中断标志 (Bus Error Interrupt Flag) 0: 仲裁成功 1: 仲裁失败	R/W

30.5.10 CAN 位时序寄存器 (CAN_BT)

CAN Bit Timing Register

偏移地址: 0xA8

复位值: 0x0102 0203

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PRESC[7:0]								-	SJW[6:0]						
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	SEG_2[6:0]								SEG_1[7:0]						

位	标记	位名	功能	读写
b31~b24	PRESC	预分频设定	预分频设定 (Prescaler) 此寄存器设定值时将模块BCLK设定位 (PRESC+1) 分频作为TQ的时钟。	R/W
b23	Reserved	-	必须保持复位值。	R
b22~b16	SJW	再同步补偿宽度 时间设定	再同步补偿宽度时间设定 (Bit Timing Segment 2) 再同步补偿宽度时间=(SJW+1)*TQ	R/W
b15	Reserved	-	必须保持复位值。	R
b14~b8	SEG_2	位段2时间设定	位段2时间单元设定 (Bit Timing Segment 2) 位段2时间=(SEG_2+1)*TQ	R/W
b7~b0	SEG_1	位段1时间设定	位段1时间单元设定 (Bit Timing Segment 1) 位段1时间=(SEG_1+2)*TQ	R/W

30.5.11 CAN 错误和仲裁失败捕捉寄存器 (CAN_EALCAP)

CAN Error and Arbitration Lost Capture Register

偏移地址: 0xB0

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
KOER[2:0]						ALC[4:0]	

位	标记	位名	功能	读写
b7~b5	KOER	错误类别	错误类别 (Kind Of Error)	R
			000: 无错误 001: 位错误 010: 形式错误 011: 填充错误 100: 应答错误 101: CRC错误 110: 其他错误 111: 保留	
			有错误时KOER位更新，正常发送接收时KOER位保持不变。	
b4~b0	ALC	仲裁失败位置捕 捉	仲裁失败位置捕捉 (Arbitration Lost Capture) 仲裁失败时ALC记录一帧数据中仲裁失败时的位置。	R/W

30.5.12 CAN 警告限定寄存器 (CAN_LIMIT)

CAN Warning Limits Register

偏移地址: 0xA7

复位值: 0x1B

b7	b6	b5	b4	b3	b2	b1	b0
AFWL[3:0]				EWL[3:0]			

位	标记	位名	功能	读写
b7~b4	AFWL	接收BUF将满 Warning Limit	接收BUF将满Warning Limit (receive buffer Almost Full Warning Limit) 设定值范围为1~10。 AFWL=0无意义，当做AFWL=1处理。	R/W
b3~b0	EWL	Error Waring Limit编程值	Error Waring Limit编程值 (Programmable Error Waring Limit) Error Waring Limit= (EWL+1) *8。 该寄存器设定值影响EIF标志。	R/W

30.5.13 CAN 接收错误计数器寄存器 (CAN_RECNT)

CAN Receive Error CouNT Register

偏移地址: 0xB2

复位值: 0x00

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

位	标记	位名	功能	读写
b7~b0	RECNT	接收错误计数器	接收错误计数器 (Receive Error CouNT) 接收错误计数器根据CAN协议规定的错误计数增加或者减少。该计数器不存在上溢，255为最大值。	R/W

30.5.14 CAN 发送错误计数器寄存器 (CAN_TECNT)

CAN Transmit Error CouNT Register

偏移地址: 0xB3

复位值: 0x00

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

位	标记	位名	功能	读写
b7~b0	TECNT	发送错误计数器	发送错误计数器 (Transmit Error CouNT) 发送错误计数器根据CAN协议规定的错误计数增加或者减少。该计数器不存在上溢，255为最大值。	R/W

30.5.15 CAN 筛选器组控制寄存器 (CAN_ACFCTRL)

CAN Acceptance Filter Control Register

偏移地址: 0xB4

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-		SELMASK	-				ACFADR

位	标记	位名	功能	读写
b7~b6	Reserved	-	必须保持复位值。	R
b5	SELMASK	选择筛选器的屏蔽寄存器	选择筛选器的屏蔽寄存器 (SElect acceptance MASK) 0: ACF指向筛选器ID寄存器 1: ACF指向筛选器MASK寄存器 通过ACFADR选择具体的筛选寄存器组	R/W
b4	Reserved	-	必须保持复位值。	R
b3~b0	ACFADR	筛选器地址	筛选器地址 (acceptance filter address) ACFADR指向具体的筛选器，通过SELMASK去区分ID和MASK。 0000: 指向ACF_1 0001: 指向ACF_2 0010: 指向ACF_3 0011: 指向ACF_4 0100: 指向ACF_5 0101: 指向ACF_6 0110: 指向ACF_7 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111: 指向ACF_8	R/W

30.5.16 CAN 筛选器组使能寄存器 (CAN_ACFEN)

CAN Acceptance Filter Enable Register

偏移地址: 0xB6

复位值: 0x01

b7	b6	b5	b4	b3	b2	b1	b0
AE_8	AE_7	AE_6	AE_5	AE_4	AE_3	AE_2	AE_1

位	标记	位名	功能	读写
b7	AE_8	ACF_8使能	ACF_8使能 (Acceptance Filter 8 Enable)	R/W
		0: 禁止		
		1: 使能		
b6	AE_7	ACF_7使能	ACF_7使能 (Acceptance Filter 7 Enable)	R/W
		0: 禁止		
		1: 使能		
b5	AE_6	ACF_6使能	ACF_6使能 (Acceptance Filter 6 Enable)	R/W
		0: 禁止		
		1: 使能		
b4	AE_5	ACF_5使能	ACF_5使能 (Acceptance Filter 5 Enable)	R/W
		0: 禁止		
		1: 使能		
b3	AE_4	ACF_4使能	ACF_4使能 (Acceptance Filter 4 Enable)	R/W
		0: 禁止		
		1: 使能		
b2	AE_3	ACF_3使能	ACF_3使能 (Acceptance Filter 3 Enable)	R/W
		0: 禁止		
		1: 使能		
b1	AE_2	ACF_2使能	ACF_2使能 (Acceptance Filter 2 Enable)	R/W
		0: 禁止		
		1: 使能		
b0	AE_1	ACF_1使能	ACF_1使能 (Acceptance Filter 1 Enable)	R/W
		0: 禁止		
		1: 使能		

30.5.17 CAN 筛选器组 code 和 mask 寄存器 (CAN_ACF)

CAN Acceptance Filter code and mask Register

偏移地址: 0xB8

复位值: 0xFFFF XXXX

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	AIDEE	AID E	ACODE[28:16] or AMASK[28:16]												
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ACODE[15:0] or AMASK[15:0]															

位	标记	位名	功能	读写
b31	Reserved	-	读出值为不定。	R
b30	AIDEE	IDE位比较使能	IDE位比较使能 (Acceptance mask IDE bit check enable) 尽在SELMASK=1时有效 0: 筛选器接收标准格式和扩展格式帧 1: 筛选器接收AIDE位定义的标准格式或者扩展格式帧	R/W
b29	AIDE	IDE位MASK	IDE位MASK 0: 筛选器仅接收标准格式 1: 筛选器仅接收扩展格式	R/W
b28~b0	ACODE/ AMASK	筛选器CODE/ 筛选器MASK	筛选器CODE (acceptance filter code) 通过ACFADR指向具体的筛选器。 SELMASK=0时表示筛选器的CODE。 标准格式时使用位10~位0, 扩展格式时使用位28~位0。 筛选器CODE (acceptance filter mask) 通过ACFADR指向具体的筛选器。 SELMASK=1时表示筛选器的MASK。 标准格式时使用位10~位0, 扩展格式时使用位28~位0。	R/W

30.5.18 TTCAN TB slot 指针寄存器 (CAN_TBSLOT)

TTCAN TB Slot Pointer Register

偏移地址: 0xBE

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
TBE	TBF	-	-	-			TBPTR[2:0]

位	标记	位名	功能	读写
b7	TBE	设置TB为空	设置TB为空 (set TB slot to "empty") 0: 无操作 1: 被TBPTR选择的SLOT被标记为空 当SLOT被标记为空并且TSFF=0时, TBE自动复位为0。 如果设定此位为1时, 被选定的SLOT中存在数据正在发送状态则TBE=1, 则等到发送完成、发送错误或者发送取消后TBE复位为0。 TBE优先级高于TBF。	R/W
b6	TBE	设置TB为已填充	设置TB已填充 (set TB slot to "Filled") 0: 无操作 1: 被TBPTR选择的SLOT被标记为已填充 当SLOT被标记为已填充并且TSFF=1时, TBE自动复位为0。	R/W
b5~b3	Reserved	-	必须保持复位值。	R
b2~b0	TBPTR	TB SLOT指针	TB SLOT指针 (Pointer to a TB message slot) 000: 指向PTB 001: 指向STB SLOT1 010: 指向STB SLOT2 011: 指向STB SLOT3 100: 指向STB SLOT4 其他: 设定禁止 被指向的TB SLOT可以通过TBUF进行读写访问, 并且可以通过TBE和TBF来标记是否已经被填充。 TTCAN模式时, TBSEL和TSNEXT寄存器无效。 注意: 仅可以在TSFF=0时对该位进行写操作。	R/W

30.5.19 TTCAN 时间触发配置寄存器 (CAN_TTCFG)

TTCAN TB Slot Pointer Register

偏移地址: 0xBF

复位值: 0x90

b7	b6	b5	b4	b3	b2	b1	b0
WTIE	WTIF	TEIF	TTIE	TTIF	T_PRESC[1:0]		TTEN

位	标记	位名	功能	读写
b7	WTIE	触发看门中断使能	触发看门中断使能 (Watch Trigger Interrupt Enable)	R/W
		使能	0: 禁止 1: 使能	
b6	WTIF	触发看门中断标志	触发看门中断标志 (Watch Trigger Interrupt Flag)	R/W
		标志	当CYCLE COUNT值=TT_WTRIG设定值时且WTIE=1时， WTIF置位。 通过应用程序写1清0。	
b5	TEIF	触发错误中断使能	触发错误中断标志 (Trigger Error Interrupt Flag)	R
		标志	TT_TTIG设定值小于实际的CYCLE_TIME时， TEIF置位。	
b4	TTIE	时间触发中断使能	时间触发中断使能 (Time Trigger Interrupt Enable)	R/W
		使能	0: 禁止 1: 使能	
b3	TTIF	时间触发中断标志	时间触发中断标志 (Time Trigger Interrupt Flag)	R/W
		标志	当CYCLE COUNT值=TT_TRIG设定值时且TTIE=1时， TTIF置位。 如果TT_TRIG没有更新，则TTIF只置位1次，下一个基本CYCLE不置位。 通过应用程序写1清0。	
b2~b1	T_PRESC	TTCAN计数器预分频	TTCAN计数器预分频 (TTCAN Timer PREScaler)	R/W
		预分频	00: BT寄存器设定的位时间的1分频 01: BT寄存器设定的位时间的2分频 10: BT寄存器设定的位时间的4分频 11: BT寄存器设定的位时间的8分频 注意: T_PRESC可在TTEN=0时进行写操作或者写TTEN=1时同时操作。	
b0	TTEN	TTCAN使能	TTCAN使能 (Time Trigger Enable)	R/W
		0: 禁止 1: 使能TTCAN，计数器开始计数。		

30.5.20 TTCAN 参考消息寄存器 (CAN_REF_MSG)

TTCAN Reference Message Register

偏移地址: 0xC0

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
REF_IDE	-	REF_ID[28:16]													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
REF_ID[15:0]															

位	标记	位名	功能	读写
b31	REF_IDE	参考消息的IDE	参考消息的IDE位 (REference message IDE bit)	R/W
		位	0: 标准格式 1: 扩展格式	
b30~b29	Reserved	-	读出值为不定。	R
b28~b0	REF_ID	参考消息的ID位	参考消息的ID位 (REference message IDentifier) REF_IDE=0: REF_ID[10:0]有效 REF_IDE=1: REF_ID[28:0]有效 REF_ID用于检测参考消息，适用于发送和接收。 检测到参考消息后，当前帧的Sync_Mark则变成Ref_Mark。 REF_ID[2:0]固定为0，并不检查其值，这样最多可以支持8个潜在的time master。 当REF_MSG的最高字节写操作后，则需要等待6个CAN时钟周期以完成REF_MSG向CAN时钟域的传递。	R/W

30.5.21 TTCAN 触发配置寄存器 (CAN_TRG_CFG)

TTCAN Reference Message Register

偏移地址: 0xC4

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TEW[3:0]				-	TTYPE[2:0]			-				TTPTR[2:0]			

位	标记	位名	功能	读写
b15~b12	TEW	发送使能窗口	发送使能窗口 (Transmit Enable Window) 用于TTCAN的单次发送触发模式 (Single Shot Transmit Trigger), 可以设定TEW+1个cycle time的窗口, 发送仅在此窗口内被允许。	R/W
b11	Reserved	-	必须保持复位值。	R
b10~b8	TTYPE	触发类型	触发类型 (Trigger Type) 000: 立即触发 (Immediate Trigger for immediate transmission) 001: 时间触发 (Time Trigger for receive triggers) 010: 单次发送触发 (Single Shot Transmit Trigger for exclusive time windows) 011: 发送开始触发 (Transmit Start Trigger for merged arbitrating time windows) 100: 发送停止触发 (Transmit Stop Trigger for merged arbitrating time windows) 其他: 保留 触发时间通过TT_TRIGGER寄存器设定, TB Slot通过TTPTR选择。	R/W
b7~b3	Reserved	-	必须保持复位值。	R
b2~b0	TTPTR	发送触发器TB slot指针	发送触发器TB slot指针 (Transmit Trigger TB slot Pointer) 000: 指向PTB 001: 指向STB SLOT1 010: 指向STB SLOT2 011: 指向STB SLOT3 100: 指向STB SLOT4 其他: 设定禁止 如果指向的TB SLOT被标记为空, 当到达触发时间后, TEIF置位。	R/W

30.5.22 TTCAN 触发时间寄存器 (CAN_TT_TRIG)

TTCAN Reference Message Register

偏移地址: 0xC6

复位值: 0x0000

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

位	标记	位名	功能	读写
b15~b0	TT_TRIG	触发时间	触发时间 (Trigger Time) 用于指定触发器的cycle time，对于发送触发器来说发送SOF时间大约是TT_TRIG设定值+1 当TT_TRIG的最高字节写操作后，TT_TRIG值开始向CAN时钟域的传递。因此如果BYTE操作，需先写低字节再写高字节。	R/W

30.5.23 TTCAN 触发看门时间寄存器 (CAN_TT_WTRIG)

TTCAN Watch Trigger Time Register

偏移地址: 0xC8

复位值: 0xFFFF

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

位	标记	位名	功能	读写
b15~b0	TT_TRIG	触发时间	触发时间 (Trigger Time) 用于指定看门触发器的cycle time。 当TT_WTRIG的最高字节写操作后，TT_WTRIG值开始向CAN时钟域的传递。因此如果BYTE操作，需先写低字节再写高字节。	R/W

30.6 使用注意事项

30.6.1 CAN 总线抗干扰措施

CAN 总线广泛应用于汽车、工业控制等行业，如果 CAN 应用现场电磁环境比较恶劣，存在电路不平衡性、空间电磁场、电网进线等因素，会导致 CAN 总线因辐射、传导干扰而产生大量通信噪声，致使总线错误帧增加、重发频繁，正确数据不能及时到达等情况，严重影响数据通信质量。所以实际应用中应该致力于消除噪声干扰，保障 CAN 总线网络稳定工作。

以下是几类常用的 CAN 总线抗干扰措施（包括但不仅限于）

- 1) 增加 CAN 总线接口电气隔离
- 2) 共收发器的信号地
- 3) 使用屏蔽双绞线缆并正确接地
- 4) 提高 CAN 传输线双绞程度
- 5) 增加信号保护器
- 6) 改进网络拓扑
- 7) 应用层软件抗干扰机制

30.6.2 CAN 控制器噪声制约

在 CAN 总线网络中应确保通信的位时间满足标准协议的要求，若引入不满足位时间宽度的噪声干扰，可能引起 CAN 控制器异常动作。

31 USB2.0 全速模块（USBFS）

31.1 USBFS 简介

USB 全速（USBFS）控制器为便携式设备提供了一套 USB 通信解决方案。USBFS 控制器支持主机模式和设备模式，且芯片内部集成全速 PHY。主机模式下，USBFS 控制器支持全速（FS, 12Mb/s）和低速（LS, 1.5Mb/s）收发器，而设备模式下则仅支持全速（FS, 12Mb/s）收发器。USBFS 控制器支持 USB 2.0 协议所定义的所有传输方式（控制传输、批量传输、中断传输和同步传输）。

31.2 USBFS 主要特性

主要分为三类：通用特性、主机模式特性和设备模式特性。

31.2.1 通用特性

- 内置片上 USB2.0 全速 PHY
- 支持主机模式和设备模式
- 支持 FS SOF 以及低速“Keep-alive”令牌并具有以下功能：
 - SOF 脉冲管脚输出功能
 - SOF 脉冲可作为芯片内部事件源去触发 TIMER, DMA 等模块工作
 - 可配置帧周期
 - 可配置的帧结束中断
 - 模块内嵌 DMA，并可软件配置 AHB 突发传输类型
 - 具备省电功能，例如 USB 挂起，停止 RAM 时钟，停止 PHY 域时钟
 - 具有采用高级 FIFO 控制的 1.25KB 专用 RAM
 - 可以将 RAM 空间划分为不同的 FIFO，以便灵活有效的使用 RAM
 - 每个 FIFO 可存储多个数据包
 - 动态分配存储区
 - FIFO 的大小可配置成为非 2 的幂次方值，以便连续使用存储单元
 - 一帧之内可以不需要应用程序干预，以达到最大 USB 带宽

- 可根据 ID 线的电平自动确定主机模式或者设备模式

31.2.2 主机模式特性

- 主机模式支持 USB2.0 全速 (FS, 12Mb/s) 和低速 (LS, 1.5Mb/s) 传输
- 需要通过外部电源芯片生成 VBUS 电压
- 多达 12 主机通道 (管道): 每个通道都可以动态实现重新配置, 可支持任何类型的 USB 传输
- 内置硬件调度器, 可:
 - 在周期性硬件队列中存储多达 8 个中断加同步传输请求
 - 在非周期性硬件队列中存储多达 8 个控制加批量传输请求
 - 管理一个共享 RX FIFO、一个周期性 TX FIFO 和一个非周期性 TX FIFO, 以有效使用 USB 数据 RAM

31.2.3 设备模式特性

- 从机模式支持 USB2.0 全速 (FS, 12Mb/s) 传输。
- 1 个双向控制端点 0
- 5 个 OUT 端点, 可以配置为支持批量传输、中断传输或同步传输
- 5 个 IN 端点, 可以配置为支持批量传输、中断传输或同步传输
- 包含 6 个发送 FIFOs(每个 IN 端点配备一个发送 FIFO)和一个接收 FIFO (由所有的 OUT 端点共享)
- 支持远程唤醒功能。
- 支持软断开功能
- VBUS PIN 支持 5V 耐压。

31.3 USBFS 系统框图

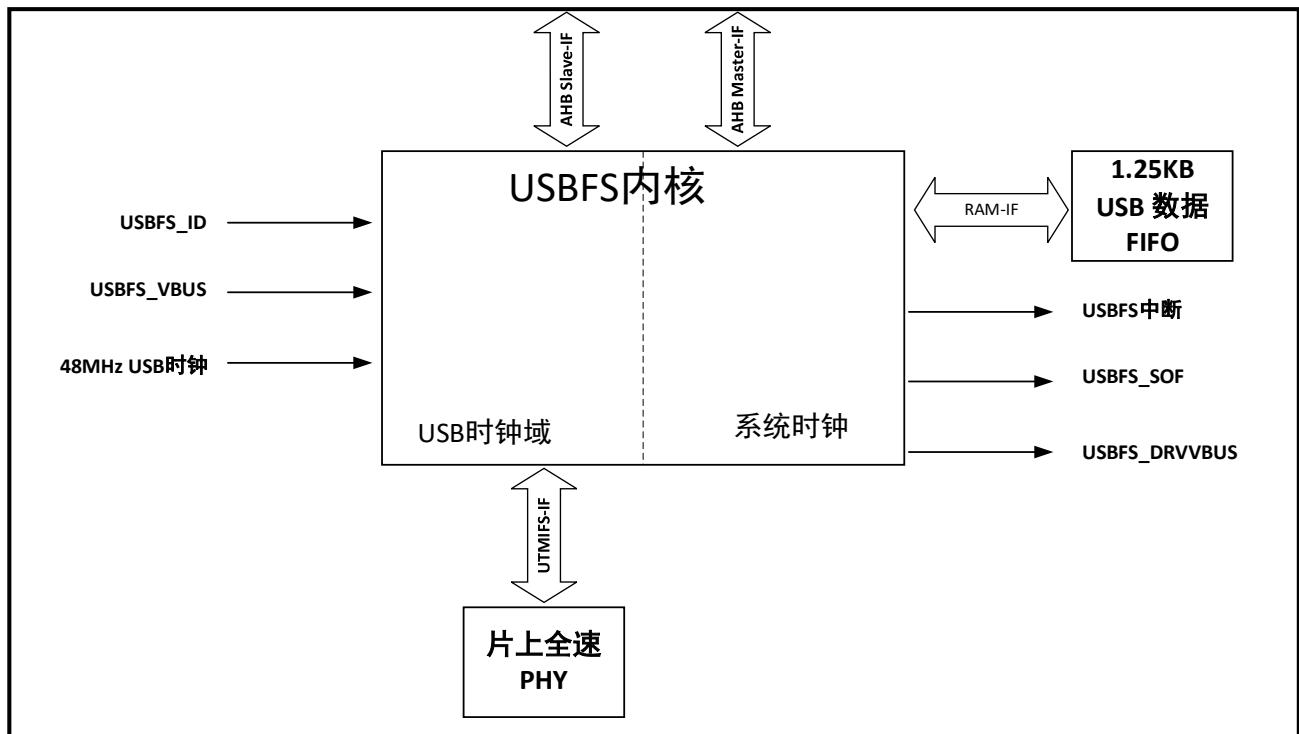


图 31-1 USBFS 系统框图

31.4 USBFS 管脚说明

管脚名	方向	功能描述
USBFS_VBUS	输入	电源端口, 5V耐压
USBFS_DP	输入/输出	差分数据D+信号
USBFS_DM	输入/输出	差分数据D-信号
USBFS_DRVVBUS	输出	外部电源芯片使能信号
USBFS_ID	输入	USB A-B器件识别信号
USBFS_SOF	输出	SOF输出脉冲信号

表 31-1 USBFS 管脚说明

由于 USBFS_DP 和 USBFS_DM 管脚和通用 GPIO 复用，在使用 USBFS 时，建议关闭其对应管脚的数字功能，具体请参考通用 IO (GPIO) 章节。另外 USBFS 功能不使用时，USBFS_DP 和 USBFS_DM 管脚对应的数字功能管脚翻转时，会产生额外的电流消耗。

31.5 USBFS 功能说明

31.5.1 USBFS 时钟以及工作模式

USBFS 所使用的时钟需要配置为 48MHz，该 48MHz 时钟由内部 PLL 电路产生，PLL 时钟源需选择外部高速振荡器，使用 USBFS 模块前，需要在 CMU 模块内配置好 USBFS 时钟。

USBFS 可以作为主机或者设备使用，并且包含一个片上全速 PHY。

片上全速 PHY 的内部集成了上拉和下拉电阻，USBFS 可以根据当前模式和连接状态自动选择。

USBFS 工作时，VCC 电压范围为 3.0~3.6V。

31.5.2 USBFS 模式决定

USBFS 决定当前工作模式有以下两种方法：

方法 1：根据 USBFS_ID 线的状态自动识别，检测到 USBFS_ID 线为高电平时，模块工作在设备模式，检测到 USBFS_ID 线为低电平时，模块工作在主机状态。

方法 2：强制主机/设备模式，通过设定寄存器 USBFS_GUSBCFG 的 FDMOD 或者 FHMOD 位为 1，使模块忽略 USBFS_ID 线的电平而强制工作在设备或者主机模式。

31.5.3 USBFS 主机功能

31.5.3.1 主机功能简介

当 USBFS 工作在主机模式下时, VBUS 为 USB 协议所规定的 5V 电源引脚。内部 PHY 不支持提供 5V 电源, 所以需要外接 USB 电源芯片为设备供电。USBFS_DRVVBUS 用来使能外接 USB 电源芯片, 外接电源芯片的过电流检出可以通过本 MCU 的外部中断 IRQ 实现。主机模式下 USB_VBUS 可以作为 GPIO 使用。

典型的 USB 主机模式系统构建图如下:

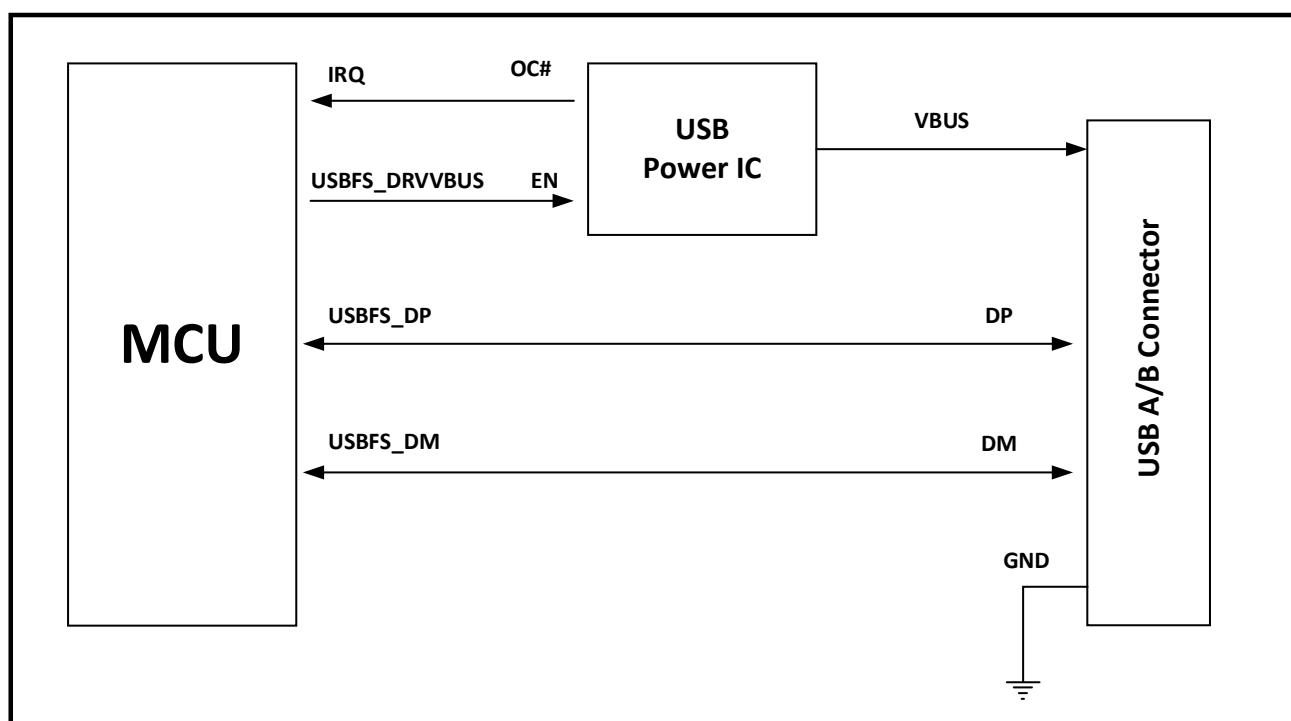


图 31-2 USBFS 主机模式系统构建图

31.5.3.2 主机端口供电

本 MCU 不能输出 5V 以提供 VBUS。为此, 必须在微控制器以外添加 USB 电源芯片或基本电源开关 (如应用电路板提供 5V 电源) 来驱动 5V VBUS 线。外部电源芯片可通过任何 GPIO 输出或者 USBFS_DRVVBUS 驱动。当应用程序确定使用 GPIO 来控制外部器件提供 VBUS, 仍须将主机端口控制和状态寄存器中的端口电源位 (USBFS_HPRT 中的 PPWR 位) 置 1。

31.5.3.3 主机检测设备连接和断开

USB 设备将在连接后立即被检测到。USBFS 模块将发出主机端口中断信号，该中断由主机端口控制和状态寄存器中的设备连接位（USBFS_HPRT 中的 PCDET 位）触发。设备断开事件将触发断开连接检测中断（USBFS_GINTSTS 中的 DISCINT 位）。

31.5.3.4 主机枚举

检测到设备连接后，若又有新的设备连接进来，主机必须通过向新的设备发送 USB 复位和配置命令来启动枚举过程。

应用程序通过将主机端口控制和状态寄存器中的端口复位位（USBFS_HPRT 中的 PRST 位）置 1，使该过程最少持续 10ms、最多持续 20 ms，以此通过 USB 驱动 USB 复位信号（单端零）。应用程序计算这个过程的持续时间，然后将端口复位位清零。

USB 复位序列完成后，端口使能/禁止更改位（USBFS_HPRT 中的 PENCHNG 位）立即触发主机端口中断，进而向应用程序发出通知，指示可从主机端口控制和状态寄存器中的端口速度字段（USBFS_HPRT 中的 PSPD）读取枚举的设备速度，以及主机已经开始驱动 SOF(FS)或 Keep-alive 令牌 (LS)。此时主机已就绪，可通过对设备发送命令来完成对设备的枚举。

31.5.3.5 主机挂起

应用程序通过将主机端口控制和状态寄存器中的端口挂起位（USBFS_HPRT 中的 PSUSP）置 1 来挂起 USB 总线。USBFS 模块停止发送 SOF 并进入挂起状态。

可由远程设备的自主活动（远程唤醒）使总线退出挂起状态。这种情况下，远程唤醒信号将触发远程唤醒中断（USBFS_GINTSTS 中的 WKUPINT 位），硬件把主机端口控制和状态寄存器中的端口恢复位（USBFS_HPRT 中的 PRES 位）自行复位，并通过 USB 自动驱动恢复信号。应用程序必须为恢复窗口定时，然后将端口恢复位清零以退出挂起状态并重新启动 SOF。

如果由主机发起退出挂起状态，则应用程序必须将端口恢复位置 1 以启动主机端口上的恢复信号，为恢复窗口定时并最终将端口恢复位清零。

31.5.3.6 主机通道

USBFS 模块实现了 12 个主机通道。每个主机通道均可用于 USB 主机传输（USB 管

道)。主机最多能同时处理 8 个传输请求。如果应用程序有 8 个以上的传输请求挂起，则在通道从之前任务释放后(即，接收到传输完成和通道停止中断后)，主机控制器驱动器(HCD)必须为未处理的传输请求重新对通道进行分配。

每个主机通道都可配置为支持输入/输出以及周期性/非周期性事务。每个主机通道都使用专用控制(HCCHARx)寄存器、传输配置(HCTSIZx)寄存器/中断(HCINTx)寄存器以及和其相关的中断屏蔽寄存器(HCINTMSKx)。

主机通道控制

应用程序可通过主机通道 x 特性寄存器(HCCHARx)对主机通道作以下控制：

- 通道使能/禁止
- 设置目标 USB 设备的速度：FS/LS
- 设置目标 USB 设备的地址
- 设置与该通道通信的目标 USB 设备上的端点的编号
- 设置该通道上的传输方向：IN/OUT
- 设置该通道上的 USB 传输的类型：控制/批量/中断/同步
- 设置与该通道通信的设备端点的最大包长
- 设置要进行周期传输的帧：奇帧/偶帧

主机通道传输

主机通道传输大小寄存器(HCTSIZx)允许应用程序对传输大小参数进行编程并读取传输状态。必须在主机通道特性寄存器中的通道使能位置 1 之前完成对此寄存器的设置。使能端点后，数据包计数字段立即变为只读状态，同时 USBFS 模块根据当前传输状态对该字段进行更新。

可对以下传输参数进行编程：

- 以字节为单位的传输大小
- 构成整个传输大小的数据包个数
- 初始数据 PID

主机通道状态/中断

主机通道 x 中断寄存器(HCINTx)指示端点在出现 USB 和 AHB 相关事件时的状态。当中断寄存器中的主机通道中断位(USBFS_GINTSTS 中的 HCINT 位)置 1 时，应用

程序必须读取这些寄存器以获得详细信息。在读取这些寄存器之前，应用程序必须先读取主机全体通道中断 (HCAINT) 寄存器，以获取主机通道 x 中断寄存器的通道编号。应用程序必须将此寄存器中的相应位清零，才能将 HAINT 和 GINTSTS 寄存器中的相应位清零。 USBFS_HCINTMSK x 寄存器还提供每个通道各中断源的屏蔽位。

主机模块提供以下状态检查和中断产生功能：

- 传输完成中断，指示应用程序 (AHB) 和 USB 端均已完成数据传输
- 通道因传输完成、USB 事务错误或应用程序发出禁止命令而停止
- 相关的发送 FIFO 为半空或全空状态 (IN 端点)
- 接收到 ACK 响应
- 接收到 NAK 响应
- 接收到 STALL 响应
- 由于 CRC 校验失败、超时、位填充错误和错误的 EOP 导致 USB 事务错误
- 串扰错误
- 帧上溢
- 用于数据同步的翻转位出错

31.5.3.7 主机调度器

主机模块内置硬件调度器，可自主对应用程序发出的 USB 事务请求重新排序和管理。每一帧开始时，主机都先执行周期性（同步和中断）事务，然后执行非周期性（控制和批量）事务，以符合 USB 规范对同步和中断传输高优先级的保证。

主机通过请求队列（一个周期性请求队列和一个非周期请求队列）处理 USB 事务。每个请求队列最多可存储 8 个条目。每个条目代表一个应用程序发起但还未得到响应的 USB 事务请求，并存储了执行该 USB 事务所用到的 IN 或 OUT 通道的编号，以及其他相关信息。USB 事务请求在队列中的写入顺序决定了事务在 USB 接口上的执行顺序。

每一帧开始时，主机都先处理周期性请求队列，然后处理非周期性请求队列。如果当前帧结束时，计划在当前帧执行的同步或中断类型的 USB 传输事务请求仍处于挂起状态，则主机将发出未完成周期性传输中断 (USBFS_GINTSTS 中的 IPXFR 位)。USBFS 模块负责对周期性和非周期性请求队列的管理。周期性发送 FIFO 和队列状态寄存器

(HPTXSTS)与非周期性发送 FIFO 和队列状态寄存器 (HNPTXSTS) 都为只读寄存器，应用程序可使用它们来读取各请求队列的状态，其中包括：

- 周期性（非周期性）请求队列中当前可用的空闲条目数（最多 8 个）
- 周期性（非周期性）TxFIFO（OUT 事务）中当前可用的空闲空间
- IN/OUT 令牌、主机通道编号和其它状态信息

由于每个请求队列最多可存储 8 个 USB 事务请求，因此应用程序可以把主机 USB 事务请求提前发送给调度器；实际的通信最晚会在调度器处理完已挂起的 8 个周期事务和 8 个非周期事务完成之后出现在 USB 总线上。

要向主机调度器（队列）发出事务请求，应用程序必须读取 USBFS_HNPTXSTS 寄存器中的 PTXQSAV 位或 USBFS_HNPTXSTS 寄存器中的 NPTQXSAV 位，确保周期性（非周期性）请求队列中至少有一个可用空间来存储当前请求。

31.5.4 USBFS 设备功能

31.5.4.1 设备功能简介

当 USBFS 工作在设备模式下时，VBUS 为 USB 协议所规定的 5V 电源引脚，为 5V 耐压管脚。本模块始终检测 VBUS 线的电平状态来连接或者断开设备。

典型的 USB 设备模式系统构建图如下：

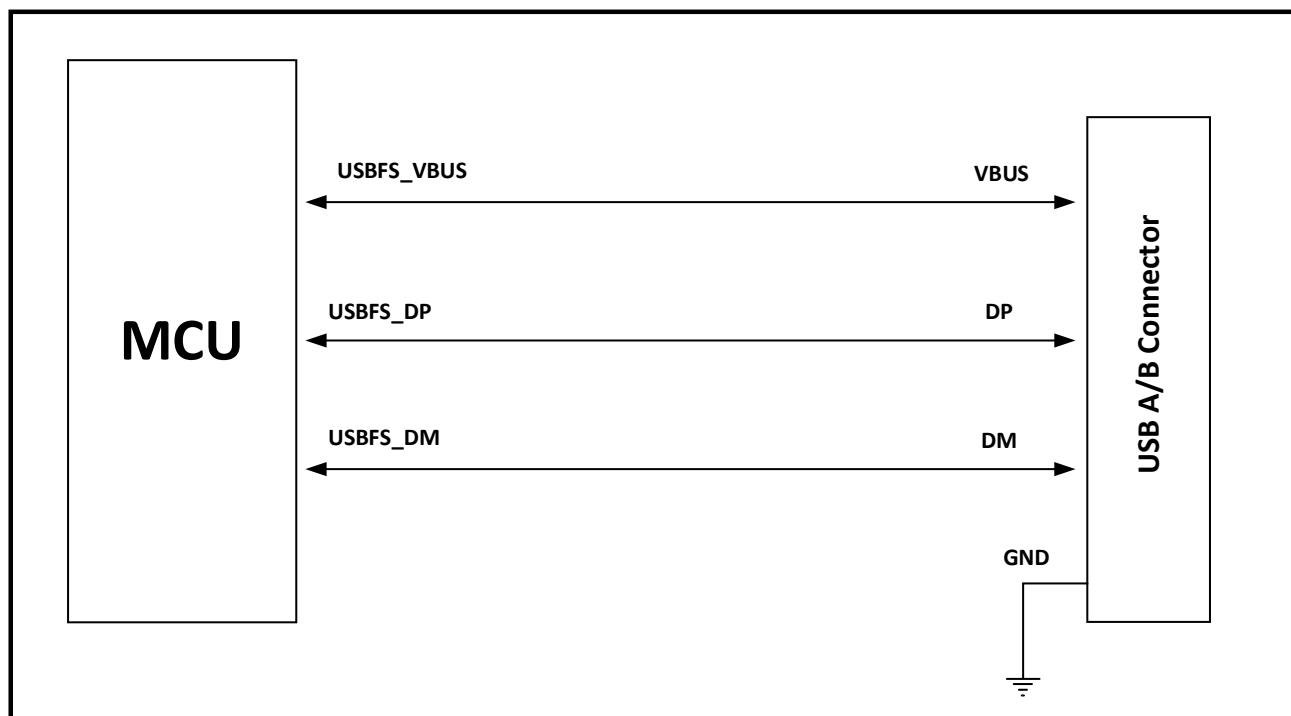


图 31-3 USBFS 设备模式系统构建图

31.5.4.2 设备供电状态

模块检测到 USBFS_VBUS 为高电平时，就会使 USB 设备进入供电状态。然后，USBFS 自动连接 DP 上拉电阻，发出全速设备与主机相连的信号并生成会话请求中断（USBFS_GINTSTS 中的 VBUSVINT 位），指示进入供电状态。

此外，USBFS_VBUS 输入还可确保主机在 USB 操作期间提供有效的 VBUS 电平。如果检测到 VBUS 电平为低电平（例如，因电源干扰或主机端口关闭引发），USBFS 将自动断开连接。

供电状态下，USBFS 期望收到来自主机的复位信号。其它 USB 操作则无法执行。收到复位信号后，立即生成检测到复位中断（USBFS_GINTST 中的 USBRST）。复位信

号结束后，将生成枚举完成中断（USBFS_GINTSTS 中的 ENUMDNE 位），USBFS 随即进入默认状态。

31.5.4.3 设备默认状态

默认状态下，USBFS 期望从主机收到 SET_ADDRESS 命令。其它 USB 操作则无法执行。当 USB 上解码出有效 SET_ADDRESS 命令时，应用程序会将相应的地址值写入设备配置寄存器中的设备地址字段（USBFS_DCFG 中的 DAD 位）。USBF 随即进入地址状态，并准备好以所配置的 USB 地址对主机事务进行应答。

31.5.4.4 设备挂起状态

USBFS 设备持续监视 USB 活动。在 USB 空闲时间达到 3ms 后，将发出早期挂起中断（USBFS_GINTSTS 中 ESUSP 位），并在 3ms 后由挂起中断（USBFS_GINTSTS 中的 USBSUSP 位）确认设备进入挂起状态。然后，设备状态寄存器中的设备挂起位（USBFS_DSTS 中的 SUSPSTS 位）自动置 1，USBFS 随即进入挂起状态。

可通过设备本身退出挂起状态。这种情况下，应用程序会将设备控制寄存器中的远程唤醒信号位（USBFS_DCTL 中的 RWUSIG 位）置 1，并在 1ms 到 15m 内将其清零。但若设备检测到主机发出的恢复信号，将生成恢复中断（USBFS_GINTSTS 中的 WKUPINT 位），设备挂起位自动清零。

31.5.4.5 设备软断开

供电状态可借助软断开功能通过软件退出。将设备控制寄存器中的软断开位（USBFS_DCTL 中的 SDIS 位）置 1 即可移除 DP 上拉电阻，此时尽管没有从主机端口实际拔出 USB 电缆，但主机端仍会发生设备断开检测中断。

31.5.4.6 设备端点

端点类别

USBFS 模块实现了以下 USB 端点：

- 控制端点 0：
 - 双向且仅处理控制消息
 - 使用一组单独的存储器来处理 IN 和 OUT 事务

- 专用控制 (USBFS_DIEPCTL0/USBFS_DOEPCTL0) 寄存器、传输配置 (USBFS_DIEPTSIZ0/USBFS_DOEPTSIZ0) 寄存器和状态中断 (USBFS_DIEPINT0/USBFS_DOEPINT0) 寄存器。控制和传输大小寄存器中可用的位组与其它端点中稍有不同
 - 5 个 IN 端点
- 每个端点都可配置为支持同步传输、批量传输或中断传输类型
- 每个端点都有专用控制 (USBFS_DIEPCTLx) 寄存器、传输配置 (USBFS_DIEPTSIZx) 寄存器和状态中断 (USBFS_DIEPINTx) 寄存器
- 设备 IN 端点通用中断屏蔽寄存器 (USBFS_DIEPMSK) 可用于使能/禁止所有 IN 端点（包括 EP0）上的同一类端点中断源
- 支持未完成的同步 IN 传输中断 (USBFS_GINTSTS 中的 IISOIXFR 位)，该中断将在当前帧中至少有一个同步 IN 端点上的传输未完成时触发。该中断和周期性帧中断 (USBFS_GINTSTS/EOPF) 一起触发
 - 5 个 OUT 端点
- 每个端点都可配置为支持同步传输、批量传输或中断传输类型
- 每个端点都有专用控制 (USBFS_DOEPCTLx) 寄存器、传输配置 (USBFS_DOEPTSIZx) 寄存器和状态中断 (USBFS_DOEPINTx) 寄存器
- 设备 OUT 端点通用中断屏蔽寄存器 (USBFS_DOEPMSK) 可用于使能/禁止所有 OUT 端点（包括 EP0）上的同一类端点中断源
- 支持未完成的同步 OUT 传输中断 (USBFS_GINTSTS 中的 INCOMPISOOUT 位)，该中断将在当前帧中至少有一个同步 OUT 端点上的传输未完成时触发。该中断和周期性帧中断 (USBFS_GINTSTS/EOPF) 一起触发

端点控制

应用程序可通过设备端点 x IN/OUT 控制寄存器 (DIEPCTLx/DOEPCTLx) 对端点采取以下控制：

- 端点使能/禁止
- 在当前配置下激活端点
- 设置 USB 传输类型（同步、批量和中断）

- 设置支持的数据包大小
- 设置与 IN 端点相关的 Tx-FIFO 编号
- 设置希望收到的或发送时要使用到的 data0/data1 PID（仅限批量/中断传输）
- 设置接收或发送事务时所对应的奇/偶帧（仅限同步传输）
- 可以设置 NAK 位，从而不论此时 FIFO 的状态如何，都对主机的请求回复 NAK
- 可以设置 STALL 位，使得主机对该端点的令牌都被硬件回复 STALL
- 可以将 OUT 端点设置为侦听模式，即对接收到的数据不进行 CRC 检查

端点传输

设备端点 x 传输长度寄存器 (DIEPTSIZx/DOEPTSIZx) 允许应用程序对传输长度参数进行编程并读取传输状态。必须在端点控制寄存器中的端点使能位置 1 之前完成对此寄存器的设置。使能端点后，这些字段立即变为只读状态，同时 USBFS 模块根据当前传输状态对这些字段进行更新。

需要对以下传输参数进行配置：

- 以字节为单位的单次传输的长度
- 构成整个传输的数据包的个数

端点状态/状态

设备端点 x 中断寄存器 (DIEPINTx/DOEPINTx) 指示端点在出现 USB 和 AHB 相关事件时的状态。当模块中断寄存器中的 OUT 端点中断位或 IN 端点中断位（分别为 USBFS_GINTSTS 中 OEPINT 位或 USBFS_GINTSTS 中的 IEPINT 位）置 1 时，应用程序必须读取这些寄存器以获得详细信息。在应用程序读取这些寄存器之前，必须先读取设备全体端点中断(USBFS_DAIINT) 寄存器，以获取设备端点 x 中断寄存器的端点编号。应用程序必须将此寄存器中的相应位清零，才能将 DAIINT 和 GINTSTS 寄存器中的相应位清零。

- 模块提供以下状态检查和中断功能：
- 传输完成中断，指示应用程序 AHB 和 USB 端均已完成数据传输
- Setup 阶段已完成（仅针对控制传输类型的 OUT 端点）

- 相关的发送 FIFO 为半空或全空状态（IN 端点）
- NAK 应答已发送到主机（仅针对同步传输的 IN 端点）
- TxFIFO 为空时接收到 IN 令牌（仅针对批量和中断传输类型的 IN 端点）
- 尚未使能端点时接收到 OUT 令牌
- 检测到 babble 错误
- 应用程序关闭端点生效
- 应用程序对端点设置 NAK 生效（仅针对同步传输类型的 IN 端点）
- 接收到 3 个以上连续 setup 数据包（仅针对控制类型的 OUT 端点）
- 检测到超时状况（仅针对控制传输类型的 IN 端点）
- 同步传输类型的数据包未产生中断而丢失

31.5.5 USBFS SOF 脉冲管脚输出功能

USBFS 在主机和设备模式下都可以监视、跟踪和配置 SOF 帧并且还具备 SOF 脉冲输出功能。SOF 脉冲通过 USBFS_SOF 管脚输出，输出宽度为 16 个系统时钟周期。

31.5.5.1 主机 SOF

主机模式下，可以在主机帧间隔寄存器 (HFIR) 中对所产生的两个连续 SOF (FS) 或 keep-alive (LS) 令牌期间所出现的 PHY 时钟数进行编程，进而应用程序可对 SOF 帧周期进行控制。帧开始 (USBFS_GINTSTS 中的 SOF 位) 时都将生成中断。当前帧编号和出现下一个 SOF 前剩余的时间应用程序在主机帧编号寄存器 (HFNUM) 中能够进行跟踪。

使用 USBFS 系统控制寄存器 USBFS_SYCTLREG 中的 SOFEN 位，可以使任何 SOF 令牌发出的同时产生的、宽度为 16 个系统时钟周期的 SOF 脉冲信号从 USBFS_SOF 引脚输出。

此外，SOF 脉冲还可以作为内部事件触发 DMA 传输、TIMER 计数等外部模块工作。

31.5.5.2 设备 SOF

在设备模式下，USB 每次接收到 SOF 令牌时，都将触发帧开始中断 (USBFS_GINTSTS 中的 SOF 位)。相应的帧编号可从设备状态寄存器 (USBFS_DSTS 中的 FNSOF 位) 读取。使用 USBFS 系统控制寄存器 USBFS_SYCTLREG 中的 SOFEN 位，还可以生成宽度为 16 个系统时钟周期的 SOF 脉冲信号，并使该信号在 USBFS_SOF 引脚输出，以实现外部可用。

此外，SOF 脉冲还可以作为内部事件触发 DMA 传输、TIMER 计数等外部模块工作。周期性帧结束中断(GINTSTS/EOPF)用于在经过了 80%、85%、90% 或 95% 的帧间隔时间时通知应用程序，具体取决于设备配置寄存器中的周期性帧间隔字段 (USBFS_DCFG 中的 PFIVL 位)。此功能可用于确定该帧的所有同步通信是否完成。

31.5.6 USBFS 功耗控制

不使用 USBFS 模块时，可以通过 CMU 模块停止 USBFS 模块的 HCLK 和 PHY 时钟，从而降低功耗。

使用 USB 模块，但设备 USB 会话未开始或设备未连接时，可以在 USB 挂起状态下使用功率降低技术。

- 停止 PHY 时钟（USBFS_GCCTL 中的 STPPCLK 位）

将时钟门控控制寄存器中的停止 PHY 时钟位置 1 时，USBFS 全速模块的大多数 48 MHz 内部时钟域均由时钟门控关闭。即使应用程序仍提供时钟输入，也会节省掉模块由于时钟信号翻转带来的动态功耗还会关掉收发器的大部分单元，只有负责检测异步恢复事件或远程唤醒事件的部分还保持工作状态。

- HCLK 门控（USBFS_GCCTL 中的 GATEHCLK 位）

将时钟门控控制寄存器中的 GATEHCLK 位置 1 时，USBFS 模块内部的大多数系统时钟域均由时钟门控关闭。只有寄存器读取和写入接口保持活动状态。即使应用程序仍提供时钟输入，也会节省掉模块由于时钟信号翻转带来的动态功耗。

为了节省动态功耗，只在 USB 数据 FIFO 被 USBFS 模块访问时为其提供时钟。

31.5.7 USBFS 动态更新 USBFS_HFIR 寄存器

主机模式下，USB 模块具有对帧周期进行动态微调的功能，能够将外部设备与 SOF 帧进行同步。如果 USBFS_HFIR 寄存器在当前 SOF 帧内发生更改，则将在下一个帧中对 SOF 周期进行相应修正，具体说明请参见图 31-4。

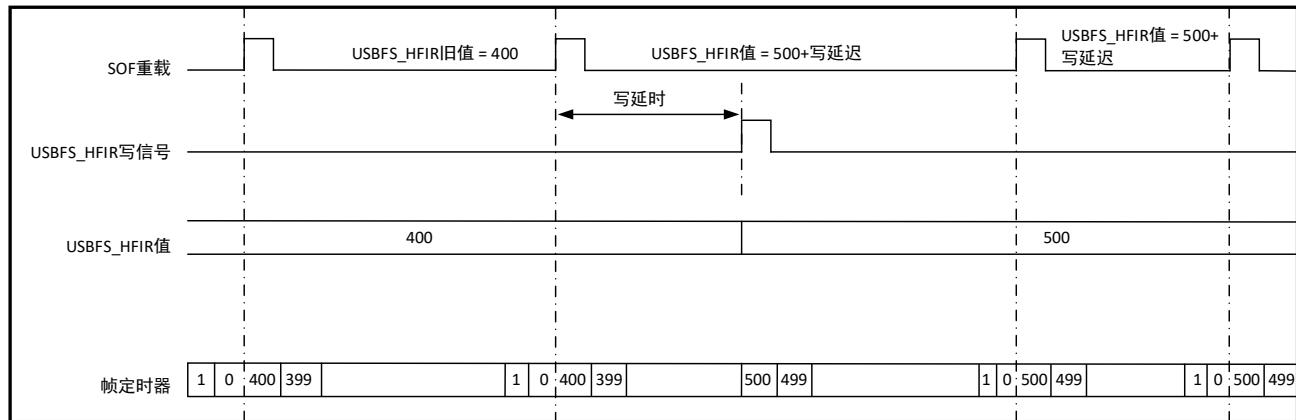


图 31-4 USBFS 动态更新 USBFS_HFIR 寄存器示意图

31.5.8 USBFS 数据 FIFO

USBFS 系统具有 1.25KB 专用 RAM，采用高效的 FIFO 控制机制。USBFS 模块中的数据包 FIFO 控制器模块将 RAM 空间划分为多个 TxFIFO（USB 传输前，应用程序将数据压入其中进行短暂存储）和单个 RxFIFO（从 USB 接收到的数据被应用程序读取之前，在其中进行短暂存储）。

RAM 中所构建的 FIFO 的数量与组织方式取决于设备的角色。设备模式下，为每个激活的 IN 端点配置一个 TxFIFO。FIFO 的大小均由软件配置，以更好地满足应用要求。

31.5.9 USBFS 主机 FIFO 架构

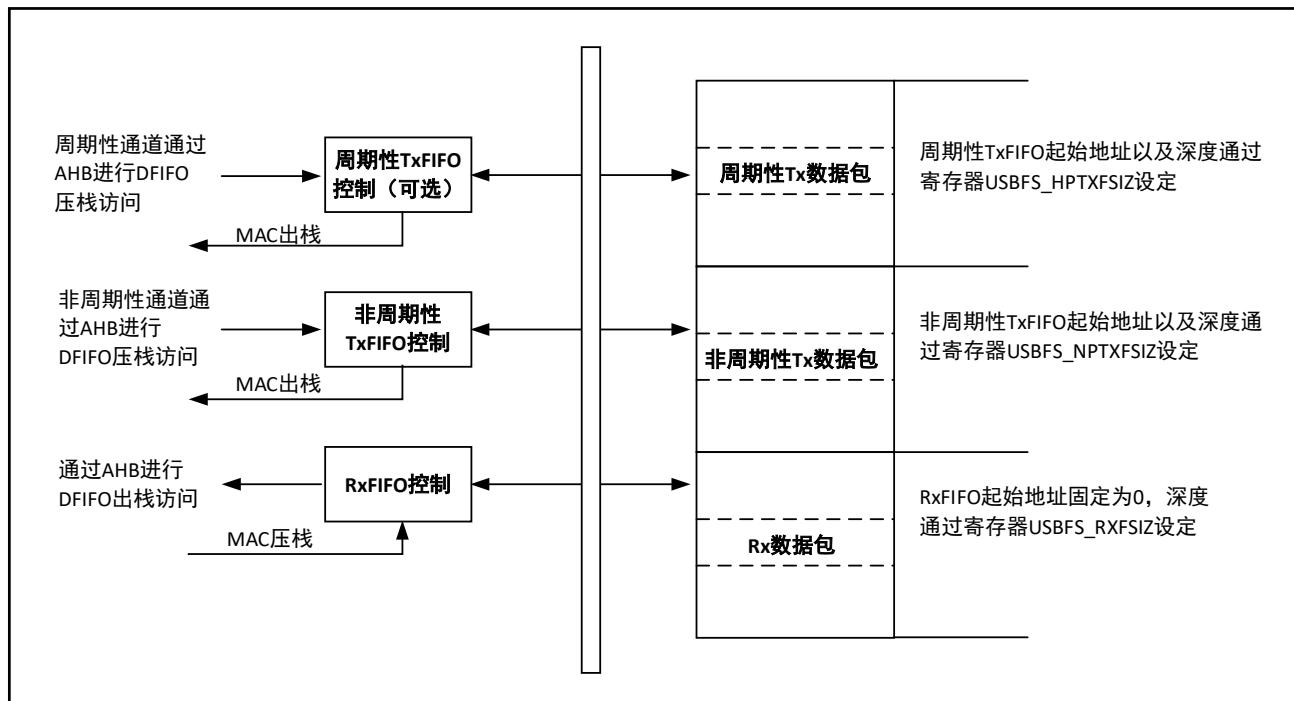


图 31-5 USBFS 主机模式下 FIFO 架构示意图

31.5.9.1 主机 RxFIFO

主机使用一个接收 FIFO 处理所有周期和非周期事务。FIFO 用作接收缓冲区以保存从 USB 接收到的数据（接收到的数据包的数据部分），直至这些数据传输到系统存储器。只要 FIFO 中有空间，来自设备 IN 端点的数据包就接收进来并挨个存储。接收到的每个数据包的状态（包含主机目标通道、字节数、数据 PID 和对所接收数据的校验）也存储在 FIFO 中。接收 FIFO 的大小在接收 FIFO 大小寄存器 (GRXFSIZ) 中配置。

单个接收 FIFO 架构使得 USB 主机高效地填充接收数据缓冲区：

- 所有 IN 配置主机通道共享同一个 RAM 缓冲区（共享 FIFO）
- 对于主机软件驱动的任意序列 IN 令牌，USBFS 模块可将接收 FIFO 填充至限值

只要至少有一个数据包在 Rx FIFO 中可供读取，应用程序就会接收 Rx FIFO 非空中断。应用程序从接收状态读取和出栈寄存器中读取数据包信息，最后从 Rx FIFO 中读出数据。

31.5.9.2 主机 TxFIFO

主机使用一个发送 FIFO 处理所有非周期（控制和批量）OUT 事务，使用另一个发送 FIFO 处理所有周期（同步和中断）OUT 事务。FIFO 用作发送缓冲区以保存要通过 USB 发送的数据（发送数据包）。周期（非周期）TxFIFO 的大小在主机周期（非周期）发送 FIFO 大小 (HPTXFSIZ/HNPTXFSIZ) 寄存器中配置。

两个 Tx FIFO 按优先级实施操作，周期性通信的优先级较高，因此在 USB 一帧的时间内首先进行周期性通信。帧起始时，内置的主机调度器先处理周期请求队列，再处理非周期请求队列。

两个发送 FIFO 的架构使得 USB 主机能够对周期和非周期发送数据缓冲区分别进行优化管理：

- 配置为支持周期（非周期）OUT 事务的所有主机通道共享同一个 RAM 缓冲区（共享 FIFO）
- 对于主机软件驱动的任意序列 OUT 令牌，USBFS 模块可将周期性（非周期性）发送 FIFO 填充至限值

只要周期性 TxFIFO 为半空或全空，USBFS 模块就会发出周期性 TxFIFO 空中断 (USBFS_GINTSTS 中的 PTXFE 位)，具体取决于 AHB 配置寄存器中的周期性 TxFIFO 空等级位 (USBFS_GAHBCFG 中的 PTXFELVL 位) 的值。只要周期性 TxFIFO 和周期性请求队列中均存在空闲空间，应用程序便可提前写入发送数据。可通过读取主机周期性发送 FIFO 和队列状态寄存器 (HPTXSTS) 来了解二者的可用空间。

只要非周期性 TxFIFO 为半空或全空，USBFS 模块就会发出非周期性 TxFIFO 空中断 (USBFS_GINTSTS 中的 NPTXFE 位)，具体取决于 AHB 配置寄存器中的非周期性 TxFIFO 空等级位 (USBFS_GAHBCFG 中的 TXFELVL 位)。只要非周期性 TxFIFO 和非周期性请求队列中均存在空闲空间，应用程序便可写入发送数据。可通过读取主机非周期性发送 FIFO 和队列状态寄存器 (HNPTXSTS) 来了解二者的可用空间。

31.5.10 USBFS 设备 FIFO 架构

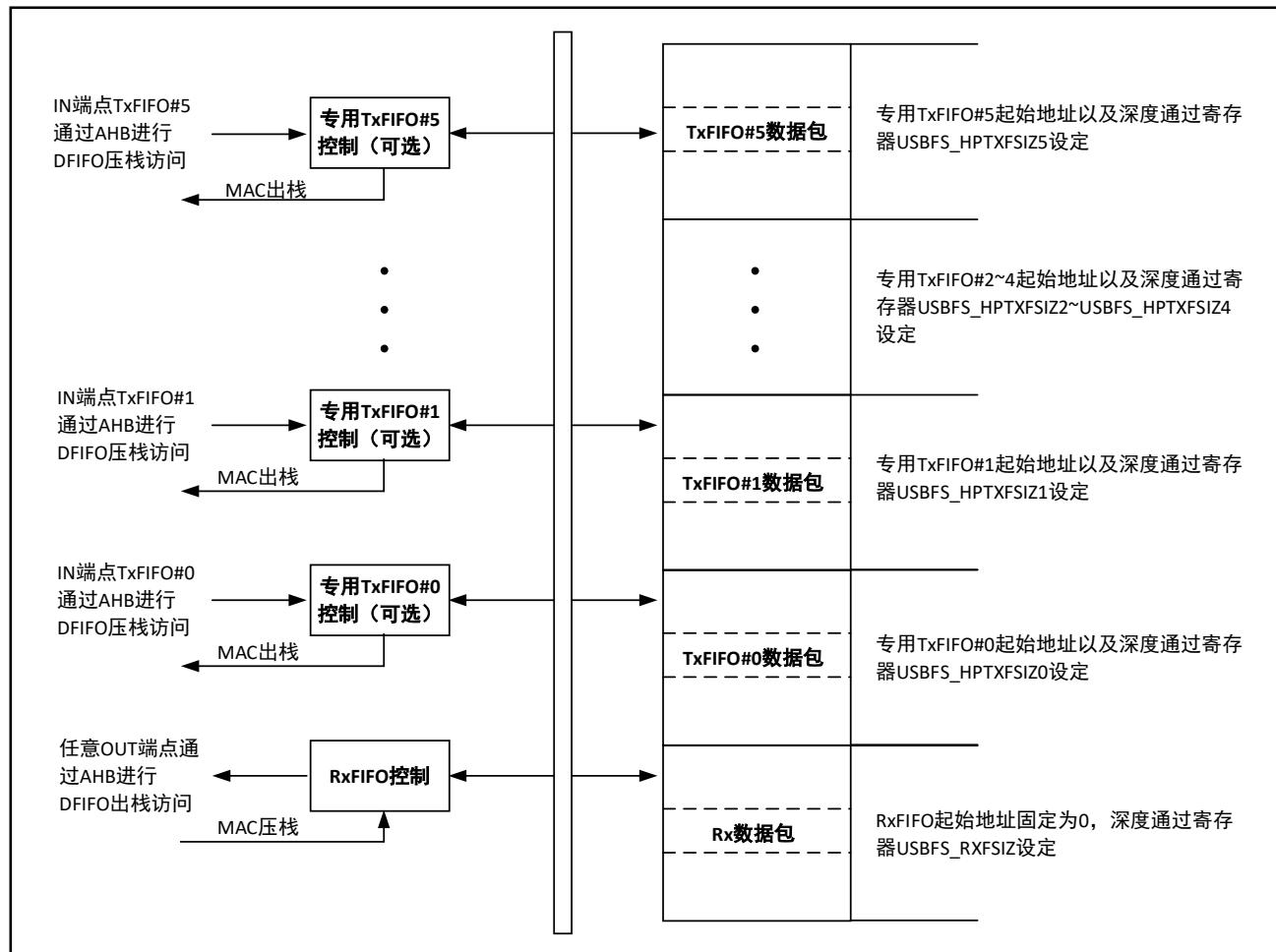


图 31-6 USBFS 设备模式下 FIFO 架构示意图

31.5.10.1 设备 RxFIFO

USBFS 设备使用单个接收 FIFO 接收发送到所有 OUT 端点的数据。只要 RxFIFO 中有空余空间，收到的数据包就挨个填入 RxFIFO。除了有效数据外，接收到的数据包状态（包含 OUT 端点目标编号、字节数、数据 PID 和对所接收数据的验证）也由模块进行存储。没有可用空间时，设备会回复主机事务 NAK 应答并在被寻址的端点上触发中断。接收 FIFO 的大小在接收 FIFO 大小寄存器 (GRXFSIZ) 中配置。

单个接收 FIFO 架构使得 USB 设备更高效地填充接收 RAM 缓冲区：

- 所有 OUT 端点共享同一个 RAM 缓冲区（共享 FIFO）
- 对于任意主机序列 OUT 令牌，USBFS 模块可将接收 FIFO 填充至限值

只要至少有一个数据包在 RxFIFO 中可供读取，应用程序就会一直接收 RxFIFO 非空

中断（USBFS_GINTSTS 中的 RXFNE 位）。应用程序从接收状态读取和出栈寄存器 (GRXSTSP) 中读取数据包信息，最后通过读取与端点相关的出栈地址从接收 FIFO 读出相应数据。

31.5.10.2 设备 TxFIFO

模块为各个 IN 端点提供了专用的 FIFO。应用程序通过非周期发送 FIFO 大小寄存器 (USBFS_DIEPTSIZ0) 为 IN 端点 0 配置 FIFO 大小；通过设备 IN 端点发送 FIFOx 寄存器 (DIEPTSIZx) 为 IN 端点 x 配置 FIFO 大小。

31.5.11 USBFS FIFO RAM 分配

31.5.11.1 主机模式

接收 FIFO RAM 分配

状态信息随各个接收数据包写入 FIFO。因此，必须至少为接收数据包分配（最大数据包大小 / 4）+ 1 的空间。如果使能了多个同步通道，则为接收连续数据包分配的空间必须至少为（最大数据包大小 / 4）的两倍+1。通常，推荐的空间为（最大数据包 / 4 + 1）的两倍，这样当上一个数据包向 CPU 传送时，USB 可同时接收后续的数据包。

传输完成状态信息和该端点收到的最后一个数据包会一起被写入 FIFO。所以必须为此分配一个位置。

以 DMA 模式运行时，每个主机通道的 DMA 地址寄存器将存储在 FIFO 中，因此需要在 FIFO 中为每个通道保留一个位置存放其地址寄存器。

发送 FIFO RAM 分配

主机非周期性发送 FIFO 所需的最小 RAM 为所支持的所有非周期性 OUT 通道上传输的最大数据包的大小。

通常，推荐的空间为最大数据包大小的两倍，这样当 USB 正在发送当前数据包的同时，AHB 可以往发送 FIFO 填入下一个数据包。

主机周期性发送 FIFO 所需的最小 RAM 为所支持的所有周期性 OUT 通道上传输的最大数据包的大小。如果至少有一个同步 OUT 端点，则空间必须至少为该通道中最大数据包大小的两倍。

以 DMA 模式运行时，每个主机通道的 DMA 地址寄存器将存储在 FIFO 中，因此需要在 FIFO 中为每个通道保留一个位置存放其地址寄存器。

31.5.11.2 设备模式

接收 FIFO RAM 分配

应用程序应为 SETUP 数据包分配 RAM：接收 FIFO 中必须保留 11 个位置以在控制端点上接收 SETUP 数据包。USBFS 模块不会向这些为 SETUP 数据包保留的位置写入任何其它数据。将会为全局 OUT NAK 分配一个位置。状态信息随各个接收数据包

写入 FIFO。因此，必须至少为接收数据包分配 $(\text{最大数据包大小}/4) + 1$ 的空间。如果使能了多个同步端点，则为接收连续数据包分配的空间必须至少为 $(\text{最大数据包大小}/4)$ 的两倍+1。通常，推荐的空间为 $(\text{最大数据包}/4 + 1)$ 的两倍，这样当上一个数据包向 CPU 传送时，USB 可同时接收后续的数据包。

传输完成状态信息和该端点收到的最后一个数据包会一起被推入 FIFO。通常情况下，推荐为每个 OUT 端点分配一个位置。

发送 FIFO RAM 分配

各个 IN 端点发送 FIFO 所需的最小 RAM 空间为该特定 IN 端点的最大数据包大小。

31.5.12 USBFS 系统性能

凭借大容量 RAM 缓冲区、高度可配置的 FIFO 大小、通过 AHB 压栈/出栈寄存器进行 32 位 FIFO 快速访问，尤其是高级 FIFO 控制机制可获得最佳 USB 和系统性能。

实际上，无论当前 USB 序列如何，USBFS 均可通过该机制高效填充可用的 RAM 空间。借助这些特性：

- 应用程序有足够的裕量来计算并校正 CPU 的负载，从而优化 CPU 带宽利用率：
- 应用程序可先积累大量发送数据，再通过 USB 发送出去
- 可带来足够的时间裕量，以从接收 FIFO 读取数据
 - USB 模块能够保持全速工作状态，也就是提供最大的全速带宽（尽量多的硬件自动运行，尽量少的软件参与）
- USB 模块可提前积累大量发送数据供其支配，从而可对 USB 数据发送进行自主管理
- 接收缓冲区中有大量空白空间，可通过 USB 中的数据自动填满

由于 USBFS 模块能够高效填充 1.25KB RAM 缓冲区且 1.25KB 发送/接收数据足以满足一个全速帧所能容纳的数据量，因此 USB 系统在一帧之内可以无需应用程序干预达到最大 USB 带宽。

31.5.13 USBFS 中断和事件

全局中断是软件需要处理的主要中断，全局中断的标志位可在 USBFS_GINTSTS 寄存器读取。

中断标志	描述	运行模式	内部事件源
WKUPINT	恢复/远程唤醒中断	主机或者设备	-
VBU SVINT	VBUS有效中断	设备	-
DISCINT	断开连接中断	主机	-
CIDSCHG	连接器ID线状态变化中断	主机或者设备	-
PTXFE	周期性Tx FIFO空中断	主机	-
HCINT	主机通道中断	主机	-
HPRTINT	主机端口中断	主机	-
DATAFSUSP	数据获取挂起	设备	-
IPXFR/INCOMPISOOUT	未完成周期性传输/未完成OUT同步传输	设备	-
IISOIXFR	未完成IN同步传输	设备	-
OEPINT	OUT端点中断	设备	-
IEPINT	IN端点中断	设备	-
EOPF	周期性帧结束中断	设备	-
ISOODRP	丢弃同步OUT数据包中断	设备	-
ENUMDNE	枚举完成	设备	-
USBRST	USB复位中断	设备	-
USBSUSP	USB挂起中断	设备	-
ESUSP	早期挂起中断	设备	-
GONAKEFF	全局OUT NAK有效中断	设备	-
GINAKEFF	全局非周期性IN NAK有效中断	设备	-
NPTXFE	非周期性Tx FIFO空中断	主机	-

RXFNE	RxFIFO非空中断	主机或者设备	-
SOF	帧起始中断	主机或者设备	是
MMIS	模式不匹配中断	主机或者设备	-

表 31-2 USBFS 中断事件表

31.6 USBFS 编程模型

31.6.1 USBFS 模块初始化

应用程序必须执行模块初始化序列。

模式决定方法请参考 31.5.2 USBFS 模式决定。

本节介绍了 USBFS 控制器上电后的初始化过程。无论是以主机模式还是设备模式工作，应用程序都必须遵循初始化序列。根据模块配置对所有模块全局寄存器进行初始化：

1. 在 USBFS_GAHBCFG 寄存器中编程以下字段：
 - 全局中断屏蔽位 GINTMSK = 1
 - RxFIFO 非空（USBFS_GINTSTS 中的 RXFNE 位）
 - 周期性 TxFIFO 空门限
2. 在 USBFS_GUSBCFG 寄存器中编程以下字段：
 - FS 超时校准字段
 - USB 周转时间字段
3. 软件必须取消对 USBFS_GINTMSK 寄存器中以下位的屏蔽：
 - 模式不匹配中断屏蔽
4. 通过读取 USBFS_GINTSTS 中的 CMOD 位，软件可确定 USBFS 控制器是在主机模式还是设备模式下工作。

31.6.2 USBFS 主机初始化

要将模块作为主机进行初始化，应用程序必须执行以下步骤：

1. 编程 USBFS_GINTMSK 寄存器中的 HPRTINT 取消屏蔽。
2. 编程 USBFS_HCFG 寄存器以选择全速主机。
3. 将 USBFS_HPRT 中的 PPWR 位编程为 1，给 USB 总线提供 VBUS。
4. 等待 USBFS_HPRT 中的 PCDET 中断。这表示某设备已连接到主机端口。
5. 将 USBFS_HPRT 中的 PRST 位编程为 1，在 USB 总线上发出复位信号。
6. 至少等待 10ms，以便完成复位过程。

7. 将 USBFS_HPRT 中的 PRST 位编程为 0。
 8. 等待 USBFS_HPRT 中的 PENCHNG 中断。
 9. 读取 USBFS_HPRT 中的 PSPD 位以获取枚举速度。
 10. 使用所选 PHY 时钟，相应地设置 HFIR 寄存器。
 11. 根据步骤 9 中检测到的设备速度编程 USBFS_HCFG 寄存器中的 FSLSPCS 字段。
如果 FSLSPCS 发生更改，则必须执行端口复位。
 12. 编程 USBFS_GRXFSIZ 寄存器以选择接收 FIFO 的大小。
 13. 编程 USBFS_HNPTXFSIZ 寄存器，以选择用于非周期性通信事务的非周期性发送 FIFO 的大小和起始地址。
 14. 编程 USBFS_HPTXFSIZ 寄存器，以选择用于周期性事务的周期性通信发送 FIFO 的大小和起始地址。
- 要与设备通信，系统软件必须初始化并使能至少一个通道。

31.6.3 USBFS 设备初始化

上电期间或者从主机模式切换为设备模式后，应用程序必须执行下列步骤来将模块作为设备进行初始化。

1. 在 USBFS_DCFG 寄存器中编程以下字段：
 - 设备速度
 - 非零长度状态 OUT 握手信号
2. 编程 USBFS_GINTMSK 寄存器以取消屏蔽以下中断：
 - USB 复位
 - 枚举完成
 - 早期挂起
 - USB 挂起
 - SOF
3. 等待 USBFS_GINTSTS 中的 VBUSVINT 中断，表示进入供电状态。
4. 等待 USBFS_GINTSTS 中的 USBRST 中断。这表示已在 USB 上检测到复位信号，复位过程自接收到此中断后约持续 10ms。

5. 等待 USBFS_GINTSTS 中的 ENUMDNE 中断。此中断指示 USB 上复位过程结束。
接收到此中断时，应用程序必须读取 USBFS_DSTS 寄存器以确定枚举速度并执行枚举完成时的端点初始化中所列的步骤。
此时，设备已准备好接受 SOF 数据包并在控制端点 0 上执行控制传输。

31.6.4 USBFS DMA 模式

USB 使用 AHB 主接口来获取发送数据包数据（AHB 到 USB）和接收数据更新（USB 到 AHB）。AHB 主接口使用经过编程的 DMA 地址（主机模式下的 HCDMAX 寄存器和设备模式下的 DIEPDMAx/DOEPDMAx 寄存器）来访问数据缓冲区。

31.6.5 USBFS 主机编程模型

31.6.5.1 通道初始化

应用程序必须初始化一个或多个通道，之后才能与所连接的设备通信。

要初始化和使能通道，应用程序必须执行以下步骤：

1. 编程 USBFS_GINTMSK 寄存器以取消对以下位的中断屏蔽：
 - 用于 OUT 事务的非周期性发送 FIFO 为空（在流水线事务级别工作且数据包计数字段编程值大于 1 时适用）。
 - 用于 OUT 事务的非周期性发送 FIFO 为半空（在流水线事务级别工作且数据包计数字段编程值大于 1 时适用）。
2. 编程 USBFS_HAINTMSK 寄存器以使能所选通道中断。
3. 编程 USBFS_HCINTMSK 寄存器，以使能主机通道中断寄存器中反映的和通信事务有关的中断。
4. 编程所选通道的 USBFS_HCCTSIZx 寄存器，指定以字节为单位的总传输大小和包括短数据包在内的预期数据包个数。应用程序必须使用初始数据 PID（用于第一个 OUT 事务或预期从第一个 IN 事务获取）编程 PID 字段。
5. 编程所选通道的 USBFS_HCCHARx 寄存器，指定设备的端点特性，例如类型、速度、方向等。（仅当应用程序准备好发送或接收数据包时，才能通过将通道使能位置 1 来使能通道）。

31.6.5.2 通道停止

应用程序可以通过编程 USBFS_HCCHARx 寄存器将 CHDIS 和 CHENA 位置 1 来禁止任何通道。这会使 USBFS 主机清空之前在该通道上发出的请求（如果有）并生成通道停止中断。应用程序在将通道重新分配给其它通信事务之前，必须等待 USBFS_HCINTx 中的 CHH 中断。USBFS 主机不会中断已在 USB 上启动的通信事务。禁止通道前，应用程序必须确保非周期性请求队列（禁止非周期性通道时）或周期性请求队列（禁止周期性通道时）中至少有一个空闲空间。应用程序可以在请求队列已满时（禁止通道之前），通过编程 USBFS_HCCHARx 寄存器将 CHDIS 位置 1 和将 CHENA 位清零，清空请求队列。出现以下任一情况时，应用程序将禁止通道：

1. IN 或 OUT 通道的 USBFS_HCINTx 中接收到 STALL、TXERR、BBERR 或 DTERR 中断。应用程序在接收到通道停止信号之前，必须能够接收相同通道的其它中断（DTERR、Nak、Data、TXERR）。
2. 接收到 USBFS_GINTSTS 中的 DISCINT（断开设备连接）中断。（应用程序将禁止所有已使能的通道）。
3. 应用程序在传输正常完成之前将其中止。

在 DMA 模式时，应用程序不能同过写寄存器停止不可分割的周期性传输。

31.6.6 USBFS 设备编程模型

31.6.6.1 USB 复位时的端点端点初始化

1. 为所有 OUT 端点将 NAK 位置 1
 - USBFS_DOEPCTLx 中，SNAK=1（对于所有 OUT 端点）
2. 取消对以下中断位的屏蔽
 - USBFS_DAINTMSK 中，INEP0=1（控制 0 IN 端点）
 - USBFS_DAINTMSK 中，OUTEP0=1（控制 0 OUT 端点）
 - DOEPMSK 中，STUP=1
 - DOEPMSK 中，XFRC=1
 - DIEPMSK 中，XFRC=1
 - DIEPMSK 中，TOC=1

3. 为每个 FIFO 设置数据 FIFO RAM
 - 对 `USBFS_GRXFSIZ` 寄存器进行编程，以能够接收控制传输的 OUT 数据和设置数据。该寄存器必须至少等于控制端点 0 的 1 个最大数据包大小+2 个字（用于控制 OUT 数据包的状态）+10 个字（用于 SETUP 数据包）。
 - 对 `USBFS_TX0FSIZ` 寄存器进行编程（取决于所选的 FIFO 编号），以能够发送控制 IN 数据。该寄存器至少必须等于控制端点 0 的 1 个最大数据包大小。
4. 对端点相关寄存器中的以下字段进行编程，以使控制 OUT 端点 0 接收 SETUP 数据包
 - `USBFS_DOEPTSIZ0` 中的 `STUPCNT=3`（接收最多 3 个连续的 SETUP 数据包）此时，接收 SETUP 数据包所需的所有初始化工作便已完成。

31.6.6.2 USB 复位时的端点初始化

1. 在枚举完成中断 (`USBFS_GINTSTS` 中的 `ENUMDNE`) 中，读取 `USBFS_DSTS` 寄存器以确定设备的枚举速度。
2. 对 `USBFS_DIEPCTL0` 中的 `MPSIZ` 字段进行编程以设置最大数据包大小。该步骤配置控制端点 0。控制端点的最大数据包大小取决于枚举速度。
此时，设备已准备好接收 SOF 数据包并配置为在控制端点 0 执行控制传输。

31.6.6.3 收到 SetAddress 命令时的端点初始化

本节介绍了应用程序在 SETUP 数据包中接收到 SetAddress 命令时必须执行的操作。

1. 使用在 SetAddress 命令中接收到的设备地址来对 `USBFS_DCFG` 寄存器进行编程
2. 对模块进行编程以发出状态阶段的 IN 数据包

31.6.6.4 收到 SetConfiguration/SetInterface 命令时的端点初始化

本节介绍了应用程序在 SETUP 包中接收 SetConfiguration 或 SetInterface 命令时必须执行的操作。

1. 接收到 SetConfiguration 命令时，应用程序必须对端点寄存器进行编程，以使用新配置中有效端点的特性来配置这些端点寄存器。
2. 接收到 SetInterface 命令时，应用程序必须对命令指定的端点的端点寄存器进行编程。

3. 在先前配置或其它设置中有效的端点在新的配置或其它设置中无效。必须停用这些无效端点。
4. 使用 USBFS_DAINTMSK 寄存器使能有效端点的中断，屏蔽无效端点的中断。
5. 为每个 FIFO 设置数据 FIFO RAM。
6. 配置完所有必需的端点后，应用程序必须对模块进行编程以发送状态阶段的 IN 数据包。

此时，设备模块已可以接收和发送任何类型的数据包

31.6.6.5 端点激活

本节介绍激活设备端点或者将现有设备端点配置为新类型所需的步骤。

1. 在 USBFS_DIEPCTLx 寄存器（对于 IN 或双向端点）或 USBFS_DOEPCTLx 寄存器（对于 OUT 或双向端点）的以下字段中，对所需端点的特性进行编程。
 - 最大数据包大小
 - USB 活动端点位置 1
 - 端点初始数据同步位（对于中断和批量端点）
 - 端点类型
 - TxFIFO 编号
2. 激活端点后，模块便开始解码发送到该端点的令牌，并在收到的令牌有效的情况下回复有效握手信号。

31.6.6.6 端点停用

本节介绍停用现有端点所需的步骤。

1. 在要停用的端点中，将 USBFS_DIEPCTLx 寄存器（对于 IN 或双向端点）或 USBFS_DOEPCTLx 寄存器（对于 OUT 或双向端点）中的 USB 活动端点位清零。
2. 停用端点后，模块便会忽略发送到该端点的令牌，从而导致 USB 超时。

31.6.7 USBFS 操作模型

31.6.7.1 SETUP 和 OUT 数据传输

本节介绍了数据 OUT 传输和 SETUP 事务期间的内部数据流和应用程序操作步骤。

数据包读取

本节介绍如何从接收 FIFO 读取数据包（OUT 数据和 SETUP 数据包）。

1. 捕获到 RXFNE 中断（USBFS_GINTSTS 寄存器）时，应用程序必须读取接收状态弹出寄存器（USBFS_GRXSTSP）。
2. 应用程序可以通过写入 RXFNE=0（在 USBFS_GINTMSK 中）来屏蔽 RXFNE 中断（在 USBFS_GINTSTS 中），直到它把数据包从接收 FIFO 中读取出来。
3. 如果已接收数据包的字节计数不是 0，则从接收数据 FIFO 中弹出这些数据并存储在存储器中。如果接收到的数据包字节计数为 0，则不会从接收数据 FIFO 中弹出任何数据。
4. 从接收 FIFO 读出的数据包状态有以下几种状态：
 - 全局 OUT NAK:
PKTSTS=全局 OUT NAK, BCNT=0x000, EPNUM 和 DPID 的值无关紧要。
这些数据表示全局 OUT NAK 位已生效。
 - SETUP 数据包:
PKTSTS=SETUP, BCNT=0x008, EPNUM=控制 EP 编号, DPID=D0。这些数据表示指定端点上收到的 SETUP 数据包现在可从接收 FIFO 中读取。
 - 建立阶段完成:
PKTSTS=建立阶段完成, BCNT=0x0, EPNUM=控制 EP 编号, DPID 值无关紧要。
这些数据表示指定端点的建立阶段完成并且数据阶段已启动。在此状态条目从接收 FIFO 中弹出后，模块将在该控制 OUT 端点上产生建立中断。
 - OUT 数据包:
PKTSTS=DataOUT, BCNT=接收的 OUT 数据包的大小 (BCNT:0~1024), EPNUM=收到数据包的端点编号, DPID=实际数据 PID。
 - 数据传输完成:

PKTSTS=OUT 数据传输完成，BCNT=0x0，EPNUM=完成数据传输的 OUT EP 编号，DPID 值无关紧要。

这些数据表示指定 OUT 端点的 OUT 数据传输完成。在此状态条目从接收 FIFO 中弹出后，模块将在指定的 OUT 端点上引发“传输完成”中断。

5. 从接收 FIFO 中弹出数据后，必须取消对 RXFNE 中断的屏蔽 (USBFS_GINTSTS)。
6. 每次应用程序检测到 USBFS_GINTSTS 中的 RXFNE 中断时，都将重复步骤 1 到 5。读取空的接收 FIFO 可能导致未定义的模块行为。

SETUP 事务

本节介绍了模块处理 SETUP 数据包的方式以及应用程序处理 SETUP 事务的顺序。

应用程序要求：

1. 要接收 SETUP 数据包，必须将控制 OUT 端点 STUPCNT 字段 (USBFS_DOEPTSIZ0) 编程为非零值。如果应用程序将 STUPCNT 字段编程为非零值，模块会接收 SETUP 数据包并将其写入接收 FIFO，而不考虑 NAK 状态和 USBFS_DOEPCTL0 中的 EPENA 位设置。控制端点每收到一个 SETUP 数据包后，STUPCNT 字段都会递减。如果在接收 SETUP 数据包之前，未将 STUPCNT 字段编程为适当值，模块仍能接收 SETUP 数据包并使 STUPCNT 字段递减，但应用程序可能无法确定在控制传输的建立阶段中接收的 SETUP 数据包正确数量。
 - 在 USBFS_DOEPTSIZ0 中，STUPCNT=3
2. 应用程序必须始终在接收数据 FIFO 中分配一些额外空间，以便能够在控制端点上接收连续的最多三个 SETUP 数据包。
 - 预留空间 10 个字。第一个 SETUP 数据包需要 3 个字，“建立阶段完成”状态双字需要 1 个字，还需要 6 个字以存储两个额外的 SETUP 数据包。
 - 每个 SETUP 数据包需要 3 个字以存储 8 个字节的 SETUP 数据和 4 个字节的 SETUP 状态。模块将在接收 FIFO 中保留这些空间。
 - 这段 FIFO 仅用于存储 SETUP 包，绝对不会将该空间用于数据包。
3. 应用程序必须从接收 FIFO 中读取 SETUP 数据包的 2 个字。
4. 应用程序必须从接收 FIFO 中读取并丢弃“建立阶段完成”状态字

内部数据流:

1. 接收到 SETUP 数据包时，模块会将接收到的数据写入接收 FIFO，而不会检查接收 FIFO 中的可用空间，且不考虑端点的 NAK 和 STALL 位设置。
 - 模块会在内部将接收到 SETUP 数据包的控制 IN/OUT 端点的 IN NAK 和 OUTNAK 位置 1。
2. USB 上接收到的每个 SETUP 数据包，模块会将 3 个字的数据写入接收 FIFO，并且将 STUPCNT 字段递减 1。
 - 第一个字包含由模块所使用的内部控制信息
 - 第二个字包含 SETUP 命令的前 4 个字节
 - 第三个字包含 SETUP 命令的最后 4 个字节
3. 当建立阶段结束，数据 IN/OUT 阶段开始时，模块会将一个状态条目（“建立阶段完成”字）写入接收 FIFO，指示建立阶段完成。
4. 在 AHB 端，SETUP 数据包被应用程序读取。
5. 当应用程序从接收 FIFO 中弹出“建立阶段完成”字时，模块将使用 STUP 中断 (USBFS_DOEPINTx) 来中断应用程序，指示其可以处理接收到的 SETUP 数据包。
 - 模块会将控制 OUT 端点的端点使能位清零。

应用程序编程顺序:

1. 对 USBFS_DOEPTSIZ0 寄存器进行编程。
 - STUPCNT=3
2. 等待 RXFNE 中断 (USBFS_GINTSTS) 并且从接收 FIFO 中读取数据包。
3. STUP 中断的触发(USBFS_DOEPINTx) 表示 SETUP 数据传输成功完成。
 - 发生该中断时，应用程序必须读取 USBFS_DOEPTSIZx 寄存器以确定接收的 SETUP 数据包数量并处理最后接收的 SETUP 数据包。

处理三个以上连续的 SETUP 数据包:

根据 USB2.0 规范，在 SETUP 数据包错误中，主机通常不会向同一个端点发送 3 个以上连续的 SETUP 数据包。但是，USB2.0 规范并未限制主机可以向同一个端点发送的连续 SETUP 数据包数量。发生这种情况时，USBFS 控制器将生成中断

(USBFS_DOEPIINTx 中的 B2BSTUP)。

将全局 OUT NAK 置 1

内部数据流：

1. 如果应用程序将全局 OUT NAK (USBFS_DCTL 中的 SGONAK 位) 置 1，模块将停止向接收 FIFO 中写入 SETUP 数据包以外的数据。无论接收 FIFO 中可用空间大小如何，设备都会对主机发送的非同步 OUT 令牌回复 NAK，而对同步 OUT 数据包直接予以忽略。
2. 模块将全局 OUT NAK 写入接收 FIFO。应用程序必须为此留出足够空间。
3. 当应用程序从接收 FIFO 中弹出全局 OUT NAK 字时，模块会将 GONAKEFF 中断 (USBFS_GINTSTS) 置 1。
4. 应用程序检测到该中断后，会认为模块处于全局 OUT NAK 模式。应用程序可以通过将 USBFS_DCTL 中的 SGONAK 位清零来清除该中断。

应用程序编程顺序：

1. 要停止接收任何类型的数据到接收 FIFO 中，应用程序必须通过编程以下字段以将全局 OUT NAK 位置 1。
 - 在 USBFS_DCTL 中，SGONAK=1
2. 等待 USBFS_GINTST 中的 GONAKEFF 中断。一旦被触发，该中断表示模块已停止接收 SETUP 数据包以外的任何类型数据。
 - 如果应用程序已将 USBFS_DCTL 中的 SGONAK 位置 1，则在模块引发 GONAKEFF 中断 (USBFS_GINTSTS) 之前，应用程序可以接收有效 OUT 数据包。
3. 应用程序可通过对 USBFS_GINTMSK 寄存器中的 GINAKEFFM 位执行写操作来暂时屏蔽此中断。
 - 在 USBFS_GINTMSK 寄存器中，GINAKEFFM=0
4. 当应用程序准备退出全局 OUT NAK 模式时，必须将 USBFS_DCTL 中的 SGONAK 位清零。此操作还会清除 GONAKEFF 中断 (USBFS_GINTSTS)。
 - 在 USBFS_DCTL 中，CGONAK=1
5. 如果应用程序在之前已屏蔽此中断，则必须按以下方式取消对该中断的屏蔽：

- 在 GINTMSK 中, GINAKEFFM=1

将全局 OUT NAK 置 1

应用程序必须使用以下顺序禁止已使能的 OUT 端点。

应用程序编程顺序:

1. 禁止任何 OUT 端点前, 应用程序必须在模块中使能全局 OUT NAK 模式。
 - 在 USBFS_DCTL 中, SGONAK=1
2. 等待 GONAKEYF 中断(USBFS_GINTSTS)
3. 通过编程以下字段来禁止 OUT 端点:
 - 在 USBFS_DOEPCTLx 中, EPDIS=1
 - 在 USBFS_DOEPCTLx 中, SNAK=1
4. 等待 EPDISD 中断 (USBFS_DOEPINTx), 该中断表示已完全禁止 OUT 端点。引发 EPDISD 中断时, 模块还会将以下位清零:
 - 在 USBFS_DOEPCTLx 中, EPDIS=0
 - 在 USBFS_DOEPCTLx 中, EPENA=0
5. 应用程序必须将全局 OUT NAK 位清零, 以开始从其它未禁止的 OUT 端点接收数据。
 - 在 USBFS_DCTL 中, SGONAK=0

通用非同步 OUT 数据传输

本节介绍一种常规非同步 OUT 数据传输 (控制、批量或中断)。

应用程序要求:

1. 建立 OUT 传输前, 应用程序必须在存储器中分配一个缓冲区, 以容纳要作为 OUT 传输的一部分而接收的所有数据。
2. 对于 OUT 传输, 端点的传输大小寄存器中的传输大小字段必须是端点的最大数据包大小的倍数 (且以字对齐)。
 - 传输大小[EPNUM] = $n \times (\text{MPSIZ}[EPNUM] + 4 - (\text{MPSIZ}[EPNUM] \bmod 4))$
 - 数据包计数[EPNUM] = n
 - $n > 0$

3. 发生 OUT 端点中断时，应用程序必须读取端点的传输大小寄存器以计算存储器中有效数据量。接收的有效数据量可能小于编程的传输大小。
 - 存储器中的有效数据量 = 应用程序设置的初始传输量 - 模块更新后的剩余传输量
 - 接收到 USB 数据包数 = 应用程序设置的初始数据包数 - 模块更新后的剩余数据包数

内部数据流：

1. 应用程序必须在端点相关寄存器中设置传输大小和数据包计数字段，将 NAK 位清零，并使能端点来接收数据。
2. NAK 位清零后，模块便开始接收数据并将数据写入接收 FIFO（只要接收 FIFO 中有空间）。对于 USB 上接收的每个数据包，数据包及其状态都会写入接收 FIFO。写入接收 FIFO 的每个数据包（数据量达到最大数据包大小的数据包或短数据包）都会使该端点的数据包计数字段递减 1。
 - 收到的数据包若 CRC 无效，则自动被从接收 FIFO 中清除。
 - 在 USB 上为数据包回复 ACK 后，模块将丢弃主机因无法检测到 ACK 而重新发送的非同步 OUT 数据包。应用程序不会在具有相同数据 PID 的相同端点上检测到多个连续的 OUT 数据包。在这种情况下，数据包计数不会递减。
 - 如果接收 FIFO 中没有空间，则会忽略同步或非同步数据包并且不会将它们写入接收 FIFO。此外，非同步 OUT 令牌将会收到 NAK 握手应答。
 - 在上述所有三种情况中，数据包计数都不会递减，因为没有任何数据写入接收 FIFO。
3. 当数据包计数变为 0 或者在端点上接收到短数据包时，该端点的 NAK 位将置 1。NAK 置位后，将忽略同步或非同步数据包并且不会将它们写入接收 FIFO，同时非同步 OUT 令牌会收到 NAK 握手应答。
4. 在数据写入接收 FIFO 后，应用程序将从接收 FIFO 中读取数据并将数据写入外部存储器，一次一个数据包，逐个端点过来。
5. 在 AHB 上向外部存储器写入完每个数据包后，端点的传输大小都会自动减去该数据包的大小。

6. 在以下情况时，OUT 端点的 OUT 数据传输完成状态将写入接收 FIFO：
 - 传输大小为 0 并且数据包计数为 0
 - 写入接收 FIFO 的最后一个 OUT 数据包是短数据包
(数据包大小:0~最大数据包大小-1)
7. 当应用程序弹出此状态条目(OUT 数据传输完成)，并生成该端点的传输完成中断，同时清零端点使能位。

应用程序编程顺序：

1. 使用传输大小和相应数据包个数对 USBFS_DOEPTSIZx 寄存器进行编程。
2. 使用端点特性对 USBFS_DOEPCTLx 寄存器进行编程，并将 EPENA 和 CNAK 位置 1。
 - 在 USBFS_DOEPCTLx 中，EPENA=1
 - 在 USBFS_DOEPCTLx 中，CNAK =1
3. 等待 RXFNE 中断（在 USBFS_GINTSTS 中）并且从接收 FIFO 中读走数据包。
 - 此步骤可重复多次，具体取决于传输大小。
4. 触发 XFRC 中断 (USBFS_DOEPINTx)，以表示非同步 OUT 数据传输成功完成。
5. 读取 USBFS_DOEPTSIZx 寄存器，以确定有效数据量。

通用同步 OUT 数据传输

本节介绍常规的同步 OUT 数据传输。

应用程序要求：

1. 非同步 OUT 数据传输的所有应用程序要求均适用于同步 OUT 数据传输。
2. 对于同步 OUT 数据传输中的传输大小和数据包计数字段，必须始终将其设置为单个帧中可接收的最大数据包大小的数据包数目。同步类型的 OUT 数据传输事务必须在一个帧内完成。
3. 在周期性帧结束 (USBFS_GINTSTS 中的 EOPF 中断) 之前，应用程序必须从接收 FIFO 中读取所有同步 OUT 数据包（数据条目和状态条目）。
4. 要接收下一帧中的数据，必须在 EOPF(USBFS_GINTSTS) 之后 SOF(USBFS_GINTSTS)之前使能一个同步 OUT 端点。

内部数据流:

1. 同步 OUT 端点的内部数据流与非同步 OUT 端点的基本相同，但稍有差异。
2. 同步 OUT 端点通过将端点使能位置 1 并将 NAK 位清零来使能时，必须相应地将偶数/奇数帧位置 1。仅当符合以下条件时，模块才会在同步 OUT 端点上接收特定帧中的数据：
 - EONUM（在 USBFS_DOEPCCTLx 中）= FNSOF [0]（在 USBFS_DSTS 中）
3. 当应用程序从接收 FIFO 中完整地读取一个同步 OUT 数据包（数据和状态）时，模块会根据从接收 FIFO 中读取的最后一个同步 OUT 数据包的数据 PID 更新 USBFS_DOEPTSIz 中的 RXDPID 字段。

应用程序编程顺序:

1. 使用传输大小和相应数据包计数对 USBFS_DOEPTSIz 寄存器进行编程
2. 使用端点特性对 USBFS_DOEPCCTLx 寄存器进行编程，并将端点使能位、清除 NAK 位和奇数/偶数帧位置 1。
 - EPENA=1
 - CNAK=1
 - EONUM= (0: 偶数/1: 奇数)
3. 等待 RXFNE 中断（在 USBFS_GINTSTS 中）并且从接收 FIFO 中读走数据包。
 - 此步骤可重复多次，具体取决于传输大小。
4. XFRC 中断（在 USBFS_DOEPINTx 中）表示同步 OUT 数据传输完成。该中断不一定意味着存储器中的数据是有效的。
5. 对于同步 OUT 传输，应用程序可能并不总会检测到该中断。相反，应用程序可能检测到 USBFS_GINTSTS 中的 IISOXXFRM 中断。
6. 读取 USBFS_DOEPTSIz 寄存器以确定接收的传输大小以及确定帧中接收的数据的有效性。仅当符合以下条件之一时，应用程序才必须将存储器中接收的数据视为有效数据：
 - RXDPID=D0（在 USBFS_DOEPTSIz 中）并且接收该有效数据的 USB 数据包数量=1

- RXDPID=D1 (在 USBFS_DOEPTSI_{Zx} 中) 并且接收该有效数据的 USB 数据包数量=2
- RXDPID=D2 (在 USBFS_DOEPTSI_{Zx} 中) 并且接收该有效数据的 USB 数据包数量=3

接收该有效数据的 USB 数据包数量=应用程序编程的初始数据包个数-模块更新后的剩余数据包个数。

应用程序可将无效数据包丢弃。

不完整的同步 OUT 数据传输

本节介绍了同步 OUT 数据包出现丢包时应用程序编程顺序。

内部数据流:

1. 对于同步 OUT 端点, 可能不会始终引发 XFRC 中断 (在 USBFS_DOEPINT_x 中)。如果模块丢弃同步 OUT 数据包, 则在以下情况下, 应用程序可能无法检测到 XFRC 中断 (USBFS_DOEPINT_x):
 - 在接收 FIFO 无法容纳完整的 ISO OUT 数据包时, 模块将丢弃接收到的 ISO OUT 数据
 - 接收到的同步 OUT 数据包存在 CRC 错误
 - 模块接收到的同步 OUT 令牌损坏
 - 应用程序从接收 FIFO 中读取数据的速度非常缓慢
2. 如果模块在所有同步 OUT 端点的传输完成前检测到周期性帧结束, 将触发未完成同步 OUT 数据中断 (USBFS_GINTSTS 中的 IISOXXFRM), 指示至少有一个同步 OUT 端点上未触发 XFRC 中断 (在 USBFS_DOEPINT_x 中)。此时, 未完成传输的端点仍保持使能, 但在 USB 的该端点上, 没有进行中的有效传输。

应用程序编程顺序:

1. 硬件触发 IISOXXFRM 中断 (USBFS_GINTSTS) 表示当前帧中至少有一个同步 OUT 端点具有未完成的传输。
2. 如果因未从端点完全读取同步 OUT 数据而发生这种情况, 应用程序必须确保首先从接收 FIFO 读取走所有同步 OUT 数据 (包括数据条目和状态条目), 然后再继续

处理。

- 从接收 FIFO 读取所有数据后，应用程序即可检测到 XFRC 中断 (USBFS_DOEPINTx)。在此情况下，应用程序必须重新使能端点以接收下一个帧中的同步 OUT 数据。
3. 当应用程序接收到 IISOOXFRM 中断（在 USBFS_GINTSTS 中）时，应用程序必须读取所有同步 OUT 端点的控制寄存器 (USBFS_DOEPCTLx)，以确定哪些端点在当前帧中具有不完整的传输。同时满足以下两个条件时，表示端点传输未完成：
 - EONUM 位（在 USBFS_DOEPCTLx 中）= FNSOF [0]（在 USBFS_DSTS 中）
 - EPENA=1（在 USBFS_DOEPCTLx 中）
 4. 在检测到 SOF 中断（在 USBFS_GINTSTS 中）前，必须执行完成上一步操作，以确保当前帧编号未发生更改。
 5. 对于具有不完整传输的同步 OUT 端点，应用程序必须丢弃存储器中的数据，并通过将 USBFS_DOEPCTLx 中的 EPDIS 位置 1 来禁止端点。
 6. 等待 EPDIS 中断（在 USBFS_DOEPINTx 中），并且使能端点以在下一帧中接收新数据。
 - 由于模块可能需要一些时间才能禁止端点，因此应用程序在接收到无效同步数据后，可能无法接收下一个帧中的数据。

停止非同步 OUT 端点

本节介绍应用程序如何才能停止非同步端点。

1. 将模块置于全局 OUT NAK 模式。
2. 禁止所需的端点
 - 禁止端点时，请设置 STALL=1（在 USBFS_DOEPCTL 中），而不是将 USBFS_DOEPCTL 中的 SNAK 位置 1。STALL 位的优先级始终高于 NAK 位。
3. 当应用程序不再需要端点回复 STALL 握手信号时，必须将 STALL 位（在 USBFS_DOEPCTLx 中）清零。
4. 如果应用程序由于收到主机的 SetFeature.Endpoint Halt 或 ClearFeature.Endpoint Halt 命令来设置或清除端点的 STALL 状态，则必须在该控制端点上的状态阶段传

输前，将 STALL 位置 1 或清零。

31.6.7.2 IN 数据传输

数据包写入

本节介绍在已使能专用发送 FIFO 的情况下应用程序如何将数据包写入端点 FIFO。

1. 应用程序可以选择轮询模式或中断模式。
 - 在轮询模式下，应用程序通过读取 USBFS_DTXFSTSx 寄存器来监视端点发送数据 FIFO 的状态，从而确定数据 FIFO 中是否有足够空间。
 - 在中断模式下，应用程序等待 TXFE 中断（在 USBFS_DIEPINTx 中），然后读取 USBFS_DTXFSTSx 寄存器以确定数据 FIFO 中是否有足够空间。
 - 要写入单个非零长度的数据包，数据 FIFO 中必须有足够的空间来容纳整个数据包。
 - 要写入零长度的数据包，应用程序不能查看 FIFO 空间。
2. 如果使用上述方法之一，当应用程序确定有足够的空间来写入发送数据包时，应用程序必须首先对端点控制寄存器进行相应写操作，然后再将数据写入数据 FIFO。通常，应用程序必须对 USBFS_DIEPCTLx 寄存器执行读-修改-写操作，以避免在将端点使能位置 1 的同时，修改寄存器中的其它内容。
如果有足够的空间，应用程序可将同一端点的多个数据包写入发送 FIFO。对于周期性 IN 端点，应用程序只能一次写入一个帧内的多个数据包。只有先前一个帧的通信事务传输完成之后，应用程序才会写入下一个帧内要发送的所有数据包。

将 IN 端点 NAK 置 1

内部数据流：

1. 当应用程序将特定端点的 IN NAK 置 1 时，模块将停止端点上的数据发送，而不考虑端点发送 FIFO 中的数据是否可用。
2. 非同步端点收到 IN 令牌，回复 NAK 握手应答。
 - 同步端点收到 IN 令牌，返回零长度数据包
3. 模块在 USBFS_DIEPINTx 中触发 INEPNE（IN 端点 NAK 有效）中断以响应 USBFS_DIEPCTLx 中的 SNAK 位。

4. 应用程序检测到该中断后，便会认为端点处于 IN NAK 模式。应用程序可通过将 USBFS_DIEPCTLx 中的 CNAK 位置 1 来清除该中断。

应用程序编程顺序：

1. 要在特定 IN 端点上停止发送任何数据，应用程序必须将 IN NAK 位置 1。要将该位置 1，必须编程以下字段。
 - USBFS_DIEPCTLx 中的 SNAK=1
2. 等待 USBFS_DIEPINTx 中的 INEPNE 中断触发。该中断表示模块已在端点上停止发送数据。
3. 在应用程序将 NAK 位置 1 但“NAK 有效”中断尚未触发时，模块可以在端点上发送有效 IN 数据。
4. 应用程序可通过写入 DIEPMSK 中的 INEPNEM 位来临时屏蔽该中断。
 - 在 DIEPMSK 中，INEPNEM = 0
5. 要退出端点 NAK 模式，应用程序必须将 USBFS_DIEPCTLx 中的 NAK 状态位 (NAKSTS) 清零。此操作还会清除 INEPNE 中断（在 USBFS_DIEPINTx 中）。
 - 在 USBFS_DIEPCTLx 中，CNAK=1
6. 如果应用程序已将该中断屏蔽，则必须按以下方式取消屏蔽：
 - 在 DIEPMSK 中，INEPNEM=1

禁止 IN 端点

使用以下顺序来禁止先前已使能的特定 IN 端点。

应用程序编程顺序：

1. 应用程序必须先停止在 AHB 上写入数据，之后才能禁止 IN 端点。
2. 应用程序必须将端点设置为 NAK 模式。
 - USBFS_DIEPCTLx 中的 SNAK=1
3. 等待 USBFS_DIEPINTx 中的 INEPNE 中断。
4. 将必须禁止的端点的 USBFS_DIEPCTLx 寄存器中的以下位置 1。
 - USBFS_DIEPCTLx 中的 EPDIS=1
 - USBFS_DIEPCTLx 中的 SNAK=1

5. USBFS_DIEPINTx 中的 EPDISD 中断的触发表示模块已完全禁止指定的端点。在触发中断的同时，模块还会将以下位清零：
 - 在 USBFS_DIEPCTLx 中，EPENA=0
 - 在 USBFS_DIEPCTLx 中，EPDIS=0
6. 应用程序必须为周期性 IN EP 读取 USBFS_DIEPTSIZx 寄存器，以计算端点上有多少数据是在 USB 上发送的。
7. 应用程序必须通过将 USBFS_GRSTCTL 寄存器中的以下字段置 1，来清空端点发送 FIFO 中的数据：
 - TXFNUM（在 USBFS_GRSTCTL 中）= 端点发送 FIFO 编号
 - TXFFLSH（在 USBFS_GRSTCTL 中）= 1

应用程序必须轮询 USBFS_GRSTCTL 寄存器，直至模块将 TXFFLSH 位清零，这表示 FIFO 清空操作结束。要在该端点上发送新数据，应用程序可以在稍后重新使能该端点。

通用非周期性 IN 数据传输

应用程序要求：

1. 建立 IN 传输前，应用程序必须确保组成一次 IN 传输的每个数据包都可以容纳在单个缓冲区中。
2. 对于 IN 传输，端点传输大小寄存器中的传输大小字段表示本次传输的有效数据量，它由多个最大数据包大小和单个短数据包组成。该短数据包在传输结束时发送。
 - 要发送多个最大数据包大小的数据包并在传输结束时外加一个短数据包：
$$\text{传输大小[EPNUM]} = x \times \text{MPSIZ[EPNUM]} + sp$$

如果 ($sp > 0$)，数据包计数[EPNUM] = $x + 1$ 。
否则，数据包计数[EPNUM] = x
 - 要发送单个零长度数据包：
$$\text{传输大小[EPNUM]} = 0$$
$$\text{数据包计数[EPNUM]} = 1$$
 - 要发送多个最大数据包大小的数据包并在传输结束时外加一个零长度数据包，应

应用程序必须将传输拆分为两个部分。

第一部分发送最大数据包大小的数据包，第二部分仅发送零长度数据包。

第一次传输：传输大小[EPNUM] = $x \times \text{MPSIZ}[\text{epnum}]$; 数据包计数 = n;

第二次传输：传输大小[EPNUM] = 0; 数据包计数 = 1;

3. 使能某个端点进行数据传输后，模块会更新传输大小寄存器。在 IN 传输结束时，应用程序必须读取传输大小寄存器，以确定送入发送 FIFO 中的数据已有多少通过 USB 发送出去。
4. 送入发送 FIFO 中的数据量 = 应用程序编程的初始传输大小 - 模块更新后的最终传输大小
 - 通过 USB 已经发送的数据量 = (应用程序编程的初始数据包计数 - 模块更新后的最终数据包计数) $\times \text{MPSIZ}[\text{EPNUM}]$
 - 要通过 USB 发送的剩余数据量 = (应用程序编程的初始传输大小 - 已通过 USB 发送的数据量)

内部数据流：

1. 应用程序必须在特定端点的寄存器中设置传输大小和数据包计数字段，并使能该端点来发送数据。
2. 应用程序还必须向该端点的发送 FIFO 写入必需的数据。
3. 应用程序每向发送 FIFO 写入一个数据包，该端点的传输大小便会自动减去该数据包大小。应用程序持续从存储器获取数据来写入发送 FIFO，直到该端点的传输大小变为 0。向 FIFO 写入数据后，“FIFO 中的数据包数”计数会递增（这是一个 3 位计数，由模块在内部进行维护，每个 IN 端点发送 FIFO 对应一个。在 IN 端点 FIFO 中，模块所维护的最大数据包数始终为八个）。对于零长度数据包，每个 FIFO 均另有一个单独的标志，FIFO 中没有任何数据。
4. 当数据写入发送 FIFO 后，模块会在接收到 IN 令牌时将这些数据送出。每个数据包发送出去并收到回复的 ACK 握手信号后，该端点的数据包计数都会递减 1，直到数据包计数变 0 为止。发生超时时，数据包计数不会递减。
5. 对于零长度数据包（由内部零长度标志指示），模块会针对 IN 令牌发出一个零长

度数据包，并递减数据包计数字段的值。

6. 如果接收到 IN 令牌的端点对应的 FIFO 中无数据，且该端点的数据包计数字段为零，则模块会针对该端点生成一个“TxFIFO 为空时接收到 IN 令牌”(ITTXFE)中断（前提是该端点的 NAK 位未置 1）。模块在该非同步端点上回复 NAK 握手信号。
7. 模块会在内部使 FIFO 指针重新返回到开头，并且不会生成超时中断。
8. 当传输大小为 0 且数据包计数为 0 时，将生成该端点的传输完成 (XFRC) 中断，同时将端点使能清零。

应用程序编程顺序：

1. 使用传输大小和相应数据包计数对 USBFS_DIEPTSIZx 寄存器进行编程。
2. 使用端点特性对 USBFS_DIEPCTLx 寄存器进行编程，并将 CNAK 和 EPENA（端点使能）位置 1。
3. 发送非零长度数据包时，应用程序必须轮询 USBFS_DTXFSTSx 寄存器（其中 x 为与该端点相关联的 FIFO 编号）以确定数据 FIFO 中是否有足够的空间。写入数据前，应用程序也可选用 TXFE 位（在 USBFS_DIEPINTx 中）。

通用周期性 IN 数据传输

本节介绍典型的周期性 IN 数据传输。

应用程序要求：

1. 通用非周期性 IN 数据传输的应用程序要求 1、2、3、4 对周期性 IN 数据传输同样适用（只是对要求 2 稍加修改）。
 - 应用程序只能发送若干个最大数据包大小的数据包或若干个最大数据包大小的包，外加传输结束时的一个短数据包。

要发送多个最大数据包大小的数据包并在传输结束时外加一个短数据包，必须满足以下条件：

$$\text{传输大小[EPNUM]} = x \times \text{MPSIZ[EPNUM]} + sp$$

（其中 x 是大于 0 的整数，且 sp 范围是 0 ~ MPSIZ[EPNUM]-1）

如果 ($sp > 0$)，数据包计数[EPNUM] = $x + 1$

否则，数据包计数[EPNUM] = x;

MCNT[EPNUM] = 数据包计数[EPNUM]

- 应用程序无法在传输结束时发送零长度数据包。应用程序可以单独发送一个零长度数据包。
- 要发送单个零长度数据包：

传输大小[EPNUM]=0

数据包计数[EPNUM]=1

MCNT[EPNUM]=数据包计数[EPNUM]

2. 应用程序一次只能安排一帧的数据传输。

- $(MCNT - 1) \times MPSIZ < XFERSIZ \leq MCNT \times MPSIZ$
- PKTCNT = MCNT (在 USBFS_DIEPTSIZx 中)
- 如果 $XFERSIZ < MCNT \times MPSIZ$, 则传输的最后一个数据包为短数据包
- 请注意: MCNT 位于 USBFS_DIEPTSIZx 中、MPSIZ 位于 USBFS_DIEPCTLx 中、PKTCNT 位于 USBFS_DIEPTSIZx 中、XFERSIZ 位于 USBFS_DIEPTSIZx 中

3. 接收到 IN 令牌前, 应用程序必须将要在帧中发送的完整数据写入到发送 FIFO 中。在接收到 IN 令牌时, 即使发送 FIFO 中该帧要发送的数据只差 1 个双字未写进来, 模块也会执行 FIFO 为空时的操作。当发送 FIFO 为空时:

- 同步端点上将回复零长度数据包
- 中断端点上将回复 NAK 握手信号

内部数据流:

1. 应用程序必须在特定端点的寄存器中设置传输大小和数据包计数字段, 并使能该端点来发送数据。
2. 应用程序还必须向与该端点相关联的发送 FIFO 写入必需的数据。
3. 应用程序每向发送 FIFO 写入一个数据包, 该端点的传输大小便会自动减去该数据包大小。应用程序持续从存储器获取数据来写入发送 FIFO, 直到该端点的传输大小变为 0。
4. 当周期性端点接收到 IN 令牌时, 模块将开始发送 FIFO 中的数据 (如果 FIFO 中

有数据)。如果 FIFO 中没有该帧要发送的数据的完整数据包，则模块将为该端点生成一个“TxFIFO 为空时接收到 IN 令牌”中断。

- 同步端点上将回复零长度数据包
 - 中断端点上将回复 NAK 握手信号
5. 端点的数据包计数会在下列情况下递减 1:
- 对于同步端点，发送一个零长度或非零长度的数据包时
 - 对于中断端点，在发送 ACK 握手信号时递减
 - 当传输大小和数据包计数均为 0 时，将生成该端点的传输完成中断，同时将端点使能位清零。
6. 在“周期性帧间隔”(由 USBFS_DCFG 中的 PFIVL 位控制)内，当模块发现任何在当前帧内应为空的同步 IN 端点 FIFO 中的数据还未发送完成时，都会在 USBFS_GINTSTS 中生成一个 IISOIXFR 中断。

应用程序编程顺序:

1. 使用端点特性对 USBFS_DIEPCTLx 寄存器进行编程，并将 CNAK 和 EPENA 位置 1。
2. 将需要在下一帧中发送的数据写入发送 FIFO。
3. 硬件触发 ITTXFE 中断(在 USBFS_DIEPINTx 中)表示应用程序尚未将需要发送的全部数据写入发送 FIFO。
4. 如果在检测到中断前已使能中断端点，则将忽略该中断。如果中断端点未使能，则使能此端点，以便数据能够在收到下一次 IN 令牌时发送出去。
5. 硬件触发 XFRC 中断(在 USBFS_DIEPINTx 中)时如果 USBFS_DIEPINTx 中未产生 ITTXFE 中断，则表示成功完成同步 IN 传输。读取 USBFS_DIEPTSIZx 寄存器时始终得到传输大小=0 且数据包计数=0，则表示所有数据都已通过 USB 发送完毕。
6. 置位 XFRC 中断(在 USBFS_DIEPINTx 中)时无论是否产生 ITTXFE 中断(在 USBFS_DIEPINTx 中)，都表示成功完成中断 IN 传输。读取 USBFS_DIEPTSIZx 寄存器时始终得到传输大小=0 且数据包计数=0，则表示所有数据都已通过 USB

发送完毕。

7. 在 USBFS_GINTSTS 中置位未完成的同步 IN 传输 (IISOIXFR) 中断时如果未产生任何前述中断，则表示在当前帧中模块至少未收到 1 个周期性的 IN 令牌。

未完成同步 IN 数据传输

本节介绍应用程序针对未完成同步 IN 数据传输必须执行的操作。

内部数据流：

1. 符合下列条件之一时，即认为同步 IN 传输未完成：
 - a) 模块在至少一个同步 IN 端点上接收到损坏的同步 IN 令牌。此时，应用程序检测到未完成同步 IN 传输中断（USBFS_GINTSTS 中的 IISOIXFR 位）。
 - b) 应用程序向发送 FIFO 写入数据的速度过慢，在将完整数据写入 FIFO 之前便接收到 IN 令牌。此时，应用程序在 USBFS_DIEPINTx 中检测到“Tx FIFO 为空时接收到 IN 令牌”中断。应用程序可忽略此中断，因为最终这将在周期性帧结束时产生一个未完成同步 IN 传输中断（USBFS_GINTSTS 中的 IISOIXFR 位）。模块会通过 USB 发送一个零长度数据包来响应接收到的 IN 令牌。
2. 应用程序必须尽快停止向发送 FIFO 写入数据。
3. 应用程序必须将端点的 NAK 位和禁止位置 1。
4. 模块会禁止该端点，将禁止位清零并触发端点的“端点禁止”中断。

应用程序编程顺序：

1. 应用程序可以在任何同步 IN 端点上忽略 USBFS_DIEPINTx 中的“Tx FIFO 为空时接收到 IN 令牌”中断，因为最终这将产生一个未完成同步 IN 传输中断（在 USBFS_GINTSTS 中）。
2. 硬件触发未完成同步 IN 传输中断（在 USBFS_GINTSTS 中）表示在至少一个同步 IN 端点上存在未完成的同步 IN 传输。
3. 应用程序必须读取所有同步 IN 端点的“端点控制”寄存器来检测存在未完成 IN 数据传输的端点。
4. 应用程序必须停止向与这些端点相关联的“周期性发送 FIFO”写入数据。

5. 对 USBFS_DIEPCTLx 寄存器中的下列字段进行编程以禁止端点：
 - USBFS_DIEPCTLx 中的 SNAK=1
 - USBFS_DIEPCTLx 中的 EPDIS=1
6. 硬件触发 USBFS_DIEPINTx 中的“端点禁止”中断表示模块已禁止该端点。
 - 此时，应用程序必须清空相关联的发送 FIFO 中的数据，或者通过在下一帧中使能新传输的端点来覆盖 FIFO 中的现有数据。要刷新数据，应用程序必须使用 USBFS_GRSTCTL 寄存器。

停止非同步 IN 端点

本节介绍应用程序如何才能停止非同步端点。

应用程序编程顺序：

1. 禁止要停止的 IN 端点。同时将 STALL 位置 1。
2. USBFS_DIEPCTLx 中的 EPDIS=1（当端点已使能时）
 - USBFS_DIEPCTLx 中的 STALL=1
 - STALL 位的优先级始终高于 NAK 位
3. 硬件触发“端点禁止”中断（在 USBFS_DIEPINTx 中）可以让应用程序知道模块已禁止指定端点。
4. 应用程序必须根据端点类型清空非周期性或周期性发送 FIFO。对于非周期性端点，应用程序必须重新使能另一个无需停止的非周期性端点来发送数据。
5. 当应用程序准备好结束该端点的 STALL 握手信号时，必须将 USBFS_DIEPCTLx 的 STALL 位清零。
6. 如果应用程序因收到来自主机的 SetFeature.Endpoint.Halt 命令或 ClearFeature.Endpoint.Halt 命令来设置或清除端点的 STALL 状态，则必须在该控制端点的状态阶段传输之前将 STALL 位置 1 或清零。

特例：停止控制 OUT 端点

如果在控制传输的数据阶段，主机发送的 IN/OUT 令牌数超过 SETUP 数据包指定的值，则模块必须对这些多余的 IN/OUT 令牌回复 STALL。在这种情况下，应用程序必须在控制传输的数据阶段使能 USBFS_DIEPINTx 的 ITTXFE 中断和

USBFS_DOEPINTx 的 OTEPDIS 中断（当模块已完成传输 SETUP 数据包指定的数据量后）。随后，当应用程序收到此中断时，必须将相应端点控制寄存器中的 STALL 位置 1 并清除此中断。

31.7 寄存器说明

应用程序通过 AHB 从接口对控制和状态寄存器进行读写操作，以此来控制 USBFS 模块。USBFS 模块所有寄存器为 32 位寄存器，其地址按 32 位对齐，因此只能以 32 位的方式访问。

控制和状态寄存器分为以下几类：

- USBFS 系统控制寄存器
- 模块全局寄存器
- 主机模式寄存器
- 设备模式寄存器
- 电源和时钟门控控制寄存器

其中 USBFS 系统控制寄存器与其它寄存器基址不同，该寄存器独立于 USBFS 模块以控制 USBFS 模块的相关设定。

只有模块全局寄存器，电源和时钟门控控制寄存器可以在主机和设备模式下进行访问。

当 USBFS 模块处以一种模式（主机或者设备）下，应用程序不得以另外一种角色模式访问寄存器，比如主机模式时，访问设备模式的寄存器。如果发生了非法访问，将会产生模式不匹配中断并在模块中断寄存器 USBFS.GINTSTS.NMIS 位反映。当模块从一种角色模式切换到另一种角色模式时，新工作模式下的寄存器必须重新编程为上电复位后的状态。

USBFS 模块寄存器列表以及基址请参考表 31-3~表 31-4 USBFS 寄存器一览表。

USBFS 系统控制寄存器基址：0x40055400

(USBFS系统控制寄存器) 寄存器名	偏移地址	复位值
USBFS系统控制寄存(USBFS_SYCTLREG)	0x00	0x0000 0000

表 31-3 USBFS 系统控制寄存器一览表

USBFS 模块寄存器基地址: 0x400C0000

(USBFS全局寄存器) 寄存器名	偏移地址	复位值
USBFS_VBUS控制寄存器(USBFS_GVBUSCFG)	0x00	0x0000 0000
USBFS_AHB控制寄存器(USBFS_GAHBCFG)	0x08	0x0000 0000
USBFS_USB配置寄存器(USBFS_GUSBCFG)	0x0c	0x0000 0A00
USBFS 复位寄存器(USBFS_GRSTCTL)	0x10	0x8000 0000
USBFS 全局中断寄存器(USBFS_GINTSTS)	0x14	0x1400 0020
USBFS 全局中断屏蔽寄存器(USBFS_GINTMSK)	0x18	0x0000 0000
USBFS 接收状态调试读取寄存器(USBFS_GRXSTSR)	0x1c	0x0000 0000
USBFS 接收状态读取和出栈寄存器(USBFS_GRXSTSP)	0x20	0x0000 0000
USBFS 接收FIFO大小寄存器(USBFS_GRXFSIZ)	0x24	0x0000 0140
USBFS 主机非周期性发送FIFO大小寄存器 (USBFS_HNPTXFSIZ)/端点0发送FIFO大小寄存器 (USBFS_DIEPRXF0)	0x28	0x0200 0140
USBFS 非周期性发送FIFO/队列状态寄存器 (USBFS_HNPTXSTS)	0x2c	0x0008 0100
USBFS 模块ID寄存器(USBFS_CID)	0x3c	0x1234 5678
USBFS 主机周期性发送FIFO大小寄存器(USBFS_HPTXFSIZ)	0x100	0x0000 0000
USBFS 设备IN端点x发送FIFO大小寄存器(USBFS_DIEPTXFx)	0x100+x*4(x=1~5)	0x0100 0240+(x-1)*0x100

(USBFS主机控制和状态寄存器) 寄存器名	偏移地址	复位值
USBFS 主机配置寄存器(USBFS_HCFG)	0x400	0x0000 0000
USBFS 主机帧时间间隔寄存器(USBFS_HFIR)	0x404	0x0000 EA60
USBFS 主机帧编号/帧剩余时间间隔寄存器 (USBFS_HFNUM)	0x408	0x0000 3FFF
USBFS 主机周期性发送FIFO/队列状态寄存器 (USBFS_HPTXSTS)	0x410	0x0008 0100
USBFS 主机全体通道中断寄存器(USBFS_HAINT)	0x414	0x0000 0000
USBFS 主机全体通道中断屏蔽寄存器 (USBFS_HINTMSK)	0x418	0x0000 0000
USBFS 主机端口控制和状态寄存器(USBFS_HPRT)	0x440	0x0000 0000
USBFS 主机通道x特性寄存器(USBFS_HCCHARx)	0x500+x*0x20(x=0~11)	0x0000 0000
USBFS 主机通道x中断寄存器(USBFS_HCINTx)	0x508+x*0x20(x=0~11)	0x0000 0000
USBFS 主机通道x中断屏蔽寄存器(USBFS_HCINTx)	0x50c+x*0x20(x=0~11)	0x0000 0000
USBFS 主机通道x传输大小寄存器(USBFS_HCTSIZx)	0x510+x*0x20(x=0~11)	0x0000 0000
USBFS 主机通道x DMA地址寄存器(USBFS_HCDMAX)	0x514+x*0x20(x=0~11)	0xXXXX XXXX

(USBFS设备控制和状态寄存器) 寄存器名	偏移地址	复位值
USBFS 设备配置寄存器(USBFS_DCFG)	0x800	0x0820 0000
USBFS 设备控制寄存器(USBFS_DCTL)	0x804	0x0000 0000
USBFS 设备状态寄存器(USBFS_DSTS)	0x808	0x0000 0002
USBFS 设备IN端点通用中断屏蔽寄存 (USBFS_DIEPMSK)	0x810	0x1400 0000
USBFS 设备OUT端点通用中断屏蔽寄存器 (USBFS_DOEPMSK)	0x814	0x0000 0000
USBFS 设备全体端点中断寄存器(USBFS_DAINT)	0x818	0x0000 0000
USBFS 设备全体端点中断屏蔽寄存(USBFS_DAINTMSK)	0x81c	0x0000 0000
USBFS 设备IN端点FIFO空中断屏蔽寄存器 (USBFS_DIEPEMPMSK)	0x834	0x0000 0000
USBFS 设备IN端点0控制寄存器(USBFS_DIEPCTL0)	0x900	0x0000 8000
USBFS 设备IN端点x控制寄存器(USBFS_DIEPCTLx)	0x900+x*0x20(n=1~5)	0x0000 0000
USBFS 设备IN端点x中断寄存器(USBFS_DIEPINTx)	0x908+x*0x20(n=0~5)	0x0000 0000
USBFS 设备IN端点x传输大小寄存器(USBFS_DIEPSIZx)	0x910+x*0x20(n=0~5)	0x0000 0000
USBFS 设备IN端点x DMA地址寄存(USBFS_DIEPDMAx)	0x914+x*0x20(n=0~5)	0x0000 0000
USBFS 设备IN端点x发送FIFO状态寄存器 (USBFS_DTXFSTSx)	0x918+x*0x20(n=0~5)	0x0000 0000
USBFS 设备OUT端点0控制寄存器(USBFS_DOEPCTL0)	0xb00	0x0000 8000
USBFS 设备OUT端点x控制寄存器(USBFS_DOEPCTLx)	0xb00+x*0x20(n=1~5)	0x0000 0000
USBFS 设备OUT端点x中断寄存器(USBFS_DOEPINTx)	0xb08+x*0x20(n=0~5)	0x0000 0000
USBFS 设备OUT端点x传输大小寄存(USBFS_DOEPSIZx)	0xb10+x*0x20(n=0~5)	0x0000 0000
USBFS 设备OUT端点x DMA地址寄存器 (USBFS_DOEPDMAx)	0xb14+x*0x20(n=0~5)	0xXXXXXXXX

(USBFS电源和时钟没空控制寄存器) 寄存器名	偏移地址	复位值
USBFS电源和门控时钟控制寄存器(USBFS_PCGCCTL)	0xe00	0x0000 0000

表 31-4 USBFS 系统控制寄存器一览表

31.7.1 USBFS 系统控制寄存器

31.7.1.1 USBFS 系统控制寄存器(USBFS_SYCTLREG)

USBFS System Control Register

偏移地址: 0x00

复位值: 0x0000 0000

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

位	标记	位名	功能	读写
b31~b2	Reserved	-	必须保持复位值。	R
b1	SOFEN	SOF脉冲输出使能位	主机发出SOF或者设备成功接收到SOF时，16个系统时钟周期宽度的SOF脉冲 从PAD输出使能 0: SOF脉冲不输出 1: SOF脉冲输出 注意：在设备模式和主机模式均可访问。	R/W
b0	DFB	VBUS/ID管脚内部去抖滤波器旁路使能位	VBUS/ID管脚模块内部去抖动滤波器旁路使能位 0: 模块内部去抖动滤波器有效 1: 旁路模块内部去抖动滤波器 注意：在设备模式和主机模式均可访问。	R/W

31.7.2 USBFS 全局寄存器

这些寄存器在主机模式和设备模式下都可用，且在这两个模式间切换时无需对其进行重新编程。除非特别说明，否则寄存器描述中的位值以二进制表示。

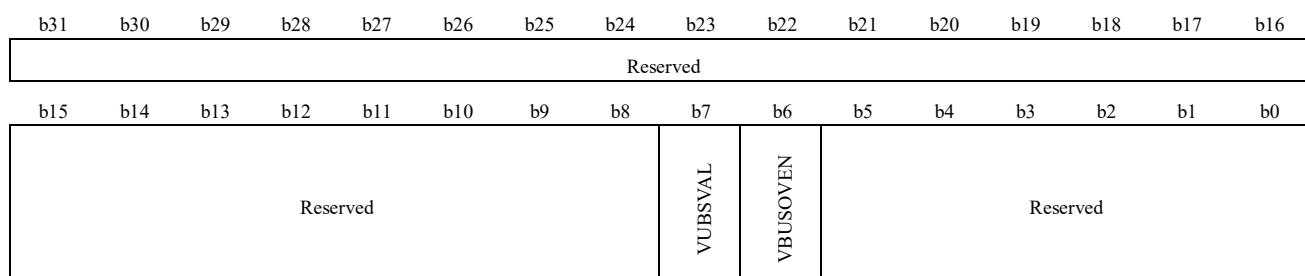
31.7.2.1 USBFS VBUS 控制寄存器(USBFS_GVBUSCFG)

VBUS Configuration Register

偏移地址：0x00

复位值：0x0000 0000

该寄存器可用于设定 VBUS 值从而忽略 VBUS 管脚的状态。



位	标记	位名	功能	读写
b31~b8	Reserved	-	必须保持复位值。	R
b7	VBUSVAL	VBUS值	VBUS值(VBUS Value) 用于设定USBFS的VBUS值，当设定为1，且VBUSOVEN设定1后对USBFS完成上电。 注意：仅可在设备模式下访问。	R/W
b6	VBUSOVEN	VBUS Override 使能	VBUS Override 使能(VBUS Override 使能) 用于将VBUSVAL设定的值反映到USBFS CORE的状态。仅当该位设置为1，VBUSVAL的值才有效。 注意：仅可在设备模式下访问。	R/W
b5~b0	Reserved	-	必须保持复位值。	R

31.7.2.2 USBFS AHB 控制寄存器(USBFS_GAHBCFG)

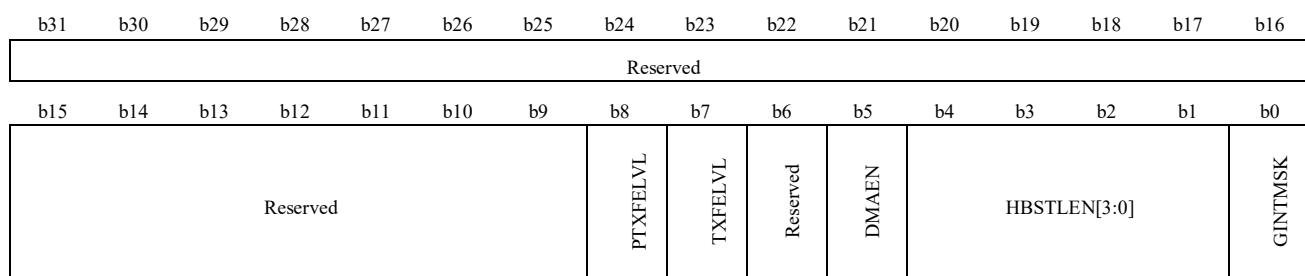
AHB Configuration Register

偏移地址: 0x08

复位值: 0x0000 0000

该寄存器可用于在上电后或更改角色模式时对模块进行配置。该寄存器主要包含 AHB 系统相关的配置参数。

应用程序必须在开始任何 AHB 或 USB 事务前对该寄存器进行编程。请勿在初始编程后更改该寄存器。



位	标记	位名	功能	读写
b31~b9	Reserved	-	必须保持复位值。	R
b8	PTXFELVL	周期性Tx FIFO空门限	周期性Tx FIFO空门限 (Periodic Tx FIFO empty level) 指示何时触发模块中断寄存器中的周期性Tx FIFO空中断位 (USBFS_GINTSTS 中的PTXFE 位)。 0: PTXFE (位于USBFS_GINTSTS) 中断指示周期性Tx FIFO为半空状态 1: PTXFE (位于USBFS_GINTSTS) 中断指示周期性Tx FIFO为全空状态 注意: 仅可在主机模式下访问。	R/W
b7	TXFELVL	设备Tx FIFO空门限	设备Tx FIFO空门限 (Tx FIFO empty level) 在设备模式下, 该位指示何时触发IN端点发送FIFO空中断 (USBFS_DIEPINTx 中的TXFE)。 0: TXFE (位于USBFS_DIEPINTx) 中断指示IN端点Tx FIFO为半空状态 1: TXFE (位于USBFS_DIEPINTx) 中断指示IN端点Tx FIFO为全空状态 注意: 仅可在主机模式下访问。	R/W
b6	Reserved	-	必须保持复位值。	R
b5	DMAEN	DMA 使能	DMA 使能 (DMA enable) 0: 模块以从模式运行 1: 模块以DMA模式运行 注意: 在设备模式和主机模式均可访问。	R/W

b4~b1	HBSTLEN	批量长度/类型	批量长度/类型 (Burst length/type)	R
			0000b:单次	
			0001b: INCR	
			0011b:INCR4	
			0101b:INCR8	
			0111b:INCR16	
			其它值: 保留	
			注意: 在设备模式和主机模式均可访问。	
b0	GINTMSK	全局中断屏蔽	全局中断屏蔽 (Global interrupt mask)	R/W
			该位用于屏蔽全局中断或对全局中断取消屏蔽。中断状态寄存器由模块进行更新, 与此位的设置无关。	
			0: 屏蔽应用程序触发的中断	
			1: 取消对应用程序触发的中断的屏蔽	
			注意: 在设备模式和主机模式均可访问。	

31.7.2.3 USBFS USB 配置寄存器 (USBFS_GUSBCFG)

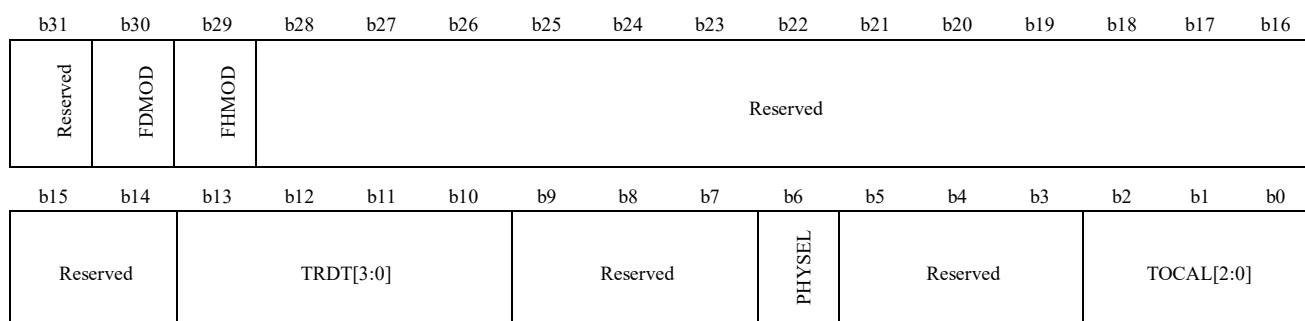
USBFS USB configuration register

偏移地址: 0x00C

复位值: 0x0000 0A00

该寄存器可用于在上电或更改角色模式后对模块进行配置。其中包含与 USB 和 USB-PHY 相关的配置参数。

应用程序必须在开始任何 AHB 或 USB 事务前对该寄存器进行编程。请勿在初始编程后更改该寄存器。



位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R
b30	FDMOD	强制设备模式	强制设备模式 (Force device mode) 向该位写入1时，可将模块强制为设备模式，忽略USBFS_ID输入引脚的状态。 0: 正常模式，取决USBFS_ID管脚的输入状态 1: 强制设备模式 将强制位置1后，应用程序必须等待至少25 ms后更改方可生效。 注意：在设备模式和主机模式均可访问。	R/W
b29	FHMOD	强制主机模式	强制主机模式 (Force host mode) 向该位写入1时，可将模块强制为主机模式，忽略USBFS_ID输入引脚的状态。 0: 正常模式，取决USBFS_ID管脚的输入状态 1: 强制主机模式 将强制位置1后，应用程序必须等待至少25 ms后更改方可生效。 注意：在设备模式和主机模式均可访问。	R/W
b28~b14	Reserved	-	必须保持复位值。	R

b13~b10	TRDT	USB周转时间	USB周转时间 (USB turnaround time) 以PHY时钟数为单位设置周转时间。 要计算TRDT的值, 请使用如下公式: $TRDT = 4 \times AHB\text{ 时钟} + 1\text{ 个PHY时钟}$ 例如: 1. 如果AHB时钟频率 = 84 MHz (PHY时钟频率 = 48 MHz), 则TRDT设置为9。 2. 如果AHB时钟频率 = 48 MHz (PHY时钟频率 = 48 MHz), 则 TRDT设置为 5。 注意: 在设备模式和主机模式均可访问。	R/W
b9~b7	Reserved	-	必须保持复位值。	R
b6	PHYSEL	全速系列收发器选择	全速系列收发器选择 (Full Speed serial transceiver select) 此位为只写位且始终为1。	W
b5~b3	Reserved	-	必须保持复位值。	R
b2~b0	TOCAL	FS 超时校准	FS 超时校准 (FS timeout calibration) PHY引入的额外延迟包括应用程序在该字段中设置的PHY时钟数, 以及模块的全速数据包间超时间隔。不同PHY引入的延迟对数据线状态的影响是不同的。全速操作的USB标准超时值为16到18 (含) 个位时间。应用程序必须根据枚举速度编程该字段。每个PHY时钟增加的位时间数为0.25个位时间。 注意: 在设备模式和主机模式均可访问。	R/W

31.7.2.4 USBFS 复位寄存器 (USBFS_GRSTCTL)

USBFS reset register

偏移地址: 0x10

复位值: 0x8000 0000

应用程序通过此寄存器复位模块中的各项硬件特性。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
AHBIDL	DMAREQ														Reserved
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
															Reserved TXFNUM[4:0] TXFFLSH RXFFLSH Reserved FCRST HSRST CSRSR

位	标记	位名	功能	读写
b31	AHBIDL	AHB主器件	AHB主器件空闲(AHB master idle) 空闲 指示AHB主器件状态机处于空闲情况。 注意：在设备模式和主机模式均可访问。	R
b30	DMAREQ	AHB主器件	DMA请求信号 (DMA request signal) 空闲 该位指示DMA请求正在进行中。用于调试。 注意：在设备模式和主机模式均可访问。	R
b29~b11	Reserved	-	读出时为“0”，写入时写“0”	R
b10~b6	TXFNUM	TxFIFO编 号	TxFIFO编号(TxFIFO number) 使用TxFIFO刷新位进行FIFO刷新的FIFO编号。只有在模块将TxFIFO刷新位清零后，方可更改此 字段。 ● 00000: — 主机模式下刷新非周期性TxFIFO — 设备模式下刷新TxFIFO 0 ● 00001: — 主机模式下刷新周期性TxFIFO — 设备模式下刷新TxFIFO 1 ● 00010: 设备模式下刷新TxFIFO 2 ... ● 00101: 设备模式下刷新TxFIFO 15 ● 10000: 在设备模式或主机模式下刷新所有的发送FIFO 注意：在设备模式和主机模式均可访问。	R/W

b5	TXFFLSH	TxFIFO刷 新	TxFIFO刷新 (TxFIFO flush) 此位选择性地刷新一个或所有的发送FIFO，但当模块处理通信事务时无法执行该操作。 只有在确认模块当前未对TxFIFO 执行读写操作后，应用程序方可对此位执行写操作。使用以下寄存器进行确认： — 读：NAK有效中断可确保模块当前未对FIFO执行读操作 — 写：USBFS_GRSTCTL中的 AHBIDL位可确保模块当前未对FIFO执行任何写操作 注意：在设备模式和主机模式均可访问。	R/W
b4	RXFFLSH	RxFIFO刷 新	RxFIFO刷新(RxFIFO flush) 应用程序可使用此位刷新整个RxFIFO，但必须首先确保模块当前未在处理通信事务。只有在确认模块当前未对RxFIFO执行读写操作后，应用程序方可对此位执行写操作。 应用程序必须等到此位清零后，方可执行其它操作。通常需要等待8个时钟周期（以PHY或AHB时钟中最慢的为准）。 注意：在设备模式和主机模式均可访问。	R/W
b3	Reserved	-	必须保持复位值。	R
b2	FCRST	主机帧计数 器复位	主机帧计数器复位(Host frame counter reset) 应用程序对该位执行写操作时，模块中的帧数计数器复位。帧计数器复位后，由模块发送的下一个SOF的帧号为 0。 注意：在设备模式和主机模式均可访问。	R/W
b1	HSRST	HCLK域逻 辑软复位	HCLK域逻辑软复位 (HCLK soft reset) 应用程序使用此位来刷新 AHB 时钟域中的控制逻辑。仅复位 AHB 时钟域流水线。 FIFO 不通过此位来刷新。 遵照协议终止 AHB 上的事务后，AHB 时钟域中的所有状态机均复位至空闲状态。 AHB 时钟域状态机所使用的 CSR 控制位清零。 要清零该中断，需要将由 AHB 时钟域状态机生成并用于控制中断状态的状态屏蔽位清零。 由于中断状态位并未清零，因此应用程序可以获取在该位置 1 后所发生的所有模块事件的状态。 此位为自清零位，模块将在其中所有必要逻辑复位后将该位清零。该过程需要若干个时钟的时间，具体取决于模块的当前状态。 注意：在设备模式和主机模式均可访问。	R/W

b0	CSRST	模块软复位	模块软复位 (Core soft reset)	R/W
按如下所述将HCLK和PCLK域复位：				
除以下各位外，将各个中断和所有CSR寄存器位清零：				
— USBFS_PCGCCTL中的RSTPDMODL位				
— USBFS_PCGCCTL中的GAYEHCLK位				
— USBFS_PCGCCTL中的PWRCLMP位				
— USBFS_PCGCCTL中的STPPCLK位				
— USBFS_HCFG 中的FSLSPCS位				
— USBFS_DCFG 中的DSPD位				
将所有模块状态机（AHB从器件除外）复位至空闲状态，并清空所有发送FIFO和接收FIFO。				
在AHB传输的最后数据阶段结束后，尽快终止AHB主器件上的所有事务。立即终止USB上的所有事务。				
应用程序可在需要复位模块时随时对该位执行写操作。该位为自清零位，模块将在其中所有必要逻辑复位后将该位清零，该过程需要若干个时钟的时间，具体取决于模块的当前状态。该位一旦清零，软件必须等待至少3个PHY时钟后才可以访问PHY域（同步延迟）。此外，软件还必须在确定该寄存器中的位31置 1（AHB主器件空闲）后方可开始运行。				
软件复位通常在两种情况下使用，一是软件开发期间，二是用户动态更改以上所列USB配置寄存器中的PHY选择位后。用户更改PHY时，将为PHY选择相应的时钟并用于PHY域中。一旦选择了新的时钟，则必须复位PHY域，才能保证正常运行。				
注意：在设备模式和主机模式均可访问。				

31.7.2.5 USBFS 全局中断状态寄存器 (USBFS_GINTSTS)

USBFS interrupt status register

偏移地址: 0x14

复位值: 0x14000020

该寄存器用于在当前模式（设备模式或主机模式）下借助系统级别的事件来中断应用程序。

该寄存器中的某些位仅在主机模式下有效，而其它位则仅在设备模式下有效。此外，该寄存器还可指示当前模式。

FIFO 状态中断为只读；如果软件在处理这些中断期间对 FIFO 执行读写操作，则 FIFO 中断标志将自动清零。

使能中断位前，应用程序必须在初始化时将 USBFS_GINTSTS 寄存器清零，才可以避免在初始化前产生任何中断。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
WKUINT	VBUCSVINT	DISCINT	CIDSCHG	Reserved	PTXFE	HCINT	HPRINTINT	Reserved	DATAFSUSP	IPXFR/ INCOMPISOUT	IISOIXFR	OEPINT	IEPINT	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EOPF	ISOODRP	ENMDNE	USBRST	USBSSUSP	ESUSP	Reserved	Reserved	GONAKEFF	GINAKEFF	NPTXFE	RXFNE	SOF	Reserved	MMIS	CMOD

位	标记	位名	功能	读写
b31	WKUINT	检测到恢复/远程唤醒中断	检测到恢复/远程唤醒中断 (Resume/remote wakeup detected interrupt) 在设备模式下，当USB总线上检测到恢复信号时，将触发该中断。 在主机模式下，当USB上检测到远程唤醒时，将触发该中断。 通过软件对该位写1清零。 注意：在设备模式和主机模式均可访问。	R/W
b30	VBUCSVINT	VBUS有效中断	VBUSy有效中断 (VBUS valid interrupt) 设备模式下，当检测到USBFS_VBUS管脚由低变高时，将触发该中断。 通过软件对该位写1清零。 注意：仅可在设备模式下访问。	R/W

b29	DISCINT	检测到断开连接中断 当检测到设备断开连接时触发该中断。 通过软件对该位写1清零。 注意：仅可在主机模式下访问。	R/W
b28	CIDSCHG	连接器ID线状态变化中断 当连接器ID线状态发生更改时，模块将该位置1。 通过软件对该位写1清零。 注意：在设备模式和主机模式均可访问。	R/W
b27	Reserved	- 必须保持复位值。	R
b26	PTXFE	周期性TxFIFO空中断 当周期性发送FIFO为半空或全空状态，且周期性请求队列中存在可写入至少一个条目的空间时，将触发该中断。该FIFO为半空状态还是全空状态由USBFS_GAHBCFG寄存器中的周期性TxFIFO空级别位（USBFS_GAHBCFG中的PTXFELVL位）决定。 注意：仅可在主机模式下访问。	R
b25	HCINT	主机通道中断 模块将该位置1时，指示模块中一个通道上存在挂起的中断（在主机模式下）。应用程序必须读取主机USBFS_HAINT寄存器，以确定发生中断的通道的准确编号，然后读取相应的USBFS_HCINTx寄存器，以确定引发中断的确切原因。应用程序必须先将USBFS_HCINTx寄存器的相应状态位清零，之后才能将该位清零。 注意：仅可在主机模式下访问。	R
b24	HPRTINT	主机端口中断 模块将该位置1时，指示主机模式下 USBFS 控制器端口的状态发生变化。应用程序必须读取USBFS_HPRT寄存器，以确定引发此中断的确切事件。应用程序必须先将USBFS_HPRT寄存器的相应状态位清零，之后才能将该位清零。 注意：仅可在主机模式下访问。	R
b23	Reserved	- 必须保持复位值。	R
b22	DATAFSUSP	数据获取挂起 该中断仅在DMA模式下有效。该中断指示，模块因TxFIFO空间或请求队列空间不可用而停止为IN端点获取数据。应用程序将该中断用于端点不匹配算法中。例如，在检测到端点不匹配后，应用程序将执行以下操作： — 将全局非周期性IN NAK握手信号置 1 — 禁止IN端点 — 清空FIFO — 根据IN令牌序列学习队列确定令牌序列 — 重新使能端点 — 如果全局非周期性IN NAK已清零但模块尚未为IN端点获取数据，同时又已接收到IN令牌，则清零全局非周期性IN NAK握手信号：模块将产生“FIFO为空时接收到IN令牌”中断。然后，USBFS将NAK响应发送到主机。 为了避免这种情况的发生，应用程序可以检查USBFS_GINTSTS 中的	R/W

DATAFSUSP中断，该中断可确保在FIFO存满后再将全局 NAK 握手信号清零。或者，应用程序可以在将全局IN NAK握手信号清零时屏蔽“当 FIFO为空时接收到IN令牌中断。

通过软件对该位写1清零。

注意：仅可在设备模式下访问。

b21	IPXFR/ INCOMPISO OUT	未完成周期性传输/ 未完成OUT同步传 输	IPXFR:未完成周期性传输 (Incomplete periodic transfer) 在主机模式下，如果存在仍处于挂起状态的未完成周期性事务，而这些事务计划在当前帧期间完成，则模块会将该中断位置1。 通过软件对该位写1清零。 注意：仅可在主机模式下访问。 INCOMPISOOUT: 未完成OUT同步传输 (Incomplete isochronous OUT transfer) 在设备模式下，模块将该中断置 1 时，指示当前帧中至少有一个同步OUT端点上的传输未完成。该中断随该寄存器中的周期性帧结束中断 (EOPF) 位一同触发。 通过软件对该位写1清零。 注意：仅可在设备模式下访问。	R/W
b20	IISOIXFR	未完成IN同步传输	未完成IN同步传输 (Incomplete isochronous IN transfer) 模块将该中断置1时，指示当前帧中至少有一个同步IN端点上的传输未完成。 该中断随该寄存器中的周期性帧结束中断 (EOPF) 位一同触发。 通过软件对该位写1清零。 注意：仅可在设备模式下访问。	R/W
b19	OEPINT	OUT端点中断	OUT端点中断(OUT endpoint interrupt) 模块将该位置1时，指示模块中一个OUT端点上存在挂起的中断（在设备模式下）。应用程序必须读取主机USBFS_DAINT寄存器，以确定发生中断的OUT端点的准确编号，然后读取相应的USBFS_DOEPINTx寄存器，以确定引发中断的确切原因。应用程序必须先将相应USBFS_DOEPINTx 寄存器的相应状态位清零，之后才能将该位清零。 注意：仅可在设备模式下访问。	R
b18	IEPINT	IN端点中断	IN端点中断 (IN endpoint interrupt) 模块将该位置1时，指示模块中一个IN端点上存在挂起的中断（在设备模式下）。应用程序必须读取主机USBFS_DAINT寄存器，以确定发生中断的IN端点的准确编号，然后读取相应的USBFS_DIEPINTx寄存器，以确定引发中断的确切原因。应用程序必须先将相应USBFS_DIEPINTx 寄存器的相应状态位清零，之后才能将该位清零。 注意：仅可在设备模式下访问。	R
b17~b16	Reserved	-	必须保持复位值。	R
b15	EOPF	周期性帧结束中断	周期性帧结束中断 (End of periodic frame interrupt) 指示当前帧已达到USBFS_DCFG 寄存器中周期性帧间隔字段 (USBFS_DCFG 中的PFIVL位) 所指定的周期。 通过软件对该位写1清零。 注意：仅可在设备模式下访问。	R/W

b14	ISOODRP	丢弃同步OUT数据包中断	丢弃同步OUT数据包中断(Isochronous OUT packet dropped interrupt) 如果由于Rx FIFO空间不足，无法容纳同步OUT端点的最大数据包，从而导致模块无法向Rx FIFO写入同步OUT数据包，模块会将该位置1。 通过软件对该位写1清零。 注意：仅可在设备模式下访问。	R/W
b13	ENUMDNE	枚举完成中断	枚举完成中断 (Enumeration done interrupt) 模块将该位置1时，指示速度枚举已完成。应用程序必须读取USBFS_DSTS寄存器来获取枚举速度。 通过软件对该位写1清零。 注意：仅可在设备模式下访问。	R/W
b12	USBRST	USB复位中断	USB复位中断(USB reset interrupt) 模块将该位置1时，指示在USB 上检测到复位信号。 通过软件对该位写1清零。 注意：仅可在设备模式下访问。	R/W
b11	USBSUSP	USB挂起中断	USB挂起中断(USB suspend interrupt) 模块将该位置1时，指示在USB上检测到挂起状态。当USB总线上的空闲状态保持3ms，模块便会进入挂起状态。通过软件对该位写1清零。 注意：仅可在设备模式下访问。	R/W
b10	ESUSP	早期挂起中断	早期挂起中断(Early suspend interrupt) 模块将该位置1时，指示已检测到USB处于空闲状态的时间达到3ms。 注意：仅可在设备模式下访问。	R/W
b9~b8	Reserved	-	必须保持复位值。	R
b7	GONAKEFF	全局OUT NAK有效中断	全局OUT NAK有效中断(Global OUT NAK effective interrupt) 指示USBFS_DCTL寄存器中由应用程序设置的“将全局OUT NAK置 1”位 (USBFS_DCTL中的SGONAK位) 已在模块中生效。通过写入USBFS_DCTL 寄存器中的“将全局 OUT NAK 清零”位 (USBFS_DCTL中的CGONAK位)，可将该位清零。 注意：仅可在设备模式下访问。	R
b6	GINAKEFF	全局非周期性IN NAK有效中断	全局非周期性IN NAK有效中断(Global IN nonperiodic NAK effective interrupt) 指示USBFS_DCTL寄存器中由应用程序设置的“将全局非周期性IN NAK置 1”位 (USBFS_DCTL中的SGINAK位) 已在模块中生效。也就是说，模块已对应用程序设置的全局IN NAK位进行采样，结果已生效。通过清零USBFS_DCTL寄存器中的“将全局非周期性IN NAK清零”位 (UBSFS_DCTL中的CGINAK位)，可将该位清零。此中断不一定表示USB上已发送了一个NAK 握手信号。STALL位优先级高于NAK 位。 注意：仅可在设备模式下访问。	R

b5	NPTXFE	非周期性Tx FIFO空 中断	非周期性Tx FIFO空中断(Non-periodic Tx FIFO empty interrupt) 当非周期性Tx FIFO为半空或全空状态，且非周期性发送请求队列中至少存在可写入一个条目的空间时，将触发该中断。该 FIFO 为半空状态还是全空状态由USBFS_GAHBCFG寄存器中的非周期性Tx FIFO空级别位(USBFS_GAHBCFG中的TXFELVL位)决定。 注意：仅可在主机模式下访问。	R
b4	RXFNE	RxFIFO非空中断	RxFIFO非空中断(RxFIFO non-empty interrupt) 指示RxFIFO中至少有一个数据包等待读取。 注意：在主机模式和设备模式均可访问。	R
b3	SOF	帧起始中断	帧起始中断 (Start of frame interrupt) 在主机模式下，模块将该位置1时，指示 USB 上已发送一个 SOF (FS) 或 Keep-Alive (LS)信号。应用程序必须将此位置1才可清除该中断。 在设备模式下，模块将该位置1时，指示 USB 上已接收到一个SOF令牌。应用程序可通过读取设备状态寄存器来获得当前的帧编号。只有在模块以 FS 模式运行时，才会出现此中断。 通过软件对该位写1清零。 注意：在主机模式和设备模式均可访问。	R/W
b2	Reserved	-	必须保持复位值。	R
b1	MMIS	模式不匹配中断	模式不匹配中断(Mode mismatch interrupt) 当应用程序尝试访问以下寄存器时，模块将该位置1： <ul style="list-style-type: none">— 模块运行在设备模式下访问主机模式寄存器— 模块运行在主机模式下访问设备模式寄存器 寄存器访问在AHB上以OKAY响应结束，但该访问在内部被模块忽略并且不影响模块运行。 通过软件对该位写1清零。 注意：在主机模式和设备模式均可访问。	R/W
b0	CMOD	当前工作模式	当前工作模式 (Current mode of operation) 指示当前模式。 0: 设备模式 1: 主机模式 注意：在主机模式和设备模式均可访问。	R

31.7.2.6 USBFS 全局中断屏蔽寄存器 (USBFS_GINTMSK)

USBFS interrupt mask register

偏移地址: 0x18

复位值: 0x00000000

该寄存器与模块中断寄存器结合使用，以中断应用程序。如果将某个中断位屏蔽，则不会产生与该位相关的中断。

但是，与该中断相对应的模块中断 (USBFS_GINTSTS) 寄存器位仍会置 1。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
WKUIM	VBUSVIM	DISCIM	CIDSCHGM	Reserved	PTXFEM	HCIIM	HPRITM	Reserved	DATAFSUSPM	IPXFRM/ INCOMPISQOLTM	IISOIXFRM	OEPIM	IEPIM	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EOPFM	ISOODRPM	ENUMDNEM	USBRSTM	USBSSUSPM	ESUSPM	Reserved	Reserved	GONAKEFFM	GINAKEFFM	NPTXFEM	RXFNEM	SOFM	Reserved	MMISM	Reserved

位	标记	位名	功能	读写
b31	WKUIM	检测到恢复/远程唤醒中断屏蔽	检测到恢复/远程唤醒中断屏蔽 (Resume/remote wakeup detected interrupt mask) 0: 屏蔽中断 1: 使能中断 注意：在主机模式和设备模式均可访问。	R/W
b30	VBUSVIM	VBUS有效中断屏蔽	VBUS有效中断屏蔽 (VBUS valid interrupt mask) 0: 屏蔽中断 1: 使能中断 注意：仅可在设备模式下访问。	R/W
b29	DISCIM	检测到断开连接中断屏蔽	检测到断开连接中断屏蔽 (Disconnect detected interrupt interrupt mask) 0: 屏蔽中断 1: 使能中断 注意：仅可在主机模式下访问。	R/W
b28	CIDSCHGM	中断连接器ID线状态变化中断屏蔽	连接器ID线状态更改中断屏蔽(Connector ID status change interrupt mask) 0: 屏蔽中断 1: 使能中断 注意：在设备模式和主机模式均可访问。	R/W

b27	Reserved	-	必须保持复位值。	R
b26	PTXFEM	周期性TxFIFO空中断屏蔽 蔽	周期性TxFIFO空中断屏蔽 (Periodic TxFIFO empty interrupt mask) 0: 屏蔽中断 1: 使能中断 注意：仅可在主机模式下访问。	R/W
b25	HCIM	主机通道中断屏蔽	主机通道中断屏蔽 (Host channels interrupt mask) 0: 屏蔽中断 1: 使能中断 注意：仅可在主机模式下访问。	R/W
b24	HPRTIM	主机端口中断屏蔽	主机端口中断屏蔽 (Host port interrupt mask) 0: 屏蔽中断 1: 使能中断 注意：仅可在主机模式下访问。	R/W
b23	Reserved	-	必须保持复位值。	R
b22	DATAFSUSP M	数据获取挂起中断屏蔽	数据获取挂起中断屏蔽 (Data fetch suspended interrupt mask) 0: 屏蔽中断 1: 使能中断 注意：仅可在设备模式下访问。	R/W
b21	IPXFRM/ INCOMPISOO UTM	未完成周期性传输中断屏蔽 蔽/ 未完成OUT同步传输中 断屏蔽	IPXFR:未完成周期性传输中断屏蔽(Incomplete periodic transfer interrupt mask) 0: 屏蔽中断 1: 使能中断 注意：仅可在主机模式下访问。 INCOMPISOOUT: 未完成OUT同步传输中断屏蔽(Incomplete isochronous OUT transfer interrupt mask) 0: 屏蔽中断 1: 使能中断 注意：仅可在设备模式下访问。	R/W
b20	IISOIXFRM	未完成IN同步传输中断 屏蔽	未完成IN同步传输中断屏蔽(Incomplete isochronous IN transfer interrupt mask) 0: 屏蔽中断 1: 使能中断 注意：仅可在设备模式下访问。	R/W
b19	OEPIM	OUT端点中断屏蔽	OUT端点中断屏蔽(OUT endpoint interrupt mask) 0: 屏蔽中断 1: 使能中断 注意：仅可在设备模式下访问。	R/W
b18	IEPIM	IN端点中断屏蔽	IN端点中断屏蔽(IN endpoint interrupt mask) 0: 屏蔽中断 1: 使能中断 注意：仅可在设备模式下访问。	R/W
b17~b16	Reserved	-	必须保持复位值。	R

b15	EOPFM	周期性帧结束中断屏蔽	周期性帧结束中断屏蔽(End of periodic frame interrupt mask)	R/W
			0: 屏蔽中断 1: 使能中断	
			注意: 仅可在设备模式下访问。	
b14	ISOODRPM	丢弃同步OUT数据包中断屏蔽	丢弃同步OUT数据包中断屏蔽(Isochronous OUT packet dropped interrupt mask)	R/W
			0: 屏蔽中断 1: 使能中断	
			注意: 仅可在设备模式下访问。	
b13	ENUMDNEM	枚举完成中断屏蔽	枚举完成中断屏蔽(Enumeration done interrupt mask)	R/W
			0: 屏蔽中断 1: 使能中断	
			注意: 仅可在设备模式下访问。	
b12	USBRSTM	USB复位中断屏蔽	USB复位中断屏蔽(USB reset interrupt mask)	R/W
			0: 屏蔽中断 1: 使能中断	
			注意: 仅可在设备模式下访问。	
b11	USBSUSPM	USB挂起中断屏蔽	USB挂起中断屏蔽(USB suspend interrupt mask)	R/W
			0: 屏蔽中断 1: 使能中断	
			注意: 仅可在设备模式下访问。	
b10	ESUSPM	早期挂起中断屏蔽	早期挂起中断屏蔽(Early suspend interrupt mask)	R/W
			0: 屏蔽中断 1: 使能中断	
			注意: 仅可在设备模式下访问。	
b9~b8	Reserved	-	必须保持复位值。	R
b7	GONAKEFFM	全局OUT NAK有效中断屏蔽	全局OUT NAK有效中断屏蔽(Global OUT NAK effective interrupt mask)	R/W
			0: 屏蔽中断 1: 使能中断	
			注意: 仅可在设备模式下访问。	
b6	GINAKEFFM	全局非周期性IN NAK有效中断屏蔽	全局非周期性IN NAK有效中断屏蔽(Global IN nonperiodic NAK effective interrupt mask)	R/W
			0: 屏蔽中断 1: 使能中断	
			注意: 仅可在设备模式下访问。	
b5	NPTXFEM	非周期性Tx FIFO空中断屏蔽	非周期性Tx FIFO空中断(Non-periodic Tx FIFO empty interrupt mask)	R/W
			0: 屏蔽中断 1: 使能中断	
			注意: 仅可在主机模式下访问。	

b4	RXFNEM	RxFIFO 非空中断屏蔽	RxFIFO非空中断屏蔽(RxFIFO non-empty interrupt mask)	R/W
		0: 屏蔽中断		
		1: 使能中断		
		注意: 在主机模式和设备模式均可访问。		
b3	SOFM	帧起始中断屏蔽	帧起始中断屏蔽(Start of frame interrupt mask)	R/W
		0: 屏蔽中断		
		1: 使能中断		
		注意: 在主机模式和设备模式均可访问。		
b2	Reserved	-	必须保持复位值。	R
b1	MMISM	模式不匹配中断中断屏蔽	模式不匹配中断屏蔽(Mode mismatch interrupt mask)	R/W
		0: 屏蔽中断		
		1: 使能中断		
		注意: 在主机模式和设备模式均可访问。		
b0	Reserved	-	必须保持复位值。	R

31.7.2.7 USBFS 接收状态调试读取 /USBFS 状态读取和出栈寄存器 (USBFS_GRXSTS/USBFS_GRXSTSP)

USBFS Receive status debug read/USBFS status read and pop registers

读取的偏移地址: 0x01C

出栈的偏移地址: 0x020

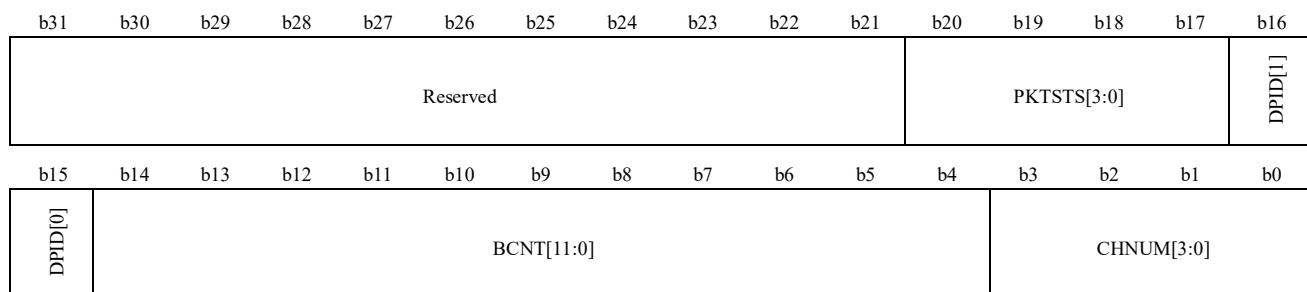
复位值: 0x0000 0000

读取接收状态调试读取寄存器将返回接收 FIFO 顶部的内容。读取接收状态读取和出栈寄存器将额外弹出 RxFIFO 顶部的数据条目。接收状态内容在主机模式和设备模式下的解释不同。

当接收 FIFO 为空时, 模块会忽略对该寄存器的读取或出栈操作, 并返回值 0x0000 0000。

当模块中断寄存器的接收 FIFO 非空位 (USBFS_GINTSTS 中的 RXFNE 位) 置位时, 应用程序必须仅弹出接收状态 FIFO。

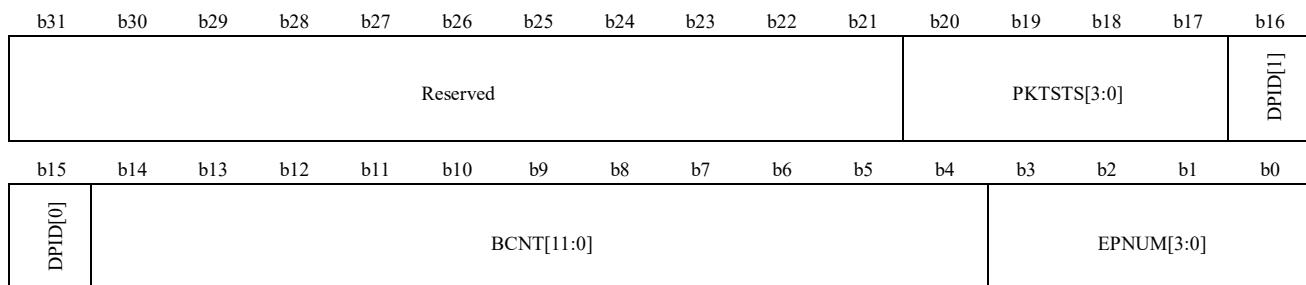
主机模式:



位	标记	位名	功能	读写
b31~b21	Reserved	-	必须保持复位值。	R
b20~b17	PKTSTS	数据包状态	数据包状态 (Packet status) 指示接收的数据包的状态 0010: 接收到IN数据包 0011: IN传输完成 (触发中断) 0101: 数据同步错误 (触发中断) 0111: 暂停通道 (触发中断) 其他值: 保留	R

b16~b15	DPID	数据PID	数据PID (Data PID) 指示接收的数据包的数据PID 00: DATA0 10: DATA1 01: DATA2 11: MDATA	R
b14~b4	BCNT	字节计数	字节计数(Byte count) 指示接收的IN数据包的字节数。	R
b3~b0	CHNUM	通道编号	通道编号(Channel number) 指示当前接收的数据包所属的通道编号。	R

设备模式:



位	标记	位名	功能	读写
b31~b21	Reserved	-	必须保持复位值。	R
b20~b17	PKTSTS	数据包状态	数据包状态(Packet status) 指示接收的数据包的状态 0001: 全局OUT NAK (触发中断) 0010: 接收到OUT数据包 0011: OUT传输完成 (触发中断) 0100: SETUP事务完成 (触发中断) 0110: 接收到SETUP数据包 其它值: 保留	R
b16~b15	DPID	数据PID	数据PID(Data PID) 指示接收的OUT数据包的数据PID 00: DATA0 10: DATA1 01: DATA2 11: MDATA	R
b14~b4	BCNT	字节计数	字节计数(Byte count) 指示接收的数据包的字节数。	R
b3~b0	EPNUM	端点编号	端点编号(Endpoint number) 指示当前接收的数据包所属的端点编号。	R

31.7.2.8 USBFS 接收 FIFO 大小寄存器 (USBFS_GRXFSIZ)

USBFS Receive FIFO size register

偏移地址: 0x024

复位值: 0x0000 0140

此应用程序可以对必须分配给 RxFIFO 的 RAM 大小进行编程。

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

位	标记	位名	功能	读写
b31~b11	Reserved	-	必须保持复位值。	R
b10~b0	RXFD	RxFifo深度	RXFD: RxFIFO深度 (RxFIFO depth) 以32位字为单位。 最小值为16 最大值为256 上电复位值为最大Rx数据FIFO深度。	R/W

31.7.2.9 USBFS 主机非周期性发送 FIFO 大小寄存器(USBFS_HNPTXFSIZ)/端点 0 发送 FIFO 大小(USBFS_DIEPTXF0)

USBFS Host non-periodic transmit FIFO size register/Device endpoint0 transmit FIFO size register

偏移地址: 0x028

复位值: 0x02000140

此应用程序可以对必须分配给 TxFIFO 的 RAM 大小进行编程。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
NPTXFD[15:0]/TX0FD[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
NPTXFSA[15:0]/TX0FSA[15:0]															

位	标记	位名	功能	读写
b31~b16	NPTXFD/	非周期性TxFIFO 深度	主机模式: NPTXFD	R/W
	TX0FD	度/端点0 TxFIFO深度	非周期性TxFIFO深度(Non-periodic TxFIFO depth)	
		度	以32位字为单位。	
			最小值为16	
			最大值为256	
			设备模式: TX0FD	
			端点0TxFIFO深度(Endpoint 0 TxFIFO depth)	
			以32位字为单位。	
			最小值为16	
			最大值为256	
b15~b0	NPTXFSA/	非周期性发送RAM	主机模式: NPTXFSA	R/W
	TX0FSA	起始地址/端点0发送	非周期性发送RAM起始地址(Non-periodic transmit RAM start address)	
		RAM起始地址	此字段包含非周期性发送FIFO RAM的存储器起始地址。	
			设备模式: TX0FSA	
			端点0发送RAM起始地址(Endpoint 0 transmit RAM start address)	
			此字段包含端点0发送FIFO RAM的存储器起始地址。	

31.7.2.10 USBFS 非周期性发送 FIFO/队列状态寄存器 (USBFS_HNPTXSTS)

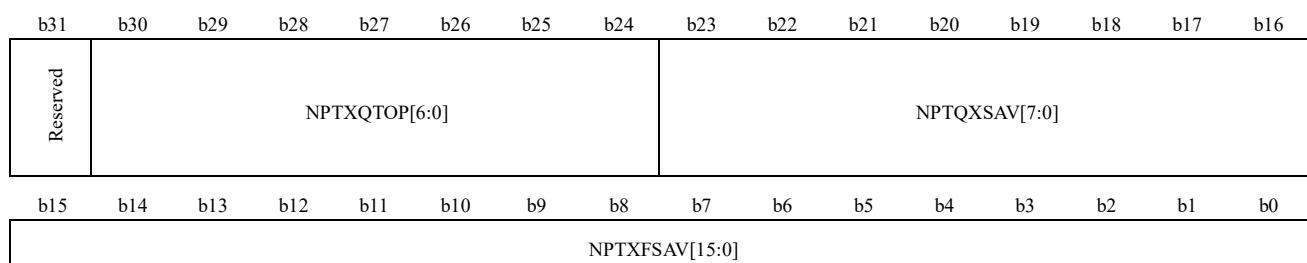
USBFS Host non-periodic transmit FIFO size register/Device endpoint0 transmit FIFO size register

偏移地址: 0x02C

复位值: 0x00080100

此只读寄存器包含非周期性 TxFIFO 和非周期性发送请求队列的自由空间信息。

此寄存器仅在主机模式有效，设备模式无效。



位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R
b30~b24	NPTXQTOP	非周期性发送请求 队列顶部	非周期性发送请求队列顶部 (Top of the non-periodic transmit request queue) 非周期性发送请求队列中MAC目前正在处理的条目。 位30:27: 通道/端点编号(Channel/endpoint number) 位26:25: <ul style="list-style-type: none">— 00: IN/OUT 令牌— 01: 长度为零的发送数据包— 11: 通道停止命令 位24: 结束 (所选通道/端点的最后一个条目) (Terminate (last entry for selected channel))	R
b23~b16	NPTQXSAV	非周期性发送请求 队列可用空间	非周期性发送请求队列可用空间 (Non-periodic transmit request queue space available) 指示非周期性发送请求队列中的可用空闲空间大小。 在主机模式下，此队列保存IN和OUT请求。 00: 非周期性发送请求队列已满 01: 1 个位置可用 10: 2 个位置可用 bxn: n 个位置可用 (其中, n范围: 0~8) 其它值: 保留	R

b15~b0	NPTXFSAV	非周期性发送请求 队列顶部	非周期性TxFIFO可用空间 (Non-periodic TxFIFO space available) 指示非周期性TxFIFO中的可用空闲空间大小。 以 32 位字为单位。 00: 非周期性 TxFIFO 已满 01: 1 个字可用 10: 2 个字可用 0xn: n 个字可用 (其中, n范围: 0~256) 其它值: 保留	R
--------	----------	------------------	---	---

31.7.2.11 USBFS 模块 ID 寄存器(USBFS_CID)

USBFS core ID register

偏移地址: 0x03C

复位值: 0x12345678

该寄存器为可编程用户配置 ID 寄存器。

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

位	标记	位名	功能	读写
b31~b0	PRODUCT_ID	产品ID字段	产品ID字段(Product ID field) 可通过应用程序编程的ID字段。	R/W

31.7.2.12 USBFS 主机周期性发送 FIFO 大小寄存器(USBFS_HPTXFSIZ)

USBFS Host periodic transmit FIFO size register

偏移地址: 0x100

复位值: 0x01400280

此应用程序可以对必须分配给周期 TxFIF 的 RAM 大小进行编程。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved				PTXFD[10:0]											
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				PTXSA[11:0]											

位	标记	位名	功能	读写
b31~b27	Reserved	-	必须保持复位值。	R
b26~b16	PTXFD	主机周期性Tx FIFO 深度	主机周期性Tx FIFO 深度 (Host periodic Tx FIFO depth) 以32位字为单位。 最小值为16 最大值为256	R/W
b15~b12	Reserved	-	必须保持复位值。	R
b11~b0	PTXSA	主机周期性Tx FIFO 起始地址	主机周期性Tx FIFO 起始地址(Host periodic Tx FIFO start address) 上电复位值是最大Rx FIFO 深度与最大非周期性Tx FIFO 深度之和。	R/W

31.7.2.13 USBFS 设备 IN 端点发送 FIFO 大小寄存器 (USBFS_DIEPTXF_x) ($x = 1..5$)

USBFS device IN endpoint transmit FIFO size register

偏移地址: 0x104+(x-1)*0x4

复位值: 0x01000240+(x-1)*0x100

此应用程序可以对必须分配给设备 TxFIFO 的大小进行编程。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								INEPTXFD[9:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								INEPTXSA[11:0]							

位	标记	位名	功能	读写
b31~b26	Reserved	-	必须保持复位值。	R
b25~b16	INEPTXFD	IN端点Tx FIFO深度	设备IN端点Tx FIFO深度 (Device IN endpoint Tx FIFO depth) 以32位字为单位。 最小值为16 最大值为256	R/W
b15~b12	Reserved	-	必须保持复位值。	R
b11~b0	INEPTXSA	IN端点Tx FIFO _x	IN端点Tx FIFO _x RAM起始地址 RAM起始地址 (IN endpoint FIFO _x transmit RAM start address) 此字段包含IN端点发送FIFO _x 的存储器起始地址。 该地址必须与32位存储器位置对齐。	R/W

31.7.3 USBFS 主机模式寄存器

主机模式寄存器会影响主机模式下的模块操作。在设备模式下不得访问主机模式寄存器，因为产生的结果不明确。

除非特别说明，否则寄存器描述中的位值以二进制表示。

31.7.3.1 USBFS 主机配置寄存器 (USBFS_HCFG)

USBFS Host configuration register

偏移地址：0x400

复位值：0x00000000

此寄存器将在上电后对模块进行配置。请勿在初始化主机后更改此寄存器。

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

位	标记	位名	功能	读写
b31~b3	Reserved	-	必须保持复位值。	R
b2	FSLSS	仅支持FS和LS	应用程序使用此位控制模块的枚举速度。使用此位，应用程序可使模块工作为FS主机，即使所连接的设备支持HS通信也是如此。请勿在初始编程后更改此字段。 1: 仅限 FS/LS，即使所连接设备可支持HS	R/W
b1~b0	FSLSPCS	FS/LS PHY时钟选择	FS/LS PHY时钟选择 (FS/LS PHY clock select) 当模块处于FS主机模式时 01: PHY时钟以48 MHz运行 其它值: 保留 当模块处于LS主机模式时 00: 保留 01: 选择48MHz PHY时钟频率 10: 选择6MHz PHY时钟频率 11: 保留 注意: 当设备连上主机时，必须依照所连接设备的速度设置 FSLSPCS (更改此位后，必须进行软件复位)。	R/W

31.7.3.2 USBFS 主机帧时间间隔寄存器 (USBFS_HFIR)

USBFS Host frame interval register

偏移地址: 0x404

复位值: 0x0000EA60

此寄存器用于存储 USBFS 控制器对已连接设备当前速度所设定的帧间隔信息。

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

位	标记	位名	功能	读写
b31~b26	Reserved	-	必须保持复位值	R
b15~b0	FRIVL	帧间隔	帧间隔 (Frame interval) 应用程序在此字段编程的值用于指定两个连续SOF(FS)或Keep-Alive令牌(LS)之间的时间间隔。此字段包含构成所需帧间隔的PHY时钟数。只有将主机端口控制和状态寄存器的端口使能位 (USBFS_HPRT的PENA位) 置1后，应用程序才能向此寄存器中写入值。如果未对值进行编程，模块将根据在主机配置寄存器的FS/LS PHY时钟选择字段 (USBFS_HCFG中的 FSLSPCS) 中指定的PHY时钟来计算。请勿在初始配置后更改此字段的值。 设定值=帧间隔 (ms) × (PHY 时钟频率) -1 注意： 只要应用程序需要更改帧间隔时间，即可对FRIVL位进行修改。	R/W

31.7.3.3 USBFS 主机帧编号/帧剩余时间寄存器 (USBFS_HFNUM)

USBFS Host frame interval register

偏移地址: 0x408

复位值: 0x0000 3FFF

此寄存器用于指示当前帧编号。它还指示当前帧的剩余时间(以PHY时钟数为单位)。

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

位	标记	位名	功能	读写
b31~b16	FTREM	帧剩余时间	帧剩余时间 (Frame time remaining) 指示当前帧的剩余时间 (以PHY时钟数为单位)。每过去1个PHY时钟，此字段递减 1。 当值达到零时，此字段将重新装载帧间隔寄存器中的值，并由模块在USB 上发送一个新 SOF。	R
b15~b0	FRNUM	帧编号	帧编号 (Frame number) 当在USB上发送1个新SOF时此字段的值将递增1，当达到 0x3FFF 时会清零。	R

31.7.3.4 USBFS 主机周期性发送 FIFO/队列状态寄存器(USBFS_HPTXSTS)

USBFS Host periodic transmit FIFO/queue status register

偏移地址: 0x410

复位值: 0x00080100

此只读寄存器包含周期性 TxFIFO 和周期性发送请求队列的空闲空间信息。

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

位	标记	位名	功能	读写
b31~b24	PTXQTOP	周期性发送请求队列顶部	周期性发送请求队列顶部 (Top of the periodic transmit request queue) 指示周期性Tx请求队列中MAC当前正在处理的项。 该寄存器用于调试。 位 31: 奇数/偶数帧 (Odd/Even frame) — 0: 以偶数帧发送 — 1: 以奇数帧发送 位 30:27: 通道/端点编号 (Channel number) 位 26:25: 类型 (Type) — 00: 输入/输出 — 01: 零长度数据包 — 11: 禁止通道命令 位 24: 结束 (所选通道的最后一个条目) (Terminate (last entry for the selected channel))	R
b23~b16	PTXQSAV	周期性发送请求队列可用空间	周期性发送请求队列可用空间 (Periodic transmit request queue space available) 指示可供写入的周期性发送请求队列的空闲位置的数量。该队列既包含IN请求，又包含OUT请求。 00: 周期性发送请求队列已满 01: 1个位置可用 10: 2个位置可用 bxn: n个位置可用 (其中, n范围: 0~8) 其它值: 保留	R

b15~b0	PTXFSAVL	周期性发送数据 FIFO可用空间	周期性发送数据FIFO可用空间 (Periodic transmit data FIFO space available) 指示可供写入的周期性 TxFIFO 的空闲位置的数量。 以32位字为单位 0000: 周期性TxFIFO已满 0001: 1个字可用 0010: 2个字可用 bxn: n个字可用 (其中, n范围: 0~PTXFD) 其它值: 保留	R
--------	----------	---------------------	--	---

31.7.3.5 USBFS 主机全体通道中断寄存器 (USBFS_HINT)

USBFS Host all channels interrupt register

偏移地址: 0x414

复位值: 0x0000 0000

当通道上有事件发生时，主机全体通道中断寄存器会使用模块中断寄存器中的主机通道中断位 (USBFS_GINTSTS 中的 HCINT 位) 中断应用程序。每个通道对应 1 个中断位，最多有 12 个位。

当应用程序通过相应主机通道 x 中断寄存器清零中断时，该寄存器中的位也会清零。

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

位	标记	位名	功能	读写
b31~b12	Reserved	-	必须保持复位值。	R
b11~b0	HINT	通道中断	通道中断(Channel interrupt) 每个通道对应一位：通道0对应位0，通道11对应位11。	R/W

31.7.3.6 USBFS 主机全体通道中断屏蔽寄存器(USBFS_HINTMSK)

USBFS Host all channels interrupt mask register

偏移地址: 0x418

复位值: 0x0000 0000

主机全体通道中断屏蔽寄存器与主机全体通道中断寄存器结合使用，进而在通道上发生事件时中断应用程序。

每个通道对应 1 个中断屏蔽位，最多有 12 个位。

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

位	标记	位名	功能	读写
b31~b12	Reserved	-	读出时为“0”，写入时写“0”	R
b11~b0	HINTM	通道中断屏蔽	通道中断屏蔽 (Channel interrupt mask) 0: 屏蔽中断 1: 使能中断 每个通道对应一位：通道0对应位0，通道11对应位11。	R/W

31.7.3.7 USBFS 主机端口控制和状态寄存器 (USBFS_HPRT)

USBFS Host port control and status register

偏移地址: 0x440

复位值: 0x0000 0000

该寄存器仅在主机模式下可用。当前，USBFS 主机仅支持一个端口。

该寄存器包含 USB 端口相关的信息，如 USB 复位、使能、挂起、恢复、连接状态。

该寄存器中的 PENCHNG/PCDET 位可通过模块中断寄存器中的主机端口中断位 (USBFS_GINTST 中 HPRTINT 位) 触发应用程序中断。发生端口中断时，应用程序必须读取该寄存器，并将引起中断的位清零。应用程序必须向该位写入 1 以清除该中断。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved												PSPD[1:0]	Reserved		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved	PWPR	PLSTS[1:0]	Reserved	PRST	PSUSP	PRES	Reserved	PENCHNG	PENA	PCDET	PCSTS				

位	标记	位名	功能	读写
b31~b19	Reserved	-	必须保持复位值。	R
b18~b17	PSPD	端口速度	端口速度(Port speed) 指示连接到该端口的设备的速度。 00/11: 保留 01: 全速 10: 低速	R
b16~b13	Reserved	-	必须保持复位值。	R
b12	PWPR	端口电源	端口电源(Port power) 应用程序使用该字段控制该端口的电源。由于本USBFS内置PHY不具备供电能力，所以此位设置为1时，通过USBFS_DRVVBUS使能外部USB电源芯片供电。 0: 掉电 1: 通电	R/W
b11~b10	PLSTS	端口线状态	指示 USB 数据线的当前逻辑电平 位11: USBFS_DM 的逻辑电平 位10: USBFS_DP 的逻辑电平	R

b9	Reserved	-	必须保持复位值。	R
b8	PRST	端口复位	端口复位(Port reset) 应用程序将该位置1时，会在该端口上启动复位序列。应用程序必须为复位周期定时，并在复位序列完成后将该位清零。 0：端口未处于复位状态 1：端口处于复位状态 应用程序必须将该位置1并最少保持 10 ms，以在端口上启动复位。	R/W
b7	PSUSP	端口挂起	端口挂起(Port suspend) 应用程序将此位置1以将此端口置于挂起模式。只有此位置1时，模块才会停止发送SOF。要停止PHY时钟，应用程序必须将端口时钟停止位置1，这会使能PHY的挂起输入引脚。 此位的读取值反映该端口的当前挂起状态。检测到远程唤醒信号，或者应用程序将此寄存器中的端口复位位或端口恢复位置1后，模块可将此位清零；或应用程序将模块中断寄存器中的恢复/远程唤醒检测中断位或断开连接检测中断位（分别为USBFS_GINTSTS中的WKUINT或 DISCINT）置1，模块也可将此位清零。 0：端口未处于挂起模式 1：端口处于挂起模式	R/W
b6	PRES	端口恢复	端口恢复 (Port resume) 应用程序将此位置1以在该端口上驱动恢复信号。模块会持续驱动恢复信号直到应用程序将此位清零。 如模块中断寄存器中的端口恢复/ 远程唤醒检测中断位（USBFS_GINTSTS中 WKUINT位）指示，如果模块检测到USB远程唤醒序列，则开始驱动恢复信号，而无需应用程序进行干预；如果模块检测到断开连接的情况，则将此位清零。 此位的读取值指示当前模块是否正在驱动恢复信号。 0：不驱动恢复信号 1：驱动恢复信号	R/W
b5~b4	Reserved	-	必须保持复位值。	R
b3	PENCHNG	端口使能/禁止变化	端口使能/禁止变化 (Port enable/disable change) 该寄存器中的端口使能位2的状态发生变化时，模块将此位置1。 通过软件对该位写1清零。	R/W

b2	PENA	端口使能	端口使能(Port enable) 端口执行复位序列后，只能由模块使能，并且可以由过流状况、断开连接状况或应用程序将此位清零来禁止。应用程序无法通过对寄存器执行写操作将此位置1。只能将此位清零来禁止端口。对此位的操作不会触发应用程序的任何中断。 0: 禁止端口 1: 使能端口	R/W
b1	PCDET	检测到端口连接	检测到端口连接 (Port connect detected) 当检测到设备连接时，模块将此位置1，以使用模块中断寄存器中的主机端口中断位（USBFS_GINTSTS中的HPRTINT位）触发应用程序的中断。 应用程序必须将此位置1才可清除该中断。	R/W
b0	PCSTS	端口连接状态	端口连接状态(Port connect status) 0: 端口未连接设备 1: 端口已连接设备	R

31.7.3.8 USBFS 主机通道 x 特性寄存器 (USBFS_HCCHARx) (x = 0..11)

USBFS Host channel-x characteristics register

偏移地址: 0x500 + (通道编号 × 0x20)

复位值: 0x0000 0000

该寄存用于设定主机通道特性。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CHENA	CHDIS	ODDFRM		DAD[6:0]						Reserved	EPTYP[1:0]		LSDEV		Reserved
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EPDIR		EPNUM[3:0]								MPSIZ[10:0]					

位	标记	位名	功能	读写
b31	CHENA	通道使能	通道使能(Channel enable) 此字段由应用程序软件置1，并由USBFS主机硬件清零。 0: 禁止通道 (Channel disabled) 1: 使能通道	R/W
b30	CHDIS	通道禁止	通道禁止(Channel disable) 应用程序将此位置1以停止通过通道发送/接收数据，即使通过该通道的传输还未完成，停止操作仍然生效。 应用程序必须等待禁止通道的中断以确认通道已经被禁止。	R/W
b29	ODDFRM	奇数帧	奇数帧(Odd frame) 此字段由应用程序置位或复位，以分别指示USBFS主机必须传输奇数帧或偶数帧。此字段只适用于周期性（同步和中断）事务。 0: 偶数帧 1: 奇数帧	R/W
b28~b22	DAD	设备地址	设备地址(Device address) 此字段用于指定要与该主机通信的特定设备。	R/W
b21~b20	Reserved	-	必须保持复位值。	R
b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 指示选择的传输类型。 00: 控制 01: 同步 10: 批量 11: 中断	R/W

b17	LSDEV	低速设备	低速设备 (Low-speed device)	R/W
此字段由应用程序置 1，表示此通道正在与一个低速设备进行通信。				
b16	Reserved	-	必须保持复位值。	R
b15 EPDIR 端点方向 端点方向(Endpoint direction) R/W				
指示通信事务的方向是输入还是输出。 0: 输出 1: 输入				
b14-b11	EPNUM	端点编号	端点编号 (Endpoint number)	R/W
指示要与该主机通道通信的 USB 设备的端点号。				
b10-b0	MPSIZ	最大数据包大小	最大数据包大小 (Maximum packet size)	R/W
指示与该主机通道通信的设备端点的最大数据包大小。				

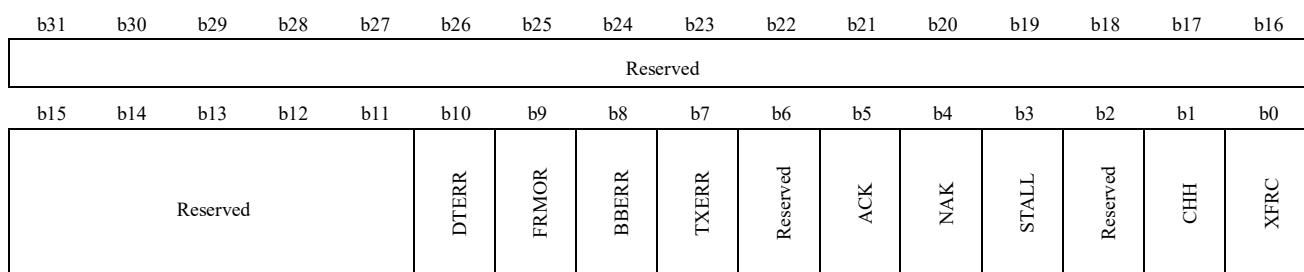
31.7.3.9 USBFS 主机通道 x 中断寄存器 (USBFS_HCINTx) (x = 0..11)

USBFS Host channel-x interrupt register

偏移地址: 0x508 + (通道编号 × 0x20)

复位值: 0x0000 0000

该寄存器指示在出现 USB 和 AHB 相关事件时通道的状态。当模块中断寄存器中的主机通道中断位 (USBFS_GINTSTS 中的 HCINT 位) 置 1 时, 应用程序必须读取该寄存器。在对寄存器执行读操作之前, 应用程序必须先读取主机全体通道中断 (USBFS_HAIN) 寄存器, 以获取主机通道 x 中断寄存器的准确通道编号。应用程序必须将该寄存器中的相应位清零, 才能将 USBFS_HAIN 和 USBFS_GINTSTS 寄存器中的对应位清零。



位	标记	位名	功能	读写
b31~b11	Reserved	-	必须保持复位值。	R
b10	DTERR	数据切换错误	数据同步错误 (Data toggle error) 应用程序需通过写1清除该位。	R/W
b9	FRMOR	帧溢出错误	帧溢出错误(Frame overrun) 应用程序需通过写1清除该位。	R/W
b8	BBERR	串扰错误	串扰错误(Babble error) 产生串扰事件的典型原因是端点发送了一个数据包, 但是数据包长度超过了端点的最大包长。 应用程序需通过写1清除该位。	R/W
b7	TXERR	通信事务错误	通信事务错误 (Transaction error) 指示 USB 上发生下列错误之一: CRC校验失败 超时 位填充错误 错误的EOP 应用程序需通过写1清除该位。	R/W
b6	Reserved	-	必须保持复位值。	R

b5	ACK	收到/发出ACK 响应	收到/发出ACK响应(ACK response received/transmitted interrupt) 应用程序需通过写1清除该位。	R/W
b4	NAK	收到NAK响应	收到NAK响应(NAK response received interrupt) 应用程序需通过写1清除该位。	R/W
b3	STALL	收到STALL响应	收到STALL响应(STALL response received interrupt)	R/W
b2	Reserved	-	读出时为“0”,写入时写“0”	R
b1	CHH	通道停止	通道停止(Channel halted) 因任意USB事务错误或为响应应用程序的禁止请求而导致传输非正常结束。 应用程序需通过写1清除该位。	R/W
b0	XFRC	传输完成	传输完成(Transfer completed) 未出现任何错误，正常完成传输。 应用程序需通过写1清除该位。	R/W

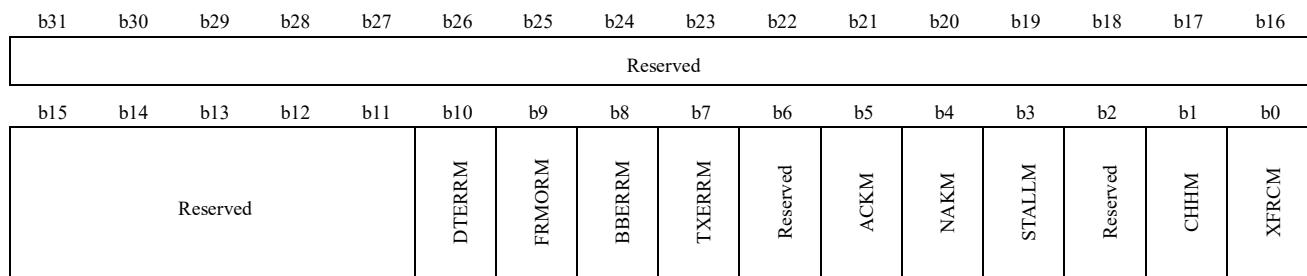
31.7.3.10 USBFS 主机通道 x 中断屏蔽寄存器 (USBFS_HCINTMSKx) (x = 0..11)

USBFS Host channel-x interrupt mask register

偏移地址: 0x50C + (通道编号 × 0x20)

复位值: 0x0000 0000

该寄存器用于选择屏蔽主机通道中断。



位	标记	位名	功能	读写
b31~b11	Reserved	-	必须保持复位值。	R
b10	DTERRM	数据切换错误中断 屏蔽	数据同步错误屏蔽(Data toggle error mask) 0: 屏蔽中断 1: 使能中断	R/W
b9	FRMORM	帧溢出错误中断屏 蔽	帧溢出错误屏蔽(Frame overrun mask) 0: 屏蔽中断 1: 使能中断	R/W
b8	BBERRM	串扰错误中断屏 蔽	串扰错误屏蔽(Babble error mask) 0: 屏蔽中断 1: 使能中断	R/W
b7	TXERRM	通信事务错误中断 屏蔽	通信事务错误屏蔽(Transaction error mask) 0: 屏蔽中断 1: 使能中断	R/W
b6	Reserved	-	必须保持复位值。	R
b5	ACKM	收到/发出ACK 响 应中断屏蔽	ACK响应接收/发送中断屏蔽 (ACK response received/transmitted interrupt mask) 0: 屏蔽中断 1: 使能中断	R/W
b4	NAKM	收到NAK响应中断 屏蔽	NAK响应接收中断屏蔽(NAK response received interrupt mask) 0: 屏蔽中断 1: 使能中断	R/W
b3	STALLM	收到STALL响应中 断屏蔽	收到STALL响应(STALL response received interrupt mask) 0: 屏蔽中断 1: 使能中断	R/W
b2	Reserved	-	必须保持复位值。	R

b1	CHHM	通道停止中断屏蔽	通道停止中断屏蔽 (Channel halted mask)	R/W
			0: 屏蔽中断	
			1: 使能中断	
b0	XFRCM	传输完成中断屏蔽	传输完成中断屏蔽 (Transfer completed mask)	R/W
			0: 屏蔽中断	
			1: 使能中断	

31.7.3.11 USBFS 主机通道 x 传输大小寄存器 (USBFS_HCTSIZx) (x = 0..11)

USBFS Host channel-x transfer size register

偏移地址: 0x510 + (通道编号 × 0x20)

复位值: 0x0000 0000

该寄存器用于设定主机通道传输大小以及数据 PID。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	DPID[1:0]		PKTCNT[9:0]										XFRSIZ[18:16]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XFRSIZ[15:0]															

位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R
b30~b29	DPID	数据PID	数据PID (Data PID) 应用程序在此字段设置数据通信的初始同步PID。 主机在此次传输事务过程中保留该字段的设置。 00: DATA0 01: 保留 10: DATA1 11: SETUP	R/W
b28~b19	PKTCNT	数据包计数	数据包计数 (Packet count) 应用程序在此字段中设置将要发送或接收的数据包数。 主机每成功发送或接收一个数据包便递减一次计数值。此值达到0后，将中断 应用程序来指示操作正常完成。	R/W
b18~b0	XFRSIZ	传输大小	传输大小(Transfer size) 对于OUT操作，此字段为传输期间主机发送的数据字节数。 对于IN操作，此字段为应用程序保留给传输的缓冲区大小。对于IN事务（周期 性和非周期性），应用程序会将此字段编程为最大数据包大小的整数倍。	R/W

31.7.3.12 USBFS 主机通道 xDMA 地址寄存器 (USBFS_HCDMAx) (x = 0..11)

USBFS Host channel-x DMA address register

偏移地址: 0x514 + (通道编号 × 0x20)

复位值: 0xFFFF XXXX

该寄存器用于设定主机 DMA 模式时 DMA 地址。

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

位	标记	位名	功能	读写
b31~b0	DMAADDR	DMA地址	DMA地址(DMA address) 此字段存储主机从设备端点获取数据或往设备端点发送数据所用DMA传输的 存储器的地址。每次AHB传输结束，该寄存器都会递增。	R/W

31.7.4 USBFS 设备模式寄存器

设备模式寄存器会影响设备模式下的模块操作。在主机模式下不得访问设备模式寄存器，因为产生的结果不明确。

除非特别说明，否则寄存器描述中的位值以二进制表示。

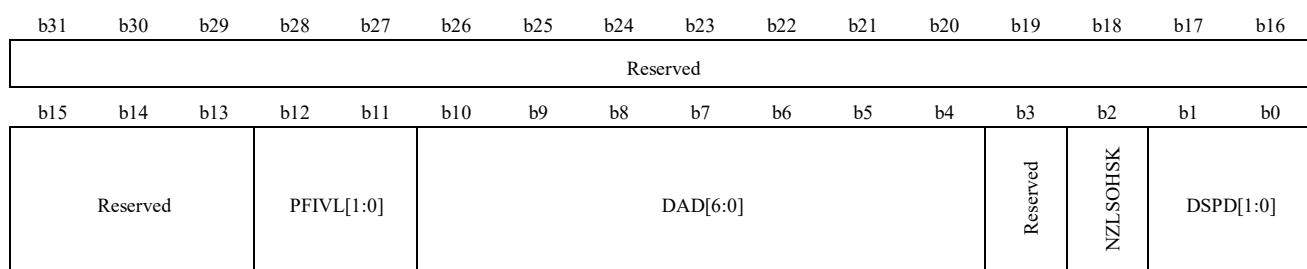
31.7.4.1 USBFS 设备配置寄存器(USBFS_DCFG)

USBFS Device configuration register

偏移地址：0x800

复位值：0x0820 0000

此寄存器在上电、执行某些控制命令或枚举后，会将模块配置为设备模式。请勿在初始编程后更改该寄存器。



位	标记	位名	功能	读写
b31~b13	Reserved	-	必须保持复位值。	R
b12~b11	PFIVL	周期性帧间隔	周期性帧间隔(Periodic frame interval) 指示一帧内必须使用周期性帧中断通知应用程序的时间点。此功能可用于确定该帧的所有同步通信是否完成。 00: 80%帧间隔 01: 85%帧间隔 10: 90%帧间隔 11: 95%帧间隔	R/W
b10~b4	DAD	设备地址	设备地址(Device address) 应用程序必须在执行每个SetAddress控制命令后根据命令参数对该字段进行设置。	R/W
b3	Reserved	-	必须保持复位值。	R

b2	NZLSOHSK	非零长度状态OUT 握手信号	非零长度状态OUT握手信号 (Non-zero-length status OUT handshake) 在控制传输状态阶段的OUT事务期间，当模块收到非零长度数据包后，应用程序可以使用此字段选择要发送的握手信号。 1: 收到非零长度状态OUT事务时，回复STALL握手信号，收到的 OUT 数据包不发送给应用程序。 0: 将收到的OUT数据包（零长度或非零长度）发送给应用程序，并基于设备端点控制寄存器中端点的NAK和STALL位回复握手信号。	R/W
b1~b0	DSPD	设备速度	设备速度 (Device speed) 指示应用程序要求模块进行枚举所采用的速度，或应用程序支持的最大速度。 但是，实际总线速度只有在完成chirp序列后才能确定，同时此速度基于与模块连接的USB主机的速度。 00: 保留 01: 保留 10: 保留 11: 全速（USB 1.1收发器时钟为48 MHz）	R/W

31.7.4.2 USBFS 设备控制寄存器(USBFS_DCTL)

USBFS Device control register

偏移地址: 0x804

复位值: 0x0000 0000

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

Reserved	POPRGDNE	CGONAK	SGONAK	CGINAK	SGINAK	Reserved	GONSTS	GINSTS	SDIS	RWUSIG
----------	----------	--------	--------	--------	--------	----------	--------	--------	------	--------

位	标记	位名	功能	读写
b31~b12	Reserved	-	必须保持复位值。	R
b11	POPRGDNE	上电编程完成	上电编程完成(Power-on programming done) 应用程序使用此位指示寄存器从掉电模式唤醒后已完成编程。	R/W
b10	CGONAK	清零全局OUT NAK	清零全局OUT NAK(Clear global OUT NAK) 对此位执行写操作会将全局OUT NAK清零。	W
b9	SGONAK	置位全局OUT NAK	置位全局OUT NAK(Set global OUT NAK) 对此位执行写操作会将全局OUT NAK置1。 应用程序使用此位在所有OUT端点发送NAK握手信号。 应用程序只有确定模块中断寄存器中全局OUT NAK有效位 (USBFS_GINTSTS中GONAKEFF位)已清零时,才可以将此位置1。	W
b8	CGINAK	清零全局IN NAK	清零全局IN NAK (Clear global IN NAK) 对此位执行写操作会将全局IN NAK清零。	W
b7	SGINAK	置位全局IN NAK	置位全局IN NAK (Set global IN NAK) 对此字段执行写操作会将全局非周期性IN NAK置1。应用程序使用此位使所有非周期性IN端点发送NAK握手信号。 应用程序只有确定模块中断寄存器中全局IN NAK有效位(USBFS_GINTSTS 中GINAKEFF位)已清零时,才可以将此位置1。	W
b6~b4	Reserved	-	必须保持复位值。	R
b3	GONSTS	全局OUT NAK状态	全局OUT NAK状态(Global OUT NAK status) 0: 将根据FIFO状态和NAK和STALL位设置发送握手信号。 1: 无论Rx FIFO中是否还有空闲空间都不接收数据。除SETUP事务之外,对所有收到的数据包回复NAK握手信号。所有同步类型的OUT数据包都将被丢弃。	R
b2	GINSTS	全局IN NAK状态	全局IN NAK状态(Global IN NAK status) 0: 将根据发送FIFO中的数据可用性回复握手信号。 1: 使所有非周期性IN端点回复NAK握手信号,无需考虑发送FIFO中的数据可用性。	R

b1	SDIS	软断连	软断连(Soft disconnect) 应用程序使用该位向USBFS模块发出执行软断开的信号。该位置1时，主机不会看到设备已连接，且该设备也不会接收USB上的信号。在应用程序将此位清零之前，模块会保持断开状态。 0：正常工作。此位在软断连之后清零，会使主机收到设备已连接的事件。 重新连接设备之后，USB主机会重新启动设备枚举。 1：使主机收到设备断开连接的事件。	R/W
b0	RWUSIG	发送远程唤醒信号	发送远程唤醒信号(Remote wakeup signaling) 应用程序将此位置1时，模块会启动远程唤醒信号，以唤醒USB主机。应用程序必须将此位置1以使模块退出挂起状态。根据USB 2.0规范，应用程序必须在将此位置1之后的1 ms到15 ms内将其清零。	R/W

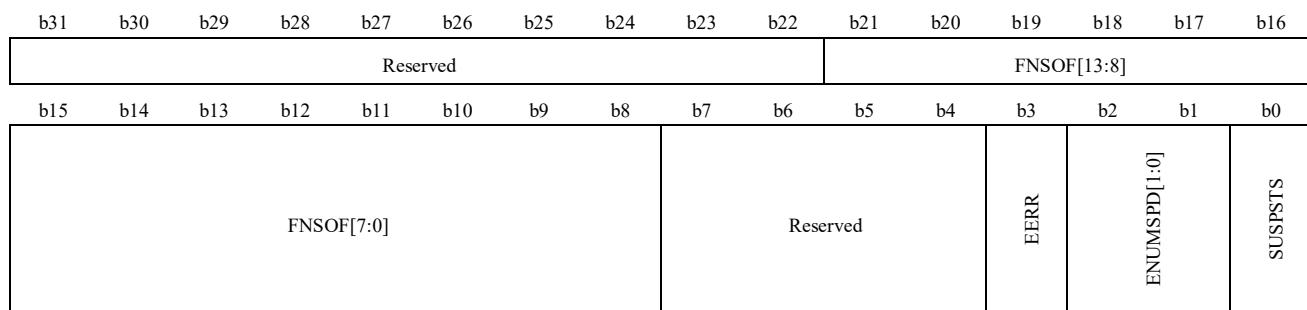
31.7.4.3 USBFS 设备状态寄存器(USBFS_DSTS)

USBFS Device status register

偏移地址: 0x808

复位值: 0x0000 0002

此寄存器指示模块在出现 USB 相关事件时的状态。发生中断时，必须从设备全体中断(USBFS_DAINT) 寄存器读取发生中断的端点信息。



位	标记	位名	功能	读写
b31~b22	Reserved	-	必须保持复位值。	R
b21~b8	FNSOF	接收SOF的帧编号	接收SOF的帧编号 (Frame number of the received SOF)	R
b7~b4	Reserved	-	必须保持复位值。	R
b3	EERR	不定错误	不定错误(Erratic error) 模块将该位置1以报告任何不定错误。 由于不定错误，USBFS控制器会进入挂起状态，并且会USBFS_GINTSTS寄存器的早期挂起位(USBFS_GINTSTS中的ESUSP 位)生成一个中断。如果早期挂起中断是由不定错误触发，则应用程序只能执行软断开以恢复通信。	R
b2~b1	ENUMSPD	枚举速度	枚举速度(Enumerated speed) 指示USBFS控制器通过chirp序列检测速度后被枚举成的速度。 01: 保留 10: 保留 11: 全速 (PHY 时钟运行频率为 48 MHz) 其它值: 保留	R
b0	SUSPSTS	挂起状态	挂起状态(Suspend status) 在设备模式下，只要在USB上检测到挂起状态，该位就会置1。当USB 总线上的空闲状态保持3ms，模块便会进入挂起状态。出现以下情况时，模块会退出挂起状态： — USB 数据线上有活动 — 应用程序对USBFS_DCTL寄存器的远程唤醒信号位(USBFS_DCTL中的RWUSIG位)执行写操作。	R

31.7.4.4 USBFS 设备 IN 端点通用中断屏蔽寄存器(USBFS_DIEPMSK)

USBFS Device IN endpoint common interrupt mask register

偏移地址: 0x810

复位值: 0x0000 0000

此寄存器与全体端点的各个 USBFS_DIEPINTx 寄存器配合使用，以便在每个 IN 端点上生成中断。通过对对此寄存器的相应位执行写操作，可屏蔽 USBFS_DIEPINTx 寄存器中的 IN 端点中断。默认情况下，状态中断都被屏蔽。

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

位	标记	位名	功能	读写
b31~b7	Reserved	-	必须保持复位值。	R
b6	INEPNEM	IN端点NAK有效中断屏蔽	IN端点NAK有效中断屏蔽 (IN endpoint NAK effective mask) 0: 屏蔽中断 1: 使能中断	R/W
b5	INEPNMM	EP不匹配时接收到IN令牌中断屏蔽	EP不匹配时接收到IN令牌中断屏蔽 (IN token received with EP mismatch mask) 0: 屏蔽中断 1: 使能中断	R/W
b4	ITTXFEMSK	TxFIFO为空时接收到IN令牌中断屏蔽	TxFIFO为空时接收到IN令牌中断屏蔽 (IN token received when TxFIFO empty mask) 0: 屏蔽中断 1: 使能中断	R/W
b3	TOM	超时中断屏蔽 (非同步端点)	超时中断屏蔽 (非同步端点) (Timeout condition mask (Non-isochronous endpoints)) 0: 屏蔽中断 1: 使能中断	R/W
b2	Reserved	-	必须保持复位值。	R
b1	EPDM	端点禁止中断屏蔽	端点禁止中断屏蔽(Endpoint disabled interrupt mask) 0: 屏蔽中断 1: 使能中断	R/W

b0	XFRCM	传输完成中断屏蔽	传输完成中断屏蔽(Transfer completed interrupt mask)	R/W
		0:	屏蔽中断	
		1:	使能中断	

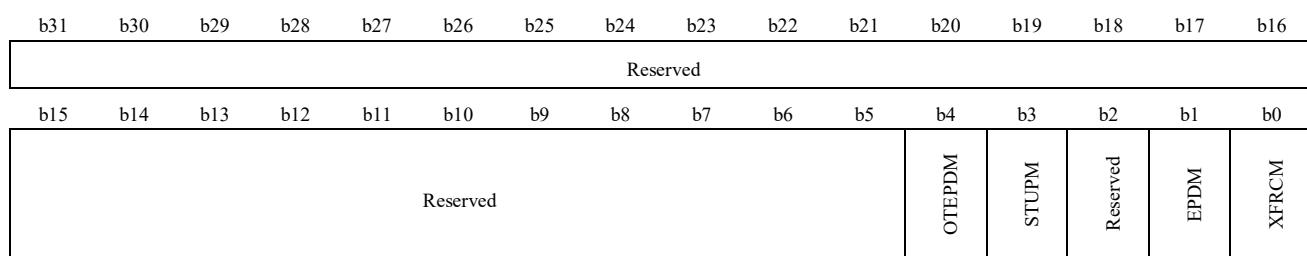
31.7.4.5 USBFS 设备 OUT 端点通用中断屏蔽寄存器(USBFS_DOEPMSK)

USBFS Device OUT endpoint common interrupt mask register

偏移地址: 0x814

复位值: 0x0000 0000

此寄存器与全体端点的各个 USBFS_DOEPINTx 寄存器配合使用, 以便在每个 OUT 端点上生成中断。通过对此寄存器的相应位执行写操作, 可屏蔽 USBFS_DOEPINTx 寄存器中的 OUT 端点中断。默认情况下, 状态中断都被屏蔽。



位	标记	位名	功能	读写
b31~b5	Reserved	-	必须保持复位值。	R
b4	OTEPM	端点禁止时接收到 OUT令牌中断屏蔽	端点禁止时接收到OUT令牌中断屏蔽 (OUT token received when endpoint disabled mask) 仅适用于控制 OUT 端点。 0: 屏蔽中断 1: 使能中断	R/W
b3	STUPM	SETUP阶段完成中 断屏蔽	SETUP阶段完成中断屏蔽(SETUP phase done mask) 仅适用于控制端点。 0: 屏蔽中断 1: 使能中断	R/W
b2	Reserved	-	必须保持复位值。	R
b1	EPDM	端点禁止中断屏蔽	端点禁止中断屏蔽(Endpoint disabled interrupt mask) 0: 屏蔽中断 1: 使能中断	R/W
b0	XFRCM	传输完成中断屏蔽	传输完成中断屏蔽(Transfer completed interrupt mask) 0: 屏蔽中断 1: 使能中断	R/W

31.7.4.6 USBFS 设备全体端点中断寄存器 (USBFS_DAINT)

USBFS Device OUT endpoint common interrupt mask register

偏移地址: 0x818

复位值: 0x0000 0000

当端点上发生有效事件时, USBFS_DAINT 寄存器将通过 USBFS_GINTSTS 寄存器中的设备 OUT 端点中断位或设备 IN 端点中断位(分别为 USBFS_GINTSTS 中的 OEPINT 或 IEPINT 位)来中断应用程序。每个端点对应一个中断位, OUT 端点和 IN 端点均最多有 16 个中断位。双向端点将使用相应的 IN 和 OUT 中断位。当应用程序将相应设备端点 x 中断寄存器(USBFS_DIEPINTx/USBFS_DOEPINTx)中的位置 1 和清零时, 此寄存器中的相应位也将置 1 和清零。

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

位	标记	位名	功能	读写
b31~b22	Reserved	-	必须保持复位值。	R
b21~b16	OEPINT	OUT端点中断位	OUT端点中断位(OUT endpoint interrupt bits) 每个OUT端点对应一位: OUT端点0对应位16, 而OUT端点5对应位21。	R/W
b15~b6	Reserved	-	必须保持复位值。	R
b5~b0	IEPINT	IN端点中断位	IN端点中断位(IN endpoint interrupt bits) 每个IN端点对应一位: IN端点0对应位0, 而IN端点5对应位5。	R/W

31.7.4.7 USBFS 设备全体端点中断屏蔽寄存器 (USBFS_DAIINTMSK)

USBFS Device all endpoints interrupt mask register

偏移地址: 0x81C

复位值: 0x0000 0000

USBFS_DAIINTMSK 寄存器与设备端点中断寄存器结合使用，在设备端点上发生事件时中断应用程序。但是，与该中断相对应的 USBFS_DAIINT 寄存器位仍会置 1。

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

位	标记	位名	功能	读写
b31~b22	Reserved	-	必须保持复位值。	R
b21~b16	OEPINTM	OUT端点中断屏蔽位	OUT端点中断屏蔽位(OUT endpoint interrupt mask bits) 每个OUT端点对应一位: OUT端点0对应位16，而OUT端点5对应位21。 0: 屏蔽中断 1: 使能中断	R/W
b15~b6	Reserved	-	必须保持复位值。	R
b5~b0	IEPINTM	IN端点中断屏蔽位	IN端点中断位(IN endpoint interrupt mask bits) 每个IN端点对应一位: IN端点0对应位0，而IN端点5对应位5。 0: 屏蔽中断 1: 使能中断	R/W

31.7.4.8 USBFS 设备 IN 端点 FIFO 空中断屏蔽寄存器(USBFS_DIEPEMPMSK)

USBFS Device IN endpoint FIFO empty interrupt mask register

偏移地址: 0x834

复位值: 0x0000 0000

此寄存器用于控制 IN 端点 FIFO 空中断的生成。

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

位	标记	位名	功能	读写
b31~b6	Reserved	-	必须保持复位值。	R
b5~b0	INEPTXFEM	IN EP TxFIFO空中 断屏蔽位	IN EP TxFIFO空中断屏蔽位 (IN EP Tx FIFO empty interrupt mask bits) 这些位用作USBFS_DIEPINTx的屏蔽位。 每个位对应一个IN端点的TXFE中断： IN端点0对应位0，而IN端点5对应位5 0: 屏蔽中断 1: 使能中断	R/W

31.7.4.9 USBFS 设备控制 IN 端点 0 控制寄存器(USBFS_DIEPCTL0)

USBFS Device control IN endpoint 0 control register

偏移地址: 0x900

复位值: 0x0000 8000

此寄存器用于控制控制传输端点 0。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EPENA	EPDIS	Reserved		SNAK	CNAK					STALL	Reserved	EPTYP[1:0]		NAKSTS	Reserved
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
USBAEP														MPSIZ[1:0]	

位	标记	位名	功能	读写
b31	EPENA	端点使能	端点使能(Endpoint enable) 应用程序将此位置1以在端点0上启动数据发送。 在此端点上触发以下任一中断之前，模块会将此位清零： — 端点禁止 — 传输完成	R/W
b30	EPDIS	端点禁止	端点禁止(Endpoint disable) 即使在该端点上的传输完成之前，应用程序也可将此位置1，以停止端点上的数据发送。应用程序必须等到发生端点禁止中断后，才能将端点视为禁止端点。在端点禁止中断位置1前，模块会将此位清零。只有在该端点的端点使能位置1后，应用程序才可将该位置 1。	R/W
b29~b28	Reserved	-	必须保持复位值。	R
b27	SNAK	置位NAK位	置位NAK位(Set NAK) 对此位进行写操作会将端点的NAK位置1。 通过此位，应用程序可以控制端点上NAK握手信号的发送。模块也可在端点接收到SETUP数据包后将该端点的此位置1。	R/W
b26	CNAK	清零NAK位	清零NAK位(Clear NAK) 对此位进行写操作会将端点的NAK位清零。	R/W
b25~b22	TXFNUM	TxFIFO编号	TxFIFO编号(TxFIFO number) 该值设置为分配给IN端点0的FIFO编号。只能使用TX-FIFO0。	R/W
b21	STALL	STALL握手	STALL握手(STALL handshake) 应用程序只能将此位置1，端点接收到SETUP令牌时，模块会将此位清零。如果NAK位、全局IN NAK或全局OUT NAK与此位均置 1，则STALL位优先。	R/W

b20	Reserved	-	必须保持复位值。	R
b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 硬件设置为‘00’，表示控制类型的端点。	R
b17	NAKSTS	NAK状态	NAK状态(NAK status) 指示以下结果： 0：模块根据FIFO状态回复非NAK握手。 1：模块在此端点上回复NAK握手。 当此位置1时（无论是被应用程序还是被模块），即使TxFIFO中仍有数据可用，模块也会停止发送数据。无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R
b16	Reserved	-	必须保持复位值。	R
b15	USBAEP	USB活动端点	USB活动端点(USB active endpoint) 此位总是置1，指示在所有配置和接口中控制端点0始终处于激活状态。	R
b14~b2	Reserved	-	必须保持复位值。	R
b1~b0	MPSIZ	最大数据包大小	最大数据包大小(Maximum packet size) 应用程序必须将此字段编程为当前逻辑端点的最大数据包大小。 00: 64字节 01: 32字节 10: 16字节 11: 8字节	R/W

31.7.4.10 USBFS 设备 IN 端点 x 控制寄存器(USBFS_DIEPCTLx)(x=1..5)

USBFS Device IN endpoint x control register

偏移地址: 0x900 + (端点编号 × 0x20)

复位值: 0x0000 0080

应用程序使用此寄存器控制各个逻辑端点（端点 0 除外）的行为。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EPENA	EPDIS	SODDFRM	SD0PID/ SEVNFRM	SNAK	CNAK	TXFNUM[3:0]				STALL	Reserved	EPTYP[1:0]		NAKSTS	EONUM/ DPID
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
USBAEP	Reserved				MPSIZ[10:0]										

位	标记	位名	功能	读写
b31	EPENA	端点使能	端点使能(Endpoint enable) 应用程序将此位置1以在端点上启动数据发送。 在此端点上触发以下任一中断之前，模块会将此位清零： — SETUP阶段完成 — 端点禁止 — 传输完成	R/W
b30	EPDIS	端点禁止	端点禁止(Endpoint disable) 即使在该端点上的传输完成之前，应用程序也可将此位置1，以停止端点上的数据发送。应用程序必须等到发生端点禁止中断后，才能将端点视为禁止端点。在端点禁止中断位置1前，模块会将此位清零。只有在该端点的端点使能位置1后，应用程序才可将该位置 1。	R/W
b29	SODDFRM	设置奇数帧	设置奇数帧(Set odd frame) 仅适用于同步IN和OUT端点。 对此字段进行写操作会将偶数/奇数帧 (EONUM) 字段设置为奇数帧。	R/W
b28	SD0PID/ SEVNFRM	设置DATA0 PID/ SEVNFRM	设置DATA0 PID (Set DATA0 PID) 仅适用于中断/批量IN 端点。 对此字段进行写操作会将此寄存器中的端点数据PID(DPID)字段设置为DATA0。 SEVNFRM: 设置偶数帧 (Set even frame) 仅适用于同步IN端点。 对此字段进行写操作会将偶数/奇数帧(EONUM)字段设置为偶数帧。	R/W

b27	SNAK	置位NAK位	置位NAK位(Set NAK) 对此位进行写操作会将端点的NAK位置 1。 通过此位，应用程序可以控制端点上NAK握手信号的发送。发生传输完成中断时或端点上接收到SETUP后，模块也可以将OUT端点的这个位置 1	R/W
b26	CNAK	清零NAK位	清零NAK位(Clear NAK) 对此位进行写操作会将端点的NAK位清零。	R/W
b25~b22	TXFNUM	TxFIFO编号	TxFIFO编号(TxFIFO number) 这些位用于指定与此端点相关联的 FIFO 编号。必须为每个有效的IN端点设置单独的 FIFO编号。 此字段仅针对IN端点有效。	R/W
b21	STALL	STALL握手	STALL握手(STALL handshake) 应用程序将此位置1使得设备对来自 USB 主机的所有令牌都回复STALL。如果NAK位、全局IN NAK或全局OUT NAK与此位同时置 1，则STALL位优先。只有应用程序能够将此位清零，而模块则不能。	R/W
b20	Reserved	-	必须保持复位值。	R
b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 以下是这个逻辑端点支持的传输类型。 00: 控制 01: 同步 10: 批量 11: 中断	R
b17	NAKSTS	NAK状态	NAK状态(NAK status) 指示以下结果： 0: 模块根据FIFO状态回复非NAK握手。 1: 模块在此端点上回复NAK握手。 当应用程序或模块将此位置1时： 对于非同步IN端点：即使TxFIFO中存在可用数据，模块也会停止通过 IN端点发送任何数据。 对于同步IN端点：即使TxFIFO中存在可用数据，模块也会发送长度为零的数据包。 无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R

b16	EONUM/ DPID	偶数/奇数帧/ 端点数据PID	偶数/奇数帧 (Even/odd frame) 仅适用于同步IN端点。 指示模块为此端点发送/接收同步的数据所在的帧的编号。应用程序必须通过此寄存器中的SEVNFRM和SODDFRM字段对偶数/奇数帧编号进行编程，以便此端点发送/接收同步数据。 0: 偶数帧 1: 奇数帧	R/W
			<p>DPID: 端点数据PID (Endpoint data PID)</p> <p>仅适用于中断/批量IN端点。</p> <p>包含此端点上将要接收或发送的数据包的PID。端点激活后，应用程序必须对要在此端点上接收或发送的首个数据包的PID进行编程。应用程序使用SD0PID寄存器字段对DATA0或DATA1 PID进行编程。</p> <p>0: DATA0 1: DATA1</p>	
b15	USBAEP	USB活动端点	USB活动端点(USB active endpoint) 指示此端点在当前配置和接口中是否激活。检测到USB复位后，模块会为所有端点（端点0除外）将此位清零。接收到SetConfiguration和 SetInterface命令后，应用程序必须相应地对端点寄存器进行编程并将此位置1。	R/W
b14~b11	Reserved	-	必须保持复位值。	R/W
b10~b0	MPSIZ	最大数据包大小	最大数据包大小(Maximum packet size) 应用程序必须将此字段编程为当前逻辑端点的最大数据包大小。 此值以字节为单位。	

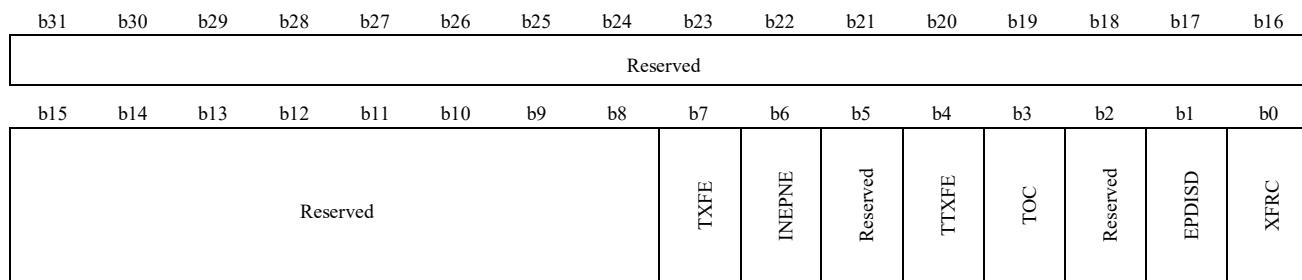
31.7.4.11 USBFS 设备 IN 端点 x 中断寄存器 (USBFS_DIEPINTx) (x=0..5)

USBFS Device IN endpoint x interrupt register

偏移地址: 0x908 + (端点编号 × 0x20)

复位值: 0x0000 0000

此寄存器指示端点在出现 USB 和 AHB 相关事件时的状态。当模块中断寄存器中的 IN 端点中断位 (USBFS_GINTSTS 中的 IEPINT 位) 置 1 时, 应用程序必须读取此寄存器。在应用程序能够读取此寄存器之前, 必须先读取设备全体端点中断 (USBFS_DAINT) 寄存器, 以获取设备端点 x 中断寄存器的准确端点编号。应用程序必须将此寄存器中的相应位清零, 才能将 USBFS_DAINT 和 USBFS_GINTSTS 寄存器中的对应位清零。



位	标记	位名	功能	读写
b31~b8	Reserved	-	必须保持复位值。	R
b7	TXFE	发送FIFO为空	发送FIFO为空 (Transmit FIFO empty) 当此端点的TxFIFO为半空或全空时, 此中断被置位。TxFIFO为半空还是全空状态由USBFS_GAHBCFG寄存器中的TxFIFO空白位 (USBFS_GAHBCFG中的TXFELVL位) 决定。	R
b6	INPNE	IN端点NAK有效	INPNE: IN端点NAK有效 (IN endpoint NAK effective) 当应用程序通过向USBFS_DIEPCTLx 中的CNAK位写入数据来将IN端点NAK清零时, 此位可被清零。 该中断指示模块已对 (由应用程序或模块) 置1的NAK采样, 结果已生效。该中断指示由应用程序置1的IN端点NAK位已在模块中起作用。 此中断不保证在USB上发送了NAK握手信号。 STALL位的优先级高于 NAK 位。 软件写1也可将此位清零。	R/W
b5	Reserved	-	必须保持复位值。	R

b4	TTXFE	TxFIFO为空时接收 到IN令牌	TxFIFO为空时接收到IN令牌 (IN token received when TxFIFO is empty) 仅适用于非周期性 IN 端点。 当和该端点相对应的TxFIFO (周期性/非周期性) 为空时, 接收到IN令牌, 从而产生中断。 通过软件写1清零。	R/W
b3	TOC	超时	超时条件(Timeout condition) 仅适用于控制IN端点。 指示该端点对最近收到的IN令牌响应超时。 通过软件写1清零。	R/W
b2	Reserved	-	必须保持复位值。	R
b1	EPDISD	端点禁止中断	端点禁止中断 (Endpoint disabled interrupt) 此位指示该端点已经由应用程序禁止掉。 通过软件写1清零。	R/W
b0	XFRC	传输完成中断	传输完成中断 (Transfer completed interrupt) 此字段指示在此端点上设置的传输已经在USB和AHB上传输完成。 通过软件写1清零。	R/W

31.7.4.12 USBFS 设备 IN 端点 0 传输大小寄存器(USBFS_DIEPTSIZ0)

USBFS Device IN endpoint 0 transfer size register

偏移地址: 0x910

复位值: 0x0000 0000

在使能端点 0 之前，应用程序必须修改此寄存器。通过设备控制端点 0 控制寄存器中的端点使能位（USBFS_DIEPCTL0 中的 EPENA）使能端点 0 后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

非零端点使用端点 1~5 的寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved										PKTCNT[1:0]	Reserved				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								XFRSIZ[6:0]							

位	标记	位名	功能	读写
b31~b21	Reserved	-	必须保持复位值。	R
b20~b19	PKTCNT	数据包计数	数据包计数 (Packet count) 指示端点0的一次数据传输包含的数据包个数。 每次从TxFIFO读取数据包（最大或短数据包）时，此字段将递减。	R/W
b18~b7	Reserved	-	必须保持复位值。	R
b6~b0	XFRSIZ	传输大小	传输大小(Transfer size) 指示端点0的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以便在每个数据包结束时中断。 每次向TxFIFO写入来自外部存储器的数据包时，模块会使此字段递减。	R/W

31.7.4.13 USBFS 设备 IN 端点 x 传输大小寄存器(USBFS_DIEPTSIZx)(x=1..5)

USBFS Device IN endpoint x transfer size register

偏移地址: 0x910 + (端点编号 × 0x20)

复位值: 0x0000 0000

在使能该端点之前，应用程序必须修改此寄存器。通过 USBFS_DIEPCTLx 寄存器中的端点使能位（USBFS_DIEPCTLx 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved		PKTCNT[9:0]										XFRSIZ[18:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XFRSIZ[15:0]															

位	标记	位名	功能	读写
b31~b29	Reserved	-	必须保持复位值。	R
b28~b19	PKTCNT	数据包计数	数据包计数 (Packet count) 指示该端点上的一次数据传输包含的数据包个数。 每次从TxFIFO读取数据包（最大大小或短数据包）时，此字段将递减。	R/W
b18~b0	XFRSIZ	传输大小	传输大小(Transfer size) 此字段包含当前端点的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。 每次向TxFIFO写入来自外部存储器的数据包时，模块会使此字段递减。	R/W

31.7.4.14 USBFS 设备 IN 端点 x DMA 地址寄存器 (USBFS_DIEPDMAx)(x=0..5)

USBFS Device IN endpoint x transfer size register

偏移地址: 0x914 + (端点编号 × 0x20)

复位值: 0x0000 0000

该寄存器用于设定设备端点 DMA 模式时 DMA 地址。

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

位	标记	位名	功能	读写
b31~b0	DMAADDR	DMA地址	DMA地址(DMA address) 该位包含使用DMA进行端点上数据存储时的外部存储区起始地址。 注意: 对于控制端点, 该字段所指向的存储区也用于存储控制OUT数据包以及SETUP事务数据包。连续接收到三个以上的SETUP数据包时, 存储器中的SETUP数据包将被覆盖。每次进行AHB传输, 该寄存器都会递增。应用程序必须设定一个双字对齐地址。	R/W

31.7.4.15 USBFS 设备 IN 端点发送 FIFO 状态寄存器 (USBFS_DTXFSTSx)(x=0..5)

USBFS Device IN endpoint transmit FIFO status register

偏移地址: 0x918 + (端点编号 × 0x20)

复位值: 0x0000 0000

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	必须保持复位值。	R
b15~b0	INEPTFSAV	IN端点TxFIFO可用空间	IN端点TxFIFO可用空间 (IN endpoint TxFIFO space available)	R
		空间	指示端点 TxFIFO 中的可用空闲空间大小。 以32位字为单位: 0x0: 端点TxFIFO已满 0x1: 1个字可用 0x2: 2个字可用 0xn: n个字可用	

31.7.4.16 USBFS 设备控制 OUT 端点 0 控制寄存器(USBFS_DOEPCTL0)

USBFS Device control OUT endpoint 0 control register

偏移地址: 0xB00

复位值: 0x0000 8000

此寄存器用于控制控制传输端点 0。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EPENA	EPDIS	Reserved		SNAK	CNAK			Reserved		STALL	SNPM	EPTYP[1:0]	NAKSTS		Reserved
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
USBAEP								Reserved					MPSIZ[1:0]		

位	标记	位名	功能	读写
b31	EPENA	端点使能	端点使能(Endpoint enable) 应用程序将此位置1以在端点0上启动数据接收。 在此端点上触发以下任一中断之前，模块会将此位清零： — SETUP 阶段完成 — 端点禁止 — 传输完成	R/W
b30	EPDIS	端点禁止	端点禁止(Endpoint disable) 应用程序无法禁止控制OUT端点0。	R
b29~b28	Reserved	-	必须保持复位值。	R
b27	SNAK	置位NAK位	置位NAK位(Set NAK) 对此位进行写操作会将端点的NAK位置1。 通过此位，应用程序可以控制端点上NAK握手信号的发送。模块也可在端点接收到SETUP数据包后将该端点的此位置1。	R/W
b26	CNAK	清零NAK位	清零NAK位(Clear NAK) 对此位进行写操作会将端点的NAK位清零。	R/W
b25~b22	Reserved	-	必须保持复位值。	R
b21	STALL	STALL握手	STALL握手(STALL handshake) 此端点接收到SETUP令牌时，应用程序只能将此位置1，而模块会将其清零。 如果NAK位、全局OUT NAK与此位同时置1，则STALL位优先。无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R/W
b20	SNPM	监听模式	监听模式 (Snoop mode) 此位用于将端点配置为监听模式。在监听模式下，模块不会在将OUT数据包传输到应用存储区前检查其是否正确。	R/W

b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 硬件设置为‘00’，表示控制类型的端点。	R/W
b17	NAKSTS	NAK状态	NAK状态(NAK status) 指示以下结果： 0：模块根据FIFO状态回复非NAK握手。 1：模块在此端点上回复NAK握手。 当应用程序或模块将此位置1时，即使Rx FIFO中存在空间可继续容纳收到的数据包，模块也会停止接收数据。无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R
b16	Reserved	-	必须保持复位值。	R
b15	USBAEP	USB活动端点	USB活动端点(USB active endpoint) 此位总是置1，指示在所有配置和接口中控制端点0始终处于激活状态。	R
b14~b2	Reserved	-	必须保持复位值。	R
b1~b0	MPSIZ	最大数据包大小	最大数据包大小(Maximum packet size) 控制OUT端点0的最大数据包大小与在控制IN端点0中进行编程的值相同。 00: 64 字节 01: 32 字节 10: 16 字节 11: 8 字节	R/W

31.7.4.17 USBFS 设备 OUT 端点 x 控制寄存器(USBFS_DOEPCTLx)(x=1..5)

USBFS Device OUT endpoint x control register

偏移地址: 0xB00 + (端点编号 × 0x20)

复位值: 0x0000 0000

应用程序使用此寄存器控制各个逻辑端点（端点 0 除外）的行为。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EPENA	EPDIS	SODDFRM/ SD1PID	SD0PID/ SEVNFRM	SNAK	CNAK	Reserved				STALL	SNPM	EPTYP[1:0]	NAKSTS	EONUM/ DPID	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
USBAEP	Reserved				MPSIZ[10:0]										

位	标记	位名	功能	读写
b31	EPENA	端点使能	端点使能(Endpoint enable) 软件置位, USBFS清零 0: 端点除能 1: 端点使能	R/W
b30	EPDIS	端点禁止	端点禁止(Endpoint disable) 即使在该端点上的传送完成之前, 应用程序也可将此位置1, 以停止端点上的数据发送/接收。应用程序必须等到发生端点禁止中断后, 才能将端点视为禁止端点。在端点禁止中断位置1前, 模块会将此位清零。只有在该端点的端点使能位置1后, 应用程序才可将该位置1。	R/W
b29	SD1PID/ SODDFRM	设置DATA1 PID/ 设置奇数帧	设置DATA1 PID (Set DATA1 PID) 仅适用于中断/批量OUT端点。对此字段进行写操作会将此寄存器中的端点数据PID(DPID) 字段设置为DATA1。 SODDFRM: 设置奇数帧 (Set odd frame) 仅适用于同步OUT端点。对此字段进行写操作会将偶数/奇数帧 (EONUM) 字段设置为奇数帧。	R
b28	SD0PID/ SEVNFRM	设置DATA0 PID/ SEVNFRM	设置DATA0 PID (Set DATA0 PID) 仅适用于中断/批量OUT端点。 对此字段进行写操作会将此寄存器中的端点数据PID(DPID)字段设置为DATA0。 SEVNFRM: 设置偶数帧 (Set even frame) 仅适用于同步OUT端点。 对此字段进行写操作会将偶数/奇数帧(EONUM)字段设置为偶数帧。	R

b27	SNAK	置位NAK位	置位NAK位(Set NAK) 对此位进行写操作会将端点的NAK位置1。 通过此位，应用程序可以控制端点上NAK握手信号的发送。发生传输完成中断时或端点上接收到SETUP后，模块也可以将OUT端点的这个位置1。	R/W
b26	CNAK	清零NAK位	清零NAK位(Clear NAK) 对此位进行写操作会将端点的NAK位清零。	R/W
b25~b22	Reserved	-	必须保持复位值。	R
b21	STALL	STALL握手	STALL握手(STALL handshake) 此端点接收到SETUP令牌时，应用程序只能将此位置1，而模块会将其清零。如果NAK位、全局OUT NAK与此位同时置1，则STALL位优先。只有应用程序能够将此位清零，而模块则不能。	R/W
b20	SNPM	监听模式	监听模式 (Snoop mode) 此位用于将端点配置为监听模式。在监听模式下，模块不会再检查接收数据的正确性。	R/W
b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 以下是这个逻辑端点支持的传输类型。 00: 控制 01: 同步 10: 批量 11: 中断	R/W
b17	NAKSTS	NAK状态	NAK状态 (NAK status) 指示以下结果： 0: 模块根据FIFO状态回复非NAK握手。 1: 模块在此端点上回复NAK握手。 当应用程序或模块将此位置1时： 即使RxFIFO存在空间可容纳传入数据包，模块也会停止在OUT端点上接收任何数据。 无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R

b16	EONUM/ DPID	偶数/奇数帧/ 端点数据PID	偶数/奇数帧 (Even/odd frame) 仅适用于同步OUT 端点。 指示模块为此端点发送/接收同步的数据所在的帧的编号。应用程序必须通过此寄存器中的SEVNFRM和SODDFRM字段对偶数/奇数帧编号进行编程，以便此端点发送/接收同步数据。 0: 偶数帧 1: 奇数帧	R/W
b15	USBAEP	USB活动端点	USB活动端点(USB active endpoint) 指示此端点在当前配置和接口中是否激活。检测到USB复位后，模块会为所有端点（端点0除外）将此位清零。接收到SetConfiguration和 SetInterface命令后，应用程序必须相应地对端点寄存器进行编程并将此位置1。	R/W
b14~b11	Reserved	-	必须保持复位值。	R/W
b10~b0	MPSIZ	最大数据包大小	最大数据包大小(Maximum packet size) 应用程序必须将此字段编程为当前逻辑端点的最大数据包大小。 此值以字节为单位。	R/W

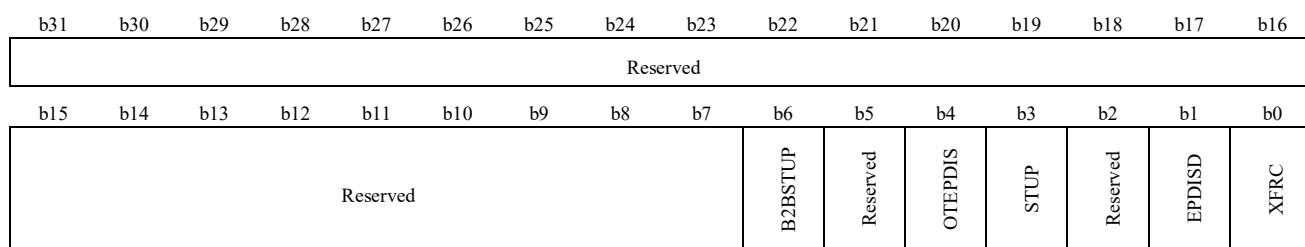
31.7.4.18 USBFS 设备 OUT 端点 x 中断寄存器 (USBFS_DOEPINTx) (x=0..5)

USBFS Device OUT endpoint x interrupt register

偏移地址: 0xb08 + (端点编号 × 0x20)

复位值: 0x0000 0080

此寄存器指示端点在出现 USB 和 AHB 相关事件时的状态。当 USBFS_GINTSTS 寄存器中的 OUT 端点中断位 (USBFS_GINTSTS 中的 OEPINT 位) 置 1 时, 应用程序必须读取此寄存器。在应用程序能够读取此寄存器之前, 必须先读取 USBFS_DAINT 寄存器, 以获取 USBFS_DOEPINTx 寄存器的准确端点编号。应用程序必须将此寄存器中的相应位清零, 才能将 USBFS_DAINT 和 USBFS_GINTSTS 寄存器中的对应位清零。



位	标记	位名	功能	读写
b31~b7	Reserved	-	必须保持复位值。	R
b6	B2BSTUP	接收到连续的 SETUP数据包	接收到连续的SETUP数据包 (Back-to-back SETUP packets received) 仅适用于控制OUT端点。此位指示该端点已接收到三个以上的连续SETUP数据包。软件写1也可将此位清零。	R/W
b5	Reserved	-	必须保持复位值。	R
b4	OTEPDIS	端点禁止时接收到 OUT令牌	端点禁止时接收到OUT令牌 (OUT token received when endpoint disabled) 仅适用于控制OUT端点。指示在尚未使能端点时接收到OUT令牌, 从而产生中断。通过软件写1清零。	R/W
b3	STUP	SETUP阶段完成	SETUP 阶段完成 (SETUP phase done) 仅适用于控制OUT端点。指示控制端点的SETUP阶段已完成, 当前控制传输中不再接收到连续的 SETUP数据包。在此中断上, 应用程序可以对接收到的 SETUP数据包进行解码。 通过软件写1清零。	R/W
b2	Reserved	-	必须保持复位值。	R
b1	EPDISD	端点禁止中断	端点禁止中断 (Endpoint disabled interrupt) 此位指示该端点已经由应用程序禁止掉。 通过软件写1清零。	R/W

b0	XFRC	传输完成中断	传输完成中断 (Transfer completed interrupt) 此字段指示在此端点上设置的传输已经在USB和AHB上传输完成。 通过软件写1清零。	R/W
----	------	--------	---	-----

31.7.4.19 USBFS 设备 OUT 端点 0 传输大小寄存器(USBFS_DOEPTSIZ0)

USBFS Device OUT endpoint 0 transfer size register

偏移地址: 0xB10

复位值: 0x0000 0000

在使能端点 0 之前，应用程序必须修改此寄存器。通过设备控制端点 0 控制寄存器中的端点使能位（USBFS_DIEPCTL0 中的 EPENA）使能端点 0 后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

非零端点使用端点 1~5 的寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved	STUPCNT[1:0]		Reserved												PKTCNT
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved														XFRSIZ[6:0]	

位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R
b30~b29	STUPCNT	SETUP数据包计数	SETUP数据包计数 (SETUP packet count) 此字段指定端点能连续接收的SETUP数据包数量。 01: 1个数据包 10: 2个数据包 11: 3个数据包	R/W
b28~b20	Reserved	-	必须保持复位值。	R
b19	PKTCNT	数据包计数	数据包计数 (Packet count) 一次传输中应该接收到数据包数量。 在端点使能前，软件设置该位，在传输开始后，每当数据包接收到后，该域数值自动减少。	R/W
b18~b7	Reserved	-	必须保持复位值。	R
b6~b0	XFRSIZ	传输大小	传输大小(Transfer size) 指示端点0的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。 每次从Rx FIFO读取数据包并将其写入外部存储器时，模块会使此字段递减。	R/W

31.7.4.20 USBFS 设备 OUT 端点 x 传输大小寄存器(USBFS_DOEPTSIZx)(x=1..5)

USBFS Device OUT endpoint x transfer size register

偏移地址: 0xB10 + (端点编号 × 0x20)

复位值: 0x0000 0000

在使能该端点之前，应用程序必须修改此寄存器。通过 USBFS_DOEPCTLx 寄存器中的端点使能位（USBFS_DOEPCTLx 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved		PKTCNT[9:0]										XFRSIZ[18:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XFRSIZ[15:0]															

位	标记	位名	功能	读写
b31~b29	Reserved	-	必须保持复位值。	R
b28~b19	PKTCNT	数据包计数	数据包计数 (Packet count) 指示该端点上的一次数据传输包含的数据包个数。 每次向Rx FIFO写入数据包（最大大小或短数据包）后，此字段将递减。	R/W
b18~b0	XFRSIZ	传输大小	传输大小(Transfer size) 此字段包含当前端点的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以便在每个数据包结束时中断。 每次从Rx FIFO读取数据包并将其写入外部存储器时，模块会使此字段递减。	R/W

31.7.4.21 USBFS 设备 OUT 端点 x DMA 地址寄存器 (USBFS_DOEPMAX)(x=0..5)

USBFS Device OUT endpoint x transfer size register

偏移地址: 0xB14 + (端点编号 × 0x20)

复位值: 0xFFFF XXXX

该寄存器用于设定设备端点 DMA 模式时 DMA 地址。

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

位	标记	位名	功能	读写
b31~b0	DMAADDR	DMA地址	DMA地址(DMA address) 该位包含使用DMA进行端点上数据发送时的外部存储区起始地址。 注意: 对于控制端点, 该字段所指向的存储区也用于存储控制OUT数据包以及SETUP事务数据包。连续接收到三个以上的SETUP数据包时, 存储器中的SETUP数据包将被覆盖。每次进行AHB传输, 该寄存器都会递增。应用程序必须设定一个双字对齐地址。	R/W

31.7.5 USBFS 时钟门控控制寄存器

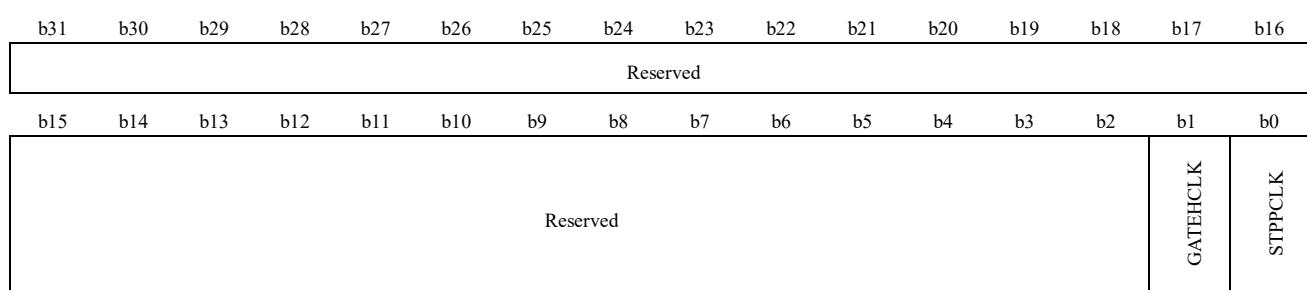
通过门控时钟控制寄存器控制 HCLK 和 PHY 时钟从而降低功耗。除非特别说明，否则寄存器描述中的位值以二进制表示。

31.7.5.1 USBFS 时钟门控控制寄存器(USBFS_GCCTL)

偏移地址：0xE00

复位值：0x0000 0000

此寄存器在主机模式和设备模式下均可用。



位	标记	位名	功能	读写
b31~b2	Reserved	-	必须保持复位值。	R
b1	GATEHCLK	门控HCLK	门控HCLK (Gate HCLK) 当USB通信挂起或会话无效时，应用程序会将此位置 1，以停止对除AHB总线从接口、主接口和唤醒逻辑之外的模块提供时钟。当USB恢复通信或新会话启动时，应用程序将此位清零。	R/W
b0	STPPCLK	停止PHY时钟	停止PHY时钟(Stop PHY clock) 当USB通信挂起、会话无效或设备断开连接时，应用程序将此位置1以停止PHY 时钟。当USB恢复通信时，应用程序将此位清零。	R/W

32 加密协处理模块（CPM）

32.1 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标准数据加密解密标准，可以实现 128 位密钥长度的加密运算和解密运算。

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 64bit 随机数。

32.2 加解密算法处理器（AES）

32.2.1 算法简介

AES 加解密算法是一个密钥迭代分组密码，包含了轮变换对状态的重复作用。加密过程的轮变换由四个操作组成：SubBytes、ShiftRows、MixColumns、AddRoundKey。其中，SubBytes 是求每个字节在 GF(2^8)中的模逆元和一个仿射变换；ShiftRows 是一个字节换位，它将状态中的行按照不同的偏移量进行循环移位；MixColumns 对状态各列进行线性变换；AddRoundKey 对状态中的各字节与轮密钥进行逐位异或操作。下图 32-1 所示是加密流程的示意图：

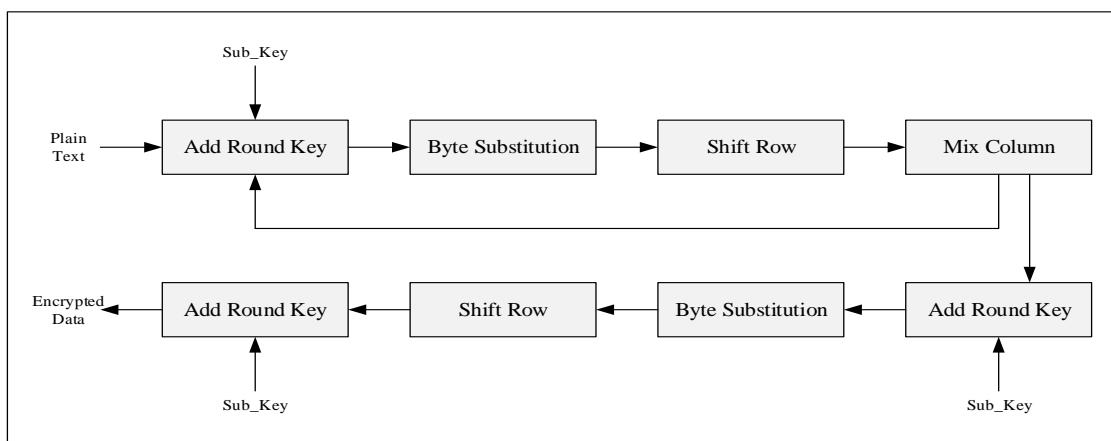


图 32-1 AES 加密流程示意图

上图中 Sub_Key 是指每一轮的子密钥，除了初始变换用初始密钥外，后面的轮变换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 128bits，加密过程运行的轮数就取决于密钥的长度。比如，密钥为 128bits 时，运行轮数为 10 轮；密钥为 192bits 时，运行轮数为 12 轮；密钥为 256bits 时，运行轮数为 14 轮。除了最后一轮缺少 MixColumn 变换，其余各轮均进行完整的轮变换操作。本产品搭载的 AES 仅支持 128bits 密钥长度的加解密过程。

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。

32.2.2 加密操作流程

AES 的加密操作流程例如下：

- 1) 将 128bits 明文写入数据寄存器（AES_DR）中。
- 2) 将加密密钥写入密钥寄存器（AES_KR）中。
- 3) 设置控制寄存器（AES_CR）中的各位，包括：
 - a) 将 CR.Mode 设置为 0
 - b) 向 AES_CR.START 写入 1，启动模块进行运算

注：步骤 a, b 可同时进行

- 4) 判断模块运算是否结束。
不断读取 AES_CR.START，如果其值变为 0，则表示运算结束
- 5) 读取数据寄存器（AES_DR），获得 128 位密文。
- 6) 如果要继续进行新的运算，回到步骤 1，否则结束。

32.2.3 解密操作流程

AES 的解密操作流程例如下：

- 1) 将 128bits 密文写入数据寄存器（AES_DR）中。
- 2) 将解密密钥写入密钥寄存器（AES_KR）中。
- 3) 设置控制寄存器（AES_CR）中的各位，包括：
 - a) 将 CR.Mode 设置为 1
 - b) 向 AES_CR.START 写入 1，启动模块进行运算

注：步骤 a, b 可同时进行

- 4) 判断模块运算是否结束。
不断读取 AES_CR.START，如果其值变为 0，则表示运算结束
- 5) 读取数据寄存器（AES_DR），获得 128 位明文。
- 6) 如果要继续进行新的运算，回到步骤 1，否则结束。

32.2.4 加密解密时间

AES 模块从启动一次运算 (AES_CR.START 写入 1) 到该次运算结束 (AES_CR.START 恢复到 0) 所需时间如下：

加密过程	440个CPU时钟周期
解密过程	580个CPU时钟周期

32.2.5 操作注意事项

- 1) 上电后，本模块进行一次异步复位操作。时钟需在复位脱离前稳定有效，并且在后续运行中持续稳定。
- 2) 在加解密过程中，数据寄存器会改变，如果下次运算操作的数据就是本次运算的结果，就无需重新写入数据。
- 3) 在用相同的密钥加解密大量数据的情况下，不需要重复写入密钥。
- 4) 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。

32.2.6 寄存器说明

BASE ADDR: 0x4000_8000

寄存器名	符号	偏移量	位宽	复位值
AES控制寄存器	AES_CR	0x0000	32	0x0000_0000
AES数据寄存器0	AES_DR0	0x0010	32	0x0000_0000
AES数据寄存器1	AES_DR1	0x0014	32	0x0000_0000
AES数据寄存器2	AES_DR2	0x0018	32	0x0000_0000
AES数据寄存器3	AES_DR3	0x001C	32	0x0000_0000
AES密钥寄存器0	AES_KR0	0x0020	32	0x0000_0000
AES密钥寄存器1	AES_KR1	0x0024	32	0x0000_0000
AES密钥寄存器2	AES_KR2	0x0028	32	0x0000_0000
AES密钥寄存器3	AES_KR3	0x002C	32	0x0000_0000

注意：

- 对所有寄存器的写操作只在本模块处于空闲状态 (AES_CR.START=0) 时有效，否则忽略写操作；对除控制寄存器 (AES_CR) 以外的寄存器的读操作只有在本模块处于空闲状态 (AES_CR.START=0) 时才能读出有效数据，否则读出未知的数据；对控制寄存器 (AES_CR) 的读操作可随时进行，都可读出有效数据。

32.2.6.1 AES 控制寄存器 (AES_CR)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b2	Reserved	-	读出时为“0”，写入时写“0”	R/W
b1	MODE	功能选择	0: 加密运算 1: 解密运算	R/W
b0	START	启动	0: 本模块运算结束或未被启动 1: 启动本模块进行运算	R/W

32.2.6.2 AES 数据寄存器 (AES_DR)

数据寄存器由四个 32 位寄存器组成 128 位数据，用于在该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，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。

位数 : 128 位

偏移地址 : 0x10 – data[31:0]

0x14 – data[63:32]

0x18 – data[95:64]

0x1C – data[127:96]

复位值 : 0x0000_0000 (每个 32 位寄存器)

数据举例 : 0x00112233445566778899AABBCCDDEEFF

偏移地址	寄存器名称	填入数据
0x10	DR0	0x3322_1100
0x14	DR1	0x7766_5544
0x18	DR2	0xBBAA_9988
0x1C	DR3	0xFFEE_DDCC

32.2.6.3 AES 密钥寄存器 (AES_KR)

密钥寄存器由四个 32 位寄存器组成 128 位密钥，用于存放输入的初始密钥。

位数 : 128 位

偏移地址 : 0x20 – key[31: 0]

0x24 – key[63: 32]

0x28 – key[95: 64]

0x2C – key[127: 96]

复位值 : 0x0000_0000 (每个 32 位寄存器)

数据举例 : 0x000102030405060708090A0B0C0D0E0F

偏移地址	寄存器名称	填入数据
0x20	KR0	0x0302_0100
0x24	KR1	0x0706_0504
0x28	KR2	0x0B0A_0908
0x2C	KR3	0x0F0E_0D0C

32.3 安全散列算法 (HASH)

32.3.1 算法简介

安全散列算法的步骤如下：

首先填充消息使其长度恰好为一个比 512 的倍数仅小 64 位的数。填充方法是附一个 1 在消息后面，后接所要求的多个 0，然后在其后附上 64 位的消息长度(填充前)，使消息长度恰好是 512 位的整数倍。

其次将 A、B、C、D、E、F、G 和 H 共 8 个 32 位变量，用十六进制初始化。然后开始算法的主循环，一次处理 512 位消息，循环次数是消息中 512 位分组的数目。

主循环一共进行 64 次操作，此操作称为压缩函数。每次操作都包含移位、循环移位、逻辑运算、模 2³² 加等，运算的过程见下图 32-2。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≤t≤63) 是 64 步循环中的一步。

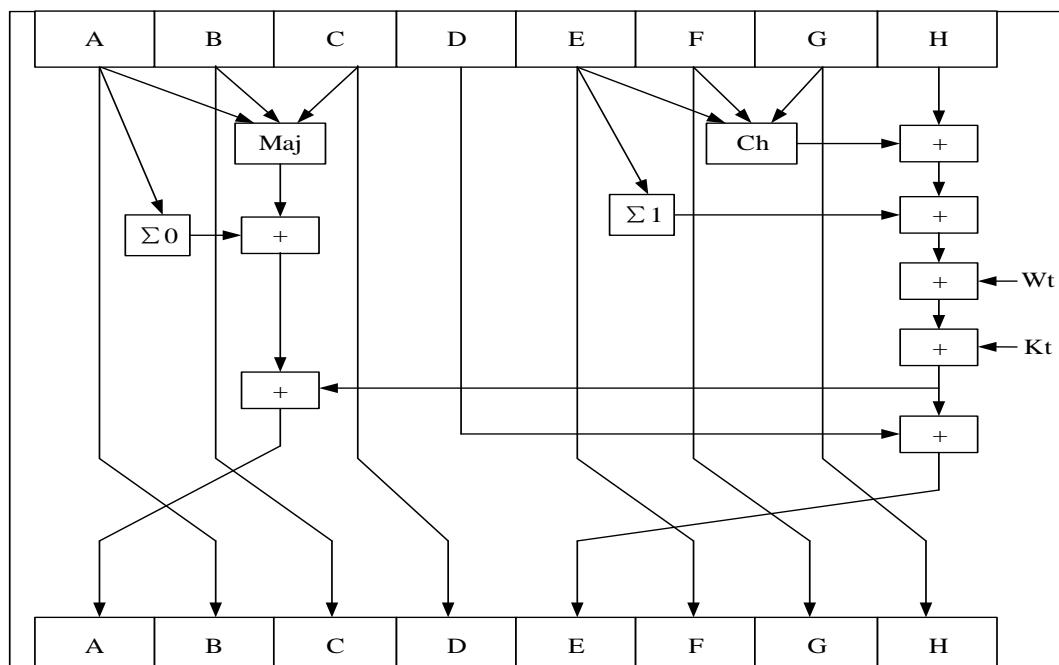


图 32-2 HASH 算法流程图

32.3.2 操作流程

上电后，本模块进行一次异步复位操作。时钟需在复位脱离前稳定有效，并且在后续运行中持续稳定。

HSAH 模块的操作流程如下：

- 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 比特进行分组。
- 2) 将被操作的数据写入数据寄存器（HASH_DR）中。
- 3) 如果本次运算为消息分组的第一组数据，对 HASH_CR.FST_GRP 位写入 1。
- 4) 对 HASH_CR.START 写入 1，启动本模块进行运算。

注：以上的 3 和 4 可以同时进行

- 5) 判断本模块本次运算是否完成，通过以下方法：

不断读取 HASH_CR.START 直到读到该位为 0 即表示运算完成

- 6) 如果本次运算不是消息分组的最后一组数据，返回到 2)。
- 7) 如果本次运算为消息分组的最后一组数据，读取摘要寄存器（HASH_HR）获得本次运算的结果。如需进行进一步运算则返回到步骤 1。

32.3.3 消息填充

SHA-256 的填充分组处理步骤如下：

1. 原始消息分组

将原始消息以 512bit 的大小分为 L 组。令原始消息总的 bit 个数为 l。如果 $l \% 512 < 448$ ，那么分组数 L 为 $l / 512$ ；如果 $l \% 512 \geq 448$ ，那么分组数 L 为 $l / 512 + 1$ 。

2. 添加长度

① 添加填充位：

在消息分组的第 $l / 512$ 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 $l \% 512 < 448$ ，填充使得数据位的长度满足长度为 $448 \bmod 512$ （后 64bit 留做原始消息长度的表示）；如果 $l \% 512 \geq 448$ ，用一个 1 和若干个 0 把第 $l / 512$ 组的 512bit 的数据块填满，并将第 L ($L = l / 512 + 1$) 组的前 448 位填 0。

② 添加原始消息长度：

一个 64bit 块，表示原始消息长度，为 64bit 无符号整数。将原始消息长度添加在第 L 个分组的最后 64bit 处。

举例说明填充分组的过程如下：

1) 填充示例 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：

A. 添加“1”。填充后的消息为“01100001 01100010 01100011 01100100 01100101 1”。

B. 添加“0”。因为原始消息长度为 40bit，所以要添加的 0 的个数为 $512 - 64 - 40 - 1 = 407$ 。

填充后的消息变为（十六进制）：

61626364 65800000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000

C. 添加原始消息长度。原始消息长度 40 的用两个 32bit 字表达为（十六进制）：

00000000 00000028。

填充后的消息变为（十六进制）：

```
61626364 65800000 00000000 00000000  
00000000 00000000 00000000 00000000  
00000000 00000000 00000000 00000000  
00000000 00000000 00000000 00000028
```

2) 填充示例 2：

原始消息为字符串“ABCDBCDECDEFDEFGEFGHFGHIGHHIJKIJKLMNOPQ”。每个字符都可通过其 ASCII 码转换为 8bit，所以消息的长度为 $1 = 56 * 8 = 448$ 。

A. 添加"1"和"0"。填充后的消息（十六进制）为第一个消息块：

```
61626364 62636465 63646566 64656667  
65666768 66676869 6768696A 68696A6B  
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F  
6D6E6F70 6E6F7071 80000000 00000000
```

B. 添加原始消息长度。原始消息长度 448 的用两个 32bit 字表达为（十六进制）：

00000000 000001C0。

填充后的消息（十六进制）为第二个消息块：

```
00000000 00000000 00000000 00000000  
00000000 00000000 00000000 00000000  
00000000 00000000 00000000 00000000  
00000000 00000000 00000000 000001C0
```

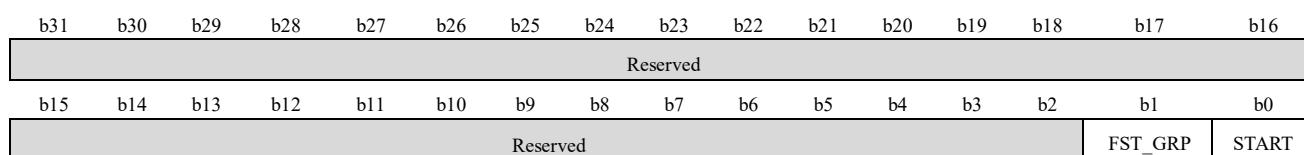
32.3.4 寄存器说明

BASE ADDR: 0x4000_8400

寄存器名	符号	偏移量	位宽	复位值
HASH控制寄存器	HASH_CR	0x0000	32	0x0000_0000
HASH摘要寄存器7	HASH_HR7	0x0010	32	0x0000_0000
HASH摘要寄存器6	HASH_HR6	0x0014	32	0x0000_0000
.....
HASH摘要寄存器1	HASH_HR1	0x0028	32	0x0000_0000
HASH摘要寄存器0	HASH_HR0	0x002C	32	0x0000_0000
HASH数据寄存器15	HASH_DR15	0x0040	32	0x0000_0000
HASH数据寄存器14	HASH_DR14	0x0044	32	0x0000_0000
.....
HASH数据寄存器1	HASH_DR1	0x0078	32	0x0000_0000
HASH数据寄存器0	HASH_DR0	0x007C	32	0x0000_0000

32.3.4.1 HASH 控制寄存器 (HASH_CR)

复位值: 0x0000_0000



位	标记	位名	功能	读写
b31~b2	Reserved	-	读出时为“0”	R/W
b1	FST_GRP	消息分组的第一组	0: 本次运算非消息分组第一组运算 1: 本次运算为消息分组第一组运算	R/W
b0	START	启动	0: 本模块运算结束或未被启动 1: 启动本模块进行运算	R/W

注意:

- START 位的操作方法是: 软件对该位写入 1 后, 本模块将启动运行; 本次运行结束后硬件会自动将该位清 0; 软件查询到该位为 0 即表示本次运行完成。
- 对该寄存器的写入操作只能在本模块非处于运算状态时 (即 START 位为 0 时) 才能进行, 否则硬件将自动忽略写操作。读操作则不受此限制。

32.3.4.2 HASH 摘要寄存器 (HASH_HR)

位数 : 256 位

偏移地址 : 0x10 – hash[255:224]

0x14 – hash[223:192]

0x18 – hash[191:160]

0x1C – hash[159:128]

0x20 – hash[127:96]

0x24 – hash[95:64]

0x28 – hash[63:32]

0x2C – hash[31:0]

复位值 : 0x0000_0000 (每个 32 位寄存器)

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

位	标记	位名	功能	读写
b31~b0	HASH[31:0]	摘要值	在模块运算完成后通过读取本寄存器得到消息摘要	R/W

注意:

- 本寄存器由 8 个 32 位寄存器拼接而成。访问时依次对 8 个 32 位寄存器进行操作，低地址对应的 32 位寄存器存放消息摘要的高字。
- 硬件将自动忽略对本寄存器的写操作。
- 对于本寄存器的读取只能在本模块非处于运算状态时 (HASH_CR.START=0) 才能进行，否则对本寄存器的读取将得到全 0。

32.3.4.3 HASH 数据寄存器 (HASH_DR)

位数 : 512 位

偏移地址 : 0x040 – data[511: 480]

0x044 – data[479: 448]

0x048 – data[447: 416]

0x04C – data[415: 384]

.....

0x070 – data[127: 96]

0x074 – data[95: 64]

0x078 – data[63: 32]

0x07C – data[31: 0]

复位值 : 0x0000_0000 (每个 32 位寄存器)

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

位	标记	位名	功能	读写
b31~b0	DATA[31:0]	数据寄存器	用于在模块运算前写入消息	R/W

注意:

- 本寄存器由 16 个 32 位寄存器拼接而成。访问时依次对 16 个 32 位寄存器进行操作，低地址对应的 32 位寄存器存放数据的高字。
- 对于本寄存器的写入只能在本模块非处于运算状态时 (HASH_CR.START) 才能进行，否则硬件将自动忽略对本寄存器的写操作。
- 对本寄存器的读取将总是得到全 0。

32.4 真随机数发生器 (TRNG)

32.4.1 模块框图

TRNG 模块提供 1 个真随机数发生器，产生 1 个 64 位随机数。

TRNG 的系统框图如下图 32-3 所示。其中随机数发生器是一个模拟随机数发振器电路，用于得到随机性的噪声；算法模块捕捉随机噪声并把结果保存至数据模块、并通过总线输出；控制模块控制 TRNG 的模式和启动。

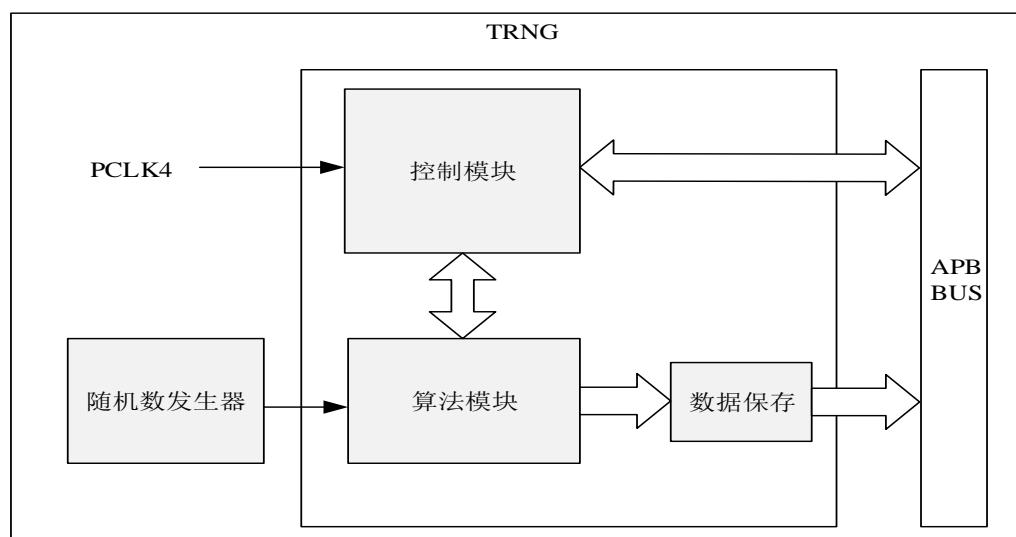


图 32-3 TRNG 系统框图

32.4.2 操作流程

真随机数生成流程如下：

1. 打开随机数发生器电路（设定 TRNG_CR 的 EN 位为 1）。
2. 配置随机数生成模式（设定 TRNG_MR）。
3. 启动随机数生成（设定 TRNG_CR 的 RUN 位为 1）。
4. 读取随机数（读取 TRNG_DR）。
5. 关闭随机数发生器电路（设定 TRNG_CR 的 EN 位为 0）。

32.4.3 中断及事件输出

当随机数生成结束后，寄存器位 TRNG_CR.RUN 硬件清零并产生随机数生成完了中断请求(TRNG_END)。随机数生成结束同时也产生事件输出，可以触发其他模块联动。

32.4.4 操作注意事项

为了获得好的随机数，请设定周边时钟 PCLK4 的频率到 1MHz 以下。

32.4.5 寄存器说明

BASE ADDR: 0x4004_1000

寄存器名	符号	偏移量	位宽	复位值
TRNG控制寄存器	TRNG_CR	0x0000	32	0x0000_0000
TRNG模式寄存器	TRNG_MR	0x0004	32	0x0000_0012
TRNG数据寄存器0	TRNG_DR0	0x000C	32	0x0800_0000
TRNG数据寄存器1	TRNG_DR1	0x0010	32	0x0800_0200

32.4.5.1 TRNG 控制寄存器 (TRNG_CR)

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b3	Reserved	-	读出时为“0”，写入时写“0”	R
b2	Reserved	-	读出时为“0”，写入时写“0”	R/W
b1	RUN	随机数运算开始	0: 随机数运算停止 1: 随机数运算开始 软件写“1”，产生新的64位随机数；运行完毕后，硬件清零。	R/W
b0	EN	模拟发振器使能	0: 关闭模拟随机数发生器电路 1: 打开模拟随机数发生器电路	R/W

32.4.5.2 TRNG 模式寄存器 (TRNG_MR)

复位值: 0x0000_0012

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved										CNT[2:0]		-		LOAD	

位	标记	位名	功能	读写
b31~b5	Reserved	-	读出时为“0”,写入时写“0”	R
b4~2	CNT[2:0]	移位次数控制位	捕捉随机噪声时, 移位次数控制位 011: 移位32次 100: 移位64次 101: 移位128次 110: 移位256次 000~010, 111: 功能保留位	R/W
b1	Reserved	-	读出时为“1”,写入时写“1”	R/W
b0	LOAD	装载控制位	随机数生成前, 数据寄存器是否从随机数发生器装载新的初始值 0: 不装载新的初始值 1: 装载新的初始值	R/W

32.4.5.3 TRNG 数据寄存器 (TRNG_DR)

复位值: DR0: 0x0800_0000

DR1: 0x0800_0200

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

位	标记	位名	功能	读写
b31~b0	DATA[31:0]	随机数	64位随机数	R

33 数据计算单元 (DCU)

33.1 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 个数据的加减和比较大小，以及窗口比较功能。本产品搭载 4 个 DCU 单元，每个单元均可独立完成自身功能。

功能概要：

- 可进行 4 种数据处理：2 个数据的加法，减法，比较以及 3 个数据窗口比较。
- 加法和减法对 DATA0 和 DATA1 寄存器中的数据进行运算，结果存放在 DATA0 中。
- 加法和减法可以选择写寄存器后计算或由其他外围电路事件触发进行运算。
- 加法和减法运算可自动将结果进行一次减半操作，并将减半的结果和加减运算的结果分别放入 2 个数据寄存器内供其他模块使用。
- 比较模式可以对 DATA0 和 DATA1 寄存器之间，以及 DATA0 和 DATA2 寄存器之间的 2 个数据进行比较，可以分别选择当满足大于，小于，等于时产生中断和标志位。
- 比较模式可以用来进行窗口比较，即分别设置 DATA1 和 DATA2 为窗口的上下限，根据 DATA0 和 DATA1 及 DATA0 和 DATA2 的比较结果判断 DATA0 在窗口内或窗口外。
- 能够由其他外围电路事件触发进行运算，并根据运算结果产生多种中断和事件信号，事件信号用于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 产生来启动外围电路开始动作。

33.2 功能描述

33.2.1 加法模式

加法模式计算 DATA0 和 DATA1 的和，其中 DATA0 作为被加数，DATA1 作为加数。每次写 DATA1 寄存器进行一次 $(\text{DATA0} + \text{DATA1})/2$ 的运算， $\text{DATA0} + \text{DATA1}$ 的结果存入 DATA0，而 $(\text{DATA0} + \text{DATA1})/2$ 的结果存入 DATA2。当 $\text{DATA0} + \text{DATA1}$ 结果超过 0xFF(8bit 模式)或 0xFFFF(16bit 模式)或 0xFFFF_FFFF(32bit 模式)时产生标志位并产生中断。

33.2.2 减法模式

减法模式计算 DATA0 和 DATA1 的差，其中 DATA0 作为被减数，DATA1 作为减数。每次写 DATA1 寄存器进行一次 $(\text{DATA0} - \text{DATA1})/2$ 的运算， $\text{DATA0} - \text{DATA1}$ 的结果存入 DATA0，而 $(\text{DATA0} - \text{DATA1})/2$ 的结果存入 DATA2。当 $\text{DATA0} - \text{DATA1}$ 结果小于 0x0(8bit, 16bit, 32bit 模式)时产生标志位并产生中断。

33.2.3 硬件触发启动模式

DCU 能够根据外围电路产生的事件触发启动运算。使用硬件触发启动模式时，需要先将功能时钟控制 0 寄存器(FCG0)的外围电路触发功能使能位置为有效。每个 DCU 单元可独立选择由其它外围电路送来的触发启动信号，选择启动信号时，在触发源选择寄存器(DCU_TRGSELx)中写入要选择的外围电路启动源的编号。当该外围电路事件发生时，会将事件信号输入 DCU 并触发 DCU 启动开始运算。硬件触发启动模式包括触发加模式和触发减模式。在触发加模式时，每发生一次事件触发，DCU 将启动并进行一次 $(\text{DATA0} + \text{DATA1})/2$ 的运算， $\text{DATA0} + \text{DATA1}$ 的结果存入 DATA0，而 $(\text{DATA0} + \text{DATA1})/2$ 的结果存入 DATA2。当 $\text{DATA0} + \text{DATA1}$ 的结果超过 0xFF(8bit 模式)或 0xFFFF(16bit 模式)或 0xFFFF_FFFF(32bit 模式)时产生标志位并产生中断。在触发减模式时，每发生一次事件触发，DCU 将启动并进行一次 $(\text{DATA0} - \text{DATA1})/2$ 的运算， $\text{DATA0} - \text{DATA1}$ 的结果存入 DATA0，而 $(\text{DATA0} - \text{DATA1})/2$ 的结果存入 DATA2。当 $\text{DATA0} - \text{DATA1}$ 的结果小于 0x0(8bit, 16bit, 32bit 模式)时产生标志位并产生中断。

33.2.4 比较模式

比较 DATA0 和 DATA1 以及 DATA0 和 DATA2 的大小, 可选择当 DATA0 大于 DATA1, DATA0 小于 DATA1, DATA0 等于 DATA1 时以及当 DATA0 大于 DATA2, DATA0 小于 DATA2, DATA0 等于 DATA2 时产生标志位并产生中断。比较模式下可以选择数据开始比较的条件, 写 DATA0 后比较或写任何数据寄存器后比较。

33.2.5 中断和事件信号输出

DCU 具有多种中断和用于触发启动其他外围电路的事件输出供用户选择。中断和事件输出的控制由中断条件选择寄存器(DCU_INTEVTSEL)控制, 当需要输出事件信号时, 用户需要将中断条件选择寄存器(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/DCU2/DCU3/DCU4。当需要对应事件在发生时产生中断, 用户需要将中断条件选择寄存器(DCU_INTEVTSEL)的相应控制位置为有效的同时, 将控制寄存器(DCU_CTL)的 INTEN 位置 1。每个 DCU 单元各输出一个 DCU 中断信号, 在中断列表中分别为 DCU1/DCU/DCU3/DCU4。

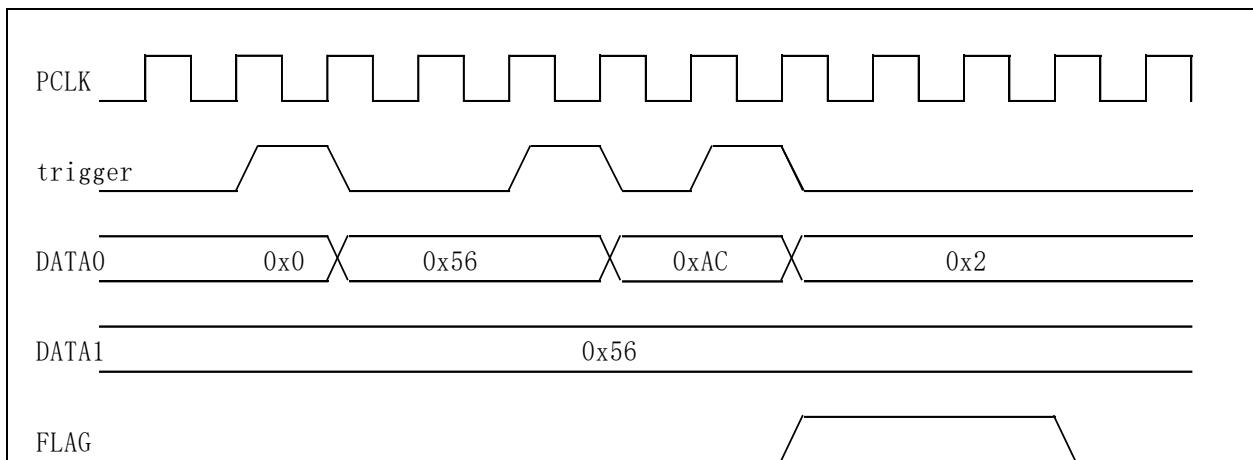
33.3 应用举例

33.3.1 加法模式

配置寄存器以实现：选择加法模式，数据宽度为 16bit。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。DATA1 中继续写入 0xFF，此时计算结果溢出，产生结果 FLAG。由软件写 FLAG 寄存器清除 FLAG。

33.3.2 触发加模式

配置寄存器以实现：选择触发加模式，数据宽度为 8bit。分别在 DATA0 和 DATA1 中写入 0x00 和 0x56，在触发源选择寄存器中写入外围电路事件编号。使该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。在连续 3 次触发后，计算结果溢出，产生结果 FLAG。由软件写 FLAG 寄存器清除 FLAG。



33.3.3 比较模式

配置寄存器以实现：选择比较模式，数据宽度为 16bit。FLAG 输出条件为 DATA0>DATA1，在写入 DATA0 后开始比较。首先在 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 FLAG。之后在 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 FLAG。当 DATA0 写入 0x9999 后满足 FLAG 产生条件，产生 FLAG，并通过写 FLAG 寄存器复位。

33.4 寄存器说明

寄存器一览

单元 1

名称	英文缩写	说明	地址
DCU1控制寄存器	DCU1_CTL	配置DCU的动作模式	0x4005_2000
DCU1标志寄存器	DCU1_FLAG	DCU的结果标识	0x4005_2004
DCU1数据寄存器0	DCU1_DATA0	存放运算数据	0x4005_2008
DCU1数据寄存器1	DCU1_DATA1	存放运算数据	0x4005_200C
DCU1数据寄存器2	DCU1_DATA2	存放运算数据	0x4005_2010
DCU1标志复位寄存器	DCU1_FLAGCLR	清除DCU的结果标识	0x4005_2014
DCU1中断条件选择寄存器	DCU1_INTEVTSEL	选择DCU产生中断的条件	0x4005_2018

单元 2

名称	英文缩写	说明	地址
DCU2控制寄存器	DCU2_CTL	配置DCU的动作模式	0x4005_2400
DCU2标志寄存器	DCU2_FLAG	DCU的结果标识	0x4005_2404
DCU2数据寄存器0	DCU2_DATA0	存放运算数据	0x4005_2408
DCU2数据寄存器1	DCU2_DATA1	存放运算数据	0x4005_240C
DCU2数据寄存器2	DCU2_DATA2	存放运算数据	0x4005_2410
DCU2标志复位寄存器	DCU2_FLAGCLR	清除DCU的结果标识	0x4005_2414
DCU2中断条件选择寄存器	DCU2_INTEVTSEL	选择DCU产生中断的条件	0x4005_2418

单元 3

名称	英文缩写	说明	地址
DCU3控制寄存器	DCU3_CTL	配置DCU的动作模式	0x4005_2800
DCU3标志寄存器	DCU3_FLAG	DCU的结果标识	0x4005_2804
DCU3数据寄存器0	DCU3_DATA0	存放运算数据	0x4005_2808
DCU3数据寄存器1	DCU3_DATA1	存放运算数据	0x4005_280C
DCU3数据寄存器2	DCU3_DATA2	存放运算数据	0x4005_2810
DCU3标志复位寄存器	DCU3_FLAGCLR	清除DCU的结果标识	0x4005_2814
DCU3中断条件选择寄存器	DCU3_INTEVTSEL	选择DCU产生中断的条件	0x4005_2818

单元 4

名称	英文缩写	说明	地址
DCU4控制寄存器	DCU4_CTL	配置DCU的动作模式	0x4005_2C00
DCU4标志寄存器	DCU4_FLAG	DCU的结果标识	0x4005_2C04
DCU4数据寄存器0	DCU4_DATA0	存放运算数据	0x4005_2C08
DCU4数据寄存器1	DCU4_DATA1	存放运算数据	0x4005_2C0C
DCU4数据寄存器2	DCU4_DATA2	存放运算数据	0x4005_2C10
DCU4标志复位寄存器	DCU4_FLAGCLR	清除DCU的结果标识	0x4005_2C14
DCU4中断条件选择寄存器	DCU4_INTEVTSEL	选择DCU产生中断的条件	0x4005_2C18

33.4.1 DCU 控制寄存器(DCU_CTL)

寄存器说明：该寄存器用来配置 DCU 的动作模式。

地址：0x4005_2000, 0x4005_2400, 0x4005_2800, 0x4005_2C00

复位值：0x8000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
INTE N	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							COM PTRG	Reserved			DATASIZE[1: 0]	MODE[2:0]			

位	标记	位名	功能	读写
b31	INTEN	中断使能	0: 不允许产生中断 1: 允许产生中断	R/W
b30~b9	Reserved	-	读时读出0, 写时请写0	R
b8	COMPTRG	比较模式触发 比较的时机	0: 写入DATA0后比较 1: 写入DATA0或DATA1或DATA2后比较	R/W
b7~b5	Reserved	-	读时读出0, 写时请写0	R
b4~b3	DATASIZE[1:0]	数据大小	00: 8bit 01: 16bit 10: 32bit	R/W
b2~b0	MODE[2:0]	动作模式	000: DCU无效 001: 加法模式, 运算在DATA1寄存器写入数据后进行 010: 减法模式, 运算在DATA1寄存器写入数据后进行 011: 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 100: 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 101: 比较模式	R/W

33.4.2 DCU 标志寄存器(DCU_FLAG)

寄存器说明：该寄存器产生 DCU 的结果标识。

地址: 0x4005_2004, 0x4005_2404, 0x4005_2804, 0x4005_2C04

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Reserved																
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Reserved										FLAG _GT1	FLAG _EQ1	FLAG _LS1	FLAG _GT2	FLAG _EQ2	FLAG _LS2	FLAG _OP

位	标记	位名	功能	读写
b31~b7	Reserved	-	读时读出0, 写时请写0	R
b6	FLAG_GT1	大于标志位1	比较模式下, 当DATA0>DATA1时置位, DCU标志复位寄存器 DCU_FLAGCLR的CLR_GT1位写1时清除	R
b5	FLAG_EQ1	等于标志位1	比较模式下, 当DATA0=DATA1时置位, DCU标志复位寄存器 DCU_FLAGCLR的CLR_EQ1位写1时清除	R
b4	FLAG_LS1	小于标志位1	比较模式下, 当DATA0<DATA1时置位, DCU标志复位寄存器 DCU_FLAGCLR的CLR_LS1位写1时清除	R
b3	FLAG_GT2	大于标志位2	比较模式下, 当DATA0>DATA2时置位, DCU标志复位寄存器 DCU_FLAGCLR的CLR_GT2位写1时清除	R
b2	FLAG_EQ2	等于标志位2	比较模式下, 当DATA0=DATA2时置位, DCU标志复位寄存器 DCU_FLAGCLR的CLR_EQ2位写1时清除	R
b1	FLAG_LS2	小于标志位2	比较模式下, 当DATA0<DATA2时置位, DCU标志复位寄存器 DCU_FLAGCLR的CLR_LS2位写1时清除	R
b0	FLAG_OP	运算标志位	加法, 减法和触发加, 触发减模式时, 当加法产生上溢或减法产生下溢时置位, DCU标志复位寄存器DCU_FLAGCLR的CLR_OP位写1时清除	R

33.4.3 DCU 数据寄存器(DCU_DATAx) (x=0,1,2)

寄存器说明：用来存放运算数据。各个数据寄存器在各模式下的功能如下：

	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

地址: 0x4005_2008, 0x4005_2408, 0x4005_2808, 0x4005_2C08

0x4005_200C, 0x4005_240C, 0x4005_280C, 0x4005_2C0C

0x4005_2010, 0x4005_2410, 0x4005_2810, 0x4005_2C10

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b0	DAT[31:0]	运算数据	存放运算数据，实际使用的位数根据DCU_CTL.DATASIZE设置， 当DCU_CTL.DATASIZE=00时DATA[7:0]为有效数据， 当DCU_CTL.DATASIZE=01时DATA[15:0]为有效数据， 当DCU_CTL.DATASIZE=10时DATA[31:0]为有效数据	R/W

33.4.4 DCU 标志复位寄存器(DCU_FLAGCLR)

寄存器说明：该寄存器用于清除 DCU 的结果标识。

地址: 0x4005_2014, 0x4005_2414, 0x4005_2814, 0x4005_2C14

复位值: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Reserved																
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Reserved										CLR _GT1	CLR _EQ1	CLR _LS1	CLR _GT2	CLR _EQ2	CLR _LS2	CLR _OP

位	标记	位名	功能	读写
b31~b7	Reserved	-	读时读出0, 写时请写0	R
b6	CLR_GT1	清除大于标志位1	写1时清除DCU_FLAG的FLAG_GT1位, 写0无任何效果	W
b5	CLR_EQ1	清除等于标志位1	写1时清除DCU_FLAG的FLAG_EQ1位, 写0无任何效果	W
b4	CLR_LS1	清除小于标志位1	写1时清除DCU_FLAG的FLAG_LS1位, 写0无任何效果	W
b3	CLR_GT2	清除大于标志位2	写1时清除DCU_FLAG的FLAG_GT2位, 写0无任何效果	W
b2	CLR_EQ2	清除等于标志位2	写1时清除DCU_FLAG的FLAG_EQ2位, 写0无任何效果	W
b1	CLR_LS2	清除小于标志位2	写1时清除DCU_FLAG的FLAG_LS2位, 写0无任何效果	W
b0	CLR_OP	清除运算标志位	写1时清除DCU_FLAG的FLAG_OP位, 写0无任何效果	W

33.4.5 DCU 中断条件选择寄存器(DCU_INTEVTSEL)

寄存器说明：该寄存器能够选择 DCU 在何种条件下产生中断和输出事件信号

地址：0x4005_2018, 0x4005_2418, 0x4005_2818, 0x4005_2C18

复位值：0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								SEL_WIN[1:0]	SEL_GTI	SEL_EQ1	SEL_LS1	SEL_GT2	SEL_EQ2	SEL_LS2	SEL_OP

位	标记	位名	功能	读写
b31~b9	Reserved	-	读时读出0, 写时请写0	R
b8~b7	SEL_WIN[1:0]	窗口比较中断条件选择	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 00: 不产生窗口比较中断和输出事件信号，此设定下由该寄存器的b1~b6选择其他中断和事件信号产生条件 01: 当DATA0数据在窗口内时产生中断和输出事件信号，即DATA2≤DATA0≤DATA1 10: 当DATA0数据在窗口外时产生中断和输出事件信号，即DATA0>DATA1或DATA0<DATA2 11: 比较模式下不产生任何中断和事件信号	R/W
b6	SEL_GTI	大于中断条件选择1	比较模式下并且SEL_WIN=00时，当DATA0>DATA1时产生中断和输出事件信号，SEL_WIN≠00时该位无效	R/W
b5	SEL_EQ1	等于中断条件选择1	比较模式下并且SEL_WIN=00时，当DATA0=DATA1时产生中断和输出事件信号，SEL_WIN≠00时该位无效	R/W
b4	SEL_LS1	小于中断条件选择1	比较模式下并且SEL_WIN=00时，当DATA0<DATA1时产生中断和输出事件信号，SEL_WIN≠00时该位无效	R/W
b3	SEL_GT2	大于中断条件选择2	比较模式下并且SEL_WIN=00时，当DATA0>DATA2时产生中断和输出事件信号，SEL_WIN≠00时该位无效	R/W
b2	SEL_EQ2	等于中断条件选择2	比较模式下并且SEL_WIN=00时，当DATA0=DATA2时产生中断和输出事件信号，SEL_WIN≠00时该位无效	R/W
b1	SEL_LS2	小于中断条件选择2	比较模式下并且SEL_WIN=00时，当DATA0<DATA2时产生中断和输出事件信号，SEL_WIN≠00时该位无效	R/W
b0	SEL_OP	运算标志位	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	R/W

34 CRC 运算 (CRC)

34.1 简介

在许多应用中，都需要 CRC 算法来校验数据的完整性和正确性。尤其是在数据传输中，CRC 校验更是被广泛应用。本模块可采用 CRC16 和 CRC32 两种算法对数据进行运算和校验。

34.2 功能框图

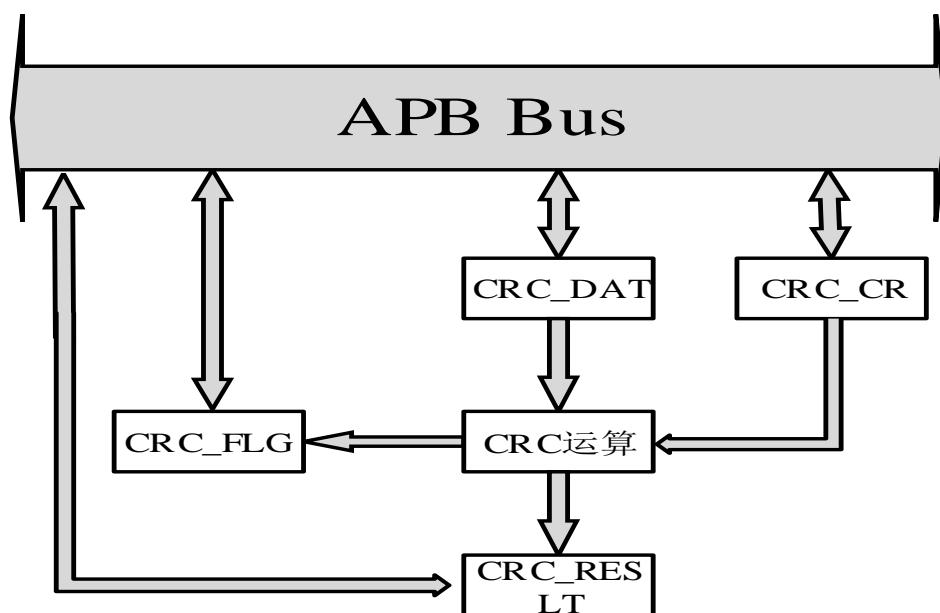


图 34-1 CRC 模块框图

34.3 功能说明

本模块 CRC 算法遵从 ISO/IEC13239 的定义，分别采用 32 位和 16 位的 CRC。CRC32 的生成多项式为 $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$ 。CRC16 的生成多项式为 $X^{16}+X^{12}+X^5+1$ 。

本模块功能包括 CRC 编码生成和 CRC 编码校验。

34.3.1 CRC 编码产生

CRC 编码就是对一串数据进行运算，产生 16/32 位的 CRC 编码结果。

操作流程如下：

1. 根据需要设定 CRC_CR 寄存器位 XOROUT,REFOUT,REFIN,CRC_SEL。
2. CRC16(CRC_SEL=0): 将 16 位初始值写入到 CRC_RESLT 寄存器的 CRC_REG[15:0] 中。
CRC32(CRC_SEL=1): 将 32 位初始值写入到 CRC_RESLT 寄存器的 CRC_REG[31:0] 中。
3. 将需要运算的数据依次写入 CRC_DAT 寄存器，每次写操作对应输入 1 个数据(16 位，32 位)。比如，有 10 个数据，就依次对 CRC_DAT 寄存器写入 10 次。
4. 在将所有需要运算的数据写入 CRC_DAT 寄存器之后，读取 CRC_RESLT 寄存器 CRC_REG，即可获得 16/32 位 CRC 编码值。

34.3.2 CRC 校验

CRC 校验就是要对一串数据以及 16/32 位 CRC 编码进行判断，校验其是否正确，仅支持 16/32 位初始值全为 1 的情况，且仅支持 ISO/IEC13239 中关于 16/32 位 CRC 的校验。

1. 根据需要设置 CRC_CR，包括 XOROUT,REFOUT,REFIN,CRC_SEL。
2. CRC16(CRC_SEL=0): 将 16 位初始值写入到 CRC_RESLT 寄存器的 CRC_REG[15:0] 中。
CRC32(CRC_SEL=1): 将 32 位初始值写入到 CRC_RESLT 寄存器的 CRC_REG[31:0] 中。

3. 将需要运算的数据写入 CRC_DAT 寄存器。
4. 将 16/32 位的 CRC 编码值写入 CRC_DAT 寄存器。
5. 读取 CRC_FLG 寄存器，为 1 则表示校验成功，为 0 则表示校验失败(对于 CRC16 来讲，也可以读取 CRC_RESLT[16],CRCFLAG_16 位来判断是否校验成功)

34.3.3 对 XOROUT,REFOUT,REFIN 不全为 1 的 CRC 校验

当选项 XOROUT,REFOUT,REFIN 不是全 1，跟随数据输入的 CRC 编码值需要进行如下变换后再写入 CRC_DAT 进行校验：

1. REFOUT=0 时，将 CRC 编码值的所有 BIT 进行颠倒；否则，不用颠倒。
2. XOROUT=0 时，将(1)的结果取反；否则，不取反；
3. REFIN=0 时，将(2)的结果中的每个 BYTE 的 BIT 完全颠倒；否则，不颠倒；
4. 将(3)的结果写入到 CRC_DAT 寄存器进行校验。

34.4 寄存器说明

表 34-1 所示，为 CRC 模块的寄存器列表。

CRC_BASE_ADDR: 0x40008C00

寄存器名	符号	偏移地址	位宽	复位值
CRC控制寄存器	CRC_CR	0x00	32	0x0000_001C
CRC结果寄存器	CRC_RESLT	0x04	32	0x0000_0000
CRC标志寄存器	CRC_FLG	0x0C	32	0x0000_0000
CRC数据寄存器	CRC_DAT	0x80~0xFC	32	0xXXXX_XXXX

表 34-1 CRC 寄存器列表

34.4.1 控制寄存器 (CRC_CR)

复位值: 0x001C

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

位	标记	位名	功能	读写
b30~b5	Reserved	-	读出时为“0”,写入时写“0”	R
b4	XOROUT	结果取反输出	0: 结果不取反输出 1: 结果取反输出	R/W
b3	REFOUT	结果所有位数颠倒输出	0: 结果所有位数不颠倒输出 1: 结果所有位数颠倒输出	R/W
b2	REFIN	输入数据字节内位数颠倒	0: 输入数据字节内位数不颠倒 1: 输入数据字节内位数颠倒	R/W
b1	CR	运算位数选择	0: CRC16 1: CRC32	R/W
b0	Reserved	-	保留位	R/W

34.4.2 结果寄存器 (CRC_RESLT)

复位值: 0x0000

选择 CRC16: (CRC_SEL=0)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CRCFLAG_16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CRC_REG[15:0]															

位	标记	位名	功能	读写
b31~b17	Reserved	-	读出时为“0”,写入时写“0”	R
b16	CRCFLAG_16	CRC16校验结果	0: CRC16运算校验错误 1: CRC16运算校验正确	R
b15~b0	CRC_REG[15:0]	结果位	本16位寄存器用于每次CRC16计算结果的更新和保存; 运算后, 读取本寄存器将得到16位的CRC编码结果	R/W

选择 CRC32: (CRC_SEL=1)

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

位	标记	位名	功能	读写
b31~b0	CRC_REG[31:0]	结果位	本32位寄存器用于每次CRC32计算结果的更新和保存; 运算后, 读取本寄存器将得到32位的CRC编码结果	R/W

34.4.3 标志寄存器 (CRC_FLAG)

复位值: 0x0000

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

-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CRC_FLAG
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----------

位	标记	位名	功能	读写
b31~b1	Reserved	-	读出时为“0”，写入时写“0”	R
b0	CRC_FLAG	CRC32校验结果	CRC校验结果标志位; 0: 当前校验错误; 1: 当前校验正确	R

34.4.4 数据寄存器 (CRC_DAT)

复位值: 0x0000

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

位	标记	位名	功能	读写
b31~b0	CRC_DAT[31:0]	数据寄存器	本寄存器用于输入需要运算的数据; 本寄存器的地址是一个范围 (0x80~0xFC), 对应了32个地址 (每个地址对应32位数据)。对该32个地址任何一个进行操作, 都会认为对本寄存器进行操作。	W

35 SDIO 控制器 (SDIOC)

35.1 简介

SDIOC 提供了一个 SD 主机接口和一个 MMC 主机接口, 用于和支持 SD2.0 协议的 SD 卡, SDIO 设备以及支持 eMMC4.2 协议的 MMC 设备进行通信。SDIOC 特点如下:

- 支持 SDSC, SDHC, SDXC 格式 SD 卡及 SDIO 设备
- 支持一线式(1bit)和四线式(4bit)SD 总线
- 支持一线式(1bit), 四线式(4bit)和八线式(8bit)MMC 总线
- SD 时钟最高 50MHz
- 具有卡识别和硬件写保护功能

本产品带有 2 个 SDIO 控制器, 能够同时与 2 个 SD/MMC/SDIO 设备进行通信。

35.2 功能描述

35.2.1 端口分配

端口名	IO	功能
SDIOx_CK(x=1~2)	O	SD时钟输出信号
SDIOx_CMD(x=1~2)	I/O	SD命令和应答信号
SDIOx_D0(x=1~2)	I/O	SD数据信号
SDIOx_D1(x=1~2)	I/O	SD数据信号
SDIOx_D2(x=1~2)	I/O	SD数据信号
SDIOx_D3(x=1~2)	I/O	SD数据信号
SDIOx_D4(x=1~2)	I/O	SD数据信号
SDIOx_D5(x=1~2)	I/O	SD数据信号
SDIOx_D6(x=1~2)	I/O	SD数据信号
SDIOx_D7(x=1~2)	I/O	SD数据信号
SDIOx_CD(x=1~2)	I	SD卡识别状态信号
SDIOx_WP(x=1~2)	I	SD写保护状态信号

上述端口根据 SD2.0 协议和 eMMC4.51 协议，SD 时钟输出信号(SDIOx_CK(x=1~2))用于在 SDIOC 和 SD/MMC 设备通信时输出 SD 时钟；SD 命令和应答信号(SDIOx_CMD(x=1~2))用于向设备输出 SD/MMC 命令以及接收设备发回的应答信息；SD 数据信号(SDIOx_Dy(x=1~2) (y=0~7))用于 SDIOC 和设备在通信过程中收发数据，当采用一线式(1bit)通信时，只有 SDIOx_D0(x=1~2)有效，SDIOx_Dy(x=1~2) (y=1~7)保持高电平，采用四线式(4bit)通信时，SDIOx_Dy(x=1~2) (y=0~3)有效，SDIOx_Dy(x=1~2) (y=4~7)保持高电平，采用八线式(8bit)通信时，SDIOx_Dy(x=1~2) (y=0~7)有效，八线式(8bit)通信仅限于 MMC 设备通信。

35.2.2 基本访问方式

用户通过读写 SDIOC 寄存器的方式启动 SDIOC 同片外 SD/MMC 设备进行通信。由于写命令寄存器将触发 SDIOC 的发送命令动作，因此写命令寄存器必须最后进行。在此之前，用户需要通过传输模式寄存器(TRANSMODE)设定传输方式，通过参数寄存器(ARG0, ARG1)设定命令参数，并在设置命令寄存器(CMD)的时候同时设置正确的命令的编号(command index)，类型，应答(response)类型等。一旦写命令寄存器执行后，SDIOC 将发送命令后，用户可以通过读取中断状态寄存器(NORINTST, ERRINTST)来查询是否接收完应答信息，有无错误信息等。当一个命令执行完成后，用户可以通过读取应答寄存器(RESP0~7)得到命令的应答。

35.2.3 数据传输

SDIOC 的数据缓冲寄存器用于 CPU/DMA 等主机设备和 SD 设备之间进行数据交换。SDIOC 内置了 FIFO 以加快数据交换的速度。当命令包含数据发送时，用户将数据依次写入数据缓冲寄存器(BUF0, BUF1)，这些数据将首先缓存在 SDIOC 的 FIFO 中，当写入的数据数量达到数据长度寄存器(BLKSIZE)的设定值或 512 字节时，SDIOC 将通过 SDIOx_Dy(x=1~2) (y=0~7)发送数据。在此期间 SDIOC 通过 FIFO 能够继续向数据缓冲寄存器写入数据。同样地，当命令包含数据接收时，SDIOC 通过 SDIOx_Dy(x=1~2) (y=0~7)接收数据，并将数据缓存在 SDIOC 的缓冲器(BUFFER)中，用户通过读取数据缓冲寄存器(BUF0, BUF1)得到数据。数据的格式请参考 SD 和 MMC 协议。

35.2.4 SD 时钟

SD 时钟(SDIOx_CK(x=1~2))由主机生成并输出给设备用于两者间的通信。SD 时钟由 SDIOC 模块的动作时钟(EXCLK)经由分频器分频产生。分频器的分频系数由时钟控制寄存器(CLKCON)设定，根据 SD2.0 协议的要求，在数据传输模式(data transfer mode)下最快应为 50MHz，因此用户需要根据 EXCLK 的频率设定合理的分频系数。

35.2.5 中断和 DMA 启动请求

35.2.5.1 SD 中断

SDIOC 提供两类中断，普通中断(normal interrupt)和错误中断(error interrupt)。普通中断是 SDIOC 和 SD/MMC 卡通信的过程中产生各种事件时发生的中断。错误中断是 SDIOC 动作通信过程中发生各种错误时产生的中断。普通中断和错误中断分别具有各自的中断状态寄存器，中断状态使能寄存器，以及中断信号使能寄存器。中断状态寄存器用于表示中断产生的要因，中断状态使能寄存器用于使能中断状态寄存器的各个状态位，中断状态寄存器的各个状态位需要在中断状态使能寄存器使能位使能时有效，中断信号使能寄存器用于是否允许各个中断要因向 CPU 申请中断。

35.2.5.2 SDIO 中断

SDIO 设备在传输空闲时可以向主机发送卡中断(card interrupt)请求。使用 SDIO 中断时需要将中断状态使能寄存器(NORINTSEN)的卡中断(CINTEN)使能，如果需要向 CPU 申请中断，还需将中断信号使能寄存器(NORINTSGEN) 的卡中断(CINTEN)使能。SDIO 设备通过将 SDIOx_D1(x=1~2)数据线拉低向主机申请中断。SDIOC 检测到 SDIOx_D1(x=1~2)被拉低后将中断状态寄存器(NORINTST)的卡中断(CINT)置位，并根据设定向 CPU 申请中断。SDIO 设备判断中断处理完成后将释放 SDIOx_D1(x=1~2)。

35.2.5.3 DMA 请求

SDIOC 在通信中的读写数据能够通过 DMA 来完成。使用 DMA 传送向设备写数据时，将 DMA 的一个通道的启动源设置为 SDIOC 的写请求，再将 DMA 的传输目标地址设置为数据缓冲寄存器(BUF0, BUF1)且为固定地址。当 SDIOC 发送完写数据命令后，如果此时数据 FIFO 空，则向 DMA 发出写请求信号，启动 DMA 向数据缓冲寄存器(BUF0, BUF1)写入数据，这些数据将依次进入数据 FIFO 中。当 FIFO 中数据达到数据长度寄存器(BLKSIZE)的设定值或 512 字节时，SDIOC 将通过 SDIOx_Dy(x=1~2)(y=0~7)发送数据。如果是多块写命令，SDIOC 将同时继续向 DMA 发送启动请求写入数据。使用 DMA 传送从设备读数据时，将 DMA 的另一个通道的启动源设置为 SDIOC 的读请求，再将 DMA 的传输源地址设置为数据缓冲寄存器(BUF0, BUF1)且为固定地

址。当 SDIOC 发送完读数据命令后, 设备将通过 SDIO_x_D_y(x=1~2) (y=0~7)发送数据, SDIOC 的数据 FIFO 开始接收数据, 当 FIFO 中数据达到数据长度寄存器(BLKSIZE)的设定值或 512 字节时, SDIOC 向 DMA 发出读请求信号, 启动 DMA 从数据缓冲寄存器(BUF0, BUF1)读出数据。如果是多块读命令, SDIOC 将同时继续从设备读取下一个数据块。

35.2.5.4 卡的插入(insert)和移除(removal)

SD/MMC/SDIO 设备的插入(insert)和移除(removal)由 SDIO_x_CD(x=1~2)信号线识别。当卡槽内无设备时, 卡槽会通过电阻将 SDIO_x_CD(x=1~2)信号拉高。当有设备插入后, SDIO_x_CD(x=1~2)信号将被拉低, 设备被移除后将重新变为高。SDIOC 通过 SDIO_x_CD(x=1~2)的电平来判断是否有设备。用户可以通过读取主机状态寄存器(PSTAT)的 CDPL 位来判断是否有设备插入。SDIOC 在设备的插入和移除时能够产生相应的中断, 通过中断状态使能寄存器(NORINTSTEN)和中断信号使能寄存器(NORINTSGEN)来使能。

35.2.6 主机和设备初始化

35.2.6.1 主机初始化

SDIOC 在使用时需要首先进行初始化设置。SDIOC 初始化步骤如下:

1. 读取主机状态寄存器(PSTAT), 查询时钟状态以及设备插入状态
2. 配置电源控制寄存器(PWRCON), 使能 SDIOC
3. 配置时钟控制寄存器(CLKCON), 使 SDIOC 输出 SD 时钟, 并根据 EXCLK 的频率配置 SD 时钟分频以保证卡识别模式下 SD 时钟频率不超过 400KHz
4. 配置主机控制寄存器(HOSTCON), 选择一线式(1bit)模式并禁用高速模式(high speed mode)
5. 根据设备特性配置超时控制寄存器(TOUTCON), 使得主机在通信超时时结束通信并报告错误
6. 配置普通和错误中断状态使能寄存器, 中断信号使能寄存器以使 SDIOC 在需要时发生中断并将标志位置起

35.2.6.2 SD 卡初始化

完成 SDIOC 初始化配置后，如果连接了 SD 卡。需要按照 SD 协议进行初始化，初始化顺序如下：

1. 卡复位，向设备发送复位命令 CMD0，CMD0 没有应答信息(response)，此时卡将进入卡识别模式
 2. 确认卡的工作状态(operation condition)，向设备发送 CMD8 并等待接收应答信息(response)
 3. 发送初始化命令 ACMD41(先发送 CMD55，再发送 CMD41)，根据应答信息判断设备是否完成初始化，否则持续发送 ACMD41 直到初始化完成
 4. 发送 CMD2 得到卡的 CID 信息
 5. 发送 CMD3 得到卡的 RCA 信息
- 此时卡将进入数据传输模式，初始化完成。

35.2.6.3 MMC 卡初始化

完成 SDIOC 初始化配置后，如果连接了 MMC 卡。需要按照 MMC 协议进行初始化，初始化顺序如下：

1. 卡复位，向设备发送复位命令 CMD0，CMD0 没有应答信息(response)，此时卡将进入卡识别模式
 2. 确认卡的工作状态(operation condition)，向设备发送 CMD1 并等待接收应答信息(response),根据应答信息判断设备是否完成初始化，否则持续发送 CMD1 直到初始化完成
 3. 发送 CMD2 得到卡的 CID 信息
 4. 发送 CMD3 得到卡的 RCA 信息
- 此时卡将进入数据传输模式，初始化完成。

35.2.6.4 SDIO 初始话

完成 SDIOC 初始化配置后，如果连接了 SDIO 设备。需要按照 SDIO 协议进行初始化，初始化顺序如下：

1. 设备复位，向设备发送复位命令 CMD0，CMD0 没有应答信息(response)

2. 确认 SDIO 设备的工作状态(operation condition), 向设备发送 CMD5 并等待接收应答信息(response)
3. 发送 CMD3 得到设备的 RCA 信息, 初始化完成

35.2.7 SD/MMC 单数据块(single block)读写

在 SD/MMC 卡进入数据传输模式后, SDIOC 就能够通过读写命令访问 SD/MMC 卡的数据。其中单数据块(block)读写的顺序如下:

1. 配置钟控制寄存器(CLKCON), 使 SDIOC 输出 SD 时钟, 并根据 EXCLK 的频率配置 SD 时钟分频使 SD 时钟频率不超过默认速度模式下的最大时钟速度(default speed mode, $f_{pp} \leq 25\text{MHz}$)。
2. 发送 CMD7, SD/MMC 卡将进入数据传输状态。
3. 如有需要, 对于 SD 卡, 可配置主机控制寄存器(HOSTCON)设置主机总线宽度, 并发送 ACMD6 设置 SD 总线宽度(1bit 或 4bit), 对于 MMC 卡, 可配置主机控制寄存器(HOSTCON)设置主机总线宽度, 并通过 CMD6(SWITCH)命令改写 MMC 卡的 Ext_CSD 寄存器来设置总线宽度(1bit, 4bit 或 8bit)。
4. 如有需要, 对于 SD 卡, 可配置数据长度寄存器(BLKSIZE)设置数据块长度, 并通过 CMD16 命令设置数据块的大小(字节数), 配置范围为 1~512 字节。对于 SDHC/SDXC 以及 MMC 卡数据块的大小为固定 512 字节。
5. 如有需要, 对于 SD 卡, 可配置主机控制寄存器(HOSTCON)设置主机为高速模式(high speed mode), 并发送 CMD6 将 SD 卡切换至高速模式(high speed mode, $f_{pp} \leq 50\text{MHz}$), 切换成功后可将 SDIOx_CK($x=1\sim 2$)设置为最高 50MHz
6. 写数据时, 首先配置传输模式寄存器(TRANSMODE), 设置传输方式为写, 单块传输。发送单 block 写命令 CMD24, 如使用 CPU 写数据, 在确认中断状态寄存器(NORINTST)的 BWR 位为 1 后, 向数据缓冲寄存器(BUF0, BUF1)写入数据, 如使用 DMA 写数据, 则等待 DMA 传输完成, SDIOC 将通过数据信号 SDIOx_Dy($x=1\sim 2$)发送数据。发送完成后, 中断状态寄存器(NORINTST)的 TC 将置 1 并产生中断请求, 如果传输过程出现错误, 则相应错误标志位会置起并产生中断请求。
7. 读数据时, 首先配置传输模式寄存器(TRANSMODE), 设置传输方式为读, 单块传

输。发送单 block 读命令 CMD17, SDIOC 将通过数据信号 SDIOx_Dy(x=1~2)接收数据。如使用 CPU 读数据, 在确认中断状态寄存器(NORINTST)的 BRR 位为 1 后, 从数据缓冲寄存器(BUF0, BUF1)读出数据, 如使用 DMA 读数据, 则等待 DMA 传输完成, 读取完成后, 中断状态寄存器(NORINTST)的 TC 将置 1 并产生中断请求, 如果传输过程出现错误, 则相应错误标志位会置起并产生中断请求。

35.2.8 SD/MMC 多数据块(multi block)读写

多数据块(block)读写与单数据块(block)读写的顺序如下:

1. 配置钟控制寄存器(CLKCON), 使 SDIOC 输出 SD 时钟, 并根据 EXCLK 的频率配置 SD 时钟分频使 SD 时钟频率不超过默认速度模式下的最大时钟速度(default speed mode, fpp<=25MHz)。
2. 发送 CMD7, SD/MMC 卡将进入数据传输状态。
3. 如有需要, 对于 SD 卡, 可配置主机控制寄存器(HOSTCON)设置主机总线宽度, 并发送 ACMD6 设置 SD 总线宽度(1bit 或 4bit), 对于 MMC 卡, 可配置主机控制寄存器(HOSTCON)设置主机总线宽度, 并通过 CMD6(SWITCH)命令改写 MMC 卡的 Ext_CSD 寄存器来设置总线宽度(1bit, 4bit 或 8bit)。
4. 如有需要, 对于 SD 卡, 可配置数据长度寄存器(BLKSIZE)设置数据块长度, 并通过 CMD16 命令设置数据块的大小(字节数), 配置范围为 1~512 字节。对于 SDHC/SDXC 以及 MMC 卡数据块的大小为固定 512 字节。
5. 配置传输模式寄存器(TRANSMODE), 设置传输方式(读/写), 多块传输。如果设置了允许数据块计数使能, 则需要在数据块计数寄存器(BLKCNT)中设置需要传输的数据块数量, 不设置数据块计数寄存器(BLKCNT)则不能启动多块传输。如果设置了禁止数据块计数使能, 则不需要设置数据块计数寄存器(BLKCNT), 此时可以进行不限数量的多块传输, 当主机决定结束传输时, 需要在最后一个数据块传输完成后向设备发送一个 CMD12 告知设备数据传输结束。
6. 如有需要, 对于 SD 卡, 可配置主机控制寄存器(HOSTCON)设置主机为高速模式 (high speed mode), 并发送 CMD6 将 SD 卡切换至高速模式(high speed mode, fpp<=50MHz), 切换成功后可将 SDIOx_CK(x=1~2)设置为最高 50MHz

7. 写数据时，发送多 block 写命令 CMD25，如使用 CPU 写数据，在确认中断状态寄存器(NORINTST)的 BWR 位为 1 后，向数据缓冲寄存器(BUF0, BUF1)写入数据，发送过程中 BWR 将保持 0，发送结束后重新置 1，用户可以在此时写入第二个 block 的数据。如使用 DMA 写数据，则等待 DMA 传输完成，SDIOC 将通过数据信号 SDIOx_Dy(x=1~2)发送数据。中断状态寄存器(NORINTST)的 TC 将置 1 并产生中断请求，如果传输过程出现错误，则相应错误标志位会置起并产生中断请求。全部数据发送完毕后，如果设置了自动发送 CMD12，则 SDIOC 将自动发送一个 CMD12 以结束传输，否则需要手动向设备发送 CMD12 告知设备数据传输结束。
8. 读数据时，发送多 block 读命令 CMD18，SDIOC 将通过数据信号 SDIOx_Dy(x=1~2)接收数据。如使用 CPU 读数据，在确认中断状态寄存器(NORINTST)的 BRR 位为 1 后，从数据缓冲寄存器(BUF0, BUF1)读出数据，接收过程中 BRR 将保持 0，发送结束后重新置 1，用户可以在此时读取第二个 block 的数据。如使用 DMA 读数据，则等待 DMA 传输完成，读取完成后，中断状态寄存器(NORINTST)的 TC 将置 1 并产生中断请求，如果传输过程出现错误，则相应错误标志位会置起并产生中断请求。全部数据接收完毕后，如果设置了自动发送 CMD12，则 SDIOC 将自动发送一个 CMD12 以结束传输，否则需要手动向设备发送 CMD12 告知设备数据传输结束。

35.2.9 传输终止(abort), 挂起(suspend)和恢复(resume)

当进行多块传输时，可以通过软件控制来终止(abort)或者挂起(suspend)。无论是否设定了传输数据块的数量都可以执行终止(abort)操作。终止操作分异步终止和同步终止。异步终止操作需要在传输进行中，通过写命令寄存器(CMD)发送 CMD12 来终止传输，此时传输将立即终止。对于写操作，SD/MMC 卡将舍弃当前数据块数据并进入烧写模式(program)，将之前接收到的数据块写入 FLASH 中。对于读操作，SD/MMC 将停止传输数据。同步终止是指传输进行中通过设置数据块间隔寄存器(BLKGAP)使传输在数据块间隔处停止传输，设置完成后，传输将在当前数据块传输结束后停止传输。此时需要发送 CMD12 以结束传输。

执行挂起(suspend)操作时，首先通过设置数据块间隔寄存器(BLKGAP) 的 SABGR 位

使传输在数据块间隔处停止传输，并且使能读等待(read wait)功能。设置完成后，传输将在当前数据块传输结束后停止传输。此时通过写命令寄存器(CMD)发送 CMD52 来将传输挂起。发送挂起命令后，需要继续通过 CMD52 读取 SDIO 的 CCCR 寄存器的 BS 位(bus status)，如果 BS=0 代表传输已挂起，SD 总线转为空闲状态。传输被挂起后，主机可以对 SDIO 的其他功能执行操作。但是如果想稍后进行恢复(resume)操作，需要在挂起后备份 SDIOC 寄存器(偏移地址 00h~0Dh)，执行完其他操作后，恢复这些寄存器设置。执行恢复(resume)操作时，首先清除数据块间隔寄存器(BLKGAP)使传输在数据块间隔处停止传输的设置，之后通过写命令寄存器(CMD)发送 CMD52 来恢复传输。发送恢复命令后，需要继续通过 CMD52 读取 SDIO 的 CCCR 寄存器的 DF 位(data flag)，如果 DF=1 代表执行恢复后有数据需要继续传输，如果 DF=0 则代表没有数据需要传输，此时应写软件复位寄存器(SFTRST)复位数据线。

注意：

- 挂起(suspend)和恢复(resume) 操作需要 SDIO 设备和 combo card 设备支持此类操作以及读等待(read wait)操作。SDIO 的 CIA 区域(common I/O area)的访问不支持挂起(suspend)和恢复(resume)操作，只能够使用读等待(read wait)功能。

35.2.10 读等待(read wait)

读等待(read wait)允许主机在连续数据传输中插入 CMD52 以访问 SDIO 设备的其他功能。执行读等待(read wait)时首先通过设置数据块间隔寄存器(BLKGAP)的 SABGR 位使传输在数据块间隔处停止传输，并且使能读等待(read wait)功能。设置完成后，主机将在当前数据块传输结束后将 SDIOx_D2(x=1~2)数据线拉低。此时传输将暂停，主机可以在此时插入 CMD52 以访问其他不需要数据传输的功能。需要结束读等待(read wait)功能时，通过设置数据块间隔寄存器(BLKGAP)的 CR 并清除 SABGR 以恢复传输。

注意：

- 读等待(read wait)需要 SDIO 设备和 combo card 设备支持，并且需要在四线式总线传输下进行。

35.2.11 唤醒(wakeup)

当长时间不工作时，可将系统转入低功耗状态以降低功耗。在低功耗状态下可以通过 SD/MMC/SDIO 设备的插入(insert)/移除(removal)和卡中断(card interrupt)来唤醒系统继续工作。使用唤醒功能时，需要将插入(insert)/移除(removal)或卡中断(card interrupt)在中断状态使能寄存器(NORINTSTEN)和中断信号使能寄存器(NORINTSGEN)中相应的使能位使能。同时使能 SDIOx_CD(x=1~2)/ SDIOx_D1(x=1~2) 端口相应的唤醒功能。配置完成后可将系统转入低功耗模式。当发生插入(insert)/移除(removal)和卡中断(card interrupt)时，SDIOx_CD(x=1~2)/ SDIOx_D1(x=1~2) 端口的唤醒功能将会唤醒系统，并发生插入(insert)/移除(removal)和卡中断(card interrupt)中断。

使用插入(insert)唤醒停止模式(stop mode)流程：

1. 配置 SDIOC 端口，选择 SDIO1_CD 为 PA10
2. 配置 PA10 的 PCR 寄存器，选择 IRQ10 有效
3. 配置电源控制模块的 PWRC3 寄存器使 CPU 执行 WFI 命令后进入停止模式
4. 配置中断控制模块 WUPEN 寄存器使能 IRQ10 的唤醒功能
5. 配置中断控制模块的 EIRQCR10 寄存器，选择下降沿触发
6. 配置 PWRCON 使能 SDIOC
7. 配置中断状态使能寄存器(NORINTSTEN)和中断信号使能寄存器(NORINTSGEN)中插入(insert)相应的使能位使能
8. 配置中断源选择寄存器选择 SDIOC 中断为中断源，并使能中断
9. 执行 WFI 指令使系统进入停止模式
10. 当设备插入时，PA10 将被拉低，通过 IRQ10 唤醒 CPU，并根据 SD 中断申请进入中断子程序
11. 清除中断状态寄存器(NORINTST)中插入(insert)状态，退出中断子程序，执行后续操作

使用插入(insert)唤醒掉电模式(power down mode)流程：

1. 配置 SDIOC 端口，选择 SDIO1_CD 为 PA10
2. 配置 PA10 的 PCR 寄存器，选择 IRQ10 有效
3. 配置电源控制模块的 PWRC3 和 PWRC0 寄存器使 CPU 执行 WFI 命令后进入掉电

模式

4. 配置电源控制模块的 PDWKE1 寄存器使 IRQ10 的掉电唤醒功能有效
5. 配置中断控制模块 WUPEN 寄存器使能 IRQ10 的唤醒功能
6. 配置中断控制模块的 EIRQCR10 寄存器，选择下降沿触发
7. 配置 PWRCON 使能 SDIOC
8. 配置中断状态使能寄存器(NORINTSTEN)和中断信号使能寄存器(NORINTSGEN)
中插入(insert)相应的使能位使能
9. 配置中断源选择寄存器选择 SDIOC 中断为中断源，并使能中断
10. 执行 WFI 指令使系统进入掉电模式
11. 当设备插入时，PA10 将被拉低，通过 IRQ10 唤醒系统重新上电

35.3 寄存器说明

SDIOC 模块根据 SD Host Controller Standard Specification 标准设计，因此寄存器也与该标准描述相同，未使用的地址和位改为保留位(Reserved)。

下表为 SDIOC 模块的寄存器列表。

BASE ADDR: 0x4006FC00 (SDIOC1)、0x40070000 (SDIOC2)

寄存器名	符号	偏移地址	位宽	复位值
数据块长度	BLKSIZE	0x04	16	0x0000h
数据块计数	BLKCNT	0x06	16	0x0000h
参数0	ARG0	0x08	16	0x0000h
参数1	ARG1	0x0A	16	0x0000h
传输模式	TRANSMODE	0x0C	16	0x0000h
命令	CMD	0x0E	16	0x0000h
应答0	RESP0	0x10	16	0x0000h
应答1	RESP1	0x12	16	0x0000h
应答2	RESP2	0x14	16	0x0000h
应答3	RESP3	0x16	16	0x0000h
应答4	RESP4	0x18	16	0x0000h
应答5	RESP5	0x1A	16	0x0000h
应答6	RESP6	0x1C	16	0x0000h
应答7	RESP7	0x1E	16	0x0000h
数据缓冲0	BUF0	0x20	16	0x0000h
数据缓冲1	BUF1	0x22	16	0x0000h
主机状态	PSTAT	0x24	32	0x00000000h
主机控制	HOSTCON	0x28	8	0x00h
电源控制	PWRCON	0x29	8	0x00h
数据块间隔控制	BLKGPCON	0x2A	8	0x00h
时钟控制	CLKCON	0x2C	16	0x0002h
超时控制	TOUTCON	0x2E	8	0x00h
软件复位	SFTRST	0x2F	8	0x00h
普通中断状态	NORINTST	0x30	16	0x0000h

错误中断状态	ERRINTST	0x32	16	0x0000h
普通中断状态使能	NORINTSTEN	0x34	16	0x0000h
错误中断状态使能	ERRINTSTEN	0x36	16	0x0000h
普通中断信号使能	NORINTSGEN	0x38	16	0x0000h
错误中断信号使能	ERRINTSGEN	0x3A	16	0x0000h
自动命令错误状态	ATCERRST	0x3C	16	0x0000h
强制自动命令错误状态控制	FEA	0x50	16	0x0000h
强制错误状态控制	FEE	0x52	16	0x0000h

此外 SDIOC1 和 SDIOC2 共用一个 MMC 模式使能寄存器，用于控制器在 SD 和 MMC 模式之间切换

寄存器名	符号	地址	位宽	复位值
MMC 模式使能寄存器	MMCER	0x40055404	32	0x00000000h

35.3.1 数据块长度寄存器(BLKSIZE)

偏移地址： 0x04

复位值： 0x0000

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

位	标记	位名	功能	读写
b15~b12	Reserved	-	读时读出0, 写时请写0	R
b11~0	TBS	数据块长度	设定传输的数据块长度(Transfer Block Size), 传输数据块的长度以字节为单位, 设定范围是1~512	R/W

35.3.2 数据块计数器(BLKCNT)

偏移地址： 0x06

复位值： 0x0000

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

位	标记	位名	功能	读写
b15~b0	BC	数据块计数	设定传输数据块的数量(Block count), 设定该寄存器需要在传输停止时进行, 并且需要传输模式寄存器的数据块计数使能位(BCE)有效	R/W

35.3.3 参数寄存器 0(ARG0)

偏移地址： 0x08

复位值： 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ARG0[15:0]															

位	标记	位名	功能	读写
b15~b0	ARG[15:0]	命令参数	设定当前发送命令包含的参数，该寄存器为参数的低16位	R/W

35.3.4 参数寄存器 1(ARG1)

偏移地址： 0x0A

复位值： 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ARG1[15:0]															

位	标记	位名	功能	读写
b15~b0	ARG[15:0]	命令参数	设定当前发送命令包含的参数，该寄存器为参数的高16位	R/W

35.3.5 传输模式寄存器(TRANSMODE)

偏移地址： 0x0C

复位值： 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved										MULB	DDIR	ATCEN[1:0]	BCE	Rsvd	

位	标记	位名	功能	读写
b15~b6	Reserved	-	读时读出0, 写时请写0	R
b5	MULB	多数据块	0: 当前传输为单数据块传输(single block) 1: 当前传输为多数据块传输(multi block)	R/W
b4	DDIR	数据传输方向	0: 写操作(主机发送数据) 1: 读操作(主机接收数据)	R/W
b3~b2	ATCEN	自动命令使能	00: 不发送自动命令 01: 多块传输结束后自动发送CMD12 10: 禁止设定 11: 禁止设定	R/W
b1	BCE	块计数使能	0: 禁用数据块计数使能 1: 允许数据块计数使能	R/W
b0	Reserved	-	读时读出0, 写时请写0	R

35.3.6 命令寄存器(CMD)

偏移地址： 0x0E

复位值： 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved						IDX[5:0]		TYP[1:0]	DAT	ICE	CCE	Rsvd			RESTYP[1:0]

位	标记	位名	功能	读写
b15~b14	Reserved	-	读时读出0, 写时请写0	R
b13~b8	IDX	命令编号	发送命令的编号	R/W
b7~b6	TYP	命令类型	00: 普通命令 01: 挂起(suspend)命令 10: 恢复(resume)命令 11: 终止(abort)命令	R/W
b5	DAT	带数据命令	0: 当前命令仅使用SDIOx_CMD(x=1~2)命令线 1: 当前命令需要使用SDIOx_Dy(x=1~2)数据线	R/W
b4	ICE	编号检查	0: 不检查应答(response)中的命令编号 1: 检查应答(response)中的命令编号	R/W
b3	CCE	CRC检查	0: 不检查应答(response)中的CRC校验码 1: 检查应答(response)中的CRC校验码	R/W
b2	Reserved	-	读时读出0, 写时请写0	R
b1~b0	RESTYP	应答类型	00: 该命令无应答 01: 命令具有长度为136bit的应答 10: 命令具有长度为48bit的应答 11: 命令具有长度为48bit并且带有忙(busy)状态的应答	R/W

35.3.7 应答寄存器 0(RESP0)

偏移地址： 0x10

复位值： 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESP0[15:0]															

位	标记	位名	功能	读写
b15~b0	RESP0[15:0]	应答信息	应答信息的15~0位	R

35.3.8 应答寄存器 1(RESP1)

偏移地址： 0x12

复位值： 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESP1[15:0]															

位	标记	位名	功能	读写
b15~b0	RESP1[15:0]	应答信息	应答信息的31~16位	R

35.3.9 应答寄存器 2(RESP2)

偏移地址： 0x14

复位值： 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESP2[15:0]															

位	标记	位名	功能	读写
b15~b0	RESP2[15:0]	应答信息	应答信息的47~32位	R

35.3.10 应答寄存器 3(RESP3)

偏移地址： 0x16

复位值： 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESP3[15:0]															

位	标记	位名	功能	读写
b15~b0	RESP3[15:0]	应答信息	应答信息的63~48位	R

35.3.11 应答寄存器 4(RESP4)

偏移地址: 0x18

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESP4[15:0]															

位	标记	位名	功能	读写
b15~b0	RESP4[15:0]	应答信息	应答信息的79~64位	R

35.3.12 应答寄存器 5(RESP5)

偏移地址: 0x1A

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESP5[15:0]															

位	标记	位名	功能	读写
b15~b0	RESP5[15:0]	应答信息	应答信息的95~80位	R

35.3.13 应答寄存器 6(RESP6)

偏移地址: 0x1C

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESP6[15:0]															

位	标记	位名	功能	读写
b15~b0	RESP6[15:0]	应答信息	应答信息的111~96位	R

35.3.14 应答寄存器 7(RESP7)

偏移地址: 0x1E

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESP7[15:0]															

位	标记	位名	功能	读写
b15~b0	RESP7[15:0]	应答信息	应答信息的127~112位	R

35.3.15 数据缓冲寄存器 0(BUF0)

偏移地址: 0x20

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BUF0[15:0]															

位	标记	位名	功能	读写
b15~b0	BUF0	数据缓冲	写入发送数据和读取接收数据, 该寄存器为数据的低16位	R/W

35.3.16 数据缓冲寄存器 1(BUF1)

偏移地址: 0x22

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BUF1[15:0]															

位	标记	位名	功能	读写
b15~b0	BUF1	数据缓冲	写入发送数据和读取接收数据, 该寄存器为数据的高16位	R/W

35.3.17 主机状态寄存器(PSTAT)

偏移地址: 0x24

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved				CMDL	DATL[3:0]				WPL	CDL	CSS	CIN			

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved		BRE	BWE	RTA	WTA	Reserved				DA	CID	CIC			

位	标记	位名	功能	读写
b31~b25	Reserved	-	读时读出0, 写时请写0	R
B24	CMDL	命令线状态	命令线(SDIOx_CMD(x=1~2))的状态	R
B23~b20	DATL	数据线状态	数据线(SDIOx_Dy(x=1~2) (y=0~3))的状态	R
B19	WPL	写保护线状态	写保护线(SDIOx_WP(x=1~2))的状态	R
B18	CDL	卡识别线状态	卡识别线(SDIOx_CD(x=1~2))的状态	R
B17	CSS	设备稳定状态	0: 卡识别线状态不稳定 1: 卡识别线状态稳定, 设备已插入或未插入	R
B16	CIN	设备插入状态	0: 无设备插入 1: 有设备插入	R
B15~b12	Reserved	-	读时读出0, 写时请写0	R
B11	BRE	数据缓冲满	0: 数据缓冲器没有足够的数据供读取 1: 数据缓冲器有足够的数据供读取	R
B10	BWE	数据缓冲空	0: 数据缓冲器可以写入数据 1: 数据缓冲器不能写入数据	R
B9	RTA	读操作状态	0: 没有正在进行的读操作 1: 有正在进行的读操作	R
B8	WTA	写操作状态	0: 没有正在进行的写操作 1: 有正在进行的写操作	R
B7~b3	Reserved	-	读时读出0, 写时请写0	R
B2	DA	数据线传输状态	0: 数据线空闲 1: 数据线正在传输数据	R
B1	CID	带数据命令抑制	0: 允许发送带数据命令 1: 禁止发送带数据命令	R
B0	CIC	命令抑制	0: 允许发送命令 1: 禁止发送命令	R

35.3.18 主机控制寄存器(HOSTCON)

偏移地址: 0x28

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
CDSS	CDTL	EXDW		Reserved	HSEN	DW	Rsvd

位	标记	位名	功能	读写
b7	CDSS	卡识别线选择	0: 选择真实的SDIOx_CD(x=1~2)线反映卡识别状态 1: 选择卡识别测试信号反映卡识别状态	R/W
b6	CDTL	卡识别测试信号状态	0: 卡识别测试信号为低电平(有设备插入) 1: 卡识别测试信号为高电平(无设备插入)	R/W
b5	EXDW	扩展数据位宽	0: 数据线位宽使用DW位的设定 1: 数据线位宽为8位(8bit)	R/W
b4~b3	Reserved	-	读时读出0, 写时请写0	R
b2	HSEN	高速模式使能	0: 禁用高速模式(high speed mode) 1: 启用高速模式	R/W
b1	DW	数据位宽选择	0: 数据线位宽为1位(1bit) 1: 数据线位宽为4位(4bit)	R/W
b0	Reserved	-	读时读出0, 写时请写0	R

35.3.19 电源控制寄存器(PWRCON)

偏移地址: 0x29

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Reserved							PWON

位	标记	位名	功能	读写
b7~b1	Reserved	-	读时读出0, 写时请写0	R
b0	PWON	SDIOC使能	0: 禁用SDIOC 1: 启用SDIOC	R/W

35.3.20 数据块间隙控制寄存器(BLKGPCON)

偏移地址: 0x2A

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
		Reserved		IABG	RWC	CR	SABGR

位	标记	位名	功能	读写
b7~b4	Reserved	-	读时读出0, 写时请写0	R
b3	IABG	数据块间隙中断控制	0: 关闭数据块间隙期间接收SDIO设备中断(card interrupt) 1: 开启数据块间隙期间接收SDIO设备中断(card interrupt)	R/W
b2	RWC	读等待控制	0: 禁用读等待功能(read wait) 1: 启用读等待功能(read wait)	R/W
b1	CR	继续传输	0: 无任何效果 1: 解除因SABGR位置位而停止的传输	R/W
b0	SABGR	数据块间隙停止传输	0: 不在数据块间隙时停止传输 1: 在数据块间隙时停止传输	R/W

35.3.21 时钟控制寄存器(CLKCON)

偏移地址: 0x2C

复位值: 0x0002

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

位	标记	位名	功能	读写
B15~b8	FS	分频选择	SDIOx_CK(x=1~2)时钟分频选择, 基准时钟为EXCLK 0x80: EXCLK的256分频 0x40: EXCLK的128分频 0x20: EXCLK的64分频 0x10: EXCLK的32分频 0x08: EXCLK的16分频 0x04: EXCLK的8分频 0x02: EXCLK的4分频 0x01: EXCLK的2分频 0x00: EXCLK 其它: 禁止设定	R/W
b2	CE	SDIOx_CK(x=1~2)输出控制	0: SDIOx_CK(x=1~2)停止输出 1: SDIOx_CK(x=1~2)输出	R/W
b0	ICE	时钟使能	0: SDIOC动作时钟开启 1: SDIOC动作时钟关闭	R/W

35.3.22 超时控制寄存器(TOUTCON)

偏移地址: 0x2E

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Reserved						DTO[3:0]	

位	标记	位名	功能	读写
b7~b4	Reserved	-	读时读出0, 写时请写0	R
b3~b0	DTO	数据超时时间	设定数据线SDIOx_Dy(x=1~2)(y=0~7)超时判定的时间, 单位为EXCLK的时钟周期 0000: EXCLK×2 ¹³ 0001: EXCLK×2 ¹⁴ 1110: EXCLK×2 ²⁷ 1111: 禁止设定	R/W

35.3.23 软件复位寄存器(SFTRST)

偏移地址: 0x2F

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
		Reserved			RSTD	RSTC	RSTA

位	标记	位名	功能	读写
b7~b3	Reserved	-	读时读出0, 写时请写0	R
b2	RSTD	数据复位	复位所有数据相关的寄存器, 包含以下寄存器位: BUF0, BUF1 PSTAT.BRE, PSTAT.BWE, PSTAT.RTA, PSTAT.WTA, PSTAT.DLA, PSTAT.CID BLKGPCON.CR, BLKGPCON.SABGR NORINTST.BRR, NORINTST.BWR, NORINTST.BGE, NORINTST.TC 0: 正常工作 1: 执行复位	R/W
b1	RSTC	命令复位	复位所有命令相关的寄存器, 包含以下寄存器位: PSTAT.CIC NORINTST.CC 0: 正常工作 1: 执行复位	R/W
b0	RSTA	全部复位	复位除卡识别功能外的全部SDIOC寄存器 0: 正常工作 1: 执行复位	R/W

35.3.24 普通中断状态寄存器(NORINTST)

偏移地址: 0x30

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EI		Reserved					CINT	CRM	CIST	BRR	BWR	Rsvd	BGE	TC	CC

位	标记	位名	功能	读写
b15	EI	错误中断	发生错误中断状态寄存器(ERRINTST)中任意错误时置位	R
b14~b9	Reserved	-	读时读出0, 写时请写0	R
b8	CINT	卡中断	SDIO设备发出卡中断(card interrupt)申请时置位, SDIO设备撤销申请后复位	R
b7	CRM	卡移除	设备移除(removal)时置位, 写1复位	R/W
b6	CIST	卡插入	设备插入(insert)时置位, 写1复位	R/W
b5	BRR	缓冲器可读	缓冲器内数据可被读取(PSTAT.BRE=1)时置位, 写1复位	R/W
b4	BWR	缓冲器可写	缓冲器可写入数据(PSTAT.BWE=1)时置位, 写1复位	R/W
b3	Reserved	-	读时读出0, 写时请写0	R
b2	BGE	数据块间隙停止传输	当传输在数据块间隙停止时置位, 写1复位	R/W
b1	TC	传输完成	读写传输完成时置位, 写1复位	R/W
b0	CC	命令完成	无应答命令命令发送完成和有应答命令的应答接收完成后置位, 写1复位	R/W

35.3.25 错误中断状态寄存器(ERRINTST)

偏移地址: 0x32

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
							Reserved	ACE	Rsvd	DEBE	DCE	DTOE	CIE	CEBE	CCE	CTOE

位	标记	位名	功能	读写
b15~b9	Reserved	-	读时读出0, 写时请写0	R
b8	ACE	自动发送命令错误	自动发送命令(auto CMD)发生错误时置位, 错误的种类可以查询自动命令 错误寄存器(ATCERRST), 写1复位	R/W
b7	Reserved	-	读时读出0, 写时请写0	R
b6	DEBE	数据停止位错误	当数据线在停止位检测到低电平时置位, 写1复位	R/W
b5	DCE	数据CRC校验错误	当数据线发生CRC校验错误时置位, 写1复位	R/W
b4	DTOE	数据超时错误	当发生数据超时时置位, 写1复位, 数据超时时间由超时控制寄存器 (TOUTCON)设定	R/W
b3	CIE	命令编号错误	接收到的应答所包含的命令编号错误时置位, 写1复位	R/W
b2	CEBE	命令停止位错误	当命令线在停止位检测到低电平时置位, 写1复位	R/W
b1	CCE	命令CRC校验错误	当命令线发生CRC校验错误时置位, 写1复位	R/W
b0	CTOE	命令超时错误	当命令发送后超过64个SDIOx_CK(x=1~2)周期未收到应答时置位, 写1复位	R/W

35.3.26 普通中断状态使能寄存器(NORINTSTEN)

偏移地址: 0x34

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
							Reserved	CINTEN	CRMEN	CISTEN	BRREN	BWREN	Rsvd	BGEEN	TCEN	CCEN

位	标记	位名	功能	读写
b15~b9	Reserved	-	读时读出0, 写时请写0	R
b8	CINTEN	卡中断状态使能	0: 禁止NORINTST.CINT置位 1: 允许NORINTST.CINT置位	R
b7	CRMEN	卡移除状态使能	0: 禁止NORINTST.CRM置位 1: 允许NORINTST.CRM置位	R/W
b6	CISTEN	卡插入状态使能	0: 禁止NORINTST.CIST置位 1: 允许NORINTST.CIST置位	R/W
b5	BRREN	缓冲器可读状态使能	0: 禁止NORINTST.BRR置位 1: 允许NORINTST.BRR置位	R/W
b4	BWREN	缓冲器可写状态使能	0: 禁止NORINTST.BWR置位 1: 允许NORINTST.BWR置位	R/W
b3	Reserved	-	读时读出0, 写时请写0	R
b2	BGEEN	数据块间隙停止传输状态使能	0: 禁止NORINTST.BGE置位 1: 允许NORINTST.BGE置位	R/W
b1	TCEN	传输完成状态使能	0: 禁止NORINTST.TC置位 1: 允许NORINTST.TC置位	R/W
b0	CCEN	命令完成状态使能	0: 禁止NORINTST.CC置位 1: 允许NORINTST.CC置位	R/W

35.3.27 错误中断状态使能寄存器(ERRINTSTEN)

偏移地址: 0x36

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
							ACEEN	Rsvd	DEBEEN	DCEEN	DTOEEN	CIEEN	CEBEEN	CCEEN	CTOEEN

位	标记	位名	功能	读写
b15~b9	Reserved	-	读时读出0, 写时请写0	R
b8	ACEEN	自动发送命令错误 状态使能	0: 禁止ERRINTST.ACE置位 1: 允许ERRINTST.ACE置位	R/W
b7	Reserved	-	读时读出0, 写时请写0	R
b6	DEBEEN	数据停止位错误状 态使能	0: 禁止ERRINTST.DEBE置位 1: 允许ERRINTST.DEBE置位	R/W
b5	DCEEN	数据CRC校验错误 状态使能	0: 禁止ERRINTST.DCE置位 1: 允许ERRINTST.DCE置位	R/W
b4	DTOEEN	数据超时错误状态 使能	0: 禁止ERRINTST.DTOE置位 1: 允许ERRINTST.DTOE置位	R/W
b3	CIEEN	命令编号错误状态 使能	0: 禁止ERRINTST.CIE置位 1: 允许ERRINTST.CIE置位	R/W
b2	CEBEEN	命令停止位错误状 态使能	0: 屏蔽ERRINTST.CEBE置位 1: 允许ERRINTST.CEBE置位	R/W
b1	CCEEN	命令CRC校验错误 状态使能	0: 禁止ERRINTST.CCE置位 1: 允许ERRINTST.CCE置位	R/W
b0	CTOEEN	命令超时错误状态 使能	0: 禁止ERRINTST.CTOE置位 1: 允许ERRINTST.CTOE置位	R/W

35.3.28 普通中断信号使能寄存器(NORINTSGEN)

偏移地址: 0x38

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
							Reserved	CINTSEN	CRMSEN	CISTSEN	BRRSEN	BWRSEN	Rsvd	BGESEN	TCESN	CCSEN

位	标记	位名	功能	读写
b15~b9	Reserved	-	读时读出0, 写时请写0	R
b8	CINTSEN	卡中断信号使能	0: 禁止NORINTST.CINT申请中断 1: 允许NORINTST.CINT申请中断	R
b7	CRMSEN	卡移除信号使能	0: 禁止NORINTST.CRM申请中断 1: 允许NORINTST.CRM申请中断	R/W
b6	CISTSEN	卡插入信号使能	0: 禁止NORINTST.CIST申请中断 1: 允许NORINTST.CIST申请中断	R/W
b5	BRRSEN	缓冲器可读信号使能	0: 禁止NORINTST.BRR申请中断 1: 允许NORINTST.BRR申请中断	R/W
b4	BWRSEN	缓冲器可写信号使能	0: 禁止NORINTST.BWR申请中断 1: 允许NORINTST.BWR申请中断	R/W
b3	Reserved	-	读时读出0, 写时请写0	R
b2	BGESEN	数据块间隙停止传输信号使能	0: 禁止NORINTST.BGE申请中断 1: 允许NORINTST.BGE申请中断	R/W
b1	TCSEN	传输完成信号使能	0: 禁止NORINTST.TC申请中断 1: 允许NORINTST.TC申请中断	R/W
b0	CCSEN	命令完成信号使能	0: 禁止NORINTST.CC申请中断 1: 允许NORINTST.CC申请中断	R/W

35.3.29 错误中断信号使能寄存器(ERRINTSGEN)

偏移地址: 0x3A

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
						Reserved	ACEESN	Rsvd	DEBESEN	DCESEN	DTOESEN	CIESEN	CEBESEN	CCESEN	CTOESEN

位	标记	位名	功能	读写
b15~b9	Reserved	-	读时读出0, 写时请写0	R
b8	ACESEN	自动发送命令错误	0: 禁止ERRINTST.ACE申请中断 1: 允许ERRINTST.ACE申请中断	R/W
b7	Reserved	-	读时读出0, 写时请写0	R
b6	DEBESEN	数据停止位错误信号使能	0: 禁止ERRINTST.DEBE申请中断 1: 允许ERRINTST.DEBE申请中断	R/W
b5	DCESEN	数据CRC校验错误	0: 禁止ERRINTST.DCE申请中断 1: 允许ERRINTST.DCE申请中断	R/W
b4	DTOESEN	数据超时错误信号使能	0: 禁止ERRINTST.DTOE申请中断 1: 允许ERRINTST.DTOE申请中断	R/W
b3	CIESEN	命令编号错误信号使能	0: 禁止ERRINTST.CIE申请中断 1: 允许ERRINTST.CIE申请中断	R/W
b2	CEBESEN	命令停止位错误信号使能	0: 屏蔽ERRINTST.CEBE申请中断 1: 允许ERRINTST.CEBE申请中断	R/W
b1	CCESEN	命令CRC校验错误信号使能	0: 禁止ERRINTST.CCE申请中断 1: 允许ERRINTST.CCE申请中断	R/W
b0	CTOESEN	命令超时错误信号使能	0: 禁止ERRINTST.CTOE申请中断 1: 允许ERRINTST.CTOE申请中断	R/W

35.3.30 自动命令错误状态寄存器(ATCERRST)

偏移地址: 0x3C

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
								CMDE		Reserved	IE	EBE	CE	TOE	NE

位	标记	位名	功能	读写
b15~b8	Reserved	-	读时读出0, 写时请写0	R
b7	CMDE	未发送错误	当本寄存器b4~b0位对应错误发生导致其它命令未发送时置位	R
b6~b5	Reserved	-	读时读出0, 写时请写0	R
b4	IE	命令编号错误	接收到的应答所包含的自动命令的编号错误时置位	R
b3	EBE	停止位错误	当命令线在停止位检测到低电平时置位	R
b2	CE	数据超时错误	当命令线发生CRC校验错误时置位	R
b1	TOE	命令超时错误	当自动命令发送后超过64个SDIOx_CK(x=1~2)周期未收到应答时置位	R
b0	NE	未执行错误	当自动命令因其它原因未发送时置位	R

35.3.31 强制自动命令错误状态控制寄存器(FEA)

偏移地址: 0x50

复位值: 0x0000

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

位	标记	位名	功能	读写
b15~b8	Reserved	-	读时读出0, 写时请写0	R
b7	FCMDE	强制未发送错误	0: 无任何效果 1: 强制发生ATCERRST.CMDE错误	W
b6~b5	Reserved	-	读时读出0, 写时请写0	W
b4	FIE	强制命令编号错误	0: 无任何效果 1: 强制发生ATCERRST.IE错误	W
b3	FEBE	强制停止位错误	0: 无任何效果 1: 强制发生ATCERRST.EBE错误	W
b2	FCE	强制数据超时错误	0: 无任何效果 1: 强制发生ATCERRST.CE错误	W
b1	FTOE	强制命令超时错误	0: 无任何效果 1: 强制发生ATCERRST.TOE错误	W
b0	FNE	强制未执行错误	0: 无任何效果 1: 强制发生ATCERRST.NE错误	W

35.3.32 强制错误状态控制寄存器(FEE)

偏移地址: 0x52

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
							Reserved	FACE	Rsvd	FDEBE	FDCE	FDTOE	FCIE	FCEBE	FCCE	FCTOE

位	标记	位名	功能	读写
b15~b9	Reserved	-	读时读出0, 写时请写0	R
b8	FACE	强制自动发送命令	0: 无任何效果 1: 强制发生ERRINTST.ACE错误	W
b7	Reserved	-	读时读出0, 写时请写0	R
b6	FDEBE	强制数据停止位错	0: 无任何效果 1: 强制发生ERRINTST.DEBE错误	W
b5	FDCE	强制数据CRC校验	0: 无任何效果 1: 强制发生ERRINTST.DCE错误	W
b4	FDTOE	强制数据超时错误	0: 无任何效果 1: 强制发生ERRINTST.DTOE错误	W
b3	FCIE	强制命令编号错误	0: 无任何效果 1: 强制发生ERRINTST.CIE错误	W
b2	FCEBE	强制命令停止位错	0: 无任何效果 1: 强制发生ERRINTST.CEBE错误	W
b1	FCCE	强制命令CRC校验	0: 无任何效果 1: 强制发生ERRINTST.CCE错误	W
b0	FCTOE	强制命令超时错误	0: 无任何效果 1: 强制发生ERRINTST.CTOE错误	W

35.3.33 MMC 模式使能寄存器(MMCER)

地址: 0x40055404

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
<hr/>															
位	标记	位名	功能	读写											
b31~b4	Reserved	-	读出时为“0”,写入时写“0”	R											
b3	SELMMC2	SDIOC2 MMC模式 使能	0: SDIOC2选择SD模式 1: SDIOC2选择MMC模式	R/W											
b2	Reserved	-	读出时为“0”,写入时写“0”	R											
b1	SELMMC1	SDIOC1 MMC模式 使能	0: SDIOC1选择SD模式 1: SDIOC1选择MMC模式	R/W											
b0	Reserved	-	读出时为“0”,写入时写“0”	R											

36 调试控制器（DBG）

本产品参考如下 ARM 技术文档：

- Cortex™-M4F r0p1 技术参考手册 (TRM)
- ARM 调试接口 V5
- ARM CoreSight 设计套件版本 r0p1 技术参考手册

36.1 简介

本 MCU 的内核是 Cortex™-M4F，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）或取访问数据（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。本 MCU 未搭载 ETM 调试器件。

提供两个调试接口：

- 串行调试跟踪接口 SWD
- 并行调试跟踪接口 JTAG

36.2 DBGc 系统框图

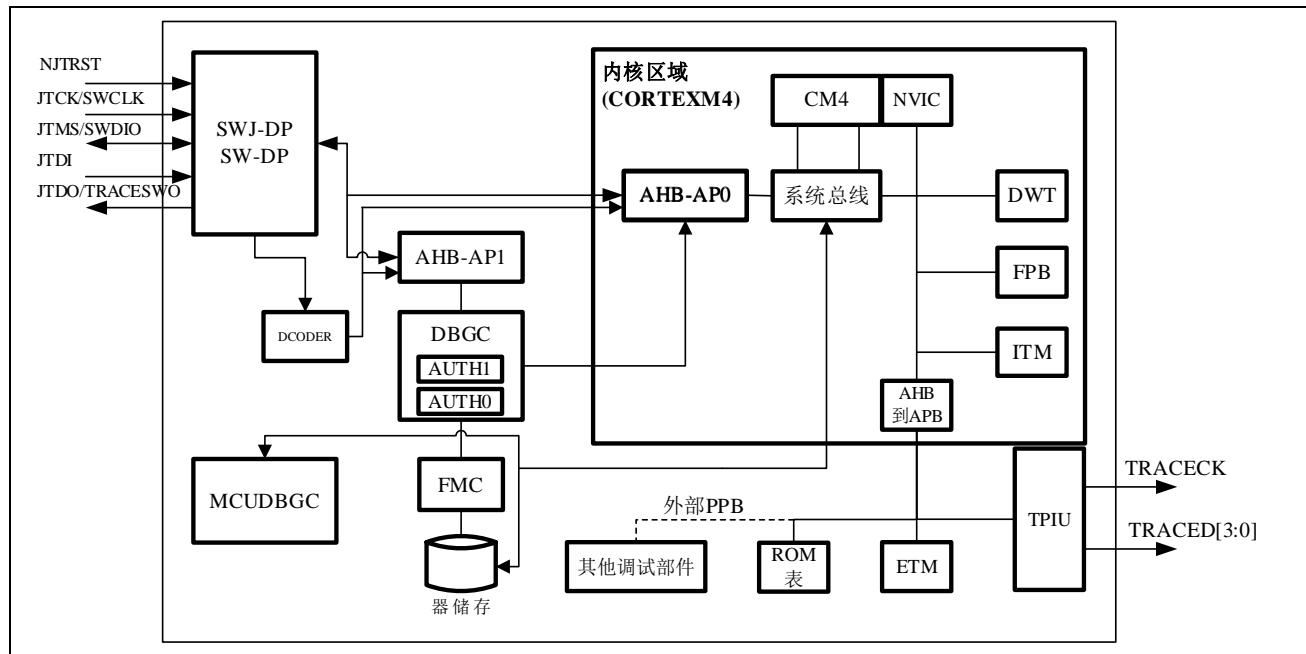


图 36-1 调试控制系统

ARM Cortex™-M4F 内核提供集成片上调试支持。它包括：

- SWJ-DP: SWD/JTAG 调试端口
- AHB-AP: AHB 访问端口
- ITM: 指令跟踪单元
- FPB: Flash 指令断点
- DWT: 数据断点触发
- TPIU: 跟踪端口单元接口（大封装上提供，其中会映射相应引脚）
- 灵活调试引脚分配

注意：

- 有关 ARM Cortex™-M4F 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》。

36.3 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 AHP-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 AHP-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。

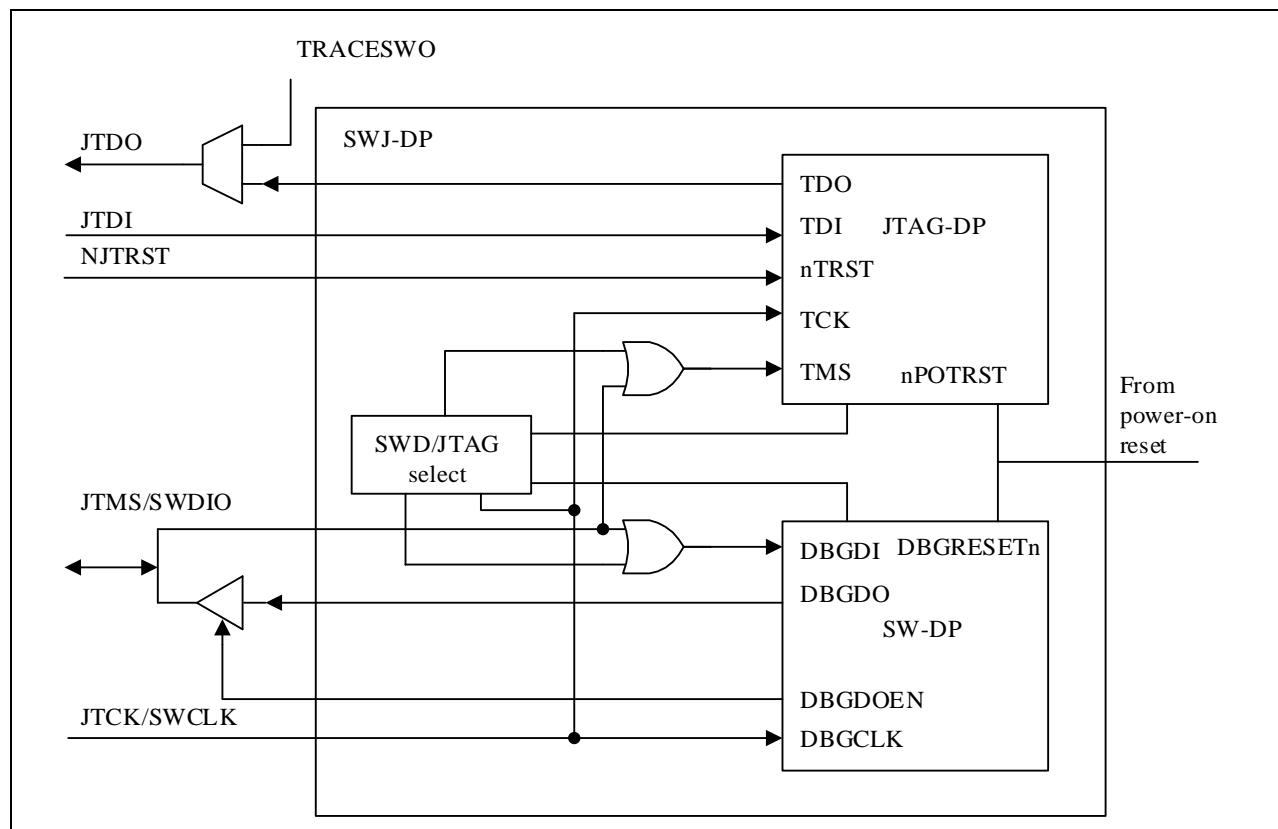


图 36-2 调试控制系统

36.3.1 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

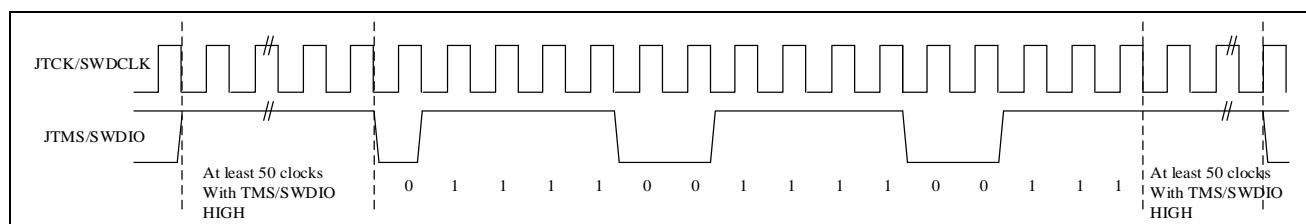


图 36-3 JTAG-DP 到 SW-DP 切换时序

36.4 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。

36.4.1 SWJ 调试端口引脚

MCU 的 5 个普通 I/O 口可用作 SWJ-DP 接口引脚。

SWJ-DP 引脚名称	JTAG调试端口		SW调试端口	
	类型	说明	类型	调试分配
JTMS/SWDIO	I	JTAG模式选择	I/O	串行线数据输入/输出
JTCK/SWCLK	I	JTAG时钟	I	串行时钟
JTDI	I	JTAG数据输入	-	-
JTDO/TRACESWO	O	JTAG数据输出	-	TRACESWO(如果使能异步跟踪)
NJTRST	I	JTAG复位	-	-

表 36-1 SWJ 调试端口引脚

36.4.2 灵活的 SWJ-DP 引脚分配

复位（上电或引脚复位）后，会将用于 SWJ-DP 的全部 5 个引脚指定为专用引脚，可供调试工具立即使用（请注意，除非明确编程，否则不分配跟踪输出）。但是，MCU 可以禁止部分或全部 SWJ-DP 端口，进而释放相关引脚以用作 GPIO。有关如何禁止 SWJ-DP 端口引脚的更多详细信息，请参见：通用 IO 特殊控制寄存器 PSPCR。

可用的调试端口	分配的SWJ IO引脚				
	JTMS/ SWDIO	JTCK/ SWCLK	JTDI	JTDO	NJTRST
全部SWJ (JTAG-DP+SW-DP) -复位状态	√	√	√	√	√
禁止JTAG-DP和使能SW-DP	√	√	可释放	可释放	可释放
禁止JTGA-DP和禁止SW-DP	可释放	可释放	可释放	可释放	可释放

表 36-2 灵活的 SWJ-DP 引脚分配

36.4.3 JTAG 引脚上的内部上拉

根据 JTAG IEEE 标准，必须确保 JTAG 输入引脚不悬空，因为这些引脚直接连接到 MCU 内部用于控制调试功能。还必须特别注意 JTCK/SWCLK 引脚，该引脚直接用于调试控制时钟功能。为避免 IO 电平浮空，MCU 在 JTAG 引脚上内置有内部上拉电阻：

- NJTRST：内部上拉
- JTDI：内部上拉
- JTMS/SWDIO：内部上拉
- JTCK/SWCLK：内部上拉
- JTDO：高阻状态

在未连接调试器状态下，用户软件可以通过设定 GPIO 特殊控制寄存器释放 JTAG IO 作为普通的 I/O 口使用。由于芯片内部上拉为<100K 欧的弱上拉，建议采用外部 10K 欧的外部上拉。

36.4.4 使用串行接口以及释放未使用的调试引脚以作 GPIO

使用 SWD 时可以释放一些 GPIO，用户软件必须在 GPIO 控制寄存器中更改 GPIO 配置，这样便可释放相应的引脚用作 GPIO。

调试时，主机执行以下操作：

- 在系统复位状态下，分配所有 SWJ 引脚(JTAG-DP+SW-DP)。
- 在系统复位状态下，调试主机发送 JTAG 序列，以从 JTAG-DP 切换到 SW-DP。
- 仍然在系统复位状态下，调试主机在复位地址处设置断点。
- 释放复位信号，内核停止在复位地址处。
- 从此调试端口切换为 SW-DP。然后通过用户软件将其它 JTAG 引脚重新分配为 GPIO。

注意：

- 对于用户软件设计，需要释放调试引脚时，在复位后一直到用户软件释放引脚这段期间，这些引脚仍然处于输入上拉 (NJTRST、JTMS、JTDI、JTCK 和 JTDO)

36.5 寄存器

寄存器描述如下：

基地址：0xE004_2000

寄存器名	符号	偏移地址	位宽	初始值	访问主机
DBG状态寄存器	MCUDBGSTAT	0x001C	32	0x00000000	CPU/调试IDE*
外设调试暂停寄存器	MCUSTPCTL	0x0020	32	0x00000003	CPU/调试IDE*
TRACE端口控制寄存器	MCUTRACECTL	0x0024	32	0x00000000	CPU/调试IDE*

注意：

- 寄存器位于 PPB 区域，只能在特权模式下 CPU 才能进行访问。

36.5.1 DBG 状态寄存器 (MCUDBGSTAT)

DBG 调试上电状态确认寄存器。

复位值: 0x0000_0001

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

位	标记	位名	功能	读写
b31~2	Reserved	-	读出时为“0”，写入时写“0”	R/W
b1	CDBG_PWRUPACK	调试器上电反馈	0: 无反馈 1: 调试上电反馈	R/W
b0	CDBG_PWRUPREQ	调试器上电请求	0: 无上电请求 1: 上电请求	R/W

36.5.2 外设调试暂停寄存器 (MCUSTPCTL)

当 CPU 处于调试状态时，周边模块暂停控制。

复位值：0x0000_0003

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TMR	-	-	-	-											
A6	A5	A4	A3	A2	A1	63	62	61	43	42	41				
STP															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TMR	TMR	-	-	-	-	-	-	-	PVD	PVD	PVD	RTC	WDT	SWD	
02	01								2	1	0	STP	STP	TSTP	
STP	STP								STP	STP	STP				

位	标记	位名	功能	读写
b31	TMRA6STP	TimerA-6计数暂停信号	0: 即使内核停止，计数器仍然计数 1: 内核停止时，计数器暂停计数	R/W
b30	TMRA5STP	TimerA-5计数暂停信号	0: 即使内核停止，计数器仍然计数 1: 内核停止时，计数器暂停计数	R/W
b29	TMRA4STP	TimerA-4计数暂停信号	0: 即使内核停止，计数器仍然计数 1: 内核停止时，计数器暂停计数	R/W
b28	TMRA3STP	TimerA-3计数暂停信号	0: 即使内核停止，计数器仍然计数 1: 内核停止时，计数器暂停计数	R/W
b27	TMRA2STP	TimerA-2计数暂停信号	0: 即使内核停止，计数器仍然计数 1: 内核停止时，计数器暂停计数	R/W
b26	TMRA1STP	TimerA-1计数暂停信号	0: 即使内核停止，计数器仍然计数 1: 内核停止时，计数器暂停计数	R/W
b25	TMR63STP	Timer6-3计数暂停信号	0: 即使内核停止，计数器仍然计数 1: 内核停止时，计数器暂停计数	R/W
b24	TMR62STP	Timer6-2计数暂停信号	0: 即使内核停止，计数器仍然计数 1: 内核停止时，计数器暂停计数	R/W
b23	TMR61STP	Timer6-1计数暂停信号	0: 即使内核停止，计数器仍然计数 1: 内核停止时，计数器暂停计数	R/W
b22	TMR43STP	Timer4-3计数暂停信号	0: 即使内核停止，计数器仍然计数 1: 内核停止时，计数器暂停计数	R/W
b21	TMR42STP	Timer4-2计数暂停信号	0: 即使内核停止，计数器仍然计数 1: 内核停止时，计数器暂停计数	R/W
b20	TMR41STP	Timer4-1计数暂停信号	0: 即使内核停止，计数器仍然计数 1: 内核停止时，计数器暂停计数	R/W
b19	Reserved	-	读出时为“0”，写入时写“0”	R/W
b18	Reserved	-	读出时为“0”，写入时写“0”	R/W
b17	Reserved	-	读出时为“0”，写入时写“0”	R/W

b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15	TMR02STP	Timer0-2计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b14	TMR01STP	Timer0-1计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	R/W
b13~6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5	PVD2STP	PVD2中断/复位屏蔽	0: 即使内核停止, 仍然产生PVD2中断申请或复位 1: 内核停止时, 屏蔽PVD2中断申请或复位	R/W
b4	PVD1STP	PVD1中断/复位屏蔽	0: 即使内核停止, 仍然产生PVD1中断申请或复位 1: 内核停止时, 屏蔽PVD1中断申请或复位	R/W
b3	PVD0STP	PVD0中断/复位屏蔽	0: 即使内核停止, 仍然产生PVD0中断申请或复位 1: 内核停止时, 屏蔽PVD0中断申请或复位	R/W
b2	RTCSTP	RTC计数暂停信号	0: 即使内核停止, RTC计数器仍然计数 1: 内核停止时, RTC计数器暂停计数	R/W
b1	WDTSTP	WDT计数暂停信号	0: 即使内核停止, WDT计数器仍然计数 1: 内核停止时, WDT计数器暂停计数	R/W
b0	SWDTSTP	SWDT计数暂停信号	0: 即使内核停止, SWDT计数器仍然计数 1: 内核停止时, SWDT计数器暂停计数	R/W

36.5.3 调试组件配置寄存器 (MCUTRACECTL)

通过该寄存器配置 TRACE 输出引脚。

复位值: 0x0000_0000

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

位	标记	位名	功能	读写
b31~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	TRACEIOEN	TRACE引脚输出控制	0: 同步跟踪引脚输出禁止 1: 同步跟踪引脚输出许可	R/W
b1~b0	TRACEMODE	TRACEDATA输出引脚控制	00: 异步跟踪 01: 同步跟踪1位TRACEDATA[0] 10: 同步跟踪2位TRACEDATA[1:0] 11: 同步跟踪4位TRACEDATA[3:0]	R/W

36.6 SW 调试端口

36.6.1 SW 协议简介

同步串行协议使用两个引脚：

- SWCLK：从主机到从机的时钟
- SWDIO：双向

传输数据时，LSB 在前。

对于 SWCLK 和 SWDIO，需要在电路板上对线路进行上拉（建议采用 10 K 欧）。

36.7 TPIU (跟踪端口接口单元)

36.7.1 简介

TPIU 是 ITM 与片上跟踪数据之间的桥接器。

输出数据流封装成跟踪源 ID，然后被跟踪端口分析器 (TPA) 捕获。

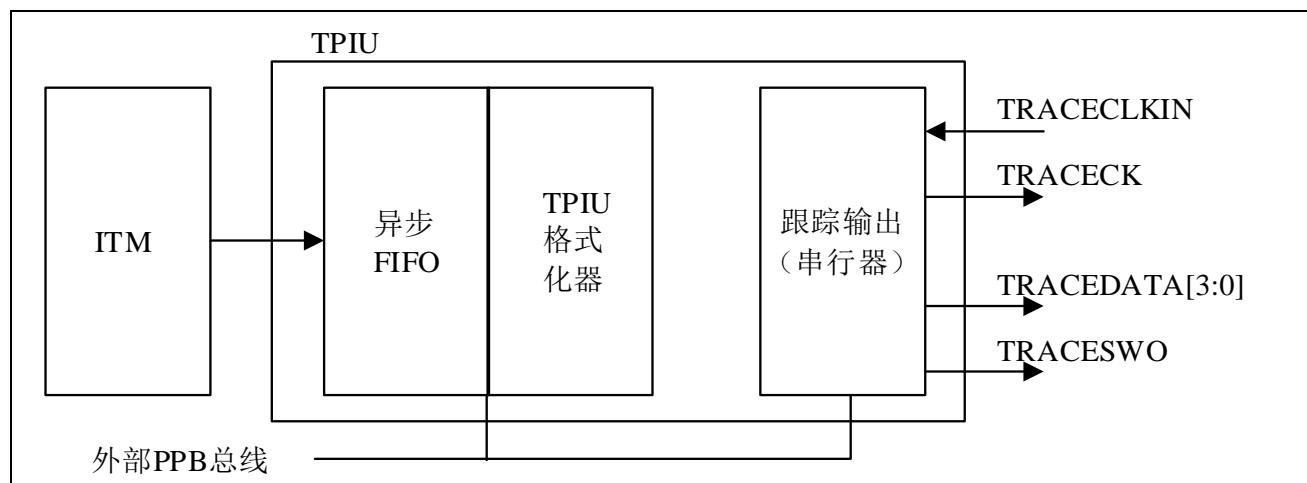


图 36-4 TPIU 框图

36.7.2 TRACE 引脚分配

- 异步模式

异步模式需要 1 个额外引脚，并且适用于所有封装。仅在使用串行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	O	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	O	TRACE时钟
TRACED[3:0]	O	TRACE同步数据输出，可以是1、2或4.

TPIU TRACE 引脚分配

默认情况下，不分配这些引脚。可通过将 MCU 调试组件配置寄存器(MCUTRACECTL) 中的 TRACE_IOEN 和 TRACE_MODE 位来配置这些引脚。必须由调试主机或 CPU 来完成此配置。

此外，要分配的引脚数目取决于跟踪配置（异步跟踪或同步跟踪）。

- 异步模式：需要 1 个额外引脚
- 同步模式：需要 5 个额外引脚
 - TRACECK
 - TRACED[0]（如果端口数据长度配置为 1, 2 或 4）
 - TRACED[1]（如果端口数据长度配置为 2 或 4）
 - TRACED[2]（如果端口数据长度配置 4）
 - TRACED[3]（如果端口数据长度配置 4）

要分配 TRACE 引脚，调试主机必须对 MCU 调试配置寄存器 (MCUTRACECTL) 的位 TRACE_IOEN 和 TRACE_MODE[1:0] 进行编程。默认情况下不分配 TRACE 引脚。此寄存器映射到外部 PPB 总线上，通过上电（而非引脚复位）复位。可在引脚复位状态下通过调试器写入该寄存器。

TPIU引脚用途	分配的TRACE IO引脚					
	JTDO/ TRACESWO	TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]
无跟踪 (默认状态) TRACE_IOEN =0 TRACE_MODE=XX	释放*	释放	释放	释放	释放	释放
异步跟踪 TRACE_IOEN =1 TRACE_MODE=00	TRACESWO	释放	释放	释放	释放	释放
同步跟踪1位 TRACE_IOEN =1 TRACE_MODE=01	释放*	TRACECK	TRACED[0]	释放	释放	释放
同步跟踪2位 TRACE_IOEN =1 TRACE_MODE=10	释放*	TRACECK	TRACED[0]	TRACED[1]	释放	释放
同步跟踪4位 TRACE_IOEN =1 TRACE_MODE=11	释放*	TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]

注意：

- 使用串行模式时，释放此引脚。但使用 JTAG 时，此引脚分配给 TDO。

36.7.3 MCU 内部 TRACECLKIN 连接

在本 MCU 中，TPIU 的时钟 TRACECLKIN 连接到内部时钟。MCU 的默认时钟是内部 MRC 振荡器。复位状态下的频率与复位释放后的频率不同。原因是，由于系统复位状态下采用默认的 MRC 校准值，而在每次系统复位释放时会更新该 MRC 校准值。因此，跟踪端口分析器 (TPA) 在系统复位状态下不应使能跟踪(使用 TRACE_IOEN 位)，原因是，在复位状态下的同步帧包的比特宽度与复位后的包不同。

36.7.4 TPIU 寄存器

仅当调试异常和监视控制寄存器 (DEMCR) 的位 TRCENA 置 1 时才能对 TPIU APB 寄存器进行读写操作。否则，这些寄存器将读为零（此位的输出会使能 TPIU 的时钟）。

36.7.5 TPIU 配置示例

- 将调试异常和监视控制寄存器 (DEMCR) 中的位 TRCENA 置 1
- 将所需值写入 TPIU 当前端口大小寄存器（对于 1 位端口大小，默认值为 0x1）
- 将 0x102 写入 TPIU 格式化器和刷新控制寄存器（默认值）
- 写入 TPIU 选择引脚协议以选择同步模式或异步模式。示例：0x2 表示异步 NRZ 模式（类似于 USART）
- 将 0x20 写入 MCUTRACECTL 控制寄存器（位 IO_TRACEN），为异步模式分配 TRACE I/O。
- 此时发送 TPIU 同步数据包 (FF_FF_FF_7F)
- 配置 ITM 并对 ITM 激励寄存器进行写操作以输出值

版本信息 & 联系方式

版本	日期	修订内容摘要
Rev1.0	2019/11/12	初版发布。
Rev1.1	2020/1/10	1) 全文增加 256KB 产品描述; 2) 全文增加 VFBGA 封装描述; 3) 初始化配置 (ICG) 描述笔误修改; 4) 控制寄存器 (WDT_CR) 笔误修改; 5) 表 34-1 CRC_RESTLT 修改为 CRC_RESLT; 6) 电气特性中修改掉电模式 105°C 的电流 max 值。
Rev1.2	2020/8/26	1) 增加超高速运行模式描述、更新 CoreMark/DMIPS、增加超高速模拟与高速模式、超低速模式之间的切换流程、增加超高速模式下 BOR/PVD 特性，电流特性。更新 200Mhz 时 SRAM 和 Flash 的等待周期，读端口时的等待周期。更新总线架构里的频率值，更新功能框图。 2) 引脚配置图增加 256KB 的型号 3) 增加 AOS 章节，增加各个 AOS 目标寄存器公共触发源使能位 4) 图 6.1 TCK_SWCLK 修改为 JTCK_SWCLK。TCK 修改为 JTCK 6.10.1 修改错别字： 6.11.7 CMU_XTAL32NFR 的 bit1-0 误记修改 6.11.13 “频率校准需在 LRC 频率保证范围内”笔误修改； 6.11.15 CMU_PLLCFGGR 的 bit27-24 误记修改 5) 7.4.3 增加为降低功耗，进入 STOP 模式前置位 PWR_FCG3 相应的 AD 功能使能位的说明； 6) 7.7 更新 PWR_PWRC3/PWR_XTAL32CS/PWR_STPMCR 的 复位 值； 删除 PWR_PDWKE0/PWR_PDWKE1 中的 WKEN_m 位的下划线； 删除 PWR_PDWKF1/PWR_PDWKE2 中 XTAL32 停止唤醒相关位； PWR_FCG0/PWR_FCG1/PWR_FCG2 的功能说明更加详细化； 7.7.13 统一 PWR_FCG 寄存器、10.2 SRAM 寄存器中 Ret-SRAM/SRAMHS/SRAMECC 的符号 统一电源控制(PWC)中预留位的描述方式 更新 PWR_PVDDSR 寄存器中 PVD2DETFLG 和 PVD1DETFLG 位的清除说明。 7) 9.6.3 修改步骤 10； 9.9.2 b1 FSLP 修改为 FSTP 9.9.3 b24 CRST0 修改为 CRST 9 概述擦除修改为扇区擦除 9.9.7 RDCOLERRITE 修改为 COLERRITE，读冲突修改为读写冲突 8) 12.3 中断向量表中 USARTx_EI /USARTx_RTO 分别修改为 USARTx_REI/USARTx_RTOI 12.4.9 增加“使用内部触发事件需要清零 PWR_FCG0.AOS 位，使能外设电路触发功能。” 12.5.2 INT_NMIENR.PRENR 修改为 INT_NMIENR.REPENR INT_NMIENR.RDEDENR 修改为 INT_NMIENR.RECCENR 12.5.3 INT_NMIFR.RPEFR 修改为 INT_NMIFR.REPFR INT_NMIFR.RDEDFFR 修改为 INT_NMIFR.RECCFFR 12.5.4 INT_NMICFR.RPECFFR 修改为 INT_NMICFR.REPCFFR INT_NMICFR.RDEDCFFR 修改为 INT_NMICFR.RECCCFR 12.5.6 INT{EIFR} 寄存器位描述中删除 INT_前缀 12.5.7 INT_EICFR 寄存器位描述中删除 INT_前缀 12.5.10 INT_VSSEL 寄存器位描述中增加前缀 V 12.5.14 INT_IER 寄存器位描述中删除 INT_前缀 9) 15.3 更新应用举例 15.4.7 更新寄存器说明

		15.3.1/15.3.2 删除非法访问动作说明 15.4.1 寄存器标记错误修改 15.4.3 b17/b16/b9/b8/b1/b0 标记和功能错误修改 10) 16.4 更新应用举例 11) 18.5.3 增加为降低功耗, 进入 STOP 模式前置位 PWR_FCG3 相应的 AD 功能使能位的说明 12) 20 章中, 优化 Timer6 部分图例; 添加[脉宽测量]、[周期测量]、[典型应用例]章节; 添加 EMB 控制的 cycle by cycle 说明; 13) 21.5.3 CCSR.DCLK 修改为 CCSR.CKDIV 14) 23 章、24 章中, 优化 Timer0、TimerA 部分图例; 修改第 23 章 TimerA CCONR 寄存器 bit9~8 及 PCONR 寄存器 bit1~0 说明 15) 25.5.3 RTC_CR2 寄存器位描述中漏记 ALMF 16) 27.5.3 USART_BRR 的初始值修改位 0x0000 FFFF 27.5.6 USART_CR3 寄存器表格中 BCN 位位宽匹配 17) 32.4.3 RAIE 修改为 RAFIE; 32.5.7CAN_RTIE 的 bit5 误记修改; 32.5.8CAN_RTIF 的 bit5 误记修改; 32.5.9 CAN_ERRINT 表格 b5 WPIE 修改为 EPIE 32.5.12 CAN_AFWL 修改为 CAN_LIMIT 32 全章节 TRG_TRIG 修改成 TT_WTRIG 32.5.12 CAN_AFWL 修改成 CAN_LIMITt 18) 37 章增加 MMC 模式使能寄存器描述; 19) 33.6.2 HFI 修改成 HFIR 增加 33.7.2.1 USBFS VBUS 控制寄存器 33.7.2.5 b31 WKUPINT 修改为 WKUINT, b7 GOUTNAKEFF 改为 GONAKEFF 20) 33.7.2.6 b31 WKUPINTM 修改为 WKUIM b30 VBUSVM 修改成 VBUSWIM b29 DISCM 修改成 DISCIM b28 CIDSCHG 修改成 CIDSCHGM b7 GOUTNAKEFFM 改为 GONAKEFFM 33.7.3.4 PTXQTOP 修改为 PTXQSAV 33.5.4.4 15ms 后修改为 15ms 内 33.7.4.8 INEPxkTXFEM 修改为 INEPTXFEM 33.7.4.2 删除表格中 TCTL 位 33.7.4.11 TO 修改成 TOC ITTXFE 修改成 TTXFE 21) 38.4.1 更新 JTAG/SWJ 调试端口引脚
Rev1.3	2021/12/10	1) 删除产品特性、引脚配置、封装信息等 (相关信息请参考最新的数据手册), 修改声明 2) 更新部分名称和优化描述, 更新笔误 3) 补充 UART 通讯的精度描述 4) timer 6 增加间隔周期有效功能的初始化方法描述 5) 增加 CAN 使用注意事项和采样点的说明 6) 18.3.4 该组端口的数字滤波功能在单元 1 间实现->该组端口的数字滤波功能由单元 1 的 FCONR 设定, 其他单元的 FCONR 对该组端口的数字滤波功能设定无效, 因此任意单元使用该功能时, 都需要将功能控制器 (PWR_FCG2) 中的 TIMER6_1 位置位 7) 19.5.7 TMR4_OCERn b13,b12 描述修改



如果您在购买与使用过程中有任何意见或建议，请随时与我们联系。

Email : mcu@hdsc.com.cn

网址 : <http://www.hdsc.com.cn/mcu.htm>

通信地址：上海市浦东新区中科路 1867 号 A 座 10 层

邮编：201203

