HDSC 华大半导体
HUADA SEMICONDUCTOR

HC32F4A0 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。

©2020 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） 63
1　存储器映射（Memory Mapping） 64
1.1　存储器映射 64
1.2　外部空间映射 72
1.3　位段空间 72
1.4　地址重映射 72
1.5　重映射寄存器 74
1.5.1　访问保护寄存器（MMF_REMPRT） 74
1.5.2　重映射寄存器 0（MMF_REMCR0） 75
1.5.3　重映射寄存器 1（MMF_REMCR1） 76
2　总线架构（BUS） 77
2.1　概述 77
2.2　总线架构 79
2.3　总线功能 81
3　复位控制（RMU） 82
3.1　简介 82
3.2　复位方式和复位标志位 83
3.3　复位时序 86
3.3.1　上电复位 86
3.3.2　NRST 引脚复位 87
3.3.3　欠压复位 87
3.3.4　可编程电压检测 1 复位、可编程电压检测 2 复位 88
3.3.5　看门狗复位、专用看门狗复位 89
3.3.6　掉电唤醒复位 90
3.3.7　软件复位 90
3.3.8　MPU 错误复位 91
3.3.9　RAM 奇偶校验复位 91
3.3.10　RAMECC 复位 91
3.3.11　时钟频率异常复位 92
3.3.12　外部高速振荡器异常停振复位 92
3.3.13　Cortex-M4 Lockup 复位 93
3.3.14　复位方式的判断 93
3.4　寄存器说明 94
3.4.1　复位控制寄存器（RMU_PRSTCR0） 94
3.4.2　复位标志寄存器 0（RMU_RSTF0） 95
4　时钟控制器（CMU） 97
4.1　简介 97
4.2　系统框图 98

4.2.1 系统框图 98
4.2.2 时钟频率测量框图 99
4.3 时钟源规格 100
4.4 工作时钟规格 102
4.5 晶振电路 105
4.5.1 外部高速振荡器 105
4.5.1.1 振荡器模式 105
4.5.1.2 时钟输入模式 106
4.5.2 外部高速振荡器故障检测 107
4.5.2.1 检测到 XTAL 故障检测动作 108
4.5.2.2 检测到 XTAL 振荡故障产生中断复位 109
4.5.3 外部低速振荡器 110
4.6 内部 RC 时钟 111
4.6.1 HRC 时钟 111
4.6.2 MRC 时钟 111
4.6.3 LRC 时钟 112
4.6.4 SWDTLRC 时钟 112
4.6.5 RTCLRC 时钟 112
4.7 PLL 时钟 113
4.8 时钟切换步骤 113
4.8.1 时钟源切换 114
4.8.2 时钟分频切换 115
4.9 时钟输出功能 116
4.10 时钟频率测量 117
4.10.1 时钟频率测量 117
4.10.2 数字滤波功能 118
4.10.3 中断/复位功能 118
4.11 寄存器说明 119
4.11.1 CMU XTAL 配置寄存器器(CMU_XTALCFGR) 121
4.11.2 CMU XTAL 配置寄存器器(CMU_XTALSTBCR) 122
4.11.3 CMU XTAL 控制寄存器(CMU_XTALCR) 123
4.11.4 CMU XTAL 振荡故障控制寄存器(CMU_XTALSTDCR) 124
4.11.5 CMU XTAL 振荡故障状态寄存器(CMU_XTALSTDSR) 125
4.11.6 CMU XTAL32 配置寄存器器(CMU_XTAL32CFGR) 125
4.11.7 CMU XTAL32 滤波寄存器(CMU_XTAL32NFR) 126
4.11.8 CMU XTAL32 控制寄存器(CMU_XTAL32CR) 126
4.11.9 CMU HRC 校准寄存器(CMU_HRCTRM) 127
4.11.10 CMU HRC 控制寄存器(CMU_HRCCR) 128
4.11.11 CMU MRC 校准寄存器(CMU_MRCTRM) 129
4.11.12 CMU MRC 控制寄存器(CMU_MRCCR) 130
4.11.13 CMU LRC 校准寄存器(CMU_LRCTRM) 131
4.11.14 CMU LRC 控制寄存器(CMU_LRCCR) 132

4.11.15 CMU RTCLRC 校准寄存器(CMU_RTCLRCTRM) 133
4.11.16 CMU PLLH 配置寄存器(CMU_PLLHCFGR) 134
4.11.17 CMU PLLH 控制存器(CMU_PLLHCR) 136
4.11.18 CMU PLLA 配置寄存器(CMU_PLLACFGR) 137
4.11.19 CMU PLLA 控制存器(CMU_PLLACR) 139
4.11.20 CMU 时钟源稳定状态器(CMU_OSCSTBSR) 140
4.11.21 CMU 系统时钟源切换寄存器(CMU_CKSWR) 141
4.11.22 CMU 时钟分频配置寄存器(CMU_SCFGR) 142
4.11.23 CMU USB 时钟配置存器(CMU_USBCKCFGR) 145
4.11.24 CMU CAN 时钟配置存器(CMU_CANCKCFGR) 146
4.11.25 CMU AD/TRNG/DA 时钟配置存器(CMU_PERICKSEL) 148
4.11.26 CMU 调试时钟配置存器(CMU_TPIUCKCFGR) 148
4.11.27 CMU MCO1 配置存器(CMU_MCO1CFGR) 149
4.11.28 CMU MCO2 配置存器(CMU_MCO2CFGR) 150
4.11.29 FCM 下限比较值寄存器(FCM_LVR) 151
4.11.30 FCM 上限比较值寄存器(FCM_UVR) 151
4.11.31 FCM 计数器值寄存器(FCM_CNTR) 152
4.11.32 FCM 开始停止寄存器(FCM_STR) 152
4.11.33 FCM 测量对象控制寄存器(FCM_MCCR) 153
4.11.34 FCM 测量基准控制寄存器(FCM_RCCR) 154
4.11.35 FCM 中断复位控制寄存器(FCM_RIER) 156
4.11.36 FCM 标志寄存器(FCM_SR) 157
4.11.37 FCM 标志位清除寄存器(FCM_CLR) 157
5 电源控制（PWC） 158
5.1 简介 158
5.2 电源分布 159
5.2.1 电池备份电源域 160
5.3 电源电压检测单元(PVD)说明 162
5.3.1 上电复位/掉电复位动作说明 162
5.3.2 欠压复位(BOR)说明 163
5.3.3 可编程电压检测 1(PVD1)、可编程电压检测 2(PVD2) 163
5.3.4 PVD1、PVD2 中断/复位框图 165
5.3.5 输入/输出引脚 165
5.3.6 PVD1 中断和复位 166
5.3.7 PVD2 中断和复位 168
5.3.8 内部电压采样和检测功能 169
5.4 唤醒定时器 171
5.5 动作模式与低功耗模式 172
5.5.1 运行模式 176
5.5.2 睡眠模式 177
5.5.3 停止模式 177
5.5.4 掉电模式 179

5.5.5 降低系统时钟速度 183
5.5.6 关闭不使用的时钟源 183
5.5.7 功能时钟停止 183
5.5.8 关闭不使用的 RAM 184
5.6 寄存器保护功能 186
5.7 寄存器说明 187
5.7.1 电源模式控制寄存器 0（PWR_PWRC0） 193
5.7.2 电源模式控制寄存器 1（PWR_PWRC1） 194
5.7.3 电源模式控制寄存器 2（PWR_PWRC2） 195
5.7.4 电源模式控制寄存器 3（PWR_PWRC3） 195
5.7.5 电源模式控制寄存器 4（PWR_PWRC4） 196
5.7.6 掉电唤醒使能寄存器 0(PWR_PDWKE0) 197
5.7.7 掉电唤醒使能寄存器 1(PWR_PDWKE1) 198
5.7.8 掉电唤醒使能寄存器 2(PWR_PDWKE2) 199
5.7.9 掉电唤醒事件边沿选择寄存器(PWR_PDWKES) 200
5.7.10 掉电唤醒标志寄存器 0(PWR_PDWKF0) 201
5.7.11 掉电唤醒标志寄存器 1(PWR_PDWKF1) 202
5.7.12 功能时钟控制 0(PWR_FCG0) 203
5.7.13 功能时钟控制 1(PWR_FCG1) 205
5.7.14 功能时钟控制 2(PWR_FCG2) 207
5.7.15 功能时钟控制 3(PWR_FCG3) 209
5.7.16 PWR_FCG0 保护控制(PWR_FCG0PC) 211
5.7.17 功能保护控制寄存器(PWR_FPRC) 212
5.7.18 STOP 模式控制寄存器(PWR_STPMCR) 213
5.7.19 RAM 功耗控制寄存器 0(PWR_RAMPC0) 214
5.7.20 RAM 运行条件寄存器(PWR_RAMOPM) 215
5.7.21 外设 RAM 低功耗控制寄存器 (PWR_PRAMLPC) 216
5.7.22 PVD 控制寄存器 0(PWR_PVDCR0) 218
5.7.23 PVD 控制寄存器 1(PWR_PVDCR1) 219
5.7.24 PVD 滤波控制寄存器(PWR_PVDFCR) 220
5.7.25 PVD 电平控制寄存器(PWR_PVDLCR) 221
5.7.26 PVD 中断控制寄存器(PWR_PVDICR) 222
5.7.27 PVD 检测状态寄存器(PWR_PVDDSR) 223
5.7.28 备份域复位寄存器(PWR_VBATRSTR) 223
5.7.29 备份域控制寄存器(PWR_VBATCR) 224
5.7.30 VBAT 备份寄存器 0~127(PWR_BKR000~PWR_BKR127) 224
5.7.31 唤醒计时器控制寄存器 0 (PWR_WKTC0) 225
5.7.32 唤醒计时器控制寄存器 1 (PWR_WKTC1) 225
5.7.33 唤醒计时器控制寄存器 2 (PWR_WKTC2) 226
6 初始化配置（ICG） 227
6.1 简介 227
6.2 寄存器说明 228

6.2.1 初始化配置寄存器 0(ICG0) 228
6.2.2 初始化配置寄存器 1(ICG1) 231
6.2.3 初始化配置寄存器 2(ICG2) 232
6.2.4 初始化配置寄存器 3(ICG3) 232
7 嵌入式 FLASH（EFM） 233
7.1 简介 233
7.2 主要特性 233
7.3 嵌入式 FLASH 234
7.4 读接口 235
7.4.1 CPU 时钟和 FLASH 读取时间之间的关系 235
7.5 FLASH 读加速缓存 236
7.6 FLASH 编程和擦除操作 237
7.6.1 解锁 EFM_KEY1 寄存器 237
7.6.2 写保护功能 237
7.6.3 单次编程无回读功能 238
7.6.4 单编程回读功能 238
7.6.5 连续编程功能 239
7.6.6 擦除功能 239
7.6.7 全擦除功能 240
7.6.8 D-BUS 读保护功能 240
7.6.9 BGO 功能 241
7.6.10 中断 241
7.7 一次性可编程字节（OTP） 243
7.8 引导交换 245
7.9 寄存器说明 247
7.9.1 访问写保护寄存器(EFM_FAPRT) 248
7.9.2 FLASH 密钥 1 寄存器(EFM_KEY1) 249
7.9.3 FLASH 密钥 2 寄存器(EFM_KEY2) 249
7.9.4 FLASH 停止寄存器(EFM_FSTP) 250
7.9.5 读模式寄存器(EFM_FRMC) 251
7.9.6 擦写模式寄存器(EFM_FWMC) 252
7.9.7 状态寄存器(EFM_FSR) 253
7.9.8 状态清除寄存器(EFM_FSCLR) 255
7.9.9 中断许可寄存器(EFM_FITE) 257
7.9.10 引导交换状态寄存器(EFM_FSWP) 258
7.9.11 华大芯片专属标志寄存器(EFM_CHIPID) 259
7.9.12 UNIQUE ID 寄存器(EFM_UQID0) 259
7.9.13 UNIQUE ID 寄存器(EFM_UQID1) 260
7.9.14 UNIQUE ID 寄存器(EFM_UQID2) 260
7.9.15 FLASH 写保护锁定寄存器(EFM_WLOCK) 261
7.9.16 FLASH0 写保护寄存器 0(EFM_F0NWPRT0) 262
7.9.17 FLASH0 写保护寄存器 1(EFM_F0NWPRT1) 263

7.9.18 FLASH0 写保护寄存器 2(EFM_F0NWPRT2) 264
7.9.19 FLASH0 写保护寄存器 3(EFM_F0NWPRT3) 265
7.9.20 FLASH1 写保护寄存器 0(EFM_F1NWPRT0) 266
7.9.21 FLASH1 写保护寄存器 1(EFM_F1NWPRT1) 267
7.9.22 FLASH1 写保护寄存器 2(EFM_F1NWPRT2) 268
7.9.23 FLASH1 写保护寄存器 3(EFM_F1NWPRT3) 269
7.10 注意事项 270
8 内置 SRAM（SRAM） 271
8.1 简介 271
8.2 寄存器说明 273
8.2.1 SRAM 等待控制寄存器（SRAM_WTCR） 274
8.2.2 SRAM 等待保护寄存器（SRAM_WTPR） 276
8.2.3 SRAM 校验控制寄存器（SRAM_CKCR） 277
8.2.4 SRAM 校验保护寄存器（SRAM_CKPR） 279
8.2.5 SRAM 校验状态寄存器（SRAM_CKSR） 280
9 通用 IO（GPIO） 281
9.1 简介 281
9.2 端口功能概要 282
9.3 动作说明 283
9.3.1 通用输入输出 GPIO 功能 283
9.3.2 周边功能 284
9.3.3 双周边功能 284
9.3.4 Event Port 输入输出功能 284
9.3.5 外部中断 EIRQ 输入功能 285
9.3.6 模拟功能 285
9.3.7 通用控制 285
9.4 寄存器说明 287
9.4.1 通用输入寄存器（PIDRx） 289
9.4.2 通用输出数据寄存器（PODRx） 289
9.4.3 通用输出许可寄存器（POERx） 290
9.4.4 通用输出置位寄存器（POSRx） 290
9.4.5 通用输出复位寄存器（PORRx） 291
9.4.6 通用输出翻转寄存器（POTRx） 291
9.4.7 特殊控制寄存器（PSPCR） 292
9.4.8 公共控制寄存器（PCCR） 293
9.4.9 输入控制寄存器（PINAER） 294
9.4.10 写保护寄存器（PWPR） 294
9.4.11 通用控制寄存器（PCRxy） 295
9.4.12 功能选择寄存器（PFSRxy） 297
9.4.13 Event Port 触发源选择寄存器（PEVNTTRGSR12, PEVNTTRGSR34） 298
9.4.14 Event Port 方向选择寄存器（PEVNTDIRRm） 298
9.4.15 Event Port 输入数据寄存器（PEVNTIDRm） 299

9.4.16 Event Port 输出数据寄存器（PEVNTODRm） 299
9.4.17 Event Port 输出数据复位寄存器（PEVNTORRm） 300
9.4.18 Event Port 输出数据置位寄存器（PEVNTOSRm） 300
9.4.19 Event Port 上升沿输入许可寄存器（PEVNTRISRm） 301
9.4.20 Event Port 下降沿输入许可寄存器（PEVNTFALRm） 302
9.4.21 Event Port 输入滤波控制寄存器（PEVNTNFCR） 303
9.4.22 32bit 访问 305
9.5 注意事项 306
10 中断控制器（INTC） 307
10.1 简介 307
10.2 INTC 系统框图 309
10.2.1 系统框图 309
10.3 向量表 310
10.3.1 中断向量表 310
10.3.2 中断事件请求序号 316
10.4 功能说明 335
10.4.1 不可屏蔽中断 335
10.4.2 外部管脚中断事件请求 335
10.4.3 中断源选择 336
10.4.4 软件中断 336
10.4.5 中断/事件选择 336
10.4.6 WFE 唤醒事件管理 337
10.4.7 数字滤波器 338
10.4.8 低功耗模式返回 339
10.4.8.1 休眠模式返回 339
10.4.8.2 停止模式返回 339
10.4.8.3 掉电模式返回 340
10.4.8.4 不可屏蔽中断和 WFI 指令 340
10.4.9 内部触发事件 340
10.5 寄存器说明 341
10.5.1 外部管脚中断滤波控制寄存器（INT_NOCCR） 347
10.5.2 不可屏蔽中断使能寄存器（INT_NMIENR） 348
10.5.3 不可屏蔽中断标志寄存器（INT_NMIFR） 349
10.5.4 不可屏蔽中断标志清除寄存器（INT_NMICFR） 350
10.5.5 外部管脚中断控制寄存器(INT_EIRQCRx)(x=0~15) 351
10.5.6 外部管脚中断标志寄存器(INT_EIFR) 352
10.5.7 外部管脚中断标志清除寄存器(INT_EIFCR) 352
10.5.8 中断/事件选择寄存器（INT_SEL0~31） 353
10.5.9 中断选择寄存器（INT_SEL32~127） 354
10.5.10 向量共享中断选择寄存器（INT_VSSEL128~143） 355
10.5.11 停止模式唤醒事件使能寄存器（INT_WUPEN） 356
10.5.12 软件中断/事件寄存器（INT_SWIER） 358

10.5.13 事件使能寄存器（INT_EVTER）..... 358
10.5.14 中断使能寄存器（INT_IER)..... 359
10.5.15 内部触发事件寄存器（INTSFTTRG）..... 359
10.6 使用注意事项..... 360
11 自动运行系统（AOS）..... 361
11.1 简介..... 361
11.1.1 功能概览..... 361
11.1.2 模块示意图..... 362
11.2 功能描述..... 363
11.2.1 AOS 源事件列表..... 363
11.2.2 AOS 目标列表..... 363
11.3 动作说明..... 365
11.3.1 专用触发源..... 365
11.3.2 公共触发源..... 365
11.4 寄存器说明..... 366
11.4.1 寄存器一览..... 366
11.4.2 外设触发事件寄存器(INTSFTTRG)..... 367
11.4.3 DCU 触发源选择寄存器(DCU_TRGSELx)(x=1~4)..... 368
11.4.4 DMA1 传输启动触发源选择寄存器(DMA1_TRGSELx)(x=0~7)..... 369
11.4.5 DMA2 传输启动触发源选择寄存器(DMA2_TRGSELx)(x=0~7)..... 370
11.4.6 DMA 通道重置触发源选择寄存器(DMA_TRGSELRC)..... 371
11.4.7 Timer6 硬件触发事件选择寄存器(TMR6_HTSSRx)(x=0~3)..... 372
11.4.8 Event Port 触发源选择寄存器(PEVNTTRGSR12, PEVNTTRGSR34)..... 373
11.4.9 Timer0 硬件触发事件选择寄存器(TMR0_HTSSR)..... 374
11.4.10 Timer2 硬件触发事件选择寄存器(TMR2_HTSSR)..... 375
11.4.11 HASH 硬件事件触发选择寄存器 A（HASH_ITRGSELA）..... 376
11.4.12 HASH 硬件事件触发选择寄存器 B（HASH_ITRGSELB）..... 377
11.4.13 TimerA 硬件触发事件选择寄存器(TMRA_HTSSRx)(x=0~3)..... 378
11.4.14 OTS 触发源选择寄存器(OTS_TRG)..... 379
11.4.15 A/D1 转换开始片内触发源选择寄存器 ADC1_ITRGSELRx(x=0,1)..... 380
11.4.16 A/D2 转换开始片内触发源选择寄存器 ADC2_ITRGSELRx(x=0,1)..... 381
11.4.17 A/D3 转换开始片内触发源选择寄存器 ADC3_ITRGSELRx(x=0,1)..... 382
11.4.18 公共触发源选择寄存器 1(AOS_COMTRG1)..... 383
11.4.19 公共触发源选择寄存器 2(AOS_COMTRG2)..... 383
12 存储保护单元（MPU）..... 384
12.1 简介..... 384
12.2 功能描述..... 385
12.2.1 区域范围设置..... 385
12.2.2 权限设置..... 385
12.2.3 MPU 动作选择..... 385
12.2.4 启动 MPU..... 385
12.3 应用举例..... 386

12.3.1 只允许部分空间访问 386
12.3.2 只禁止部分空间访问 386
12.4 寄存器说明 387
12.4.1 区域范围描述寄存器 MPU_RGDn（n=0～15） 389
12.4.2 状态标志寄存器 MPU_SR 390
12.4.3 标志清除寄存器 MPU_ECLR 391
12.4.4 写保护寄存器 MPU_WP 392
12.4.5 SMPU1 区域使能寄存器 MPU_S1RGE 393
12.4.6 SMPU1 区域写权限寄存器 MPU_S1RGWP 395
12.4.7 SMPU1 区域读权限寄存器 MPU_S1RGRP 397
12.4.8 SMPU1 控制寄存器 MPU_S1CR 399
12.4.9 SMPU2 区域使能寄存器 MPU_S2RGE 400
12.4.10 SMPU2 区域写权限寄存器 MPU_S2RGWP 402
12.4.11 SMPU2 区域读权限寄存器 MPU_S2RGRP 404
12.4.12 SMPU2 控制寄存器 MPU_S2CR 406
12.4.13 FMPU 区域使能寄存器 MPU_FRGE 407
12.4.14 FMPU 区域写权限寄存器 MPU_FRGWP 408
12.4.15 FMPU 区域读权限寄存器 MPU_FRGRP 409
12.4.16 FMPU 控制寄存器 MPU_FCR 410
12.4.17 HMPU 区域使能寄存器 MPU_HRGE 411
12.4.18 HMPU 区域写权限寄存器 MPU_HRGWP 412
12.4.19 HMPU 区域读权限寄存器 MPU_HRGRP 413
12.4.20 HMPU 控制寄存器 MPU_HCR 414
12.4.21 EMPU 区域使能寄存器 MPU_ERGE 415
12.4.22 EMPU 区域写权限寄存器 MPU_ERGWP 416
12.4.23 EMPU 区域读权限寄存器 MPU_ERGRP 417
12.4.24 EMPU 控制寄存器 MPU_ECR 418
12.4.25 IP 访问保护寄存器 MPU_IPPR 419
13 键盘扫描控制模块（KEYSCAN） 422
13.1 简介 422
13.2 KEYSCAN 系统框图 423
13.3 管脚说明 423
13.4 功能说明 424
13.4.1 按键识别功能 424
13.4.2 键盘扫描功能 424
13.4.3 使用上的注意事项 425
13.5 寄存器说明 425
13.5.1 KEYSCAN 扫描控制寄存器（KEYSCAN_SCR） 426
13.5.2 KEYSCAN 扫描使能寄存器（KEYCAN_SER） 428
13.5.3 KEYSCAN 扫描状态寄存器（KEYSCAN_SSR） 429
14 内部时钟校准器（CTC） 430
14.1 简介 430

14.2 结构框图.....430
14.3 功能说明.....431
14.3.1 参考时钟.....431
14.3.2 频率校准.....433
14.3.3 编程指南.....434
14.4 寄存器说明.....436
14.4.1 时钟校准控制寄存器 1（CTC_CR1）.....437
14.4.2 时钟校准控制寄存器 2（CTC_CR2）.....439
14.4.3 时钟校准状态寄存器（CTC_STR）.....440
15 DMA 控制器（DMA）.....441
15.1 简介.....441
15.2 模块示意图.....442
15.3 功能描述.....443
15.3.1 使能 DMA 控制器.....443
15.3.2 通道选择和通道优先级.....443
15.3.3 启动 DMA.....443
15.3.4 数据块.....443
15.3.5 传输地址控制.....444
15.3.6 传输次数.....444
15.3.7 中断和事件信号输出.....444
15.3.8 连锁传输.....445
15.3.9 不连续地址传输.....447
15.3.10 通道重置.....448
15.3.11 传输提前终止.....449
15.4 应用举例.....451
15.4.1 存储器到存储器的传输.....451
15.4.2 存储器到外围电路的传输.....452
15.4.3 存储器到存储器的连锁传输.....454
15.5 寄存器说明.....457
15.5.1 寄存器一览.....457
15.5.2 DMA 使能寄存器(DMA_EN).....459
15.5.3 中断状态寄存器 0(DMA_INTSTAT0).....459
15.5.4 中断状态寄存器 0(DMA_INTSTAT1).....460
15.5.5 中断屏蔽寄存器(DMA_INTMASK0).....460
15.5.6 中断屏蔽寄存器(DMA_INTMASK1).....461
15.5.7 中断复位寄存器(DMA_INTCLR0).....461
15.5.8 中断复位寄存器(DMA_INTCLR1).....462
15.5.9 通道使能寄存器(DMA_CHEN).....462
15.5.10 通道使能复位寄存器(DMA_CHENCLR).....463
15.5.11 通道重置控制寄存器(DMA_RCFGCTL).....464
15.5.12 传输请求状态寄存器(DMA_REQSTAT).....466
15.5.13 通道状态观测寄存器(DMA_CHSTAT).....467

15.5.14 传输启动触发源选择寄存器(DMA_TRGSELx) (x=0~7) 467
15.5.15 通道重置触发源选择寄存器(DMA_TRGSELRC) 468
15.5.16 传输源地址寄存器(DMA_SARx) (x=0~7) 468
15.5.17 传输目标地址寄存器(DMA_DARx) (x=0~7) 469
15.5.18 数据控制寄存器(DMA_DTCTLx) (x=0~7) 469
15.5.19 重复区域大小寄存器(DMA_RPTx) (x=0~7) 470
15.5.20 重复区域大小寄存器 B(DMA_RPTBx) (x=0~7) 471
15.5.21 源设备不连续地址传输控制寄存器(DMA_SNSEQCTLx) (x=0~7) 472
15.5.22 源设备不连续地址传输控制寄存器 B(DMA_SNSEQCTLBx) (x=0~7) 473
15.5.23 目标设备不连续地址传输控制寄存器(DMA_DNSEQCTLx) (x=0~7) 474
15.5.24 目标设备不连续地址传输控制寄存器 B(DMA_DNSEQCTLBx) (x=0~7) 475
15.5.25 链指针寄存器(DMA_LLPx) (x=0~7) 476
15.5.26 通道控制寄存器(DMA_CHCTLx) (x=0~7) 477
15.5.27 通道监视寄存器(DMA_MONSARx， DMA_MONDARx, DMA_MONDTCTLx, DMA_MONRPTx, DMA_MONSNSEQCTLx, DMA_MONDNSEQCTLx) (x=0~7) 478
15.6 使用注意事项 479
16 电压比较器（CMP） 480
16.1 简介 480
16.2 功能框图 481
16.3 功能说明 483
16.3.1 普通比较模式 483
16.3.2 窗口比较模式 484
16.3.3 定时器窗口输出 485
16.3.4 数字滤波器 485
16.3.5 比较器中断 486
16.3.6 外设触发事件 486
16.3.7 外部管脚输出 486
16.4 注意事项 487
16.4.1 模块停止功能 487
16.4.2 模块停止时的动作 487
16.4.3 停止低功耗模式时的动作 487
16.5 寄存器说明 488
16.5.1 比较器模式设定寄存器（CMPx_MDR，x=1~4） 489
16.5.2 比较器滤波和中断控制寄存器（CMPx_FIR，x=1~4） 490
16.5.3 比较器输出控制寄存器（CMPx_OCR，x=1~4） 491
16.5.4 比较器正负端输入选择寄存器（CMPx_PMSR，x=1~4） 492
16.5.5 比较器电压输入源选择寄存器（CMPx_VISR，x=1，3） 493
16.5.6 比较器定时器窗口选择寄存器（CMPx_TWSR，x=1~4） 495
16.5.7 比较器定时器窗口极性寄存器（CMPx_TWPR，x=1~4） 496
17 模数转换模块（ADC） 497
17.1 简介 497
17.2 ADC 系统框图 499

17.3 功能说明 501
17.3.1 ADC 时钟 501
17.3.2 通道选择 501
17.3.3 触发源选择 503
17.3.4 序列 A 单次扫描模式 503
17.3.5 序列 A 连续扫描模式 504
17.3.6 双序列扫描模式 505
17.3.7 模拟看门狗功能 507
17.3.8 模拟输入的采样时间和转换时间 508
17.3.9 A/D 数据寄存器自动清除功能 510
17.3.10 转换数据平均计算功能 510
17.3.11 可编程增益放大器 PGA 511
17.3.12 通道专用采样保持电路 SH 513
17.3.13 多 ADC 协同工作模式 514
17.3.14 中断和事件信号输出 519
17.4 寄存器说明 520
17.4.1 寄存器一览 520
17.4.2 A/D 启动寄存器 ADC_STR 522
17.4.3 A/D 控制寄存器 0 ADC_CR0 523
17.4.4 A/D 控制寄存器 1 ADC_CR1 524
17.4.5 A/D 转换开始触发寄存器 ADC_TRGSR 525
17.4.6 A/D 转换开始片内触发源选择寄存器 ADC_ITRGSELR0，ADC_ITRGSELR1 526
17.4.7 A/D 通道选择寄存器 A ADC_CHSELRA 527
17.4.8 A/D 通道选择寄存器 B ADC_CHSELRB 528
17.4.9 A/D 平均通道选择寄存器 ADC_AVCHSELR 529
17.4.10 A/D 扩展通道选择寄存器 ADC_EXCHSELR 529
17.4.11 A/D 采样状态寄存器 ADC_SSTRx 530
17.4.12 A/D 通道映射控制寄存器 ADC_CHMUXR 531
17.4.13 A/D 中断状态寄存器 ADC_ISR 532
17.4.14 A/D 中断许可寄存器 ADC_ICR 532
17.4.15 A/D 中断状态复位寄存器 ADC_ISCLRR 533
17.4.16 A/D 协同模式控制寄存器 ADC_SYNCCR 534
17.4.17 A/D 数据寄存器 ADC_DR 536
17.4.18 模拟看门狗控制寄存器 ADC_AWDCR 537
17.4.19 模拟看门狗状态寄存器 ADC_AWDSR 538
17.4.20 模拟看门狗状态复位寄存器 ADC_AWDSCLRR 538
17.4.21 模拟看门狗阀值寄存器 ADC_AWD0DR0、ADC_AWD0DR1、ADC_AWD1DR0、ADC_AWD1DR1 539
17.4.22 模拟看门狗比较通道选择寄存器 ADC_AWD0CHSR、ADC_AWD1CHSR 540
17.4.23 专用采样保持电路控制寄存器 ADC_SHCR 541
17.4.24 可编程增益放大器控制寄存器 ADC_PGACRx 542
17.4.25 可编程增益放大器参考地选择寄存器 ADC_PGAVSSENR 543

17.5 使用注意事项.....544
17.5.1 数据寄存器读取时注意事项.....544
17.5.2 扫描完成中断处理注意事项.....544
17.5.3 模块停止与低功耗设定的注意事项.....544
17.5.4 A/D 转换模拟通道输入的引脚设定.....544
17.5.5 噪声控制.....544
18 数模转换器（DAC）.....545
18.1 简介.....545
18.2 功能框图.....546
18.3 功能说明.....547
18.3.1 D/A 转换.....547
18.3.2 同步转换.....547
18.3.3 外部数据转换.....548
18.3.4 A/D 转换优先模式.....548
18.4 注意事项.....550
18.4.1 模块停止功能的设置.....550
18.4.2 模块停止状态下的 DAC 动作.....550
18.4.3 停止低功耗模式下的 DAC 动作.....550
18.4.4 掉电低功耗模式下的 DAC 动作.....550
18.4.5 用输出放大器的注意事项.....550
18.5 寄存器说明.....551
18.5.1 DAC 数据寄存器（DADRx_y x,y=1,2）.....552
18.5.2 DAC 控制寄存器（DACRx, x=1,2）.....553
18.5.3 DAC 模拟输出控制寄存器（DAOCRx, x=1,2）.....554
18.5.4 DAC A/D 转换优先控制寄存器（DAADPCRx, x=1,2）.....555
19 温度传感器（OTS）.....556
19.1 简介.....556
19.2 使用说明.....557
19.3 寄存器说明.....559
19.3.1 OTS 控制寄存器（OTS_CTL）.....560
19.3.2 OTS 数据寄存器 1（OTS_DR1）.....560
19.3.3 OTS 数据寄存器 2（OTS_DR2）.....561
19.3.4 OTS 误差补偿寄存器（OTS_ECR）.....561
19.3.5 OTS 预置温度数据寄存器（OTS_PDR1,2,3）.....561
19.3.6 OTS 触发源选择寄存器（OTS_TRG）.....562
20 高级控制定时器（Timer6）.....563
20.1 简介.....563
20.2 基本框图.....564
20.3 功能说明.....566
20.3.1 波形模式.....566
20.3.2 时钟源选择.....566
20.3.3 计数方向.....567

20.3.3.1 锯齿波计数方向 567
20.3.3.2 三角波计数方向 567
20.3.4 比较输出 567
20.3.5 捕获输入 568
20.3.6 计数器刷新 569
20.3.7 软件同步 570
20.3.7.1 软件同步启动 570
20.3.7.2 软件同步停止 570
20.3.7.3 软件同步清零 570
20.3.7.4 软件同步刷新 570
20.3.8 硬件同步 571
20.3.8.1 硬件同步启动 571
20.3.8.2 硬件同步停止 572
20.3.8.3 硬件同步清零 572
20.3.8.4 硬件同步刷新 572
20.3.8.5 硬件同步捕获输入 572
20.3.8.6 硬件同步计数 572
20.3.9 脉宽测量 573
20.3.10 周期测量 574
20.3.11 缓存功能 575
20.3.11.1 单缓存动作 575
20.3.11.2 双缓存动作 576
20.3.11.3 缓存传送节点 577
比较输出（锯齿波）缓存传送 577
比较输出（三角波）缓存传送 578
捕获输入缓存传送 580
20.3.12 数字滤波 580
20.3.13 通用 PWM 输出 580
20.3.13.1 单边对齐独立 PWM 输出 580
20.3.13.2 双边对称独立 PWM 输出 581
20.3.13.3 双边对称互补 PWM 输出 582
软件设定 GCMBR 互补 PWM 输出 582
硬件设定 GCMBR 互补 PWM 输出 583
20.3.13.4 双边非对称 PWM 输出 584
20.3.13.5 单元间多相 PWM 输出 585
20.3.14 周期间隔响应 588
20.3.15 正交编码计数 589
20.3.15.1 位置计数模式 589
基本计数 589
相位差计数 590
方向计数 591
20.3.15.2 公转计数模式 591

Z 相计数 591
位置溢出计数 592
混合计数 592
20.3.15.3 Z 相动作屏蔽 593
20.3.16 EMB 控制 594
20.3.17 典型应用例 595
20.3.17.1 基本计数及中断动作 595
20.3.17.2 比较输出及中断动作 595
20.3.17.3 捕获输入及中断动作 596
20.3.17.4 缓存传送动作（周期基准值） 596
20.3.17.5 缓存传送动作（通用比较基准值） 597
20.3.17.6 缓存传送动作（专用比较基准值） 597
20.3.17.7 缓存传送动作（死区基准值） 598
20.3.17.8 同步启动动作（软件方式） 598
20.3.17.9 同步启动动作（硬件方式） 598
20.3.17.10 正交编码计数动作（2 相） 599
20.3.17.11 正交编码计数动作（3 相） 599
20.3.17.12 单路 PWM 输出 600
20.3.17.13 互补 PWM 输出（软件死区） 600
20.3.17.14 互补 PWM 输出（硬件死区） 601
20.3.17.15 EMB 监控及中断动作 601
20.3.18 功能汇总表 602
20.4 中断及事件说明 604
20.4.1 中断输出 604
20.4.1.1 计数比较匹配中断 604
20.4.1.2 计数周期匹配中断 604
20.4.1.3 死区时间错误中断 604
20.4.2 事件输出 605
20.5 寄存器说明 606
20.5.1 通用计数值寄存器（TMR6_CNTER） 609
20.5.2 通用刷新值寄存器（TMR6_UPDAR） 609
20.5.3 通用周期基准值寄存器（TMR6_PERmR）（m=A~C） 610
20.5.4 通用比较基准值寄存器（TMR6_GCMmR）（m=A~F） 610
20.5.5 专用比较基准值寄存器（TMR6_SCMmR）（m=A~F） 611
20.5.6 死区时间基准值寄存器（TMR6_DTmnR）（m=D、U&&n=A、B） 611
20.5.7 通用控制寄存器（TMR6_GCONR） 612
20.5.8 中断控制寄存器（TMR6_ICONR） 614
20.5.9 缓存控制寄存器（TMR6_BCONR） 616
20.5.10 死区控制寄存器（TMR6_DCONR） 618
20.5.11 端口控制寄存器（TMR6_PCNAR） 619
20.5.12 端口控制寄存器（TMR6_PCNBR） 622
20.5.13 滤波控制寄存器（TMR6_FCNGR） 625

20.5.14 滤波控制寄存器（TMR6_FCNTR）..... 626
20.5.15 有效周期寄存器（TMR6_VPERR）..... 627
20.5.16 状态标志寄存器（TMR6_STFLR）..... 628
20.5.17 硬件启动事件选择寄存器（TMR6_HSTAR）..... 630
20.5.18 硬件停止事件选择寄存器（TMR6_HSTPR）..... 632
20.5.19 硬件清零事件选择寄存器（TMR6_HCLRR）..... 634
20.5.20 硬件刷新事件选择寄存器（TMR6_HUPDR）..... 636
20.5.21 硬件捕获事件选择寄存器（TMR6_HCPAR）..... 638
20.5.22 硬件捕获事件选择寄存器（TMR6_HCPBR）..... 640
20.5.23 硬件递加事件选择寄存器（TMR6_HCUPR）..... 642
20.5.24 硬件递减事件选择寄存器（TMR6_HCDOR）..... 644
20.5.25 硬件触发事件选择寄存器（TMR6_HTSSRm）（m=0~3）..... 646
20.5.26 软件同步启动控制寄存器（TMR6_SSTAR）..... 647
20.5.27 软件同步停止控制寄存器（TMR6_SSTPR）..... 648
20.5.28 软件同步清零控制寄存器（TMR6_SCLRR）..... 649
20.5.29 软件同步刷新控制寄存器（TMR6_SUPDR）..... 650
20.6 使用注意事项..... 651
21 高精度 PWM（HRPWM）..... 653
21.1 简介..... 653
21.2 基本框图..... 653
21.3 功能说明..... 654
21.3.1 校准功能..... 654
21.3.2 高分辨率 PWM 调整功能..... 655
21.3.3 使用注意事项..... 655
21.4 寄存器说明..... 657
21.4.1 HRPWM 控制寄存器(HRPWM_CRn, n=1…16)..... 658
21.4.2 HRPWM 校准控制寄存器 0(HRPWM_CALCRn, n=0,1)..... 659
22 通用控制定时器（Timer4）..... 660
22.1 简介..... 660
22.2 基本框图..... 660
22.3 功能说明..... 662
22.3.1 波形模式..... 662
22.3.2 计数动作..... 662
22.3.3 比较输出..... 663
22.3.4 缓存功能..... 664
22.3.4.1 周期基准寄存器缓存功能..... 665
22.3.4.2 通用比较寄存器缓存功能..... 666
22.3.4.3 专用比较寄存器缓存功能..... 669
22.3.5 通用 PWM 输出..... 671
22.3.5.1 独立 PWM 输出..... 671
22.3.5.2 扩展 PWM 输出..... 672
22.3.5.3 互补 PWM 输出..... 673

22.3.6 周期间隔响应 676
22.3.7 EMB 控制 678
22.3.8 监测输出 679
22.4 中断及事件说明 681
22.4.1 计数比较匹配中断 681
22.4.2 计数周期匹配中断 681
22.4.3 重载计数匹配中断 681
22.4.4 专用比较匹配事件 681
22.5 寄存器说明 683
22.5.1 计数值寄存器（TMR4_CNTR） 686
22.5.2 周期基准寄存器（TMR4_CPSR） 686
22.5.3 控制状态寄存器（TMR4_CCSR） 687
22.5.4 有效周期寄存器（TMR4_CVPR） 689
22.5.5 通用比较基准寄存器（TMR4_OCCRm） 689
22.5.6 通用控制状态寄存器（TMR4_OCSRn） 690
22.5.7 通用扩展控制寄存器（TMR4_OCERn） 691
22.5.8 通用模式控制寄存器（TMR4_OCMRm） 693
22.5.9 专用比较基准寄存器（TMR4_SCCRm） 698
22.5.10 专用控制状态寄存器（TMR4_SCSRm） 699
22.5.11 专用扩展控制寄存器（TMR4_SCER） 701
22.5.12 专用模式控制寄存器（TMR4_SCMRm） 701
22.5.13 PWM 基本控制寄存器（TMR4_POCRn） 702
22.5.14 PWM 状态控制寄存器（TMR4_PSCR） 703
22.5.15 PWM 滤波控制寄存器（TMR4_PFSRn） 706
22.5.16 PWM 死区控制寄存器（TMR4_PDA<B>Rn） 706
22.5.17 重载控制状态寄存器（TMR4_RCSR） 707
23 紧急刹车模块（EMB） 709
23.1 简介 709
23.2 功能描述 711
23.2.1 概述 711
23.2.2 外部端口输入电平变化时控制 PWM 信号输出 713
23.2.3 PWM 输出端口电平发生同相（同高或同低）时停止 PWM 信号输出 714
23.2.4 根据电压比较器比较结果停止 PWM 信号输出 715
23.2.5 外部振荡器停止振荡时停止 PWM 信号输出 716
23.2.6 写寄存器软件控制 PWM 信号输出 716
23.3 寄存器说明 717
23.3.1 EMB 控制寄存器 1_0~3(EMB_CTL1_0~3) 718
23.3.2 EMB 控制寄存器 1_4~6(EMB_CTL1_4~6) 719
23.3.3 EMB 控制寄存器 2_0~3(EMB_CTL2_0~3) 721
23.3.4 EMB 控制寄存器 2_4~6(EMB_CTL2_4~6) 723
23.3.5 EMB 软件输出使能控制寄存器(EMB_SOEx)(x=0~6) 725
23.3.6 EMB 状态寄存器(EMB_STATx) (x=0~6) 726

23.3.7 EMB 状态复位寄存器(EMB_STATCLRx) (x=0~6) 728
23.3.8 EMB 中断许可寄存器(EMB_INTENx)(x=0~6) 729
23.3.9 EMB 控制 PWM 输出释放方式选择寄存器(EMB_RLSSELx)(x=0~6) 730
24 通用定时器（TimerA） 731
24.1 简介 731
24.2 基本框图 731
24.3 功能说明 733
24.3.1 波形模式 733
24.3.2 时钟源选择 733
24.3.3 比较输出 734
24.3.4 捕获输入 734
24.3.5 同步启动 735
24.3.6 数字滤波 736
24.3.7 缓存功能 737
24.3.8 级联计数 738
24.3.9 PWM 输出 739
24.3.9.1 单边对齐 PWM 输出 739
24.3.9.2 双边对称 PWM 输出 739
24.3.10 正交编码计数 740
24.3.10.1 位置计数模式 741
24.3.10.2 公转计数模式 742
24.4 中断及事件说明 745
24.4.1 比较匹配中断及事件 745
24.4.2 周期匹配中断及事件 745
24.5 寄存器说明 746
24.5.1 通用计数值寄存器（TMRA_CNTER） 749
24.5.2 周期基准值寄存器（TMRA_PERAR） 749
24.5.3 比较基准值寄存器（TMRA_CMPARm）（m=1~4） 749
24.5.4 控制状态寄存器（TMRA_BCSTR） 750
24.5.5 中断控制寄存器（TMRA_ICONR） 752
24.5.6 事件控制寄存器（TMRA_ECONR） 753
24.5.7 滤波控制寄存器（TMRA_FCONR） 754
24.5.8 状态标志寄存器（TMRA_STFLR） 755
24.5.9 缓存控制寄存器（TMRA_BCONRm）（m=1~2） 756
24.5.10 捕获控制寄存器（TMRA_CCONRm）（m=1~4） 757
24.5.11 端口控制寄存器（TMRA_PCONRm）（m=1~4） 759
24.5.12 硬件触发事件选择寄存器（TMRA_HCONR） 761
24.5.13 硬件递加事件选择寄存器（TMRA_HCUPR） 763
24.5.14 硬件递减事件选择寄存器（TMRA_HCDOR） 765
24.5.15 内部触发事件选择寄存器（TMRA_HTSSRm）（m=0~3） 767
25 通用定时器（Timer2） 768
25.1 简介 768

25.2 基本框图.....768
25.3 功能说明.....770
25.3.1 时钟源选择.....770
25.3.1.1 同步计数时钟源.....770
25.3.1.2 异步计数时钟源.....770
25.3.2 比较输出.....771
25.3.3 硬件触发.....771
25.3.3.1 启动停止清零.....772
25.3.3.2 捕获输入动作.....772
25.3.3.3 脉宽测量.....773
25.3.3.4 周期测量.....773
25.3.4 数字滤波.....774
25.4 中断及事件说明.....775
25.4.1 中断输出.....775
25.4.2 事件输出.....775
25.5 寄存器说明.....776
25.5.1 计数值寄存器（TMR2_CNTmR）（m=A、B）.....777
25.5.2 基准值寄存器（TMR2_CMPmR）（m=A、B）.....777
25.5.3 基本控制寄存器（TMR2_BCONR）.....778
25.5.4 中断控制寄存器（TMR2_ICONR）.....781
25.5.5 端口控制寄存器（TMR2_PCONR）.....782
25.5.6 硬件控制寄存器（TMR2_HCONR）.....784
25.5.7 触发选择寄存器（TMR2_HTSSR）.....787
25.5.8 状态标志寄存器（TMR2_STFLR）.....787
25.6 使用注意事项.....788
26 通用定时器（Timer0）.....789
26.1 简介.....789
26.2 基本框图.....789
26.3 功能说明.....790
26.3.1 时钟源选择.....790
26.3.1.1 同步计数时钟源.....790
26.3.1.2 异步计数时钟源.....790
26.3.2 基本计数.....790
26.3.3 硬件触发.....791
26.4 中断及事件说明.....792
26.4.1 中断输出.....792
26.4.2 事件输出.....792
26.5 寄存器说明.....793
26.5.1 计数值寄存器（TMR0_CNTmR）（m=A~B）.....794
26.5.2 基准值寄存器（TMR0_CMPmR）（m=A~B）.....794
26.5.3 基本控制寄存器（TMR0_BCONR）.....795
26.5.4 触发选择寄存器（TMR0_HTSSR）.....798

26.5.5 状态标志寄存器（TMR0_STFLR）..... 798
26.6 使用注意事项..... 799
27 实时时钟（RTC）..... 800
27.1 简介..... 800
27.2 基本框图..... 801
27.3 功能说明..... 802
27.3.1 上电设定..... 802
27.3.2 RTC 计数开始设定..... 802
27.3.3 系统低功耗模式切换..... 802
27.3.4 读出计数寄存器..... 803
27.3.5 写入计数寄存器..... 803
27.3.6 闹钟设定..... 803
27.3.7 时钟误差补偿..... 803
27.3.8 1Hz 输出..... 804
27.3.9 入侵检测..... 805
27.3.10 时间戳功能..... 806
27.4 中断说明..... 807
27.4.1 闹钟中断..... 807
27.4.2 定周期中断..... 807
27.4.3 入侵检测中断..... 807
27.5 寄存器说明..... 808
27.5.1 控制寄存器 0（RTC_CR0）..... 809
27.5.2 控制寄存器 1（RTC_CR1）..... 810
27.5.3 控制寄存器 2（RTC_CR2）..... 811
27.5.4 控制寄存器 3（RTC_CR3）..... 812
27.5.5 秒计数寄存器（RTC_SEC）..... 812
27.5.6 分计数寄存器（RTC_MIN）..... 813
27.5.7 时计数寄存器（RTC_HOUR）..... 814
27.5.8 日计数寄存器（RTC_DAY）..... 816
27.5.9 周计数寄存器（RTC_WEEK）..... 817
27.5.10 月计数寄存器（RTC_MON）..... 818
27.5.11 年计数寄存器（RTC_YEAR）..... 818
27.5.12 分闹钟寄存器（RTC_ALMMIN）..... 819
27.5.13 时闹钟寄存器（RTC_ALMHOUR）..... 819
27.5.14 周闹钟寄存器（RTC_ALMWEEK）..... 820
27.5.15 时钟误差补偿寄存器（RTC_ERRCRH、RTC_ERRCRL）..... 821
27.5.16 入侵控制寄存器 0（RTC_TPCR0）..... 823
27.5.17 入侵控制寄存器 1（RTC_TPCR1）..... 824
27.5.18 入侵状态寄存器（RTC_TPSR）..... 825
27.5.19 秒时间戳寄存器（RTC_SECTP）..... 825
27.5.20 分时间戳寄存器（RTC_MINTP）..... 826
27.5.21 时时间戳寄存器（RTC_HOURTP）..... 826

27.5.22 日时间戳寄存器（RTC_DAYTP） 827
27.5.23 月时间戳寄存器（RTC_MONTP） 827
28 看门狗计数器（WDT/SWDT） 828
28.1 简介 828
28.2 功能说明 829
28.2.1 启动看门狗 829
28.2.2 硬件启动方式 830
28.2.3 软件启动方式 831
28.2.4 刷新动作 832
28.2.5 标志位 832
28.2.6 中断复位 833
28.2.7 计数下溢 834
28.2.8 刷新错误 835
28.3 寄存器说明 836
28.3.1 控制寄存器（SWDT_CR、WDT_CR） 837
28.3.2 状态寄存器（SWDT_SR、WDT_SR） 839
28.3.3 刷新寄存器（SWDT_RR、WDT_RR） 839
28.4 使用注意事项 839
29 通用同步异步收发器（USART） 840
29.1 简介 840
29.2 USART 系统框图 843
29.3 管脚说明 843
29.4 功能说明 844
29.4.1 UART 844
29.4.1.1 时钟 844
29.4.1.2 数据格式 845
29.4.1.3 调制解调器操作 845
29.4.1.4 发送器 846
29.4.1.5 接收器 848
29.4.1.6 UART 接收 TIMEOUT 功能 851
29.4.1.7 RX 线唤醒停止模式功能 852
29.4.1.8 UART 半双工通信方式 853
29.4.1.9 UART 中断和事件 853
29.4.2 多处理器通信 854
29.4.2.1 功能简介 854
29.4.2.2 数据格式 854
29.4.2.3 动作说明 855
29.4.2.4 中断和事件 858
29.4.3 UART-LIN 859
29.4.3.1 功能简介 859
29.4.3.2 LIN 数据格式 859
29.4.3.3 LIN 发送 859

29.4.3.4 LIN 接收 860
29.4.3.5 LIN 中断和事件 861
29.4.4 智能卡 862
29.4.4.1 连接示意图 862
29.4.4.2 时钟 862
29.4.4.3 数据格式 863
29.4.4.4 智能卡的初始化设定步骤 864
29.4.4.5 智能卡模式动作说明 864
29.4.4.6 中断和事件 866
29.4.5 时钟同步模式 867
29.4.5.1 时钟 867
29.4.5.2 数据格式 867
29.4.5.3 调制解调器操作 868
29.4.5.4 发送器 868
29.4.5.5 接收器 870
29.4.5.6 同时发送接收数据 872
29.4.5.7 时钟同步模式中断和事件 873
29.4.6 数字滤波功能 874
29.4.7 中断 874
29.5 寄存器说明 875
29.5.1 状态寄存器（USART_SR） 876
29.5.2 数据寄存器（USART_DR） 879
29.5.3 波特率寄存器（USART_BRR） 880
29.5.4 控制寄存器 1（USART_CR1） 882
29.5.5 控制寄存器 2（USART_CR2） 885
29.5.6 控制寄存器 3（USART_CR3） 888
29.5.7 预分频寄存器（USART_PR） 890
29.5.8 LIN 波特率测量计数寄存器（USART_LBMC） 891
29.5.9 USART1 滤波控制寄存器（USART1_NFC） 892
29.6 使用注意事项 893
29.6.1 UART 注意事项 893
29.6.2 时钟同步模式注意事项 893
29.6.3 其他注意事项 893
30 集成电路总线（I²C） 894
30.1 简介 894
30.2 I²C 系统框图 895
30.2.1 系统框图 895
30.2.2 结构图 896
30.3 动作说明 897
30.3.1 I²C 协议 897
30.3.1.1 开始条件 897
30.3.1.2 地址传送 898

30.3.1.3 数据传送 899
30.3.1.4 停止条件 903
30.3.1.5 重新开始条件 903
30.3.1.6 SCL 时钟同步 904
30.3.1.7 仲裁 904
30.3.1.8 握手 905
30.3.2 地址匹配 907
30.3.2.1 从机地址匹配 907
30.3.2.2 广播地址匹配 910
30.3.2.3 SMBus 主机地址匹配 911
30.3.2.4 SMBus 报警响应地址匹配 912
30.3.2.5 SMBus 默认地址匹配 913
30.3.3 SMBus 动作 913
30.3.3.1 SMBus 超时测量 913
30.3.3.2 数据包错误码（PEC） 915
30.3.4 复位 915
30.3.5 中断和事件信号输出 916
30.3.6 可编程数字滤波 918
30.4 应用软件设定 I2C 初始化流程 918
30.5 寄存器说明 919
30.5.1 I2C 控制寄存器 1(I2C_CR1) 920
30.5.2 I2C 控制寄存器 1(I2C_CR2) 922
30.5.3 I2C 控制寄存器 1(I2C_CR3) 924
30.5.4 I2C 控制寄存器 4(I2C_CR4) 925
30.5.5 I2C 从机地址寄存器 0(I2C_SLR0) 926
30.5.6 I2C 从机地址寄存器 1(I2C_SLR1) 927
30.5.7 I2C SCL 电平超时控制寄存器(I2C_SLTR) 928
30.5.8 I2C 状态寄存器(I2C_SR) 929
30.5.9 I2C 状态清零寄存器(I2C_CLR) 934
30.5.10 I2C 数据发送寄存器(I2C_DTR) 936
30.5.11 I2C 数据接收寄存器(I2C_DRR) 936
30.5.12 I2C 数据移位寄存器(I2C_DSR) 937
30.5.13 I2C 时钟控制寄存器(I2C_CCR) 938
30.5.14 I2C 滤波控制寄存器(I2C_FLTR) 940
31 串行外设接口（SPI） 941
31.1 简介 941
31.2 SPI 系统框图 942
31.3 管脚说明 943
31.4 SPI 动作系统说明 944
31.4.1 主机模式的管脚状态 944
31.4.2 从机模式的管脚状态 945
31.4.3 SPI 系统连接实例 946

31.5 数据通信说明.....948
31.5.1 波特率.....948
31.5.2 数据格式.....949
31.5.3 传送格式.....951
31.5.4 通信方式.....953
31.6 运行说明.....955
31.6.1 运行模式概要.....955
31.6.2 SPI 运行模式时的主机动作.....957
31.6.3 SPI 运行模式时的从机动作.....958
31.6.4 时钟同步运行模式中的主机动作.....959
31.6.5 时钟同步运行模式中的从机动作.....960
31.6.6 几种 SPI 动作的处理流程.....962
31.7 奇偶校验位自诊断.....963
31.8 错误检测.....964
31.8.1 欠载错误.....965
31.8.2 模式错误.....965
31.8.3 过载错误.....965
31.8.4 奇偶校验错误.....967
31.9 SPI 的初始化.....969
31.9.1 清除 SPE 位进行初始化.....969
31.9.2 系统复位初始化.....969
31.10 中断.....970
31.11 可供使用的事件触发源.....971
31.12 寄存器说明.....972
31.12.1 SPI 数据寄存器（SPI_DR）.....973
31.12.2 SPI 控制寄存器（SPI_CR1）.....974
31.12.3 SPI 通信配置寄存器 1（SPI_CFG1）.....976
31.12.4 SPI 状态寄存器（SPI_SR）.....978
31.12.5 SPI 通信配置寄存器 2（SPI_CFG2）.....979
32 四线式串行外设接口（QSPI）.....981
32.1 简介.....981
32.2 内存映射.....983
32.2.1 内部总线空间.....983
32.2.2 ROM 空间及总线的地址宽度.....984
32.3 QSPI 总线.....985
32.3.1 SPI 协议.....985
32.3.2 SPI 模式.....987
32.4 QSPI 总线的时序调整.....988
32.4.1 QSPI 总线基准时钟.....988
32.4.2 SPI 总线基准时钟.....990
32.4.3 QSSN 信号最小高电平宽度.....990
32.4.4 QSSN 的建立时间.....990

32.4.5 QSSN 的保持时间 992
32.4.6 串行数据接收延迟 993
32.5 用于 ROM 访问的 SPI 指令介绍 994
32.5.1 现有的 QSPI-ROM 指令参考 994
32.5.2 标准读指令 995
32.5.3 快速读指令 995
32.5.4 二线式输出快速读指令 997
32.5.5 二线式输入输出快速读指令 998
32.5.6 四线式输出快速读指令 1000
32.5.7 四线式输入输出快速读指令 1002
32.5.8 进入 4-Byte 模式指令 1004
32.5.9 退出 4-Byte 模式指令 1005
32.5.10 写许可指令 1006
32.6 QSPI 总线周期的安排 1007
32.6.1 独立转换的单一闪存读取 1007
32.6.2 使用预读取功能进行闪存读取 1007
32.6.3 预读取的终止 1008
32.6.4 预读取状态监测 1008
32.6.5 使用 QSPI 总线周期延长功能进行闪存读取 1009
32.7 XIP 控制 1010
32.7.1 XIP 模式的设置 1011
32.7.2 XIP 模式的退出 1011
32.8 QSIO2 和 QSIO3 的管脚状态 1012
32.9 直接通信模式 1013
32.9.1 关于直接通信模式 1013
32.9.2 直接通信模式的设置 1013
32.9.3 直接通信模式下 QSPI 总线周期的生成 1013
32.10 中断 1014
32.11 使用上的注意事项 1015
32.11.1 QSPI 寄存器的设置顺序 1015
32.11.2 模块停止信号的设置 1015
32.12 寄存器说明 1015
32.12.1 QSPI 控制寄存器（QSCR） 1016
32.12.2 QSPI 片选控制寄存器（QSCSCR） 1019
32.12.3 QSPI 格式控制寄存器（QSFCR） 1020
32.12.4 QSPI 状态寄存器（QSSR） 1022
32.12.5 QSPI 指令代码寄存器（QSCCMD） 1024
32.12.6 QSPI 直接通信指令寄存器（QSDCOM） 1024
32.12.7 QSPI XIP 模式代码寄存器（QSXCMD） 1025
32.12.8 QSPI 系统配置寄存器（QSSR2） 1025
32.12.9 QSPI 外部扩展地址寄存器（QSEXAR） 1026
33 集成电路内置音频总线模块（I2S） 1027

33.1 简介.....1027
33.2 I2S 系统框图.....1028
33.3 管脚说明.....1029
33.4 功能说明.....1030
33.4.1 I2S 一般说明.....1030
33.4.2 通信方式.....1030
33.4.3 支持的音频协议.....1031
33.4.3.1 I2S Philips 标准.....1031
33.4.3.2 MSB 对齐标准.....1032
33.4.3.3 LSB 对齐标准.....1033
33.4.3.4 PCM 标准.....1035
33.4.4 时钟发生器.....1037
33.4.5 I2S 主模式.....1039
33.4.6 I2S 从模式.....1040
33.4.7 I2S 中断.....1042
33.4.8 使用上的注意事项.....1043
33.4.8.1 作为主机使用时的注意事项.....1043
33.4.8.2 作为从机使用时的注意事项.....1045
33.5 寄存器说明.....1046
33.5.1 I2S 控制寄存器（I2S_CTRL）.....1047
33.5.2 I2S 状态寄存器（I2S_SR）.....1049
33.5.3 I2S 错误状态寄存器（I2S_ER）.....1050
33.5.4 I2S 配置寄存器（I2S_CFGR）.....1051
33.5.5 I2S 发送缓冲 FIFO 数据寄存器（I2S_TXBUF）.....1052
33.5.6 I2S 接收缓冲 FIFO 数据寄存器（I2S_RXBUF）.....1053
33.5.7 I2S 分频寄存器（I2S_PR）.....1054
33.5.8 CMU I2S 时钟配置存器(CMU_I2SCKSEL).....1055
34 USB2.0 高速模块（USBHS）.....1057
34.1 USBHS 简介.....1057
34.2 USBHS 主要特性.....1058
34.2.1 通用特性.....1058
34.2.2 主机模式特性.....1059
34.2.3 设备模式特性.....1059
34.3 USBHS 系统框图.....1060
34.4 USBHS 管脚说明.....1061
34.5 USBHS 功能说明.....1062
34.5.1 USBHS 时钟以及工作模式.....1062
34.5.2 USBHS 模式决定.....1062
34.5.3 USBHS 主机功能.....1062
34.5.3.1 主机功能简介.....1062
34.5.3.2 主机端口供电.....1063
34.5.3.3 主机检测设备连接和断开.....1063

34.5.3.4 主机枚举.....1063
34.5.3.5 主机挂起.....1064
34.5.3.6 主机通道.....1064
34.5.3.7 主机调度器.....1066
34.5.4 USBHS 设备功能.....1068
34.5.4.1 设备功能简介.....1068
34.5.4.2 设备供电状态.....1068
34.5.4.3 设备默认状态.....1069
34.5.4.4 设备挂起状态.....1069
34.5.4.5 设备软断开.....1069
34.5.4.6 设备端点.....1069
34.5.5 USBHS SOF 脉冲管脚输出功能.....1073
34.5.5.1 主机 SOF.....1073
34.5.5.2 设备 SOF.....1073
34.5.6 USBHS 功耗控制.....1074
34.5.7 USBHS 动态更新 USBHS_HFIR 寄存器.....1075
34.5.8 USBHS 数据 FIFO.....1075
34.5.9 USBHS 主机 FIFO 架构.....1076
34.5.9.1 主机 RxFIFO.....1076
34.5.9.2 主机 TxFIFO.....1077
34.5.10 USBHS 设备 FIFO 架构.....1078
34.5.10.1 设备 RxFIFO.....1078
34.5.10.2 设备 TxFIFO.....1079
34.5.11 USBHS FIFO RAM 分配.....1079
34.5.11.1 主机模式.....1079
34.5.11.2 设备模式.....1080
34.5.12 USBHS 系统性能.....1080
34.5.13 USBHS 中断和事件.....1082
34.6 USBHS 编程模型.....1084
34.6.1 USBHS 模块初始化.....1084
34.6.2 USBHS 主机初始化.....1084
34.6.3 USBHS 设备初始化.....1085
34.6.4 USBHS DMA 模式.....1086
34.6.5 USBHS 主机编程模型.....1086
34.6.5.1 通道初始化.....1086
34.6.5.2 通道停止.....1087
34.6.5.3 Ping 协议.....1087
34.6.6 USBHS 设备编程模型.....1088
34.6.6.1 USB 复位时的端点端点初始化.....1088
34.6.6.2 USB 复位时的端点端点初始化.....1089
34.6.6.3 收到 SetAddress 命令时的端点初始化.....1089
34.6.6.4 收到 SetConfiguration/SetInterface 命令时的端点初始化.....1089

34.6.6.5 端点激活.....1090
34.6.6.6 端点停用.....1090
34.6.7 USBHS 操作模型.....1090
34.6.7.1 SETUP 和 OUT 数据传输.....1090
34.6.7.2 IN 数据传输.....1100
34.7 寄存器说明.....1110
34.7.1 USB 系统控制寄存器.....1116
34.7.1.1 USB 系统控制寄存器(USB_SYCTLREG).....1116
34.7.2 USBHS 全局寄存器.....1118
34.7.2.1 USBHS VBUS 控制寄存器(USBHS_GVBUSCFG).....1118
34.7.2.2 USBHS AHB 控制寄存器(USBHS_GAHBCFG).....1119
34.7.2.3 USBHS USB 配置寄存器 (USBHS_GUSBCFG).....1121
34.7.2.4 USBHS 复位寄存器 (USBHS_GRSTCTL).....1124
34.7.2.5 USBHS 全局中断状态寄存器 (USBHS_GINTSTS).....1127
34.7.2.6 USBHS 全局中断屏蔽寄存器 (USBHS_GINTMSK).....1132
34.7.2.7 USBHS 接收状态调试读取/USBHS 状态读取和出栈寄存器 (USBHS_GRXSTSR/USBHS_GRXSTSP).....1136
34.7.2.8 USBHS 接收 FIFO 大小寄存器 (USBHS_GRXFSIZ).....1138
34.7.2.9 USBHS 主机非周期性发送 FIFO 大小寄存器(USBHS_HNPTXFSIZ)/端点 0 发送 FIFO 大小(USBHS_DIEPTXF0).....1139
34.7.2.10 USBHS 非周期性发送 FIFO/队列状态寄存器 (USBHS_HNPTXSTS).....1140
34.7.2.11 USBHS 模块 ID 寄存器(USBHS_CID).....1142
34.7.2.12 USBHS LPM 配置寄存器(USBHS_GLPMCFG).....1143
34.7.2.13 USBHS 主机周期性发送 FIFO 大小寄存器(USBHS_HPTXFSIZ).....1146
34.7.2.14 USBHS 设备 IN 端点发送 FIFO 大小寄存器 (USBHS_DIEPTXFx)（x = 1..15）1147
34.7.3 USBHS 主机模式寄存器.....1148
34.7.3.1 USBHS 主机配置寄存器 (USBHS_HCFG).....1148
34.7.3.2 USBHS 主机帧时间间隔寄存器 (USBHS_HFIR).....1149
34.7.3.3 USBHS 主机帧编号/帧剩余时间寄存器 (USBHS_HFNUM).....1150
34.7.3.4 USBHS 主机周期性发送 FIFO/队列状态寄存器(USBHS_HPTXSTS).....1151
34.7.3.5 USBHS 主机全体通道中断寄存器 (USBHS_HAINT).....1153
34.7.3.6 USBHS 主机全体通道中断屏蔽寄存器(USBHS_HAINTMSK).....1154
34.7.3.7 USBHS 主机端口控制和状态寄存器 (USBHS_HPRT).....1155
34.7.3.8 USBHS 主机通道 x 特性寄存器 (USBHS_HCCHARx)（x = 0..15）.....1158
34.7.3.9 USBHS 主机通道 x 分离控制寄存器 (USBHS_HCSPLTx)（x = 0..15）.....1160
34.7.3.10 USBHS 主机通道 x 中断寄存器 (USBHS_HCINTx)（x = 0..15）.....1161
34.7.3.11 USBHS 主机通道 x 中断屏蔽寄存器 (USBHS_HCINTMSKx)（x = 0..15）.....1163
34.7.3.12 USBHS 主机通道 x 传输大小寄存器 (USBHS_ HCTSIZx)（x = 0..15）.....1165
34.7.3.13 USBHS 主机通道 xDMA 地址寄存器 (USBHS_HCDMAx)（x = 0..15）.....1166
34.7.4 USBHS 设备模式寄存器.....1167
34.7.4.1 USBHS 设备配置寄存器(USBHS_DCFG).....1167

34.7.4.2 USBHS 设备控制寄存器(USBHS_DCTL) 1169
34.7.4.3 USBHS 设备状态寄存器(USBHS_DSTS) 1171
34.7.4.4 USBHS 设备 IN 端点通用中断屏蔽寄存器(USBHS_DIEPMSK) 1173
34.7.4.5 USBHS 设备 OUT 端点通用中断屏蔽寄存器(USBHS_DOEPMSK) 1175
34.7.4.6 USBHS 设备全体端点中断寄存器 (USBHS_DAINT) 1177
34.7.4.7 USBHS 设备全体端点中断屏蔽寄存器 (USBHS_DAINTMSK) 1178
34.7.4.8 USBHS 设备阈值控制寄存器(USBHS_DTHRCTL) 1179
34.7.4.9 USBHS 设备 IN 端点 FIFO 空中断屏蔽寄存器(USBHS_DIEPEMPMSK) 1180
34.7.4.10 USBHS 设备单个端点中断寄存器(USBHS_DEACHINT) 1181
34.7.4.11 USBHS 设备单个端点中断屏蔽寄存器(USBHS_DEACHINTMASK) 1182
34.7.4.12 USBHS 设备 IN 端 1 中断屏蔽寄存器 (USBHS_DIEPEACHMSK1) 1183
34.7.4.13 USBHS 设备 OUT 端点 1 中断屏蔽寄存器 (USBHS_DOEPEACHMSK1) 1185
34.7.4.14 USBHS 设备控制 IN 端点 0 控制寄存器(USBHS_DIEPCTL0) 1187
34.7.4.15 USBHS 设备 IN 端点 x 控制寄存器(USBHS_DIEPCTLx)(x=1..15) 1189
34.7.4.16 USBHS 设备 IN 端点 x 中断寄存器 (USBHS_DIEPINTx) (x=0..15) 1192
34.7.4.17 USBHS 设备 IN 端点 0 传输大小寄存器(USBHS_DIEPTSIZ0) 1194
34.7.4.18 USBHS 设备 IN 端点 x 传输大小寄存器(USBHS_DIEPTSIZx)(x=1..15) 1195
34.7.4.19 USBHS 设备 IN 端点 x DMA 地址寄存器（USBHS_DIEPDMAx)(x=0..15) 1196
34.7.4.20 USBHS 设备 IN 端点发送 FIFO 状态寄存器（USBHS_ DTXFSTSx)(x=0..15)... 1197
34.7.4.21 USBHS 设备控制 OUT 端点 0 控制寄存器(USBHS_DOEPCTL0) 1198
34.7.4.22 USBHS 设备 OUT 端点 x 控制寄存器(USBHS_DOEPCTLx)(x=1..15)..... 1200
34.7.4.23 USBHS 设备 OUT 端点 x 中断寄存器 (USBHS_DOEPINTx) (x=0..15) 1203
34.7.4.24 USBHS 设备 OUT 端点 0 传输大小寄存器(USBHS_DOEPTSIZ0)..... 1205
34.7.4.25 USBHS 设备 OUT 端点 x 传输大小寄存器(USBHS_DOEPTSIZx)(x=1..15)..... 1206
34.7.4.26 USBHS 设备 OUT 端点 x DMA 地址寄存器（USBHS_DOEPDMAx)(x=0..15).. 1207
34.7.5 USBHS 时钟门控控制寄存器 1208
34.7.5.1 USBHS 时钟门控控制寄存器(USBHS_GCCTL) 1208
35 USB2.0 全速模块（USBFS） 1209
35.1 USBFS 简介 1209
35.2 USBFS 主要特性 1209
35.2.1 通用特性 1209
35.2.2 主机模式特性 1210
35.2.3 设备模式特性 1210
35.3 USBFS 系统框图 1211
35.4 USBFS 管脚说明 1212
35.5 USBFS 功能说明 1213
35.5.1 USBFS 时钟以及工作模式 1213
35.5.2 USBFS 模式决定 1213
35.5.3 USBFS 主机功能 1213
35.5.3.1 主机功能简介 1213
35.5.3.2 主机端口供电 1214
35.5.3.3 主机检测设备连接和断开 1214

35.5.3.4 主机枚举.....1214
35.5.3.5 主机挂起.....1215
35.5.3.6 主机通道.....1215
35.5.3.7 主机调度器.....1217
35.5.4 USBFS 设备功能.....1219
35.5.4.1 设备功能简介.....1219
35.5.4.2 设备供电状态.....1219
35.5.4.3 设备默认状态.....1220
35.5.4.4 设备挂起状态.....1220
35.5.4.5 设备软断开.....1220
35.5.4.6 设备端点.....1220
35.5.5 USBFS SOF 脉冲管脚输出功能.....1224
35.5.5.1 主机 SOF.....1224
35.5.5.2 设备 SOF.....1224
35.5.6 USBFS 功耗控制.....1225
35.5.7 USBFS 动态更新 USBFS_HFIR 寄存器.....1226
35.5.8 USBFS 数据 FIFO.....1226
35.5.9 USBFS 主机 FIFO 架构.....1227
35.5.9.1 主机 RxFIFO.....1227
35.5.9.2 主机 TxFIFO.....1227
35.5.10 USBFS 设备 FIFO 架构.....1229
35.5.10.1 设备 RxFIFO.....1229
35.5.10.2 设备 TxFIFO.....1230
35.5.11 USBFS FIFO RAM 分配.....1230
35.5.11.1 主机模式.....1230
35.5.11.2 设备模式.....1231
35.5.12 USBFS 系统性能.....1231
35.5.13 USBFS 中断和事件.....1232
35.6 USBFS 编程模型.....1234
35.6.1 USBFS 模块初始化.....1234
35.6.2 USBFS 主机初始化.....1234
35.6.3 USBFS 设备初始化.....1235
35.6.4 USBFS DMA 模式.....1236
35.6.5 USBFS 主机编程模型.....1236
35.6.5.1 通道初始化.....1236
35.6.5.2 通道停止.....1237
35.6.6 USBFS 设备编程模型.....1237
35.6.6.1 USB 复位时的端点端点初始化.....1237
35.6.6.2 USB 复位时的端点端点初始化.....1238
35.6.6.3 收到 SetAddress 命令时的端点初始化.....1238
35.6.6.4 收到 SetConfiguration/SetInterface 命令时的端点初始化.....1238
35.6.6.5 端点激活.....1239

35.6.6.6 端点停用.....1239
35.6.7 USBFS 操作模型.....1240
35.6.7.1 SETUP 和 OUT 数据传输.....1240
35.6.7.2 IN 数据传输.....1250
35.7 寄存器说明.....1260
35.7.1 USB 系统控制寄存器.....1265
35.7.1.1 USB 系统控制寄存器(USB_SYCTLREG).....1265
35.7.2 USBFS 全局寄存器.....1267
35.7.2.1 USBFS VBUS 控制寄存器(USBFS_GVBUSCFG).....1267
35.7.2.2 USBFS AHB 控制寄存器(USBFS_GAHBCFG).....1268
35.7.2.3 USBFS USB 配置寄存器 (USBFS_GUSBCFG).....1270
35.7.2.4 USBFS 复位寄存器 (USBFS_GRSTCTL).....1272
35.7.2.5 USBFS 全局中断状态寄存器 (USBFS_GINTSTS).....1275
35.7.2.6 USBFS 全局中断屏蔽寄存器 (USBFS_GINTMSK).....1280
35.7.2.7 USBFS 接收状态调试读取/USBFS 状态读取和出栈寄存器 (USBFS_GRXSTSR/USBFS_GRXSTSP).....1284
35.7.2.8 USBFS 接收 FIFO 大小寄存器 (USBFS_GRXFSIZ).....1286
35.7.2.9 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).....1287
35.7.2.10 USBFS 非周期性发送 FIFO/队列状态寄存器 (USBFS_HNPTXSTS).....1288
35.7.2.11 USBFS 模块 ID 寄存器(USBFS_CID).....1290
35.7.2.12 USBFS LPM 配置寄存器(USBFS_GLPMCFG).....1291
35.7.2.13 USBFS 主机周期性发送 FIFO 大小寄存器(USBFS_HPTXFSIZ).....1294
35.7.2.14 USBFS 设备 IN 端点发送 FIFO 大小寄存器 (USBFS_DIEPTXFx)（x = 1..15）1295
35.7.3 USBFS 主机模式寄存器.....1296
35.7.3.1 USBFS 主机配置寄存器 (USBFS_HCFG).....1296
35.7.3.2 USBFS 主机帧时间间隔寄存器 (USBFS_HFIR).....1297
35.7.3.3 USBFS 主机帧编号/帧剩余时间寄存器 (USBFS_HFNUM).....1298
35.7.3.4 USBFS 主机周期性发送 FIFO/队列状态寄存器(USBFS_HPTXSTS).....1299
35.7.3.5 USBFS 主机全体通道中断寄存器 (USBFS_HAINT).....1301
35.7.3.6 USBFS 主机全体通道中断屏蔽寄存器(USBFS_HAINTMSK).....1302
35.7.3.7 USBFS 主机端口控制和状态寄存器 (USBFS_HPRT).....1303
35.7.3.8 USBFS 主机通道 x 特性寄存器 (USBFS_HCCHARx)（x = 0..15）.....1306
35.7.3.9 USBFS 主机通道 x 中断寄存器 (USBFS_HCINTx)（x = 0..15）.....1308
35.7.3.10 USBFS 主机通道 x 中断屏蔽寄存器 (USBFS_HCINTMSKx)（x = 0..15）.....1310
35.7.3.11 USBFS 主机通道 x 传输大小寄存器 (USBFS_ HCTSIZx)（x = 0..15）.....1312
35.7.3.12 USBFS 主机通道 xDMA 地址寄存器 (USBFS_HCDMAx)（x = 0..15）.....1313
35.7.4 USBFS 设备模式寄存器.....1314
35.7.4.1 USBFS 设备配置寄存器(USBFS_DCFG).....1314
35.7.4.2 USBFS 设备控制寄存器(USBFS_DCTL).....1316
35.7.4.3 USBFS 设备状态寄存器(USBFS_DSTS).....1318
35.7.4.4 USBFS 设备 IN 端点通用中断屏蔽寄存器(USBFS_DIEPMSK).....1320

35.7.4.5 USBFS 设备 OUT 端点通用中断屏蔽寄存器(USBFS_DOEPMSK) 1322
35.7.4.6 USBFS 设备全体端点中断寄存器 (USBFS_DAINT) 1323
35.7.4.7 USBFS 设备全体端点中断屏蔽寄存器 (USBFS_DAINTMSK) 1324
35.7.4.8 USBFS 设备 IN 端点 FIFO 空中断屏蔽寄存器(USBFS_DIEPEMPMSK) 1325
35.7.4.9 USBFS 设备控制 IN 端点 0 控制寄存器(USBFS_DIEPCTL0) 1326
35.7.4.10 USBFS 设备 IN 端点 x 控制寄存器(USBFS_DIEPCTLx)(x=1..15) 1328
35.7.4.11 USBFS 设备 IN 端点 x 中断寄存器 (USBFS_DIEPINTx) (x=0..15) 1331
35.7.4.12 USBFS 设备 IN 端点 0 传输大小寄存器(USBFS_DIEPTSIZ0) 1333
35.7.4.13 USBFS 设备 IN 端点 x 传输大小寄存器(USBFS_DIEPTSIZx)(x=1..15) 1334
35.7.4.14 USBFS 设备 IN 端点 x DMA 地址寄存器（USBFS_DIEPDMAx)(x=0..15) 1335
35.7.4.15 USBFS 设备 IN 端点发送 FIFO 状态寄存器（USBFS_ DTXFSTSx)(x=0..15) 1336
35.7.4.16 USBFS 设备控制 OUT 端点 0 控制寄存器(USBFS_DOEPCTL0) 1337
35.7.4.17 USBFS 设备 OUT 端点 x 控制寄存器(USBFS_DOEPCTLx)(x=1..15) 1339
35.7.4.18 USBFS 设备 OUT 端点 x 中断寄存器 (USBFS_DOEPINTx) (x=0..15) 1342
35.7.4.19 USBFS 设备 OUT 端点 0 传输大小寄存器(USBFS_DOEPTSIZ0) 1344
35.7.4.20 USBFS 设备 OUT 端点 x 传输大小寄存器(USBFS_DOEPTSIZx)(x=1..15) 1345
35.7.4.21 USBFS 设备 OUT 端点 x DMA 地址寄存器（USBFS_DOEPDMAx)(x=0..15) ... 1346
35.7.5 USBFS 时钟门控控制寄存器 1347
35.7.5.1 USBFS 时钟门控控制寄存器(USBFS_GCCTL) 1347
36 CAN FD 控制器（CAN FD） 1348
36.1 简介 1348
36.2 CAN FD 系统框图 1349
36.3 管脚说明 1350
36.4 功能说明 1351
36.4.1 动作模式 1351
36.4.2 波特率设定 1351
36.4.3 发送缓冲器 1355
36.4.4 接收缓冲器 1355
36.4.5 接收筛选寄存器组 1356
36.4.6 数据发送 1357
36.4.7 单次数据发送 1358
36.4.8 取消数据发送 1358
36.4.9 数据接收 1359
36.4.10 错误处理 1359
36.4.11 节点关闭 1360
36.4.12 仲裁失败位置捕捉 1360
36.4.13 回环模式 1360
36.4.14 静默模式 1361
36.4.15 软件复位功能 1362
36.4.16 向上兼容 CAN-FD 功能 1364
36.4.17 时间触发 TTCAN 1364
36.4.17.1 TTCAN 模式下的 TBUF 行为 1364

36.4.17.2 TTCAN 功能描述 1364
36.4.17.3 TTCAN 时序 1365
36.4.17.4 TTCAN 触发方式 1365
36.4.17.5 TTCAN 触发看门时间 1366
36.4.18 TDC 和 RDC 1367
36.4.19 中断 1368
36.5 寄存器说明 1369
36.5.1 CAN 接收 BUF 寄存器（CAN_RBUF） 1370
36.5.2 CAN 发送 BUF 寄存器（CAN_TBUF） 1373
36.5.3 CAN 配置和状态寄存器（CAN_CFG_STAT） 1375
36.5.4 CAN 命令寄存器（CAN_TCMD） 1376
36.5.5 CAN 发送控制寄存器（CAN_TCTRL） 1378
36.5.6 CAN 接收控制寄存器（CAN_RCTRL） 1380
36.5.7 CAN 接收和发送中断使能寄存器（CAN_RTIE） 1381
36.5.8 CAN 接收和发送中断状态寄存器（CAN_RTIF） 1382
36.5.9 CAN 错误中断使能和标志寄存器（CAN_ERRINT） 1384
36.5.10 CAN Slow 位时序寄存器（CAN_SBT） 1385
36.5.11 CAN Fast 位时序寄存器（CAN_FBT） 1386
36.5.12 CAN 发送器延迟补偿寄存器（CAN_TDC） 1387
36.5.13 CAN 错误和仲裁失败捕捉寄存器（CAN_EALCAP） 1388
36.5.14 CAN 警告限定寄存器（CAN_LIMIT） 1388
36.5.15 CAN 接收错误计数器寄存器（CAN_RECNT） 1389
36.5.16 CAN 发送错误计数器寄存器（CAN_TECNT） 1389
36.5.17 CAN 筛选器组控制寄存器（CAN_ACFCTRL） 1390
36.5.18 CAN 筛选器组使能寄存器（CAN_ACFEN） 1391
36.5.19 CAN 筛选器组 code 和 mask 寄存器（CAN_ACF） 1392
36.5.20 TTCAN TB slot 指针寄存器（CAN_TBSLOT） 1393
36.5.21 TTCAN 时间触发配置寄存器（CAN_TTCFG） 1394
36.5.22 TTCAN 参考消息寄存器（CAN_REF_MSG） 1395
36.5.23 TTCAN 触发配置寄存器（CAN_TRG_CFG） 1396
36.5.24 TTCAN 触发时间寄存器（CAN_TT_TRIG） 1397
36.5.25 TTCAN 触发看门时间寄存器（CAN_TT_WTRIG） 1397
37 CAN2.0B 控制器（CAN2.0B） 1398
37.1 简介 1398
37.2 CAN 系统框图 1399
37.3 管脚说明 1400
37.4 功能说明 1401
37.4.1 动作模式 1401
37.4.2 波特率设定 1401
37.4.3 发送缓冲器 1402
37.4.4 接收缓冲器 1403
37.4.5 接收筛选寄存器组 1403

37.4.6 数据发送 1404
37.4.7 单次数据发送 1405
37.4.8 取消数据发送 1405
37.4.9 数据接收 1406
37.4.10 错误处理 1406
37.4.11 节点关闭 1407
37.4.12 仲裁失败位置捕捉 1407
37.4.13 回环模式 1407
37.4.14 静默模式 1408
37.4.15 软件复位功能 1409
37.4.16 向上兼容 CAN-FD 功能 1411
37.4.17 时间触发 TTCAN 1411
37.4.17.1 TTCAN 模式下的 TBUF 行为 1411
37.4.17.2 TTCAN 功能描述 1411
37.4.17.3 TTCAN 时序 1412
37.4.17.4 TTCAN 触发方式 1412
37.4.17.5 TTCAN 触发看门时间 1413
37.4.18 中断 1414
37.5 寄存器说明 1415
37.5.1 CAN 接收 BUF 寄存器（CAN_RBUF） 1416
37.5.2 CAN 发送 BUF 寄存器（CAN_TBUF） 1419
37.5.3 CAN 配置和状态寄存器（CAN_CFG_STAT） 1421
37.5.4 CAN 命令寄存器（CAN_TCMD） 1422
37.5.5 CAN 发送控制寄存器（CAN_TCTRL） 1424
37.5.6 CAN 接收控制寄存器（CAN_RCTRL） 1426
37.5.7 CAN 接收和发送中断使能寄存器（CAN_RTIE） 1427
37.5.8 CAN 接收和发送中断状态寄存器（CAN_RTIF） 1428
37.5.9 CAN 错误中断使能和标志寄存器（CAN_ERRINT） 1430
37.5.10 CAN Slow 位时序寄存器（CAN_SBT） 1431
37.5.11 CAN 错误和仲裁失败捕捉寄存器（CAN_EALCAP） 1432
37.5.12 CAN 警告限定寄存器（CAN_LIMIT） 1432
37.5.13 CAN 接收错误计数器寄存器（CAN_RECNT） 1433
37.5.14 CAN 发送错误计数器寄存器（CAN_TECNT） 1433
37.5.15 CAN 筛选器组控制寄存器（CAN_ACFCTRL） 1434
37.5.16 CAN 筛选器组使能寄存器（CAN_ACFEN） 1435
37.5.17 CAN 筛选器组 code 和 mask 寄存器（CAN_ACF） 1436
37.5.18 TTCAN TB slot 指针寄存器（CAN_TBSLOT） 1437
37.5.19 TTCAN 时间触发配置寄存器（CAN_TTCFG） 1438
37.5.20 TTCAN 参考消息寄存器（CAN_REF_MSG） 1439
37.5.21 TTCAN 触发配置寄存器（CAN_TRG_CFG） 1440
37.5.22 TTCAN 触发时间寄存器（CAN_TT_TRIG） 1441
37.5.23 TTCAN 触发看门时间寄存器（CAN_TT_WTRIG） 1441

38 SDIO 控制器（SDIOC）.....1442
38.1 简介.....1442
38.2 功能描述.....1443
38.2.1 端口分配.....1443
38.2.2 基本访问方式.....1444
38.2.3 数据传输.....1444
38.2.4 SD 时钟.....1444
38.2.5 中断和 DMA 启动请求.....1445
38.2.5.1 SD 中断.....1445
38.2.5.2 SDIO 中断.....1445
38.2.5.3 DMA 请求.....1445
38.2.5.4 卡的插入(insert)和移除(removal).....1446
38.2.6 主机和设备初始化.....1446
38.2.6.1 主机初始化.....1446
38.2.6.2 SD 卡初始化.....1447
38.2.6.3 MMC 卡初始化.....1447
38.2.6.4 SDIO 初始化.....1447
38.2.7 SD/MMC 单数据块(single block)读写.....1448
38.2.8 SD/MMC 多数据块(multi block)读写.....1449
38.2.9 传输终止(abort)，挂起(suspend)和恢复(resume).....1450
38.2.10 读等待(read wait).....1451
38.2.11 唤醒(wakeup).....1452
38.3 寄存器说明.....1454
38.3.1 数据块长度寄存器(BLKSIZE).....1456
38.3.2 数据块计数存器(BLKCNT).....1456
38.3.3 参数寄存器 0(ARG0).....1456
38.3.4 参数寄存器 1(ARG1).....1457
38.3.5 传输模式寄存器(TRANSMODE).....1457
38.3.6 命令寄存器(CMD).....1458
38.3.7 应答寄存器 0(RESP0).....1458
38.3.8 应答寄存器 1(RESP1).....1459
38.3.9 应答寄存器 2(RESP2).....1459
38.3.10 应答寄存器 3(RESP3).....1459
38.3.11 应答寄存器 4(RESP4).....1460
38.3.12 应答寄存器 5(RESP5).....1460
38.3.13 应答寄存器 6(RESP6).....1460
38.3.14 应答寄存器 7(RESP7).....1461
38.3.15 数据缓冲寄存器 0(BUF0).....1461
38.3.16 数据缓冲寄存器 1(BUF1).....1461
38.3.17 主机状态寄存器(PSTAT).....1462
38.3.18 主机控制寄存器(HOSTCON).....1463
38.3.19 电源控制寄存器(PWRCON).....1463

38.3.20 数据块间隙控制寄存器(BLKGPCON) 1464
38.3.21 时钟控制寄存器(CLKCON) 1465
38.3.22 超时控制寄存器(TOUTCON) 1466
38.3.23 软件复位寄存器(SFTRST) 1467
38.3.24 普通中断状态寄存器(NORINTST) 1468
38.3.25 错误中断状态寄存器(ERRINTST) 1469
38.3.26 普通中断状态使能寄存器(NORINTSTEN) 1470
38.3.27 错误中断状态使能寄存器(ERRINTSTEN) 1471
38.3.28 普通中断信号使能寄存器(NORINTSGEN) 1472
38.3.29 错误中断信号使能寄存器(ERRINTSGEN) 1473
38.3.30 自动命令错误状态寄存器(ATCERRST) 1474
38.3.31 强制自动命令错误状态控制寄存器(FEA) 1475
38.3.32 强制错误状态控制寄存器(FEE) 1476
38.3.33 MMC 模式使能寄存器(MMCER) 1477
39 以太网 MAC 控制器（ETHMAC） 1478
39.1 概要 1478
39.2 基本特性 1479
39.2.1 基本框图 1479
39.2.2 ETH_MAC 特性 1479
39.2.3 ETH_PTP 特性 1482
39.2.4 ETH_DMA 特性 1482
39.3 接口说明 1484
39.3.1 MII 接口 1484
39.3.2 RMII 接口 1486
39.3.3 SMI 接口 1487
39.3.4 以太网端口配置 1487
39.4 功能说明 1489
39.4.1 ETH_MAC 功能 1489
39.4.1.1 MAC 802.3 帧格式 1489
39.4.1.2 MAC 帧发送动作 1490
发送基本协议 1491
调度算法程序 1492
冲突期间重发 1493
TxFIFO 刷新 1493
CRC 处理和 PAD 生成 1493
源地址 SA 处理 1494
VLAN 标识处理 1494
发送状态字 1494
发送暂停帧 1494
COE 引擎 1495
发送时序 1497
39.4.1.3 MAC 帧接收动作 1498

接收基本协议 1499
接收错误处理 1499
CRC 计算和 PAD 去除 1499
接收状态字 1500
接收暂停帧 1500
COE 引擎 1500
接收时序 1501
39.4.1.4 MAC 帧过滤 1503
MAC 源地址过滤 1503
MAC 目标地址过滤-单播 1504
MAC 目标地址过滤-多播 1504
MAC 目标地址过滤-广播 1504
MAC 目标地址 Hash 过滤 1504
VLAN 标识符过滤 1505
VLAN 标识符 Hash 过滤 1505
L3 源地址&目标地址过滤 1505
L4 源端口&目标端口过滤 1506
39.4.1.5 MAC Loopback 模式 1506
39.4.1.6 MAC 管理计数器（MMC） 1506
39.4.1.7 MAC 电源管理（PMT） 1507
远程唤醒帧过滤 1507
远程唤醒帧检测 1509
魔术数据包检测 1509
掉电期间系统状态 1510
39.4.2 ETH_PTP 功能 1511
39.4.2.1 PTP 参考时序源 1512
39.4.2.2 PTP 报文类型 1512
39.4.2.3 PTP 帧发送功能 1513
39.4.2.4 PTP 帧接收功能 1513
39.4.2.5 PTP 系统时间校准 1513
39.4.2.6 PTP 系统时间生成初始化 1515
39.4.2.7 PTP PPS 输出 1516
39.4.3 ETH_DMA 功能 1517
39.4.3.1 DMA 初始化 1518
总线突发访问 1519
数据缓冲区对齐 1519
缓冲区大小计算 1520
DMA 仲裁器 1520
39.4.3.2 DMA 错误响应 1521
39.4.3.3 DMA 发送配置 1521
TxDMA 操作 1521
发送帧处理 1525

发送轮询挂起 1526
常规型 Tx 描述符 1526
增强型 Tx 描述符 1530
39.4.3.4 DMA 接收配置 1532
RxDMA 操作 1532
接收帧获取 1534
接收帧处理 1535
接收过程挂起 1535
常规型 Rx 描述符 1536
增强型 Rx 描述符 1539
39.5 中断说明 1542
39.5.1 DMA 中断 1542
39.5.2 PMT 中断 1543
39.5.3 PTP 中断 1543
39.5.4 MMC 中断 1543
39.6 寄存器说明 1545
39.6.1 ETH_MAC 寄存器 1548
39.6.1.1 ETH_MAC 接口选择寄存器（ETH_MAC_IFCONFR） 1548
39.6.1.2 ETH_MAC MAC 配置寄存器（ETH_MAC_CONFIGR） 1549
39.6.1.3 ETH_MAC MAC 流控寄存器（ETH_MAC_FLOCTLR） 1552
39.6.1.4 ETH_MAC MAC 状态寄存器（ETH_MAC_MACSTSR） 1554
39.6.1.5 ETH_MAC 中断状态寄存器（ETH_MAC_INTSTSR） 1556
39.6.1.6 ETH_MAC 中断屏蔽寄存器（ETH_MAC_INTMSKR） 1557
39.6.1.7 ETH_MAC SMI 地址寄存器（ETH_MAC_SMIADDR） 1558
39.6.1.8 ETH_MAC SMI 数据寄存器（ETH_MAC_SMIDATR） 1559
39.6.1.9 ETH_MAC PMT 控制状态寄存器（ETH_MAC_PMTCTLR） 1560
39.6.1.10 ETH_MAC 远程唤醒帧过滤寄存器（ETH_MAC_RTWKFFR） 1562
39.6.1.11 ETH_MAC MAC 帧过滤控制寄存器（ETH_MAC_FLTCTLR） 1563
39.6.1.12 ETH_MAC MAC 地址高位寄存器 0（ETH_MAC_MACADHR0） 1565
39.6.1.13 ETH_MAC MAC 地址低位寄存器 0（ETH_MAC_MACADLR0） 1565
39.6.1.14 ETH_MAC MAC 地址高位寄存器 1（ETH_MAC_MACADHR1） 1566
39.6.1.15 ETH_MAC MAC 地址低位寄存器 1（ETH_MAC_MACADLR1） 1566
39.6.1.16 ETH_MAC MAC 地址高位寄存器 2（ETH_MAC_MACADHR2） 1567
39.6.1.17 ETH_MAC MAC 地址低位寄存器 2（ETH_MAC_MACADLR2） 1567
39.6.1.18 ETH_MAC MAC 地址高位寄存器 3（ETH_MAC_MACADHR3） 1568
39.6.1.19 ETH_MAC MAC 地址低位寄存器 3（ETH_MAC_MACADLR3） 1568
39.6.1.20 ETH_MAC MAC 地址高位寄存器 4（ETH_MAC_MACADHR4） 1569
39.6.1.21 ETH_MAC MAC 地址低位寄存器 4（ETH_MAC_MACADLR4） 1569
39.6.1.22 ETH_MAC MAC Hash 表高位寄存器（ETH_MAC_HASHTHR） 1570
39.6.1.23 ETH_MAC MAC Hash 表低位寄存器（ETH_MAC_HASHTLR） 1570
39.6.1.24 ETH_MAC VLAN 标记发送控制寄存器（ETH_MAC_VTACTLR） 1571
39.6.1.25 ETH_MAC VLAN 标记接收过滤寄存器（ETH_MAC_VTAFLTR） 1572

39.6.1.26 ETH_MAC VLAN Hash 表寄存器（ETH_MAC_VLAHTBR） 1572
39.6.1.27 ETH_MAC LAY3LAY4 控制寄存器（ETH_MAC_L34CTLR） 1573
39.6.1.28 ETH_MAC LAY4 端口寄存器 （ETH_MAC_L4PORTR） 1575
39.6.1.29 ETH_MAC LAY3 地址寄存器 0（ETH_MAC_L3ADDRR0） 1575
39.6.1.30 ETH_MAC LAY3 地址寄存器 1（ETH_MAC_L3ADDRR1） 1576
39.6.1.31 ETH_MAC LAY3 地址寄存器 2（ETH_MAC_L3ADDRR2） 1576
39.6.1.32 ETH_MAC LAY3 地址寄存器 3（ETH_MAC_L3ADDRR3） 1577
39.6.2 ETH_PTP 寄存器 1578
39.6.2.1 ETH_PTP 时间戳控制寄存器（ETH_PTP_TSPCTLR） 1578
39.6.2.2 ETH_PTP 时间戳状态寄存器（ETH_PTP_TSPSTSR） 1580
39.6.2.3 ETH_PTP 时间戳基本加数寄存器（ETH_PTP_TSPADDR） 1580
39.6.2.4 ETH_PTP 时间戳亚秒加数寄存器（ETH_PTP_TSPNSAR） 1581
39.6.2.5 ETH_PTP 时间戳系统秒寄存器（ETH_PTP_TMSSECR） 1581
39.6.2.6 ETH_PTP 时间戳系统亚秒寄存器（ETH_PTP_TMSNSER） 1581
39.6.2.7 ETH_PTP 时间戳更新秒寄存器（ETH_PTP_TMUSECR） 1582
39.6.2.8 ETH_PTP 时间戳更新亚秒寄存器（ETH_PTP_TMUNSER） 1582
39.6.2.9 ETH_PTP 时间戳目标秒寄存器 0（ETH_PTP_TMTSECR0） 1582
39.6.2.10 ETH_PTP 时间戳目标亚秒寄存器 0（ETH_PTP_TMTNSER0） 1583
39.6.2.11 ETH_PTP 时间戳目标秒寄存器 1（ETH_PTP_TMTSECR1） 1583
39.6.2.12 ETH_PTP 时间戳目标亚秒寄存器 1（ETH_PTP_TMTNSER1） 1583
39.6.2.13 ETH_PTP PPS 输出控制寄存器（ETH_PTP_PPSCTLR） 1584
39.6.3 ETH_DMA 寄存器 1586
39.6.3.1 ETH_DMA 总线模式寄存器（ETH_DMA_BUSMODR） 1586
39.6.3.2 ETH_DMA 操作模式寄存器（ETH_DMA_OPRMODR） 1588
39.6.3.3 ETH_DMA 动作状态寄存器（ETH_DMA_DMASTSR） 1591
39.6.3.4 ETH_DMA 中断使能寄存器（ETH_DMA_INTENAR） 1594
39.6.3.5 ETH_DMA 帧丢失统计寄存器（ETH_DMA_RFRCNTR） 1596
39.6.3.6 ETH_DMA 看门狗定时寄存器（ETH_DMA_REVWDTR） 1596
39.6.3.7 ETH_DMA 发送轮询要求寄存器（ETH_DMA_TXPOLLR） 1597
39.6.3.8 ETH_DMA 接收轮询要求寄存器（ETH_DMA_RXPOLLR） 1597
39.6.3.9 ETH_DMA 发送描述符列表地址寄存器（ETH_DMA_TXDLADR） 1598
39.6.3.10 ETH_DMA 接收描述符列表地址寄存器（ETH_DMA_RXDLADR） 1598
39.6.3.11 ETH_DMA 当前主机发送描述符寄存器（ETH_DMA_CHTXDER） 1598
39.6.3.12 ETH_DMA 当前主机接收描述符寄存器（ETH_DMA_CHRXDER） 1599
39.6.3.13 ETH_DMA 当前主机发送缓冲区寄存器（ETH_DMA_CHTXBFR） 1599
39.6.3.14 ETH_DMA 当前主机接收缓冲区寄存器（ETH_DMA_CHRXBFR） 1599
39.6.4 ETH_MMC 寄存器 1600
39.6.4.1 ETH_MMC MMC 控制寄存器（ETH_MMC_MMCCTLR） 1600
39.6.4.2 ETH_MMC Rx 统计状态寄存器（ETH_MMC_REVSTSR） 1601
39.6.4.3 ETH_MMC Tx 统计状态寄存器（ETH_MMC_TRSSTSR） 1602
39.6.4.4 ETH_MMC Rx 中断控制寄存器（ETH_MMC_RITCTLR） 1603
39.6.4.5 ETH_MMC Tx 中断控制寄存器（ETH_MMC_TITCTLR） 1604

39.6.4.6 ETH_MMC Rx 单播好帧统计寄存器（ETH_MMC_RXUNGFR）.....1605
39.6.4.7 ETH_MMC Rx 多播好帧统计寄存器（ETH_MMC_RXMUGFR）.....1605
39.6.4.8 ETH_MMC Rx 广播好帧统计寄存器（ETH_MMC_RXBRGFR）.....1605
39.6.4.9 ETH_MMC RxCRC 错误帧统计寄存器（ETH_MMC_RXCREFR）.....1606
39.6.4.10 ETH_MMC Rx 对齐错误帧统计寄存器（ETH_MMC_RXALEFR）.....1606
39.6.4.11 ETH_MMC Rx 矮帧错误帧统计寄存器（ETH_MMC_RXRUEFR）.....1606
39.6.4.12 ETH_MMC Rx 长度错误帧统计寄存器（ETH_MMC_RXLEEFR）.....1607
39.6.4.13 ETH_MMC Rx 超范围错误帧统计寄存器（ETH_MMC_RXOREFR）.....1607
39.6.4.14 ETH_MMC Tx 单播好帧统计寄存器（ETH_MMC_TXUNGFR）.....1607
39.6.4.15 ETH_MMC Tx 多播好帧统计寄存器（ETH_MMC_TXMUGFR）.....1608
39.6.4.16 ETH_MMC Tx 广播好帧统计寄存器（ETH_MMC_TXBRGFR）.....1608
39.6.4.17 ETH_MMC Tx 延迟错误帧统计寄存器（ETH_MMC_TXDEEFR）.....1608
39.6.4.18 ETH_MMC Tx 载波错误帧统计寄存器（ETH_MMC_TXCAEFR）.....1609
39.6.4.19 ETH_MMC Tx 延迟冲突错误帧统计寄存器（ETH_MMC_TXLCEFR）.....1609
39.6.4.20 ETH_MMC Tx 过度冲突错误帧统计寄存器（ETH_MMC_TXECEFR）.....1609
39.6.4.21 ETH_MMC Tx 过度延迟错误帧统计寄存器（ETH_MMC_TXEDEFR）.....1610
40 外部存储器控制器（EXMC）.....1611
40.1 概要.....1611
40.2 基本特性.....1612
40.2.1 功能列表.....1612
40.2.2 控制器架构.....1612
40.2.3 基本访问规范.....1613
40.2.4 地址映射.....1614
40.2.5 协议接口.....1619
40.2.5.1 SMC 协议接口.....1619
40.2.5.2 DMC 协议接口.....1620
40.2.5.3 NFC 协议接口.....1620
40.3 功能说明.....1622
40.3.1 SMC-SRAM/PSRAM/NOR Flash 控制器.....1622
40.3.1.1 SRAM/PSRAM/NOR Flash 介绍.....1622
40.3.1.2 SMC 基本功能.....1622
40.3.1.3 SMC 初始设定.....1623
40.3.1.4 SMC 访问动作.....1624
访问方式.....1624
访问时序.....1625
FIFO 管理.....1634
40.3.1.5 SMC 低功耗管理.....1634
40.3.2 DMC-SDRAM 控制器.....1635
40.3.2.1 SDRAM 介绍.....1635
Bank 特性.....1635
行激活.....1635
预充电.....1636

刷新动作 1637
命令真值表 1638
40.3.2.2 DMC 基本功能 1639
40.3.2.3 DMC 初始设定 1640
40.3.2.4 DMC 访问动作 1642
访问时序 1642
FIFO 管理 1644
40.3.2.5 DMC 低功耗管理 1644
Low Power 状态控制 1644
DMC_CLK 输出控制 1645
DMC_CKE 输出控制 1645
40.3.3 NFC-NAND Flash 控制器 1646
40.3.3.1 NAND Flash 介绍 1646
40.3.3.2 NFC 基本功能 1646
40.3.3.3 NFC 访问命令 1647
40.3.3.4 NFC 访问动作 1648
Page Read 1648
Page Write 1648
Block Erase 1648
Reset 1649
Read ID 1649
Read Status 1649
40.4 中断说明 1650
40.5 寄存器说明 1651
40.5.1 SMC-SRAM/PSRAM/NOR Flash 寄存器 1653
40.5.1.1 使能寄存器（SMC_ENAR） 1653
40.5.1.2 状态寄存器（SMC_STSR） 1653
40.5.1.3 命令寄存器（SMC_CMDR） 1654
40.5.1.4 状态控制寄存器（SMC_STCR0） 1654
40.5.1.5 状态控制寄存器（SMC_STCR1） 1655
40.5.1.6 刷新时间寄存器（SMC_RFTR） 1655
40.5.1.7 基本控制寄存器（SMC_BACR） 1656
40.5.1.8 片选控制寄存器（SMC_CSCR0） 1656
40.5.1.9 片选控制寄存器（SMC_CSCR1） 1657
40.5.1.10 CHIP 配置寄存器（SMC_CPCR） 1658
40.5.1.11 CHIP 状态寄存器（SMC_CPSR0~3） 1659
40.5.1.12 时序配置寄存器（SMC_TMCR） 1660
40.5.1.13 时序状态寄存器（SMC_TMSR0~3） 1661
40.5.2 DMC-SDRAM 寄存器 1662
40.5.2.1 使能寄存器（DMC_ENAR） 1662
40.5.2.2 状态寄存器（DMC_STSR） 1662
40.5.2.3 命令寄存器（DMC_CMDR） 1663

40.5.2.4 状态控制寄存器（DMC_STCR）.....1664
40.5.2.5 刷新时间寄存器（DMC_RFTR）.....1664
40.5.2.6 基本控制寄存器（DMC_BACR）.....1665
40.5.2.7 片选控制寄存器（DMC_CSCR0~3）.....1666
40.5.2.8 CHIP 配置寄存器（DMC_CPCR）.....1667
40.5.2.9 时序配置寄存器（DMC_TMCR_t_casl）.....1669
40.5.2.10 时序配置寄存器（DMC_TMCR_t_dqss）.....1669
40.5.2.11 时序配置寄存器（DMC_TMCR_t_mrd）.....1670
40.5.2.12 时序配置寄存器（DMC_TMCR_t_ras）.....1670
40.5.2.13 时序配置寄存器（DMC_TMCR_t_rc）.....1671
40.5.2.14 时序配置寄存器（DMC_TMCR_t_rcd）.....1671
40.5.2.15 时序配置寄存器（DMC_TMCR_t_rfc）.....1672
40.5.2.16 时序配置寄存器（DMC_TMCR_t_rp）.....1672
40.5.2.17 时序配置寄存器（DMC_TMCR_t_rrd）.....1673
40.5.2.18 时序配置寄存器（DMC_TMCR_t_wr）.....1673
40.5.2.19 时序配置寄存器（DMC_TMCR_t_wtr）.....1674
40.5.2.20 时序配置寄存器（DMC_TMCR_t_xp）.....1674
40.5.2.21 时序配置寄存器（DMC_TMCR_t_xsr）.....1675
40.5.2.22 时序配置寄存器（DMC_TMCR_t_esr）.....1675
40.5.3 NFC-NAND Flash 寄存器.....1676
40.5.3.1 使能寄存器（NFC_ENAR）.....1676
40.5.3.2 状态寄存器（NFC_STSR）.....1676
40.5.3.3 控制寄存器（NFC_STCR）.....1677
40.5.3.4 数据寄存器（NFC_DATR）.....1677
40.5.3.5 命令寄存器（NFC_CMDR）.....1678
40.5.3.6 索引寄存器（NFC_IDXR0）.....1680
40.5.3.7 索引寄存器（NFC_IDXR1）.....1681
40.5.3.8 基本配置寄存器（NFC_BACR）.....1682
40.5.3.9 中断使能寄存器（NFC_IENR）.....1683
40.5.3.10 中断状态寄存器（NFC_ISTR）.....1684
40.5.3.11 中断结果寄存器（NFC_IRSR）.....1685
40.5.3.12 时序配置寄存器（NFC_TMCR0）.....1685
40.5.3.13 时序配置寄存器（NFC_TMCR1）.....1686
40.5.3.14 时序配置寄存器（NFC_TMCR2）.....1687
40.5.3.15 ECC 校验寄存器（NFC_ECCR）.....1688
40.6 使用注意事项.....1692
41 数字视频接口（DVP）.....1693
41.1 简介.....1693
41.2 系统框图.....1693
41.3 功能说明.....1695
41.3.1 视频数据格式.....1695
41.3.1.1 单色格式.....1695

41.3.1.2 YCbCr 格式 1695
41.3.1.3 RGB565 格式 1696
41.3.1.4 JPEG 格式 1696
41.3.2 并口存储格式 1697
41.3.2.1 8Bit 数据 1697
41.3.2.2 10Bit 数据 1698
41.3.2.3 12Bit 数据 1698
41.3.2.4 14Bit 数据 1698
41.3.3 模式选择 1699
41.3.3.1 单帧模式 1699
41.3.3.2 连续模式 1699
41.3.4 同步控制 1701
41.3.4.1 软件同步 1701
41.3.4.2 硬件同步 1701
41.3.5 窗口裁剪 1702
41.3.6 FIFO 控制 1703
41.3.7 DMA 控制 1703
41.4 中断及事件说明 1704
41.4.1 帧传送状态中断及事件 1704
41.4.2 软件同步错误中断及事件 1704
41.4.3 FIFO 溢出错误中断及事件 1705
41.5 寄存器说明 1706
41.5.1 控制寄存器（DVP_CTR） 1707
41.5.2 数据寄存器（DVP_DTR） 1708
41.5.3 状态寄存器（DVP_STR） 1708
41.5.4 中断寄存器（DVP_IER） 1709
41.5.5 DMA 数据传输寄存器（DVP_DMR） 1709
41.5.6 软件同步数据寄存器（DVP_SSYNDR） 1710
41.5.7 软件同步屏蔽寄存器（DVP_SSYNMR） 1710
41.5.8 窗口裁剪偏移寄存器（DVP_CPSFTR） 1711
41.5.9 窗口裁剪尺寸寄存器（DVP_CPSZER） 1711
42 加密协处理模块（CPM） 1712
42.1 简介 1712
42.2 加解密算法处理器（AES） 1713
42.2.1 算法简介 1713
42.2.2 AES 模块功能描述 1715
42.2.3 加密操作流程 1715
42.2.4 解密操作流程 1715
42.2.5 数据示例 1716
42.2.6 运行时间说明 1719
42.2.7 操作注意事项 1719
42.2.8 寄存器说明 1720

42.2.8.1 AES 控制寄存器（AES_CR） 1721
42.2.8.2 AES 数据寄存器（AES_DRx）（x=0~3） 1722
42.2.8.3 AES 密钥寄存器（AES_KRx）（x=0~7） 1723
42.3 安全散列算法（HASH） 1725
42.3.1 算法简介 1725
42.3.2 操作流程 1726
42.3.3 消息填充 1727
42.3.4 HMAC 运算 1729
42.3.4.1 HMAC 运算过程 1729
42.3.4.2 HMAC 软件操作流程 1729
42.3.4.3 HMAC 使用 DMA 操作流程 1730
42.3.4.4 HMAC 运算举例 1731
42.3.4.5 注意事项 1731
42.3.5 中断说明 1731
42.3.6 硬件触发事件选择 1732
42.3.7 寄存器说明 1733
42.3.7.1 HASH 控制寄存器（HASH_CR） 1735
42.3.7.2 HASH 摘要寄存器（HASH_HR） 1737
42.3.7.3 HASH 数据寄存器（HASH_DR） 1738
42.3.7.4 HASH 硬件事件触发选择寄存器 A（HASH_ITRGSELA） 1739
42.3.7.5 HASH 硬件事件触发选择寄存器 B（HASH_ITRGSELB） 1740
42.4 真随机数发生器（TRNG） 1741
42.4.1 模块框图 1741
42.4.2 操作流程 1741
42.4.3 中断及事件输出 1741
42.4.4 操作注意事项 1742
42.4.5 寄存器说明 1742
42.4.5.1 TRNG 控制寄存器（TRNG_CR） 1742
42.4.5.2 TRNG 模式寄存器（TRNG_MR） 1743
42.4.5.3 TRNG 数据寄存器（TRNG_DR） 1743
43 CRC 运算（CRC） 1744
43.1 简介 1744
43.2 功能框图 1744
43.3 功能说明 1745
43.3.1 CRC16 编码模式 1745
43.3.2 CRC16 校验模式 1745
43.3.3 CRC32 编码模式 1746
43.3.4 CRC32 校验模式 1746
43.4 寄存器说明 1747
43.4.1 控制寄存器（CRC_CR） 1747
43.4.2 结果寄存器（CRC_RESLT） 1748
43.4.3 数据寄存器（CRC_DAT） 1748

44 数据计算单元（DCU） 1749
44.1 概要....1749
44.2 功能描述....1750
44.2.1 加法模式....1750
44.2.2 减法模式....1750
44.2.3 硬件触发启动模式....1750
44.2.4 比较模式....1751
44.2.5 中断和事件信号输出....1752
44.2.6 三角波输出模式....1752
44.2.7 递增锯齿波输出模式....1755
44.2.8 递减锯齿波输出模式....1756
44.3 寄存器说明....1759
44.3.1 DCU 控制寄存器(DCUx_CTL)(x=1~8)....1764
44.3.2 DCU 标志寄存器(DCUx_FLAG)(x=1~4)....1765
44.3.3 DCU 标志寄存器(DCUx_FLAG)(x=5~8)....1766
44.3.4 DCU 数据寄存器(DCUx_DATAy)(x=1~8,y=0,1,2)....1767
44.3.5 DCU 标志复位寄存器(DCUx_FLAGCLR) (x=1~4)....1770
44.3.6 DCU 标志复位寄存器(DCUx_FLAGCLR) (x=5~8)....1771
44.3.7 DCU 中断和事件寄存器(DCUx_INTEVTSEL) (x=1~4)....1772
44.3.8 DCU 中断和事件寄存器(DCUx_INTEVTSEL) (x=5~8)....1774
44.3.9 DCU 触发源选择寄存器(DCUx_TRGSEL)(x=1~4)....1776
45 数学运算单元（MAU）....1777
45.1 简介....1777
45.2 功能说明....1777
45.2.1 开方运算....1777
45.2.1.1 操作流程....1777
45.2.1.2 结果处理....1777
45.2.1.3 定点数处理....1778
45.2.2 正弦运算....1778
45.2.2.1 操作流程....1778
45.3 中断及事件说明....1779
45.3.1 中断输出....1779
45.3.2 事件输出....1779
45.4 寄存器说明....1780
45.4.1 控制状态寄存器（MAU_CSR）....1780
45.4.2 数据输入寄存器 0（MAU_DTR0）....1781
45.4.3 结果输出寄存器 0（MAU_RTR0）....1781
45.4.4 数据输入寄存器 1（MAU_DTR1）....1781
45.4.5 结果输出寄存器 1（MAU_RTR1）....1782
46 滤波数学加速器（FMAC）....1783
46.1 简介....1783
46.2 基本框图....1783

46.3 操作流程.....1784
46.4 模块使能.....1784
46.5 系数归一化.....1784
46.6 中断及事件说明.....1785
46.6.1 中断输出.....1785
46.6.2 事件输出.....1785
46.7 寄存器说明.....1786
46.7.1 模块使能寄存器（FMAC_ENR）.....1786
46.7.2 基本控制寄存器（FMAC_CTR）.....1787
46.7.3 中断控制寄存器（FMAC_IER）.....1787
46.7.4 数据输入寄存器（FMAC_DTR）.....1788
46.7.5 滤波系数寄存器（FMAC_COR0~16）.....1788
46.7.6 结果输出寄存器 0（FMAC_RTR0）.....1788
46.7.7 结果输出寄存器 1（FMAC_RTR1）.....1789
46.7.8 运算状态寄存器（FMAC_STR）.....1789
47 调试控制器（DBGC）.....1790
47.1 简介.....1790
47.2 DBGC 系统框图.....1791
47.3 SWJ-DP 调试端口（SWD 和 JTAG）.....1792
47.3.1 JTAG-DP 或 SW-DP 的切换机制.....1793
47.4 引脚排列和调试端口引脚.....1794
47.4.1 SWJ 调试端口引脚.....1794
47.4.2 灵活的 SWJ-DP 引脚分配.....1794
47.4.3 JTAG 引脚上的内部上拉.....1795
47.4.4 使用串行接口以及释放未使用的调试引脚以作 GPIO.....1795
47.5 寄存器说明.....1796
47.5.1 DBG 状态寄存器（MCUDBGSTAT）.....1797
47.5.2 外设调试暂停寄存器（MCUSTPCTL）.....1798
47.5.3 调试组件配置寄存器（MCUTRACECTL）.....1800
47.5.4 外设调试暂停寄存器 2（MCUSTPCTL2）.....1801
47.6 SW 调试端口.....1803
47.6.1 SW 协议简介.....1803
47.7 TPIU（跟踪端口接口单元）.....1804
47.7.1 简介.....1804
47.7.2 TRACE 引脚分配.....1804
47.7.3 MCU 内部 TRACECLKIN 连接.....1806
47.7.4 TPIU 寄存器.....1806
47.7.5 TPIU 配置示例.....1807
版本信息 & 联系方式.....1808

表目录

表 1-1 存储器映射 71
表 1-2 QSPI 地址空间分配 72
表 1-3 目标地址配置主闪存示例(MMF_REMCR0.EN0=1 或 MMF_REMCR1.EN1=1) 73
表 1-4 目标地址配置高速 SRAM 示例(MMF_REMCR0.EN0=1 或 MMF_REMCR1.EN1=1) 73
表 1-5 寄存器列表 74
表 3-1 复位方式及产生条件 83
表 3-2 复位方式及复位标志 85
表 3-3 RMU 寄存器一览 94
表 4-1 各个内部时钟的规格 103
表 5-1 BOR 配置 163
表 5-2 PVD1/PVD2 特性 164
表 5-3 运行模式 172
表 5-4 低功耗模式 172
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 175
表 5-6 运行模式说明 176
表 5-7 掉电模式子模式 180
表 5-8 RAM 模块与 RAM 掉电控制位 185
表 5-9 寄存器保护列表 186
表 5-10 寄存器一览表 192
表 6-1 寄存器一览表 227
表 7-1 CPU 时钟频率和 FLASH 读等待周期对照表 235
表 7-2 FLASH 实际读周期数 236
表 7-3 OTP 地址分布表 243
表 7-4 寄存器一览表 247
表 8-1 SRAM 读写访问的等待周期设定和 CPU 时钟频率的关系 271
表 8-2 SRAM 空间分配 271
表 9-1 PORT 寄存器一览 1 287
表 9-2 PORT 寄存器一览 2 288
表 9-3 32bit 访问时 PORT 寄存器一览 305
表 10-1 中断向量表 316
表 10-2 中断事件请求序号与选择 334
表 10-3 寄存器列表 346
表 13-1 KEYSCAN 管脚说明 423
表 13-2 KEYSCAN 寄存器一览表 425
表 14-1 HRC 目标频率为 20MHz 时的测量误差 431
表 14-2 HRC 目标频率为 16MHz 时的测量误差 432
表 14-3 CTC 寄存器一览表 436
表 15-1 通道重置说明 448
表 15-2 寄存器一览表 458
表 16-1 CMP 引脚 482

表 16-2 CMP 寄存器一览表 488
表 16-3 正端输入电压一览表 494
表 16-4 负端输入电压一览表 494
表 16-5 定时器窗口 PWM 一览表 495
表 17-1 各 ADC 单元规格 500
表 17-2 序列 A 和 B 的各种竞争 506
表 17-3 AD 转换时间 509
表 17-4 ADC 寄存器一览 1/2 520
表 17-5 ADC 寄存器一览 2/2 521
表 18-1 DAC 引脚 546
表 18-2 DAC1 寄存器一览表 551
表 18-3 DAC2 寄存器一览表 551
表 18-4 D/A 转换和模拟输出控制 554
表 20-1 Timer6 的基本功能及特性 564
表 20-2 Timer6 端口列表 565
表 20-3 不同模式下的功能对比表 603
表 20-4 Timer6 寄存器列表 607
表 20-5 计数器（CNTER）控制优先级 651
表 20-6 PWMA 端口输出控制优先级 652
表 20-7 PWMB 端口输出控制优先级 652
表 21-1 寄存器列表 657
表 22-1 Timer4 的基本功能及特性 660
表 22-2 Timer4 端口列表 661
表 22-3 Timer4 寄存器列表 685
表 22-4 PWM 端口输出状态与寄存器设定值 705
表 24-1 TimerA 的基本功能及特性 731
表 24-2 TimerA 端口列表 732
表 24-3 TimerA 寄存器列表 747
表 24-4 内部触发事件 HTSSR 选择关系对应表 748
表 25-1 Timer2 的基本功能及特性 768
表 25-2 Timer2 端口列表 769
表 25-3 Timer2 寄存器列表 776
表 26-1 Timer0 寄存器列表 793
表 27-1 RTC 的基本规格 800
表 27-2 寄存器列表 808
表 28-1 看门狗计数器的基本特性 828
表 28-2 寄存器列表 836
表 29-1 USART 管脚说明 843
表 29-2 UART 中断/事件表 853
表 29-3 多处理器模式中断/事件表 858
表 29-4 LIN 中断/事件表 861
表 29-5 智能卡模式中断/事件表 866

表 29-6 时钟同步模式中断/事件表 873
表 29-7 USART 整体中断一览表 874
表 29-8 USART 寄存器一览表 875
表 29-9 波特率计算公式（小数波特率无效 FBME=0） 880
表 29-10 波特率计算公式（小数波特率有效 FBME=1） 881
表 30-1 输入/ 输出引脚 896
表 30-2 寄存器一览表 919
表 31-1 SPI 的特性要点 942
表 31-2 管脚说明 943
表 31-3 主机模式时 SPI 管脚状态说明 944
表 31-4 从机模式时 SPI 管脚状态说明 945
表 31-5 部分设定值的位速 948
表 31-6 SPI 模式和寄存器设定关系 956
表 31-7 错误检测对应表 964
表 31-8 SPI 中断源说明 970
表 32-1 QSPI 主要规格 981
表 32-2 QSPI 管脚 982
表 32-3 QSPI 总线基准时钟选择一览表 989
表 32-4 参考指令一览表 994
表 32-5 QIO2 和 QIO3 的管脚状态 1012
表 32-6 QSPI 寄存器一览表 1015
表 33-1 I2S 主要特性 1027
表 33-2 I2S 管脚说明 1029
表 33-3 音频频率精度(针对 VCO 输入频率=1MHz) 1038
表 33-4 I2S 中断请求 1042
表 33-5 I2S 寄存器一览表 1046
表 34-1 USBHS 管脚说明 1061
表 34-2 USBHS_GLB 中断事件表 1083
表 34-3 USBHS 系统控制寄存器一览表 1111
表 34-4 USBHS 系统控制寄存器一览表 1115
表 35-1 USBFS 管脚说明 1212
表 35-2 USBFS_GLB 中断事件表 1233
表 35-3 USBFS 系统控制寄存器一览表 1261
表 35-4 USBFS 系统控制寄存器一览表 1264
表 36-1 CAN 管脚说明 1350
表 36-2 CAN 位时间设定规则 1353
表 36-3 20MHz 通信时钟时波特率设定建议 1353
表 36-4 40MHz 通信时钟时波特率设定建议 1354
表 36-5 80MHz 通信时钟时波特率设定建议 1354
表 36-6 软件复位范围表 1363
表 36-7 CAN 中断表 1368
表 36-8 CAN 寄存器一览表 1369

表 36-9 标准格式 CAN 接收邮箱格式.....1370
表 36-10 扩展格式 CAN 接收邮箱格式.....1371
表 36-11 DLC 控制位.....1372
表 36-12 标准格式 CAN 发送邮箱格式.....1373
表 36-13 扩展格式 CAN 发送邮箱格式.....1374
表 37-1 CAN 管脚说明.....1400
表 37-2 CAN 位时间设定规则.....1402
表 37-3 软件复位范围表.....1410
表 37-4 CAN 中断表.....1414
表 37-5 CAN 寄存器一览表.....1415
表 37-6 标准格式 CAN 接收邮箱格式.....1416
表 37-7 扩展格式 CAN 接收邮箱格式.....1417
表 37-8 DLC 控制位.....1418
表 37-9 标准格式 CAN 发送邮箱格式.....1419
表 37-10 扩展格式 CAN 发送邮箱格式.....1420
表 39-1 MII 接口信号说明.....1485
表 39-2 发送数据时信号状态说明.....1486
表 39-3 发送数据时信号状态说明.....1486
表 39-4 SMI 接口信号说明.....1487
表 39-5 SMI 帧格式.....1487
表 39-6 ETHMAC 端口功能分配.....1488
表 39-7 时间戳快照目标报文.....1513
表 39-8 常规型 Tx 描述符.....1527
表 39-9 增强型 Tx 描述符.....1531
表 39-10 常规型 Rx 描述符.....1536
表 39-11 增强型描述符.....1539
表 39-12 ETHMAC 寄存器列表.....1547
表 40-1 EXMC 的基本功能.....1612
表 40-2 AHB 访问宽度和存储器位宽对应访问方式表.....1614
表 40-3 SMC 协议接口.....1619
表 40-4 DMC 协议接口.....1620
表 40-5 NFC 协议接口.....1620
表 40-6 EXMC 端口功能分配.....1621
表 40-7 单次读动作基本设定例.....1625
表 40-8 地址数据线复用单次读动作基本设定例.....1630
表 40-9 单次写动作基本设定例.....1631
表 40-10 地址数据线复用单次写动作基本设定例.....1633
表 40-11 突发读动作基本设定例.....1633
表 40-12 突发写动作基本设定例.....1634
表 40-13 DMC 的命令真值表.....1638
表 40-14 ONFI 访问命令.....1647
表 40-15 EXMC 寄存器列表.....1652

表 40-16 CS0 设定和访问地址对应关系表 1666
表 40-17 命令参数列表 1679
表 40-18 索引寄存器值与 MEM 地址对应表 1681
表 40-19 4bitECC 模式时的 ECCR 寄存器列表 1689
表 41-1 DVP 基本功能及特性 1693
表 41-2 DVP 端口列表 1694
表 41-3 单色格式视频数据存储方式 1695
表 41-4 YCbCr 格式视频数据存储方式 1695
表 41-5 RGB565 格式视频数据存储方式 1696
表 41-6 8bit 数据 DVP 存储方式 1697
表 41-7 10bit 数据 DVP 存储方式 1698
表 41-8 12bit 数据 DVP 存储方式 1698
表 41-9 14bit 数据 DVP 存储方式 1698
表 41-10 帧传送中断 1704
表 41-11 同步码错误序列 1704
表 41-12 DVP 寄存器列表 1706
表 42-1 128 位操作寄存器示例 1716
表 42-2 192 位操作寄存器示例 1717
表 42-3 256 位操作寄存器示例 1718
表 42-4 AES 加解密运行时间 1719
表 42-5 寄存器列表 1720
表 43-1 CRC 寄存器列表 1747
表 45-1 MAU 寄存器列表 1780
表 46-1 FMAC 寄存器列表 1786
表 47-1 SWJ 调试端口引脚 1794
表 47-2 灵活的 SWJ-DP 引脚分配 1794

图目录

图 3-1 上电复位 86
图 3-2 NRST 复位时序 87
图 3-3 欠压复位 87
图 3-4 可编程电压检测 1 复位 88
图 3-5 可编程电压检测 2 复位 89
图 3-6 看门狗和专用看门狗复位 90
图 3-7 掉电唤醒复位 90
图 3-8 软件复位 90
图 3-9 MPU 错误复位 91
图 3-10 RAM 奇偶校验复位 91
图 3-11 RAMECC 复位 92
图 3-12 时钟频率异常复位 92
图 3-13 外部高速振荡异常复位 92
图 3-14 M4 Lockup 复位 93
图 4-1 时钟系统框图 98
图 4-2 时钟频率测量框图 99
图 4-3 外部高速振荡器连接事例 105
图 4-4 外部时钟输入的连接事例图 106
图 4-5 外部高速振荡器故障检测例 107
图 4-6 系统时钟选择 XTAL，检测到 XTAL 振荡故障例 108
图 4-7 外部低速振荡器连接事例 110
图 4-8 时钟频率测量时序图 117
图 5-1 电源构成图 160
图 5-2 电池备用域电源切换图 161
图 5-3 上电复位、掉电复位波形 162
图 5-4 欠压复位波形 163
图 5-5 PVD1 中断/复位框图 165
图 5-6 PVD2 中断/复位框图 165
图 5-7 电源监视 1 中断时序图 166
图 5-8 电源监视 1 复位时序图 167
图 5-9 电源监视 2 中断运行时序图 168
图 5-10 电源监视 2 复位运行时序图 169
图 5-11 内部电压采样示意图 169
图 5-12 PTWKn 结构框图 182
图 7-1 FLASH 地址分配 234
图 7-2 OTP 功能未使能时，FLASH 地址分配 245
图 7-3 OTP 功能使能时，FLASH 地址分配 245
图 7-4 启动引导交换 1 246
图 7-5 启动引导交换 2 246
图 9-1 端口基本结构示意图 282

图 10-1 中断系统框图 309
图 10-2 中断事件选择 337
图 10-3 数字滤波器工作示意图 339
图 13-1 KEYSCAN 系统框图 423
图 13-2 键盘扫描功能示意图 424
图 14-1 CTC 的基本框图 430
图 14-2 CTC 校准示意图 433
图 15-1 DMA 结构图 442
图 15-2 连锁传输示意图 446
图 15-3 不连续地址传输示意图（以源地址为例） 447
图 15-4 不连续式重置示意图 449
图 15-5 应用例 1：存储器到存储器传输 452
图 15-6 应用例 2：存储器到外围电路的传输 454
图 16-1 功能框图 481
图 16-2 普通比较模式工作示意图 483
图 16-3 窗口比较模式工作示意图 485
图 17-1 ADC 框图 499
图 17-2 通道映射示意图 501
图 17-3 内部模拟通道选择 502
图 17-4 序列 A 单次扫描模式 503
图 17-5 连续扫描 504
图 17-6 双序列扫描模式（序列 A 从被中断通道重新启动） 506
图 17-7 双序列扫描模式（序列 A 从第一个通道重新启动） 507
图 17-8 模拟看门狗保护区域（比较模式） 508
图 17-9 A/D 转换时间 509
图 17-10 平均功能有效时的转换动作 511
图 17-11 ADC 单元 1,单元 2 的 PGA 与 SH 通道示意图 512
图 17-12 通道专用采样保持电路有效 513
图 17-13 单次并行触发模式（三 ADC） 514
图 17-14 单次延迟触发模式（三 ADC） 516
图 17-15 循环并行触发模式（三 ADC） 517
图 17-16 循环延迟触发模式（两 ADC） 518
图 17-17 循环延迟触发模式（三 ADC） 518
图 17-18 ADC 中断和事件输出时序 519
图 18-1 D/A 转换通道框图 546
图 18-2 D/A 转换示意图 547
图 18-3 A/D 转换优先模式动作示意图 549
图 19-1 OTS 功能框图 556
图 20-1 Timer6 基本框图 565
图 20-2 锯齿波波形（递加计数） 566
图 20-3 三角波波形 566
图 20-4 比较输出动作 568

图 20-5 捕获输入动作 569
图 20-6 硬件刷新动作 569
图 20-7 软件同步动作 571
图 20-8 硬件同步动作 573
图 20-9 脉宽测量 574
图 20-10 周期测量 574
图 20-11 单缓存方式比较输出时序 576
图 20-12 双缓存方式捕获输入时序 577
图 20-13 锯齿波模式时计数缓存动作 578
图 20-14 三角波模式时计数缓存动作 1 579
图 20-15 三角波模式时计数缓存动作 2 579
图 20-16 捕获输入端口的滤波功能 580
图 20-17 单边对齐独立 PWM 581
图 20-18 双边对称独立 PWM 582
图 20-19 软件设定 GCMBR 互补 PWM 波输出 583
图 20-20 硬件设定 GCMBR 互补 PWM 波输出 584
图 20-21 双边非对称 PWM 输出 585
图 20-22 6 相单边对齐独立 PWM 586
图 20-23 带死区时间 3 相双边对称互补 PWM 587
图 20-24 周期间隔有效请求信号动作 588
图 20-25 位置模式-基本计数 590
图 20-26 位置计数模式-相位差计数（1 倍计数） 590
图 20-27 位置计数模式-相位差计数（2 倍计数） 590
图 20-28 位置计数模式-相位差计数（4 倍计数） 591
图 20-29 位置计数模式-方向计数 591
图 20-30 公转计数模式-Z 相计数 592
图 20-31 公转计数模式-位置溢出计数 592
图 20-32 公转计数模式-混合计数 593
图 20-33 公转计数模式-混合计数 Z 相屏蔽动作例 1 593
图 20-34 公转计数模式-混合计数 Z 相屏蔽动作例 2 594
图 20-35 锯齿波模式时中断&事件输出例 605
图 21-1 HRPWM 基本框图 654
图 21-2 HRPWM 调整波形图 655
图 22-1 Timer4 基本框图 661
图 22-2 Timer4 锯齿波波形 662
图 22-3 Timer4 三角波波形 662
图 22-4 Timer4 锯齿波模式计数动作 662
图 22-5 Timer4 三角波模式计数动作 663
图 22-6 锯齿波模式波形输出例 664
图 22-7 三角波模式波形输出例 664
图 22-8 缓存无效时修改锯齿波计数周期 665
图 22-9 缓存使能时修改锯齿波计数周期 666

图 22-10 缓存使能时修改三角波计数周期 666
图 22-11 OCCR 缓冲数据传输（周期间隔响应链接禁止时） 667
图 22-12 OCCR 缓冲数据传输（周期间隔响应链接使能） 668
图 22-13 输出比较缓冲数据传输（OCMR 缓冲使能） 669
图 22-14 SCCR 缓冲传输操作（周期间隔响应链接传输禁止时） 670
图 22-15 SCCR 缓冲传输操作（周期间隔响应链接传输使能时） 671
图 22-16 锯齿波独立 PWM 输出例 672
图 22-17 三角波独立 PWM 输出例 672
图 22-18 三角波扩展 PWM 输出 673
图 22-19 软件实现互补 PWM 输出 673
图 22-20 死区定时器模式下的互补 PWM 输出 674
图 22-21 脉宽异常时的死区定时器模式下的波形输出 675
图 22-22 死区定时器滤波模式下的互补 PWM 输出 676
图 22-23 周期间隔响应时序图 677
图 22-24 专用事件输出信号周期间隔响应输出 678
图 22-25 计数方向信号输出示例 680
图 22-26 延时启动模式下专用事件输出信号的输出时序 682
图 23-1 EMB 结构框图 710
图 24-1 TimerA 基本框图 732
图 24-2 锯齿波波形（递加计数） 733
图 24-3 三角波波形 733
图 24-4 比较输出动作 734
图 24-5 捕获输入动作 735
图 24-6 软件同步动作 736
图 24-7 时钟输入端口的滤波功能 736
图 24-8 锯齿波模式时缓存动作 738
图 24-9 32 位级联计数动作 739
图 24-10 单边对齐 PWM 输出例 739
图 24-11 双边对称 PWM 输出例 740
图 24-12 位置模式-基本计数 741
图 24-13 位置计数模式-相位差计数（1 倍计数） 741
图 24-14 位置计数模式-相位差计数（2 倍计数） 742
图 24-15 位置计数模式-相位差计数（4 倍计数） 742
图 24-16 位置计数模式-方向计数 742
图 24-17 公转计数模式-Z 相计数 743
图 24-18 公转计数模式-位置溢出计数 743
图 24-19 公转计数模式-混合计数 744
图 25-1 Timer2 基本框图 769
图 25-2 比较输出动作 771
图 25-3 硬件启动、清零动作 772
图 25-4 捕获输入动作 773
图 25-5 脉宽测量 773

图 25-6 周期测量 774
图 25-7 TRIG 输入端口的数字滤波 774
图 26-1 Timer0 基本框图 789
图 26-2 Timer0 计数时序图 791
图 27-1 RTC 的基本框图 801
图 28-1 硬件启动例 830
图 28-2 软件启动例 831
图 28-3 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） 832
图 28-4 计数器下溢动作例 834
图 28-5 计数器刷新动作例 835
图 29-1 USART 系统框图 843
图 29-2 UART 数据格式 845
图 29-3 UART 发送数据图例 1 847
图 29-4 UART 发送数据图例 2 847
图 29-5 UART 内部同步和采样时序 848
图 29-6 UART 接收数据图例 1 850
图 29-7 UART 接收数据图例 2 850
图 29-8 多处理器通信图例 854
图 29-9 多处理器模式数据格式 854
图 29-10 多处理器模式发送数据图例 856
图 29-11 多处理器模式接收数据图例 1 857
图 29-12 多处理器模式接收数据图例 2 858
图 29-13 LIN 总线数据行为 859
图 29-14 智能卡连接示意图 862
图 29-15 智能卡模式同步时序和采样时序图 863
图 29-16 智能卡模式数据格式 863
图 29-17 智能卡模式发送数据图例 865
图 29-18 智能卡模式接收数据图例 865
图 29-19 时钟同步模式数据格式 868
图 29-20 时钟同步模式发送数据图例 1 870
图 29-21 时钟同步模式发送数据图例 2 870
图 29-22 时钟同步模式接收数据图例 1 871
图 29-23 时钟同步模式接收数据图例 2 872
图 30-1 I²C 系统框图 895
图 30-2 I2C 总线的结构例 896
图 30-3 I²C 总线的时序图 897
图 30-4 I²C 总线的数据格式 898
图 30-5 7 位地址格式的主机发送数据时序图（例） 900
图 30-6 7 位地址格式的主机接收数据的时序图（例） 901
图 30-7 7 位地址格式的从机发送模式时序图（例） 902
图 30-8 7 位地址格式从机接收模式时序图（例） 903
图 30-9 SCL 同步时序 904

图 30-10 从机发送时序图（1） 905
图 30-11 从机发送时序图（2） 906
图 30-12 选择 7 位地址格式时的时序 908
图 30-13 选择 10 位地址格式时的时序 909
图 30-14 数字滤波电路框图 918
图 31-1 系统框图 942
图 31-2 主机模式结构 946
图 31-3 三线式时钟同步运行 947
图 31-4 数据格式 949
图 31-5 数据传送格式图（CPHA=0） 951
图 31-6 数据传送格式（CPHA=1） 952
图 31-7 全双工同步串行通信 953
图 31-8 只进行发送通信 954
图 31-9 奇偶校验流程 963
图 31-10 过载错误处理 965
图 31-11 启用时钟自动停止功能时的动作示意图（CPHA=1） 967
图 31-12 启用时钟自动停止功能时的动作示意图（CPHA=0） 967
图 31-13 奇偶校验错误 968
图 32-1 QSPI 的模块构成图 982
图 32-2 默认区域设定及 AHB 总线空间内存映射关系图 983
图 32-3 QSPI-ROM 空间内存映像图 984
图 32-4 扩展式 SPI 协议动作示意图 1（快速读模式） 985
图 32-5 扩展式 SPI 协议动作示意图 2（四线式输入输出快速读模式） 985
图 32-6 二线式 SPI 协议动作示意图（快速读模式） 986
图 32-7 四线式 SPI 协议动作示意图（快速读模式） 986
图 32-8 串行接口的基本时序图 987
图 32-9 基准时钟选择 HCLK 三分频时输出时钟占空比补正示意图 990
图 32-10 QSSL 建立时间配置示意图 991
图 32-11 QSSN 保持时间配置示意图 992
图 32-12 数据接收延迟示意图 993
图 32-13 标准读总线周期示意图 995
图 32-14 快速读总线周期示意图 996
图 32-15 选择 XIP 模式的快速读总线周期示意图 996
图 32-16 二线式输出快速读总线周期示意图 997
图 32-17 选择 XIP 模式的二线式输出快速读总线周期示意图 998
图 32-18 二线式输入输出快速读总线周期示意图 999
图 32-19 选择 XIP 模式的二线式输入输出快速读总线周期示意图 999
图 32-20 四线式输出快速读总线周期示意图 1000
图 32-21 选择 XIP 模式的四线式输出快速读总线周期示意图 1001
图 32-22 四线式输入输出快速读总线周期示意图 1002
图 32-23 选择 XIP 模式的四线式输入输出快速读总线周期示意图 1003
图 32-24 进入 4-Byte 模式指令总线周期示意图 1004

图 32-25 退出 4-Byte 模式指令总线周期示意图 1005
图 32-26 写许可指令总线周期示意图 1006
图 32-27 独立转换的单一闪存数据读取操作示意图 1007
图 32-28 预读取功能有效时数据读取操作示意图 1008
图 32-29 使用 QSPI 总线周期延长功能的数据读取操作示意图 1009
图 32-30 XIP 模式控制示意图 1010
图 33-1 I2S 系统框图 1028
图 33-2 I2S Philips 协议波形（16/32 位全精度） 1031
图 33-3 I2S Philips 协议波形（16 位数据封装在 32 位帧中） 1032
图 33-4 I2S Philips 协议波形（24 位数据封装在 32 位帧中） 1032
图 33-5 I2S MSB 协议波形（16/32 位全精度） 1032
图 33-6 I2S MSB 协议波形（16 位数据封装在 32 位帧中） 1033
图 33-7 I2S MSB 协议波形（24 位数据封装在 32 位帧中） 1033
图 33-8 I2S LSB 协议波形（16/32 位全精度） 1034
图 33-9 I2S LSB 协议波形（16 位数据封装在 32 位帧中） 1034
图 33-10 I2S LSB 协议波形（24 位数据封装在 32 位帧中） 1034
图 33-11 I2S PCM 协议波形（16/32 位全精度） 1035
图 33-12 I2S PCM 协议波形（16 位数据封装在 32 位帧中） 1035
图 33-13 I2S PCM 协议波形（24 位数据封装在 32 位帧中） 1036
图 33-14 音频采样频率定义 1037
图 33-15 时钟发生器架构 1037
图 33-16 主机只接收暂时停止接收 1044
图 33-17 PCM 短帧主机发送暂停后重新发送方式一 1045
图 33-18 PCM 短帧主机发送暂停后重新发送方式二 1045
图 34-1 USBHS 系统框图 1060
图 34-2 USBHS 主机模式系统构建图 1063
图 34-3 USBHS 设备模式系统构建图 1068
图 34-4 USBHS 动态更新 USBHS_HFIR 寄存器示意图 1075
图 34-5 USBHS 主机模式下 FIFO 架构示意图 1076
图 34-6 USBHS 设备模式下 FIFO 架构示意图 1078
图 34-7 USBHS 控制状态 寄存器存储器映射 1111
图 35-1 USBFS 系统框图 1211
图 35-2 USBFS 主机模式系统构建图 1214
图 35-3 USBFS 设备模式系统构建图 1219
图 35-4 USBFS 动态更新 USBFS_HFIR 寄存器示意图 1226
图 35-5 USBFS 主机模式下 FIFO 架构示意图 1227
图 35-6 USBFS 设备模式下 FIFO 架构示意图 1229
图 35-7 USBFS 控制状态寄存器存储器映射 1261
图 36-1 CANFD 系统框图 1349
图 36-2 CAN 位时间定义图 1352
图 36-3 CANFD TBUF 寄存器写发送缓冲器和示意图 1355
图 36-4 CAN RBUF 寄存器读接收缓冲器示意图 1356

图 36-5 CAN ACF 寄存器访问筛选器组示意图 1357
图 36-6 CANFD LBMI 和 LBME 示意图 1361
图 36-7 TDC 功能示意图 1367
图 37-1 CAN 系统框图 1399
图 37-2 CAN 位时间定义图 1401
图 37-3 CAN TBUF 寄存器写发送缓冲器和示意图 1403
图 37-4 CAN RBUF 寄存器读接收缓冲器示意图 1403
图 37-5 CAN ACF 寄存器访问筛选器组示意图 1404
图 37-6 CAN LBMI 和 LBME 示意图 1408
图 39-1 以太网 MAC 控制器架构图 1479
图 39-2 MII 接口连接图 1484
图 39-3 RMII 接口连接图 1486
图 39-4 SMI 接口连接图 1487
图 39-5 MAC 帧结构 1490
图 39-6 MII/RMII 发送位序 1497
图 39-7 无冲突发送图 1497
图 39-8 有冲突发送图 1498
图 39-9 MII/RMII 模式下的发送图 1498
图 39-10 MII/RMII 接收位序 1502
图 39-11 无错误发送图 1502
图 39-12 有错误发送图 1502
图 39-13 假载波指示下的接收图 1503
图 39-14 远程唤醒帧过滤寄存器 1508
图 39-15 时钟同步图 1511
图 39-16 系统时间校准 1514
图 39-17 描述符结构 1518
图 39-18 TxDMA 动作流程（非 OSF 模式） 1523
图 39-19 TxDMA 动作流程（OSF 模式） 1525
图 39-20 RxDMA 动作流程 1534
图 39-21 ETHMAC 中断方案 1542
图 39-22 DMA 中断组成 1543
图 40-1 EXMC 架构图 1613
图 40-2 外部空间地址分配 1615
图 40-3 SMC 的地址空间划分 1616
图 40-4 DMC 的地址空间划分 1617
图 40-5 NFC 的地址空间划分 1618
图 40-6 SMC 状态图 1623
图 40-7 SMC 初始设定流程 1624
图 40-8 单次读动作基本时序（异步方式（RSYN=0）&16 位位宽（MW=01）） 1626
图 40-9 单次读动作基本时序（异步方式（RSYN=0）&32 位位宽（MW=10）） 1627
图 40-10 单次读动作基本时序（同步方式（RSYN=1）&16 位位宽（MW=01）） 1628
图 40-11 单次读动作基本时序（同步方式（RSYN=1）&32 位位宽（MW=10）） 1629

图 40-12 地址数据线复用单次读动作基本时序（异步方式（RSYN=0）&32 位位宽（MW=10））.....1630
图 40-13 单次写动作基本时序（异步方式（WSYN=0）&16 位位宽（MW=01）&BLS=0）.....1631
图 40-14 单次写动作基本时序（异步方式（WSYN=0）&32 位位宽（MW=10）&BLS=1）.....1631
图 40-15 单次写动作基本时序（同步方式（WSYN=1）&16 位位宽（MW=01）&BLS=1）.....1632
图 40-16 单次写动作基本时序（同步方式（WSYN=1）&32 位位宽（MW=10）&BLS=0）.....1632
图 40-17 地址数据线复用单次写动作基本时序（异步方式（WSYN=0）&16 位位宽（MW=01））.....1633
图 40-18 突发读动作基本时序（异步方式（RSYN=0）&32 位位宽（MW=10）&RBL=010）.....1633
图 40-19 突发写动作基本时序（异步方式（WSYN=0）&32 位位宽（MW=10）&WBL=011）.....1634
图 40-20 行激活到读写操作时序.....1636
图 40-21 命令方式预充电.....1636
图 40-22 读写后自动预充电.....1637
图 40-23 自刷新动作的进入和退出时序.....1637
图 40-24 DMC 状态图.....1640
图 40-25 DMC 初始设定流程.....1641
图 40-26 单次读动作基本时序（16 位位宽（DMCMW=00））.....1642
图 40-27 单次写动作基本时序（32 位位宽（DMCMW=01））.....1643
图 40-28 突发读动作基本时序（32 位位宽（DMCMW=10）&BURST=010）.....1644
图 40-29 突发写动作基本时序（32 位位宽（DMCMW=10）&BURST=011）.....1644
图 40-30 DMC_CKE 端口输出控制时序.....1645
图 40-31 NFC 时序控制图.....1686
图 41-1 DVP 基本框图.....1694
图 41-2 DVP 信号波形示意图.....1697
图 41-3 单帧模式数据采集动作.....1699
图 41-4 连续模式数据采集动作.....1699
图 41-5 帧采集频率控制.....1700
图 41-6 DVP 窗口设定图.....1702
图 41-7 窗口裁剪数据采集图.....1702
图 42-1 AES 加解密流程示意图.....1713
图 42-2 AES 加密流程图.....1714
图 42-3 HASH 算法流程图.....1725
图 42-4 TRNG 系统框图.....1741
图 43-1 CRC 应用示意图.....1744
图 45-1 正弦运算角度值示意图.....1778
图 46-1 FIR 基本框图.....1783
图 47-1 调试控制系统.....1791
图 47-2 调试控制系统.....1792
图 47-3 JTAG-DP 到 SW-DP 切换时序.....1793
图 47-4 TPIU 框图.....1804

简介（Overview）

HC32F4A0 系列是基于 ARM® Cortex®-M4 32-bit RISC CPU，最高工作频率 240MHz 的高性能 MCU。Cortex-M4 内核集成了浮点运算单元（FPU）和 DSP，实现单精度浮点算术运算，支持所有 ARM 单精度数据处理指令和数据类型，支持完整 DSP 指令集。内核集成了 MPU 单元，同时叠加 DMAC 专用 MPU 单元，保障系统运行的安全性。

HC32F4A0 系列集成了高速片上存储器，包括最大 2MB 的 Flash，最大 512KB 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，ETHMAC 专用 DMA。除总线矩阵外，支持外设间数据传递，基本算术运算和事件相互触发，可以显著降低 CPU 的事务处理负荷。

HC32F4A0 系列集成了丰富的外设功能。包括 3 个独立的 12bit 2.5MSPS ADC，4 个增益可调 PGA，4 个 12 位 15MSPS 的 DAC，4 个高速电压比较器（CMP），8 个多功能 PWM Timer（Timer6），支持 16 路互补 PWM 输出，16 个高精度 PWM（HRPWM）扩展了 Timer6 的 PWM 信号的分辨率，3 个电机 PWM Timer（Timer4）支持 18 路互补 PWM 输出，12 个 16bit 通用 Timer（TimerA）支持 6 路 3 相正交编码输入及 48 路 Duty 独立可设 PWM 输出，22 个串行通信接口（I2C/UART/SPI），1 个 QSPI 接口，2 路 CAN，4 个 I2S 支持音频 PLL，2 个 SDIO，1 个 ETHMAC，内置 USBFS PHY 的 USBFS Controller 和 USBHS Controller，1 个外部扩张总线控制器，包括 NFC 控制器，SMC 控制器和 DMC 控制器，1 个数字视频接口 DVP，1 个数学运算单元（MAU）和 4 个滤波数学加速器（FMAC）。

HC32F4A0 系列支持宽电压范围（1.8-3.6V），宽温度范围（-40-105℃）和各种低功耗模式。支持低功耗模式的快速唤醒，STOP 模式唤醒最快至 2us，Power Down 模式唤醒最快至 25us。

典型应用

HC32F4A0 系列提供 100pin、144pin、176pin 的 LQFP 封装，176pin 的 VFBGA 封装，208pin 的 TFBGA 封装，用于高性能变频控制、数字电源，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

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

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		0xE0100000	0xFFFFFFFF	511MB	Reserved		自定义空间
	私有外设 外部总线	0xE00FF000 0xE0042400 0xE0042000 0xE0041000 0xE0040000	0xE00FFFFF 0xE00FEFFF 0xE00423FF 0xE0041FFF 0xE0040FFF	4KB 755KB 1KB 4KB 4KB	ROMTABLE DBGC ETM TPIU		调试控制寄存器区域
	私有外设 内部总线	0xE000F000 0xE000E000 0xE0003000 0xE0002000 0xE0001000 0xE0000000	0xE003FFFF 0xE000EFFF 0xE000DFFF 0xE0002FFF 0xE0001FFF 0xE0000FFF	196KB 4KB 44KB 4KB 4KB 4KB	 SCS FPB DWT ITM		系统控制空间 NVIC/MPU 等
外部设备	-	0xA0000000	0xDFFFFFFF	1024MB	Reserved		
外部存储	AHB5 时钟： HCLK	0x98000000 0x88200000	0x9FFFFFFF 0x97FFFFFF	128MB 254MB	QSPI Reserved		
	AHB5	0x88100000 0x88000800	0x881FFFFF 0x880FFFFF	1MB 1022KB	NFC 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
	时钟： BCLK	0x88000400	0x880007FF	1KB	DMCR		
		0x88000000	0x880003FF	1KB	SMCR		
		0x80000000	0x87FFFFFF	128MB	DMC		
		0x60000000	0x7FFFFFFF	512MB	SMC		
外设	-	0x44000000	0x5FFFFFFF	448MB	Reserved		
		0x42000000	0x43FFFFFF	32MB	PeriBitBand		
		0x40100000	0x41FFFFFF	31MB	Reserved		
	AHB4 时钟： PCLK1	0x400C0000	0x400FFFFF	256KB	USBHS		
	AHB2 时钟： PCLK1	0x40080000	0x400BFFFF	256KB	USBFS		
		0x40078800	0x4007FFFF	30KB	BLANK		
		0x40078400	0x400787FF	1KB	SDIOC_2		
		0x40078000	0x400703FF	1KB	CAN_2		
	AHB3 时钟： PCLK1	0x40070400	0x40077FFF	31KB	BLANK		
		0x40070000	0x400703FF	1KB	SDIOC_1		
		0x40060000	0x4006FFFF	64KB	ETHMAC		
	AHB1 时钟： HCLK	0x40059000	0x4005FFFF	28KB	BLANK		
		0x40058C00	0x40058FFF	1KB	FMAC_4		
		0x40058800	0x40058BFF	1KB	FMAC_3		
		0x40058400	0x400587FF	1KB	FMAC_2		
		0x40058000	0x400583FF	1KB	FMAC_1		
		0x40057C00	0x40057FFF	1KB	DCU_8		
		0x40057800	0x40057BFF	1KB	DCU_7		
		0x40057400	0x400577FF	1KB	DCU_6		
		0x40057000	0x400573FF	1KB	DCU_5		
		0x40056C00	0x40056FFF	1KB	DCU_4		

存储器分类		开始地址	结束地址	空间大小	模块*3	保护*4	说明
		0x40056800	0x40056BFF	1KB	DCU_3		
		0x40056400	0x400567FF	1KB	DCU_2		
		0x40056000	0x400563FF	1KB	DCU_1		
		0x40055800	0x40055BFF	1KB	BLANK		
		0x40055800	0x40055BFF	1KB	DVP		
		0x40055400	0x400557FF	1KB	PERIC		周边模块控制寄存器
		0x40055000	0x400553FF	1KB	MAU		
		0x40054400	0x40054FFF	3KB	BLANK		禁止访问
		0x40054000	0x400543FF	1KB	CMU	带保护	
		0x40053800	0x40053FFF	2KB	GPIO		
		0x40053400	0x400537FF	1KB	DMA_2		
		0x40053000	0x400533FF	1KB	DMA_1		
		0x40052000	0x40052FFF	4KB	BLANK		
		0x40051000	0x40051FFF	4KB	INTC	带保护	
		0x40050C00	0x40050FFF	1KB	KEYSCAN		
		0x40050800	0x40050BFF	1KB	RAMIF	带保护	
		0x40050400	0x400507FF	1KB	BLANK		
		0x40050000	0x400503FF	1KB	DMPU	带保护	
	APB4 时钟： PCLK3	0x4004F800	0x4004FFFF	2KB	BLANK		
		0x4004F400	0x4004F7FF	1KB	I2C_6		
		0x4004F000	0x4004F3FF	1KB	I2C_5		
		0x4004EC00	0x4004EFFF	1KB	I2C_4		
		0x4004E800	0x4004EBFF	1KB	I2C_3		
		0x4004E400	0x4004E7FF	1KB	I2C_2		
		0x4004E000	0x4004E3FF	1KB	I2C_1		
		0x4004D000	0x4004DFFF	1KB	BLANK		
		0x4004CC00	0x4004CFFF	1KB	EMU	带保护	

存储器分类		开始地址	结束地址	空间大小	模块*3	保护*4	说明
		0x4004C800	0x4004CBFF	1KB	BLANK		
		0x4004C400	0x4004C7FF	1KB	VBAT	带保护	
		0x4004C000	0x4004C3FF	1KB	RTC	带保护	
		0x4004AC00	0x4004BFFF	5KB	BLANK		
		0x4004A800	0x4004ABFF	1KB	OTS		
		0x4004A400	0x4004A7FF	1KB	CMP_2		
		0x4004A000	0x4004A3FF	1KB	CMP_1		
		0x40049C00	0x40049FFF	1KB	CTC		
		0x40049800	0x40049BFF	1KB	BLANK		
		0x40049400	0x400497FF	1KB	SWDT	带保护	
		0x40049000	0x400493FF	1KB	WDT	带保护	
		0x400488000	0x40048FFF	2KB	BLANK		
		0x40048400	0x400487FF	1KB	FCM		
		0x40048000	0x400483FF	1KB	FCG	带保护	
外设	APB3	0x40042400	0x40047FFF	23KB	BLANK		
	时钟：	0x40042000	0x400423FF	1KB	TRNG	带保护	
	PCLK4	0x40041800	0x40041FFF	2KB	BLANK		
		0x40041400	0x400417FF	1KB	DAC_2		
		0x40041000	0x400413FF	1KB	DAC_1		
		0x40040C00	0x40040FFF	1KB	BLANK		
		0x40040800	0x40040BFF	1KB	ADC_3		
		0x40040400	0x400407FF	1KB	ADC_2		
		0x40040000	0x400403FF	1KB	ADC_1		
	APB5	0x4003C400	0x4003FFFF	15KB	BLANK		
	时钟：	0x4003C000	0x4003C3FF	1KB	HRPWM		
	PCLK0	0x4003B000	0x4003BFFF	4KB	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
		0x4003AC00	0x4003AFFF	1KB	TimerA_4		
		0x4003A800	0x4003ABFF	1KB	TimerA_3		
		0x4003A400	0x4003A7FF	1KB	TimerA_2		
		0x4003A000	0x4003A3FF	1KB	TimerA_1		
		0x40038C00	0x40039FFF	1KB	BLANK		
		0x40038800	0x40038BFF	1KB	Timer4_3		
		0x40038400	0x400387FF	1KB	Timer4_2		
		0x40038000	0x400383FF	1KB	Timer4_1		
	APB2	0x40028000	0x40037FFF	64KB	BLANK		
	时钟：	0x40027C00	0x40027FFF	1KB	TimerA_12		
	PCLK1	0x40027800	0x40027BFF	1KB	TimerA_11		
		0x40027400	0x400277FF	1KB	TimerA_10		
		0x40027000	0x400273FF	1KB	TimerA_9		
		0x40026C00	0x40026FFF	1KB	TimerA_8		
		0x40026800	0x40026BFF	1KB	TimerA_7		
		0x40026400	0x400267FF	1KB	TimerA_6		
		0x40026000	0x400263FF	1KB	TimerA_5		
		0x40026800	0x40026FFF	2KB	BLANK		
		0x40026400	0x400267FF	1KB	Timer2_4		
		0x40025000	0x400253FF	1KB	Timer2_3		
		0x40024C00	0x40024FFF	1KB	Timer2_2		
		0x40024800	0x40024BFF	1KB	Timer2_1		
		0x40024400	0x400247FF	1KB	Timer0_2		
		0x40024000	0x400243FF	1KB	Timer0_1		
		0x40022800	0x40023FFF	6KB	BLANK		
		0x40022400	0x400227FF	1KB	I2S_4		
		0x40022000	0x400223FF	1KB	I2S_3		

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		0x40021C00	0x40021FFF	1KB	USART_10		
		0x40021800	0x40021BFF	1KB	USART_9		
		0x40021400	0x400217FF	1KB	USART_8		
		0x40021000	0x400213FF	1KB	USART_7		
		0x40020C00	0x40020FFF	1KB	USART_6		
		0x40020800	0x40020BFF	1KB	SPI_6		
		0x40020400	0x400207FF	1KB	SPI_5		
		0x40020000	0x400203FF	1KB	SPI_4		
	APB1	0x4001E800	0x4001FFFF	6KB	BLANK		
	时钟： PCLK1	0x4001E400	0x4001E7FF	1KB	I2S_2		
		0x4001E000	0x4001E3FF	1KB	I2S_1		
		0x4001DC00	0x4001DFFF	1KB	USART_5		
		0x4001D800	0x4001DBFF	1KB	USART_4		
		0x4001D400	0x4001D7FF	1KB	USART_3		
		0x4001D000	0x4001D3FF	1KB	USART_2		
		0x4001CC00	0x4001CFFF	1KB	USART_1		
		0x4001C800	0x4001CBFF	1KB	SPI_3		
		0x4001C400	0x4001C7FF	1KB	SPI_2		
		0x4001C000	0x4001C3FF	1KB	SPI_1		
		0x40018000	0x4001BFFF	16KB	Timer6		计数时钟:PCLK0
		0x40017C00	0x40017FFF	1KB	EMB		
		0x40010C00	0x40017BFF	28KB	BLANK		
		0x40010800	0x40010BFF	1KB	-		内部触发事件寄存器区域
		0x40010400	0x400107FF	1KB	EFM	带保护	
		0x40010000	0x400103FF	1KB	BLANK		
	AHB4	0x40009400	0x4000FFFF	27KB	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
	时钟： PCLK1	0x40009000	0x400093FF	1KB	CAN_1		
		0x40008C00	0x40008FFF	1KB	CRC	带保护	
		0x40008800	0x40008BFF	1KB	BLANK		
		0x40008400	0x400087FF	1KB	HASH	带保护	
		0x40008000	0x400083FF	1KB	AES	带保护	
		0x40000000	0x40007FFF	32KB	Reserved		
SRAM	SRAM 时钟： HCLK	0x24000000	0x3FFFFFFF	448MB	Reserved		
		0x22000000	0x23FFFFFF	32MB	SRAMBitBand		
		0x20100000	0x21FFFFFF	31MB	Reserved		
		0x200F1000	0x200FFFFF	60KB	BLANK		
		0x200F0000	0x200F0FFF	4KB	SRAMB		ECC RAM
		0x20060000	0x200EFFFF	576KB	BLANK		
		0x20058000	0x2005FFFF	32KB	SRAM4		ECC RAM
		0x20040000	0x20057FFF	96KB	SRAM3		
		0x20020000	0x2003FFFF	128KB	SRAM2		
		0x20000000	0x2001FFFF	128KB	SRAM1		
CODE	SRAM 时钟： HCLK	0x1FFE0000	0x1FFFFFFF	128KB	SRAMH		
		0x03004000	0x1FFDFFFF	463.86MB	BLANK		
	OTP 时钟： HCLK	0x03002004	0x03003FFF	8187B	BLANK		
		0x03002000	0x03002003	4 B	引导交换		仅 CPU 可以访问
		0x03001ADC	0x03001FFF	1316B	BLANK		
		0x03000000	0x03001ADB	6876B	OTP		仅 CPU 可以访问
	-	0x02100000	0x02FFFFFF	15M	BLANK		
	REMAP 时钟： HCLK	0x02080000	0x020FFFFF	512KB	REMAP1		地址重映射区域 1
		0x02000000	0x0207FFFF	512KB	REMAP0		地址重映射区域 0

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
	-	0x00200000	0x01FFFFFF	30M	BLANK		
	Flash 时 钟 ： HCLK	0x00100000 0x00000000	0x001FFFFF 0x000FFFFF	1MB 1MB	Embedded Flash 1 Embedded Flash 0		

表 1-1 存储器映射

*1 请参考 ARM Cortex-M4 说明手册“存储器系统”。

*2 总线说明请参考总线章节。

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

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

1.2 外部空间映射

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

QSPI	0x98000000	0x9FFFFFFF	128MB

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

在该 MCU 中，外设寄存器和 SRAM 均映射到一个位段区域，这样可实现单个位段的读写操作。这些操作仅适用于 Cortex™-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 提供了个存储器地址重映射功能,可以使用 MMF_REMCR0 和 MMF_REMCR1 来配置重映射目标地址。本 MCU 提供 2 个重映射地址供自由配置,目标地址可以是主闪存地址也可以是高速 SRAM 地址。

重映射地址 0：

0x0200_0000H~0x0208_0000H(视设定重映射大小而定 MMF_REMCR0.RM0SIZE[4:0])

重映射地址 1：

0x0208_0000H~0x0210_0000H(视设定重映射大小而定 MMF_REMCR1.RM1SIZE[4:0])

当使用重映射功能有效(MMF_REMCR0.EN0=1 或 MMF_REMCR1.EN1=1)时，地址对应表如下：

	重映射地址 （CPU地址—CPUADDR[31:0]）	重映射目标地址（主闪存地址）		
		高12位	中6位	低14位
RM0SIZE[4:0]=01110情形 （重映射空间：16K）	0x0200_0000h~0x0200_3FFFh	全0	RM0TADDR[5:0]	CPUADDR[13:0]
RM0SIZE[4:0]=01111情形 （重映射空间：32K）	0x0200_0000h~0x0200_7FFFh	全0	RM0TADDR[4:0] CPUADDR[14]	CPUADDR[13:0]
RM1SIZE[3:0]=10000情形 （重映射空间：64K）	0x0208_0000h~0x0208_7FFFh	全0	RM1TADDR[3:0] CPUADDR[15:14]	CPUADDR[13:0]
RM1SIZE[4:0]=10001情形 （重映射空间：128K）	0x0208_0000h~0x0208_FFFFh	全0	RM1TADDR[2:0] CPUADDR[16:14]	CPUADDR[13:0]

表 1-3 目标地址配置主闪存示例(MMF_REMCR0.EN0=1 或 MMF_REMCR1.EN1=1)

	重映射地址 （CPU地址—CPUADDR[31:0]）	重映射目标地址（主闪存地址）		
		高3位	中17位	低12位
RM0SIZE[4:0]=01100情形 （重映射空间：4K）	0x0200_0000h~0x0200_0FFFh	全0	RM0TADDR[16:0]	CPUADDR[11:0]
RM0SIZE[4:0]=01101情形 （重映射空间：8K）	0x0200_0000h~0x0200_1FFFh	全0	RM0TADDR[15:0] CPUADDR[12]	CPUADDR[11:0]
RM1SIZE[3:0]= 01100情形 （重映射空间：4K）	0x0208_0000h~0x0208_0FFFh	全0	RM1TADDR[16:0]	CPUADDR[11:0]
RM1SIZE[4:0]= 01101情形 （重映射空间：8K）	0x0208_0000h~0x0208_1FFFh	全0	RM1TADDR[15:0] CPUADDR[12]	CPUADDR[11:0]

表 1-4 目标地址配置高速 SRAM 示例(MMF_REMCR0.EN0=1 或 MMF_REMCR1.EN1=1)

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：

寄存器基地址：0x40010500

寄存器名	符号	偏移地址	位宽	复位值
访问保护寄存器	MMF_REMPRT	0000h	32	00000000h
重映射寄存器0	MMF_REMCR0	0004h	32	00000000h
重映射寄存器1	MMF_REMCR1	0008h	32	00000000h

表 1-5 寄存器列表

1.5.1 访问保护寄存器（MMF_REMPRT）

复位值： 0x00000000h

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

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

位	标记	位名	功能	读写
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0123H接着写入3210H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 重映射寄存器 0（MMF_REMCR0）

复位值：0x0000_0000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EN0	-		RM0TADDR[16:4]												

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RM0TADDR[3:0]				-							RM0SIZE[4:0]				

位	标记	位名	功能	读写
b31	EN0	重映射0有效位	0：重映射0无效 1：重映射0有效	R/W
b30~29	Reserved	-	读出时为“0”,写入时写“0”	R
b28~12	RM0TADDR[16:0]	重映射目标地址	有效位数和RM0SIZE[4:0]设定有关。 设定可参考表格3.1地址重映射功能有效时示例	R/W
b11~b5	Reserved	-	读出时为“0”,写入时写“0”	R
b4~b0	RM0SIZE[4:0]	重映射空间大小	设定重映射 0 区域的大小 00000~01011：保留，设定禁止 01100：4KByte 01101：8KByte 01110：16KByte 01111：32KByte 10000：64KByte 10001：128KByte 10010：256KByte 10011：512KByte 10100～11111：保留，设定禁止	R/W

1.5.3 重映射寄存器 1（MMF_REMCR1）

复位值：0x0000_0000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EN1	-														
							RM1TADDR[16:4]								

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RM1TADDR[3:0]				-							RM1SIZE[4:0]				

位	标记	位名	功能	读写
b31	EN1	重映射1有效位	0：重映射1无效 1：重映射1有效	R/W
b30~29	Reserved	-	读出时为“0”,写入时写“0”	R
b28~12	RM1TADDR[16:0]	重映射目标地址	有效位数和RM1SIZE[4:0]设定有关。 设定可参考表格3.1地址重映射功能有效时示例	R/W
b11~b5	Reserved	-	读出时为“0”,写入时写“0”	R
b4~b0	RM1SIZE[4:0]	重映射空间大小	设定重映射 1 区域的大小 00000~01011：保留，设定禁止 01100：4KByte 01101：8KByte 01110：16KByte 01111：32KByte 10000：64KByte 10001：128KByte 10010：256KByte 10011：512KByte 10100～11111：保留，设定禁止	R/W

2 总线架构（BUS）

2.1 概述

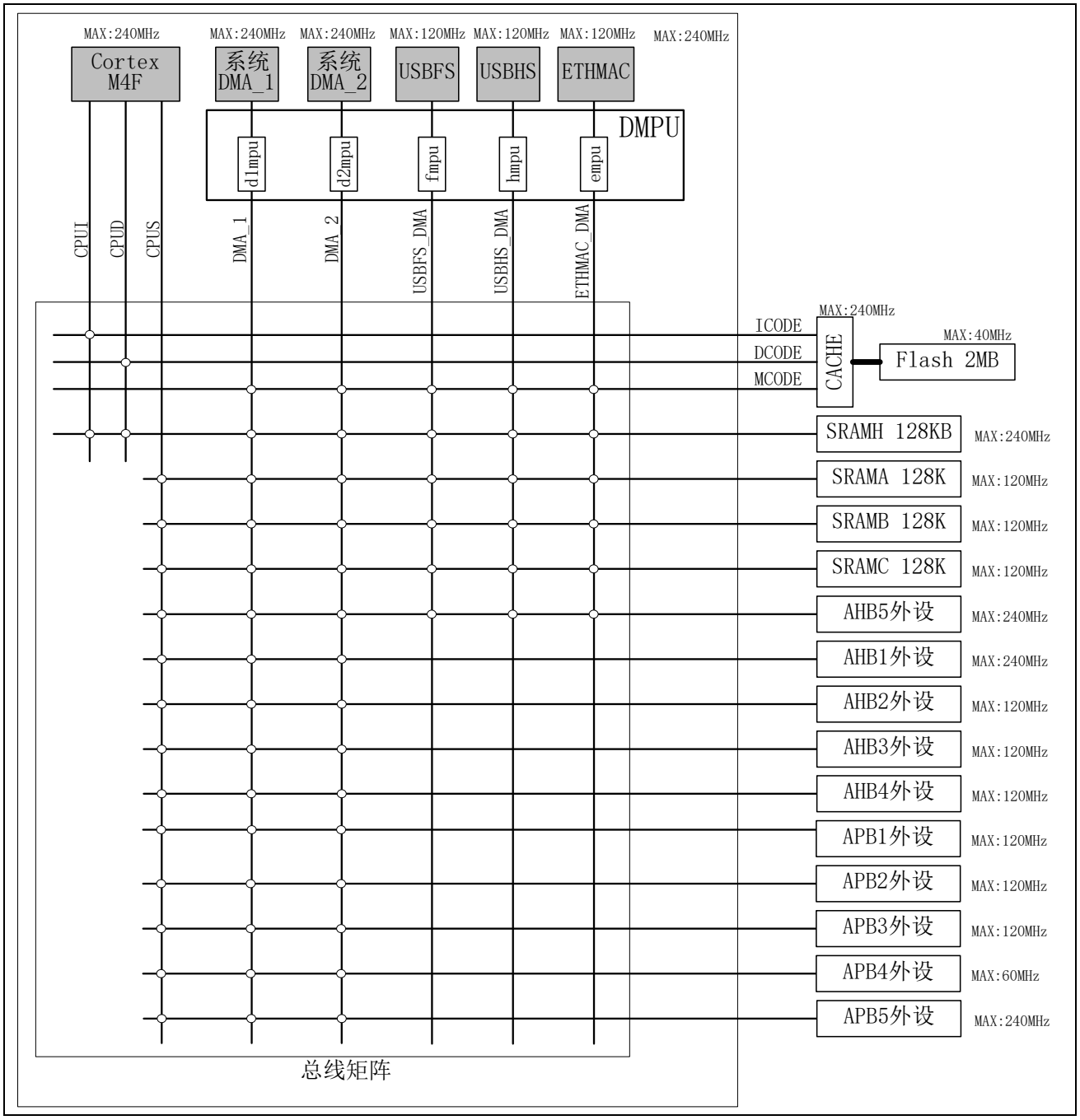
主系统由 32 位多层 AHB 总线矩阵构成，可实现以下主机总线和从机总线的互连：

- 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
  - USBHS_DMA 总线
  - ETHMAC_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 128kB)总线
  - 系统 SRAMA(SRAM1 128KB)总线
  - 系统 SRAMB(SRAM2 128KB)总线
  - 系统 SRAMC(SRAM3 96KB,SRAM4 32KB,SRAMB 4KB)
  - APB1 外设总线(EMB/Timers/SPI/USART/I2S/HRPWM/EFM)
  - APB2 外设总线(Timers/SPI/USART/I2S)
  - APB3 外设总线(ADC/DAC/TRNG)
  - APB4 外设总线(FCM/WDT/SWDT/CMP/EMU/CTC/OTS/RTC/VBAT/WKTM/I2C)
  - APB5 外设总线(Timers/HRPWM)
  - AHB1 外设总线
    (DMPU/KEYSCAN/INTC/DCU/GPIO/DMA/CMU/DVP/MAU/FMAC)
  - AHB2 外设总线(CAN/ SDIOC/USBFS)
  - AHB3 外设总线(SDIOC/ETHMAC)
  - AHB4 外设总线(AES/HASH/CRC/CAN/USBHS)

– AHB5 外设总线(SMC/DMC/SMCR/DMCR/NFC/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


MAX:240MHz
MAX:240MHz
MAX:240MHz
MAX:120MHz
MAX:120MHz
MAX:120MHz
MAX:240MHz
Cortex M4F
系统 DMA_1
系统 DMA_2
USBFS
USBHS
ETHMAC
DMPU
d1mpu
d2mpu
fmpu
hmpu
empu
CPUI
CPUD
CPUS
DMA_1
DMA_2
USBFS_DMA
USBHS_DMA
ETHMAC_DMA
ICODE
DCODE
MCODE
MAX:240MHz
CACHE
MAX:40MHz
Flash 2MB
SRAMH 128KB
MAX:240MHz
SRAMA 128K
MAX:120MHz
SRAMB 128K
MAX:120MHz
SRAMC 128K
MAX:120MHz
AHB5外设
MAX:240MHz
AHB1外设
MAX:240MHz
AHB2外设
MAX:120MHz
AHB3外设
MAX:120MHz
AHB4外设
MAX:120MHz
APB1外设
MAX:120MHz
APB2外设
MAX:120MHz
APB3外设
MAX:120MHz
APB4外设
MAX:60MHz
APB5外设
MAX:240MHz
总线矩阵

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 SRAM1/SRAM2/SRAM3/SRAM4/SRAMB/所有外设以及外部扩展空间。

- DMA_1 总线，DMA_2 总线

  系统 DMA_1/系统 DMA_2 专用总线，DMA_1/DMA_2 通过此总线访问数据存储器和外设，访问对象为 Flash/SRAMH/SRAM1/SRAM2/SRAM3/SRAM4/SRAMB/所有外设以及外部扩展空间。

- USBFS-DMA 总线

  USBFS 的 DMA 专用总线，USBFS 通过此总线访问所有存储器空间。访问对象为 Flash/ SRAMH/SRAM1/SRAM2/SRAM3/SRAM4/SRAMB 以及外部扩展空间。

- USBHS-DMA 总线

  USBHS 的 DMA 专用总线，USBHS 通过此总线访问所有存储器空间。访问对象为 Flash/ SRAMH/SRAM1/SRAM2/SRAM3/SRAM4/SRAMB 以及外部扩展空间。

- ETHMAC-DMA 总线

  ETHMAC 的 DMA 专用总线，ETH 通过此总线访问所有存储器空间。访问对象为 Flash/ SRAMH/SRAM1/SRAM2/SRAM3/SRAM4/SRAMB 以及外部扩展空间。

2.3 总线功能

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

通过总线矩阵，不同主机总线的访问目标不冲突时，各访问能够同时进行。例如 CPU-I 访问 Flash，CPU-D 访问 SRAMH，CPUS 访问 APB 外设，DMA_1 访问 SRAM1，DMA_2 访问 SRAM2，USBFS-DMA 访问 AHB5 的外部扩展空间，这些访问可以同时进行。

3 复位控制（RMU）

3.1 简介

芯片配置了 15 种复位方式。

- 上电复位（POR）
- NRST 引脚复位（NRST）
- 欠压复位（BOR）
- 可编程电压检测 1 复位（PVD1R）
- 可编程电压检测 2 复位（PVD2R）
- 看门狗复位（WDTR）
- 专用看门狗复位（SWDTR）
- 掉电唤醒复位（PDRST）
- 软件复位（SRST）
- MPU 错误复位（MPUR）
- RAM 奇偶校验复位（RAMPR）
- RAMECC 复位（RAMECCR）
- 时钟异常复位（CKFER）
- 外部高速振荡器异常停振复位（XTALER）
- CortexM4+LockUP 复位（LKUPR）

3.2 复位方式和复位标志位

复位方式及产生条件如表 3-1 所示。

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
CortexM4+LockUP 复位(LKUPR)	当 Cortex-M4+ 遇到严重的异常时，它会将自己的 PC 指针停在当前地址处，并锁死自己，在几个时钟周期延时之后复位整个芯片

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	M4+ Lockup 复位
上电复位标志（RMU_RSTF0.PORF）	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—
引脚复位标志（RMU_RSTF0.PINRF）	—	√	—	—	—	—	—	—	—	—	—	—	—	—	—
欠压复位标志（RMU_RSTF0.BORF）	—	—	√	—	—	—	—	—	—	—	—	—	—	—	—
可编程电压检测 1 复位标志（RMU_RSTF0.PVD1RF）	—	—	—	√	—	—	—	—	—	—	—	—	—	—	—
可编程电压检测 2 复位标志（RMU_RSTF0.PVD2RF）	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—
看门狗复位标志（RMU_RSTF0.WDRF）	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—
专用看门狗复位标（RMU_RSTF0.SWDRF）	—	—	—	—	—	—	√	—	—	—	—	—	—	—	—
掉电唤醒复位标志（RMU_RSTF0. PDRF）	—	—	—	—	—	—	—	√	—	—	—	—	—	—	—
软件复位标志（RMU_RSTF0.SWRF）	—	—	—	—	—	—	—	—	√	—	—	—	—	—	—
MPU 错误复位(RMU_RSTF0.MPUERF)	—	—	—	—	—	—	—	—	—	√	—	—	—	—	—
RAM 奇偶错误复位(RMU_RSTF0.RAPERF)	—	—	—	—	—	—	—	—	—	—	√	—	—	—	—
RAM ECC 复位(RMU_RSTF0.RAECRF)	—	—	—	—	—	—	—	—	—	—	—	√	—	—	—
时钟频率异常复位（RMU_RSTF0.CKFERF）	—	—	—	—	—	—	—	—	—	—	—	—	√	—	—
外部高速振荡器异常停振复位（RMU_RSTF0.XTALERF）	—	—	—	—	—	—	—	—	—	—	—	—	—	√	—
M4+ Lockup 复位（RSTF0.CPULKUPRF）	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√

√：置位 —：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 VPOR 后经过一定的时间（TPOR）之后芯片内部复位解除，CPU 开始执行代码。产生上电复位时，上电复位标志 RMU_RSTF0.PORF 被置位。上电复位的详细说明请参照【上电复位/掉电复位动作说明】。

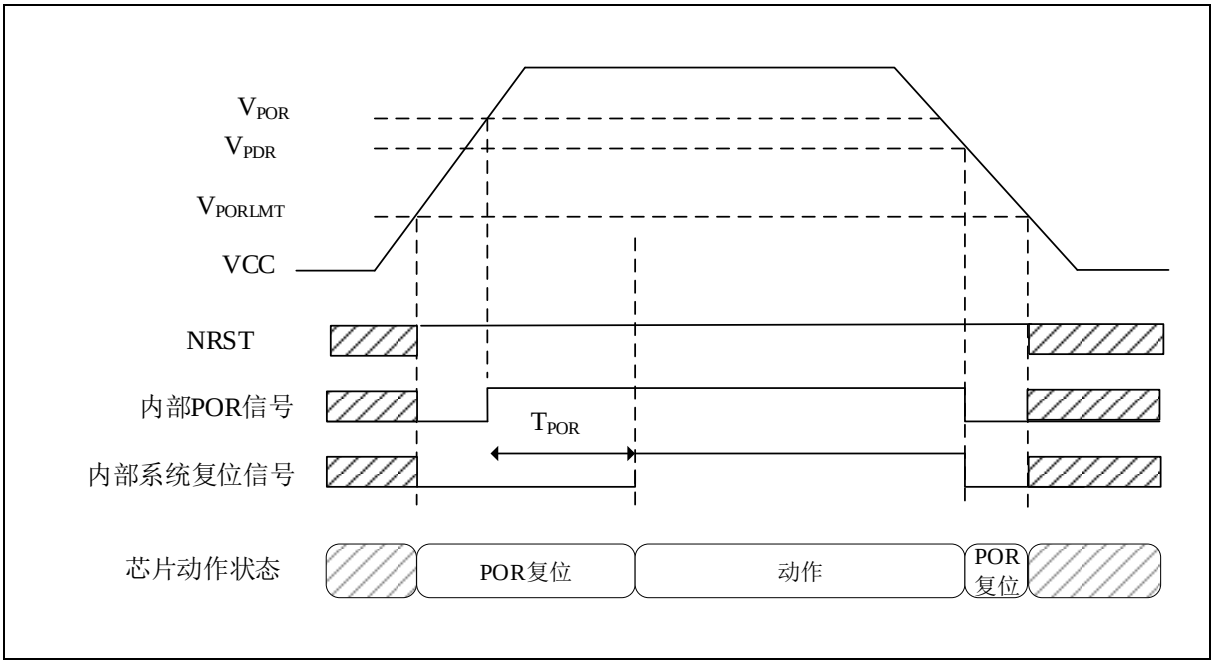


图 3-1 上电复位

3.3.2 NRST 引脚复位

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

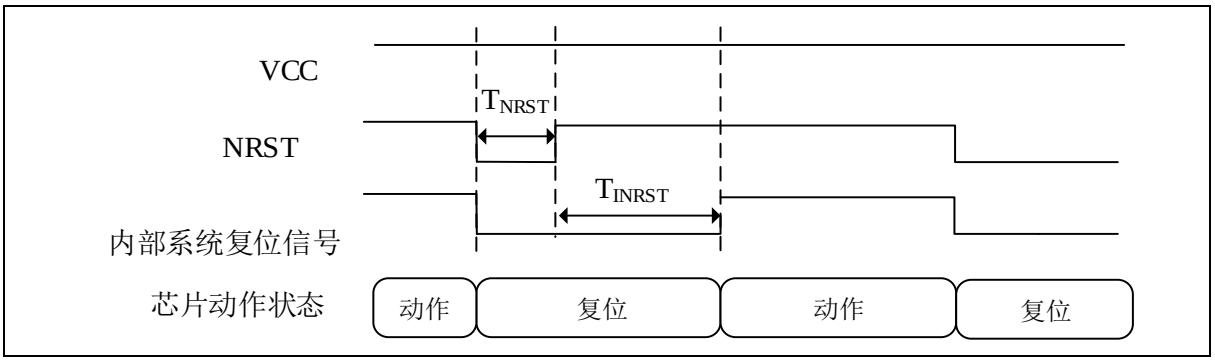


图 3-2 NRST 复位时序

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 VBOR，RMU_RSTF0.VBORF 被置位。当 VCC 电压高于监测电压 VBOR 经过 VBOR 的复位时间(TIBOR)后解除复位。

欠压的复位设定请参考【欠压复位(BOR)说明】。

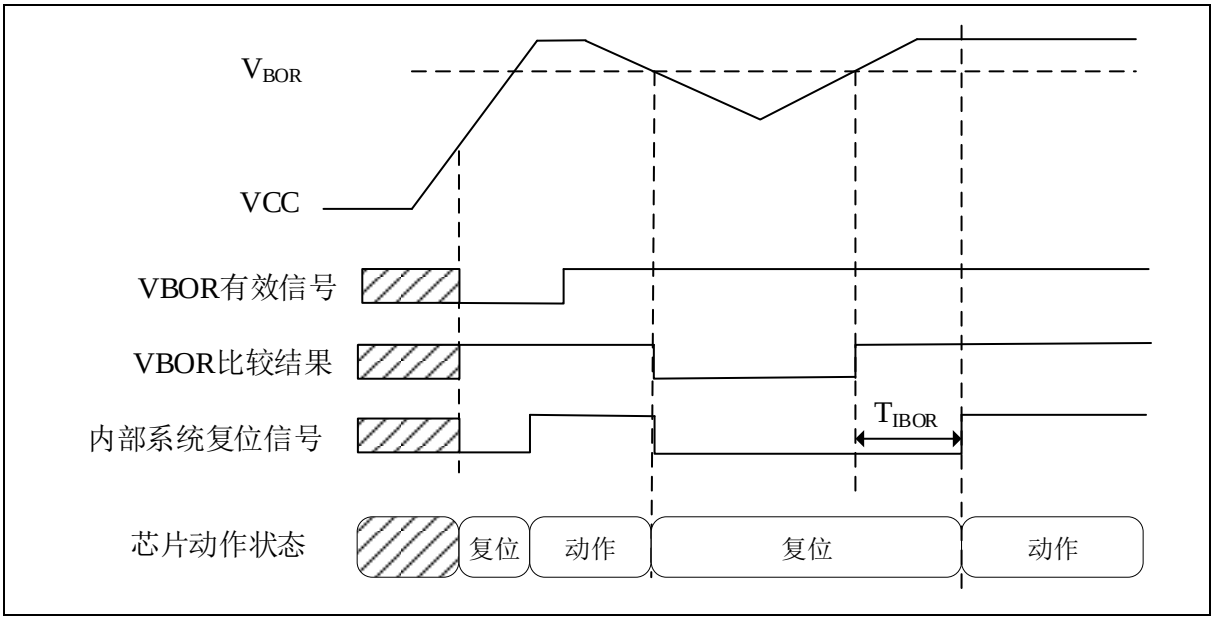


图 3-3 欠压复位

3.3.4 可编程电压检测 1 复位、可编程电压检测 2 复位

可编程电压检测 1 和可编程电压检测 2 复位时电压监测电路引起的复位。

在可编程电压检测 1 有效并设置为复位使能后，如果 VCC 低于可编程电压检测 1 的监测电压，产生可编程电压检测 1 复位，RMU_RSTF0.PVD1F 被置位。当 VCC 电压高于可编程电压检测 1 的监测电压后，经过 PVD1 的复位时间(tIPPVD1)后解除复位。

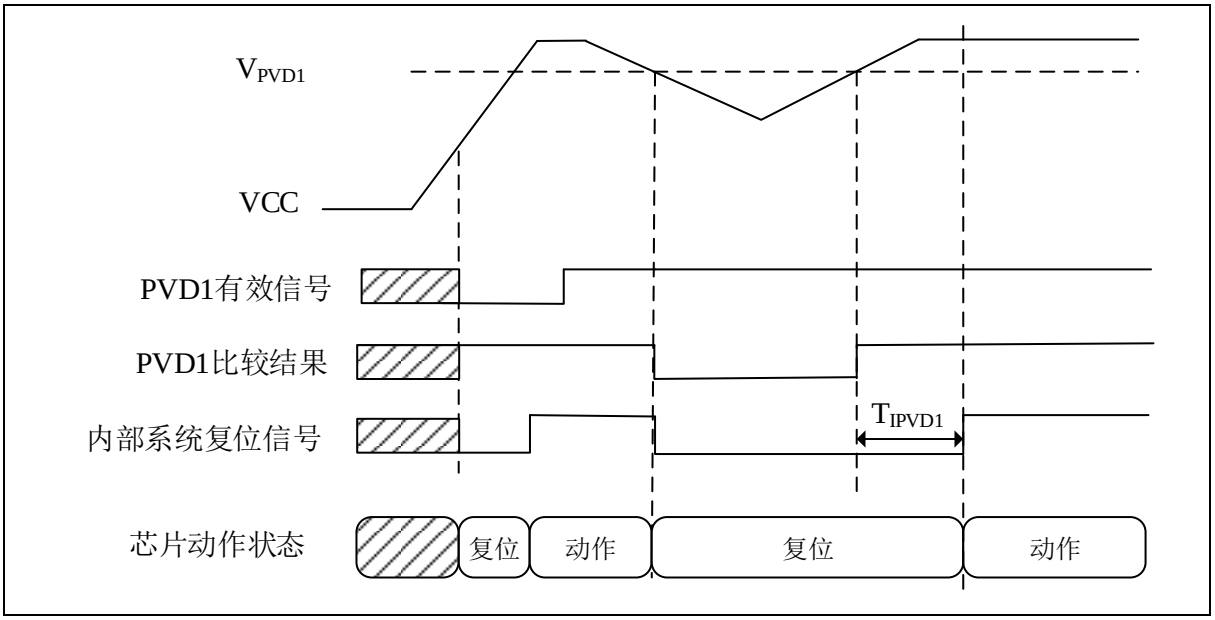


图 3-4 可编程电压检测 1 复位

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(tIPPVD2)后解除复位。

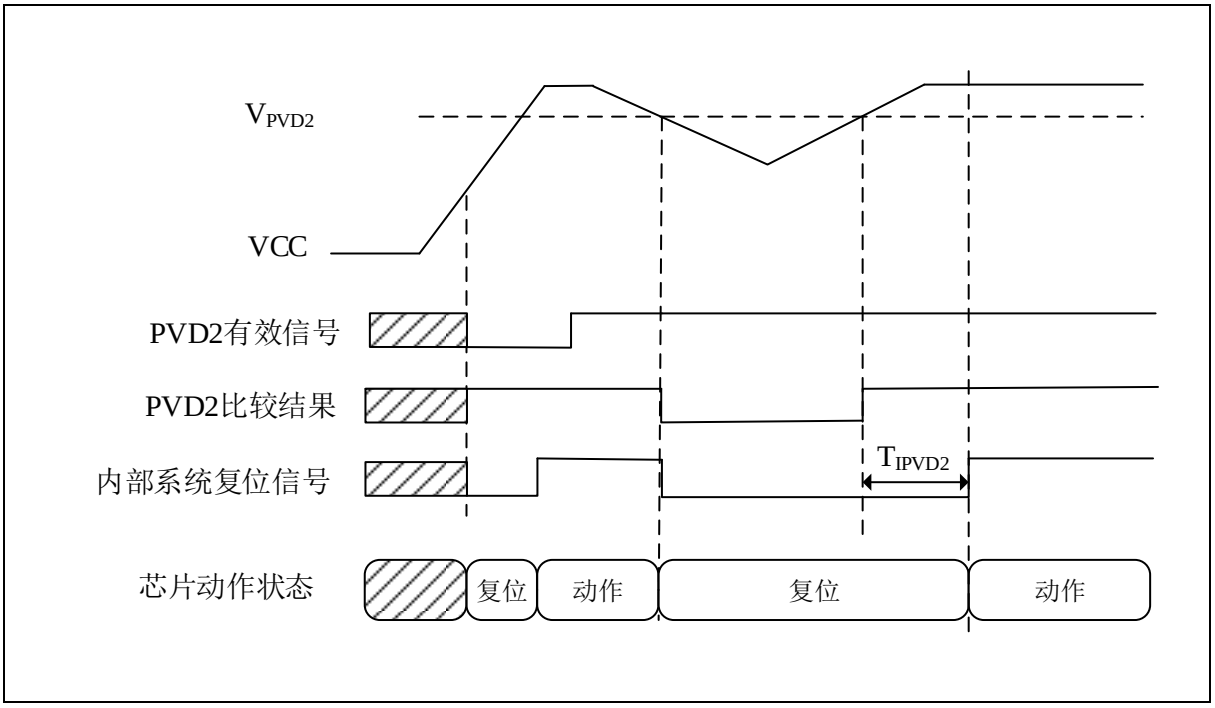


图 3-5 可编程电压检测 2 复位

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

3.3.5 看门狗复位、专用看门狗复位

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

设定看门狗复位有效后，在看门狗定时器产生下溢或者不在刷新允许期间进行写操作时产生看门狗复位。看门狗复位将 RMU_RSTF0.WDRF 置位。产生看门狗复位后，经过内部复位时间 TRIPT 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 TRIPT 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）】。

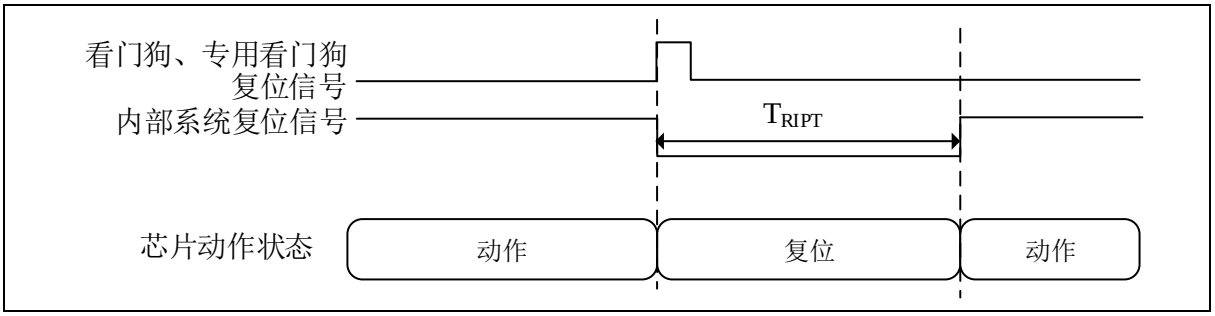


图 3-6 看门狗和专用看门狗复位

3.3.6 掉电唤醒复位

掉电唤醒复位是芯片在设置 PWR_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(TIPDWK)后，解除掉电唤醒复位。返回时间根据设定的具体掉电模式有所不同，在掉电模式 1 时最小，掉电模式 3 时最大。

有关掉电唤醒复位的详细内容，请参照【掉电模式】。

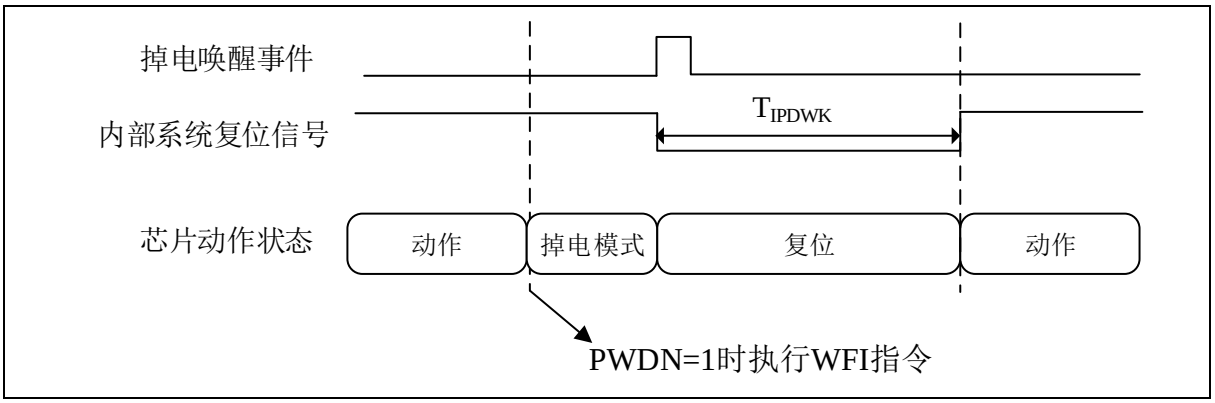


图 3-7 掉电唤醒复位

3.3.7 软件复位

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

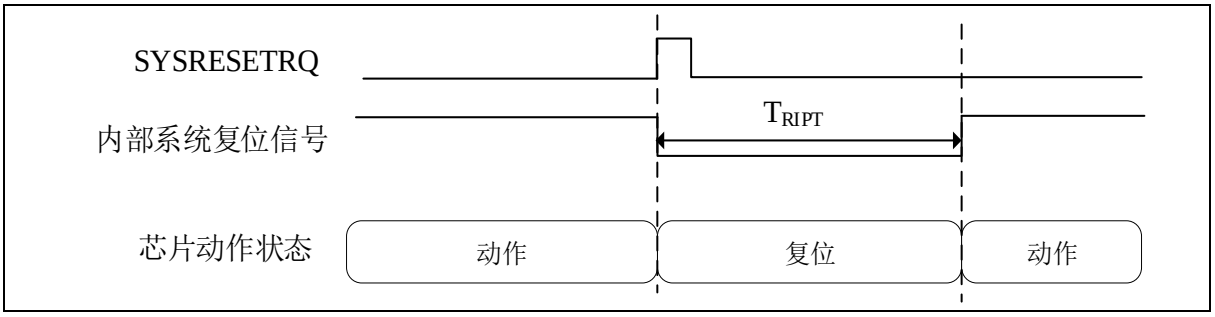


图 3-8 软件复位

3.3.8 MPU 错误复位

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

有关 MPU 错误复位的设定，请参照【存储保护单元（MPU）】。

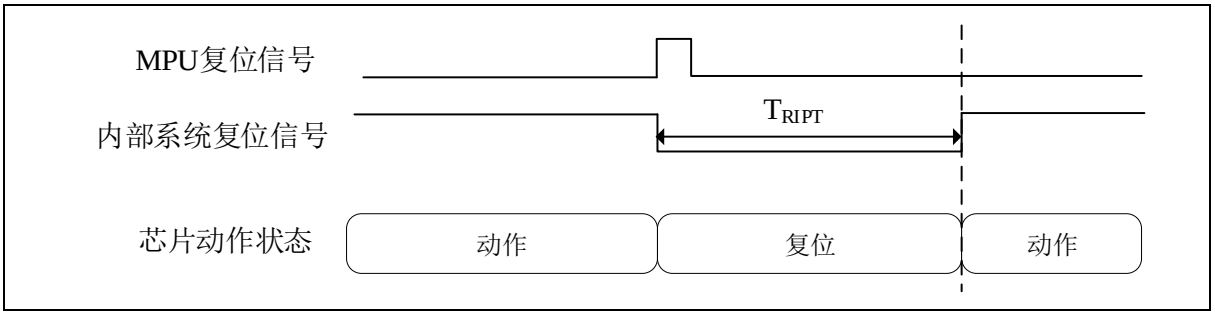


图 3-9 MPU 错误复位

3.3.9 RAM 奇偶校验复位

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

RAM 奇偶校验错误复位的设定，请参照【内置 SRAM（SRAM）】。

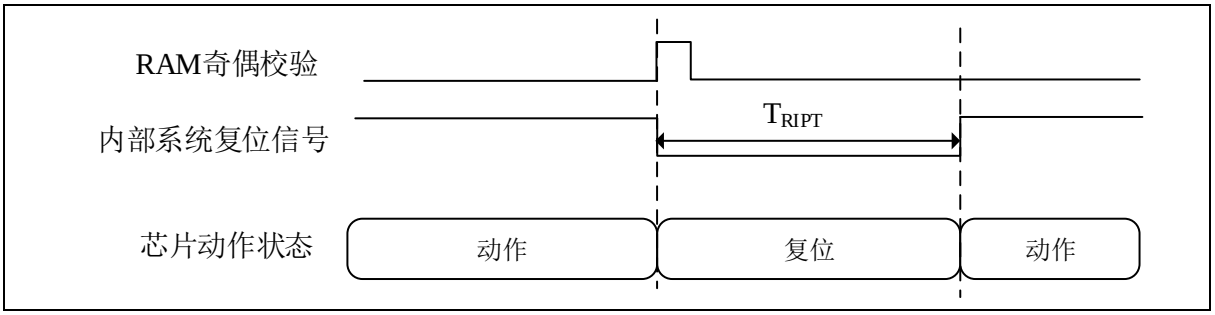


图 3-10 RAM 奇偶校验复位

3.3.10 RAMECC 复位

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

RAMECC 复位的设定，请参照【内置 SRAM（SRAM）】。

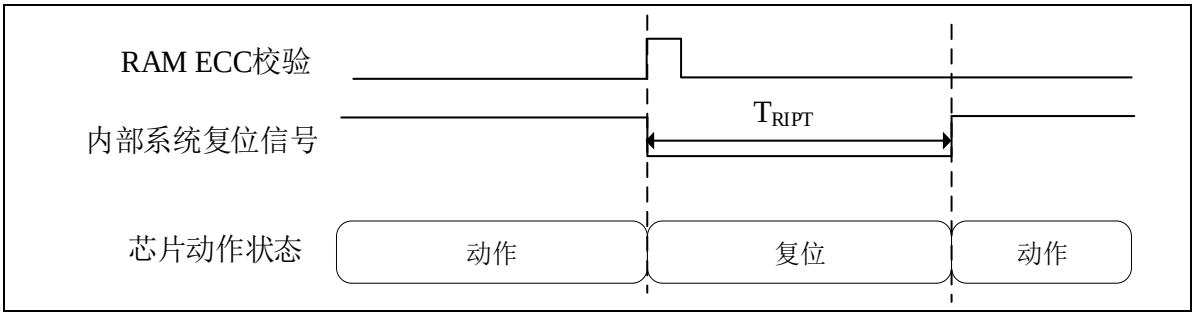


图 3-11 RAMECC 复位

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 TRIPT 之后，芯片解除复位。

时钟频率异常复位的设定，请参照【时钟频率测量】。

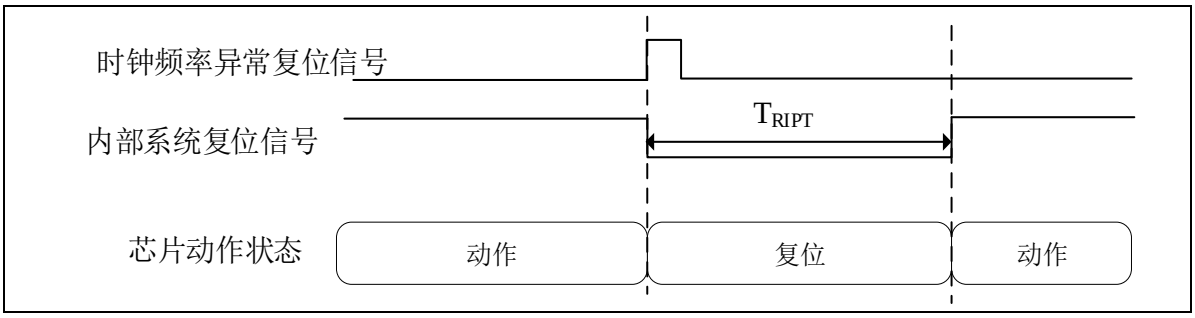


图 3-12 时钟频率异常复位

3.3.12 外部高速振荡器异常停振复位

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

外部高速振荡器异常停振复位的设定，请参照【外部高速振荡器故障检测】。

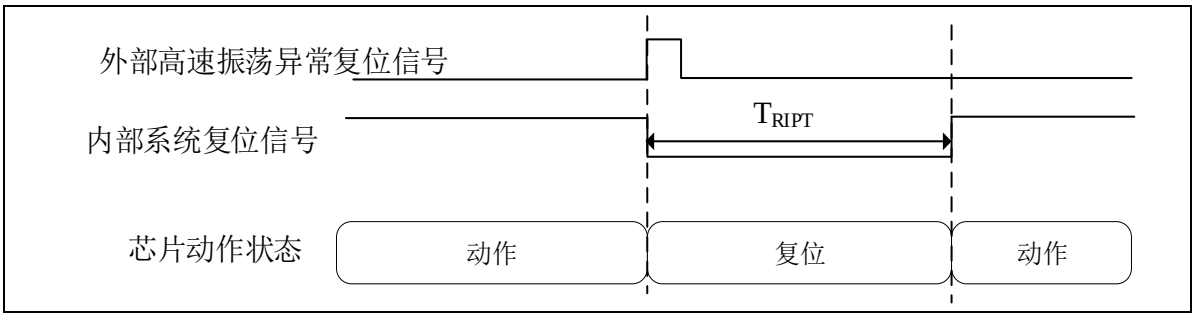


图 3-13 外部高速振荡异常复位

3.3.13 Cortex-M4 Lockup 复位

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

Cortex-M4 Lockup 复位必须在 RMU_PRSTCR0.LKUPREN 置位后才能使用。

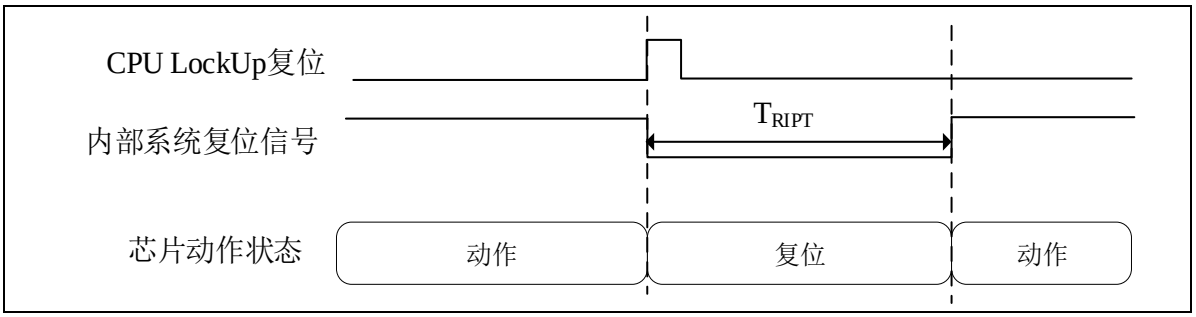


图 3-14 M4 Lockup 复位

3.3.14 复位方式的判断

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

3.4 寄存器说明

寄存器一览如表 3-3 所示。

BASE ADDR: 0x4004CCF0

寄存器名	符号	偏移地址	位宽	复位值
复位控制寄存器	RMU_PRSTCR0	0x08	8	0x40
复位状态寄存器	RMU_RSTF0	0x0C	32	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 复位控制寄存器（RMU_PRSTCR0）

复位值：0x40h

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

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

3.4.2 复位标志寄存器 0（RMU_RSTF0）

复位值：0xXXXXh（根据复位方式不同，复位值不同）

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

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

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

4 时钟控制器（CMU）

4.1 简介

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

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

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

1） 外部高速振荡器（XTAL）

2） 外部低速振荡器（XTAL32）

3） PLLH 时钟（PLLH）

4） 内部高速振荡器（HRC）

5） 内部中速振荡器（MRC）

6） 内部低速振荡器（LRC）

系统时钟的最大运行时钟频率可以达到 240MHz。SWDT 有独立的时钟源：SWDT 专用内部低速振荡器（SWDTLRC）。实时时钟（RTC）使用外部低速振荡器或者内部低速振荡器作为时钟源。USB-FS 的 48MHz 时钟可以选择系统时钟、PLLH、PLLA 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

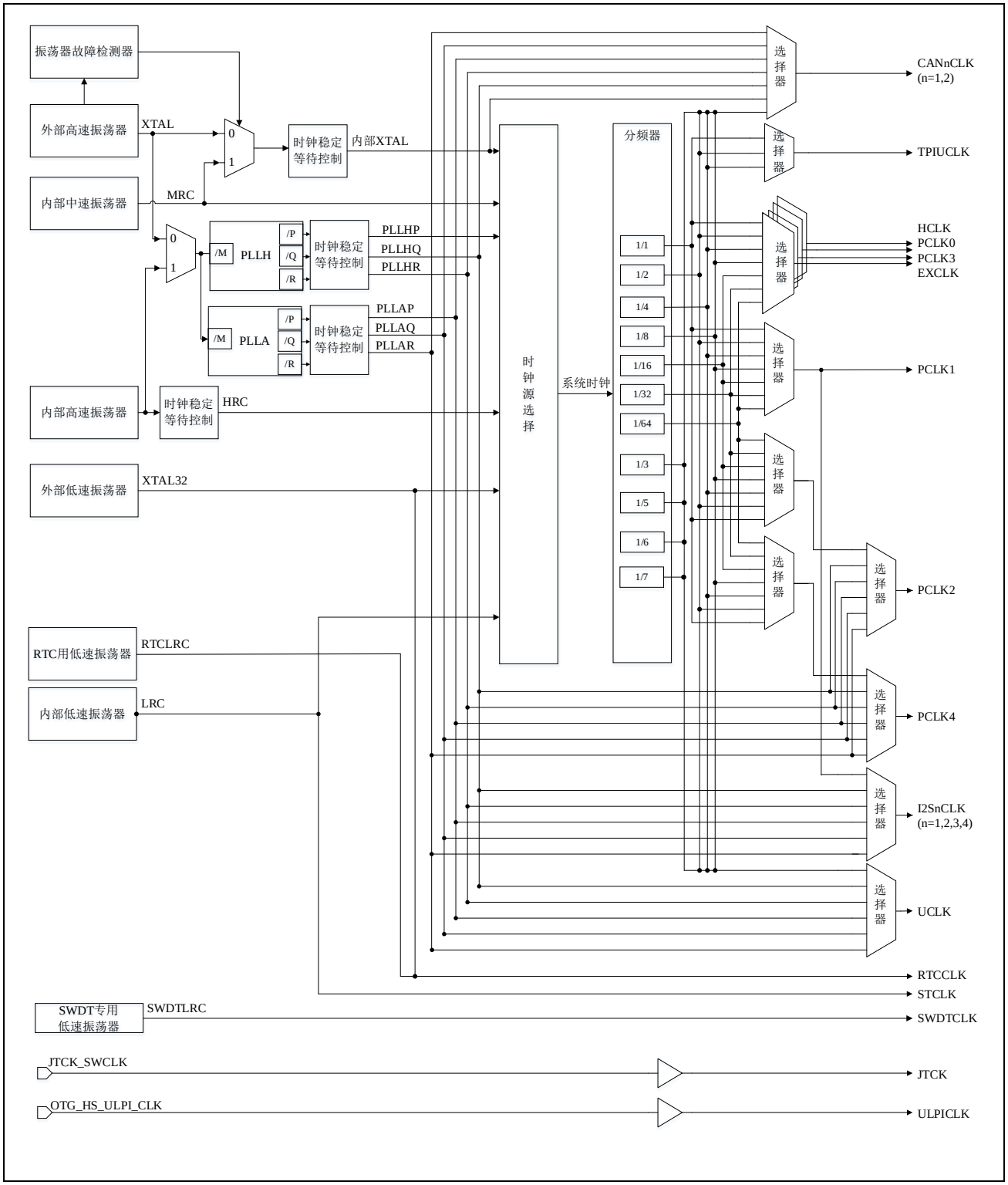


图 4-1 时钟系统框图

4.2.2 时钟频率测量框图

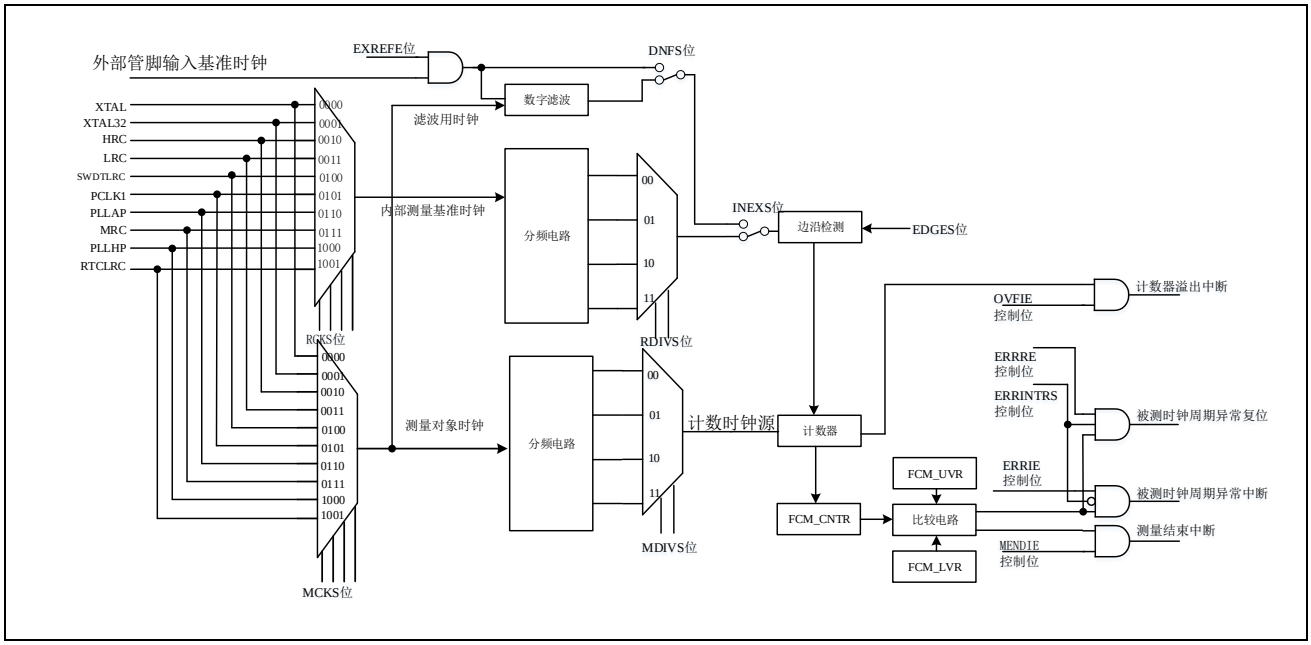


图 4-2 时钟频率测量框图

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：4~24MHz 外部时钟输入：最高 24MHz 振荡器故障检测功能
外部低速振荡器（XTAL32）	晶振的频率范围：32.768kHz
PLLH 时钟（PLLH）	输入时钟：外部高速振荡器或者内部高速振荡器 PLLH 输入时钟分频：1~4 任意分频可选 PFD 输入时钟频率=输入时钟/PLLH 输入时钟分频，频率范围 8MHz~24MHz PLLH 倍频系数：25~150 倍 VCO 振荡频率：600MHz~1200MHz PLLHQ 输出分频比： 2~16 任意分频 PLLHP 输出分频比： 2~16 任意分频 PLLHR 输出分频比： 2~16 任意分频 PLLHP 输出频率=（输入时钟/PLLH 输入时钟分频）*PLLH 倍频系数/PLLHP 输出分频比 PLLHQ 输出频率=（输入时钟/PLLH 输入时钟分频）*PLLH 倍频系数/PLLHQ 输出分频比 PLLHR 输出频率=（输入时钟/PLLH 输入时钟分频）*PLLH 倍频系数/PLLHR 输出分频比
PLLA 时钟（PLLA）	输入时钟：外部高速振荡器或者内部高速振荡器 PLLA 输入时钟分频：1~24 任意分频可选 PFD 输入时钟频率=输入时钟/PLLA 输入时钟分频，频率范围 1MHz~24MHz PLLA 倍频系数：20~480 倍 VCO 振荡频率：240MHz~480MHz PLLAP 输出分频比：2~16 任意分频 PLLAQ 输出分频比：2~16 任意分频

	PLLAR 输出分频比：2~16 任意分频 PLLAP 输出频率=（输入时钟/PLLA 输入时钟分频）*PLLA 倍频系数/PLLAP 输出分频比 PLLAQ 输出频率=（输入时钟/PLLA 输入时钟分频）*PLLA 倍频系数/PLLAQ 输出分频比 PLLAR 输出频率=（输入时钟/PLLA 输入时钟分频）*PLLA 倍频系数/PLLAR 输出分频比
内部高速振荡器（HRC）	频率：16MHz 或者 20MHz 用户可写寄存器对频率微调
内部中速振荡器（MRC）	频率：8MHz 用户可写寄存器对频率微调
内部低速振荡器（LRC）	频率：32.768kHz 用户可写寄存器对频率微调
RTC 用内部低速振荡器（RTCLRC）	频率：32.768kHz 用户可写寄存器对频率微调
SWDT 专用内部低速振荡器（SWDTLRC）	频率：10kHz

4.4 工作时钟规格

时钟	作用范围	规格
HCLK	CPU、DMAn(n=1、2)、EFM(主闪存)、SRAMn(n=1~4)、SRAMHS、Ret-RAM、MPU、GPIO、DCUn(n=1~8)、INTC，QSPI、DVP、KEY、FIRn(n=1~4)、CORDIC	最高频率 240MHz。 由 CMU_SCFGR 寄存器 HCLKS 位配置。 可选系统时钟的分频：1，2，4，8，16，32，64
PCLK0	Timer6 计数器用时钟、Timer4n(n=1~3)、HRPWM、TimerAn(n=1~4)	最高频率 240MHz 由 CMU_SCFGR 寄存器 PCLK0S 位配置。 可选系统时钟的分频：1，2，4，8，16，32，64
PCLK1	USBHS/USBFS 控制逻辑、USARTn（n=1~10）、SPIn(n=1~6)、Timer0n(n=1、2)、Timer2n(n=1~4)、TimerAn(n=5~12)、Timer6（控制逻辑）、EMB、CRC、HASH、AES、I2Sn(n=1~4)控制逻辑、SDIOC、Ether、CANn(n=1、2)控制逻辑	最高频率 120MHz 由 CMU_SCFGR 寄存器 PCLK1S 位配置。 可选系统时钟的分频：1，2，4，8，16，32，64
PCLK2	ADC 转换时钟	最高频率 60MHz 由 CMU_SCFGR 寄存器的 PCLK2S 位和 CMU_PERICKSEL 寄存器 PERICKSEL 位配置。 可选系统时钟的分频：1，2，4，8，16，32，64 可选独立时钟源: PLLAP,PLLAQ,PLLAR,PLLHQ,PLLHR
PCLK3	RTC（控制逻辑）、I2Cn(n=1~6)、CMPn(n=1、2)、WDT、SWDT（控制逻辑）、WKTM、OTS、FCM、VBAT 备份寄存器、CTC	最高频率 60MHz 由 CMU_SCFGR 寄存器 PCLK3S 位配置。 可选系统时钟的分频：1，2，4，8，16，32，64
PCLK4	ADCn(n=1~3)（控制逻辑）、DACn(n=1~2)（控制逻辑）、TRNG	最高频率 120MHz 由 CMU_SCFGR 寄存器的 PCLK4S 位和 CMU_PERICKSEL 寄存器 PERICKSEL 位配置。 可选系统时钟的分频：1，2，4，8，16，32，64 可选独立时钟源: PLLAP,PLLAQ,PLLAR,PLLHQ,PLLHR

EXCLK	NFC、DMC、SMC	最高频率 120MHz 由 CMU_SCFGR 寄存器的 EXCKS 位配置。 可选系统时钟的分频：1，2，4，8，16，32，64
UCLK	USBFS/USBHS FullSpeed 通信用时钟	频率 48MHz 由 CMU_ USBCKCFGR 寄存器的 USBCKS 位配置。 可选系统时钟分频 2，3，4，5，6，7，8。 可选独立时钟源: PLLAP,PLLAQ,PLLAR,PLLHQ,PLLHR
CANnCLK	CAN1/CAN2 通信时钟	最高频率 60MHz 由 CMU_ CANCKCFGR 寄存器的 CAN2CKS/CAN1CKS 位配置。 可选系统时钟分频 2，3，4，5，6，7，8。 可选独立时钟源: PLLAP,PLLAQ,PLLAR,PLLHQ,PLLHR,XTAL
STCLK	SYSTICK 定时器外部参考时钟	内部低速振荡器时钟。 CPU 的 SYSTICK 控制和状态寄存器的 CLKSOURCE 位配置 SYSTICK 定时器时钟。
SWDTCLK	SWDT 计数器用时钟	频率 10kHz
TCK	JTAG 用时钟	最高频率 25MHz
TPIUCLK	Cortex-M4 调试跟踪器用时钟	最高频率 60MHz 由 CMU_TPIUCKCFGR 寄存器的 TPIUCKS 位配置。 可选系统时钟的分频：1，2，4
ULPICLK	USB-HS HighSpeed 通信用时钟	频率 60MHz
I2SnCLK (n=1~4)	I2Sn(n=1~4)	最高频率 240MHz 由 CMU_I2SCKSEL 寄存器配置。 可选择时钟: PLLAP,PLLAQ,PLLAR,PLLHQ,PLLHR,PCLK3

表 4-1　各个内部时钟的规格

各时钟之间需遵守下列规则：

- HCLK 频率>= PCLK2 频率、PCLK3 频率、PCLK4 频率
- ETHER/ SDIOC 使用时：HCLK 频率> PCLK1 频率

  ETHER/ SDIOC 未使用时：HCLK 频率>= PCLK1 频率
- NFC 使用时：HCLK 频率> PCLK1 频率

  NFC 未使用时：HCLK 频率>= PCLK1 频率
- 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=1:8,1:4,1:2,1:1,2:1,4:1,8:1

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 4~24MHz。

XTAL 通过 CMU_XTALCR 的 XTALSTP 位打开和关闭。

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

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

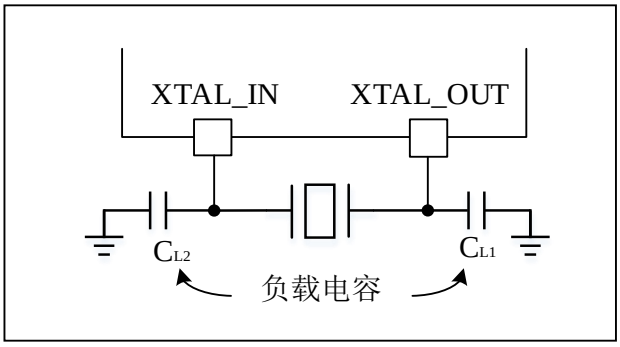


图 4-3 外部高速振荡器连接事例

4.5.1.2 时钟输入模式

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

外部时钟输入的连接例如下图所示。

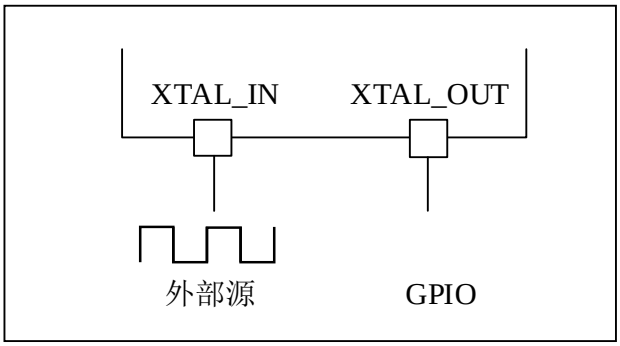


图 4-4 外部时钟输入的连接事例图

4.5.2 外部高速振荡器故障检测

振荡器故障检测是检测外部高速振荡器（XTAL）振荡是否正常。

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

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

PLLH、PLLA 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】。

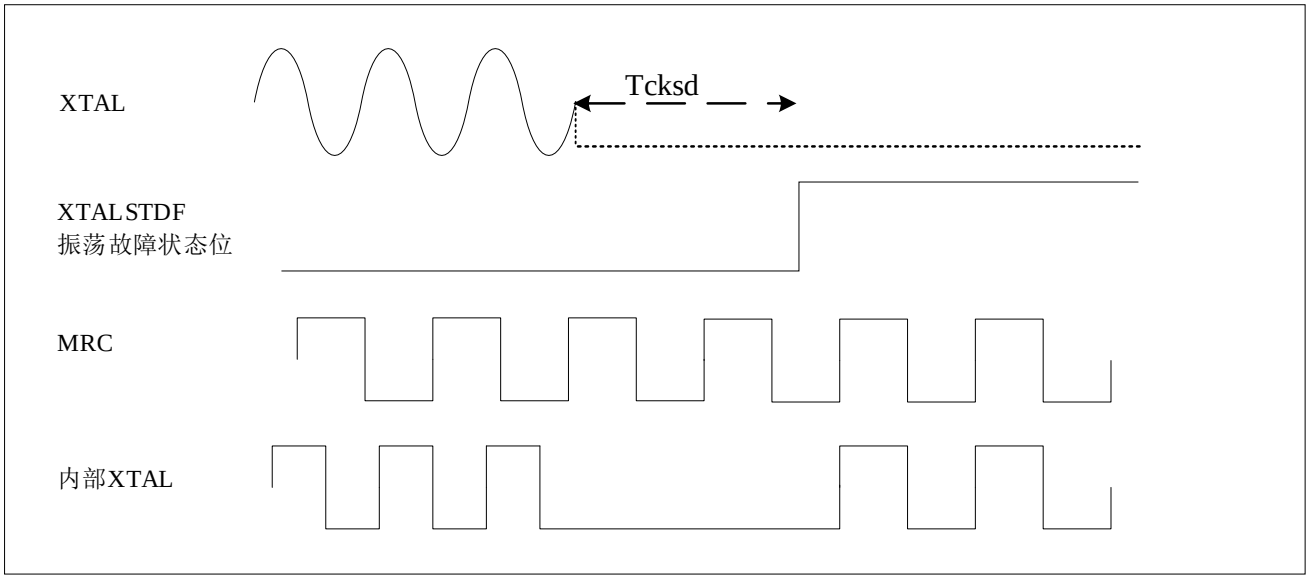


图 4-5　外部高速振荡器故障检测例

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。

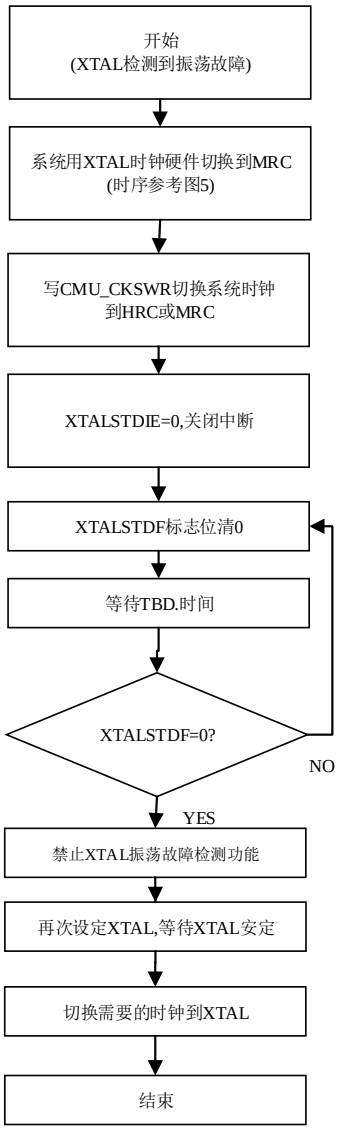


图 4-6　系统时钟选择 XTAL，检测到 XTAL 振荡故障例

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。

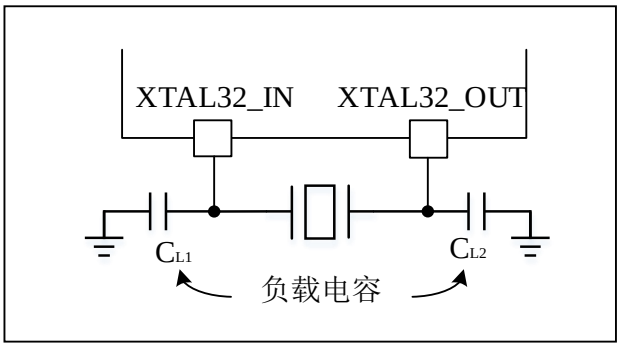


图 4-7　外部低速振荡器连接事例

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. 写 VBATRSTR 寄存器初始化
3. 通过 CMU_XTAL32FGR 设定匹配的 XTAL32 驱动能力
4. 通过 CMU_XTAL32NFR 设定滤波功能
5. CMU_XTAL32CR.XTAL32STP 位写 0，XTAL32 振荡
6. 软件等待 XTAL32 稳定，稳定时间参考电器特性章节

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

4.6 内部 RC 时钟

4.6.1 HRC 时钟

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

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

频率校准

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

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

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

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

4.6.2 MRC 时钟

MRC 时钟信号由内部 8MHz 中速振荡器生成，可直接用作系统时钟。

MRC 振荡器的优点是启动速度快。

频率校准

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

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

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

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

4.6.3 LRC 时钟

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

LRC 振荡器的启动速度快。

频率校准

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

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

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

4.6.4 SWDTLRC 时钟

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

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

4.6.5 RTCLRC 时钟

RTCLRC 时钟信号由内部 32.768kHz 低速振荡器生成。RTCLRC 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/ WKTM 使用。

RTCLRC 振荡器的启动速度快。

频率校准

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

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

RTCLRC 可通过 RTC_CR3 控制寄存器中的 LRCEN 位打开或关闭。

4.7 PLL 时钟

HC32F4Axx 器件具有两个 PLL：

- PLLH 由 XTAL 或 HRC 振荡器提供时钟信号，并具有三个不同的输出时钟：
  - P 分频器输出用于生成系统时钟（最高达 240 MHz）
  - 三个输出都可用于生成 USB、TRNG、ADC、I2S、CAN 时钟。
- PLLA 三个输出亦可用于生成 USB、TRNG、ADC、I2S、CAN 时钟。

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

PLLH/PLLA 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。

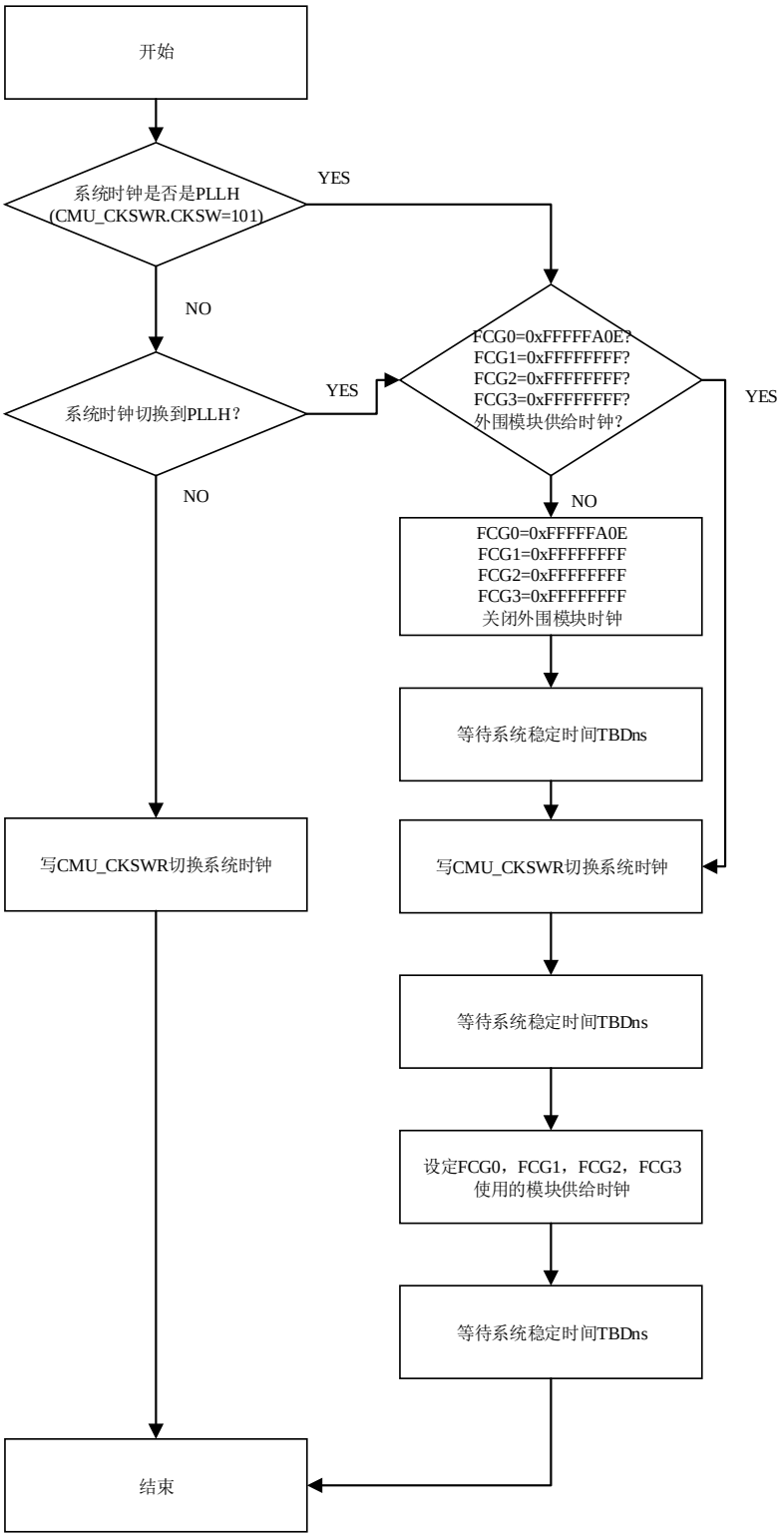
当进入掉电和停止模式后，两个 PLL 将由硬件禁止。

4.8 时钟切换步骤

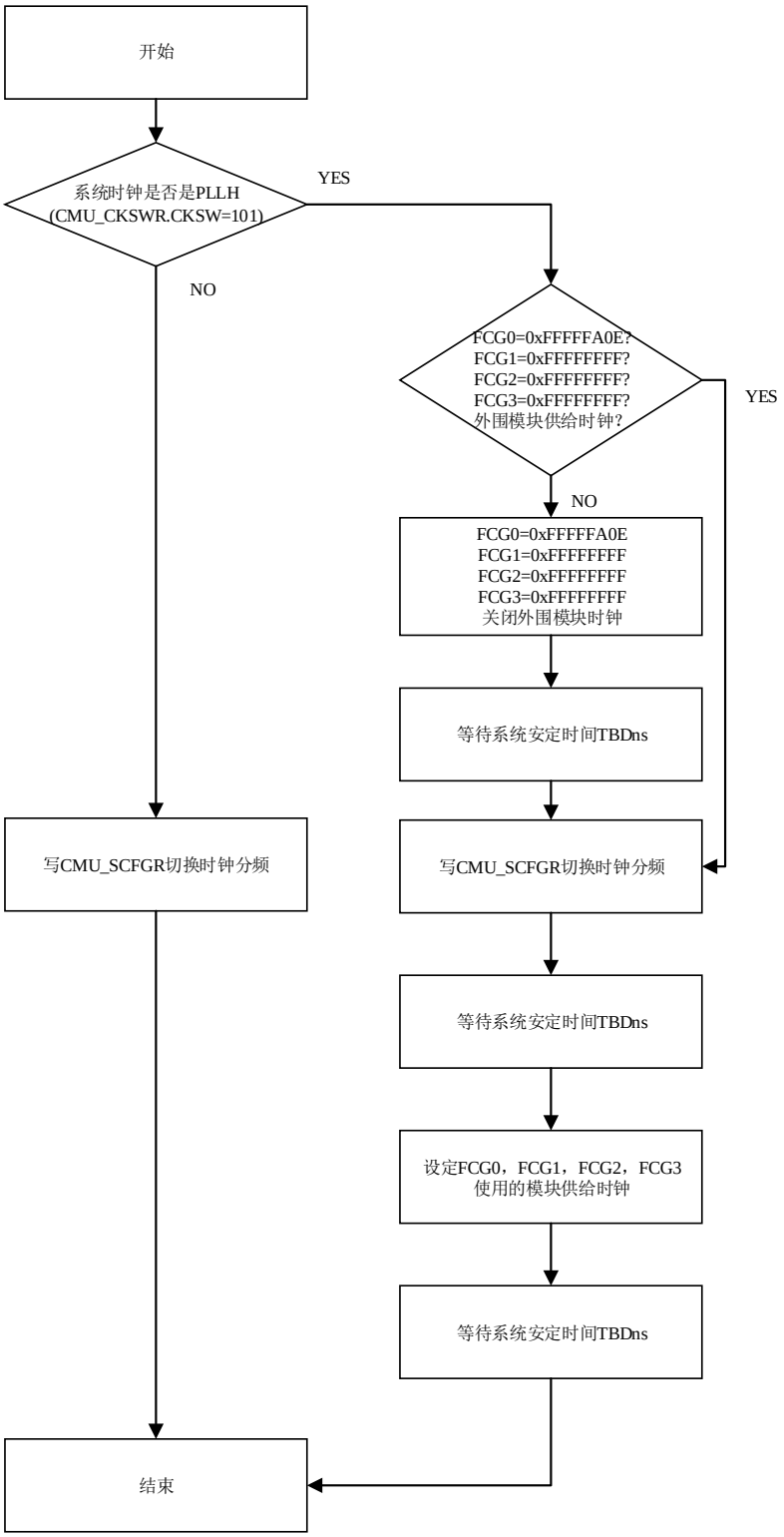
在系统复位后，默认系统时钟为 MRC。通过设定寄存器 CMU_CKSW 切换时钟源，切换步骤参照时钟源切换。只有在目标时钟源已稳定的状态下，才可以从一个时钟源切换到另一个时钟源。

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）】章节进行配置。

4.8.1 时钟源切换


开始
系统时钟是否是PLLH
(CMU_CKSWR.CKSW=101)
YES
NO
系统时钟切换到PLLH？
YES
NO
FCG0=0xFFFFFA0E?
FCG1=0xFFFFFFFF?
FCG2=0xFFFFFFFF?
FCG3=0xFFFFFFFF?
外围模块供给时钟?
YES
NO
FCG0=0xFFFFFA0E
FCG1=0xFFFFFFFF
FCG2=0xFFFFFFFF
FCG3=0xFFFFFFFF
关闭外围模块时钟
等待系统稳定时间TBDns
写CMU_CKSWR切换系统时钟
写CMU_CKSWR切换系统时钟
等待系统稳定时间TBDns
设定FCG0，FCG1，FCG2，FCG3
使用的模块供给时钟
等待系统稳定时间TBDns
结束

4.8.2 时钟分频切换


开始
系统时钟是否是PLLH
(CMU_CKSWR.CKSW=101)
YES
NO
FCG0=0xFFFFFA0E?
FCG1=0xFFFFFFFF?
FCG2=0xFFFFFFFF?
FCG3=0xFFFFFFFF?
外围模块供给时钟?
YES
NO
FCG0=0xFFFFFA0E
FCG1=0xFFFFFFFF
FCG2=0xFFFFFFFF
FCG3=0xFFFFFFFF
关闭外围模块时钟
等待系统安定时间TBDns
写CMU_SCFGR切换时钟分频
写CMU_SCFGR切换时钟分频
等待系统安定时间TBDns
设定FCG0，FCG1，FCG2，FCG3
使用的模块供给时钟
等待系统安定时间TBDns
结束

4.9 时钟输出功能

共有两路时钟输出：

- MCO_1

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

  – HRC 时钟
  – MRC 时钟
  – LRC 时钟
  – XTAL 时钟
  – XTAL32 时钟
  – PLLHP/PLLHQ 时钟
  – PLLAP/PLLAQ/PLLAR 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
  – PLLHP/PLLHQ 时钟
  – PLLAP/PLLAQ/PLLAR 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

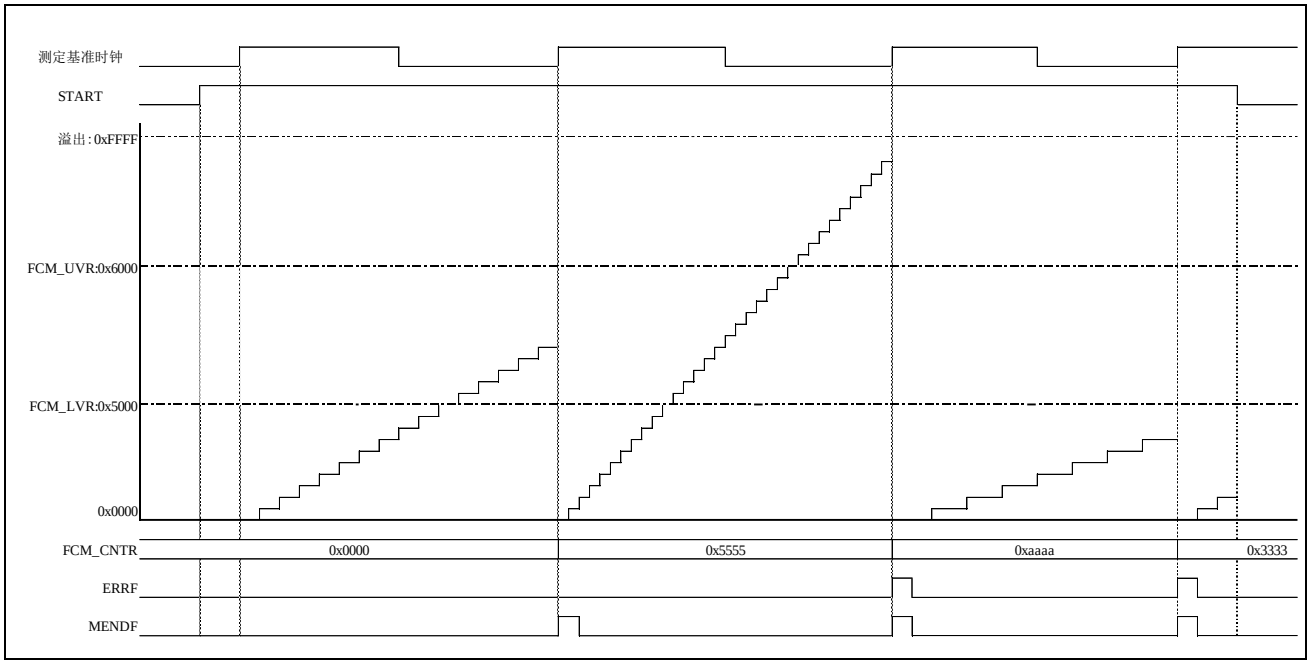


图 4-8 时钟频率测量时序图

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<=FCM_CNTR<=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: 0x40048400

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

基准地址 2: 0x4004C400

CMU_XTAL32控制寄存器	CMU_XTAL32CR	0x000	8	0x00
CMU_XTAL32配置寄存器	CMU_XTALC32CFGR	0x004	8	0xXX
CMU_XTAL32滤波寄存器	CMU_XTAL32NFR	0x014	8	0xXX
CMU_LRC控制寄存器	CMU_LRCCR	0x01C	8	0x00
CMU_LRCTRM校准寄存器	CMU_LRCTRM	0x024	8	0x00
CMU_RTCLRCTRM校准寄存器	CMU_RTCLRCTRM	0x02C	8	0x00

基准地址 3: 0x4004CC00

CMU_XTAL配置寄存器	CMU_XTALCFGR	0x78	8	0x80

基准地址 4: 0x40054000

寄存器名	符号	偏移地址	位宽	复位值
CMU_XTAL稳定配置寄存器	CMU_XTALSTBCR	0x0A2	8	0x05
CMU_XTAL控制寄存器	CMU_XTALCR	0x032	8	0x01
CMU_XTAL振荡故障控制寄存器	CMU_XTALSTDCR	0x040	8	0x00
CMU_XTAL振荡故障状态寄存器	CMU_XTALSTDSR	0x041	8	0x00
CMU_HRC控制寄存器	CMU_HRCCR	0x036	8	由ICG1.HRCSTP值决定
CMU_HRC校准寄存器	CMU_HRCTRM	0x062	8x	0x00
CMU_MRC控制寄存器	CMU_MRCCR	0x038	8	0x00
CMU_MRC校准寄存器	CMU_MRCTRM	0x061	8	0x00
CMU_PLLH配置寄存器	CMU_PLLHCFGR	0x100	32	0x11101300
CMU_PLLH控制寄存器	CMU_PLLHCR	0x02A	8	0x01
CMU_PLLA配置寄存器	CMU_PLLACFGR	0x104	32	0x11101300
CMU_PLLA控制寄存器	CMU_PLLACR	0x02E	8	0x01
CMU_时钟源稳定状态寄存器	CMU_OSCSTBSR	0x03C	8	0x00
CMU_系统时钟源切换寄存器	CMU_CKSWR	0x026	8	0x01
CMU_时钟分频配置寄存器	CMU_SCFGR	0x020	32	0x00000000
CMU_USB时钟配置寄存器	CMU_USBCKCFGR	0x024	8	0x40
CMU_CAN时钟配置寄存器	CMU_CANCKCFGR	0x018	8	0xdd
CMU_AD/TRNG时钟配置寄存器	CMU_PERICKSEL	0x010	16	0x0000
CMU_调试时钟配置寄存器	CMU_TPIUCKCFGR	0x03F	8	0x01
CMU_MCO1时钟输出配置寄存器	CMU_MCO1CFGR	0x03D	8	0x00
CMU_MCO2时钟输出配置寄存器	CMU_MCO2CFGR	0x03E	8	0x00

4.11.1 CMU XTAL 配置寄存器器(CMU_XTALCFGR)

复位值：0x80

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

位	标记	位名	功能	读写
b7	Reserved	-	写入时写“1”	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20~24MHz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 配置寄存器器(CMU_XTALSTBCR)

复位值：0x05

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

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

4.11.3 CMU XTAL 控制寄存器(CMU_XTALCR)

复位值：0x01

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

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

注意：

- XTAL 选作系统时钟或者 PLLH/PLLA 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_XTALSTDCR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
XTALSTDE	-	-	-	-	XTALSTDRIS	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	XTALSTDRIS	XTAL振荡故障复位中断选择	0：XTAL振荡故障产生中断 1：XTAL振荡故障产生复位 注： PLLH、PLLA选择XTAL时钟作为输入源时，只能选择XTAL振荡故障产生复位功能。	R/W
b1	XTALSTDRE	XTAL振荡故障复位允许	0：禁止XTAL振荡故障复位 1：允许XTAL振荡故障复位	R/W
b0	XTALSTDIE	XTAL振荡故障中断允许	0：禁止XTAL振荡故障中断 1：允许XTAL振荡故障中断 通过EMB将Timer6/Timer4的PWM输出置成Hiz输出，XTALSTDIE位需要设置成1。	R/W

注意：

– XTAL 选作系统时钟或者 PLLH/PLLA 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 配置寄存器器(CMU_XTAL32CFGR)

复位值：0xXX

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

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

4.11.7 CMU XTAL32 滤波寄存器(CMU_XTAL32NFR)

复位值：0xXX

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

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

4.11.8 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
HRCTRIM[7:0]							

位	标记	位名	功能	读写
b7~b0	HRCTRIM[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

注意：

– 频率校准需在 HRC 频率保证范围内。

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

4.11.11 CMU MRC 校准寄存器(CMU_MRCTRM)

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）PWR_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
LRCTRIM[7:0]							

位	标记	位名	功能	读写
b7~b0	LRCTRIM[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
-	-	-	-	-	-	-	LRCSTP

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

注意：

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

4.11.15 CMU RTCLRC 校准寄存器(CMU_RTCLRCTRM)

复位值：0x00

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

位	标记	位名	功能	读写
b7~b0	RTCLRCTRIM[7:0]	RTCLRC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 RTCLRC 频率保证范围内。

4.11.16 CMU PLLH 配置寄存器(CMU_PLLHCFGR)

复位值：0x11101300

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PLLHP[3:0]				PLLHQ[3:0]				PLLHR[3:0]				-	-	-	-

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

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

b19-b16	-	-	读出时为“0”,写入时写“0”	R/W
b15-b8	PLLHN[7:0]	PLLH倍频系数	用于控制PLLH的VCO的倍频系数，在PLLH停止条件下写PLLHN。确保PLLH的VCO频率介于600MHz到1200MHz之间。 PLLH的VCO频率=PLLH的PFD输入时钟频率 * PLLHN 00011000：25 00011001：26 00011010：27 00011011：28 …… 10010100：149 10010101：150	R/W
b7	PLLSRC	PLLH/PLLA输入时钟源选择	0：选择外部高速振荡器作为PLLH/PLLA的输入时钟 1：选择内部高速振荡器作为PLLH/PLLA的输入时钟	R/W
b6-b2	-	-	读出时为“0”,写入时写“0”	R/W
b1-b0	PLLHM[1:0]	PLLH输入时钟分频系数	用于在PLLH的VCO之前对PLLH输入时钟进行分频。在PLLH停止条件下写PLLHM。确保PLLH的PFD输入时钟频率介于8MHz到24MHz之间。 00：1分频 01：2分频 10：3分频 11：4分频	R/W

4.11.17 CMU PLLH 控制存器(CMU_PLLHCR)

复位值：0x01

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

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

注意：

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

4.11.18 CMU PLLA 配置寄存器(CMU_PLLACFGR)

复位值：0x11101300

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PLLAP[3:0]				PLLAQ[3:0]				PLLAR[3:0]				-	-	-	PLLAN[8]

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PLLAN[7:0]								-	-	-	PLLAM[4:0]				

位	标记	位名	功能	读写
b31-b28	PLLAP[3:0]	系统时钟用PLLA分频系数	用于控制PLLAP时钟的频率，在PLLA停止条件下写PLLAP。 PLLA输出时钟频率=PLLA的VCO频率/PLLAP 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	PLLAQ[3:0]	系统时钟用PLLA分频系数	用于控制PLLAQ时钟的频率，在PLLA停止条件下写PLLAQ。 PLLA输出时钟频率=PLLA的VCO频率/PLLAQ 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	PLLAR[3:0]	系统时钟用PLLA分频系数	用于控制PLLAR时钟的频率，在PLLA停止条件下写PLLAR。 PLLA输出时钟频率=PLLA的VCO频率/PLLAR 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	R/W

			1111：16分频	
b19-b17	-	-	读出时为“0”,写入时写“0”	R/W
b16-b8	PLLAN[8:0]	PLLA倍频系数	用于控制PLLA的VCO的倍频系数，在PLLA停止条件下写PLLAN。确保PLLA的VCO频率介于240MHz到480MHz之间。 PLLA的VCO频率=PLLA的PFD输入时钟频率 * PLLAN 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	PLLAM[4:0]	PLLA输入时钟分频系数	用于在PLLA的VCO之前对PLLA输入时钟进行分频。在PLLA停止条件下写PLLAM。确保PLLA的PFD输入时钟频率介于1MHz到24MHz之间。 00000：1分频 00001：2分频 00010：3分频 …… 10111：24分频	R/W

4.11.19 CMU PLLA 控制存器(CMU_PLLACR)

复位值：0x01

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

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	PLLAOFF	PLLA使能	用于开始停止PLLA。 0：PLLA动作开始 1：PLLA停止	R/W

注意：

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

4.11.20 CMU 时钟源稳定状态器(CMU_OSCSTBSR)

复位值：0x00

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

4.11.21 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	0 0 0：选择HRC时钟作为系统时钟 0 0 1：选择MRC时钟作为系统时钟 0 1 0：选择LRC时钟作为系统时钟 0 1 1：选择XTAL时钟作为系统时钟 1 0 0：选择XTAL32时钟作为系统时钟 1 0 1：选择PLLH作为系统时钟 1 1 0：禁止设定 1 1 1：禁止设定 注： 1，切换的目标时钟源，需保证在时钟稳定状态。 2，流程参照【时钟源切换】章节 3，PWR_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。	R/W

4.11.22 CMU 时钟分频配置寄存器(CMU_SCFGR)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	HCLKS[2:0]			-	EXCKS[2:0]			-	PCLK4S[2:0]		

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	PCLK3S[2:0]			-	PCLK2S[2:0]			-	PCLK1S[2:0]			-	PCLK0S[2:0]		

位	标记	位名	功能	读写
b31~b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26~24	HCLKS[2:0]	HCLK时钟分频选择位	000：系统时钟的1分频 001：系统时钟的2分频 010：系统时钟的4分频 011：系统时钟的8分频 100：系统时钟的16分频 101：系统时钟的32分频 110：系统时钟的64分频 111：禁止设定 注：PWR_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：PWR_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	R/W

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

	101：系统时钟的32分频 110：系统时钟的64分频 111：禁止设定 注：PWR_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。

4.11.23 CMU USB 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/USB-HS的48MHz时钟源选择	0001：系统时钟2分频 0010：系统时钟3分频 0011：系统时钟4分频 0100：系统时钟5分频 0101：系统时钟6分频 0110：系统时钟7分频 0111：系统时钟8分频 1000：PLLH/Q 1001：PLLH/R 1010：PLLA/P 1011：PLLA/Q 1100：PLLA/R 其他禁止设定 注： 1. 切换的目标时钟源为PLLH/PLLA时，需保证在PLLH/PLLA时钟处于稳定状态。 2. PWR_STPMCR.CKSMRC位为1时，停止模式唤醒后，此寄存器初始化，USBCLK为系统时钟的5分频。	R/W
b3~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

4.11.24 CMU CAN 时钟配置存器(CMU_CANCKCFGR)

复位值：0xdd

b7	b6	b5	b4	b3	b2	b1	b0
CAN2CKS[3:0]				CAN1CKS[3:0]			

位	标记	位名	功能	读写
b7~b4	CAN2CKS[3:0]	CAN2通讯时钟选择	0001：系统时钟2分频 0010：系统时钟3分频 0011：系统时钟4分频 0100：系统时钟5分频 0101：系统时钟6分频 0110：系统时钟7分频 0111：系统时钟8分频 1000：PLLH/Q 1001：PLLH/R 1010：PLLA/P 1011：PLLA/Q 1100：PLLA/R 1101：XTAL 其他禁止设定 注： 1. 切换的目标时钟源为PLLH/PLLA时，需保证在PLLH/PLLA时钟处于稳定状态。 2. 系统时钟选择PLLH时，需要将USB， CAN，QSPI，SPI，通用定时器，FCM，ADC，DAC设定成模块停止状态，再写CMU_SCFGR寄存器切换时钟分频。写完CMU_UCCKCFGR寄存器后，软件等待系统稳定时间TBDns。 3. PWR_STPMCR.CKSMRC位为1时，停止模式唤醒后，此寄存器初始化，CANCLK为XTAL时钟源。	R/W
b3~b0	CAN1CKS[3:0]	CAN通讯时钟选择	0001：系统时钟2分频 0010：系统时钟3分频 0011：系统时钟4分频 0100：系统时钟5分频 0101：系统时钟6分频 0110：系统时钟7分频 0111：系统时钟8分频 1000：PLLH/Q 1001：PLLH/R 1010：PLLA/P 1011：PLLA/Q	R/W

	1100：PLLA/R 1101：XTAL 其他禁止设定 注： 1. 切换的目标时钟源为PLLH/PLLA时，需保证在PLLH/PLLA时钟处于稳定状态。 2. 系统时钟选择PLLH时，需要将USB， CAN，QSPI，SPI，通用定时器，FCM，ADC，DAC设定成模块停止状态，再写CMU_SCFGR寄存器切换时钟分频。写完CMU_UCCKCFGR寄存器后，软件等待系统稳定时间TBDns。 3. PWR_STPMCR.CKSMRC位为1时，停止模式唤醒后，此寄存器初始化，CANCLK为XTAL时钟源。

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

复位值：0x0000

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

位	标记	位名	功能	读写
b15~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3~b0	PERICKSEL[3:0]	AD/TRNG时钟源选择	0000：CMU_SCFGR设定的PCLK2/PCLK4	R/W
			1000：PCLK2/PCLK4配置为PLLHQ	
			1001：PCLK2/PCLK4配置为PLLHR	
			1010：PCLK2/PCLK4配置为PLLAP	
			1011：PCLK2/PCLK4配置为PLLAQ	
			1100：PCLK2/PCLK4配置为PLLAR	
			除此以外禁止设定。	

注意：

– 切换的目标时钟源为 PLLH/PLLA 时，需保证在 PLLH/PLLA 时钟稳定状态。

4.11.26 CMU 调试时钟配置存器(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	R/W
			1：允许	
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	R/W
			01：2分频	
			10：4分频	
			其他禁止设定	

4.11.27 CMU MCO1 配置存器(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：PLLHP 0111：PLLAP 1000：PLLHQ 1001：PLLAQ 1010： PLLAR 1011：系统时钟 其他禁止设定。	R/W

4.11.28 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：PLLHP 0111：PLLAP 1000：PLLHQ 1001：PLLAQ 1010： PLLAR 1011：系统时钟 其他禁止设定。	R/W

4.11.29 FCM 下限比较值寄存器(FCM_LVR)

复位值：0x00000000

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

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

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

4.11.30 FCM 上限比较值寄存器(FCM_UVR)

复位值：0x00000000

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

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

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

4.11.31 FCM 计数器值寄存器(FCM_CNTR)

复位值：0x00000000

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

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

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

4.11.32 FCM 开始停止寄存器(FCM_STR)

复位值：0x00000000

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

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

4.11.33 FCM 测量对象控制寄存器(FCM_MCCR)

复位值：0x00000000

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

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

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7~b4	MCKS[3:0]	测量对象时钟选择位	0000：XTAL	R/W
			0001：XTAL32	
			0010：HRC	
			0011：LRC	
			0100：SWDTLRC	
			0101：PCLK1	
			0110：PLLAP	
			0111：MRC	
			1000：PLLHP	
			1001：RTCLRC	
			其他：设定禁止	
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	

4.11.34 FCM 测量基准控制寄存器(FCM_RCCR)

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EXREFE	-	EDGES[1:0] -		-	-	DNFS[1:0]		INEXS	RCKS[3:0]				-	RDIVS[1:0]	

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

10：1024分频
11：8192分频

4.11.35 FCM 中断复位控制寄存器(FCM_RIER)

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	ERRE	-	-	ERRINTRS	-	OVFIE	MENDIE	ERRIE

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

4.11.36 FCM 标志寄存器(FCM_SR)

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	OVF	MENDF	ERRF

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

4.11.37 FCM 标志位清除寄存器(FCM_CLR)

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
					-	-	-	-	-	-	-	-	OVFCLR	MENDFCLR	ERRFCLR

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

5 电源控制（PWC）

5.1 简介

电源控制器用来控制芯片的多个电源域在多个运行模式和低功耗模式下的电源供给、切换、检测。电源控制器由功耗控制逻辑(PWC)、电源电压检测单元(PVD)、自动切换 VCC 与电池供电的电池备份控制模块（BATBKUP）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)在掉电模式或者 VCC 掉电情况下为 VDDR 域供电。芯片通过功耗控制逻辑(PWC)提供了高速、超低速等两种运行模式，睡眠、停止和掉电等三种低功耗模式。

电源电压检测单元(PVD)提供了上电复位(POR)、掉电复位(PDR)、欠压复位(BOR)、可编程电压检测 1(PVD1)、可编程电压检测 2(PVD2)、基准电压测量通路、VBAT 电压检测、VBAT 电压测量等功能，其中 POR、PDR、BOR 通过检测 VCC 电压，控制芯片复位动作。PVD1 通过检测 VCC 电压，根据寄存器设定使芯片产生复位或者中断。PVD2 通过检测 VCC 电压或者外部输入检测电压，根据寄存器选择产生复位或者中断。基准电压测量通路，是使用 ADC 测量基准电压的功能。VBAT 电压检测，是读取寄存器获得 VBAT 电压高于或者低于 VBAT 检测电压的功能，VBAT 检测电压可以使用寄存器选择 1.8V 或者 2.1V。VBAT 电压测量功能，是指使用 ADC 测量 VBAT 的 1/2 分压，从而获得 VBAT 电压值的功能。

电池备份域在 VCC 掉电情况下通过 VBAT 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，并为 RLDO 提供电源。VDDR 区域在芯片进入掉电模式或者 VCC 掉电情况下可以通过 RLDO 维持电源，保持 4KB 的备份 SRAM(Backup-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、电池备份域和 VDDR 域构成。

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

VDD 域由 CPU、数字外设等数字逻辑、RAM、FLASH 等构成，通过 LDO 产生的 VDD 供电。在 VDD 域中的 RAM 被分为 5 组，由 11 个独立模块构成，可以通过寄存器独立控制每个模块的断电。

BATBKUP 电源域为电池备份域，该区域的电源经电源切换开关自动选择由 VCC 或 VBAT 供电。当 VCC 断电时，备份域的电源自动切换成 VBAT。电池备份域由实时时钟(RTC)、唤醒定时器(WKTM)、外部低速振荡器（XTAL32）、RTC 用内部低速振荡器(RTCLRC)构成。

VDDR 域由 4KB 备份 RAM(Backup-SRAM)构成。Backup-SRAM 在掉电模式下或者 VCC 掉电情况下通过 RLDO 供电，除此以外的模式中通过 LDO 供电。在掉电模式下，Backup-SRAM 能够保持数据。不需要使用备份 RAM 时，可以置位 PWR_BATCR.VBTRSD，使芯片进入掉电模式或者 VCC 掉电时切断 VDDR 域电源，进一步降低功耗。

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

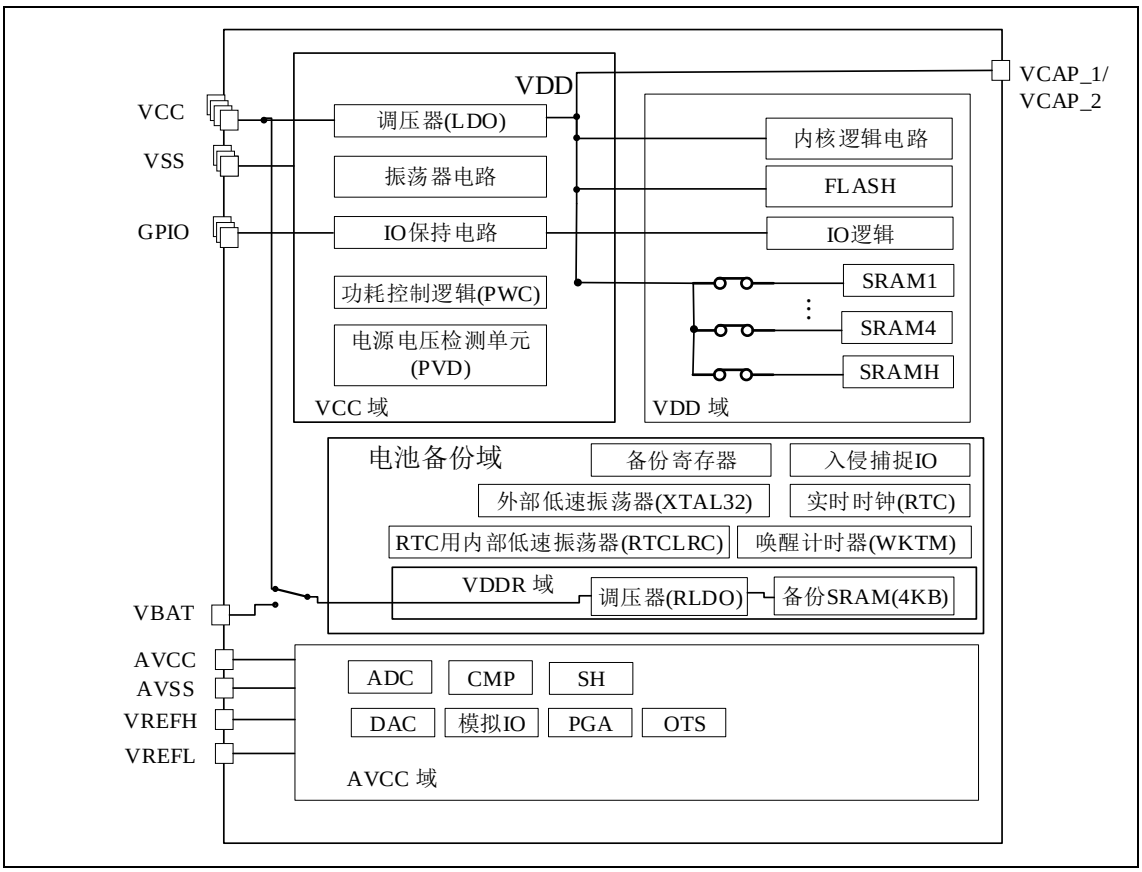


图 5-1 电源构成图

5.2.1 电池备份电源域

电池备份域由内部电源切换器来选择 VCC 供电或 VBAT（电池）供电，该备份域包含 RTC（实时时钟）、XTAL32（外部低速振荡器）、128 字节备份寄存器、4K 字节备份 SRAM、内部低速振荡器、以及 RTCIC0-1 等 2 个管脚构成。当 VCC 引脚的电压降低时，由专用的电池备用电源引脚（VBAT 引脚）给实时时钟（RTC）和 RTC 用内部低速振荡器供电。对于没有外部电池的应用，建议将 VBAT 引脚通过 100nF 的外部陶瓷去耦电容连接到 VCC 引脚上。

电池备用功能的切换运行如图 5-2 所示。VCC 电压降低到 Vswvbat 以下时，电源备份域切换成由 VBAT 供电；VCC 电压升高到 Vswvbat 以上后，电源备份域切换成由 VCC 供电。如果芯片使用掉电模式 3 或者掉电模式 4，Vswvbat 的检测电平不能确定，在 VCC 电压下降时，不能保证芯片可以正确切换到 VBAT 电源，因此，芯片不能同时使用掉电模式 3 和电池备份功能，也不能同时使用掉电模式 4 和电池备份功能。

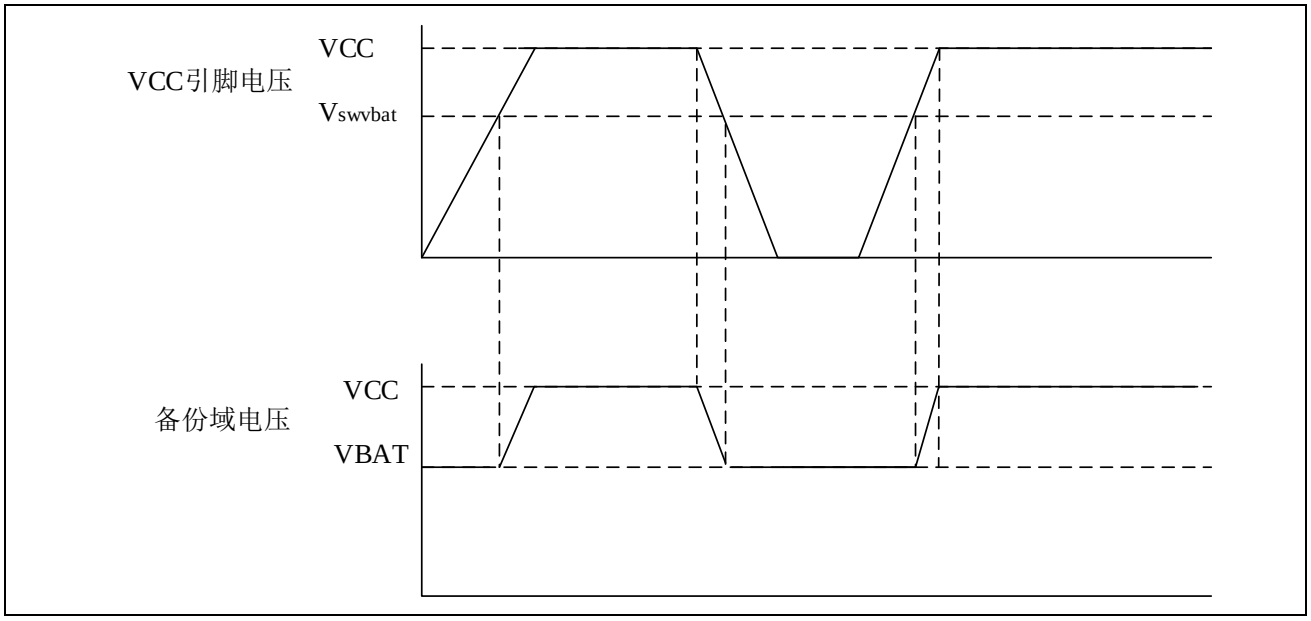


图 5-2 电池备用域电源切换图

RTC 的时钟源可以是 RTC 用内部低速振荡器（RTCLRC）或外部低速振荡器（XTAL32）。

在通过 WFI/WFE 指令进入省电模式之前，Cortex™-M4 需要通过 RTC 寄存器设置预期的唤醒时间并启用唤醒功能，以实现 RTC 定时器唤醒事件。

备份 SRAM 在掉电模式甚至 VCC 掉电后仍然可以保持数据，因此如果持续给 VBAT 供电时，可以将备份 SRAM 视为内部 EEPROM。发生入侵事件时，根据软件设定可以选择备份域 SRAM 是否发生掉电动作，及备份寄存器是否需要初始化。备份 SRAM 可以通过置位寄存器 BRAMC0 的 VBTRSD 位关闭。在备份 SRAM 被关闭后，为备份 SRAM 供电的调压器等电路也随之关闭，可以进一步降低功耗。备份 SRAM 在 VBTRSD 位置位后要等到 RAMVALID 标志位变成 1 后才可以进行读写操作。

使用电池备份域时，需要先写 PWR_VBATRSTR 寄存器将 VBAT 域初始化。PWR_VBATRSTR 写 0xA5 后，外部低速振荡器 XTAL32、备份 RAM 用的 RLDO 启动、唤醒计时器 WKTM 被初始化。

5.3 电源电压检测单元(PVD)说明

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

5.3.1 上电复位/掉电复位动作说明

芯片集成有上电复位、掉电复位电路。上电复位、掉电复位波形如图 5-3 所示，当 VCC 高于指定阈值 VPOR 之后，经过 TRSTTEMPO 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 VPDR 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。

VPOR、VPDR、TRSTTEMPO 等参数的详细信息，请参考**数据手册中电气特性**。

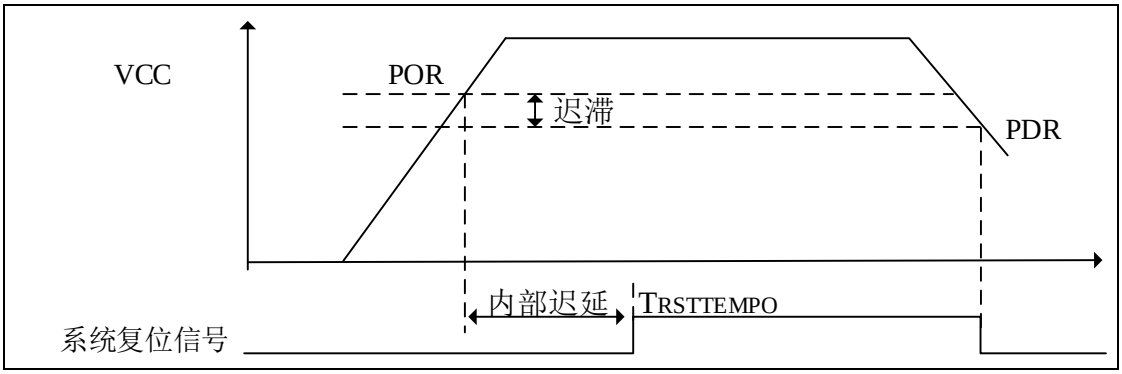


图 5-3 上电复位、掉电复位波形

5.3.2 欠压复位(BOR)说明

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

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

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

表 5-1 BOR 配置

BOR 阈值的电气特性，请参考电气特性章节。

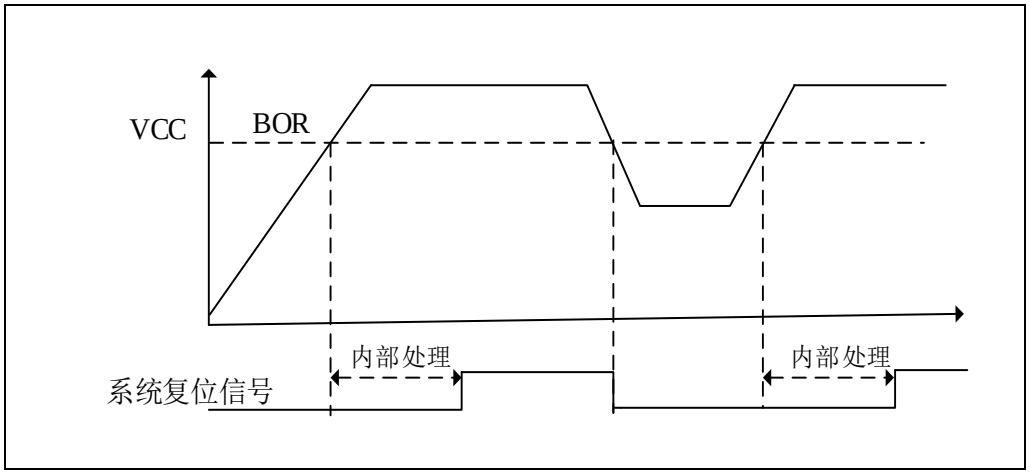


图 5-4 欠压复位波形

5.3.3 可编程电压检测 1(PVD1)、可编程电压检测 2(PVD2)

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

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

可编程电压检测主要特性如表 5-2 所示。

项目	PVD1	PVD2
检测对象	VCC下降/上升过程中是否经过阈值电压点(VPVD1)	VCC下降/上升过程中是否经过阈值电压点(VPVD2)、外部输入电压的上升下降是否经过阈值电压点(VPVD2)
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	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(VPVD1) 或者 VCC上升经过阈值电压点(VPVD1) 或者VCC上升/下降经过阈值电压点(VPVD1)	VCC下降经过阈值电压点(VPVD2) 或者 VCC上升经过阈值电压点(VPVD2) 或者VCC上升/下降经过阈值电压点(VPVD2)
滤波功能	数字滤波	数字滤波
AOS触发功能	VCC下降经过阈值电压点(VPVD1)	VCC下降经过阈值电压点(VPVD2)

表 5-2 PVD1/PVD2 特性

5.3.4 PVD1、PVD2 中断/复位框图

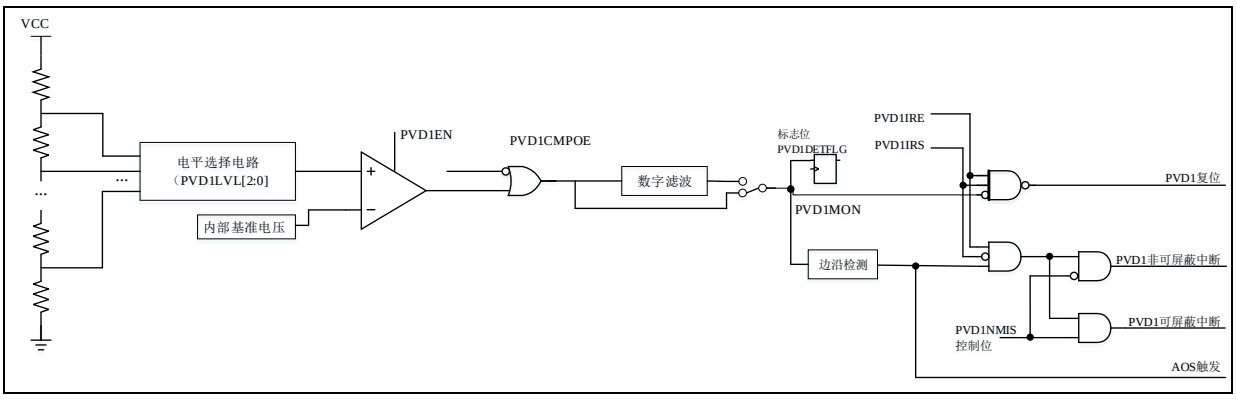


图 5-5 PVD1 中断/复位框图

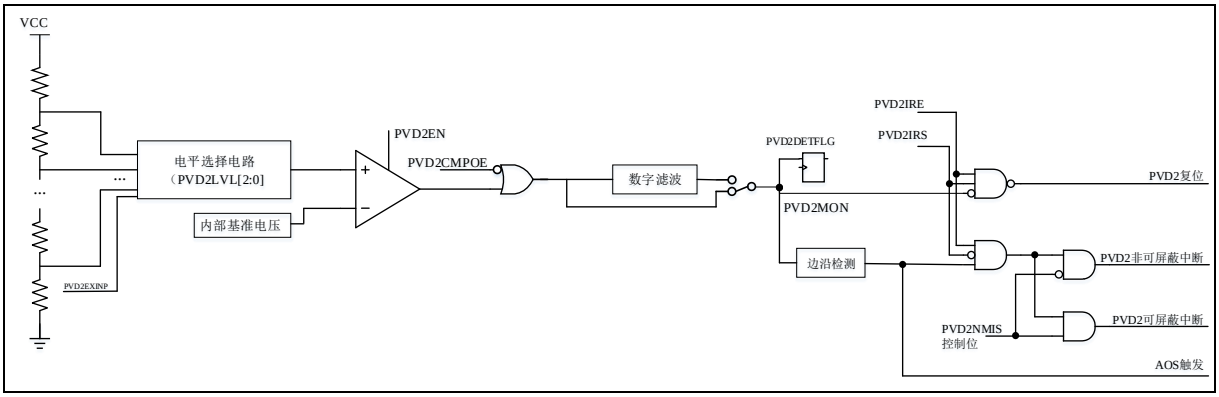


图 5-6 PVD2 中断/复位框图

5.3.5 输入/输出引脚

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

5.3.6 PVD1 中断和复位

在停止模式或者掉电模式中使用 PVD1 电路时，请遵守如下注意事项。

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

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

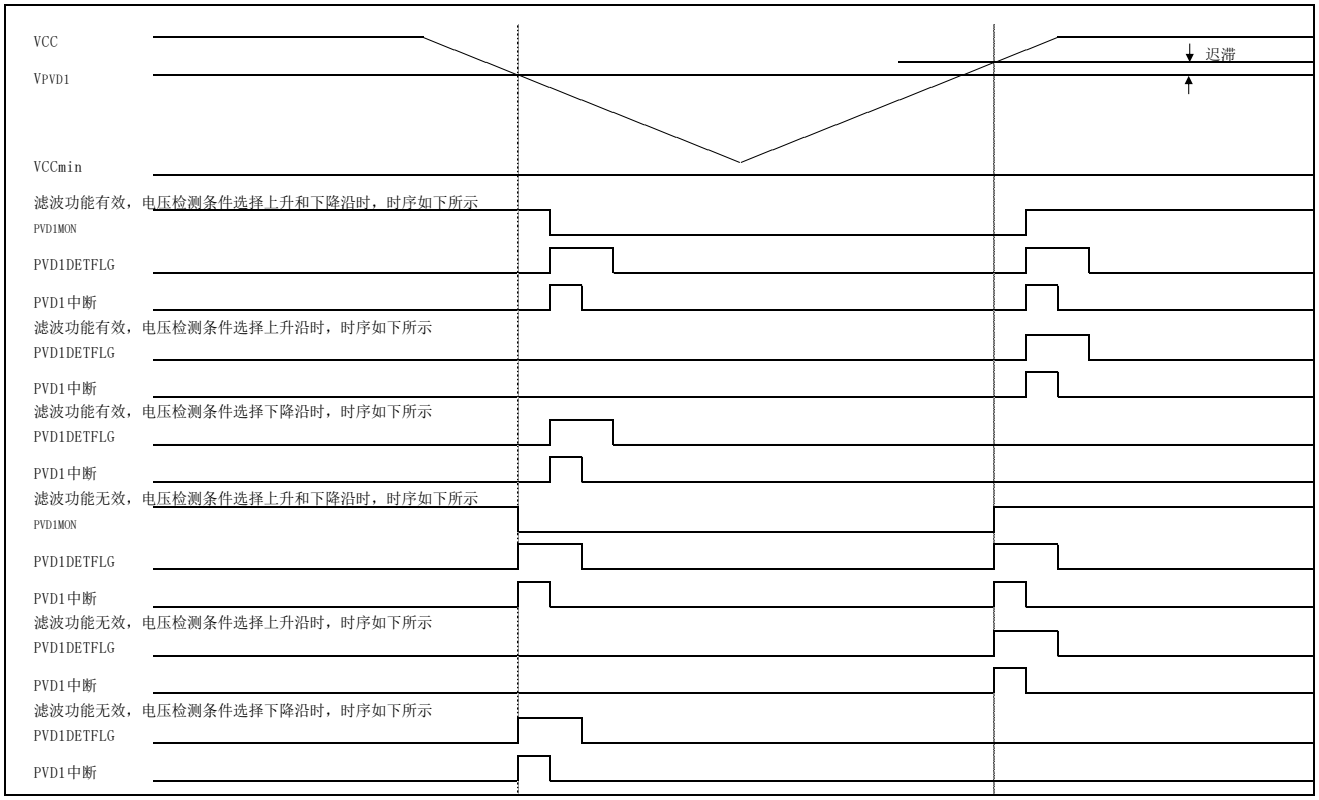


图 5-7 电源监视 1 中断时序图

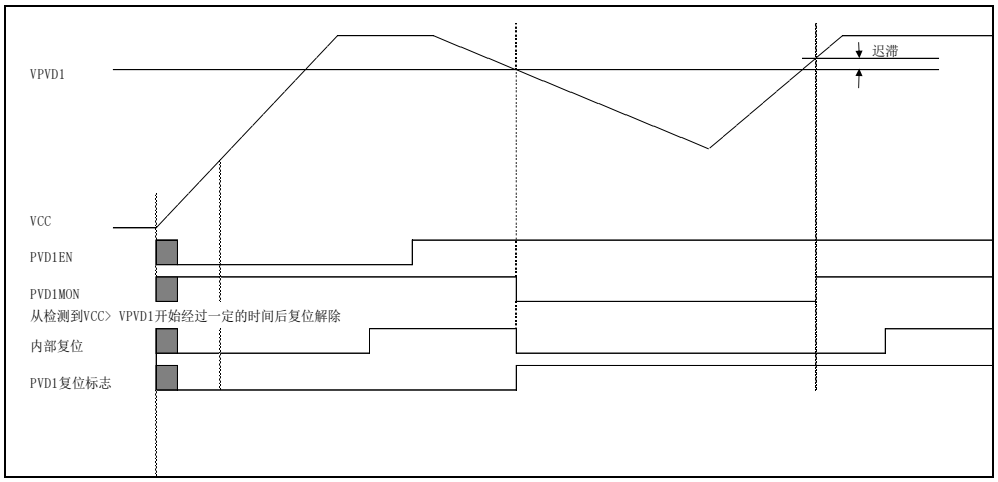


图 5-8 电源监视 1 复位时序图

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) PVD2INTRS 置 0，选择 PVD2 产生中断；选择复位功能时，不能进入掉电模式。

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

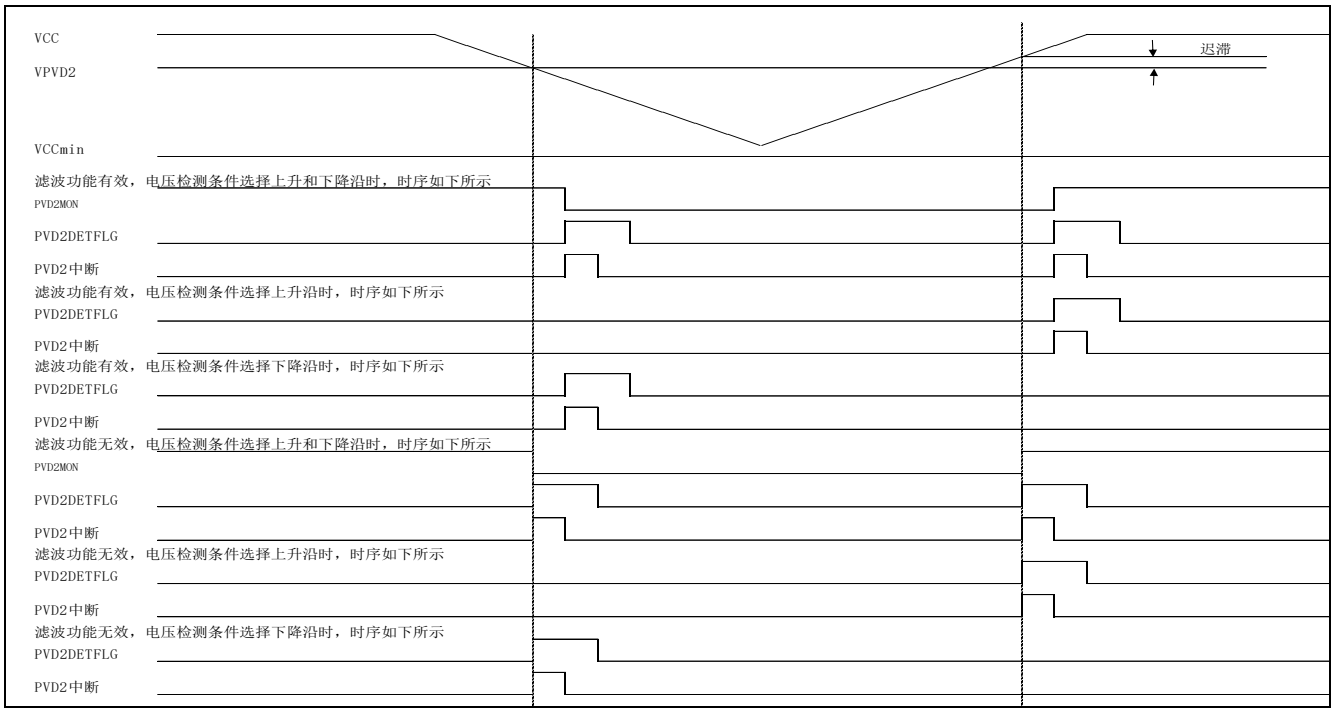


图 5-9 电源监视 2 中断运行时序图

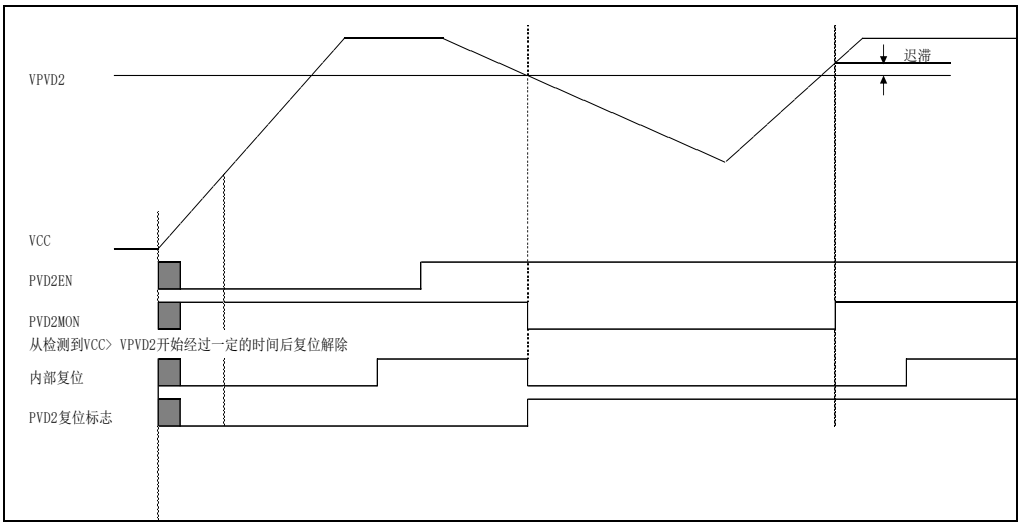


图 5-10 电源监视 2 复位运行时序图

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包括基准电压测量，VBAT 分压检测，VBAT 电压测量等三个功能。基准电压测量通路，是使用 ADC 测量基准电压的功能。VBAT 电压检测，是读取寄存器获得 VBAT 电压高于或者低于 VBAT 检测电压的功能，VBAT 检测电压可以使用寄存器选择 1.8V 或者 2.1V。VBAT 电压测量功能，是指使用 ADC 测量 VBAT 的 1/2 分压，从而获得 VBAT 电压值的功能。

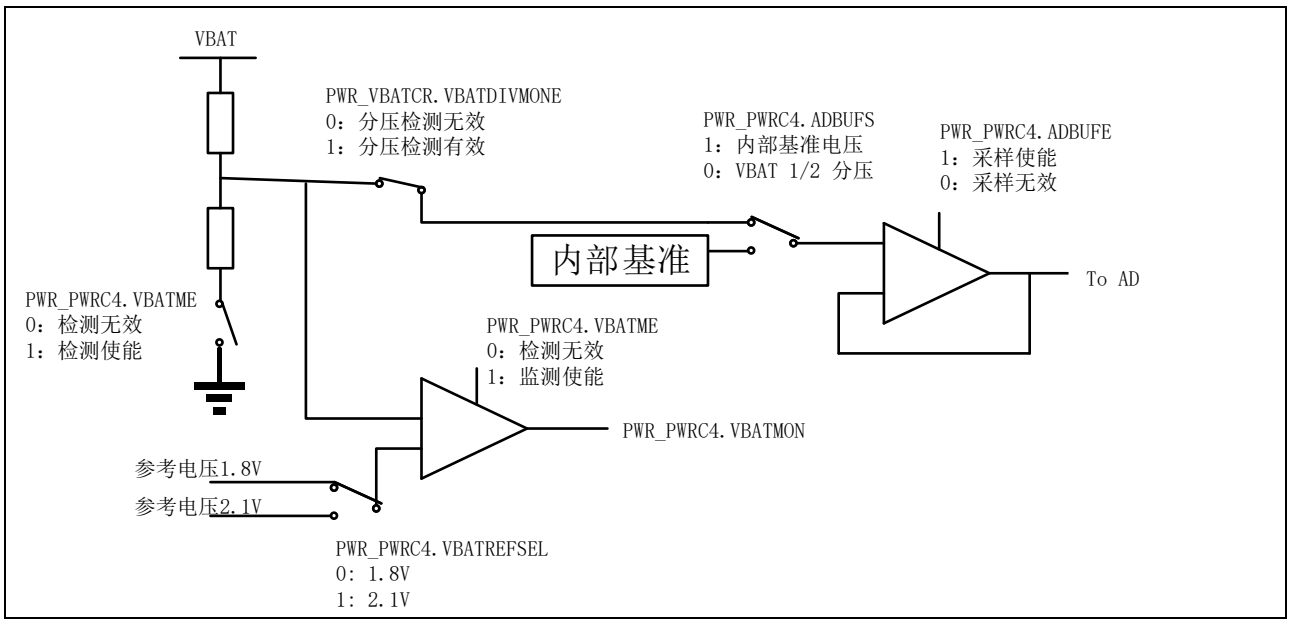


图 5-11 内部电压采样示意图

- 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. PWR_PWRC4.ADBUFS=0，选择内部基准电压
  2. PWR_PWRC4.ADBUFE=1，使能内部电压测量功能
  3. ADC_EXCHSELR. EXCHSEL=1，使 ADC 选通内部电压测量通道
  4. 等待 50uS 后使用 ADC 测量内部基准电压

- VBAT 电压检测

  使用 VBAT 电压检测通路时，需要按照下述步骤选中 VBAT 电压检测通路。

  1. PWR_PWRC4.VBATREFSEL=1，选择 VBATREF=2.1V

     PWR_PWRC4.VBATREFSEL=0，选择 VBATREF=1.8V
  2. PWR_PWRC4.VBATME=1，使能 VBAT 分压电压电路
  3. 等待 50uS
  4. 读 PWRC4.VBATMON 位，读出为 0 时，VBAT>VBATREF。读出为 1 时，VBAT<VBATREF

- VBAT 电压测量

  使用 VBAT 电压测量通路时，需要按照下述步骤选中 VBAT 电压测量通路。

  1. VBAT_CR0.VBATDIVMONE=1，使能 VBAT 分压测量
  2. PWR_PWRC4.VBATME=1，使能 VBAT 分压电路
  3. PWR_PWRC4.ADBUFS=1，选择 VBAT 的 1/2 分压
  4. PWR_PWRC4.ADBUFE=1，使能内部电压测量功能
  5. ADC_EXCHSELR. EXCHSEL=1，使 ADC 选通内部电压测量通道
  6. 等待 50uS 后使用 ADC 测量 VBAT 电压
  7. VBAT_CR0.VBATDIVMONE=0，关闭 VBAT 分压测量

5.4 唤醒定时器

芯片内置了唤醒定时器 WKTM，该计数器可以选择 RTC 用内部低速振荡器(RTCLRC)、外部低速振荡器作为时钟源，在 RTC 使用外部低速振荡器作为时钟动作时还可以选择 64Hz 的内部时钟信号作为时钟源。该计数器是加计数的计数器，在 PWR_WKTC2.WKTCE 置位后，计数器开始计数，当计数值等于 PWR_WKTC1 和 PWR_WKTC0 的 WKTCMP[11:0]设置值后，计数器计数清零后重新计数的同时，产生匹配一致事件，该事件可以用作普通中断，可以用于唤醒停止模式，用于唤醒掉电模式。WKTM 寄存器值上电复位后不定，上电复位后，必须先设定 VBAT 区域复位寄存器后，设置 PWR_WKTC0/PWR_WKTC1/PWR_WKTC2。在启动 WKTM 前，要确保计数时钟处于开启状态。

5.5 动作模式与低功耗模式

系统复位或者上电复位之后，芯片的所有电源域都处于供电状态，芯片进入高速运行模式。在运行模式下，CPU 通过 HCLK 提供时钟，并执行程序代码。芯片提供了高速运行模式、超低速运行模式两种运行模式。芯片可以配置的运行模式如表 5-3 所示。

为了节省 CPU 不需要运行时的功耗，系统提供了睡眠模式、停止模式、掉电模式等三种低功耗模式。芯片可以配置的低功耗模式如表 5-4 所示。在睡眠模式中，芯片的 Cortex™-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。位于电池备份域中的实时时钟、唤醒定时器在低功耗模式中可以动作，备份 SRAM 可以保持数据。

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
高速运行模式 (High Speed Run Mode)	主频240MHz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 Mode)	掉电模式1(PDMD1)	VDD域掉电
	掉电模式2(PDMD2)	VDD域掉电之外，电压检测单元无效
	掉电模式3(PDMD3)	VDD域掉电之外，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PWR_PDWKF0/PWR_PDWKF1/RSTF0外芯片彻底复位。
	掉电模式4(PDMD4)	VDD域掉电之外，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	PWR_STPMCR.STOP=0 PWR_PWRC0.PWDN=0, WFI	PWR_STPMCR.STOP=1 PWR_PWRC0.PWDN=0, WFI	PWR_STPMCR.STOP=1 PWR_PWRC0.PWDN=1, WFI
解除	任意中断或者复位	停止模式下可以使用的中断或复位	掉电模式下可以使用的唤醒事件或者复位
外部高速振荡器	工作可设	停止	停止
外部低速振荡器	工作可设	工作可设	工作可设
内部高速振荡器	工作可设	停止	掉电
内部中速振荡器	工作可设	停止	掉电
内部低速振荡器	工作可设	工作可设	工作可设
WDT专用时钟振荡器	工作可设	工作可设	掉电
PLLA	工作可设	停止	掉电
PLLH	工作可设	停止	掉电
CPU	停止(保持)	停止(保持)	掉电
RAM	工作可设 可以设置工作，掉电	停止(保持) 根据进入待机前的设定，可以维持掉电或者睡眠	掉电
备份RAM	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	可以设置掉电，睡眠
Flash	工作可设	停止(保持)	掉电，内容保持
DMA	工作可设	停止(保持)	掉电
调压器	工作 驱动可以调整	工作 驱动可以调整	停止
上电复位电路	工作	工作	工作 掉电模式1、掉电模式2复位电路精度可以保证、掉电模式3和掉电模式4下上电复位电压不保证

项目	睡眠模式	停止模式	掉电模式
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	工作可设
USB-FS	工作可设	停止(保持)	掉电
Timer0	工作可设	工作可设	掉电
Timer2	工作可设	工作可设	掉电
WKTM	工作可设	工作可设	工作可设
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-PB13，PC0-PC13，PD0-PD15,PE0-PE15, PH2-PH15,PI0-PI12，PI14-15	工作可设	保持	保持或者高阻
PC14-PC15	工作可设	作为外部低速振荡器的管脚使用时，保持振荡器动作；设为GPIO或者其他外设功能时，请设定保持两个管脚为同样的电平	设为GPIO或者其他外设功能时，PC14和PC15的状态可以设置为保持或者高阻，请设定保持两个管脚为同样的电平
PH0-PH1	工作可设	作为外部高速振荡器使用	作为外部高速振荡器使用

项目	睡眠模式	停止模式	掉电模式
		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	时，振荡器停止震荡，管脚状态保持进入掉电模式前的状态；设为GPIO或者其他外设功能时，保持掉电模式之前的状态
NRST复位管脚	芯片外面通过电路上拉到VCC	芯片外面通过电阻上拉到VCC	芯片外面通过电阻上拉到VCC
PA11-PA12、PB14-PB15	工作可设	保持； 由于该管脚的电平被拉高时会产生电流，不使用USB的STOP模式唤醒功能时，进入STOP模式时禁止上拉,。	保持或者高阻； 由于该管脚的电平被拉高时会产生冗余电流，进入掉电模式时禁止上拉
PI13/MD	工作可设	保持；	保持； 芯片外面通过电阻接到地
PA13-PA15，PB3,PB4	工作可设; 作为JTAG功能时，内置上拉电路有效	保持； 作为JTAG功能时，内置上拉电路有效	保持； 作为JTAG功能时，内置上拉电路有效

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

5.5.1 运行模式

芯片有高速、超低速等两种运行模式。根据系统的速度要求，设置最佳的运行模式，使内核电压、驱动能力适配于系统时钟频率，从而达到降低功耗的目的。如表 5-6 所示，根据 PWR_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 FRMC.LVM=1、PWR_RAMOPM=0x9062。

运行模式	频率范围	寄存器设定		
		PWR_PWRC2.DVS	PWR_PWRC2.DDAS	PWR_PWRC3.DDAS
高速运行模式	240MHz以下	11	1111	11111111
超低速运行模式*	8MHz以下	10	0000	00000000

* ：超低速模式只支持 Ta=-40℃~85℃

表 5-6 运行模式说明

高速运行模式与超低速运行模式之间的切换需要遵循如下流程 1 和流程 2。

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 PWR_RAMOPM 设置成 0x9062
4) 确认 FRMC.LVM=1，RAMOPT=0x9062
5) 设置 PWR_PWRC2.DDAS[3:0]为 0000; 设置 PWR_PWRC3.DDAS[7:0]为 0x00; PWR_PWRC2.DVS[1:0]为 10
6) 等待 TSWMD1(30uS)
7) 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 PWR_PWRC2. DDAS[3:0]为 1111
PWR_PWRC3.DDAS[7:0]为 0xff;

PWR_PWRC2. DVS[3:0]根据系统频率要求设置为 11

2) 等待 TSWMD2(30uS)
3) 设置 FLASH 的 FRMC.LVM=0、RAM 动作模式寄存器 PWR_RAMOPM 设置成 0x8043
4) 确认 FRMC.LVM=0，RAMOPT=0x8043
5) 芯片在高速模式下动作

5.5.2 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 SWDTSLPOFF 位为 1，专用看门狗在睡眠模式下停止计数；如果 SWDTSLPOFF 位为 0，专用看门狗在睡眠模式下继续计数。

- 进入睡眠模式

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

- 退出睡眠模式

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

5.5.3 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 SWDTSLPOFF 位为 1，专用看门狗在停止模式下停止计数；如果 SWDTSLPOFF 位为 0，专用看门狗在停止模式下继续计数。

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

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”，置位 PWR_FCG3.ADC3、PWR_FCG3.ADC2、PWR_FCG3.ADC1 后，执行 WFI 指令进入停止模式。

STOP 模式唤醒时，通过 PWR_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 PWR_PWRC1.STPDAS=11 后再进入 STOP 模式时；如果 PWR_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 PWR_STPMCR.STOP=1,PWR_PWRC0.PWDN=0 时执行 WFI 指令即可进入停止模式。表 5-5 给出了芯片的外设和时钟源在停止模式下的状态。

- 解除停止模式

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

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

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

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

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

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

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 PWR_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 PWR_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 PWR_PWRC3.PDTS 位置位。

通过设置 PWR_PWRC0.PDMD[1:0]，可以进一步降低掉电模式的功耗。掉电模式的子模式如表 5-7 所示。掉电模式 1 中电压检测电路可以使用，上电复位检测电路处于动作状态，由于唤醒时不需要等待 VCC 域参考电压、电压检测电路和上电复位检测电路的稳定，因此在实现低功耗的同时，唤醒时间最短。掉电模式 2 中，VCC 域参考电压电路，电压检测电路停止工作，上电复位检测电路处于动作状态，唤醒时需要等待 VCC 域参考电压电路、电压检测电路的稳定时间。掉电模式 3 中，VCC 域参考电压电路、电压检测电路、上电复位检测电路都停止工作，唤醒时需要等待这些电路的稳定，因此在实现最低功耗的同时，唤醒时间比掉电模式 2 和掉电模式 1 长。掉电模式 4 与掉电模式 3 中停止工作的电路相同，因此掉电模式 4 与掉电模式 3 具有相同的功耗。具体功耗数值和唤醒时间，请参考电气特性。

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	01	I_{PD2}	T_{PD2}	VCC域POR、PDR检测电路有效，BOR、PVD1、PVD2无效
掉电模式3	10	I_{PD3}	T_{PD3}	VCC域POR、PDR，BOR、PVD1、PVD2无效
掉电模式4	11	I_{PD4}	T_{PD4}	VCC域POR、PDR，BOR、PVD1、PVD2无效

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 PWR_STPMCR.STOP=1,PWR_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、上电复位和电压检测 0 复位。能够用于掉电模式唤醒的事件包括：

WKUPn_0/1/2/3(n=0/1/2/3)唤醒事件、实时时钟的闹钟和定时事件、电压检测1唤醒事件、电压检测2唤醒事件、唤醒定时器唤醒事件

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

在掉电模式 3 中 POR、PDR，BOR、PVD1、PVD2 电路都处于无效状态，从掉电模式 3 唤醒后，除 PWR_PDWKF0/PWR_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。

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。

掉电唤醒事件通过掉电唤醒使能寄存器(PWR_PDWKE0-PDWKE3)、掉电唤醒事件边沿选择寄存器(PWR_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(PWR_PDWKF0-PWR_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 PWR_PDWKES 选择。

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

掉电模式	不被复位的寄存器
掉电模式1 掉电模式2 掉电模式4	PWR_PWRC0 PWR_PWRC1 PWR_PWRC3 PWR_PDWKE0 PWR_PDWKE1 PWR_PDWKE2 PWR_PDWKES PWR_PDWKF0 PWR_PDWKF1 PWR_PVDCR0 PWR_PVDCR1 PWR_PVDFCR PWR_PVDLCR PWR_PVDICR PWR_PVDDSR
掉电模式3	PWR_PDWKF0 PWR_PDWKF1 RMU_RSTF0

- 解除掉电模式后的管脚状态

  在掉电模式中，根据寄存器设定值，芯片管脚将保持进入掉电模式之前的状态或者高阻状态。如果 PWR_PWRC0.IORTN[1:0]=10 或者 11,管脚状态在掉电模式中为高阻状态，掉电模式解除后管脚初始成高阻状态。如果 PWR_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 PWR_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 PWR_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。

- 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。PTWKn 的结构框图如图 5-12 所示。

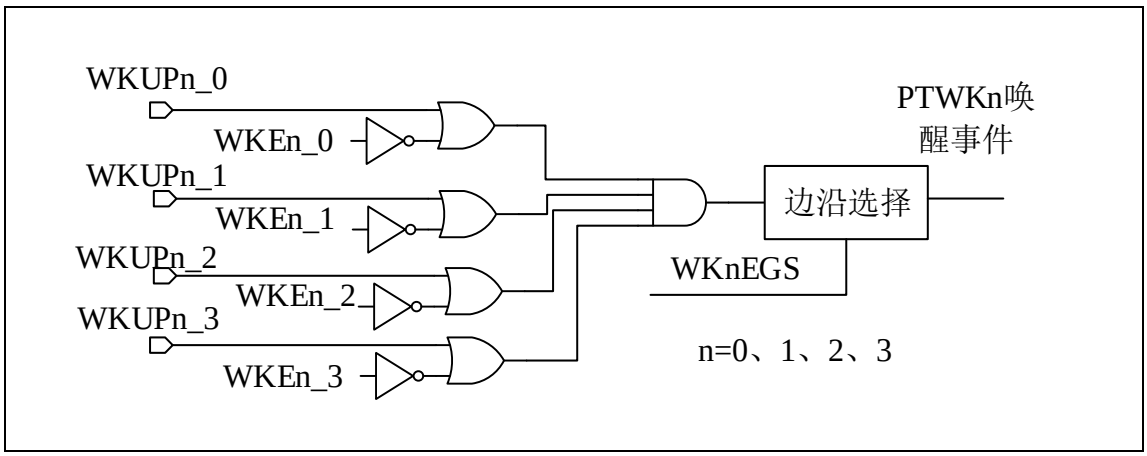


图 5-12 PTWKn 结构框图

- 电池备份域在掉电模式时的动作说明

  电池备份域在芯片进入掉电模式后，备份 RAM 通过 RLDO 继续供电，RTC/WKTM/备份寄存器通过 VCC 继续供电，因此 RTC /WKTM/Backup-SRAM 都能继续动作或者保持数据。掉电唤醒后 VDDR 域由 LDO 供电。进入掉电模式前，需要关闭 VBAT 电压测量功能。

- 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 PWR_FCGn（n=0/1/2/3）关闭不需要使用的功能
5. 关闭 RAM 的电源

5.5.5 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】。

5.5.6 关闭不使用的时钟源

芯片的系统时钟有 6 个时钟源：

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

SWDT 具有独立的专用内部低速振荡器(SWDTLRC)；RTC 可以选择外部低速振荡器或者 RTC 用内部低速振荡器（RTCLRC）作为时钟源。HRC 和 PLL 都配有独立的电源电路，在 HRC 关闭后，可以通过置位 PWR_PWRC1.VHRCSD 位关闭 HRC 的电源，进一步降低功耗；在 PLLA 和 PLLH 都被关闭后，可以通过设置 PWR_PWRC1.VPLLSD=11 关闭 PLLA 和 PLLH 的电源。

有关详细信息，请参考【时钟控制器 CMU】。

5.5.7 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。

5.5.8 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 PWR_RAMPC0 寄存器的相应 RAMPDCn(n=0-10)、PWR_VBATCR. VBTRSD 位可以使对应的 RAM 掉电。

RAM模块	说明	掉电控制位
SRAM1	0x20000000~0x20000FFFF 地址空间用RAM	PWR_RAMPC0.RAMPDC0
	0x20010000~0x20001FFFF 地址空间用RAM	PWR_RAMPC0.RAMPDC1
SRAM2	0x20020000~0x2002FFFF 地址空间用RAM	PWR_RAMPC0.RAMPDC2
	0x20030000~0x2003FFFF 地址空间用RAM	PWR_RAMPC0.RAMPDC3
SRAM3	0x20040000~0x2004FFFF 地址空间用RAM	PWR_RAMPC0.RAMPDC4
	0x20050000~0x20057FFF 地址空间用RAM	PWR_RAMPC0.RAMPDC5
SRAM4	0x20058000~0x2005FFFF 地址空间用RAM	PWR_RAMPC0.RAMPDC6
SRAMH	0x1FFE0000~0x1FFE7FFF 地址空间用RAM	PWR_RAMPC0.RAMPDC7
	0x1FFE8000~0x1FFEFFFF 地址空间用RAM	PWR_RAMPC0.RAMPDC8
	0x1FFF0000~0x1FFF7FFF 地址空间用RAM	PWR_RAMPC0.RAMPDC9
	0x1FFF8000~0x1FFFFFFF 地址空间用RAM	PWR_RAMPC0.RAMPDC10
Backup-SRAM	0x200F0000~0x200F0FFF	PWR_VBATCR. VBTRSD

	地址空间用RAM	
CAN_1 RAM	CAN单元1用RAM	PWR_PRAMLPC.PRAMDC0
CAN_2 RAM	CAN单元2用RAM	PWR_PRAMLPC.PRAMDC1
CACHERAM	Cache用RAM	PWR_PRAMLPC.PRAMDC2
USBFS RAM	USBFS用RAM	PWR_PRAMLPC.PRAMDC3
USBHS RAM	USBHS用RAM	PWR_PRAMLPC.PRAMDC4
ETHMAC TX RAM	ETHMAC TX用RAM	PWR_PRAMLPC.PRAMDC5
ETHMAC RX RAM	ETHMAC RX用RAM	PWR_PRAMLPC.PRAMDC6
SDIOC1 RAM	SDIOC1用RAM	PWR_PRAMLPC.PRAMDC7
SDIOC2 RAM	SDIOC2用RAM	PWR_PRAMLPC.PRAMDC8
NFCRAM	NFC用RAM	PWR_PRAMLPC.PRAMDC9

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
PWR_FPRC. FPRCB0	CMU_XTALCFGR,CMU_XTALSTBCR,CMU_XTALCR,CMU_XTALSTDCR, CMU_XTALSTDSR,CMU_HRCTRM,CMU_HRCCR,CMU_MRCTRM, CMU_MRCCR,CMU_PLLHCFGR,CMU_PLLHCR,CMU_PLLACFGR, CMU_PLLACR,CMU_OSCSTBSR,CMU_CKSWR,CMU_SCFGR, CMU_UsBCKCFGR,CMU_CANCKCFGR, CMU_TPIUCKCFGR,CMU_MCO1CFGR,CMU_MCO2CFGR, CMU_XTAL32CR,CMU_XTALC32CFGR,CMU_XTAL32NFR, CMU_LRCCR,CMU_LRCTRM,CMU_RTCLRCTRM
PWR_FPRC. FPRCB1	PWR_PWRC0，PWR_PWRC1，PWR_PWRC2，PWR_PWRC3， PWR_PWRC4,PWR_PDWKE0，PWR_PDWKE1， PWR_PDWKE2，PWR_PDWKES，PWR_PDWKF0，PWR_PDWKF1， CMU_PERICKSEL, CMU_I2SCKSEL, PWR_STPMCR，PWR_RAMPC0，PWR_RAMOPM，PWR_PRAMLPC， RMU_RSTF0,PWR_VBATRSTR,PWR_BATCR0,PWR_WKTC0,PWR_WKTC1, PWR_WKTC2, RMU_PRSTCR0
PWR_FPRC. FPRCB3	PWR_PVDCR0，PWR_PVDCR1，PWR_PVDFCR，PWR_PVDLCR， PWR_PVDICR， PWR_PVDDSR

保护寄存器位	被保护寄存器
PWR_FCG0PC.B0	PWR_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: 0x4004CC00

寄存器名	符号	偏移地址	位宽	复位值
电源控制寄存器0	PWR_PWRC0	0x00	8	0x00
电源模式控制寄存器1	PWR_PWRC1	0x04	8	0x00
电源模式控制寄存器2	PWR_PWRC2	0x08	8	0xFF
电源模式控制寄存器3	PWR_PWRC3	0x0C	8	0xFF
电源模式控制寄存器4	PWR_PWRC4	0x10	8	0x00
PVD控制寄存器0	PWR_PVDCR0	0x14	8	0x00
PVD控制寄存器1	PWR_PVDCR1	0x18	8	0x00
PVD滤波控制寄存器	PWR_PVDFCR	0x1C	8	0x11
PVD电平控制寄存器	PWR_PVDLCR	0x20	8	0x00
掉电唤醒使能寄存器0	PWR_PDWKE0	0x28	8	0x00
掉电唤醒使能寄存器1	PWR_PDWKE1	0x2C	8	0x00
掉电唤醒使能寄存器2	PWR_PDWKE2	0x30	8	0x00
掉电唤醒事件边沿选择寄存器	PWR_PDWKES	0x34	8	0x00
掉电唤醒标志寄存器0	PWR_PDWKF0	0x38	8	0x00
掉电唤醒标志寄存器1	PWR_PDWKF1	0x3C	8	0x00
RAM运行条件寄存器	PWR_RAMPC0	0xE0	32	0x00000000
RAM功耗控制寄存器	PWR_RAMOPM	0xE4	16	0x8043
外设RAM低功耗控制寄存器	PWR_PRAMLPC	0xE8	32	0x00000000
PVD中断控制寄存器	PWR_PVDICR	0xF0	8	0x00
PVD检测状态寄存器	PWR_PVDDSR	0xF4	8	0x11

BASE ADDR:0x40054000

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	PWR_STPMCR	0x0C	16	0x0000
功能保护控制寄存器	PWR_FPRC	0x3FE	8	0x00

BASE ADDR: 0x4004C400

寄存器名	符号	偏移地址	位宽	复位值

备份域复位寄存器	PWR_VBATRSTR	0x30	8	0x00
备份域控制寄存器	PWR_VBATCR	0x40	8	0x00
唤醒计时器控制寄存器0	PWR_WKTC0	0x50	8	0x00
唤醒计时器控制寄存器1	PWR_WKTC1	0x54	8	0x00
唤醒计时器控制寄存器2	PWR_WKTC2	0x58	8	0x00
备份寄存器000-127	PWR_BKR000	0x200	8	0x00
	PWR_BKR001	0x204	8	0x00
	PWR_BKR002	0x208	8	0x00
	PWR_BKR003	0x20C	8	0x00
	PWR_BKR004	0x210	8	0x00
	PWR_BKR005	0x214	8	0x00
	PWR_BKR006	0x218	8	0x00
	PWR_BKR007	0x21C	8	0x00
	PWR_BKR008	0x220	8	0x00
	PWR_BKR009	0x224	8	0x00
	PWR_BKR010	0x228	8	0x00
	PWR_BKR011	0x22C	8	0x00
	PWR_BKR012	0x230	8	0x00
	PWR_BKR013	0x234	8	0x00
	PWR_BKR014	0x238	8	0x00
	PWR_BKR015	0x23C	8	0x00
	PWR_BKR016	0x240	8	0x00
	PWR_BKR017	0x244	8	0x00
	PWR_BKR018	0x248	8	0x00
	PWR_BKR019	0x24C	8	0x00
	PWR_BKR020	0x250	8	0x00
	PWR_BKR021	0x254	8	0x00
	PWR_BKR022	0x258	8	0x00
	PWR_BKR023	0x25C	8	0x00
	PWR_BKR024	0x260	8	0x00

	PWR_BKR025	0x264	8	0x00
	PWR_BKR026	0x268	8	0x00
	PWR_BKR027	0x26C	8	0x00
	PWR_BKR028	0x270	8	0x00
	PWR_BKR029	0x274	8	0x00
	PWR_BKR030	0x278	8	0x00
	PWR_BKR031	0x27C	8	0x00
	PWR_BKR032	0x280	8	0x00
	PWR_BKR033	0x284	8	0x00
	PWR_BKR034	0x288	8	0x00
	PWR_BKR035	0x28C	8	0x00
	PWR_BKR036	0x290	8	0x00
	PWR_BKR037	0x294	8	0x00
	PWR_BKR038	0x298	8	0x00
	PWR_BKR039	0x29C	8	0x00
	PWR_BKR040	0x2A0	8	0x00
	PWR_BKR041	0x2A4	8	0x00
	PWR_BKR042	0x2A8	8	0x00
	PWR_BKR043	0x2AC	8	0x00
	PWR_BKR044	0x2B0	8	0x00
	PWR_BKR045	0x2B4	8	0x00
	PWR_BKR046	0x2B8	8	0x00
	PWR_BKR047	0x2BC	8	0x00
	PWR_BKR048	0x2C0	8	0x00
	PWR_BKR049	0x2C4	8	0x00
	PWR_BKR050	0x2C8	8	0x00
	PWR_BKR051	0x2CC	8	0x00
	PWR_BKR052	0x2D0	8	0x00
	PWR_BKR053	0x2D4	8	0x00
	PWR_BKR054	0x2D8	8	0x00

	PWR_BKR055	0x2DC	8	0x00
	PWR_BKR056	0x2E0	8	0x00
	PWR_BKR057	0x2E4	8	0x00
	PWR_BKR058	0x2E8	8	0x00
	PWR_BKR059	0x2EC	8	0x00
	PWR_BKR060	0x2F0	8	0x00
	PWR_BKR061	0x2F4	8	0x00
	PWR_BKR062	0x2F8	8	0x00
	PWR_BKR063	0x2FC	8	0x00
	PWR_BKR064	0x300	8	0x00
	PWR_BKR065	0x304	8	0x00
	PWR_BKR066	0x308	8	0x00
	PWR_BKR067	0x30C	8	0x00
	PWR_BKR068	0x310	8	0x00
	PWR_BKR069	0x314	8	0x00
	PWR_BKR070	0x318	8	0x00
	PWR_BKR071	0x31C	8	0x00
	PWR_BKR072	0x320	8	0x00
	PWR_BKR073	0x324	8	0x00
	PWR_BKR074	0x328	8	0x00
	PWR_BKR075	0x32C	8	0x00
	PWR_BKR076	0x330	8	0x00
	PWR_BKR077	0x334	8	0x00
	PWR_BKR078	0x338	8	0x00
	PWR_BKR079	0x33C	8	0x00
	PWR_BKR080	0x340	8	0x00
	PWR_BKR081	0x344	8	0x00
	PWR_BKR082	0x348	8	0x00
	PWR_BKR083	0x34C	8	0x00
	PWR_BKR084	0x350	8	0x00

	PWR_BKR085	0x354	8	0x00
	PWR_BKR086	0x358	8	0x00
	PWR_BKR087	0x35C	8	0x00
	PWR_BKR088	0x360	8	0x00
	PWR_BKR089	0x364	8	0x00
	PWR_BKR090	0x368	8	0x00
	PWR_BKR091	0x36C	8	0x00
	PWR_BKR092	0x370	8	0x00
	PWR_BKR093	0x374	8	0x00
	PWR_BKR094	0x378	8	0x00
	PWR_BKR095	0x37C	8	0x00
	PWR_BKR096	0x380	8	0x00
	PWR_BKR097	0x384	8	0x00
	PWR_BKR098	0x388	8	0x00
	PWR_BKR099	0x38C	8	0x00
	PWR_BKR100	0x390	8	0x00
	PWR_BKR101	0x394	8	0x00
	PWR_BKR102	0x398	8	0x00
	PWR_BKR103	0x39C	8	0x00
	PWR_BKR104	0x3A0	8	0x00
	PWR_BKR105	0x3A4	8	0x00
	PWR_BKR106	0x3A8	8	0x00
	PWR_BKR107	0x3AC	8	0x00
	PWR_BKR108	0x3B0	8	0x00
	PWR_BKR109	0x3B4	8	0x00
	PWR_BKR110	0x3B8	8	0x00
	PWR_BKR111	0x3BC	8	0x00
	PWR_BKR112	0x3C0	8	0x00
	PWR_BKR114	0x3C4	8	0x00
	PWR_BKR115	0x3C8	8	0x00

	PWR_BKR116	0x3CC	8	0x00
	PWR_BKR117	0x3D0	8	0x00
	PWR_BKR118	0x3D4	8	0x00
	PWR_BKR119	0x3D8	8	0x00
	PWR_BKR120	0x3DC	8	0x00
	PWR_BKR121	0x3E0	8	0x00
	PWR_BKR122	0x3E4	8	0x00
	PWR_BKR123	0x3E8	8	0x00
	PWR_BKR124	0x3EC	8	0x00
	PWR_BKR125	0x3F0	8	0x00
	PWR_BKR126	0x3F4	8	0x00
	PWR_BKR127	0x3F8	8	0x00

BASE ADDR:0x40048000

寄存器名	符号	偏移地址	位宽	复位值
功能时钟控制0	PWR_FCG0	0x00	32	0xFFFFFA0E
功能时钟控制1	PWR_FCG1	0x04	32	0xFFFFFFFF
功能时钟控制2	PWR_FCG2	0x08	32	0xFFFFFFFF
功能时钟控制3	PWR_FCG3	0x0C	32	0xFFFFFFFF
PWR_FCG0保护控制	PWR_FCG0PC	0x10	32	0x00000000

表 5-10 寄存器一览表

5.7.1 电源模式控制寄存器 0（PWR_PWRC0）

复位值：0x00

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

位	标记	位名	功能	读写
b7	PWDN	掉电模式控制位	0：掉电模式无效 1：掉电模式使能	R/W
b6	Reserved	-	读出时为“0”，写入时写“0”	R/W
b5-b4	IORTN[1:0]	掉电模式下IO保持控制	00：掉电模式中IO保持状态，掉电唤醒后硬件解除IO保持状态 01：掉电模式中IO保持状态，掉电唤醒后将IORTN[1:0]设为00b，解除IO保持状态 1x：掉电模式中及掉电唤醒后IO为高阻	R/W
b3-b2	Reserved	-	读出时为“0”，写入时写“0”	R/W
b1-b0	PDMDS[1:0]	掉电模式选择控制	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（PWR_PWRC1）

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
STPDAS[1:0]		-		PDTS	VHRCSD	VPLLSD[1:0]	

位	标记	位名	功能	读写
b7-b6	STPDAS[1:0]	STOP模式LDO驱动选择	00：高速模式模式进入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 10/01：禁止设定。	R/W
b5-b4	Reserved	-	读出时为“0”，写入时写“0”	R/W
b3	PDTS	掉电唤醒时间控制位	0：当VCAP_1/VCAP_2总电容是2个0.1uF时或者1个0.22uF时，进入掉电模式之前，PDTS位必须清零。 1：VCAP_1/VCAP_2总电容是2个0.047uF时或者1个0.1uF时,，进入掉电模式之前，PDTS位必须置位。	R/W
b2	VHRCSD	HRC电源关闭	0：HRC电源使能 1：HRC电源关闭 在HRC不使用时，置位VHRCSD后关闭HRC用电源，进一步降低功耗。VHRCSD清零后，需要等待25uS后再启动HRC模块	R/W
b1-b0	VPLLSD[1:0]	PLL电源关闭	00：PLL电源使能 11：PLL电源关闭 01: 设定禁止 10：设定禁止 在PLLA关闭, PLLH关闭并等待50us后，设置VPLLSD=11后关闭PLL用电源，进一步降低功耗。设置VPLLSD=00后，需要等待25uS后再启动PLLA或者PLLH模块。	R/W

5.7.3 电源模式控制寄存器 2（PWR_PWRC2）

复位值：0xFF

b7	b6	b5	b4	b3	b2	b1	b0
-		DVS[1:0]		DDAS[11:8]			

位	标记	位名	功能	读写
b7-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	10：选择超低速动作电压 11：选择高速动作模式电压 00/01：设定禁止	R/W
b3-b0	DDAS[11:8]	电源驱动选择	与PWR_PWRC3中的DDAS[7:0]联合，用以调整驱动能力 DDAS[11:0]= 0xfff：高速动作模式 0x000：低速驱动模式 其他： 设定禁止	R/W

5.7.4 电源模式控制寄存器 3（PWR_PWRC3）

复位值：0xFF

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

位	标记	位名	功能	读写
b7-b0	DDAS[7:0]	电源驱动选择	与PWR_PWRC2中的DDAS[11:8]联合，用以调整驱动能力 DDAS[11:0]= 0xfff：高速动作模式 0x000：超低速驱动模式 其他： 设定禁止	R/W

5.7.5 电源模式控制寄存器 4（PWR_PWRC4）

复位值:0x00

b7	b6	b5	b4	b3	b2	b1	b0
ADBUFE	ADBUFS	-	VBATMON	-	-	VBATME	VBATREFSEL

位	标记	位名	功能	读写
b7	ADBUFE	内部电压采样使能	使用AD采样芯片内部电压时，需要设置本位为1 0：无效 1：有效	R/W
b6	ADBUFS	采样内部电压选择位	0：内部基准电压 1：VBAT的1/2分压	R/W
b5	Reserved	-	读出时为“0”，写入时写“0”	R/W
b4	VBATMON	VBAT电压检测状态	0：VBAT > VBATREF 1：VBAT < VBATREF	R
b3-b2	Reserved	-	读出时为“0”，写入时写“0”，禁止写入“1”	R/W
b1	VBATME	VBAT检测使能	0：检测无效 1：检测使能	R/W
b0	VBATREFSEL	VBAT检测用参考电压(VBATREF)选择	0：1.8V 1：2.1V	R/W

5.7.6 掉电唤醒使能寄存器 0(PWR_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

5.7.7 掉电唤醒使能寄存器 1(PWR_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

5.7.8 掉电唤醒使能寄存器 2(PWR_PDWKE2)

复位值：0x00

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

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

5.7.9 掉电唤醒事件边沿选择寄存器(PWR_PDWKES)

复位值：0x00

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

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

5.7.10 掉电唤醒标志寄存器 0(PWR_PDWKF0)

复位值：0x00

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

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

5.7.11 掉电唤醒标志寄存器 1(PWR_PDWKF1)

复位值：0x00

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

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

5.7.12 功能时钟控制 0(PWR_FCG0)

复位值:0xFFFFFA0E

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DCU8	DCU7	DCU6	DCU5	DCU4	DCU3	DCU2	DCU1	CRC	TRNG	HASH	AES	MAU	CTC	AOS	FCM
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DMA2	DMA1	KEY	-	-	SRAMB	-	-	SRAM4	SRAM3	SRAM2	SRAM1	-	-	-	SRAMH

位	标记	位名	功能	读写
b31	DCU8	DCU8功能控制	0：数字计算单元DCU8功能使能 1：数字计算单元DCU8功能无效	R/W
b30	DCU7	DCU7功能控制	0：数字计算单元DCU7功能使能 1：数字计算单元DCU7功能无效	R/W
b29	DCU6	DCU6功能控制	0：数字计算单元DCU6功能使能 1：数字计算单元DCU6功能无效	R/W
b28	DCU5	DCU5功能控制	0：数字计算单元DCU5功能使能 1：数字计算单元DCU5功能无效	R/W
b27	DCU4	DCU4功能控制	0：数字计算单元DCU4功能使能 1：数字计算单元DCU4功能无效	R/W
b26	DCU3	DCU3功能控制	0：数字计算单元DCU3功能使能 1：数字计算单元DCU3功能无效	R/W
b25	DCU2	DCU2功能控制	0：数字计算单元DCU2功能使能 1：数字计算单元DCU2功能无效	R/W
b24	DCU1	DCU1功能控制	0：数字计算单元DCU1功能使能 1：数字计算单元DCU1功能无效	R/W
b23	CRC	CRC功能控制	0：CRC功能使能 1：CRC功能无效	R/W
b22	TRNG	TRNG功能控制	0：加密协处理模块CPM中的真随机发生器TRNG功能使能 1：加密协处理模块CPM中的真随机发生器TRNG功能无效	R/W
b21	HASH	HASH功能控制	0：加密协处理模块CPM中的安全散列算法模块HASH功能使能 1：加密协处理模块CPM中的安全散列算法模块HASH功能无效	R/W

b20	AES	AES功能控制	0：加密协处理模块CPM中的加解密算法处理器AES功能使能 1：加密协处理模块CPM中的加解密算法处理器AES功能无效	R/W
b19	MAU	MAU功能控制	0：数学运算单元MAU功能使能 1：数学运算单元MAU功能无效	R/W
b18	CTC	CTC功能控制	0：内部时钟校准器CTC功能使能 1：内部时钟校准器CTC功能无效	R/W
b17	AOS	AOS功能控制	0：自动运行系统AOS功能使能 1：自动运行系统AOS功能无效	R/W
b16	FCM	FCM功能控制	0：时钟控制器CMU中的时钟频率测量模块FCM功能使能 1：时钟控制器CMU中的时钟频率测量模块FCM功能无效	R/W
b15	DMA2	DMA2功能控制	0：DMA控制器DMA2功能使能 1：DMA控制器DMA2功能无效	R/W
b14	DMA1	DMA1功能控制	0：DMA控制器DMA1功能使能 1：DMA控制器DMA1功能无效	R/W
b13	KEY	KEYSCAN功能控制	0：键盘扫描控制模块KEYSCAN功能使能 1：键盘扫描控制模块KEYSCAN功能无效	R/W
b12-b11	Reserved	-	读出时为“1”，写入时写“1”	R/W
b10	SRAMB	BACKUP-SRAM功能控制	0：BACKUP-SRAM功能使能 1：BACKUP-SRAM功能无效	R/W
b9-b8	Reserved	-	读出时为“10”，写入时写“10”	R/W
b7	SRAM4	SRAM4功能控制	0：SRAM4（0x20058000~0x2005FFFF）功能使能 1：SRAM4（0x20058000~0x2005FFFF）功能无效	R/W
b6	SRAM3	SRAM3功能控制	0：SRAM3（0x20040000~0x20057FFF）功能使能 1：SRAM3（0x20040000~0x20057FFF）功能无效	R/W
b5	SRAM2	SRAM2功能控制	0：SRAM2（0x20020000~0x2003FFFF）功能使能 1：SRAM2（0x20020000~0x2003FFFF）功能无效	R/W
b4	SRAM1	SRAM1功能控制	0：SRAM1（0x20000000~0x2001FFFF）功能使能 1：SRAM1（0x20000000~0x2001FFFF）功能无效	R/W
b3-b1	Reserved	-	读出时为“1”，写入时写“1”	R/W
b0	SRAMH	SRAMH功能控制	0：SRAMH（0x1FFE0000~0x1FFFFFFF）功能使能 1：SRAMH（0x1FFE0000~0x1FFFFFFF）功能使能	R/W

5.7.13 功能时钟控制 1(PWR_FCG1)

复位值：0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	FMAC4	FMAC3	FMAC2	FMAC1
b23	b22	b21	b20	b19	b18	b17	b16
USBHS	USBFS	SPI6	SPI5	SPI4	SPI3	SPI2	SPI1
b15	b14	b13	b12	b11	b10	b9	b8
I2S4	I2S3	I2S2	I2S1	SDIOC2	SDIOC1	I2C6	I2C5
b7	b6	b5	b4	b3	b2	b1	b0
I2C4	I2C3	I2C2	I2C1	QSPI	ETHMAC	CAN2	CAN1

位	标记	位名	功能	读写
b31-b28	Reserved	-	读出时为“1”，写入时写“1”	R/W
b27	FMAC4	FMAC4功能控制	0：滤波数学加速器FMAC单元4功能使能 1：滤波数学加速器FMAC单元4功能无效	R/W
b26	FMAC3	FMAC3功能控制	0：滤波数学加速器FMAC单元3功能使能 1：滤波数学加速器FMAC单元3功能无效	R/W
b25	FMAC2	FMAC2功能控制	0：滤波数学加速器FMAC单元2功能使能 1：滤波数学加速器FMAC单元2功能无效	R/W
b24	FMAC1	FMAC1功能控制	0：滤波数学加速器FMAC单元1功能使能 1：滤波数学加速器FMAC单元1功能无效	R/W
b23	USBHS	USBHS功能控制	0：USB2.0高速模块USBHS功能使能 1：USB2.0高速模块USBHS功能无效	R/W
b22	USBFS	USBFS功能控制	0：USB2.0全速模块USBFS功能使能 1：USB2.0全速模块USBFS功能无效	R/W
b21	SPI6	SPI6功能控制	0：串行外设接口SPI单元6功能使能 1：串行外设接口SPI单元6功能无效	R/W
b20	SPI5	SPI5功能控制	0：串行外设接口SPI单元5功能使能 1：串行外设接口SPI单元5功能无效	R/W
b19	SPI4	SPI4功能控制	0：串行外设接口SPI单元4功能使能 1：串行外设接口SPI单元4功能无效	R/W
b18	SPI3	SPI3功能控制	0：串行外设接口SPI单元3功能使能 1：串行外设接口SPI单元3功能无效	R/W
b17	SPI2	SPI2功能控制	0：串行外设接口SPI单元2功能使能 1：串行外设接口SPI单元2功能无效	R/W

b16	SPI1	SPI1功能控制	0：串行外设接口SPI单元1功能使能 1：串行外设接口SPI单元1功能无效	R/W
b15	I2S4	I2S4功能控制	0：集成电路内置音频总线模块I2S单元4功能使能 1：集成电路内置音频总线模块I2S单元4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I2S单元3功能使能 1：集成电路内置音频总线模块I2S单元3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I2S单元2功能使能 1：集成电路内置音频总线模块I2S单元2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I2S单元1功能使能 1：集成电路内置音频总线模块I2S单元1功能无效	R/W
b11	SDIOC2	SDIOC2功能控制	0：SDIO控制器SDIOC单元2功能使能 1：SDIO控制器SDIOC单元2功能无效	R/W
b10	SDIOC1	SDIOC1功能控制	0：SDIO控制器SDIOC单元1功能使能 1：SDIO控制器SDIOC单元1功能无效	R/W
b9	I2C6	I2C6功能控制	0：集成电路总线I2C单元6功能使能 1：集成电路总线I2C单元6功能无效	R/W
b8	I2C5	I2C5功能控制	0：集成电路总线I2C单元5功能使能 1：集成电路总线I2C单元5功能无效	R/W
b7	I2C4	I2C4功能控制	0：集成电路总线I2C单元4功能使能 1：集成电路总线I2C单元4功能无效	R/W
b6	I2C3	I2C3功能控制	0：集成电路总线I2C单元3功能使能 1：集成电路总线I2C单元3功能无效	R/W
b5	I2C2	I2C2功能控制	0：集成电路总线I2C单元2功能使能 1：集成电路总线I2C单元2功能无效	R/W
b4	I2C1	I2C1功能控制	0：集成电路总线I2C单元1功能使能 1：集成电路总线I2C单元1功能无效	R/W
b3	QSPI	QSPI功能控制	0：四线式串行外设接口QSPI功能使能 1：四线式串行外设接口QSPI功能无效	R/W
b2	ETHMAC	ETHMAC功能控制	0：以太网MAC控制器ETHMAC功能使能 1：以太网MAC控制器ETHMAC功能无效	R/W
b1	CAN2	CAN2功能控制	0：CAN单元2功能使能 1：CAN单元2功能无效	R/W
b0	CAN1	CAN1功能控制	0：CAN单元1功能使能 1：CAN单元1功能无效	R/W

5.7.14 功能时钟控制 2(PWR_FCG2)

复位值：0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TIMERA12	TIMERA11	TIMERA10	TIMERA9	TIMERA8	TIMERA7	TIMERA6	TIMERA5	TIMERA4	TIMERA3	TIMERA2	TIMERA1	TIMER2_4	TIMER2_3	TIMER2_2	TIMER2_1
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EMB	-	TIMER0_2	TIMER0_1	HRPWM	TIMER4_3	TIMER4_2	TIMER4_1	TIMER6_8	TIMER6_7	TIMER6_6	TIMER6_5	TIMER6_4	TIMER6_3	TIMER6_2	TIMER6_1

位	标记	位名	功能	读写
b31	TIMERA12	TIMERA12功能控制	0：TimerA单元12功能使能 1：TimerA单元12功能无效	R/W
b30	TIMERA11	TIMERA11功能控制	0：TimerA单元11功能使能 1：TimerA单元11功能无效	R/W
b29	TIMERA10	TIMERA10功能控制	0：TimerA单元10功能使能 1：TimerA单元10功能无效	R/W
b28	TIMERA9	TIMERA9功能控制	0：TimerA单元9功能使能 1：TimerA单元9功能无效	R/W
b27	TIMERA8	TIMERA8功能控制	0：TimerA单元8功能使能 1：TimerA单元8功能无效	R/W
b26	TIMERA7	TIMERA7功能控制	0：TimerA单元7功能使能 1：TimerA单元7功能无效	R/W
b25	TIMERA6	TIMERA6功能控制	0：TimerA单元6功能使能 1：TimerA单元6功能无效	R/W
b24	TIMERA5	TIMERA5功能控制	0：TimerA单元5功能使能 1：TimerA单元5功能无效	R/W
b23	TIMERA4	TIMERA4功能控制	0：TimerA单元4功能使能 1：TimerA单元4功能无效	R/W
b22	TIMERA3	TIMERA3功能控制	0：TimerA单元3功能使能 1：TimerA单元3功能无效	R/W
b21	TIMERA2	TIMERA2功能控制	0：TimerA单元2功能使能 1：TimerA单元2功能无效	R/W
b20	TIMERA1	TIMERA1功能控制	0：TimerA单元1功能使能 1：TimerA单元1功能无效	R/W
b19	TIMER2_4	TIMER2_4功能控制	0：TIMER2单元4功能使能 1：TIMER2单元4功能无效	R/W

b18	TIMER2_3	TIMER2_3功能控制	0：TIMER2单元3功能使能 1：TIMER2单元3功能无效	R/W
b17	TIMER2_2	TIMER2_2功能控制	0：TIMER2单元2功能使能 1：TIMER2单元2功能无效	R/W
b16	TIMER2_1	TIMER2_1功能控制	0：TIMER2单元1功能使能 1：TIMER2单元1功能无效	R/W
b15	EMB	EMB功能控制	0：紧急刹车模块EMB功能使能 1：紧急刹车模块EMB功能无效	R/W
b14	Reserved	-	读出时为“1”，写入时写“1”	R/W
b13	TIMER0_2	TIMER0_2功能控制	0：TIMER0_2功能使能 1：TIMER0_2功能无效	R/W
b12	TIMER0_1	TIMER0_1功能控制	0：TIMER0_1功能使能 1：TIMER0_1功能无效	R/W
b11	HRWPM	HRPWM功能控制	0：HRPWM功能使能 1：HRPWM功能无效	R/W
b10	TIMER4_3	TIMER4_3功能控制	0：TIMER4_3功能使能 1：TIMER4_3功能无效	R/W
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	TIMER6_8	TIMER6_8功能控制	0：TIMER6单元8功能使能 1：TIMER6单元8功能无效	R/W
b6	TIMER6_7	TIMER6_7功能控制	0：TIMER6单元7功能使能 1：TIMER6单元7功能无效	R/W
b5	TIMER6_6	TIMER6_6功能控制	0：TIMER6单元6功能使能 1：TIMER6单元6功能无效	R/W
b4	TIMER6_5	TIMER6_5功能控制	0：TIMER6单元5功能使能 1：TIMER6单元5功能无效	R/W
b3	TIMER6_4	TIMER6_4功能控制	0：TIMER6单元4功能使能 1：TIMER6单元4功能无效	R/W
b2	TIMER6_3	TIMER6_3功能控制	0：TIMER6单元3功能使能 1：TIMER6单元3功能无效	R/W
b1	TIMER6_2	TIMER6_2功能控制	0：TIMER6单元2功能使能 1：TIMER6单元2功能无效	R/W
b0	TIMER6_1	TIMER6_1功能控制	0：TIMER6单元1功能使能 1：TIMER6单元1功能无效	R/W

5.7.15 功能时钟控制 3(PWR_FCG3)

复位值：0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24
-	-	USART10	USART9	USART8	USART7	USART6	USART5
b23	b22	b21	b20	b19	b18	b17	b16
USART4	USART3	USART2	USART1	-	EXMC_NFC	EXMC_DMC	EXMC_SMC
b15	b14	b13	b12	b11	b10	b9	b8
DVP	-	-	OTS	-	-	CMP2	CMP1
b7	b6	b5	b4	b3	b2	b1	b0
-	-	DAC2	DAC1	CMBIAS	ADC3	ADC2	ADC1

位	标记	位名	功能	读写
b31-b30	Reserved	-	读出时为“1”，写入时写“1”	R/W
b29	USART10	USART10功能控制	0：通用同步异步收发器USART单元10功能使能 1：通用同步异步收发器USART单元10功能无效	R/W
b28	USART9	USART9功能控制	0：通用同步异步收发器USART单元9功能使能 1：通用同步异步收发器USART单元9功能无效	R/W
b27	USART8	USART8功能控制	0：通用同步异步收发器USART单元8功能使能 1：通用同步异步收发器USART单元8功能无效	R/W
b26	USART7	USART7功能控制	0：通用同步异步收发器USART单元7功能使能 1：通用同步异步收发器USART单元7功能无效	R/W
b25	USART6	USART6功能控制	0：通用同步异步收发器USART单元6功能使能 1：通用同步异步收发器USART单元6功能无效	R/W
b24	USART5	USART5功能控制	0：通用同步异步收发器USART单元5功能使能 1：通用同步异步收发器USART单元5功能无效	R/W
b23	USART4	USART4功能控制	0：通用同步异步收发器USART单元4功能使能 1：通用同步异步收发器USART单元4功能无效	R/W
b22	USART3	USART3功能控制	0：通用同步异步收发器USART单元3功能使能 1：通用同步异步收发器USART单元3功能无效	R/W
b21	USART2	USART2功能控制	0：通用同步异步收发器USART单元2功能使能 1：通用同步异步收发器USART单元2功能无效	R/W
b20	USART1	USART1功能控制	0：通用同步异步收发器USART单元1功能使能 1：通用同步异步收发器USART单元1功能无效	R/W
b19	Reserved	-	读出时为“1”，写入时写“1”	R/W
b18	EXMC_NFC	EXMC_NFC功能控制	0：外部存储控制器(EXMC)的NAND FLASH控制器功能有效	R/W

			1：外部存储控制器(EXMC)的NAND FLASH控制器功能无效	
b17	EXMC_DMC	EXMC_DMC功能控制	0：外部存储控制器(EXMC)的DMC控制器功能有效 1：外部存储控制器(EXMC)的DMC控制器功能无效	R/W
b16	EXMC_SMC	EXMC_SMC功能控制	0：外部存储控制器(EXMC)的SMC控制器功能有效 1：外部存储控制器(EXMC)的SMC控制器功能无效	R/W
b15	DVP	DVP功能控制	0：数字视频接口DVP功能有效 1：数字视频接口DVP功能无效	R/W
b14-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	
b11-b10	Reserved	-	读出时为“1”，写入时写“1”	R/W
b9	CMP2	CMP2功能控制	0：电压比较器CMP单元2功能使能 1：电压比较器CMP单元2功能无效	R/W
b8	CMP1	CMP1功能控制	0：电压比较器CMP单元1功能使能 1：电压比较器CMP单元1功能无效	R/W
b7-b6	Reserved	-	读出时为“1”，写入时写“1”	R/W
b5	DAC2	DAC2功能控制	0：数模转换器DAC单元2功能使能 1：数模转换器DAC单元2功能无效	R/W
b4	DAC1	DAC1功能控制	0：数模转换器DAC单元1功能使能 1：数模转换器DAC单元1功能无效	R/W
b3	CMBIAS	CMP/PGA/SH基准电流源控制	0：CMP/PGA/SH基准电流源有效 1：CMP/PGA/SH基准电流源无效	R/W
b2	ADC3	ADC3功能控制	0：模数转换模块ADC单元3功能使能 1：模数转换模块ADC单元3功能无效	R/W
b1	ADC2	ADC2功能控制	0：模数转换模块ADC单元2功能使能 1：模数转换模块ADC单元2功能无效	R/W
b0	ADC1	ADC1功能控制	0：模数转换模块ADC单元1功能使能 1：模数转换模块ADC单元1功能无效	R/W

5.7.16 PWR_FCG0 保护控制(PWR_FCG0PC)

复位值：0x00000000

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

位	标记	位名	功能	读写
b31~b16	FCG0PCWE[15:0]	PWR_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	PWR_FCG0写使能控制位 0：PWR_FCG0写无效 1：PWR_FCG0写使能	R/W

5.7.17 功能保护控制寄存器(PWR_FPRC)

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PWR_FPRCWE[7:0]								FPRCB7	FPRCB6	FPRCB5	FPRCB4	FPRCB3	FPRCB2	FPRCB1	FPRCB0

位	标记	位名	功能	读写
b15~b8	PWR_FPRCWE[7:0]	PWR_FPRC寄存器写使能	写入0xA5h的同时能够更新PWR_FPRC值，否则对低8位写入值无效。读出时为0x00。	R/W
b7	FPRCB7	FPRC位7	预留，读出时为“0”，写入时写“0”	R/W
b6	FPRCB6	FPRC位6	预留，读出时为“0”，写入时写“0”	R/W
b5	FPRCB5	FPRC位5	预留，读出时为“0”，写入时写“0”	R/W
b4	FPRCB4	FPRC位4	预留，读出时为“0”，写入时写“0”	R/W
b3	FPRCB3	FPRC位3	保护寄存器位，保护对象参考表7-9 0：写保护 1：写使能	R/W
b2	FPRCB2	FPRC位2	预留，读出时为“0”，写入时写“0”	R/W
b1	FPRCB1	FPRC位1	保护寄存器位，保护对象参考表7-9 0：写保护 1：写使能	R/W
b0	FPRCB0	FPRC位0	保护寄存器位，保护对象参考表7-9 0：写保护 1：写使能	R/W

5.7.18 STOP 模式控制寄存器(PWR_STPMCR)

复位值：0x0000

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

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

5.7.19 RAM 功耗控制寄存器 0(PWR_RAMPC0)

复位值：0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	RAMDC10	RAMDC9	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-b11	Reserved	-	读出时为“0”，写入时写“0”	R/W
b10	RAMPDC10	RAM掉电控制位10	0：0x 1FFF8000~0x1FFFFFFF空间RAM动作 1：0x 1FFF8000~0x1FFFFFFF空间RAM掉电	R/W
b9	RAMPDC9	RAM掉电控制位9	0：0x 1FFF0000~0x1FFF7FFF空间RAM动作 1：0x 1FFF0000~0x1FFF7FFF RAM掉电	R/W
b8	RAMPDC8	RAM掉电控制位8	0：0x1FFE8000~0x1FFEFFFF空间RAM动作 1：0x 1FFE8000~0x1FFFEFFF RAM掉电	R/W
b7	RAMPDC7	RAM掉电控制位7	0：0x1FFE0000~0x1FFE7FFF空间RAM动作 1：0x 1FFE0000~0x1FFE7FFF空间RAM掉电	R/W
b6	RAMPDC6	RAM掉电控制位6	0：0x20058000~0x2005FFFF空间RAM动作 1：0x20058000~0x2005FFFF空间RAM掉电	R/W
b5	RAMPDC5	RAM掉电控制位5	0：0x20050000~0x20057FFF空间RAM动作 1：0x20050000~0x20057FFF空间RAM掉电	R/W
b4	RAMPDC4	RAM掉电控制位4	0：0x20040000~0x2004FFFF空间RAM动作 1：0x20040000~0x2004FFFF空间RAM掉电	R/W
b3	RAMPDC3	RAM掉电控制位3	0：0x20030000~0x2003FFFF空间RAM动作 1：0x20030000~0x2003FFFF空间RAM掉电	R/W
b2	RAMPDC2	RAM掉电控制位2	0：0x20020000~0x2002FFFF空间RAM动作 1：0x20020000~0x2002FFFF空间RAM掉电	R/W
b1	RAMPDC1	RAM掉电控制位1	0：0x20010000~0x2001FFFF空间RAM动作 1：0x20010000~0x2001FFFF空间RAM掉电	R/W
b0	RAMPDC0	RAM掉电控制位0	0：0x20000000~0x2000FFFF空间RAM动作 1：0x20000000~0x2000FFFF空间RAM掉电	R/W

5.7.20 RAM 运行条件寄存器(PWR_RAMOPM)

复位值：0x8043

15	14	13	12	11	10	9	8
PWR_RAMOPM[15:8]							
7	6	5	4	3	2	1	0
PWR_RAMOPM[7:0]							

位	标记	位名	功能	读写
b15-b0	PWR_RAMOPM[15:0]	RAM动作模式选择位	芯片工作在高速运行模式时， PWR_RAMOPM设置成0x8043 芯片工作在超低速模式动作时， PWR_RAMOPM设置成0x9062	R/W

5.7.21 外设 RAM 低功耗控制寄存器 (PWR_PRAMLPC)

复位值：0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	PRAMPDC9	PRAMPDC8
7	6	5	4	3	2	1	0
PRAMPDC7	PRAMPDC6	PRAMPDC5	PRAMPDC4	PRAMPDC3	PRAMPDC2	PRAMPDC1	PRAMPDC0

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	PRAMPDC9	外设RAM掉电控制位9	NFC_RAM掉电控制 0：不掉电 1：掉电	R/W
b8	PRAMPDC8	外设RAM掉电控制位8	SDIO2_RAM掉电控制 0：不掉电 1：掉电	R/W
b7	PRAMPDC7	外设RAM掉电控制位7	SDIO1_RAM掉电控制 0：不掉电 1：掉电	R/W
b6	PRAMPDC6	外设RAM掉电控制位6	ETHER RX RAM掉电控制 0：不掉电 1：掉电	R/W
b5	PRAMPDC5	外设RAM掉电控制位5	ETHER TX RAM掉电控制 0：不掉电 1：掉电	R/W
b4	PRAMPDC4	外设RAM掉电控制位4	USBHS RAM掉电控制 0：不掉电 1：掉电	R/W
b3	PRAMPDC3	外设RAM掉电控制位3	USBFS RAM掉电控制 0：不掉电 1：掉电	R/W
b2	PRAMPDC2	外设RAM掉电控制位2	CACHE_RAM掉电控制	R/W

			0：不掉电 1：掉电	
b1	PRAMPDC1	外设RAM掉电控制位1	CAN_2 RAM掉电控制 0：不掉电 1：掉电	R/W
b0	PRAMPDC0	外设RAM掉电控制位0	CAN_1 RAM掉电控制 0：不掉电 1：掉电	R/W

5.7.22 PVD 控制寄存器 0(PWR_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-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

5.7.23 PVD 控制寄存器 1(PWR_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：在VCC变化满足检测条件时产生PVD2中断 1：在下降过程中经过VPVD2产生PVD2复位 注：当PVD1IRS位为“1”或者PVD2IRS位为“1”时，不能进入掉电模式，要进入PD模式时，必须将PVD1IRS位置“0”并且将PVD2IRS位置“0”	R/W
b4	PVD2IRE	PVD2中断复位使能	0：禁止 1：允许 注：请在PVD2EN位为“1”且PVD2CMPOE位为“1”状态下将PVD2INTRE位写“1”	R/W
b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2	PVD1CMPOE	PVD1比较结果输出使能	0：禁止输出比较器1的比较结果 1：允许输出比较器1的比较结果	R/W
b1	PVD1IRS	PVD1中断复位选择	0：在VCC变化满足检测条件时产时产生PVD1中断 1：在下降过程中经过VPVD1产生PVD1复位 注1：当PVD1IRS位为“1”或者PVD2IRS位为“1”时，不能进入掉电模式，要进入PD模式时，必须将PVD1IRS位置“0”并且将PVD2IRS位置“0”	R/W
b0	PVD1IRE	PVD1中断复位使能	0：禁止 1：允许 注：请在PVD1EN位为“1”且PVD1CMPOE位为“1”状态下将PVD2INTRE位写“1”	R/W

5.7.24 PVD 滤波控制寄存器(PWR_PVDFCR)

复位值：0x11

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

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

5.7.25 PVD 电平控制寄存器(PWR_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[2:0]	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PWR_PVDCR0.EXVCCINEN=1时有效，其它情况请不要设定该值)	R/W
b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2~b0	PVD1LVL[2:0]	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	R/W

5.7.26 PVD 中断控制寄存器(PWR_PVDICR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	PVD2EDGS[1:0]		PVD2NMIS	-	PVD1EDGS[1:0]		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	PVD2EDGS[1:0]	PVD2检测条件选择	00：当检测到VCC＜VPVD2(下降)时 01：当检测到VCC≥VPVD2(上升)时 10：当检测到VCC＜VPVD2(下降)时或者当检测到VCC≥VPVD2(上升)时 11：请勿设定	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	PVD1EDGS[1:0]	PVD1检测条件选择	00：当检测到VCC＜VPVD1(下降)时 01：当检测到VCC≥VPVD1(上升)时 10：当检测到VCC＜VPVD1(下降)时或者当检测到VCC≥VPVD1(上升)时 11：请勿设定	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

5.7.27 PVD 检测状态寄存器(PWR_PVDDSR)

复位值：0x11

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

5.7.28 备份域复位寄存器(PWR_VBATRSTR)

复位值：0x00

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

位	标记	位名	功能	读写
b7-0	RST[7:0]	备份域复位寄存器	写入0xA5复位备份域	W

5.7.29 备份域控制寄存器(PWR_VBATCR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
CSDIS	-	-	-	VBATDIVMONE	RAMPDF	RAMVALID	VBTRSD

位	标记	位名	功能	读写
b7	CSDIS	VBAT域电流源无效控制	0：VBAT域电流源有效 1：VBAT域电流源无效，只有当备份RAM、XTAL32、XTAL32NF都不使用的时候，这个位才可以设置成1	R/W
b6-b4	Reserved	-	读出时为“0”，写入时写“0”	R/W
b3	VBATDIVMONE	VBAT分压电路使能位	0：分压电路无效 1：分压电路使能	R/W
b2	RAMPDF	RAM掉电标志位	0：Backup-RAM没有掉电 1：Backup-RAM掉电	R
b1	RAMVALID	RAM有效标志	0：Backup-RAM不可读写 1：Backup-RAM可以读写	R
b0	VBTRSD	VBAT RLDO关闭控制位	0：VBAT RLDO启动 1：VBAT RLDO关闭	R/W

5.7.30 VBAT 备份寄存器 0~127(PWR_BKR000~PWR_BKR127)

复位值：0x00

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

N=0~127

位	标记	位名	功能	读写
b7-b0	bkrNb[7:0]	备份寄存器	备份寄存器	R/W

5.7.31 唤醒计时器控制寄存器 0 (PWR_WKTC0)

复位值：0x00

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

位	标记	位名	功能	读写
b7-0	WKTMCMP[7:0]	WKTM计数器的比较值	WKTM计数器的比较值	R/W

5.7.32 唤醒计时器控制寄存器 1 (PWR_WKTC1)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	WKTMCMP[11:8]			

位	标记	位名	功能	读写
b7-b4	Reserved	-	读出时为“0”，写入时写“0”	R/W
b3-0	WKTMCMP[11:8]	WKTM计数器的比较值	WKTM计数器的比较值	R/W

5.7.33 唤醒计时器控制寄存器 2 (PWR_WKTC2)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
WKTCE	WKCKS[1:0]		WKOVF	-	-	-	-

位	标记	位名	功能	读写
b7	WKTCE	WKTM使能	0： WKTM停止 1： WKTM计数 WKTM计数时清零WKTCE位会将计数器清零，计数器停止动作。	R/W
b6-b5	WKCKS[1:0]	WKTM时钟选择	00：64Hz时钟 01：XTAL32 10：RTCLRC 11：预留	R/W
b4	WKOVF	计时器比较结果标志	0：计数器与WKTMCMP值不一致 1：计数器与WKTMCMP值一致 写0清除本标志位。	R/W
b3-b0	Reserved	-	读出时为“0”，写入时写“0”	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 0x00000400H~0x0000045FH 把数据加载到初始化配置寄存器。地址 0x00000410H~0x0000045FH 为预约功能，请写入全 1 保证芯片正常动作。FLASH 引导交换有效，且 OTP 不使能时，该区域位于 FLASH 块 1 扇区 0；否则该区域存在 FLASH 块 0 扇区 0。用户可通过编程或擦除扇区 0 来修改初始化配置(Initial ConfiG)寄存器。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:0x00000400

寄存器名	符号	偏移地址	位宽	复位值
初始化配置寄存器0	ICG0	0x000	32	不定
初始化配置寄存器1	ICG1	0x004	32	不定
初始化配置寄存器2	ICG2	0x008	32	不定
初始化配置寄存器3	ICG3	0x00C	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
-	-	-	WDTSLPOFF	WDTWDPT]3:0]				WDTCKS[3:0]				WDTPERI[1:0]		WDTITS	WDTAUTS

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	SWDTSLPOFF	SWDTWDPT]3:0]				SWDTCKS[3:0]				SWDTPERI[1:0]		SWDTITS	SWDTAUTS

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
-	-	-	-	-	-	-	-	-	-	-	-	-	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~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 2(ICG2)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	BGO1M[23:16]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BGO1M[15:0]															

位	标记	位名	功能	读写
b31~b24	Reserved	-	功能预留位	R
b23~b0	BGO1M[23:0]	1Mbytes产品FLASH配置	对FLASH空间为1Mbytes 产品，根据BGO1M[23:0]设定，可实现BGO功能。 数据为0x4b4b4b4b时，地址0x0000_0000~0x0007_FFFF空间由FLASH块0提供；地址0x0008_0000~0x000F_FFFF空间由FLASH块1提供。 数据非0x4b4b4b4b时，地址0x0000_0000~0x000F_FFFF空间全由FLASH0提供。 在2Mbytes产品中，该位无效。	R

6.2.4 初始化配置寄存器 3(ICG3)

复位值：不定

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

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

7 嵌入式 FLASH（EFM）

7.1 简介

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

7.2 主要特性

- 两块独立 FLASH 构成 dual bank，可实现 BGO（BackGroud Operation）功能
- 134Kbytes 的 OTP 空间
- ICODE 总线 16Bytes 预取指
- 两个独立缓存区：ICODE 总线缓存空间 4Kbytes(256x128)；DCODE 总线缓存空间 1Kbytes(64x128)
- 支持引导交换功能
- 支持安全保护*1

  *1：关于 FLASH 安全保护的具体规格，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 2Mbytes，由两块 1Mbytes 的 FLASH 构成，共 256 个扇区，每个扇区为 8Kbytes。块 0 中扇区 0~扇区 15 为可配置为 OTP 区域。
- OTP(One Time Program)区域共 134KBytes，其中 128Kbytes 配置在块 0 地址 0x0000_0000~0x0001_FFFF，6Kbytes 配置在地址 0x0300_0000~0x0300_17FF。地址 0x0300_1800~0x0300_1AD7 为 OTP 数据锁存区。
- 128 位宽数据读取，读缓存 128 位宽缓冲，加速代码执行。
- 编程单位为 4bytes,擦除单位为 8Kbytes。

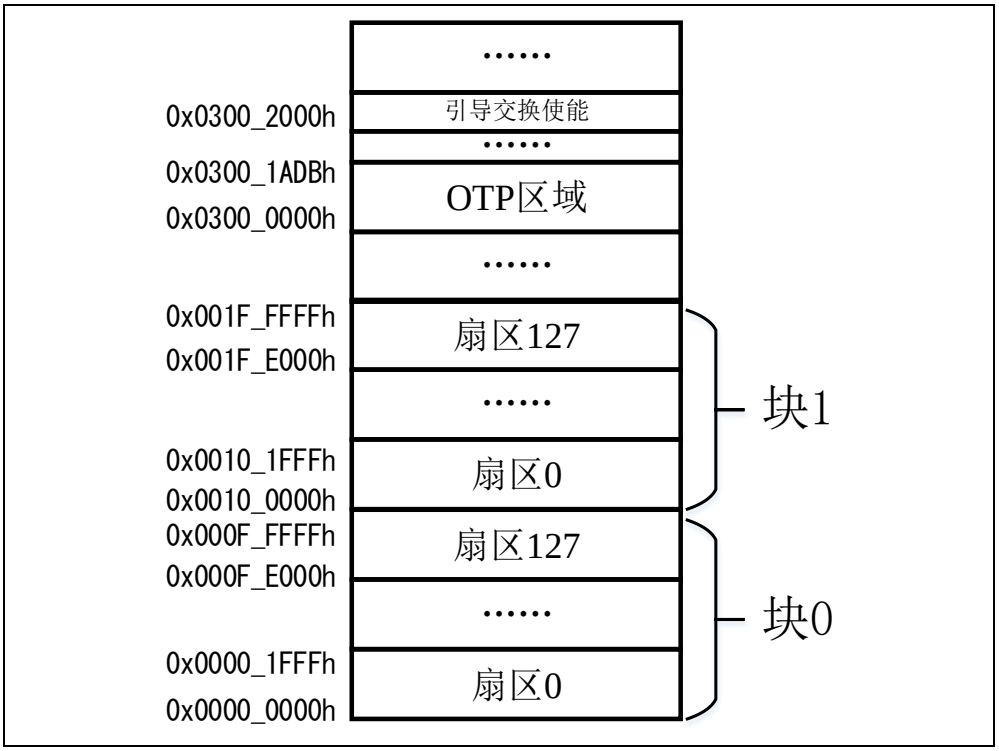


图 7-1 FLASH 地址分配

OTP 区域 0x0300_0000~0x0300_1ADB 及引导交换使能地址 0x0300_2000 分别单独享有 1 个扇区，可进行编程，擦除，但全擦除禁止；这两扇区物理上从属于 FLASH 块 0，擦写标志位跟随 FLASH 块 0，但对 FLASH 块 0 全擦除这两扇区数据不变。

7.4 读接口

7.4.1 CPU 时钟和 FLASH 读取时间之间的关系

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

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

CPU 频率提高步骤：

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

CPU 频率降低步骤：

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

CPU时钟频率(hclk)	读等待周期设定
200MHz<Fhclk≤240MHz	FLWT[3:0]=4’b0101
160MHz<Fhclk≤200MHz	FLWT[3:0]=4’b0100
120MHz<Fhclk≤160MHz	FLWT[3:0]=4’b0011
80MHz<Fhclk≤120MHz	FLWT[3:0]=4’b0010
40MHz<Fhclk≤80MHz	FLWT[3:0]=4’b0001
Fhclk≤40MHz	FLWT[3:0]=4’b0000

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 5Kbytes 空间做为缓存存储器，可以有效地减少因指令跳转而产生的时间损耗。通过 EFM_FRMC 寄存器中的缓存使能 (ICACHE/DCACHE) 位置 1，来使缓存功能有效。每当出现指令或数据缺失（即请求的指令或数据未存在于当前使用的指令行或缓存存储器中）时，系统会将新读取的数据行(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		缓存使能	
	缓冲命中	缓冲不命中	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
N(N>2)	1	N+1	1	N+1

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

7.6.1 解锁 EFM_KEY1 寄存器

复位解除后，FLASH 编程，擦除模式寄存器（EFM_FWMC）处于写禁止状态，需要先解除 FLASH 访问保护寄存器（EFM_FAPRT），然后再解除 EFM_KEY1 的保护。以下步骤用于解除上述保护。

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

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

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

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

7.6.2 写保护功能

FLASH 每个扇区有 1 个写保护位，存在于写保护寄存器 EFM_F0/1NWPRTx(x=0~3)中。每个写保护寄存器存在 1 个写保护锁定位(WLCOK[7:0])，写保护锁定位一旦设定为 1，写保护寄存器位只能从写许可设定为写禁止。当 FLASH 扇区设定为写禁止时，对该扇区地址进行编程和擦除操作将不发生，并发生错误标志(EFM_FSR.PRTWERR0/1=1)。

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

7.6.3 单次编程无回读功能

单编程无回读模式设定步骤如下：

1) 解除 FLASH 的寄存器写保护。(EFM_FAPRT 先写 0x0123，再写 0x3210)
2) 解除 EFM_KEY1 锁定。(EFM_KEY1 先写 0x01234567，再写 0xFEDCBA98)
3) 设定单次编程模式。(EFM_FWMC.PEMODE[2:0]=001)
4) 解除写保护。(EFM_ F0/1NWPRTx(x=0~3)对应位设定为 1)
5) 对编程地址写入 32 位数据
6) 等待 FLASH 处于空闲状态。(EFM_FSR.RDY0/1=1)
7) 读出编程地址值判断是否和写入值一致；

   一致，表示编程成功，不一致，表示该 FLASH 地址已遭破坏，永久废弃。
8) 清除编程结束标志位。(EFM_FSR.OPTEND0/1)

   对已锁存的 OTP 地址发行单编程不回读写操作，编程不成功，标志位 EFM_FSR.OTPWERR0 置位。

7.6.4 单编程回读功能

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

单编程回读模式设定步骤如下：

1) 解除 FLASH 的寄存器写保护。(EFM_FAPRT 先写 0x0123，再写 0x3210)
2) 解除 EFM_KEY1 锁定。(EFM_KEY1 先写 0x01234567，再写 0xFEDCBA98)
3) 设定单次编程回读模式。(EFM_FWMC.PEMODE[2:0]=010)
4) 解除写保护。(EFM_ F0/1NWPRTx(x=0~3)对应位设定为 1)
5) 对编程地址写入 32 位数据
6) 等待 FLASH 处于空闲状态。(EFM_FSR.RDY0/1=1)
7) 判断编程自读取结果标志位。(EFM_FSR.MISMTCH0/1)

   如为 0，则表示编程成功；为 1 表示该 FLASH 地址已遭破坏，永久废弃。
8) 清除编程结束标志位。(EFM_FSR.OPTEND0/1)

   对已锁存的 OTP 地址发行单编程回读写操作，编程不成功，标志位

EFM_FSR.OTPWERR0 置位。

7.6.5 连续编程功能

当连续对 FLASH 地址进行编程时，推荐使用连续编程模式。连续编程模式比单编程模式可以节省时间 50%以上。连续编程模式时，对 FLASH 地址写命令间隔不能超过 16us。

连续编程操作设定步骤如下：

1) 解除 FLASH 的寄存器写保护。(EFM_FAPRT 先写 0x0123，再写 0x3210)
2) 解除 EFM_KEY1 锁定。(EFM_KEY1 先写 0x01234567，再写 0xFEDCBA98)
3) 设定连续编程模式。(EFM_FWMC.PEMOD[2:0]=011)
4) 解除写保护。(EFM_ F0/1NWPRTx(x=0~3)对应位设定为 1)
5) 对编程地址写 32 位数据。（编程地址需要和执行程序分属不同 FLASH 块或者程序在 FLASH 以外区域运行）
6) 等待操作结束标志位(EFM_FSR.OPTEND0/1)置位。
7) 清除操作结束标志位，直至读到 EFM_FSR.OPTEND0/1 为 0。
8) 重复 5)，6)，7)直至所有数据写完。
9) 修改连续编程模式为非连续编程模式。(EFM_FWMC.PEMOD[2:0]=000)
10) 等待 FLASH 处于空闲状态。(EFM_FSR.RDY0/1=1)
11) 读取编程地址并判断是否和写入值一致。
12) 一致，表示编程成功，不一致，表示该 FLASH 地址已遭破坏，永久废弃。

对已锁存的 OTP 地址发行连续编程写操作，编程不成功，标志位 EFM_FSR.OTPWERR0 置位。

7.6.6 擦除功能

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

1) 解除 FLASH 的寄存器写保护(EFM_FAPRT 先写 0x0123，再写 0x3210)。
2) 解除 EFM_KEY1 锁定。(EFM_KEY1 先写 0x01234567，再写 0xFEDCBA98)
3) 设定擦除模式(EFM_FWMC.PEMOD[2:0]=100)。
4) 解除写保护。(EFM_ F0/1NWPRTx(x=0~3)对应位设定为 1)

5) 对需要擦除扇区内的任意地址(地址需以 4 对齐)写入 32 位任意值。
6) 等待 FLASH 处于空闲状态。(EFM_FSR.RDY0/1=1)
7) 清除擦除结束标志位。(EFM_FSR.OPTEND0/1)

   对已锁存的 OTP 地址发行擦除操作，擦除不成功，OTP 区域数据保留，标志位 EFM_FSR.OTPWERR0 置位。

7.6.7 全擦除功能

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

1) 解除 FLASH 的寄存器写保护(EFM_FAPRT 先写 0x0123，再写 0x3210)。
2) 解除 EFM_KEY1 锁定。(EFM_KEY1 先写 0x01234567，再写 0xFEDCBA98)
3) 单块 FLASH 全擦除时，设定 EFM_FWMC.PEMOD[2:0]=101。
4) 两块 FLASH 同时全擦除时，设定 EFM_FWMC.PEMOD[2:0]=110。
5) 解除写保护。(EFM_ F0/1NWPRTx(x=0~3)FLASH 块对应所有位都设定为 1)
6) 单块 FLASH 全擦除：对需要擦除 FLASH 的任意地址(地址需以 4 对齐)写入 32 位任意值。
7) 两块 FLASH 全擦除：对 FLASH0 或 FLASH1 地址(地址需以 4 对齐)写入 32 位任意值。

   全擦动作硬件会自动同步到另一块 FLASH。
8) 等待 FLASH 处于空闲状态。(EFM_FSR.RDY0/1=1)
9) 清除擦除结束标志位。(EFM_FSR.OPTEND0/1)

   OTP 使能后，对 FLASH 块 0 地址发行单块全擦除写操作，全擦除不成功，FLASH0 数据保留，标志位 EFM_FSR.OTPWERR0 置位；对 FLASH 块 0 地址发行两块全擦除写操作，FLASH0 数据保留，FLASH1 数据被全擦，标志位 EFM_FSR.OTPWERR0 置位。

7.6.8 D-BUS 读保护功能

对地址 0x0000_0000h~0x0001_FFFFh(128Kbytes)空间提供了 D-BUS 读保护。使能 D-BUS 保护，需设定 ICG3 寄存器 DBUSPRT3[15:0]为 0x4450，复位重启后，功能使

能。CPU-PC 指针在此 128K 空间时（即 CPU 在此区域执行程序），CPU 和 DMAC 可以正常访问；当 CPU-PC 指针在此 128K 空间以外时（即 CPU 在 128K 以外区域执行程序），CPU 和 DMAC 对 128K 空间内进行 D-BUS 读则被禁止，并返回总线错误。D-BUS 是 CPU 的数据访问总线，包括操作数访问，堆栈数据访问，一般数据访问等。

D-BUS 读保护使能时，用户需要注意以下事项。

1）单个独立程序空间不能跨越 0x0002_0000h 的边界。

2）CPU 在执行 128K 以外区域程序时，不能读取配置在 128K 空间上的中断向量表。

3）CPU 在执行 128K 以外区域程序时，只有通过跳转指令和子程序调用可以回到 128K 区域执行程序。

7.6.9 BGO 功能

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

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

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

3) 设定编程，擦除模式。(EFM_FWMC.PEMODE[2:0]=001，010，011，100，101)

4) 解除写保护。(目标地址对应扇区位(EFM_F0/1NWPRTx (x=0~3))设定为 1)

5) 设定编程，擦除期间总线释放（EFM_FWMC.BUSHLDCTL=1）

6) 对编程地址写入 32 位数据。（编程地址和执行程序分属不同块 FLASH）

   若执行程序在 RAM 上，可以对两块 FLASH 同时编程，(全)擦除。

7) 等待擦写结束标志位置位。(EFM_FSR.OPTEND0/1=1)

7.6.10 中断

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

1. PE 错误中断 EFM_PEERR：

   置位：

   - 对已锁存的 OTP 地址发出编程，擦除，全擦除操作(OTPWERR0=1)。

- OTP 功能启用后，对 0x0300_0000~0x0300_01AB 空间进行擦除，全擦除操作(OTPWERR0=1)。
- 对写保护扇区发出编程，擦除，全擦除操作(PRTWERR0/1=1)。
- 编程地址非以 4 对齐或者数据大小非 32 位(PGSZERR0/1=1)。
- 单编程回读模式时，编程地址硬件自动读取值与写入值不一致(MISMTCH0/1=1)。

清零：

寄存器 EFM_FSCLR 对应标志清除位写 1，状态位清零。

2. FLASH 读写冲突中断 EFM_COLERR：

置位：

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

清零：

寄存器 EFM_FSCLR 对应清除位置 1，状态位清零。

3. 操作结束中断 EFM_OPTEND：

置位：

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

清零：

寄存器 EFM_FSCLR 对应清除位置 1，状态位清零。

7.7 一次性可编程字节（OTP）

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

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

表 7-3 OTP 地址分布表

启用 OTP 功能，需要预先对地址 0x0300_1AD8~0x0300_1ADB 编程写入 32 位包含 0 数据（推荐写入 32 位全 0）。当 OTP 块锁存地址数据包含 0 时，则锁存地址对应的块空间无法再次编程，擦除和全擦除。OTP 的 sector0~15 与 FLASH 块 0 地址 0x0000_0000~0x0001_FFFF 共享 128Kbytes 物理空间；其空间的锁存地址数据若为全 1，此区域为普通 FLASH 空间，可多次编程和擦除；若对应的锁存地址数据包含 0，则对应的扇区为只读，FLASH 块 0 将无法进行全擦除。对 OTP 锁存地址编程需要解锁 EFM_KEY2 寄存器(EFM_KEY2 先写 0x10325476，再写 0xEFCDAB89)。如对 EFM_KEY2 寄存器写入错误的序列，则会发生引发一次总线错误，并且 EFM_KEY2 寄存器会锁定

直至发生复位。

对 OTP 区域操作请按以下步骤进行：

1) 解除 FLASH 的寄存器写保护。(EFM_FAPRT 先写 0x0123，再写 0x3210)。
2) 解除 EFM_KEY1 锁定。(EFM_KEY1 先写 0x01234567，再写 0xFEDCBA98)。
3) 设定编程模式。(EFM_FWMC.PEMODE[2:0]=001，010，011)。
4) 根据需要对 OTP 地址写入数据。
5) 解除 EFM_KEY2 锁定。(EFM_KEY2 先写 0x10325476，再写 0xEFCDAB89)。
6) 对 OTP 使能地址 0x0300_1AD8~0x0300_1ADB 的 32bit 任一 bit 写 0。（推荐写入 32bit 全 0）
7) 对 OTP 锁存地址任一 bit 写入 0。（推荐写入 32bit 全 0）

芯片复位启动后，硬件会自动加载 OTP 锁存地址数据到电路中，因此，若 OTP 使能地址 0x0300_1AD8~0x0300_1ADB 已编程写入过 0，则需省略 step 6。

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

OTP 锁存地址写完后，如需回归到 EFM_KEY2 锁定状态，可对 EFM_FWMC.KEY2LOCK 写入 1。

7.8 引导交换

对地址 FLASH 地址（0x0300_2000）编程写入数据 0x5A5A5A，可实现引导交换功能。当 OTP 功能未启用(0x0300_1AD8~0x0300_1ADB 全 1)，编程写入复位后，FLASH 块 0 和 FLASH 块 1 全体地址互换；此时，OTP 使能地址无法进行编程(地址 0x0300_1AD8 无法进行编程)。可通过 EFM_FSWP.FSWP 位查询引导交换是否使能。当 OTP 功能启用后(0x0300_1AD8~0x0300_1ADB 数据非全 1)，编程写入复位后，FLASH 块 0 和 FLASH 块 1 地址部分互换。此时，EFM_FSWP.FSWP 为 0，通过读取地址 0x0300_2000 数据来判断地址交换功能，变换规则见下图 7-2，图 7-3。

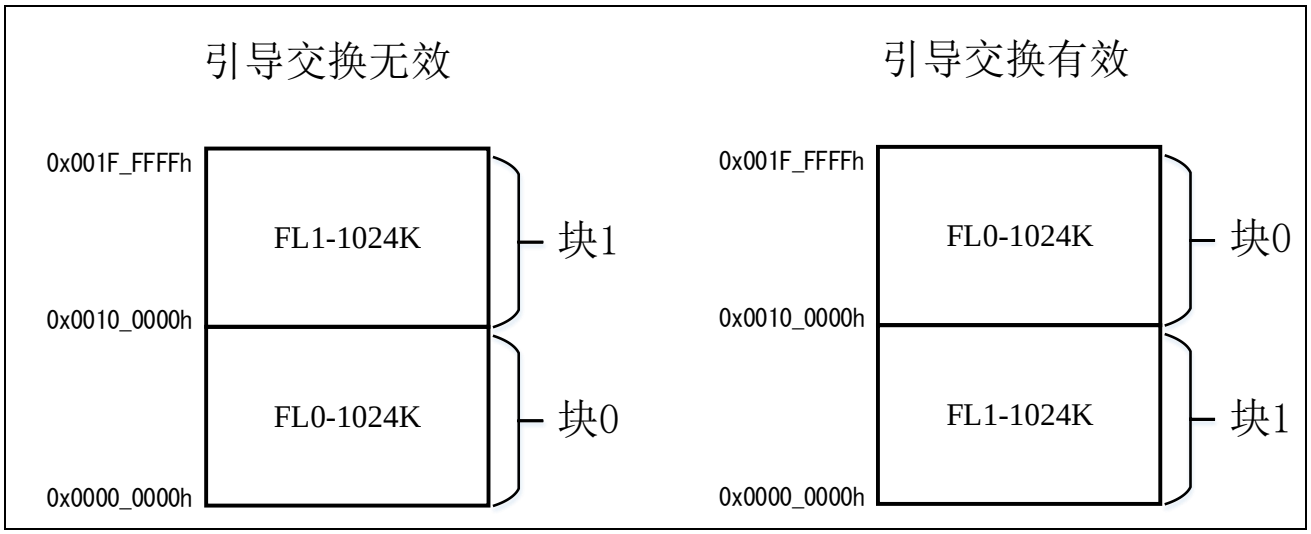


图 7-2 OTP 功能未使能时，FLASH 地址分配

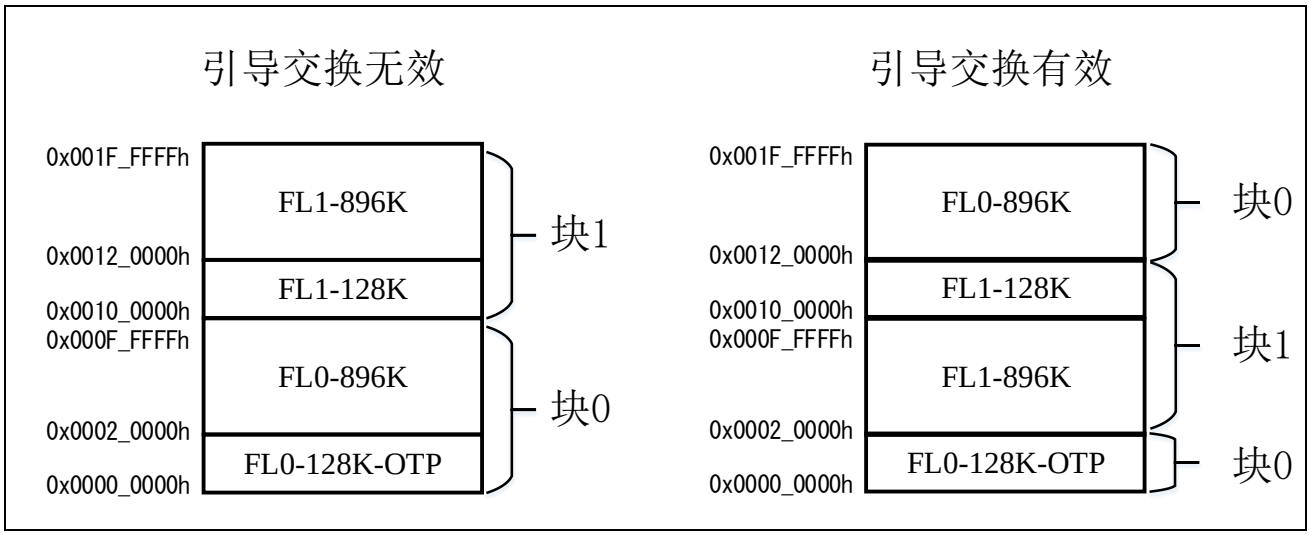


图 7-3 OTP 功能使能时，FLASH 地址分配

以下以 OTP 未启用的引导交换为例，介绍该功能的使用示意图 7-4。

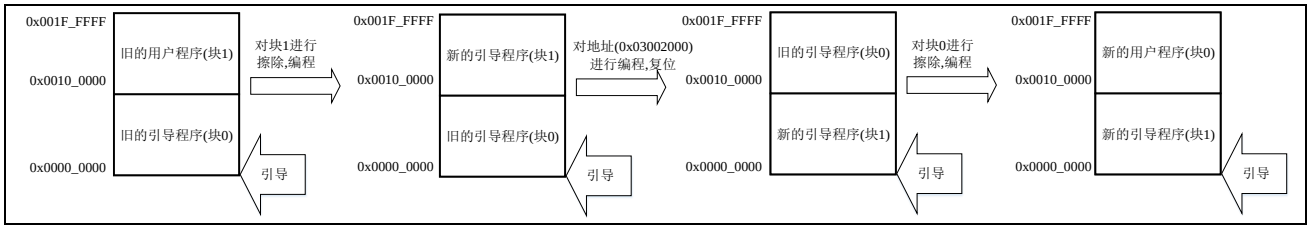


图 7-4 启动引导交换 1

当用户需要再次升级启动程序，由于地址 0x0300_2000 已经被编程过了，再次升级则需要对地址 0x0300_2000 进行擦除。操作流程如图 7-5。

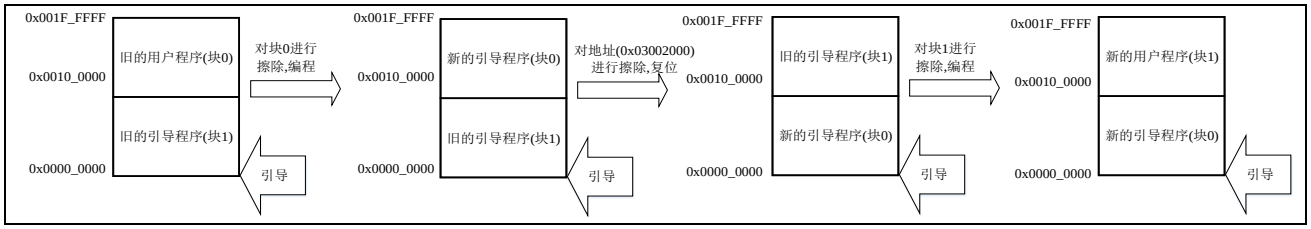


图 7-5 启动引导交换 2

7.9 寄存器说明

EFM_BASE_ADDR：0x40010400

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	0x0000h	32	0x0000_0000
FLASH密钥1寄存器	EFM_KEY1	0x0004h	32	0x0000_0000
FLASH密钥2寄存器	EFM_KEY2	0x0008h	32	0x0000_0000
FLASH停止寄存器	EFM_FSTP	0x0014h	32	0x0000_0000
FLASH读模式寄存器	EFM_FRMC	0x0018h	32	0x0000_0000
FLASH擦写模式寄存器	EFM_FWMC	0x001Ch	32	0x0003_0000
FLASH状态寄存器	EFM_FSR	0x0020h	32	0x0100_0100
FLASH状态清除寄存器	EFM_FSCLR	0x0024h	32	0x0000_0000
FLASH中断许可寄存器	EFM_FITE	0x0028h	32	0x0000_0000
FLASH引导交换状态寄存器	EFM_FSWP	0x002Ch	32	不定
CHIPID寄存器	EFM_CHIPID	0x0040h	32	0x4844_04A0
MCUuniqueID寄存器0	FEM_UQID0	0x0050h	32	不定
MCUuniqueID寄存器1	EFM_UQID1	0x0054h	32	不定
MCUuniqueID寄存器2	EFM_UQID2	0x0058h	32	不定
FLASH写保护锁存寄存器	EFM_WLOCK	0x0180h	32	0x0000_0000
FLASH0写保护寄存器0	EFM_F0NWPRT0	0x0190h	32	0x0000_0000
FLASH0写保护寄存器1	EFM_F0NWPRT1	0x0194h	32	0x0000_0000
FLASH0写保护寄存器2	EFM_F0NWPRT2	0x0198h	32	0x0000_0000
FLASH0写保护寄存器3	EFM_F0NWPRT3	0x019Ch	32	0x0000_0000
FLASH1写保护寄存器0	EFM_F1NWPRT0	0x01A0h	32	0x0000_0000
FLASH1写保护寄存器1	EFM_F1NWPRT1	0x01A4h	32	0x0000_0000
FLASH1写保护寄存器2	EFM_F1NWPRT2	0x01A8h	32	0x0000_0000
FLASH1写保护寄存器3	EFM_F1NWPRT3	0x01ACh	32	0x0000_0000

表 7-4 寄存器一览表

7.9.1 访问写保护寄存器(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/W
b15-b0	FAPRT[15:0]	EFM寄存器写保护	EFM寄存器访问保护寄存器。 解除方法：对FAPRT先写”16位数据0x0123”再写”16位数据0x3210”。 在解除保护状态下，写入任意数据，EFM寄存器再次进入保护状态。 EFM寄存器访问保护有效时，该寄存器读出值为0x00000000。 EFM寄存器访问保护无效时，该寄存器读出值为0x00000001。	R/W

7.9.2 FLASH 密钥 1 寄存器(EFM_KEY1)

复位值：0x0000_0000

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

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

7.9.3 FLASH 密钥 2 寄存器(EFM_KEY2)

复位值：0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
KEY2[31:16]-															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
KEY2[15:0]															

位	标记	位名	功能	读写
b31-b0	KEY2[31:0]	密钥2寄存器	对OTP锁存地址的写保护寄存器。 对EFM_KEY2先写0x10325476，再写0xEFCDAB89， 解除对OTP锁存地址的写保护。 该寄存器读出值为0x00000000。	R/W

7.9.4 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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	F1STP	F0STP

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	F1STP	FLASH1停止模式控制	0：FLASH1活动状态 1：FLASH1处于停止模式 当寄存器位由1设为0后，请在确认FSR.RDY1位为1后，进行FLASH1访问。	R/W
b0	F0STP	FLASH0停止模式控制	0：FLASH0活动状态 1：FLASH0处于停止模式 当寄存器位由1设为0后，请在确认FSR.RDY0位为1后，进行FLASH0访问。	R/W

7.9.5 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	PREFE	DCACHE	ICACHE
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	LVM	-	-	-	-	FLWT[3:0]			

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

7.9.6 擦写模式寄存器(EFM_FWMC)

复位值：0x0003_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	KEY2LOCK	KEY1LOCK

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

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

7.9.7 状态寄存器(EFM_FSR)

复位值：0x0100_0100

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	RDY1	-	-	COLERR1	OPTEND1	MISMTCH1	PGSZERR1	PRTWERR1	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	RDY0	-	-	COLERR0	OPTEND0	MISMTCH0	PGSZERR0	PRTWERR0	OTPWERR0

位	标记	位名	功能	读写
b31~b25	Reserved	-	读出时为“0”，写入时写“0”	R/W
b24	RDY1	FLASH块1忙/空闲状态	0：FLASH块1忙状态 1：FLASH块1空闲状态	R
b23~b22	Reserved	-	读出时为“0”，写入时写“0”	R/W
b21	COLERR1	FLASH块1读写冲突错误标志位	置位条件： RDY1未置位情况下，对FLASH1读写操作。 连续编程模式下，对FLASH1写不置位。 清零条件： EFM_FSCLR对应清零位写1。	R
b20	OPTEND1	FLASH块1空间编程，擦除，全擦除操作结束标志位	置位条件： 编程/擦除/全擦除操作结束。 清零条件： EFM_FSCLR对应清零位写1。	R
b19	MISMTCH1	FLASH块1空间单编程回读值不一致标志位	置位条件： 单编程回读模式，FLASH1编程后读数据和写入值不一致。 清零条件： EFM_FSCLR对应清零位写1	R
b18	PGSZERR1	FLASH块1空间编程地址和大小不对齐标志位	置位条件： 编程地址非以4对齐，或者发生非32位写操作。 清零条件： EFM_FSCLR对应清零位写1	R
b17	PRTWERR1	FLASH块1写保护错误标志位	置位条件： 对写保护扇区进行编程，擦除，全擦除操作。	R

			清零条件： EFM_FSCLR对应清零位写1	
b16~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8	RDY0	FLASH块0忙/空闲状态	0：FLASH块0忙状态 1：FLASH块0空闲状态	R
b7~b6	Reserved	-	读出时为“0”，写入时写“0”	R/W
b5	COLERR0	FLASH块0读写冲突错误标志位	置位条件： RDY0未置位情况下，对FLASH0读写操作。 连续编程模式下，对FLASH0写不置位。 清零条件： EFM_FSCLR对应清零位写1。	R
b4	OPTEND0	FLASH块0空间编程，擦除，全擦除操作结束标志位	置位条件： 编程/擦除/全擦除操作结束。 清零条件： EFM_FSCLR对应清零位写1。	R
b3	MISMTCH0	FLASH块0空间单编程回读值不一致标志位	置位条件： 单编程回读模式，FLASH0编程后读数据和写入值不一致。 清零条件： EFM_FSCLR对应清零位写1	R
b2	PGSZERR0	FLASH块0编程地址和大小不对齐标志位	置位条件： 编程地址非以4对齐，或者发生非32位写操作。 清零条件： EFM_FSCLR对应清零位写1	R
b1	PRTWERR0	FLASH块0写保护地址错误标志位	置位条件： 对写保护扇区进行编程，擦除，全擦除操作。 清零条件： EFM_FSCLR对应清零位写1	R
b0	OTPWERR0	FLASH块0的OTP擦写错误标志位	置位条件： 对OTP锁存区域进行编程，擦除，全擦除操作。 清零条件： EFM_FSCLR对应清零位写1	R

7.9.8 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
-	-	-	-	-	-	-	-	-	-	COLERRCLR1	OPTENDCLR1	MISMTCHCLR1	PGSZERRCLR1	PRTWERRCLR1	-

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	COLERRCLR0	OPTENDCLR0	MISMTCHCLR0	PGSZERRCLR0	PRTWERRCLR0	OTPWERRCLR0

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

b2	PGSZERRCLR0	清除编程地址和大小不对齐标志位	0：不发生清除动作 1：清除FSR.PGSZERR0位 该位读时，始终为0。	R/W
b1	PEPRTERRCLR0	清除写保护错误标志位	0：不发生清除动作 1：清除FSR.PRTWERR0位 该位读时，始终为0。	R/W
b0	OTPWERRCLR0	清除OTP擦写错误标志位	0：不发生清除动作 1：清除FSR.OTPWERR0位 该位读时，始终为0。	R/W

7.9.9 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
-	-	-	-	-	-	-	-	-	-	-	-	-	COLERRITE	OPTENDITE	PEERRITE

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

7.9.10 引导交换状态寄存器(EFM_FSWP)

复位值：不定

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

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

位	标记	位名	功能	读写
b31-b1	Reserved	-	读出时为“0”，写入时写“0”	R/W
b0	FSWP	扇区0和扇区1地址交换位	0：FLASH块0和块1地址不交换 复位后，CPU从FLASH块0启动。 1：块0和块1地址交换 复位后，CPU从FLASH块1启动。 寄存器初始值由FLASH 0x03002000~0x03002003的值决定，其数据为0x5A5A5A5A时，复位启动后，初值为1，否则为0。	R

7.9.11 华大芯片专属标志寄存器(EFM_CHIPID)

复位值：0x4844_04A0

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

位	标记	位名	功能	读写
b31-b0	CHIPID[31:0]	华大芯片专属标志	CHIPID[31:16]：HD的对应的ASCII码 CHIPID[15:0]：04A0，产品型名 该寄存器只读。	R

7.9.12 UNIQUE ID 寄存器(EFM_UQID0)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
UQID0[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UQID0[15:0]															

位	标记	位名	功能	读写
b31-b0	UQID0[31:0]	唯一码	芯片唯一码	R

7.9.13 UNIQUE ID 寄存器(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

7.9.14 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]															

位	标记	位名	功能	读写
b31-b0	UQID2[31:0]	唯一码	芯片唯一码	R

7.9.15 FLASH 写保护锁定寄存器(EFM_WLOCK)

复位值：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
-	-	-	-	-	-	-	-	WLOCK[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	WLOCK[7]	F1NWPRT3寄存器锁定位	0：F1NWPRT3寄存器控制的扇区写禁止未lockup 1：F1NWPRT3寄存器控制的扇区写禁止lockup 该位一旦设定为1，只能通过复位才能恢复成0。	R/W
b6	WLOCK[6]	F1NWPRT2寄存器锁定位	0：F1NWPRT2寄存器控制的扇区写禁止未lockup 1：F1NWPRT2寄存器控制的扇区写禁止lockup 该位一旦设定为1，只能通过复位才能恢复成0。	R/W
b5	WLOCK[5]	F1NWPRT1寄存器锁定位	0：F1NWPRT1寄存器控制的扇区写禁止未lockup 1：F1NWPRT1寄存器控制的扇区写禁止lockup 该位一旦设定为1，只能通过复位才能恢复成0。	R/W
b4	WLOCK[4]	F1NWPRT0寄存器锁定位	0：F1NWPRT0寄存器控制的扇区写禁止未lockup 1：F1NWPRT0寄存器控制的扇区写禁止lockup 该位一旦设定为1，只能通过复位才能恢复成0。	R/W
b3	WLOCK[3]	F0NWPRT3寄存器锁定位	0：F0NWPRT3寄存器控制的扇区写禁止未lockup 1：F0NWPRT3寄存器控制的扇区写禁止lockup 该位一旦设定为1，只能通过复位才能恢复成0。	R/W
b2	WLOCK[2]	F0NWPRT2寄存器锁定位	0：F0NWPRT2寄存器控制的扇区写禁止未lockup 1：F0NWPRT2寄存器控制的扇区写禁止lockup 该位一旦设定为1，只能通过复位才能恢复成0。	R/W
b1	WLOCK[1]	F0NWPRT1寄存器锁定位	0：F0NWPRT1寄存器控制的扇区写禁止未lockup 1：F0NWPRT1寄存器控制的扇区写禁止lockup 该位一旦设定为1，只能通过复位才能恢复成0。	R/W
b0	WLOCK[0]	F0NWPRT0寄存器锁定位	0：F0NWPRT0寄存器控制的扇区写禁止未lockup 1：F0NWPRT0寄存器控制的扇区写禁止lockup 该位一旦设定为1，只能通过复位才能恢复成0。	R/W

7.9.16 FLASH0 写保护寄存器 0(EFM_F0NWPRT0)

复位值：0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
F0NWPRT31	F0NWPRT30	F0NWPRT29	F0WPRT28	F0NWPRT27	F0NWPRT26	F0NWPRT25	F0NWPRT24	F0NWPRT23	F0NWPRT22	F0NWPRT21	F0NWPRT20	F0NWPRT19	F0NWPRT18	F0NWPRT17	F0NWPRT16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
F0NWPRT15	F0NWPRT14	F0NWPRT13	F0NWPRT12	F0NWPRT11	F0NWPRT10	F0NWPRT9	F0NWPRT8	F0NWPRT7	F0NWPRT6	F0NWPRT5	F0NWPRT4	F0NWPRT3	F0NWPRT2	F0NWPRT1	F0NWPRT0

位	标记	位名	功能	读写
b31	F0NWPRT31	FLASH块0扇区31的写许可位	0：FLASH块0扇区31的写禁止 1：FLASH块0扇区31的写许可 WLOCK[0]=0时，该位写0，写1许可。 WLOCK[0]=1时，该位只能0，写1无效。	R/W
b30	F0NWPRT30	FLASH块0扇区30的写保护位	0：FLASH块0扇区30的写禁止 1：FLASH块0扇区30的写许可 WLOCK[0]=0时，该位写0，写1许可。 WLOCK[0]=1时，该位只能0，写1无效。	R/W
b29~b2	…	…	…	R/W
b1	F0NWPRT1	FLASH块0扇区1的写保护位	0：FLASH块0扇区1的写禁止 1：FLASH块0扇区1的写许可 WLOCK[0]=0时，该位写0，写1许可。 WLOCK[0]=1时，该位只能0，写1无效。	R/W
b0	F0NWPRT0	FLASH块0扇区0的写保护位	0：FLASH块0扇区0的写禁止 1：FLASH块0扇区0的写许可 WLOCK[0]=0时，该位写0，写1许可。 WLOCK[0]=1时，该位只能0，写1无效。	R/W

7.9.17 FLASH0 写保护寄存器 1(EFM_F0NWPRT1)

复位值：0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
F0NWPRT63	F0NWPRT62	F0NWPRT61	F0NWPRT60	F0NWPRT59	F0NWPRT58	F0NWPRT57	F0NWPRT56	F0NWPRT55	F0NWPRT54	F0NWPRT53	F0NWPRT52	F0NWPRT51	F0NWPRT50	F0NWPRT49	F0NWPRT48
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
F0NWPRT47	F0NWPRT46	F0NWPRT45	F0NWPRT44	F0NWPRT43	F0NWPRT42	F0NWPRT41	F0NWPRT40	F0NWPRT39	F0NWPRT38	F0NWPRT37	F0NWPRT36	F0NWPRT35	F0NWPRT34	F0NWPRT33	F0NWPRT32

位	标记	位名	功能	读写
b31	F0NWPRT63	FLASH块0扇区63的写保护位	0：FLASH块0扇区63的写禁止 1：FLASH块0扇区63的写许可 WLOCK[1]=0时，该位写0，写1许可。 WLOCK[1]=1时，该位只能0，写1无效。	R/W
b30	F0NWPRT62	FLASH块0扇区62的写保护位	0：FLASH块0扇区62的写禁止 1：FLASH块0扇区62的写许可 WLOCK[1]=0时，该位写0，写1许可。 WLOCK[1]=1时，该位只能0，写1无效。	R/W
b29~b2	…	…	…	R/W
b1	F0NWPRT33	FLASH块0扇区33的写保护位	0：FLASH块0扇区33的写禁止 1：FLASH块0扇区33的写许可 WLOCK[1]=0时，该位写0，写1许可。 WLOCK[1]=1时，该位只能0，写1无效。	R/W
b0	F0NWPRT32	FLASH块0扇区32的写保护位	0：FLASH块0扇区32的写禁止 1：FLASH块0扇区32的写许可 WLOCK[1]=0时，该位写0，写1许可。 WLOCK[1]=1时，该位只能0，写1无效。	R/W

7.9.18 FLASH0 写保护寄存器 2(EFM_F0NWPRT2)

复位值：0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
F0NWPRT95	F0NWPRT94	F0NWPRT93	F0NWPRT92	F0NWPRT91	F0NWPRT90	F0NWPRT89	F0NWPRT88	F0NWPRT87	F0NWPRT86	F0NWPRT85	F0NWPRT84	F0NWPRT83	F0NWPRT82	F0NWPRT81	F0NWPRT80
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
F0NWPRT79	F0NWPRT78	F0NWPRT77	F0NWPRT76	F0NWPRT75	F0NWPRT74	F0NWPRT73	F0NWPRT72	F0NWPRT71	F0NWPRT70	F0NWPRT69	F0NWPRT68	F0NWPRT67	F0NWPRT66	F0NWPRT65	F0NWPRT64

位	标记	位名	功能	读写
b31	F0NWPRT95	FLASH块0扇区95的写保护位	0：FLASH块0扇区95的写禁止 1：FLASH块0扇区95的写许可 WLOCK[2]=0时，该位写0，写1许可。 WLOCK[2]=1时，该位只能0，写1无效。	R/W
b30	F0NWPRT94	FLASH块0扇区94的写保护位	0：FLASH块0扇区94的写禁止 1：FLASH块0扇区94的写许可 WLOCK[2]=0时，该位写0，写1许可。 WLOCK[2]=1时，该位只能0，写1无效。	R/W
b29~b2	…	…	…	R/W
b1	F0NWPRT65	FLASH块0扇区65的写保护位	0：FLASH块0扇区65的写禁止 1：FLASH块0扇区65的写许可 WLOCK[2]=0时，该位写0，写1许可。 WLOCK[2]=1时，该位只能0，写1无效。	R/W
b0	F0NWPRT64	FLASH块0扇区64的写保护位	0：FLASH块0扇区64的写禁止 1：FLASH块0扇区64的写许可 WLOCK[2]=0时，该位写0，写1许可。 WLOCK[2]=1时，该位只能0，写1无效。	R/W

7.9.19 FLASH0 写保护寄存器 3(EFM_F0NWPRT3)

复位值：0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
F0N WPR T127	F0N WPR T126	F0N WPR T125	F0N WPR T124	F0N WPR T123	F0N WPR T122	F0N WPR T121	F0N WPR T120	F0N WPR T119	F0N WPR T118	F0N WPR T117	F0N WPR T116	F0N WPR T115	F0N WPR T114	F0N WPR T113	F0N WPR T112
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
F0N WPR T111	F0N WPR T110	F0N WPR T109	F0N WPR T108	F0N WPR T107	F0N WPR T106	F0N WPR T105	F0N WPR T104	F0N WPR T103	F0N WPR T102	F0N WPR T101	F0N WPR T100	F0N WPR T99	F0N WPR T98	F0N WPR T97	F0N WPR T96

位	标记	位名	功能	读写
b31	F0NWPRT127	FLASH块0扇区127的写保护位	0：FLASH块0扇区127的写禁止 1：FLASH块0扇区127的写许可 WLOCK[3]=0时，该位写0，写1许可。 WLOCK[3]=1时，该位只能0，写1无效。	R/W
b30	F0NWPRT126	FLASH块0扇区126的写保护位	0：FLASH块0扇区126的写禁止 1：FLASH块0扇区126的写许可 WLOCK[3]=0时，该位写0，写1许可。 WLOCK[3]=1时，该位只能0，写1无效。	R/W
b29~b2	…	…	…	R/W
b1	F0NWPRT97	FLASH块0扇区97的写保护位	0：FLASH块0扇区97的写禁止 1：FLASH块0扇区97的写许可 WLOCK[3]=0时，该位写0，写1许可。 WLOCK[3]=1时，该位只能0，写1无效。	R/W
b0	F0NWPRT96	FLASH块0扇区96的写保护位	0：FLASH块0扇区96的写禁止 1：FLASH块0扇区96的写许可 WLOCK[3]=0时，该位写0，写1许可。 WLOCK[3]=1时，该位只能0，写1无效。	R/W

7.9.20 FLASH1 写保护寄存器 0(EFM_F1NWPRT0)

复位值：0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
F1NWPRT31	F1NWPRT30	F1NWPRT29	F1NWPRT28	F1NWPRT27	F1NWPRT26	F1NWPRT25	F1NWPRT24	F1NWPRT23	F1NWPRT22	F1NWPRT21	F1NWPRT20	F1NWPRT19	F1NWPRT18	F1NWPRT17	F1NWPRT16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
F1NWPRT15	F1NWPRT14	F1NWPRT13	F1NWPRT12	F1NWPRT11	F1NWPRT10	F1NWPRT9	F1NWPRT8	F1NWPRT7	F1NWPRT6	F1NWPRT5	F1NWPRT4	F1NWPRT3	F1NWPRT2	F1NWPRT1	F1NWPRT0

位	标记	位名	功能	读写
b31	F1NWPRT31	FLASH块1扇区31的写保护位	0：FLASH块1扇区31的写禁止 1：FLASH块1扇区31的写许可 WLOCK[4]=0时，该位写0，写1许可。 WLOCK[4]=1时，该位只能0，写1无效。	R/W
b30	F1NWPRT30	FLASH块1扇区30的写保护位	0：FLASH块1扇区30的写禁止。 1：FLASH块1扇区30的写许可。 WLOCK[4]=0时，该位写0，写1许可。 WLOCK[4]=1时，该位只能0，写1无效。	R/W
b29~b2	…	…	…	R/W
b1	F1NWPRT1	FLASH块1扇区1的写保护位	0：FLASH块1扇区1的写禁止 1：FLASH块1扇区1的写许可 WLOCK[4]=0时，该位写0，写1许可。 WLOCK[4]=1时，该位只能0，写1无效。	R/W
b0	F1NWPRT0	FLASH块1扇区0的写保护位	0：FLASH块1扇区0的写禁止 1：FLASH块1扇区0的写许可 WLOCK[4]=0时，该位写0，写1许可。 WLOCK[4]=1时，该位只能0，写1无效。	R/W

7.9.21 FLASH1 写保护寄存器 1(EFM_F1NWPRT1)

复位值：0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
F1NWPRT63	F1NWPRT62	F1NWPRT61	F1NWPRT60	F1NWPRT59	F1NWPRT58	F1NWPRT57	F1NWPRT56	F1NWPRT55	F1NWPRT54	F1NWPRT53	F1NWPRT52	F1NWPRT51	F1NWPRT50	F1NWPRT49	F1NWPRT48
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
F1NWPRT47	F1NWPRT46	F1NWPRT45	F1NWPRT44	F1NWPRT43	F1NWPRT42	F1NWPRT41	F1NWPRT40	F1NWPRT39	F1NWPRT38	F1NWPRT37	F1NWPRT36	F1NWPRT35	F1NWPRT34	F1NWPRT33	F1NWPRT32

位	标记	位名	功能	读写
b31	F1NWPRT63	FLASH块1扇区63的写保护位	0：FLASH块1扇区63的写禁止 1：FLASH块1扇区63的写许可 WLOCK[5]=0时，该位写0，写1许可。 WLOCK[5]=1时，该位只能0，写1无效。	R/W
b30	F1NWPRT62	FLASH块1扇区62的写保护位	0：FLASH块1扇区62的写禁止 1：FLASH块1扇区62的写许可 WLOCK[5]=0时，该位写0，写1许可。 WLOCK[5]=1时，该位只能0，写1无效。	R/W
b29~b2	…	…	…	R/W
b1	F1NWPRT33	FLASH块1扇区33的写保护位	0：FLASH块1扇区33的写禁止 1：FLASH块1扇区33的写许可 WLOCK[5]=0时，该位写0，写1许可。 WLOCK[5]=1时，该位只能0，写1无效。	R/W
b0	F1NWPRT32	FLASH块1扇区32的写保护位	0：FLASH块1扇区32的写禁止 1：FLASH块1扇区32的写许可 WLOCK[5]=0时，该位写0，写1许可。 WLOCK[5]=1时，该位只能0，写1无效。	R/W

7.9.22 FLASH1 写保护寄存器 2(EFM_F1NWPRT2)

复位值：0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
F1NWPRT95	F1NWPRT94	F1NWPRT93	F1NWPRT92	F1NWPRT91	F1NWPRT90	F1NWPRT89	F1NWPRT88	F1NWPRT87	F1NWPRT86	F1NWPRT85	F1NWPRT84	F1NWPRT83	F1NWPRT82	F1NWPRT81	F1NWPRT80
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
F1NWPRT79	F1NWPRT78	F1NWPRT77	F1NWPRT76	F1NWPRT75	F1NWPRT74	F1NWPRT73	F1NWPRT72	F1NWPRT71	F1NWPRT70	F1NWPRT69	F1NWPRT68	F1NWPRT67	F1NWPRT66	F1NWPRT65	F1NWPRT64

位	标记	位名	功能	读写
b31	F1NWPRT95	FLASH块1扇区95的写保护位	0：FLASH块1扇区95的写禁止 1：FLASH块1扇区95的写许可 WLOCK[6]=0时，该位写0，写1许可。 WLOCK[6]=1时，该位只能0，写1无效。	R/W
b30	F1NWPRT94	FLASH块1扇区94的写保护位	0：FLASH块1扇区94的写禁止 1：FLASH块1扇区94的写许可 WLOCK[6]=0时，该位写0，写1许可。 WLOCK[6]=1时，该位只能0，写1无效。	R/W
b29~b2	…	…	…	R/W
b1	F1NWPRT65	FLASH块1扇区65的写保护位	0：FLASH块1扇区65的写禁止 1：FLASH块1扇区65的写许可 WLOCK[6]=0时，该位写0，写1许可。 WLOCK[6]=1时，该位只能0，写1无效。	R/W
b0	F1NWPRT64	FLASH块1扇区64的写保护位	0：FLASH块1扇区64的写禁止 1：FLASH块1扇区64的写许可 WLOCK[6]=0时，该位写0，写1许可。 WLOCK[6]=1时，该位只能0，写1无效。	R/W

7.9.23 FLASH1 写保护寄存器 3(EFM_F1NWPRT3)

复位值：0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
F1NWPRT127	F1NWPRT126	F1NWPRT125	F1NWPRT124	F1NWPRT123	F1NWPRT122	F1NWPRT121	F1NWPRT120	F1NWPRT119	F1NWPRT118	F1NWPRT117	F1NWPRT116	F1NWPRT115	F1NWPRT114	F1NWPRT113	F1NWPRT112
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
F1NWPRT111	F1NWPRT110	F1NWPRT109	F1NWPRT108	F1NWPRT107	F1NWPRT106	F1NWPRT105	F1NWPRT104	F1NWPRT103	F1NWPRT102	F1NWPRT101	F1NWPRT100	F1NWPRT99	F1NWPRT98	F1NWPRT97	F1NWPRT96

位	标记	位名	功能	读写
b31	F1NWPRT127	FLASH块1扇区127的写保护位	0：FLASH块1扇区127的写禁止 1：FLASH块1扇区127的写许可 WLOCK[7]=0时，该位写0，写1许可。 WLOCK[7]=1时，该位只能0，写1无效。	R/W
b30	F1NWPRT126	FLASH块1扇区126的写保护位	0：FLASH块1扇区126的写禁止 1：FLASH块1扇区126的写许可 WLOCK[7]=0时，该位写0，写1许可。 WLOCK[7]=1时，该位只能0，写1无效。	R/W
b29~b2	…	…	…	R/W
b1	F1NWPRT97	FLASH块1扇区97的写保护位	0：FLASH块1扇区97的写禁止 1：FLASH块1扇区97的写许可 WLOCK[7]=0时，该位写0，写1许可。 WLOCK[7]=1时，该位只能0，写1无效。	R/W
b0	F1NWPRT96	FLASH块1扇区96的写保护位	0：FLASH块1扇区96的写禁止 1：FLASH块1扇区96的写许可 WLOCK[7]=0时，该位写0，写1许可。 WLOCK[7]=1时，该位只能0，写1无效。	R/W

7.10 注意事项

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

8 内置 SRAM（SRAM）

8.1 简介

本产品带有 4KB 备份 SRAM（SRAMB）和 512KB 系统 SRAM（SRAMH/SRAM1/SRAM2/SRAM3/SRAM4）。

各 SRAM 可按照字节、半字（16 位）或全字（32 位）访问。高速 SRAM（SRAMH）读写操作最快可以在 CPU 最高速度（240MHz）下 0 等待（即 1 周期）执行，也可插入等待周期。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	访问掉电模式保持SRAM（SRAMB）允许的CPU时钟频率范围	访问其它SRAM（SRAM1,2,3,4）允许的CPU时钟频率范围
0wait(1个CPU周期访问)	0~240MHz	0~120MHz	0~200MHz
1wait(2个CPU周期访问)	0~240MHz	0~240MHz	0~240MHz
2wait(3个CPU周期访问)	0~240MHz	0~240MHz	0~240MHz
3wait(4个CPU周期访问)	0~240MHz	0~240MHz	0~240MHz
4wait(5个CPU周期访问)	0~240MHz	0~240MHz	0~240MHz
5wait(6个CPU周期访问)	0~240MHz	0~240MHz	0~240MHz
6wait(7个CPU周期访问)	0~240MHz	0~240MHz	0~240MHz
7wait(8个CPU周期访问)	0~240MHz	0~240MHz	0~240MHz

表 8-1 SRAM 读写访问的等待周期设定和 CPU 时钟频率的关系

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

SRAM4 和 SRAMB 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/SRAM3 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	128KB	0x20000000~0x2001FFFF	Even-parity check
SRAM2	128KB	0x20020000~0x2003FFFF	Even-parity check
SRAM3	96KB	0x20040000~0x20057FFF	Even-parity check
SRAM4	32KB	0x20058000~0x2005FFFF	ECC check
SRAMB	4KB	0x200F0000~0x200F0FFF	ECC check
SRAMH	128KB	0x1FFE0000~0x1FFFFFFF	Even-parity check

表 8-2 SRAM 空间分配

注意：

- 在允许产生 RAM 奇偶校验错误产生 NMI 中断和复位的情况下，当存取数据时，必须对所用 RAM 空间以字为单位进行初始化；当从 RAMH 空间执行指令时，必须对所用 RAM 空间+3 字的区域以字为单位进行初始化。
- 在允许产生 RAM ECC 校验错误产生 NMI 中断和复位的情况下，当存取数据时，必须对所用 RAM 空间以字为单位进行初始化。
- 备份 SRAM 在使用前需要将 VBAT 域初始化，初始化方法参考【电池备份电源域】

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）	0x40050800	0x00000000
SRAM等待控制保护寄存器（SRAM_WTPR）	0x40050804	0x00000000
SRAM校验控制寄存器（SRAM_CKCR）	0x40050808	0x00000000
SRAM校验控制保护寄存器（SRAM_CKPR）	0x4005080C	0x00000000
SRAM校验状态寄存器（SRAM_ CKSR）	0x40050810	0x00000000

8.2.1 SRAM 等待控制寄存器（SRAM_WTCR）

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Rev	SRAMBWWT[2:0]			Rev	SRAMBRWT[2:0]			Rev	SRAMHWWT[2:0]			Rev	SRAMHRWT[2:0]		

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Rev	SRAM4WWT[2:0]			Rev	SRAM4RWT[2:0]			Rev	SRAM123WWT[2:0]			Rev	SRAM123RWT[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	SRAMBKWWT[2:0]	SRAMB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	SRAMBKRWT[2:0]	SRAMB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	SRAMHWWT[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	SRAMHRWT[1:0]	SRAMH读取周期选择	000b：1周期读 001b：2周期读 010b：3周期读	R/W

			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	SRAM4WWT[2:0]	SRAM4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
b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10~b8	SRAM4RWT[2:0]	SRAM4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
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6~b4	SRAM123WWT[2:0]	SRAM1, SRAM2和 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
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~b0	SRAM123RWT[2:0]	SRAM1, SRAM2和 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

8.2.2 SRAM 等待保护寄存器（SRAM_WTPR）

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Rev								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 3bh。

8.2.3 SRAM 校验控制寄存器（SRAM_CKCR）

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Rev				BECCMOD[1:0]		ECCMOD[1:0]		Rev						BECCOAD	ECCOAD

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

位	标记	位名	功能	读写
b31~b28	Reserved	-	读出时为“0”,写入时写“0”	R/W
b27~b26	BECCMOD[1:0]	SRAMB的ECC校验允许位	00：禁止ECC校验功能 01：若1位错误，ECC纠错， 不产生1位错误标志，不产生中断/复位； 若2位错误，ECC检错， 产生2位错误标志，产生中断/复位. 10：若1位错误，ECC纠错， 产生1位错误标志，不产生中断/复位； 若2位错误，ECC检错， 产生2位错误标志，产生中断/复位. 11：若1位错误，ECC纠错， 产生1位错误标志，产生中断/复位； 若2位错误，ECC检错， 产生2位错误标志，产生中断/复位.	R/W
b25~b24	ECCMOD[1:0]	SRAM4的ECC校验允许位	00：禁止ECC校验功能 01：若1位错误，ECC纠错， 不产生1位错误标志，不产生中断/复位； 若2位错误，ECC检错， 产生2位错误标志，产生中断/复位. 10：若1位错误，ECC纠错， 产生1位错误标志，不产生中断/复位； 若2位错误，ECC检错， 产生2位错误标志，产生中断/复位. 11：若1位错误，ECC纠错， 产生1位错误标志，产生中断/复位； 若2位错误，ECC检错， 产生2位错误标志，产生中断/复位.	R/W
b23~b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17	BECCOAD	SRAMB的ECC校验出错后操作	0：Non-maskable interrupt 1：Reset	R/W
b16	ECCOAD	SRAM4的ECC校验出错	0：Non-maskable interrupt	R/W

		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 出错后操作	0：Non-maskable interrupt 1：Reset	R/W

注意：

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

8.2.4 SRAM 校验保护寄存器（SRAM_CKPR）

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

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

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

8.2.5 SRAM 校验状态寄存器（SRAM_CKSR）

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							CAHCERAM_PYERR	SRAMB_2ERR	SRAMB_1ERR	SRAM4_2ERR	SRAM4_1ERR	SRAMH_PYERR	SRAM3_PYERR	SRAM2_PYERR	SRAM1_PYERR

位	标记	位名	功能	读写
b31~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8	CACHERAM_PYERR	CACHE RAM奇偶校验错误标志	0：无奇偶校验错误发生 1：有奇偶校验错误发生	R/W （注1）
b7	SRAMB_2ERR	SRAMB发生ECC 2-bit错误标志	0：无2-bit ECC错误发生 1：有2-bit ECC错误发生	R/W （注1）
b6	SRAMB_1ERR	SRAMB发生ECC 1-bit错误标志	0：无1-bit ECC错误发生 1：有1-bit ECC错误发生	R/W （注1）
b5	SRAM4_2ERR	SRAM4发生ECC 2-bit错误标志	0：无2-bit ECC错误发生 1：有2-bit ECC错误发生	R/W （注1）
b4	SRAM4_1ERR	SRAM4发生ECC 1-bit错误标志	0：无1-bit ECC错误发生 1：有1-bit ECC错误发生	R/W （注1）
b3	SRAMH_PYERR	SRANH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	SRAM3_PYERR	SRAM3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	SRAM2_PYERR	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）
b0	SRAM1_PYERR	SRAM1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）

注 1：写 1 清 0。

9 通用 IO（GPIO）

本章中使用的一些略称：

- Px （x=A~I）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~I, 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 CMOS/Schmitt 两种输入模式自由切换
- 支持外部中断的输入
- 支持 I/O pin 周边功能复用，一个 I/O pin 最多可具有 64 个可选择的复用功能
- 各个 I/O pin 可独立编程
- 各个 I/O pin 可以选择 2 个功能同时有效（不支持 2 个输出功能同时有效）

9.2 端口功能概要

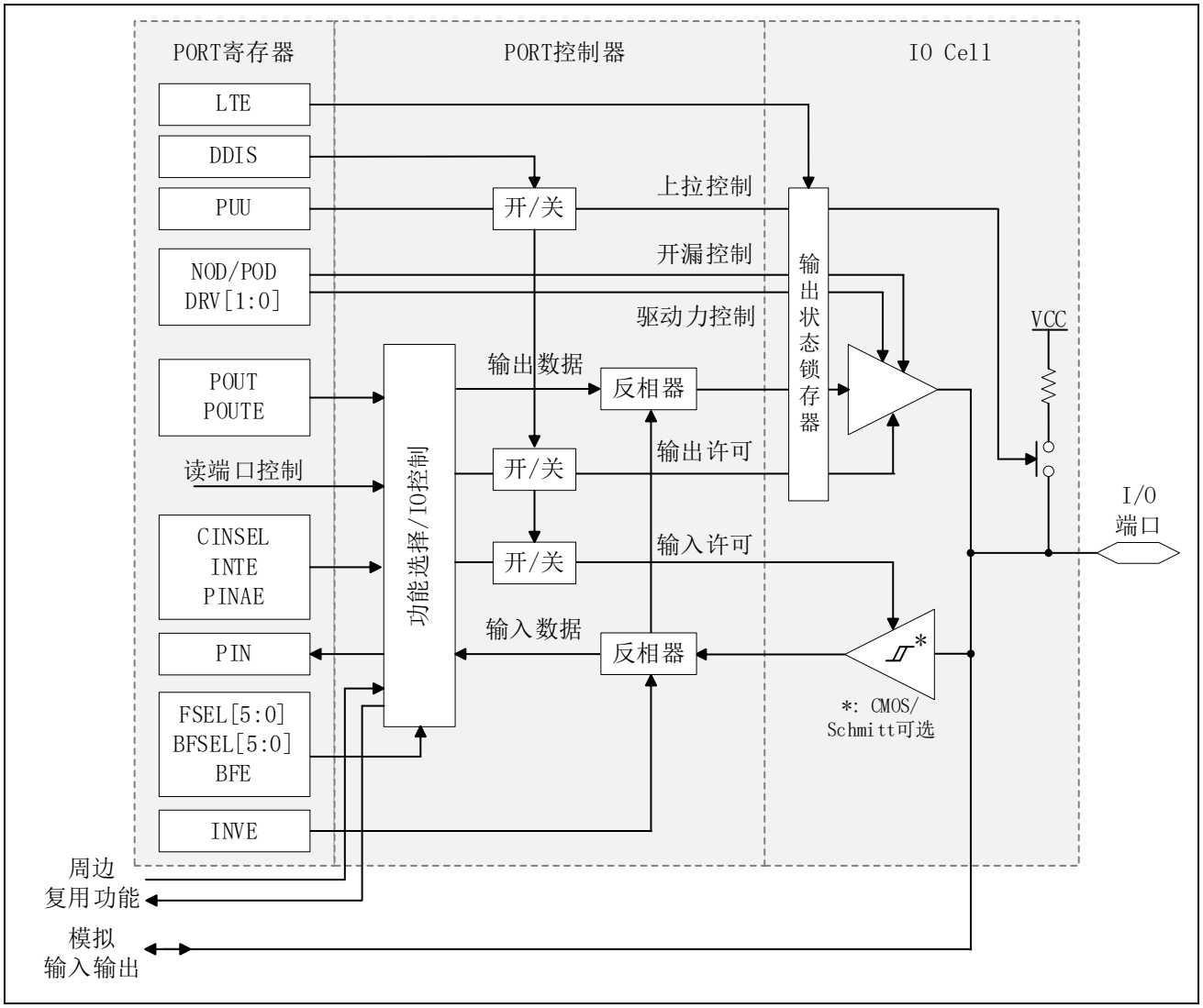


图 9-1 端口基本结构示意图

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

9.3 动作说明

9.3.1 通用输入输出 GPIO 功能

通用输入功能 GPI：

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

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

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

通用输出功能 GPO：

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

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

上面的寄存器都是 16 个 PORT 一组一起操作的。为了方便对单个 I/O 的控制，也可通过设置 PCRxy.POUTE 来允许或者禁止 I/O 的输出，PCRxy.POUTE 寄存器位与 POERx.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 POUTE=0)。

注意：

– 端口 PI13 与 MD 复用，复位端口 NRST 解除时 PI13/MD 须固定为低电平，以便系统能够进入用户模式。模式确立之后 PI13 可作为普通 IO 使用。

9.3.2 周边功能

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

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

9.3.3 双周边功能

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

9.3.4 Event Port 输入输出功能

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

作为触发源时，设置 PEVNTRISRm,PEVNTFALRm,PEVNTNFCR 选择上升沿或下降沿检测，以及数字滤波功能，并设置功能选择寄存器 PFSRxy 选择 EVNTPmn 功能。当

选择的边沿从端口输入时，则生成事件 EVENT_PORTm，输出到其它周边设备以触发其开始动作。

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

使用 Event Port 功能时，需要先将功能时钟控制 0 寄存器(FCG0)的外围电路触发功能使能位置设为有效。

9.3.5 外部中断 EIRQ 输入功能

除 PI13/MD 引脚外，每个 I/O 端口都具有外部中断输入功能。当 PCRxy.INTE 位设为 1 时，此 I/O 将作为外部中断源 EIRQy 被允许输入（如：PA0 对应 EIRQ0，PA2 对应 EIRQ2）。每个 EIRQy 可配置的 I/O 不止一个，使用时每个 EIRQy 不要同时允许多个 I/O 输入。EIRQy 输入功能与 PFSRxy.FSEL 选择的周边功能（包括 GPIO）可同时有效。

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

9.3.6 模拟功能

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

9.3.7 通用控制

1. 上/下拉电阻

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

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

PA11/USBFS_DM，PA12/USBFS_DP，PB14/USBHS_DM 及 PB14/USBHS_DP 引脚与 USB 端口复用，4 个引脚均内藏约 400KΩ 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 PCRxy.DRV[1:0]。

   本功能只在端口处于输出状态时才有效。

3. 开漏输出模式

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

   当 I/O 端口选择 I2Cx_SCL/I2Cx_SDA 功能时，将无视寄存器 NOD 的设定，开漏输出模式强制有效。

4. CMOS/Schmitt 输入模式

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

   在使用 I/O 输入功能时，请根据实际需求设置输入类型。

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

9.4 寄存器说明

BASE_ADDR：0x40053800

寄存器名	符号	偏移地址	位宽	复位值
通用输入数据寄存器	PIDRx	0x00+0x10*n *1	16/32	0xXXXX
通用输出数据寄存器	PODRx	0x04+0x10*n	16/32	0x0000
通用输出许可寄存器	POERx	0x06+0x10*n	16/32	0x0000
通用输出置位寄存器	POSRx	0x08+0x10*n	16/32	0x0000
通用输出复位寄存器	PORRx	0x0A+0x10*n	16/32	0x0000
通用输出翻转寄存器	POTRx	0x0C+0x10*n	16/32	0x0000
特殊控制寄存器	PSPCR	0x3F4	16/32	0x001F
公共控制寄存器	PCCR	0x3F8	16/32	0x1000
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~I 对应 n=0~8

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

BASE_ADDR：0x40010800

寄存器名	符号	偏移地址	位宽	复位值
Event Port1,2触发源选择寄存器	PEVNTTRGSR12	0x68	32	0x000001ff
Event Port3,4触发源选择寄存器	PEVNTTRGSR34	0x6C	32	0x000001ff
Event Port方向选择寄存器	PEVNTDIRRm	0x100+0x1C*(m-1)	32	0x00000000
Event Port输入数据寄存器	PEVNTIDRm	0x104+0x1C*(m-1)	32	0x00000000
Event Port输出数据寄存器	PEVNTODRm	0x108+0x1C*(m-1)	32	0x00000000
Event Port输出数据复位寄存器	PEVNTORRm	0x10C+0x1C*(m-1)	32	0x00000000
Event Port输出数据置位寄存器	PEVNTOSRm	0x110+0x1C*(m-1)	32	0x00000000
Event Port上升沿输入许可寄存器	PEVNTRISRm	0x114+0x1C*(m-1)	32	0x00000000
Event Port下降沿输入许可寄存器	PEVNTFALRm	0x118+0x1C*(m-1)	32	0x00000000
Event Port输入滤波控制寄存器	PEVNTNFCR	0x170	32	0x00000000

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
POR[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
PORR[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
POTR[15:0]															

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

此寄存器的读出值始终为 0x0000。

9.4.7 特殊控制寄存器（PSPCR）

复位值：0x001f

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

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

注意：

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

9.4.8 公共控制寄存器（PCCR）

复位值：0x1000

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

位	标记	位名	功能	读写
b15	Reserved	-	读出时为0，写时请写0	R/W
b14-b12	RDWT[2:0]	读端口等待	设置读寄存器PIDRx，PCRxy时插入的等待周期数	R/W

设定值	等待周期	推荐工作频率
000	无等待	~50MHz
001（默认值）	1周期	50~100MHz
010	2周期	100~150MHz
011	3周期	150~200MHz
100	4周期	200~250MHz
101	5周期	250MHz以上
110	6周期	250MHz以上
111	7周期	250MHz以上

b11~b6	Reserved	-	读出时为0，写时请写0	R/W
b5~b0	BFSEL[5:0]	副功能选择	各端口的功能配置请参考**数据手册中引脚功能表**	R/W

9.4.9 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[8:0]								

位	标记	位名	功能	读写
b15~b9	Reserved	-	读出时为0，写时请写0	R/W
b8~b0	PINAE[8:0]	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PF0~PF15， PINAE[6]控制PG0~PG15， PINAE[7]控制PH0~PH15， PINAE[8]控制PI0~PI13	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/W
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	-	CINSEL	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/W
b12	INTE	外部中断许可	0：外部中断输入禁止 1：外部中断输入许可	R/W
b11	Reserved	-	读出时为0，写时请写0	R/W
b10	CINSEL	输入模式选择	0：Schmitt输入 1：CMOS输入	R/W
b9	INVE	反相许可	0：输入输出数据不反相 1：输入输出数据反相	R/W
b8	PIN	输入状态	0：I/O端口输入状态为低电平 1：I/O端口输入状态为高电平 与寄存器PIDRx中PIN[y]功能一致	R
b7	Reserved	-	读出时为0，写时请写0	R/W
b6	PUU	上拉许可	0：内部上拉(pullup)电阻无效 1：内部上拉(pullup)电阻有效	R/W
b5~b4	DRV[1:0]	驱动模式选择	b00：低驱动力模式 b01：中驱动力模式 b10，b11：高驱动力模式	R/W
b3	Reserved	-	读出时为0，写时请写0	R/W
b2	NOD	NMOS开漏	0：正常CMOS输出模式 1：NMOS开漏输出	R/W
b1	POUTE	输出许可	0：输出禁止 1：输出许可 与寄存器POERx中POUTE[y]功能一致	R/W
b0	POUT	输出数据	0：输出低电平 1：输出高电平 与寄存器PODRx中POUT[y]功能一致	R/W

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

DDIS 位设为 1。

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

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

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

*1：以下端口通用控制寄存器 PCR 的复位值不是 b0000_000x_0000_0000，请注意。XTAL32_IN，XTAL32_OUT 复用端口 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/W
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/W
b5~b0	FSEL[5:0]	功能选择	各端口的功能配置请参考**数据手册中引脚功能表**	R/W

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

注意：

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

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

复位值：0x000001ff

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
-							TRGSEL[8:0]								

位	标记	位名	功能	读写
b31~b9	Reserved	-	读出时为0，写时请写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）】。	R/W

9.4.14 Event Port 方向选择寄存器（PEVNTDIRRm）

复位值：0x00000000

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

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

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

9.4.15 Event Port 输入数据寄存器（PEVNTIDRm）

复位值：0x00000000

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

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

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。

9.4.16 Event Port 输出数据寄存器（PEVNTODRm）

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-															
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/W
b15~b0	POUT[15:0]	端口输出值	0：Event Port输出低电平	R/W
			1：Event Port输出高电平	

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。

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

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-															
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/W
b15~b0	POR[15:0]	输出值复位	0：事件触发时对应PEVNTODRm.POUT无变化 1：事件触发时对应PEVNTODRm.POUT复位	R/W

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

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

复位值：0x00000000

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

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

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

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

复位值：0x00000000

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

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

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

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

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

复位值：0x00000000

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

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

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

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

复位值：0x00000000

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

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

1：Event Port1 数字滤波有效

9.4.22 32bit 访问

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

地址	b31 ~ b16	b15 ~ b0
0x40053800+0x10*n *1	Reserved	PIDRx
0x40053804+0x10*n	POERx	PODRx
0x40053808+0x10*n	PORRx	POSRx
0x4005380C+0x10*n	Reserved	POTRx
0x40053BF4	Reserved	PSPCR
0x40053BF8	PINAER	PCCR
0x40053BFC	Reserved	PWPR
0x40053C00+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~I 对应 n=0~8

9.5 注意事项

请不要将同一功能设置到多个端口上。

使用模拟功能时，请将相应端口的数字功能关闭(DDIS=1)。

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

10 中断控制器（INTC）

10.1 简介

中断控制器（INTC）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 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 Cortex™-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 Cortex™-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：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、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
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

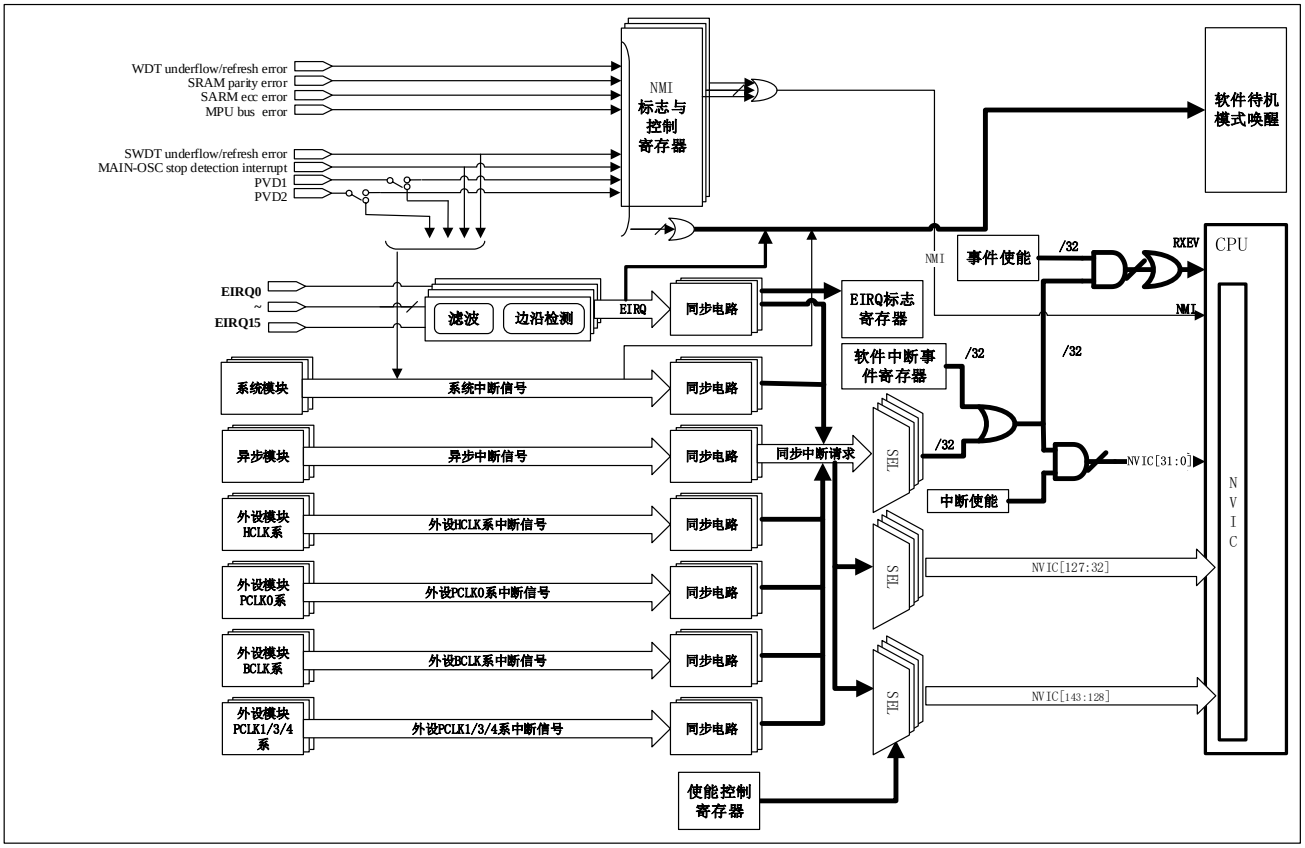


图 10-1 中断系统框图

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 ervice(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/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。
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。

地址	向量序号	IRQ序号	中断源	说明
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。
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。
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。

地址	向量序号	IRQ序号	中断源	说明
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。
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。

地址	向量序号	IRQ序号	中断源	说明
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。
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。

地址	向量序号	IRQ序号	中断源	说明
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。
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。

地址	向量序号	IRQ序号	中断源	说明
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。
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

地址	向量序号	IRQ序号	中断源	说明
				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，被称为中断源；中断事件请求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	是否可选择为内部触发源	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]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
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]
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_1	DMA_1_TC0	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[0]
33	021h		DMA_1_TC1	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[1]
34	022h		DMA_1_TC2	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[2]
35	023h		DMA_1_TC3	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[3]
36	024h		DMA_1_TC4	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[4]
37	025h		DMA_1_TC5	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[5]
38	026h		DMA_1_TC6	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[6]
39	027h		DMA_1_TC7	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[7]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
40	028h		DMA_1_BTC0	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[8]
41	029h		DMA_1_BTC1	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[9]
42	02Ah		DMA_1_BTC2	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[10]
43	02Bh		DMA_1_BTC3	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[11]
44	02Ch		DMA_1_BTC4	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[12]
45	02Dh		DMA_1_BTC5	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[13]
46	02Eh		DMA_1_BTC6	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[14]
47	02Eh		DMA_1_BTC7	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[15]
48	030h		DMA_1_ERR	√	-	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[16]
49	031h	EFM	EFM_PEERR	√	-	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[17]
50	032h		EFM_RDCOL	√	-	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[18]
51	033h		EFM_OPTEND	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[19]
52	034h	USBFS	USBFS_SOF	-	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[20]
53	035h	USBHS	USBHS_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		DCU5	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[27]
60	03Ch		DCU6	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[28]
61	03Dh		DCU7	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[29]
62	03Eh		DCU8	√	√	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[30]
63	03Fh	-	-	-	-	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[31]
64	040h	DMA_2	DMA_2_TC0	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[0]
65	041h		DMA_2_TC1	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[1]
66	042h		DMA_2_TC2	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[2]
67	043h		DMA_2_TC3	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[3]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
68	044h		DMA_2_TC4	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[4]
69	045h		DMA_2_TC5	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[5]
70	046h		DMA_2_TC6	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[6]
71	047h		DMA_2_TC7	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[7]
72	048h		DMA_2_BTC0	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[8]
73	049h		DMA_2_BTC1	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[9]
74	04Ah		DMA_2_BTC2	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[10]
75	04Bh		DMA_2_BTC3	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[11]
76	04Ch		DMA_2_BTC4	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[12]
77	04Dh		DMA_2_BTC5	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[13]
78	04Eh		DMA_2_BTC6	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[14]
79	04Fh		DMA_2_BTC7	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[15]
80	050h		DMA_2_ERR	√	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[16]
81	051h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[17]
82	052h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[18]
83	053h	MAU	MAU_SQRT	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[19]
84	054h	DVP	DVP_FRAMSTA	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[20]
85	055h		DVP_LINESTA	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[21]
86	056h		DVP_LINEEND	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[22]
87	057h		DVP_FRAMEND	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[23]
88	058h		DVP_SQUERR	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[24]
89	059h		DVP_FIFOERR	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[25]
90	05Ah		DVP_DMAREQ	-	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[26]
91	05Bh	FMAC	FMAC_1_FIR	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[27]
92	05Ch		FMAC_2_FIR	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[28]
93	05Dh		FMAC_3_FIR	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[29]
94	05Eh		FMAC_4_FIR	√	√	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[30]
95	05Fh	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[31]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
96	060h	Timer0_1	TMR0_1_CMPA	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[0]
97	061h		TMR0_1_CMPB	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[1]
98	062h	Timer0_2	TMR0_2_CMPA	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[2]
99	063h		TMR0_2_CMPB	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[3]
100	064h	Timer2_1	TMR2_1_CMPA	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[4]
101	065h		TMR2_1_CMPB	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[5]
102	066h		TMR2_1_OVFA	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[6]
103	067h		TMR2_1_OVFB	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[7]
104	068h	Timer2_2	TMR2_2_CMPA	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[8]
105	069h		TMR2_2_CMPB	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[9]
106	06Ah		TMR2_2_OVFA	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[10]
107	06Bh		TMR2_2_OVFB	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[11]
108	06Ch	Timer2_3	TMR2_3_CMPA	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[12]
109	06Dh		TMR2_3_CMPB	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[13]
110	06Eh		TMR2_3_OVFA	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[14]
111	06Fh		TMR2_3_OVFB	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[15]
112	070h	Timer2_4	TMR2_4_CMPA	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[16]
113	071h		TMR2_4_CMPB	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[17]
114	072h		TMR2_4_OVFA	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[18]
115	073h		TMR2_4_OVFB	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[19]
116	074h	-	-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[20]
117	075h	-	-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[21]
118	076h	-	-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[22]
119	077h	-	-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[23]
120	078h	RTC	RTC_TP	√	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[24]
121	079h		RTC_ALM	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[25]
122	07Ah		RTC_PRD	√	√	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[26]
123	07Bh		-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[27]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
124	07Ch	-	-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[28]
125	07Dh	XTAL	XTAL_STOP	√	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[29]
126	07Eh	WKTM	WKTM_PRD	√	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[30]
127	07Fh	SWDT	SWDT_REFUDF	√	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[31]
128	080h	Timer6_1	TMR6_1_GCMA	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[0]
129	081h		TMR6_1_GCMB	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[1]
130	082h		TMR6_1_GCMC	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[2]
131	083h		TMR6_1_GCMD	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[3]
132	084h		TMR6_1_GCME	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[4]
133	085h		TMR6_1_GCMF	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[5]
134	086h		TMR6_1_GOVF	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[6]
135	087h		TMR6_1_GUDF	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[7]
136	088h	Timer4_1	TMR4_1_G/SCMUH	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[8]
137	089h		TMR4_1_G/SCMUL	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[9]
138	08Ah		TMR4_1_G/SCMVH	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[10]
139	08Bh		TMR4_1_G/SCMVL	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[11]
140	08Ch		TMR4_1_G/SCMWH	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[12]
141	08Dh		TMR4_1_G/SCMWL	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[13]
142	08Eh		TMR4_1_GOVF	√	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[14]
143	08Fh		TMR4_1_GUDF	√	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[15]
144	090h	Timer6_2	TMR6_2_GCMA	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[16]
145	091h		TMR6_2_GCMB	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[17]
146	092h		TMR6_2_GCMC	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[18]
147	093h		TMR6_2_GCMD	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[19]
148	094h		TMR6_2_GCME	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[20]
149	095h		TMR6_2_GCMF	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[21]
150	096h		TMR6_2_GOVF	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[22]
151	097h		TMR6_2_GUDF	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[23]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
152	098h	Timer4_2	TMR4_2_G/SCMUH	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[24]
153	099h		TMR4_2_G/SCMUL	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[25]
154	09Ah		TMR4_2_G/SCMVH	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[26]
155	09Bh		TMR4_2_G/SCMVL	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[27]
156	09Ch		TMR4_2_G/SCMWH	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[28]
157	09Dh		TMR4_2_G/SCMWL	√	√	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[29]
158	09Eh		TMR4_2_GOVF	√	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[30]
159	09Fh		TMR4_2_GUDF	√	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[31]
160	0A0h	Timer6_3	TMR6_3_GCMA	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[0]
161	0A1h		TMR6_3_GCMB	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[1]
162	0A2h		TMR6_3_GCMC	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[2]
163	0A3h		TMR6_3_GCMD	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[3]
164	0A4h		TMR6_3_GCME	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[4]
165	0A5h		TMR6_3_GCMF	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[5]
166	0A6h		TMR6_3_GOVF	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[6]
167	0A7h		TMR6_3_GUDF	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[7]
168	0A8h	Timer4_3	TMR4_3_G/SCMUH	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[8]
169	0A9h		TMR4_3_G/SCMUL	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[9]
170	0AAh		TMR4_3_G/SCMVH	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[10]
171	0ABh		TMR4_3_G/SCMVL	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[11]
172	0ACh		TMR4_3_G/SCMWH	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[12]
173	0ADh		TMR4_3_G/SCMWL	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[13]
174	0AEh		TMR4_3_GOVF	√	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[14]
175	0AFh		TMR4_3_GUDF	√	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[15]
176	0B0h	Timer6_1	TMR6_1_GDTE	√	-	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		TMR6_1_SCMA	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[19]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
180	0B4h		TMR6_1_SCMB	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[20]
181	0B5h	Timer4_1	TMR4_1_RLOU	√	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[21]
182	0B6h		TMR4_1_RLOV	√	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[22]
183	0B7h		TMR4_1_RLOW	√	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[23]
184	0B8h	Timer6_2	TMR6_2_GDTE	√	-	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		TMR6_2_SCMA	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[27]
188	0BCh		TMR6_2_SCMB	√	√	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[28]
189	0BDh	Timer4_2	TMR4_2_RLOU	√	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[29]
190	0BEh		TMR4_2_RLOV	√	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[30]
191	0BFh		TMR4_2_RLOW	√	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[31]
192	0C0h	Timer6_3	TMR6_3_GDTE	√	-	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		TMR6_3_SCMA	√	√	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[3]
196	0C4h		TMR6_3_SCMB	√	√	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[4]
197	0C5h	Timer4_3	TMR4_3_RLOU	√	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[5]
198	0C6h		TMR4_3_RLOV	√	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[6]
199	0C7h		TMR4_3_RLOW	√	-	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]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
208	0D0h	Timer6_4	TMR6_4_GCMA	√	√	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[16]
209	0D1h		TMR6_4_GCMB	√	√	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[17]
210	0D2h		TMR6_4_GCMC	√	√	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[18]
211	0D3h		TMR6_4_GCMD	√	√	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[19]
212	0D4h		TMR6_4_GCME	√	√	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[20]
213	0D5h		TMR6_4_GCMF	√	√	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[21]
214	0D6h		TMR6_4_GOVF	√	√	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[22]
215	0D7h		TMR6_4_GUDF	√	√	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[23]
216	0D8h		TMR6_4_GDTE	√	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[24]
217	0D9h		-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[25]
218	0DAh		-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[26]
219	0DBh		TMR6_4_SCMA	√	√	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[27]
220	0DCh		TMR6_4_SCMB	√	√	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	Timer6_5	TMR6_5_GCMA	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[0]
225	0E1h		TMR6_5_GCMB	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[1]
226	0E2h		TMR6_5_GCMC	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[2]
227	0E3h		TMR6_5_GCMD	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[3]
228	0E4h		TMR6_5_GCME	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[4]
229	0E5h		TMR6_5_GCMF	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[5]
230	0E6h		TMR6_5_GOVF	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[6]
231	0E7h		TMR6_5_GUDF	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[7]
232	0E8h		TMR6_5_GDTE	√	-	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		TMR6_5_SCMA	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[11]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
236	0ECh		TMR6_5_SCMB	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[12]
237	0EDh	TimerA_1	TMRA_1_OVF	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[13]
238	0EEh		TMRA_1_UDF	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[14]
239	0EFh		TMRA_1_CMP	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[15]
240	0F0h	Timer6_6	TMR6_6_GCMA	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[16]
241	0F1h		TMR6_6_GCMB	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[17]
242	0F2h		TMR6_6_GCMC	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[18]
243	0F3h		TMR6_6_GCMD	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[19]
244	0F4h		TMR6_6_GCME	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[20]
245	0F5h		TMR6_6_GCMF	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[21]
246	0F6h		TMR6_6_GOVF	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[22]
247	0F7h		TMR6_6_GUDF	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[23]
248	0F8h		TMR6_6_GDTE	√	-	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		TMR6_6_SCMA	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[27]
252	0FCh		TMR6_6_SCMB	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[28]
253	0FDh	TimerA_2	TMRA_2_OVF	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[29]
254	0FEh		TMRA_2_UDF	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[30]
255	0FFh		TMRA_2_CMP	√	√	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[31]
256	100h	Timer6_7	TMR6_7_GCMA	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[0]
257	101h		TMR6_7_GCMB	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[1]
258	102h		TMR6_7_GCMC	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[2]
259	103h		TMR6_7_GCMD	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[3]
260	104h		TMR6_7_GCME	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[4]
261	105h		TMR6_7_GCMF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[5]
262	106h		TMR6_7_GOVF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[6]
263	107h		TMR6_7_GUDF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[7]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
264	108h		TMR6_7_GDTE	√	-	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[8]
265	109h		-	-	-	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[9]
266	10Ah		-	-	-	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[10]
267	10Bh		TMR6_7_SCMA	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[11]
268	10Ch		TMR6_7_SCMB	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[12]
269	10Dh	TimerA_3	TMRA_3_OVF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[13]
270	10Eh		TMRA_3_UDF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[14]
271	10Fh		TMRA_3_CMP	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[15]
272	110h	Timer6_8	TMR6_8_GCMA	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[16]
273	111h		TMR6_8_GCMB	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[17]
274	112h		TMR6_8_GCMC	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[18]
275	113h		TMR6_8_GCMD	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[19]
276	114h		TMR6_8_GCME	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[20]
277	115h		TMR6_8_GCMF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[21]
278	116h		TMR6_8_GOVF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[22]
279	117h		TMR6_8_GUDF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[23]
280	118h		TMR6_8_GDTE	√	-	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[24]
281	119h		-	-	-	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[25]
282	11Ah		-	-	-	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[26]
283	11Bh		TMR6_8_SCMA	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[27]
284	11Ch		TMR6_8_SCMB	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[28]
285	11Dh	TimerA_4	TMRA_4_OVF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[29]
286	11Eh		TMRA_4_UDF	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[30]
287	11Fh		TMRA_4_CMP	√	√	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[31]
288	120h	EMB	EMB_GR0	√	-	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[0]
289	121h		EMB_GR1	√	-	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[1]
290	122h		EMB_GR2	√	-	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[2]
291	123h		EMB_GR3	√	-	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[3]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
292	124h		EMB_GR4	√	-	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[4]
293	125h		EMB_GR5	√	-	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[5]
294	126h		EMB_GR6	√	-	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[6]
295	127h	USBHS	USBHS_EP1_OUT	√	-	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[7]
296	128h		USBHS_EP1_IN	√	-	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[8]
297	129h		USBHS_GLB	√	-	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[9]
298	12Ah		USBHS_WKUP	√	-	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[10]
299	12Bh	AOS_STRG	AOS_STRG	-	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[11]
300	12Ch	USART_1	USART_1_EI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[12]
301	12Dh		USART_1_RI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[13]
302	12Eh		USART_1_TI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[14]
303	12Fh		USART_1_TCI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[15]
304	130h		USART_1_RTO	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[16]
305	131h	USART_2	USART_2_EI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[17]
306	132h		USART_2_RI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[18]
307	133h		USART_2_TI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[19]
308	134h		USART_2_TCI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[20]
309	135h		USART_2_RTO	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[21]
310	136h	SPI_1	SPI_1_SPRI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[22]
311	137h		SPI_1_SPTI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[23]
312	138h		SPI_1_SPII	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[24]
313	139h		SPI_1_SPEI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[25]
314	13Ah		SPI_1_SPEND	-	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[26]
315	13Bh	SPI_2	SPI_2_SPRI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[27]
316	13Ch		SPI_2_SPTI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[28]
317	13Dh		SPI_2_SPII	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[29]
318	13Eh		SPI_2_SPEI	√	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[30]
319	13Fh		SPI_2_STEND	-	√	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[31]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
320	140h	TimerA_5	TMRA_5_OVF	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[0]
321	141h		TMRA_5_UDF	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[1]
322	142h		TMRA_5_CMP	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[2]
323	143h	TimerA_6	TMRA_6_OVF	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[3]
324	144h		TMRA_6_UDF	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[4]
325	145h		TMRA_6_CMP	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[5]
326	146h	TimerA_7	TMRA_7_OVF	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[6]
327	147h		TMRA_7_UDF	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[7]
328	148h		TMRA_7_CMP	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[8]
329	149h	TimerA_8	TMRA_8_OVF	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[9]
330	14Ah		TMRA_8_UDF	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[10]
331	14Bh		TMRA_8_CMP	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[11]
332	14Ch	USART_3	USART_3_EI	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[12]
333	14Dh		USART_3_RI	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[13]
334	14Eh		USART_3_TI	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[14]
335	14Fh		USART_3_TCI	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[15]
336	150h	USART_4	USART_4_EI	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[16]
337	151h		USART_4_RI	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[17]
338	152h		USART_4_TI	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[18]
339	153h		USART_4_TCI	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[19]
340	154h	CAN_1	CAN_1_HOST	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[20]
341	155h	CAN_2	CAN_2_HOST	√	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[21]
342	156h	SPI_3	SPI_3_SPRI	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[22]
343	157h		SPI_3_SPTI	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[23]
344	158h		SPI_3_SPII	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[24]
345	159h		SPI_3_SPEI	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[25]
346	15Ah		SPI_3_SPEND	-	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[26]
347	15Bh	SPI_4	SPI_4_SPRI	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[27]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
348	15Ch		SPI_4_SPTI	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[28]
349	15Dh		SPI_4_SPII	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[29]
350	15Eh		SPI_4_SPEI	√	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[30]
351	15Fh		SPI_4_SPEND	-	√	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[31]
352	160h	TimerA_9	TMRA_9_OVF	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[0]
353	161h		TMRA_9_UDF	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[1]
354	162h		TMRA_9_CMP	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[2]
355	163h	TimerA_10	TMRA_10_OVF	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[3]
356	164h		TMRA_10_UDF	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[4]
357	165h		TMRA_10_CMP	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[5]
358	166h	TimerA_11	TMRA_11_OVF	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[6]
359	167h		TMRA_11_UDF	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[7]
360	168h		TMRA_11_CMP	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[8]
361	169h	TimerA_12	TMRA_12_OVF	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[9]
362	16Ah		TMRA_12_UDF	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[10]
363	16Bh		TMRA_12_CMP	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[11]
364	16Ch	USART_5	USART_5_BRKWKPI	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[12]
365	16Dh		USART_5_EI	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[13]
366	16Eh		USART_5_RI	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[14]
367	16Fh		USART_5_TI	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[15]
368	170h		USART_5_TCI	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[16]
369	171h	USART_6	USART_6_EI	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[17]
370	172h		USART_6_RI	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[18]
371	173h		USART_6_TI	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[19]
372	174h		USART_6_TCI	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[20]
373	175h		USART_6_RTO	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[21]
374	176h	SPI_5	SPI_5_SPRI	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[22]
375	177h		SPI_5_SPTI	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[23]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
376	178h		SPI_5_SPII	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[24]
377	179h		SPI_5_SPEI	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[25]
378	17Ah		SPI_5_SPEND	-	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[26]
379	17Bh	SPI_6	SPI_6_SPRI	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[27]
380	17Ch		SPI_6_SPTI	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[28]
381	17Dh		SPI_6_SPII	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[29]
382	17Eh		SPI_6_SPEI	√	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[30]
383	17Fh		SPI_6_SPEND	-	√	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[31]
384	180h	I2S_1	I2S_1_TXIRQOUT	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[0]
385	181h		I2S_1_RXIRQOUT	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[1]
386	182h		I2S_1_ERRIRQOUT	√	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[2]
387	183h	I2S_2	I2S_2_TXIRQOUT	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[3]
388	184h		I2S_2_RXIRQOUT	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[4]
389	185h		I2S_2_ERRIRQOUT	√	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[5]
390	186h	USART_7	USART_7_EI	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[6]
391	187h		USART_7_RI	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[7]
392	188h		USART_7_TI	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[8]
393	189h		USART_7_TCI	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[9]
394	18Ah		USART_7_RTO	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[10]
395	18Bh	USART_8	USART_8_EI	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[11]
396	18Ch		USART_8_RI	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[12]
397	18Dh		USART_8_TI	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[13]
398	18Eh		USART_8_TCI	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[14]
399	18Fh	USBFS	USBFS_GLB	√	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[15]
400	190h		USBFS_WKUP	√	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[16]
401	191h	HASH	HASH_INT	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[17]
402	192h	SDIOC_1	SDIOC_1_DMAR	-	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[18]
403	193h		SDIOC_1_DMAW	-	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[19]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
404	194h		SDIOC_1_SD	√	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[20]
405	195h	SDIOC_2	SDIOC_2_DMAR	-	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[21]
406	196h		SDIOC_2_DMAW	-	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[22]
407	197h		SDIOC_2_SD	√	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[23]
408	198h	EVENT port	EVENT_PORT1	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[24]
409	199h		EVENT_PORT2	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[25]
410	19Ah		EVENT_PORT3	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[26]
411	19Bh		EVENT_PORT4	√	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[27]
412	19Ch	ETHER	ETH_GLB_INT	√	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[28]
413	19Dh		ETH_WKP_INT	√	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[29]
414	19Eh		ETH_PPS_OUT_0	-	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[30]
415	19Fh		ETH_PPS_OUT_1	-	√	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[31]
416	1A0h	I2S_3	I2S_3_TXIRQOUT	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[0]
417	1A1h		I2S_3_RXIRQOUT	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[1]
418	1A2h		I2S_3_ERRIRQOUT	√	-	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[2]
419	1A3h	I2S_4	I2S_4_TXIRQOUT	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[3]
420	1A4h		I2S_4_RXIRQOUT	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[4]
421	1A5h		I2S_4_ERRIRQOUT	√	-	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[5]
422	1A6h	USART_9	USART_9_EI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[6]
423	1A7h		USART_9_RI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[7]
424	1A8h		USART_9_TI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[8]
425	1A9h		USART_9_TCI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[9]
426	1AAh	USART_10	USART_10_BRKWKPI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[10]
427	1ABh	-	USART_10_EI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[11]
428	1ACh	-	USART_10_RI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[12]
429	1ADh	-	USART_10_TI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[13]
430	1AEh		USART_10_TCI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[14]
431	1AFh	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[15]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
432	1B0h	I2C_1	I2C_1_RXI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[16]
433	1B1h		I2C_1_TXI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[17]
434	1B2h		I2C_1_TEI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[18]
435	1B3h		I2C_1_EEI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[19]
436	1B4h	I2C_2	I2C_2_RXI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[20]
437	1B5h		I2C_2_TXI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[21]
438	1B6h		I2C_2_TEI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[22]
439	1B7h		I2C_2_EEI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[23]
440	1B8h	I2C_3	I2C_3_RXI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[24]
441	1B9h		I2C_3_TXI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[25]
442	1BAh		I2C_3_TEI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[26]
443	1BBh		I2C_3_EEI	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[27]
444	1BCh	ACMP	CMP1	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[28]
445	1BDh		CMP2	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[29]
446	1BEh		CMP3	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[30]
447	1BFh		CMP4	√	√	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[31]
448	1C0h	I2C_4	I2C_4_RXI	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[0]
449	1C1h		I2C_4_TXI	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[1]
450	1C2h		I2C_4_TEI	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[2]
451	1C3h		I2C_4_EEI	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[3]
452	1C4h	I2C_5	I2C_5_RXI	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[4]
453	1C5h		I2C_5_TXI	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[5]
454	1C6h		I2C_5_TEI	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[6]
455	1C7h		I2C_5_EEI	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[7]
456	1C8h	I2C_6	I2C_6_RXI	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[8]
457	1C9h		I2C_6_TXI	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[9]
458	1CAh		I2C_6_TEI	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[10]
459	1CBh		I2C_6_EEI	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[11]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
460	1CCh	USART_1	USART_1_WUPI	√	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[12]
461	1CDh	PVD	PVD_PVD1	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[13]
462	1CEh		PVD_PVD2	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[14]
463	1CFh	OTS	OTS	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[15]
464	1D0h	FCM	FCMFERRI	√	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[16]
465	1D1h		FCMMENDI	√	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[17]
466	1D2h		FCMCOVFI	√	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[18]
467	1D3h	WDT	WDT_REFUDF	√	√	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[19]
468	1D4h	CTC	CTC_ERR	√	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[20]
469	1D5h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[21]
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	ADC_1	ADC_1_EOCA	√	√	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[0]
481	1E1h		ADC_1_EOCB	√	√	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[1]
482	1E2h		ADC_1_CMP0	√	√	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[2]
483	1E3h		ADC_1_CMP1	√	√	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[3]
484	1E4h	ADC_2	ADC_2_EOCA	√	√	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[4]
485	1E5h		ADC_2_EOCB	√	√	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[5]
486	1E6h		ADC_2_CMP0	√	√	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[6]
487	1E7h		ADC_2_CMP1	√	√	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[7]

编号	中断事件请求序号	功能	功能名称	是否可选择为中断	是否可选择为内部触发源	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
488	1E8h	ADC_3	ADC_3_EOCA	√	√	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[8]
489	1E9h		ADC_3_EOCB	√	√	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[9]
490	1EAh		ADC_3_CMP0	√	√	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[10]
491	1EBh		ADC_3_CMP1	√	√	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[11]
492	1ECh	TRNG	TRNG_END	√	√	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	NFC	NFC_INT	√	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[16]
497	1F1h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[17]
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 由软件设定寄存器 INTSFTTRG 的 STRG 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：

- 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”。

使用寄存器设定时，请按照下面流程设定：

1. 使用不可屏蔽中断事件请求时，请配置相应的功能。
2. 对 INT_NMICFR 各寄存器位写“1”，清除 INT_NMIFR 标志位，防止误动作。
3. 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。

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=0~15），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

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。具体中断向量分配请参考 10.3.2 中断事件请求序号。

第一种方式

共 32 个中断向量，对应 NVIC 的中断向量为 0~31。通过中断/事件选择寄存器 INT_SEL0~31 进行选择，可以在所有中断事件请求中任选 1 个作为中断源。0~31 中断向量需要通过 INT_IER 寄存器使能。

第二种方式

共 96 个中断向量，对应的中断向量为 32~127，通过中断选择寄存器 INT_SEL32~127 进行选择。 96 个中断向量以 6 个向量为 1 组分成 16 组，同时所有中断事件请求以 32 个请求为 1 组也分成 16 组。每组中的 6 个中断向量通过中断选择寄存器，选择对应的中断事件请求组中 32 个请求里的任一请求。

第三种方式

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

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。

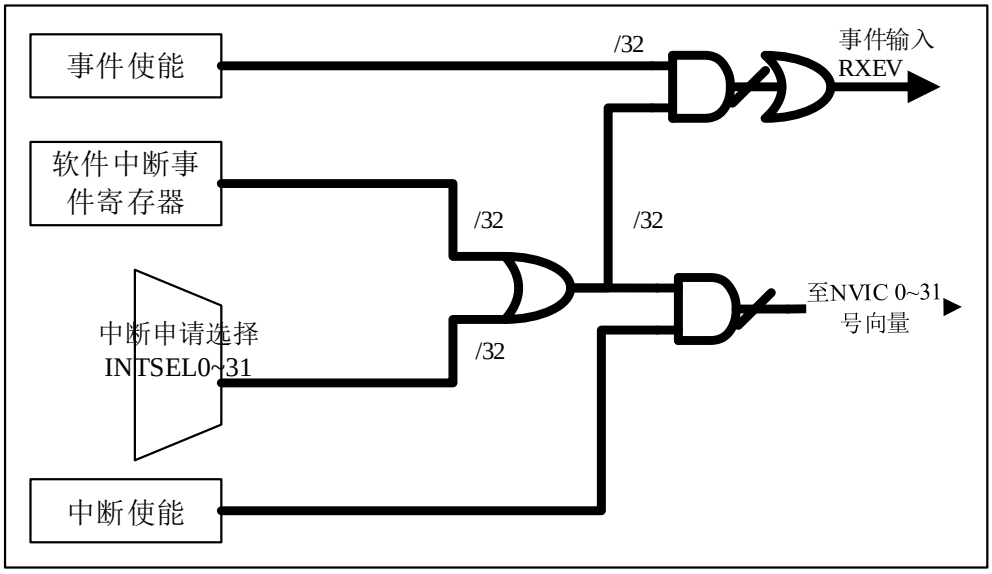


图 10-2 中断事件选择

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_SELn 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

__WFE(); 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

     __WFE(); 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

对于从 EIRQx（x=0~15）管脚输入的事件，可选择以下两种数字滤波器进行噪声滤波。滤波器 A 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：

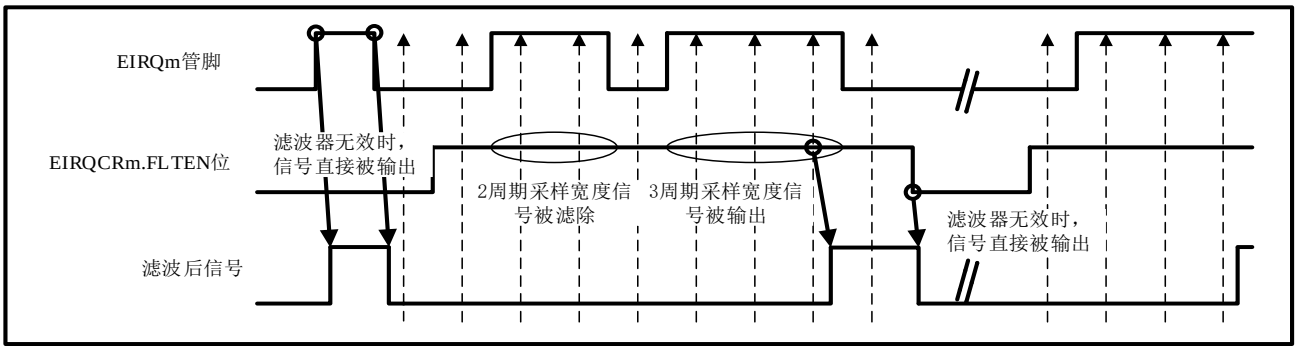


图 10-3 数字滤波器工作示意图

滤波器 B 以 INT_NOCCR.NOCSEL 设定的固定宽度对信号进行滤波，它不使用 PCLK3 作为采样时钟，因此可以在停止模式返回时使用。

滤波器 A 和滤波器 B 彼此独立，可单独使能或者关闭。

在进入停止模式前，请设定 INT_EIRQCRm.EFEN，禁止使用数字滤波器 A。从停止模式返回后再使能数字滤波器 A。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) 关闭数字滤波器 A
4) 启动数字滤波器 B（可选）
5) 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

  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

在执行 WFI 指令前，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，也可以通过配置外设触发事件寄存器，选择内部触发源来触发开始工作。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:0x40051000

寄存器名	符号	偏移地址	位宽	复位值
外部管脚中断滤波控制	INT_NOCCR	0x0000	32	0x00000000
不可屏蔽中断使能寄存器	INT_NMIENR	0x0004	32	0x00000000
不可屏蔽中断标志寄存器	INT_NMIFR	0x0008	32	0x00000000
不可屏蔽中断标志清除寄存器	INT_NMICFR	0x000C	32	0x00000000
外部管脚中断控制寄存器0	INT_EIRQCR0	0x0010	32	0x00000000
外部管脚中断控制寄存器1	INT_EIRQCR1	0x0014	32	0x00000000
外部管脚中断控制寄存器2	INT_EIRQCR2	0x0018	32	0x00000000
外部管脚中断控制寄存器3	INT_EIRQCR3	0x001C	32	0x00000000
外部管脚中断控制寄存器4	INT_EIRQCR4	0x0020	32	0x00000000
外部管脚中断控制寄存器5	INT_EIRQCR5	0x0024	32	0x00000000
外部管脚中断控制寄存器6	INT_EIRQCR6	0x0028	32	0x00000000
外部管脚中断控制寄存器7	INT_EIRQCR7	0x002C	32	0x00000000
外部管脚中断控制寄存器8	INT_EIRQCR8	0x0030	32	0x00000000
外部管脚中断控制寄存器9	INT_EIRQCR9	0x0034	32	0x00000000
外部管脚中断控制寄存器10	INT_EIRQCR10	0x0038	32	0x00000000
外部管脚中断控制寄存器11	INT_EIRQCR11	0x003C	32	0x00000000
外部管脚中断控制寄存器12	INT_EIRQCR12	0x0040	32	0x00000000
外部管脚中断控制寄存器13	INT_EIRQCR13	0x0044	32	0x00000000
外部管脚中断控制寄存器14	INT_EIRQCR14	0x0048	32	0x00000000
外部管脚中断控制寄存器15	INT_EIRQCR15	0x004C	32	0x00000000
停止模式唤醒事件使能寄存器	INT_WUPEN	0x0050	32	0x00000000
外部管脚中断标志寄存器	INT_EIFR	0x0054	32	0x00000000
外部管脚中断标志清除寄存器	INT_EIFCR	0x0058	32	0x00000000
中断/事件选择寄存器0	INT_SEL0	0x005C	32	0x000001FF
中断/事件选择寄存器1	INT_SEL1	0x0060	32	0x000001FF
中断/事件选择寄存器2	INT_SEL2	0x0064	32	0x000001FF

寄存器名	符号	偏移地址	位宽	复位值
中断/事件选择寄存器3	INT_SEL3	0x0068	32	0x000001FF
中断/事件选择寄存器4	INT_SEL4	0x006C	32	0x000001FF
中断/事件选择寄存器5	INT_SEL5	0x0070	32	0x000001FF
中断/事件选择寄存器6	INT_SEL6	0x0074	32	0x000001FF
中断/事件选择寄存器7	INT_SEL7	0x0078	32	0x000001FF
中断/事件选择寄存器8	INT_SEL8	0x007C	32	0x000001FF
中断/事件选择寄存器9	INT_SEL9	0x0080	32	0x000001FF
中断/事件选择寄存器10	INT_SEL10	0x0084	32	0x000001FF
中断/事件选择寄存器11	INT_SEL11	0x0088	32	0x000001FF
中断/事件选择寄存器12	INT_SEL12	0x008C	32	0x000001FF
中断/事件选择寄存器13	INT_SEL13	0x0090	32	0x000001FF
中断/事件选择寄存器14	INT_SEL14	0x0094	32	0x000001FF
中断/事件选择寄存器15	INT_SEL15	0x0098	32	0x000001FF
中断/事件选择寄存器16	INT_SEL16	0x009C	32	0x000001FF
中断/事件选择寄存器17	INT_SEL17	0x00A0	32	0x000001FF
中断/事件选择寄存器18	INT_SEL18	0x00A4	32	0x000001FF
中断/事件选择寄存器19	INT_SEL19	0x00A8	32	0x000001FF
中断/事件选择寄存器20	INT_SEL20	0x00AC	32	0x000001FF
中断/事件选择寄存器21	INT_SEL21	0x00B0	32	0x000001FF
中断/事件选择寄存器22	INT_SEL22	0x00B4	32	0x000001FF
中断/事件选择寄存器23	INT_SEL23	0x00B8	32	0x000001FF
中断/事件选择寄存器24	INT_SEL24	0x00BC	32	0x000001FF
中断/事件选择寄存器25	INT_SEL25	0x00C0	32	0x000001FF
中断/事件选择寄存器26	INT_SEL26	0x00C4	32	0x000001FF
中断/事件选择寄存器27	INT_SEL27	0x00C8	32	0x000001FF
中断/事件选择寄存器28	INT_SEL28	0x00CC	32	0x000001FF
中断/事件选择寄存器29	INT_SEL29	0x00D0	32	0x000001FF
中断/事件选择寄存器30	INT_SEL30	0x00D4	32	0x000001FF
中断/事件选择寄存器31	INT_SEL31	0x00D8	32	0x000001FF
中断选择寄存器32	INT_SEL32	0x00DC	32	0x000001FF

寄存器名	符号	偏移地址	位宽	复位值
中断选择寄存器33	INT_SEL33	0x00E0	32	0x000001FF
中断选择寄存器34	INT_SEL34	0x00E4	32	0x000001FF
中断选择寄存器35	INT_SEL35	0x00E8	32	0x000001FF
中断选择寄存器36	INT_SEL36	0x00EC	32	0x000001FF
中断选择寄存器37	INT_SEL37	0x00F0	32	0x000001FF
中断选择寄存器38	INT_SEL38	0x00F4	32	0x000001FF
中断选择寄存器39	INT_SEL39	0x00F8	32	0x000001FF
中断选择寄存器40	INT_SEL40	0x00FC	32	0x000001FF
中断选择寄存器41	INT_SEL41	0x0100	32	0x000001FF
中断选择寄存器42	INT_SEL42	0x0104	32	0x000001FF
中断选择寄存器43	INT_SEL43	0x0108	32	0x000001FF
中断选择寄存器44	INT_SEL44	0x010C	32	0x000001FF
中断选择寄存器45	INT_SEL45	0x0110	32	0x000001FF
中断选择寄存器46	INT_SEL46	0x0114	32	0x000001FF
中断选择寄存器47	INT_SEL47	0x0118	32	0x000001FF
中断选择寄存器48	INT_SEL48	0x011C	32	0x000001FF
中断选择寄存器49	INT_SEL49	0x0120	32	0x000001FF
中断选择寄存器50	INT_SEL50	0x0124	32	0x000001FF
中断选择寄存器51	INT_SEL51	0x0128	32	0x000001FF
中断选择寄存器52	INT_SEL52	0x012C	32	0x000001FF
中断选择寄存器53	INT_SEL53	0x0130	32	0x000001FF
中断选择寄存器54	INT_SEL54	0x0134	32	0x000001FF
中断选择寄存器55	INT_SEL55	0x0138	32	0x000001FF
中断选择寄存器56	INT_SEL56	0x013C	32	0x000001FF
中断选择寄存器57	INT_SEL57	0x0140	32	0x000001FF
中断选择寄存器58	INT_SEL58	0x0144	32	0x000001FF
中断选择寄存器59	INT_SEL59	0x0148	32	0x000001FF
中断选择寄存器60	INT_SEL60	0x014C	32	0x000001FF
中断选择寄存器61	INT_SEL61	0x0150	32	0x000001FF
中断选择寄存器62	INT_SEL62	0x0154	32	0x000001FF

寄存器名	符号	偏移地址	位宽	复位值
中断选择寄存器63	INT_SEL63	0x0158	32	0x000001FF
中断选择寄存器64	INT_SEL64	0x015C	32	0x000001FF
中断选择寄存器65	INT_SEL65	0x0160	32	0x000001FF
中断选择寄存器66	INT_SEL66	0x0164	32	0x000001FF
中断选择寄存器67	INT_SEL67	0x0168	32	0x000001FF
中断选择寄存器68	INT_SEL68	0x016C	32	0x000001FF
中断选择寄存器69	INT_SEL69	0x0170	32	0x000001FF
中断选择寄存器70	INT_SEL70	0x0174	32	0x000001FF
中断选择寄存器71	INT_SEL71	0x0178	32	0x000001FF
中断选择寄存器72	INT_SEL72	0x017C	32	0x000001FF
中断选择寄存器73	INT_SEL73	0x0180	32	0x000001FF
中断选择寄存器74	INT_SEL74	0x0184	32	0x000001FF
中断选择寄存器75	INT_SEL75	0x0188	32	0x000001FF
中断选择寄存器76	INT_SEL76	0x018C	32	0x000001FF
中断选择寄存器77	INT_SEL77	0x0190	32	0x000001FF
中断选择寄存器78	INT_SEL78	0x0194	32	0x000001FF
中断选择寄存器79	INT_SEL79	0x0198	32	0x000001FF
中断选择寄存器80	INT_SEL80	0x019C	32	0x000001FF
中断选择寄存器81	INT_SEL81	0x01A0	32	0x000001FF
中断选择寄存器82	INT_SEL82	0x01A4	32	0x000001FF
中断选择寄存器83	INT_SEL83	0x01A8	32	0x000001FF
中断选择寄存器84	INT_SEL84	0x01AC	32	0x000001FF
中断选择寄存器85	INT_SEL85	0x01B0	32	0x000001FF
中断选择寄存器86	INT_SEL86	0x01B4	32	0x000001FF
中断选择寄存器87	INT_SEL87	0x01B8	32	0x000001FF
中断选择寄存器88	INT_SEL88	0x01BC	32	0x000001FF
中断选择寄存器89	INT_SEL89	0x01C0	32	0x000001FF
中断选择寄存器90	INT_SEL90	0x01C4	32	0x000001FF
中断选择寄存器91	INT_SEL91	0x01C8	32	0x000001FF
中断选择寄存器92	INT_SEL92	0x01CC	32	0x000001FF

寄存器名	符号	偏移地址	位宽	复位值
中断选择寄存器93	INT_SEL93	0x01D0	32	0x000001FF
中断选择寄存器94	INT_SEL94	0x01D4	32	0x000001FF
中断选择寄存器95	INT_SEL95	0x01D8	32	0x000001FF
中断选择寄存器96	INT_SEL96	0x01DC	32	0x000001FF
中断选择寄存器97	INT_SEL97	0x01E0	32	0x000001FF
中断选择寄存器98	INT_SEL98	0x01E4	32	0x000001FF
中断选择寄存器99	INT_SEL99	0x01E8	32	0x000001FF
中断选择寄存器100	INT_SEL100	0x01EC	32	0x000001FF
中断选择寄存器101	INT_SEL101	0x01F0	32	0x000001FF
中断选择寄存器102	INT_SEL102	0x01F4	32	0x000001FF
中断选择寄存器103	INT_SEL103	0x01F8	32	0x000001FF
中断选择寄存器104	INT_SEL104	0x01FC	32	0x000001FF
中断选择寄存器105	INT_SEL105	0x0200	32	0x000001FF
中断选择寄存器106	INT_SEL106	0x0204	32	0x000001FF
中断选择寄存器107	INT_SEL107	0x0208	32	0x000001FF
中断选择寄存器108	INT_SEL108	0x020C	32	0x000001FF
中断选择寄存器109	INT_SEL109	0x0210	32	0x000001FF
中断选择寄存器110	INT_SEL110	0x0214	32	0x000001FF
中断选择寄存器111	INT_SEL111	0x0218	32	0x000001FF
中断选择寄存器112	INT_SEL112	0x021C	32	0x000001FF
中断选择寄存器113	INT_SEL113	0x0220	32	0x000001FF
中断选择寄存器114	INT_SEL114	0x0224	32	0x000001FF
中断选择寄存器115	INT_SEL115	0x0228	32	0x000001FF
中断选择寄存器116	INT_SEL116	0x022C	32	0x000001FF
中断选择寄存器117	INT_SEL117	0x0230	32	0x000001FF
中断选择寄存器118	INT_SEL118	0x0234	32	0x000001FF
中断选择寄存器119	INT_SEL119	0x0238	32	0x000001FF
中断选择寄存器120	INT_SEL120	0x023C	32	0x000001FF
中断选择寄存器121	INT_SEL121	0x0240	32	0x000001FF
中断选择寄存器122	INT_SEL122	0x0244	32	0x000001FF

寄存器名	符号	偏移地址	位宽	复位值
中断选择寄存器123	INT_SEL123	0x0248	32	0x000001FF
中断选择寄存器124	INT_SEL124	0x024C	32	0x000001FF
中断选择寄存器125	INT_SEL125	0x0250	32	0x000001FF
中断选择寄存器126	INT_SEL126	0x0254	32	0x000001FF
中断选择寄存器127	INT_SEL127	0x0258	32	0x000001FF
向量共享中断选择寄存器128	INT_VSSEL128	0x025C	32	0x00000000
向量共享中断选择寄存器129	INT_VSSEL129	0x0260	32	0x00000000
向量共享中断选择寄存器130	INT_VSSEL130	0x0264	32	0x00000000
向量共享中断选择寄存器131	INT_VSSEL131	0x0268	32	0x00000000
向量共享中断选择寄存器132	INT_VSSEL132	0x026C	32	0x00000000
向量共享中断选择寄存器133	INT_VSSEL133	0x0270	32	0x00000000
向量共享中断选择寄存器134	INT_VSSEL134	0x0274	32	0x00000000
向量共享中断选择寄存器135	INT_VSSEL135	0x0278	32	0x00000000
向量共享中断选择寄存器136	INT_VSSEL136	0x027C	32	0x00000000
向量共享中断选择寄存器137	INT_VSSEL137	0x0280	32	0x00000000
向量共享中断选择寄存器138	INT_VSSEL138	0x0284	32	0x00000000
向量共享中断选择寄存器139	INT_VSSEL139	0x0288	32	0x00000000
向量共享中断选择寄存器140	INT_VSSEL140	0x028C	32	0x00000000
向量共享中断选择寄存器141	INT_VSSEL141	0x0290	32	0x00000000
向量共享中断选择寄存器142	INT_VSSEL142	0x0294	32	0x00000000
向量共享中断选择寄存器143	INT_VSSEL143	0x0298	32	0x00000000
软件中断事件寄存器	INT_SWIER	0x029C	32	0x00000000
事件使能寄存器	INT_EVTER	0x02A0	32	0x00000000
中断使能寄存器	INT_IER	0x02A4	32	0xFFFFFFFF

外设事件触发寄存器

外设触发事件寄存器	INTSFTTRG	0x40010800	32	0x00000000

表 10-3 寄存器列表

10.5.1 外部管脚中断滤波控制寄存器（INT_NOCCR）

EIRQ Noise Cancel Control Register (INT_NOCCR)

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		NOCSEL[1:0]		Reserved											

位	标记	位名	功能	读写
b31~b14	Reserved	-	读出时为“0”，写入时写“0”	R/W
b13~b12	NOCSEL[1:0]	滤波器B滤波宽度选择	0 0：滤波宽度档位1 0 1：滤波宽度档位2 1 0：滤波宽度档位3 1 1：滤波宽度档位4 各档位具体值请参考**数据手册电气特性中EIRQ滤波特性**章节。 该滤波宽度对所有EIRQ管脚中断均有效。	R/W
b11~b0	Reserved	-	读出时为“0”，写入时写“0”	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
Reserved				WDTENR	BUSMENR	RDEDENR	PREENR	Reserved		XTALSTPENR	Res	PVD2ENR	PVD1ENR	SWDTENR	Res

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	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~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/W
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	Reserved	-	读出时为“0”，写入时写“0”	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
Reserved				WDTFR	BUSMFR	RDEDFR	RPEFR	Reserved		XTALSTPFR	Res	PVD2FR	PVD1FR	SWDTFR	Res

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	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~b6	Reserved	-	读出时为“0”，写入时写“0”	R/W
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/W
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	Reserved	-	读出时为“0”，写入时写“0”	R/W

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
Reserved				WDTCFR	BUSMCFR	RDEDCFR	RPECFR	Reserved		XTALSTPCFR	Res	PVD2CFR	PVD1CFR	SWDTCFR	Res

位	标记	位名	功能	读写
b31~b12	Reserved	-	读出时为“0”，写入时写“0”	R/W[注1]
b11	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	RDEDCFR	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	RPECFR	SRAM奇偶校验错误中断标志清除	0：无效 1：清除SRAM奇偶校验错误标志	R/W[注1]
b7~b6	Reserved	-	读出时为“0”，写入时写“0”	R/W[注1]
b5	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/W
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	Reserved	-	读出时为“0”，写入时写“0”	R/W[注1]

[注 1] 只可写“1”，读出时为“0”。

10.5.5 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
NOCEN	Reserved							EFEN	Res	EISMPCLK[1:0]		Reserved		EIRQTRG[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	NOCEN	EIRQ数字滤波器B使能	0：禁止数字滤波器B功能	R/W
			1：允许数字滤波器B功能	
b14~b8	Reserved	-	读出时为“0”，写入时写“0”	R/W
b7	EFEN	EIRQ数字滤波器A使能	0：禁止数字滤波器A功能	R/W
			1：允许数字滤波器A功能	
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	R/W
			0 1：PCLK3/8	
			1 0：PCLK3/32	
			1 1：PCLK3/64	
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	R/W
			0 1：上升沿	
			1 0：双边沿	
			1 1：低电平	

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
INT_EIFR15	INT_EIFR14	INT_EIFR13	INT_EIFR12	INT_EIFR11	INT_EIFR10	INT_EIFR9	INT_EIFR8	INT_EIFR7	INT_EIFR6	INT_EIFR5	INT_EIFR4	INT_EIFR3	INT_EIFR2	INT_EIFR1	INT_EIFR0

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	INT_EIFR	INT_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	

10.5.7 外部管脚中断标志清除寄存器(INT_EIFCR)

EIRQ Flag Clear Register (INT_EIFCR)

复位值：0x0000 0000

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

Interrup 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
Reserved							INT_SEL[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	INT_SEL[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）

Interrup 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
Reserved							INT_SEL[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	INT_SEL[8:0]	中断事件请求选择	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： 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。	R/W

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
SEL31	SEL30	SEL29	SEL28	SEL27	SEL26	SEL25	SEL24	SEL23	SEL22	SEL21	SEL20	SEL19	SEL18	SEL17	SEL16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SEL15	SEL14	SEL13	SEL12	SEL11	SEL10	SEL9	SEL8	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0

位	标记	位名	功能	读写
b31~b0	SEL[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
Reserved		ETH WUEN	USF WUEN	USH WUEN	RX WUEN	TMR2 OVFW UEN	TMR2 GCM WUEN	TMR0 GCM WUEN	RTC PRD WUEN	RTC ALM WUEN	WKTM WUEN	CMP WUEN	PVD2 WUEN	PVD1 WUEN	SWDT WUEN

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	Reserved	-	读出时为“0”，写入时写“0”	R/W
b30	Reserved	-	读出时为“0”，写入时写“0”	R/W
b29	ETHWUEN	ETH_PMT停止模式唤醒使能	0：唤醒禁止 1：唤醒许可	R/W
b28	USFWUEN	USBFS_WKUP停止模式唤醒使能	0：唤醒禁止 1：唤醒许可	R/W
b27	USHWUEN	USBHS_WKUP停止模式唤醒使能	0：唤醒禁止 1：唤醒许可	R/W
b26	RXWUEN	USART1_WUPI停止模式唤醒使能	0：唤醒禁止 1：唤醒许可	R/W
b25	TMR2OVFWUEN	TMR21_OVFA停止模式唤醒使能	0：唤醒禁止 1：唤醒许可	R/W
b24	TMR2GCMWUEN	TMR21_GCMA停止模式唤醒使能	0：唤醒禁止 1：唤醒许可	R/W
b23	TMR0GCMWUEN	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	CMPWUEN	CMP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	R/W

			1：唤醒许可	
b15~b0	EIRQWUEN[15:0]	EIRQ停止模式唤醒使能	0：唤醒禁止	R/W
			1：唤醒许可	

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
SWIE31	SWIE30	SWIE29	SWIE28	SWIE27	SWIE26	SWIE25	SWIE24	SWIE23	SWIE22	SWIE21	SWIE20	SWIE19	SWIE18	SWIE17	SWIE16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SWIE15	SWIE14	SWIE13	SWIE12	SWIE11	SWIE10	SWIE9	SWIE8	SWIE7	SWIE6	SWIE5	SWIE4	SWIE3	SWIE2	SWIE1	SWIE0

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
EVTE 31	EVTE 30	EVTE 29	EVTE 28	EVTE 27	EVTE 26	EVTE 25	EVTE 24	EVTE 23	EVTE 22	EVTE 21	EVTE 20	EVTE 19	EVTE 18	EVTE 17	EVTE 16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EVTE 15	EVTE 14	EVTE 13	EVTE 12	EVTE 11	EVTE 10	EVTE 9	EVTE 8	EVTE 7	EVTE 6	EVTE 5	EVTE 4	EVTE 3	EVTE 2	EVTE 1	EVTE0

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)

Interrup 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
INT_IER31	INT_IER30	INT_IER29	INT_IER28	INT_IER27	INT_IER26	INT_IER25	INT_IER24	INT_IER23	INT_IER22	INT_IER21	INT_IER20	INT_IER19	INT_IER18	INT_IER17	INT_IER16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
INT_IER15	INT_IER14	INT_IER13	INT_IER12	INT_IER11	INT_IER10	INT_IER 9	INT_IER 8	INT_IER 7	INT_IER 6	INT_IER 5	INT_IER 4	INT_IER 3	INT_IER 2	INT_IER 1	INT_IER 0

位	标记	位名	功能	读写
b31~b0	INT_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INT_SEL[8:0]选择的中断事件请求将不能被NVIC接收。 0：INT_SEL[8:0]选择的中断事件请求与软件中断事件请求被禁止 1：INT_SEL[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

10.5.15 内部触发事件寄存器（INTSFTTRG）

Peripheral Event Soft Trigger Register （INTSFTTRG)

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															STRG

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	STRG	周边外设事件软件触发位	0：不产生触发事件 1：产生触发事件 软件每次将该位置1，产生一次外设触发事件（AOS_STRG）。软件写0无效。	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)。

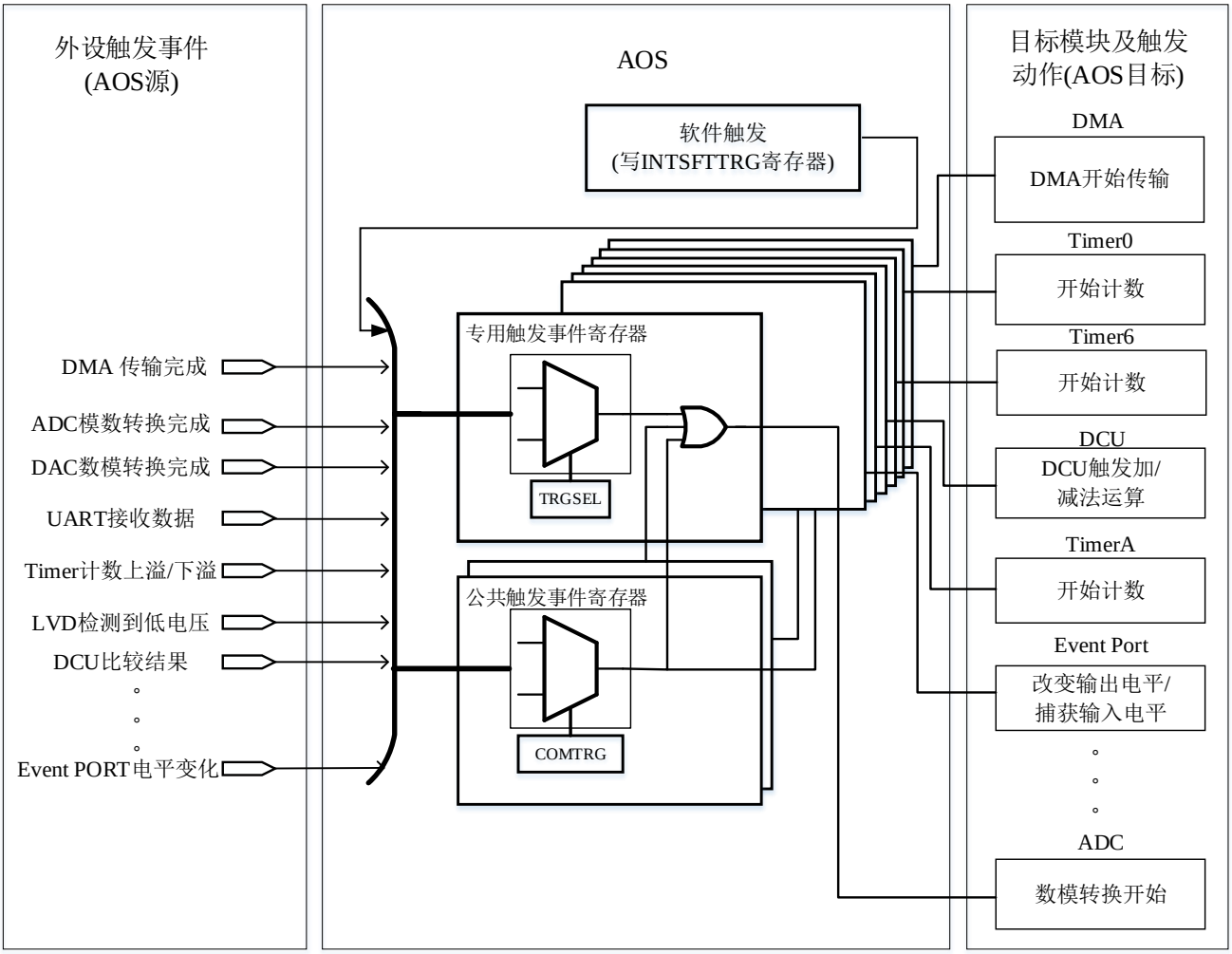
11.1.1 功能概览

- 共 368 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，其中 DCU1 和 DCU5 共用一个 AOS 目标，DCU2 和 DCU6 共用一个 AOS 目标，DCU3 和 DCU7 共用一个 AOS 目标，DCU4 和 DCU8 共用一个 AOS 目标
  - 17 个 DMA 触发目标，用于 2 个 8 通道 DMA 开始数据传输以及一个 DMA 事件触发通道重置
  - 4 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
  - 1 个通用定时器 2(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 HASH 触发目标
  - 4 个通用定时器 A(TimerA) 触发目标

– 1 个温度传感器(OTS) 触发目标

– 3 组每组 2 个 AD 触发目标，用于 AD1~AD3 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，三角波/锯齿波模式下触发数据递增或递减
DCU2	触发加/减法运算，三角波/锯齿波模式下触发数据递增或递减
DCU3	触发加/减法运算，三角波/锯齿波模式下触发数据递增或递减
DCU4	触发加/减法运算，三角波/锯齿波模式下触发数据递增或递减
DCU5	触发加/减法运算
DCU6	触发加/减法运算
DCU7	触发加/减法运算
DCU8	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 通道 4 开始传输 通道 5 开始传输 通道 6 开始传输 通道 7 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 通道 4 开始传输 通道 5 开始传输 通道 6 开始传输 通道 7 开始传输
DMA1&2	事件触发通道重置
Timer6	开始计数
Timer0	开始计数
Timer2	开始计数
HASH	启动 HASH 运算
Event Port	Event Port1&2 触发动作

	Event Port3&4 触发动作
TimerA	开始计数
OTS	开始测温
ADC1	开始模数转换
ADC2	开始模数转换
ADC3	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

11.4.1 寄存器一览

寄存器基地址：0x40010800

缩写	名称	偏移地址	所在章节
INTSFTTRG	外设触发事件寄存器	0x00	10.5.15
DCU_TRGSELx(x=1~4)	DCU 触发源选择寄存器	0x04,0x08,0x0C,0x10	44.3.9
DMA1_TRGSELx(x=0~7)	DMA1 传输启动触发源选择寄存器	0x14,0x18,0x1C,0x20 0x24,0x28,0x2C,0x30	15.5.14
DMA2_TRGSELx(x=0~7)	DMA2 传输启动触发源选择寄存器	0x34,0x38,0x3C,0x40 0x44,0x48,0x4C,0x50	15.5.14
DMA_TRGSELRC	DMA 通道重置触发源选择寄存器	0x54	15.5.15
TMR6_HTSSRx(x=0~3)	Timer6 硬件触发事件选择寄存器	0x58,0x5C,0x60,0x64	20.5.25
PEVNTTRGSR12	Event Port1,2 触发源选择寄存器	0x68	9.4.13
PEVNTTRGSR34	Event Port3,4 触发源选择寄存器	0x6C	9.4.13
TMR0_HTSSR	Timer0 触发选择寄存器	0x70	26.5.4
TMR2_HTSSR	Timer0 触发选择寄存器	0x74	25.5.7
HASH_ITRGSELA	HASH 硬件事件触发选择寄存器 A	0x78	42.3.7.4
HASH_ITRGSELB	HASH 硬件事件触发选择寄存器 B	0x7C	42.3.7.5
TMRA_HTSSRx(x=0~3)	TimerA 内部触发事件选择寄存器	0x80,0x84,0x88,0x8C	24.5.15
OTS_TRG	OTS 触发源选择寄存器	0x90	19.3.6
ADC1_ITRGSELRx(x=0,1)	A/D1 启动片内触发源选择寄存器	0x94，0x98	17.4.6
ADC2_ITRGSELRx(x=0,1)	A/D2 启动片内触发源选择寄存器	0x9C，0xA0	17.4.6
ADC3_ITRGSELRx(x=0,1)	A/D3 启动片内触发源选择寄存器	0xA4，0xA8	17.4.6
AOS_COMTRG1	公共触发源选择寄存器 1	0xAC	11.4.18
AOS_COMTRG2	公共触发源选择寄存器 2	0xB0	11.4.19

11.4.2 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：0x00000000

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

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

11.4.3 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。DCU1 将和 DCU5 共用寄存器 DCU_TRGSEL1，即当 DCU_TRGSEL1 写入事件编号，该编号事件发生时，会同时触发 DCU1 和 DCU5；DCU2 将和 DCU6 共用寄存器 DCU_TRGSEL2，即当 DCU_TRGSEL2 写入事件编号，该编号事件发生时，会同时触发 DCU2 和 DCU6；DCU3 将和 DCU7 共用寄存器 DCU_TRGSEL3，即当 DCU_TRGSEL3 写入事件编号，该编号事件发生时，会同时触发 DCU3 和 DCU7；DCU4 将和 DCU8 共用寄存器 DCU_TRGSEL4，即当 DCU_TRGSEL4 写入事件编号，该编号事件发生时，会同时触发 DCU4 和 DCU8。

偏移地址：0x04, 0x08, 0x0C, 0x10

复位值：0x000001FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0：禁止AOS_COMTRG1的公共触发事件触发本AOS目标 1：允许AOS_COMTRG1的公共触发事件触发本AOS目标	R/W
b30	COMEN[0]	公共触发使能	0：禁止AOS_COMTRG2的公共触发事件触发本AOS目标 1：允许AOS_COMTRG2的公共触发事件触发本AOS目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

11.4.4 DMA1 传输启动触发源选择寄存器(DMA1_TRGSELx)(x=0~7)

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,，x20，0x24，0x28，0x2C，0x30

复位值: 0x000001FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0：禁止AOS_COMTRG1的公共触发事件触发本AOS目标 1：允许AOS_COMTRG1的公共触发事件触发本AOS目标	R/W
b30	COMEN[0]	公共触发使能	0：禁止AOS_COMTRG2的公共触发事件触发本AOS目标 1：允许AOS_COMTRG2的公共触发事件触发本AOS目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

11.4.5 DMA2 传输启动触发源选择寄存器(DMA2_TRGSELx)(x=0~7)

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：0x34，0x38，0x3C,，x40，0x44，0x48，0x4C，0x50

复位值: 0x000001FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0：禁止AOS_COMTRG1的公共触发事件触发本AOS目标 1：允许AOS_COMTRG1的公共触发事件触发本AOS目标	R/W
b30	COMEN[0]	公共触发使能	0：禁止AOS_COMTRG2的公共触发事件触发本AOS目标 1：允许AOS_COMTRG2的公共触发事件触发本AOS目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

11.4.6 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：0x54

复位值: 0x000001FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0：禁止AOS_COMTRG1的公共触发事件触发本AOS目标 1：允许AOS_COMTRG1的公共触发事件触发本AOS目标	R/W
b30	COMEN[0]	公共触发使能	0：禁止AOS_COMTRG2的公共触发事件触发本AOS目标 1：允许AOS_COMTRG2的公共触发事件触发本AOS目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

11.4.7 Timer6 硬件触发事件选择寄存器(TMR6_HTSSRx)(x=0~3)

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：0x58，0x5C，0x60，0x64

复位值: 0x000001FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0：禁止AOS_COMTRG1的公共触发事件触发本AOS目标 1：允许AOS_COMTRG1的公共触发事件触发本AOS目标	R/W
b30	COMEN[0]	公共触发使能	0：禁止AOS_COMTRG2的公共触发事件触发本AOS目标 1：允许AOS_COMTRG2的公共触发事件触发本AOS目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

11.4.8 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：0x68，0x6C

复位值: 0x000001FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0：禁止AOS_COMTRG1的公共触发事件触发本AOS目标 1：允许AOS_COMTRG1的公共触发事件触发本AOS目标	R/W
b30	COMEN[0]	公共触发使能	0：禁止AOS_COMTRG2的公共触发事件触发本AOS目标 1：允许AOS_COMTRG2的公共触发事件触发本AOS目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

11.4.9 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：0x70

复位值: 0x000001FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0：禁止AOS_COMTRG1的公共触发事件触发本AOS目标 1：允许AOS_COMTRG1的公共触发事件触发本AOS目标	R/W
b30	COMEN[0]	公共触发使能	0：禁止AOS_COMTRG2的公共触发事件触发本AOS目标 1：允许AOS_COMTRG2的公共触发事件触发本AOS目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

11.4.10 Timer2 硬件触发事件选择寄存器(TMR2_HTSSR)

寄存器说明：在 Timer2 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，Timer2 将由该事件触发启动。

偏移地址：0x74

复位值: 0x000001FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0：禁止AOS_COMTRG1的公共触发事件触发本AOS目标 1：允许AOS_COMTRG1的公共触发事件触发本AOS目标	R/W
b30	COMEN[0]	公共触发使能	0：禁止AOS_COMTRG2的公共触发事件触发本AOS目标 1：允许AOS_COMTRG2的公共触发事件触发本AOS目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

11.4.11 HASH 硬件事件触发选择寄存器 A（HASH_ITRGSELA）

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

偏移地址：0x78

复位值: 0x000001FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0：禁止AOS_COMTRG1的公共触发事件触发本AOS目标 1：允许AOS_COMTRG1的公共触发事件触发本AOS目标	R/W
b30	COMEN[0]	公共触发使能	0：禁止AOS_COMTRG2的公共触发事件触发本AOS目标 1：允许AOS_COMTRG2的公共触发事件触发本AOS目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

11.4.12 HASH 硬件事件触发选择寄存器 B（HASH_ITRGSELB）

寄存器说明：当最后一组数据写入 HASH_DR 后，通过该寄存器选择硬件事件触发源，以通知 HASH 进行最后一次运算。注意，请勿选择 DMA_TCx(x=0~7)以外的其他触发信号。

偏移地址：0x7C

复位值: 0x000001FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0：禁止AOS_COMTRG1的公共触发事件触发本AOS目标 1：允许AOS_COMTRG1的公共触发事件触发本AOS目标	R/W
b30	COMEN[0]	公共触发使能	0：禁止AOS_COMTRG2的公共触发事件触发本AOS目标 1：允许AOS_COMTRG2的公共触发事件触发本AOS目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

11.4.13 TimerA 硬件触发事件选择寄存器(TMRA_HTSSRx)(x=0~3)

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：0x80，0x84，0x88，0x8C

复位值: 0x000001FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0：禁止AOS_COMTRG1的公共触发事件触发本AOS目标 1：允许AOS_COMTRG1的公共触发事件触发本AOS目标	R/W
b30	COMEN[0]	公共触发使能	0：禁止AOS_COMTRG2的公共触发事件触发本AOS目标 1：允许AOS_COMTRG2的公共触发事件触发本AOS目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

11.4.14 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：0x90

复位值: 0x000001FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0：禁止AOS_COMTRG1的公共触发事件触发本AOS目标 1：允许AOS_COMTRG1的公共触发事件触发本AOS目标	R/W
b30	COMEN[0]	公共触发使能	0：禁止AOS_COMTRG2的公共触发事件触发本AOS目标 1：允许AOS_COMTRG2的公共触发事件触发本AOS目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

11.4.15 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：0x94，0x98

复位值: 0x000001FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0：禁止AOS_COMTRG1的公共触发事件触发本AOS目标 1：允许AOS_COMTRG1的公共触发事件触发本AOS目标	R/W
b30	COMEN[0]	公共触发使能	0：禁止AOS_COMTRG2的公共触发事件触发本AOS目标 1：允许AOS_COMTRG2的公共触发事件触发本AOS目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

11.4.16 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：0x9C，0xA0

复位值: 0x000001FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0：禁止AOS_COMTRG1的公共触发事件触发本AOS目标 1：允许AOS_COMTRG1的公共触发事件触发本AOS目标	R/W
b30	COMEN[0]	公共触发使能	0：禁止AOS_COMTRG2的公共触发事件触发本AOS目标 1：允许AOS_COMTRG2的公共触发事件触发本AOS目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

11.4.17 A/D3 转换开始片内触发源选择寄存器 ADC3_ITRGSELRx(x=0,1)

寄存器说明：在 ADC3 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，ADC3 将由该事件触发启动。

偏移地址：0xA4，0xA8

复位值: 0x000001FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0：禁止AOS_COMTRG1的公共触发事件触发本AOS目标 1：允许AOS_COMTRG1的公共触发事件触发本AOS目标	R/W
b30	COMEN[0]	公共触发使能	0：禁止AOS_COMTRG2的公共触发事件触发本AOS目标 1：允许AOS_COMTRG2的公共触发事件触发本AOS目标	R/W
b29~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

11.4.18 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：0xAC

复位值: 0x000001FF

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

位	标记	位名	功能	读写
b31~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	COMTRG[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

11.4.19 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：0xB0

复位值: 0x000001FF

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

位	标记	位名	功能	读写
b31~b9	Reserved	-	读出时为“0”，写入时写“0”	R/W
b8~b0	COMTRG[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

12 存储保护单元（MPU）

12.1 简介

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

本芯片内置了六个针对主机的 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
USBHS-DMA MPU：HMPU	USBHS-DMA 的存储保护单元 8 区域，所有 DMA 共用
ETH-DMA MPU：EMPU	ETH-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/HMPU/EMPU 分别提供系统 DMA_1/系统 DMA_2/USBFS-DMA/USBHS-DMA/ETH-DMA 对全部 4G 地址空间的读写访问权限控制。对禁止空间发生访问时，可以设置 MPU 动作为无视/总线错误/不可屏蔽中断/复位。

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

12.2 功能描述

12.2.1 区域范围设置

MPU 以区域为单位对存储空间进行权限管理。每个区域可以独立设定基地址和区域大小，可设的范围为 32Byte～4GByte，大小必须为 2^nByte（n=5~32），对应的基地址低 n 位为 0。

没有被任何区域覆盖的地址空间称为背景区域。

12.2.2 权限设置

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

12.2.3 MPU 动作选择

发生被禁止的访问时，该访问被无视（读访问读到 0，写访问忽略）的同时，可以设置对应的动作，可以设置为：

- 无视
- 总线错误
- 不可屏蔽中断
- 复位

12.2.4 启动 MPU

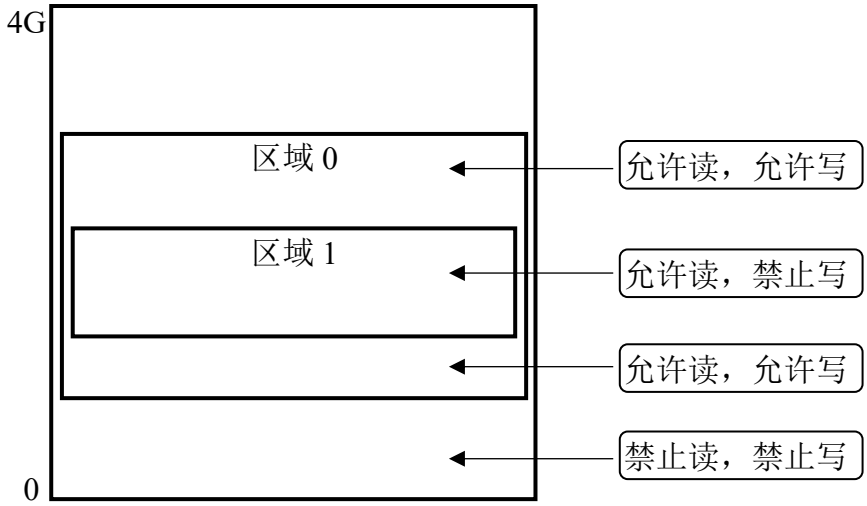
SMPU1/ SMPU2/FMPU/HMPU/EMPU 可以独立使能。

建议在设置好区域范围/权限设置/动作选择后再使能 MPU。

12.3 应用举例

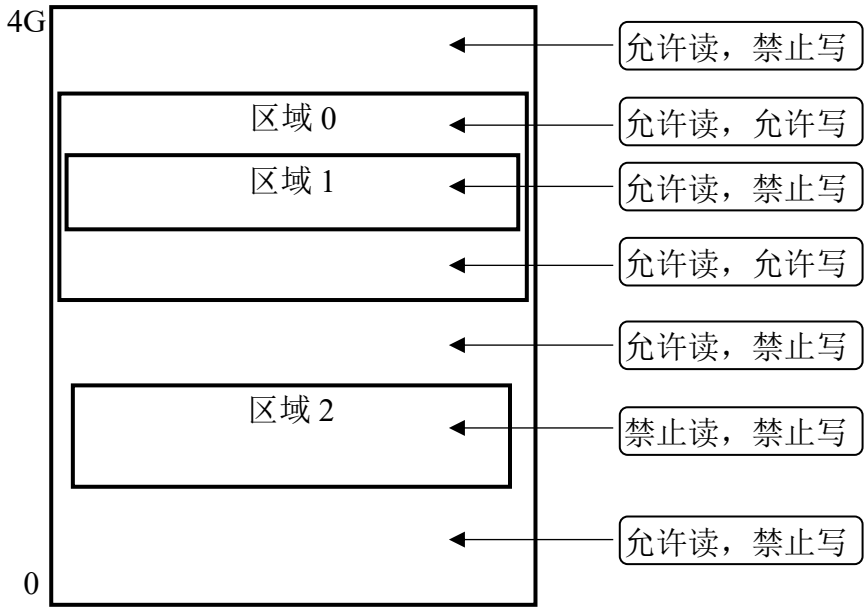
12.3.1 只允许部分空间访问

例：将背景区域权限设置为禁止读/禁止写，区域 0 设置为允许读/允许写，区域 1 设置为允许读禁止写，区域 0 范围覆盖区域 1。



12.3.2 只禁止部分空间访问

例：将背景区域权限设置为允许读/禁止写，区域 0 设置为允许读/允许写，区域 1 设置为允许读/禁止写，区域 0 覆盖区域 1，区域 2 设置为禁止读/禁止写。



12.4 寄存器说明

本模块的寄存器只能通过 CPU 来设置。

MPU 基准地址: 0x40050000

偏移地址	寄存器名	初始值	名称	写保护
+00～+3C	MPU_RGD0~15	0x00000000	区域 0~15 范围描述寄存器	MPUWE
+40	MPU_SR	0x00000000	MPU 状态寄存器	无
+44	MPU_ECLR	0x00000000	MPU 错误标志清除寄存器	无
+48	MPU_WP	0x00000000	MPU 写保护寄存器	WKEY
+4C	MPU_IPPR	0x00000000	IP 访问保护寄存器	MPUWE
+50	MPU_S1RGE	0x00000000	SMPU1 区域使能寄存器	MPUWE
+54	MPU_S1RGWP	0x00000000	SMPU1 区域写权限寄存器	MPUWE
+58	MPU_S1RGRP	0x00000000	SMPU1 区域读权限寄存器	MPUWE
+5C	MPU_S1CR	0x00000000	SMPU1 控制寄存器	MPUWE
+60	MPU_S2RGE	0x00000000	SMPU2 区域使能寄存器	MPUWE
+64	MPU_S2RGWP	0x00000000	SMPU2 区域写权限寄存器	MPUWE
+68	MPU_S2RGRP	0x00000000	SMPU2 区域读权限寄存器	MPUWE
+6C	MPU_S2CR	0x00000000	SMPU2 控制寄存器	MPUWE
+70	MPU_FRGE	0x00000000	FMPU 区域使能寄存器	MPUWE
+74	MPU_FRGWP	0x00000000	FMPU 区域写权限寄存器	MPUWE
+78	MPU_FRGRP	0x00000000	FMPU 区域读权限寄存器	MPUWE
+7C	MPU_FCR	0x00000000	FMPU 控制寄存器	MPUWE
+80	MPU_HRGE	0x00000000	HMPU 区域使能寄存器	MPUWE
+84	MPU_HRGWP	0x00000000	HMPU 区域写权限寄存器	MPUWE
+88	MPU_HRGRP	0x00000000	HMPU 区域读权限寄存器	MPUWE
+8C	MPU_HCR	0x00000000	HMPU 控制寄存器	MPUWE
+90	MPU_ERGE	0x00000000	EMPU 区域使能寄存器	MPUWE
+94	MPU_ERGWP	0x00000000	EMPU 区域写权限寄存器	MPUWE

+98	MPU_ERGRP	0x00000000	EMPU 区域读权限寄存器	MPUWE
+9C	MPU_ECR	0x00000000	EMPU 控制寄存器	MPUWE

12.4.1 区域范围描述寄存器 MPU_RGDn（n=0～15）

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
MPURGnADDR [31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MPURGnADDR[15:5]											MPURGnSIZE[4:0]				

位	标记	位名	功能	读写
b31~b5	MPURGnADDR[31:5]	区域基地址	设定区域 n 的基地址，有效位数与区域大小有关，低(MPURGnSIZE+1)位固定为 0	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

12.4.2 状态标志寄存器 MPU_SR

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved											EMPUEAF	HMPUEAF	FMPUEAF	SMPU2EAF	SMPU1EAF

位	标记	位名	功能	读写
b31~b5	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b4	EMPUEAF	EMPU 错误标志	0：ETH-DMA 未发生错误访问 1：ETH-DMA 发生了错误访问	R
b3	HMPUEAF	HMPU 错误标志	0：USBHS -DMA 未发生错误访问 1：USBHS -DMA 发生了错误访问	R
b2	FMPUEAF	FMPU 错误标志	0：USBFS-DMA 未发生错误访问 1：USBFS-DMA 发生了错误访问	R
b1	SMPU2EAF	SMPU2 错误标志	0：系统 DMA_2 未发生错误访问 1：系统 DMA_2 发生了错误访问	R
b0	SMPU1EAF	SMPU1 错误标志	0：系统 DMA_1 未发生错误访问 1：系统 DMA_1 发生了错误访问	R

对本寄存器的写操作会被忽略，清除错误标志请使用 MPUECLR。

12.4.3 标志清除寄存器 MPU_ECLR

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved											EMPUECLR	HMPUECLR	FMPUECLR	SMPU2ECLR	SMPU1ECLR

位	标记	位名	功能	读写
b31~b5	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b4	EMPUECLR	EMPU 错误标志清除	写入 1 可以将 EMPUEAF 清 0	R/W
b3	HMPUECLR	HMPU 错误标志清除	写入 1 可以将 HMPUEAF 清 0	R/W
b2	FMPUECLR	FMPU 错误标志清除	写入 1 可以将 FMPUEAF 清 0	R/W
b1	SMPU2ECLR	SMPU2 错误标志清除	写入 1 可以将 SMPU2EAF 清 0	R/W
b0	SMPU1ECLR	SMPU1 错误标志清除	写入 1 可以将 SMPU1EAF 清 0	R/W

本寄存器读出值固定为 0x00000000。

12.4.4 写保护寄存器 MPU_WP

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
WKEY[15:1]															MPUWE

位	标记	位名	功能	读写
b31~b6	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b15~b1	WKEY[15:1]	写入码	在写入 MPUWE 时，必须同时向 WKEY 写入 ’b1001_0110_1010_0101，读出为 0	R/W
b0	MPUWE	MPU 寄存器写允许	0：MPU 地址寄存器/控制寄存器不允许写入 1：MPU 地址寄存器/控制寄存器允许写入	RW

向本寄存器写入 0x96a5 可以将 MPUWE 置 1，写入 0x96a4 可以将 MPUWE 清 0，写入其他值不能改变 MPUWE。

12.4.5 SMPU1 区域使能寄存器 MPU_S1RGE

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S1RG15E	S1RG14E	S1RG13E	S1RG12E	S1RG11E	S1RG10E	S1RG9E	S1RG8E	S1RG7E	S1RG6E	S1RG5E	S1RG4E	S1RG3E	S1RG2E	S1RG1E	S1RG0E

位	标记	位名	功能	读写
b31~b16	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b15	S1RG15E	SMPU1 区域 15 使能	0：SMPU1 的区域 15 无效 1：SMPU1 的区域 15 有效	R/W
b14	S1RG14E	SMPU1 区域 14 使能	0：SMPU1 的区域 14 无效 1：SMPU1 的区域 14 有效	R/W
b13	S1RG13E	SMPU1 区域 13 使能	0：SMPU1 的区域 13 无效 1：SMPU1 的区域 13 有效	R/W
b12	S1RG12E	SMPU1 区域 12 使能	0：SMPU1 的区域 12 无效 1：SMPU1 的区域 12 有效	R/W
b11	S1RG11E	SMPU1 区域 11 使能	0：SMPU1 的区域 11 无效 1：SMPU1 的区域 11 有效	R/W
b10	S1RG10E	SMPU1 区域 10 使能	0：SMPU1 的区域 10 无效 1：SMPU1 的区域 10 有效	R/W
b9	S1RG9E	SMPU1 区域 9 使能	0：SMPU1 的区域 9 无效 1：SMPU1 的区域 9 有效	R/W
b8	S1RG8	SMPU1 区域 8 使能	0：SMPU1 的区域 8 无效 1：SMPU1 的区域 8 有效	R/W
b7	S1RG7E	SMPU1 区域 7 使能	0：SMPU1 的区域 7 无效 1：SMPU1 的区域 7 有效	R/W

b6	S1RG6E	SMPU1 区域 6 使能	0：SMPU1 的区域 6 无效 1：SMPU1 的区域 6 有效	R/W
b5	S1RG5E	SMPU1 区域 5 使能	0：SMPU1 的区域 5 无效 1：SMPU1 的区域 5 有效	R/W
b4	S1RG4E	SMPU1 区域 4 使能	0：SMPU1 的区域 4 无效 1：SMPU1 的区域 4 有效	R/W
b3	S1RG3E	SMPU1 区域 3 使能	0：SMPU1 的区域 3 无效 1：SMPU1 的区域 3 有效	R/W
b2	S1RG2E	SMPU1 区域 2 使能	0：SMPU1 的区域 2 无效 1：SMPU1 的区域 2 有效	R/W
b1	S1RG1E	SMPU1 区域 1 使能	0：SMPU1 的区域 1 无效 1：SMPU1 的区域 1 有效	R/W
b0	S1RG0E	SMPU1 区域 0 使能	0：SMPU1 的区域 0 无效 1：SMPU1 的区域 0 有效	R/W

12.4.6 SMPU1 区域写权限寄存器 MPU_S1RGWP

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S1RG15WP	S1RG14WP	S1RG13WP	S1RG12WP	S1RG11WP	S1RG10WP	S1RG9WP	S1RG8WP	S1RG7WP	S1RG6WP	S1RG5WP	S1RG4WP	S1RG3WP	S1RG2WP	S1RG1WP	S1RG0WP

位	标记	位名	功能	读写
b31~b16	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b15	S1RG15WP	SMPU1 区域 15 写权限	0：区域 15 允许系统 DMA_1 写 1：区域 15 禁止系统 DMA_1 写	R/W
b14	S1RG14WP	SMPU1 区域 14 写权限	0：区域 14 允许系统 DMA_1 写 1：区域 14 禁止系统 DMA_1 写	R/W
b13	S1RG13WP	SMPU1 区域 13 写权限	0：区域 13 允许系统 DMA_1 写 1：区域 13 禁止系统 DMA_1 写	R/W
b12	S1RG12WP	SMPU1 区域 12 写权限	0：区域 12 允许系统 DMA_1 写 1：区域 12 禁止系统 DMA_1 写	R/W
b11	S1RG11WP	SMPU1 区域 11 写权限	0：区域 11 允许系统 DMA_1 写 1：区域 11 禁止系统 DMA_1 写	R/W
b10	S1RG10WP	SMPU1 区域 10 写权限	0：区域 10 允许系统 DMA_1 写 1：区域 10 禁止系统 DMA_1 写	R/W
b9	S1RG9WP	SMPU1 区域 9 写权限	0：区域 9 允许系统 DMA_1 写 1：区域 9 禁止系统 DMA_1 写	R/W
b8	S1RG8WP	SMPU1 区域 8 写权限	0：区域 8 允许系统 DMA_1 写 1：区域 8 禁止系统 DMA_1 写	R/W
b7	S1RG7WP	SMPU1 区域 7 写权限	0：区域 7 允许系统 DMA_1 写 1：区域 7 禁止系统 DMA_1 写	R/W

b6	S1RG6WP	SMPU1 区域 6 写权限	0：区域 6 允许系统 DMA_1 写 1：区域 6 禁止系统 DMA_1 写	R/W
b5	S1RG5WP	SMPU1 区域 5 写权限	0：区域 5 允许系统 DMA_1 写 1：区域 5 禁止系统 DMA_1 写	R/W
b4	S1RG4WP	SMPU1 区域 4 写权限	0：区域 4 允许系统 DMA_1 写 1：区域 4 禁止系统 DMA_1 写	R/W
b3	S1RG3WP	SMPU1 区域 3 写权限	0：区域 3 允许系统 DMA_1 写 1：区域 3 禁止系统 DMA_1 写	R/W
b2	S1RG2WP	SMPU1 区域 2 写权限	0：区域 2 允许系统 DMA_1 写 1：区域 2 禁止系统 DMA_1 写	R/W
b1	S1RG1WP	SMPU1 区域 1 写权限	0：区域 1 允许系统 DMA_1 写 1：区域 1 禁止系统 DMA_1 写	R/W
b0	S1RG0WP	SMPU1 区域 0 写权限	0：区域 0 允许系统 DMA_1 写 1：区域 0 禁止系统 DMA_1 写	R/W

12.4.7 SMPU1 区域读权限寄存器 MPU_S1RGRP

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S1RG15RP	S1RG14RP	S1RG13RP	S1RG12RP	S1RG11RP	S1RG10RP	S1RG9RP	S1RG8RP	S1RG7RP	S1RG6RP	S1RG5RP	S1RG4RP	S1RG3RP	S1RG2RP	S1RG1RP	S1RG0RP

位	标记	位名	功能	读写
b31~b16	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b15	S1RG15RP	SMPU1 区域 15 读权限	0：区域 15 允许系统 DMA_1 读 1：区域 15 禁止系统 DMA_1 读	R/W
b14	S1RG14RP	SMPU1 区域 14 读权限	0：区域 14 允许系统 DMA_1 读 1：区域 14 禁止系统 DMA_1 读	R/W
b13	S1RG13RP	SMPU1 区域 13 读权限	0：区域 13 允许系统 DMA_1 读 1：区域 13 禁止系统 DMA_1 读	R/W
b12	S1RG12RP	SMPU1 区域 12 读权限	0：区域 12 允许系统 DMA_1 读 1：区域 12 禁止系统 DMA_1 读	R/W
b11	S1RG11RP	SMPU1 区域 11 读权限	0：区域 11 允许系统 DMA_1 读 1：区域 11 禁止系统 DMA_1 读	R/W
b10	S1RG10RP	SMPU1 区域 10 读权限	0：区域 10 允许系统 DMA_1 读 1：区域 10 禁止系统 DMA_1 读	R/W
b9	S1RG9RP	SMPU1 区域 9 读权限	0：区域 9 允许系统 DMA_1 读 1：区域 9 禁止系统 DMA_1 读	R/W
b8	S1RG8RP	SMPU1 区域 8 读权限	0：区域 8 允许系统 DMA_1 读 1：区域 8 禁止系统 DMA_1 读	R/W
b7	S1RG7RP	SMPU1 区域 7 读权限	0：区域 7 允许系统 DMA_1 读 1：区域 7 禁止系统 DMA_1 读	R/W

b6	S1RG6RP	SMPU1 区域 6 读权限	0：区域 6 允许系统 DMA_1 读 1：区域 6 禁止系统 DMA_1 读	R/W
b5	S1RG5RP	SMPU1 区域 5 读权限	0：区域 5 允许系统 DMA_1 读 1：区域 5 禁止系统 DMA_1 读	R/W
b4	S1RG4RP	SMPU1 区域 4 读权限	0：区域 4 允许系统 DMA_1 读 1：区域 4 禁止系统 DMA_1 读	R/W
b3	S1RG3RP	SMPU1 区域 3 读权限	0：区域 3 允许系统 DMA_1 读 1：区域 3 禁止系统 DMA_1 读	R/W
b2	S1RG2RP	SMPU1 区域 2 读权限	0：区域 2 允许系统 DMA_1 读 1：区域 2 禁止系统 DMA_1 读	R/W
b1	S1RG1RP	SMPU1 区域 1 读权限	0：区域 1 允许系统 DMA_1 读 1：区域 1 禁止系统 DMA_1 读	R/W
b0	S1RG0RP	SMPU1 区域 0 读权限	0：区域 0 允许系统 DMA_1 读 1：区域 0 禁止系统 DMA_1 读	R/W

12.4.8 SMPU1 控制寄存器 MPU_S1CR

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								SMPU1E	Reserved			SMPU1ACT[1:0]		SMPU1BWP	SMPU1BRP

位	标记	位名	功能	读写
b31~b8	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b7	SMPU1E	SMPU1 使能	0：SMPU1 无效 1：SMPU1 有效	R/W
b6~b4	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b3~b2	SMPU1ACT[1:0]	SMPU1 动作选择	设定系统 DMA_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
b1	SMPU1BWP	SMPU1 背景写权限设置	0：SMPU1 背景空间允许系统 DMA_1 写 1：SMPU1 背景空间禁止系统 DMA_1 写	R/W
b0	SMPU1BRP	SMPU1 背景读权限设置	0：SMPU1 背景空间允许系统 DMA_1 读 1：SMPU1 背景空间禁止系统 DMA_1 读	R/W

多个区域设定重叠时，优先级为：设定的禁止>设定的允许

12.4.9 SMPU2 区域使能寄存器 MPU_S2RGE

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S2RG15E	S2RG14E	S2RG13E	S2RG12E	S2RG11E	S2RG10E	S2RG9E	S2RG8E	S2RG7E	S2RG6E	S2RG5E	S2RG4E	S2RG3E	S2RG2E	S2RG1E	S2RG0E

位	标记	位名	功能	读写
b31~b16	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b15	S2RG15E	SMPU2 区域 15 使能	0：SMPU2 的区域 15 无效 1：SMPU2 的区域 15 有效	R/W
b14	S2RG14E	SMPU2 区域 14 使能	0：SMPU2 的区域 14 无效 1：SMPU2 的区域 14 有效	R/W
b13	S2RG13E	SMPU2 区域 13 使能	0：SMPU2 的区域 13 无效 1：SMPU2 的区域 13 有效	R/W
b12	S2RG12E	SMPU2 区域 12 使能	0：SMPU2 的区域 12 无效 1：SMPU2 的区域 12 有效	R/W
b11	S2RG11E	SMPU2 区域 11 使能	0：SMPU2 的区域 11 无效 1：SMPU2 的区域 11 有效	R/W
b10	S2RG10E	SMPU2 区域 10 使能	0：SMPU2 的区域 10 无效 1：SMPU2 的区域 10 有效	R/W
b9	S2RG9E	SMPU2 区域 9 使能	0：SMPU2 的区域 9 无效 1：SMPU2 的区域 9 有效	R/W
b8	S2RG8	SMPU2 区域 8 使能	0：SMPU2 的区域 8 无效 1：SMPU2 的区域 8 有效	R/W
b7	S2RG7E	SMPU2 区域 7 使能	0：SMPU2 的区域 7 无效 1：SMPU2 的区域 7 有效	R/W

b6	S2RG6E	SMPU2 区域 6 使能	0：SMPU2 的区域 6 无效 1：SMPU2 的区域 6 有效	R/W
b5	S2RG5E	SMPU2 区域 5 使能	0：SMPU2 的区域 5 无效 1：SMPU2 的区域 5 有效	R/W
b4	S2RG4E	SMPU2 区域 4 使能	0：SMPU2 的区域 4 无效 1：SMPU2 的区域 4 有效	R/W
b3	S2RG3E	SMPU2 区域 3 使能	0：SMPU2 的区域 3 无效 1：SMPU2 的区域 3 有效	R/W
b2	S2RG2E	SMPU2 区域 2 使能	0：SMPU2 的区域 2 无效 1：SMPU2 的区域 2 有效	R/W
b1	S2RG1E	SMPU2 区域 1 使能	0：SMPU2 的区域 1 无效 1：SMPU2 的区域 1 有效	R/W
b0	S2RG0E	SMPU2 区域 0 使能	0：SMPU2 的区域 0 无效 1：SMPU2 的区域 0 有效	R/W

12.4.10 SMPU2 区域写权限寄存器 MPU_S2RGWP

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S2RG15WP	S2RG14WP	S2RG13WP	S2RG12WP	S2RG11WP	S2RG10WP	S2RG9WP	S2RG8WP	S2RG7WP	S2RG6WP	S2RG5WP	S2RG4WP	S2RG3WP	S2RG2WP	S2RG1WP	S2RG0WP

位	标记	位名	功能	读写
b31~b16	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b15	S2RG15WP	SMPU2 区域 15 写权限	0：区域 15 允许系统 DMA_2 写 1：区域 15 禁止系统 DMA_2 写	R/W
b14	S2RG14WP	SMPU2 区域 14 写权限	0：区域 14 允许系统 DMA_2 写 1：区域 14 禁止系统 DMA_2 写	R/W
b13	S2RG13WP	SMPU2 区域 13 写权限	0：区域 13 允许系统 DMA_2 写 1：区域 13 禁止系统 DMA_2 写	R/W
b12	S2RG12WP	SMPU2 区域 12 写权限	0：区域 12 允许系统 DMA_2 写 1：区域 12 禁止系统 DMA_2 写	R/W
b11	S2RG11WP	SMPU2 区域 11 写权限	0：区域 11 允许系统 DMA_2 写 1：区域 11 禁止系统 DMA_2 写	R/W
b10	S2RG10WP	SMPU2 区域 10 写权限	0：区域 10 允许系统 DMA_2 写 1：区域 10 禁止系统 DMA_2 写	R/W
b9	S2RG9WP	SMPU2 区域 9 写权限	0：区域 9 允许系统 DMA_2 写 1：区域 9 禁止系统 DMA_2 写	R/W
b8	S2RG8WP	SMPU2 区域 8 写权限	0：区域 8 允许系统 DMA_2 写 1：区域 8 禁止系统 DMA_2 写	R/W
b7	S2RG7WP	SMPU2 区域 7 写权限	0：区域 7 允许系统 DMA_2 写 1：区域 7 禁止系统 DMA_2 写	R/W

b6	S2RG6WP	SMPU2 区域 6 写权限	0：区域 6 允许系统 DMA_2 写 1：区域 6 禁止系统 DMA_2 写	R/W
b5	S2RG5WP	SMPU2 区域 5 写权限	0：区域 5 允许系统 DMA_2 写 1：区域 5 禁止系统 DMA_2 写	R/W
b4	S2RG4WP	SMPU2 区域 4 写权限	0：区域 4 允许系统 DMA_2 写 1：区域 4 禁止系统 DMA_2 写	R/W
b3	S2RG3WP	SMPU2 区域 3 写权限	0：区域 3 允许系统 DMA_2 写 1：区域 3 禁止系统 DMA_2 写	R/W
b2	S2RG2WP	SMPU2 区域 2 写权限	0：区域 2 允许系统 DMA_2 写 1：区域 2 禁止系统 DMA_2 写	R/W
b1	S2RG1WP	SMPU2 区域 1 写权限	0：区域 1 允许系统 DMA_2 写 1：区域 1 禁止系统 DMA_2 写	R/W
b0	S2RG0WP	SMPU2 区域 0 写权限	0：区域 0 允许系统 DMA_2 写 1：区域 0 禁止系统 DMA_2 写	R/W

12.4.11 SMPU2 区域读权限寄存器 MPU_S2RGRP

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S2RG15RP	S2RG14RP	S2RG13RP	S2RG12RP	S2RG11RP	S2RG10RP	S2RG9RP	S2RG8RP	S2RG7RP	S2RG6RP	S2RG5RP	S2RG4RP	S2RG3RP	S2RG2RP	S2RG1RP	S2RG0RP

位	标记	位名	功能	读写
b31~b16	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b15	S2RG15RP	SMPU2 区域 15 读权限	0：区域 15 允许系统 DMA_2 读 1：区域 15 禁止系统 DMA_2 读	R/W
b14	S2RG14RP	SMPU2 区域 14 读权限	0：区域 14 允许系统 DMA_2 读 1：区域 14 禁止系统 DMA_2 读	R/W
b13	S2RG13RP	SMPU2 区域 13 读权限	0：区域 13 允许系统 DMA_2 读 1：区域 13 禁止系统 DMA_2 读	R/W
b12	S2RG12RP	SMPU2 区域 12 读权限	0：区域 12 允许系统 DMA_2 读 1：区域 12 禁止系统 DMA_2 读	R/W
b11	S2RG11RP	SMPU2 区域 11 读权限	0：区域 11 允许系统 DMA_2 读 1：区域 11 禁止系统 DMA_2 读	R/W
b10	S2RG10RP	SMPU2 区域 10 读权限	0：区域 10 允许系统 DMA_2 读 1：区域 10 禁止系统 DMA_2 读	R/W
b9	S2RG9RP	SMPU2 区域 9 读权限	0：区域 9 允许系统 DMA_2 读 1：区域 9 禁止系统 DMA_2 读	R/W
b8	S2RG8RP	SMPU2 区域 8 读权限	0：区域 8 允许系统 DMA_2 读 1：区域 8 禁止系统 DMA_2 读	R/W
b7	S2RG7RP	SMPU2 区域 7 读权限	0：区域 7 允许系统 DMA_2 读 1：区域 7 禁止系统 DMA_2 读	R/W

b6	S2RG6RP	SMPU2 区域 6 读权限	0：区域 6 允许系统 DMA_2 读 1：区域 6 禁止系统 DMA_2 读	R/W
b5	S2RG5RP	SMPU2 区域 5 读权限	0：区域 5 允许系统 DMA_2 读 1：区域 5 禁止系统 DMA_2 读	R/W
b4	S2RG4RP	SMPU2 区域 4 读权限	0：区域 4 允许系统 DMA_2 读 1：区域 4 禁止系统 DMA_2 读	R/W
b3	S2RG3RP	SMPU2 区域 3 读权限	0：区域 3 允许系统 DMA_2 读 1：区域 3 禁止系统 DMA_2 读	R/W
b2	S2RG2RP	SMPU2 区域 2 读权限	0：区域 2 允许系统 DMA_2 读 1：区域 2 禁止系统 DMA_2 读	R/W
b1	S2RG1RP	SMPU2 区域 1 读权限	0：区域 1 允许系统 DMA_2 读 1：区域 1 禁止系统 DMA_2 读	R/W
b0	S2RG0RP	SMPU2 区域 0 读权限	0：区域 0 允许系统 DMA_2 读 1：区域 0 禁止系统 DMA_2 读	R/W

12.4.12 SMPU2 控制寄存器 MPU_S2CR

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								SMPU2E	Reserved			SMPU2ACT[1:0]		SMPU2BWP	SMPU2BRP

位	标记	位名	功能	读写
b31~b8	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b7	SMPU2E	SMPU2 使能	0：SMPU2 无效 1：SMPU2 有效	R/W
b6~b4	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b3~b2	SMPU2ACT[1:0]	SMPU2 动作选择	设定系统 DMA_2 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
b1	SMPU2BWP	SMPU2 背景写权限设置	0：SMPU2 背景空间允许系统 DMA_2 写 1：SMPU2 背景空间禁止系统 DMA_2 写	R/W
b0	SMPU2BRP	SMPU2 背景读权限设置	0：SMPU2 背景空间允许系统 DMA_2 读 1：SMPU2 背景空间禁止系统 DMA_2 读	R/W

多个区域设定重叠时，优先级为：设定的禁止>设定的允许

12.4.13 FMPU 区域使能寄存器 MPU_FRGE

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								FRG7E	FRG6E	FRG5E	FRG4E	FRG3E	FRG2E	FRG1E	FRG0E

位	标记	位名	功能	读写
b31~b8	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b7	FRG7E	FMPU 区域 7 使能	0：FMPU 的区域 7 无效 1：FMPU 的区域 7 有效	R/W
b6	FRG6E	FMPU 区域 6 使能	0：FMPU 的区域 6 无效 1：FMPU 的区域 6 有效	R/W
b5	FRG5E	FMPU 区域 5 使能	0：FMPU 的区域 5 无效 1：FMPU 的区域 5 有效	R/W
b4	FRG4E	FMPU 区域 4 使能	0：FMPU 的区域 4 无效 1：FMPU 的区域 4 有效	R/W
b3	FRG3E	FMPU 区域 3 使能	0：FMPU 的区域 3 无效 1：FMPU 的区域 3 有效	R/W
b2	FRG2E	FMPU 区域 2 使能	0：FMPU 的区域 2 无效 1：FMPU 的区域 2 有效	R/W
b1	FRG1E	FMPU 区域 1 使能	0：FMPU 的区域 1 无效 1：FMPU 的区域 1 有效	R/W
b0	FRG0E	FMPU 区域 0 使能	0：FMPU 的区域 0 无效 1：FMPU 的区域 0 有效	R/W

12.4.14 FMPU 区域写权限寄存器 MPU_FRGWP

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								FRG7WP	FRG6WP	FRG5WP	FRG4WP	FRG3WP	FRG2WP	FRG1WP	FRG0WP

位	标记	位名	功能	读写
b31~b8	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b7	FRG7WP	FMPU 区域 7 写权限	0：区域 7 允许 USBFS-DMA 写 1：区域 7 禁止 USBFS-DMA 写	R/W
b6	FRG6WP	FMPU 区域 6 写权限	0：区域 6 允许 USBFS-DMA 写 1：区域 6 禁止 USBFS-DMA 写	R/W
b5	FRG5WP	FMPU 区域 5 写权限	0：区域 5 允许 USBFS-DMA 写 1：区域 5 禁止 USBFS-DMA 写	R/W
b4	FRG4WP	FMPU 区域 4 写权限	0：区域 4 允许 USBFS-DMA 写 1：区域 4 禁止 USBFS-DMA 写	R/W
b3	FRG3WP	FMPU 区域 3 写权限	0：区域 3 允许 USBFS-DMA 写 1：区域 3 禁止 USBFS-DMA 写	R/W
b2	FRG2WP	FMPU 区域 2 写权限	0：区域 2 允许 USBFS-DMA 写 1：区域 2 禁止 USBFS-DMA 写	R/W
b1	FRG1WP	FMPU 区域 1 写权限	0：区域 1 允许 USBFS-DMA 写 1：区域 1 禁止 USBFS-DMA 写	R/W
b0	FRG0WP	FMPU 区域 0 写权限	0：区域 0 允许 USBFS-DMA 写 1：区域 0 禁止 USBFS-DMA 写	R/W

12.4.15 FMPU 区域读权限寄存器 MPU_FRGRP

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								FRG7RP	FRG6RP	FRG5RP	FRG4RP	FRG3RP	FRG2RP	FRG1RP	FRG0RP

位	标记	位名	功能	读写
b31~b8	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b7	FRG7RP	FMPU 区域 7 读权限	0：区域 7 允许 USBFS-DMA 读 1：区域 7 禁止 USBFS-DMA 读	R/W
b6	FRG6RP	FMPU 区域 6 读权限	0：区域 6 允许 USBFS-DMA 读 1：区域 6 禁止 USBFS-DMA 读	R/W
b5	FRG5RP	FMPU 区域 5 读权限	0：区域 5 允许 USBFS-DMA 读 1：区域 5 禁止 USBFS-DMA 读	R/W
b4	FRG4RP	FMPU 区域 4 读权限	0：区域 4 允许 USBFS-DMA 读 1：区域 4 禁止 USBFS-DMA 读	R/W
b3	FRG3RP	FMPU 区域 3 读权限	0：区域 3 允许 USBFS-DMA 读 1：区域 3 禁止 USBFS-DMA 读	R/W
b2	FRG2RP	FMPU 区域 2 读权限	0：区域 2 允许 USBFS-DMA 读 1：区域 2 禁止 USBFS-DMA 读	R/W
b1	FRG1RP	FMPU 区域 1 读权限	0：区域 1 允许 USBFS-DMA 读 1：区域 1 禁止 USBFS-DMA 读	R/W
b0	FRG0RP	FMPU 区域 0 读权限	0：区域 0 允许 USBFS-DMA 读 1：区域 0 禁止 USBFS-DMA 读	R/W

12.4.16 FMPU 控制寄存器 MPU_FCR

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								FMPUE	Reserved			FMPUACT[1:0]		FMPUBWP	FMPUBRP

位	标记	位名	功能	读写
b31~b8	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b7	FMPUE	FMPU 使能	0：FMPU 无效 1：FMPU 有效	R/W
b6~b4	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b3~b2	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
b1	FMPUBWP	FMPU 背景写权限设置	0：FMPU 背景空间允许 USBFS-DMA 写 1：FMPU 背景空间禁止 USBFS-DMA 写	R/W
b0	FMPUBRP	FMPU 背景读权限设置	0：FMPU 背景空间允许 USBFS-DMA 读 1：FMPU 背景空间禁止 USBFS-DMA 读	R/W

多个区域设定重叠时，优先级为 设定的禁止>设定的允许

12.4.17 HMPU 区域使能寄存器 MPU_HRGE

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								HRG7E	HRG6E	HRG5E	HRG4E	HRG3E	HRG2E	HRG1E	HRG0E

位	标记	位名	功能	读写
b31~b8	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b7	HRG7E	HMPU 区域 7 使能	0：HMPU 的区域 7 无效 1：HMPU 的区域 7 有效	R/W
b6	HRG6E	HMPU 区域 6 使能	0：HMPU 的区域 6 无效 1：HMPU 的区域 6 有效	R/W
b5	HRG5E	HMPU 区域 5 使能	0：HMPU 的区域 5 无效 1：HMPU 的区域 5 有效	R/W
b4	HRG4E	HMPU 区域 4 使能	0：HMPU 的区域 4 无效 1：HMPU 的区域 4 有效	R/W
b3	HRG3E	HMPU 区域 3 使能	0：HMPU 的区域 3 无效 1：HMPU 的区域 3 有效	R/W
b2	HRG2E	HMPU 区域 2 使能	0：HMPU 的区域 2 无效 1：HMPU 的区域 2 有效	R/W
b1	HRG1E	HMPU 区域 1 使能	0：HMPU 的区域 1 无效 1：HMPU 的区域 1 有效	R/W
b0	HRG0E	HMPU 区域 0 使能	0：HMPU 的区域 0 无效 1：HMPU 的区域 0 有效	R/W

12.4.18 HMPU 区域写权限寄存器 MPU_HRGWP

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								HRG7WP	HRG6WP	HRG5WP	HRG4WP	HRG3WP	HRG2WP	HRG1WP	HRG0WP

位	标记	位名	功能	读写
b31~b8	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b7	HRG7WP	HMPU 区域 7 写权限	0：区域 7 允许 USBHS-DMA 写 1：区域 7 禁止 USBHS-DMA 写	R/W
b6	HRG6WP	HMPU 区域 6 写权限	0：区域 6 允许 USBHS-DMA 写 1：区域 6 禁止 USBHS-DMA 写	R/W
b5	HRG5WP	HMPU 区域 5 写权限	0：区域 5 允许 USBHS-DMA 写 1：区域 5 禁止 USBHS-DMA 写	R/W
b4	HRG4WP	HMPU 区域 4 写权限	0：区域 4 允许 USBHS-DMA 写 1：区域 4 禁止 USBHS-DMA 写	R/W
b3	HRG3WP	HMPU 区域 3 写权限	0：区域 3 允许 USBHS-DMA 写 1：区域 3 禁止 USBHS-DMA 写	R/W
b2	HRG2WP	HMPU 区域 2 写权限	0：区域 2 允许 USBHS-DMA 写 1：区域 2 禁止 USBHS-DMA 写	R/W
b1	HRG1WP	HMPU 区域 1 写权限	0：区域 1 允许 USBHS-DMA 写 1：区域 1 禁止 USBHS-DMA 写	R/W
b0	HRG0WP	HMPU 区域 0 写权限	0：区域 0 允许 USBHS-DMA 写 1：区域 0 禁止 USBHS-DMA 写	R/W

12.4.19 HMPU 区域读权限寄存器 MPU_HRGRP

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								HRG7RP	HRG6RP	HRG5RP	HRG4RP	HRG3RP	HRG2RP	HRG1RP	HRG0RP

位	标记	位名	功能	读写
b31~b8	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b7	HRG7RP	HMPU 区域 7 读权限	0：区域 7 允许 USBHS-DMA 读 1：区域 7 禁止 USBHS-DMA 读	R/W
b6	HRG6RP	HMPU 区域 6 读权限	0：区域 6 允许 USBHS-DMA 读 1：区域 6 禁止 USBHS-DMA 读	R/W
b5	HRG5RP	HMPU 区域 5 读权限	0：区域 5 允许 USBHS-DMA 读 1：区域 5 禁止 USBHS-DMA 读	R/W
b4	HRG4RP	HMPU 区域 4 读权限	0：区域 4 允许 USBHS-DMA 读 1：区域 4 禁止 USBHS-DMA 读	R/W
b3	HRG3RP	HMPU 区域 3 读权限	0：区域 3 允许 USBHS-DMA 读 1：区域 3 禁止 USBHS-DMA 读	R/W
b2	HRG2RP	HMPU 区域 2 读权限	0：区域 2 允许 USBHS-DMA 读 1：区域 2 禁止 USBHS-DMA 读	R/W
b1	HRG1RP	HMPU 区域 1 读权限	0：区域 1 允许 USBHS-DMA 读 1：区域 1 禁止 USBHS-DMA 读	R/W
b0	HRG0RP	HMPU 区域 0 读权限	0：区域 0 允许 USBHS-DMA 读 1：区域 0 禁止 USBHS-DMA 读	R/W

12.4.20 HMPU 控制寄存器 MPU_HCR

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								HMPUE	Reserved			HMPUACT[1:0]		HMPUBWP	HMPUBRP

位	标记	位名	功能	读写
b31~b8	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b7	HMPUE	HMPU 使能	0：HMPU 无效 1：HMPU 有效	R/W
b6~b4	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b3~b2	HMPUACT[1:0]	HMPU 动作选择	设定 USBHS-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
b1	HMPUBWP	HMPU 背景写权限设置	0：HMPU 背景空间允许 USBHS-DMA 写 1：HMPU 背景空间禁止 USBHS-DMA 写	R/W
b0	HMPUBRP	HMPU 背景读权限设置	0：HMPU 背景空间允许 USBHS-DMA 读 1：HMPU 背景空间禁止 USBHS-DMA 读	R/W

多个区域设定重叠时，优先级为 设定的禁止>设定的允许

12.4.21 EMPU 区域使能寄存器 MPU_ERGE

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								ERG7E	ERG6E	ERG5E	ERG4E	ERG3E	ERG2E	ERG1E	ERG0E

位	标记	位名	功能	读写
b31~b8	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b7	ERG7E	EMPU 区域 7 使能	0：EMPU 的区域 7 无效 1：EMPU 的区域 7 有效	R/W
b6	ERG6E	EMPU 区域 6 使能	0：EMPU 的区域 6 无效 1：EMPU 的区域 6 有效	R/W
b5	ERG5E	EMPU 区域 5 使能	0：EMPU 的区域 5 无效 1：EMPU 的区域 5 有效	R/W
b4	ERG4E	EMPU 区域 4 使能	0：EMPU 的区域 4 无效 1：EMPU 的区域 4 有效	R/W
b3	ERG3E	EMPU 区域 3 使能	0：EMPU 的区域 3 无效 1：EMPU 的区域 3 有效	R/W
b2	ERG2E	EMPU 区域 2 使能	0：EMPU 的区域 2 无效 1：EMPU 的区域 2 有效	R/W
b1	ERG1E	EMPU 区域 1 使能	0：EMPU 的区域 1 无效 1：EMPU 的区域 1 有效	R/W
b0	ERG0E	EMPU 区域 0 使能	0：EMPU 的区域 0 无效 1：EMPU 的区域 0 有效	R/W

12.4.22 EMPU 区域写权限寄存器 MPU_ERGWP

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								ERG7WP	ERG6WP	ERG5WP	ERG4WP	ERG3WP	ERG2WP	ERG1WP	ERG0WP

位	标记	位名	功能	读写
b31~b8	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b7	ERG7WP	EMPU 区域 7 写权限	0：区域 7 允许 ETH-DMA 写 1：区域 7 禁止 ETH-DMA 写	R/W
b6	ERG6WP	EMPU 区域 6 写权限	0：区域 6 允许 ETH-DMA 写 1：区域 6 禁止 ETH-DMA 写	R/W
b5	ERG5WP	EMPU 区域 5 写权限	0：区域 5 允许 ETH-DMA 写 1：区域 5 禁止 ETH-DMA 写	R/W
b4	ERG4WP	EMPU 区域 4 写权限	0：区域 4 允许 ETH-DMA 写 1：区域 4 禁止 ETH-DMA 写	R/W
b3	ERG3WP	EMPU 区域 3 写权限	0：区域 3 允许 ETH-DMA 写 1：区域 3 禁止 ETH-DMA 写	R/W
b2	ERG2WP	EMPU 区域 2 写权限	0：区域 2 允许 ETH-DMA 写 1：区域 2 禁止 ETH-DMA 写	R/W
b1	ERG1WP	EMPU 区域 1 写权限	0：区域 1 允许 ETH-DMA 写 1：区域 1 禁止 ETH-DMA 写	R/W
b0	ERG0WP	EMPU 区域 0 写权限	0：区域 0 允许 ETH-DMA 写 1：区域 0 禁止 ETH-DMA 写	R/W

12.4.23 EMPU 区域读权限寄存器 MPU_ERGRP

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								ERG7RP	ERG6RP	ERG5RP	ERG4RP	ERG3RP	ERG2RP	ERG1RP	ERG0RP

位	标记	位名	功能	读写
b31~b8	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b7	ERG7RP	EMPU 区域 7 读权限	0：区域 7 允许 ETH-DMA 读 1：区域 7 禁止 ETH-DMA 读	R/W
b6	ERG6RP	EMPU 区域 6 读权限	0：区域 6 允许 ETH-DMA 读 1：区域 6 禁止 ETH-DMA 读	R/W
b5	ERG5RP	EMPU 区域 5 读权限	0：区域 5 允许 ETH-DMA 读 1：区域 5 禁止 ETH-DMA 读	R/W
b4	ERG4RP	EMPU 区域 4 读权限	0：区域 4 允许 ETH-DMA 读 1：区域 4 禁止 ETH-DMA 读	R/W
b3	ERG3RP	EMPU 区域 3 读权限	0：区域 3 允许 ETH-DMA 读 1：区域 3 禁止 ETH-DMA 读	R/W
b2	ERG2RP	EMPU 区域 2 读权限	0：区域 2 允许 ETH-DMA 读 1：区域 2 禁止 ETH-DMA 读	R/W
b1	ERG1RP	EMPU 区域 1 读权限	0：区域 1 允许 ETH-DMA 读 1：区域 1 禁止 ETH-DMA 读	R/W
b0	ERG0RP	EMPU 区域 0 读权限	0：区域 0 允许 ETH-DMA 读 1：区域 0 禁止 ETH-DMA 读	R/W

12.4.24 EMPU 控制寄存器 MPU_ECR

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								EMPUE	Reserved			EMPUACT[1:0]		EMPUBWP	EMPUBRP

位	标记	位名	功能	读写
b31~b8	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b7	EMPUE	EMPU 使能	0：EMPU 无效 1：EMPU 有效	R/W
b6~b4	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b3~b2	EMPUACT[1:0]	EMPU 动作选择	设定 ETH-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
b1	EMPUBWP	EMPU 背景写权限设置	0：EMPU 背景空间允许 ETH-DMA 写 1：EMPU 背景空间禁止 ETH-DMA 写	R/W
b0	EMPUBRP	EMPU 背景读权限设置	0：EMPU 背景空间允许 ETH-DMA 读 1：EMPU 背景空间禁止 ETH-DMA 读	R/W

多个区域设定重叠时，优先级为 设定的禁止>设定的允许

12.4.25 IP 访问保护寄存器 MPU_IPPR

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
BUSERRE	-	MSTPWRP	MSTPRDP	SYSCWRP	SYSCRDP	INTCWRP	INTCRDP	SRAMCWRP	SRAMCRDP	DMPUWRP	DMPURDP	RTCWRP	RTCRDP	BKSRAMWRP	BKSRAMRDP

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SWDTWRP	SWDTRDP	WDTWRP	WDTRDP	-	-	FMCWRP	FMCRDP	CRCWRP	CRCRDP	TRNGWRP	TRNGRDP	HASHWRP	HASHRDP	AESWRP	AESRDP

位	标记	位名	功能	读写
b31	BUSERRE	总线错误允许	0：忽略对保护 IP 的访问 1：发生对保护 IP 的访问时，返回总线错误	R/W
b30	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b29	MSTPWRP	MSTP 写保护	0：允许对 PWR_FCG0/1/2/3、PWR_FCG0PC 的写操作 1：禁止对 PWR_FCG0/1/2/3、PWR_FCG0PC 的写操作	R/W
b28	MSTPRDP	MSTP 读保护	0：允许对 PWR_FCG0/1/2/3、PWR_FCG0PC 的读操作 1：禁止对 PWR_FCG0/1/2/3、PWR_FCG0PC 的读操作	R/W
b27	SYSCWRP	SYSC 写保护	0：允许对 RMU/CMU/PWC 的写操作 1：禁止对 RMU/CMU/PWC 的写操作 注意：BUSERRE 设置为 1 时，对 RMU/PWC 的写操作不会发生总线错误	R/W
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 注意：BUSERRE 设置为 1 时，对 RMU/PWC 的读操作不会发生总线错误	R/W
b25	INTCWRP	INTC 写保护	0：允许对 INTC 的写操作 1：禁止对 INTC 的写操作	R/W
b24	INTCRDP	INTC 读保护	0：允许对 INTC 的读操作 1：禁止对 INTC 的读操作	R/W

b23	SRAMCWRP	SRAMC 写保护	0：允许对 SRAMC 的写操作 1：禁止对 SRAMC 的写操作	R/W
b22	SRAMCRDP	SRAMC 读保护	0：允许对 SRAMC 的读操作 1：禁止对 SRAMC 的读操作	R/W
b21	DMPUWRP	DMPU 写保护	0：允许对 DMPU 的写操作 1：禁止对 DMPU 的写操作	R/W
b20	DMPURDP	DMPU 读保护	0：允许对 DMPU 的读操作 1：禁止对 DMPU 的读操作	R/W
b19	RTCWRP	RTC 写保护	0：允许对 RTC 的写操作 1：禁止对 RTC 的写操作	R/W
b18	RTCRDP	RTC 读保护	0：允许对 RTC 的读操作 1：禁止对 RTC 的读操作	R/W
b17	BKSRAMWRP	BKSRAM 写保护	0：允许对 BKSRAM 的写操作 1：禁止对 BKSRAM 的写操作	R/W
b16	BKSRAMRDP	BKSRAM 读保护	0：允许对 BKSRAM 的读操作 1：禁止对 BKSRAM 的读操作	R/W
b15	SWDTWRP	SWDT 写保护	0：允许对 SWDT 的写操作 1：禁止对 SWDT 的写操作	R/W
b14	SWDTRDP	SWDT 读保护	0：允许对 SWDT 的读操作 1：禁止对 SWDT 的读操作	R/W
b13	WDTWRP	WDT 写保护	0：允许对 WDT 的写操作 1：禁止对 WDT 的写操作	R/W
b12	WDTRDP	WDT 读保护	0：允许对 WDT 的读操作 1：禁止对 WDT 的读操作	R/W
b11-b10	Reserved	-	保留位，读出为 0，写入时写 0	R/W
b9	FMCWRP	FMC 写保护	0：允许对 FMC 的写操作 1：禁止对 FMC 的写操作	R/W
b8	FMCRDP	FMC 读保护	0：允许对 FMC 的读操作	R/W

			1：禁止对 FMC 的读操作	
b7	CRCWRP	CRC 写保护	0：允许对 CRC 的写操作	R/W
			1：禁止对 CRC 的写操作	
b6	CRCRDP	CRC 读保护	0：允许对 CRC 的读操作	R/W
			1：禁止对 CRC 的读操作	
b5	TRNGWRP	TRNG 写保护	0：允许对 TRNG 的写操作	R/W
			1：禁止对 TRNG 的写操作	
b4	TRNGRDP	TRNG 读保护	0：允许对 TRNG 的读操作	R/W
			1：禁止对 TRNG 的读操作	
b3	HASHWRP	HASH 写保护	0：允许对 HASH 的写操作	R/W
			1：禁止对 HASH 的写操作	
b2	HASHRDP	HASH 读保护	0：允许对 HASH 的读操作	R/W
			1：禁止对 HASH 的读操作	
b1	AESWRP	AES 写保护	0：允许对 AES 的写操作	R/W
			1：禁止对 AES 的写操作	
b0	AESRDP	AES 读保护	0：允许对 AES 的读操作	R/W
			1：禁止对 AES 的读操作	

特权模式下不受本寄存器影响可以读写访问对象 IP。

13 键盘扫描控制模块（KEYSCAN）

13.1 简介

本产品搭载键盘扫描控制模块（KEYSCAN）1 个单元。KEYSCAN 模块支持键盘阵列（行和列）扫描，列是由独立的扫描输出 KEYOUTm(m=0~7)驱动，而行 KEYINn(n=0~15)则作为 EIRQn(n=0~15)输入被检测。本模块通过行扫描查询法实现按键识别功能。

KEYSCAN 主要特性：

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

13.2 KEYSCAN 系统框图

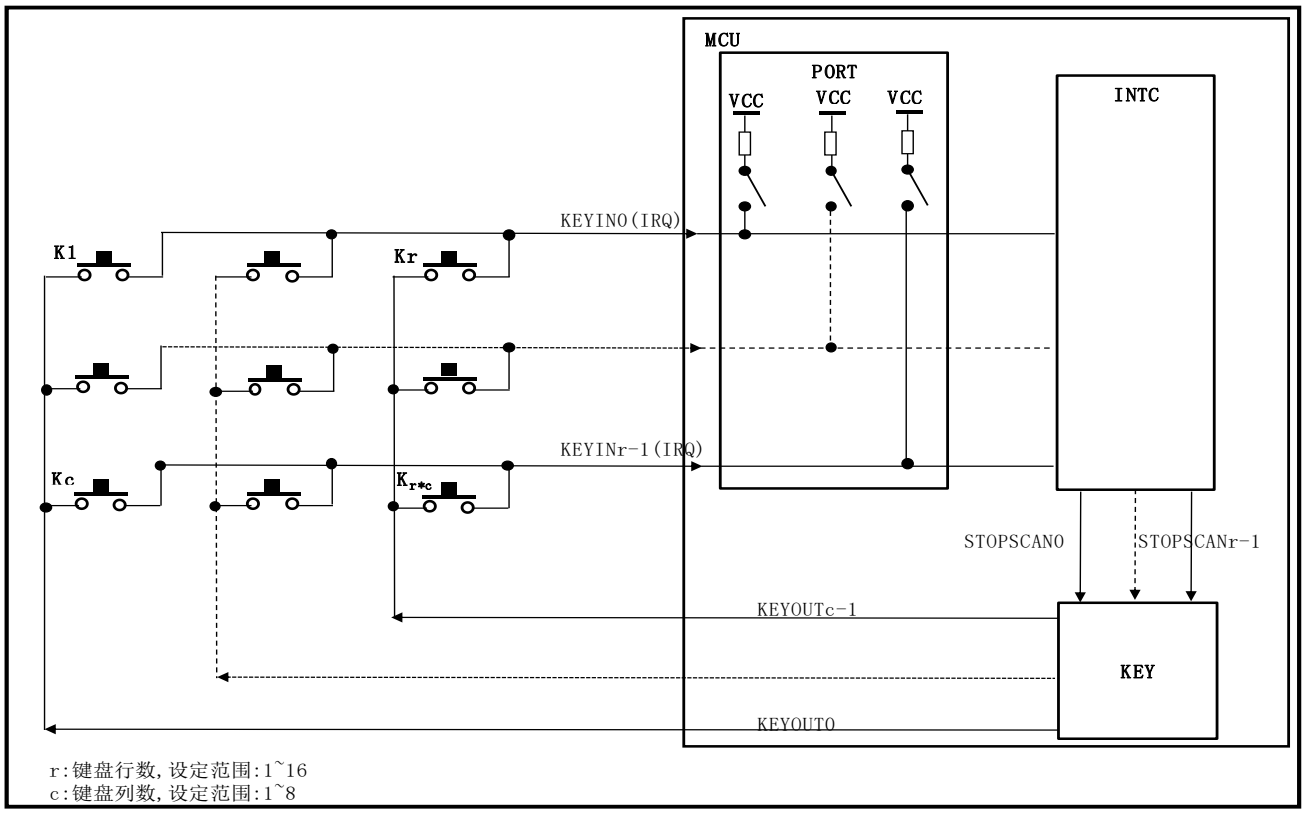


图 13-1 KEYSCAN 系统框图

13.3 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 13-1 KEYSCAN 管脚说明

13.4 功能说明

本章将对键盘扫描功能和按键识别功能详细说明。

13.4.1 按键识别功能

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

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

13.4.2 键盘扫描功能

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

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

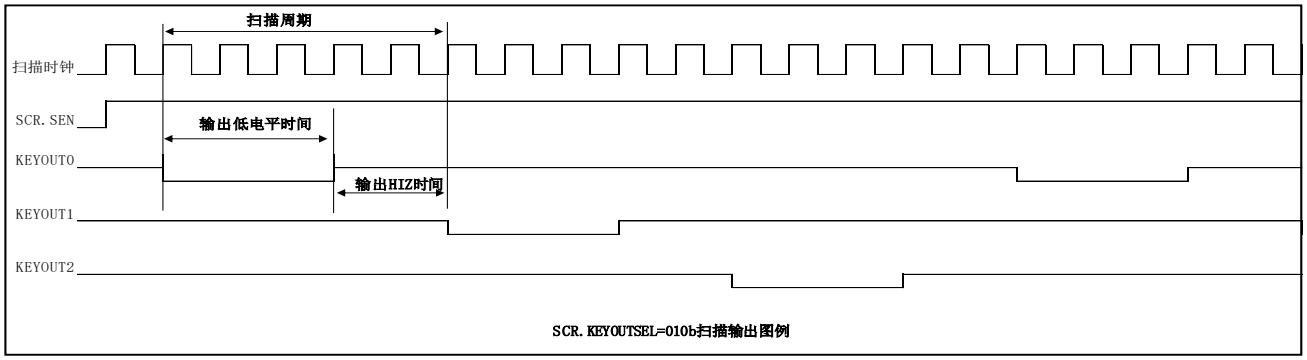


图 13-2 键盘扫描功能示意图

13.4.3 使用上的注意事项

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

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

如果使用芯片内部上拉电阻，则请参考 PORT 特性，选择合适的扫描时间以及滤波时间。

13.5 寄存器说明

KEYSCAN_BASE_ADDR:0x40050C00

寄存器名	符号	偏移地址	位宽	复位值
KEYSCAN扫描控制寄存器	KEYSCAN_SCR	0x00	32	0x0000 0000
KEYSCAN扫描使能寄存器	KEYSCAN_SER	0x04	32	0x0000 0000
KEYSCAN扫描状体寄存器	KEYSCAN_SSR	0x08	32	0x0000 0000

表 13-2 KEYSCAN 寄存器一览表

13.5.1 KEYSCAN 扫描控制寄存器（KEYSCAN_SCR）

KEYSCAN Scan Control Register

偏移地址：0x00

复位值：0x00000000

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

位	标记	位名	功能	读写
b31~b29	T-HIZ[2:0]	输出HIZ时间	KEYOUT输出低电平间HIZ时间（扫描时钟数） 扫描周期=输出低电平时间+输出HIZ时间 设定值：HIZ周期数 000b： 4 001b： 8 010b： 16 011b： 32 100b： 64 101b： 256 110b： 512 111b： 1024 注意：SCR.T-HIZ[2：0]只能在SER.SEN=0时设定有效	R/W
b28~b24	T_LLEVEL[4:0]	输出低电平时间	KEYOUT输出低电平间时间（扫描时钟数） 扫描周期=输出低电平时间+输出HIZ时间 输出低电平时间=2的T_LLEVEL次幂个扫描时钟数 注意：SCR.T-LLEVEL[4:0]只能在SER.SEN=0时设定有效,且00000b和00001b设定禁止，最大可以设定值为11000b	R/W
b23~b22	Reserved	-	读出时为“0”,写入时写“0”	R/W
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/W

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 注意：SCR.KEYOUTSEL[2:0]只能在SER.SEN=0时设定有效	R/W
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 注意：SCR. KEYINSEL[15:0]只能在SER.SEN=0时设定有效	R/W

13.5.2 KEYSCAN 扫描使能寄存器（KEYCAN_SER）

KEYSCAN Scan Enable Register

偏移地址：0x04

复位值：0x00000000

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

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

位	标记	位名	功能	读写
b31~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	SEN	扫描使能位	扫描使能位	R/W
			0：扫描禁止	
			1：扫描使能	

13.5.3 KEYSCAN 扫描状态寄存器（KEYSCAN_SSR）

KEYSCAN Scan Status Register

偏移地址：0x08

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	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/W
b2~b0	INDEX[2:0]	当前工作的SCAN管脚索引位	当前工作的SCAN管脚索引位 000：当前工作的SCAN管脚为KEYOUT0 001：当前工作的SCAN管脚为KEYOUT1 010：当前工作的SCAN管脚为KEYOUT2 011：当前工作的SCAN管脚为KEYOUT3 100：当前工作的SCAN管脚为KEYOUT4 101：当前工作的SCAN管脚为KEYOUT5 110：当前工作的SCAN管脚为KEYOUT6 111：当前工作的SCAN管脚为KEYOUT7 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	R

14 内部时钟校准器（CTC）

14.1 简介

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

CTC 的主要特性如下：

- 三个外部参考时钟源：XTAL，XTAL32 以及 CTCREF
- 用于频率测量并具有重载功能的 16 位校准计数器
- 用于频率校准的 8 位校准偏差值和 6 位校准值
- 用于提示校准失败的错误中断

14.2 结构框图

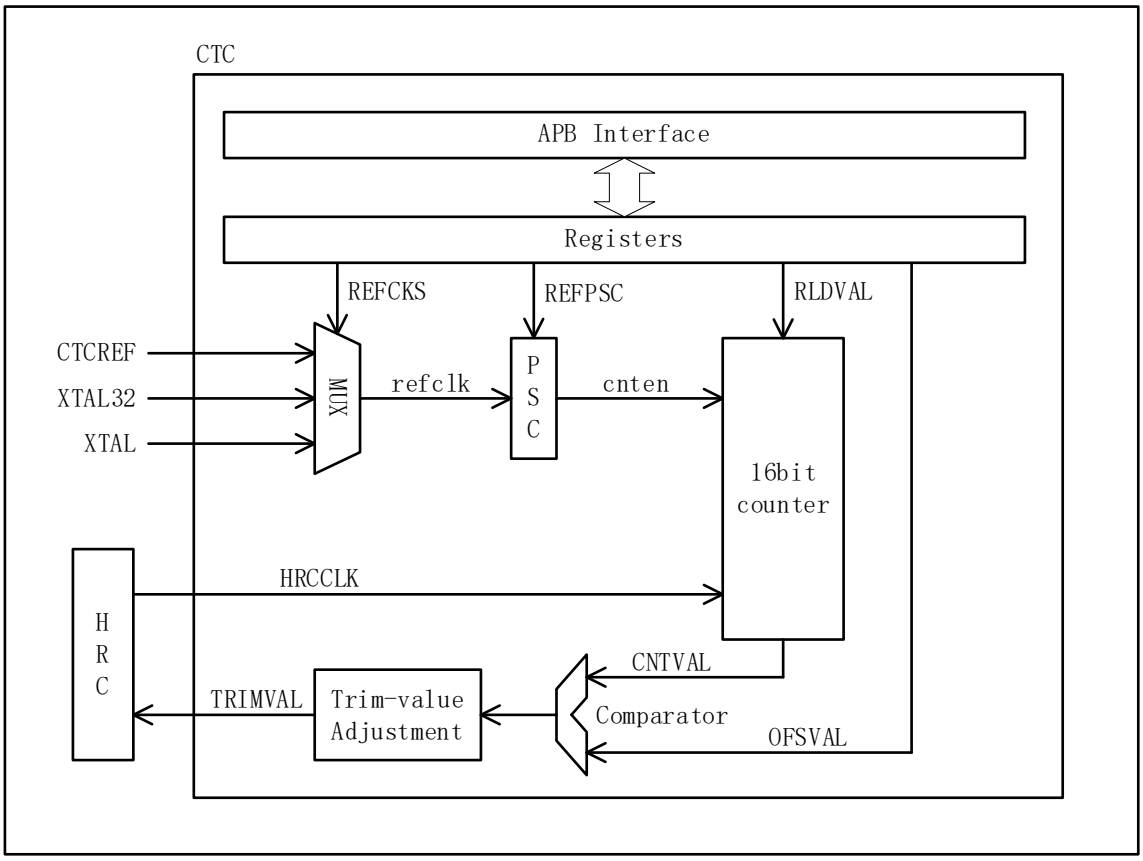


图 14-1 CTC 的基本框图

14.3 功能说明

14.3.1 参考时钟

CTC 有三个外部时钟源可作为校准 HRC 频率的参考时钟，分别是外部高速时钟（XTAL，~24MHz）、外部低速时钟（XTAL32，32.768kHz）和外部基准时钟（CTCREF）。可以通过设置 CTC_CR1 寄存器中的 REFCKS[1:0]位来选择一个作为参考时钟。

HRC 的 Jitter 以及频率测量过程中的计数误差都会影响校准精度，可以通过设置 CTC_CR1 寄存器中的 REFPSC[2:0]位为参考时钟选择合适的分频来减小误差带来的影响。表 14-1 和表 14-2 是不同频率的参考时钟在不同分频条件下的误差估算值，请参考选择合适的参考时钟源和分频。如果参考时钟的频率未在表中列出，可以参考以下公式计算误差。基本上，应该使测量误差小于校准精度。

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

注：Fhrc 表示 HRC 的目标频率；

Fref 表示参考时钟的频率；

PSC 表示参考时钟分频数；

(Fhrc ÷ Fref)×PSC 表示测量 HRC 频率时校准计数器的计数值；

Fref / PSC	32.768kHz	4MHz	8MHz	12MHz	24MHz
8	0.530%	不推荐	不推荐	不推荐	不推荐
32	0.244%	不推荐	不推荐	不推荐	不推荐
128	不可设	1.863%	3.000%	不推荐	不推荐
256	不可设	1.188%	1.863%	2.453%	不推荐
512	不可设	0.775%	1.188%	1.540%	2.453%
1024	不可设	0.516%	0.775%	0.992%	1.540%
2048	不可设	0.348%	0.516%	0.653%	0.992%
4096	不可设	0.238%	0.348%	0.437%	0.653%

表 14-1 HRC 目标频率为 20MHz 时的测量误差

注：“不可设”表示该设定超出了校准计数器的范围，无法完成校准。

注：“不推荐”表示该设定误差过大，无法精确校准。

Fref / PSC	32.768kHz	4MHz	8MHz	12MHz	24MHz
8	0.603%	不推荐	不推荐	不推荐	不推荐
32	0.276%	不推荐	不推荐	不推荐	不推荐
128	不可设	2.165%	3.519%	不推荐	不推荐
256	不可设	1.369%	2.165%	2.866%	不推荐
512	不可设	0.887%	1.369%	1.784%	2.866%
1024	不可设	0.587%	0.887%	1.140%	1.784%
2048	不可设	0.395%	0.587%	0.746%	1.140%
4096	不可设	0.269%	0.395%	0.497%	0.746%

表 14-2 HRC 目标频率为 16MHz 时的测量误差

注：“不可设”表示该设定超出了校准计数器的范围，无法完成校准。

注：“不推荐”表示该设定误差过大，无法精确校准。

14.3.2 频率校准

在 CTC_CR1 寄存器的 CTCEN 位置为 1 后，16 位校准计数器从 CTC_CR2 寄存器中重载 RLDVAL 并向下计数，计数时钟由 HRC 提供。之后，如果检测到 PSC 发出的计数停止同步信号，16 位校准计数器将停止计数。如果始终检测不到 PSC 的计数停止同步信号，16 位校准计数器会持续计数直到下溢为 0xFFFF 时停止，并等待 PSC 的计数停止同步信号的到来。

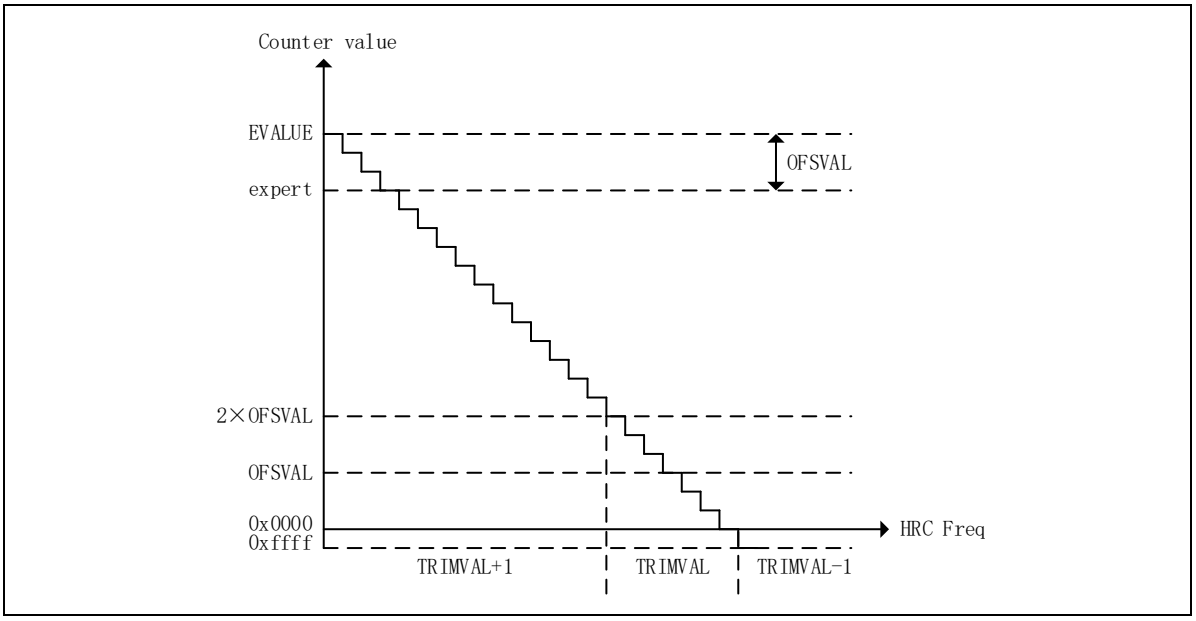


图 14-2 CTC 校准示意图

检测到 PSC 发出的计数停止同步信号后，HRC 频率校准功能开始执行。由于在设定校准计数重载值 RLDVAL 时加上了校准计数偏差值 OFSVAL，因此如果 16 位校准计数器的值大于 2×OFSVAL 且没有发生下溢，说明当前频率比期望频率慢，需要增大 CTC_CR1 寄存器中 TRMVAL[5:0]的值来提高频率。反之，如果 16 位校准计数器的值发生下溢，则说明当前频率比期望频率快，需要减小 TRMVAL 的值以降低频率。

- Counter > (OFSVAL × 2)时停止计数；

  CTC_CR1 寄存器中的 TRMVAL 的值自动加 1。

- 0 ≤ Counter ≤(OFSVAL × 2)时停止计数；

  CTC_CR1 寄存器中的 TRMVAL 的值不变，CTC_STR 寄存器中的 TRIMOK 位被置成 1。

- Counter = 0xFFFF；

CTC_CR1 寄存器中的 TRMVAL 的值自动减 1。

TRMVAL 的中间值是 0x00。如果自动加 1 的结果大于 0x1F，TRMVAL 的值将保持 0x1F，同时发生校准上溢，CTC_STR 寄存器中的 TRMOVF 位置 1。反之，如果自动减 1 的结果小于 0x20，TRMVAL 的值将保持 0x20，同时发生校准下溢，CTC_STR 寄存器中的 TRMUDF 位置 1。

发生校准上溢或下溢时校准会自动停止（CTCEN 自动清零），此时如果 CTC_CR1 寄存器中的 ERRIE 位为 1，就会发生一次校准错误中断。

14.3.3 编程指南

以使用 XTAL32 的 8 分频作为参考时钟将 HRC 校准到（20 ± 0.5%）MHz 为例。

查表可知频率测量误差为 0.244%，小于校准允许偏差 0.5%。

（1） 设定 CTC_CR2 = 0x132b0018，其中

OFSVAL=（20 ÷ 0.032768）× 8 × 0.5% ≈ 24（0x18）

RLDVAL=（20 ÷ 0.032768）×8 + 24 ≈ 4907（0x132b）

（2） 设定 CTC_CR1 = 0x000000e0

TRMVAL=0x00（从中间值开始校准）

CTCEN=1（启动 CTC）

ERRIE=1（允许错误中断）

REFCKS=10b（选择 XTAL32 作为参考时钟）

REFPSC=000b（选择参考时钟的 8 分频）

（3） 一段时间后，确认 CTC_STR.TRIMOK 的状态

如果 TRIMOK=1，CTCEN 清零

如果 TRIMOK=0，回到（3）

（4） 确认 CTC_STR.CTCBSY=0，后续操作。

途中如果发生错误中断，则表示由于超出可校准范围发生了校准上溢或者校准下溢，CTCEN 位会自动清零，校准停止。这时请修正 OFSVAL，RLDVAL 以及参考时钟设定后再次尝试校准。

需要注意：频率校准是通过 CTC_CR1 寄存器中的 CTCEN 位来启动或停止的。因此，

启动频率校准后，请勿改变任何其他寄存器的设定值。如需修改设置，请先将 CTCEN 位写 0，并确认 CTC_STR 寄存器中的 CTCBSY 位为 0 后再进行修改。

14.4 寄存器说明

基准地址： 0x40049C00

寄存器名	符号	偏移地址	位宽	复位值
时钟校准控制寄存器1	CTC_CR1	0x00	32	0x00000000
时钟校准控制寄存器2	CTC_CR2	0x04	32	0x00000000
时钟校准状态寄存器	CTC_STR	0x08	32	0x00000000

表 14-3 CTC 寄存器一览表

14.4.1 时钟校准控制寄存器 1（CTC_CR1）

复位值：0x00000000

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

位	标记	位名	功能	读写
b31~b22	Reserved	-	读出时为“0”，写入时写“0”	R/W
b21~b16	TRMVAL[5:0]	HRC校准值	TRMVAL最高位为符号位，中间值是0。 0x20：-32 0x21：-31 …… 0x3F：-1 0x00：0（中间值） 0x01：+1 …… 0x1E：+30 0x1F：+31 CTCEN位置1时TRMVAL为只读，由硬件自动修改。如需修改请将CTCEN位置0并确认CTC_STR.CTCBSY位为0后再修改。	R/W
b15~b8	Reserved	-	读出时为“0”，写入时写“0”	R/W
b7	CTCEN	校准使能	用于启动频率校准。 0：停止频率校准 1：启动频率校准 置1条件： 软件置1； 清0条件： (1)软件清0 (2)校准发生上溢或者下溢时硬件清0；	R/W
b6	ERRIE	错误中断允许	用于选择发生校准上溢或者下溢时是否产生错误中断。 0：禁止发生错误中断 1：允许发生错误中断	R/W
b5~b4	REFCKS[1:0]	参考时钟选择	用于为频率校准选择参考时钟源。 0xb：选择CTCREF作为参考时钟 10b：选择XTAL32作为参考时钟 11b：选择XTAL作为参考时钟 CTCEN位置1时禁止修改REFCKS的值。如需修改请将CTCEN位置0并确认CTC_STR.CTCBSY位为0后再修改。	R/W
b3	Reserved	-	读出时为“0”，写入时写“0”	R/W

b2~b0	REFPSC[2:0]	参考时钟分频	用于选择参考时钟的分频。 000b：参考时钟8分频 001b：参考时钟32分频 010b：参考时钟128分频 011b：参考时钟256分频 100b：参考时钟512分频 101b：参考时钟1024分频 110b：参考时钟2048分频 111b：参考时钟4096分频 CTCEN位置1时禁止修改REFPSC的值。如需修改请将CTCEN位置0并确认CTC_STR.CTCBSY位为0后再修改。	R/W

14.4.2 时钟校准控制寄存器 2（CTC_CR2）

复位值：0x00000000

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

位	标记	位名	功能	读写
b31~b16	RLDVAL[15:0]	校准计数重载值	RLDVAL设定的是16位校准计数器的重载值，每次校准开始后16位校准计数器以该值为起点向下计数。 RLDVAL = (Fhrc ÷ Fref) + OFSVAL Fhrc：HRC标准频率值； Fref：参考时钟频率÷参考时钟预分频数； OFSVAL：校准计数偏差值； CTCEN位置1时禁止修改RLDVAL的值。如需修改请将CTCEN位置0并确认CTC_STR.CTCBSY位为0后再修改。	R/W
b15~b8	Reserved	-	读出时为“0”，写入时写“0”	R/W
b7~b0	OFSVAL[7:0]	校准计数偏差值	OFSVAL设定的是HRC在校准时允许的偏差值。 OFSVAL = (Fhrc ÷ Fref) × TA Fhrc：HRC标准频率值； Fref：参考时钟频率÷参考时钟预分频数； TA：校准精度，例如0.5%； CTCEN位置1时禁止修改OFSVAL的值。如需修改请将CTCEN位置0并确认CTC_STR.CTCBSY位为0后再修改。	R/W

14.4.3 时钟校准状态寄存器（CTC_STR）

复位值：0x00000000

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

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	CTCBSY	CTC状态标志	0：CTC处于停止状态 1：CTC处于工作状态	R
b2	TRMUDF	校准下溢标志	校准下溢表示TRMVAL已减至0x20，但仍未校准到目标范围。校准下溢时自动停止校准，ERRIE为1时将发生错误中断。 0：未发生校准下溢 1：发生了校准下溢	R
b1	TRMOVF	校准上溢标志	校准上溢表示TRMVAL已增至0x1F，但仍未校准到目标范围。校准上溢时自动停止校准，ERRIE为1时将发生错误中断。 0：未发生校准上溢 1：发生了校准上溢	R
b0	TRIMOK	校准成功标志	0：HRC频率未校准到目标范围 1：HRC频率已校准到目标范围	R

15 DMA 控制器（DMA）

15.1 简介

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

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

15.2 模块示意图

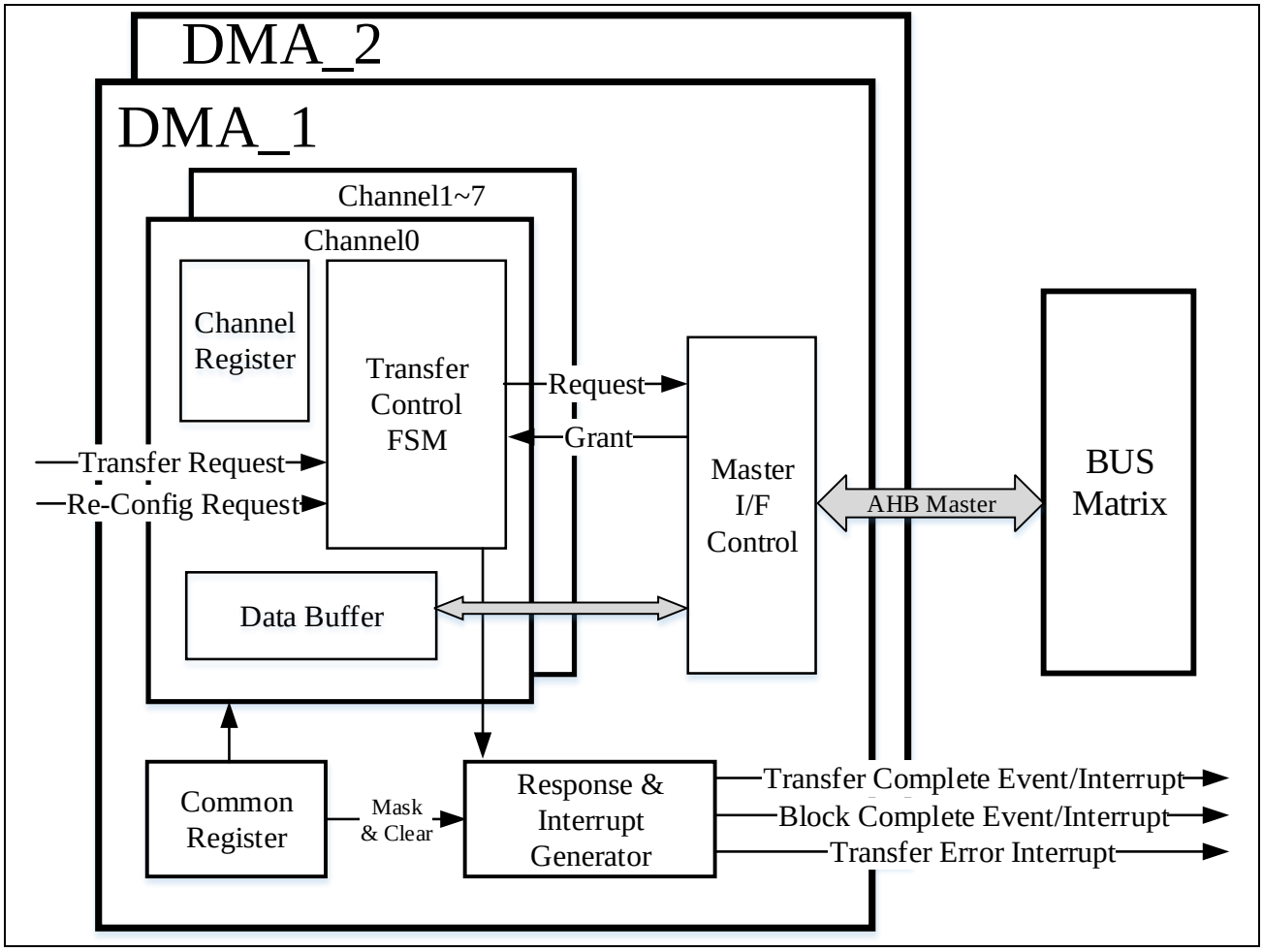


图 15-1 DMA 结构图

15.3 功能描述

15.3.1 使能 DMA 控制器

使用 DMA 时需要先将功能时钟控制 0 寄存器 FCG0.DMAx 位写 0 允许 DMA 的时钟供给，再将 DMA 使能寄存器 DMA_EN.EN 位写 1。同时，DMA 需要外围电路来启动传输，所以 FCG0.AOS 外设电路触发功能位也需写 0。

当不使用 DMA，或者芯片需要进入 STOP 模式时，请将 DMA_EN.EN 设定为 0，并写 FCG0 关闭 DMA 时钟。EN 写 0 前请先确认寄存器 DMA_CHSTAT.DMAACT 为 0，确保 DMA 已完成所有传输。

15.3.2 通道选择和通道优先级

每个 DMA 控制单元中包含 8 个通道，每个通道可以独立配置传输功能。

8 个通道优先级顺序为：通道 0>通道 1>通道 2>...>通道 7。

当一个 DMA 单元有多个通道请求传输时将按照优先级顺序执行。但已处于传输中的通道不会被打断，高优先通道需等当前通道传输完成后才会启动。

15.3.3 启动 DMA

DMA 通过外围电路产生的请求启动，这些请求通过触发源选择寄存器 DMA_TRGSELx(x=0~7)进行配置，各通道独立配置启动请求源。当外围电路产生启动请求或软件写寄存器产生启动请求时，并且 DMA 传输使能有效 DMA_EN.EN=1，传输通道处于许可状态 DMA_CHEN.CHEN[x]=1，则启动通道 x 传输。

使用前，需要先将功能时钟控制 0 寄存器(FCG0)的外设电路触发功能及 DMA 功能使能位置为有效。

15.3.4 数据块

DMA 每次启动传输的数据量用块(block)表示，块的大小由数据控制寄存器 DMA_DTCTLx.BLKSIZE 设定，最大可以设置 1024 个数据。每个的数据宽度由 DMA_CHCTLx.HIZE 决定，可以是 8bit，16bit 或 32bit。

15.3.5 传输地址控制

传输的源地址和目标地址可以由寄存器设定为固定、递增、递减、重载或者不连续跳转。

固定：源地址、目标地址将在传输过程中固定不变。

递增及递减：源地址、目标地址将在每传输完成 1 个数据后根据 HSIZE 的值进行增减。例如当 HSIZE 为 8bit 时，地址将每次增加/减少 1，为 16bit 时每次增加/减少 2，为 32bit 时每次增加/减少 4。

重载：传输指定数量的数据后，源、目标地址将重新返回至最初的地址设定值。地址重载前需要传输的数据量，即重复区域的大小由寄存器 DMA_RPT 设定。

不连续地址传输：传输指定数量的数据后，源、目标地址将跳过指定偏移量。地址跳转的偏移量，以及跳转前需要传输的数据量，即不连续区域的大小，由寄存器 DMA_SNSEQCTL/DMA_DNSEQCTL 设定。当地址重载与不连续跳转的条件同时满足时，执行地址重载。

15.3.6 传输次数

DMA 传输的总数据块的个数由数据控制寄存器 DMA_DTCTLx 的 CNT 位设定。传输次数最多可设置 65535 次。每传输一个数据块寄存器值减 1，当寄存器值减为 0 时代表本通道全部数据传输完成，自动清除通道传输许可位 DMA_CHEN.CHEN[x]，并产生传输完成中断。如果传输开始时 DMA_DTCTLx.CNT 被设为 0，表示无限次传输，每次启动请求传输一个数据块，但不清除通道传输许可位，也不会产生传输完成中断。

注意：

- CHEN[x]自动清 0 后，需要再次启动此通道时，需要重新设置通道配置寄存器后再将 CHEN[x]写 1 使能本通道。否则将按照上次传输结束时的状态开始后续的传输，即源/目标地址为上次传输对应的下个地址，CNT 为 0 的无限次传输。

15.3.7 中断和事件信号输出

DMA 控制器可以产生以下 3 种中断：

数据块完成中断 DMA_BTCx：完成一个数据块传输后产生。

传输完成中断 DMA_TCx：完成寄存器 DMA_DTCTLx.CNT 设置的传输次数后产生。

传输错误中断 DMA_ERR：当启动请求溢出（即通道的上一次请求还未响应时此通道再次触发启动请求）时，或者，传输过程中发生总线错误（如访问了非法地址或受保护的地址）时产生中断，其中，总线错误时会立即终止本次传输。

以上中断除了启动请求溢出错误外，其他中断都可以通过寄存器 DMA_CHCTLx.IE 设置中断的有效或无效。另外所有中断还配有独立的 MASK 寄存器，对中断进行屏蔽。

上述的 DMA_BTCx，DMA_TCx 中断同时也可以作为事件信号输出，可用作其它外围电路的触发源，事件输出受 MASK 寄存器控制，但不受中断许可位 DMA_CHCTLx.IE 控制。DMA_BTCx，DMA_TCx，DMA_ERR 事件发生会将对应的状态位寄存器置位，不受 DMA_CHCTLx.IE、或 MASK 寄存器影响。

15.3.8 连锁传输

DMA 控制器具有连锁传输功能。连锁传输需要配置以下 8 个寄存器共 8 个 word，称为一个描述符(descriptor)，包含了连锁传输的源地址、目标地址、数据控制信息、地址控制信息、链指针以及传输控制信息。

DMA_SARx

DMA_DARx

DMA_DTCTLx

DMA_RPTx

DMA_SNSEQCTLx

DMA_DNSEQCTLx

DMA_LLPx

DMA_CHCTLx

其中 LLP 称为链指针(Linked-List Pointer)，其中的值代表下一个描述符在存储器中的首地址。使用连锁传输时首先写通道控制寄存器 DMA_CHCTLx 的 LLPEN 将连锁传输使能，并将第一个传输的描述符信息写入相应寄存器中。再将后续传输的描述符按顺序初始化在存储器中。需要结束连锁传输时，将最后一个描述符中 DMA_CHCTLx 的 LLPEN 置为无效，DMA 控制器将在传输完成后结束连锁传输。

当一个描述符的最后一次传输结束时，LLP 指定的下一个描述符将被从存储器中载入通道配置寄存器。等待下一次传输请求输入，开始新描述符的第一次传输。或者根据

寄存器 DMA_CHCTLx.LLPRUN 的设置，在载入新描述符后直接开始第一次传输。

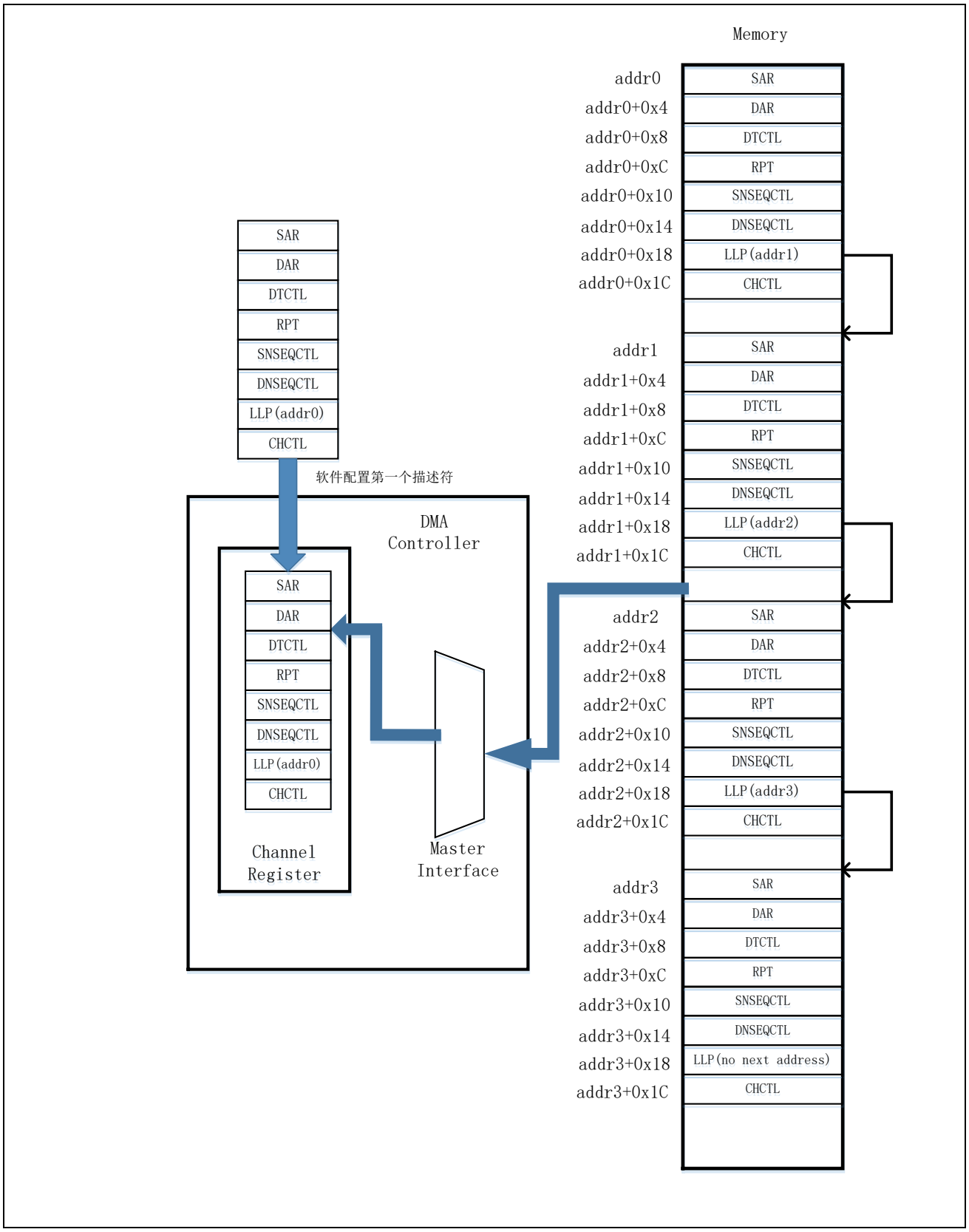


图 15-2 连锁传输示意图

15.3.9 不连续地址传输

使用不连续地址传输能够实现源地址和目标地址在传输完一定量的数据后按照一定的偏移量跳转。以源地址不连续传输为例：首先根据需要将通道控制寄存器DMA_CHCTLx.SNSEQEN 置 1 使不连续地址传输有效，跳转的方向按照DMA_CHCTLx.SINC 的设定向前或向后跳转，然后配置制寄存器 DMA_SNSEQCTLx设置具体的跳转方式。目标地址的控制与源地址相似，对应控制寄存器分别为DMA_CHCTLx.DNSEQEN、DMA_CHCTLx.DINC、DMA_DNSEQCTLx。传输过程按照下图方式进行。

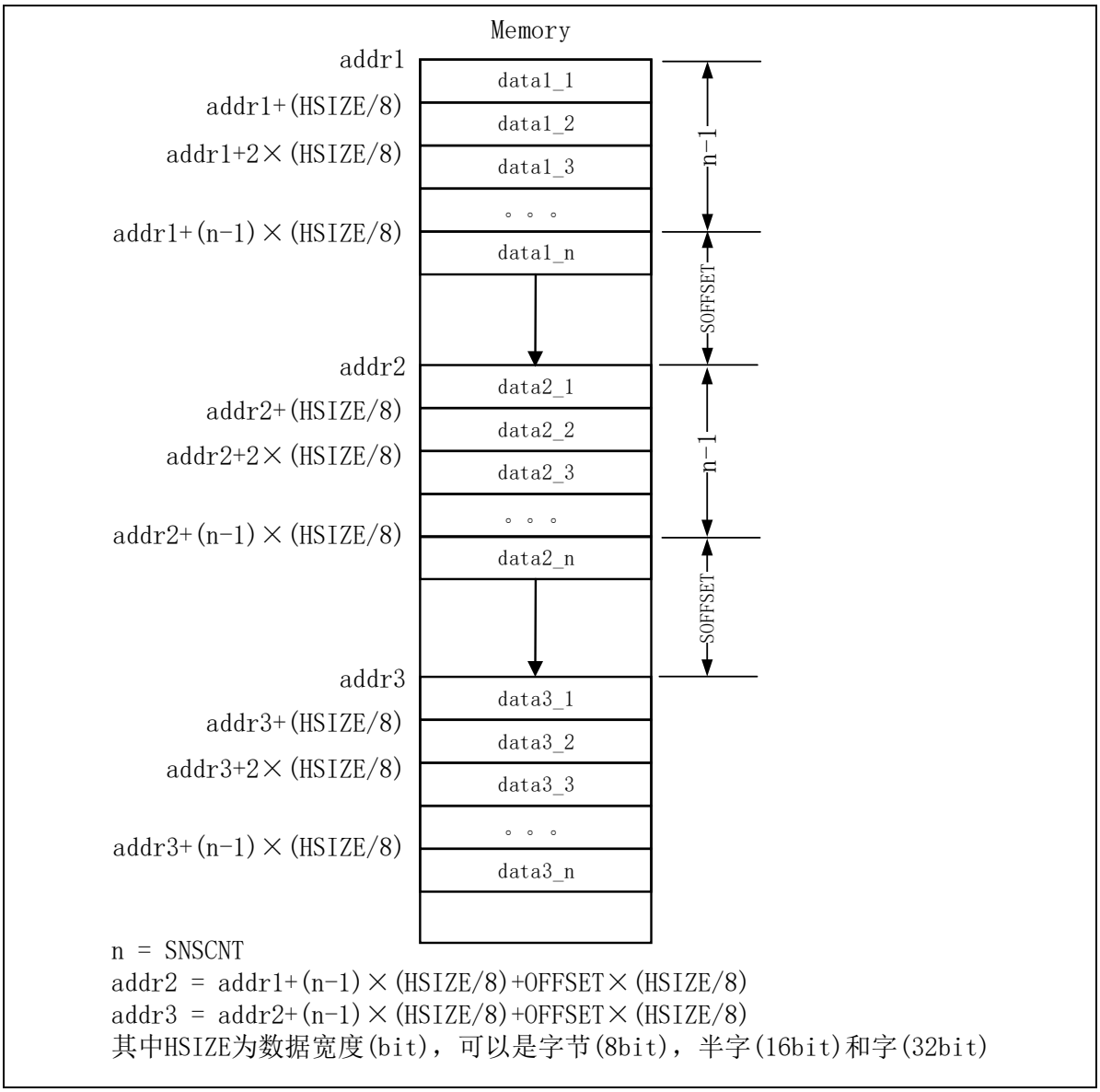


图 15-3 不连续地址传输示意图（以源地址为例）

15.3.10通道重置

通道重置功能是指通过外围电路的事件请求来修改通道内部状态寄存器，重新配置下一次数据的传输方式。设置寄存器 DMA_RCFGCTL.RCFGEN 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 15-1 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。

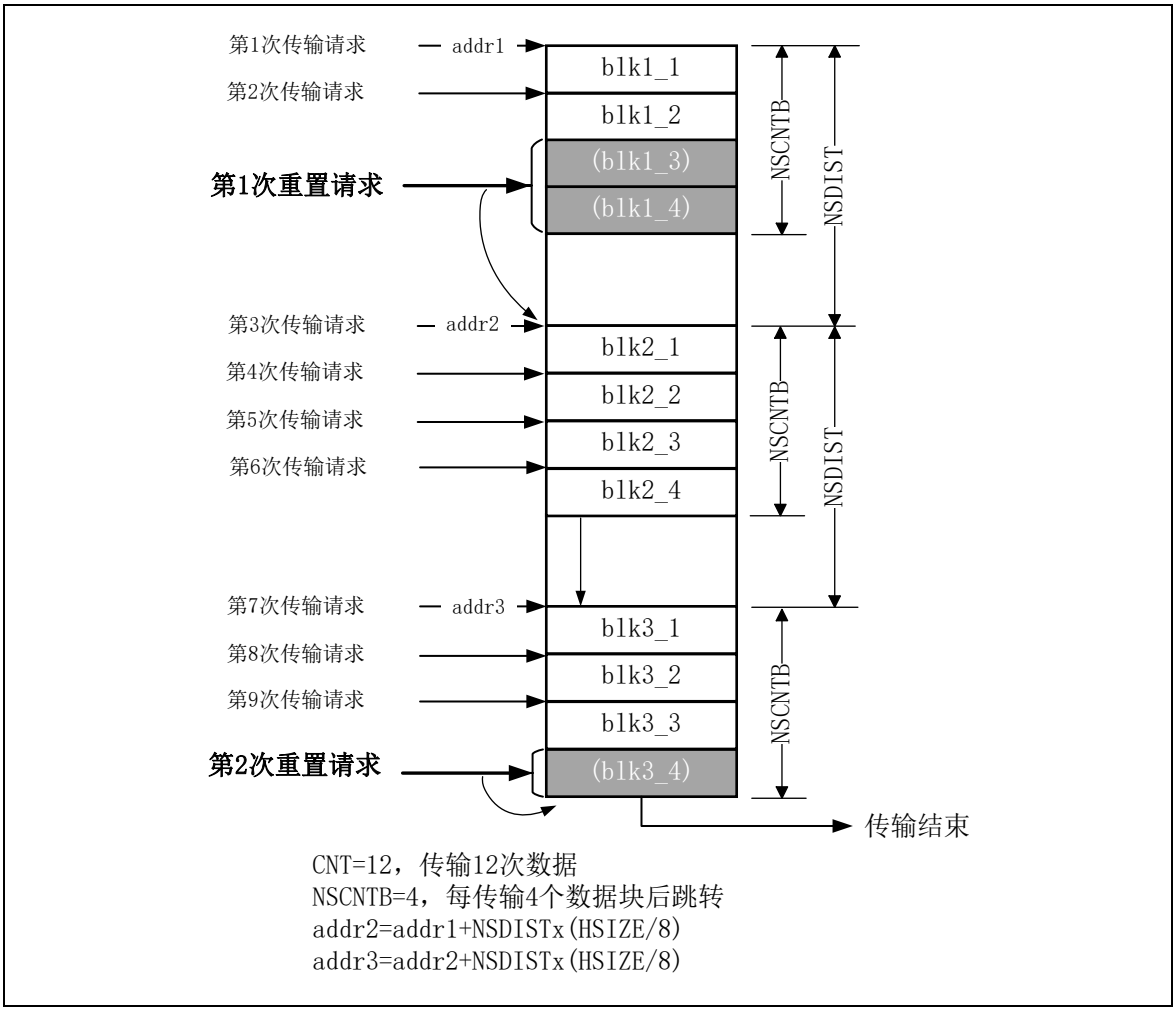


图 15-4 不连续式重置示意图

图 15-4 所示 DMA 动作中，每次传输请求启动一个数据块的传输。第 1 次重置请求发生后，控制器跳过了数据块 blk1_3,blk1_4，传输地址更新为下个不连续区域的首地址即 addr2。第 2 次重置请求发生后，剩余传输次数更新为 0，即所有数据传输完成，通道许可位自动清 0，产生传输完成中断和事件。

15.3.11传输提前终止

传输过程中通道使能寄存器 DMA_CHEN.CHENx 保持有效，非连锁传输时，数据控制寄存器 DMA_DTCTLx 设定的传输次数完成后自动置为无效，连锁传输时，最后一次连锁传输的传输次数完成后自动置为无效。传输过程中如果软件对 DMA_CHENCLR.CHENCLRx 写 1，则 DMA 将在完成当次数据读写后终止传输，且 DMA_CHEN.CHENx 位清 0。

注意：

– 软件对 CHENCLRx 位写 1 提前终止传输时，DMA 内部不会保存被终止时的传输状态。在没有重新设置通道配置寄存器（描述符）状态，将 CHENx 写 1 再次允许此通道，传输请求输入后，DMA 将重新传输被终止的数据块，而不是断点续传。

15.4 应用举例

15.4.1 存储器到存储器的传输

目标：将 22 个数据从 RAM 地址 0x20000000 传送到 0x20001000，数据宽度为 32bit。

1. 寄存器设定

- DMA_EN.EN 写 1 使能 DMA 控制器
- 选择一个通道，例如通道 0，配置通道寄存器以实现：
  - 写 DMA_SAR0 配置源地址为 SRAM 区域 0x20000000
  - 写 DMA_DAR0 配置源地址为 SRAM 区域 0x20001000
  - 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
  - 配置触发源控制器 DMA_TRGSEL0，选择软件触发作为 DMA 通道 0 的启动请求
  - 通道使能位 DMA_CHEN.CHEN0 写 1，使能通道 0
- 写外设事件软件触发寄存器 INTSFTTRG.STRG 为 1，发送第一个软件启动请求，DMA 开始传输数据

2. 传输过程

由于传输的数据块大小为 4，当软件写 INTSFTTRG.STRG 为 1 后开始第一次传输，当一个数据块传输完成后，传输次数 DMA_DTCTL0.CNT 减 1，并产生一个块传输完成中断，软件可以在中断子程序中继续写 INTSFTTRG.STRG 开始第二次传输。在第二次传输中，由于设置了源地址重复区域大小为 6，因此在传输完 2 个地址后源地址将重新载入初始地址 0x20000000 并继续传输剩余的 2 个地址。第二次传输完成后，传输次数 DMA_DTCTL0.CNT 减 1，并产生一个块传输完成中断，软件可以在中断子程序中继续写 INTSFTTRG.STRG 开始第三次传输。第三次传输完成后，传输次数 DMA_DTCTL0.CNT 减为 0，即本次传输全部完成，DMA 产生一个块传输完成中断和一个传输完成中断，同时通道使能位 DMA_CHEN.CHEN0 将被自动清零。

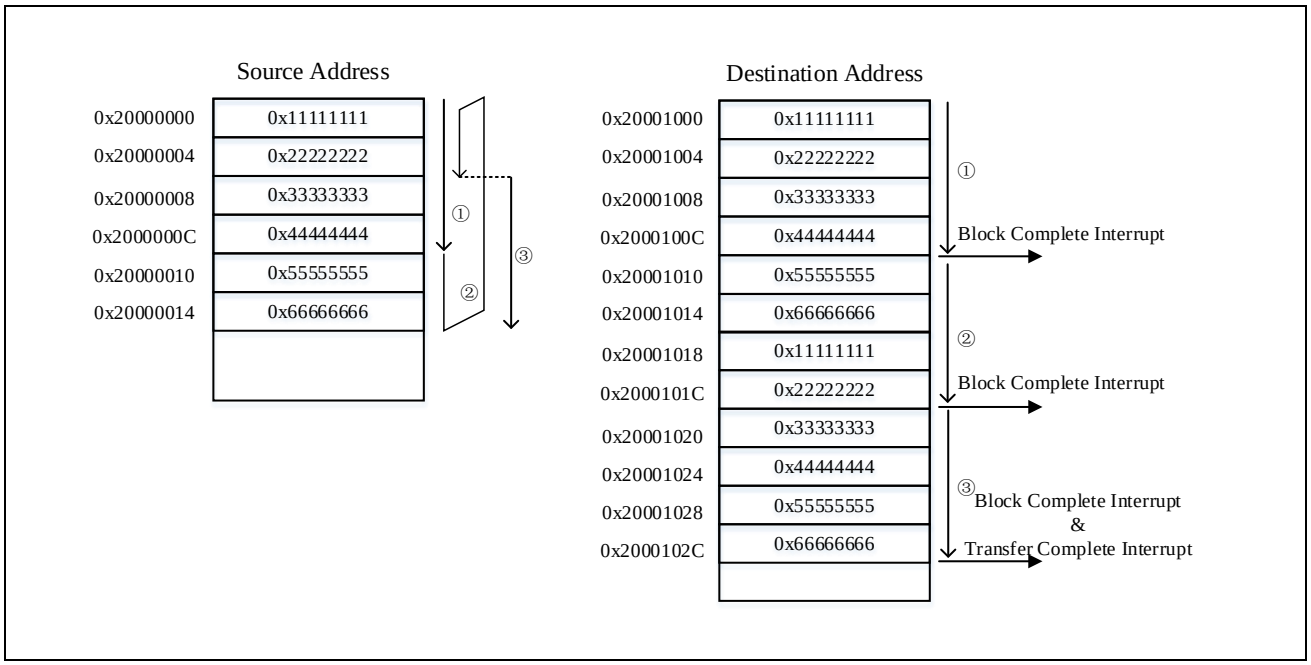


图 15-5 应用例 1：存储器到存储器传输

15.4.2 存储器到外围电路的传输

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

1. 寄存器设定

- DMA_EN.EN 写 1 使能 DMA 控制器

- 配置 DMA_INTMSK 寄存器，屏蔽块传输完成中断，使得传输完成中断
- 选择一个通道，配置通道寄存器，例如选择通道 0
  - 写 DMA_SAR0 配置源地址为 SRAM 区域 0x20000000
  - 写 DMA_DAR0 配置源地址为外围电路的寄存器地址 0x40000000
  - 写 DMA_DTCTL0 配置数据块的大小为 1，传输次数为 10 次，每个传输请求传送一次，每次 1 个数据。
  - 配置通道控制寄存器 DMA_CH0CTL 以实现：
    * 源地址和目标地址连锁传输无效
    * 源地址更新方式为自增，目标地址为固定
    * 源地址和目标地址数据宽度为半字(16bit)
    * 中断使能有效
  - 配置触发源控制器 DMA_TRGSEL0，选择通信模块的发送寄存器空作为 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。

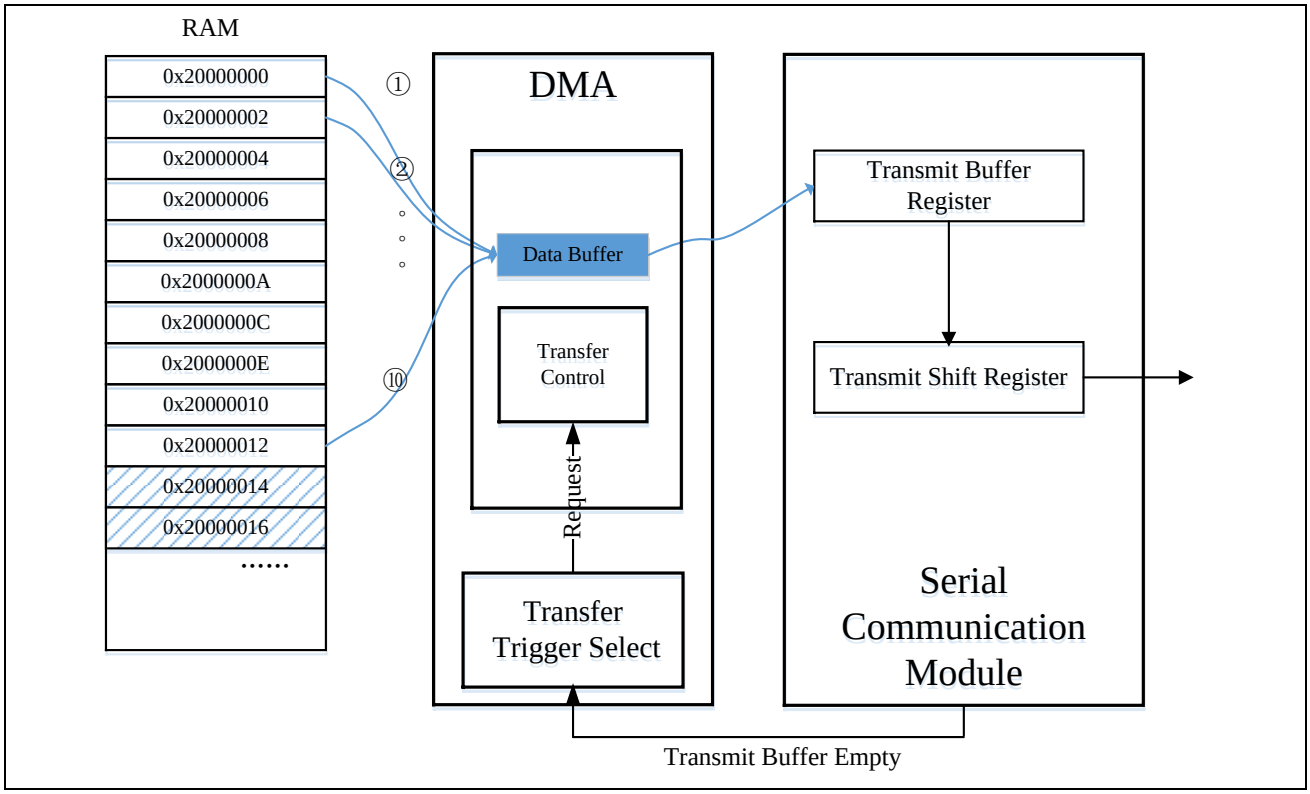


图 15-6 应用例 2：存储器到外围电路的传输

15.4.3 存储器到存储器的连锁传输

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 0x20000000
  - 写 DMA_DAR0 配置源地址为 SRAM 区域 0x20001000
  - 写 DMA_DTCTL0 配置数据块的大小为 10
  - 在链指针寄存器 DMA_LLP0 中写入第二个描述符(descriptor1)的地址 0x20002000
  - 配置通道控制寄存器 DMA_CH0CTL，配置第一个数据块的传输参数以实现：
    * 连锁传输有效
    * 连锁传输模式为直接启动下一次传输
    * 源地址和目标地址更新方式为自增
    * 数据宽度为字(32bit)

    * 中断使能无效

- 在 RAM 空间的 0x20002000 地址中配置第二次传输的描述符(descriptor1)，包括：
  - 0x20002000 中写入 32 位数据 0x20000100，此为第二次传输的源地址
  - 0x20002004 中写入 32 位数据 0x20001100，此为第二次传输的目标地址
  - 0x20002008 中配置数据块的大小为 20
  - 0x20002018 中写入 32 位数据 0x20002020，此为第三次传输的描述符(descriptor2)所在地址
  - 0x2000201C 中写入第二次传输的控制数据，该控制数据实现：
    * 连锁传输有效
    * 连锁传输模式为直接启动下一次传输
    * 源地址和目标地址更新方式为自增
    * 数据宽度为半字(16bit)
    * 中断使能无效
- 在 RAM 空间的 0x20002000 地址中配置第三次传输的描述符(descriptor2)，包括：
  - 0x20002020 中写入 32 位数据 0x20000200，此为第三次传输的源地址
  - 0x20002024 中写入 32 位数据 0x20001200，此为第三次传输的目标地址
  - 0x20002028 配置数据块的大小为 40
  - 0x20002028 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
  - 0x2000202C 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
- 配置传输启动触发源悬着寄存器 DMA_TRGSEL0，选择软件触发作为 DMA 通道 0 的启动请求
- 通道使能位 DMA_CHEN.CHEN0 写 1，使能通道 0
- 写软件触发寄存器 INTSFTTRG.STRG 为 1，发送一个启动请求，DMA 开始传输数据

2. 传输过程

软件启动 DMA 开始传输。第一次传输完成后，由于设置了连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。

15.5 寄存器说明

15.5.1 寄存器一览

单元1 BASE_ADDR:0x40053000

单元2 BASE_ADDR:0x40053400

寄存器名	符号	偏移地址	位宽	复位值
DMA使能寄存器	DMA_EN	0x00	32	0x00000000
中断状态寄存器0	DMA_INTSTAT0	0x04	32	0x00000000
中断状态寄存器1	DMA_INTSTAT1	0x08	32	0x00000000
中断屏蔽寄存器0	DMA_INTMASK0	0x0C	32	0x00000000
中断屏蔽寄存器1	DMA_INTMASK1	0x10	32	0x00000000
中断复位寄存器0	DMA_INTCLR0	0x14	32	0x00000000
中断复位寄存器1	DMA_INTCLR1	0x18	32	0x00000000
通道使能寄存器	DMA_CHEN	0x1c	32	0x00000000
通道使能复位寄存器	DMA_CHENCLR	0x34	32	0x00000000
传输请求状态寄存器	DMA_REQSTAT	0x20	32	0x00000000
传输中通道监视寄存器	DMA_CHSTAT	0x24	32	0x00000000
通道重置控制寄存器	DMA_RCFGCTL	0x2c	32	0x00000000
传输源地址寄存器	DMA_SARx *1	0x40+0x40*x	32	0x00000000
传输目标地址寄存器	DMA_DARx	0x44+0x40*x	32	0x00000000
数据控制寄存器	DMA_DTCTLx	0x48+0x40*x	32	0x00000001
重复区域大小寄存器	DMA_RPTx	0x4C+0x40*x	32	0x00000000
重复区域大小寄存器B	DMA_RPTBx			
源设备不连续地址传输控制寄存器	DMA_SNSEQCTLx	0x50+0x40*x	32	0x00000000
源设备不连续地址传输控制寄存器B	DMA_SNSEQCTLBx			
目标设备不连续地址传输控制寄存器	DMA_DNSEQCTLx	0x54+0x40*x	32	0x00000000
目标设备不连续地址传输控制寄存器B	DMA_DNSEQCTBx			
链指针寄存器	DMA_LLPx	0x58+0x40*x	32	0x00000000
通道控制寄存器	DMA_CHCTLx	0x5C+0x40*x	32	0x00001000
传输源地址监视寄存器	DMA_MONSARx	0x60+0x40*x	32	0x00000000
传输目标地址监视寄存器	DMA_MONDARx	0x64+0x40*x	32	0x00000000
数据控制监视寄存器	DMA_MONDTCTLx	0x68+0x40*x	32	0x00000001
重复区域计数器监视寄存器	DMA_MONRPTx	0x6C+0x40*x	32	0x00000000
源设备不连续传输计数器监视寄存器	DMA_MONSNSEQCTLx	0x70+0x40*x	32	0x00000000
目标设备不连续传输计数器监视寄存器	DMA_MONDNSEQCTLx	0x74+0x40*x	32	0x00000000

寄存器名	符号	地址	位宽	复位值
DMA1传输启动触发源选择寄存器	DMA1_TRGSELx	0x40010814+0x4*x	32	0x000001ff
DMA2传输启动触发源选择寄存器	DMA2_TRGSELx	0x40010834+0x4*x	32	0x000001ff
DMA1,2共用通道重置触发源选择寄存器	DMA_TRGSELRC	0x40010854	32	0x000001ff

注*1: x=0~7

表 15-2 寄存器一览表

15.5.2 DMA 使能寄存器(DMA_EN)

复位值: 0x00000000

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

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

位	标记	位名	功能	读写
b31-b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	EN	DMA使能位	0：DMA无效 1：DMA使能	R/W

15.5.3 中断状态寄存器 0(DMA_INTSTAT0)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	REQERR[7:0]							

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

位	标记	位名	功能	读写
b31-b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23-b16	REQERR[7:0]	传输请求溢出错误中断位	0：该通道没有发生传输请求溢出错误 1：该通道发生了传输请求溢出错误，即上次请求还处在等待状态时又来了传输请求	R
b15-b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7-b0	TRNERR[7:0]	传输错误中断位	0：该通道没有发生传输错误 1：该通道发生了传输错误	R

15.5.4 中断状态寄存器 0(DMA_INTSTAT1)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	BTC[7:0]							

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

位	标记	位名	功能	读写
b31-b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23-b16	BTC[7:0]	块传输完成中断位	该中断在完成一个数据块的传输后发生 0：该通道没有块传输中断发生 1：该通道发生块传输中断	R
b15-b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7-b0	TC[7:0]	传输完成中断位	该中断在完成传输次数寄存器DMA_CNTx设置的传输次数后发生 0：该通道没有传输完成中断发生 1：该通道发生传输完成中断	R

15.5.5 中断屏蔽寄存器(DMA_INTMASK0)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	MSKREQERR[7:0]							

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

位	标记	位名	功能	读写
b31-b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23-b16	MSKREQERR[7:0]	传输请求溢出中断屏蔽	0：不屏蔽传输请求溢出中断 1：屏蔽传输请求溢出中断	R/W
b15-b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7-b0	MSKTRNERR[7:0]	传输错误中断屏蔽	0：不屏蔽传输错误中断 1：屏蔽传输错误中断	R/W

15.5.6 中断屏蔽寄存器(DMA_INTMASK1)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	MSKBTC[7:0]							

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

位	标记	位名	功能	读写
b31-b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23-b16	MSKBTC[7:0]	块传输完成中断屏蔽	0：不屏蔽块传输完成中断 1：屏蔽块传输完成中断	R/W
b15-b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7-b0	MSKTC[7:0]	传输完成中断屏蔽	0：不屏蔽传输完成中断 1：屏蔽传输完成中断	R/W

15.5.7 中断复位寄存器(DMA_INTCLR0)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	CLRREQERR[7:0]							

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

位	标记	位名	功能	读写
b31-b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23-b16	CLRREQERR[7:0]	传输请求溢出错误中断复位	写0无任何效果，写1复位传输请求溢出错误中断状态位 读出永远为0	R/W
b15-b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7-b0	CLRTRNERR[7:0]	传输错误中断复位	写0无任何效果，写1复位传输错误中断状态位 读出永远为0	R/W

15.5.8 中断复位寄存器(DMA_INTCLR1)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	CLRBTC[7:0]							

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

位	标记	位名	功能	读写
b31-b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23-b16	CLRBTC[7:0]	块传输完成中断复位	写0无任何效果，写1复位块传输完成中断状态位 读出永远为0	R/W
b15-b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7-b0	CLRTC[7:0]	传输完成中断复位	写0无任何效果，写1复位传输完成中断状态位 读出永远为0	R/W

15.5.9 通道使能寄存器(DMA_CHEN)

复位值: 0x00000000

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

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

位	标记	位名	功能	读写
b31-b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7-b0	CHEN[7:0]	通道使能位	每位对应一通道。 0：该通道无效 1：该通道有效 对CHEN[x]写1，CHEN[x]置1。对CHEN[x]写0无效。 对DMA_CHENCLR.CHENCLR[x]位写1时，CHEN[x]清0，并强制终止通道x正在进行的传输。传输过程中使能位保持为1，当传输完传输次数寄存器DMA_DTCTLx.CNT设置的传输次数后将会自动清0。如果DMA_DTCTLx.CNT设置为0，则传输完成后不会自动清零，即无限次传输。 注意：请在通道x的8个描述符寄存器全部设置结束后将CHEN[x]写1，使能本通道。 CHEN[x]为1时，对通道x的描述符寄存器的写操作无效。	R/W

15.5.10 通道使能复位寄存器(DMA_CHENCLR)

复位值: 0x00000000

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

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

位	标记	位名	功能	读写
b31-b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7-b0	CHENCLR[7:0]	通道使能位复位	写0无任何效果，写1复位通道使能位CHEN 读出永远为0 写1时若本通道正在传输，则DMA将在完成当次数据读写后强制终止传输。	R/W

15.5.11 通道重置控制寄存器(DMA_RCFGCTL)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	CNTMD[1:0]		DARMD[1:0]		SARMD[1:0]	

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

位	标记	位名	功能	读写
b31-b22	Reserved	-	读出时为“0”,写入时写“0”	R/W
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_CHCTLx.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_CHCTLx.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_CHCTLx.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_CHCTLx.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 依此类推 0x7：通道7 其它：Reserved，设定禁止	R/W
b7-b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	RCFGLLP	链指针式通道重置	0：链指针式重置无效 1：链指针式重置有效 注意：RCFGPLL设为1时，通道将重新载入存储器中的新描述符，故本寄存器的bit16-bit25全部无效。	R/W
b0	RCFGEN	通道重置许可	0：禁止事件触发通道配置寄存器强制更新 1：允许事件触发通道配置寄存器强制更新	R/W

注意：

– 请在 DMA_EN.EN 为 0 时设置本寄存器，本寄存器必须在重置通道的第一次传输前设置。

15.5.12 传输请求状态寄存器(DMA_REQSTAT)

复位值: 0x00000000

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

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

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	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
b14-b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7~b0	CHREQ[7:0]	通道传输请求标志位	每位对应一个通道。 当外部传输请求输入后对应位置1， 当该通道传输启动后，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，但本标志位保持1，继续等待传输。 0：该通道无传输请求 1：该通道有传输请求	R

15.5.13 通道状态观测寄存器(DMA_CHSTAT)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	CHACT[7:0]							

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	RCFGACT	DMAAC

位	标记	位名	功能	读写
b31-b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23-b16	CHACT[7:0]	传输动作中通道监视位	每位对应一个通道。 0：该通道空闲中 1：该通道处于动作中	R
b15-b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	RCFGACT	DMA通道重置动作中监视位	0：DMA未处于通道重置动作中 1：DMA处于通道重置动作中	R
b0	DMAACT	DMA动作中监视位	0：DMA未处于传输动作中 1：DMA处于传输动作中	R

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

复位值: 0x000001ff

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]		-	-	-	-	-	-	-	-	-	-	-	-	-	-

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	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_COMTRG1的公共触发事件触发通道传输 1：允许AOS_COMTRG1的公共触发事件触发通道传输	R/W
b30	COMEN[0]	公共触发使能	0：禁止AOS_COMTRG0的公共触发事件触发通道传输 1：允许AOS_COMTRG0的公共触发事件触发通道传输	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【】。	R/W

15.5.15 通道重置触发源选择寄存器(DMA_TRGSELRC)

复位值: 0x000001ff

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]		-	-	-	-	-	-	-	-	-	-	-	-	-	-

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	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_COMTRG1的公共触发事件触发通道重置 1：允许AOS_COMTRG1的公共触发事件触发通道重置	R/W
b30	COMEN[0]	公共触发使能	0：禁止AOS_COMTRG0的公共触发事件触发通道重置 1：允许AOS_COMTRG0的公共触发事件触发通道重置	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【】。 DMA_1, DMA_2共用一个重置触发源。	R/W

15.5.16 传输源地址寄存器(DMA_SARx) (x=0~7)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SAR[31:16]															

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

位	标记	位名	功能	读写
b8-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_CHCTLx.HSIZE=01，SAR[0]无效。传输数据宽度为32bit时，即DMA_CHCTLx.HSIZE=1x，SAR[1:0]无效。	R/W

15.5.17 传输目标地址寄存器(DMA_DARx) (x=0~7)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DAR[31:16]															

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

位	标记	位名	功能	读写
b31-b0	DAR[31:0]	传输目标地址	设置传输目标地址 注意： – 传输数据宽度为16bit时，即DMA_CHCTLx.HSIZE=01，DAR[0]无效。传输数据宽度为32bit时，即DMA_CHCTLx.HSIZE=1x，DAR[1:0]无效。	R/W

15.5.18 数据控制寄存器(DMA_DTCTLx) (x=0~7)

复位值: 0x00000001

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CNT[15:0]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	BLKSIZE[9:0]									

位	标记	位名	功能	读写
b31-b16	CNT[15:0]	传输次数	总的传输次数，每次请求启动一个数据块的传输，完成时传输次数计数器减1，当减到0时发生传输完成中断。如果设置为0，表示无限次传输，每次启动请求传输一个数据块，完成时传输次数计数器保持0不变，不会产生传输完成中断。	R/W
b15-b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9-b0	BLKSIZE[9:0]	数据块的大小	设置数据块的大小，最大可以配置1024个数据。每个数据的宽度由DMA_CHCTLx寄存器的HSIZE位决定。寄存器值设为1则每次传输1个数据，设为0则每次传输1024个数据。	R/W

15.5.19 重复区域大小寄存器(DMA_RPTx) (x=0~7)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	DRPT[9:0]									

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	SRPT[9:0]									

位	标记	位名	功能	读写
b31-b26	Reserved	-	读出时为“0”,写入时写“0”	R/W
b25-b16	DRPT[9:0]	目标地址重复区域大小	设置目标地址重复区域大小 目标设备在每传输DRPT个数据后目标地址重载为DMA_DARx寄存器初始设定值。寄存器设为10则每传输10个数据后地址重载，设为0则每传1024个数据后地址重载。每个数据的宽度由DMA_CHCTLx寄存器的HSIZE位决定。	R/W
b15-b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9-b0	SRPT[9:0]	源地址重复区域大小	设置源地址重复区域大小 源设备在每传输SRPT个数据后源地址重载为DMA_SARx寄存器初始设定值。寄存器设为10则每传输10个数据后地址重载，设为0则每传1024个数据后地址重载。每个数据的宽度由DMA_CHCTLx寄存器的HSIZE位决定。	R/W

该寄存器配置源地址和目标地址重复区域的大小。使用重复地址传输需要配置 DMA_CHCTLx 寄存器的 SRPTEN/DRPREN 位有效，并且配置 DMA_CHCTLx 寄存器的 SINC/DINC 位使地址更新方式为自增或自减,如果为固定则地址重载功能无效。该寄存器在本通道的重置功能打开时无效。

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

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	DRPT[9:0]									

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	SRPT[9:0]									

位	标记	位名	功能	读写
b31-b26	Reserved	-	读出时为“0”,写入时写“0”	R/W
b25-b16	DRPTB[9:0]	目标地址重复区域大小	设置目标地址重复区域大小 目标设备在每传输DRPTB个数据块后目标地址重载为DMA_DARx寄存器初始设定值。数据块大小由DMA_DTCTLx.BLKSIZE与DMA_CHCTLx.HSIZE决定。	R/W
b15-b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9-b0	SRPTB[9:0]	源地址重复区域大小	设置源地址重复区域大小 源设备在每传输SRPTB个数据块后源地址重载为DMA_SARx寄存器初始设定值。数据块大小由DMA_DTCTLx.BLKSIZE与DMA_CHCTLx.HSIZE决定。	R/W

该寄存器配置源地址和目标地址重复区域的大小。使用重复地址传输需要配置 DMA_CHCTLx 寄存器的 SRPTEN/DRPREN 位有效，并且配置 DMA_CHCTLx 寄存器的 SINC/DINC 位使地址更新方式为自增或自减，如果为固定则重复地址传输功能无效。

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

15.5.21 源设备不连续地址传输控制寄存器(DMA_SNSEQCTLx) (x=0~7)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SNSCNT[11:0]												SOFFSET[19:16]			

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

位	标记	位名	功能	读写
b31-b20	SNSCNT[11:0]	源地址跳转的数据量	设置源地址跳转前传输的数据量大小。 源设备在每传输SNSCNT个数据后源地址按SOFFSET指定偏移量跳转。寄存器设为10则每传输10个数据后地址跳转，设为0则每传4096个数据后地址跳转。	R/W
b19-b0	SOFFSET[19:0]	源地址跳转的地址偏移量	当不连续地址传输时，设置源地址跳转的偏移量。偏移量是相对于当前传输地址，即跳转前最后一次传输地址而言的。跳转的方向根据通道控制寄存器DMA_CHCTLx.SINC的值向前或向后跳转。参考图 15-3。当DMA_CHCTLx.SINC设为地址固定时，不连续地址传输无效。 跳转地址将根据数据的宽度(DMA_CHCTLx.HSIZE)设置的bit数和SOFFSET的值进行计算。 地址偏移量=SOFFSET×(HSIZE(bit)/8) 例如，当SOFFSET设为10，HSIZE为字(32bit)时，地址偏移量为10×4=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_CHCTLx 寄存器的 SNSEQEN 位有效，并且配置 DMA_CHCTLx 寄存器的 SINC 位使地址更新方式为自增或自减。

该寄存器在本通道的重置功能打开时无效。

15.5.22 源设备不连续地址传输控制寄存器 B(DMA_SNSEQCTLBx) (x=0~7)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SNSCNTB[11:0]												SNSDIST[19:16]			

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

位	标记	位名	功能	读写
b31-b20	SNSCNTB[11:0]	源地址跳转的数据量	设置源地址跳转前传输的数据量大小。 源设备在每传输SNSCNTB个数据块后源地址按SNSDIST指定地址间距跳转。数据块大小由DMA_DTCTLx.BLKSIZE与DMA_CHCTLx.HSIZE决定。	R/W
b19-b0	SNSDIST[19:0]	源不连续区域地址间距	当不连续地址传输时，设置源设备两个不连续区域的间距。 跳转的方向根据通道控制寄存器DMA_CHCTLx.SINC的值向前或向后跳转。参考图 15-4。当DMA_CHCTLx.SINC设为地址固定时，不连续地址传输无效。 地址间距将根据数据的宽度(DMA_CHCTLx.HSIZE)设置的bit数和SNSDIST的值进行计算。 地址间距=SNSDIST×(HSIZE(bit)/8) 例如，当SNSDIST设为10，HSIZE为字(32bit)时，地址间距为10×4=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_CHCTLx 寄存器的 SNSEQEN 位有效，并且配置 DMA_CHCTLx 寄存器的 SINC 位使地址更新方式为自增或自减。

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

15.5.23 目标设备不连续地址传输控制寄存器(DMA_DNSEQCTLx) (x=0~7)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DNSCNT[11:0]												DOFFSET[19:16]			

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

位	标记	位名	功能	读写
b31-b20	DNSCNT[11:0]	目标地址跳转的数据量	设置目标地址跳转前传输的数据量大小。 目标设备在每传输DNSCNT个数据后目标地址按DOFFSET指定偏移量跳转。寄存器设为10则每传输10个数据后地址跳转，设为0则每传4096个数据后地址跳转。	R/W
b9-b0	DSOFFSET[19:0]	目标地址跳转的地址偏移量	当不连续地址传输时，设置目标地址跳转的偏移量。偏移量是相对于当前传输地址，即跳转前最后一次传输地址而言的。跳转的方向根据通道控制寄存器DMA_CHCTLx.DINC的值向前或向后跳转。 参考图 15-3。当DMA_CHCTLx.DINC设为地址固定时，不连续地址传输将无效。 跳转地址将根据数据的宽度(DMA_CHCTLx.HSIZE)设置的bit数和DOFFSET的值进行计算。 地址偏移量=DOFFSET×(HSIZE(bit)/8) 例如，当DOFFSET设为10，HSIZE为字(32bit)时，地址偏移量为10×4=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_CHCTLx 寄存器的 DNSEQEN 位有效，并且配置 DMA_CHCTLx 寄存器的 DINC 位使地址更新方式为自增或自减。

该寄存器在本通道的重置功能打开时无效。

15.5.24 目标设备不连续地址传输控制寄存器 B(DMA_DNSEQCTLBx) (x=0~7)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DNSCNTB[11:0]												DNSDIST[19:16]			

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

位	标记	位名	功能	读写
b31-b20	DNSCNTB[11:0]	目标地址跳转的数据量	设置目标地址跳转前传输的数据量大小。 目标设备在每传输SNSCNTB个数据块后目标地址按DNSDIST指定地址间距跳转。数据块大小由DMA_DTCTLx.BLKSIZE与DMA_CHCTLx.HSIZE决定。	R/W
b19-b0	DNSDIST[19:0]	目标不连续区域地址间距	当不连续地址传输时，设置目标设备两个不连续区域的间距。 跳转的方向根据通道控制寄存器DMA_CHCTLx.DINC的值向前或向后跳转。参考图 15-4。当DMA_CHCTLx.DINC设为地址固定时，不连续地址传输将无效。 地址间距将根据数据的宽度(DMA_CHCTLx.HSIZE)设置的bit数和DNSDIST的值进行计算。 地址间距=DNSDIST×(HSIZE(bit)/8) 例如，当DNSDIST设为10，HSIZE为字(32bit)时，地址间距为10×4=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_CHCTLx 寄存器的 DNSEQEN 位有效，并且配置 DMA_CHCTLx 寄存器的 DINC 位使地址更新方式为自增或自减。

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

15.5.25 链指针寄存器(DMA_LLPx) (x=0~7)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
LLP[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LLP[15:2]														-	-

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

15.5.26 通道控制寄存器(DMA_CHCTLx) (x=0~7)

复位值: 0x00001000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b199	b188	b177	b166
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	IE	LLPRUN	LLPEN	HSIZE[1:0]		DNSEQEN	SNSEQEN	DRPTEN	SRPTEN	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	连锁传输模式选择	连锁传输有效时，设置当前传输完成时，载入链指针指向的新描述符后是否立即开始新描述符对应的传输 0：不立即传输，等待下一个传输请求产生后开始传输 1：新描述符载入后，立即开始传输	R/W
b10	LLPEN	连锁传输使能	0：连锁传输无效 1：连锁传输有效	R/W
b9-b8	HSIZE[1:0]	传输数据的宽度	00：8bit 01：16bit 10, 11：32bit	R/W
b7	DNSEQEN	目标地址不连续传输使能	0：不允许不连续地址传输 1：允许不连续地址传输	R/W
b6	SNSEQEN	源地址不连续传输使能	0：不允许不连续地址传输 1：允许不连续地址传输	R/W
b5	DRPTEN	目标重复传输功能使能位	设置是否允许目标地址重新载入初始值 0：不重载 1：重载	R/W
b4	SRPTEN	源重复传输功能使能位	设置是否允许源地址重新载入初始值 0：不重载 1：重载	R/W
b3-b2	DINC[1:0]	目标地址的更新方式	00：固定 01：递增 10, 11：递减	R/W
b1-b0	SINC[1:0]	源地址的更新方式	00：固定 01：递增 10, 11：递减	R/W

15.5.27 通道监视寄存器(DMA_MONSARx，DMA_MONDARx, DMA_MONDTCTLx, DMA_MONRPTx, DMA_MONSNSEQCTLx, DMA_MONDNSEQCTLx) (x=0~7)

这些监视寄存器与相应的通道配置寄存器对应，寄存器位配置一致，但全部为只读寄存器。

通道配置寄存器在 DMA 传输前后保持不变，而通道监视寄存器会在 DMA 每完成一个请求对应的传输后，即每完成一个数据块传输后更新。更新内容和方式如下：

- DMA_MONSARx.SAR[31:0]，DMA_MONDARx.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_MONDTCTLx.CNT[15:0]：减 1，若已经是 0 则保持为 0。
- DMA_MONRPTx.SRPT[9:0]，DRPT[9:0]：通道重置无效时，减去块大小，减到 0 时重载 DMA_RPTx 设定值。重置有效时，减 1，减到 0 时重载原始设定值。
- DMA_MONSNSEQCTLx.SNSCNT[11:0]，DMA_MONDNSEQCTLx.DNSCNT[11:0]：通道重置无效时，减去块大小，减到 0 时重载 DMA_SNSEQCTLx/DMA_DNSEQCTLx 原始设定值。重置有效时，减 1，减到 0 时重载原始设定值。
- 以上之外的监视寄存器位保持与配置寄存器一致。

15.6 使用注意事项

DMA 的寄存器只支持 32bit 读写，8/16bit 读写操作无效。

16 电压比较器（CMP）

16.1 简介

电压比较器（Comparator，以下简称 CMP）是将两个模拟电压进行比较的外设模块，共有四个比较通道 CMP1~4。

CMP 主要特性：

- CMP1~4 均可独立进行电压比较
- CMP1，2 或者 CMP3，4 同时使用时可完成窗口比较
- 正端电压和负端电压均有多个输入电压源供选择
- 数字噪声滤波器的采样时钟可选
- 可使用定时器输出的 PWM 波形对输出进行开关控制
- 可产生触发其他外设启动的事件
- 可产生中断并可唤醒系统停止模式
- 比较结果可输出到外部管脚 VCOUT

16.2 功能框图

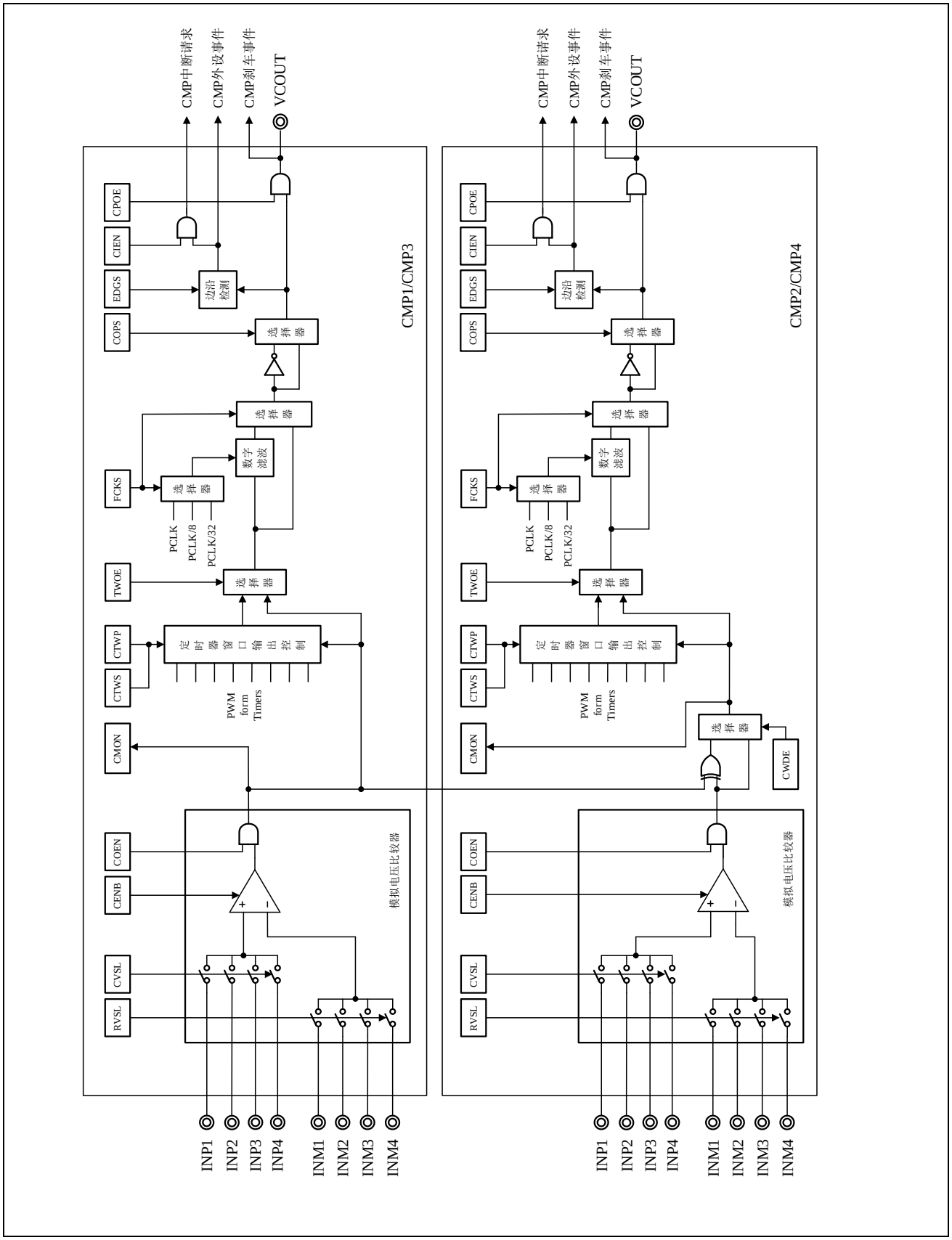


图 16-1 功能框图

引脚名称	输入/输出	功能
AVCC0	输入	模拟电源
AVSS0	输入	模拟地
CMP1_INP2	输入	CMP1正端模拟输入
CMP1_INP3	输入	CMP1正端模拟输入
CMP1_INP4	输入	CMP1正端模拟输入
CMP123_INM3	输入	CMP1，2，3负端模拟输入
CMP1_INM4	输入	CMP1负端模拟输入
CMP2_INP3	输入	CMP2正端模拟输入
CMP2_INP4	输入	CMP2正端模拟输入
CMP2_INM4	输入	CMP2负端模拟输入
CMP3_INP2	输入	CMP3正端模拟输入
CMP3_INP3	输入	CMP3正端模拟输入
CMP3_INP4	输入	CMP3正端模拟输入
CMP3_INM4	输入	CMP3负端模拟输入
CMP4_INP3	输入	CMP4正端模拟输入
CMP4_INP4	输入	CMP4正端模拟输入
CMP4_INM3	输入	CMP4负端模拟输入
CMP4_INM4	输入	CMP4负端模拟输入
VCOUT1	输出	CMP1的外部端口输出
VCOUT2	输出	CMP2的外部端口输出
VCOUT3	输出	CMP3的外部端口输出
VCOUT4	输出	CMP4的外部端口输出
VCOUT	输出	CMP1~4的外部端口输出

表 16-1 CMP 引脚

16.3 功能说明

16.3.1 普通比较模式

普通比较模式时，CMP1~4 可以独立运行。以 CMP1 为例，设定步骤如下：

1） PWR_FCG3.bit3 写 0，并等待至少 2us。

2） 关闭 CMP 的模块停止功能（PWR_FCG3.bit8 写 0）。

3） 设定 CMP_PMSR 寄存器的 CVSL 位，选择正端电压；

设定 CMP_PMSR 寄存器的 RVSL 位，选择负端电压。

4） CMP_MDR 寄存器的 CENB 位设为 1，启动比较器。

5） 等待模拟电压比较器的稳定时间 tCMP。

6） 设定 CMP_FIR 寄存器的 FCKS 位，设置数字滤波器；

设定 CMP_FIR 寄存器的 EDGS、CIEN 位，设置中断和边沿检测条件。

7） 设定 CMP_OCR 寄存器的 CPOE、COPS 位，设置 VCOUT 的输出；

设定 CMP_OCR 寄存器的 TWOE、TWOL 位和 CMP_TWSR 和 CMP_TWPR 寄存器，设置定时器窗口输出。

8） CMP_OCR 寄存器的 COEN 位设为 1，允许比较器输出。

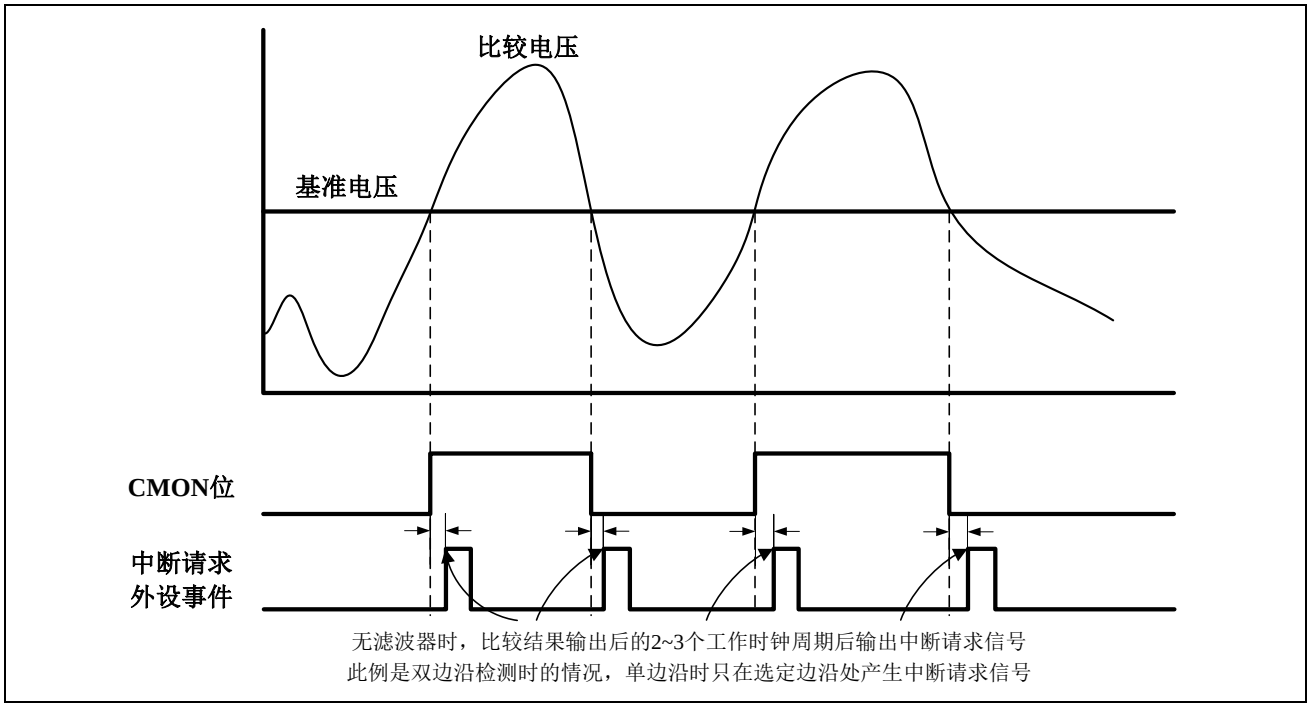


图 16-2 普通比较模式工作示意图

如图 16-2，在普通比较模式且为正极性输出时，正端电压高于负端电压，CMP_MDR 寄存器的 CMON 位为 1；正端电压低于负端电压，CMON 位为 0。

对于 CMP1 和 CMP3 来说，如果选择 INP2 或者 INP3 作为正端电压，那么还需要通过设定 CMP_VISR 选择电压输入源。具体参考表 16-3。

16.3.2 窗口比较模式

CMP1，2 或者 CMP3，4 同时运行时可实现窗口比较。以 CMP1，2 为例，设定步骤如下：

1） PWR_FCG3.bit3 写 0，并等待至少 2us。

2） 关闭 CMP1，2 的模块停止功能（PWR_FCG3.bit8 写 0）。

3） 设定 CMP1_PMSR 寄存器的 CVSL 位，选择正端电压；

设定 CMP1_PMSR 寄存器的 RVSL 位，选择下限负端电压；

4） 设定 CMP2_PMSR 寄存器的 CVSL 位，选择正端电压；

注意所选输入电压源必须与（2）中一致。

设定 CMP2_PMSR 寄存器的 RVSL 位，选择上限负端电压。

5） CMP2_MDR 寄存器的 CWDE 位设为 1，选择窗口比较模式。

6） CMP1_MDR 和 CMP2_MDR 寄存器的 CENB 位设为 1，启动比较器。

7） 等待电压比较器的稳定时间 tCMP。

8） 设定 CMP2_FIR 寄存器的 FCKS 位，设置数字滤波器；

设定 CMP2_FIR 寄存器的 EDGS、CIEN 位，设置中断和边沿检测条件。

9） 设定 CMP2_OCR 寄存器的 CPOE、COPS 位，设置 VCOUT 的输出；

设定 CMP2_OCR 寄存器的 TWOE、TWOL 位和 CMP2_TWSR 和 CMP2_TWPR 寄存器，设置定时器窗口输出。

10） CMP1_OCR，CMP2_OCR 寄存器的 COEN 位设为 1，允许比较器输出。

窗口比较时，由 CMP2 或者 CMP4 完成对比较结果的监视、滤波、中断和输出等操作。

如图 16-3 所示，如果选择正极性输出，当正端电压介于下限负端电压和上限负端电压之间时 CMP2_MDR 寄存器的 CMON 位为 1；否则该 CMON 位为 0。

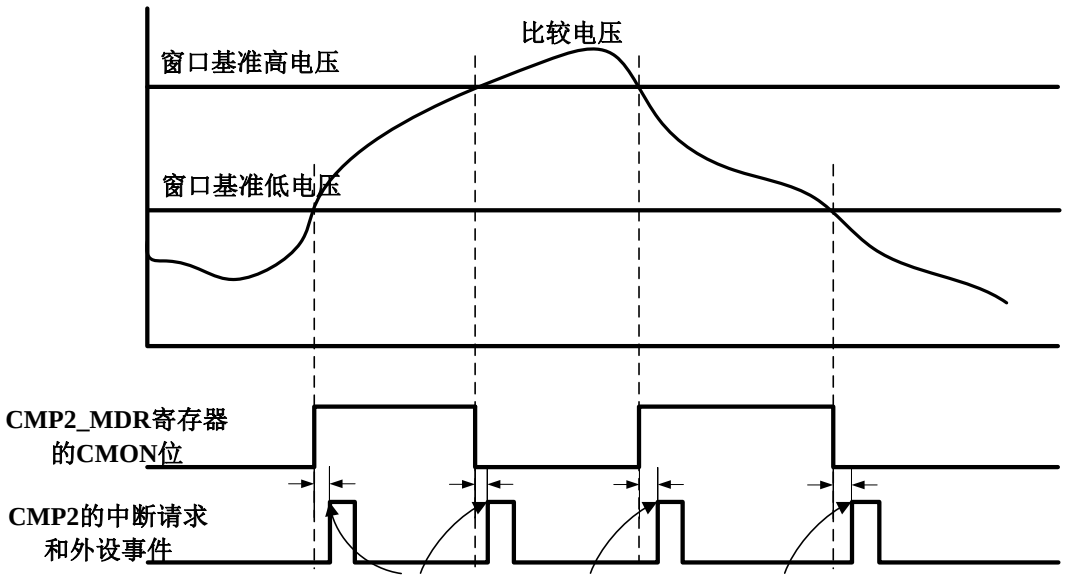


图 16-3 窗口比较模式工作示意图

16.3.3 定时器窗口输出

定时器窗口输出无效（CMP_OCR 寄存器的 TWOE 位为 0）时，电压比较器的输出只受 CMP_OCR 寄存器的 COEN 位控制。COEN 位为 1 时允许输出，否则输出禁止并固定为低电平。

定时器窗口输出有效（CMP_OCR 寄存器的 TWOE 位为 1）时，电压比较器的输出除了受 CMP_OCR 寄存器的 COEN 位控制外，还受到由 CMP_TWSR 寄存器选择的定时器窗口信号的控制。可以通过 CMP_TWPR 寄存器设定允许输出时定时器窗口信号的极性，并通过 CMP_OCR 寄存器的 TWOL 位设定禁止输出时定时器窗口输出端的固定电平。

16.3.4 数字滤波器

四个比较通道都配有数字滤波器，可对各自的电压比较器的输出进行噪声滤波。数字滤波器按照采样时钟对电压比较器的输出进行采样，在三次采样电平相同后的下一个采样时钟将采样值输出，否则保持输出不变。采样时钟可以通过 CMP_FIR 寄存器的 FCKS 位选择，当 FCKS 位设为 00b 时数字滤波器关闭。

16.3.5 比较器中断

四个比较通道均可产生比较器中断请求。使用中断时，必须将 CMP_FIR 寄存器的 CIEN 位设置成 1，并通过 CMP_FIR 寄存器的 EDGS 位（2'b00 以外的值）设定产生中断的比较器输出边沿。如果还设定了数字滤波和反相输出，那么 EDGS 位的设定则是检测经过数字滤波以及反相输出后信号的边沿。

比较器中断还可以唤醒停止低功耗模式。设定中断有效（CIEN=1）并且关闭数字滤波和边沿检测，中断将在比较器输出由低变高时发生并唤醒停止低功耗模式。如果还设定了反相输出，那么中断将在反相输出由低变高时发生并唤醒停止低功耗模式。回到工作模式后，请先设定中断无效（CIEN=0）后再进行后续操作。

16.3.6 外设触发事件

和中断请求的产生条件相同，通过检测 CMP_FIR 寄存器设定的数字滤波器的输出边沿产生触发其他外设的事件信号。但是和中断请求不同的是外设事件总是输出的，与 CMP_FIR 寄存器的 CIEN 位无关。必须通过目标外设的硬件触发源寄存器启动或关闭事件触发功能。

16.3.7 外部管脚输出

可以将经过数字滤波器的比较结果输出到外部端口 VCOUT。通过 CMP_OCR 寄存器的 CPOE、COPS 两位可分别设定是否允许输出以及输出极性（正相输出或者反相输出）。复位后 VCOUT 对应的端口为通用端口且为输入状态，因此还必须通过端口寄存器将其设定为 VCOUT 输出功能。

16.4 注意事项

16.4.1 模块停止功能

CMP 有模块停止功能，通过设定模块停止寄存器可将模块的数字部分关闭。CMP1,2 共用一个模块停止控制位 PWR_FCG3.bit8，CMP3,4 共用一个模块停止控制位 PWR_FCG3.bit9。CMP1~4 初始均为停止状态，设定模块工作时才可以访问各自的寄存器。相关寄存器设定请参考低功耗章节。

16.4.2 模块停止时的动作

CMP 在工作状态进入到模块停止状态时，比较器将继续保持工作，功耗等同于工作状态。如需进一步降低功耗，请将 CMP_MDR 寄存器的 CENB 位清“0”。

16.4.3 停止低功耗模式时的动作

芯片进入停止低功耗模式时，如果 CMP 处于工作状态，在进入停止低功耗模式时后将继续保持工作，功耗等同于进入停止低功耗模式之前的水平。如需进一步降低功耗，请在进入停止低功耗模式前将 CMP_MDR 寄存器的 CENB 位清“0”。

16.5 寄存器说明

CMP1 基准地址： 0x4000A000

CMP2 基准地址： 0x4000A010

CMP3 基准地址： 0x4000A400

CMP4 基准地址： 0x4000A410

寄存器名	符号	偏移地址	位宽	复位值
比较器模式设定寄存器	CMPx_MDR， x=1~4	0x00	8	0x00
比较器滤波和中断控制寄存器	CMPx_FIR， x=1~4	0x01	8	0x00
比较器输出控制寄存器	CMPx_OCR， x=1~4	0x02	8	0x00
比较器正负端输入选择寄存器	CMPx_PMSR， x=1~4	0x03	8	0x00
比较器定时器窗口选择寄存器	CMPx_TWSR， x=1~4	0x04	16	0x0000
比较器定时器窗口极性寄存器	CMPx_TWPR， x=1~4	0x06	16	0x0000
比较器电压输入源选择寄存器	CMPx_VISR，x=1，3	0x08	16	0x0000

表 16-2 CMP 寄存器一览表

16.5.1 比较器模式设定寄存器（CMPx_MDR，x=1~4）

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
CMON	-	-	-	-	-	CWDE	CENB

位	标记	位名	功能	读写
b7	CMON	比较器结果监视	普通比较模式 0：正端电压＜负端电压 1：正端电压＞负端电压 窗口比较模式（此位只对CMP2和CMP4有效） 0：正端电压＜下限负端电压，或者 正端电压＞上限负端电压 1：下限负端电压＜正端电压＜上限负端电压	R
b6~b2	Reserved	-	读出时为“0”，写入时写“0”	R/W
b1	CWDE	窗口比较选择	0：普通比较模式 1：窗口比较模式 注意： 此位只存在于CMP2和CMP4中。设为窗口比较模式后，对窗口比较结果的滤波、边沿检测以及输出等控制均由CMP2或者CMP4完成。	R/W
b0	CENB	比较器工作许可	0：电压比较器停止 1：电压比较器工作 注意： 每次将CENB由“0”设为“1”后，需要等待约300ns的工作稳定时间才可以进行后续操作。	R/W

16.5.2 比较器滤波和中断控制寄存器（CMPx_FIR，x=1~4）

复位值：0x00

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

位	标记	位名	功能	读写
b7	Reserved	-	读出时为“0”，写入时写“0”	R/W
b6	CIEN	比较器中断许可	0：禁止比较器中断 1：允许比较器中断	R/W
b5~b4	EDGS[1:0]	比较器输出边沿检测选择	0 0：不检测比较器输出的边沿 0 1：检测比较器输出的上升沿 1 0：检测比较器输出的下降沿 1 1：检测比较器输出的上升沿和下降沿 注意： 改变EDGS[1:0]可能会引起中断或者外设触发事件，因此请在中断禁止或者外设触发功能无效状态下设定该寄存器。寄存器设定后，请清除相应的中断标志。	R/W
b3~b2	Reserved	-	读出时为“0”，写入时写“0”	R/W
b1~b0	FCKS[1:0]	比较器结果滤波采样选择	0 0：不使用噪声滤波器 0 1：使用噪声滤波器，通过PCLK进行采样 1 0：使用噪声滤波器，通过PCLK/8进行采样 1 1：使用噪声滤波器，通过PCLK/32进行采样 注意： 请在比较器输出禁止（即COEN位为 “0”）时改写FCKS[1:0]。 FCKS[1:0]从 “00b ” 切换到其他值时，请使用经过4次采样后的滤波输出作为中断请求或外设触发事件。 改变FCKS[1:0]可能会引起中断或者外设触发事件，因此请在中断禁止或者外设触发功能无效状态下设定该寄存器。寄存器设定后，请清除相应的中断标志。	R/W

16.5.3 比较器输出控制寄存器（CMPx_OCR，x=1~4）

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	TWOL	TWOE	CPOE	COPS	COEN

位	标记	位名	功能	读写
b7~b5	Reserved	-	读出时为“0”，写入时写“0”	R/W
b4	TWOL	定时器窗口输出电平选择	0：定时器窗口输出在输出禁止时固定为低电平 1：定时器窗口输出在输出禁止时固定为高电平	R/W
b3	TWOE	定时器窗口输出使能	0：关闭定时器窗口输出功能（普通输出模式） 1：打开定时器窗口输出功能	R/W
b2	CPOE	VCOUT输出允许	0：禁止VCOUT输出 1：允许VCOUT输出	R/W
b1	COPS	比较器输出极性选择	0：比较器正相输出 1：比较器反相输出 注意： 请在比较器输出禁止（即COEN位为 ”0”）时改写COPS。改变COPS位可能会引起中断或者外设触发事件，因此请在中断禁止或者外设触发功能无效状态下设定该寄存器。寄存器设定后，请清除相应的中断标志。	R/W
b0	COEN	比较器输出允许	0：禁止比较器输出（比较器输出固定成低电平） 1：允许比较器输出	R/W

16.5.4 比较器正负端输入选择寄存器（CMPx_PMSR，x=1~4）

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
CVSL[3:0]				RVSL[3:0]			

位	标记	位名	功能	读写
b7~b4	CVSL[3:0]	正端输入选择	选择比较器的正端输入电压 0000：无正端电压输入 0001：选择INP1作为正端电压（所选输入见表 16-3） 0010：选择INP2作为正端电压（所选输入见表 16-3） 0100：选择INP3作为正端电压（所选输入见表 16-3） 1000：选择INP4作为正端电压（所选输入见表 16-3） 其他：设定禁止 注意： 请在比较器输出禁止（即COEN位为 “0”）时改写CVSL。每次改写CVSL后请等待300ns再允许比较器输出（即COEN位设为“1”）。改变CVSL可能会引起中断或者外设触发事件，因此请在中断禁止或者外设触发功能无效状态下设定该寄存器。寄存器设定后，请清除相应的中断标志。	R/W
b3~b0	RVSL[3:0]	负端输入选择	选择模拟电压比较器的负端输入电压 0000：无负端电压输入 0001：选择INM1作为负端电压（所选输入见表 16-4） 0010：选择INM2作为负端电压（所选输入见表 16-4） 0100：选择INM3作为负端电压（所选输入见表 16-4） 1000：选择INM4作为负端电压（所选输入见表 16-4） 其他：设定禁止 注意： 请在比较器输出禁止（即COEN位为 “0”）时改写RVSL。每次改写RVSL后请等待300ns再允许比较器输出（即COEN位设为“1”）。改变RVSL可能会引起中断或者外设触发事件，因此请在中断禁止或者外设触发功能无效状态下设定该寄存器。寄存器设定后，请清除相应的中断标志。	R/W

16.5.5 比较器电压输入源选择寄存器（CMPx_VISR，x=1，3）

CMP1_VISR 复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	-	-	-	-

b7	b6	b5	b4	b3	b2	b1	b0
-	-	P3SL[1:0]		-	P2SL[2:0]		

位	标记	位名	功能	读写
b15~b6	Reserved	-	读出时为“0”，写入时写“0”	R/W
b5~b4	P3SL[1:0]	INP3输入选择	选择CMP1_PMSR.CVSL[3:0]为3’b0100时的输入电压 00：无电压输入 01/10：所选输入见表 16-3 11：设定禁止 注意：请在CMP1_PMSR.CVSL[2]为“0”时改写P3SL。	R/W
b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2~b0	P2SL[2:0]	INP2输入选择	选择CMP1_PMSR.CVSL[3:0]为3’b0010时的输入电压 000：无电压输入 001/010/100：所选输入见表 16-3 其他：设定禁止 注意：请在CMP1_PMSR.CVSL[1]为“0”时改写P2SL。	R/W

CMP3_VISR 复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	-	-	-	-

b7	b6	b5	b4	b3	b2	b1	b0
-	-	P3SL[1:0]		-	P2SL[2:0]		

位	标记	位名	功能	读写
b15~b6	Reserved	-	读出时为“0”，写入时写“0”	R/W
b5~b4	P3SL[1:0]	INP3输入选择	选择CMP3_PMSR.CVSL[3:0]为3’b0100时的输入电压 00：无电压输入 01/10：所选电压见表 16-3 11：设定禁止 注意：请在CMP3_PMSR.CVSL[2]为“0”时改写P3SL。	R/W
b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2~b0	P2SL[2:0]	INP2输入选择	选择CMP3_PMSR.CVSL[3:0]为3’b0010时的输入电压 000：无电压输入 001/010/100：所选电压见表 16-3 其他：设定禁止 注意：请在CMP3_PMSR.CVSL[1]为“0”时改写P2SL。	R/W

比较通道	CVSL=0001b	CVSL=0010b		CVSL=0100b		CVSL=1000b
CMP1	PGA1_BP	**P2SL=001b**	PGA1	**P3SL=01b**	CMP1_INP3	CMP1_INP4
		P2SL=010b	PGA2	**P3SL=10b**	CMP2_INP3	
		P2SL=100b	CMP1_INP2			
CMP2	PGA2_BP	PGA2		CMP2_INP3		CMP2_INP4
CMP3	PGA3_BP	**P2SL=001b**	PGA3	**P3SL=01b**	CMP3_INP3	CMP3_INP4
		P2SL=010b	PGA4	**P3SL=10b**	CMP4_INP3	
		P2SL=100b	CMP3_INP2			
CMP4	PGA4_BP	PGA4		CMP4_INP3		CMP4_INP4

注：表中 PGA1_BP/PGA2_BP/PGA3_BP/PGA4_BP 分别为 PGA1~4 输出的未经放大的电压信号，使用时也要将 PGA 设为有效。

表 16-3 正端输入电压一览表

比较通道	RVSL=0001b	RVSL=0010b	RVSL=0100b	RVSL=1000b
CMP1	DA1O1	DA1O2	CMP123_INM3	CMP1_INM4
CMP2	DA1O1	DA1O2	CMP123_INM3	CMP2_INM4
CMP3	DA2O1	DA2O2	CMP123_INM3	CMP3_INM4
CMP4	DA2O1	DA2O2	CMP4_INM3	CMP4_INM4

表 16-4 负端输入电压一览表

16.5.6 比较器定时器窗口选择寄存器（CMPx_TWSR，x=1~4）

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8
CTWS15	CTWS14	CTWS14	CTWS12	CTWS11	CTWS10	CTWS9	CTWS8
b7	b6	b5	b4	b3	b2	b1	b0
CTWS7	CTWS6	CTWS5	CTWS4	CTWS3	CTWS2	CTWS1	CTWS0

位	标记	位名	功能	读写
b15~0	CTWS15~0	定时器窗口使能位	定时器窗口输出有效时选择作为比较结果输出窗口的定时器PWM输出。 0：不选择定时器的PWM输出作为窗口信号 1：选择定时器的PWM输出作为窗口信号	R/W

CTWS bit	CMP1	CMP2	CMP3	CMP4
15	TIM4_1_OWL	TIM4_2_OWL	TIM4_3_OWL	TIM4_3_OWL
14	TIM4_1_OWH	TIM4_2_OWH	TIM4_3_OWH	TIM4_3_OWH
13	TIM4_1_OVL	TIM4_2_OVL	TIM4_3_OVL	TIM4_3_OVL
12	TIM4_1_OVH	TIM4_2_OVH	TIM4_3_OVH	TIM4_3_OVH
11	TIM4_1_OUL	TIM4_2_OUL	TIM4_3_OUL	TIM4_3_OUL
10	TIM4_1_OUH	TIM4_2_OUH	TIM4_3_OUH	TIM4_3_OUH
9	TIM6_4_PWMA	TIM6_8_PWMA	TIM6_4_PWMB	TIM6_8_PWMB
8	TIM6_3_PWMA	TIM6_7_PWMA	TIM6_3_PWMB	TIM6_7_PWMB
7	TIM6_2_PWMA	TIM6_6_PWMA	TIM6_2_PWMB	TIM6_6_PWMB
6	TIM6_1_PWMA	TIM6_5_PWMA	TIM6_1_PWMB	TIM6_5_PWMB
5	TIMA_2_PWM3	TIMA_4_PWM3	TIMA_3_PWM3	TIMA_4_PWM3
4	TIMA_2_PWM2	TIMA_4_PWM2	TIMA_3_PWM2	TIMA_4_PWM2
3	TIMA_2_PWM1	TIMA_4_PWM1	TIMA_3_PWM1	TIMA_4_PWM1
2	TIMA_1_PWM3	TIMA_3_PWM3	TIMA_1_PWM3	TIMA_2_PWM3
1	TIMA_1_PWM2	TIMA_3_PWM2	TIMA_1_PWM2	TIMA_2_PWM2
0	TIMA_1_PWM1	TIMA_3_PWM1	TIMA_1_PWM1	TIMA_2_PWM1

表 16-5 定时器窗口 PWM 一览表

16.5.7 比较器定时器窗口极性寄存器（CMPx_TWPR，x=1~4）

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8
CTWP15	CTWP14	CTWP13	CTWP12	CTWP11	CTWP10	CTWP9	CTWP8
b7	b6	b5	b4	b3	b2	b1	b0
CTWP7	CTWP6	CTWP5	CTWP4	CTWP3	CTWP2	CTWP1	CTWP0

位	标记	位名	功能	读写
b15~b0	CTWP15~0	定时器窗口极性选择	CTWP15~0用于设定CMPx_TWSR选定的窗口信号的有效电平。 0：窗口信号为低电平时允许比较器输出 1：窗口信号为高电平时允许比较器输出	R/W

17 模数转换模块（ADC）

17.1 简介

12 位 ADC 是一种采用逐次逼近方式的模拟数字转换器。本 MCU 搭载 3 个 ADC 单元，单元 1 和 2 支持 16 个通道，单元 3 支持 20 个通道，可以转换来自外部引脚、以及芯片内部的模拟信号。模拟输入通道可以任意组合成一个序列，一个序列可以进行单次扫描转换，或连续扫描转换。支持对任意指定通道进行连续多次转换并对转换结果进行平均。ADC 模块还搭载模拟看门狗功能，对任意指定通道的转换结果进行监视，检测其是否超出用户设定的范围。

ADC 主要特性

- 高性能
  - 可配置 12 位、10 位和 8 位分辨率
  - ADC 数字接口时钟 PCLK4 和转换时钟 PCLK2（也称作 ADCLK）的频率比可设置为 1:1、 2:1、 4:1、 8:1、 1:2、 1:4

    PCLK2 可选与系统时钟 HCLK 异步的 PLL 时钟，此时频率比

    PCLK4:PCLK2=1:1

    PCLK2 频率最高支持 60MHz
  - 采样率：2.5MSPS（PCLK2=60MHz，12 位，采样 11 周期，变换 13 周期）
  - 各通道采样时间独立编程
  - 各通道独立数据寄存器
  - 数据寄存器可配置左/右对齐方式
  - 连续多次转换平均功能
  - 模拟看门狗，监视转换结果
  - 不使用时可以将 ADC 模块设定成停止状态
- 模拟输入通道
  - 总计 28 个外部模拟输入，单个 ADC 单元最大支持 20 通道
  - 2 个内部模拟输入：内部基准电压，VBAT 分压
- 转换开始条件

  - 软件设置转换开始
  - 周边外设同步触发转换开始
  - 外部引脚触发转换开始
- 转换模式
  - 2 个扫描序列 A、B，可任意指定单个或多个通道
  - 序列 A 单次扫描
  - 序列 A 连续扫描
  - 双序列扫描，序列 A、B 独立选择触发源，序列 B 优先级高于 A
  - 协同工作模式（适用于具有两个或三个 ADC 的设备）
- 中断与事件信号输出
  - 序列 A 扫描结束中断和事件 ADC_EOCA
  - 序列 B 扫描结束中断和事件 ADC_EOCB
  - 模拟看门狗 0 比较中断和事件 ADC_CMP0
  - 模拟看门狗 1 比较中断和事件 ADC_CMP1
  - 上述的 4 个事件输出都可启动 DMA，仅搭载了 DMA 的产品支持

17.2 ADC 系统框图

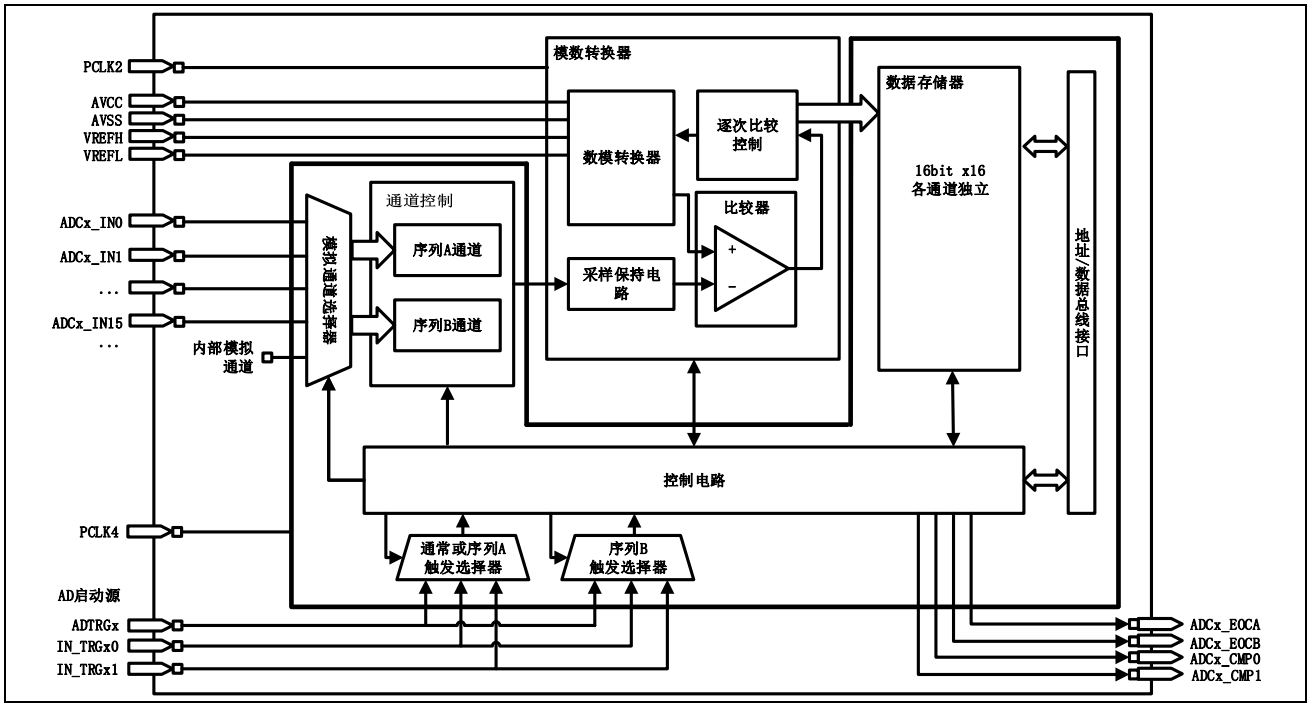


图 17-1 ADC 框图

本芯片搭载了 3 个 ADC 模块单元，每个单元的配置有所不同，具体参考下表：

项目		单元1（ADC1）	单元2（ADC2）	单元2（ADC3）
电源		AVCC		
		AVSS/VREFL		
基准电压		VREFH		
模拟通道 *1	CH0	ADC123_IN0	ADC123_IN0	ADC123_IN0
	CH1	ADC123_IN1	ADC123_IN1	ADC123_IN1
	CH2	ADC123_IN2	ADC123_IN2	ADC123_IN2
	CH3	ADC123_IN3	ADC123_IN3	ADC123_IN3
	CH4	ADC12_IN4	ADC12_IN4	ADC3_IN4
	CH5	ADC12_IN5	ADC12_IN5	ADC3_IN5
	CH6	ADC12_IN6	ADC12_IN6	ADC3_IN6
	CH7	ADC12_IN7	ADC12_IN7	ADC3_IN7
	CH8	ADC12_IN8	ADC12_IN8	ADC3_IN8
	CH9	ADC12_IN9	ADC12_IN9	ADC3_IN9
	CH10	ADC123_IN10	ADC123_IN10	ADC123_IN10
	CH11	ADC123_IN11	ADC123_IN11	ADC123_IN11
	CH12	ADC123_IN12	ADC123_IN12	ADC123_IN12
	CH13	ADC123_IN13	ADC123_IN13	ADC123_IN13
	CH14	ADC12_IN14	ADC12_IN14	ADC3_IN14
	CH15	ADC12_IN15/ 内部模拟通道	ADC12_IN15/ 内部模拟通道	ADC3_IN15/ 内部模拟通道
	CH16	-	-	ADC3_IN16
	CH17	-	-	ADC3_IN17
	CH18	-	-	ADC3_IN18
	CH19	-	-	ADC3_IN19
可编程增益放大器PGA	PGA1/4	ADC123_IN0	ADC12_IN6	-
	PGA2	ADC123_IN1	-	-
	PGA3	ADC123_IN2	-	-
	COM	PGA123_VSS	PGA4_VSS	-
专用采样保持电路SH	SH1	ADC123_IN0	-	-
	SH2	ADC123_IN1	-	-
	SH3	ADC123_IN2	-	-
硬件触发源	外部引脚	ADTRG1	ADTRG2	ADTRG3
	片内周边	IN_TRG10	IN_TRG20	IN_TRG30
		IN_TRG11	IN_TRG21	IN_TRG31

表 17-1 各 ADC 单元规格

注意：ADC 中虚拟通道 CH0~CH15 与物理通道 ADCx_INy（实际的模拟输入源）可以设置寄存器进行自由映射，本表所示为复位后默认的映射关系。

17.3 功能说明

17.3.1 ADC 时钟

ADC 模块需要使用 2 个时钟：数字接口时钟 PCLK4，模拟电路时钟 PCLK2。PCLK4 与 PCLK2 是同步关系，频率比率可设置为 1：1，2：1，4：1，8：1，1：2，1：4。PCLK2 可以选择与系统时钟 HCLK 异步的 PLL 时钟源，此时 PCLK4 与 PCLK2 相同，为同步同频率关系。

注意：

– PCLK2 的频率请设置在 1MHz~60MHz 以内。

17.3.2 通道选择

ADC 模块支持通道映射，即模块中的虚拟通道与实际的物理通道间的映射。虚拟通道是指 ADC 模块中假定的通道，如寄存器 ADC_CHSELRA 设置为 0x1 表示序列 A 选择转换 CH0，这个 CH0 就是虚拟通道，而寄存器 ADC_DR0，是虚拟通道 CH0 的转换结果寄存器。物理通道是指实际存在的模拟通道，包含外部引脚的模拟输入 ADCx_INy 以及内部基准电压等内部模拟通路。虚拟通道与物理通道的映射可以通过寄存器 ADC_CHMUXR 进行配置，具体参考寄存器说明。本章中若无特别说明，通道 n 或 CHn 均表示虚拟通道。

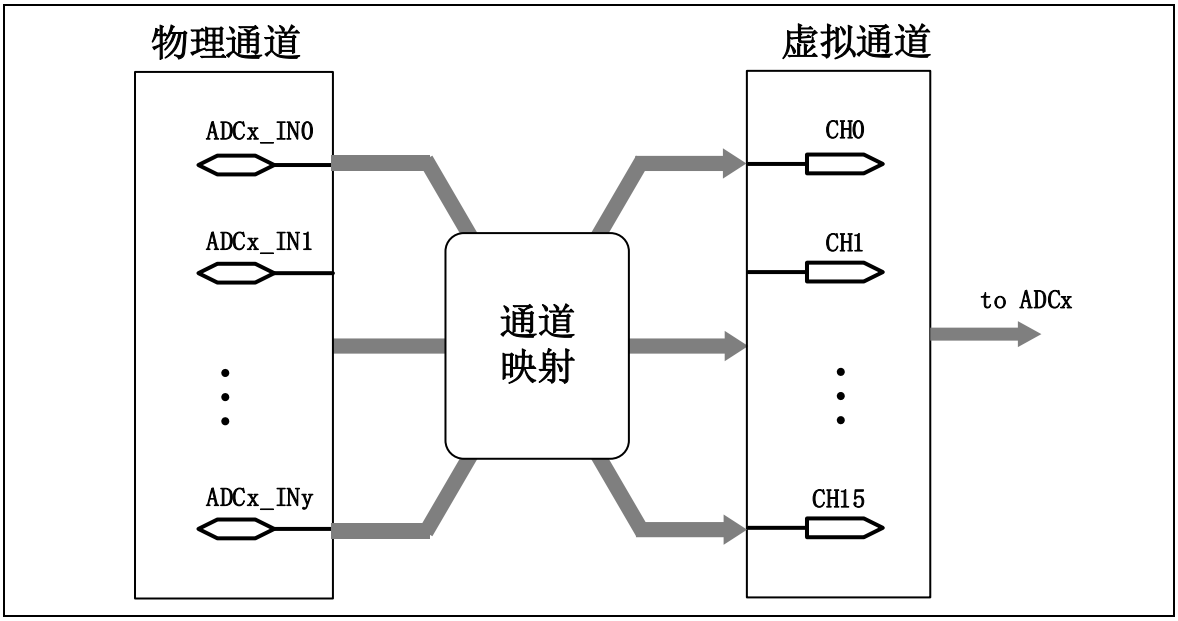


图 17-2 通道映射示意图

ADC 模块有多个通道，可配置为两个序列：序列 A，序列 B 进行转换。序列 A 和 B

配有独立的通道选择寄存器 ADC_CHSELRA，ADC_CHSELRB。寄存器每位代表一个通道，如 bit0 位写 1 表示转换 CH0，写 0 表示不转换 CH0。两个序列可独立选择任意 1 个或多个通道进行转换。例如：ADC_CHSELRA 设置为 0x0055，ADCHSELRB 设置为 0x0002，则序列 A 的触发条件发生时，将依次转换 CH0，CH2，CH4 和 CH6 这 4 个通道。序列 B 的触发条件发生时，将转换 CH1 这一个通道。

其中，内部模拟输出与外部输入 ADCx_IN15 共用 CH15。设置 ADC_CHMUX 同时改变内部模拟输出和 ADCx_IN15 与 CHn 的映射关系。当需要转换内部模拟通道或 ADCx_IN15 时，需要先设置扩展通道寄存器 ADC_EXCHSELR，然后将 ADC_CHSELRA/ADC_CHSELB 中与之对应的虚拟通道位写 1（默认映射条件下为 bit15）。当 ADC_EXCHSELR 设置为 0x0 时，表示转换外部模拟输入，当 ADC_EXCHSELR 设置为 0x1 时，表示转换内部模拟通道。2 个内部模拟量的选择请参考电源控制(PWC)章节的寄存器 PWR_PWRC4 说明。

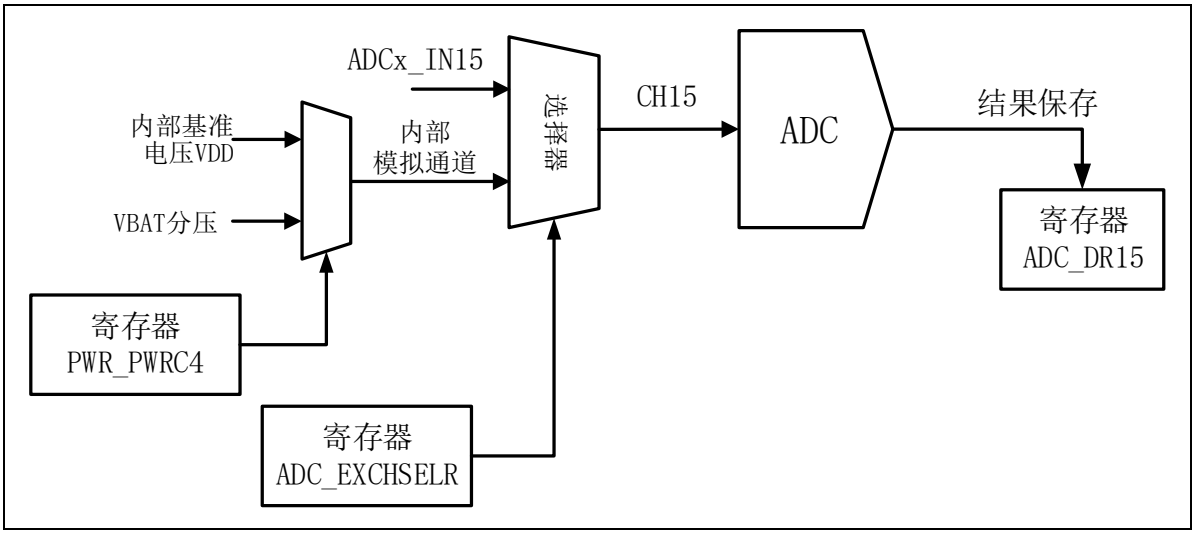


图 17-3 内部模拟通道选择

另外，大部分的物理通道可以输入到多个 ADC 模块单元，如 ADC123_IN0 可输入到 ADC_1,2,3 三个单元，ADC12_IN4 可输入 ADC_1,2 两单元。结合多 ADC 协同工作模式以及通道映射功能，可以方便的对指定通道实现高采样率的转换。

注意：

– 不要在序列 A 和 B 中选择相同的通道。对于不存在的通道，请不要设置相应的寄存器，保持其复位后的状态。

17.3.3 触发源选择

序列 A，序列 B 独立选择触发源。可选择的触发源包括外部端口 ADTRGx，内部事件 IN_TRGx0, IN_TRGx1。其中，端口 ADTRGx 下降沿输入有效。IN_TRGx0,IN_TRGx1 由寄存器 ADC_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 可生成序列 A 软件触发信号，序列 A 软件触发只能在 ADC 处于待机状态时使用。软件触发与触发源选择寄存器 ADC_TRGSR 的设置无关。

17.3.4 序列 A 单次扫描模式

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。

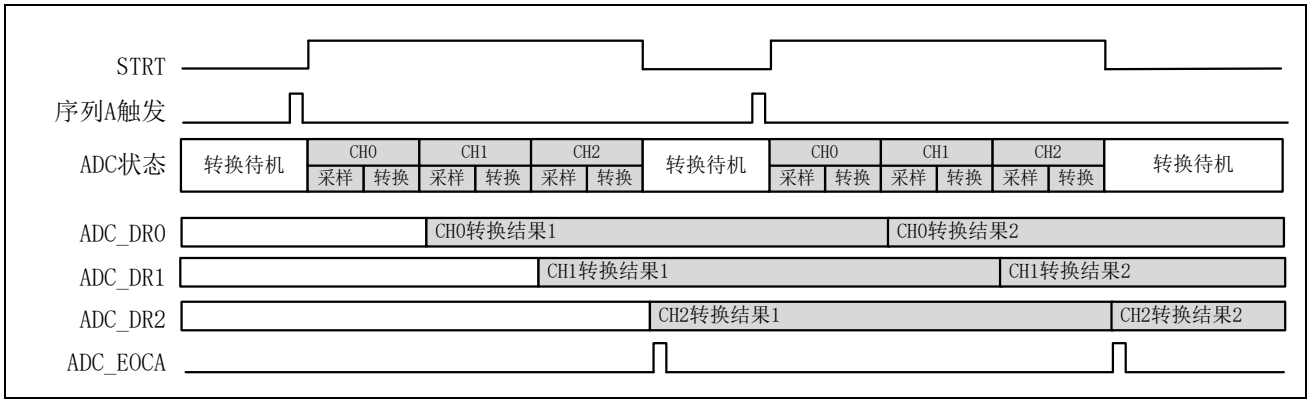


图 17-4 序列 A 单次扫描模式

序列 A 单次扫描模式的软件流程：

1. 确认 ADC_STR.STRT 为 0，ADC 处于转换待机状态。
2. 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.STRT 写 1 软件触发序列 A，或者设置寄存器 ADC_TRGSR 选择序列 A 触发条件。
6. 查询序列 A 转换结束标志位 EOCAF。
7. 读取各通道数据寄存器 ADC_DR。
8. 写 0 清除 EOCAF 标志位，为下次转换准备。

上面的步骤 6~8 的 CPU 查询方式也可以替换成中断方式，利用 ADC_EOCA 中断来处理转换数据。或者利用 ADC_EOCA 事件启动 DMA 读取数据。

17.3.5 序列 A 连续扫描模式

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。

序列 A 连续扫描模式与序列 A 单次扫描模式动作类似，不同点在于，连续模式在所用通道转换结束后不是进入转换待机状态，而是重新开始转换序列 A。STRT 位也不会自动清 0。

当需要停止连续扫描时，对 STRT 位写 0，并读 STRT 确认为 0 以判断 ADC 进入转换待机状态。

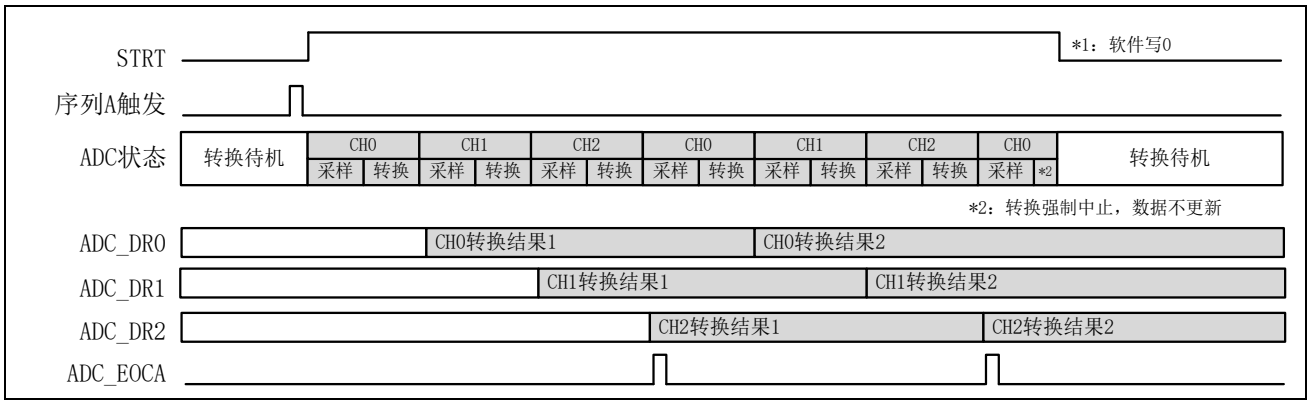


图 17-5 连续扫描

序列 A 连续扫描模式的软件流程：

1. 确认 ADC_STR.STRT 为 0，ADC 处于转换待机状态。
2. 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.STRT 写 1 软件触发序列 A，或者设置寄存器 ADC_TRGSR 选择序列 A 触发条件。

6. 查询序列 A 转换结束标志位 EOCAF。
7. 读取各通道数据寄存器 ADC_DR。
8. 写 0 清楚 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。

17.3.6 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 17-2 序列 A 和 B 的各种竞争

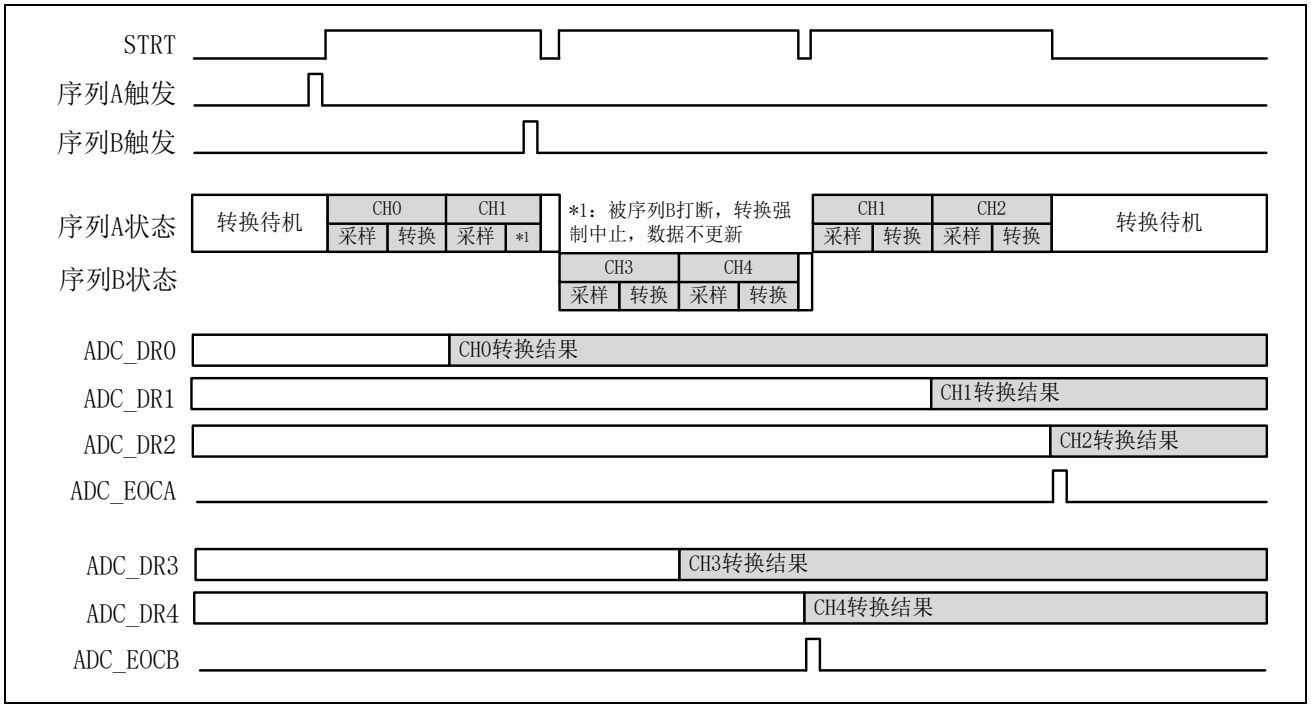


图 17-6 双序列扫描模式（序列 A 从被中断通道重新启动）

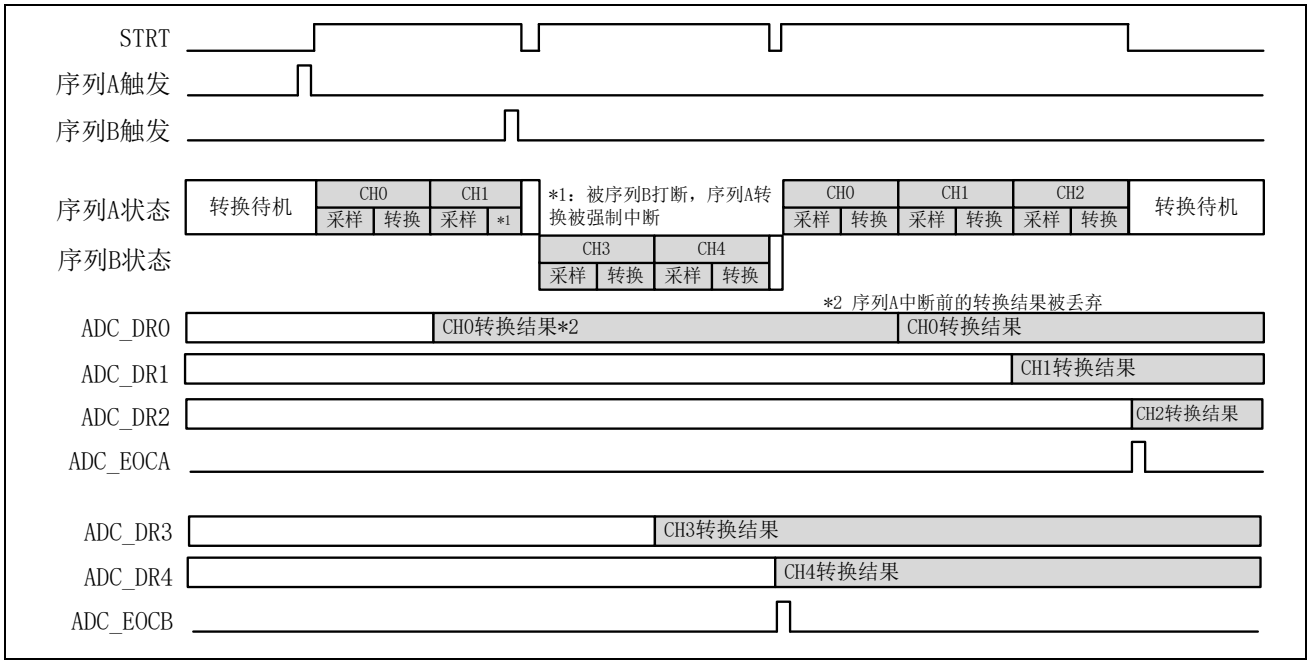


图 17-7 双序列扫描模式（序列 A 从第一个通道重新启动）

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。

17.3.7 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，本 ADC 支持 2 个比较窗口：比较窗口 0、比较窗口 1。以比较窗口 0 为例，如下图所示，若转换结果在保护区域内，则生成看门狗比较中断和事件 ADC_CMP0。

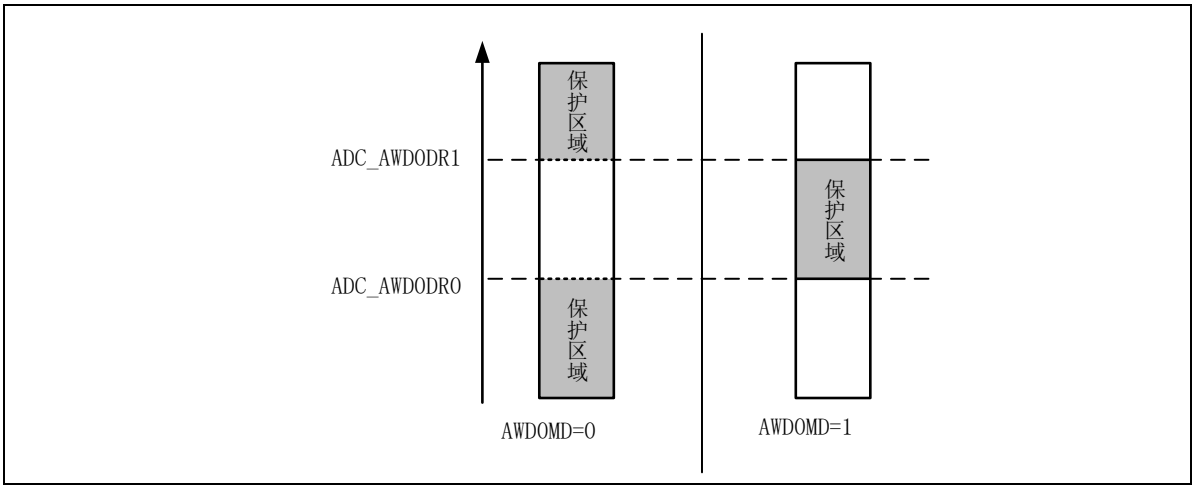


图 17-8 模拟看门狗保护区域（比较模式）

使用模拟看门狗功能的软件流程，以比较窗口 0 为例：

1. 设置阀值寄存器 AD_AWD0DR0，ADC_AWD0DR1
2. 设置比较通道寄存器 ADC_AWD0CHSR，选择需要比较的通道
3. 设置 ADC_AWDCR.AWD0MD 选择比较模式
4. 设置 ADC_AWDCR.AWD0IEN 中断许可位。
5. 设置 ADC_AWDCR.AWD0EN 允许模拟看门狗 0 功能
6. 根据前文，设置扫描模式，启动 AD 进行转换。
7. 在 ADC_CMP0 中断中，或 A/D 转换结束后，查询比较状态寄存器 ADC_AWDSR.AWD0F，对比较结果进行做相应的处理。

比较窗口 1 的使用方法与比较窗口 0 相同。

两个比较窗口可以组合使用。当窗口组合功能有效时，在窗口 1 选的通道转换结束时，比较中断 ADC_CMP1 输出的不再是窗口 1 单独的比较结果，而是按照设置，对窗口 0 和 1 的比较结果进行逻辑或、逻辑与或者逻辑异或后产生的组合结果。使用窗口组合比较功能的软件流程与窗口单独使用时相似，需要在设置好窗口 0 和 1 之后，追加设置 ADC_AWDCR.AWDCM[1:0]寄存器选择组合方式。

17.3.8 模拟输入的采样时间和转换时间

在单次扫描模式中，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_SSTR 调整采样周期数。t_{CMP} 表示逐次比较时间，12 位精度 13 个 PCLK2，10 位精度 11 个 PCLK2，8 位精度 9 个 PCLK2。

一次扫描转换的时间 $t_{SCAN}=t_D+\sum t_{CONV}+t_{ED}$。其中$\sum t_{CONV}$ 表示所有扫描通道的转换时间总和，由于可以独立设置采样时间 t_{SPL}，各通道的转换时间 t_{CONV} 可以不同。

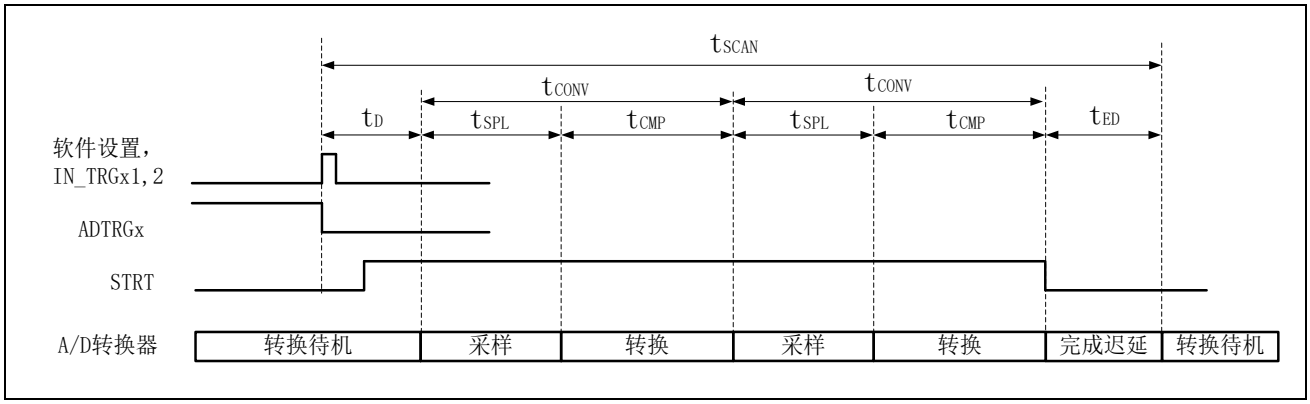


图 17-9 A/D 转换时间

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		扫描开始处理时间	ADC空闲中，启动转换	4 PCLK2	3 PCLK4 + 4 PCLK2 + 1 PCLK4_SYNC	2 PCLK4 + 4 PCLK2	3 PCLK2
			序列A转换中被打断，启动序列B转换	5 PCLK2	4 PCLK4 + 5 PCLK2 + 1 PCLK4_SYNC	2 PCLK4 + 5 PCLK2	-
t_{CONV}	t_{SPL}	采样时间		ADSSTRx.SST[7:0] x PCLK2			
	t_{CMP}	逐次转换时间	12位分辨率	13 PCLK2			
			10位分辨率	11 PCLK2			
			8位分辨率	9 PCLK2			
t_{ED}		扫描完成处理时间		3 PCLK2			
t_{TD}		最小连续触发时间间隔		$\sum t_{CONV}$ + 6 PCLK2			

表 17-3 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(即 CMU_SCFGR 设置的时钟)，此时的 PCLK4，PCLK2 相同，均为异步的 PLL 时钟。

17.3.9 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，待测模拟量（0x0222）由于某种原因未被转换或结果未被更新到数据寄存器 ADC_DR 中，ADC_DR 寄存器继续保持前次转换值（0x0111）。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 转换值是否有效，需要额外将以前的转换值存储到 RAM 中，通过对比转换结果来判断。
- 如果 ADC_CR0.CLREN 为“1”，自动清除功能许可的情况下，前次的转换结果（0x0111）被 CPU 或 DMA 读取后，ADC_DR 寄存器将自动被清除为“0x0000”，此后进行 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。

17.3.10 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_AVCHSELR 选择任意一个或多个需要平均的通道。

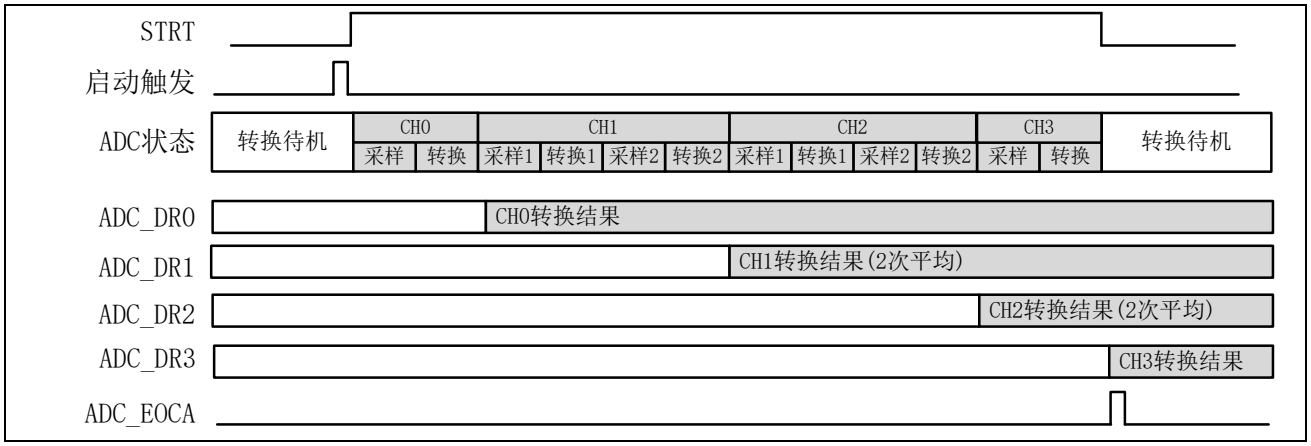


图 17-10 平均功能有效时的转换动作

图 17-10 中，序列 A 单次扫描模式，选择转换 CH0~3 这 4 个通道，其中 CH1、2 设置为 2 次平均模式。在扫描过程中，CH1、2 都会进行连续两次转换，并将平均后的结果保存至对应通道的数据寄存器 ADC_DR1、2 中。

17.3.11 可编程增益放大器 PGA

本 MCU 搭载了可编程增益放大器 PGA，可以设置寄存器 ADC_PGACR，有效 PGA 电路并选择增益倍数，增益范围 x2~x32 可选择。此时，模拟输入先经过 PGA 电路进行放大，然后再输入到 ADC 模块进行转换。

使用 PGA 前请先设置寄存器 PWR_FCG3 的 bit3 允许 PGA，并等待 2us 的启动稳定时间。

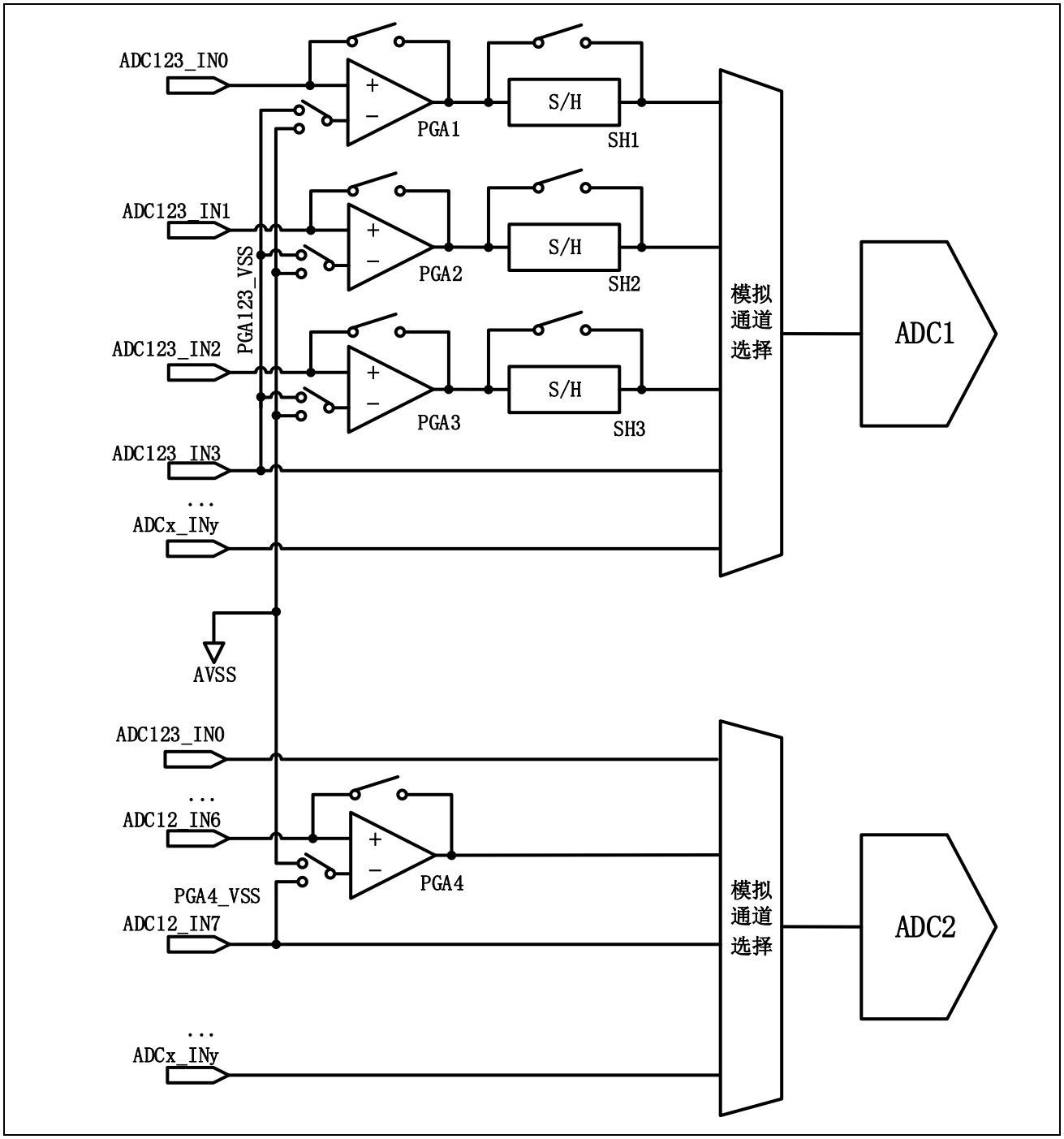


图 17-11 ADC 单元 1,单元 2 的 PGA 与 SH 通道示意图

如上图，ADC1 包含 PGA1~3 共 3 个 PGA 通道，分别与物理通道 ADC123_IN0~2 对应。ADC2 包含 PGA4 共 1 个 PGA 通道，与物理通道 ADC12_IN6 对应。

PGA 的参考地可以选择内部的 AVSS 也可以选择外部的 PGAx_VSS，当选择外部的 PGAx_VSS 时请将其接与 AVSS 相同的电压。

17.3.12 通道专用采样保持电路 SH

本 MCU 搭载 3 个单元专用的采样保持电路（SH）。当专用采样保持电路有效时，每次序列启动时，先同时对所有 SH 有效的通道进行采样，然后再启动 ADC 开始依次对序列中的每个通道进行 A/D 转换。连续扫描模式时，序列在第二次以及之后的扫描启动时都会插入 SH 的采样时间。

可以设置寄存器 ADC_SHCR.SHSEL 位，有效 SH 电路，设置寄存器 ADC_SHCR.SHSST[7:0]修改 SH 电路的采样周期数。

使用 SH 前请先设置寄存器 PWR_FCG3 的 bit3 允许 SH，并等待 2us 的启动稳定时间。

参考图 17-12，ADC1 包含 3 路 SH，可以与 PGA 组合使用，即先经过 PGA 放大，再由 SH 采样保持，输入到 ADC 进行转换。也可以无效 PGA，外部引脚输入直接由 SH 采样，再输入到 ADC 进行转换。

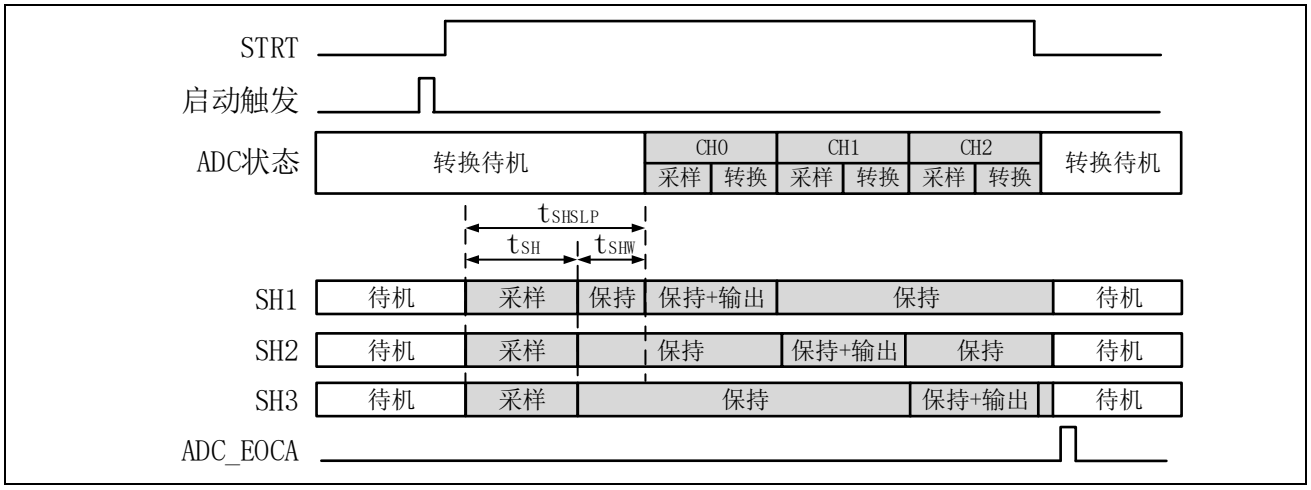


图 17-12 通道专用采样保持电路有效

如上图所示，CH0~2 的专用采样保持电路 SH1~3 均设置为有效。序列启动触发后，SH1~3 电路预先对通道 CH0~2 的模拟输入同时进行采样，即图中的 t_{SHSLP} 时间。然后对 SH0~2 保持的模拟电压进行 A/D 转换。其中，$t_{SHSLP}=t_{SH}+t_{SHW}$。t_{SH} 表示专用采样保持电路采样时间，由寄存器 ADC_SHCR.SHSST[7:0]设定，请设置在 0.4μs 以上。t_{SHW} 表示采样保持电路采样完成后的等待时间，为 12 个 PCLK2 周期。

注意：

– 双序列扫描模式时，序列 A 中不要使用 SH。因为序列 A 可能会被序列 B 打断导致各专用采样保持电路的采样时间不一致而输出期待之外的转换结果。

17.3.13 多 ADC 协同工作模式

在搭载有两个或三个 ADC 模块的芯片上，可以使用 ADC 协同工作模式。

在 ADC 协同工作模式下，ADC1 作为主控单元，通过 ADC1 的触发信号来同步 ADC2 和 ADC3 的转换。即 ADC2 和 ADC3 的序列 A 触发源选择寄存器 ADC_TRGSR.TRGSELA[2:0]设置无效。所有 ADC 模块均由 ADC1 的序列 A 触发源选择寄存器选定的触发源来触发。该模式下 ADC_STR.STRT 寄存器写 1 不会启动转换，即软件启动无效。

使用协同工作模式时，请禁止序列 B 动作，以免打乱同步。

可以设置 ADC1，ADC2 两个 ADC 模块协同工作，也可以 ADC1，ADC2，ADC3 三个 ADC 模块协同工作。根据产品具体规格，ADC3 可能未搭载。

ADC 可配置成以下四种协同工作模式：

- 单次并行触发模式
- 单次延迟触发模式
- 循环并行触发模式
- 循环延迟触发模式

单次并行触发模式

ADC1 的序列 A 触发条件同时触发处于协同工作模式的所有 ADC 模块，且只触发一次。

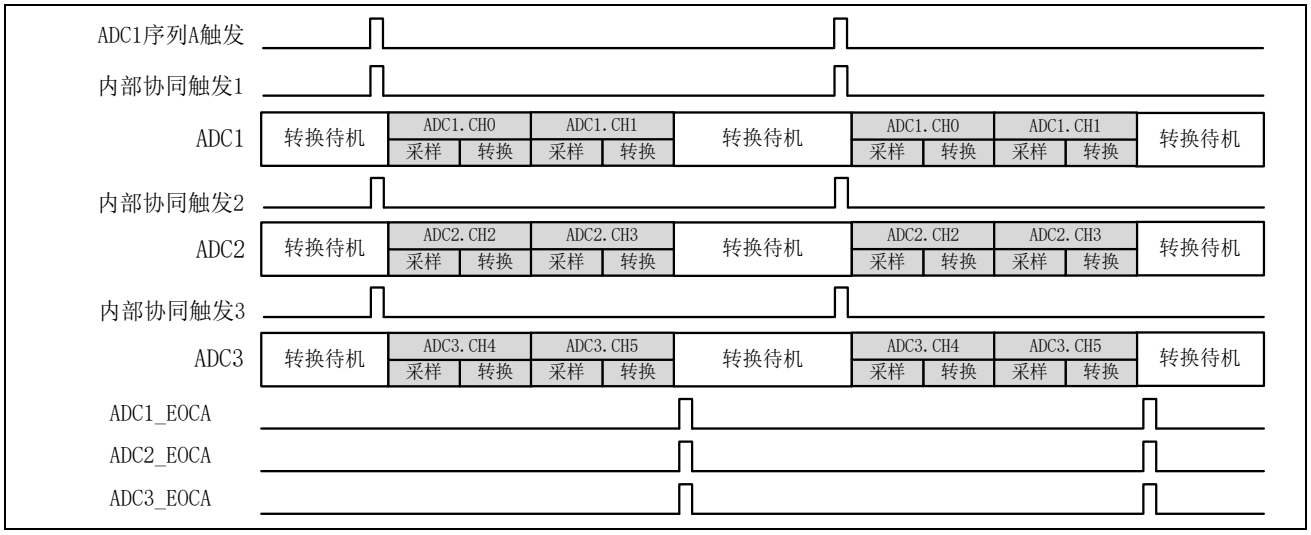


图 17-13 单次并行触发模式（三 ADC）

注意：

– 禁止多个 ADC 同时对同一个模拟输入进行转换，一个模拟通道同一时间只能给一个 ADC 模块采样，否则精度不做保证，下同。

本模式的软件设定流程如下：

1. 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 0，确认协同工作无效。
2. 设置 ADC1 模块
   a) 确认 ADC1_STR.STRT 为 0，ADC1 处于转换待机状态。
   b) 设置控制寄存器 ADC1_CR0.MS[1:0]为 00b：序列 A 单次扫描模式，或 01b：序列 A 连续扫描模式
   c) 设置序列 A 通道选择寄存器 ADC1_CHSELRA
   d) 设置采样时间寄存器 ADC1_SSTR
   e) 设置序列 A 触发源选择寄存器 ADC1_TRGSR
3. 设置 ADC2 模块
   a) 确认 ADC2_STR.STRT 为 0，ADC2 处于转换待机状态。
   b) 设置控制寄存器 ADC2_CR0.MS[1:0]，通道选择寄存器 ADC2_CHSELRA，通道采样时间寄存器 ADC2_SSTR。

      注意：

      – 为保证 ADC2 与 ADC1 的同步工作，上述寄存器尽量与 ADC1 的寄存器设置相同的值。具体通道无需相同，只要保持通道数，以及对应通道的采样时间一致即可。
4. 设置 ADC3 模块（三 ADC 协同工作时）
   a) 确认 ADC3_STR.STRT 为 0，ADC3 处于转换待机状态。
   b) 设置控制寄存器 ADC3_CR0.MS[1:0]，通道选择寄存器 ADC3_CHSELRA，通道采样时间寄存器 ADC3_SSTR。

      注意：

      – 同 ADC2，为保证 ADC3 与 ADC1 的同步工作，上述寄存器尽量与 ADC1 的寄存器设置相同的值。
5. 设置协同模式控制寄存器 ADC_SYNCCR.SYNCMD[2:0]，写 010b：ADC1，ADC2 两 ADC 协同工作。或者写 011b：ADC1，ADC2，ADC3 三 ADC 协同工作。

6. 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 1，协同工作有效。
7. 等待 ADC1 序列 A 触发源输入，在 ADC1，2，3 完成转换后处理结果。

单次延迟触发模式

ADC1 的序列 A 触发条件触发 ADC1 之后，经过设定的延迟后触发 ADC2 启动转换，再经过设定的延迟后触发 ADC3 启动转换，每个 ADC 模块只触发一次。

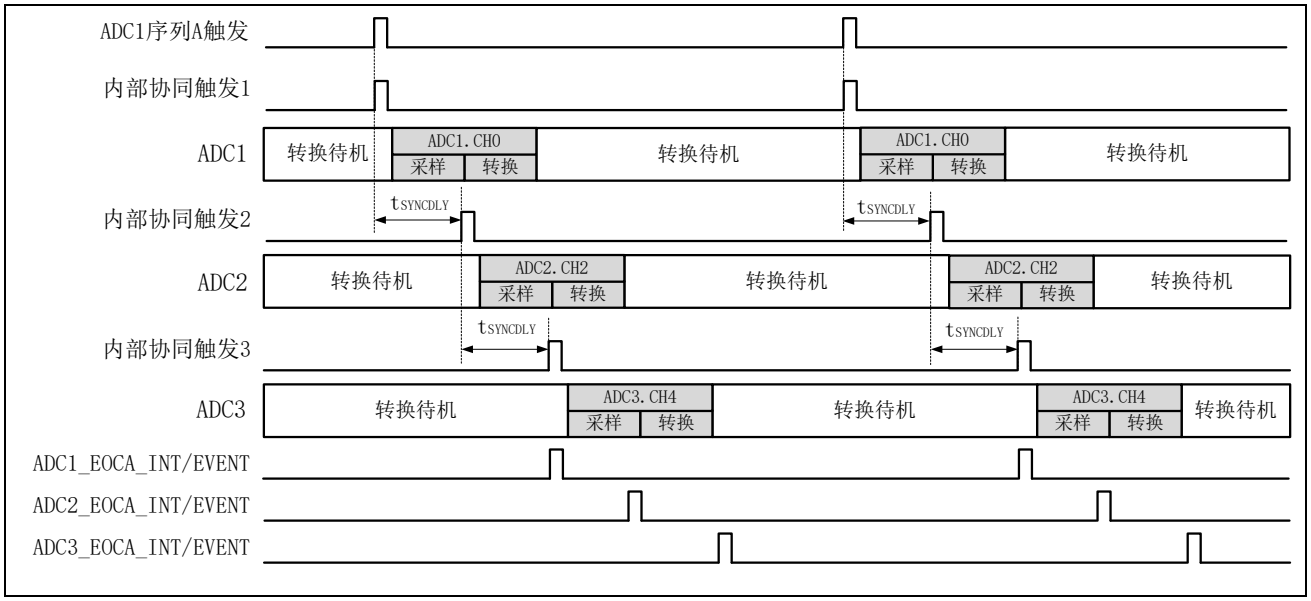


图 17-14 单次延迟触发模式（三 ADC）

注意：

- ADC1 序列 A 触发第一次输入后，ADC3 协同触发发生前，再次输入 ADC1 序列 A 触发将被忽略。
- 若各 ADC 单元转换的是同一模拟通道，需要错开采样时间，即延迟时间 $t_{SYNCDLY}$ 与通道开采样时间 t_{SPL} 需满足：$t_{SYNCDLY}>t_{SPL}$。

本模式的软件设定流程如下：

1. 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 0，确认协同工作无效。
2. 设置 ADC1，2，3 模块（参考单次平行模式）
3. 设置协同模式控制寄存器 ADC_SYNCCR.SYNCDLY[7:0]，设置两个 ADC 的启动延迟。
4. 设置协同模式控制寄存器 ADC_SYNCCR.SYNCMD[2:0]，写 000b：ADC1，ADC2 两 ADC 协同工作。或者写 001b：ADC1，ADC2，ADC3 三 ADC 协同工作。

5. 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 1，协同工作有效。
6. 等待 ADC1 序列 A 触发源输入，在 ADC1，2，3 完成转换后处理结果。

循环并行触发模式

ADC1 的序列 A 触发条件同时触发处于协同工作模式的所有 ADC 模块，且之后每经过指定延迟之后会再次同时触发所有 ADC 模块。直至用户主动软件停止 ADC1 模块，或者禁止协同工作模式。

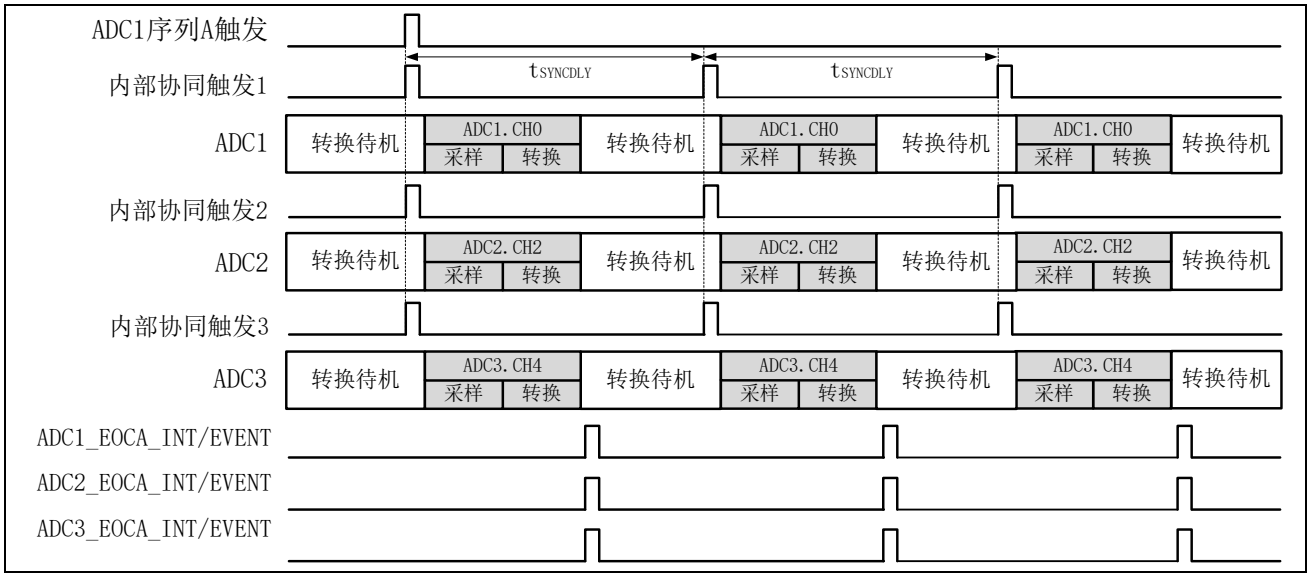


图 17-15 循环并行触发模式（三 ADC）

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 ADC1，2，3 模块，参考单次并行模式。ADC_CR0.MS[1:0]设置为 00b：序列 A 单次扫描模式
3. 设置协同模式控制寄存器 ADC_SYNCCR.SYNCDLY[7:0]，设置每次并行触发的延迟。
4. 设置协同模式控制寄存器 ADC_SYNCCR.SYNCMD[2:0]，写 110b：ADC1，ADC2 两 ADC 协同工作。或者写 111b： ADC1，ADC2，ADC3 三 ADC 协同工作。
5. 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 1，协同工作有效。
6. 等待 ADC1 序列 A 触发源输入，在 ADC1，2，3 完成转换后处理结果。

循环延迟触发模式

ADC1 的序列 A 触发条件触发 ADC1 之后，每经过设定的延迟后，依次循环不断触发 ADC2，ADC3，ADC1，ADC2...，直至用户主动软件停止 ADC1 模块，或者禁止协同工作模式。

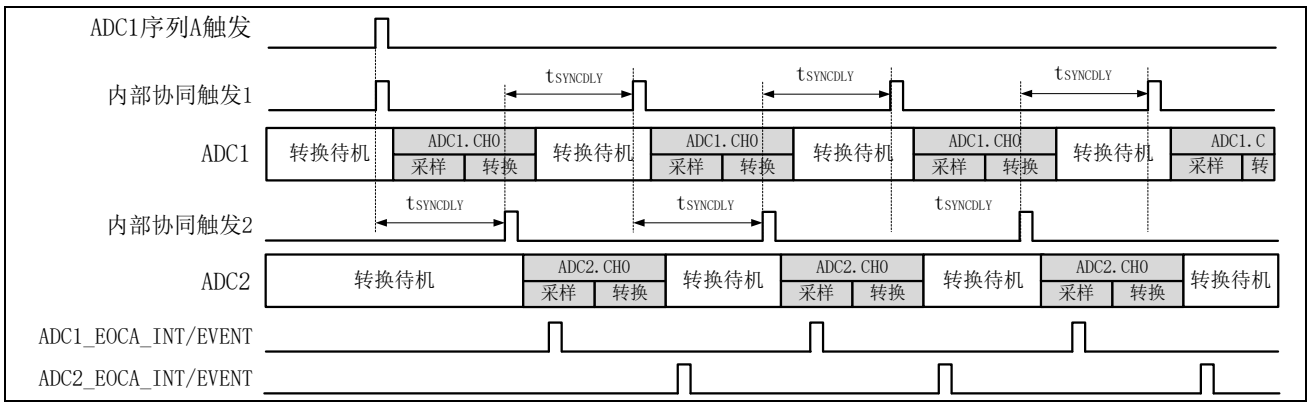


图 17-16 循环延迟触发模式（两 ADC）

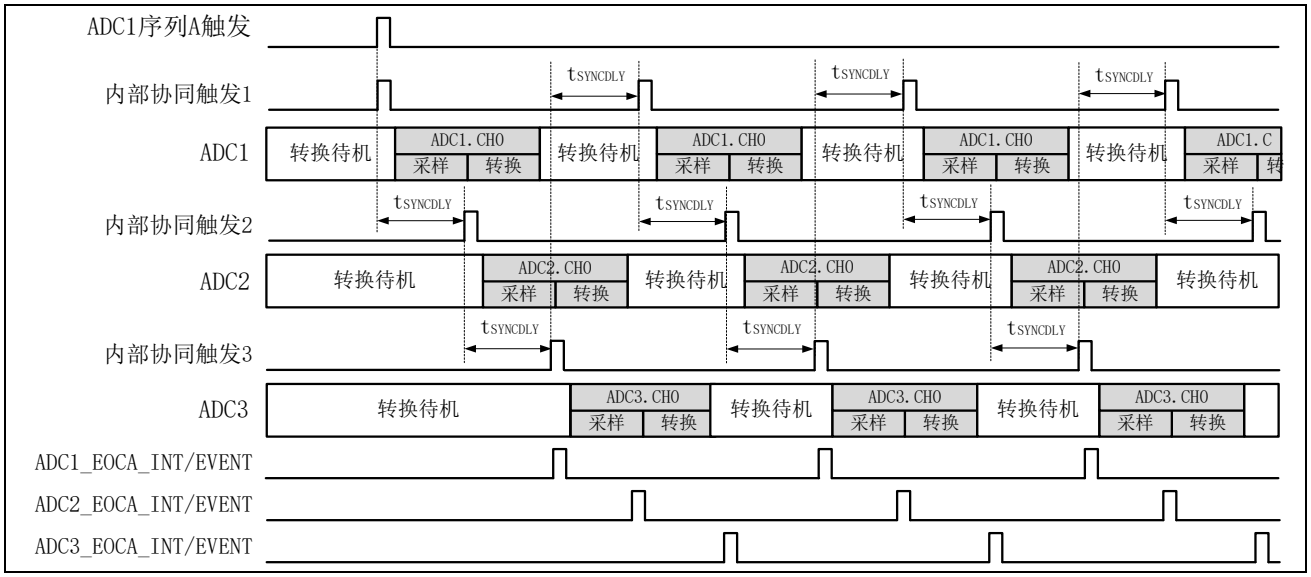


图 17-17 循环延迟触发模式（三 ADC）

注意：

– 两 ADC 协同工作时，延迟时间 $t_{SYNCDLY}$ 与一次扫描转换的时间 t_{SCAN} 需满足：$t_{SYNCDLY}>t_{SCAN}/2$。三 ADC 协同工作时需满足：$t_{SYNCDLY}>t_{SCAN}/3$。同时，如 ADC1，ADC2，ADC3 转换的是同一模拟通道，还需要错开采样时间，即 $t_{SYNCDLY}>t_{SPL}$。

本模式的软件设定流程如下：

1. 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 0，确认协同工作无效。
2. 设置 ADC1，2，3 模块，参考循环并行触发模式。

3. 设置协同模式控制寄存器 ADC_SYNCCR.SYNCDLY[7:0]，设置每次触发的延迟。
4. 设置协同模式控制寄存器 ADC_SYNCCR.SYNCMD[2:0]，写 100b：ADC1，ADC2 两 ADC 协同工作。或者写 101b：ADC1，ADC2，ADC3 三 ADC 协同工作。
5. 协同工作许可寄存器 ADC_SYNCCR.SYNCEN 写 1，协同工作有效。
6. 等待 ADC1 序列 A 触发源输入，在 ADC1，2，3 完成转换后处理结果。

17.3.14 中断和事件信号输出

ADC 模块可以产生以下四种事件输出，每个事件发生时，若对应的中断许可寄存器设置为有效时，同时输出中断申请。

1. 序列 A 扫描结束 ADC_EOCA，对应中断许可寄存器 ADC_ICR.EOCAIEN
2. 序列 B 扫描结束 ADC_EOCB，对应中断许可寄存器 ADC_ICR.EOCBIEN
3. 模拟看门狗 0 ADC_CMP0，对应中断许可寄存器 ADC_AWDCR.AWD0IEN
4. 模拟看门狗 1 ADC_CMP1，对应中断许可寄存器 ADC_AWDCR.AWD1IEN

上述四种事件输出，可以启动其他片内周边模块，包括启动 DMA 传送。利用 DMA 传送可以连续读取 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。

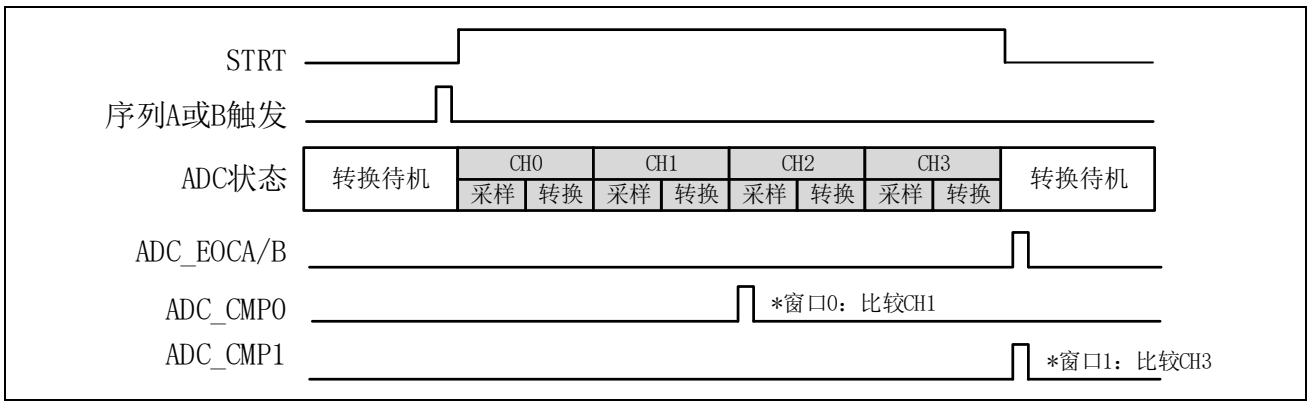


图 17-18 ADC 中断和事件输出时序

17.4 寄存器说明

17.4.1 寄存器一览

单元1 BASE_ADDR: 0x40040000
单元2 BASE_ADDR: 0x40040400
单元3 BASE_ADDR: 0x40040800

寄存器名	符号	偏移地址	位宽	复位值
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	0x00000000
A/D通道选择寄存器A	ADC_CHSELRA	0x0c	32	0x00000000
A/D通道选择寄存器B	ADC_CHSELRB	0x10	32	0x00000000
A/D平均通道选择寄存器	ADC_AVCHSELR	0x14	32	0x00000000
A/D扩展通道选择寄存器	ADC_EXCHSELR	0x18	8	0x00
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	ADC_CHMUXR0	0x38	16	0x3210
A/D通道映射控制寄存器1	ADC_CHMUXR1	0x3a	16	0x7654
A/D通道映射控制寄存器2	ADC_CHMUXR2	0x3c	16	0xba98
A/D通道映射控制寄存器3	ADC_CHMUXR3	0x3e	16	0xfedc
A/D中断状态寄存器	ADC_ISR	0x44	8	0x00
A/D中断许可寄存器	ADC_ICR	0x45	8	0x03
A/D中断状态复位寄存器	ADC_ISCLRR	0x46	8	0x00
A/D协同模式控制寄存器	ADC_SYNCCR	0x4c	16	0x0c00
A/D数据寄存器	ADC_DRy	0x50+2*y	16	0x0000
模拟看门狗控制寄存器	ADC_AWDCR	0xa0	16	0x0000
模拟看门狗状态寄存器	ADC_AWDSR	0xa2	8	0x00
模拟看门狗状态复位寄存器	ADC_AWDSCLRR	0xa3	8	0x00
模拟看门狗窗口0阀值寄存器	ADC_AWD0DR0	0xa4	16	0x0000
	ADC_AWD0DR1	0xa6	16	0xffff
模拟看门狗窗口0比较通道选择寄存器	ADC_AWD0CHSR	0xa8	8	0x00
模拟看门狗窗口1阀值寄存器	ADC_AWD1DR0	0xac	16	0x0000
	ADC_AWD1DR1	0xae	16	0xffff
模拟看门狗窗口1比较通道选择寄存器	ADC_AWD1CHSR	0xb0	8	0x00
采样保持电路控制寄存器	ADC_SHCR	0x1a	16	0x0018
可编程增益放大器控制寄存器1	ADC_PGACR1	0xc0	8	0x00
可编程增益放大器控制寄存器2	ADC_PGACR2	0xc1	8	0x00
可编程增益放大器控制寄存器3	ADC_PGACR3	0xc2	8	0x00
可编程增益放大器参考地选择寄存器	ADC_PGAVSSENR	0xc4	8	0x00

表 17-4 ADC 寄存器一览 1/2

寄存器名	符号	地址	位宽	复位值
ADC1启动片内触发源0选择寄存器	ADC1_ITRGSELR0	0x4001088c	32	0x000001ff
ADC1启动片内触发源1选择寄存器	ADC1_ITRGSELR1	0x40010890	32	0x000001ff
ADC2启动片内触发源0选择寄存器	ADC2_ITRGSELR0	0x40010894	32	0x000001ff
ADC2启动片内触发源1选择寄存器	ADC2_ITRGSELR1	0x40010898	32	0x000001ff
ADC3启动片内触发源0选择寄存器	ADC3_ITRGSELR0	0x4001089c	32	0x000001ff
ADC3启动片内触发源1选择寄存器	ADC3_ITRGSELR1	0x400108a0	32	0x000001ff

表 17-5 ADC 寄存器一览 2/2

17.4.2 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	Reserved	—	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

17.4.3 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-b12	Reserved	—	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	Reserved	—	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.STRT 为“0”时设置本寄存器。

17.4.4 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	Reserved	—	读出时为0，写入时写0	R/W
b2	RSCHSEL	序列A重启通道选择	0：被序列B中断后，序列A重启时从被中断通道开始继续扫描 1：被序列B中断后，序列A重启时从被第一个通道开始重新扫描	R/W
b1-b0	Reserved	—	读出时为0，写入时写0	R/W

注意：

– 请在 ADC_STR.STRT 为“0”时设置本寄存器。

17.4.5 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[1]	TRGSELB[0]
b7	b6	b5	b4	b3	b2	b1	b0
TRGENA	—	—	—	—	—	TRGSELA[1]	TRGSELA[0]

位	标记	位名	功能	R/W
b15	TRGENB	序列B触发使能	0：序列B片内或外部引脚触发禁止 1：序列B片内或外部引脚触发许可 注意：选择外部引脚触发有效。如果ADTRGx由”High”变为“Low”，检测到下降沿，则扫描转换开始，请保持“Low”1.5*PCLK4周期以上。	R/W
b14-b10	Reserved	—	读出时为0，写入时写0	R/W
b9-b8	TRGSELB[1:0]	序列B触发条件选择	在序列B有效模式下(ADC_CR0.MS[1]=1)，作为序列B的触发条件 00b：ADTRGx 01b：IN_TRGx0 10b：IN_TRGx1 11b：IN_TRGx0 + IN_TRGx1 注意：只在序列B有效模式下有效。其他模式设定无效。 两次触发的间隔必须大于或等于扫描周期t_{SCAN}，若小于则触发无效。	R/W
b7	TRGENA	序列A触发使能	0：序列A片内或外部引脚触发禁止 1：序列A片内或外部引脚触发许可 注意：选择外部引脚触发有效。如果ADTRGx由”High”变为“Low”，检测到下降沿，则扫描转换开始，请保持“Low”1.5*PCLK4周期以上。	R/W
b6-b2	Reserved	—	读出时为0，写入时写0	R/W
b1-b0	TRGSELA[1:0]	序列A触发条件选择	序列A的触发条件。 00b：ADTRGx (x=1~3,代表ADC单元编号) 01b：IN_TRGx0 10b：IN_TRGx1 11b：IN_TRGx0 + IN_TRGx1 注意： ADC空闲中对ADC_STR.STRT写1软件触发，无视TRGENA, TRGSELA[1:0]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t_{SCAN}，若小于则触发无效。	R/W

注意：

– 在 ADC_STR.STRT 为“0”时设置本寄存器。

17.4.6 A/D 转换开始片内触发源选择寄存器 ADC_ITRGSELR0，ADC_ITRGSELR1

复位值：0x000001ff

b31	b30	b29	b28	b27	b26	b25	b24
COMEN[1]	COMEN[0]	—	—	—	—	—	—
b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—	—	—
b15	b14	b13	b12	b11	b10	b9	b8
—	—	—	—	—	—	—	TRGSEL[8]
b7	b6	b5	b4	b3	b2	b1	b0
TRGSEL[7]	TRGSEL[6]	TRGSEL[5]	TRGSEL[4]	TRGSEL[3]	TRGSEL[2]	TRGSEL[1]	TRGSEL[0]

位	标记	位名	功能	R/W
b31	COMEN[1]	公共触发使能	0：禁止内部触发IN_TRGx0,1中包含寄存器AOS_COMTRG1选择的公共触发事件 1：允许内部触发IN_TRGx0,1中包含寄存器AOS_COMTRG1选择的公共触发事件 详见【自动运行系统（AOS】章节。	R/W
b30	COMEN[0]	公共触发使能	0：禁止内部触发IN_TRGx0,1中包含寄存器AOS_COMTRG0选择的公共触发事件 1：允许内部触发IN_TRGx0,1中包含寄存器AOS_COMTRG0选择的公共触发事件 详见【自动运行系统（AOS】章节。	R/W
b29-b9	Reserved	—	读出时为0，写入时写0	R/W
b8-b0	TRGSEL[8:0]	片内触发源选择	参考【中断控制器（INTC）】，写入选定的事件编号。 根据COMEN[1:0]设置，IN_TRGx0,1最多可包含TRGSEL[8:0]，AOSCOMTRG0，AOSCOMTRG1所选择的3个触发源。	R/W

注意：

- 触发源 IN_TRGx0 由寄存器 ADCx_ITRGSEL0 设置
- 触发源 IN_TRGx1 由寄存器 ADCx_ITRGSEL1 设置

17.4.7 A/D 通道选择寄存器 A ADC_CHSELRA

复位值：0x00000000

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

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

注意：

– 请在 ADC_STR.STRT 为“0”时设置本寄存器。

17.4.8 A/D 通道选择寄存器 B ADC_CHSELRB

复位值：0x00000000

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

位	标记	位名	功能	R/W
b31-b0	CHSELB[31:0]	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：请不要在序列A和序列B中选择相同的通道。	R/W

注意：

– 请在 ADC_STR.STRT 为“0”时设置本寄存器。

17.4.9 A/D 平均通道选择寄存器 ADC_AVCHSELR

复位值：0x00000000

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

位	标记	位名	功能	R/W
b31-b0	AVCHSEL[31:0]	平均通道选择	每一位代表一个通道，AVCHSEL[x]代表通道CHx，可选择任意组合。 0：未选择对应通道 1：选择对应通道 不存在通道的对应位为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。

17.4.10 A/D 扩展通道选择寄存器 ADC_EXCHSELR

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	EXCHSEL

位	标记	位名	功能	R/W
b7-b1	—	—	读出时为0，写入时写0	R/W
b0	EXCHSEL	扩展通道选择	ADCx_IN15与片内模拟通道的选择 0：选择外部引脚ADCx_IN15 1：选择内部模拟通道，内部模拟源的设置请参考【电源控制（PWC）】说明	R/W

注意：

– 请在 ADC_STR.STRT 为“0”时设置本寄存器。

17.4.11 A/D 采样状态寄存器 ADC_SSTRx

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	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：PCLK2频率为50MHz时，一个采样周期为20ns，初始的转换状态有11个采样周期。当外部输入阻抗R_{AIN}太大采样时间不足或者PCLK2频率低时可以设置寄存器进行调整采样时间。采样时间不要少于5周期。 SST ≧（R_{AIN}+R_{ADC}）*C_{ADC}*ln(2^{N+2})*f_{ADC}+1 其中：R_{AIN}表示外部输入阻抗（Ω），R_{ADC}表示内部采样开关电阻（Ω），C_{ADC}表示内部采样和保持电容（F），N表示AD分辨率（12/10/8），f_{ADC}表示PCLK2频率（Hz）。具体参考电器特性相关说明。	R/W

注意：

- 请在 ADC_STR.STRT 为“0”时设置本寄存器。
- 内部基准电压的采样时间不要小于 1us。
- VBAT 分压的采样时间不要小于 4us。
- 通道专用采样保持电路 SH 有效时，对应通道采样时间不要小于 0.4us

17.4.12 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]			

<table>
<tr><th>位</th><th>标记</th><th>位名</th><th>功能</th><th>R/W</th></tr>
<tr><td></td><td>CHxMUX[3:0]</td><td>通道x映射选择
x=0~15</td><td>不存在的通道对应位读出时为0，写入时写0
对不同的ADC单元CHx的映射关系如下：
<table>
<tr><th>设定值</th><th>ADC1 映射对象</th><th>ADC2 映射对象</th><th>ADC3 映射对象</th></tr>
<tr><td>0x0</td><td>ADC123_IN0</td><td>ADC123_IN0</td><td>ADC123_IN0</td></tr>
<tr><td>0x1</td><td>ADC123_IN1</td><td>ADC123_IN1</td><td>ADC123_IN1</td></tr>
<tr><td>0x2</td><td>ADC123_IN2</td><td>ADC123_IN2</td><td>ADC123_IN2</td></tr>
<tr><td>0x3</td><td>ADC123_IN3</td><td>ADC123_IN3</td><td>ADC123_IN3</td></tr>
<tr><td>0x4</td><td>ADC12_IN4</td><td>ADC12_IN4</td><td>ADC3_IN4</td></tr>
<tr><td>0x5</td><td>ADC12_IN5</td><td>ADC12_IN5</td><td>ADC3_IN5</td></tr>
<tr><td>0x6</td><td>ADC12_IN6</td><td>ADC12_IN6</td><td>ADC3_IN6</td></tr>
<tr><td>0x7</td><td>ADC12_IN7</td><td>ADC12_IN7</td><td>ADC3_IN7</td></tr>
<tr><td>0x8</td><td>ADC12_IN8</td><td>ADC12_IN8</td><td>ADC3_IN8</td></tr>
<tr><td>0x9</td><td>ADC12_IN9</td><td>ADC12_IN9</td><td>ADC3_IN9</td></tr>
<tr><td>0xa</td><td>ADC123_IN10</td><td>ADC123_IN10</td><td>ADC123_IN10</td></tr>
<tr><td>0xb</td><td>ADC123_IN11</td><td>ADC123_IN11</td><td>ADC123_IN11</td></tr>
<tr><td>0xc</td><td>ADC123_IN12</td><td>ADC123_IN12</td><td>ADC123_IN12</td></tr>
<tr><td>0xd</td><td>ADC123_IN13</td><td>ADC123_IN13</td><td>ADC123_IN13</td></tr>
<tr><td>0xe</td><td>ADC12_IN14</td><td>ADC12_IN14</td><td>ADC3_IN14</td></tr>
<tr><td>0xf</td><td>ADC12_IN15</td><td>ADC12_IN15</td><td>ADC3_IN15</td></tr>
</table>
注意：请不要设置到不存在的模拟输入上。</td><td>R/W</td></tr>
</table>

注意：

– 请在 ADC_STR.STRT 为“0”时设置本寄存器。

17.4.13 A/D 中断状态寄存器 ADC_ISR

复位值：0x00

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

位	标记	位名	功能	R/W
b7-b5	Reserved	—	读出时为0，写入时写0	R/W
b4	SASTPDF	序列A被打断标志	双序列扫描模式，序列A扫描过程中被高优先级的序列B打断时置1 本寄存器位为只读位	R
b3-b2	Reserved	—	读出时为0，写入时写0	R/W
b1	EOCBF	序列B转换完成标志	序列B所选通道全部扫描完成后置1 本寄存器位为只读位	R/W
b0	EOCAF	序列A转换完成标志	序列A所选通道全部扫描完成后置1 本寄存器位为只读位	R/W

17.4.14 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	Reserved	—	读出时为0，写入时写0	R/W
b1	EOCBIEN	序列B转换完成中断使能	0：序列B转换完成中断禁止 1：序列B转换完成中断许可	R/W
b0	EOCAIEN	序列A转换完成中断使能	0：序列A转换完成中断禁止 1：序列A转换完成中断许可	R/W

17.4.15 A/D 中断状态复位寄存器 ADC_ISCLRR

复位值：0x00

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

位	标记	位名	功能	R/W
b7-b5	Reserved	—	读出时为0，写入时写0	R/W
b4	CLRSASTPDF	序列A被打断标志复位	写0无任何效果，写1复位SASTPDF状态位， 读出永远为0	R/W
b3-b2	Reserved	—	读出时为0，写入时写0	R/W
b1	CLREOCBF	序列B转换完成标志复位	写 0 无任何效果，写 1 复位 EOCBF 状态位， 读出永远为0	R/W
b0	CLREOCAF	序列A转换完成标志复位	写 0 无任何效果，写 1 复位 EOCAF 状态位， 读出永远为0。	R/W

17.4.16 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*PCLK2，0xff表示$t_{SYNCDLY}$ = 255*PCLK2 注意：在SYNCEN为“0”时设置本寄存器。请不要写入0x00。 根据各ADC的采样时间，转换时间，设置合理的延迟时间，避免多个ADC同时处于采样状态引起的误差增大，避免ADC还未转换结束就再次发生触发，导致同步失败。推荐设置如下： 单次延迟触发模式： $t_{SYNCDLY} > t_{SPL}$ 两ADC循环延迟触发模式： $t_{SYNCDLY} > t_{SPL}$，且$t_{SYNCDLY} > t_{SCAN}/2$ 三ADC循环延迟触发模式： $t_{SYNCDLY} > t_{SPL}$，且$t_{SYNCDLY} > t_{SCAN}/3$ 单次并行触发模式：本寄存器设置无效。 循环并行触发模式： $t_{SYNCDLY} > t_{SCAN}$	R/W
b7	Reserved	—	读出时为0，写入时写0	R/W
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 注意：在SYNCEN为“0”时设置本寄存器。使用单次触发时，请将需要同步的ADC设置为序列A单次扫描，或序列A连续扫描模式。使用循环触发模式时，请将ADC设置为序列A单次扫描模式。	R/W
b3-b1	Reserved	—	读出时为0，写入时写0	R/W
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.STRT写0强制停止转换时，SYNCEN自动清0。	R/W

注意：

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

17.4.17 A/D 数据寄存器 ADC_DR

ADC_DRx （ADC1 x=0~15，ADC2 x=0~15，ADC3 x=0~19） 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]												0	0	0	0

数据左对齐-10 位分辨率

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AD[9:0]										0	0	0	0	0	0

数据左对齐-8 位分辨率

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

17.4.18 模拟看门狗控制寄存器 ADC_AWDCR

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8
—	—	—	—	—	—	AWDCD[1]	AWDCM[0]

b7	b6	b5	b4	b3	b2	b1	b0
—	AWD1MD	AWD1IEN	AWD1EN	—	AWD0MD	AWD0IEN	AWD0EN

位	标记	位名	功能	R/W
b15-b10	Reserved	—	读出时为0，写入时写0	R/W
b9-b8	AWDCM[1:0]	看门狗窗口组合选择	00：窗口组合无效，ADC_CMP1输出窗口1独立比较结果 01：窗口组合有效，ADC_CMP1输出窗口0与窗口1比较结果的逻辑或 10：窗口组合有效，ADC_CMP1输出窗口0与窗口1比较结果的逻辑与 11：窗口组合有效，ADC_CMP1输出窗口0与窗口1比较结果的逻辑异或 注意：使用窗口组合功能时需要窗口0和窗口1都使能，即AWD0EN写1，AWD1EN写1。 若窗口0与窗口1选择的比较通道不相同，请保证扫描转换过程中，窗口1所选通道在窗口0所选通道之后转换。ADC_CMP1中断或事件在窗口1所选通道转换结束时输出。	R/W
b7	Reserved	—	读出时为0，写入时写0	R/W
b6	AWD1MD	看门狗窗口1比较模式	窗口1保护区域选择 0：转换结果<AWD1DR0，或转换结果>AWD1DR1 1：转换结果≧AWD1DR0，且转换结果≦AWD1DR1。	R/W
b5	AWD1IEN	看门狗窗口1中断使能	0：看门狗窗口1比较中断ADC_CMP1无效 1：看门狗窗口1比较中断ADC_CMP1有效	R/W
b4	AWD1EN	看门狗窗口1比较功能使能	0：看门狗窗口1比较功能无效 1：看门狗窗口1比较功能有效	R/W
b3	Reserved	—	读出时为0，写入时写0	R/W
b2	AWD0MD	看门狗窗口0比较模式	窗口0保护区域选择 0：转换结果<AWD0DR0，或转换结果>AWD0DR1 1：转换结果≧AWD0DR0，且转换结果≦AWD0DR1。	R/W
b1	AWD0IEN	看门狗窗口0中断使能	0：看门狗窗口0比较中断ADC_CMP0无效 1：看门狗窗口0比较中断ADC_CMP0有效	R/W
b0	AWD0EN	看门狗窗口0比较功能使能	0：看门狗窗口0比较功能无效 1：看门狗窗口0比较功能有效	R/W

17.4.19 模拟看门狗状态寄存器 ADC_AWDSR

复位值：0x00

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

位	标记	位名	功能	R/W
b7-b5	Reserved	—	读出时为0，写入时写0	R/W
b4	AWDCMF	看门狗窗口组合比较状态位	看门狗窗口组合比较功能有效（即AWDCM[1:0]=01b/10b/11b）时，窗口1所选通道转换结束时，窗口0与窗口1比较结果满足组合条件时置1。 对本寄存器位写无效。	R
b3-b2	Reserved	—	读出时为0，写入时写0	R/W
b1	AWD1F	看门狗窗口1比较状态位	窗口1所选通道转换结束时，转换结果满足比较条件时置1。 对本寄存器位写无效。	R
b0	AWD0F	看门狗窗口0比较状态位	窗口0所选通道转换结束时，转换结果满足比较条件时置1。 对本寄存器位写无效。	R

17.4.20 模拟看门狗状态复位寄存器 ADC_AWDSCLRR

复位值：0x00

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

位	标记	位名	功能	R/W
b7-b5	Reserved	—	读出时为0，写入时写0	R/W
b4	CLRAWDCMF	看门狗窗口组合比较状态位复位	写0无任何效果，写1复位AWDCMF状态位， 读出永远为0	R/W
b3-b2	Reserved	—	读出时为0，写入时写0	R/W
b1	CLRAWD1F	看门狗窗口1比较状态复位	写0无任何效果，写1复位AWD1F状态位， 读出永远为0	R/W
b0	CLRAWD0F	看门狗窗口0比较状态复位	写0无任何效果，写1复位AWD1F状态位， 读出永远为0	R/W

17.4.21 模拟看门狗阀值寄存器 ADC_AWD0DR0、ADC_AWD0DR1、ADC_AWD1DR0、ADC_AWD1DR1

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

ADC_AWD1DR0=0x0000, ADC_AWD1DR1=0xffff

b15	b14	b13	b12	b11	b10	b9	b8
AWDDR[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
AWDDR[7:0]							

位	标记	位名	功能	R/W
b15-b0	AWDDR[15:0]	比较数据	比较数据	R/W

AWD0DR0 设置窗口 0 低阀置，AWD0DR1 设置窗口 0 高阀值。

AWD1DR0 设置窗口 1 低阀置，AWD1DR1 设置窗口 1 高阀值。

AWD0DR0、AWD0DR1、AWD1DR0、AWD1DR1 根据对齐方式（数据右对齐或左对齐）,分辨率（12 位、10 位或者 8 位）会有差异。

- 数据右对齐-12 位分辨率　低 12 位[11:0]可用
- 数据右对齐-10 位分辨率　低 10 位[9:0]可用
- 数据右对齐-8 位分辨率　低 8 位[7:0]可用
- 数据左对齐-12 位分辨率　高 12 位[15:4]可用
- 数据左对齐-10 位分辨率　高 10 位[15:6]可用
- 数据左对齐-8 位分辨率　高 8 位[15:8]可用

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

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	AWDCH[4]	AWDCH[3]	AWDCH[2]	AWDCH[1]	AWDCH[0]

位	标记	位名	功能	R/W
b7-b5	Reserved	—	读出时为0，写入时写0	R/W
b4-b0	AWDCH[4:0]	看门狗比较通道选择	ADC_AWD0CHSR选择窗口0的比较通道，ADC_AWD1CHSR选择窗口1的比较通道。 0x00：CH0 0x01：CH1 以此类推 注意：请不要设置成不存在的通道	R/W

17.4.23 专用采样保持电路控制寄存器 ADC_SHCR

复位值：0x0018

b15	b14	b13	b12	b11	b10	b9	b8
—	—	—	—	—	SHSEL[2]	SHSEL[1]	SHSEL[0]
b7	b6	b5	b4	b3	b2	b1	b0
SHSST[7]	SHSST[6]	SHSST[5]	SHSST[4]	SHSST[3]	SHSST[2]	SHSST[1]	SHSST[0]

位	标记	位名	功能	R/W
b15-b11	Reserved	—	读出时为0，写入时写0	R/W
b10-b8	SHSEL[2:0]	专用采样保持电路通道选择	0：旁路专用采样保持电路 1：选通专用采样保持电路 每一位对应一个通道。 注意：在双序列扫描模式下，序列A中不要使用专用采样保持电路。因为序列A可能会被序列B打断导致各专用采样保持电路的采样时间不一致而输出期待之外的转换结果。	
b7-b0	SHSST[7:0]	采样时间选择	采样时间可选择为4到255周期。 注意：PCLK2为50MHz工作时，1周期时间为20ns。初始的周期设定为24周期，当输入阻抗高采样时间不足，或者PCLK2频率低的状况下可以调整采样时间。采样时间不要少于0.4us。	R/W

注意：

- 单元 1（ADC1）支持 SH1~3 共 3 个通道，参考表 17-1。寄存器 ADC_SHCR.SHSEL[0]对应 SH1，SHSEL[1]对应 SH2，SHSEL[2]对应 SH3。
- 单元 2（ADC2）不支持 SH，无此寄存器。
- 单元 3（ADC3）不支持 SH，无此寄存器。

17.4.24 可编程增益放大器控制寄存器 ADC_PGACRx

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
PGAGAIN[3:0]				PGACTL[3:0]			

位	标记	位名	功能	R/W
b7-b4	PGAGAIN[3:0]	增益设定	0 0 0 0：× 2.000 0 0 0 1：× 2.133 0 0 1 0：× 2.286 0 0 1 1：× 2.667 0 1 0 0：× 2.909 0 1 0 1：× 3.2 0 1 1 0：× 3.556 0 1 1 1：× 4.000 1 0 0 0：× 4.571 1 0 0 1：× 5.333 1 0 1 0：× 6.4 1 0 1 1：× 8 1 1 0 0：× 10.667 1 1 0 1：× 16 1 1 1 0：× 32 注意：其他值禁止设定	
b3-b0	PGACTL[3:0]	放大器控制	0000：放大器无效 1110：放大器有效，信号按PGAGAIN[3:0]设定值放大 注意：禁止设定上述以外的值。	R/W

注意：

- 单元 1（ADC1）支持 PGA1~3 共 3 个通道，参考表 17-1。寄存器 ADC1_PGACR1~3 分别对应控制 PGA1~3。
- 单元 2（ADC2）支持 PGA4 共 1 个通道，参考表 17-1。寄存器 ADC2_PGACR1 对应控制 PGA4。
- 单元 3（ADC3）不支持 PGA。

17.4.25 可编程增益放大器参考地选择寄存器 ADC_PGAVSSENR

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	PGAVSSEN[2:0]		

位	标记	位名	功能	R/W
b7-b3	Reserved	—	读出时为0，写入时写0	R/W
b2-b0	PGAVSSEN[2:0]	放大器参考地选择	0：使用外部引脚PGAx_VSS作为PGA参考地 1：使用模拟地AVSS作为PGA参考地 其中： 单元1 ADC1_PGAVSSENR.PGAVSSEN[0]选择PGA1参考地PGA123_VSS或AVSS 单元1 ADC1_PGAVSSENR.PGAVSSEN[1]选择PGA2参考地PGA123_VSS或AVSS 单元1 ADC1_PGAVSSENR.PGAVSSEN[2]选择PGA3参考地PGA123_VSS或AVSS 单元2 ADC2_PGAVSSENR.PGAVSSEN[0]选择PGA4参考地PGA4_VSS或AVSS 单元2 ADC2_PGAVSSENR.PGAVSSEN[2:1]不存在。	R/W

注意：

– 单元 1，2（ADC1，2）支持此寄存器，单元 3（ADC3）无此寄存器。

17.5 使用注意事项

17.5.1 数据寄存器读取时注意事项

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。

17.5.2 扫描完成中断处理注意事项

当对同一通道进行连续两次扫描转换时，从第一次转换完成中断处理到第二次转完成中断处理期间，如果 CPU 没有及时将第一次的转换数据读取，那么第二次的转换数据会将第一次的转换数据覆盖。

17.5.3 模块停止与低功耗设定的注意事项

通过设定寄存器 PWR_FCG，可以设定 ADC 模块停止，降低功耗。ADC 初始状态为停止状态。需要 A/D 模块工作时请先设定 PWR_FCG 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.STRT 位为“0”。

在设置系统进入停止模式（STOP）前，请先将 ADC 设定为模块停止模式。

具体请参考低功耗说明章节。

17.5.4 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。

17.5.5 噪声控制

为防止浪涌等异常电压破坏模拟输入引脚，推荐使用**数据手册中电气特性**章节所示保护电路。

18 数模转换器（DAC）

18.1 简介

本 MCU 搭载了两个 12 位的数字-模拟电压转换器单元 DAC1 和 DAC2。每个 DAC 单元包含两个 D/A 转换通道，可以独立转换也可以通过转换数据的同步更新实现同步转换。每个转换通道配有一个输出放大器，可以在没有外部运放时直接驱动外部负载。独立管脚输入参考电压 VREFH 和 VREFL 可用来提高转换精度。

DAC 主要特性：

- 两个 DAC 单元，每个单元有两个 D/A 转换通道
- 12 位转换数据可配置成左对齐或者右对齐格式
- 同一个 DAC 模块的两个转换通道可实现同步转换
- 转换外部数据（来自 DCU）可输出三角波和锯齿波
- 输出放大功能，可直接驱动外部负载
- A/D 转换优先模式可减少对 A/D 转换的干扰
- 输出可供电压比较器作为负端电压
- 独立管脚输入参考电压 VREFH/VREFL

18.2 功能框图

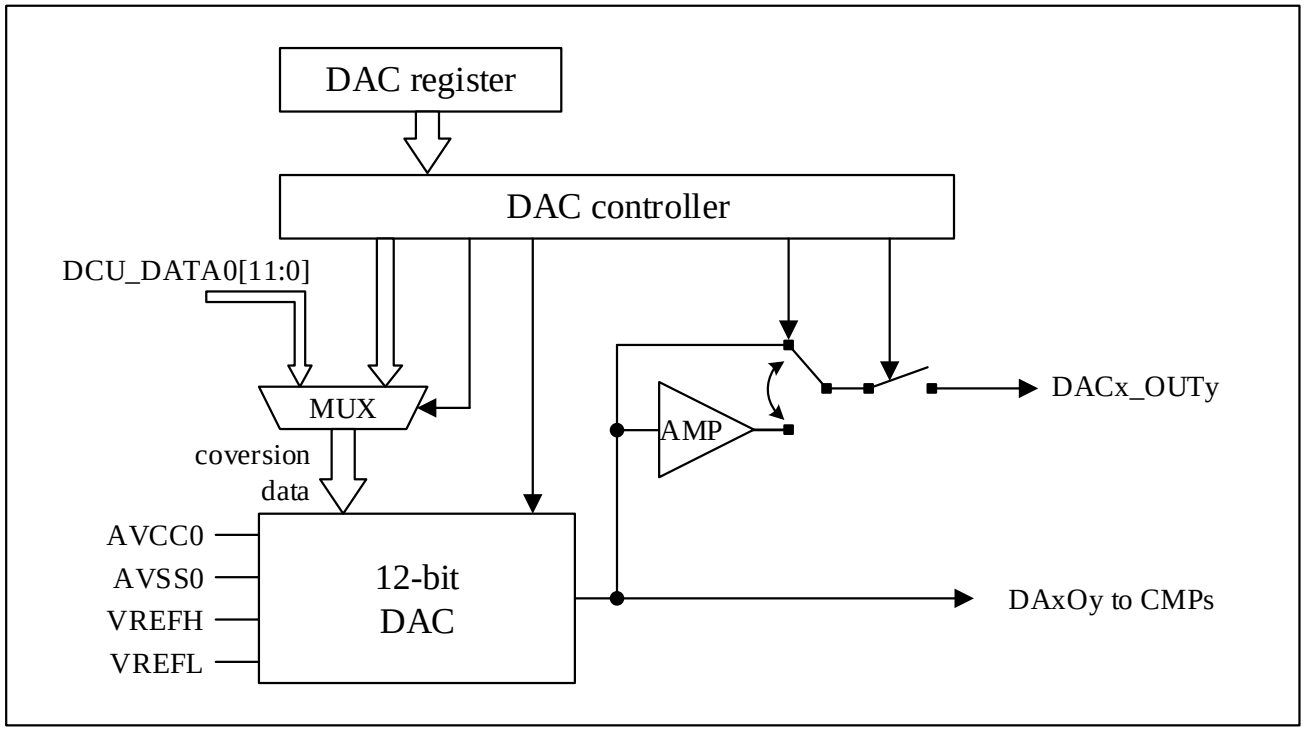


图 18-1 D/A 转换通道框图

引脚名称	输入/输出	功能
AVCC0	输入	模拟电源
AVSS0	输入	模拟地
VREFH	输入	参考电压
VREFL	输入	参考地
DACx_OUTy	输出	D/A转换模拟输出（x,y=1~2）

表 18-1 DAC 引脚

18.3 功能说明

18.3.1 D/A 转换

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

下面是一个使用 DAC1_ch1 进行单通道转换的例子，动作时序见图 18-2。

1. 设置 D/A 转换数据（DADR1_1）和数据格式（DACR1.DPSEL）。
2. 设置 DACR1.DA1E 为 1，D/A 转换开始。经过转换时间 tDCONV 后，转换结果从端口 DAC1_OUT1 输出并一直保持到 DADR 的值被改写或者 DACR1.DA1E 被设置成 0。输出的模拟电压值可由以下公式计算：

$$DACoutput = ConversionData/4096 \times VREFH$$

3. DACR1.DA1E 为 1 时，改写 DADR 的值将触发一次新的 D/A 转换。同样地，经过转换时间 tDCONV 后，从端口 DAC1_OUT1 输出新的转换结果。
4. 设置 DACR1.DA1E 位为 0，关闭 DAC1_ch1，DAC1_OUT1 输出高阻态。

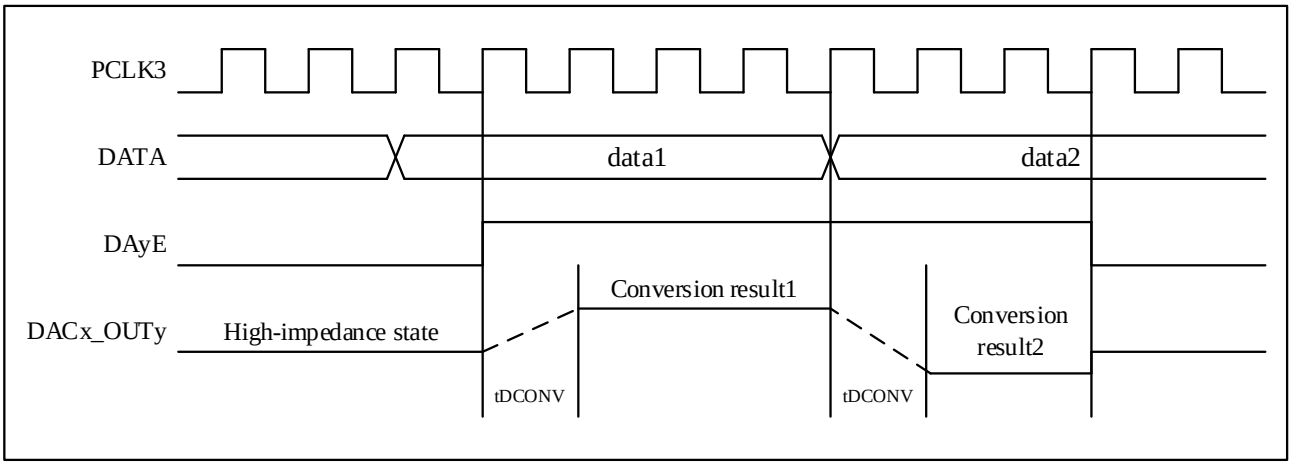


图 18-2 D/A 转换示意图

18.3.2 同步转换

通过转换数据的同步更新可以实现同一个 DAC 单元的两个转换通道的同步转换。

下面是一个使用 DAC1 进行两通道同步转换的例子。

1. 设置 D/A 转换数据（DADR1_1，DADR1_2）和数据格式（DACR.DPSEL）。

2. 设置 DACR1.DAE 为 1，两个通道的 D/A 转换同步开始。经过转换时间 tDCONV 后，转换结果分别从端口 DAC1_OUT1 和 DAC1_OUT2 输出并一直保持到 DADR 的值被改写，或者 DACR1.DAE 被设置成 0。
3. DACR1.DAE 为 1 的状态下，同时改写 DADR1_1 和 DADR1_2 的值将同步触发两个通道开始一次新的 D/A 转换。同样地，经过转换时间 tDCONV 后，从 DAC1_OUT1 和 DAC1_OUT2 输出新的转换结果。
4. 设置 DACR1.DAE 为 0，两个通道同步关闭，DAC1_OUT1 和 DAC1_OUT 输出高阻态。

18.3.3 外部数据转换

DAC 除了可以转换 DADR 中的数据，还可以转换来自数据计算单元（DCU）的数据。DAC1 的通道 1 和通道 2 分别对应 DCU 的单元 1 和单元 2，DAC2 的通道 1 和通道 2 分别对应 DCU 的单元 3 和单元 4。设置 DACRx.EXTDSLy 为 1 即选择转换来自相应 DCU 单元的数据寄存器 0 的低 12 位数据。

转换外部数据时禁止使用输出放大器和 A/D 转换优先模式。

18.3.4 A/D 转换优先模式

DAC 开始 D/A 转换时模拟电源上可能出现短暂的浪涌电流，这会对正在进行中的 A/D 转换产生干扰。A/D 转换优先模式通过改变 D/A 转换数据的更新时序可以有效地避免这种干扰的发生。

设置 DAADPCR.ADPEN 为 1，DAC 进入 A/D 转换优先模式。这时，如果在 A/D 转换期间改写 DADR 的值，D/A 转换并不会立即开始，而是要等到 A/D 转换完成后才开始。也就是说，从改写 DADR 到实际开始 D/A 转换最长需要等待一次 A/D 转换的时间。因此，在此期间 DADR 的值与模拟输出值并不一致。但是，如果改写 DADR 时 ADC 处于停止状态（ADCSR.ADST 为 0）， D/A 转换则会在 2 个 ADCLK 周期后开始。DAADPCR.ADPSLn（n=0~2）用于选择 A/D 转换优先的 ADC 通道。

选择转换外部数据（DACRx.EXTDSLy=1）时，A/D 转换优先模式无效。

下面以 DAC1_ch1 为例，说明 A/D 转换优先模式的设定步骤，动作时序见图 18-3。

1. 确认 ADC 处于停止状态，设置 DACR1.ADPSL。

2. 确认 ADC 处于停止状态，设置 DACR1.DA1E 为 1。
3. 将转换数据写入 DADR1_1。
   - 转换数据 A 被写入 DADR1_1 时，ADCSR.ADST 为 0，ADC 处于停止状态，D/A 转换在 2 个 ADCLK 周期后开始。
   - 转换数据 B 被写入 DADR1_1 时，ADCSR.ADST 为 1，ADC 正在进行 A/D 转换，D/A 转换要到 A/D 转换完成后才会开始。而在 D/A 转换开始前转换数据 C 已被写入 DADR1_1，因此最终被转换的是转换数据 C，而不是转换数据 B。

为避免转换数据丢失，请先查看 D/A 转换状态寄存器（DAADPCRx.DAySF，xy=1~2），确认当前数据转换完成后再改写 DADR 的值。

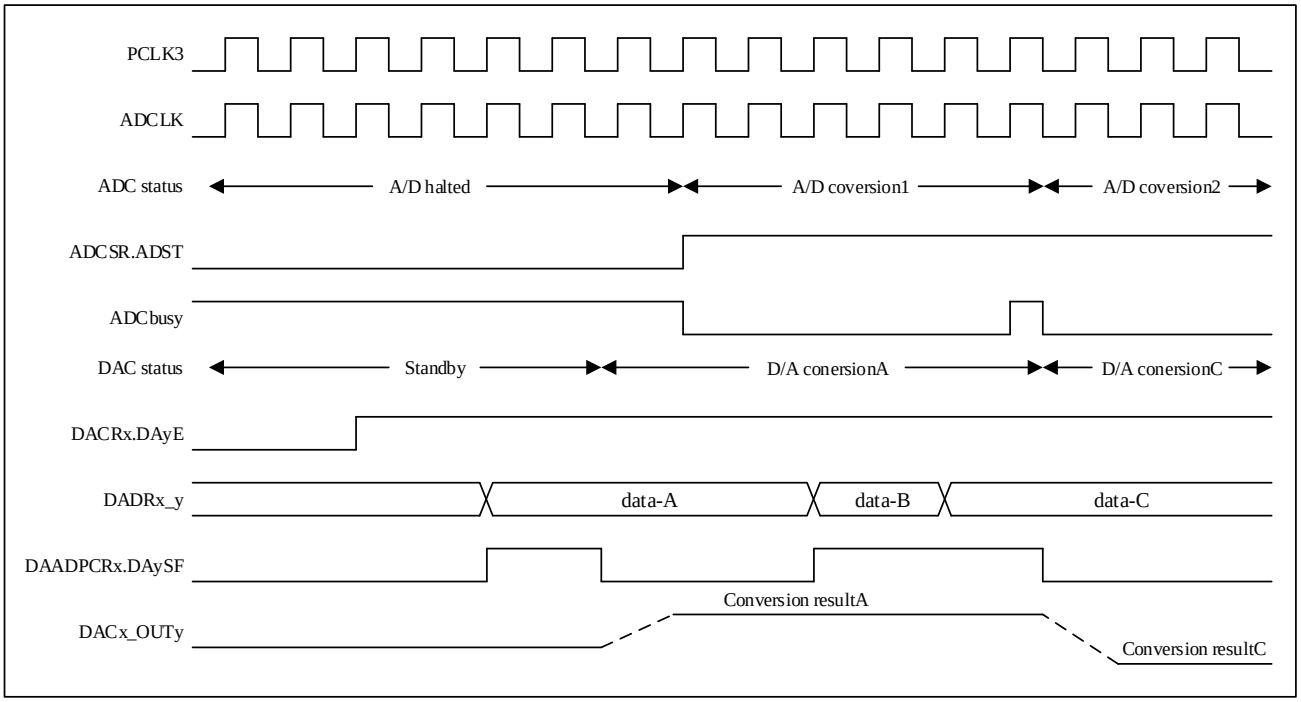


图 18-3 A/D 转换优先模式动作示意图

18.4 注意事项

18.4.1 模块停止功能的设置

可以使用模块停止控制寄存器设定 DAC1，2 有效或者停止。系统复位后 DAC1，2 都是停止的，需要解除模块停止状态后才可以访问寄存器。详见【动作模式与低功耗模式】。

18.4.2 模块停止状态下的 DAC 动作

如果 DAC 是在 D/A 转换时进入模块停止状态的，模拟输出将被保持，这时流过模拟电源的电流与 D/A 转换时相同。如果需要进一步减小模块停止状态下的功耗，请将 DACRx 的 DAyE 和 DAE 位全部设为 0。

18.4.3 停止低功耗模式下的 DAC 动作

如果系统是在 D/A 转换时进入停止低功耗模式的，模拟输出将被保持，这时流过模拟电源的电流与 D/A 转换时相同。如果需要进一步减小停止模式下的功耗，请将 DACRx 的 DAyE 和 DAE 位全部设为 0。

18.4.4 掉电低功耗模式下的 DAC 动作

如果系统是在 D/A 转换时进入掉电低功耗模式的，模拟输出将被置为高阻态。

18.4.5 用输出放大器的注意事项

使用输出放大器时请使用以下的初始化程序。

1. 给 DADRx_y 写全 0。
2. 设置 DACRx.DAAMPy 为 1。
3. 设置 DACRx.DAE 或者 DAyE 为 1。
4. 等待 3us 后将变换数据写入 DADRx_y。

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

18.5 寄存器说明

DAC1 基准地址： 0x40041000

寄存器名	符号	偏移地址	位宽	复位值
DAC1 数据寄存器1	DADR1_1	0x00	16	0x0000
DAC1 数据寄存器2	DADR1_2	0x02	16	0x0000
DAC1 控制寄存器	DACR1	0x04	16	0x0000
DAC1 A/D转换优先控制寄存器	DAADPCR1	0x06	16	0x0000
DAC1 模拟输出控制寄存器	DAOCR1	0x1c	16	0x0000

表 18-2 DAC1 寄存器一览表

DAC2 基准地址： 0x40041400

寄存器名	符号	偏移地址	位宽	复位值
DAC2 数据寄存器1	DADR2_1	0x00	16	0x0000
DAC2 数据寄存器2	DADR2_2	0x02	16	0x0000
DAC2 控制寄存器	DACR2	0x04	16	0x0000
DAC2 A/D转换优先控制寄存器	DAADPCR2	0x06	16	0x0000
DAC2 模拟输出控制寄存器	DAOCR2	0x1c	16	0x0000

表 18-3 DAC2 寄存器一览表

18.5.1 DAC 数据寄存器（DADRx_y x,y=1,2）

DACR.DPSEL=0（数据右对齐）时

复位值：0x0000

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

位	标记	位名	功能	读写
b15~b12	Reserved	-	读时读出0，写时请写0	R/W
b11~b0	DADR[11:0]	变换数据	变换数据设定值	R/W

DACR.DPSEL=1（数据左对齐）时

复位值：0x0000

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

位	标记	位名	功能	读写
b15~b4	DADR[11:0]	变换数据	变换数据设定值	R/W
b3~b0	Reserved	-	读时读出0，写时请写0	R/W

DADR 用于存放 D/A 变换的数据，每个通道对应有一个。只要允许 D/A 转换，DADR 中的值就被转换成模拟电压并从模拟端口输出。对同一单元的 DADR 进行 32 位操作可实现双通道同步转换。

18.5.2 DAC 控制寄存器（DACRx, x=1,2）

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	EXTDSL2	EXTSL1	DAAMP2	DAAMP1	DPSEL	-	-	-	-	-	DA2E	DA1E	DAE

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	EXTDSL2	通道2外部数据选择	0：选择DADR作为通道2的转换数据 1：选择外部数据作为通道2的转换数据	R/W
b11	EXTDSL1	通道1外部数据选择	0：选择DADR作为通道1的转换数据 1：选择外部数据作为通道1的转换数据	R/W
b10	DAAMP2	通道2输出放大器使能	0：禁止通道2输出放大器 1：启动通道2输出放大器	R/W
b9	DAAMP1	通道1输出放大器使能	0：禁止通道1输出放大器 1：启动通道1输出放大器	R/W
b8	DPSEL	数据寄存器格式选择	0：右对齐格式 1：左对齐格式	R/W
b7~b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2	DA2E	通道2使能	DAE=0时此位设定有效 0：通道2禁止 1：通道2使能	R/W
b1	DA1E	通道1使能	DAE=0时此位设定有效 0：通道1禁止 1：通道1使能	R/W
b0	DAE	通道总使能	0：通道1，2禁止 1：通道1，2使能	R/W

DACR 用于控制 D/A 转换和 D/A 输出。详见表 18-4。

DA1E 位为 0 并且 DAE 位也为 0 时，通道 1 停止 D/A 转换，这时不管 DAAMP1 是什么值，通道 1 的模拟输出被禁止，端口呈现高阻态。DA2E 位为 0 并且 DAE 位也为 0 时，通道 2 停止 D/A 转换，这时不管 DAAMP2 是什么值，通道 2 的模拟输出被禁止，端口呈现高阻态。

如果选择 A/D 转换优先模式，请在 ADC 停止（ADCSR.ADST=0）的状态下设定 DA2E，DA1E 和 DAE 位。同时，为了避免 ADC 被意外启动，请将 ADC 的触发选择设定为软件触发。

18.5.3 DAC 模拟输出控制寄存器（DAOCRx, x=1,2）

复位值：0x0000

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

位	标记	位名	功能	读写
b15	DAODIS2	DACx_OUT2端口输出控制	1：禁止DACx_OUT2端口输出模拟电压 0：允许DACx_OUT2端口输出模拟电压	R/W
b14	DAODIS1	DACx_OUT1端口输出控制	1：禁止DACx_OUT1端口输出模拟电压 0：允许DACx_OUT1端口输出模拟电压	R/W
b13~b0	Reserved	-	读出时为“0”，写入时写“0”	R/W

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

表 18-4 D/A 转换和模拟输出控制

18.5.4 DAC A/D 转换优先控制寄存器（DAADPCRx, x=1,2）

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADP EN	-	-	-	-	-	DA2 SF	DA1 SF	-	-	-	-	-	ADC SL3	ADC SL2	ADC SL1

位	标记	位名	功能	读写
b15	ADPEN	A/D转换优先模式选择	0：A/D转换优先模式无效 1：A/D转换优先模式有效	R/W
b14~b10	Reserved	-	读出时为“0”，写入时写“0”	R/W
b9	DA2SF	通道2数据更新状态	本标志位只在A/D转换优先模式时变化 0：通道2数据更新完 1：通道2数据更新中	R/W
b8	DA1SF	通道1数据更新状态	本标志位只在A/D转换优先模式时变化 0：通道1数据更新完 1：通道1数据更新中	R/W
b7~b3	Reserved	-	读出时为“0”，写入时写“0”	R/W
b2	ADCSL3	ADC_3优先选择	0：不选择ADC_3作为A/D转换优先对象 1：选择ADC_3作为A/D转换优先对象	R/W
b1	ADCSL2	ADC_2优先选择	0：不选择ADC_2作为A/D转换优先对象 1：选择ADC_2作为A/D转换优先对象	R/W
b0	ADCSL1	ADC_1优先选择	0：不选择ADC_1作为A/D转换优先对象 1：选择ADC_1作为A/D转换优先对象	R/W

DAADPCR 用于 A/D 转换优先模式的控制。

ADPEN 设定成 1 时 A/D 转换优先模式有效，请在设定 ADPEN 的同时选择作为优先对象的 ADC 单元。

DAxSF(x=1,2)为 DA 转换状态标志位，反映的是当前该通道数据寄存器中的数据是否已被转换。

19 温度传感器（OTS）

19.1 简介

温度传感器（On-chip Temperature Sensor，以下简称 OTS）可以获取芯片内部的温度，以支持系统的可靠性操作。OTS 提供了一组与温度相关的数字量，通过计算可以得到温度值。不使用时可以通过模块停止功能将其关闭以减少系统功耗。

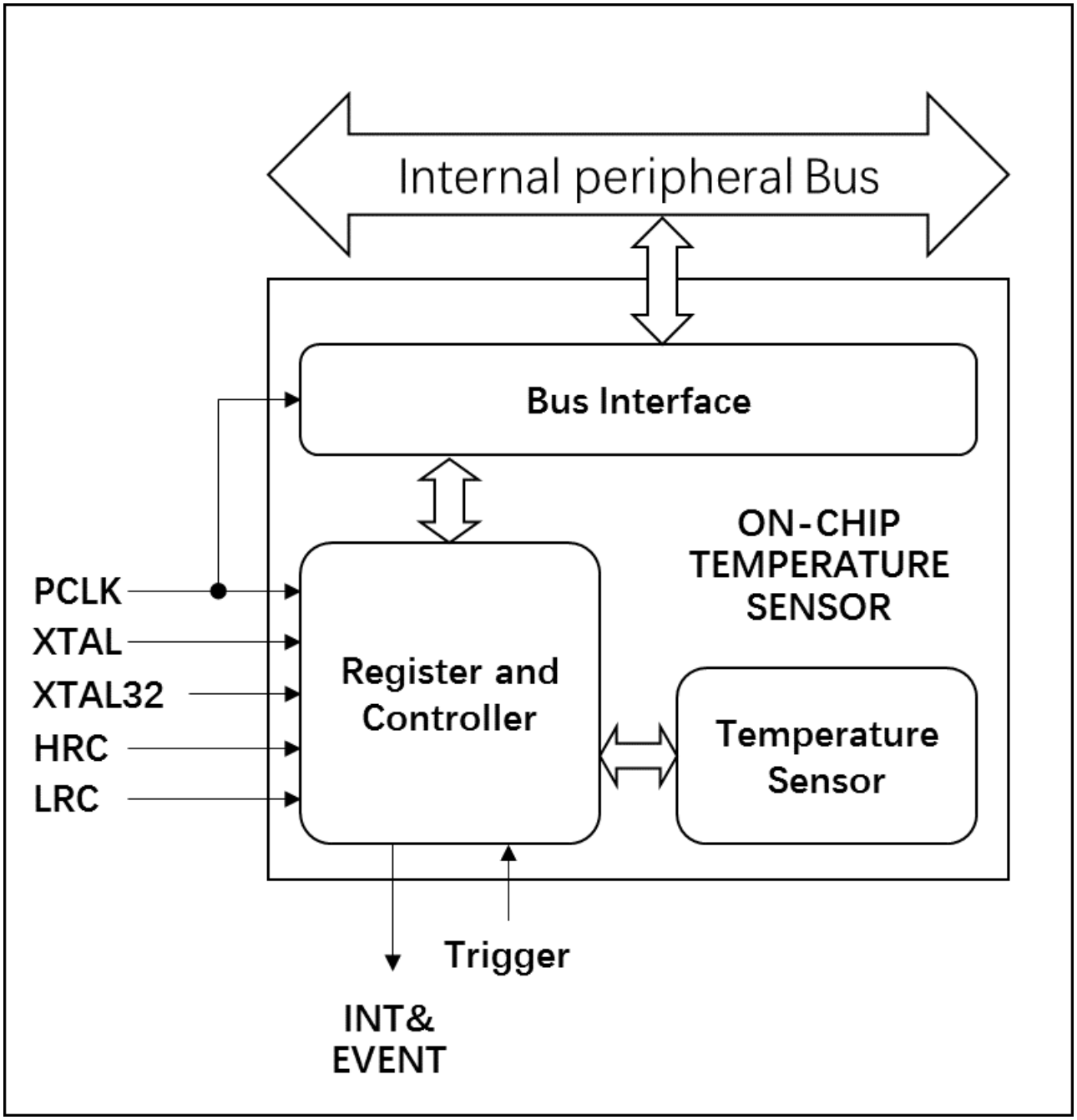


图 19-1 OTS 功能框图

19.2 使用说明

使用 OTS 获取芯片内部温度之前，先要对其进行定标，且测量精度与定标方式有关，请参考下述定标实验的说明进行定标。使用时请先关闭模块停止功能并启动内部低速时钟 LRC,同时请根据使用情况选择启动内部高速时钟 HRC、外部高速时钟 XTAL 以及外部低速时钟 XTAL32。将寄存器 OTS_CTL.OTSST 设置成“1”后测温开始，测温完成后 OTSST 位会自动清零。确认 OTSST 为“0”后，读取寄存器 OTS_DR1,2 以及 OTS_ECR 中的温度参数，并利用以下计算公式求得温度值。

$T = K \times (1.7 / D1 - 1 / D2) \times E_{hrc} + M$

【参数说明】

T：温度（℃）

K：温度斜率（通过定标实验确定）

D1：温度数据 1（从寄存器 OTS_DR1 读取）

D2：温度数据 2（从寄存器 OTS_DR2 读取）

E_{hrc}：HRC 频率误差补偿数据（从寄存器 OTS_ECR 读取）

M：温度偏移量（通过定标实验确定）

寄存器 OTS_CTL.OTSCK 用来选择测温时钟。选择 HRC 动作时，其频率误差可能会影响到最终计算出的温度的精度。为了消除这一误差，请在测温前启动 XTAL32 并在定标和计算温度时使用 E_{hrc}。选择 XTAL 动作时请忽略 E_{hrc} 并将其示为常量 1。

【定标实验】

请在两个确定的温度下进行定标实验，计算 K 和 M。

$K = (T2 - T1) / (A2 - A1)$

$M = T1 - K \times A1 = T2 - K \times A2$

T1：实验温度 1

T2：实验温度 2

A1：$(1.7 / D1_{T1} - 1 / D2_{T1}) \times E_{hrcT1}$

$D1_{T1}$, $D1_{T1}$, E_{hrcT1} 分别是在温度 T1 下测到的 D1，D2，E_{hrc}；

A2：$(1.7 / D1_{T2} - 1 / D2_{T2}) \times E_{hrcT2}$

$D1_{T2}$, $D1_{T2}$, E_{hrcT2} 分别是在温度 T2 下测到的 D1，D2，E_{hrc}；

作为参考，芯片内部预置了三组温度数据，可任意选择两个温度点的数据进行定标。由于测试环境温度会有偏差，预置数据仅供参考使用。

测试温度	读取方式		测试条件
	温度数据 1（D1）	温度数据 2（D2）	
Ta= -40℃	OTS_PDR3[15:0]	OTS_PDR3[31:16]	XTAL=8MHz
Ta= 25℃	OTS_PDR1[15:0]	OTS_PDR1[31:16]	XTAL=8MHz
Ta= 125℃	OTS_PDR2[15:0]	OTS_PDR2[31:16]	XTAL=8MHz

表 19-1 OTS 预置温度数据

E_{hrc} 是为了消除 HRC 本身的误差对测量温度影响的一个校准值，可以从寄存器 OTS_ECR 中读取，使用方法如下。

动作时钟 (OTSCK)	定标实验（计算 K,M）		温度测量（计算 T）
	使用预置温度数据	重新获取温度数据	
XTAL	E_{hrc}=1	E_{hrc}=1	E_{hrc}=1
HRC	$E_{hrc}=f_{hrc}(MHz)/0.032768$	E_{hrc}=OTS_ECR[15:0]	E_{hrc}=OTS_ECR[15:0]

f_{hrc} 为 HRC 的标准频率，即 16MHz 或者 20MHz。

表 21-2 E_{hrc} 使用及设定方法

寄存器 OTS_CTL.TSSTP 用来选择测温完成后是否关闭模拟温度传感器。TSSTP 初始值为“0”，表示在一次测温完成后将模拟温度传感器置于开启状态，这样在下一次测温时将自动跳过模拟温度传感器由关闭到开启时的稳定时间。如需在每次测温后关闭模拟温度传感器，请将 TSSTP 设置成“1”。

可以通过其他的外设事件触发开始测温，使用时请将触发源的触发目标设置成 OTS。测温完成时也可以产生触发其他的外设启动的事件，使用时请设定寄存器 OTS_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”。

19.3 寄存器说明

基准地址：0x4004A800

寄存器名	符号	偏移地址	位宽	复位值
OTS控制寄存器	OTS_CTL	0x00	16	0x0000
OTS数据寄存器1	OTS_DR1	0x02	16	0x0000
OTS数据寄存器2	OTS_DR2	0x04	16	0x0000
OTS误差补偿寄存器	OTS_ECR	0x06	16	0x0000

基准地址：0x40010600

寄存器名	符号	偏移地址	位宽	复位值
OTS预置数据寄存器1	OTS_PDR1	0xe0	32	不定值
OTS预置数据寄存器2	OTS_PDR2	0xf4	32	不定值
OTS预置数据寄存器3	OTS_PDR3	0xf8	32	不定值

基准地址：0x40010800

寄存器名	符号	地址	位宽	复位值
OTS触发源选择寄存器	OTS_TRG	0x90	16	0x0000

表 19-3 OTS 寄存器一览表

19.3.1 OTS 控制寄存器（OTS_CTL）

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	TSSTP	OTSIE	OTSCK	OTSST

复位值：0x0000

位	标记	位名	功能	读写
b15~b4	Reserved	-	读时读出0，写时请写0	R/W
b3	TSSTP	关闭模拟温度传感器	选择测温结束后是否自动关闭模拟温度传感器 0：不关闭模拟温度传感器 1：关闭模拟温度传感器	R/W
b2	OTSIE	中断使能位	0：禁止测温结束中断请求 1：允许测温结束中断请求	R/W
b1	OTSCK	时钟选择位	0：选择外部高速时钟（XTAL）动作 1：选择内部高速时钟（HRC）动作	R/W
b0	OTSST	测温开始位	0：停止测温 1：开始测温 置“1”条件： （1）软件置“1” （2）硬件触发置“1” 清“0”条件： （1）软件清“0” （2）测温结束后自动清“0”	R/W

19.3.2 OTS 数据寄存器 1（OTS_DR1）

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

复位值：0x0000

位	标记	位名	功能	读写
b15~b0	TSDC[15:0]	温度数据D1	温度数据D1 测温完成后自动更新。请确认OTS_CTL.OTSST为“0”后再读取。	R

19.3.3 OTS 数据寄存器 2（OTS_DR2）

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

复位值：0x0000

位	标记	位名	功能	读写
b15~b0	TSDC[15:0]	温度数据D2	温度数据D2 测温完成后自动更新。请确认OTS_CTL.OTSST为“0”后再读取。	R

19.3.4 OTS 误差补偿寄存器（OTS_ECR）

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

复位值：0x0000

位	标记	位名	功能	读写
b15~b0	TSEC[15:0]	误差系数	误差系数Ehrc 测温完成后自动更新。请确认OTS_CTL.OTSST为“0”后再读取。	R

19.3.5 OTS 预置温度数据寄存器（OTS_PDR1,2,3）

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TSPD2[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TSPD1[15:0]															

复位值：不定值

位	标记	位名	功能	读写
b31~b16	TSPD2[15:0]	预置温度数据D2	预置温度数据，供定标时使用。 该数据为8MHz时测得，使用前请先与用由OTSCK选择的动作时钟的实际频率进行换算。 D2= TSPD2×f_{OTSCK}/8，f_{OTSCK}为实际动作时钟的频率，单位MHz。	R
b15~b0	TSPD1[15:0]	预置温度数据D1	预置温度数据，供定标时使用。 该数据为8MHz时测得，使用前请先与用由OTSCK选择的动作时钟的实际频率进行换算。 D1= TSPD1×f_{OTSCK}/8，f_{OTSCK}为实际动作时钟的频率，单位MHz。	R

19.3.6 OTS 触发源选择寄存器（OTS_TRG）

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	TSTSL[8:0]								

复位值：0x0000

位	标记	位名	功能	读写
b15~b9	Reserved	-	读时读出0，写时请写0	R/W
b8~b0	TSTSL	触发源选择	选择硬件触发启动时的触发源编号 请参考“10.3.2”章节的表 10-2进行设置，表中“是否可选择为事件”列打“√”的中断事件源可用作触发源。	R/W

20 高级控制定时器（Timer6）

20.1 简介

高级控制定时器 6（Timer6）是一个 16 位计数宽度的高性能定时器，能在各种复杂应用场景中提供丰富、灵活的搭配组合和各种中断、事件、PWM 输出。该定时器支持锯齿波和三角波两种波形模式，可生成各种 PWM 波形（单边对齐独立 PWM、双边对称独立 PWM、双边对称互补 PWM、双边非对称 PWM 等）；单元间可实现软件同步和硬件同步（同步启动、停止、清零、刷新等）；各基准值寄存器支持缓存功能（单级缓存和双级缓存）；支持脉宽测量和周期测量；支持 2 相正交编码和 3 相正交编码；支持 EMB 控制。本系列产品中搭载 8 个单元的 Timer6（U1~4 为 32bit 定时器；U5~U8 为 16bit 定时器）。

20.2 基本框图

Timer6 基本的功能及特性如表 20-1 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
中断输出	计数比较匹配中断
	计数周期匹配中断
	死区时间错误中断
事件输出	计数比较匹配事件
	计数周期匹配事件

表 20-1 Timer6 的基本功能及特性

Timer6 的基本框图如图 20-1 所示。框图中所示”<t>”表示单元编号，即”<t>”为 1~8，本章节后文提到“<t>”时均指单元编号，不再赘述。

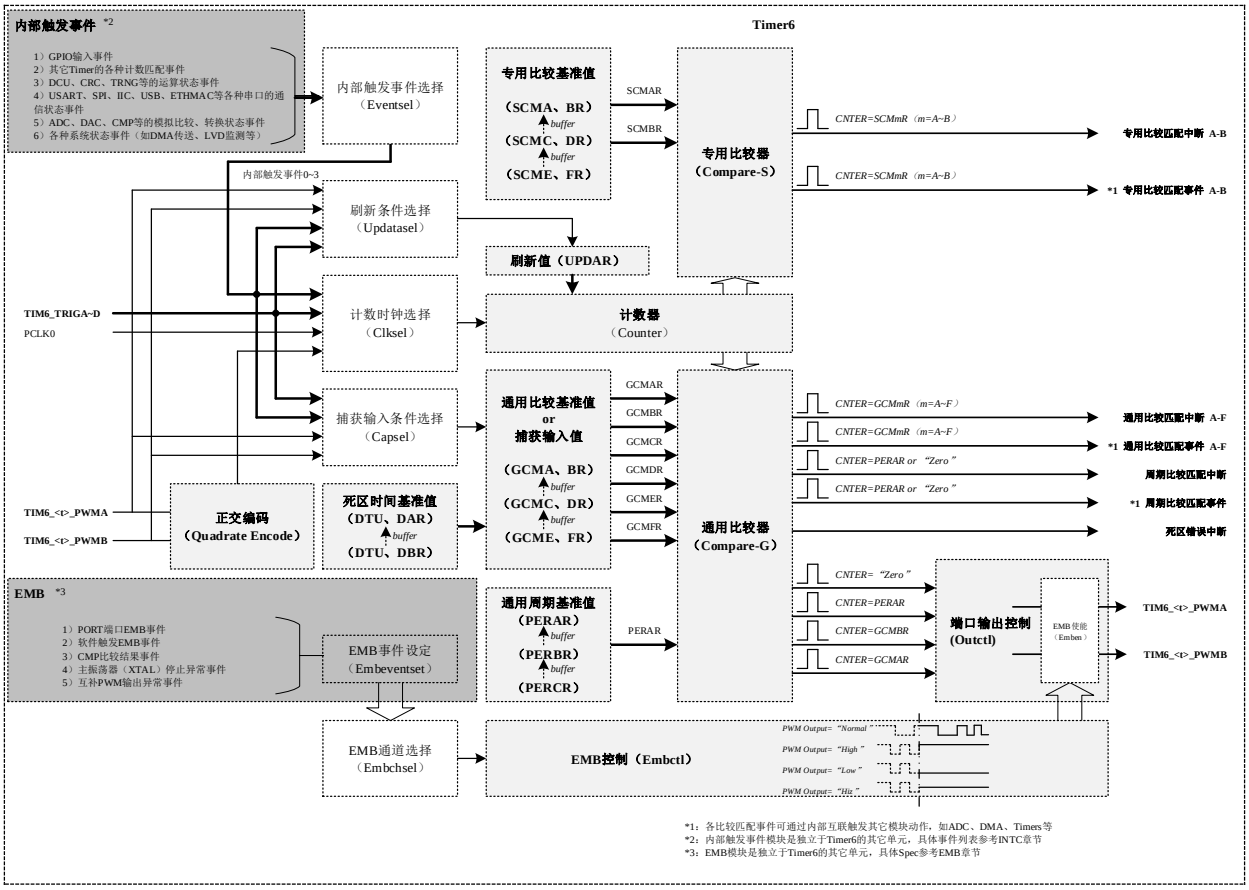


图 20-1 Timer6 基本框图

表 20-2 所示为 Timer6 的输入输出端口列表。

端口名	方向	功能
TIM6_<t>_PWMA	in or out	1) 正交编码计数时钟输入端口或捕获输入端口或比较输出端口 2) 硬件启动、停止、清零、刷新条件输入端口
TIM6_<t>_PWMB		
TIM6_TRIGA	in	1) 硬件计数时钟输入端口或捕获输入端口 2) 硬件启动、停止、清零、刷新条件输入端口
TIM6_TRIGB		
TIM6_TRIGC		
TIM6_TRIGD		

表 20-2 Timer6 端口列表

20.3 功能说明

20.3.1 波形模式

Timer6 有 2 种基本计数波形模式，锯齿波模式和三角波模式。两种波形模式的基本波形如图 20-2、图 20-3 所示。

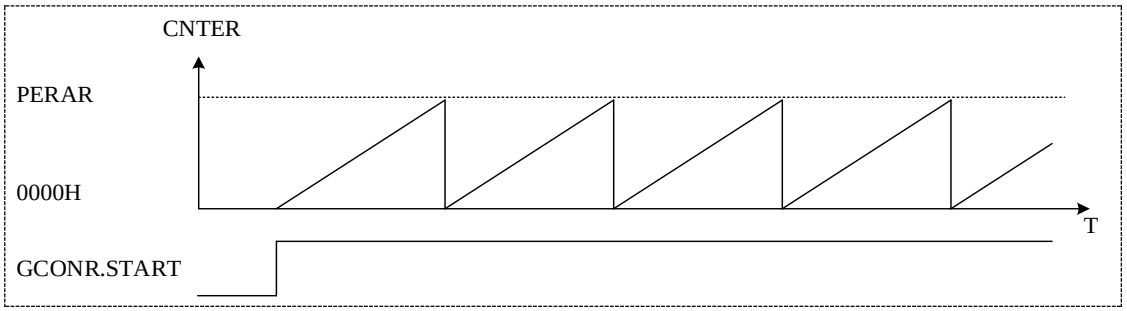


图 20-2 锯齿波波形（递加计数）

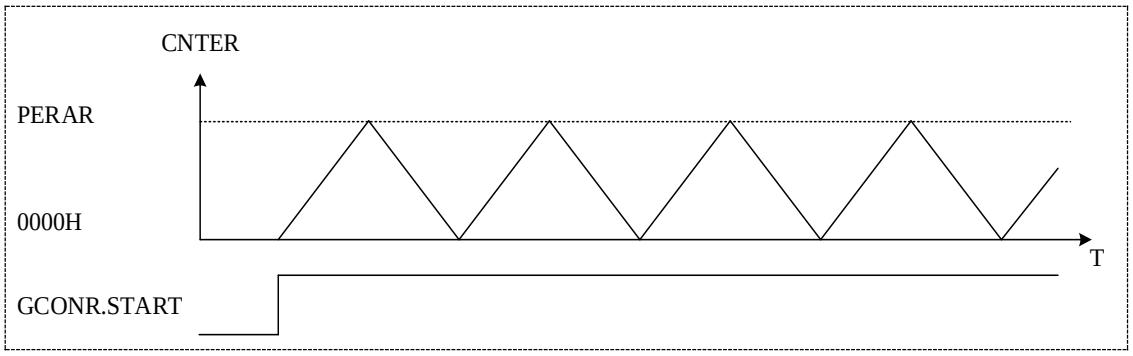


图 20-3 三角波波形

20.3.2 时钟源选择

Timer6 的计数时钟可以有以下几种选择：

a) PCLK0 及 PCLK0 的 2、4、8、16、32、64、128、256、512、1024 分频（GCONR.CKDIV[3:0]设定）

b) 内部触发事件输入 0~3（HCUPR[11:8]或 HCDOR[11:8]设定）

c) TIM6_TRIGA~D 的端口输入（HCUPR[23:16]或 HCDOR[23:16]设定）

d) TIM6_<t>_PWMA 和 TIM6_<t>_PWMB 的正交编码关系输入（HCUPR[7:0]或 HCDOR[7:0]设定）

计数时钟源选择 a 时为软件计数模式，计数时钟源选择 b、c、d 时为硬件计数模式。上述描述可以看到，b、c、d 时钟互相独立，可分别设定有效或无效，并且当选择 b、c、d 时钟时，a 时钟自动无效。

20.3.3 计数方向

Timer6 的定时器计数方向可通过软件方式改变。不同波形模式时，改变计数方向的方法略有不同。

20.3.3.1 锯齿波计数方向

锯齿波模式时，计数方向可在定时器计数中或停止时设定。

在递加计数中时，设定 GCONR.DIR=0（递减计数），则定时器计数到上溢后变为递减计数模式；在递减计数中时，设定 GCONR.DIR=1（递加计数），则定时器计数到下溢后变为递加计数模式。

在计数停止时，设定 GCONR.DIR 位， 计数开始后直至上溢或下溢时，GCONR.DIR 的设定反映到计数中。

20.3.3.2 三角波计数方向

三角波模式时，设定计数方向无效， 计数至计数峰点或计数谷点时，自动改变计数方向。

20.3.4 比较输出

每个单元的 Timer6 有 2 个比较输出端口（TIM6_<t>_PWMA、TIM6_<t>_PWMB），可在计数值与比较基准值比较匹配时输出指定的电平。GCMAR、GCMBR 寄存器对应了 TIM6_<t>_PWMA、TIM6_<t>_PWMB 的计数比较基准值。当定时器的计数值和 GCMA<B>R 相等时，TIM6_<t>_PWMA 或 TIM6_<t>_PWMB 端口输出指定电平。

TIM6_<t>_PWMA 端口的计数开始时的电平、计数停止时的电平、计数溢出时的电平、计数比较匹配时的电平等，可通过端口控制寄存器（PCNAR）的 PCNAR.STACA、PCNAR.STPCA 、 PCNAR.OVFCA 、 PCNAR.UDFCA 、 PCNAR.CMA<B>U<D>A 、PCNAR.FORCA 位设定。

TIM6_<t>_PWMB 端口的计数开始时的电平、计数停止时的电平、计数溢出时的电平、计数比较匹配时的电平等，可通过端口控制寄存器（PCNBR）的 PCNBR.STACB、PCNBR.STPCB 、 PCNBR.OVFCB 、 PCNBR.UDFCB 、 PCNBR.CMA<B>U<D>B 、PCNBR.FORCB 位设定。图 20-4 为比较输出的动作例。

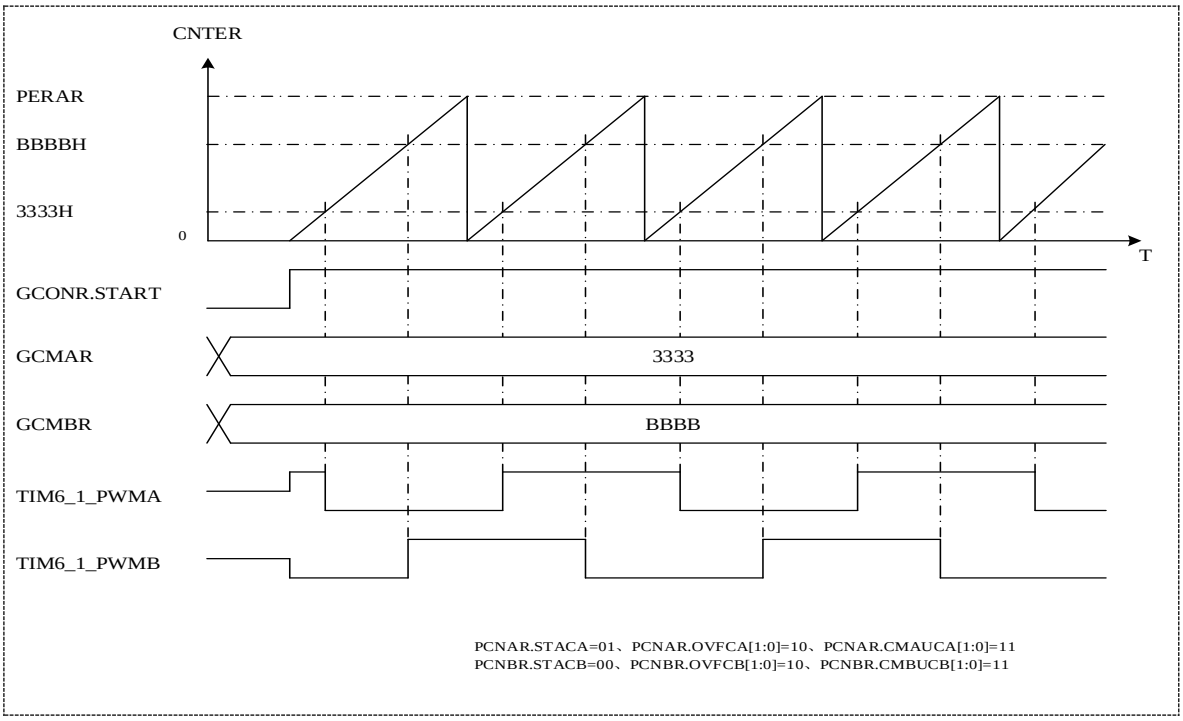


图 20-4 比较输出动作

20.3.5 捕获输入

每个单元都具有捕获输入功能，具备 2 组捕获输入寄存器（GCMAR、GCMBR），用于保存捕获到的当前计数值。设定端口控制寄存器（PCNA<B>R）的 CAPMDA<B>位为 1，捕获输入功能变为有效。当设定了对应的捕获输入条件且该条件有效时，当前的计数值就被保存到相应的捕获寄存器（GCMAR、GCMBR）中。

各单元的每组捕获输入的条件可以是内部触发事件输入 0~3、TIM6_TRIGA~D 的端口输入、TIM6_<t>_PWMA 或 TIM6_<t>_PWMB 的端口输入等，具体的条件选择可通过硬件捕获事件选择寄存器（HCPAR、HCPBR）来设定。图 20-5 为捕获输入的动作例。

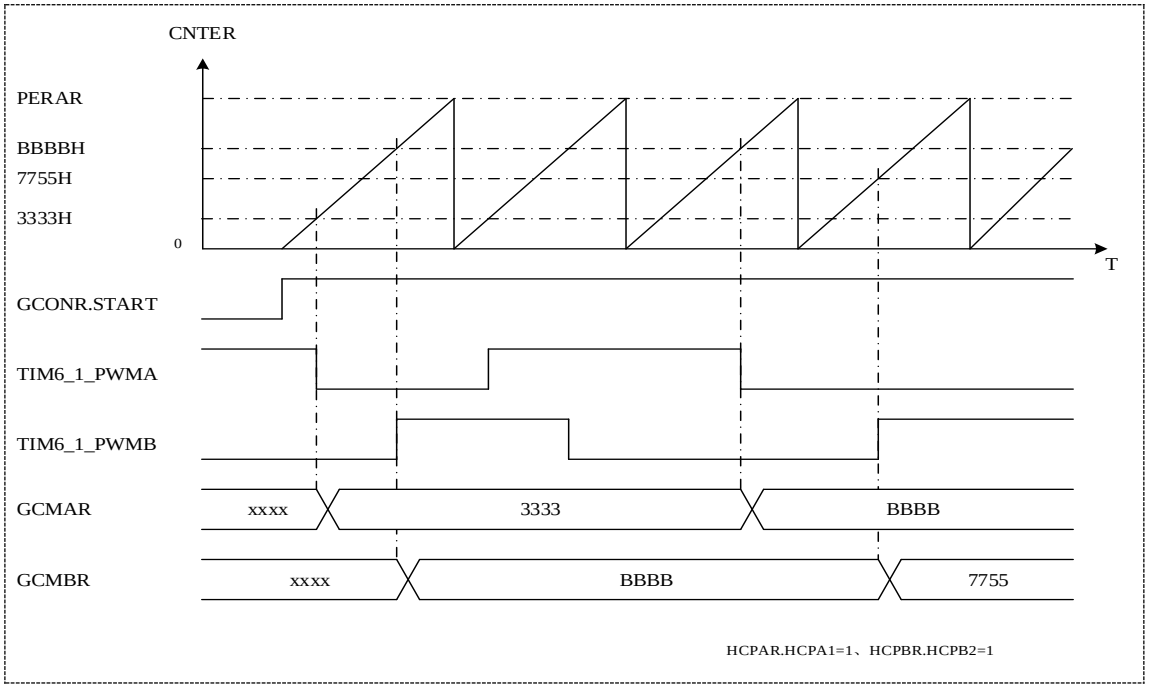


图 20-5 捕获输入动作

20.3.6 计数器刷新

Timer6 具有一个通用刷新值寄存器（UPDAR），可以在计数停止或计数中时，对通用计数值寄存器（CNTER）的计数值实时进行刷新。

计数值的刷新条件由硬件刷新事件选择寄存器（HUPDR）或软件同步刷新控制寄存器（SUPDR）的对应位设定。当设定的刷新事件有效时，通用计数值寄存器（CNTER）的值更新为通用刷新值寄存器（UPDAR）中指定的值。图 20-6 是锯齿波模式在计数中的硬件刷新动作例。

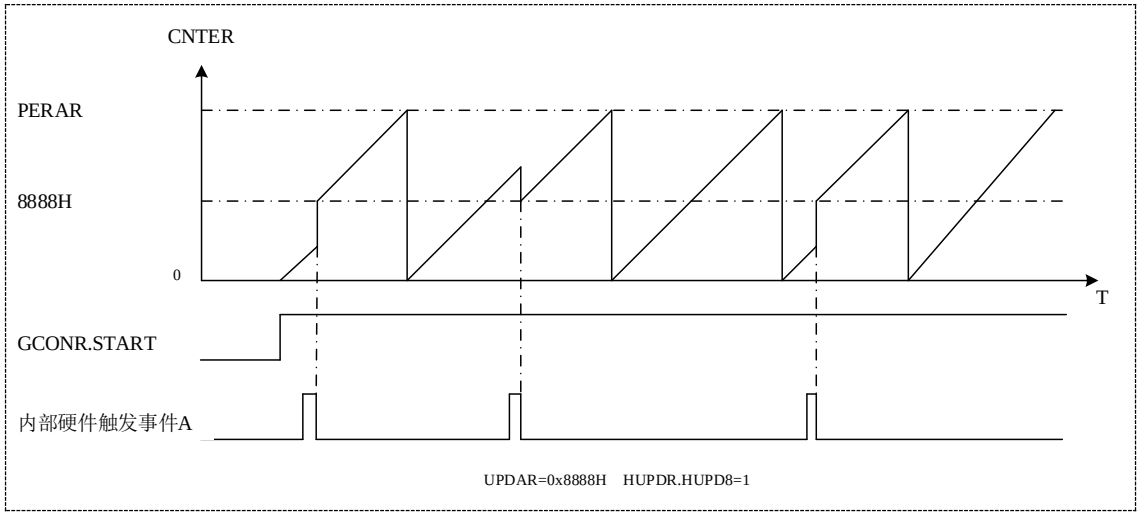


图 20-6 硬件刷新动作

20.3.7 软件同步

20.3.7.1 软件同步启动

各单元可通过设定软件同步启动控制寄存器（SSTAR）的相关位，实现目标单元计数器（CNTER）的同步启动。

20.3.7.2 软件同步停止

各单元可通过设定软件同步停止控制寄存器（SSTPR）的相关位，实现目标单元计数器（CNTER）的同步停止。

20.3.7.3 软件同步清零

各单元可通过设定软件同步清零控制寄存器（SCLRR）的相关位，实现目标单元计数器（CNTER）的同步清零。

20.3.7.4 软件同步刷新

各单元可通过设定软件同步刷新控制寄存器（SUPDR）的相关位，实现目标单元计数器（CNTER）的同步刷新。

如图 20-7 所示、若设定 SSTAR.SSTA1=SSTAR.SSTA2=SSTAR.SSTA3=1，即可实现单元 1~3 的软件同步启动、设定 SSTPR.SSTP1=SSTPR.SSTP2=SSTPR.SSTP3=1，即可实现单元 1~3 的软件同步停止。

软件同步动作相关寄存器（SSTAR、SSTPR、SCLRR、SUPDR）是一组独立于单元外、各个单元间共用的寄存器，这组寄存器的各个位只在写 1 时有效，写 0 无效。在读取 SSTAR 寄存器时，会读出各个单元的定时器状态（计数停止或计数中），在读取 SSTPR、SCLRR 或 SUPDR 时，会读出零。

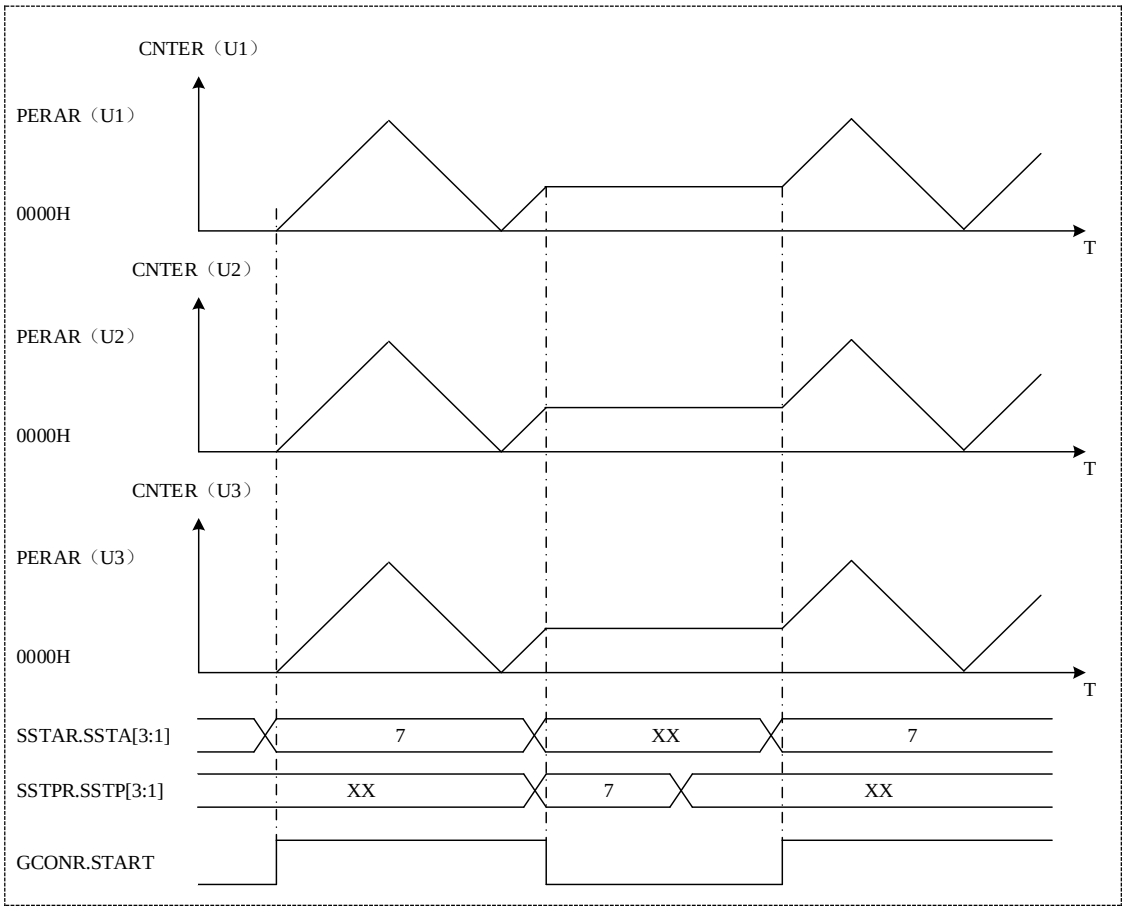


图 20-7 软件同步动作

20.3.8 硬件同步

每个单元除独立拥有 2 个通用输入端口（TIM6_<t>_PWMA、TIM6_<t>_PWMB）外，还共同拥有 4 个通用外部触发输入端口（TIM6_TRIGA、TIM6_TRIGB、TIM6_TRIGC、TIM6_TRIGD）及 4 个内部触发事件输入条件，可实现单元间的硬件同步动作。

内部触发事件的事件源可通过硬件触发事件选择寄存器（HTSSR0~3）中对应的编号设定来选择，具体的事件对应关系请参考 INTC 章节。使用内部触发功能时，需要先将功能时钟控制寄存器 0（PWR_FCG0）的外围电路触发功能使能位置 1。

20.3.8.1 硬件同步启动

各单元均可选择用硬件方式启动定时器，选择相同硬件启动条件的单元即可在启动条件有效时实现同步启动。具体的硬件启动条件由硬件启动事件选择寄存器（HSTAR）的设定来决定。

20.3.8.2 硬件同步停止

各单元均可选择用硬件方式停止定时器，选择相同硬件停止条件的单元即可在停止条件有效时实现同步停止。具体的硬件停止条件由硬件停止事件选择寄存器（HSTPR）的设定来决定。

20.3.8.3 硬件同步清零

各单元均可选择用硬件方式清零定时器，选择相同硬件清零条件的单元即可在清零条件有效时实现同步清零。具体的硬件清零条件由硬件清零事件选择寄存器（HCLRR）的设定来决定。

20.3.8.4 硬件同步刷新

各单元均可选择用硬件方式刷新定时器，选择相同硬件刷新条件的单元即可在刷新条件有效时实现同步刷新。具体的硬件刷新条件由硬件刷新事件选择寄存器（HUPDR）的设定来决定。

20.3.8.5 硬件同步捕获输入

各单元均可选择用硬件方式实现捕获输入功能，选择相同捕获输入功能条件的单元即可在捕获输入功能条件有效时实现同步捕获输入。具体的硬件捕获输入功能条件由硬件捕获事件选择寄存器（HCPAR、HCPBR）的设定来决定。

20.3.8.6 硬件同步计数

各单元均可选择用硬件输入作为 CLOCK 进行计数，选择相同硬件计数条件的单元即可在硬件计数时钟有效时实现同步计数。具体的硬件计数条件由硬件递加事件选择寄存器（HCUPR）和硬件递减事件选择寄存器（HCDOR）的设定来决定。

选择硬件同步计数功能时，只是选择了外部输入时钟源，不影响定时器的启动、停止、清零动作。定时器的启动、停止、清零等还需要单独设定。

图 20-8 所示、是单元 1~3 的硬件同步动作例。

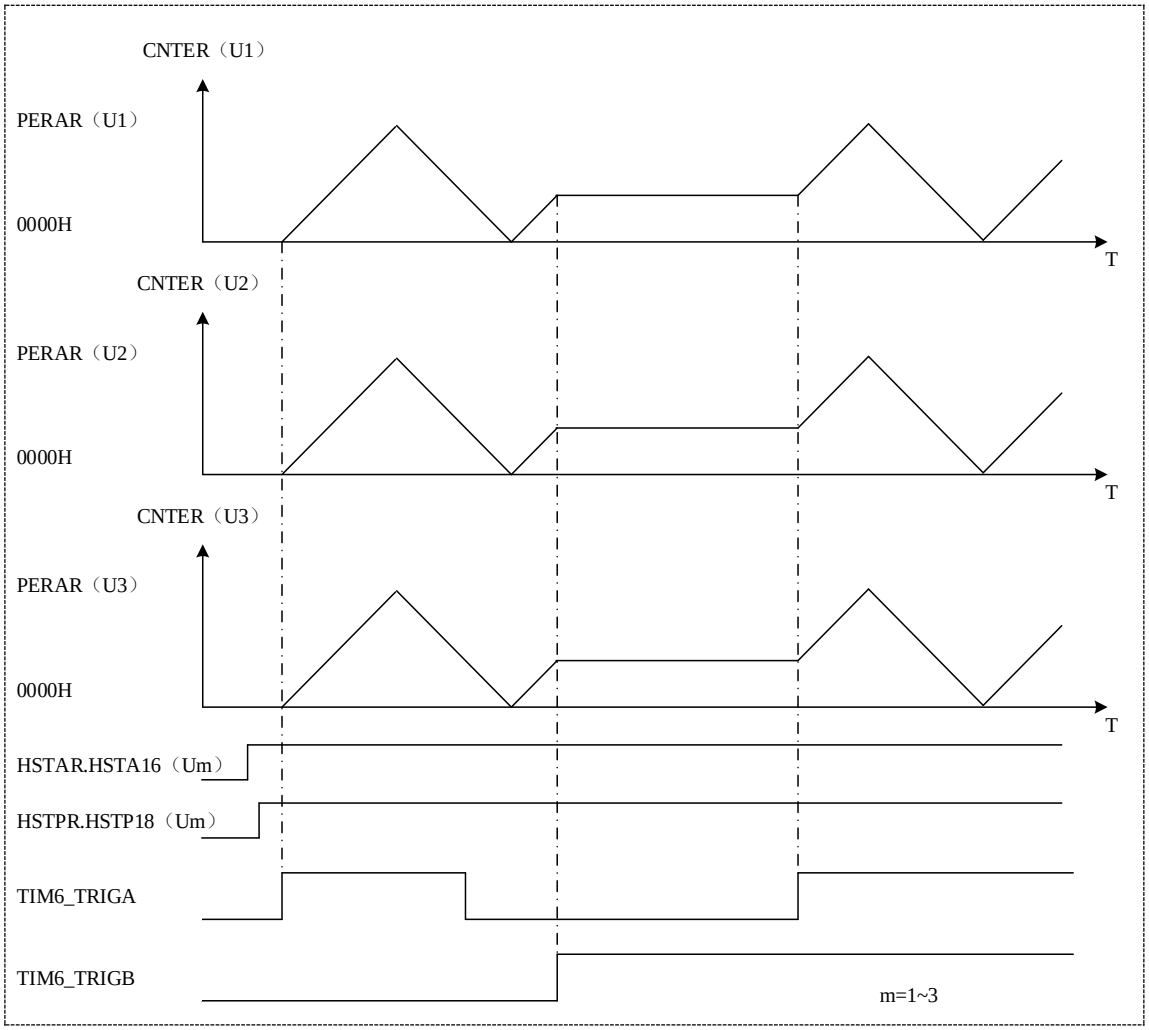


图 20-8 硬件同步动作

20.3.9 脉宽测量

在使用 TIM6_<t>_TRIGA~D 端口的硬件触发相关功能（参考硬件同步章节）时，每个单元都可以实现 2 路独立的脉宽测量功能。

例如，将计数器的硬件启动条件设为 TIM6_<t>_TRIGA 的上升沿，硬件清零条件、停止条件和 GCMAR 寄存器的捕获输入条件均设为 TIM6_<t>_TRIGA 的下降沿，就可以实现连续的脉冲宽度测量。对应动作如图 20-9 所示。

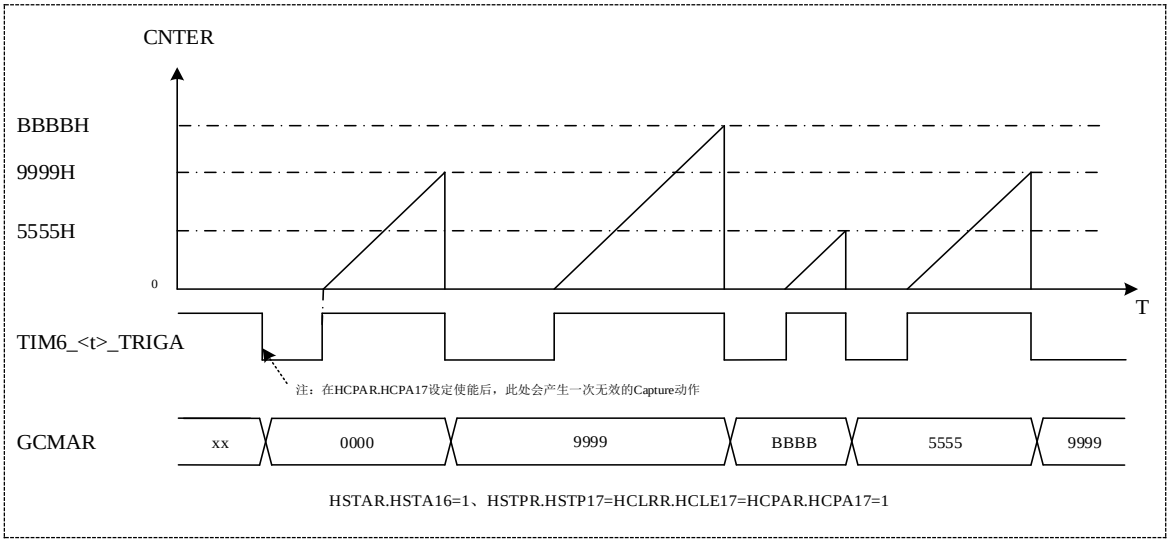


图 20-9 脉宽测量

20.3.10 周期测量

在使用 TIM6_<t>_TRIGA~D 端口的硬件触发相关功能（参考硬件同步章节）时，每个单元都可以实现 2 路独立的周期测量功能。

例如，将计数器的硬件启动条件、硬件清零条件、GCMBR 寄存器的捕获输入条件均设为 TIM6_<t>_TRIGB 的上升沿，就可以实现连续的周期测量。对应动作如图 20-10 所示。

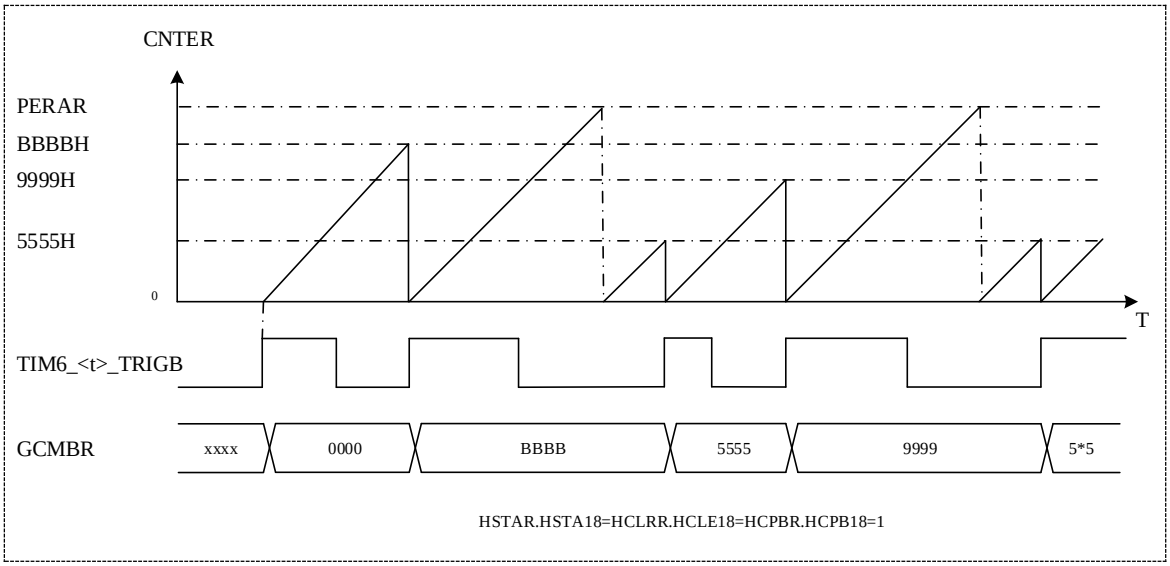


图 20-10 周期测量

20.3.11 缓存功能

Timer6 的通用周期基准值、通用比较基准值、专用比较基准值、死区时间基准值等都具有缓存功能，可在计数期间硬件实现周期变化、占空比变化、死区变化等。周期基准值、通用比较基准值和专用比较基准值等有单缓存、双缓存功能，死区时间基准值有单缓存功能。

20.3.11.1 单缓存动作

单缓存动作是指通过设定缓存控制寄存器（ BCONR.BENA<B><P><SPA><SPB>=1 、BCONR.BSEA<B><P><SPA><SPB>=0）、死区控制寄存器（DCONR.DTBENU<D>=1），在缓存传送时间点，选择发生以下事件：

a) 通用周期基准值缓存寄存器（PERBR）的值自动传送到通用周期基准值寄存器（PERAR）中
b) 通用比较基准值缓存寄存器（GCMCR、GCMDR）的值自动传送到通用比较基准值寄存器（GCMAR、GCMBR）中（比较输出时）
c) 通用比较基准值寄存器（GCMAR、GCMBR）的值自动传送到通用比较基准值缓存寄存器（GCMCR、GCMDR）中（捕获输入时）
d) 专用比较基准值缓存寄存器（SCMCR、SCMDR）的值自动传送到专用比较基准值寄存器（SCMAR、SCMBR）中
e) 死区时间基准值缓存寄存器（DTUBR、DTDBR）的值自动传送到死区时间基准值寄存器（DTUAR、DTDAR）中

如图 20-11 所示，是单元 1 比较输出动作（PCNAR.CAPMDA=0）时、通用比较基准值寄存器的单缓存方式的时序图。从图中可以看到，在计数期间改变通用比较基准值寄存器（GCMAR）的值可以调整输出占空比，改变通用周期基准值寄存器（PERAR）的值可以调整输出周期。

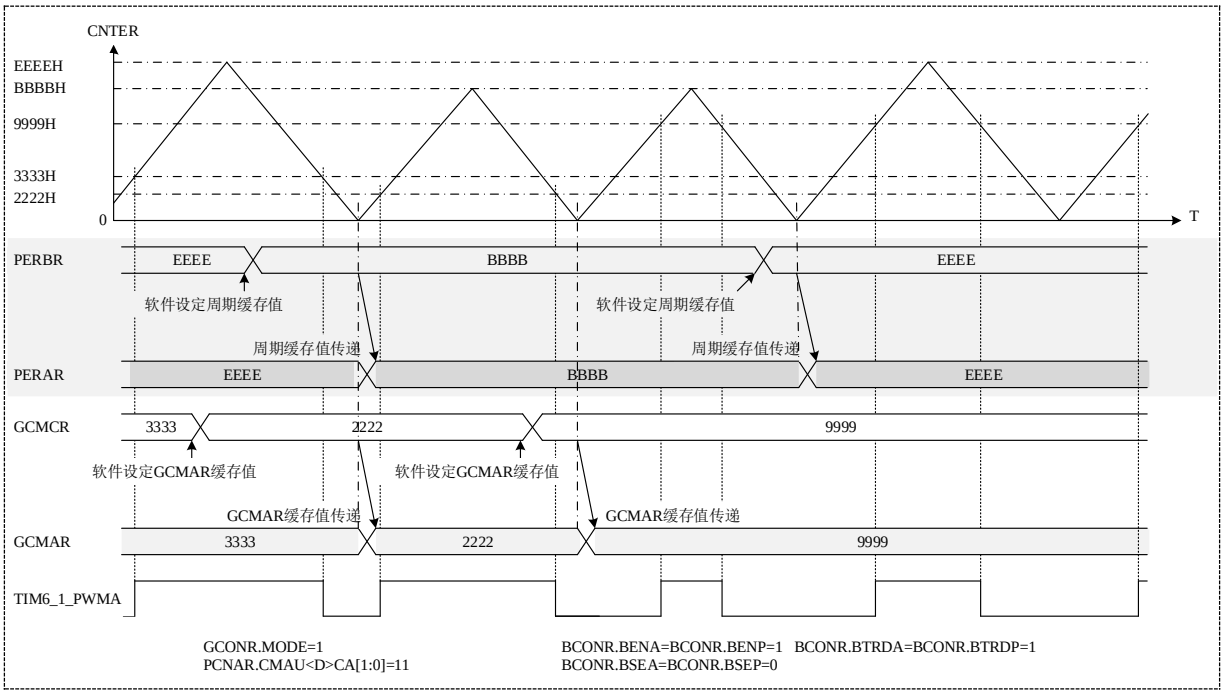


图 20-11 单缓存方式比较输出时序

20.3.11.2 双缓存动作

双缓存动作是指通过设定缓存控制寄存器（ BCONR.BENA<B><P><SPA><SPB>=1 、BCONR.BSEA<B><P><SPA><SPB>=1），在缓存传送时间点，选择发生以下事件：

a) 通用周期基准值缓存寄存器（PERBR）的值自动传送到通用周期基准值寄存器（PERAR）中、通用周期基准值双缓存寄存器（PERCR）的值自动传送到通用周期基准值缓存寄存器（PERBR）中

b) 通用比较基准值缓存寄存器（GCMCR、GCMDR）的值自动传送到通用比较基准值寄存器（GCMAR、GCMBR）、通用比较基准值双缓存寄存器（GCMER、GCMFR）的值自动传送到通用比较基准值缓存寄存器（GCMCR、GCMDR）中（比较输出时）

c) 通用比较基准值缓存寄存器（GCMCR、GCMDR）的值自动传送到通用比较基准值双缓存寄存器（GCMER、GCMFR）中、通用比较基准值寄存器（GCMAR、GCMBR）的值自动传送到通用比较基准值缓存寄存器（GCMCR、GCMDR）中（捕获输入时）

d) 专用比较基准值缓存寄存器（SCMCR、SCMDR）的值自动传送到专用比较基准值寄存器（SCMAR、SCMBR）中、专用比较基准值双缓存寄存器（SCMER、SCMFR）

的值自动传送到专用比较基准值缓存寄存器（SCMCR、SCMDR）中图 20-12 所示，是内部触发事件 0 触发捕获输入时、双缓存方式的时序图。

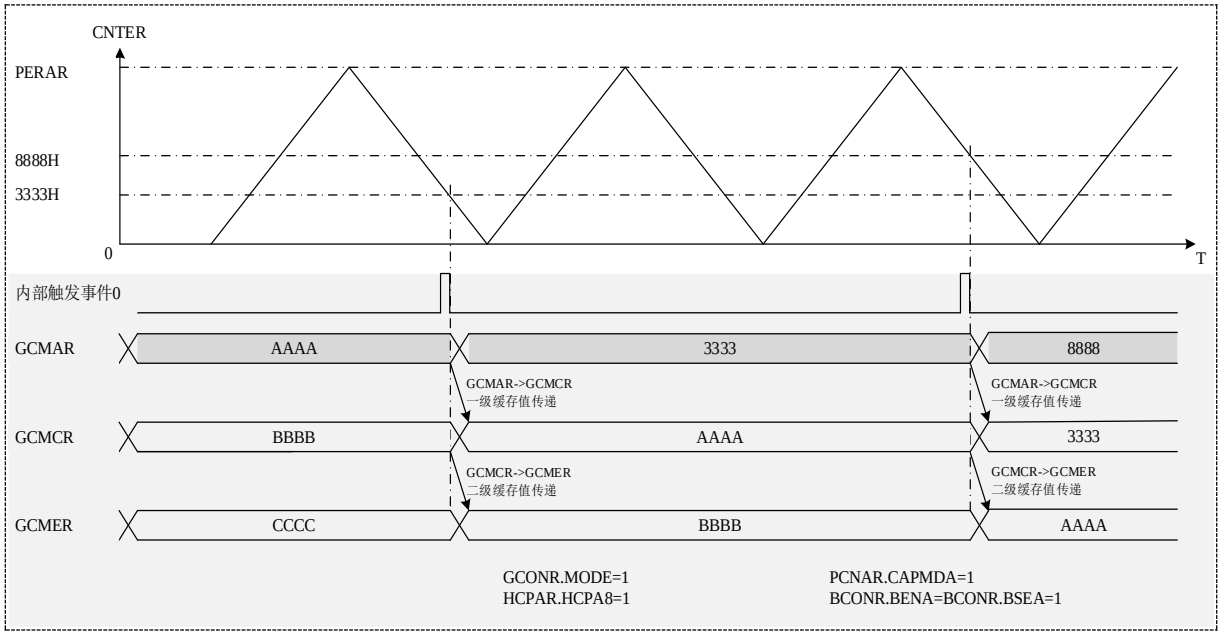


图 20-12　双缓存方式捕获输入时序

20.3.11.3 缓存传送节点

比较输出（锯齿波）缓存传送

在缓存功能有效（BCONR.BENA<B><P><SPA><SPB>=DCONR.DTBENU<D>=1）&& 比较输出计数（PCNA<B>R.CAPMDA<B>=0）&& 锯齿波计数模式（GCONR.MODE=0）时，通用周期基准值、通用比较基准值、专用比较基准值、死区时间基准值的缓存传送发生在递加计数上溢点或递减计数下溢点。

如图 20-13 所示，是锯齿波递加计数时的缓存动作。

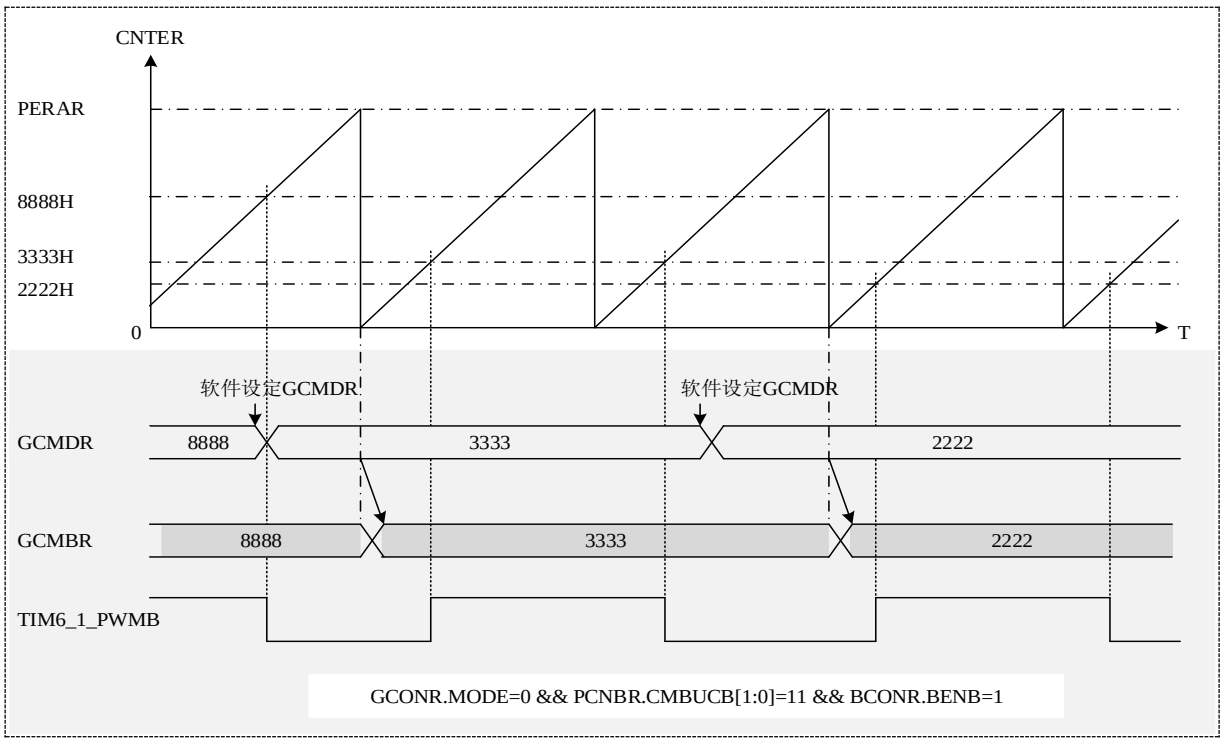


图 20-13　锯齿波模式时计数缓存动作

注意：

– 在锯齿波计数模式时，若产生清零动作，也视为一次计数溢出。各基准值会根据相应的缓存动作设定状况（单缓存、双缓存等）发生一次缓存传送。

– 在硬件计数模式时，若产生清零动作，也视为一次计数溢出。通用周期基准值、通用比较基准值会根据相应的缓存动作设定状况（单缓存、双缓存、上溢传送、下溢传送等）发生一次缓存传送，其它基准值不发生缓存传送。

比较输出（三角波）缓存传送

在缓存功能有效（BCONR.BENA<B><P><SPA><SPB>=DCONR.DTBENU<D>=1）&& 比较输出计数（PCNA<B>R.CAPMDA<B>=0）&& 三角波计数模式（GCONR.MODE=0）时，通用周期基准值、通用比较基准值、专用比较基准值、死区时间基准值的缓存传送时间点由对应的寄存器控制位决定。当 BCONR.BTRUA<B><P><SPA><SPB>=1 或 DCONR.DTBTRU=1 时，在计数器计数到三角波的峰点时，发生对应的缓存传送；当 BCONR.BTRDA<B><P><SPA><SPB>=1 或 DCONR.DTBTRD=1 时，在计数器计数到三角波的谷点时，发生对应的缓存传送。如图 20-14 所示，是三角波计数到谷点时的缓存动作，图 20-15 所示，是三角波计数到峰点和谷点时均发生缓存动作的示意图。

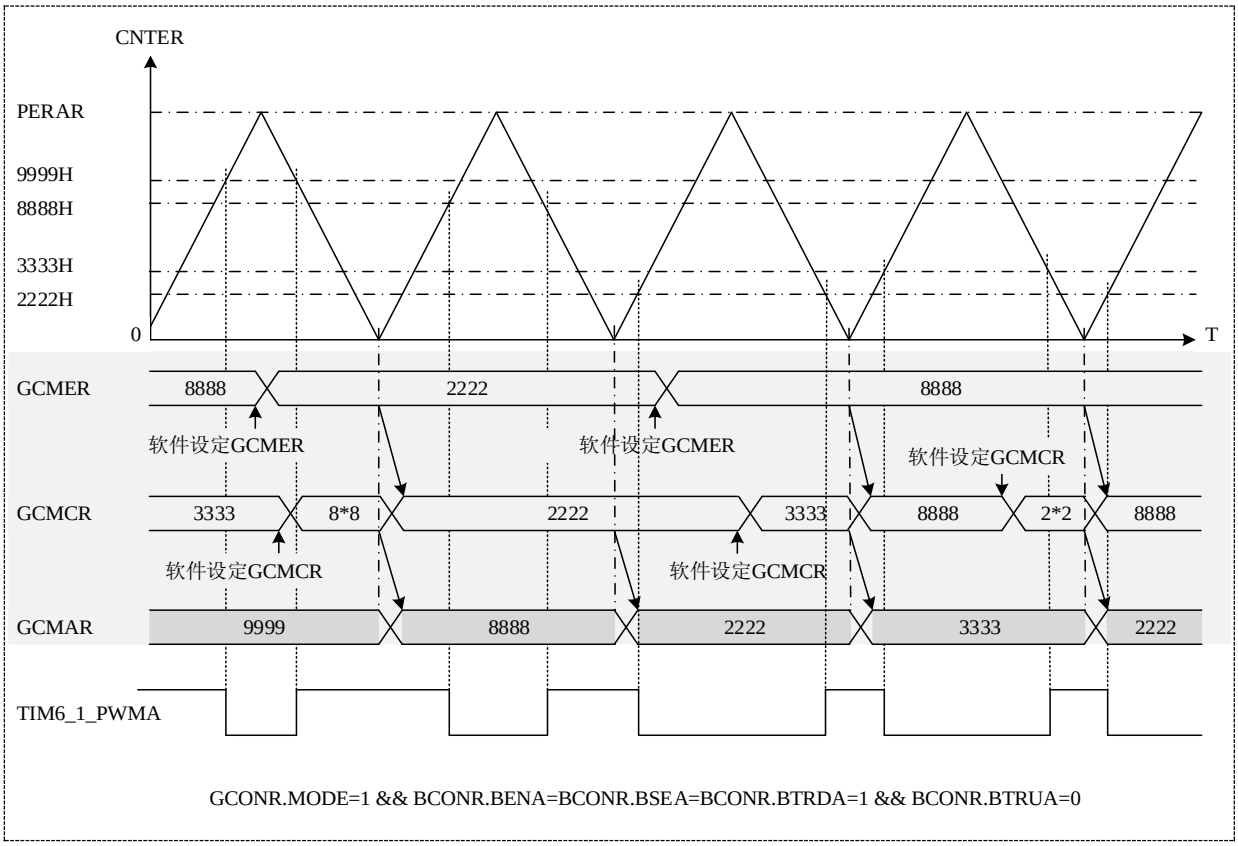


图 20-14 三角波模式时计数缓存动作 1

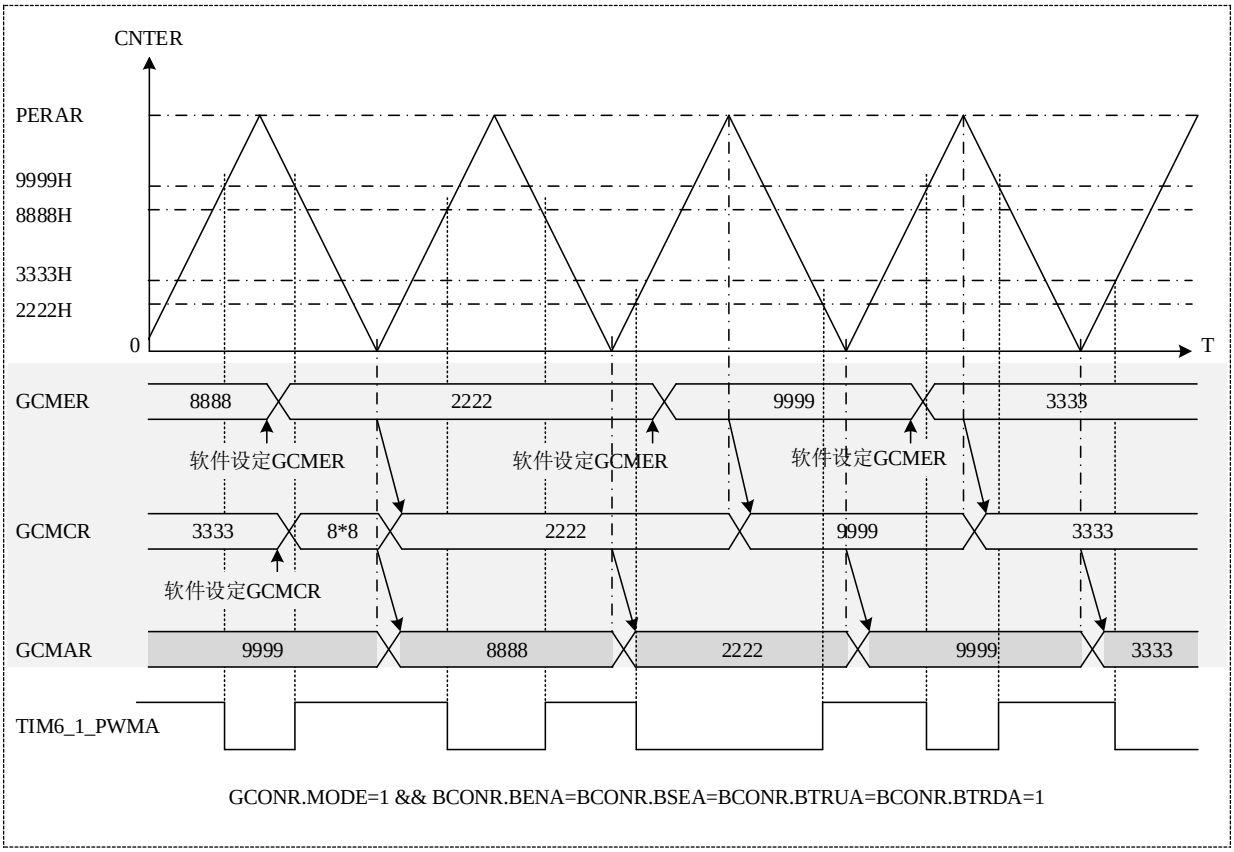


图 20-15 三角波模式时计数缓存动作 2

捕获输入缓存传送

在捕获输入动作（PCNA<B>R.CAPMDA<B>=1）有效时，通用比较基准值支持缓存功能。若缓存设定有效（BCONR.BENA<B>=1），在捕获输入动作点会发生一次缓存传送。捕获输入动作可选择单缓存功能或双缓存功能（由 BCONR.BSEA<B>设定）。

20.3.12 数字滤波

TIM6_<t>_PWMA、TIM6_<t>_PWMB、TIM6_TRIGA~D 端口输入都具有数字滤波功能。TIM6_<t>_PWMA、TIM6_<t>_PWMB 通过设定滤波控制寄存器（FCNGR）的相关使能位开启对应端口的滤波功能，滤波有效时的滤波基准时钟也可通过滤波控制寄存器（FCNGR）设定；TIM6_TRIGA~D 端口是一组单元间共用的端口通过设定滤波控制寄存器（FCNTR）的相关使能位开启对应端口的滤波功能，滤波有效时的滤波基准时钟也可通过滤波控制寄存器（FCNTR）设定。

在滤波采样基准时钟采样到端口上 3 次一致的电平时，该电平被当作有效电平传送到模块内部；小于 3 次一致的电平会被当作外部干扰滤掉，不传送到模块内部。其动作例如图 20-16 所示。

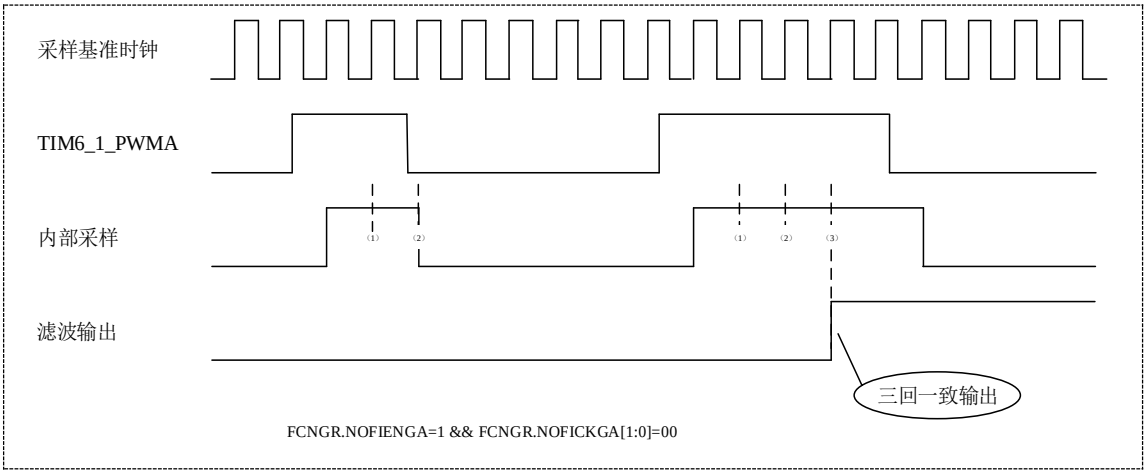


图 20-16 捕获输入端口的滤波功能

20.3.13 通用 PWM 输出

20.3.13.1 单边对齐独立 PWM 输出

在锯齿波计数模式（GCONR.MODE=0）时，每个单元的 2 个端口 TIM6_<t>_PWMA、TIM6_<t>_PWMB 都能独立的输出 PWM 波。在计数溢出点若设定同样的电平变化，则可实现单边对齐独立 PWM 输出。如下图 20-17 所示。

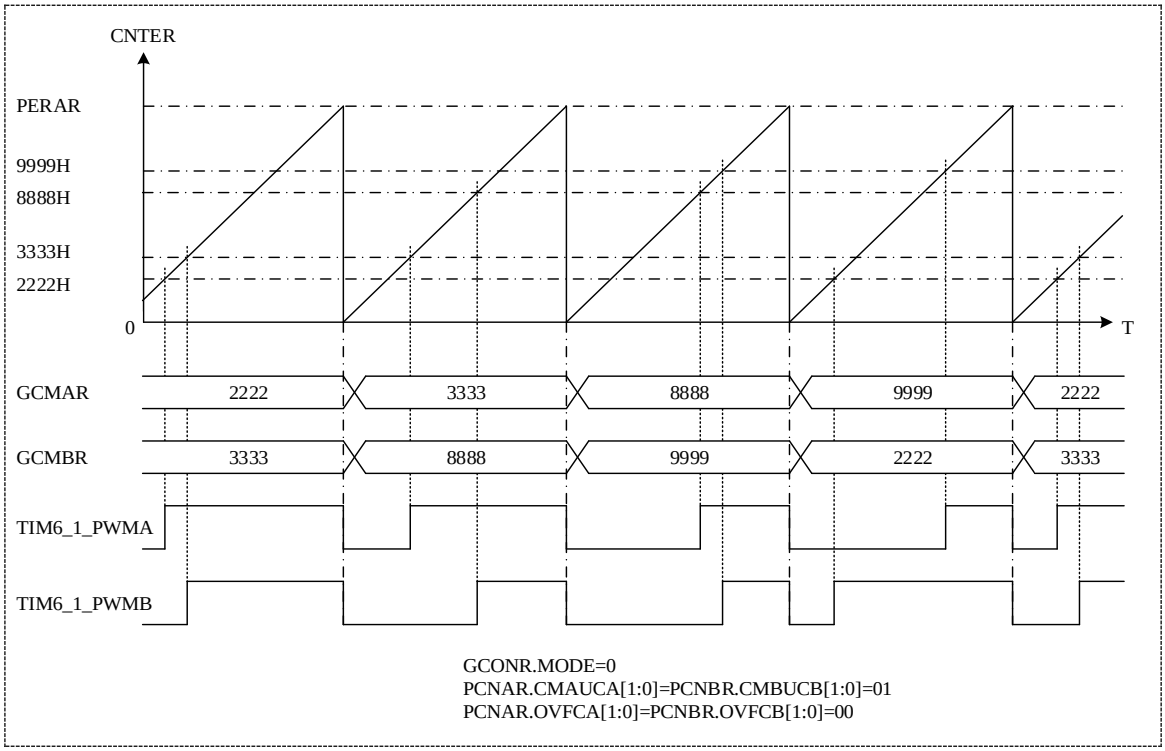


图 20-17　单边对齐独立 PWM

20.3.13.2 双边对称独立 PWM 输出

在三角波计数模式（GCONR.MODE=1）时，每个单元的 2 个端口 TIM6_<t>_PWMA、TIM6_<t>_PWMB 都能独立的输出 PWM 波。在计数比较匹配点若设定电平变化、在计数峰点和谷点设定电平不变化，则可实现双边对称独立 PWM 输出。如图 20-18 所示，是 TIM6_<t>_PWMA 端口和 TIM6_<t>_PWMB 端口实现双边对称独立输出 PWM。

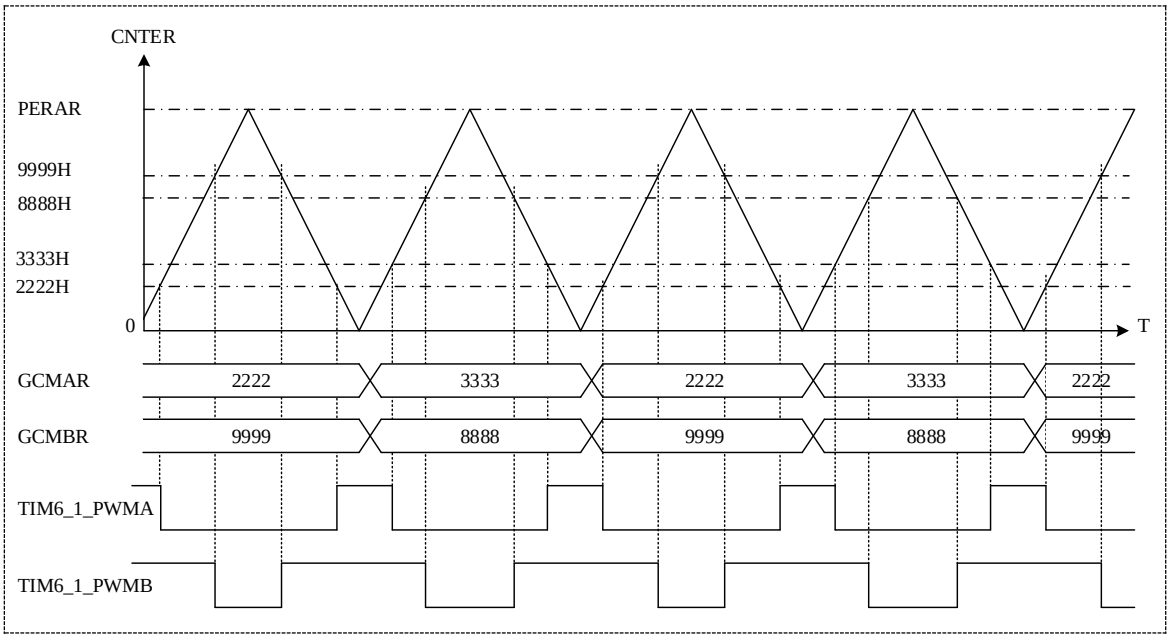


图 20-18　双边对称独立 PWM

20.3.13.3 双边对称互补 PWM 输出

在三角波计数模式（GCONR.MODE=1）时，设定好端口在计数启动、比较匹配、计数溢出时的电平状态，可实现 TIM6_<t>_PWMA 和 TIM6_<t>_PWMB 端口上输出一对互补 PWM 波形。

根据 GCMBR 基准值的赋值方式不同，双边对称互补 PWM 的输出可分为“软件设定 GCMBR 互补 PWM 输出”和“硬件设定 GCMBR 互补 PWM 输出”。

软件设定GCMBR互补PWM输出

软件设定方式是指在三角波模式下，用于 TIM6_<t>_PWMB 端口波形输出的通用比较基准值寄存器（GCMBR）直接由 CPU 等写入，与 GCMAR 的值没有直接关系。

图 20-19 为软件设定 GCMBR 互补 PWM 波的输出例。

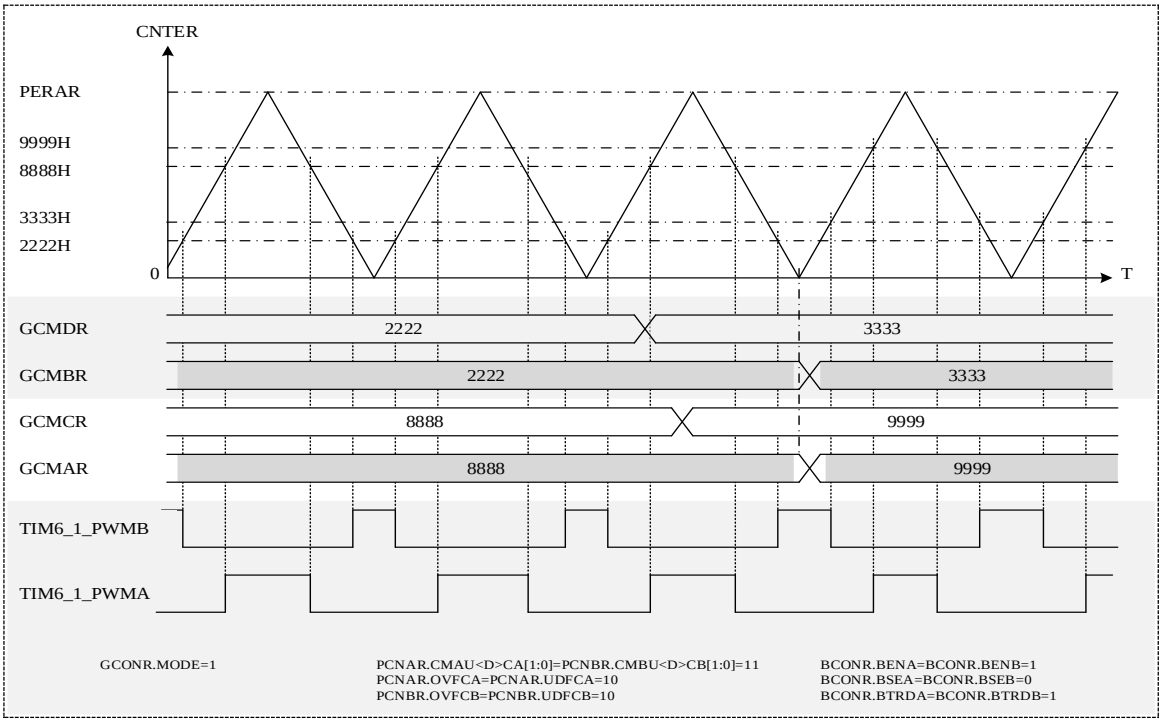


图 20-19 软件设定 GCMBR 互补 PWM 波输出

硬件设定GCMBR互补PWM输出

硬件设定方式是指在三角波模式下，用于 TIM6_<t>_PWMB 端口波形输出的通用比较基准值寄存器（GCMBR）的值由通用比较基准值寄存器（GCMAR）和死区时间基准值寄存器（DTU<D>AR）的值运算决定。

图 20-20 为硬件设定 GCMBR 互补 PWM 波输出例。

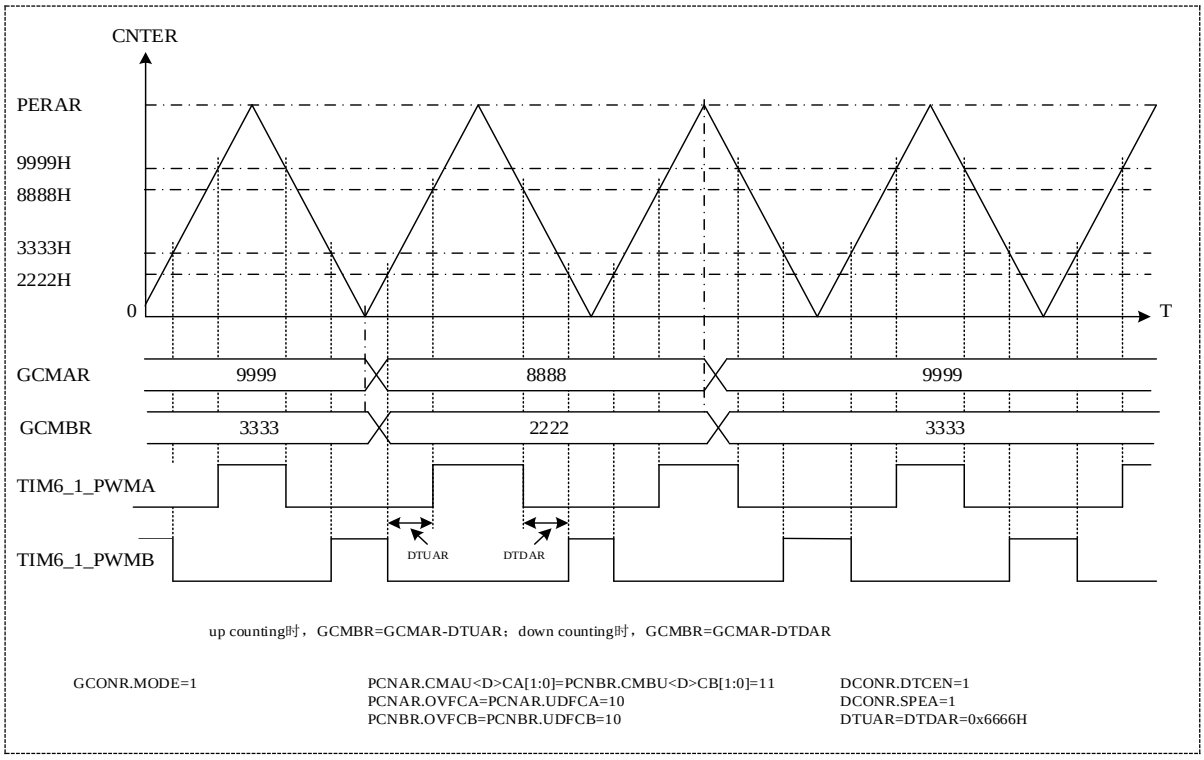


图 20-20　硬件设定 GCMBR 互补 PWM 波输出

20.3.13.4 双边非对称 PWM 输出

在三角波计数模式（GCONR.MODE=1）时，每个单元的 2 个端口 TIM6_<t>_PWMA、TIM6_<t>_PWMB 都能独立的输出 PWM 波，且每个端口都能根据通用比较基准值寄存器（GCMAR、GCMBR）的基准进行对应的电平输出变化。若在计数器的向上计数期间和向下计数期间，TIM6_<t>_PWMA 的端口电平变化分别由 GCMAR、GCMBR 基准值的比较结果控制，则在 TIM6_<t>_PWMA 端口上可实现双边非对称 PWM 输出。如图 20-21 所示，是 TIM6_<t>_PWMA 端口的双边非对称 PWM 输出。

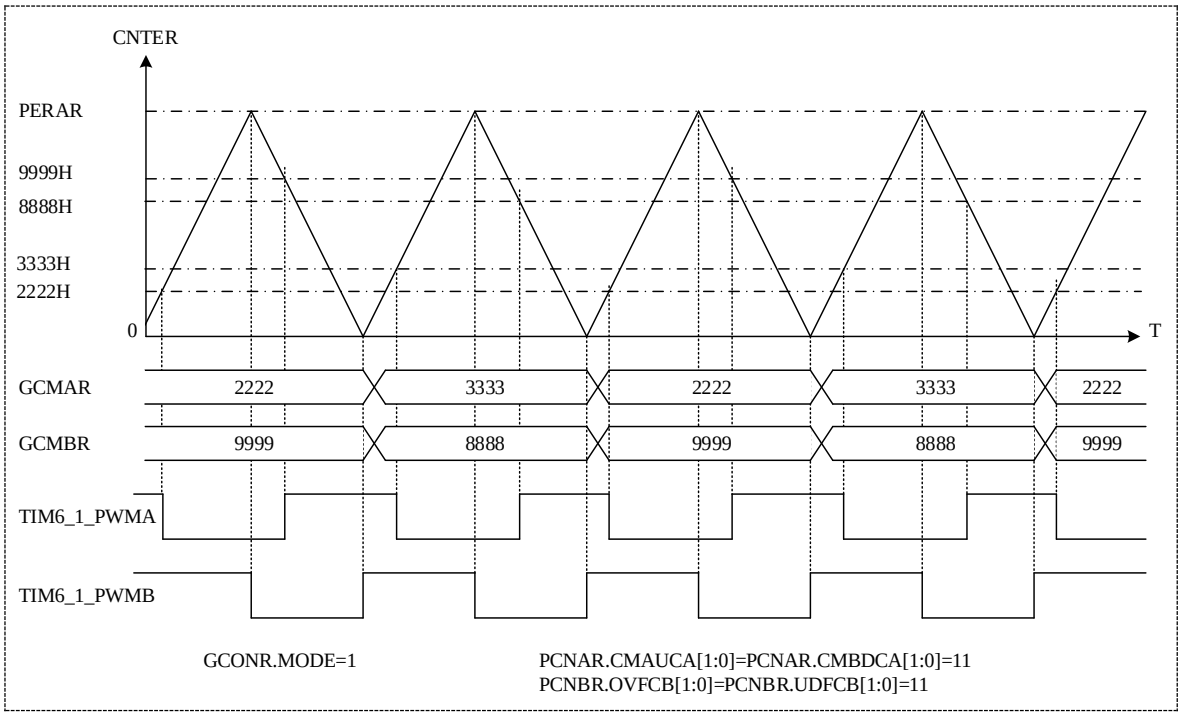


图 20-21 双边非对称 PWM 输出

20.3.13.5 单元间多相 PWM 输出

每个单元的 TIM6_<t>_PWMA、TIM6_<t>_PWMB 端口都能输出 2 相 PWM 波，多个单元间组合，同时结合软件、硬件同步动作就可实现多相 PWM 波输出。如图 20-22，单元 1、单元 2、单元 3 组合输出 6 相单边对齐独立 PWM；如图 20-23，单元 1、单元 2、单元 3 组合输出 3 相双边对称互补 PWM。

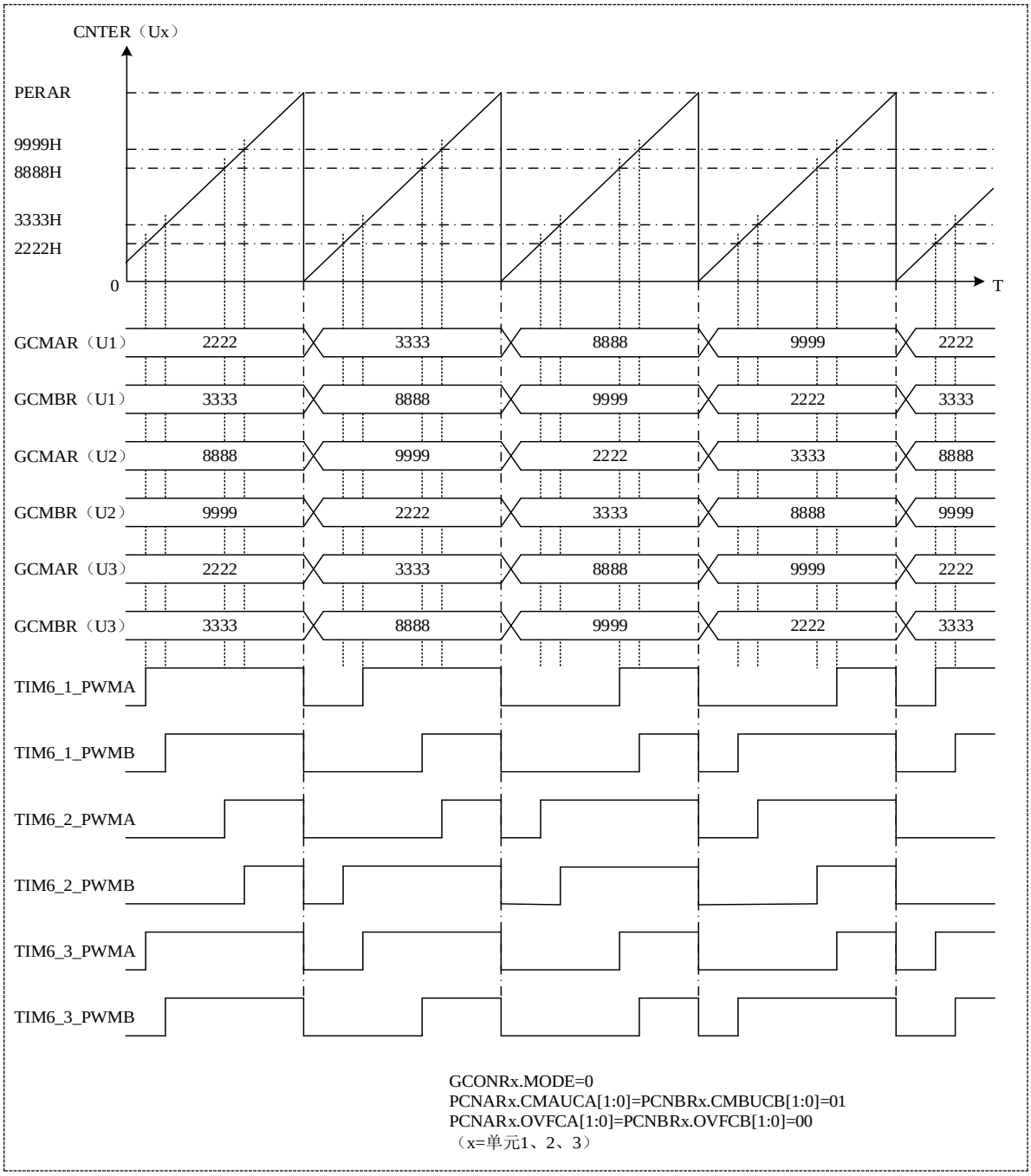


图 20-22　6 相单边对齐独立 PWM

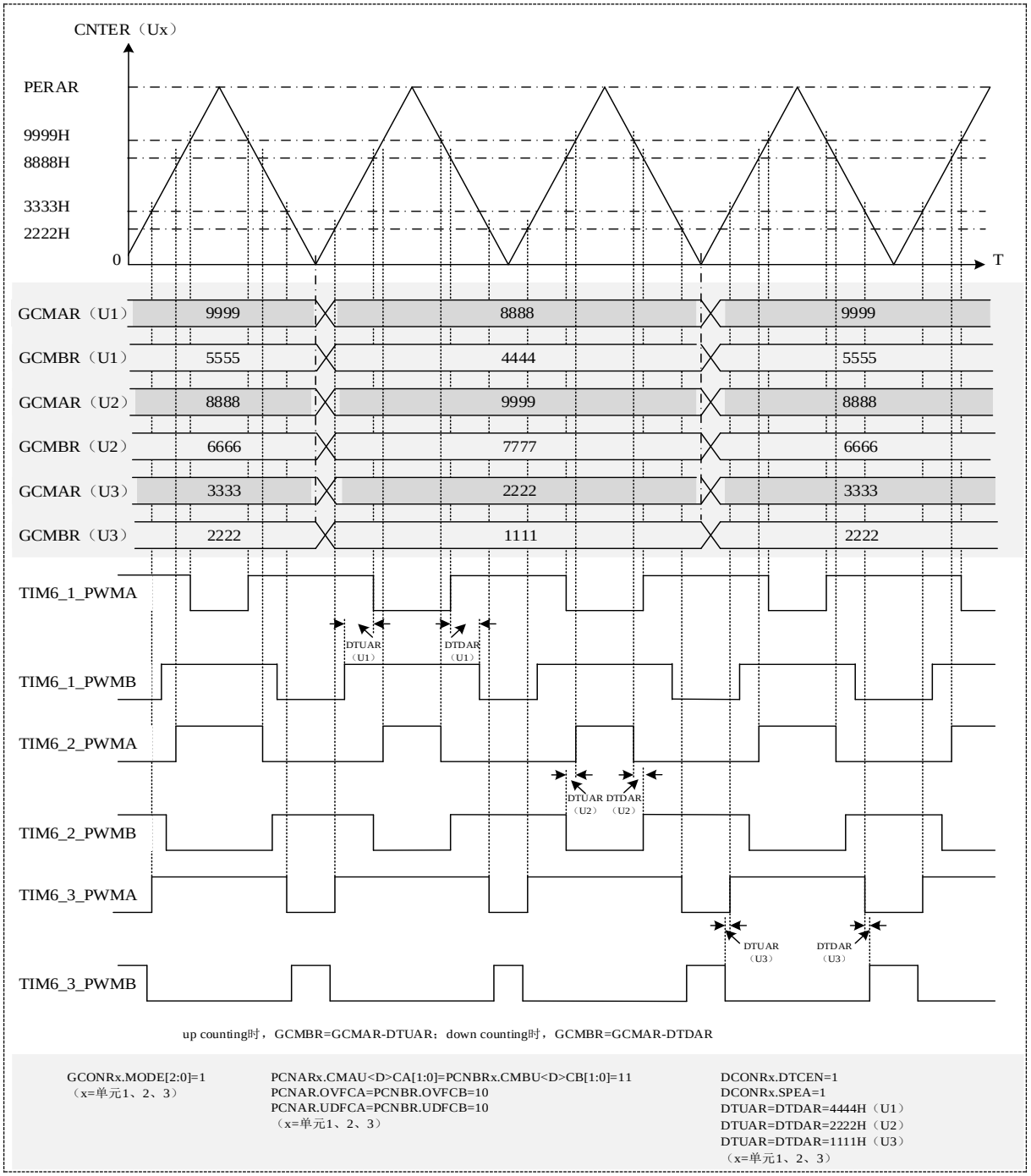


图 20-23　带死区时间 3 相双边对称互补 PWM

20.3.14 周期间隔响应

Timer6 的 2 个专用比较基准值寄存器（SCMAR、SCMBR），在计数比较匹配时可分别输出专用比较匹配中断 A 信号、专用比较匹配中断 B 信号到 INTC 产生对应的中断；同时可分别输出专用比较匹配事件 A 信号、专用比较匹配事件 B 信号，用于和其它模块关联动作，多用于启动 ADC 等。

该中断和事件的请求信号可以每间隔几个周期后产生一次有效的请求信号,即实现周期间隔响应。该功能通过设定有效周期寄存器（VPERR）的 VPERR.PCNTE[1:0]位和 VPERR.SPPERIA/B 位使能。设定 VPERR.PCNTS[2:0]位来指定每隔多少个周期请求信号有效一次，其它周期内即使计数值和专用比较基准值寄存器 SCMAR 或 SCMBR 的值相等，也不会输出有效的请求信号。

该功能有效后，各波形模式下的周期匹配中断和周期匹配事件也只在专用比较匹配中断和事件输出的有效周期里（下图中 STFLR.VPERNUM=0 的周期）输出。图 20-24 所示是周期间隔有效请求信号的动作例。

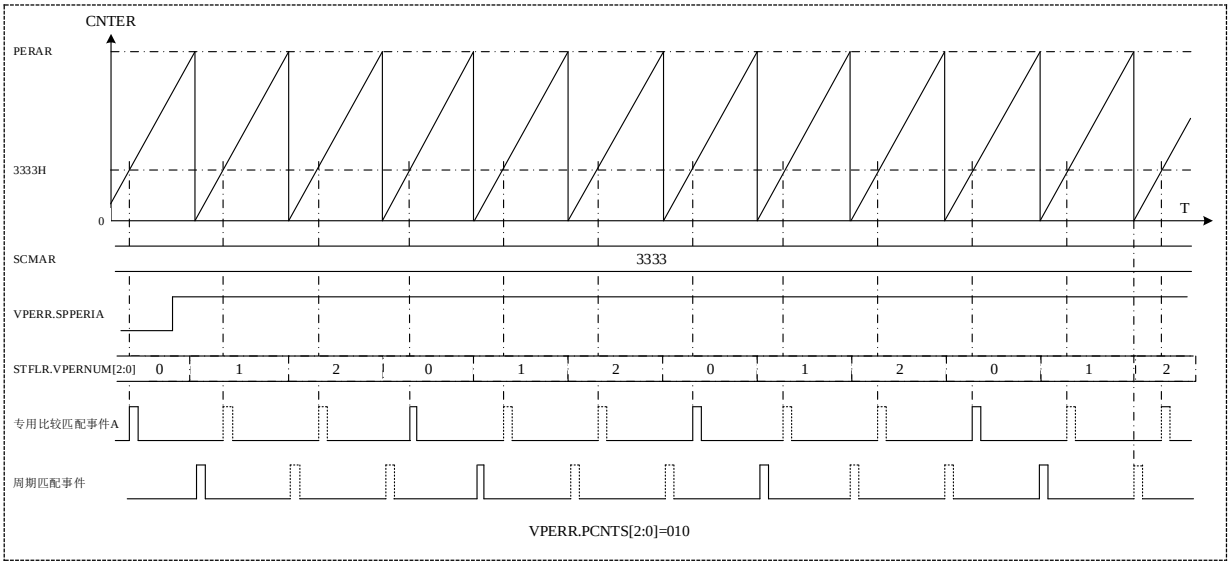


图 20-24 周期间隔有效请求信号动作

20.3.15 正交编码计数

将 TIM6_<t>_PWMA 输入看作 AIN 输入、TIM6_<t>_PWMB 输入看作 BIN 输入、TIM6_TRIGA~D 中的任意一个输入看作 ZIN 输入，Timer6 就可以实现三路输入的正交编码计数。

一个单元的 AIN、BIN 单独动作可以实现位置计数模式；两个单元的 AIN、BIN、ZIN 组合动作可以实现公转计数模式，一个单元用于位置计数，一个单元用于公转计数。公转计数模式时，Timer6 的 8 个单元可任意组合选择做位置计数单元或公转计数单元，但在 Z 相屏蔽功能有效时组合固定，单元 1、2 组合，单元 1 作为位置计数单元，单元 2 作为公转计数单元，分别实现位置计数和公转计数；单元 3、4 组合，单元 3 作为位置计数单元，单元 4 作为公转计数单元，分别实现位置计数和公转计数；单元 5、6 组合，单元 5 作为位置计数单元，单元 6 作为公转计数单元，分别实现位置计数和公转计数；单元 7、8 组合，单元 7 作为位置计数单元，单元 8 作为公转计数单元，分别实现位置计数和公转计数。

AIN 和 BIN 的计数条件是通过设定硬件递加事件选择寄存器（HCUPR）和硬件递减事件选择寄存器（HCDOR）中 TIM6_<t>_PWMA 和 TIM6_<t>_PWMB 的正交关系实现；ZIN 的输入动作通过设定位置单元的硬件清零事件选择寄存器（HCLRR）实现位置计数单元的位置定时器清零、通过设定公转单元的硬件递加事件选择寄存器（HCUPR）实现公转计数单元的公转定时器计数。

20.3.15.1 位置计数模式

正交编码位置计数模式，是指根据 AIN、BIN 的输入实现基本计数功能、相位差计数功能和方向计数功能。

基本计数

基本计数动作是根据 AIN 或 BIN 端口的输入时钟进行计数，如下图 20-25 所示。

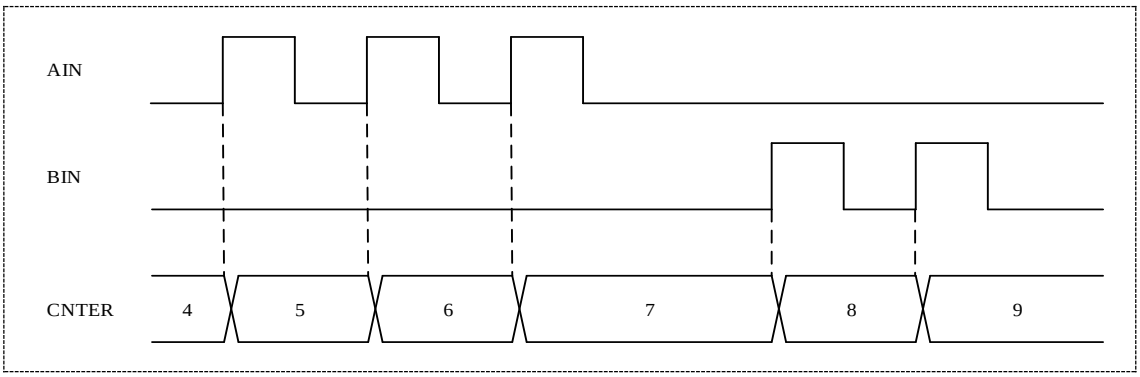


图 20-25 位置模式-基本计数

相位差计数

相位差计数是指根据 AIN 和 BIN 的相位关系进行计数。根据设定的不同，可以实现 1 倍计数、2 倍计数、4 倍计数等，如下图 20-26~图 20-28 所示。

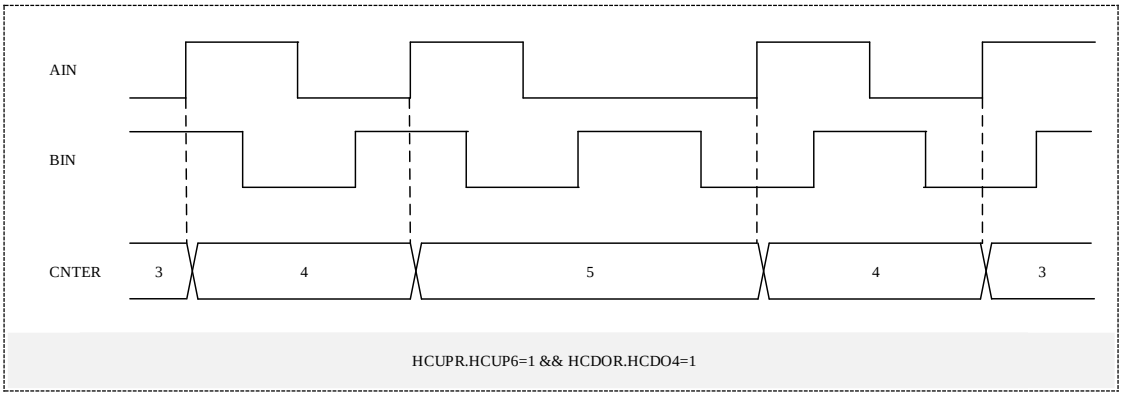


图 20-26 位置计数模式-相位差计数（1 倍计数）

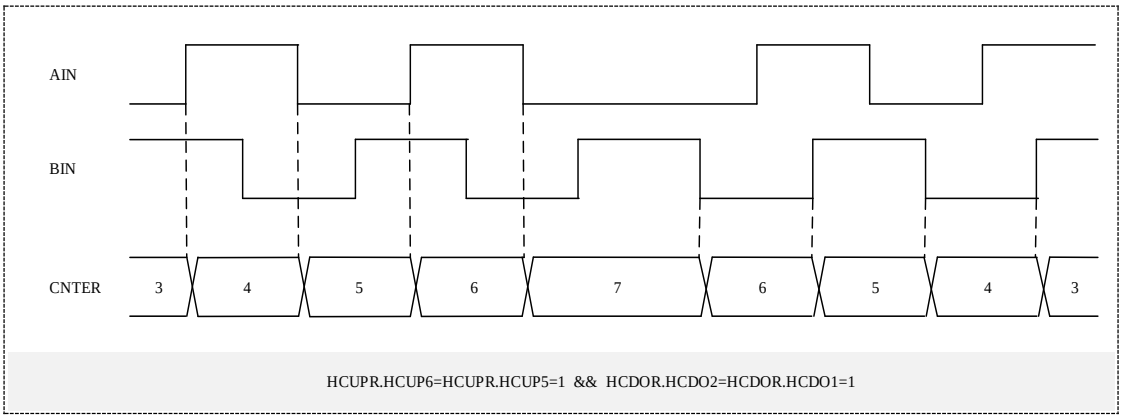


图 20-27 位置计数模式-相位差计数（2 倍计数）

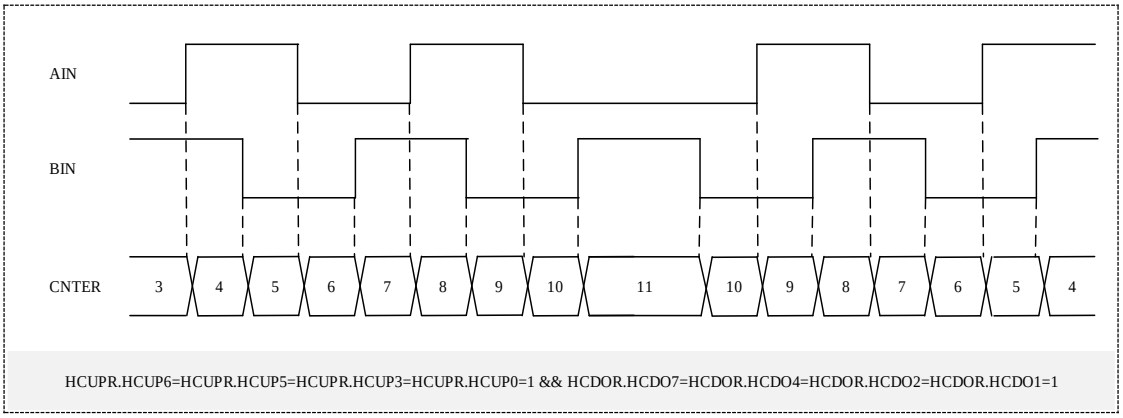


图 20-28 位置计数模式-相位差计数（4 倍计数）

方向计数

方向计数是指将 AIN 的输入状态设定为方向控制，将 BIN 的输入作为时钟计数，如下图 20-29 所示。

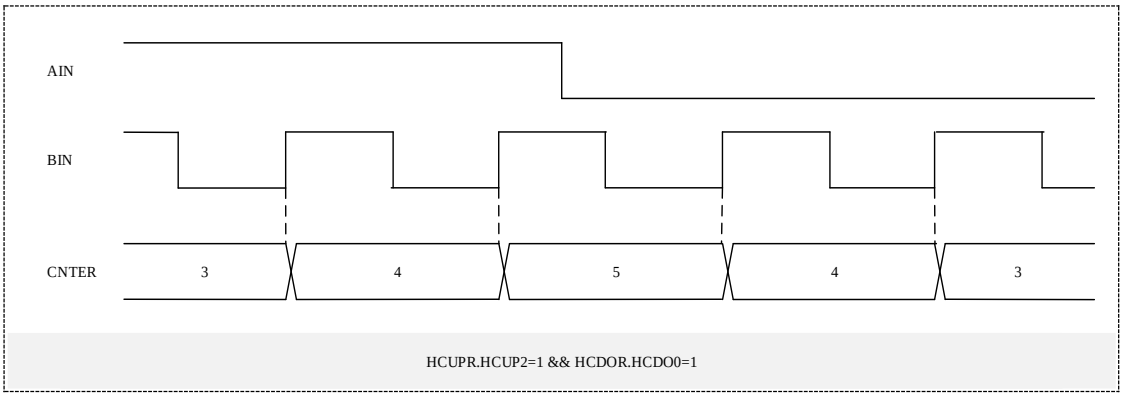


图 20-29 位置计数模式-方向计数

20.3.15.2 公转计数模式

正交编码公转计数模式，是指在 AIN、BIN 计数的基础上，加入 ZIN 的输入事件以实现对公转圈数等的判断。公转计数模式时根据公转定时器的计数方式，可实现 Z 相计数功能、位置溢出计数功能和混合计数功能。

Z相计数

Z 相计数是指根据 ZIN 的输入，公转计数单元进行计数，同时将位置计数单元清零的计数动作。如下图 20-30 所示。

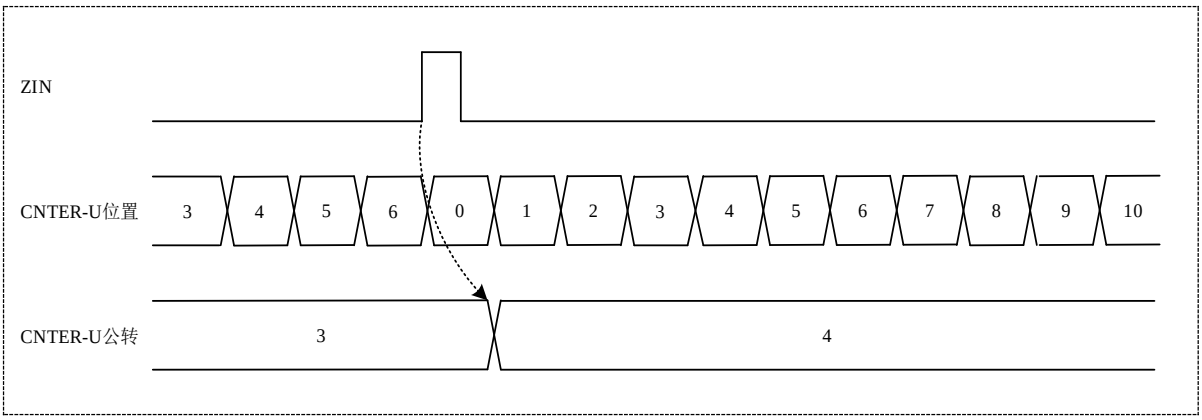


图 20-30 公转计数模式-Z 相计数

位置溢出计数

位置溢出计数是指位置计数单元计数发生上溢或下溢时，产生一个溢出事件，从而触发公转计数单元的定时器进行一次计数（在该计数方式时 ZIN 的输入不进行公转计数单元的计数动作和位置计数单元的清零动作）。

位置计数单元的溢出事件通过内部触发事件接口选通实现公转计数单元计数，即可实现位置溢出计数。公转计数单元的硬件递加（递减）事件选择寄存器（HCUPR 或 HCDOR）的递加（递减）事件选择 Bit8~Bit11 中的 1 位，同时将对应的硬件触发事件选择寄存器（HTSSR0~3）中的事件编号设定为位置计数单元的上溢或下溢事件。具体事件编号参考 INTC 章节。如下图 20-31 所示。

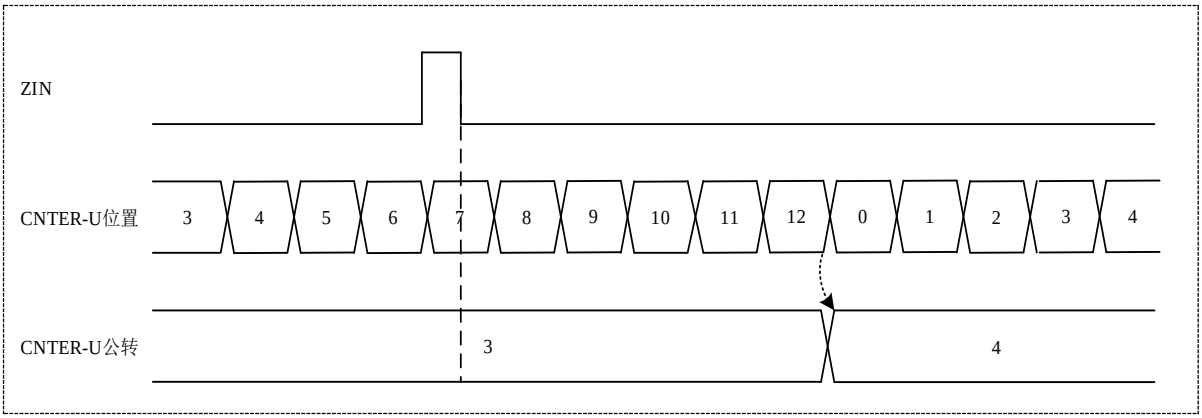


图 20-31 公转计数模式-位置溢出计数

混合计数

混合计数是指上述 Z 相计数和位置溢出计数两种计数方式合并起来的计数动作，其实现方式也是上述两种计数方式的组合。如下图 20-32 所示。

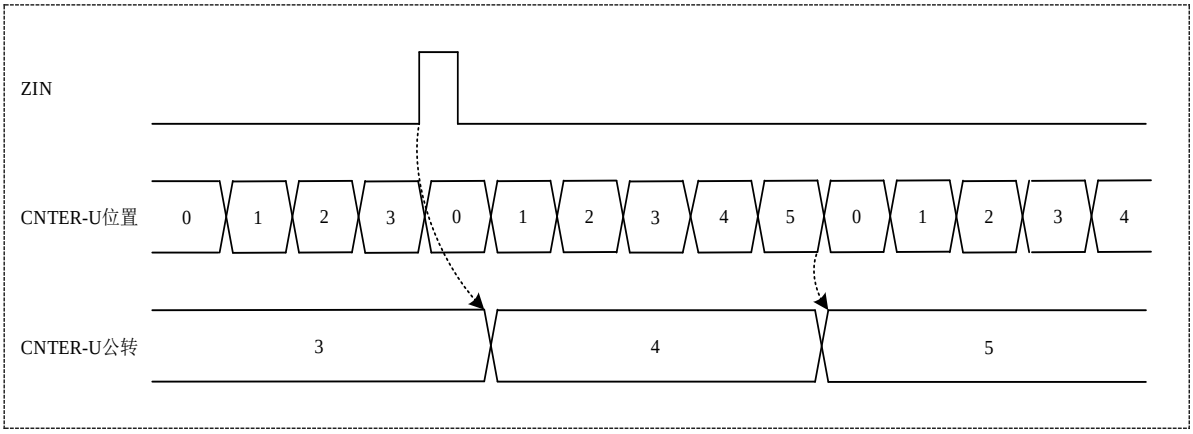


图 20-32 公转计数模式-混合计数

20.3.15.3 Z 相动作屏蔽

在公转计数模式的 Z 相计数功能或混合计数功能时，可以设定在位置定时器的上溢点或下溢点后的几个周期内（GCONR.ZMSKVAL[0:1]设定），将 ZIN 的有效输入屏蔽，不进行公转计数单元的计数和位置计数单元的清零。

位置计数单元的通用控制寄存器（GCONR）的 GCONR.ZMSKPOS 为 1 时，位置计数单元的 Z 相屏蔽功能使能，Z 相屏蔽的周期数由 GCONR.ZMSKVAL 设定；公转计数单元的通用控制寄存器（GCONR）的 GCONR.ZMSKREV 为 1 时，公转计数单元的 Z 相屏蔽功能使能。

图 20-33 是公转计数模式混合计数时，在位置计数单元计数上溢后的 4 个计数周期内有 ZIN 相输入时，ZIN 相输入的动作无效，即公转计数单元不计数、位置计数单元不清零；之后再来的 ZIN 相输入正常动作。

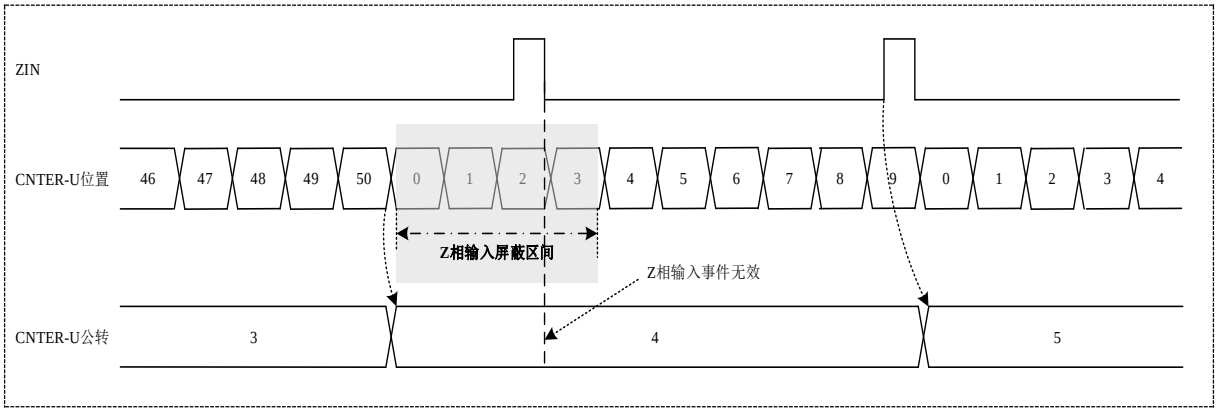


图 20-33 公转计数模式-混合计数 Z 相屏蔽动作例 1

图 20-34 是公转计数模式混合计数时，在位置计数单元计数上溢后的第 3 个周期，计数方向发生变化，此时设定的 4 个周期的屏蔽周期变为无效（实际 ZIN 相屏蔽功能维

持了 3 个周期），开始向下计数。在位置计数单元发生计数下溢后，ZIN 相屏蔽功能重新开启，维持 4 个周期后变为无效。在 ZIN 相屏蔽期间，ZIN 相的输入功能无效，即公转计数单元不计数、位置计数单元不清零；之后再来的 ZIN 相输入正常动作。

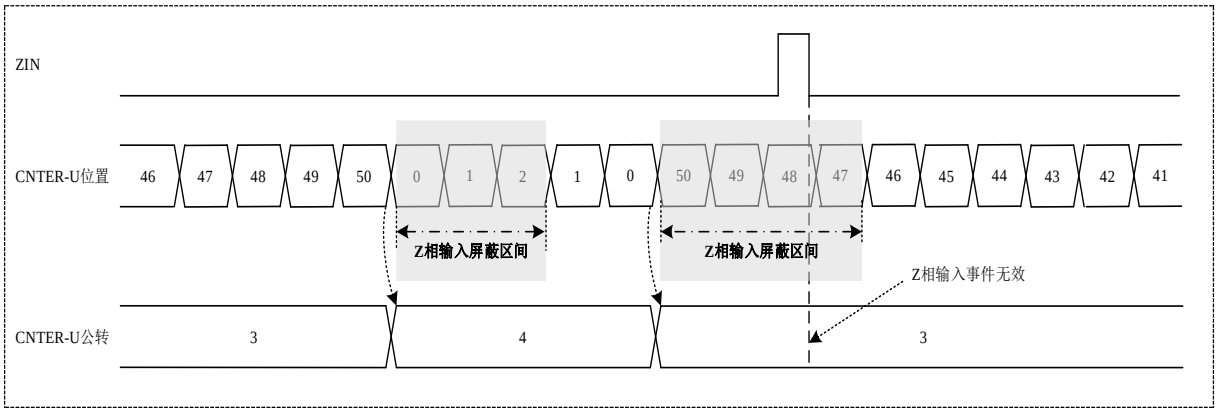


图 20-34　公转计数模式-混合计数 Z 相屏蔽动作例 2

20.3.16 EMB 控制

Timer6 可以对端口的输出状态进行保护控制，在发生异常时将端口状态固定成预先设定好的安全状态。所有单元具有 4 个共用的端口输出控制接口，每个单元通过端口控制寄存器（PCNA<B>R.EMBSA<B>）的设定选择要使用的 EMB 事件连接接口，这个接口连接 EMB 模块输出的 1 组 EMB 事件。同时接口上选通的异常状况事件可从 EMB 侧设定（参见 EMB 章节），当这些接口上监测到异常状况时，可以实现对通用 PWM 输出的控制。

端口在正常输出期间，若监测到从 EMB 过来的 EMB 事件，则端口的输出状态可变为预先设定好的状态。通用 PWM 输出端口在 EMB 异常事件发生时，端口状态可以变为输出高阻态、输出低电平或输出高电平（根据 PCNA<B>R.EMBCA<B>的设定决定）。例如，若 PCNAR.EMBCA=01 设定时，则在 TIM6_<t>_PWMA 端口正常输出期间，若产生 EMB 事件，则 TIM6_<t>_PWMA 端口上输出变为高阻态。

在 EMB 事件无效后（从 EMB 模块连接到 Timer6 的异常事件消失，信号变为正常电平），PWM 端口的输出可以自动恢复到正常的输出。此时，可以通过端口控制寄存器的设定选择立即恢复 PWM 正常输出（PCNA<B>R.EMBRA<B>=00；这种方式称之为 One Shot 方式释放）或在计数到下一个溢出点之后再恢复 PWM 的正常输出（PCNA<B>R.EMBRA<B>=01、10、11；这种方式称之为 Cycle By Cycle 方式释放）。

20.3.17 典型应用例

下面描述几种典型应用情况下，Timer6 相关寄存器的基本设定，供用户参考。

20.3.17.1 基本计数及中断动作

a) 设定通用周期基准值（PERAR）

b) 设定需要的比较基准值，包括通用比较基准值（GCMAR~GCMFR）、专用比较基准值（SCMAR~SCMBR）等

c) 设定需要的中断使能位，包括计数上溢中断（ICONR.INTENOVF）、计数下溢中断（ICONR.INTENUDF）、计数匹配中断（ICONR.INTENA~F、ICONR.INTENSAU、ICONR.INTENSAD、ICONR.INTENSBU、ICONR.INTENSBD）等

d) 设定内部计数时钟分频（GCONR.CKDIV[3:0]）

e) 设定波形模式（GCONR.MODE）

f) 设定计数方向（仅在锯齿波模式 GCONR.MODE=0 时需要设定）

g) 启动计数器（GCONR.START=1）

20.3.17.2 比较输出及中断动作

a) 设定通用周期基准值（PERAR）

b) 设定各通道的比较基准值，包括通用比较基准值 A（GCMAR）、通用比较基准值 B（GCMBR）

c) 设定需要的中断使能位，包括计数上溢中断（ICONR.INTENOVF）、计数下溢中断（ICONR.INTENUDF）、计数匹配中断（ICONR.INTENA~B）等

d) 设定各通道在不同计数状态时的端口输出状态（参考 PCNAR 或 PCNBR 的 bit17~bit0 相关控制）

e) 设定内部计数时钟分频（GCONR.CKDIV[3:0]）

f) 设定波形模式（GCONR.MODE）

g) 设定计数方向（仅在锯齿波模式 GCONR.MODE=0 时需要设定）

h) 设定各通道比较输出模式（PCNAR.CAPMDA=0、PCNBR.CAPMDB=0）

i) 设定各通道输出使能（PCNAR.OUTENA=1、PCNBR.OUTENB=1）

j) 启动计数器（GCONR.START=1）

20.3.17.3 捕获输入及中断动作

a) 设定通用周期基准值（PERAR）

b) 设定需要的中断使能位，包括计数上溢中断（ICONR.INTENOVF）、计数下溢中断（ICONR.INTENUDF）、捕获输入中断（ICONR.INTENA~B）等

c) 设定各通道的捕获输入外部条件（参考 HCPAR 或 HCPBR 的全部有效控制位。该有效控制位互相独立，可以同时选择多个均作为某通道的捕获输入条件）

d) 设定内部计数时钟分频（GCONR.CKDIV[3:0]）

e) 设定波形模式（GCONR.MODE）

f) 设定计数方向（仅在锯齿波模式 GCONR.MODE=0 时需要设定）

g) 设定捕获输入模式（PCNAR.CAPMDA=1、PCNBR.CAPMDB=1）

h) 启动计数器（GCONR.START=1）

i) 等待捕获输入条件产生，读取对应通道的捕获输入值（GCMAR 或 GCMBR）或等待相应的中断产生

20.3.17.4 缓存传送动作（周期基准值）

a) 设定需要的通用周期基准值（PERAR、PERBR、PERCR）

b) 设定单、双缓存传送方式（BCONR.BSEP）

c) 设定缓存传送时间点（BCONR.BTRUP、BCONR.BTRDP，这两个控制位互相独立，可以同时选择，均作为缓存传送时间点）（该设定仅在三角波模式时有效，在锯齿波模式时无效）

d) 设定内部计数时钟分频（GCONR.CKDIV[3:0]）

e) 设定波形模式（GCONR.MODE）

f) 设定计数方向（仅在锯齿波模式 GCONR.MODE=0 时需要设定）

g) 设定缓存功能有效（BCONR.BENP=1）

h) 启动计数器（GCONR.START=1）

i) 等待对应的缓存传送时间点，发生缓存动作（PERBR->PERAR（BCONR.BSEP=0 时）、PERCR->PERBR->PERAR（BCONR.BSEP=1 时））

20.3.17.5 缓存传送动作（通用比较基准值）

a) 设定需要的通用比较基准值（GCMAR、GCMCR、GCMER、GCMBR、GCMDR、GCMFR）

b) 设定各通道单、双缓存传送方式（BCONR.BSEA、BCONR.BSEB）

c) 设定各通道缓存传送时间点（BCONR.BTRUA、BCONR.BTRDA、BCONR.BTRUB、BCONR.BTRDB，每个通道的两个控制位互相独立，可以同时选择，均作为缓存传送时间点）（该设定仅在三角波模式时有效，在锯齿波模式时无效）

d) 设定内部计数时钟分频（GCONR.CKDIV[3:0]）

e) 设定波形模式（GCONR.MODE）

f) 设定计数方向（仅在锯齿波模式 GCONR.MODE=0 时需要设定）

g) 设定各通道缓存功能有效（BCONR.BENA=1、BCONR.BENB=1）

h) 启动计数器（GCONR.START=1）

i) 等待各通道所设定的对应缓存传送时间点，发生缓存动作（GCMCR->GCMAR（BCONR.BSEA=0 时）、GCMER->GCMCR->GCMAR（BCONR.BSEA=1 时）、GCMDR->GCMBR（BCONR.BSEB=0 时）、GCMFR->GCMDR->GCMBR（BCONR.BSEB=1 时））

20.3.17.6 缓存传送动作（专用比较基准值）

a) 设定需要的专用比较基准值（SCMAR、SCMCR、SCMER、SCMBR、SCMDR、SCMFR）

b) 设定各通道单、双缓存传送方式（BCONR.BSESPA、BCONR.BSESPB）

c) 设定各通道缓存传送时间点（BCONR.BTRUSPA、BCONR.BTRDSPA、BCONR.BTRUSPB、BCONR.BTRDSPB，每个通道的两个控制位互相独立，可以同时选择，均作为缓存传送时间点）（该设定仅在三角波模式时有效，在锯齿波模式时无效）

d) 设定内部计数时钟分频（GCONR.CKDIV[3:0]）

e) 设定波形模式（GCONR.MODE）

f) 设定计数方向（仅在锯齿波模式 GCONR.MODE=0 时需要设定）

g) 设定各通道缓存功能有效（BCONR.BENSPA=1、BCONR.BENSPB=1）

h) 启动计数器（GCONR.START=1）

i) 等待各通道所设定的对应缓存传送时间点，发生缓存动作（SCMCR->SCMAR（BCONR.BSESPA=0 时）、SCMER->SCMCR->SCMAR（BCONR.BSESPA=1 时）、SCMDR->SCMBR （ BCONR.BSESPB=0 时 ）、 SCMFR->SCMDR->SCMBR（BCONR.BSESPB=1 时））

20.3.17.7 缓存传送动作（死区基准值）

a) 设定需要的死区时间基准值（DTUAR、DTUBR、DTDAR、DTDBR）

b) 设定缓存传送时间点（DCONR.DTBTRU、DCONR.DTBTRD，这两个控制位互相独立，可以同时选择，均作为缓存传送时间点）（该设定仅在三角波模式时有效，在锯齿波模式时无效）

c) 设定内部计数时钟分频（GCONR.CKDIV[3:0]）

d) 设定波形模式（GCONR.MODE）

e) 设定计数方向（仅在锯齿波模式 GCONR.MODE=0 时需要设定）

f) 设定缓存功能有效（DCONR.DTBENU=1、DCONR.DTBEND=1）

g) 设定硬件死区功能有效（DCONR.DTCEN=1）

h) 启动计数器（GCONR.START=1）

i) 等待对应缓存传送时间点，发生缓存动作（DTUBR->DTUAR、DTDBR->DTDAR）

20.3.17.8 同步启动动作（软件方式）

a) 参考[基本计数及中断动作]章节的 a~f 步骤，对需要同步启动的各个单元做设定

b) 同步启动计数器（设定 SSTAR 寄存器的对应位为 1，每个单元对应一个寄存器位）

20.3.17.9 同步启动动作（硬件方式）

a) 设定通用周期基准值（PERAR）

b) 设定需要的比较基准值，包括通用比较基准值（GCMAR~GCMFR）、专用比较基准值（SCMAR~SCMBR）等

c) 设定需要的中断使能位，包括计数上溢中断（ICONR.INTENOVF）、计数下溢中

断（ICONR.INTENUDF）、计数匹配中断（ICONR.INTENA~F、ICONR.INTENSAU、ICONR.INTENSAD、ICONR.INTENSBU、ICONR.INTENSBD）等

d) 设定硬件启动条件（通过 HSTAR.HSTAx 选择，x=8~31）

e) 设定硬件启动使能（HSTAR.STAS=1）

f) 重复上述 a~e 步骤对需要同步启动的各个单元做设定（需要同步启动的各个单元中，步骤 d 的设定要一致）

g) 等待设定的触发事件产生，确认各个单元的计数器同步启动

20.3.17.10 正交编码计数动作（2 相）

a) 设定通用周期基准值（PERAR）

b) 设定需要的比较基准值，包括通用比较基准值（GCMAR~GCMFR）、专用比较基准值（SCMAR~SCMBR）等

c) 设定需要的中断使能位，包括计数上溢中断（ICONR.INTENOVF）、计数下溢中断（ICONR.INTENUDF）、计数匹配中断（ICONR.INTENA~F、ICONR.INTENSAU、ICONR.INTENSAD、ICONR.INTENSBU、ICONR.INTENSBD）等

d) 设定需要的硬件向上计数条件（通过 HCUPR.HCUPx 选择，x=0~7）

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

20.3.17.11 正交编码计数动作（3 相）

a) 参考[正交编码计数动作（2 相）]章节的 a~e 步骤，对位置计数单元做设定

b) 设定位置计数单元的硬件清零条件（通过 HCLRR.HCLRx 选择，x=16~31）

c) 设定位置计数单元硬件清零使能（HCLRR.CLES=1）

d) 设定公转计数单元的通用周期基准值（PERAR）

e) 设定公转计数单元的比较基准值，包括通用比较基准值（GCMAR~GCMFR）、专用比较基准值（SCMAR~SCMBR）等

f) 设定公转计数单元需要的中断使能位，包括计数上溢中断（ICONR.INTENOVF）、计数下溢中断（ICONR.INTENUDF）、计数匹配中断（ICONR.INTENA~F、

ICONR.INTENSAU 、 ICONR.INTENSAD 、 ICONR.INTENSBU 、 ICONR.INTENSBD）等

g) 设定公转计数单元的硬件向上计数条件 1（ZIN 相输入）（通过 HCUPR.HCUPx 选择，x=16~31，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.HCUP8 选择内部硬件触发事件 0）

i) 设定公转计数单元的硬件向下计数条件（位置计数单元的下溢事件输入）（通过 HCDOR.HCDO9 选择内部硬件触发事件 1）

j) 设定 HTSSR0 中的触发源编号为位置计数单元的计数上溢事件（该上溢事件编号参考 INTC 章节）

k) 设定 HTSSR1 中的触发源编号为位置计数单元的计数下溢事件（该下溢事件编号参考 INTC 章节）

l) 启动公转计数单元计数器（GCONR.START=1）

m) 启动位置计数单元计数器（GCONR.START=1）

n) 等待设定的 AIN、BIN、ZIN 相计数事件产生，确认计数器正常计数

20.3.17.12 单路 PWM 输出

a) 参考[比较输出及中断动作]章节 a~j 步骤的设定（每个单元内部的 2 个 PWM 通道 TIM6_<t>_PWMA 和 TIM6_<t>_PWMB 的输出状态均可独立设定，形成 2 个互不相关的单路 PWM 输出）

20.3.17.13 互补 PWM 输出（软件死区）

a) 设定通用周期基准值（PERAR）

b) 设定通用比较基准值 A（GCMAR）、通用比较基准值 B（GCMBR）

c) 设定需要的中断使能位，包括计数上溢中断（ICONR.INTENOVF）、计数下溢中断（ICONR.INTENUDF）、计数匹配中断（ICONR.INTENA~B）等

d) 设定在不同计数状态时的端口输出状态（参考 PCNAR 和 PCNBR 的 bit17~bit0 相关控制，结合 GCMAR 和 GCMBR 的设定值，需要保证 2 个 PWM 输出之间形成互补死区）

e) 设定内部计数时钟分频（GCONR.CKDIV[3:0]）

f) 设定波形模式为三角波模式（GCONR.MODE=1）

g) 设定比较输出模式（PCNAR.CAPMDA=0、PCNBR.CAPMDB=0）

h) 设定输出使能（PCNAR.OUTENA=1、PCNBR.OUTENB=1）

i) 启动计数器（GCONR.START=1）

20.3.17.14 互补 PWM 输出（硬件死区）

a) 设定通用周期基准值（PERAR）

b) 设定通用比较基准值 A（GCMAR）、死区时间基准值（DTUAR、DTDAR）

c) 设定需要的中断使能位，包括计数上溢中断（ICONR.INTENOVF）、计数下溢中断（ICONR.INTENUDF）、计数匹配中断（ICONR.INTENA~B）、死区错误中断（ICONR.INTENDTE）等

d) 设定在不同计数状态时的端口输出状态（参考 PCNAR 和 PCNBR 的 bit17~bit0 相关控制，结合 GCMAR、DTUAR 和 DTDAR 的设定值，需要保证 2 个 PWM 输出之间形成互补死区）

e) 设定内部计数时钟分频（GCONR.CKDIV[3:0]）

f) 设定波形模式为三角波模式（GCONR.MODE=1）

g) 设定各通道比较输出模式（PCNAR.CAPMDA=0、PCNBR.CAPMDB=0）

h) 设定各通道输出使能（PCNAR.OUTENA=1、PCNBR.OUTENB=1）

i) 设定硬件死区功能有效（DCONR.DTCEN=1）

j) 启动计数器（GCONR.START=1）

20.3.17.15 EMB 监控及中断动作

a) 参考[互补 PWM 输出（软件死区）]章节的 a~h 步骤或[互补 PWM 输出（硬件死区）]章节 a~i 步骤，对互补 PWM 输出动作进行设定

b) 设定 EMB 事件发生时 PWM 端口的状态（PCNAR.EMBCA、PCNBR.EMBCB）（根据系统应用的不同，选择相应的保护状态）

c) 设定 EMB 事件变为无效时 PWM 端口恢复正常输出的时间点（PCNAR.EMBRA、PCNBR.EMBRB）

d) 设定选择从 EMB 模块输入的 EMB 事件源（PCNAR.EMBSA、PCNBR.EMBSB）

e) 设定 EMB 模块的相关寄存器（包括 EMB 中断许可寄存器（EMB_INTEN0~3）、EMB 控制寄存器 1/2（EMB_CTL1/2_0~3）、EMB 控制 PWM 输出释放方式选择寄存器（EMB_RLSSEL0~3）等）

f) 启动计数器（GCONR.START=1），EMB 模块实时监控系统状态

20.3.18 功能汇总表

Timer6 的锯齿波模式和三角波 A、B 模式下，主要功能的汇总表如下表 20-3 所示。

PWM输出功能				锯齿波	三角波	相关主要寄存器
独立PWM输出	端口状态控制	启动时		支持	支持	PCNA<B>R.STACA<B>
		停止时		支持	支持	PCNA<B>R.STPCA<B>
		上溢时		支持	支持	PCNA<B>R.OVFCA<B>
		下溢时		支持	支持	PCNA<B>R.UDFCA<B>
		计数匹配时（Up Counting）		支持	支持	PCNA<B>R.CMA<B>UCA<B>
		计数匹配时（Down Counting）		支持	支持	PCNA<B>R.CMA<B>DCA<B>
	缓存传送	周期基准值	单缓存	支持	支持	PERBR->PERAR
			双缓存	支持	支持	PERCR->PERBR PERBR->PERAR
		比较基准值	单缓存	支持	支持	GCMDR->GCMBR GCMCR->GCMAR
			双缓存	支持	支持	GCMFR->GCMDR GCMER->GCMCR GCMDR->GCMBR GCMCR->GCMAR
	EMB			支持	支持	PCNA<B>R.EMBCA<B>
互补PWM输出	端口状态控制	启动时		支持	支持	PCNA<B>R.STACA<B>
		停止时		支持	支持	PCNA<B>R.STPCA<B>
		上溢时		支持	支持	PCNA<B>R.OVFCA<B>
		下溢时		支持	支持	PCNA<B>R.UDFCA<B>
		计数匹配时（Up Counting）		支持	支持	PCNA<B>R.CMA<B>UCA<B>

		计数匹配时 （Down Counting）		支持	支持	PCNA<B>R.CMA<B>DCA<B>
	缓存传送	周期基准值	单缓存	支持	支持	PERBR->PERAR
			双缓存	支持	支持	PERCR->PERBR PERBR->PERAR
		比较基准值	单缓存	支持	支持	GCMDR->GCMBR GCMCR->GCMAR
			双缓存	支持	支持	GCMFR->GCMDR GCMER->GCMCR GCMDR->GCMBR GCMCR->GCMAR
		死区基准值	单缓存	支持	支持	DTUBR->DTUAR DTDBR->DTDAR
	无死区PWM输出			支持	支持	GCMAR=GCMBR
	带死区PWM输出		软件方式	支持	支持	GCMAR≠GCMBR
			硬件方式	不支持	支持	GCMBR=GCMAR-DTUAR GCNBR=GCMAR-DTDAR
	EMB			支持	支持	PCNA<B>R.EMBCA<B>

表 20-3 不同模式下的功能对比表

20.4 中断及事件说明

20.4.1 中断输出

Timer6 含有 6 个通用计数比较匹配中断（含 2 个捕获输入中断）、2 个专用计数比较匹配中断、2 个计数周期匹配中断、1 个死区时间错误中断。

20.4.1.1 计数比较匹配中断

通用比较基准值寄存器（GCMAR~GCMFR）共计 6 个，可分别与计数值比较产生比较匹配。计数比较匹配时，状态标志寄存器（STFLR）中的 STFLR.CMAF~STFLR.CMFF 位分别会被置为 1。此时若设定中断控制寄存器（ICONR）的 INTENA~INTENF 中相应位为 1 使能中断，则对应的中断请求（TMR6_<t>_GCMA~F）也会被触发。

在硬件捕获事件选择寄存器（HCPAR、HCPBR）选择的捕获输入有效条件产生时，捕获输入动作发生。此时若设置中断控制寄存器（ICONR）的 INTENA 或 INTENB 位为 1 使能中断，则对应的中断请求（TMR6_<t>_GCMA~B）被触发。

2 个专用比较基准值寄存器（SCMAR~SCMBR）也可分别与计数值比较产生比较匹配。计数比较匹配时，状态标志寄存器（STFLR）中的 STFLR.CMSPAF~CMSPBF 位分别会被置为 1。此时若设定中断控制寄存器（ICONR）的 INTENSAU<D>或 INTENSBU<D>中相应位为 1 使能中断，则对应的中断请求（TMR6_<t>_SCMA~B）也会被触发。

20.4.1.2 计数周期匹配中断

锯齿波递加计数至上溢点、锯齿波递减计数至下溢点、三角波计数至谷点或三角波计数至峰点时，状态标志寄存器（STFLR）的 STFLR.OVFF 或 STFLR.UDFF 位会被置为 1。此时若设定中断控制寄存器（ICONR）的 ICONR.INTENOVF 或 ICONR.INTENUDF 位使能中断，则在对应的时间点可触发计数周期匹配中断（TMR6_<t>_GOVF 和 TMR6_<t>_GUDF）。

20.4.1.3 死区时间错误中断

死区时间基准值寄存器（DTU<D>AR）的值加载到通用比较基准值寄存器（GCMBR）中时，若超过周期限制，则会产生死区时间错误，状态标志寄存器（STFLR）的 STFLR.DTEF 位会被置为 1。此时若设置中断控制寄存器（ICONR）的 INTENDTE 位

使能中断，则会在该时刻触发死区时间错误中断（TMR6_<t>_GDTE）。

20.4.2 事件输出

在时钟计数过程中，若产生周期匹配事件（三角波的上溢点或下溢点、锯齿波的计数峰点或谷点）、通用计数比较匹配事件、专用计数比较匹配事件时，会产生相应的事件输出信号，用于选择触发别的模块，如 ADC、DMA 等。

下图是单元 1 的通用比较匹配中断 A~F&&事件 A~F、专用比较匹配中断 A~B&&事件 A~B、周期匹配中断&&事件的动作例。

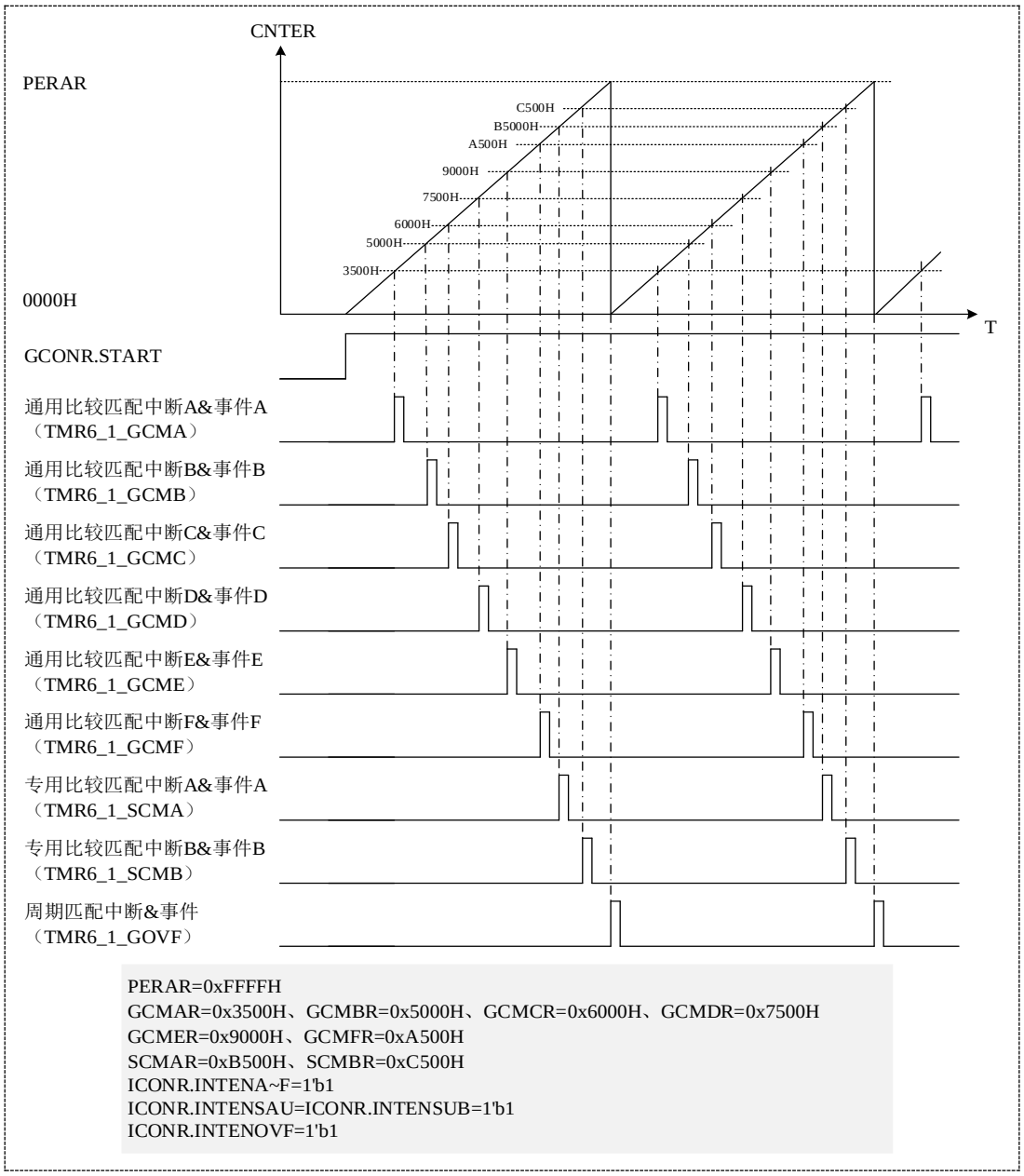


图 20-35 锯齿波模式时中断&事件输出例

20.5 寄存器说明

表 20-4 所示，为 Timer6 模块的寄存器列表。

BASE ADDR:

0x40018000（U1）、0x40018400（U2）、0x40018800（U3）、0x40018C00（U4）

0x40019000（U5）、0x40019400（U6）、0x40019800（U7）、0x40019C00（U8）

寄存器名	符号	偏移量	位宽	复位值
通用计数值寄存器	TMR6_CNTER	0x0000h	32	0x00000000h
通用刷新值寄存器	TMR6_UPDAR	0x0004h	32	0x00000000h
通用周期基准值寄存器A	TMR6_PERAR	0x0040h	32	0x0000FFFFh
通用周期基准值寄存器B	TMR6_PERBR	0x0044h	32	0x0000FFFFh
通用周期基准值寄存器C	TMR6_PERCR	0x0048h	32	0x0000FFFFh
通用比较基准值寄存器A	TMR6_GCMAR	0x0080h	32	0x0000FFFFh
通用比较基准值寄存器B	TMR6_GCMBR	0x0084h	32	0x0000FFFFh
通用比较基准值寄存器C	TMR6_GCMCR	0x0088h	32	0x0000FFFFh
通用比较基准值寄存器D	TMR6_GCMDR	0x008Ch	32	0x0000FFFFh
通用比较基准值寄存器E	TMR6_GCMER	0x0090h	32	0x0000FFFFh
通用比较基准值寄存器F	TMR6_GCMFR	0x0094h	32	0x0000FFFFh
专用比较基准值寄存器A	TMR6_SCMAR	0x00C0h	32	0x0000FFFFh
专用比较基准值寄存器B	TMR6_SCMBR	0x00C4h	32	0x0000FFFFh
专用比较基准值寄存器C	TMR6_SCMCR	0x00C8h	32	0x0000FFFFh
专用比较基准值寄存器D	TMR6_SCMDR	0x00CCh	32	0x0000FFFFh
专用比较基准值寄存器E	TMR6_SCMER	0x00D0h	32	0x0000FFFFh
专用比较基准值寄存器F	TMR6_SCMFR	0x00D4h	32	0x0000FFFFh
死区时间基准值寄存器UA	TMR6_DTUAR	0x0100h	32	0x0000FFFFh
死区时间基准值寄存器DA	TMR6_DTDAR	0x0104h	32	0x0000FFFFh
死区时间基准值寄存器UB	TMR6_DTUBR	0x0108h	32	0x0000FFFFh
死区时间基准值寄存器DB	TMR6_DTDBR	0x010Ch	32	0x0000FFFFh

通用控制寄存器	TMR6_GCONR	0x0140h	32	0x00000002h
中断控制寄存器	TMR6_ICONR	0x0144h	32	0x00000000h
缓存控制寄存器	TMR6_BCONR	0x0148h	32	0x00000000h
死区控制寄存器	TMR6_DCONR	0x014Ch	32	0x00000000h
端口控制寄存器A	TMR6_PCNAR	0x0154h	32	0x00000000h
端口控制寄存器B	TMR6_PCNBR	0x0158h	32	0x00000000h
滤波控制寄存器G	TMR6_FCNGR	0x015Ch	32	0x00000000h
滤波控制寄存器T	TMR6_FCNTR	U1_base+0x03ECh	32	0x00000000h
有效周期寄存器	TMR6_VPERR	0x0160h	32	0x00000000h
状态标志寄存器	TMR6_STFLR	0x0164h	32	0x80000000h
硬件启动事件选择寄存器	TMR6_HSTAR	0x0180h	32	0x00000000h
硬件停止事件选择寄存器	TMR6_HSTPR	0x0184h	32	0x00000000h
硬件清零事件选择寄存器	TMR6_HCLRR	0x0188h	32	0x00000000h
硬件刷新事件选择寄存器	TMR6_HUPDR	0x018Ch	32	0x00000000h
硬件捕获事件选择寄存器A	TMR6_HCPAR	0x0190h	32	0x00000000h
硬件捕获事件选择寄存器B	TMR6_HCPBR	0x0194h	32	0x00000000h
硬件递加事件选择寄存器	TMR6_HCUPR	0x0198h	32	0x00000000h
硬件递减事件选择寄存器	TMR6_HCDOR	0x019Ch	32	0x00000000h
硬件触发事件选择寄存器0	TMR6_HTSSR0	（0x40010858h）	32	0x000001FFh
硬件触发事件选择寄存器1	TMR6_HTSSR1	（0x4001085Ch）	32	0x000001FFh
硬件触发事件选择寄存器2	TMR6_HTSSR2	（0x40010860h）	32	0x000001FFh
硬件触发事件选择寄存器3	TMR6_HTSSR3	（0x40010864h）	32	0x000001FFh
软件同步启动控制寄存器	TMR6_SSTAR	U1_base+0x03F0h	32	0x00000000h
软件同步停止控制寄存器	TMR6_SSTPR	U1_base+0x03F4h	32	0x00000000h
软件同步清零控制寄存器	TMR6_SCLRR	U1_base+0x03F8h	32	0x00000000h
软件同步刷新控制寄存器	TMR6_SUPDR	U1_base+0x03FCh	32	0x00000000h

表 20-4 Timer6 寄存器列表

注意：

– U1~4 单元的通用计数值寄存器、通用刷新值寄存器及各基准值寄存器的位宽均

为 32 位，各复位值与 U5~U8 单元不一样。其中，通用计数值寄存器、通用刷新值寄寄存器的复位值为 0x00000000h；各基准值寄存器的复位值为 0xFFFFFFFFh。

- 触发选择寄存器（TMR6_HTSSR0~3）是 4 个独立于单元的寄存器，为 8 个单元 Timer6 所共有。
- 软件同步寄存器（TMR6_SSTAR、TMR6_SSTPR、TMR6_SCLLR、TMR6_SUPDR）是 4 个独立于单元的寄存器，为 8 个单元 Timer6 所共有。
- 滤波控制寄存器（TMR6_FCNTR）是 1 个独立于单元的寄存器，为 8 个单元 Timer6 所共有。

20.5.1 通用计数值寄存器（TMR6_CNTER）

复位值: 0x00000000h

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

20.5.2 通用刷新值寄存器（TMR6_UPDAR）

复位值: 0x00000000h

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”	R
b15~b0	UPDA[15:0]	刷新值	设定要更新进定时器的刷新值	R/W

20.5.3 通用周期基准值寄存器（TMR6_PERmR）（m=A~C）

复位值: 0x0000FFFFh

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

20.5.4 通用比较基准值寄存器（TMR6_GCMmR）（m=A~F）

复位值: 0x0000FFFFh

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

20.5.5 专用比较基准值寄存器（TMR6_SCMmR）（m=A~F）

复位值: 0x0000FFFFh

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

20.5.6 死区时间基准值寄存器（TMR6_DTmnR）（m=D、U&&n=A、B）

复位值: 0x0000FFFFh

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

20.5.7 通用控制寄存器（TMR6_GCONR）

复位值: 0x00000002h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved												ZMSK VAL[1:0]		ZMSK POS	ZMSK REV

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							OV STP	CKDIV[3:0]				-	MODE	DIR	START

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 00：Z相输入屏蔽功能无效 01：位置计数上溢后或下溢后的4个计数周期内的Z相输入被屏蔽 10：位置计数上溢后或下溢后的8个计数周期内的Z相输入被屏蔽 11：位置计数上溢后或下溢后的16个计数周期内的Z相输入被屏蔽	R/W
b17	ZMSKPOS	Z相输入位置定时器选择	0：Z相输入时该单元作为位置定时器，在屏蔽周期期间内位置定时器清零功能正常动作 1：Z相输入时该单元作为位置定时器，在屏蔽周期期间内位置定时器清零功能被屏蔽	R/W
b16	ZMSKREV	Z相输入公转定时器选择	0：Z相输入时该单元作为公转定时器，在屏蔽周期期间内公转定时器计数功能正常动作 1：Z相输入时该单元作为公转定时器，在屏蔽周期期间内公转定时器计数功能被屏蔽	R/W
b15~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8	OVSTP	计数溢出停止控制	0：计数器在计数上溢或下溢后，继续计数 1：计数器在计数上溢或下溢后，停止计数	R/W
b7~b4	CKDIV[2:0]	计数时钟选择	0000：PCLK0 0001：PCLK0/2 0010：PCLK0/4 0011：PCLK0/8 0100：PCLK0/16 0101：PCLK0/32 0110：PCLK0/64 0111：PCLK0/128 1000：PCLK0/256 1001：PCLK0/512 1010：PCLK0/1024	R/W
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	MODE	计数模式	0：锯齿波模式	R/W

			1：三角波模式	
b1	DIR	计数方向	0：递减计数	R/W
			1：递加计数	
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	

20.5.8 中断控制寄存器（TMR6_ICONR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved												INTENSBD	INTENSBU	INTENSAD	INTENSAU

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							INTENDTE	INTENUDF	INTENOVF	INTENF	INTENE	INTEND	INTENC	INTENB	INTENA

位	标记	位名	功能	读写
b31~b20	Reserved	-	读出时为“0”,写入时写“0”	R/W
b19	INTENSBD	专用向下计数中断使能B	0：向下计数期间，SCMBR寄存器和计数值相等时，该中断无效 1：向下计数期间，SCMBR寄存器和计数值相等时，该中断使能	R/W
b18	INTENSBU	专用向上计数中断使能B	0：向上计数期间，SCMBR寄存器和计数值相等时，该中断无效 1：向上计数期间，SCMBR寄存器和计数值相等时，该中断使能	R/W
b17	INTENSAD	专用向下计数中断使能A	0：向下计数期间，SCMAR寄存器和计数值相等时，该中断无效 1：向下计数期间，SCMAR寄存器和计数值相等时，该中断使能	R/W
b16	INTENSAU	专用向上计数中断使能A	0：向上计数期间，SCMAR寄存器和计数值相等时，该中断无效 1：向上计数期间，SCMAR寄存器和计数值相等时，该中断使能	R/W
b15~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8	INTENDTE	死区时间错误中断使能	0：死区时间错误时，该中断无效 1：死区时间错误时，该中断使能	R/W
b7	INTENUDF	下溢中断使能	0：计数器计数下溢时，该中断无效 1：计数器计数下溢时，该中断使能	R/W
b6	INTENOVF	上溢中断使能	0：计数器计数上溢时，该中断无效 1：计数器计数上溢时，该中断使能	R/W
b5	INTENF	计数匹配中断使能F	0：GCMFR寄存器与计数值相等时，该中断无效 1：GCMFR寄存器与计数值相等时，该中断使能	R/W
b4	INTENE	计数匹配中断使能E	0：GCMER寄存器与计数值相等时，该中断无效 1：GCMER寄存器与计数值相等时，该中断使能	R/W
b3	INTEND	计数匹配中断使能D	0：GCMDR寄存器与计数值相等时，该中断无效 1：GCMDR寄存器与计数值相等时，该中断使能	R/W
b2	INTENC	计数匹配中断使能C	0：GCMCR寄存器与计数值相等时，该中断无效 1：GCMCR寄存器与计数值相等时，该中断使能	R/W
b1	INTENB	计数匹配中断使能B	0：GCMBR寄存器与计数值相等时，或者发生捕获输入事件时，该中断无效 1：GCMBR寄存器与计数值相等时，或者发生捕获输入事件时，该中断使能	R/W
b0	INTENA	计数匹配中断使能A	0：GCMAR寄存器与计数值相等时，或者发生捕获输入事件时，该中断无效	R/W

	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

20.5.9 缓存控制寄存器（TMR6_BCONR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								BTRDSPB	BTRUSPB	BSESPB	BENSPB	BTRDSPA	BTRUSPA	BSESPA	BENSPA

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	BTRDP	BTRUP	BSEP	BENP	BTRDB	BTRUB	BSEB	BENB	BTRDA	BTRUA	BSEA	BENA

位	标记	位名	功能	读写
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	BTRDSPB	专用比较基准值缓存传送时间设定DB	0：三角波计数模式计数到谷点时，缓存值不传送 1：三角波计数模式计数到谷点时，发生一次缓存值传送 注：锯齿波模式时，不需要该位控制	R/W
b22	BTRUSPB	专用比较基准值缓存传送时间设定UB	0：三角波计数模式计数到峰点时，缓存值不传送 1：三角波计数模式计数到峰点时，发生一次缓存值传送 注：锯齿波模式时，不需要该位控制	R/W
b21	BSESPB	专用比较基准值缓存传送选择B	0：单缓存传送（SCMDR->SCMBR） 1：双缓存传送（SCMFR->SCMDR->SCMBR）	R/W
b20	BENSPB	专用比较基准值缓存传送B	0：缓存传送无效 1：缓存传送使能	R/W
b19	BTRDSPA	专用比较基准值缓存传送时间设定DA	0：三角波计数模式计数到谷点时，缓存值不传送 1：三角波计数模式计数到谷点时，发生一次缓存值传送 注：锯齿波模式时，不需要该位控制	R/W
b18	BTRUSPA	专用比较基准值缓存传送时间设定UA	0：三角波计数模式计数到峰点时，缓存值不传送 1：三角波计数模式计数到峰点时，发生一次缓存值传送 注：锯齿波模式时，不需要该位控制	R/W
b17	BSESPA	专用比较基准值缓存传送选择A	0：单缓存传送（SCMCR->SCMAR） 1：双缓存传送（SCMER->SCMCR->SCMAR）	R/W
b16	BENSPA	专用比较基准值缓存传送A	0：缓存传送无效 1：缓存传送使能	R/W
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	BTRDP	周期值缓存传送时间设定D	0：三角波计数模式计数到谷点时，缓存值不传送 1：三角波计数模式计数到谷点时，发生一次缓存值传送 注：锯齿波模式时，不需要该位控制	R/W
b10	BTRUP	周期值缓存传送时间设定U	0：三角波计数模式计数到峰点时，缓存值不传送 1：三角波计数模式计数到峰点时，发生一次缓存值传送 注：锯齿波模式时，不需要该位控制	R/W
b9	BSEP	周期值缓存传送选择	0：单缓存传送（PERBR->PERAR） 1：双缓存传送（PERCR->PERBR->PERAR）	R/W

			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	BTRDB	通用比较值缓存传送时间设定DB	0：三角波计数模式计数到谷点时，缓存值不传送 1：三角波计数模式计数到谷点时，发生一次缓存值传送 注：锯齿波模式时，不需要该位控制	R/W
b6	BTRUB	通用比较值缓存传送时间设定UB	0：三角波计数模式计数到峰点时，缓存值不传送 1：三角波计数模式计数到峰点时，发生一次缓存值传送 注：锯齿波模式时，不需要该位控制	R/W
b5	BSEB	通用比较值缓存传送选择B	比较输出功能时： 0：单缓存传送（GCMDR->GCMBR） 1：双缓存传送（GCMFR->GCMDR->GCMBR） 捕获输入功能时： 0：单缓存传送（GCMBR->GCMDR） 1：双缓存传送（GCMBR->GCMDR->GCMFR）	R/W
b4	BENB	通用比较值缓存传送B	0：缓存传送无效 1：缓存传送使能	R/W
b3	BTRDA	通用比较值缓存传送时间设定DA	0：三角波计数模式计数到谷点时，缓存值不传送 1：三角波计数模式计数到谷点时，发生一次缓存值传送 注：锯齿波模式时，不需要该位控制	R/W
b2	BTRUA	通用比较值缓存传送时间设定UA	0：三角波计数模式计数到峰点时，缓存值不传送 1：三角波计数模式计数到峰点时，发生一次缓存值传送 注：锯齿波模式时，不需要该位控制	R/W
b1	BSEA	通用比较值缓存传送选择A	比较输出功能时： 0：单缓存传送（GCMCR->GCMAR） 1：双缓存传送（GCMER->GCMCR->GCMAR） 捕获输入功能时： 0：单缓存传送（GCMAR->GCMCR） 1：双缓存传送（GCMAR->GCMCR->GCMER）	R/W
b0	BENA	通用比较值缓存传送A	0：缓存传送无效 1：缓存传送使能	R/W

20.5.10 死区控制寄存器（TMR6_DCONR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								DTB TRD	DTB TRU	DTB END	DTB ENU	-	-	SEPA	DTC EN

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	DTBTRD	死区时间值缓存传送时间设定D	0：三角波计数模式计数到谷点时，死区缓存值不传送 1：三角波计数模式计数到谷点时，发生一次死区缓存值传送 注：锯齿波模式时，不需要该位控制	R/W
b6	DTBTRU	死区时间值缓存传送时间设定U	0：三角波计数模式计数到峰点时，死区缓存值不传送 1：三角波计数模式计数到峰点时，发生一次死区缓存值传送 注：锯齿波模式时，不需要该位控制	R/W
b5	DTBEND	死区时间值缓存传送D	0：缓存传送无效 1：缓存传送使能（DTDBR->DTDAR）	R/W
b4	DTBENU	死区时间值缓存传送U	0：缓存传送无效 1：缓存传送使能（DTUBR->DTUAR）	R/W
b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	SEPA	分离设定	0：DTUAR和DTDAR分别设定 1：DTDAR的值和DTUAR的值自动相等	R/W
b0	DTCEN	死区功能	0：死区功能无效 1：死区功能有效	R/W

20.5.11 端口控制寄存器（TMR6_PCNAR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25 b24	b23 b22	b21 b20	b19	b18	b17 b16
CAPMDA	-	-	OUTENA	-	-	EMBSA[1:0]	EMBRA[1:0]	EMBCA[1:0]	-	-	FORCA[1:0]

b15 b14	b13 b12	b11 b10	b9 b8	b7 b6	b5 b4	b3 b2	b1 b0
CMBDCA[1:0]	CMBUCA[1:0]	CMADCA[1:0]	CMAUCA[1:0]	UDFCA[1:0]	OVFCA[1:0]	STPCA[1:0]	STACA[1:0]

位	标记	位名	功能	读写
b31	CAPMDA	功能模式选择A	0：比较输出功能 1：捕获输入功能	R/W
b30~b29	Reserved	-	读出时为“0”,写入时写“0”	R/W
b28	OUTENA	输出使能A	0：Timer6功能时的TIM6_<t>_PWMA端口输出无效 1：Timer6功能时的TIM6_<t>_PWMA端口输出有效	R/W
b27~b26	Reserved	-	读出时为“0”,写入时写“0”	R/W
b25~b24	EMBSA[1:0]	EMB事件时端口状态选择A	00：选择EMB事件通道0有效 01：选择EMB事件通道1有效 10：选择EMB事件通道2有效 11：选择EMB事件通道3有效	R/W
b23~b22	EMBRA[1:0]	EMB释放方式选择A	00：被选择的通道EMB事件无效时，立即释放TIM6_<t>_PWMA端口（One Shot） 01：被选择的通道EMB事件无效时，等到计数器计数到上溢时释放TIM6_<t>_PWMA端口（Cycle By Cycle 1） 10：被选择的通道EMB事件无效时，等到计数器计数到下溢时释放TIM6_<t>_PWMA端口（Cycle By Cycle 2） 11：被选择的通道EMB事件无效时，等到计数器计数到上溢或下溢时释放TIM6_<t>_PWMA端口（Cycle By Cycle 3）	R/W
b21~b20	EMBCA[1:0]	EMB事件时端口状态设定A	00：被选择的通道发生EMB事件时，TIM6_<t>_PWMA端口正常输出 01：被选择的通道发生EMB事件时，TIM6_<t>_PWMA端口输出高阻态 10：被选择的通道发生EMB事件时，TIM6_<t>_PWMA端口输出低电平 11：被选择的通道发生EMB事件时，TIM6_<t>_PWMA端口输出高电平	R/W
b19~b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17~b16	FORCA[1:0]	强制端口状态设定A	0x：设定无效定 10：下周期开始，TIM6_<t>_PWMA端口输出设定为低电平 11：下周期开始，TIM6_<t>_PWMA端口输出设定为高电平	R/W

			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	
b15~b14	CMBDCA[1:0]	向下计数&&比较值B匹配时端口状态设定A	00：在向下计数期间，定时器计数值与GCMBR相等时，TIM6_<t>_PWMA端口输出设定为低电平 01：在向下计数期间，定时器计数值与GCMBR相等时，TIM6_<t>_PWMA端口输出设定为高电平 10：在向下计数期间，定时器计数值与GCMBR相等时，TIM6_<t>_PWMA端口输出保持先前状态 11：在向下计数期间，定时器计数值与GCMBR相等时，TIM6_<t>_PWMA端口输出设定为反转电平	R/W
b13~b12	CMBUCA[1:0]	向上计数&&比较值B匹配时端口状态设定A	00：在向上计数期间，定时器计数值与GCMBR相等时，TIM6_<t>_PWMA端口输出设定为低电平 01：在向上计数期间，定时器计数值与GCMBR相等时，TIM6_<t>_PWMA端口输出设定为高电平 10：在向上计数期间，定时器计数值与GCMBR相等时，TIM6_<t>_PWMA端口输出保持先前状态 11：在向上计数期间，定时器计数值与GCMBR相等时，TIM6_<t>_PWMA端口输出设定为反转电平	R/W
b11~b10	CMADCA[1:0]	向下计数&&比较值A匹配时端口状态设定A	00：在向下计数期间，定时器计数值与GCMAR相等时，TIM6_<t>_PWMA端口输出设定为低电平 01：在向下计数期间，定时器计数值与GCMAR相等时，TIM6_<t>_PWMA端口输出设定为高电平 10：在向下计数期间，定时器计数值与GCMAR相等时，TIM6_<t>_PWMA端口输出保持先前状态 11：在向下计数期间，定时器计数值与GCMAR相等时，TIM6_<t>_PWMA端口输出设定为反转电平	R/W
b9~b8	CMAUCA[1:0]	向上计数&&比较值A匹配时端口状态设定A	00：在向上计数期间，定时器计数值与GCMAR相等时，TIM6_<t>_PWMA端口输出设定为低电平 01：在向上计数期间，定时器计数值与GCMAR相等时，TIM6_<t>_PWMA端口输出设定为高电平 10：在向上计数期间，定时器计数值与GCMAR相等时，TIM6_<t>_PWMA端口输出保持先前状态 11：在向上计数期间，定时器计数值与GCMAR相等时，TIM6_<t>_PWMA端口输出设定为反转电平	R/W
b7~b6	UDFCA[1:0]	计数下溢端口状态设定A	00：计数下溢时，TIM6_<t>_PWMA端口输出设定为低电平 01：计数下溢时，TIM6_<t>_PWMA端口输出设定为高电平 10：计数下溢时，TIM6_<t>_PWMA端口输出保持先前状态 11：计数下溢时，TIM6_<t>_PWMA端口输出设定为反转电平	R/W
b5~b4	OVFCA[1:0]	计数上溢端口状态设定A	00：计数上溢时，TIM6_<t>_PWMA端口输出设定为低电平 01：计数上溢时，TIM6_<t>_PWMA端口输出设定为高电平 10：计数上溢时，TIM6_<t>_PWMA端口输出保持先前状态	R/W

			11：计数上溢时，TIM6_<t>_PWMA端口输出设定为反转电平	
b3~b2	STPCA[1:0]	计数停止端口状态设定A	00：计数停止时，TIM6_<t>_PWMA端口输出设定为低电平 01：计数停止时，TIM6_<t>_PWMA端口输出设定为高电平 10：计数停止时，TIM6_<t>_PWMA端口输出保持先前状态 11：计数停止时，TIM6_<t>_PWMA端口输出保持先前状态	R/W
b1~b0	STACA[1:0]	计数开始端口状态设定A	00：计数开始时，TIM6_<t>_PWMA端口输出设定为低电平 01：计数开始时，TIM6_<t>_PWMA端口输出设定为高电平 10：计数开始时，TIM6_<t>_PWMA端口输出保持先前状态 11：计数开始时，TIM6_<t>_PWMA端口输出保持先前状态	R/W

20.5.12 端口控制寄存器（TMR6_PCNBR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25 b24	b23 b22	b21 b20	b19	b18	b17 b16
CAPMDB	-	-	OUTENB	-	-	EMBSB[1:0]	EMBRB[1:0]	EMBCB[1:0]	-	-	FORCB[1:0]

b15 b14	b13 b12	b11 b10	b9 b8	b7 b6	b5 b4	b3 b2	b1 b0
CMBDCB[1:0]	CMBUCB[1:0]	CMADCB[1:0]	CMAUCB[1:0]	UDFCB[1:0]	OVFCB[1:0]	STPCB[1:0]	STACB[1:0]

位	标记	位名	功能	读写
b31	CAPMDB	功能模式选择B	0：比较输出功能 1：捕获输入功能	R/W
b30~b29	Reserved	-	读出时为“0”,写入时写“0”	R/W
b28	OUTENB	输出使能B	0：Timer6功能时的TIM6_<t>_PWMB端口输出无效 1：Timer6功能时的TIM6_<t>_PWMB端口输出有效	R/W
b27~b26	Reserved	-	读出时为“0”,写入时写“0”	R/W
b25~b24	EMBSB[1:0]	EMB事件时端口状态选择B	00：选择EMB事件通道0有效 01：选择EMB事件通道1有效 10：选择EMB事件通道2有效 11：选择EMB事件通道3有效	R/W
b23~b22	EMBRB[1:0]	EMB释放方式选择B	00：被选择的通道EMB事件无效时，立即释放TIM6_<t>_PWMB端口（One Shot） 01：被选择的通道EMB事件无效时，等到计数器计数到上溢时释放TIM6_<t>_PWMB端口（Cycle By Cycle 1） 10：被选择的通道EMB事件无效时，等到计数器计数到下溢时释放TIM6_<t>_PWMB端口（Cycle By Cycle 2） 11：被选择的通道EMB事件无效时，等到计数器计数到上溢或下溢时释放TIM6_<t>_PWMB端口（Cycle By Cycle 3）	R/W
b21~b20	EMBCB[1:0]	EMB事件时端口状态设定B	00：被选择的通道发生EMB事件时，TIM6_<t>_PWMB端口正常输出 01：被选择的通道发生EMB事件时，TIM6_<t>_PWMB端口输出高阻态 10：被选择的通道发生EMB事件时，TIM6_<t>_PWMB端口输出低电平 11：被选择的通道发生EMB事件时，TIM6_<t>_PWMB端口输出高电平	R/W
b19~b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17~b16	FORCB[1:0]	强制端口状态设定B	0x：设定无效定 10：下周期开始，TIM6_<t>_PWMB端口输出设定为低电平 11：下周期开始，TIM6_<t>_PWMB端口输出设定为高电平	R/W

			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	
b15~b14	CMBDCB[1:0]	向下计数&&比较值B匹配时端口状态设定B	00：在向下计数期间，定时器计数值与GCMBR相等时，TIM6_<t>_PWMB端口输出设定为低电平 01：在向下计数期间，定时器计数值与GCMBR相等时，TIM6_<t>_PWMB端口输出设定为高电平 10：在向下计数期间，定时器计数值与GCMBR相等时，TIM6_<t>_PWMB端口输出保持先前状态 11：在向下计数期间，定时器计数值与GCMBR相等时，TIM6_<t>_PWMB端口输出设定为反转电平	R/W
b13~b12	CMBUCB[1:0]	向上计数&&比较值B匹配时端口状态设定B	00：在向上计数期间，定时器计数值与GCMBR相等时，TIM6_<t>_PWMB端口输出设定为低电平 01：在向上计数期间，定时器计数值与GCMBR相等时，TIM6_<t>_PWMB端口输出设定为高电平 10：在向上计数期间，定时器计数值与GCMBR相等时，TIM6_<t>_PWMB端口输出保持先前状态 11：在向上计数期间，定时器计数值与GCMBR相等时，TIM6_<t>_PWMB端口输出设定为反转电平	R/W
b11~b10	CMADCB[1:0]	向下计数&&比较值A匹配时端口状态设定B	00：在向下计数期间，定时器计数值与GCMAR相等时，TIM6_<t>_PWMB端口输出设定为低电平 01：在向下计数期间，定时器计数值与GCMAR相等时，TIM6_<t>_PWMB端口输出设定为高电平 10：在向下计数期间，定时器计数值与GCMAR相等时，TIM6_<t>_PWMB端口输出保持先前状态 11：在向下计数期间，定时器计数值与GCMAR相等时，TIM6_<t>_PWMB端口输出设定为反转电平	R/W
b9~b8	CMAUCB[1:0]	向上计数&&比较值A匹配时端口状态设定B	00：在向上计数期间，定时器计数值与GCMAR相等时，TIM6_<t>_PWMB端口输出设定为低电平 01：在向上计数期间，定时器计数值与GCMAR相等时，TIM6_<t>_PWMB端口输出设定为高电平 10：在向上计数期间，定时器计数值与GCMAR相等时，TIM6_<t>_PWMB端口输出保持先前状态 11：在向上计数期间，定时器计数值与GCMAR相等时，TIM6_<t>_PWMB端口输出设定为反转电平	R/W
b7~b6	UDFCB[1:0]	计数下溢端口状态设定B	00：计数下溢时，TIM6_<t>_PWMB端口输出设定为低电平 01：计数下溢时，TIM6_<t>_PWMB端口输出设定为高电平 10：计数下溢时，TIM6_<t>_PWMB端口输出保持先前状态 11：计数下溢时，TIM6_<t>_PWMB端口输出设定为反转电平	R/W
b5~b4	OVFCB[1:0]	计数上溢端口状态设定B	00：计数上溢时，TIM6_<t>_PWMB端口输出设定为低电平 01：计数上溢时，TIM6_<t>_PWMB端口输出设定为高电平 10：计数上溢时，TIM6_<t>_PWMB端口输出保持先前状态	R/W

			11：计数上溢时，TIM6_<t>_PWMB端口输出设定为反转电平	
b3~b2	STPCB[1:0]	计数停止端口状态设定B	00：计数停止时，TIM6_<t>_PWMB端口输出设定为低电平 01：计数停止时，TIM6_<t>_PWMB端口输出设定为高电平 10：计数停止时，TIM6_<t>_PWMB端口输出保持先前状态 11：计数停止时，TIM6_<t>_PWMB端口输出保持先前状态	R/W
b1~b0	STBCB[1:0]	计数开始端口状态设定B	00：计数开始时，TIM6_<t>_PWMB端口输出设定为低电平 01：计数开始时，TIM6_<t>_PWMB端口输出设定为高电平 10：计数开始时，TIM6_<t>_PWMB端口输出保持先前状态 11：计数开始时，TIM6_<t>_PWMB端口输出保持先前状态	R/W

20.5.13 滤波控制寄存器（TMR6_FCNGR）

复位值: 0x00000000h

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									NOFI CKGB[1:0]		NOFI ENGB	-	NOFI CKGA[1:0]		NOFI ENGA

位	标记	位名	功能	读写
b31~b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6~b5	NOFICKGB[1:0]	滤波采样基准时钟选择GB	00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64	R/W
b4	NOFIENGB	捕获输入端口滤波GB	0：该单元TIM6_<t>_PWMB输入端口滤波功能无效 1：该单元TIM6_<t>_PWMB输入端口滤波功能使能	R/W
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~b1	NOFICKGA[1:0]	滤波采样基准时钟选择GA	00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64	R/W
b0	NOFIENGA	捕获输入端口滤波GA	0：该单元TIM6_<t>_PWMA输入端口滤波功能无效 1：该单元TIM6_<t>_PWMA输入端口滤波功能使能	R/W

20.5.14 滤波控制寄存器（TMR6_FCNTR）

复位值: 0x00000000h

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	NOFI CKTD[1:0]		NOFI ENTD	-	NOFI CKTC[1:0]		NOFI ENTC	-	NOFI CKTB[1:0]		NOFI ENTB	-	NOFI CKTA[1:0]		NOFI ENTA

位	标记	位名	功能	读写
b31~b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14~b13	NOFICKTD[1:0]	滤波采样基准时钟选择TD	00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64	R/W
b12	NOFIENTD	捕获输入端口滤波TD	0：TIM6_TRIGD端口输入滤波功能无效 1：TIM6_TRIGD端口输入滤波功能使能	R/W
b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10~b9	NOFICKTC[1:0]	滤波采样基准时钟选择TC	00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64	R/W
b8	NOFIENTC	捕获输入端口滤波TC	0：TIM6_TRIGC端口输入滤波功能无效 1：TIM6_TRIGC端口输入滤波功能使能	R/W
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6~b5	NOFICKTB[1:0]	滤波采样基准时钟选择TB	00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64	R/W
b4	NOFIENTB	捕获输入端口滤波TB	0：TIM6_TRIGB端口输入滤波功能无效 1：TIM6_TRIGB端口输入滤波功能使能	R/W
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~b1	NOFICKTA[1:0]	滤波采样基准时钟选择TA	00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64	R/W
b0	NOFIENTA	捕获输入端口滤波TA	0：TIM6_TRIGA端口输入滤波功能无效 1：TIM6_TRIGA端口输入滤波功能使能	R/W

20.5.15 有效周期寄存器（TMR6_VPERR）

复位值: 0x00000000h

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	R/W
			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	
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	R/W
			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	
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	R/W
			1：有效周期选择功能使能	
b8	SPPERIA	专用信号有效周期选择A	0：有效周期选择功能无效	R/W
			1：有效周期选择功能使能	
b7~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

20.5.16 状态标志寄存器（TMR6_STFLR）

复位值: 0x80000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DIRF	Reserved							VPERNUM[2:0]			Reserved				

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	CMSBDF	CMSBUF	CMSADF	CMSAUF	DTEF	UDFF	OVFF	CMFF	CMEF	CMDF	CMCF	CMBF	CMAF

位	标记	位名	功能	读写
b31	DIRF	计数方向	0：递减计数 1：递加计数	R
b30-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准值匹配B	0：向下计数时，SCMBR寄存器的值与计数值不相等 1：向下计数时，SCMBR寄存器的值与计数值相等	R/W
b11	CMSBUF	向上计数专用比较基准值匹配B	0：向上计数时，SCMBR寄存器的值与计数值不相等 1：向上计数时，SCMBR寄存器的值与计数值相等	R/W
b10	CMSADF	向下计数专用比较基准值匹配A	0：向下计数时，SCMAR寄存器的值与计数值不相等 1：向下计数时，SCMAR寄存器的值与计数值相等	R/W
b09	CMSAUF	向上计数专用比较基准值匹配A	0：向上计数时，SCMAR寄存器的值与计数值不相等 1：向上计数时，SCMAR寄存器的值与计数值相等	R/W
b8	DTEF	死区时间错误	0：未发生死区时间错误 1：发生死区时间错误	R
b7	UDFF	下溢匹配	0：未发生锯齿波下溢或三角波计数到谷点 1：发生锯齿波下溢或三角波计数到谷点	R/W
b6	OVFF	上溢匹配	0：未发生锯齿波上溢或三角波计数到峰点 1：发生锯齿波上溢或三角波计数到峰点	R/W
b5	CMFF	计数匹配F	0：GCMFR寄存器的值与计数值不相等 1：GCMFR寄存器的值与计数值相等	R/W
b4	CMEF	计数匹配E	0：GCMER寄存器的值与计数值不相等 1：GCMER寄存器的值与计数值相等	R/W
b3	CMDF	计数匹配D	0：GCMDR寄存器的值与计数值不相等 1：GCMDR寄存器的值与计数值相等	R/W
b2	CMCF	计数匹配C	0：GCMCR寄存器的值与计数值不相等 1：GCMCR寄存器的值与计数值相等	R/W
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_<t>_PWMB捕获完成动作 1：GCMBR寄存器的值与计数值相等，或发生TIM6_<t>_PWMB	R/W

			捕获完成动作	
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

20.5.17 硬件启动事件选择寄存器（TMR6_HSTAR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HSTA23	HSTA22	HSTA21	HSTA20	HSTA19	HSTA18	HSTA17	HSTA16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HSTA11	HSTA10	HSTA9	HSTA8	STAS	-	-	-	HSTA3	HSTA2	HSTA1	HSTA0

位	标记	位名	功能	读写
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	HSTA23	硬件启动条件23	条件：TIM6_TRIGD端口上采样到下降沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
b22	HSTA22	硬件启动条件22	条件：TIM6_TRIGD端口上采样到上升沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
b21	HSTA21	硬件启动条件21	条件：TIM6_TRIGC端口上采样到下降沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
b20	HSTA20	硬件启动条件20	条件：TIM6_TRIGC端口上采样到上升沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
b19	HSTA19	硬件启动条件19	条件：TIM6_TRIGB端口上采样到下降沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
b18	HSTA18	硬件启动条件18	条件：TIM6_TRIGB端口上采样到上升沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
b17	HSTA17	硬件启动条件17	条件：TIM6_TRIGA端口上采样到下降沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
b16	HSTA16	硬件启动条件16	条件：TIM6_TRIGA端口上采样到上升沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	HSTA11	硬件启动条件11	条件：内部硬件触发事件3有效 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
b10	HSTA10	硬件启动条件10	条件：内部硬件触发事件2有效	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	
b9	HSTA9	硬件启动条件9	条件：内部硬件触发事件1有效 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
b8	HSTA8	硬件启动条件8	条件：内部硬件触发事件0有效 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
b7	STAS	硬件启动使能	0：硬件启动无效 1：硬件启动有效 注：硬件启动有效时，SSTAR的设定无效	R/W
b6~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	HSTA3	硬件启动条件3	条件：TIM6_<t>_PWMB端口上采样到下降沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
b2	HSTA2	硬件启动条件2	条件：TIM6_<t>_PWMB端口上采样到上升沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
b1	HSTA1	硬件启动条件1	条件：TIM6_<t>_PWMA端口上采样到下降沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
b0	HSTA0	硬件启动条件0	条件：TIM6_<t>_PWMA端口上采样到上升沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W

20.5.18 硬件停止事件选择寄存器（TMR6_HSTPR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HSTP23	HSTP22	HSTP21	HSTP20	HSTP19	HSTP18	HSTP17	HSTP16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HSTP11	HSTP10	HSTP9	HSTP8	STPS	-	-	-	HSTP3	HSTP2	HSTP1	HSTP0

位	标记	位名	功能	读写
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	HSTP23	硬件停止条件23	条件：TIM6_TRIGD端口上采样到下降沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b22	HSTP22	硬件停止条件22	条件：TIM6_TRIGD端口上采样到上升沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b21	HSTP21	硬件停止条件21	条件：TIM6_TRIGC端口上采样到下降沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b20	HSTP20	硬件停止条件20	条件：TIM6_TRIGC端口上采样到上升沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b19	HSTP19	硬件停止条件19	条件：TIM6_TRIGB端口上采样到下降沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b18	HSTP18	硬件停止条件18	条件：TIM6_TRIGB端口上采样到上升沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b17	HSTP17	硬件停止条件17	条件：TIM6_TRIGA端口上采样到下降沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b16	HSTP16	硬件停止条件16	条件：TIM6_TRIGA端口上采样到上升沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	HSTP11	硬件停止条件11	条件：内部硬件触发事件3有效 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b10	HSTP10	硬件停止条件10	条件：内部硬件触发事件2有效	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	
b9	HSTP9	硬件停止条件9	条件：内部硬件触发事件1有效 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b8	HSTP8	硬件停止条件8	条件：内部硬件触发事件0有效 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b7	STPS	硬件停止使能	0：硬件停止无效 1：硬件停止有效 注：硬件停止有效时，SSTPR的设定无效	R/W
b6~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	HSTP3	硬件停止条件3	条件：TIM6_<t>_PWMB端口上采样到下降沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b2	HSTP2	硬件停止条件2	条件：TIM6_<t>_PWMB端口上采样到上升沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b1	HSTP1	硬件停止条件1	条件：TIM6_<t>_PWMA端口上采样到下降沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b0	HSTP0	硬件停止条件0	条件：TIM6_<t>_PWMA端口上采样到上升沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W

20.5.19 硬件清零事件选择寄存器（TMR6_HCLRR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HCLE 23	HCLE 22	HCLE 21	HCLE 20	HCLE 19	HCLE 18	HCLE 17	HCLE 16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HCLE 11	HCLE 10	HCLE 9	HCLE 8	CLES	-	-	-	HCLE 3	HCLE 2	HCLE 1	HCLE 0

位	标记	位名	功能	读写
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	HCLE23	硬件清零条件23	条件：TIM6_TRIGD端口上采样到下降沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b22	HCLE22	硬件清零条件22	条件：TIM6_TRIGD端口上采样到上升沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b21	HCLE21	硬件清零条件21	条件：TIM6_TRIGC端口上采样到下降沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b20	HCLE20	硬件清零条件20	条件：TIM6_TRIGC端口上采样到上升沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b19	HCLE19	硬件清零条件19	条件：TIM6_TRIGB端口上采样到下降沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b18	HCLE18	硬件清零条件18	条件：TIM6_TRIGB端口上采样到上升沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b17	HCLE17	硬件清零条件17	条件：TIM6_TRIGA端口上采样到下降沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b16	HCLE16	硬件清零条件16	条件：TIM6_TRIGA端口上采样到上升沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	HCLE11	硬件清零条件11	条件：内部硬件触发事件3有效 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b10	HCLE10	硬件清零条件10	条件：内部硬件触发事件2有效	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	
b9	HCLE9	硬件清零条件9	条件：内部硬件触发事件1有效 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b8	HCLE8	硬件清零条件8	条件：内部硬件触发事件0有效 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b7	CLES	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
b6~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	HCLE3	硬件清零条件3	条件：TIM6_<t>_PWMB端口上采样到下降沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b2	HCLE2	硬件清零条件2	条件：TIM6_<t>_PWMB端口上采样到上升沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b1	HCLE1	硬件清零条件1	条件：TIM6_<t>_PWMA端口上采样到下降沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b0	HCLE0	硬件清零条件0	条件：TIM6_<t>_PWMA端口上采样到上升沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W

20.5.20 硬件刷新事件选择寄存器（TMR6_HUPDR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HUPD23	HUPD22	HUPD21	HUPD20	HUPD19	HUPD18	HUPD17	HUPD16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HUPD11	HUPD10	HUPD9	HUPD8	UPDS	-	-	-	HUPD3	HUPD2	HUPD1	HUPD0

位	标记	位名	功能	读写
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	HUPD23	硬件刷新条件23	条件：TIM6_TRIGD端口上采样到下降沿 0：条件匹配时，硬件刷新无效 1：条件匹配时，硬件刷新有效	R/W
b22	HUPD22	硬件刷新条件22	条件：TIM6_TRIGD端口上采样到上升沿 0：条件匹配时，硬件刷新无效 1：条件匹配时，硬件刷新有效	R/W
b21	HUPD21	硬件刷新条件21	条件：TIM6_TRIGC端口上采样到下降沿 0：条件匹配时，硬件刷新无效 1：条件匹配时，硬件刷新有效	R/W
b20	HUPD20	硬件刷新条件20	条件：TIM6_TRIGC端口上采样到上升沿 0：条件匹配时，硬件刷新无效 1：条件匹配时，硬件刷新有效	R/W
b19	HUPD19	硬件刷新条件19	条件：TIM6_TRIGB端口上采样到下降沿 0：条件匹配时，硬件刷新无效 1：条件匹配时，硬件刷新有效	R/W
b18	HUPD18	硬件刷新条件18	条件：TIM6_TRIGB端口上采样到上升沿 0：条件匹配时，硬件刷新无效 1：条件匹配时，硬件刷新有效	R/W
b17	HUPD17	硬件刷新条件17	条件：TIM6_TRIGA端口上采样到下降沿 0：条件匹配时，硬件刷新无效 1：条件匹配时，硬件刷新有效	R/W
b16	HUPD16	硬件刷新条件16	条件：TIM6_TRIGA端口上采样到上升沿 0：条件匹配时，硬件刷新无效 1：条件匹配时，硬件刷新有效	R/W
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	HUPD11	硬件刷新条件11	条件：内部硬件触发事件3有效 0：条件匹配时，硬件刷新无效 1：条件匹配时，硬件刷新有效	R/W
b10	HUPD10	硬件刷新条件10	条件：内部硬件触发事件2有效	R/W

			0：条件匹配时，硬件刷新无效 1：条件匹配时，硬件刷新有效	
b9	HUPD9	硬件刷新条件9	条件：内部硬件触发事件1有效 0：条件匹配时，硬件刷新无效 1：条件匹配时，硬件刷新有效	R/W
b8	HUPD8	硬件刷新条件8	条件：内部硬件触发事件0有效 0：条件匹配时，硬件刷新无效 1：条件匹配时，硬件刷新有效	R/W
b7	UPDS	硬件刷新使能	0：硬件刷新无效 1：硬件刷新有效 注：硬件刷新有效时，SUPDR的设定无效	R/W
b6~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	HUPD3	硬件刷新条件3	条件：TIM6_<t>_PWMB端口上采样到下降沿 0：条件匹配时，硬件刷新无效 1：条件匹配时，硬件刷新有效	R/W
b2	HUPD2	硬件刷新条件2	条件：TIM6_<t>_PWMB端口上采样到上升沿 0：条件匹配时，硬件刷新无效 1：条件匹配时，硬件刷新有效	R/W
b1	HUPD1	硬件刷新条件1	条件：TIM6_<t>_PWMA端口上采样到下降沿 0：条件匹配时，硬件刷新无效 1：条件匹配时，硬件刷新有效	R/W
b0	HUPD0	硬件刷新条件0	条件：TIM6_<t>_PWMA端口上采样到上升沿 0：条件匹配时，硬件刷新无效 1：条件匹配时，硬件刷新有效	R/W

20.5.21 硬件捕获事件选择寄存器（TMR6_HCPAR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HCPA23	HCPA22	HCPA21	HCPA20	HCPA19	HCPA18	HCPA17	HCPA16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HCPA11	HCPA10	HCPA9	HCPA8	-	-	-	-	HCPA3	HCPA2	HCPA1	HCPA0

位	标记	位名	功能	读写
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	HCPA23	硬件捕获A条件23	条件：TIM6_TRIGD端口上采样到下降沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效	R/W
b22	HCPA22	硬件捕获A条件22	条件：TIM6_TRIGD端口上采样到上升沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效	R/W
b21	HCPA21	硬件捕获A条件21	条件：TIM6_TRIGC端口上采样到下降沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效	R/W
b20	HCPA20	硬件捕获A条件20	条件：TIM6_TRIGC端口上采样到上升沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效	R/W
b19	HCPA19	硬件捕获A条件19	条件：TIM6_TRIGB端口上采样到下降沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效	R/W
b18	HCPA18	硬件捕获A条件18	条件：TIM6_TRIGB端口上采样到上升沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效	R/W
b17	HCPA17	硬件捕获A条件17	条件：TIM6_TRIGA端口上采样到下降沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效	R/W
b16	HCPA16	硬件捕获A条件16	条件：TIM6_TRIGA端口上采样到上升沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效	R/W
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	HCPA11	硬件捕获A条件11	条件：内部硬件触发事件3有效 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效	R/W
b10	HCPA10	硬件捕获A条件10	条件：内部硬件触发事件2有效	R/W

			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	
b9	HCPA9	硬件捕获A条件9	条件：内部硬件触发事件1有效 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效	R/W
b8	HCPA8	硬件捕获A条件8	条件：内部硬件触发事件0有效 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效	R/W
b7~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	HCPA3	硬件捕获A条件3	条件：TIM6_<t>_PWMB端口上采样到下降沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效	R/W
b2	HCPA2	硬件捕获A条件2	条件：TIM6_<t>_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
b1	HCPA1	硬件捕获A条件1	条件：TIM6_<t>_PWMA端口上采样到下降沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效	R/W
b0	HCPA0	硬件捕获A条件0	条件：TIM6_<t>_PWMA端口上采样到上升沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效	R/W

20.5.22 硬件捕获事件选择寄存器（TMR6_HCPBR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HCPB23	HCPB22	HCPB21	HCPB20	HCPB19	HCPB18	HCPB17	HCPB16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HCPB11	HCPB10	HCPB9	HCPB8	-	-	-	-	HCPB3	HCPB2	HCPB1	HCPB0

位	标记	位名	功能	读写
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	HCPB23	硬件捕获B条件23	条件：TIM6_TRIGD端口上采样到下降沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效	R/W
b22	HCPB22	硬件捕获B条件22	条件：TIM6_TRIGD端口上采样到上升沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效	R/W
b21	HCPB21	硬件捕获B条件21	条件：TIM6_TRIGC端口上采样到下降沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效	R/W
b20	HCPB20	硬件捕获B条件20	条件：TIM6_TRIGC端口上采样到上升沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效	R/W
b19	HCPB19	硬件捕获B条件19	条件：TIM6_TRIGB端口上采样到下降沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效	R/W
b18	HCPB18	硬件捕获B条件18	条件：TIM6_TRIGB端口上采样到上升沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效	R/W
b17	HCPB17	硬件捕获B条件17	条件：TIM6_TRIGA端口上采样到下降沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效	R/W
b16	HCPB16	硬件捕获B条件16	条件：TIM6_TRIGA端口上采样到上升沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效	R/W
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	HCPB11	硬件捕获B条件11	条件：内部硬件触发事件3有效 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效	R/W
b10	HCPB10	硬件捕获B条件10	条件：内部硬件触发事件2有效	R/W

			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	
b9	HCPB9	硬件捕获B条件9	条件：内部硬件触发事件1有效 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效	R/W
b8	HCPB8	硬件捕获B条件8	条件：内部硬件触发事件0有效 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效	R/W
b7~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	HCPB3	硬件捕获B条件3	条件：TIM6_<t>_PWMB端口上采样到下降沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效	R/W
b2	HCPB2	硬件捕获B条件2	条件：TIM6_<t>_PWMB端口上采样到上升沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效	R/W
b1	HCPB1	硬件捕获B条件1	条件：TIM6_<t>_PWMA端口上采样到下降沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效	R/W
b0	HCPB0	硬件捕获B条件0	条件：TIM6_<t>_PWMA端口上采样到上升沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效	R/W

20.5.23 硬件递加事件选择寄存器（TMR6_HCUPR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HCUP23	HCUP22	HCUP21	HCUP20	HCUP19	HCUP18	HCUP17	HCUP16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HCUP11	HCUP10	HCUP9	HCUP8	HCUP7	HCUP6	HCUP5	HCUP4	HCUP3	HCUP2	HCUP1	HCUP0

位	标记	位名	功能	读写
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	HCUP23	硬件递加条件23	条件：TIM6_TRIGD端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b22	HCUP22	硬件递加条件22	条件：TIM6_TRIGD端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b21	HCUP21	硬件递加条件21	条件：TIM6_TRIGC端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b20	HCUP20	硬件递加条件20	条件：TIM6_TRIGC端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b19	HCUP19	硬件递加条件19	条件：TIM6_TRIGB端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	
b18	HCUP18	硬件递加条件18	条件：TIM6_TRIGB端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	
b17	HCUP17	硬件递加条件17	条件：TIM6_TRIGA端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	
b16	HCUP16	硬件递加条件16	条件：TIM6_TRIGA端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	HCUP11	硬件递加条件11	条件：内部硬件触发事件3有效 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b10	HCUP10	硬件递加条件10	条件：内部硬件触发事件2有效	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	
b9	HCUP9	硬件递加条件9	条件：内部硬件触发事件1有效 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b8	HCUP8	硬件递加条件8	条件：内部硬件触发事件0有效 0：条件匹配时，硬件递加无效 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 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：TIM6_<t>_PWMB端口为低电平时， TIM6_<t>_PWMA端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b3	HCUP3	硬件递加条件3	条件：TIM6_<t>_PWMA端口为高电平时， TIM6_<t>_PWMB端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b2	HCUP2	硬件递加条件2	条件：TIM6_<t>_PWMA端口为高电平时， TIM6_<t>_PWMB端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b1	HCUP1	硬件递加条件1	条件：TIM6_<t>_PWMA端口为低电平时， TIM6_<t>_PWMB端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b0	HCUP0	硬件递加条件0	条件：TIM6_<t>_PWMA端口为低电平时， TIM6_<t>_PWMB端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W

20.5.24 硬件递减事件选择寄存器（TMR6_HCDOR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HCDO23	HCDO22	HCDO21	HCDO20	HCDO19	HCDO18	HCDO17	HCDO16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HCDO11	HCDO10	HCDO9	HCDO8	HCDO7	HCDO6	HCDO5	HCDO4	HCDO3	HCDO2	HCDO1	HCDO0

位	标记	位名	功能	读写
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	HCDO23	硬件递减条件23	条件：TIM6_TRIGD端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b22	HCDO22	硬件递减条件22	条件：TIM6_TRIGD端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b21	HCDO21	硬件递减条件21	条件：TIM6_TRIGC端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b20	HCDO20	硬件递减条件20	条件：TIM6_TRIGC端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b19	HCDO19	硬件递减条件19	条件：TIM6_TRIGB端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	
b18	HCDO18	硬件递减条件18	条件：TIM6_TRIGB端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	
b17	HCDO17	硬件递减条件17	条件：TIM6_TRIGA端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	
b16	HCDO16	硬件递减条件16	条件：TIM6_TRIGA端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	HCDO11	硬件递减条件11	条件：内部硬件触发事件3有效 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b10	HCDO10	硬件递减条件10	条件：内部硬件触发事件2有效	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	
b9	HCDO9	硬件递减条件9	条件：内部硬件触发事件1有效 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b8	HCDO8	硬件递减条件8	条件：内部硬件触发事件0有效 0：条件匹配时，硬件递减无效 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 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：TIM6_<t>_PWMB端口为低电平时， TIM6_<t>_PWMA端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b3	HCDO3	硬件递减条件3	条件：TIM6_<t>_PWMA端口为高电平时， TIM6_<t>_PWMB端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b2	HCDO2	硬件递减条件2	条件：TIM6_<t>_PWMA端口为高电平时， TIM6_<t>_PWMB端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b1	HCDO1	硬件递减条件1	条件：TIM6_<t>_PWMA端口为低电平时， TIM6_<t>_PWMB端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b0	HCDO0	硬件递减条件0	条件：TIM6_<t>_PWMA端口为低电平时， TIM6_<t>_PWMB端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W

20.5.25 硬件触发事件选择寄存器（TMR6_HTSSRm）（m=0~3）

复位值: 0x000001FFh

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1	0：禁止 AOS_COMTRG1 的公共触发事件触发 Timer6 1：允许 AOS_COMTRG1 的公共触发事件触发 Timer6 详见自动运行系统（AOS）章节	R/W
b30	COMEN[0]	公共触发使能0	0：禁止 AOS_COMTRG2 的公共触发事件触发 Timer6 1：允许 AOS_COMTRG2 的公共触发事件触发 Timer6 详见自动运行系统（AOS）章节	R/W
b29~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8~b0	TRGSEL	触发源选择	触发源编号写入，参考【中断控制器（INTC）】章节	R/W

20.5.26 软件同步启动控制寄存器（TMR6_SSTAR）

复位值: 0x00000000h

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								SSTA8	SSTA7	SSTA6	SSTA5	SSTA4	SSTA3	SSTA2	SSTA1

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	SSTA8	单元8软件启动	0：软件启动无效 1：软件启动使能	R/W
b6	SSTA7	单元7软件启动	0：软件启动无效 1：软件启动使能	R/W
b5	SSTA6	单元6软件启动	0：软件启动无效 1：软件启动使能	R/W
b4	SSTA5	单元5软件启动	0：软件启动无效 1：软件启动使能	R/W
b3	SSTA4	单元4软件启动	0：软件启动无效 1：软件启动使能	R/W
b2	SSTA3	单元3软件启动	0：软件启动无效 1：软件启动使能	R/W
b1	SSTA2	单元2软件启动	0：软件启动无效 1：软件启动使能	R/W
b0	SSTA1	单元1软件启动	0：软件启动无效 1：软件启动使能	R/W

20.5.27 软件同步停止控制寄存器（TMR6_SSTPR）

复位值: 0x00000000h

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								SSTP8	SSTP7	SSTP6	SSTP5	SSTP4	SSTP3	SSTP2	SSTP1

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	SSTP8	单元8软件停止	0：软件停止无效 1：软件停止使能	R/W
b6	SSTP7	单元7软件停止	0：软件停止无效 1：软件停止使能	R/W
b5	SSTP6	单元6软件停止	0：软件停止无效 1：软件停止使能	R/W
b4	SSTP5	单元5软件停止	0：软件停止无效 1：软件停止使能	R/W
b3	SSTP4	单元4软件停止	0：软件停止无效 1：软件停止使能	R/W
b2	SSTP3	单元3软件停止	0：软件停止无效 1：软件停止使能	R/W
b1	SSTP2	单元2软件停止	0：软件停止无效 1：软件停止使能	R/W
b0	SSTP1	单元1软件停止	0：软件停止无效 1：软件停止使能	R/W

20.5.28 软件同步清零控制寄存器（TMR6_SCLRR）

复位值: 0x00000000h

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								SCLE8	SCLE7	SCLE6	SCLE5	SCLE4	SCLE3	SCLE2	SCLE1

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	SCLE8	单元8软件清零	0：软件清零无效 1：软件清零使能	R/W
b6	SCLE7	单元7软件清零	0：软件清零无效 1：软件清零使能	R/W
b5	SCLE6	单元6软件清零	0：软件清零无效 1：软件清零使能	R/W
b4	SCLE5	单元5软件清零	0：软件清零无效 1：软件清零使能	R/W
b3	SCLE4	单元4软件清零	0：软件清零无效 1：软件清零使能	R/W
b2	SCLE3	单元3软件清零	0：软件清零无效 1：软件清零使能	R/W
b1	SCLE2	单元2软件清零	0：软件清零无效 1：软件清零使能	R/W
b0	SCLE1	单元1软件清零	0：软件清零无效 1：软件清零使能	R/W

20.5.29 软件同步刷新控制寄存器（TMR6_SUPDR）

复位值: 0x00000000h

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								SUPD8	SUPD7	SUPD6	SUPD5	SUPD4	SUPD3	SUPD2	SUPD1

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	SUPD8	单元8软件刷新	0：软件刷新无效 1：软件刷新使能	R/W
b6	SUPD7	单元7软件刷新	0：软件刷新无效 1：软件刷新使能	R/W
b5	SUPD6	单元6软件刷新	0：软件刷新无效 1：软件刷新使能	R/W
b4	SUPD5	单元5软件刷新	0：软件刷新无效 1：软件刷新使能	R/W
b3	SUPD4	单元4软件刷新	0：软件刷新无效 1：软件刷新使能	R/W
b2	SUPD3	单元3软件刷新	0：软件刷新无效 1：软件刷新使能	R/W
b1	SUPD2	单元2软件刷新	0：软件刷新无效 1：软件刷新使能	R/W
b0	SUPD1	单元1软件刷新	0：软件刷新无效 1：软件刷新使能	R/W

20.6 使用注意事项

1）计数器在计数中时 CPU 不能对 CNTER 寄存器执行写入动作，只能在计数器停止时对 CNTER 寄存器执行写入动作。

2）GCMCR~GCMFR 在对应的缓存功能有效时，其中断输出、事件输出无效。

3）在使用双边对称互补 PWM 输出功能时，用户需计算、设定好相关的基准值和端口输出状态，保证端口为预期的输出状态。

4）在一些特定场合，对计数器的控制事件有可能同时出现，此时，计数器（CNTER）的变化控制优先级如下表 20-5 所示。

计时器控制事件		计数器（CNTER）值	优先级
CPU对CNTER执行写操作		CPU的写入值	高
刷新事件（软件刷新或硬件刷新）		UPDAR中的值	↓
清零事件（软件清零或硬件清零）	硬件计数或锯齿波模式时	根据计数方向变为0或PERAR	↓
	三角波模式时	0	↓
硬件计数		正常计数值	↓
三角波模式计数		正常计数值	↓
锯齿波模式计数		正常计数值	低

表 20-5 计数器（CNTER）控制优先级

5）在一些特定场合，对 PWM 端口输出的控制事件有可能同时出现，此时，两个端口（TIM6_<t>_PWMA 和 TIM6_<t>_PWMB）的输出变化控制优先级分别如下表 20-6 所示、表 20-7 所示。

PWMA端口输出控制事件		端口状态	优先级
EMB选择的事件发生		PCNAR.EMBCA设定状态	高
设定强制端口输出		PCNAR.FORCA设定状态	↓
计数开始时		PCNAR.STACA设定状态	↓
锯齿波向上计数时	CNTER=PERAR	PCNAR.OVFCA设定状态	↓
	CNTER=GCMAR	PCNAR.CMAUCA设定状态	↓
	CNTER=GCMBR	PCNAR.CMBUCA设定状态	↓
锯齿波向下计数时	CNTER=0	PCNAR.UDFCA设定状态	↓

	CNTER=GCMAR	PCNAR.CMADCA设定状态	↓
	CNTER=GCMBR	PCNAR.CMBDCA设定状态	↓
三角波向上计数时	CNTER=GCMAR	PCNAR.CMAUCA设定状态	↓
	CNTER=GCMBR	PCNAR.CMBUCA设定状态	↓
	CNTER=PERAR	PCNAR.OVFCA设定状态	↓
三角波向下计数时	CNTER=GCMAR	PCNAR.CMADCA设定状态	↓
	CNTER=GCMBR	PCNAR.CMBDCA设定状态	↓
	CNTER=0	PCNAR.UDFCA设定状态	↓
计数停止时		PCNAR.STPCA设定状态	低

表 20-6 PWMA 端口输出控制优先级

PWMB端口输出控制事件		**端口状态**	**优先级**
EMB选择的事件发生		PCNBR.EMBCB设定状态	高
设定强制端口输出		PCNBR.FORCB设定状态	↓
计数开始时		PCNBR.STACB设定状态	↓
锯齿波向上计数时	CNTER=PERAR	PCNBR.OVFCB设定状态	↓
	CNTER=GCMBR	PCNBR.CMBUCB设定状态	↓
	CNTER=GCMAR	PCNBR.CMAUCB设定状态	↓
锯齿波向下计数时	CNTER=0	PCNBR.UDFCB设定状态	↓
	CNTER=GCMBR	PCNBR.CMBDCB设定状态	↓
	CNTER=GCMAR	PCNBR.CMADCB设定状态	↓
三角波向上计数时	CNTER=GCMBR	PCNBR.CMBUCB设定状态	↓
	CNTER=GCMAR	PCNBR.CMAUCB设定状态	↓
	CNTER=PERAR	PCNBR.OVFCB设定状态	↓
三角波向下计数时	CNTER=GCMBR	PCNBR.CMBDCB设定状态	↓
	CNTER=GCMAR	PCNBR.CMADCB设定状态	↓
	CNTER=0	PCNBR.UDFCB设定状态	↓
计数停止时		PCNBR.STPCB设定状态	低

表 20-7 PWMB 端口输出控制优先级

21 高精度 PWM（HRPWM）

21.1 简介

高精度 PWM（HRPWM）扩展了 Timer6 的 PWM 信号的分辨率。本模块搭配 TMR6，能够产生最多 16 个通道的高分辨率 PWM 波形。

HRPWM 的特征如下：

- 扩展 PWM 信号分辨率
- 用于调整 PWM 波形的占空比和相位
- 可以用于上升沿、下降沿、上升沿和下降沿控制
- 自动校准功能，用于提供单位迟延量

21.2 基本框图

HRPWM 的基本框图如图 21-1 所示。

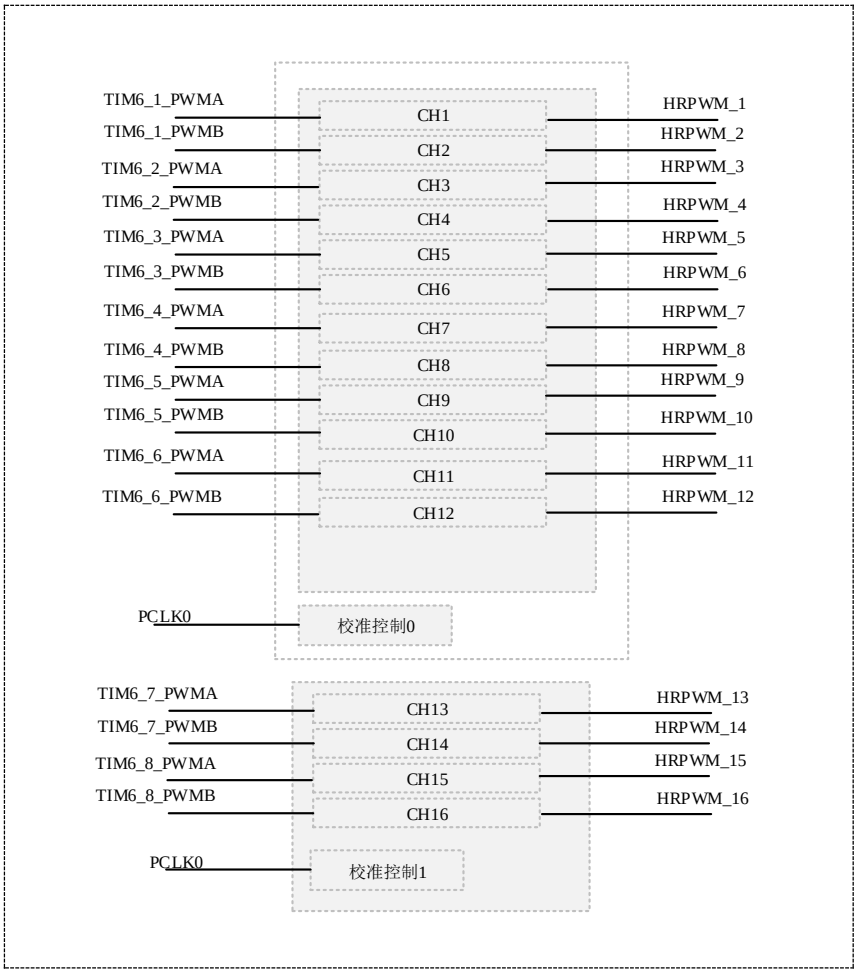


图 21-1 HRPWM 基本框图

21.3 功能说明

21.3.1 校准功能

本模块配置有两个校准控制模块，其中校准控制 0 用于通道 1 到通道 12 的单元迟延量校准，校准控制 1 用于通道 13-通道 16 的单元迟延量校准。校准控制单元与通道 1-16 之间相互独立，因此校准过程不会影响通道 1-16 的功能。通过校准控制单元，可以获得校准码，该校准码表示 1 个 PCLK0 可以分割成多少个单位迟延量。

校准模块使用流程(n=0/1)：

1. 配置使能时钟 PCLK0
2. HRPWM_CALCRn.CALEN=1
3. 等待 HRPWM_CALCRn.ENDF=1
4. 读出 HRPWM_CALCRn.CALCODE[7:0]

21.3.2 高分辨率 PWM 调整功能

HRPWM 可以通过软件配置，分别设置 PWM 信号的上升沿、下降沿、上升和下降沿调整。调整的最小刻度由单位迟延量决定，通过软件可以分别配置上升沿、下降沿的迟延量，可以选择的最小迟延量为单位迟延量，最大为 256 倍的最小迟延量长度。

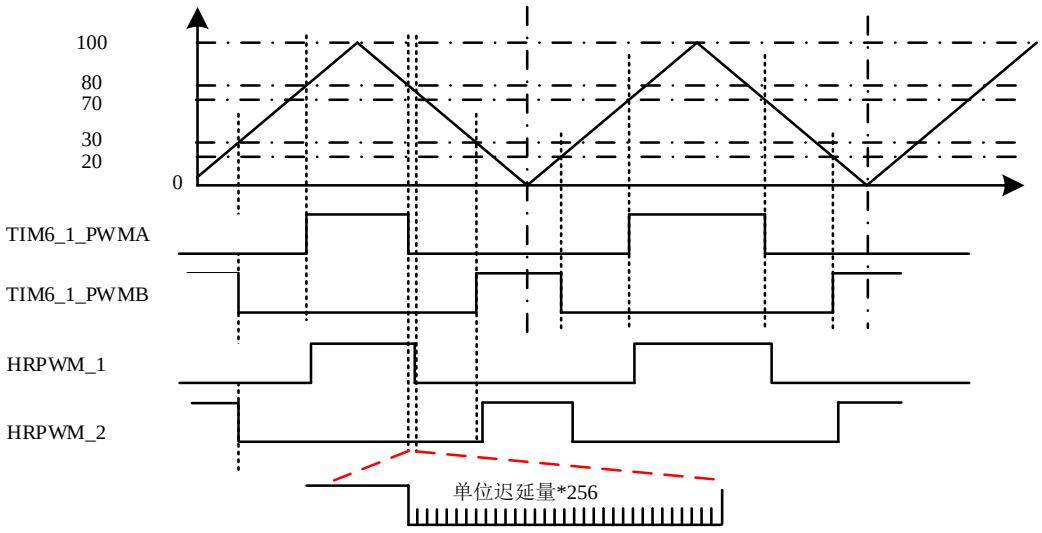


图 21-2 HRPWM 调整波形图

初始配置 HRPWM 模块流程(n=1-16):

1. 根据 21.3.1 校准功能中的使用流程说明，进行校准，获取校准码
2. 配置 HRPWM_CHnCR.NSEL，改变下降沿的迟延调整量
3. 配置 HRPWM_CHnCR.PSEL，改变上升沿的迟延调整量
4. 配置 HRPWM_CHnCR.NE，使能下降沿调整
5. 配置 HRPWM_CHnCR.PE，使能上升沿调整
6. 配置 HRPWM_CHnCR.EN=1，使能 HRPWM 功能

21.3.3 使用注意事项

1. 在 PWM 波形连续输出时，只能在 TIM6 的计数峰点和谷点动态配置 HRPWM_CRn.NSEL 和 HRPWM_CRn.PSEL。
2. PCLK0 的频率必须高于 120MHz。

3. 芯片进入 STOP 模式后，TIM6_n_PWMA/TIM6_n_PWMB(n=1~6)、HRPWM_1~HRPWM_12 输出低电平，不会保持 STOP 模式之前的状态。如果这些功能对应的输出管脚需要维持在 STOP 模式之前的状态，需要配置管脚的功能选择寄存器 PFSR 为通用 GPIO 功能，并通过管脚的通用控制寄存器 PCR 设置成对应的状态。

21.4 寄存器说明

表 21-1 所示，为 HRPWM 模块的寄存器列表。

BASE ADDR：0x4003C000 (HRPWM_BASE_ADDR)

寄存器名	符号	偏移地址	位宽	复位值
HRPWM 单元 1 控制寄存器	HRPWM_CR1	0x00	32	0x00000000h
HRPWM 单元 2 控制寄存器	HRPWM_CR2	0x04	32	0x00000000h
HRPWM 单元 3 控制寄存器	HRPWM_CR3	0x08	32	0x00000000h
HRPWM 单元 4 控制寄存器	HRPWM_CR4	0x0c	32	0x00000000h
HRPWM 单元 5 控制寄存器	HRPWM_CR5	0x10	32	0x00000000h
HRPWM 单元 6 控制寄存器	HRPWM_CR6	0x14	32	0x00000000h
HRPWM 单元 7 控制寄存器	HRPWM_CR7	0x18	32	0x00000000h
HRPWM 单元 8 控制寄存器	HRPWM_CR8	0x1c	32	0x00000000h
HRPWM 单元 9 控制寄存器	HRPWM_CR9	0x20	32	0x00000000h
HRPWM 单元 10 控制寄存器	HRPWM_CR10	0x24	32	0x00000000h
HRPWM 单元 11 控制寄存器	HRPWM_CR11	0x28	32	0x00000000h
HRPWM 单元 12 控制寄存器	HRPWM_CR12	0x2c	32	0x00000000h
HRPWM 单元 13 控制寄存器	HRPWM_CR13	0x30	32	0x00000000h
HRPWM 单元 14 控制寄存器	HRPWM_CR14	0x34	32	0x00000000h
HRPWM 单元 15 控制寄存器	HRPWM_CR15	0x38	32	0x00000000h
HRPWM 单元 16 控制寄存器	HRPWM_CR16	0x3c	32	0x00000000h
HRPWM 校准控制寄存器 0	HRPWM_CALCR0	0x50	32	0x00000000h
HRPWM 校准控制寄存器 1	HRPWM_CALCR1	0x54	32	0x00000000h

表 21-1 寄存器列表

21.4.1 HRPWM 控制寄存器(HRPWM_CRn, n=1…16)

复位值：0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EN	PE	NE	-												

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

位	标记	位名	功能	读写
b31	EN	高分辨率PWM使能	0:高分辨率PWM无效，PWM信号直通 1:高分辨率PWM使能	R/W
b30	PE	HRPWM上升沿调整使能	0: 不使用HRPWM调整上升沿 1: 使用HRPWM调整上升沿	R/W
b29	NE	HRPWM下降沿调整使能	0: 不使用HRPWM调整下降沿 1: 使用HRPWM调整下降沿沿	R/W
b28-b16	Reserved	-	读出为0，写入时写0	R/W
b15-b8	PSEL[7:0]	HRPWM上升沿校准选择	0x00h: 选择一段迟延 0x01h:选择两段迟延 …. 0xfeh:选择255段迟延 0xffh:选择256段迟延	R/W
b7-b0	NSEL[7:0]	HRPWM下降沿校准选择	0x00h: 选择一段迟延 0x01h:选择两段迟延 …. 0xfeh:选择255段迟延 0xffh:选择256段迟延	R/W

21.4.2 HRPWM 校准控制寄存器 0(HRPWM_CALCRn, n=0,1)

复位值：0x00000000h

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
CALEN	-		ENDF	-				CALCODE[7:0]							

位	标记	位名	功能	读写
b31-b16	Reserved	-	读时读出0，写时请写0	R/W
b15	CALEN	校准使能	0：校准功能关闭 1：校准功能使能	R/W
b14-b13	Reserved	-	读时读出0，写时请写0	R/W
b12	ENDF	校准结束Flag	0:校准中 1:校准完成	R
b11-b8	Reserved	-	读时读出0，写时请写0	R/W
b7-b0	CALCODE	校准值	自动校准结果，表示校准用pclk由多少个单位延迟量构成。	R/W

CALCR0 是通道 1 到通道 12 校准使用寄存器。

CALCR1 是通道 13-通道 16 校准使用寄存器。

22 通用控制定时器（Timer4）

22.1 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。

22.2 基本框图

Timer4 基本的功能及特性如表 22-1 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
	• 专用事件输出启动ADC转换
	• 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 22-1 Timer4 的基本功能及特性

图 22-1 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；框图中所示 PCLK 是指 PCLK0，即 Timer4 的内部计数时钟为 PCLK0。

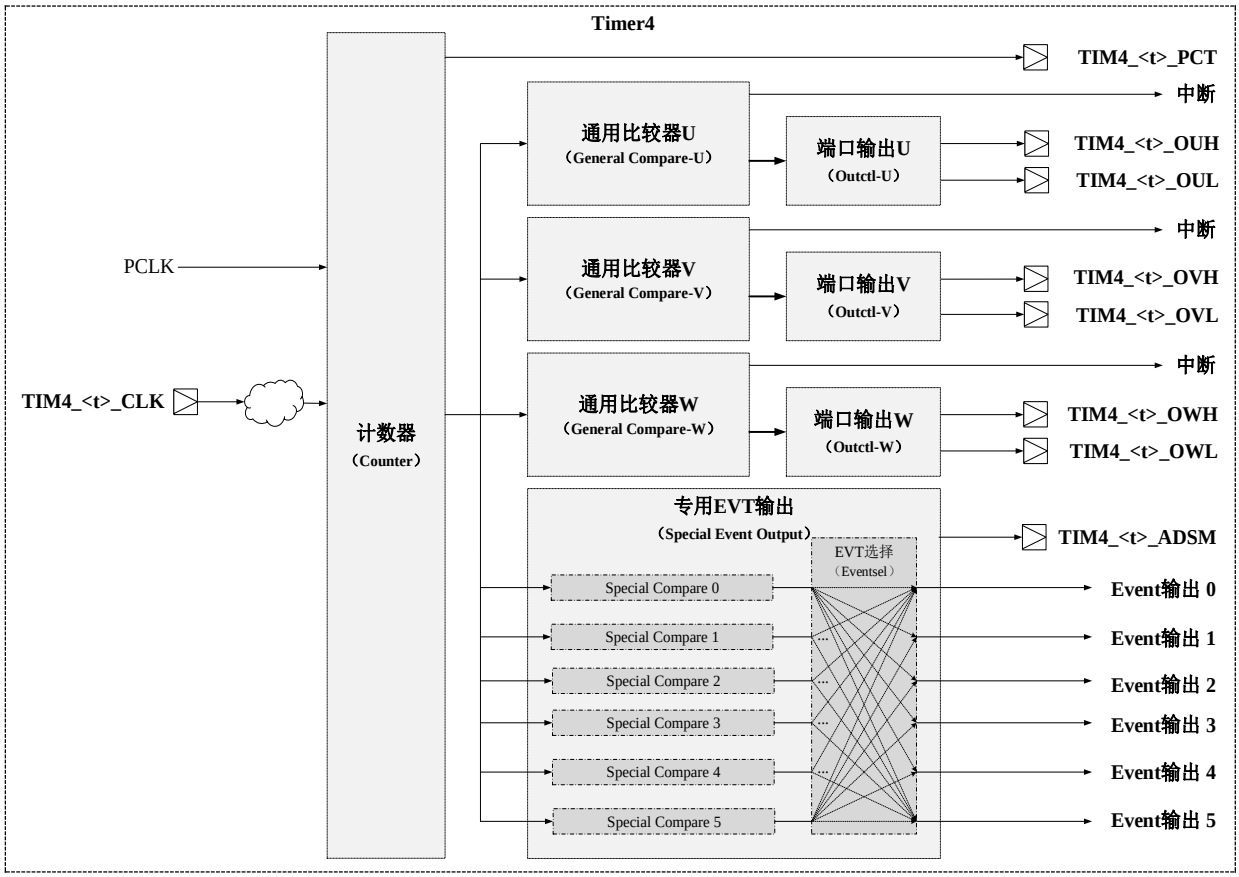


图 22-1 Timer4 基本框图

表 22-2 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	out	PWM输出端口
TIM4_<t>_OUL		
TIM4_<t>_OVH		
TIM4_<t>_OVL		
TIM4_<t>_OWH		
TIM4_<t>_OWL		
TIM4_<t>_PCT	out	PWM周期输出监测
TIM4_<t>_ADSM	out	专用事件输出监测

表 22-2 Timer4 端口列表

22.3 功能说明

22.3.1 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 22-2 和图 22-3 分别是锯齿波和三角波的波形图。

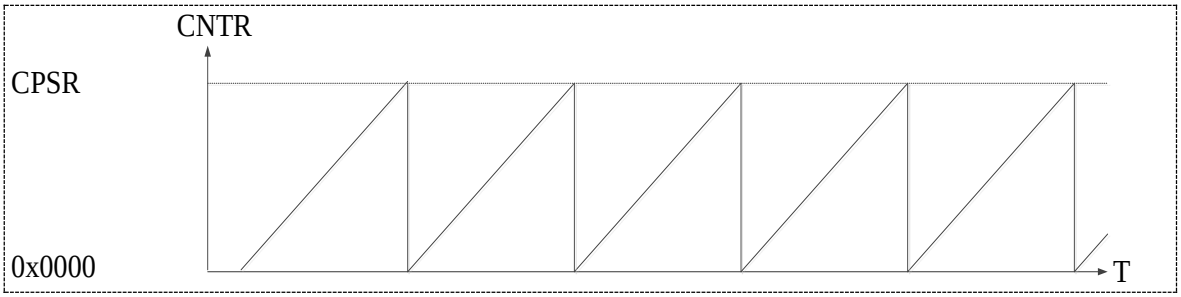


图 22-2 Timer4 锯齿波波形

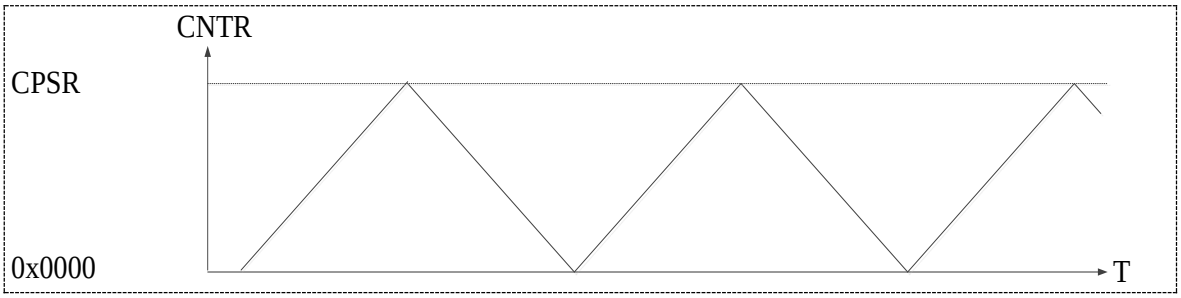


图 22-3 Timer4 三角波波形

22.3.2 计数动作

1) 锯齿波计数操作与控制流程如图 22-4 所示。

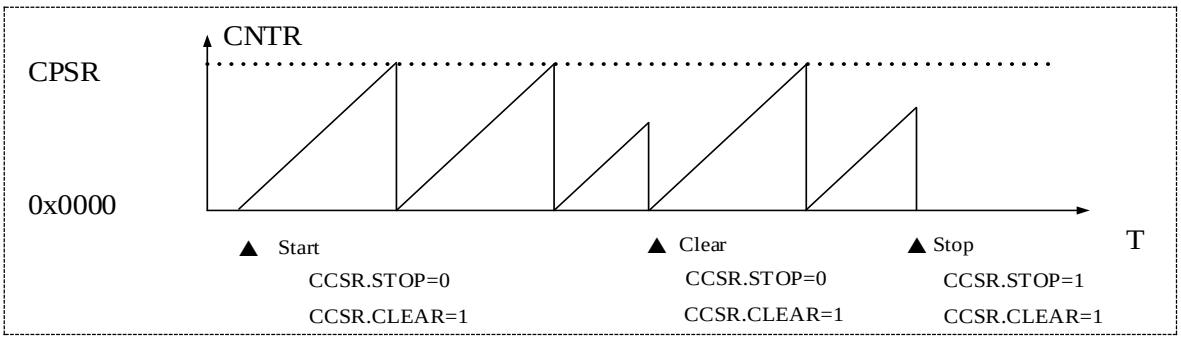


图 22-4 Timer4 锯齿波模式计数动作

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 22-5 所示。

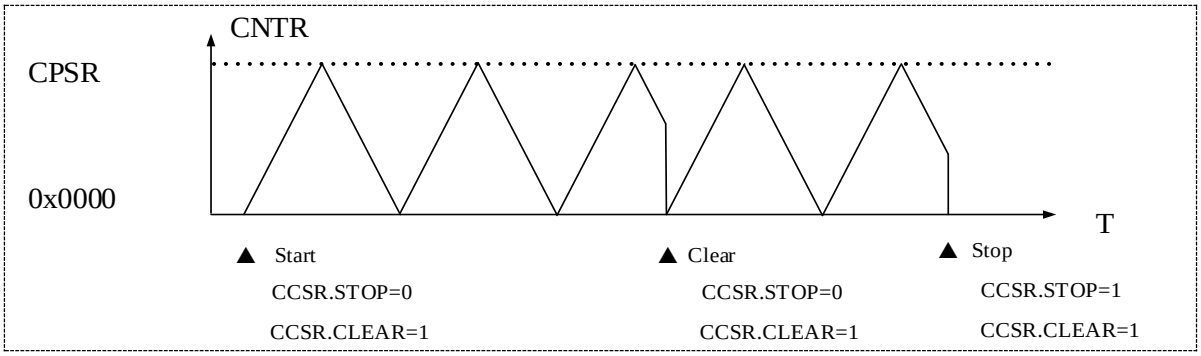


图 22-5 Timer4 三角波模式计数动作

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。

22.3.3 比较输出

1) 下图 22-6 是锯齿波计数模式下的比较输出模块的波形输出示例。

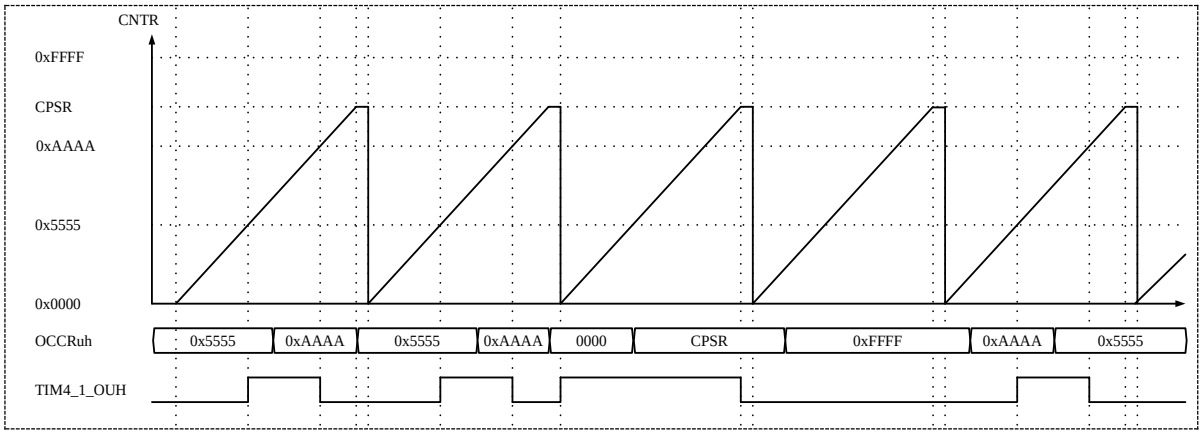


图 22-6 锯齿波模式波形输出例

2) 下图 22-7 是三角波计数模式下的比较输出模块的波形输出示例。

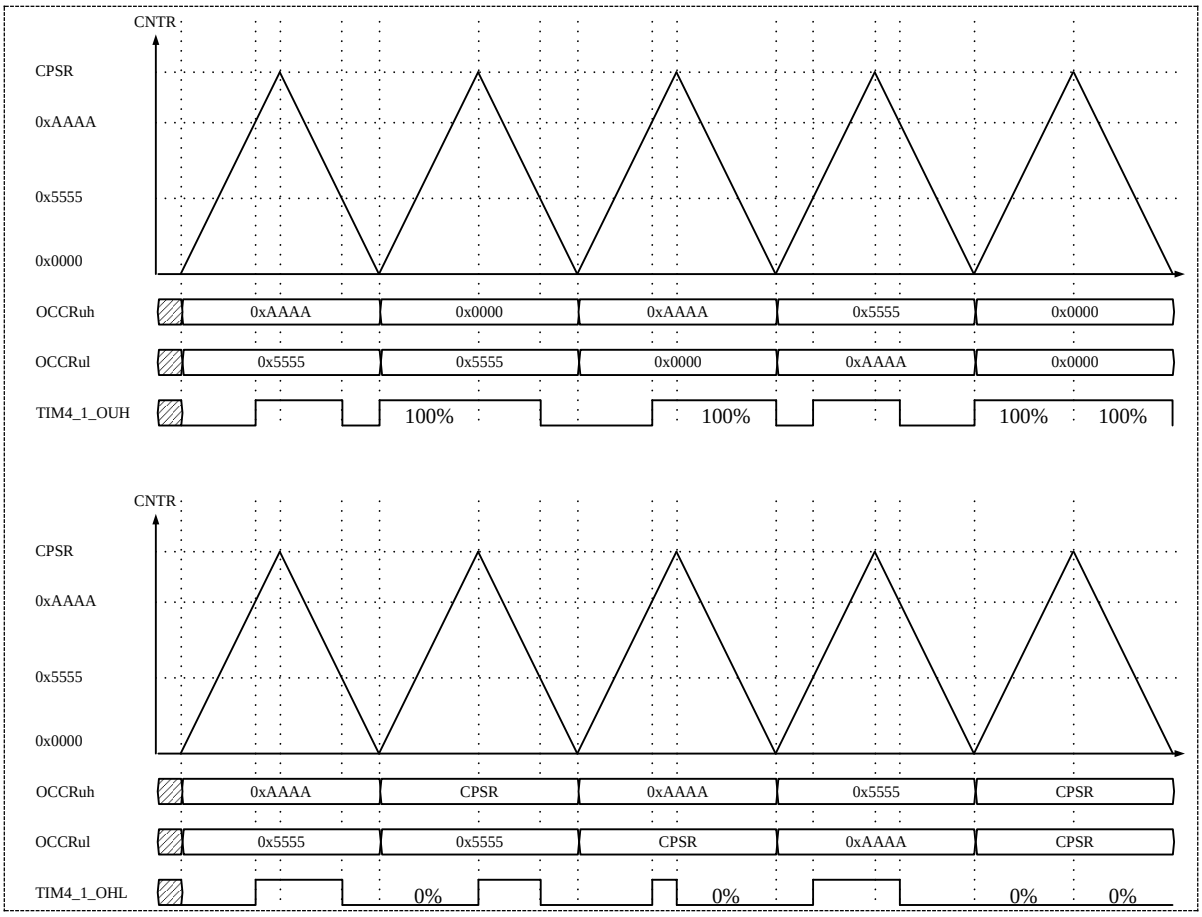


图 22-7 三角波模式波形输出例

22.3.4 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。

22.3.4.1 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。

1) 缓冲功能禁止时（CCSR.BUFEN=0），写入数据立即从缓冲寄存器传输到 CPSR 寄存器。
2) 缓冲功能使能时（CCSR.BUFEN=1），在计数器停止时（CCSR.STOP=1）或者计数器的计数值为"0x0000"时，数据从缓冲寄存器传输到 CPSR 寄存器。

注意：

– 从 CPSR 读取数据时，读取的并非 CPSR 缓冲寄存器的值，而是 CPSR 寄存器的值。当缓冲功能使能时，在传输完成之前所读取的值不是最近写入的值，而是上一次写入的 CPSR 值。

图 22-8 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。

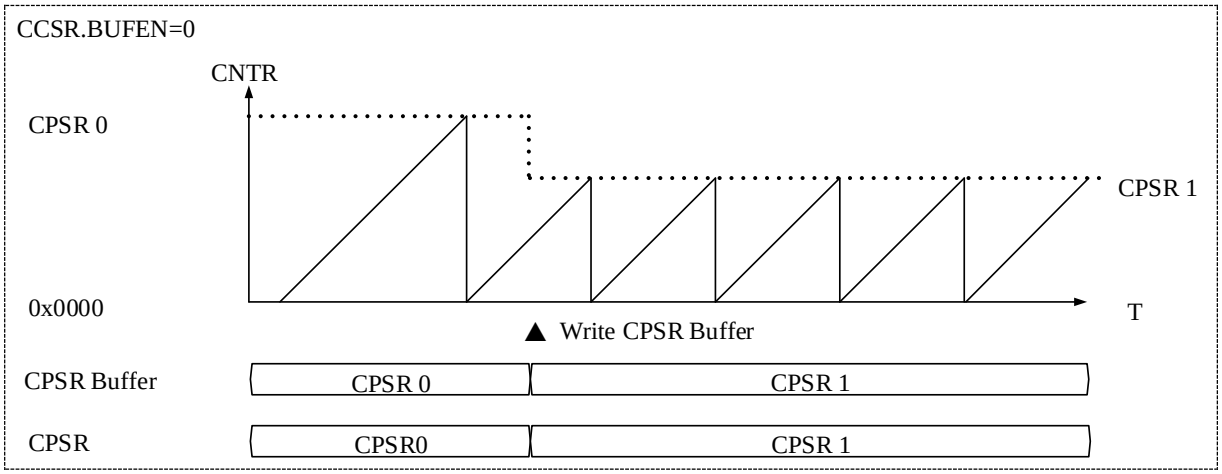


图 22-8 缓存无效时修改锯齿波计数周期

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 22-9 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。

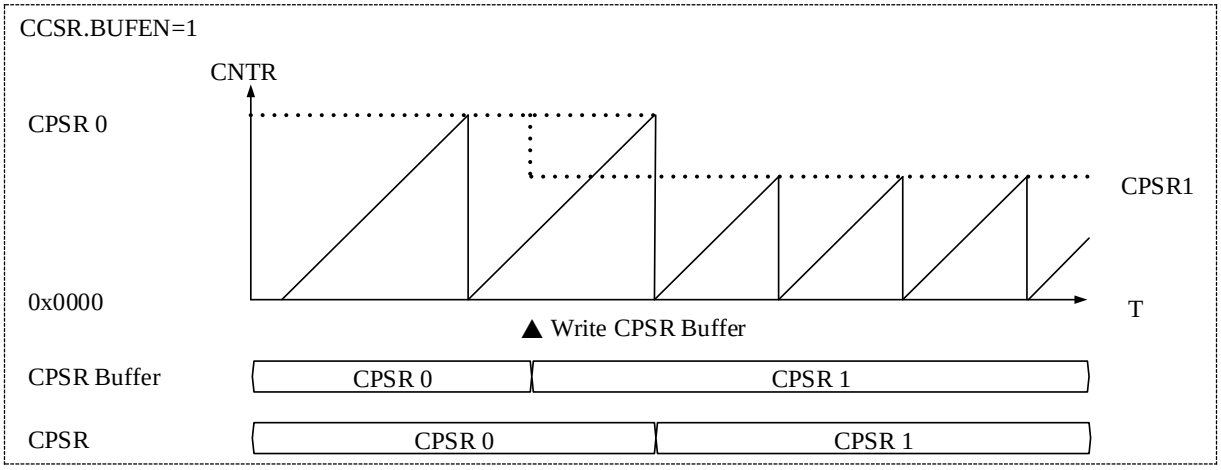


图 22-9 缓存使能时修改锯齿波计数周期

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 22-10 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。

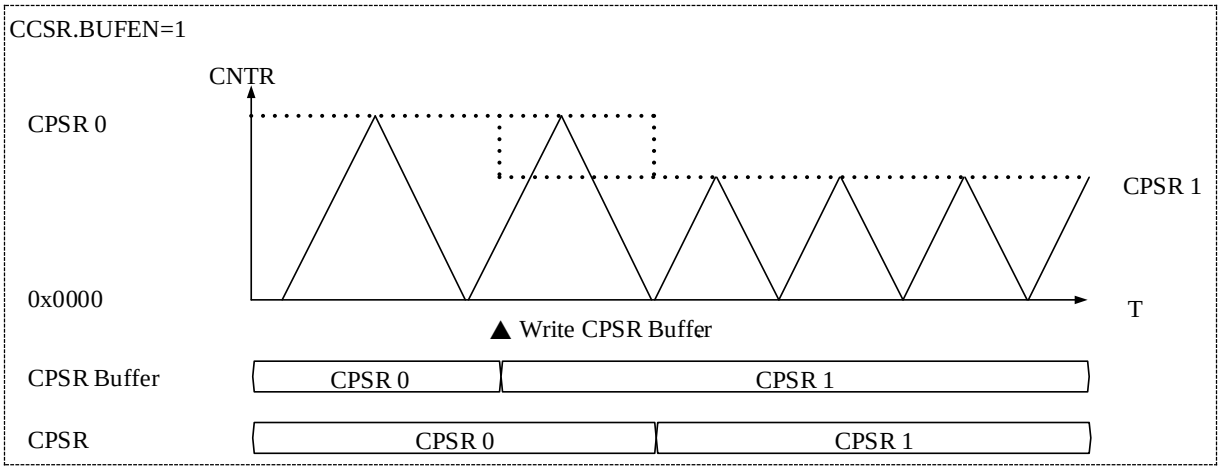


图 22-10 缓存使能时修改三角波计数周期

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。

22.3.4.2 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 22-11 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）。

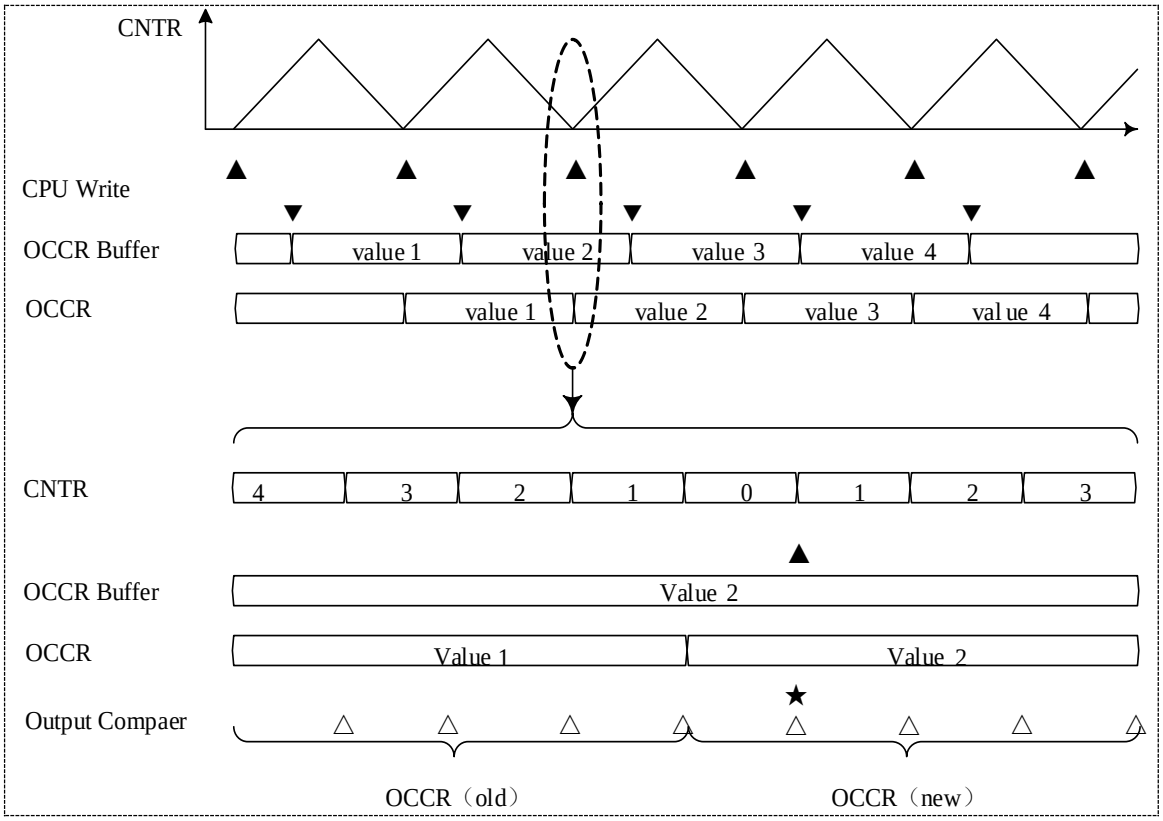


图 22-11 OCCR 缓冲数据传输（周期间隔响应链接禁止时）

图中的上半部分是全局示意图，下半部分是传输操作时放大图。

计数器为三角波计数模式，下溢中断在标志▲（计数谷点）时刻产生。在▼时刻，CPU 改写 OCCR 寄存器，写入的数据存储在 OCCR 缓冲寄存器中。之后当计数器计数值为 0x0000 时，进行数据加载操作，数据从缓冲寄存器到 OCCR 寄存器，同时产生中断标志 IRQZF。

在△时刻，输出比较根据设定的 OCCR 寄存器值与计数值匹配事件，执行改变 PWM 输出和置位 OCSR.OCFx 位（x=L 或 H）。在时刻★（计数值=0x0000）之后，端口输出根据新的 OCCR 数据执行操作。在时刻★之前根据原 OCCR 数据执行操作。

图中示意了 OCCR 缓冲寄存器在计数谷点的传输操作，OCMR 的缓冲寄存器的缓存传送操作与之类似；同理，在计数峰点的传输操作也类似。新的数据在传输时刻之后即

刻生效（新写入数据将控制 PWM 输出和中断标志置位）。

b) 当计数器周期间隔响应链接使能时，缓冲值在设定的计数状态且周期间隔计数器计数值为 0 时执行缓冲寄存器传输操作。

图 22-12 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）。

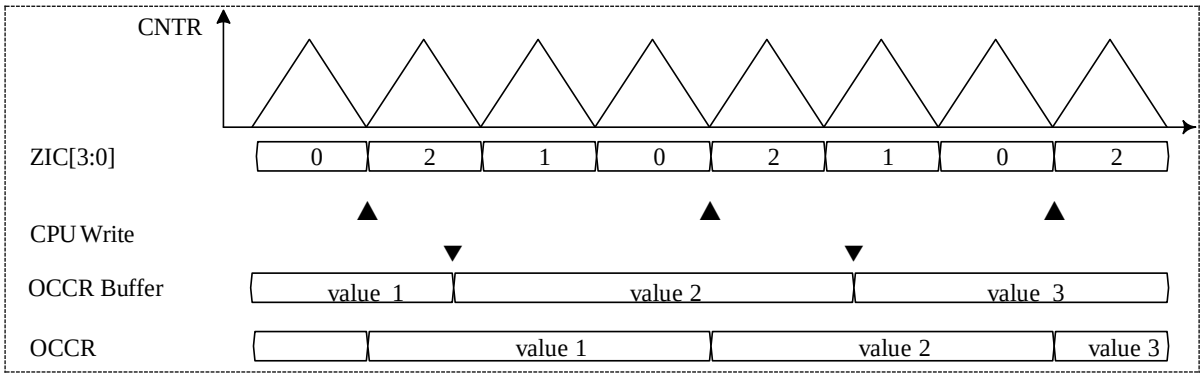


图 22-12 OCCR 缓冲数据传输（周期间隔响应链接使能）

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 22-13 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。

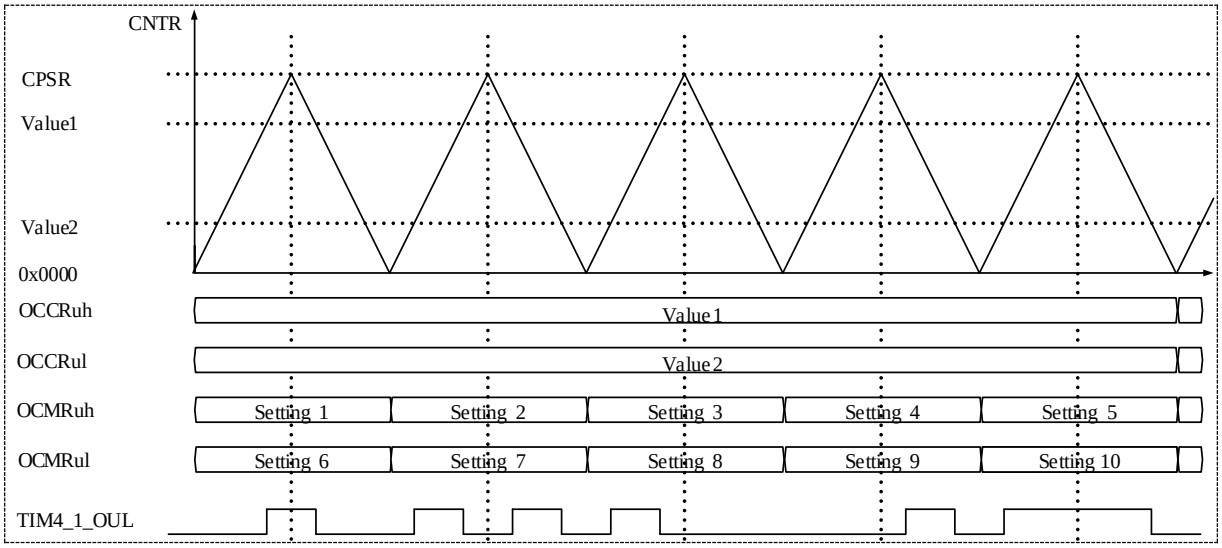


图 22-13 输出比较缓冲数据传输（OCMR 缓冲使能）

22.3.4.3 专用比较寄存器缓存功能

专用比较基准寄存器（SCCR）和在专用模式控制寄存器（SCMR）都带有缓冲功寄存器。当使能缓冲功能时，CPU 写入 SCCR 和 SCMR 缓冲寄存器的值在设定的计数器状态下加载到 SCCR 和 SCMR 寄存器。

a) 当禁止计数器周期间隔响应链接传输时，缓冲传输操作仅与计数器状态有关，不受计数器周期间隔计数器影响。

图 22-14 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。

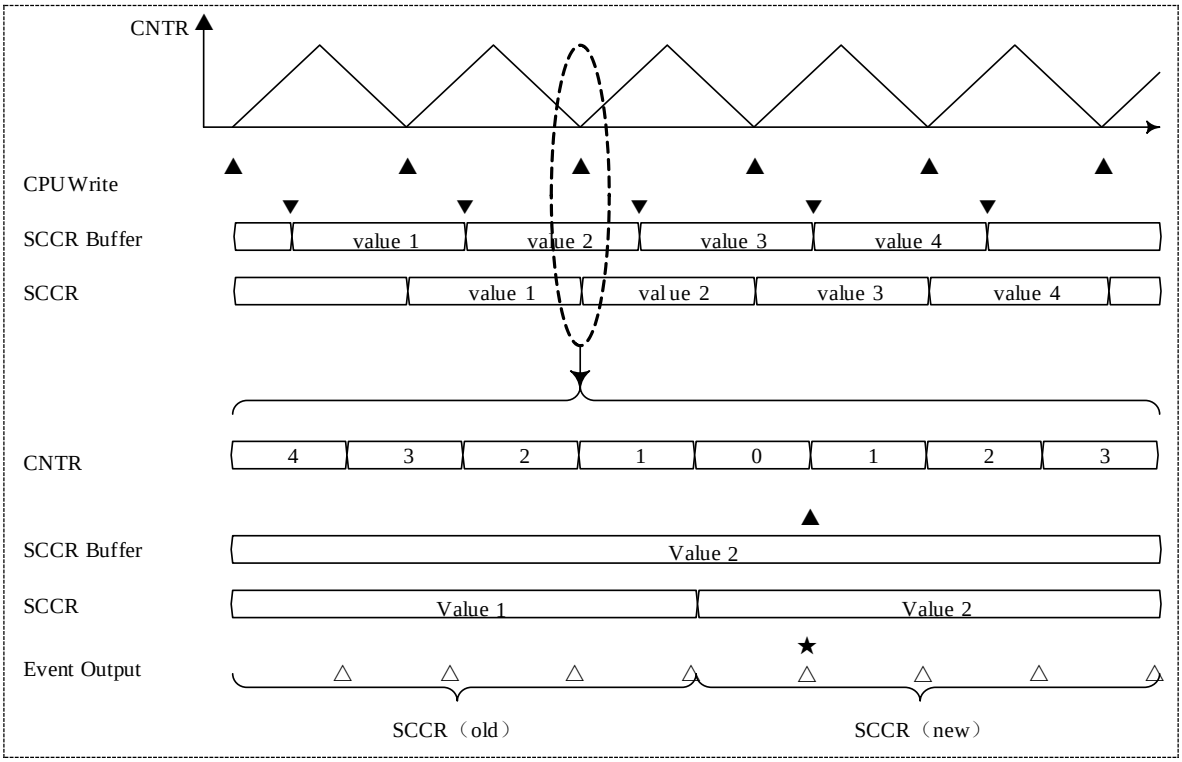


图 22-14 SCCR 缓冲传输操作（周期间隔响应链接传输禁止时）

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 22-15 所示，是使能 SCCR 缓冲功能、计数器零值时加载（SCSR.BUFEN=01），计数器周期期间隔响应链接传输使能时（SCSR.LMC=1）的示意图。

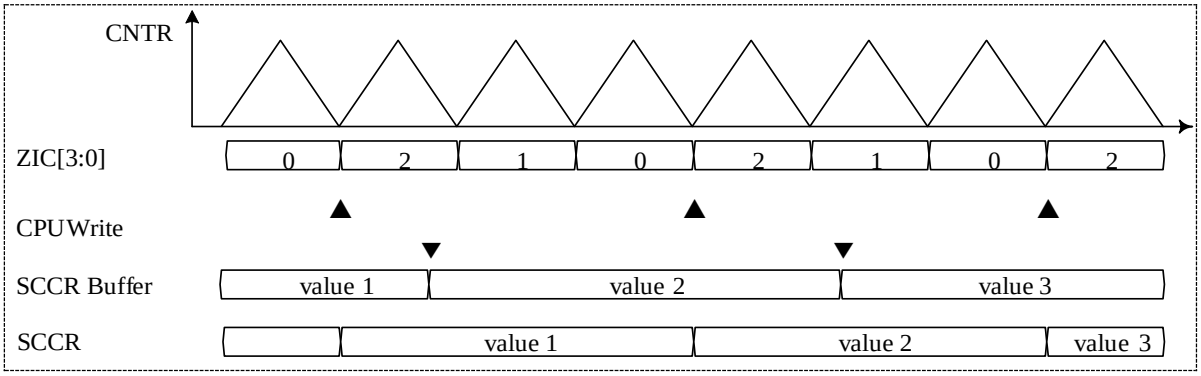


图 22-15 SCCR 缓冲传输操作（周期间隔响应链接传输使能时）

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。

22.3.5 通用 PWM 输出

22.3.5.1 独立 PWM 输出

在直通模式（POCR.PWMMD=00）下,分别设定 OCCRxh、OCCRxl 的基准值和 OCMRxh、OCMRxl 的端口输出状态（x=u、v、w），可实现不同的 PWM 输出。此时，每个端口的 PWM 输出都独立控制。使用独立 PWM 输出模式前，需要设定 MOE、OExy 位以使能 Timer4 的输出，详细描述请参考 PWM 状态控制寄存器（PSCR）。图 22-16 和图 22-17 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）。

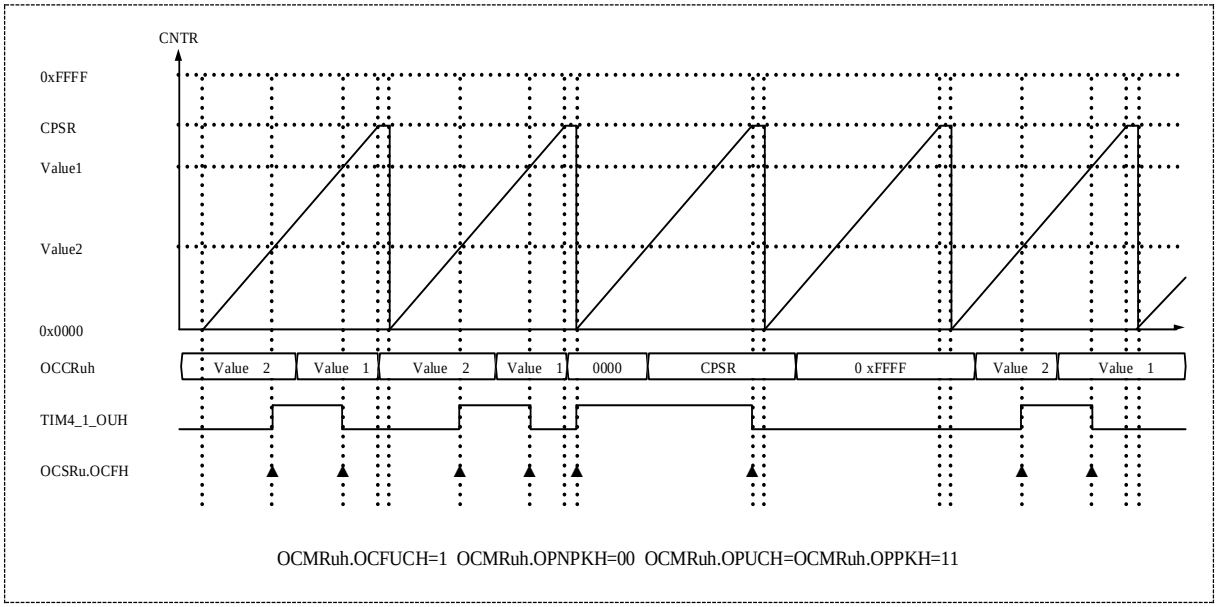


图 22-16 锯齿波独立 PWM 输出例

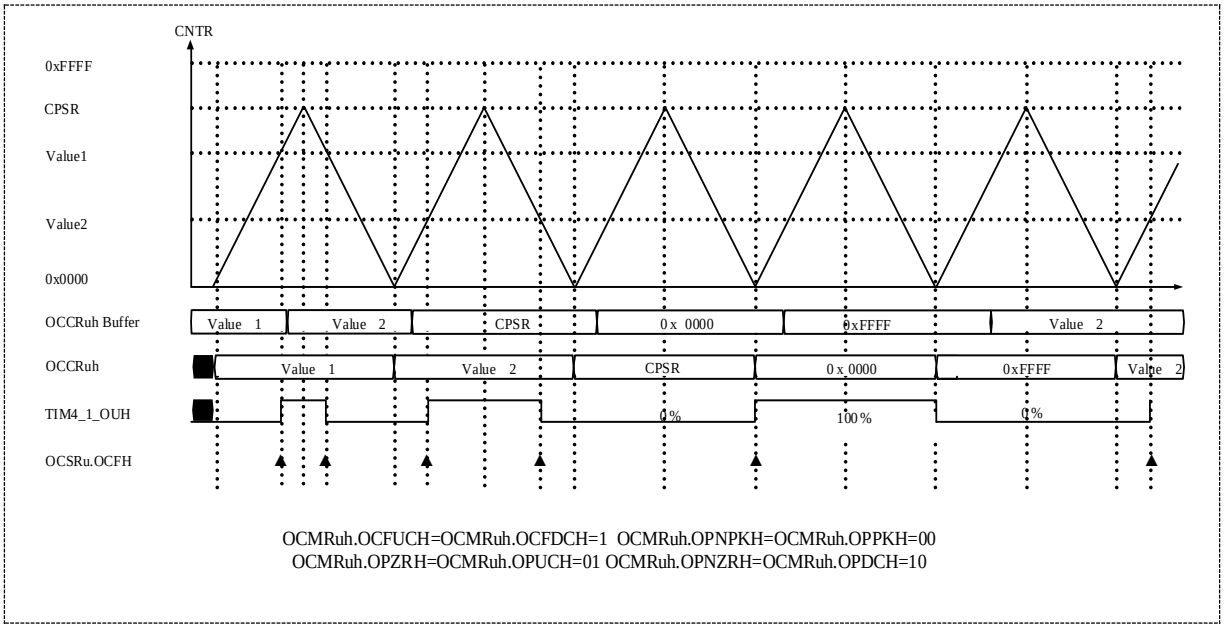


图 22-17 三角波独立 PWM 输出例

22.3.5.2 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 OCCRxh 基准值相关，从而在 TIM4_<t>_OXL 端口上实现扩展 PWM 输出（X=U、V、W，x=u、v、w）。使用扩展 PWM 输出模式前，需要设定 MOE、OExy 位以使能 Timer4 的输出，详细描述请参考 PWM 状态控制寄存器（PSCR）。图 22-18 所示是该方式下 TIM4_<t>_OUH 和 TIM4_<t>_OUL 端口的 PWM 输出。

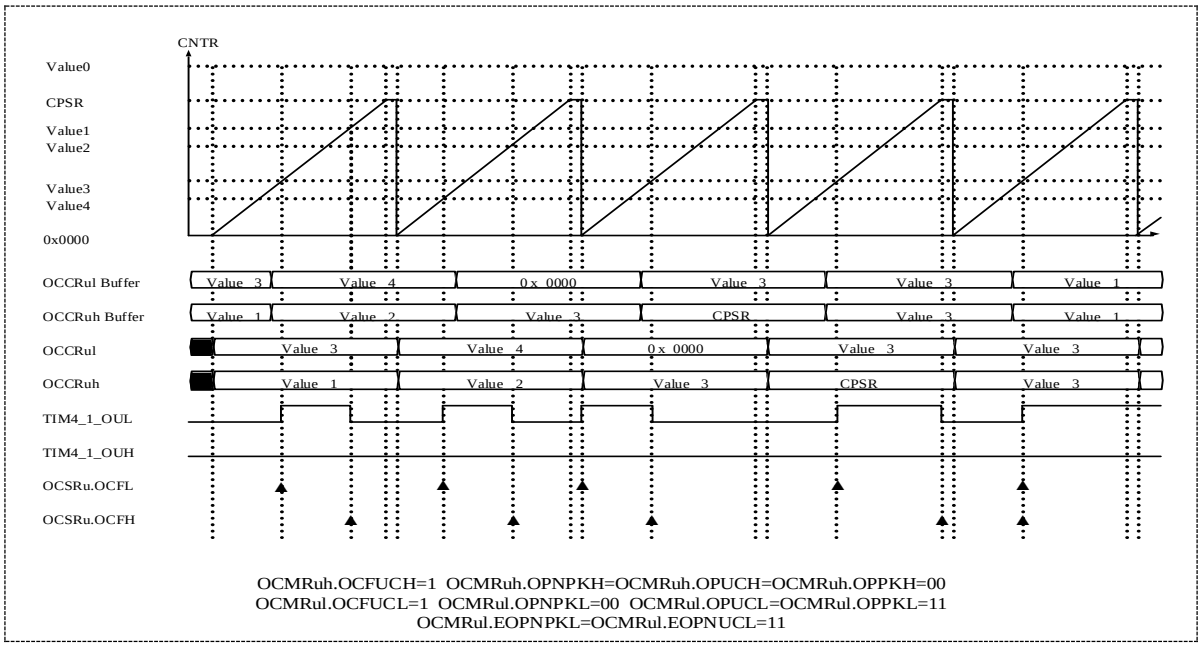


图 22-18 三角波扩展 PWM 输出

注意：

– 独立PWM输出方式下，TIM4_<t>_OXL的端口状态由OCMRxl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）。

22.3.5.3 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 22-19 所示。

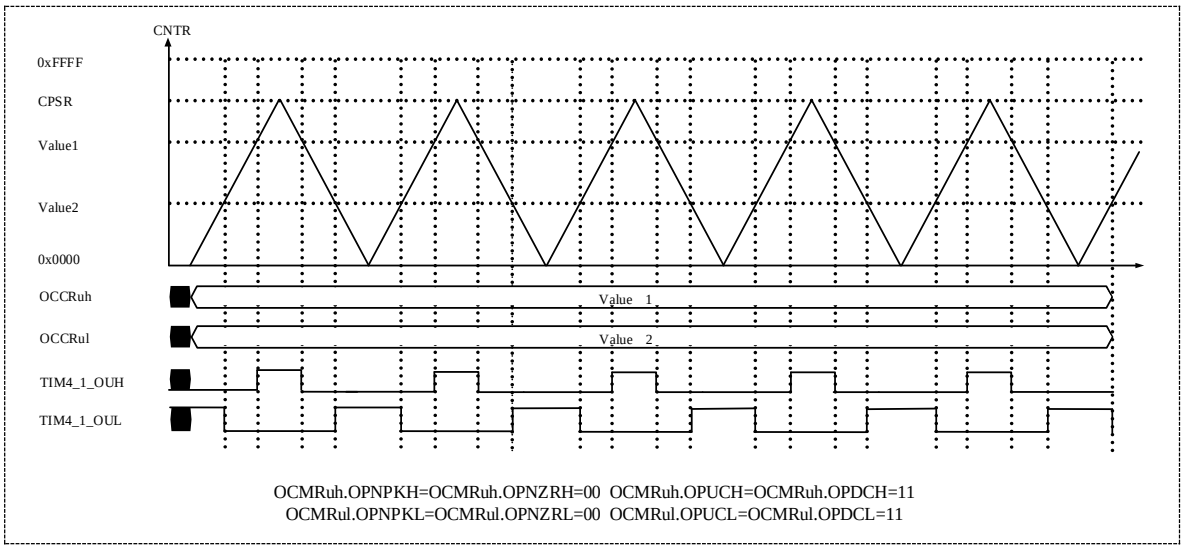


图 22-19 软件实现互补 PWM 输出

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 22-20 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。

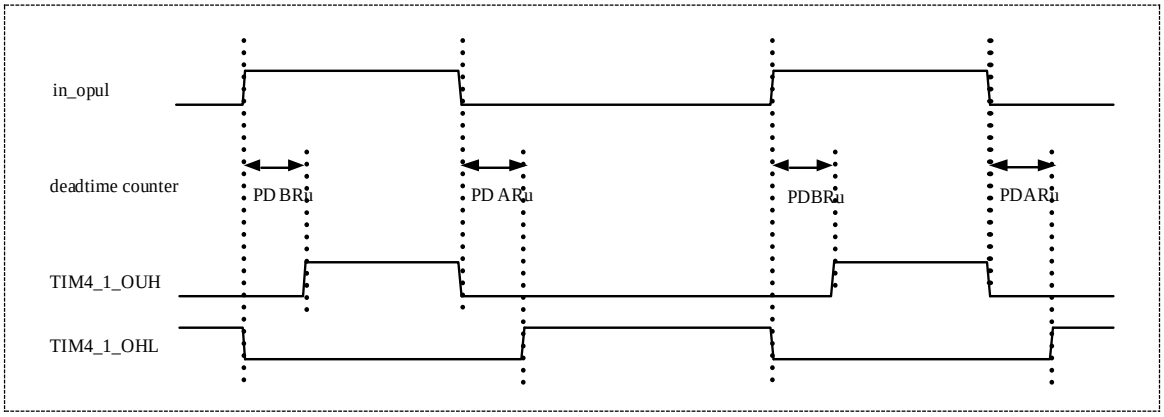


图 22-20 死区定时器模式下的互补 PWM 输出

in_opxl 的高电平脉冲宽度小于 PDBR 设定的死区时间时，仅 TIM4_<t>_OXL 输出变为低电平。TIM4_<t>_OXL 输出电平变为高的条件是当 in_opxl 下降沿之后经过 PDAR 寄存器设定的死区时间之后。在此情况下，TIM4_<t>_OXH 输出将持续保持为低电平。

in_opxl 的低电平脉冲宽度小于 PDAR 设定的死区时间时，仅 TIM4_<t>_OXH 输出变为低电平。TIM4_<t>_OXH 输出电平变为高的条件是当 in_opxl 上升沿之后经过 PDBR 寄存器设定的死区时间之后。在此情况下，TIM4_<t>_OXL 输出将持续保持为低电平（X=U、V、W，x=u、v、w）。如下图 22-21 所示。

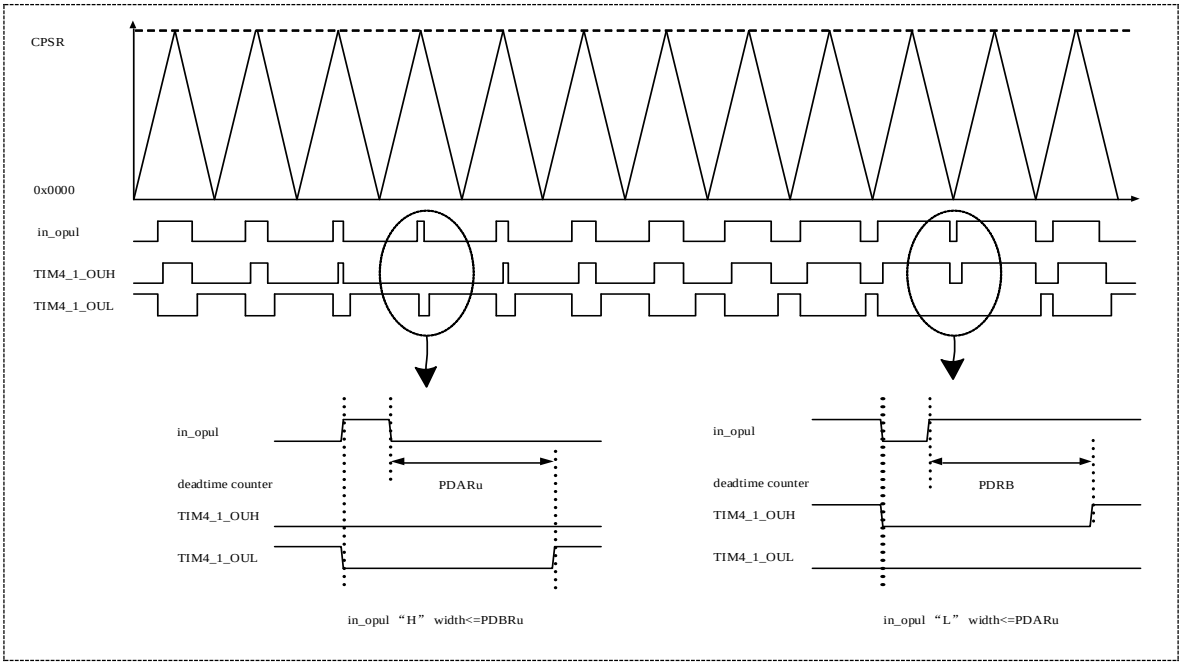


图 22-21 脉宽异常时的死区定时器模式下的波形输出

在上述硬件实现死区输出方式的基础上，还可以对内部的比较匹配信号（in_opxl）进行脉冲宽度的监测，从而实现对 in_opxl 信号的滤波控制。这种 in_opxl 带脉宽滤波的死区输出实现方式叫死区计数器滤波模式（POCR.PWMMD=10）（x=u、v、w）。

在死区计数器滤波模式下，滤波宽度由 PWM 滤波控制寄存器（PFSRn）的设定值决定。当 in_opxl 的脉冲宽度大于寄存器 PFSRn 设定的时间时，滤波计数器将 in_opxl 信号推迟 PFSR 设定的时间后输出，再经过死区定时器模式所描述的方式产生互补 PWM 输出（x=u、v、w）。

如果检测到信号 in_opxl 上升沿，滤波计数器加载 PFSR 寄存器值并启动计量 in_opxl 的高电平宽度，当 in_opxl 的高电平脉冲宽度大于寄存器 PFSR 设定的时间时，经过 PFSR 所设定时间之后，TIM4_<t>_OXL 输出变为低电平，死区计数器加载 PDBR 寄存器的设定值并启动递减计数，当计数值变为 0x0000 时，计数器停止并使 TIM4_<t>_OXH 输出高电平；如果检测到信号 in_opxl 下降沿，滤波计数器加载 PFSR 寄存器值并启动计量 in_opxl 的低电平宽度，当 in_opxl 的低电平脉冲宽度大于寄存器 PFSR 设定的时间时，经过 PFSR 所设定时间之后，TIM4_<t>_OXH 输出变为低电平，死区计数器加载 PDAR 寄存器的设定值并启动递减计数，当计数值变为 0x0000 时，计数器停止并使 TIM4_<t>_OXL 输出高电平。当 in_opxl 的电平脉冲宽度小于寄存器 PFSR 设定的时间时，输出 TIM4_<t>_OXH 和 TIM4_<t>_OXL 将保持不变（X=U、V、

W，x=u、v、w）。

图 22-22 所示，是死区计数器滤波模式下的互补 PWM 输出例。

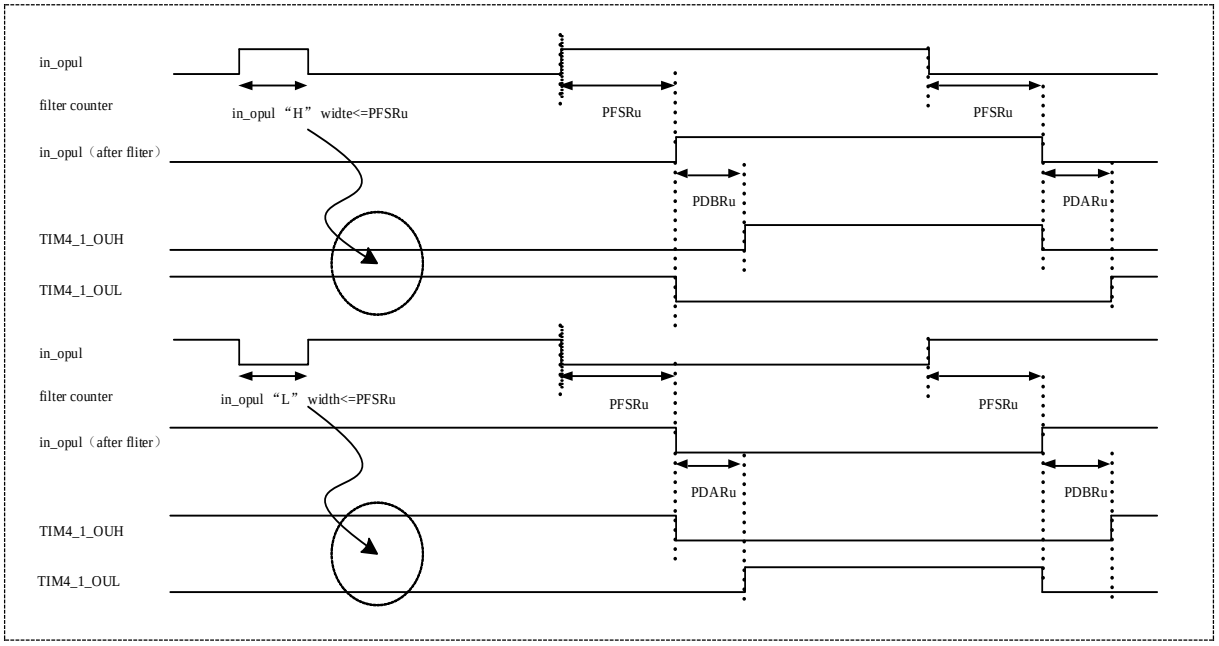


图 22-22 死区定时器滤波模式下的互补 PWM 输出

注意：

– 使用软件或硬件互补 PWM 输出模式前，需要设定 MOE、OExy 位以使能 Timer4 的输出，详细描述请参考 PWM 状态控制寄存器（PSCR）。

22.3.6 周期间隔响应

下溢中断屏蔽计数器用于降低（掩盖）下溢标志位（CCSR.IRQZF）的被置位次数。下溢中断屏蔽计数器（CVPR.ZIC[3:0]）作为递减计数器操作，在开始时载入 CVPR.ZIM[3:0]所设定的值，当 CVPR.ZIC[3:0]= “0”时，下溢标志位(CCSR.IRQZF)被置位为“1”。

上溢中断屏蔽计数器用于降低（掩盖）上溢标志位（CCSR.IRQPF）的被置位次数。上溢中断屏蔽计数器（CVPR.PIC[3:0]）作为递减计数器操作，在开始时载入 CVPR.PIM[3:0]所设定的值，当 CVPR.PIC[3:0]= “0”时，上溢标志位(CCSR.IRQPF)被置位为“1”。

如下图 22-23 所示，是周期间隔响应时的 IRQZF、IRQPF 的置位时序图。

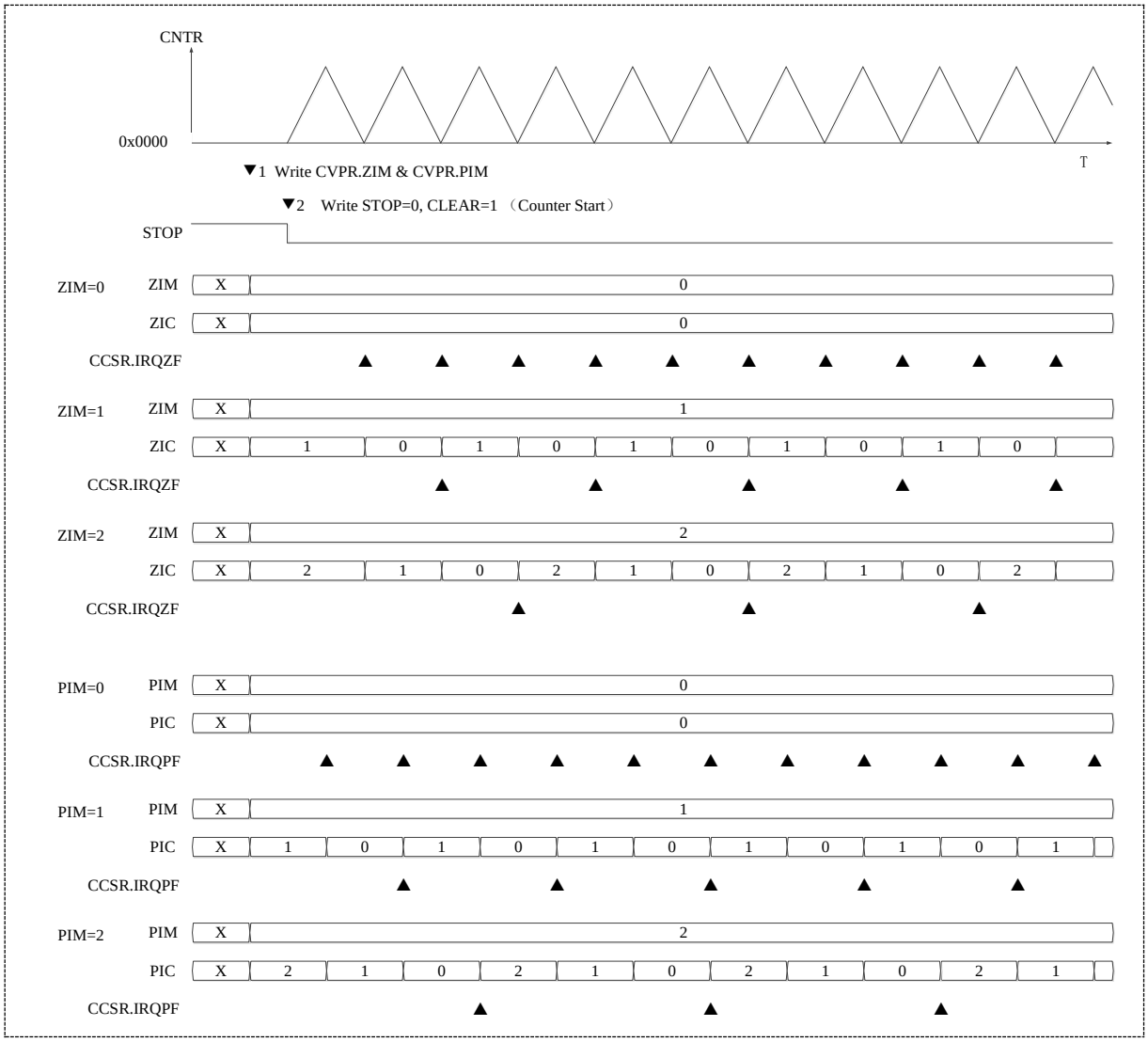


图 22-23 周期间隔响应时序图

▼1 计数器停止时，写入 CVPR.ZIM 和 CVPR.PIM 初始化值，初始值立即反映在内部计数器(CVPR.ZIC、CVPR.PIC)。

▼2 初始化并启动计数器(STOP=0 且 CLEAR=1)，计数器在总线复位或者从软件初始化 CLEAR=1 后从零开始进行递增计数， 此刻 CCSR.IRQZF 标志不会立即置位, 之后每当中断屏蔽计数器的计数值为 0x0000 且计数器的计数值为 0x0000 和 CPSR 时，标志▲为 CCSR.IRQZF 或 CCSR.IRQPF 置位的时刻。

注意：

– 在计数器运行中，写入 CVPR.ZIM 和 CVPR.PIM，该设置值不会立即反应到中断屏蔽计数器(CVPR.ZIC 和 CVPR.PIC)中。如果写入软复位(CLEAR=1)，写入的 CVPR.ZIM 和 CVPR.PIM 值将立即加载为中断屏蔽计数器的初值。

专用比较基准寄存器（SCCRm）的比较匹配事件（专用事件输出）也具有周期间隔响应功能。如下图 22-24 所示，是专用事件输出信号的周期间隔响应输出图。

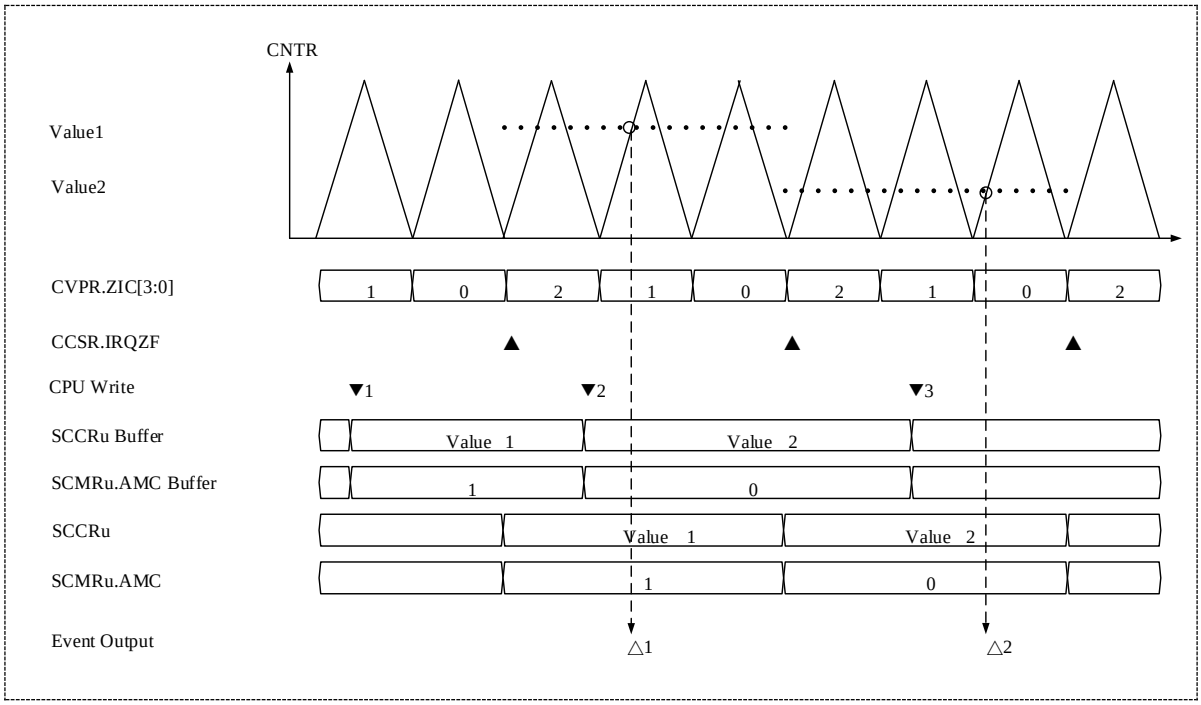


图 22-24 专用事件输出信号周期间隔响应输出

计数器是三角波计数模式，下溢中断屏蔽计数器(CVPR.ZIC)从 2-0 做递减计数。下溢中断在时刻▲产生。

在时刻▼1，向 SCCRu 缓冲寄存器写入 value 1。同时，MZCE=1、MPCE=0、AMC=0001 写入 SCMRu 缓冲寄存器。之后，缓冲寄存器 SCCRu 和 SCMRu 传送操作在▲时刻执行。因为 MZCE=1 且 AMC=1 且计数值=SCCRu=Value 1，在时刻△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 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。

22.3.7 EMB 控制

每个 Timer4 单元都有一个输出无效事件接口，连接 EMB 模块输出的 EMB 事件。该接口上选通的异常状况事件可从 EMB 侧设定（参见 EMB 章节）。

各单元内 3 组 PWM 端口在正常输出期间，若监测到从 EMB 过来的 EMB 异常事件，PSCR.MOE 位由硬件异步清零，各 PWM 端口的输出状态变为预先设定好的状态。根据 PSCR.OSxy 的设定，该预设端口状态可以为输出高阻态、输出低电平或输出高电平

（x=U、V、W，y=H、L）。

例如，若设定 PSCR.OSUH=01 时，在 Timer4 的 TIM4_<t>_OUH 端口正常输出期间产生 EMB 事件，则 TIM4_<t>_OUH 端口上输出变为高阻态。

EMB 异常事件清除后，若此时 PSCR.AOE=1 则 PSCR.MOE 将由硬件自动置位，随后 PWM 端口恢复到正常输出；若此时 PSCR.AOE=0 则需要软件置位 PSCR.MOE，以恢复 PWM 端口到正常输出。

需要注意，硬件上控制 PWM 输出使能使用的是 MOE 的异步信号，当 MOE 的值发生变化后，需要等待几个时钟周期才能在总线上读取到实际的值。关于 EMB 的详细操作方法，请参考 EMB 章节的描述。

22.3.8 监测输出

每个 Timer4 单元都有一个 PWM 周期输出监测端口 TIM4_<t>_PCT 和专用事件输出监测端口 TIM4_<t>_ADSM。这些监测端口输出到外部，用户可根据端口的变化信息灵活对应用系统进行控制。

PWM 周期输出监测端口 TIM4_<t>_PCT 用来监测当前计数器的计数状态。在三角波模式时，每次计数到波峰或波谷时，检测端口 TIM4_<t>_PCT 上就发生一次翻转。图 22-25 是计数方向信号的示例。

TIM4_<t>_PCT 端口的周期输出监测功能只在计数时钟不分频（CCSR.CKDIV=4’h0）的情况下使用。

专用事件输出监测端口 TIM4_<t>_ADSM 可以用来监测内部计数器的专用比较事件。通过专用扩展控制寄存器（SCER）的 SCER.EVTRS 位选择需要监测的专用比较输出通道，在该通道上的比较基准值和计数器产生比较匹配时，监测端口 TIM4_<t>_ADSM 上就会置位（输出为高）；在该通道上的计数器计数归零时（锯齿波溢出和三角波计数到波谷），监测端口 TIM4_<t>_ADSM 上就会复位（输出为低）。

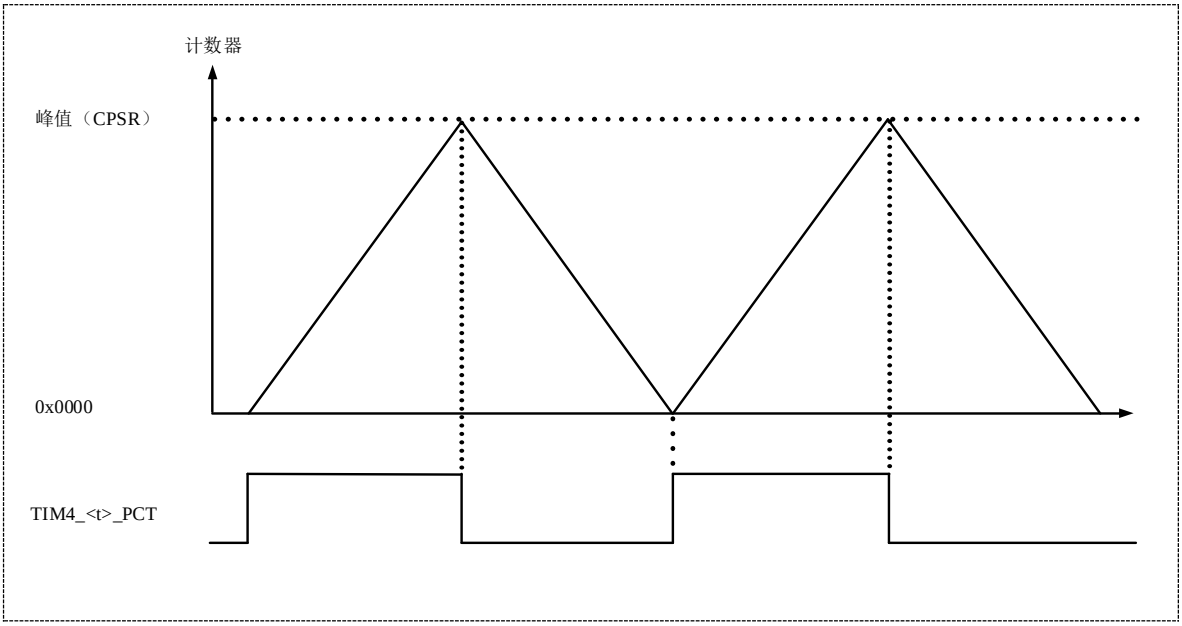


图 22-25　计数方向信号输出示例

22.4 中断及事件说明

22.4.1 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_<t>_GCMmn，m=U、V、W；n=H、L）也会被触发。

22.4.2 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，控制状态寄存器（CCSR）的 CCSR.IRQPF 或 CCSR.IRQZF 位会被置为 1。此时若设置 CCSR.IRQPEN 或 CCSR.IRQZEN 位使能中断，则在对应的时间点可触发计数周期匹配中断（TMR4_<t>_GOVF 和 TMR4_<t>_GUDF）。

22.4.3 重载计数匹配中断

在重载功能有效时，PWM 滤波控制寄存器（PFSRn）与计数值比较产生比较匹配信号。计数比较匹配时，重载控制状态寄存器（RCSR）中的 RCSR.RTIFU、RCSR.RTIFV、RCSR.RTIFW 位分别会被置为 1。此时若设定 RCSR.RTIDU、RCSR.RTIDV、RCSR.RTIDW 中断屏蔽无效，则对应的重载计数匹配中断请求（TMR4_<t>_RLOm，m=U、V、W）也会被触发。

22.4.4 专用比较匹配事件

Timer4 的 6 个专用比较基准寄存器（SCCRm）对应产生 6 个专用事件输出信号可以用于选择触发别的模块，如启动 ADC 等。

时钟计数过程中，若专用比较基准值（SCCRm）发生计数比较匹配事件（TMR4_<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 22-26 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。

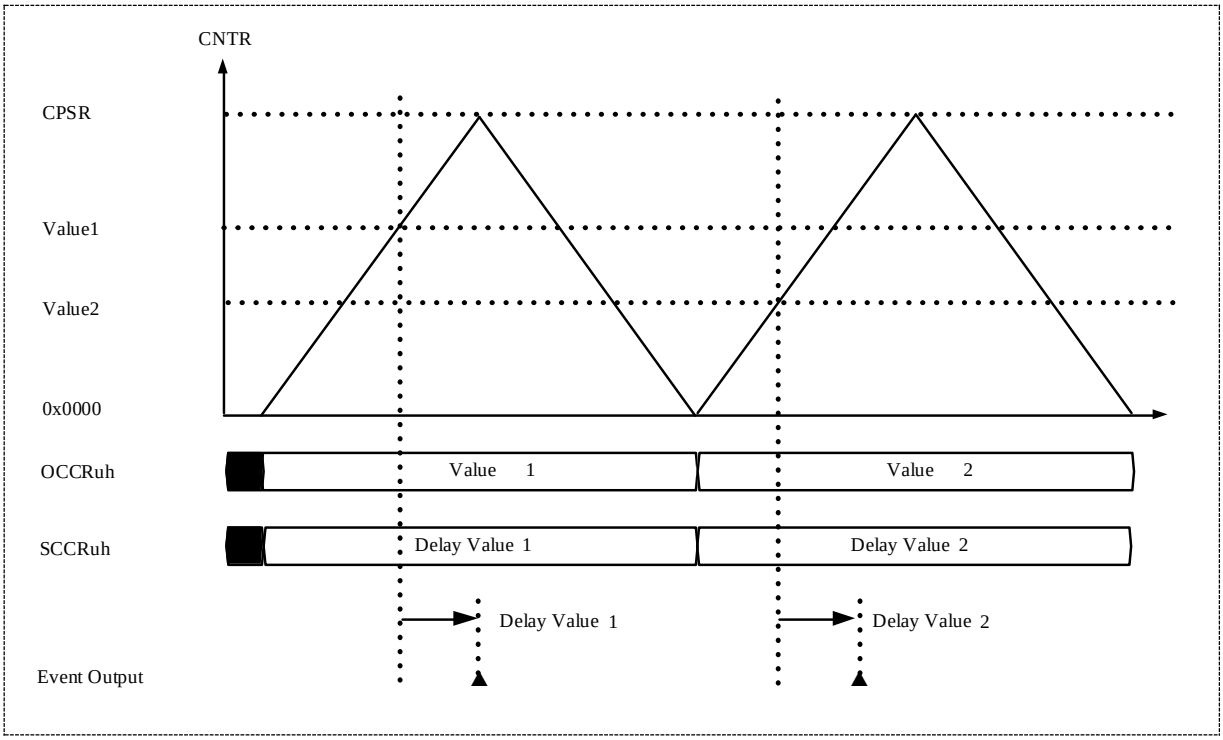


图 22-26 延时启动模式下专用事件输出信号的输出时序

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。

22.5 寄存器说明

表 22-3 所示，为 Timer4 模块的寄存器列表。

BASE ADDR: 0x40038000（U1）、0x40038400（U2）、0x40038800（U3）

寄存器名	符号	偏移量	位宽	复位值
计数值寄存器	TMR4_CNTR	0x0046h	16	0x0000h
周期基准寄存器	TMR4_CPSR	0x0042h	16	0xFFFFh
控制状态寄存器	TMR4_CCSR	0x0048h	16	0x0040h
有效周期寄存器	TMR4_CVPR	0x004Ah	16	0x0000h
通用比较基准寄存器UH	TMR4_OCCRuh	0x0002h	16	0x0000h
通用比较基准寄存器UL	TMR4_OCCRul	0x0006h	16	0x0000h
通用比较基准寄存器VH	TMR4_OCCRvh	0x000Ah	16	0x0000h
通用比较基准寄存器VL	TMR4_OCCRvl	0x000Eh	16	0x0000h
通用比较基准寄存器WH	TMR4_OCCRwh	0x0012h	16	0x0000h
通用比较基准寄存器WL	TMR4_OCCRwl	0x0016h	16	0x0000h
通用控制状态寄存器U	TMR4_OCSRu	0x0018h	16	0xFF00h
通用控制状态寄存器V	TMR4_OCSRv	0x001Ch	16	0xFF00h
通用控制状态寄存器W	TMR4_OCSRw	0x0020h	16	0xFF00h
通用扩展控制寄存器U	TMR4_OCERu	0x001Ah	16	0x0000h
通用扩展控制寄存器V	TMR4_OCERv	0x001Eh	16	0x0000h
通用扩展控制寄存器W	TMR4_OCERw	0x0022h	16	0x0000h
通用模式控制寄存器UH	TMR4_OCMRuh	0x0024h	16	0x0000h
通用模式控制寄存器UL	TMR4_OCMRul	0x0028h	32	0x00000000h
通用模式控制寄存器VH	TMR4_OCMRvh	0x002Ch	16	0x0000h
通用模式控制寄存器VL	TMR4_OCMRvl	0x0030h	32	0x00000000h
通用模式控制寄存器WH	TMR4_OCMRwh	0x0034h	16	0x0000h
通用模式控制寄存器WL	TMR4_OCMRwl	0x0038h	32	0x00000000h
专用比较基准寄存器UH	TMR4_SCCRuh	0x00B0h	16	0x0000h

专用比较基准寄存器UL	TMR4_SCCRul	0x00B4h	16	0x0000h
专用比较基准寄存器VH	TMR4_SCCRvh	0x00B8h	16	0x0000h
专用比较基准寄存器VL	TMR4_SCCRvl	0x00BCh	16	0x0000h
专用比较基准寄存器WH	TMR4_SCCRwh	0x00C0h	16	0x0000h
专用比较基准寄存器WL	TMR4_SCCRwl	0x00C4h	16	0x0000h
专用控制状态寄存器UH	TMR4_SCSRuh	0x00C8h	16	0x0000h
专用控制状态寄存器UL	TMR4_SCSRul	0x00CCh	16	0x0000h
专用控制状态寄存器VH	TMR4_SCSRvh	0x00D0h	16	0x0000h
专用控制状态寄存器VL	TMR4_SCSRvl	0x00D4h	16	0x0000h
专用控制状态寄存器WH	TMR4_SCSRwh	0x00D8h	16	0x0000h
专用控制状态寄存器WL	TMR4_SCSRwl	0x00DCh	16	0x0000h
专用扩展控制寄存器	TMR4_SCER	0x00E4h	16	0xFF00h
专用模式控制寄存器UH	TMR4_SCMRuh	0x00CAh	16	0xFF00h
专用模式控制寄存器UL	TMR4_SCMRul	0x00CEh	16	0xFF00h
专用模式控制寄存器VH	TMR4_SCMRvh	0x00D2h	16	0xFF00h
专用模式控制寄存器VL	TMR4_SCMRvl	0x00D6h	16	0xFF00h
专用模式控制寄存器WH	TMR4_SCMRwh	0x00DAh	16	0xFF00h
专用模式控制寄存器WL	TMR4_SCMRwl	0x00DEh	16	0xFF00h
PWM基本控制寄存器U	TMR4_POCRu	0x0098h	16	0xFF00h
PWM基本控制寄存器V	TMR4_POCRv	0x009Ch	16	0xFF00h
PWM基本控制寄存器W	TMR4_POCRw	0x00A0h	16	0xFF00h
PWM状态控制寄存器	TMR4_PSCR	0x00E0h	32	0x05550000h
PWM滤波控制寄存器U	TMR4_PFSRu	0x0082h	16	0x0000h
PWM滤波控制寄存器V	TMR4_PFSRv	0x008Ah	16	0x0000h
PWM滤波控制寄存器W	TMR4_PFSRw	0x0092h	16	0x0000h
PWM死区控制寄存器AU	TMR4_PDARu	0x0084h	16	0x0000h
PWM死区控制寄存器BU	TMR4_PDBRu	0x0086h	16	0x0000h
PWM死区控制寄存器AV	TMR4_PDARv	0x008Ch	16	0x0000h

PWM死区控制寄存器BV	TMR4_PDBRv	0x008Eh	16	0x0000h
PWM死区控制寄存器AW	TMR4_PDARw	0x0094h	16	0x0000h
PWM死区控制寄存器BW	TMR4_PDBRw	0x0096h	16	0x0000h
重载控制状态寄存器	TMR4_RCSR	0x00A4h	16	0x0000h
EMB控制状态寄存器	TMR4_ECSR	（0x40006C18h）	32	0x00000000h

表 22-3 Timer4 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。

22.5.1 计数值寄存器（TMR4_CNTR）

复位值: 0x0000h

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

22.5.2 周期基准寄存器（TMR4_CPSR）

复位值: 0xFFFFh

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

22.5.3 控制状态寄存器（TMR4_CCSR）

复位值: 0x0040h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ECKEN	IRQZF	IRQZEN	-	-	-	IRQPF	IRQPEN	BUFEN	STOP	MODE	CLEAR	CKDIV[3:0]			

位	标记	位名	功能	读写
b15	EXKEN	时钟源选择	0：内部PCLK0时钟 1：外部TIM4_<t>_CLK端口输入时钟 注1：该位在计数器停止时设定 注2：使用外部输入时钟进行操作的场合，计数器操作开始后，最初的外部输入时钟的边沿，无论是上升沿还是下降沿都被忽略，计数操作从之后的边沿开始	R/W
b14	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	IRQPEN	上溢中断使能	0：禁止IRQPF产生中断到CPU 1：允许IRQPF产生中断到CPU	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PCLK0 0001：计数时钟为PCLK0/2 0010：计数时钟为PCLK0/4 0011：计数时钟为PCLK0/8 0100：计数时钟为PCLK0/16 0101：计数时钟为PCLK0/32 0110：计数时钟为PCLK0/64 0111：计数时钟为PCLK0/128 1000：计数时钟为PCLK0/256 1001：计数时钟为PCLK0/512 1010：计数时钟为PCLK0/1024 请不要设定其它值

22.5.4 有效周期寄存器（TMR4_CVPR）

复位值: 0x0000h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIC[3:0]				ZIC[3:0]				PIM[3:0]				ZIM[3:0]			

位	标记	位名	功能	读写
b15~b12	PIC[3:0]	上溢中断屏蔽状态	当前被屏蔽的上溢中断个数	R
b11~b8	ZIC[3:0]	下溢中断屏蔽状态	当前被屏蔽的下溢中断个数	R
b7~b4	PIM[3:0]	上溢中断屏蔽设定	设置屏蔽的上溢中断个数	R/W
b3~b0	ZIM[3:0]	上溢中断屏蔽设定	设置屏蔽的下溢中断个数	R/W

22.5.5 通用比较基准寄存器（TMR4_OCCRm）

复位值: 0x0000h

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

22.5.6 通用控制状态寄存器（TMR4_OCSRn）

复位值: 0xFF00h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								OCFL	OCFH	OCIEL	OCIEH	OCPL	OCPH	OCEL	OCEH

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	OCPL	比较输出无效时端口状态L	0：OCEL=0时，TIM4_<t>_OxL上输出低电平 1：OCEL=0时，TIM4_<t>_OxL上输出高电平 （m=U、V、W）	R/W
b2	OCPH	比较输出无效时端口状态H	0：OCEH=0时，TIM4_<t>_OxH上输出低电平 1：OCEH=0时，TIM4_<t>_OxH上输出高电平 （m=U、V、W）	R/W
b1	OCEL	端口输出选择L	0：比较输出无效，TIM4_<t>_OxL端口状态由OCPL决定 1：比较输出有效，TIM4_<t>_OxL端口状态由OCMRyl的设定和OCFL状态决定 （x=U、V、W，y=u、v、w）	R/W
b0	OCEH	端口输出选择H	0：比较输出无效，TIM4_<t>_OxH端口状态由OCPH决定 1：比较输出有效，TIM4_<t>_OxH端口状态由OCMRyh的设定和OCFH状态决定 （x=U、V、W，y=u、v、w）	R/W

22.5.7 通用扩展控制寄存器（TMR4_OCERn）

复位值: 0x0000h

b15	b14	b13	b12	b11	b10	b9	b8	b7 b6	b5 b4	b3 b2	b1 b0
-	-	MCECL	MCECH	LMML	LMMH	LMCL	LMCH	MLBUFEN[1:0]	MHBUFEN[1:0]	CLBUFEN[1:0]	CHBUFEN[1:0]

位	标记	位名	功能	读写
b15~b14	Reserved	-	读出时为“0”,写入时写“0”	R/W
b13	MCECL	扩展控制使能L	0：禁止计数值与OCCRxl匹配条件的扩展控制功能 1：使能计数值与OCCRxl匹配条件的扩展控制功能 （x=u、v、w） 注1：只能在输出比较功能无效（OCSR.OCEL=0）时修改该位 注2：具体请参考TMR4_OCMRm寄存器说明的注意事项	R/W
b12	MCECH	扩展控制使能H	0：禁止计数值与OCCRxh匹配条件的扩展控制功能 1：使能计数值与OCCRxh匹配条件的扩展控制功能 （x=u、v、w） 注1：只能在输出比较功能无效（OCSR.OCEH=0）时修改该位 注2：具体请参考TMR4_OCMRm寄存器说明的注意事项	R/W
b11	LMML	周期间隔响应功能链接L	0：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MLBUFEN设定决定 1：周期间隔响应功能链接有效，OCMRxh的缓存传送在MLBUFEN设定的基础上，还必须满足CVPR.PIC[3:0]=0000（计数上溢时）或CVPR.ZIC[3:0]=0000（计数下溢时） （x=u、v、w）	R/W
b9	LMCL	周期间隔响应功能链接L	0：周期间隔响应功能链接无效，OCCRxl的缓存传送由MLBUFEN设定决定 1：周期间隔响应功能链接有效，OCCRxl的缓存传送在MLBUFEN设定的基础上，还必须满足CVPR.PIC[3:0]=0000（计数上溢时）或CVPR.ZIC[3:0]=0000（计数下溢时） （x=u、v、w）	R/W
b8	LMCH	周期间隔响应功能链接H	0：周期间隔响应功能链接无效，OCCRxh的缓存传送由MLBUFEN设定决定 1：周期间隔响应功能链接有效，OCCRxh的缓存传送在MLBUFEN设定的基础上，还必须满足CVPR.PIC[3:0]=0000（计	R/W

			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

22.5.8 通用模式控制寄存器（TMR4_OCMRm）

复位值: 0x0000h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OPN ZRH[1:0]		OPN PKH[1:0]		OP ZRH[1:0]		OP UCH[1:0]		OP PKH[1:0]		OP DCH[1:0]		OCF ZRH	OCF UCH	OCF PKH	OCF DCH

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_<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）	
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: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EOPN ZRL[1:0]		EOPN PKL[1:0]		EOP ZRL[1:0]		EOP UCL[1:0]		EOP PKL[1:0]		EOP DCL[1:0]		EOPN UCL[1:0]		EOPN DCL[1:0]	

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OPN ZRL[1:0]		OPN PKL[1:0]		OP ZRL[1:0]		OP UCL[1:0]		OP PKL[1:0]		OP DCL[1:0]		OCF ZRL	OCF UCL	OCF PKL	OCF DCL

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_<t>_OyL端口位保持不变 01：满足条件时，TIM4_<t>_OyL端口位输出高电平 10：满足条件时，TIM4_<t>_OyL端口位输出低电平 11：满足条件时，TIM4_<t>_OyL端口位输出反转 （y=U、V、W）	R/W
b29~b28	EOPNPKL[1:0]	扩张上溢点端口状态L	条件：计数上溢 & OCCRxl计数不匹配 & OCCRxh计数匹配（x=u、v、w） 00：满足条件时，TIM4_<t>_OyL端口位保持不变 01：满足条件时，TIM4_<t>_OyL端口位输出高电平 10：满足条件时，TIM4_<t>_OyL端口位输出低电平 11：满足条件时，TIM4_<t>_OyL端口位输出反转 （y=U、V、W）	R/W
b27~b26	EOPZRL[1:0]	扩张下溢点端口	条件：计数下溢 & OCCRxl计数匹配 & OCCRxh计数匹配（x=u、	R/W

		状态L	v、w） 00：满足条件时，TIM4_<t>_OyL端口位保持不变 01：满足条件时，TIM4_<t>_OyL端口位输出高电平 10：满足条件时，TIM4_<t>_OyL端口位输出低电平 11：满足条件时，TIM4_<t>_OyL端口位输出反转 （y=U、V、W）	
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 & OCCRxl计数匹配 & OCCRxh计数匹配 （x=u、v、w） 00：满足条件时，TIM4_<t>_OyL端口位保持不变 01：满足条件时，TIM4_<t>_OyL端口位输出高电平 10：满足条件时，TIM4_<t>_OyL端口位输出低电平 11：满足条件时，TIM4_<t>_OyL端口位输出反转 （y=U、V、W）	R/W
b23~b22	EOPPKL[1:0]	扩张上溢点端口状态L	条件：计数上溢 & OCCRxl计数匹配 & OCCRxh计数匹配（x=u、v、w） 00：满足条件时，TIM4_<t>_OyL端口位保持不变 01：满足条件时，TIM4_<t>_OyL端口位输出高电平 10：满足条件时，TIM4_<t>_OyL端口位输出低电平 11：满足条件时，TIM4_<t>_OyL端口位输出反转 （y=U、V、W）	R/W
b21~b20	EOPDCL[1:0]	扩张向下计数端口状态L	条件：计数器向下计数 & OCCRxl计数匹配 & OCCRxh计数匹配 （x=u、v、w） 00：满足条件时，TIM4_<t>_OyL端口位保持不变 01：满足条件时，TIM4_<t>_OyL端口位输出高电平 10：满足条件时，TIM4_<t>_OyL端口位输出低电平 11：满足条件时，TIM4_<t>_OyL端口位输出反转 （y=U、V、W）	R/W
b19~b18	EOPNUCL[1:0]	扩张向上计数端口状态L	条件：计数器向上计数 & OCCRxl计数不匹配 & OCCRxh计数匹配 （x=u、v、w） 00：满足条件时，TIM4_<t>_OyL端口位保持不变 01：满足条件时，TIM4_<t>_OyL端口位输出高电平 10：满足条件时，TIM4_<t>_OyL端口位输出低电平 11：满足条件时，TIM4_<t>_OyL端口位输出反转 （y=U、V、W）	R/W
b17~b16	EOPNDCL[1:0]	扩张向下计数端口状态L	条件：计数器向下计数 & OCCRxl计数不匹配 & OCCRxh计数匹配 （x=u、v、w） 00：满足条件时，TIM4_<t>_OyL端口位保持不变 01：满足条件时，TIM4_<t>_OyL端口位输出高电平 10：满足条件时，TIM4_<t>_OyL端口位输出低电平 11：满足条件时，TIM4_<t>_OyL端口位输出反转 （y=U、V、W）	R/W
b15~b14	OPNZRL[1:0]	下溢点端口状态	条件：计数下溢 & OCCRxl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_<t>_OyL端口位保持不变 01：满足条件时，TIM4_<t>_OyL端口位输出高电平 10：满足条件时，TIM4_<t>_OyL端口位输出低电平 11：满足条件时，TIM4_<t>_OyL端口位输出反转 （y=U、V、W）	
b13~b12	OPNPKL[1:0]	上溢点端口状态L	条件：计数上溢 & OCCRxl计数不匹配 & OCCRxh计数不匹配 （x=u、v、w） 00：满足条件时，TIM4_<t>_OyL端口位保持不变 01：满足条件时，TIM4_<t>_OyL端口位输出高电平 10：满足条件时，TIM4_<t>_OyL端口位输出低电平 11：满足条件时，TIM4_<t>_OyL端口位输出反转 （y=U、V、W）	R/W
b11~b10	OPZRL[1:0]	下溢点端口状态L	条件：计数下溢 & OCCRxl计数匹配 & OCCRxh计数不匹配 （x=u、v、w） 00：满足条件时，TIM4_<t>_OyL端口位保持不变 01：满足条件时，TIM4_<t>_OyL端口位输出高电平 10：满足条件时，TIM4_<t>_OyL端口位输出低电平 11：满足条件时，TIM4_<t>_OyL端口位输出反转 （y=U、V、W）	R/W
b9~b8	OPUCL[1:0]	向上计数端口状态L	条件：计数器向上计数 & OCCRxl计数匹配 & OCCRxh计数不匹配 （x=u、v、w） 00：满足条件时，TIM4_<t>_OyL端口位保持不变 01：满足条件时，TIM4_<t>_OyL端口位输出高电平 10：满足条件时，TIM4_<t>_OyL端口位输出低电平 11：满足条件时，TIM4_<t>_OyL端口位输出反转 （y=U、V、W）	R/W
b7~b6	OPPKL[1:0]	上溢点端口状态L	条件：计数上溢 & OCCRxl计数匹配 & OCCRxh计数不匹配 （x=u、v、w） 00：满足条件时，TIM4_<t>_OyL端口位保持不变 01：满足条件时，TIM4_<t>_OyL端口位输出高电平 10：满足条件时，TIM4_<t>_OyL端口位输出低电平 11：满足条件时，TIM4_<t>_OyL端口位输出反转 （y=U、V、W）	R/W
b5~b4	OPDCL[1:0]	向下计数端口状态L	条件：计数器向下计数 & OCCRxl计数匹配 & OCCRxh计数不匹配 （x=u、v、w） 00：满足条件时，TIM4_<t>_OyL端口位保持不变 01：满足条件时，TIM4_<t>_OyL端口位输出高电平 10：满足条件时，TIM4_<t>_OyL端口位输出低电平 11：满足条件时，TIM4_<t>_OyL端口位输出反转 （y=U、V、W）	R/W
b3	OCFZRL	下溢点OCFL状	条件：计数下溢 & OCCRxl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状态L	条件：计数器向上计数 & OCCRxl计数匹配（x=u、v、w） 0：满足条件时，OCSR.OCFL位保持不变 1：满足条件时，OCSR.OCFL位置位	R/W
b1	OCFPKL	上溢点OCFL状态L	条件：计数上溢 & OCCRxl计数匹配（x=u、v、w） 0：满足条件时，OCSR.OCFL位保持不变 1：满足条件时，OCSR.OCFL位置位	R/W
b0	OCFDCL	向下计数OCFL状态L	条件：计数器向下计数 & OCCRxl计数匹配（x=u、v、w） 0：满足条件时，OCSR.OCFL位保持不变 1：满足条件时，OCSR.OCFL位置位	R/W

注意：

- 从本地址区域读取数据时，读取的并非缓冲器寄存器的值，而是 OCMR 寄存器的值。
- TIM4_<t>_OyL 可以由 OCCRxl 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 OCCRxl 与计数器的计数值（链接操作模式）来决定。向寄存器 OCMRxl 的 bit[31:20]和 bit[15:4]写入相同的 12 位值，同时将 OCMRxl[19:16]写入“0000”，此时，TIM4_<t>_OyL 输出将不受 OCCRxh 的影响，而仅由 OCCRxl 决定。此模式叫做独立操作模式----通道 yH 由 OCCRxh 决定，通道 yL 由 OCCRxl 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 OCCRxl 影响。

  （x=u、v、w，y=U、V、W）
- 下面 2 种情况满足时，也认为是符合 OCMRxh. OPPKH[7:6]设定的条件（x=u、v、w）：

  1） CCSR.MODE=1 && 计数上溢 && OCCRxh=0xFFFF

  2） OCERx.MCECH=1 && 计数上溢 && OCCRxh≥CNTR
- 下面 2 种情况满足时，也认为是符合 OCMRxl. OPPKL[7:6]设定的条件（x=u、v、w）：

  1） CCSR.MODE=1 && 计数上溢 && OCCRxl=0xFFFF

  2） OCERx.MCECL=1 && 计数上溢 && OCCRxl≥CNTR

22.5.9 专用比较基准寄存器（TMR4_SCCRm）

复位值: 0x0000h

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

22.5.10 专用控制状态寄存器（TMR4_SCSRm）

复位值: 0x0000h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ZEN	UEN	PEN	DEN	-	-	EVTDS	EVTMS	-	-	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，OCCRxl作为延时比较匹配对象 （x=u、v、w） 注：该位在EVTMS=0时无效	R/W
b8	EVTMS	EVT模式选择	0：比较启动模式（CNTR和SCCR的比较结果触发） 1：延时启动模式（比较匹配事件经SCCR延迟后触发）	R/W
b7~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5	LMC	周期间隔响应功能链接	0：周期间隔响应功能链接无效，SCCR的缓存传送由BUFEN设定决定 1：周期间隔响应功能链接有效，SCCR的缓存传送在BUFEN设定的基础上，还必须满足CVPR.PIC[3:0]=0000（计数上溢时）或CVPR.ZIC[3:0]=0000（计数下溢时）	R/W

b4~b2	EVTOS[2:0]	EVT输出选择	000：Special Event 0的EVT输出有效 001：Special 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缓存传送	00：SCCR、SCMR缓存寄存器的值直接写入SCCR、SCMR 01：SCCR、SCMR缓存寄存器的值在计数下溢时写入SCCR、SCMR 10：SCCR、SCMR缓存寄存器的值在计数上溢时写入SCCR、SCMR 11：SCCR、SCMR缓存寄存器的值在计数下溢或上溢时写入SCCR、SCMR	R/W

22.5.11 专用扩展控制寄存器（TMR4_SCER）

复位值: 0xFF00h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												PCTS	EVTRS[2:0]		

位	标记	位名	功能	读写
b15~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	PCTS	计数方向输出使能	计数器工作在三角波模式时，控制计数方向输出到端口 0：禁止计数方向信号输出到TIM4_<t>_PCT端口 1：允许计数方向信号输出到TIM4_<t>_PCT端口	R/W
b2~b0	EVTRS[2:0]	专用事件输出选择	000：禁止专用事件输出到端口 001：Special Event 0的EVT输出至TIM4_<t>_ADSM端口 010：Special Event1的EVT输出至TIM4_<t>_ADSM端口 011：Special Event 2的EVT输出至TIM4_<t>_ADSM端口 100：Special Event 3的EVT输出至TIM4_<t>_ADSM端口 101：Special Event 4的EVT输出至TIM4_<t>_ADSM端口 110：Special Event 5的EVT输出至TIM4_<t>_ADSM端口 111：设定禁止	R/W

22.5.12 专用模式控制寄存器（TMR4_SCMRm）

复位值: 0xFF00h

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。

22.5.13 PWM 基本控制寄存器（TMR4_POCRn）

复位值: 0xFF00h

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_<t>_OxH和TIM4_<t>_OxL的输出都不反转 01：TIM4_<t>_OxH和TIM4_<t>_OxL的输出都反转 10：TIM4_<t>_OxH的输出反转，TIM4_<t>_OxL的输出不反转 11：TIM4_<t>_OxH的输出不反转，TIM4_<t>_OxL的输出反转	R/W
b5~b4	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PCLK0 001：计数时钟为PCLK0/2 010：计数时钟为PCLK0/4 011：计数时钟为PCLK0/8 100：计数时钟为PCLK0/16 101：计数时钟为PCLK0/32 110：计数时钟为PCLK0/64 111：计数时钟为PCLK0/128	R/W

22.5.14 PWM 状态控制寄存器（TMR4_PSCR）

复位值: 0x05550000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved				OSWL[1:0]		OSWH[1:0]		OSVL[1:0]		OSVH[1:0]		OSUL[1:0]		OSUH[1:0]	

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved						AOE	MOE	ODT[1:0]		OEWL	OEWH	OEVL	OEVH	OEUL	OEUH

位	标记	位名	功能	读写
b31~b28	Reserved	-	读出时为“0”,写入时写“0”	R/W
b27~b26	OSWL[1:0]	TIM4_<t>_OWL输出禁止状态	条件：1.发生EMB事件；2.软件写MOE=0；3. MOE=1&OEWL=0 00：满足以上条件之一时，TIM4_<t>_OWL端口正常输出 01：满足以上条件之一时，TIM4_<t>_OWL端口输出为Hi-z 10：满足以上条件之一时，TIM4_<t>_OWL端口输出固定为低电平 11：满足以上条件之一时，TIM4_<t>_OWL端口输出固定为高电平 注：只能在Timer4停止（CCSR.STOP=1）时修改该位	R/W
b25~b24	OSWH[1:0]	TIM4_<t>_OWH输出禁止状态	条件：1.发生EMB事件；2.软件写MOE=0；3. MOE=1&OEWH=0 00：满足以上条件之一时，TIM4_<t>_OWH端口正常输出 01：满足以上条件之一时，TIM4_<t>_OWH端口输出为Hi-z 10：满足以上条件之一时，TIM4_<t>_OWH端口输出固定为低电平 11：满足以上条件之一时，TIM4_<t>_OWH端口输出固定为高电平 注：只能在Timer4停止（CCSR.STOP=1）时修改该位	R/W
b23~b22	OSVL[1:0]	TIM4_<t>_OVL输出禁止状态	条件：1.发生EMB事件；2.软件写MOE=0；3. MOE=1&OEVL=0 00：满足以上条件之一时，TIM4_<t>_OVL端口正常输出 01：满足以上条件之一时，TIM4_<t>_OVL端口输出为Hi-z 10：满足以上条件之一时，TIM4_<t>_OVL端口输出固定为低电平 11：满足以上条件之一时，TIM4_<t>_OVL端口输出固定为高电平 注：只能在Timer4停止（CCSR.STOP=1）时修改该位	R/W
b21~b20	OSVH[1:0]	TIM4_<t>_OVH输出禁止状态	条件：1.发生EMB事件；2.软件写MOE=0；3. MOE=1&OEVH=0 00：满足以上条件之一时，TIM4_<t>_OVH端口正常输出 01：满足以上条件之一时，TIM4_<t>_OVH端口输出为Hi-z 10：满足以上条件之一时，TIM4_<t>_OVH端口输出固定为低电平 11：满足以上条件之一时，TIM4_<t>_OVH端口输出固定为高电平 注：只能在Timer4停止（CCSR.STOP=1）时修改该位	R/W
b19~b18	OSUL[1:0]	TIM4_<t>_OUL输出禁止状态	条件：1.发生EMB事件；2.软件写MOE=0；3. MOE=1&OEUL=0 00：满足以上条件之一时，TIM4_<t>_OUL端口正常输出 01：满足以上条件之一时，TIM4_<t>_OUL端口输出为Hi-z 10：满足以上条件之一时，TIM4_<t>_OUL端口输出固定为低电平 11：满足以上条件之一时，TIM4_<t>_OUL端口输出固定为高电平 注：只能在Timer4停止（CCSR.STOP=1）时修改该位	R/W
b17~b16	OSUH[1:0]	TIM4_<t>_OUH输出	条件：1.发生EMB事件；2.软件写MOE=0；3. MOE=1&OEUH=0	R/W

		禁止状态	00：满足以上条件之一时，TIM4_<t>_OUH端口正常输出 01：满足以上条件之一时，TIM4_<t>_OUH端口输出为Hi-z 10：满足以上条件之一时，TIM4_<t>_OUH端口输出固定为低电平 11：满足以上条件之一时，TIM4_<t>_OUH端口输出固定为高电平 注：只能在Timer4停止（CCSR.STOP=1）时修改该位	
b15~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9	AOE	自动输出使能	0：EMB事件清除后MOE位只能软件置1以恢复PWM正常输出 1：EMB事件清除后MOE位硬件自动置1，PWM输出恢复正常 注：置位EMB_STATCLR或清零EMB_SOE可以清除EMB事件	R/W
b8	MOE	主输出使能	0：TIM4_<t>_Oxy端口输出由OSxy位指定的状态（x=U、V、W，y=H、L） 1：OExy位设定为使能时TIM4_<t>_Oxy端口输出使能（x=U、V、W，y=H、L） 注：EMB事件发生后此位由硬件立即清零。EMB事件清除后，依据AOE位的设置此位可以软件置1或硬件自动置1。	R/W
b7~b6	ODT	端口使能位生效时间	条件：端口使能位OExy的值改变后（x=U、V、W，y=H、L） 0X：满足条件时，立即生效 10：满足条件时，计数器下次下溢时生效 11：满足条件时，计数器下次上溢时生效	R/W
b5	OEWL	TIM4_<t>_OWL输出使能	0：TIM4_<t>_OWL端口输出状态由OSWL位设定 1：MOE=1时，TIM4_<t>_OWL端口输出正常PWM波形	R/W
b4	OEWH	TIM4_<t>_OWH输出使能	0：TIM4_<t>_OWH端口输出状态由OSWH位设定 1：MOE=1时，TIM4_<t>_OWH端口输出正常PWM波形	R/W
b3	OEVL	TIM4_<t>_OVL输出使能	0：TIM4_<t>_OVL端口输出状态由OSVL位设定 1：MOE=1时，TIM4_<t>_OVL端口输出正常PWM波形	R/W
b2	OEVH	TIM4_<t>_OVH输出使能	0：TIM4_<t>_OVH端口输出状态由OSVH位设定 1：MOE=1时，TIM4_<t>_OVH端口输出正常PWM波形	R/W
b1	OEUL	TIM4_<t>_OUL输出使能	0：TIM4_<t>_OUL端口输出状态由OSUL位设定 1：MOE=1时，TIM4_<t>_OUL端口输出正常PWM波形	R/W
b0	OEUH	TIM4_<t>_OUH输出使能	0：TIM4_<t>_OUH端口输出状态由OSUH位设定 1：MOE=1时，TIM4_<t>_OUH端口输出正常PWM波形	R/W

表 22-4 为 PWM 端口输出状态与寄存器设定值的关系说明。

MOE 位	OExy 位	OSxy 位	TIM4_<t>_Oxy 输出状态
1	0	00	正常输出（Timer4 驱动）
		01	Hi-z
		10	低电平
		11	高电平

	1	X	正常输出（Timer4 驱动）
0	X	00	正常输出（Timer4 驱动）
		01	Hi-z
		10	低电平
		11	高电平

表 22-4　PWM 端口输出状态与寄存器设定值

22.5.15 PWM 滤波控制寄存器（TMR4_PFSRn）

复位值: 0x0000h

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

22.5.16 PWM 死区控制寄存器（TMR4_PDA<B>Rn）

复位值: 0x0000h

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

22.5.17 重载控制状态寄存器（TMR4_RCSR）

复位值: 0x0000h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RTSW	RTEW	RTICW	RTIFW	RTSV	RTEV	RTICV	RTIFV	RTSU	RTEU	RTICU	RTIFU	-	RTIDW	RTIDV	RTIDU

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

23 紧急刹车模块（EMB）

23.1 简介

紧急刹车模块是在满足一定条件时产生控制事件输出给定时器，以控制定时器停止向外部电机输出 PWM 信号的功能模块，下列要因用于产生控制事件：

- 外部端口输入电平变化
- PWM 输出端口电平发生同相（同高或同低）
- 电压比较器比较结果
- 外部振荡器停止振荡
- 写寄存器软件控制

EMB 结构框图如图 23-1 所示。

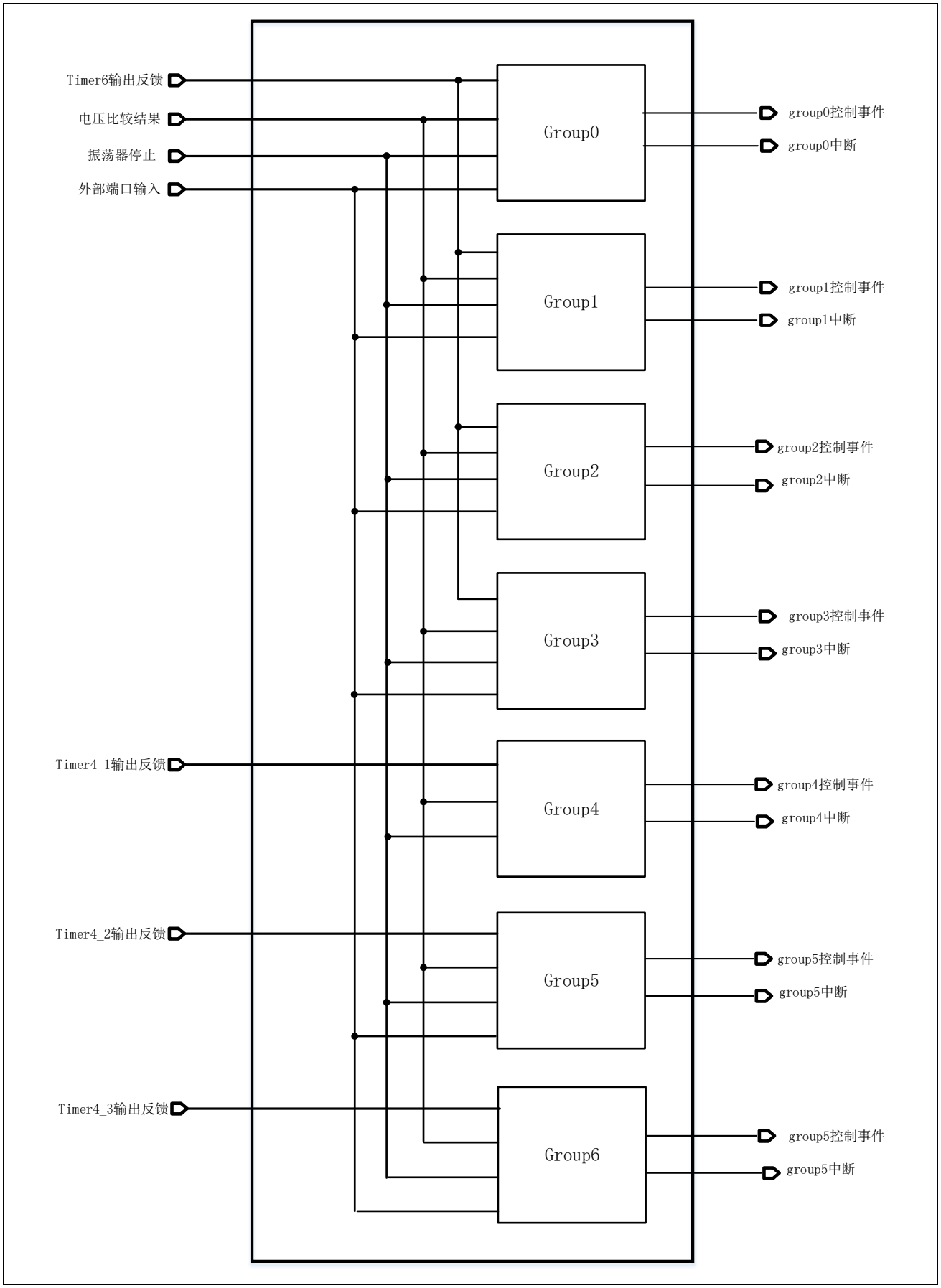
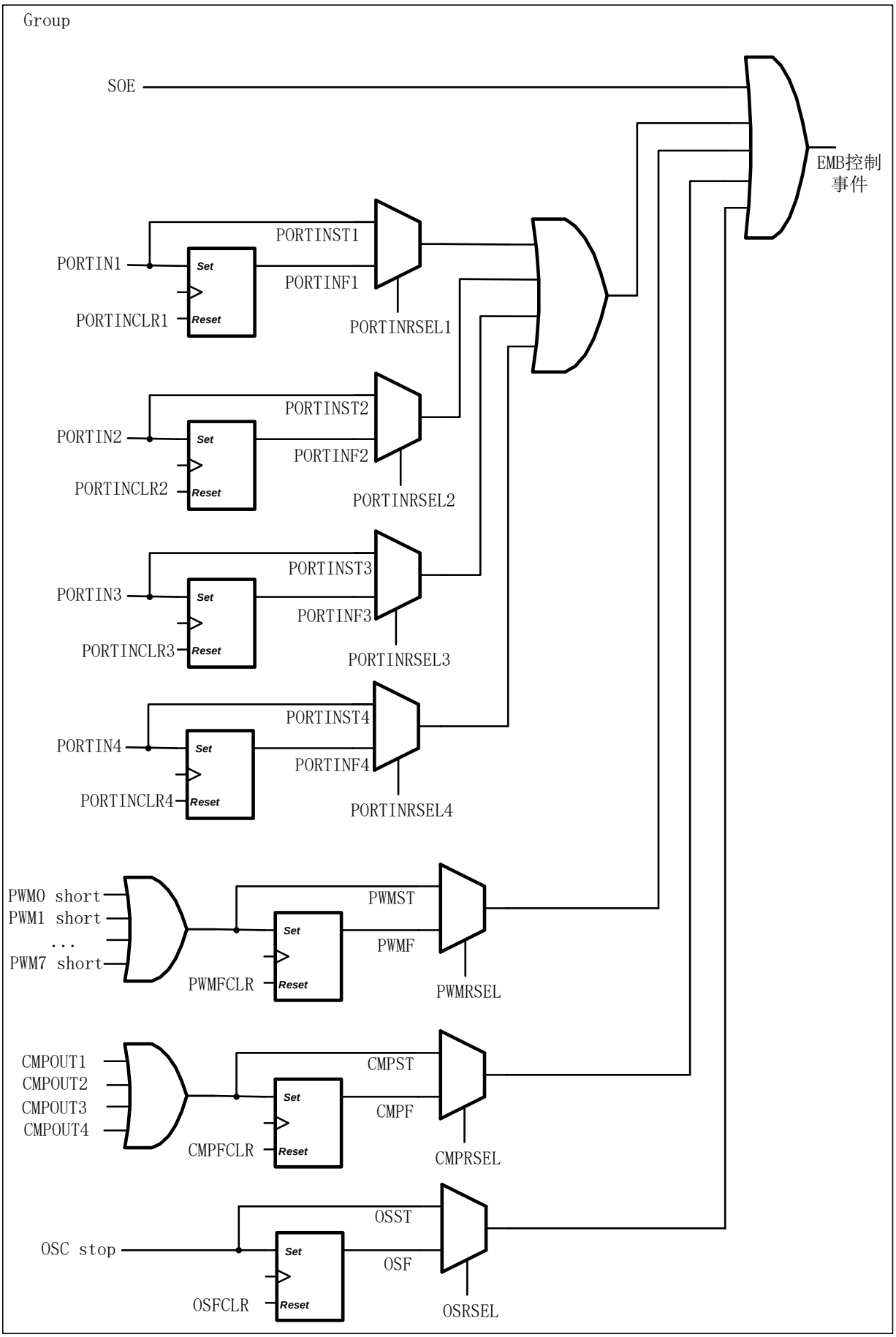


图 23-1 EMB 结构框图

23.2 功能描述

23.2.1 概述

EMB 用于向具有 PWM 功能的定时器模块(Timer4，Timer6)在满足一定条件时输出一个控制事件信号，通知定时器模块关闭当前 PWM 输出。EMB 模块具有 7 个组群(group)，其中 group0~group3 用于控制 Timer6，由 Timer6 寄存器设定选择使用，group4~group6 用于控制 Timer4，分别对应 Timer4 的 3 个单元。


Group
SOE
EMB控制
事件
PORTIN1
PORTINCLR1
Set
Reset
PORTINST1
PORTINF1
PORTINRSEL1
PORTIN2
PORTINCLR2
PORTINST2
PORTINF2
PORTINRSEL2
PORTIN3
PORTINCLR3
PORTINST3
PORTINF3
PORTINRSEL3
PORTIN4
PORTINCLR4
PORTINST4
PORTINF4
PORTINRSEL4
PWM0 short
PWM1 short
...
PWM7 short
PWMFCLR
PWMST
PWMF
PWMRSEL
CMPOUT1
CMPOUT2
CMPOUT3
CMPOUT4
CMPFCLR
CMPST
CMPF
CMPRSEL
OSC stop
OSFCLR
OSST
OSF
OSRSEL

23.2.2 外部端口输入电平变化时控制 PWM 信号输出

EMB 共有 4 个外部端口用于实现当输入电平变化时控制 PWM 信号输出。每个组群能够独立设置 4 个外部端口中的 1 个或多个有效，端口分配如下表所示。

端口	PIN 名
PORT1	PA11,PB2,PC5,PD7
PORT2	PA6,PE15,PB11,PB12
PORT3	PA7,PC2,PD9
PORT4	PD4,PH2,PB15,PI4

使用外部端口输入电平变化时控制 PWM 信号输出时，首先将使能位，即 EMB 控制寄存器 EMB_CTL1_x. PORTINENy (x=0~5， y=1~4) 置为有效，同时通过 EMB_CTL1_x.INVSEL y(x=0~5,y=1~4)设置在端口电平为高(INVSEL=0)时或端口电平为低(INVSEL=1)时产生控制信号。根据需要通过 EMB_CTL2_x.NFENy(x=0~6,y=1~4)和 EMB_CTL2_x.NFSELy (x=0~6,y=1~4)使能滤波功能和设置滤波时钟。

滤波器根据滤波时钟对输入信号进行采样后，采用三次比较一致的方式进行滤波，即当滤波时钟采样到端口上 3 次一致的电平时，该电平被当作有效电平传送到模块内部；小于 3 次一致的电平会被当作外部干扰滤掉，不传送到模块内部。

当端口上产生符合条件的有效电平时，EMB 状态寄存器的端口输入状态 EMB_STATx.PORTINSTy(x=0~6,y=1~4) 被置位，同时端口输入控制标志位 EMB_STATx.PORTINFy(x=0~6,y=1~4) 被置位。当中断许可寄存器 EMB_INTENx.PORTINEN(x=0~6)=1 时将产生中断。

EMB 的控制事件通过 EMB 释放方式选择寄存器 EMB_RLSSELx.PORTINRSELy (x=0~6,y=1~4) 选择由 EMB_STATx.PORTINSTy(x=0~6,y=1~4) 或者 EMB_STATx.PORTINFy(x=0~6,y=1~4)向 Timer6 和 Timer4 输出。Timer6 和 Timer4 在接到控制事件后，可根据寄存器设定将输出端口置为高电平，低电平或高阻态。

释放控制时，如果 EMB_RLSSELx. PORTINRSELy(x=0~6,y=1~4)=1，当端口输入电平转为无效时，EMB_STATx.PORTINSTy(x=0~6,y=1~4)将自动清零，同时控制事件将立即释放。如果 EMB_RLSSELx.PORTINRSELy(x=0~6,y=1~4)=0，当端口输入电平转为无效后，需要写 EMB 状态复位寄存器 EMB_STATCLRx .PORTINFCLR (x=0~6)使

EMB_STATx.PORTINF (x=0~6)清 0，将控制事件释放。

23.2.3 PWM 输出端口电平发生同相（同高或同低）时停止 PWM 信号输出

使用时首先将使能位，即 EMB 控制寄存器 EMB_CTL1_x. PWMSEN y(x=0~3,y=0~7 或 x=4~6,y=0~2) 置为有效，同时通过 EMB 控制寄存器 EMB_CTL2_x.PWMLVy(x=0~3,y=0~7 或 x=4~6,y=0~2)选择需要监控的有效电平。

完成设置后，EMB 监控 Timer6 和 Timer4 的互补 PWM 输出信号，当输出信号出现同高或同低情况时，EMB 状态寄存器的 PWM 输出状态 EMB_STATx.PWMST(x=0~6)被置位，同时 PWM 输出同相位控制标志位 EMB_STATx.PWMF(x=0~6)被置位。当中断许可寄存器 EMB_INTENx. PWMEN (x=0~6)=1 时将产生中断。

EMB 的控制事件通过 EMB 释放方式选择寄存器 EMB_RLSSELx.PWMRSEL(x=0~6)选择由 EMB_STATx. PWMST (x=0~6)或者 EMB_STATx. PWMF (x=0~6)向 Timer6 和 Timer4 输出。Timer6 和 Timer4 在接到控制事件后可根据寄存器设定将输出端口置为高电平，低电平或高阻态。

释放控制时，如果 EMB_RLSSELx. PWMRSEL(x=0~6)=1，当 PWM 输出同高或同低状态解除，EMB_STATx.PWMST(x=0~6)将自动清零，同时控制事件将立即释放。如果 EMB_RLSSELx. PWMST(x=0~6)=0，当 PWM 输出同高或同低状态解除后，需要写 EMB 状态复位寄存器 EMB_STATCLRx .PWMFCLR (x=0~6)使 EMB_STATxPWMF (x=0~6)清 0 后，将控制事件释放。

group0-3 可用于监控 Timer6 的互补 PWM 输出信号，group4 用于监控 Timer4_1 的互补 PWM 输出信号，group5 用于监控 Timer4_2 的互补 PWM 输出信号，group6 用于监控 Timer4_3 的互补 PWM 输出信号。

端口名	功能	对应组群	EMB_CTL1 控制位	EMB_CTL2 控制位
TIM6_m_PWMA(m=1~8)	Timer6的互补PWM输出信号	group0 group1 group2 group3	PWMSEN[7:0]	PWMLV[7:0]
TIM6_m_PWMB(m=1~8)				

TIM4_1_OUH	Timer4_1的互补PWM输出信号	group4	PWMSEN[2]	PWMLV[2]
TIM4_1_OUL				
TIM4_1_OVH			PWMSEN[1]	PWMLV[1]
TIM4_1_OVL				
TIM4_1_OWH			PWMSEN[0]	PWMLV[0]
TIM4_1_OWL				
TIM4_2_OUH	Timer4_2的互补PWM输出信号	group5	PWMSEN[2]	PWMLV[2]
TIM4_2_OUL				
TIM4_2_OVH			PWMSEN[1]	PWMLV[1]
TIM4_2_OVL				
TIM4_2_OWH			PWMSEN[0]	PWMLV[0]
TIM4_2_OWL				
TIM4_3_OUH	Timer4_3的互补PWM输出信号	group6	PWMSEN[2]	PWMLV[2]
TIM4_3_OUL				
TIM4_3_OVH			PWMSEN[1]	PWMLV[1]
TIM4_3_OVL				
TIM4_3_OWH			PWMSEN[0]	PWMLV[0]
TIM4_3_OWL				

23.2.4 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 4 组比较结果向 Timer6 和 Timer4 发送控制事件信号。电压比较器输出结果的设定请参考电压比较器章节。

使用时首先将使能位，即 EMB 控制寄存器 EMB_CTL1_x. CMPEN y(x=0~6)置为有效。

当电压比较器比较结果标志位被置起时，EMB 状态寄存器的电压比较器状态 EMB_STATx.CMPST(x=0~6) 被置位，同时 EMB 电压比较器控制标志位 EMB_STATx.CMPF(x=0~6) 被置位。当中断许可寄存器 EMB_INTENx. CMPINTEN(x=0~6)=1 时将产生中断。

EMB 的控制事件通过 EMB 释放方式选择寄存器 EMB_RLSSELx.CMPRSELy(x=0~6)

选择由 EMB_STATx.CMPST(x=0~6)或者 EMB_STATx.CMPF (x=0~6)向 Timer6 和 Timer4 输出。Timer6 和 Timer4 在接到控制事件信号后可根据寄存器设定将输出端口置为高电平，低电平或高阻态。

释放控制时，如果 EMB_RLSSELx. CMPRSEL(x=0~6)=1，当电压比较器结果转为无效时，EMB_STATx.CMPST(x=0~6)将自动清零，同时控制事件将立即释放。如果 EMB_RLSSELx.CMPRSELy(x=0~6)=0，当电压比较器结果转为无效后，需要写 EMB 状态复位寄存器 EMB_STATCLRx .CMPFCLR(x=0~6)使 EMB_STATx.CMPF (x=0~6)清 0 后，将控制事件释放。

23.2.5 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振荡时向 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。

使用时首先将使能位，即 EMB 控制寄存器 EMB_CTL1_x. OSCSTPEN y(x=0~6)置为有效。当外部振荡器停止震荡标志被置起时，EMB 状态寄存器的振荡器状态 EMB_STATx.OSST(x=0~6)被置位，同时 EMB 振荡器停止振荡控制标志位 EMB_STATx.OSF(x=0~6) 被置位。当中断许可寄存器 EMB_INTENx. OSINTEN(x=0~6)=1 时还将产生中断。

EMB 的控制事件通过 EMB 释放方式选择寄存器 EMB_RLSSELx.OSRSELy(x=0~6)选择由 EMB_STATx.OSST(x=0~6)或者 EMB_STATx.OSF (x=0~6)向 Timer6 和 Timer4 输出。Timer6 和 Timer4 在接到控制事件信号后可根据寄存器设定将输出端口置为高电平，低电平或高阻态。

释放控制时，如果 EMB_RLSSELx. OSRSEL(x=0~6)=1，当外部振荡器停止震荡标志转为无效时 EMB_STATx.OSST(x=0~6)将自动清零，同时控制事件将立即释放。如果 EMB_RLSSELx.OSRSELy(x=0~6)=0，当外部振荡器停止震荡标志转为无效后，需要写 EMB 状态复位寄存器 EMB_STATCLRx .OSFCLR(x=0~6)使 EMB_STATx.OSF (x=0~6)清 0 后，将控制事件释放。

23.2.6 写寄存器软件控制 PWM 信号输出

EMB 的软件输出使能控制寄存器(EMB_SOEx)(x=0~6)能够允许用户通过软件直接置

位和复位的方式向 Timer6 和 Timer4 发送控制信号，软件控制 PWM 输出时不会产生中断请求。

23.3 寄存器说明

寄存器一览

名称	英文缩写	说明	偏移地址
EMB控制寄存器1	EMB_CTL1	各个PWM输出控制事件的使能	0x0
EMB控制寄存器2	EMB_CTL2	端口输入控制事件的滤波以及选择PWM反馈信号的有效电平	0x4
EMB软件输出使能控制寄存器	EMB_SOE	软件产生PWM输出控制事件	0x8
EMB状态寄存器	EMB_STAT	表示PWM输出控制的状态	0xC
EMB状态复位寄存器	EMB_STATCLR	清除PWM输出控制的状态	0x10
EMB中断许可寄存器	EMB_INTEN	中断使能	0x14
EMB控制PWM输出释放方式选择寄存器	EMB_RSEL	选择各个PWM输出控制事件的释放方式	0x18

23.3.1 EMB 控制寄存器 1_0~3(EMB_CTL1_0~3)

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：0x40017C00, 0x40017C20, 0x40017C40, 0x40017C60

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved						INVSEL4	INVSEL3	INVSEL2	INVSEL1	Reserved		PORTINEN4	PORTINEN3	PORTINEN2	PORTINEN1

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved			PWMSEN7	PWMSEN6	PWMSEN5	PWMSEN4	PWMSEN3	PWMSEN2	PWMSEN1	PWMSEN0	OSCSTPEN	CMPEN[3:0]			

位	标记	位名	功能	读写
b31~b26	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b25	INVSEL4	端口4输入有效电平选择	0：高电平有效 1：低电平有效	R/W
b24	INVSEL3	端口3输入有效电平选择	0：高电平有效 1：低电平有效	R/W
b23	INVSEL2	端口2输入有效电平选择	0：高电平有效 1：低电平有效	R/W
b22	INVSEL1	端口1输入有效电平选择	0：高电平有效 1：低电平有效	R/W
b21~b20	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b19	PORTINEN4	端口4输入控制使能	0：端口输入控制无效 1：端口输入控制有效	R/W
b18	PORTINEN3	端口3输入控制使能	0：端口输入控制无效 1：端口输入控制有效	R/W
b17	PORTINEN2	端口2输入控制使能	0：端口输入控制无效 1：端口输入控制有效	R/W
b16	PORTINEN1	端口1输入控制使能	0：端口输入控制无效 1：端口输入控制有效	R/W
b15~b13	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b12~b5	PWMSEN[7:0]	TIM6_m_PWMA/B(m=1~8) 短路输出控制使能	对应8组PWM反馈输入 0：短路时输出控制无效 1：短路时输出控制有效	R/W
b4	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
b3~b0	CMPEN[3:0]	电压比较器比较结果控制使能	对应 4 组电压比较器比较结果 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

23.3.2 EMB 控制寄存器 1_4~6(EMB_CTL1_4~6)

该寄存器为单次写入寄存器，即复位之后仅能够写入一次

地址：0x40017C80, 0x40017CA0, 0x40017CC0

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved						INVSEL4	INVSEL3	INVSEL2	INVSEL1	Reserved		PORTINEN4	PORTINEN3	PORTINEN2	PORTINEN1

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								PWMSEN2	PWMSEN1	PWMSEN0	OSCSTPEN	CMPEN[3:0]			

位	标记	位名	功能	读写
b31~b24	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b25	INVSEL4	端口4输入有效电平选择	0：高电平有效 1：低电平有效	R/W
b24	INVSEL3	端口3输入有效电平选择	0：高电平有效 1：低电平有效	R/W
b23	INVSEL2	端口2输入有效电平选择	0：高电平有效 1：低电平有效	R/W
b22	INVSEL1	端口1输入有效电平选择	0：高电平有效 1：低电平有效	R/W
b21~b20	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b19	PORTINEN4	端口4输入控制使能	0：端口输入控制无效 1：端口输入控制有效	R/W
b18	PORTINEN3	端口3输入控制使能	0：端口输入控制无效 1：端口输入控制有效	R/W
b17	PORTINEN2	端口2输入控制使能	0：端口输入控制无效 1：端口输入控制有效	R/W
b16	PORTINEN1	端口1输入控制使能	0：端口输入控制无效 1：端口输入控制有效	R/W
b15~b8	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b7	PWMSEN2	TIM4_m_OUH/L 短路输出控制使能（m=1~3）	0：短路时输出控制无效 1：短路时输出控制有效	R/W
b6	PWMSEN1	TIM4_m_OVH/L 短路输出控制使能（m=1~3）	0：短路时输出控制无效 1：短路时输出控制有效	R/W
b5	PWMSEN0	TIM4_m_OWH/L 短路输出控制使能（m=1~3）	0：短路时输出控制无效 1：短路时输出控制有效	R/W
b4	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

b3~b0	CMPEN[3:0]	电压比较器比较结果控制使能	对应 4 组电压比较器比较结果 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

23.3.3 EMB 控制寄存器 2_0~3(EMB_CTL2_0~3)

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：0x40017C04, 0x40017C24, 0x40017C44, 0x40017C64

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved				NFEN4	NFSEL4[1:0]		NFEN3	NFSEL3[1:0]		NFEN2	NFSEL2[1:0]		NFEN1	NFSEL1[1:0]	

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								PWMLV7	PWMLV6	PWMLV5	PWMLV4	PWMLV3	PWMLV2	PWMLV1	PWMLV0

位	标记	位名	功能	读写
b31~28	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b27	NFEN4	端口4输入数字滤波器使能	0：滤波器无效 1：滤波器有效	R/W
b26~b25	NFSEL4[1:0]	端口4数字滤波器滤波时钟选择	00：使用总线时钟滤波 01：使用总线时钟的8分频滤波 10：使用总线时钟的32分频滤波 11：使用总线时钟的128分频滤波	R/W
b24	NFEN3	端口3输入数字滤波器使能	0：滤波器无效 1：滤波器有效	R/W
b23~b22	NFSEL3[1:0]	端口3数字滤波器滤波时钟选择	00：使用总线时钟滤波 01：使用总线时钟的8分频滤波 10：使用总线时钟的32分频滤波 11：使用总线时钟的128分频滤波	R/W
b21	NFEN2	端口2输入数字滤波器使能	0：滤波器无效 1：滤波器有效	R/W
b20~b19	NFSEL2[1:0]	端口2数字滤波器滤波时钟选择	00：使用总线时钟滤波 01：使用总线时钟的8分频滤波 10：使用总线时钟的32分频滤波 11：使用总线时钟的128分频滤波	R/W
b18	NFEN1	端口1输入数字滤波器使能	0：滤波器无效 1：滤波器有效	R/W
b17~b16	NFSEL1[1:0]	端口1数字滤波器滤波时钟选择	00：使用总线时钟滤波 01：使用总线时钟的8分频滤波 10：使用总线时钟的32分频滤波 11：使用总线时钟的128分频滤波	R/W
b15~b8	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b7~b0	PWMLV[7:0]	TIM6_m_PWMA/B(m=1~8)	对应8组PWM反馈输入	R/W

	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

23.3.4 EMB 控制寄存器 2_4~6(EMB_CTL2_4~6)

该寄存器为单次写入寄存器，即复位之后仅能够写入一次

地址：0x40017C84, 0x40017CA4, 0x40017CC4

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved				NFEN4	NFSEL4[1:0]		NFEN3	NFSEL3[1:0]		NFEN2	NFSEL2[1:0]		NFEN1	NFSEL1[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved													PWMLV2	PWMLV1	PWMLV0

位	标记	位名	功能	读写
b31~28	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b27	NFEN4	端口4输入数字滤波器使能	0：滤波器无效 1：滤波器有效	R/W
b26~b25	NFSEL4[1:0]	端口4数字滤波器滤波时钟选择	00：使用总线时钟滤波 01：使用总线时钟的8分频滤波 10：使用总线时钟的32分频滤波 11：使用总线时钟的128分频滤波	R/W
b24	NFEN3	端口3输入数字滤波器使能	0：滤波器无效 1：滤波器有效	R/W
b23~b22	NFSEL3[1:0]	端口3数字滤波器滤波时钟选择	00：使用总线时钟滤波 01：使用总线时钟的8分频滤波 10：使用总线时钟的32分频滤波 11：使用总线时钟的128分频滤波	R/W
b21	NFEN2	端口2输入数字滤波器使能	0：滤波器无效 1：滤波器有效	R/W
b20~b19	NFSEL2[1:0]	端口2数字滤波器滤波时钟选择	00：使用总线时钟滤波 01：使用总线时钟的8分频滤波 10：使用总线时钟的32分频滤波 11：使用总线时钟的128分频滤波	R/W
b18	NFEN1	端口1输入数字滤波器使能	0：滤波器无效 1：滤波器有效	R/W
b17~b16	NFSEL1[1:0]	端口1数字滤波器滤波时钟选择	00：使用总线时钟滤波 01：使用总线时钟的8分频滤波 10：使用总线时钟的32分频滤波 11：使用总线时钟的128分频滤波	R/W
b15~b3	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b2	PWMLV2	TIM4_m_OUH/L输出有效电	0：低电平为有效电平	R/W

		平选择（m=1~3）	1：高电平为有效电平	
b1	PWMLV1	TIM4_m_OVH/L输出有效电平选择（m=1~3）	0：低电平为有效电平 1：高电平为有效电平	R/W
b0	PWMLV0	TIM4_m_OWH/L输出有效电平选择（m=1~3）	0：低电平为有效电平 1：高电平为有效电平	R/W

23.3.5 EMB 软件输出使能控制寄存器(EMB_SOEx)(x=0~6)

地址：0x40017C08, 0x40017C28, 0x40017C48, 0x40017C68, 0x40017C88, 0x40017CA8, 0x40017CC8

复位值：0x00000000

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

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

位	标记	位名	功能	读写
b31~1	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b0	SOE	软件控制输出	0：PWM正常输出 1：PWM停止输出	R/W

23.3.6 EMB 状态寄存器(EMB_STATx) (x=0~6)

地址：0x40017C0C, 0x40017C2C, 0x40017C4C, 0x40017C6C, 0x40017C8C, 0x40017CAC, 0x40017CCC

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved														PORTINST4	PORTINST3

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PORTINST2	PORTINST1	Reserved		PORTINF4	PORTINF3	PORTINF2	PORTINF1	OSST	CMPST	PWMST	Reserved	OSF	CMPF	PWMSF	Reserved

位	标记	位名	功能	读写
b31~18	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b17	PORTINST4	端口4输入控制状态	0：端口4输入控制处于无效电平状态 1：端口4输入控制处于有效电平状态	R
b16	PORTINST3	端口3输入控制状态	0：端口3输入控制处于无效电平状态 1：端口3输入控制处于有效电平状态	R
b15	PORTINST2	端口2输入控制状态	0：端口2输入控制处于无效电平状态 1：端口2输入控制处于有效电平状态	R
b14	PORTINST1	端口1输入控制状态	0：端口1输入控制处于无效电平状态 1：端口1输入控制处于有效电平状态	R
b13~12	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b11	PORTINF4	端口4输入控制标志位	EMB_STATx. PORTINST4为1时置位 EMB_STATCLRx. PORTINFCLR4写1时清0	R
b10	PORTINF3	端口3输入控制标志位	EMB_STATx. PORTINST3为1时置位 EMB_STATCLRx. PORTINFCLR3写1时清0	R
b9	PORTINF2	端口2输入控制标志位	EMB_STATx. PORTINST2为1时置位 EMB_STATCLRx. PORTINFCLR2写1时清0	R
b8	PORTINF1	端口1输入控制标志位	EMB_STATx. PORTINST1为1时置位 EMB_STATCLRx. PORTINFCLR1写1时清0	R
b7	OSST	振荡器状态	0：振荡器未停止震荡 1：振荡器停止震荡	R
b6	CMPST	电压比较器状态	电压比较器比较结果	R
b5	PWMST	PWM输出状态	0：没有发生PWM输出同相 1：发生PWM输出同相	R
b4	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b3	OSF	振荡器停止震荡控制标志位	EMB_STATx. OSST为1时置位 EMB_STATCLRx.OSFCLR写1时清0	R
b2	CMPF	电压比较器控制标志位	EMB_STATx. CMPST为1时置位	R

			EMB_STATCLRx. CMPFCLR写1时清0	
b1	PWMSF	PWM输出同相位控制标志位	EMB_STATx. PWMST为1时置位 EMB_STATCLRx. PWMSCLR写1时清0	R
b0	Reserved	-	读时读出“0”，写入时请写“0”	R/W

23.3.7 EMB 状态复位寄存器(EMB_STATCLRx) (x=0~6)

地址：0x40017C10, 0x40017C30, 0x40017C50, 0x40017C70, 0x40017C90, 0x40017CB0, 0x40017CD0

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				PORTINFCLR4	PORTINFCLR3	PORTINFCLR2	PORTINFCLR1	Reserved				OSFCLR	CMPFCLR	PWMSFCLR	Reserved

位	标记	位名	功能	读写
b31~b12	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b11	PORTINFCLR4	复位 EMB_STAT.PORTINF4	0：无任何效果 1：当EMB_STAT.PORTINST4=0时，将EMB_STAT. PORTINF4清0 读该寄存器位无任何效果	R/W
b10	PORTINFCLR3	复位 EMB_STAT.PORTINF3	0：无任何效果 1：当EMB_STAT.PORTINST3=0时，将EMB_STAT. PORTINF3清0 读该寄存器位无任何效果	R/W
b9	PORTINFCLR2	复位 EMB_STAT.PORTINF2	0：无任何效果 1：当EMB_STAT.PORTINST2=0时，将EMB_STAT. PORTINF2清0 读该寄存器位无任何效果	R/W
b8	PORTINFCLR1	复位 EMB_STAT.PORTINF1	0：无任何效果 1：当EMB_STAT.PORTINST1=0时，将EMB_STAT. PORTINF1清0 读该寄存器位无任何效果	R/W
b7~b4	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b3	OSFCLR	复位 EMB_STAT.OSF	0：无任何效果 1：当EMB_STAT.OSST=0时，将EMB_STAT.OSF清0 读该寄存器位无任何效果	R/W
b2	CMPFCLR	复位 EMB_STAT.CMPF	0：无任何效果 1：当EMB_STAT.CMPST=0时，将EMB_STAT. CMPF清0 读该寄存器位无任何效果	R/W
b1	PWMSCLR	复位 EMB_STAT.PWMSF	0：无任何效果 1：当EMB_STAT.PWMST=0时，将EMB_STAT. PWMSF清0 读该寄存器位无任何效果	R/W
b0	Reserved	-	读时读出“0”，写入时请写“0”	R/W

23.3.8 EMB 中断许可寄存器(EMB_INTENx)(x=0~6)

地址：0x40017C14, 0x40017C34, 0x40017C54, 0x40017C74, 0x40017C94, 0x40017CB4, 0x40017CD4

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				PORTINTEN4	PORTINTEN3	PORTINTEN2	PORTINTEN1	Reserved				OSINTEN	CMPINTEN	PWMSINTEN	Reserved

位	标记	位名	功能	读写
b31~b12	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b11	PORTINTEN4	端口4输入控制中断使能	0：端口4输入控制不产生中断 1：端口4输入控制产生中断	R/W
b10	PORTINTEN3	端口3输入控制中断使能	0：端口3输入控制不产生中断 1：端口3输入控制产生中断	R/W
b9	PORTINTEN2	端口2输入控制中断使能	0：端口2输入控制不产生中断 1：端口2输入控制产生中断	R/W
b8	PORTINTEN1	端口1输入控制中断使能	0：端口1输入控制不产生中断 1：端口1输入控制产生中断	R/W
b7~b4	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b3	OSINTEN	振荡器停止震荡控制中断使能	0：振荡器停止震荡控制不产生中断 1：振荡器停止震荡控制产生中断	R/W
b2	CMPINTEN	电压比较器比较结果控制中断使能	0：电压比较器比较结果控制不产生中断 1：电压比较器比较结果控制产生中断	R/W
b1	PWMSINTEN	PWM输出同相位控制中断使能	0：PWM输出同相位控制不产生中断 1：PWM输出同相位控制产生中断	R/W
b0	Reserved	-	读时读出“0”，写入时请写“0”	R/W

23.3.9 EMB 控制 PWM 输出释放方式选择寄存器(EMB_RLSSELx)(x=0~6)

地址：0x40017C18, 0x40017C38, 0x40017C58, 0x40017C78, 0x40017C98, 0x40017CB8, 0x40017CD8

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				PORTINRSEL4	PORTINRSEL3	PORTINRSEL2	PORTINRSEL1	Reserved				OSRSEL	CMPRSEL	PWMRSEL	Reserved

位	标记	位名	功能	读写
b31~b12	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b11	PORTINRSEL4	端口输入4控制释放方式选择	0：EMB_STATx. PORTINF4=0时释放PWM输出控制 1：EMB_STATx. PORTINST4=0时释放PWM输出控制	R/W
b10	PORTINRSEL3	端口输入3控制释放方式选择	0：EMB_STATx. PORTINF3=0时释放PWM输出控制 1：EMB_STATx. PORTINST3=0时释放PWM输出控制	R/W
b9	PORTINRSEL2	端口输入2控制释放方式选择	0：EMB_STATx. PORTINF2=0时释放PWM输出控制 1：EMB_STATx. PORTINST2=0时释放PWM输出控制	R/W
b8	PORTINRSEL1	端口输入1控制释放方式选择	0：EMB_STATx. PORTINF1=0时释放PWM输出控制 1：EMB_STATx. PORTINST1=0时释放PWM输出控制	R/W
b7~b4	Reserved	-	读时读出“0”，写入时请写“0”	R/W
b3	OSRSEL	振荡器停止震荡控制释放方式选择	0：EMB_STATx. OSF=0时释放PWM输出控制 1：EMB_STATx. OSST=0时释放PWM输出控制	R/W
b2	CMPRSEL	比较器比较结果控制释放方式选择	0：EMB_STATx. CMPF=0时释放PWM输出控制 1：EMB_STATx. CMPST=0时释放PWM输出控制	R/W
b1	PWMRSEL	PWM输出控制释放方式选择	0：EMB_STATx. PWMSF=0时释放PWM输出控制 1：EMB_STATx. PWMST=0时释放PWM输出控制	R/W
b0	Reserved	-	读时读出“0”，写入时请写“0”	R/W

24 通用定时器（TimerA）

24.1 简介

通用定时器 A（TimerA）是一个具有 16 位计数宽度、4 路 PWM 输出的定时器。该定时器支持三角波和锯齿波两种波形模式，可生成各种 PWM 波形（单边对齐 PWM、双边对称 PWM）；支持计数器同步启动；比较基准值寄存器支持缓存功能；支持单元间级联实现 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 12 个单元 TimerA，最大可实现 48 路 PWM 输出。

24.2 基本框图

TimerA 基本的功能及特性如表 24-1 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
	• 4路PWM输出
	• 比较匹配事件输出
中断类型	• 比较匹配中断
	• 周期匹配中断

表 24-1 TimerA 的基本功能及特性

TimerA 的基本框图如图 24-1 所示。图中”<t>”为单元编号，即”<t>”为 1~12，本章节后文提到“<t>”时均指单元编号，不再赘述。

本系列产品搭载的 12 个单元的 TimerA，其中单元 1~单元 4 的总线时钟和计数时钟均为 PCLK0（此时，框图中所示 PCLK 是指 PCLK0）；单元 5~单元 12 的总线时钟和计数时钟为 PCLK1（此时，框图中所示 PCLK 是指 PCLK1）。各单元的计数分频时钟源及端口数字滤波的采样基准时钟源与其总线时钟和计数时钟相同。

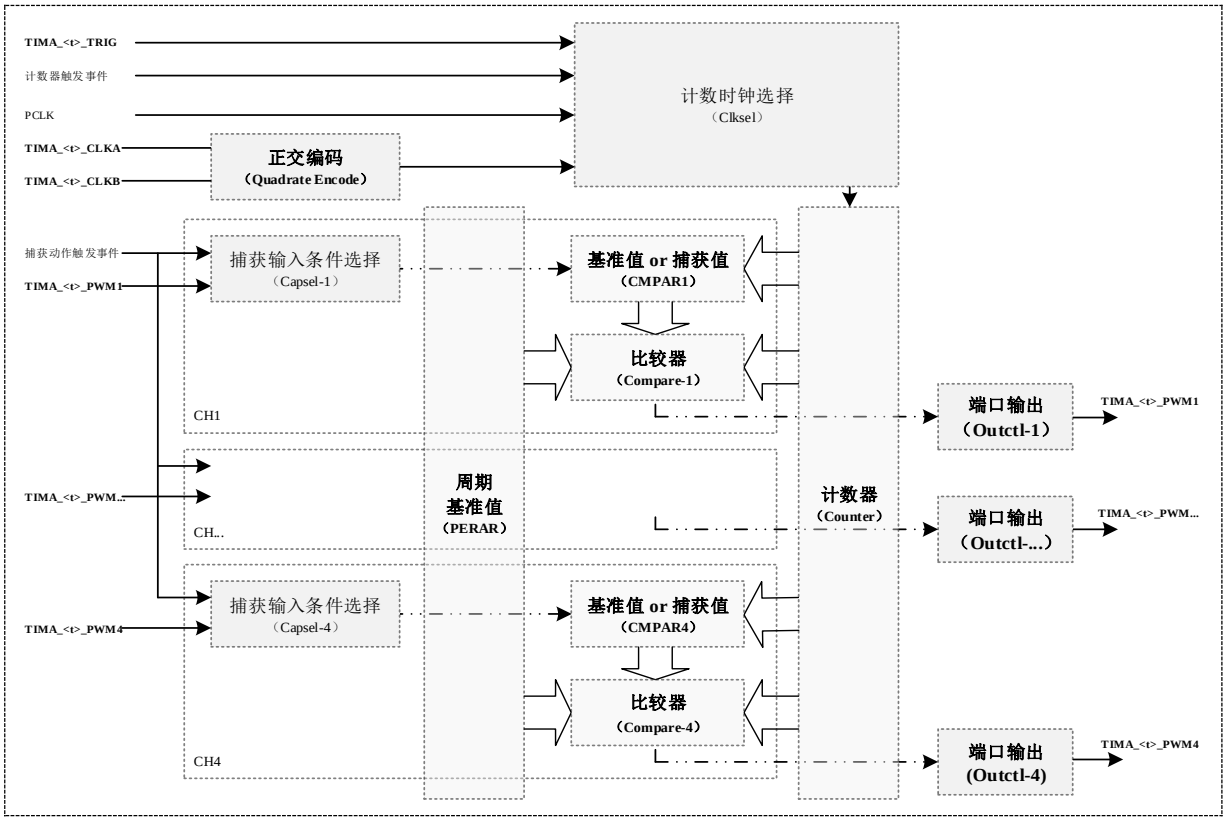


图 24-1 TimerA 基本框图

表 24-2 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=1~4）
TIMA_<t>_CLKA	in	正交编码计数事件输入端口
TIMA_<t>_CLKB		
TIMA_<t>_TRIG	in	硬件触发启动、停止、清零事件输入端口

表 24-2 TimerA 端口列表

24.3 功能说明

24.3.1 波形模式

TimerA 有 2 种基本计数波形模式，锯齿波模式和三角波模式。两种波形模式的基本波形如图 24-2、图 24-3 所示。

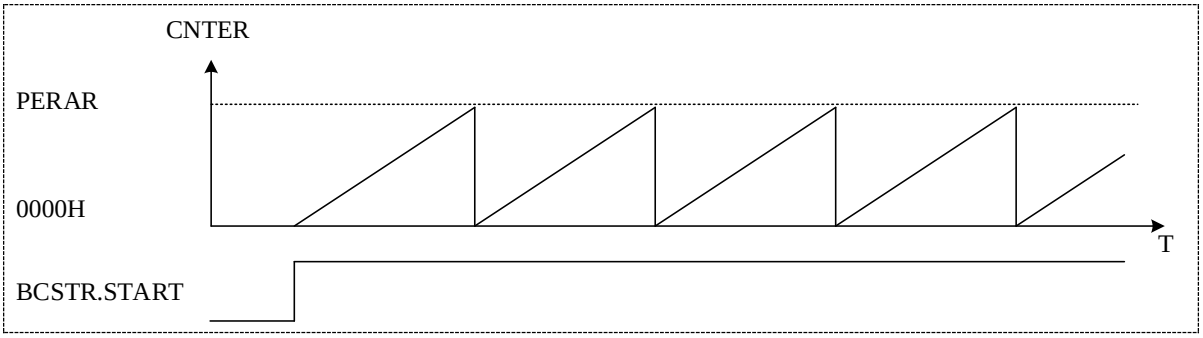


图 24-2 锯齿波波形（递加计数）

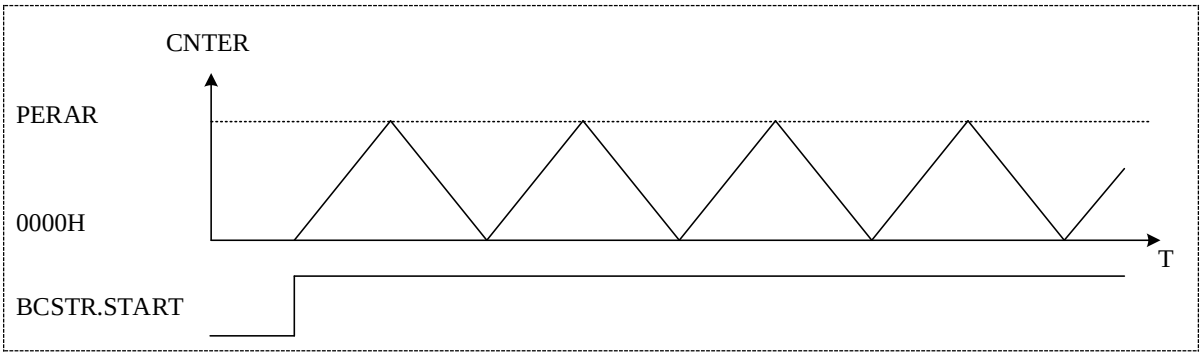


图 24-3 三角波波形

24.3.2 时钟源选择

TimerA 的计数时钟可以有以下几种选择：

a) PCLK 的 1、2、4、8、16、32、64、128、256、512、1024 分频（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。

24.3.3 比较输出

各个 TimerA 单元内部均含有 4 个通道的比较输出（TIMA_<t>_PWMn），可在计数值与比较基准值比较匹配时输出指定的电平。TMRA_CMPARn 寄存器分别对应了 TIMA_<t>_PWMn 输出端口的计数比较基准值。当定时器的计数值和 TMRA_CMPARn 相等时，TIMA_<t>_PWMn 端口输出指定的电平（n=1~4）。

TIMA_<t>_PWMn 端口的计数开始时的电平、计数停止时的电平、计数比较匹配时的电平、计数周期匹配时的电平等，可通过端口控制寄存器（PCONRn）的 STAC、STPC、CMPC、PERC、FORC 位设定控制（n=1~4）。图 24-4 为单元 1 的比较输出动作例。

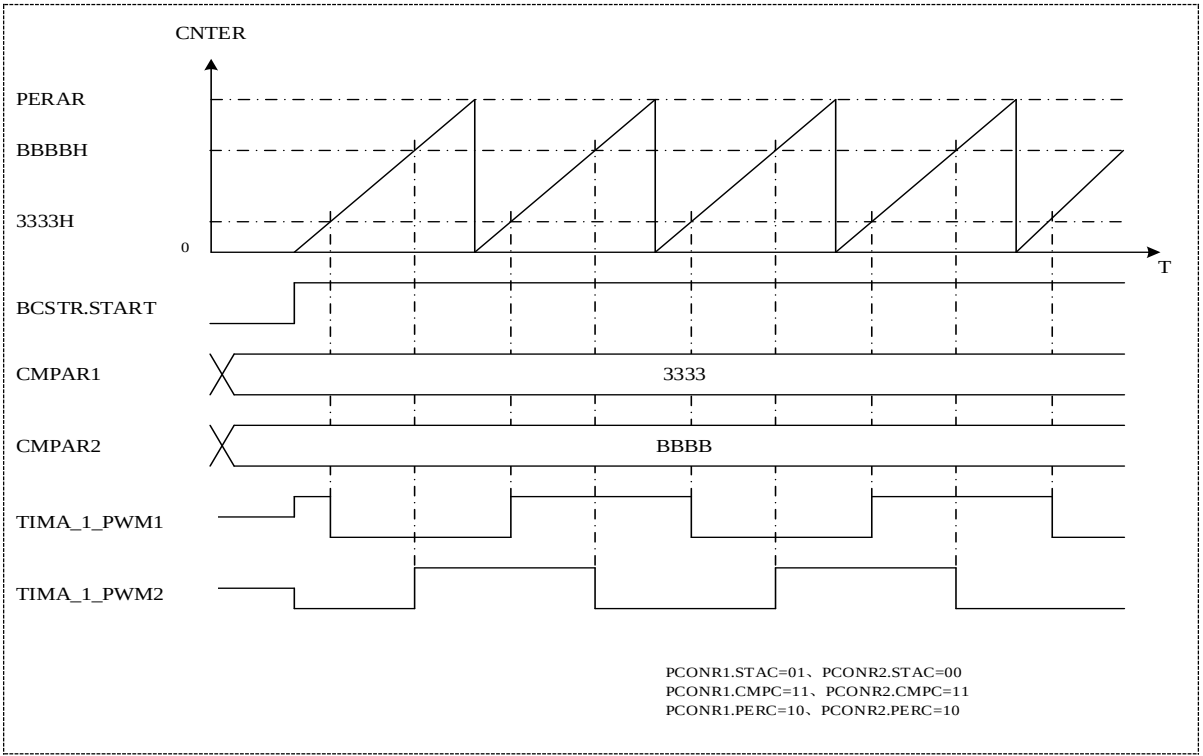


图 24-4 比较输出动作

24.3.4 捕获输入

各个 TimerA 单元的每个 PWM 输出通道都具有捕获输入功能，用于保存捕获到的计数值。设定捕获控制寄存器（CCONRn）的 CCONR.CAPMD 位为 1，捕获输入功能变

为有效。此时选择对应的捕获输入条件且该条件有效时，当前的计数值就被保存到相应的寄存器（CMPARn）中（n=1~4）。

捕获输入条件可以选择内部捕获动作触发事件（通过 HTSSR0~3 寄存器选择，具体参考寄存器说明章节）、TIMA_<t>_PWMn 端口输入等，具体的条件选择可通过捕获控制寄存器（CCONRn）的 HICP 位来设定（n=1~4）。图 24-5 为捕获输入动作例。

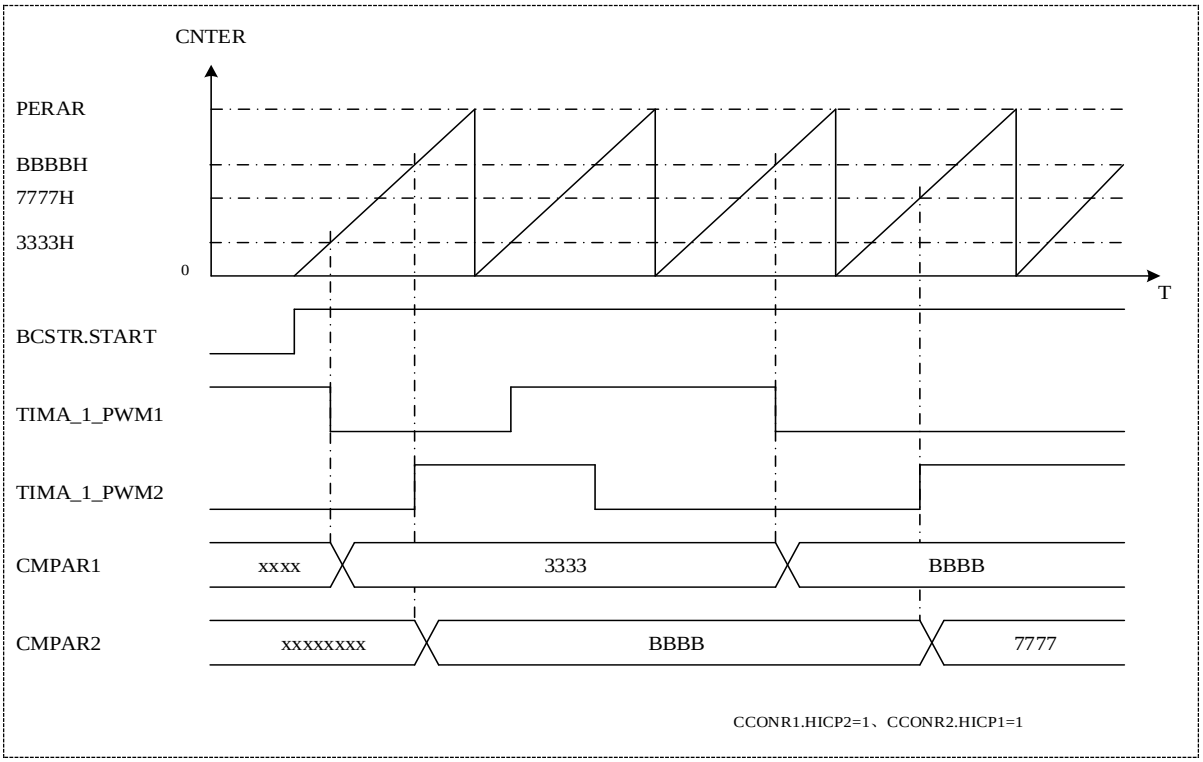


图 24-5　捕获输入动作

24.3.5 同步启动

本产品所搭载 12 个单元的 TimerA，可以实现软件同步启动或硬件同步启动。当本单元为单元 m 时，单元 m 可以选择与单元 n 同步启动（当 m=2、4、6、8、10、12 时，n=1、3、5、7、9、11）。

当单元 m 中的 BCSTR.SYNST 位设定为 1 时，单元 m 与单元 n 的同步启动功能有效。此时，若软件设定单元 n 的 BCSTR.START 位为 1，被同步单元（单元 m）的计数器开始软件同步计数；若硬件设定单元 n 的 HCONR.HSTA1~0 中任意位为 1，且单元 n 的对应硬件事件发生时，被同步单元（单元 m）的计数器开始硬件同步计数。在选择硬件同步计数启动功能时，被同步单元（单元 m）的 HCONR.HSTA1~0 的对应位也必须设定为有效（当 m=2、4、6、8、10、12 时，n=1、3、5、7、9、11）。

图 24-6 为设定单元 4 的 BCSTR.SYNST=1 时的软件同步启动例。

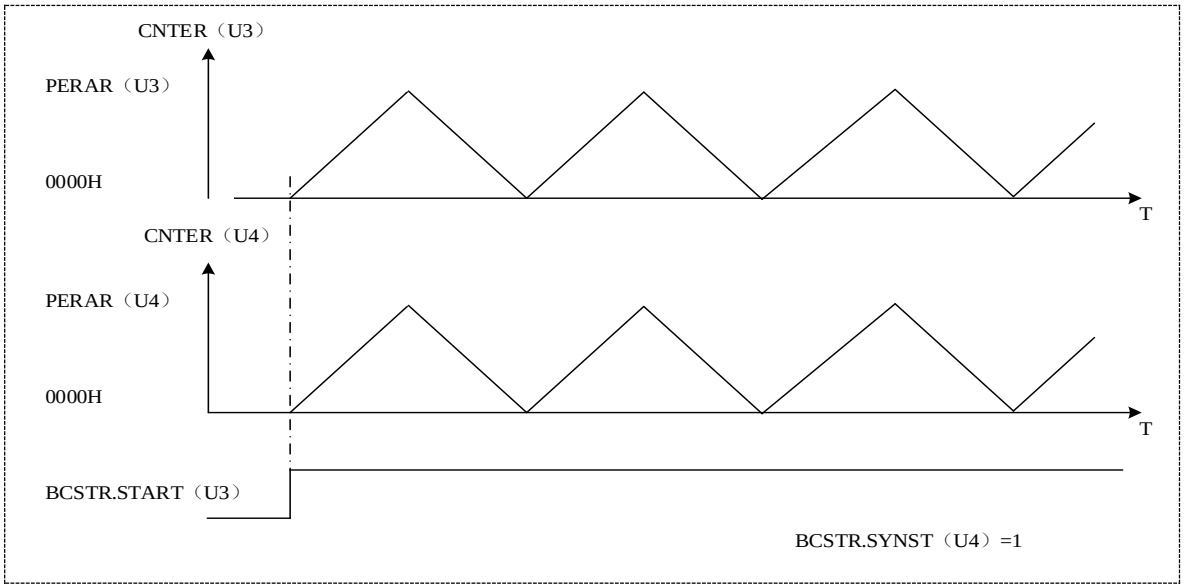


图 24-6 软件同步动作

24.3.6 数字滤波

各个单元的 TIMA_<t>_CLKA、TIMA_<t>_CLKB、TIMA_<t>_TRIG、TIMA_<t>_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（CCONRn）的对应位来实现（n=1~4）。

在滤波采样基准时钟采样到端口上 3 次一致的电平时，该电平被当作有效电平传送到模块内部；小于 3 次一致的电平会被当作外部干扰滤掉，不传送到模块内部。图 24-7 所示为 TIMA_1_CLKA 端口滤波动作例。

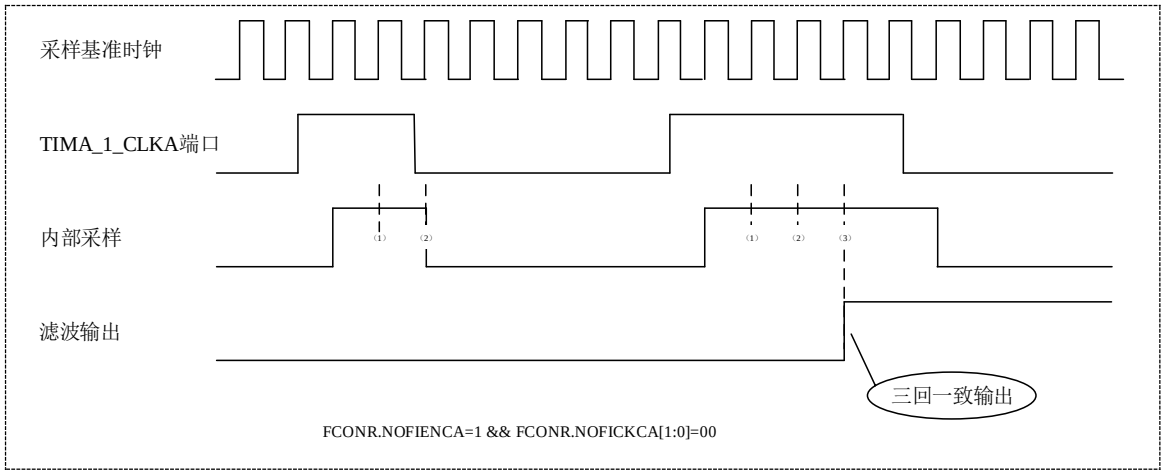


图 24-7 时钟输入端口的滤波功能

24.3.7 缓存功能

TimerA 的共计 4 个比较基准寄存器（CMPARn）可以成对实现缓存功能（n=1~4）。即 CMPAR2 作为 CMPAR1 的缓存基准值、CMPAR4 作为 CMPAR3 的缓存基准值。缓存控制寄存器（BCONRm）分别实现对四组缓存功能的控制（m=1~2）。

当缓存控制寄存器（BCONRm）的 BEN 位被置位时，缓存功能变为有效（m=1~2）。计数器计数到特定时间点时就发生一次缓存传送（CMPAR4/2->CMPAR3/1）。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）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=1~2）

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=1~2）

e) 硬件计数模式或锯齿波计数模式时，发生清零动作

下图 24-8 所示，是锯齿波模式时的缓存传送示意图。

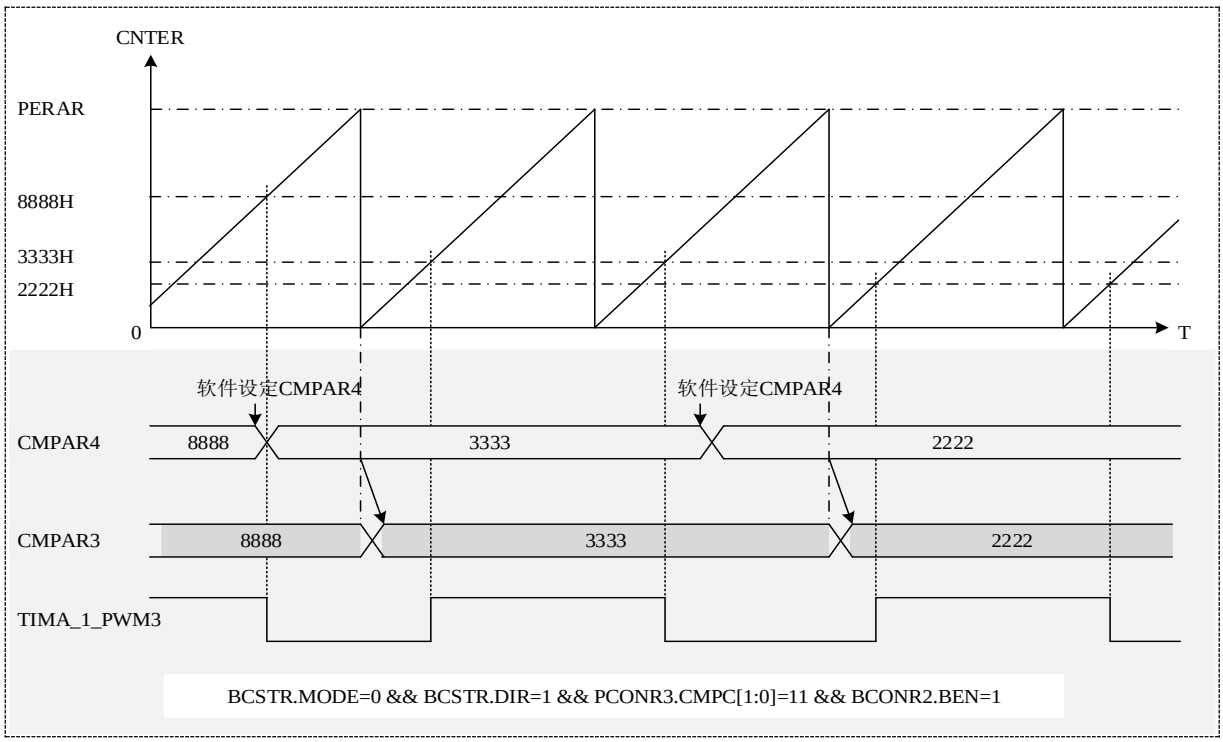


图 24-8 锯齿波模式时缓存动作

24.3.8 级联计数

在计数时钟源选择章节中，当时钟源选择 d)时，本单元的计数时钟就选择为对称单元的计数溢出（计数上溢或计数下溢）事件，此时两个单元级联合并可实现 32 位计数器。在级联计数中，对称单元的 CNTER 为低 16 位计数器，本单元的 CNTER 为高 16 位计数器。

例如，在三角波向上计数模式（BCSTR.MODE=0、BCSTR.DIR=1）时，设定单元 1 的计数时钟为 PCLK，设定单元 2 的计数时钟源为单元 1 的计数上溢事件（单元 2 的 TMRA_HCUPR.HCUP11=1），启动单元 1、2 计数（先启动单元 2，再启动单元 1）就实现级联计数。单元 1 的 CNTER 位低 16 位计数器，单元 2 的 CNTER 为高 16 位计数器。如图 24-9 所示是单元 1、2 级联计数的示意图。

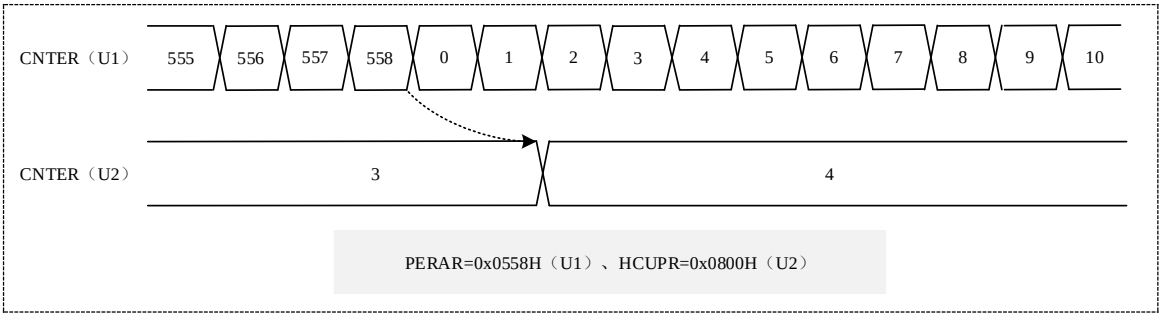


图 24-9　32 位级联计数动作

24.3.9 PWM 输出

24.3.9.1 单边对齐 PWM 输出

锯齿波计数模式时，一个单元内的各个通道通过各种端口控制设定，可以实现单边对齐 PWM 输出（在计数周期点对齐）。例如，设定在比较基准值比较匹配时翻转（PCONR.CMPC=11）、在周期基准值比较匹配时翻转（PCONR.PERC=11），就可实现在一个周期内产生单边对齐的 PWM 输出。

图 24-10 所示，是锯齿波模式下，单边对齐 PWM 输出波形例。

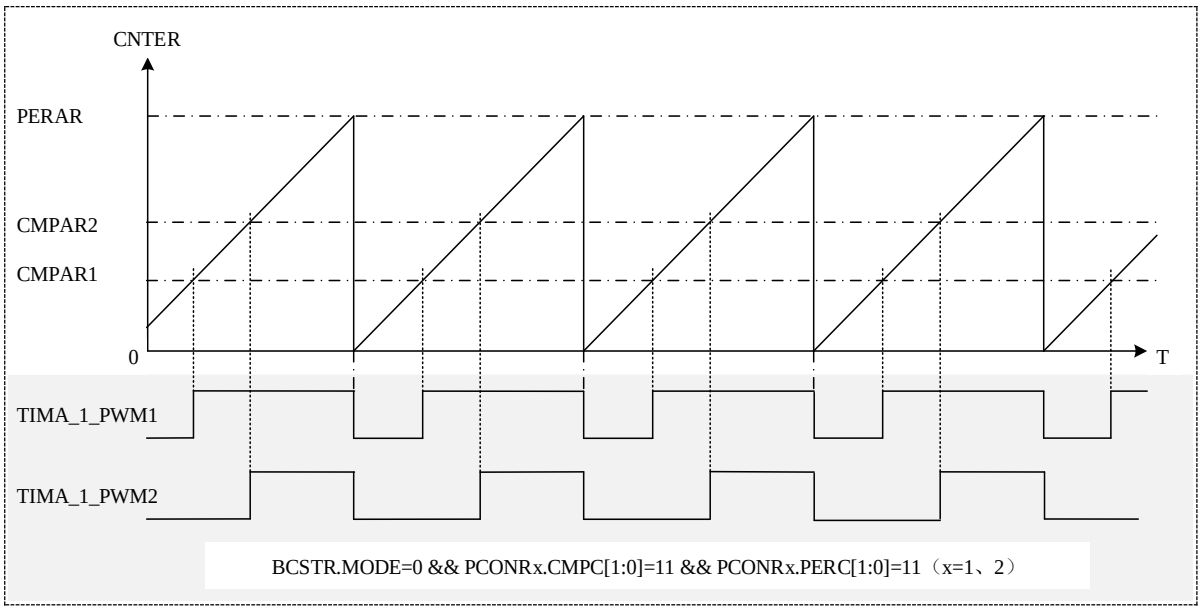


图 24-10　单边对齐 PWM 输出例

24.3.9.2 双边对称 PWM 输出

三角波计数模式时，一个单元内的各个通道通过各种端口控制设定，可以实现双边对称 PWM 输出（以计数峰点对称）。根据通道间的输出关系，可以实现独立的 PWM 输出或互补的 PWM 输出。

图 24-11 所示，是三角波模式下，通道 1、2、3、4 的双边对称 PWM 输出波形例。其中，通道 1、2 可以作为一对互补 PWM 输出。

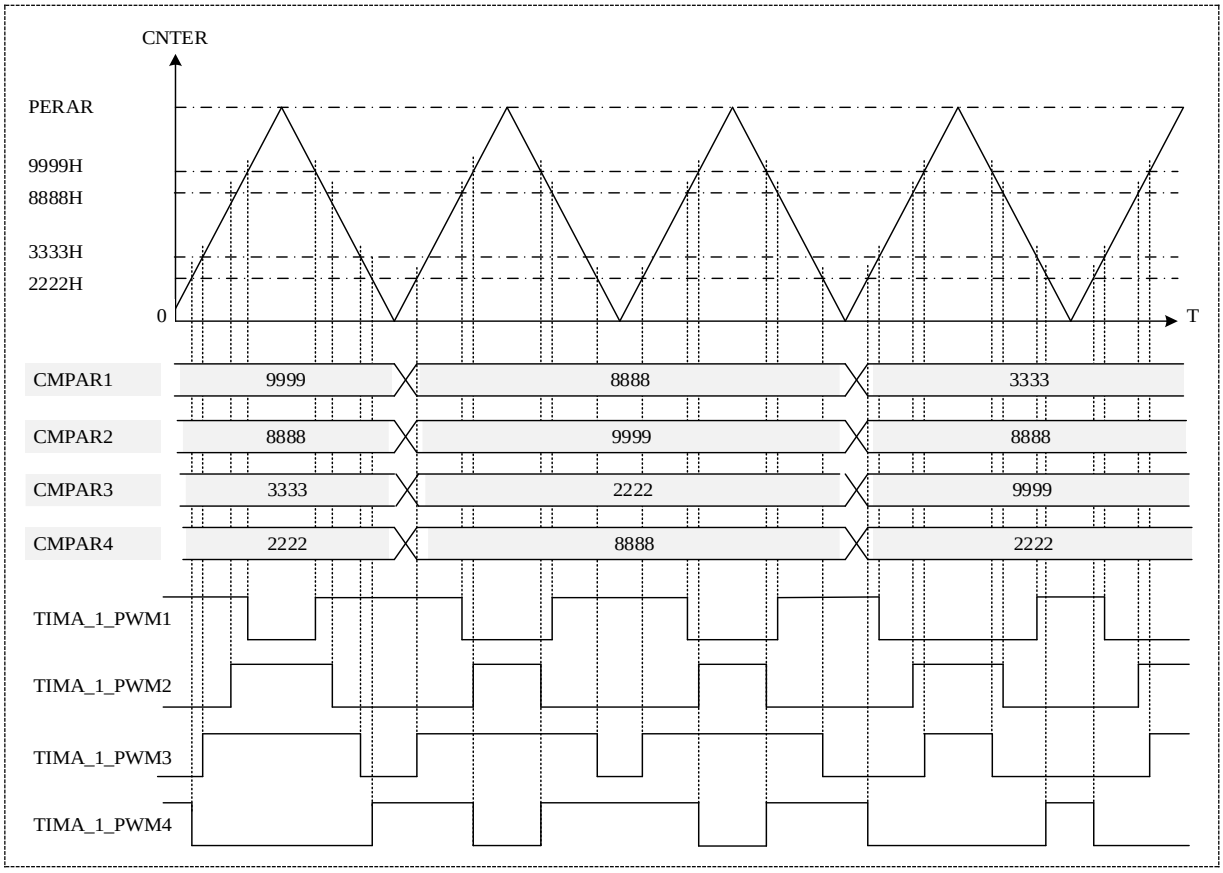


图 24-11　双边对称 PWM 输出例

24.3.10 正交编码计数

将 TIMA_<t>_CLKA 输入看作 AIN 输入、TIMA_<t>_CLKB 输入看作 BIN 输入、TIMA_<t>_TRIG 输入看作 ZIN 输入，TimerA 就可以实现三路输入的正交编码计数。每个单元的 AIN、BIN 单独动作可以实现位置计数模式；两个单元的 AIN、BIN、ZIN 组合动作可以实现公转计数模式，其中用于位置计数的单元称之为位置计数单元、用于公转计数的单元称之为公转计数单元。公转计数模式时，每两个单元间互相组合（单元 1、2 组合；单元 3、4 组合；单元 5、6 组合；单元 7、8 组合；单元 9、10 组合；单元 11、12 组合）,组合内位置计数单元和公转计数单元可以任意指定。

AIN 和 BIN 的计数条件使能通过设定硬件递加事件选择寄存器（HCUPR）和硬件递减事件选择寄存器（HCDOR）中 TIMA_<t>_CLKA 和 TIMA_<t>_CLKB 的正交关系来实现；ZIN 的输入动作通过设定位置计数单元的硬件触发事件选择寄存器（HCONR）

的清零使能位实现位置定时器清零、通过设定公转单元的硬件递加事件选择寄存器（HCUPR）实现公转定时器计数。

24.3.10.1 位置计数模式

正交编码位置计数模式，是指根据 AIN、BIN 的输入实现基本计数功能、相位差计数功能和方向计数功能。

基本计数

基本计数动作是根据 AIN 或 BIN 端口的输入时钟进行计数，如下图 24-12 所示。

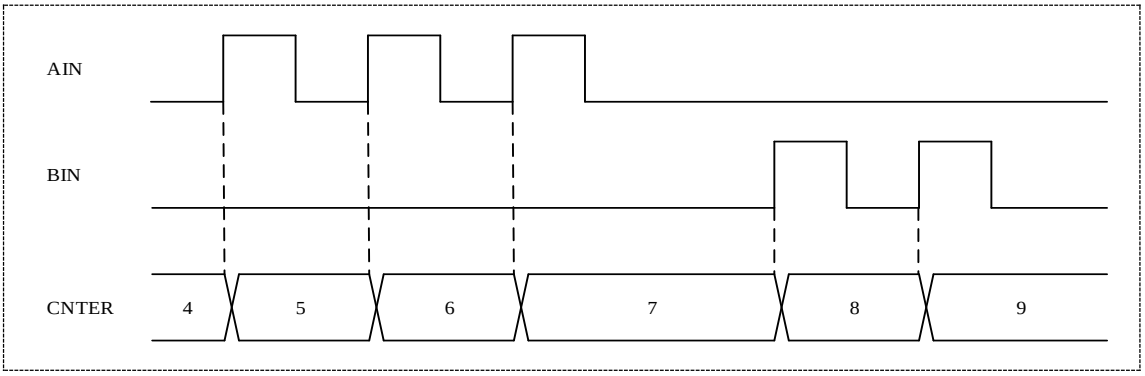


图 24-12 位置模式-基本计数

相位差计数

相位差计数是指根据 AIN 和 BIN 的相位关系进行计数。根据设定的不同，可以实现 1 倍计数、2 倍计数、4 倍计数等，如下图 24-13~图 24-15 所示。

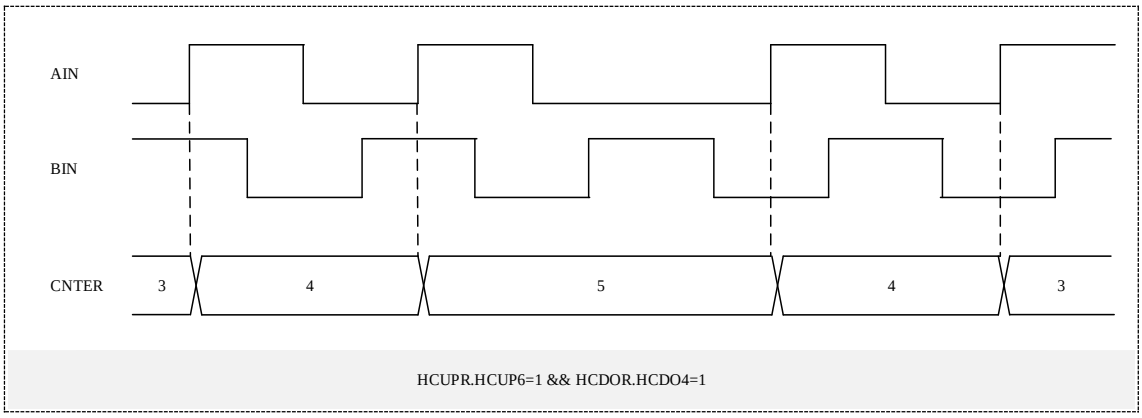


图 24-13 位置计数模式-相位差计数（1 倍计数）

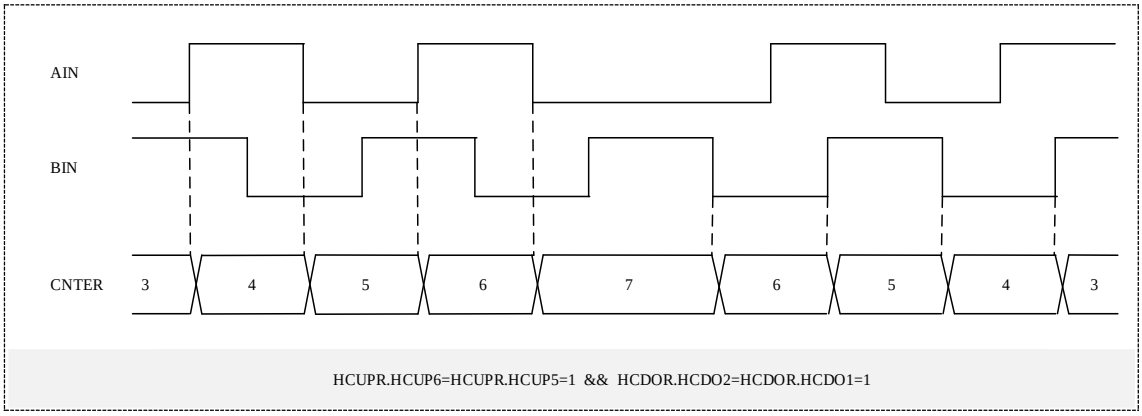


图 24-14 位置计数模式-相位差计数（2 倍计数）

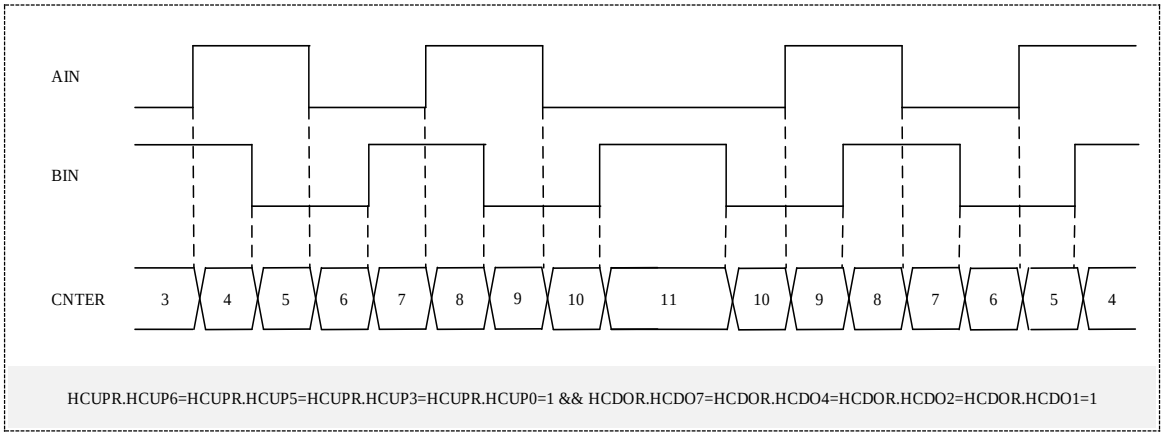


图 24-15 位置计数模式-相位差计数（4 倍计数）

方向计数

方向计数是指将 AIN 的输入状态设定为方向控制，将 BIN 的输入作为时钟计数，如下图 24-16 所示。

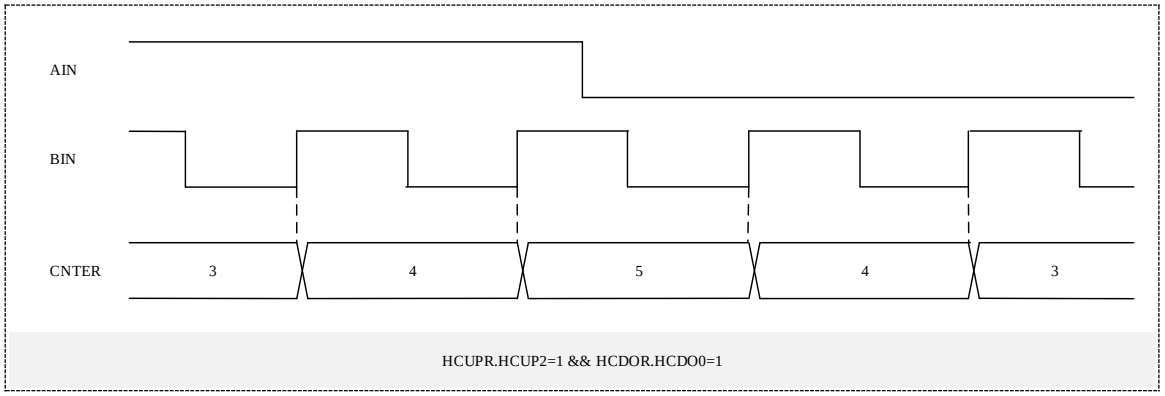


图 24-16 位置计数模式-方向计数

24.3.10.2 公转计数模式

正交编码公转计数模式，是指在 AIN、BIN 计数的基础上，加入 ZIN 的输入事件以实现对公转圈数等的判断。公转计数模式时根据公转定时器的计数方式，可实现 Z 相计

数功能、位置溢出计数功能和混合计数功能。

Z 相计数

Z 相计数是指根据 ZIN 的输入，公转计数单元进行计数，同时将位置计数单元清零的计数动作。如下图 24-17 所示。

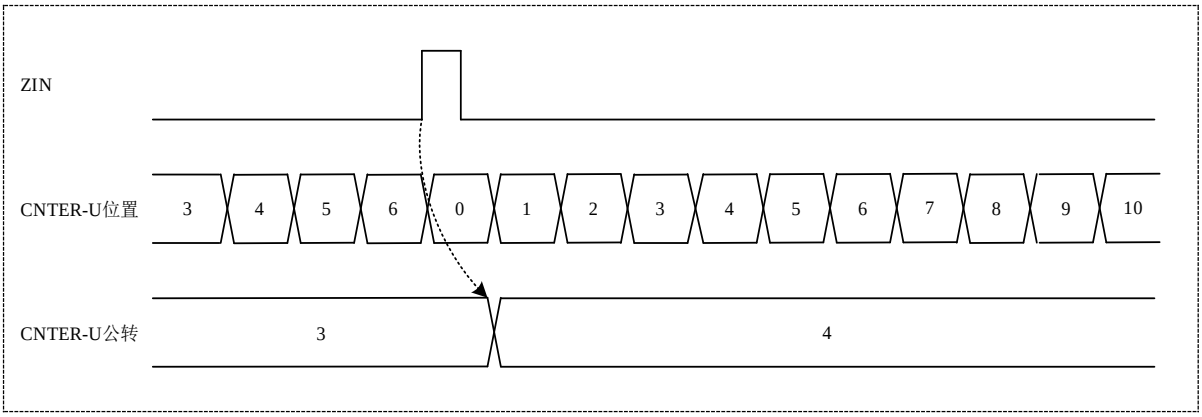


图 24-17　公转计数模式-Z 相计数

位置溢出计数

位置溢出计数是指位置计数单元计数发生上溢或下溢时，产生一个溢出事件，从而触发公转计数单元的定时器进行一次计数（在该计数方式时 ZIN 的输入不进行公转计数单元的计数动作和位置计数单元的清零动作）。

公转计数单元的硬件递加（递减）事件选择寄存器（HCUPR 或 HCDOR）的递加（递减）事件 bit12~11 位使能，位置计数单元的溢出事件就可以触发公转计数单元实现一次计数。如下图 24-18 所示。

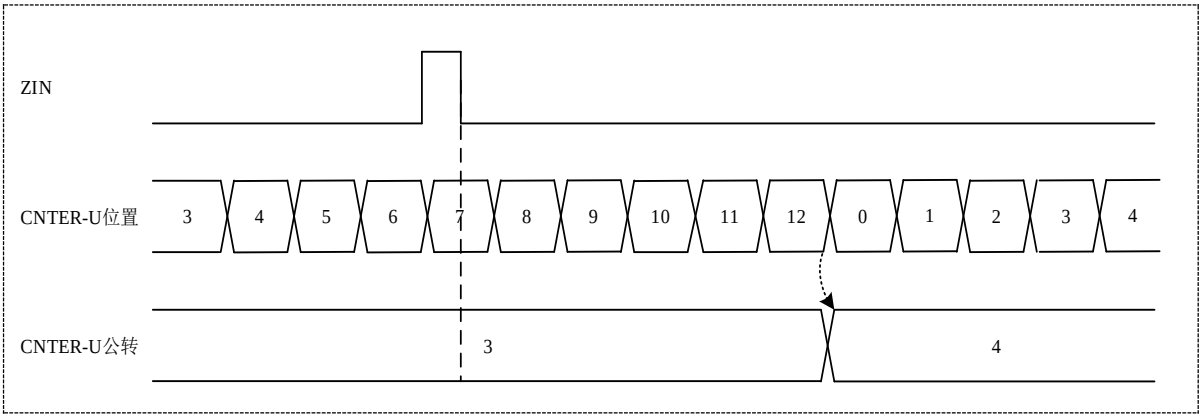


图 24-18　公转计数模式-位置溢出计数

混合计数

混合计数是指上述 Z 相计数和位置溢出计数两种计数方式合并起来的计数动作，其实现方式也是上述两种计数方式的组合。如下图 24-19 所示。

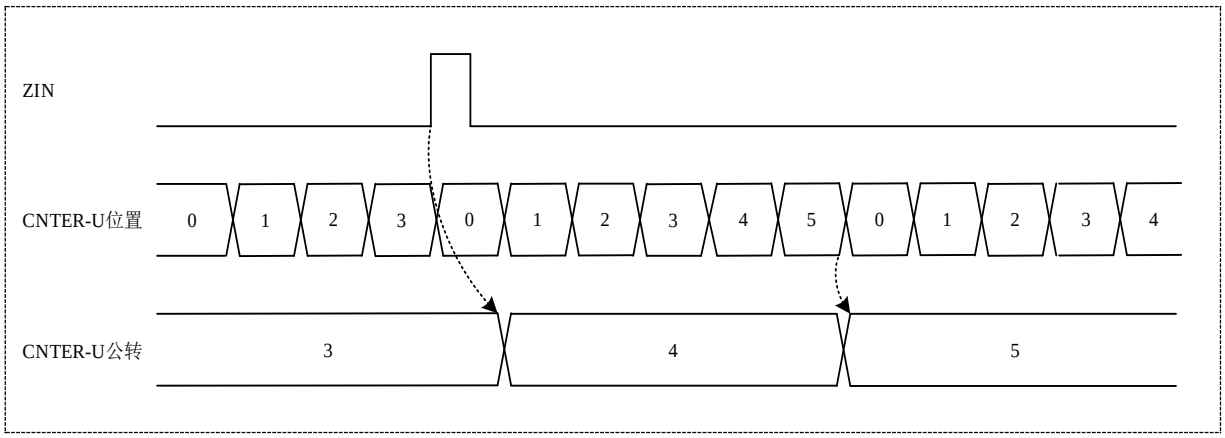


图 24-19　公转计数模式-混合计数

24.4 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。

24.4.1 比较匹配中断及事件

比较基准值寄存器（CMPARn）与计数值比较发生比较匹配时，状态标志寄存器（STFLR）中的对应位（STFLR.CMPFn）会被置为 1。此时，若中断控制寄存器（ICONR）的对应位（ICONR.ITENn）设定为 1，则对应的中断请求（TMRA_<t>_CMP）会被触发；若事件控制寄存器（ECONR）的对应位（ECONR.ETENn）设定为 1，则对应的事件请求（TMRA_<t>_CMP）会被触发（n=1~4）。

捕获控制寄存器（CCONRn）选择的捕获输入有效条件产生时，捕获输入动作发生。此时，若中断控制寄存器（ICONR）的对应位（ICONR.ITENn）设定为 1，则对应的中断请求（TMRA_<t>_CMP）被触发；若事件控制寄存器（ECONR）的对应位（ECONR.ETENn）设定为 1，则对应的事件请求（TMRA_<t>_CMP）会被触发（n=1~4）。

每个单元内部的 4 个基准值的比较匹配中断和比较匹配事件并非独立输出，比较匹配中断通过“或逻辑”汇总成一个中断输出至中断模块（参见 INTC 章节），比较匹配事件通过“或逻辑”汇总成一个事件输出用于选择触发别的模块。

24.4.2 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_<t>_OVF 和 TMRA_<t>_UDF）输出至中断模块（INTC）；周期匹配事件则无对应的使能位控制，在对应的计数周期点就触发周期匹配事件（TMRA_<t>_OVF 和 TMRA_<t>_UDF）输出用于选择触发别的模块。

24.5 寄存器说明

表 24-3 所示，为 TimerA 模块的寄存器列表。

BASE ADDR:

0x4003A000（U1）、0x4003A400（U2）、0x4003A800（U3）、
0x4003AC00（U4）、0x40026000（U5）、0x40026400（U6）、
0x40026800（U7）、0x40026C00（U8）、0x40027000（U9）、
0x40027400（U10）、0x40027800（U11）、0x40027C00（U12）

寄存器名	符号	偏移量	位宽	复位值
通用计数值寄存器	TMRA_CNTER	0x0000h	16	0x0000h
周期基准值寄存器	TMRA_PERAR	0x0004h	16	0xFFFFh
比较基准值寄存器1	TMRA_CMPAR1	0x0040h	16	0xFFFFh
比较基准值寄存器2	TMRA_CMPAR2	0x0044h	16	0xFFFFh
比较基准值寄存器3	TMRA_CMPAR3	0x0048h	16	0xFFFFh
比较基准值寄存器4	TMRA_CMPAR4	0x004Ch	16	0xFFFFh
控制状态寄存器	TMRA_BCSTR	0x0080h	16	0x0002h
中断控制寄存器	TMRA_ICONR	0x0090h	16	0x0000h
事件控制寄存器	TMRA_ECONR	0x0094h	16	0x0000h
滤波控制寄存器	TMRA_FCONR	0x0098h	16	0x0000h
状态标志寄存器	TMRA_STFLR	0x009Ch	16	0x0000h
缓存控制寄存器1	TMRA_BCONR1	0x00C0h	16	0x0000h
缓存控制寄存器2	TMRA_BCONR2	0x00C8h	16	0x0000h
捕获控制寄存器1	TMRA_CCONR1	0x0100h	16	0x0000h
捕获控制寄存器2	TMRA_CCONR2	0x0104h	16	0x0000h
捕获控制寄存器3	TMRA_CCONR3	0x0108h	16	0x0000h
捕获控制寄存器4	TMRA_CCONR4	0x010Ch	16	0x0000h
端口控制寄存器1	TMRA_PCONR1	0x0140h	16	0x0000h
端口控制寄存器2	TMRA_PCONR2	0x0144h	16	0x0000h

端口控制寄存器3	TMRA_PCONR3	0x0148h	16	0x0000h
端口控制寄存器4	TMRA_PCONR4	0x014Ch	16	0x0000h
硬件触发事件选择寄存器	TMRA_HCONR	0x0084h	16	0x0000h
硬件递加事件选择寄存器	TMRA_HCUPR	0x0088h	16	0x0000h
硬件递减事件选择寄存器	TMRA_HCDOR	0x008Ch	16	0x0000h
内部触发事件选择寄存器0	TMRA_HTSSR0	（0x40010880h）	32	0x000001FFh
内部触发事件选择寄存器1	TMRA_HTSSR1	（0x40010884h）	32	0x000001FFh
内部触发事件选择寄存器2	TMRA_HTSSR2	（0x40010888h）	32	0x000001FFh
内部触发事件选择寄存器3	TMRA_HTSSR3	（0x4001088Ch）	32	0x000001FFh

表 24-3 TimerA 寄存器列表

注意：

– 内部触发事件选择寄存器（TMRA_HTSSR0~3）是 4 个独立的寄存器，为 12 个单元的 TimerA 所共有。具体分配如下表所示。

单元数	事件分类	所对应的选择寄存器
U1	计数器触发事件： HCONR、HCUPR、HCDOR 所指的 HTSSR 选择事件	TMRA_HTSSR0
U2		TMRA_HTSSR1
U3		TMRA_HTSSR2
U4		TMRA_HTSSR3
U5		TMRA_HTSSR0
U6		TMRA_HTSSR1
U7		TMRA_HTSSR2
U8		TMRA_HTSSR3
U9		TMRA_HTSSR0
U10		TMRA_HTSSR1
U11		TMRA_HTSSR2
U12		TMRA_HTSSR3
U1	捕获动作触发事件： CCONR 中所指的 HTSSR 选择事件	TMRA_HTSSR1
U2		TMRA_HTSSR0

U3		TMRA_HTSSR3
U4		TMRA_HTSSR2
U5		TMRA_HTSSR1
U6		TMRA_HTSSR0
U7		TMRA_HTSSR3
U8		TMRA_HTSSR2
U9		TMRA_HTSSR1
U10		TMRA_HTSSR0
U11		TMRA_HTSSR3
U12		TMRA_HTSSR2

表 24-4 内部触发事件 HTSSR 选择关系对应表

24.5.1 通用计数值寄存器（TMRA_CNTER）

复位值: 0x0000h

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

24.5.2 周期基准值寄存器（TMRA_PERAR）

复位值: 0xFFFFh

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

24.5.3 比较基准值寄存器（TMRA_CMPARm）（m=1~4）

复位值: 0xFFFFh

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

24.5.4 控制状态寄存器（TMRA_BCSTR）

复位值: 0x0002h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UDFF	OVFF	ITENUDF	ITENOVF	-	-	-	OVSTP	CKDIV[3:0]				SYNST	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~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8	OVSTP	计数溢出停止控制	0：计数器在计数上溢或下溢后，继续计数 1：计数器在计数上溢或下溢后，停止计数	R/W
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：本单元为单元m时，与单元n的同步启动功能无效 1：本单元为单元m时，与单元n的同步启动功能有效 （当m=2、4、6、8、10、12时，n=1、3、5、7、9、11） 注：单元n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	R/W

	注2：本单元m的同步启动功能有效时，单元n软件启动后本单元的该位也会被置位（当m=2、4、6、8、10、12时，n=1、3、5、7、9、11）

24.5.5 中断控制寄存器（TMRA_ICONR）

复位值: 0x0000h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												ITEN4	ITEN3	ITEN2	ITEN1

位	标记	位名	功能	读写
b15~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	ITEN4	计数匹配中断使能4	0：CMPAR4寄存器与计数值相等时，或者发生捕获输入事件时，该中断无效 1：CMPAR4寄存器与计数值相等时，或者发生捕获输入事件时，该中断使能	R/W
b2	ITEN3	计数匹配中断使能3	0：CMPAR3寄存器与计数值相等时，或者发生捕获输入事件时，该中断无效 1：CMPAR3寄存器与计数值相等时，或者发生捕获输入事件时，该中断使能	R/W
b1	ITEN2	计数匹配中断使能2	0：CMPAR2寄存器与计数值相等时，或者发生捕获输入事件时，该中断无效 1：CMPAR2寄存器与计数值相等时，或者发生捕获输入事件时，该中断使能	R/W
b0	ITEN1	计数匹配中断使能1	0：CMPAR1寄存器与计数值相等时，或者发生捕获输入事件时，该中断无效 1：CMPAR1寄存器与计数值相等时，或者发生捕获输入事件时，该中断使能	R/W

24.5.6 事件控制寄存器（TMRA_ECONR）

复位值: 0x0000h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												ETEN4	ETEN3	ETEN2	ETEN1

位	标记	位名	功能	读写
b15~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	ETEN4	计数匹配事件使能4	0：CMPAR4寄存器与计数值相等时，或者发生捕获输入事件时，该事件输出无效 1：CMPAR4寄存器与计数值相等时，或者发生捕获输入事件时，该事件输出使能	R/W
b2	ETEN3	计数匹配事件使能3	0：CMPAR3寄存器与计数值相等时，或者发生捕获输入事件时，该事件输出无效 1：CMPAR3寄存器与计数值相等时，或者发生捕获输入事件时，该事件输出使能	R/W
b1	ETEN2	计数匹配事件使能2	0：CMPAR2寄存器与计数值相等时，或者发生捕获输入事件时，该事件输出无效 1：CMPAR2寄存器与计数值相等时，或者发生捕获输入事件时，该事件输出使能	R/W
b0	ETEN1	计数匹配事件使能1	0：CMPAR1寄存器与计数值相等时，或者发生捕获输入事件时，该事件输出无效 1：CMPAR1寄存器与计数值相等时，或者发生捕获输入事件时，该事件输出使能	R/W

24.5.7 滤波控制寄存器（TMRA_FCONR）

复位值: 0x0000h

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_<t>_CLKB端口输入滤波功能无效 1：TIMA_<t>_CLKB端口输入滤波功能使能	R/W
b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
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_<t>_CLKA端口输入滤波功能无效 1：TIMA_<t>_CLKA端口输入滤波功能使能	R/W
b7~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
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_<t>_TRIG输入端口滤波功能无效 1：TIMA_<t>_TRIG输入端口滤波功能使能	R/W

24.5.8 状态标志寄存器（TMRA_STFLR）

复位值: 0x0000h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												CMPF4	CMPF3	CMPF2	CMPF1

位	标记	位名	功能	读写
b15-b4	Reserved	-	读出时为“0”,写入时写“0”	R
b3	CMPF4	计数匹配标志4	0：CMPAR4寄存器的值与计数值不相等，且未发生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

24.5.9 缓存控制寄存器（TMRA_BCONRm）（m=1~2）

复位值: 0x0000h

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，n=1、3）	R/W
b1	BSE0	三角波缓存传送选择0	0：三角波计数模式计数到峰点时，缓存值不传送 1：三角波计数模式计数到峰点时，缓存值传送，即：CMMARm -> CMPARn（m=2、4，n=1、3）	R/W
b0	BEN	缓存使能	0：CMPARn基准值的缓存功能无效 1：CMPARn基准值的缓存功能有效 （n=1、3）	R/W

24.5.10 捕获控制寄存器（TMRA_CCONRm）（m=1~4）

复位值: 0x0000h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	NOFICKCP[1:0]		NOFIENCP	-	-	HICP4	HICP3	-	HICP2	HICP1	HICP0	-	-	-	CAPMD

位	标记	位名	功能	读写
b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14~b13	NOFICKCP[1:0]	滤波采样基准时钟选择CP	00：PCLK 01：PCLK/4 10：PCLK/16 11：PCLK/64	R/W
b12	NOFIENCP	捕获输入端口滤波CP	0：TIMA_<t>_PWMn端口输入滤波功能无效 1：TIMA_<t>_PWMn端口输入滤波功能使能 （n=1~4）	R/W
b11~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9	HICP4	捕获输入条件使能4	0：TIMA_<t>_TRIG端口输入采样到下降沿时，通道m+1不发生捕获输入动作 1：TIMA_<t>_TRIG端口输入采样到下降沿时，通道m+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_<t>_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_<t>_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_HTSSR寄存器中指定的事件发生时，不发生捕获输入动作 1：TMRA_HTSSR寄存器中指定的事件发生时，产生捕获输入动作 注：具体请参考寄存器说明章节的注意事项说明	R/W
b5	HICP1	捕获输入条件使能1	0：TIMA_<t>_PWMn端口输入采样到下降沿时，不发生捕获输入动作 1：TIMA_<t>_PWMn端口输入采样到下降沿时，产生捕获输入动作 （n=1~4）	R/W
b4	HICP0	捕获输入条件使能0	0：TIMA_<t>_PWMn端口输入采样到上升沿时，不发生捕获输入动作 1：TIMA_<t>_PWMn端口输入采样到上升沿时，产生捕获输入动作 （n=1~4）	R/W

b3~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	CAPMD	功能模式选择	0：比较输出功能 1：捕获输入功能	R/W

24.5.11 端口控制寄存器（TMRA_PCONRm）（m=1~4）

复位值: 0x0000h

b15	b14	b13	b12	b11	b10	b9 b8	b7 b6	b5 b4	b3 b2	b1 b0
-	-	-	OUTEN	-	-	FORC[1:0]	PERC[1:0]	CMPC[1:0]	STPC[1:0]	STAC[1:0]

位	标记	位名	功能	读写
b15~b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	OUTEN	输出使能	0：PWM输出功能时的TIMA_<t>_PWMn端口输出无效 1：PWM输出功能时的TIMA_<t>_PWMn端口输出有效 （n=1~4）	R/W
b11~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9~b8	FORC[1:0]	强制端口状态设定	0x：设定无效定 10：下周期开始，TIMA_<t>_PWMn端口输出设定为低电平 11：下周期开始，TIMA_<t>_PWMn端口输出设定为高电平 （n=1~4） 注1：下周期是指硬件计数模式或锯齿波计数到上溢点或下溢点、三角波计数到谷点 注2：该寄存器位可用于实现PWM输出占空比0%或100%的控制	R/W
b7~b6	PERC[1:0]	周期值匹配时端口状态设定	00：计数值与PERAR相等时，TIMA_<t>_PWMn端口输出设定为低电平 01：计数值与PERAR相等时，TIMA_<t>_PWMn端口输出设定为高电平 10：计数值与PERAR相等时，TIMA_<t>_PWMn端口输出保持先前状态 11：计数值与PERAR相等时，TIMA_<t>_PWMn端口输出设定为反转电平 （n=1~4）	R/W
b5~b4	CMPC[1:0]	比较值匹配时端口状态设定	00：计数值与CMPARn相等时，TIMA_<t>_PWMn端口输出设定为低电平 01：计数值与CMPARn相等时，TIMA_<t>_PWMn端口输出设定为高电平 10：计数值与CMPARn相等时，TIMA_<t>_PWMn端口输出保持先前状态 11：计数值与CMPARn相等时，TIMA_<t>_PWMn端口输出设定为反转电平 （n=1~4）	R/W
b3~b2	STPC[1:0]	计数停止时端口状态设定	00：计数停止时，TIMA_<t>_PWMn端口输出设定为低电平 01：计数停止时，TIMA_<t>_PWMn端口输出设定为高电平 10：计数停止时，TIMA_<t>_PWMn端口输出保持先前状态 11：计数停止时，TIMA_<t>_PWMn端口输出保持先前状态	R/W

			（n=1~4）	
b1~b0	STAC[1:0]	计数开始时端口状态设定	00：计数开始时，TIMA_<t>_PWMn端口输出设定为低电平 01：计数开始时，TIMA_<t>_PWMn端口输出设定为高电平 10：计数开始时，TIMA_<t>_PWMn端口输出保持先前状态 11：计数开始时，TIMA_<t>_PWMn端口输出保持先前状态 （n=1~4） 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	R/W

24.5.12 硬件触发事件选择寄存器（TMRA_HCONR）

复位值: 0x0000h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HCLE6	HCLE5	HCLE4	HCLE3	-	HCLE2	HCLE1	HCLE0	-	HSTP2	HSTP1	HSTP0	-	HSTA2	HSTA1	HSTA0

位	标记	位名	功能	读写
b15	HCLE6	硬件清零条件6	条件：TIMA_<t>_PWM3端口输入采样到下降沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b14	HCLE5	硬件清零条件5	条件：TIMA_<t>_PWM3端口输入采样到上升沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b13	HCLE4	硬件清零条件4	条件：本单元为单元m时，单元n的TRIG端口输入采样到下降沿 （当m=1、3、5、7、9、11时，n=2、4、6、8、10、12；当m=2、4、6、8、10、12时，n=1、3、5、7、9、11） 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b12	HCLE3	硬件清零条件3	条件：本单元为单元m时，单元n的TRIG端口输入采样到上升沿 （当m=1、3、5、7、9、11时，n=2、4、6、8、10、12；当m=2、4、6、8、10、12时，n=1、3、5、7、9、11） 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10	HCLE2	硬件清零条件2	条件：TMRA_HTSSR寄存器中指定的事件发生 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效 注：具体请参考寄存器说明章节的注意事项说明	R/W
b9	HCLE1	硬件清零条件1	条件：TIMA_TRIG端口输入采样到下降沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b8	HCLE0	硬件清零条件0	条件：TIMA_TRIG端口输入采样到上升沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6	HSTP2	硬件停止条件2	条件：TMRA_HTSSR寄存器中指定的事件发生 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效 注：具体请参考寄存器说明章节的注意事项说明	R/W
b5	HSTP1	硬件停止条件1	条件：TIMA_<t>_TRIG端口输入采样到下降沿	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	
b4	HSTP0	硬件停止条件0	条件：TIMA_<t>_TRIG端口输入采样到上升沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	HSTA2	硬件启动条件2	条件：TMRA_HTSSR寄存器中指定的事件发生 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效 注：具体请参考寄存器说明章节的注意事项说明	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_n_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）中，当本单元为2、4、6、8、10、12时，n=1、3、5、7、9、11；当本单元为1、3、5、7、9、11时，该功能无效	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_n_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）中，当本单元为2、4、6、8、10、12时，n=1、3、5、7、9、11；当本单元为1、3、5、7、9、11时，该功能无效	R/W

24.5.13 硬件递加事件选择寄存器（TMRA_HCUPR）

复位值: 0x0000h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	HCUP12	HCUP11	HCUP10	HCUP9	HCUP8	HCUP7	HCUP6	HCUP5	HCUP4	HCUP3	HCUP2	HCUP1	HCUP0

位	标记	位名	功能	读写
b15~b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	HCUP12	硬件递加条件12	条件：本单元为单元m时，单元n发生计数下溢（当m=1、3、5、7、9、11时，n=2、4、6、8、10、12；当m=2、4、6、8、10、12时，n=1、3、5、7、9、11） 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b11	HCUP11	硬件递加条件11	条件：本单元为单元m时，单元n发生计数上溢（当m=1、3、5、7、9、11时，n=2、4、6、8、10、12；当m=2、4、6、8、10、12时，n=1、3、5、7、9、11） 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b10	HCUP10	硬件递加条件10	条件：TMRA_HTSSR寄存器中指定的事件发生 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效 注：具体请参考寄存器说明章节的注意事项说明	R/W
b9	HCUP9	硬件递加条件9	条件：TIMA_<t>_TRIG端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b8	HCUP8	硬件递加条件8	条件：TIMA_<t>_TRIG端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b7	HCUP7	硬件递加条件7	条件：TIMA_<t>_CLKB端口为高电平时，TIMA_<t>_CLKA端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b6	HCUP6	硬件递加条件6	条件：TIMA_<t>_CLKB端口为高电平时，TIMA_<t>_CLKA端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b5	HCUP5	硬件递加条件5	条件：TIMA_<t>_CLKB端口为低电平时，TIMA_<t>_CLKA端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W

b4	HCUP4	硬件递加条件4	条件：TIMA_<t>_CLKB端口为低电平时，TIMA_<t>_CLKA端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b3	HCUP3	硬件递加条件3	条件：TIMA_<t>_CLKA端口为高电平时，TIMA_<t>_CLKB端口上采样到下降沿 0：条件匹配时，硬件递加无效 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_<t>_CLKA端口为高电平时，TIMA_<t>_CLKB端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b1	HCUP1	硬件递加条件1	条件：TIMA_<t>_CLKA端口为低电平时，TIMA_<t>_CLKB端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W
b0	HCUP0	硬件递加条件0	条件：TIMA_<t>_CLKA端口为低电平时，TIMA_<t>_CLKB端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效	R/W

24.5.14 硬件递减事件选择寄存器（TMRA_HCDOR）

复位值: 0x0000h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	HCDO12	HCDO11	HCDO10	HCDO9	HCDO8	HCDO7	HCDO6	HCDO5	HCDO4	HCDO3	HCDO2	HCDO1	HCDO0

位	标记	位名	功能	读写
b15~b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	HCDO12	硬件递减条件12	条件：本单元为单元m时，单元n发生计数下溢（当m=1、3、5、7、9、11时，n=2、4、6、8、10、12；当m=2、4、6、8、10、12时，n=1、3、5、7、9、11） 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b11	HCDO11	硬件递减条件11	条件：本单元为单元m时，单元n发生计数上溢（当m=1、3、5、7、9、11时，n=2、4、6、8、10、12；当m=2、4、6、8、10、12时，n=1、3、5、7、9、11） 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b10	HCDO10	硬件递减条件10	条件：TMRA_HTSSR寄存器中指定的事件发生 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效 注：具体请参考寄存器说明章节的注意事项说明	R/W
b9	HCDO9	硬件递减条件9	条件：TIMA_<t>_TRIG端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b8	HCDO8	硬件递减条件8	条件：TIMA_<t>_TRIG端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b7	HCDO7	硬件递减条件7	条件：TIMA_<t>_CLKB端口为高电平时，TIMA_<t>_CLKA端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b6	HCDO6	硬件递减条件6	条件：TIMA_<t>_CLKB端口为高电平时，TIMA_<t>_CLKA端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b5	HCDO5	硬件递减条件5	条件：TIMA_<t>_CLKB端口为低电平时，TIMA_<t>_CLKA端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W

b4	HCDO4	硬件递减条件4	条件：TIMA_<t>_CLKB端口为低电平时，TIMA_<t>_CLKA端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效	R/W
b3	HCDO3	硬件递减条件3	条件：TIMA_<t>_CLKA端口为高电平时，TIMA_<t>_CLKB端口上采样到下降沿 0：条件匹配时，硬件递减无效 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

24.5.15 内部触发事件选择寄存器（TMRA_HTSSRm）（m=0~3）

复位值: 0x000001FFh

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1	0：禁止 AOS_COMTRG1 的公共触发事件触发 TimerA 1：允许 AOS_COMTRG1 的公共触发事件触发 TimerA 详见自动运行系统（AOS）章节	R/W
b30	COMEN[0]	公共触发使能0	0：禁止 AOS_COMTRG2 的公共触发事件触发 TimerA 1：允许 AOS_COMTRG2 的公共触发事件触发 TimerA 详见自动运行系统（AOS）章节	R/W
b29~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8~b0	TRGSEL	触发源选择	触发源编号写入，参考INTC章节	R/W

25 通用定时器（Timer2）

25.1 简介

通用定时器 2（Timer2）是一个可以实现同步计数、异步计数方式的基本定时器。该定时器内含 2 个通道（CH-A 和 CH-B）。每个通道均有一个输出端口，可实现基本的方波输出；每个通道均有 2 个输入端口，一个是时钟输入端口，可实现端口异步计数；一个是触发输入端口，可实现定时器启动、停止、清零、计数动作及计数值捕获输入；支持脉宽测量和周期测量。本系列产品中搭载 4 个单元的 Timer2。

25.2 基本框图

Timer2 基本的功能及特性如下：

基本动作	● 硬件触发计数器启动、停止、清零、计数及捕获输入
	● 同步计数、异步计数
	● 脉宽测量
	● 周期测量
	● 方波输出
	● 事件信号输出
中断类型	● 计数比较匹配中断
	● 计数溢出中断

表 25-1 Timer2 的基本功能及特性

Timer2 的基本框图如图 25-1 所示。框图中所示”<t>”表示单元编号，即”<t>”为 1~4，本章节后文提到“<t>”时均指单元编号，不再赘述。框图中 TIM2_<t>_CLKA/B 端口输入上的 ANF 是一个模拟滤波单元，滤波宽度为 40ns（典型条件）。

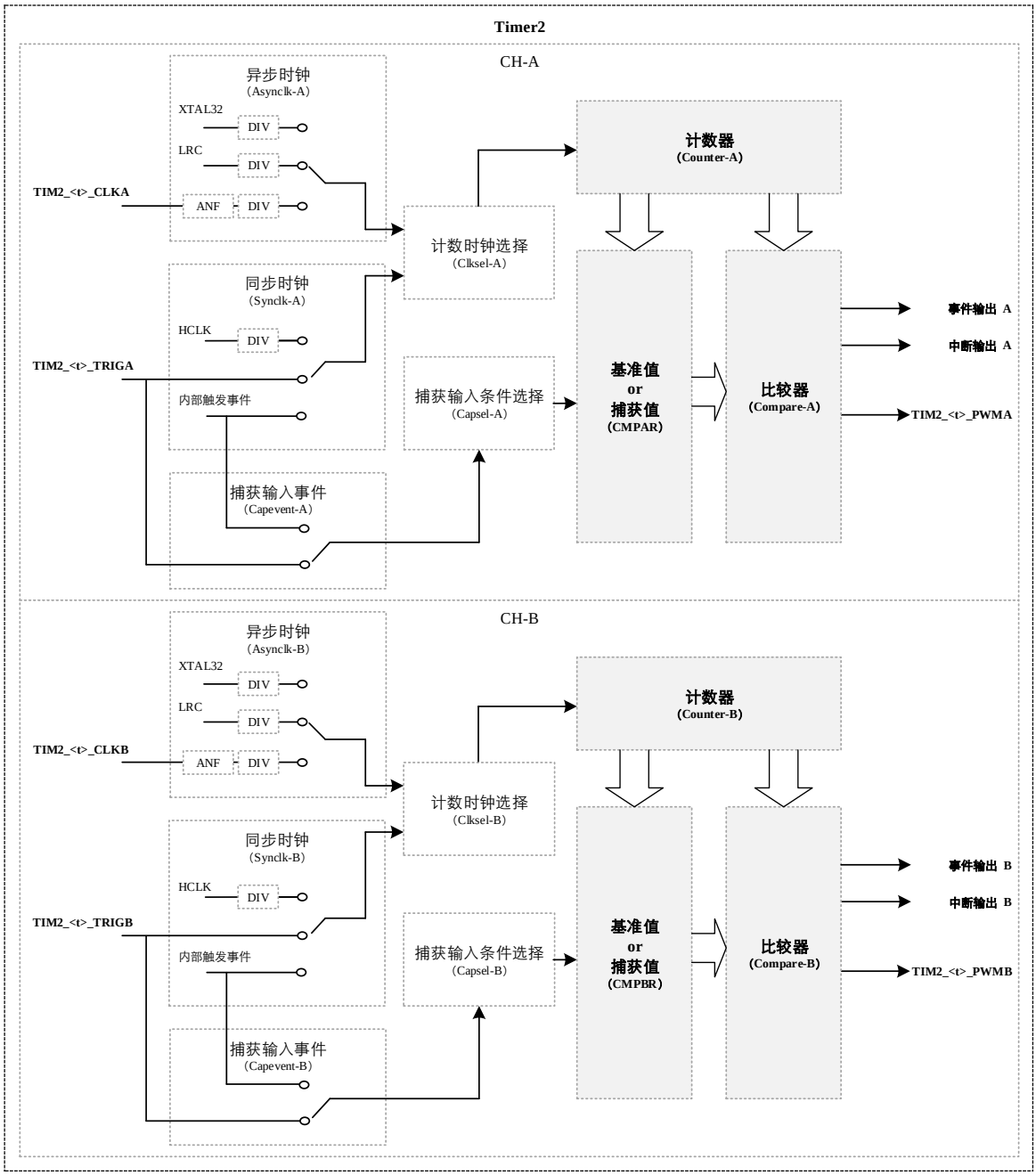


图 25-1 Timer2 基本框图

表 25-2 所示为 Timer2 的输入输出端口列表。

端口名	方向	功能
TIM2_<t>_CLKA/B	in	异步时钟输入端口
TIM2_<t>_TRIGA/B	in	1）硬件启动、停止、清零、计数输入端口 2）捕获输入端口
TIM2_<t>_PWMA/B	out	方波输出端口

表 25-2 Timer2 端口列表

25.3 功能说明

25.3.1 时钟源选择

Timer2 的计数方式可以选择同步计数方式或异步计数方式。

同步计数方式是指定时器的计数时钟和总线访问时钟（寄存器读写操作时钟）有同步时序关系；异步计数方式是指定时器的计数时钟和总线访问时钟（寄存器读写操作时钟）是非同步的时序关系。异步计数方式时对寄存器读操作时，定时器等的状态可能正在发生变化、读出不可预期的状态。因此，在异步计数方式时，寄存器读操作须在计数停止状态下实现。

25.3.1.1 同步计数时钟源

同步计数方式时（BCONR.SYNSA<B>=0），时钟源可以有以下几种选择（BCONR.SYNCLK A<B> [1:0]或 BCONR.SYNCLK A<B> T[1:0]设定）：

a) PCLK1 及 PCLK1 的 2、4、8、16、32、64、128、256、512、1024 分频作为同步计数时钟（BCONR.SYNCLK A<B> [1:0]=00 & BCONR.CKDIV A<B> [3:0]设定）

b) TIM2_<t>_TRIGA、TIM2_<t>_TRIGB 端口的输入有效选择沿被内部 PCLK1 采样后作为同步计数时钟（BCONR.SYNCLK A<B> [1:0]=01 或 10）

c) 内部硬件触发事件输入作为同步计数时钟（BCONR.SYNCLK A<B> [1:0]=11）

d) Timer6 的计数上溢事件（BCONR.SYNCLKA<B>T[0]=1）

e) Timer6 的计数下溢事件（BCONR.SYNCLKA<B>T[1]=1）

计数时钟源选择 a 时为软件计数模式，计数时钟源选择 b、c、d、e 时为硬件计数模式。

上述描述可以看到，b、c、d、e 时钟互相独立，可分别设定有效或无效，并且当选择 b、c、d、e 时钟时，a 时钟自动无效。选择 d、e 时钟源时，请设定 PCLK0 和 PCLK1 同频。

25.3.1.2 异步计数时钟源

异步计数方式时（BCONR.SYNS A<B>=1），时钟源可以有以下几种选择（BCONR.ASYNCLK A<B> [1:0]设定选）：

a) LRC 时钟源输入及其 2、4、8、16、32、64、128、256、512、1024 分频作为异步

计数时钟（BCONR.ASYNCLK A<B>=00 & BCONR.CKDIV A<B> [3:0]设定）

b) XTAL32 时钟源输入及其 2、4、8、16、32、64、128、256、512、1024 分频作为异步计数时钟（BCONR.ASYNCLK A<B>=01 & BCONR.CKDIV A<B> [3:0]设定）

c) TIM2_<t>_ CLKA、TIM2_<t>_ CLKB 时钟输入端口及其 2、4、8、16、32、64、128、256、512、1024 分频作为异步计数时钟（BCONR.ASYNCLK A<B>=10 & BCONR.CKDIV A<B> [3:0]设定）

25.3.2 比较输出

Timer2 有 2 个输出端口（TIM2_<t>_PWMA、TIM2_<t>_PWMB），可以实现基本的方波输出。

CMPAR、CMPBR 寄存器对应了 TIM2_<t>_PWMA、TIM2_<t>_PWMB 的计数比较基准值。当定时器的计数值 CNTAR 和 CMPAR 相等时，TIM2_<t>_PWMA 端口输出指定的电平；当定时器的计数值 CNTBR 和 CMPBR 相等时，TIM2_<t>_PWMB 端口输出指定的电平。

TIM2_<t>_PWMA/B 端口的计数起始电平、停止电平、计数比较匹配时的电平等，可由端口控制寄存器（PCONR）的 PCONR.STACA<B>[1:0]、PCONR.STPCA<B>[1:0]、PCONR.CMPCA<B>[1:0]位设定。图 25-2 为比较输出动作例。

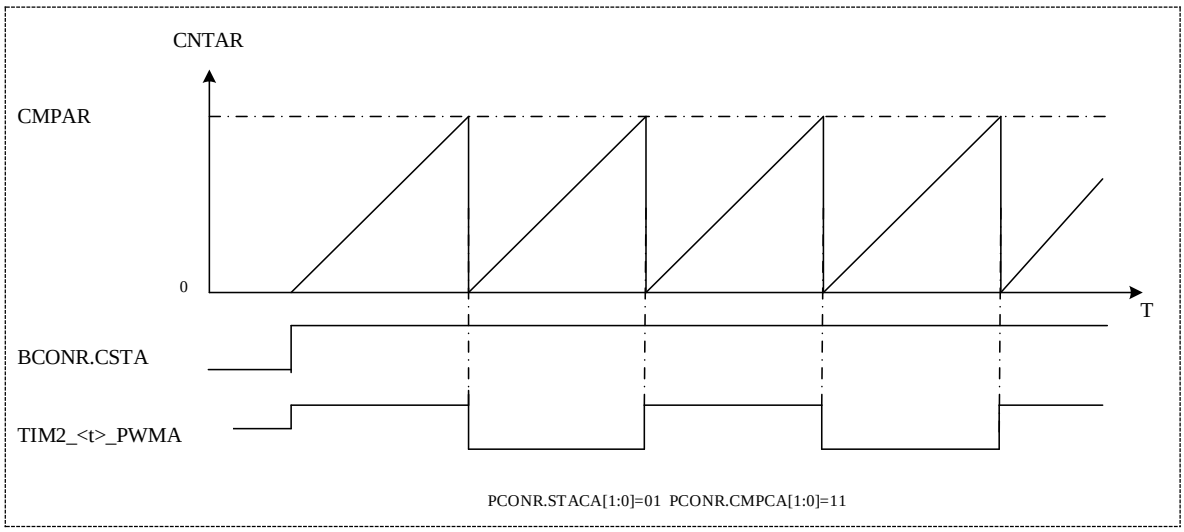


图 25-2 比较输出动作

25.3.3 硬件触发

Timer2 的硬件触发动作有两类，一个是 TIM2_<t>_TRIGA/B 端口的输入事件，另一个

是内部硬件触发事件。这两类事件均可以实现计数器的启动、停止、清零及硬件捕获输入功能。各种功能组合设定也可实现脉宽测量、周期测量等。

TIM2_<t>_TRIGA/B 端口输入类型选择（上升沿或下降沿）及功能选择（启动、停止、清零、捕获输入）、内部硬件触发事件的功能选择（启动、停止、清零、捕获输入）可通过硬件控制寄存器（HCONR）的相应位设定。

Timer2 有一个内部硬件触发源，事件源可通过触发选择寄存器（HTSSR）中对应的编号设定来选择，具体的事件对应关系请参考 INTC 章节。使用内部硬件触发功能时，需要先将功能时钟控制寄存器（PWR_FCG0）的外围电路触发功能使能位置 1。

25.3.3.1 启动停止清零

每个通道的硬件启动、停止、清零条件由硬件控制寄存器（HCONR）的相关设定来决定。图 25-3 是硬件启动、清零的动作例。

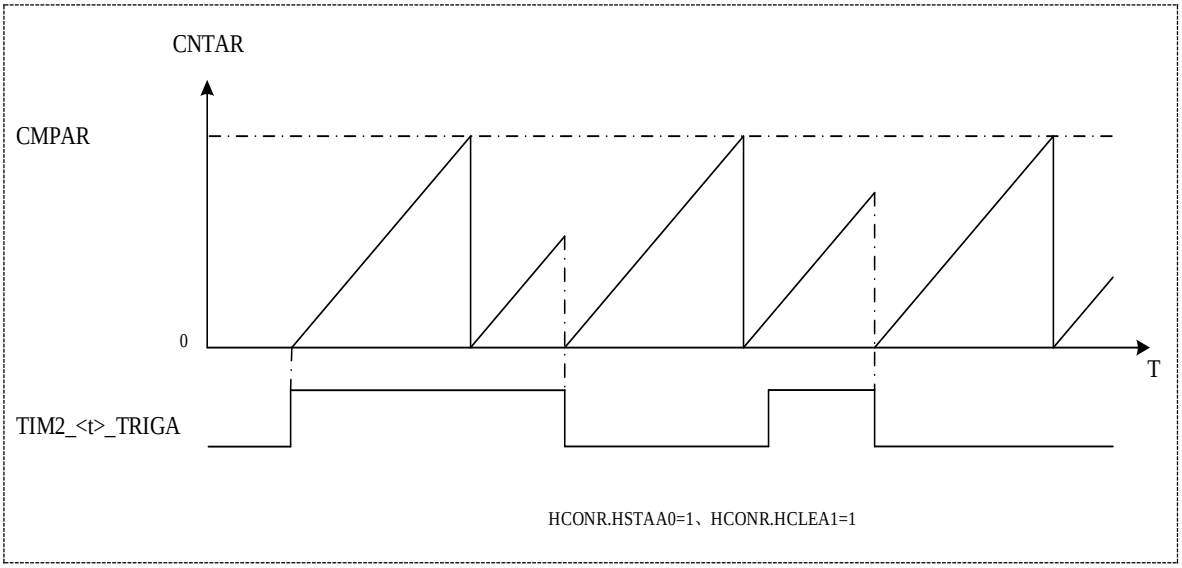


图 25-3　硬件启动、清零动作

25.3.3.2 捕获输入动作

TIM2_<t>_TRIGA/B 端口输入或内部触发事件作为捕获输入条件时，可实现捕获输入功能。当硬件控制寄存器（HCONR）选择的捕获输入条件有效时，当前的计数值就被保存到相应的寄存器（CMPAR、CMPBR）中。图 25-4 为捕获输入的动作例。

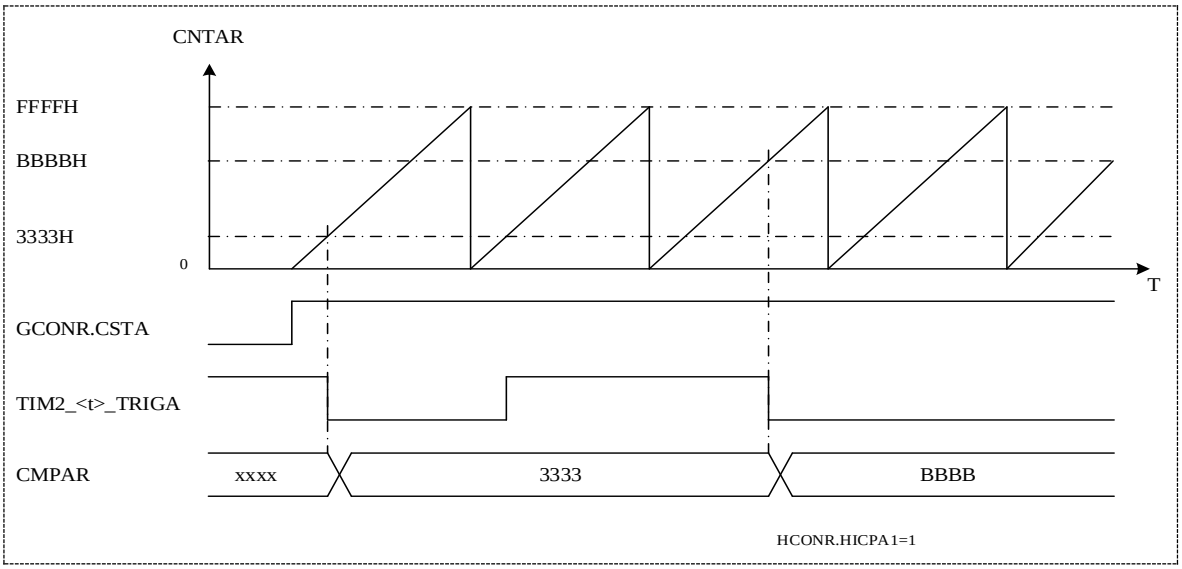


图 25-4 捕获输入动作

25.3.3.3 脉宽测量

将 A 通道的硬件启动条件设为 TIM2_<t>_TRIGA 的上升沿，硬件清零条件、停止条件和捕获输入条件均设为 TIM2_<t>_TRIGA 的下降沿，就可以实现连续的脉冲宽度测量。B 通道可实现同样功能。对应动作如图 25-5 所示。

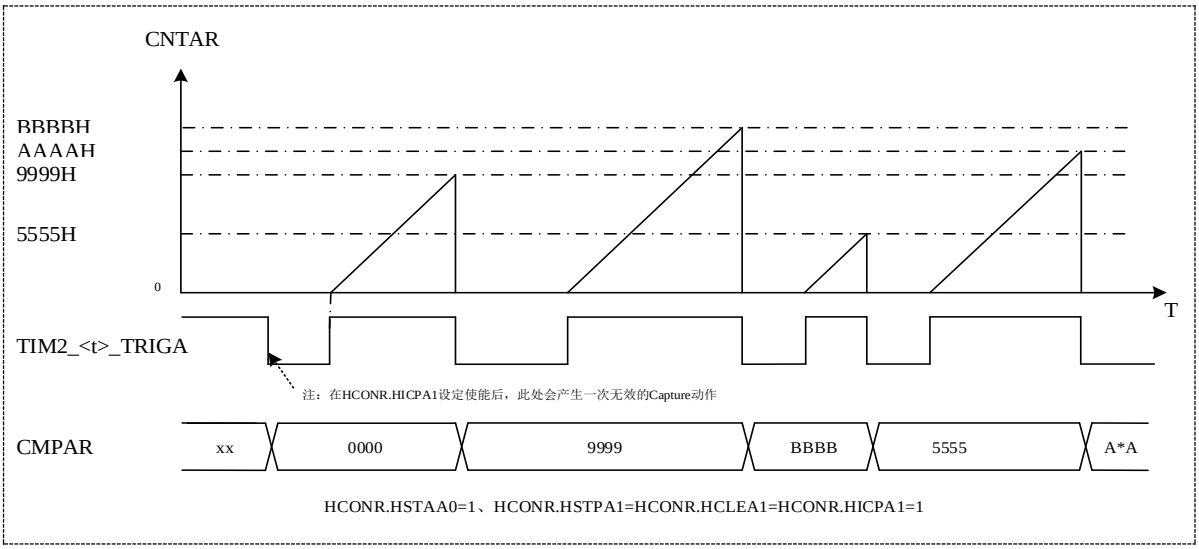


图 25-5 脉宽测量

25.3.3.4 周期测量

将 A 通道的硬件启动条件、硬件清零条件和捕获输入条件设为 TIM2_<t>_TRIGA 的相同沿（上升沿或下降沿），就可以实现连续的周期宽度测量。B 通道可实现同样功能。对应动作如图 25-6 所示。

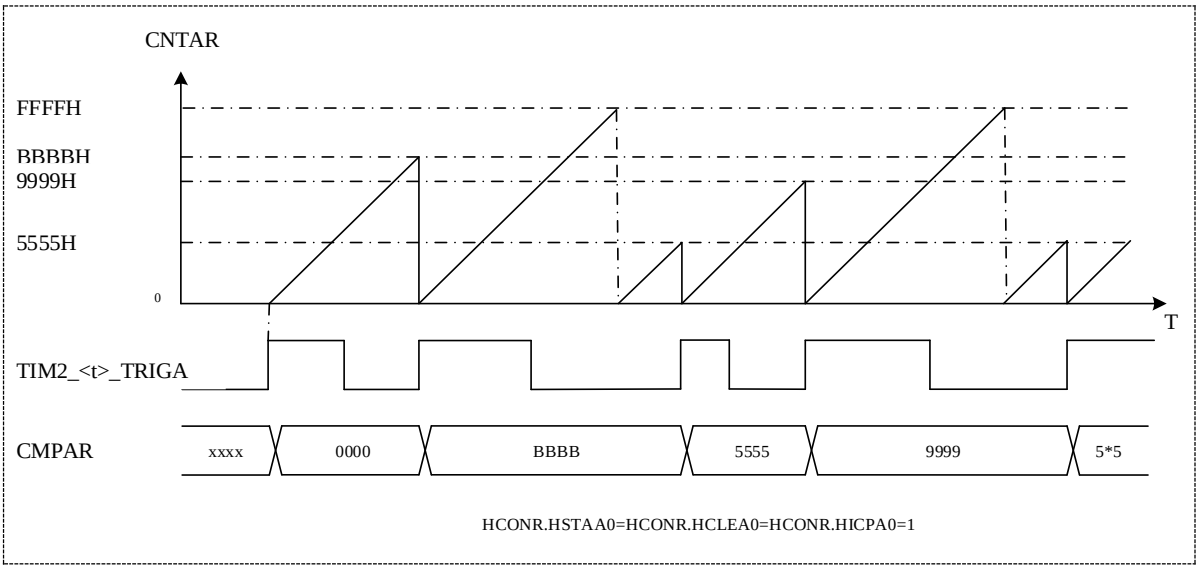


图 25-6 周期测量

25.3.4 数字滤波

Timer2 的 TRIG 输入端口（TIM2_<t>_TRIGA/B）具有数字滤波功能。可通过设定端口控制寄存器（PCONR.NOFIENA<B>）的使能位开启对应端口的滤波功能。滤波用的基准时钟也通过端口控制寄存器（PCONR.NOFICKA<B>[1:0]）设定。

在滤波采样基准时钟采样到端口上 3 次一致的电平时，该电平被当作有效电平传送到模块内部；小于 3 次一致的电平会被当作外部干扰滤掉，不传送到模块内部。其动作例如图 25-7 所示。

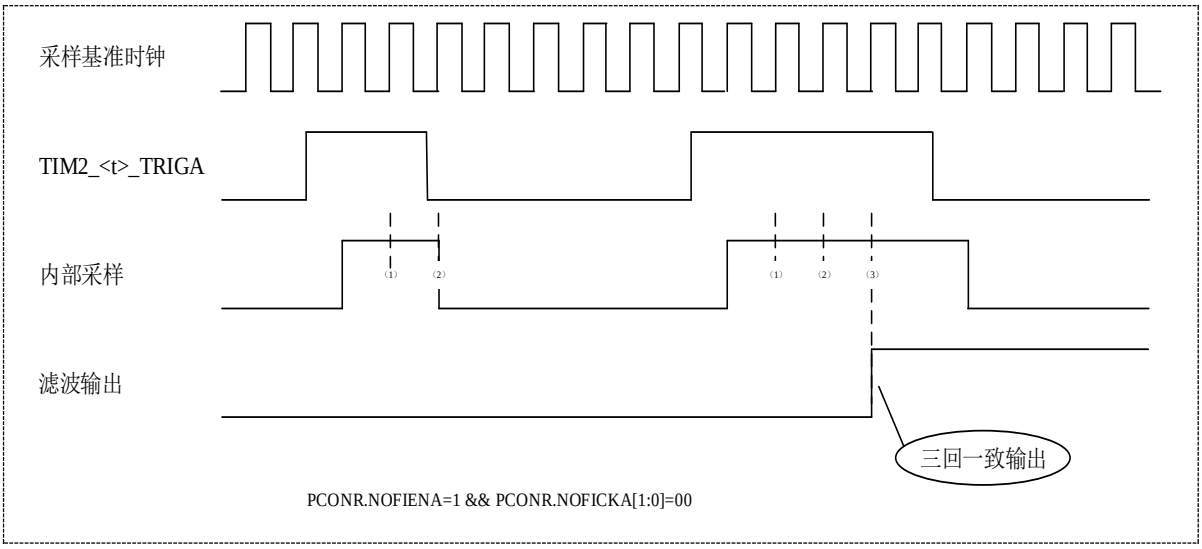


图 25-7 TRIG 输入端口的数字滤波

25.4 中断及事件说明

25.4.1 中断输出

一个 Timer2 含有 4 个中断，分别是通道 A 和通道 B 的计数比较匹配中断或捕获输入中断、通道 A 和通道 B 的计数溢出中断。

基准值寄存器（CMPAR、CMPBR）共计 2 个，可分别与计数值寄存器（CNTAR、CNTBR）比较产生比较匹配有效信号。计数比较匹配时，状态标志寄存器（STFLR）中的 STFLR.CMFA<B>位分别会被置为 1。此时若设定中断控制寄存器（ICONR）的 ICONR.CMENA<B>位使能中断，则对应的中断请求（TMR2_<t>_CMPm，m=A、B）也会被触发。

在硬件控制寄存器（HCONR）选择的捕获输入有效条件产生时，可以产生相应的捕获输入动作。此时若设定中断控制寄存器（ICONR）的 ICONR.CMENA<B>位使能中断，则对应的中断请求（TMR2_<t>_CMPm，m=A、B）被触发。

计数值寄存器（CNTAR、CNTBR）计数到 0XFFFF 时发生计数溢出事件，此时状态标志寄存器（STFLR）中的 STFLR.OVFA<B>位会被置为 1。若设定中断控制寄存器（ICONR）的 ICONR.OVENA<B>位使能中断，则对应的中断请求（TMR2_<t>_OVFm，m=A、B）也会被触发。

在选择异步计数模式（BCONR.ASYNCLKA[1:0]=2’b10 除外）时，单元 1 的基准值寄存器（CMPAR）产生的比较匹配中断可用于在低功耗模式时对系统进行唤醒，具体请参考 INTC 章节。

25.4.2 事件输出

一个 Timer2 含有 4 个事件，分别是通道 A 和通道 B 的计数比较匹配事件或捕获输入事件、通道 A 和通道 B 的计数溢出事件。

在计数过程中发生计数比较匹配或捕获输入动作或计数溢出时，会产生相应的事件请求（TMR2_<t>_CMPm 或 TMR2_<t>_OVFm，m=A、B）输出信号，可以用于选择触发其它模块。

25.5 寄存器说明

表 25-3 所示，为 Timer2 模块的寄存器列表。

BASE ADDR:

0x40024800（U1）、0x40024C00（U2）、0x40025000（U3）、0x40025400（U4）

寄存器名	符号	偏移量	位宽	复位值
计数值寄存器	TMR2_CNTAR	0x0000h	32	0x00000000h
计数值寄存器	TMR2_CNTBR	0x0004h	32	0x00000000h
基准值寄存器	TMR2_CMPAR	0x0008h	32	0x0000FFFFh
基准值寄存器	TMR2_CMPBR	0x000Ch	32	0x0000FFFFh
基本控制寄存器	TMR2_BCONR	0x0010h	32	0x00000000h
中断控制寄存器	TMR2_ICONR	0x0014h	32	0x00000000h
端口控制寄存器	TMR2_PCONR	0x0018h	32	0x00000000h
硬件控制寄存器	TMR2_HCONR	0x001ch	32	0x00000000h
触发选择寄存器	TMR2_HTSSR	（0x40010874h）	32	0x000001FFh
状态标志寄存器	TMR2_STFLR	0x0020h	32	0x00000000h

表 25-3 Timer2 寄存器列表

注意：

– 触发选择寄存器（TMR2_HTSSR）是一个独立的寄存器，为 4 个单元的 Timer2 所共有。

25.5.1 计数值寄存器（TMR2_CNTmR）（m=A、B）

复位值: 0x00000000h

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”	R
b15~b0	CNTA<B>[15:0]	计数值	当前定时器的计数值	R/W

25.5.2 基准值寄存器（TMR2_CMPmR）（m=A、B）

复位值: 0x0000FFFFh

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”	R
b15~b0	CMPA<B>[15:0]	基准值	设定计数基准值，产生Compare Match事件	R/W

25.5.3 基本控制寄存器（TMR2_BCONR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	SYNCLKBT[1:0]		ASYNCLKB[1:0]		SYNCLKB[1:0]		CKDIVB[3:0]				SYNSB	-	CAPMDB	CSTB
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	SYNCLKAT[1:0]		ASYNCLKA[1:0]		SYNCLKA[1:0]		CKDIVA[3:0]				SYNSA	-	CAPMDA	CSTA

位	标记	位名	功能	读写
b31~b30	Reserved	-	读出时为“0”,写入时写“0”	R/W
b29	SYNCLKBT[1]	同步计数同时钟源选择BT	条件：Timer6 的单元 n 发生计数下溢 0：条件匹配时，Timer2 单元 m 同步计数无效 1：条件匹配时，Timer2单元m进行一次同步计数 （当 m=1、2、3、4 时，n=2、4、6、8）	R/W
b28	SYNCLKBT[0]	同步计数同时钟源选择BT	条件：Timer6 的单元 n 发生计数上溢 0：条件匹配时，Timer2 单元 m 同步计数无效 1：条件匹配时，Timer2单元m进行一次同步计数 （当m=1、2、3、4时，n=2、4、6、8）	R/W
b27~b26	ASYNCLKB[1:0]	异步计数同时钟源选择B	00：LRC 01：XTAL32 10：TIM2_<t>_CLKB时钟输入 11：设定禁止	R/W
b25~b24	SYNCLKB[1:0]	同步计数同时钟源选择B	00：PCLK1 01：TIM2_<t>_TRIGB上升沿（内部PCLK1同步） 10：TIM2_<t>_TRIGB下降沿（内部PCLK1同步） 11：内部触发事件输入	R/W
b23~b20	CKDIVB[3:0]	计数时钟分频选择B	计数时钟分频选择： 0000：时钟源 0001：时钟源/2 0010：时钟源/4 0011：时钟源/8 0100：时钟源/16 0101：时钟源/32 0110：时钟源/64 0111：时钟源/128 1000：时钟源/256 1001：时钟源/512 1010：时钟源/1024 请不要设定其它值	R/W

			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	SYNSB	计数方式选择B	0：同步计数方式 1：异步计数方式	R/W
b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17	CAPMDB	功能模式选择B	0：比较输出功能 1：捕获输入功能	R/W
b16	CSTB	定时器启动B	0：通道B定时器关闭 1：通道B定时器启动 注：该位在硬件触发停止条件有效时，会自动变为0	R/W
b15~b14	Reserved	-	读出时为“0”,写入时写“0”	R/W
b13	SYNCLKAT[1]	同步计数同时钟源选择AT	条件：Timer6 的单元 n 发生计数下溢 0：条件匹配时，Timer2 单元 m 同步计数无效 1：条件匹配时，Timer2单元m进行一次同步计数 （当 m=1、2、3、4 时，n=1、3、5、7）	R/W
b12	SYNCLKAT[0]	同步计数同时钟源选择AT	条件：Timer6 的单元 n 发生计数上溢 0：条件匹配时，Timer2 单元 m 同步计数无效 1：条件匹配时，Timer2单元m进行一次同步计数 （当m=1、2、3、4时，n=1、3、5、7）	R/W
b11~b10	ASYNCLKA[1:0]	异步计数同时钟源选择A	00：LRC 01：XTAL32 10：TIM2_<t>_CLKA时钟输入 11：设定禁止	R/W
b9~b8	SYNCLKA[1:0]	同步计数同时钟源选择A	00：PCLK1 01：TIM2_<t>_TRIGA上升沿（内部HCLK同步） 10：TIM2_<t>_TRIGA下降沿（内部HCLK同步） 11：内部触发事件输入	R/W
b7~b4	CKDIVA[3:0]	计数时钟分频选择A	计数时钟分频选择： 0000：时钟源 0001：时钟源/2 0010：时钟源/4 0011：时钟源/8 0100：时钟源/16 0101：时钟源/32 0110：时钟源/64 0111：时钟源/128 1000：时钟源/256 1001：时钟源/512 1010：时钟源/1024 请不要设定其它值 注：被分频的时钟源可以是异步计数时的各种时钟源、同步计数时的PCLK1	R/W

b3	SYNSA	计数方式选择A	0：同步计数方式 1：异步计数方式	R/W
b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	CAPMDA	功能模式选择A	0：比较输出功能 1：捕获输入功能	R/W
b0	CSTA	定时器启动A	0：通道A定时器关闭 1：通道A定时器启动 注：该位在硬件触发停止条件有效时，会自动变为0	R/W

25.5.4 中断控制寄存器（TMR2_ICONR）

复位值: 0x00000000h

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

位	标记	位名	功能	读写
b31~b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17	OVENB	计数溢出中断使能 B	0：计数值（CNTBR）=0xFFFFh 时，该中断无效 1：计数值（CNTBR）=0xFFFFh 时，该中断使能	R/W
b16	CMENB	计数匹配中断使能 B	0：CMPBR 寄存器与计数值（CNTBR）相等时，或者发生捕获输入事件时，该中断无效 1：CMPBR 寄存器与计数值（CNTBR）相等时，或者发生捕获输入事件时，该中断使能	R/W
b15~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	OVENA	计数溢出中断使能 A	0：计数值（CNTAR）=0xFFFFh 时，该中断无效 1：计数值（CNTAR）=0xFFFFh 时，该中断使能	R/W
b0	CMENA	计数匹配中断使能 A	0：CMPAR 寄存器与计数值（CNTAR）相等时，或者发生捕获输入事件时，该中断无效 1：CMPAR 寄存器与计数值（CNTAR）相等时，或者发生捕获输入事件时，该中断使能	R/W

25.5.5 端口控制寄存器（TMR2_PCONR）

复位值: 0x00000000h

b31	b30 b29	b28	b27	b26	b25	b24	b23	b22	b21 b20	b19 b18	b17 b16
-	NOFI CKB[1:0]	NOFI ENB	-	-	-	OUT ENB	-	-	CMP CB[1:0]	STP CB[1:0]	STA CB[1:0]

b15	b14 b13	b12	b11	b10	b9	b8	b7	b6	b5 b4	b3 b2	b1 b0
-	NOFI CKA[1:0]	NOFI ENA	-	-	-	OUT ENA	-	-	CMP CA[1:0]	STP CA[1:0]	STA CA[1:0]

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“0”,写入时写“0”	R/W
b30~b29	NOFICKB	滤波采样基准时钟选择 B	00：时钟源 01：时钟源/4 10：时钟源/16 11：时钟源/64 注：该时钟源为异步计数时的各种时钟源、同步计数时的 PCLK1	R/W
b28	NOFIENB	触发端口滤波 B	0：TIM2_<t>_TRIGB 输入端口滤波功能无效 1：TIM2_<t>_TRIGB 输入端口滤波功能使能	R/W
b27~b25	Reserved	-	读出时为“0”,写入时写“0”	R/W
b24	OUTENB	端口输出使能 B	0：TIM2_<t>_PWMB 端口输出无效 1：TIM2_<t>_PWMB 端口输出有效	R/W
b23~b22	Reserved	-	读出时为“0”,写入时写“0”	R/W
b21~b20	CMPCB[1:0]	比较值匹配时端口状态设定 B	00：计数值（CNTBR）与 CMPBR 相等时，TIM2_<t>_PWMB 端口输出设定为低电平 01：计数值（CNTBR）与 CMPBR 相等时，TIM2_<t>_PWMB 端口输出设定为高电平 10：计数值（CNTBR）与 CMPBR 相等时，TIM2_<t>_PWMB 端口输出保持先前状态 11：计数值（CNTBR）与 CMPBR 相等时，TIM2_<t>_PWMB 端口输出设定为反转电平	R/W
b19~b18	STPCB[1:0]	计数停止时端口状态设定 B	00：计数停止时，TIM2_<t>_PWMB 端口输出设定为低电平 01：计数停止时，TIM2_<t>_PWMB 端口输出设定为高电平 10：计数停止时，TIM2_<t>_PWMB 端口输出保持先前状态 11：计数停止时，TIM2_<t>_PWMB 端口输出保持先前状态	R/W
b17~b16	STACB[1:0]	计数启动时端口状态设定 B	00：计数启动时，TIM2_<t>_PWMB 端口输出设定为低电平 01：计数启动时，TIM2_<t>_PWMB 端口输出设定为高电平 10：计数启动时，TIM2_<t>_PWMB 端口输出保持先前状态 11：计数启动时，TIM2_<t>_PWMB 端口输出保持先前状态 注：该位设定只在不分频（BCONR.CKDIVB=4‘h0）的情况下有效，其它分频请设定为 2’b10 或 2‘b11	R/W

b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14~b13	NOFICKA	滤波采样基准时钟选择 A	00：时钟源 01：时钟源/4 10：时钟源/16 11：时钟源/64 注：该时钟源为异步计数时的各种时钟源、同步计数时的 PCLK1	R/W
b12	NOFIENA	触发端口滤波 A	0：TIM2_<t>_TRIGA 输入端口滤波功能无效 1：TIM2_<t>_TRIGA 输入端口滤波功能使能	R/W
b11~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8	OUTENA	端口输出使能 A	0：TIM2_<t>_PWMA 端口输出无效 1：TIM2_<t>_PWMA 端口输出有效	R/W
b7~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5~b4	CMPCA[1:0]	比较值匹配时端口状态设定 A	00：计数值（CNTAR）与 CMPAR 相等时，TIM2_<t>_PWMA 端口输出设定为低电平 01：计数值（CNTAR）与 CMPAR 相等时，TIM2_<t>_PWMA 端口输出设定为高电平 10：计数值（CNTAR）与 CMPAR 相等时，TIM2_<t>_PWMA 端口输出保持先前状态 11：计数值（CNTAR）与 CMPAR 相等时，TIM2_<t>_PWMA 端口输出设定为反转电平	R/W
b3~b2	STPCA[1:0]	计数停止时端口状态设定 A	00：计数停止时，TIM2_<t>_PWMA 端口输出设定为低电平 01：计数停止时，TIM2_<t>_PWMA 端口输出设定为高电平 10：计数停止时，TIM2_<t>_PWMA 端口输出保持先前状态 11：计数停止时，TIM2_<t>_PWMA 端口输出保持先前状态	R/W
b1~b0	STACA[1:0]	计数启动时端口状态设定 A	00：计数启动时，TIM2_<t>_PWMA 端口输出设定为低电平 01：计数启动时，TIM2_<t>_PWMA 端口输出设定为高电平 10：计数启动时，TIM2_<t>_PWMA 端口输出保持先前状态 11：计数启动时，TIM2_<t>_PWMA 端口输出保持先前状态 注：该位设定只在不分频（BCONR.CKDIVA=4‘h0）的情况下有效，其它分频请设定为 2’b10 或 2‘b11	R/W

25.5.6 硬件控制寄存器（TMR2_HCONR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	HICPB2	HICPB1	HICPB0	-	HCLEB2	HCLEB1	HCLEB0	-	HSTPB2	HSTPB1	HSTPB0	-	HSTAB2	HSTAB1	HSTAB0
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	HICPA2	HICPA1	HICPA0	-	HCLEA2	HCLEA1	HCLEA0	-	HSTPA2	HSTPA1	HSTPA0	-	HSTAA2	HSTAA1	HSTAA0

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“0”,写入时写“0”	R/W
b30	HICPB2	硬件捕获输入条件 B2	条件：内部硬件触发事件有效 0：条件匹配时，硬件捕获输入无效 1：条件匹配时，硬件捕获输入有效	R/W
b29	HICPB1	硬件捕获输入条件 B1	条件：TIM2_<t>_TRIGB 端口上采样到下降沿 0：条件匹配时，硬件捕获输入无效 1：条件匹配时，硬件捕获输入有效	R/W
b28	HICPB0	硬件捕获输入条件 B0	条件：TIM2_<t>_TRIGB 端口上采样到上升沿 0：条件匹配时，硬件捕获输入无效 1：条件匹配时，硬件捕获输入有效	R/W
b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26	HCLEB2	硬件清零条件 B2	条件：内部硬件触发事件有效 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b25	HCLEB1	硬件清零条件 B1	条件：TIM2_<t>_TRIGB 端口上采样到下降沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b24	HCLEB0	硬件清零条件 B0	条件：TIM2_<t>_TRIGB 端口上采样到上升沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b23	Reserved	-	读出时为“0”,写入时写“0”	R/W
b22	HSTPB2	硬件停止条件 B2	条件：内部硬件触发事件有效 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b21	HSTPB1	硬件停止条件 B1	条件：TIM2_<t>_TRIGB 端口上采样到下降沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b20	HSTPB0	硬件停止条件 B0	条件：TIM2_<t>_TRIGB 端口上采样到上升沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W

b19	Reserved	-	读出时为“0”,写入时写“0”	R/W
b18	HSTAB2	硬件启动条件 B2	条件：内部硬件触发事件有效 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
b17	HSTAB1	硬件启动条件 B1	条件：TIM2_<t>_TRIGB 端口上采样到下降沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
b16	HSTAB0	硬件启动条件 B0	条件：TIM2_<t>_TRIGB 端口上采样到上升沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效	R/W
b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14	HICPA2	硬件捕获输入条件 A2	条件：内部硬件触发事件有效 0：条件匹配时，硬件捕获输入无效 1：条件匹配时，硬件捕获输入有效	R/W
b13	HICPA1	硬件捕获输入条件 A1	条件：TIM2_<t>_TRIGA 端口上采样到下降沿 0：条件匹配时，硬件捕获输入无效 1：条件匹配时，硬件捕获输入有效	R/W
b12	HICPA0	硬件捕获输入条件 A0	条件：TIM2_<t>_TRIGA 端口上采样到上升沿 0：条件匹配时，硬件捕获输入无效 1：条件匹配时，硬件捕获输入有效	R/W
b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10	HCLEA2	硬件清零条件 A2	条件：内部硬件触发事件有效 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效	R/W
b9	HCLEA1	硬件清零条件 A1	条件：TIM2_<t>_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	HCLEA0	硬件清零条件 A0	条件：TIM2_<t>_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	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6	HSTPA2	硬件停止条件 A2	条件：内部硬件触发事件有效 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效	R/W
b5	HSTPA1	硬件停止条件 A1	条件：TIM2_<t>_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
b4	HSTPA0	硬件停止条件 A0	条件：TIM2_<t>_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
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	HSTAA2	硬件启动条件 A2	条件：内部硬件触发事件有效	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	
b1	HSTAA1	硬件启动条件 A1	条件：TIM2_<t>_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
b0	HSTAA0	硬件启动条件 A0	条件：TIM2_<t>_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

25.5.7 触发选择寄存器（TMR2_HTSSR）

复位值: 0x000001FFh

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1	0：禁止 AOS_COMTRG1 的公共触发事件触发 Timer2 1：允许 AOS_COMTRG1 的公共触发事件触发 Timer2 详见自动运行系统（AOS）章节	R/W
b30	COMEN[0]	公共触发使能0	0：禁止 AOS_COMTRG2 的公共触发事件触发 Timer2 1：允许 AOS_COMTRG2 的公共触发事件触发 Timer2 详见自动运行系统（AOS）章节	R/W
b29~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8~b0	TRGSEL	触发源选择	触发源编号写入，参考【中断控制器（INTC）】章节	R/W

25.5.8 状态标志寄存器（TMR2_STFLR）

复位值: 0x00000000h

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

位	标记	位名	功能	读写
b31~b18	Reserved	-	读出时为“0”,写入时写“0”	R
b17	OVFB	计数溢出标志B	0：计数值（CNTBR）未计数到0xFFFFh 1：计数值（CNTBR）计数到0xFFFFh	R/W
b16	CMFB	计数匹配标志B	0：CMPBR寄存器的值与计数值（CNTBR）不相等且未发生捕获输入动作 1：CMPBR寄存器的值与计数值（CNTBR）相等或发生捕获输入动作	R/W
b15~b2	Reserved	-	读出时为“0”,写入时写“0”	R
b1	OVFA	计数溢出标志A	0：计数值（CNTAR）未计数到0xFFFFh 1：计数值（CNTAR）计数到0xFFFFh	R/W
b0	CMFA	计数匹配标志A	0：CMPAR寄存器的值与计数值（CNTAR）不相等且未发生捕获输入动作 1：CMPAR寄存器的值与计数值（CNTAR）相等或发生捕获输入动作	R/W

25.6 使用注意事项

1) 在异步计数动作时，需先设定 BCONR.ASYNCLKA<B>位选择异步时钟源,再设定 BCONR.SYNSA<B>位选择异步计数方式，然后再启动 Timer2。
2) 在选择异步计数的情况下，修改计数值(CNTA<B>R)、基准值（CMPA<B>R）、启动位（BCONR.CSTA<B>）、状态位（STFLR.CMFA<B>）时，Timer2 从接收到写动作后经过 3 个异步计数时钟才将修改值写入对应的寄存器中。
3) 在选择异步计数的情况下，连续对计数值(CNTA<B>R)、基准值（CMPA<B>R）、启动位（BCONR.CSTA<B>）、状态位（STFLR.CMFA<B>）进行写动作时，需间隔至少 3 个异步计数时钟。
4) 在选择异步计数的情况下，请将 BCONR.SYNCLKA<B>[1:0]设定为 0。

26 通用定时器（Timer0）

26.1 简介

通用定时器 0（Timer0）是一个可以实现同步计数、异步计数方式的基本定时器。该定时器内含 2 个通道（CH-A 和 CH-B），可以在计数期间产生比较匹配事件。该事件可以触发中断，也可作为事件输出来控制其它模块等。本系列产品中搭载 2 个单元的 Timer0。

26.2 基本框图

Timer0 的基本框图如图 26-1 所示。

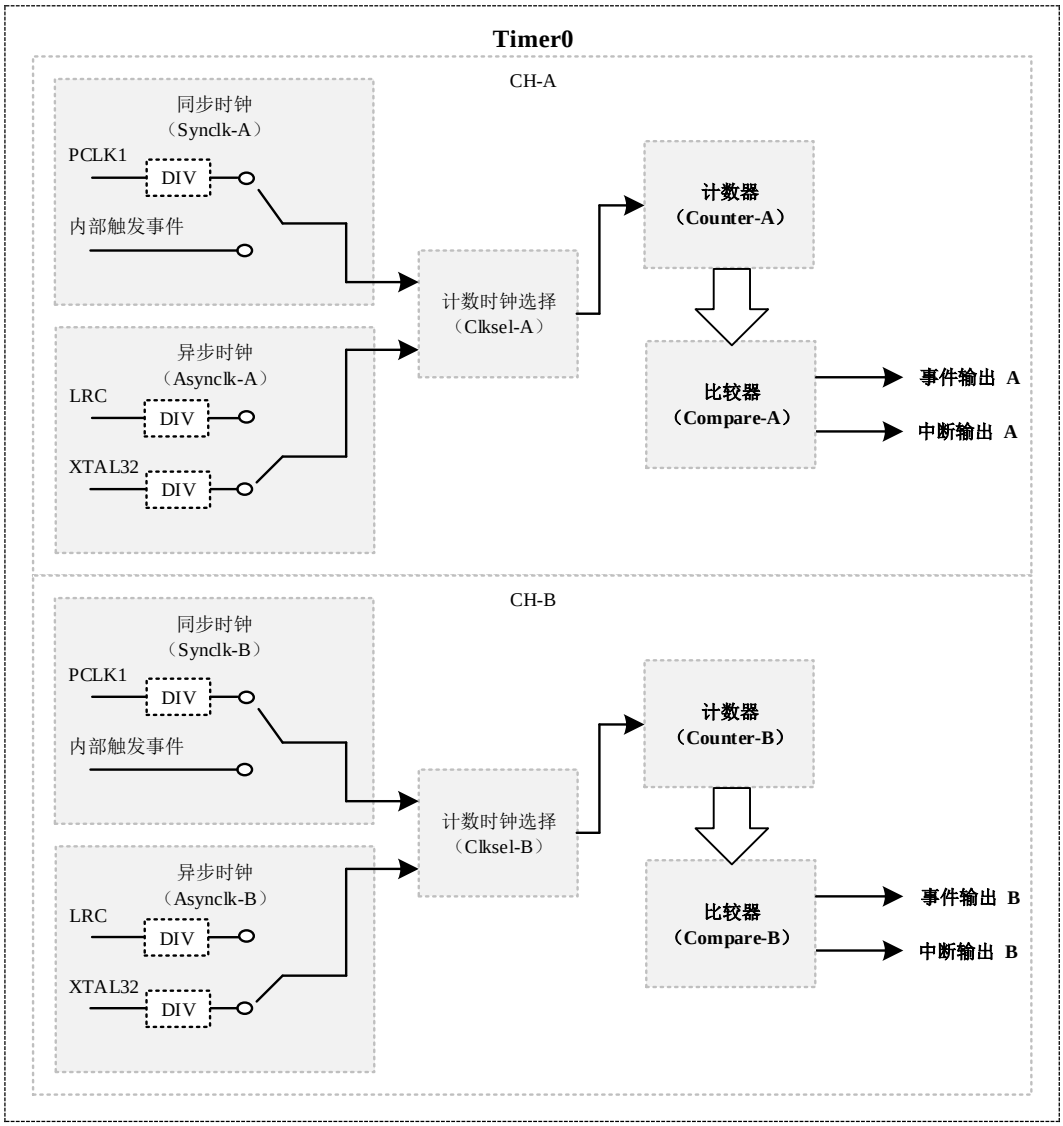


图 26-1 Timer0 基本框图

26.3 功能说明

26.3.1 时钟源选择

Timer0 的计数方式可以选择同步计数方式或异步计数方式。

同步计数方式是指定时器的计数时钟和总线访问时钟（寄存器读写操作时钟）有同步时序关系；异步计数方式是指定时器的计数时钟和总线访问时钟（寄存器读写操作时钟）是非同步的时序关系。异步计数方式时对寄存器读操作时，定时器等的状态可能正在发生变化、读出不可预期的状态。因此，在异步计数方式时，寄存器读操作须在计数停止状态下实现。

26.3.1.1 同步计数时钟源

同步计数方式时（BCONR.SYNSA<B>=0），时钟源可以有以下几种选择（BCONR.SYNCLKA<B>设定）：

a) PCLK1 及 PCLK1 的 2、4、8、16、32、64、128、256、512、1024 分频作为同步计数时钟（BCONR.SYNCLK A<B>=0 & BCONR.CKDIV A<B> [3:0]设定）

b) 内部硬件触发事件输入作为同步计数时钟（BCONR.SYNCLK A<B>=1）

26.3.1.2 异步计数时钟源

异步计数方式时（BCONR.SYNS A<B>=1），时钟源可以有以下几种选择（BCONR.ASYNCLK A<B>设定选择）：

a) LRC 时钟源输入及其 2、4、8、16、32、64、128、256、512、1024 分频作为异步计数时钟（BCONR.ASYNCLK A<B>=0 & BCONR.CKDIV A<B> [3:0]设定）

b) XTAL32 时钟源输入及其 2、4、8、16、32、64、128、256、512、1024 分频作为异步计数时钟（BCONR.ASYNCLK A<B>=1 & BCONR.CKDIV A<B> [3:0]设定）

26.3.2 基本计数

Timer0 的每个通道可设定基准计数值，在计数值和基准值相等时产生计数比较匹配事件。如图 26-2 所示。

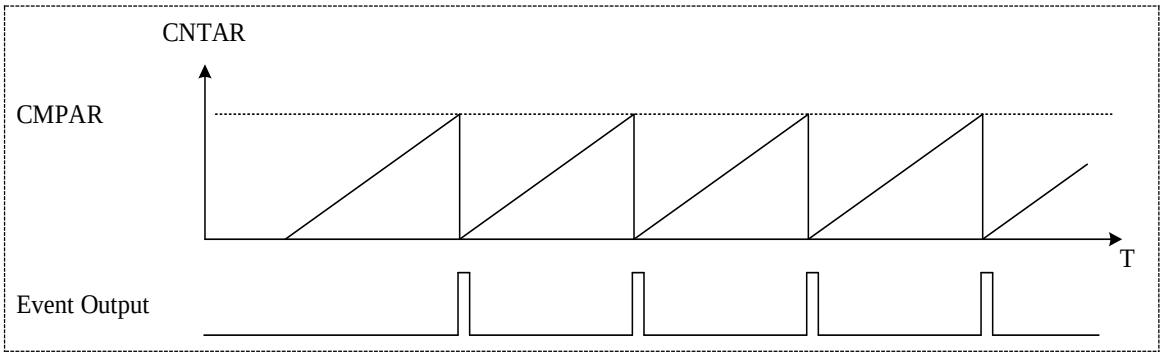


图 26-2 Timer0 计数时序图

26.3.3 硬件触发

Timer0 的 2 个通道有一个共用的内部硬件触发源，可以通过基本控制寄存器（BCONR）的相关设定来控制定时器的状态（计数、启动、停止、清零）以及捕获输入动作等。该硬件触发源的源选择通过向硬件触发选择寄存器（HTSSR）中输入对应的编号来实现，具体的事件对应关系请参考 INTC 章节。使用内部硬件触发功能时，需要先将功能时钟控制寄存器 0（PWR_FCG0）的外围电路触发功能位使能。

26.4 中断及事件说明

26.4.1 中断输出

一个 Timer0 含有 2 个中断输出，分别是通道 A 和通道 B 的计数比较匹配中断或捕获输入中断。

基准值寄存器（CMPAR、CMPBR）共计 2 个，可分别与计数值寄存器（CNTAR、CNTBR）比较产生比较匹配有效信号。计数比较匹配时，状态标志寄存器（STFLR）中的 STFLR.CMFA<B>位分别会被置为 1。此时若设定基本控制寄存器（BCONR）的 BCONR.INTENA<B>位使能中断，则对应的中断请求（TMR0_m_CMPn，m=1、2；n=A、B）也会被触发。

在内部硬件触发输入作为捕获输入条件时，可以产生相应的捕获输入动作。此时若设定基本控制寄存器（BCONR）的 BCONR.INTENA<B>位使能中断，则对应的中断请求（TMR0_m_CMPn，m=1、2；n=A、B）被触发。

在选择异步计数模式时，单元 1 的基准值寄存器（CMPAR）产生的比较匹配中断可用于在低功耗模式时对系统进行唤醒，具体请参考 INTC 章节。

26.4.2 事件输出

一个 Timer0 含有 2 个事件输出，分别是通道 A 和通道 B 的计数比较匹配事件或捕获输入事件。

在计数过程中发生计数比较匹配或捕获输入动作时，会分别产生相应的事件请求（TMR0_m_CMPn，m=1、2；n=A、B）输出信号，可以用于选择触发其它模块。

26.5 寄存器说明

表 26-1 所示，为 Timer0 模块的寄存器列表。

BASE ADDR: 0x40024000（U1）、0x40024400（U2）

寄存器名	符号	偏移量	位宽	复位值
计数值寄存器	TMR0_CNTAR	0x0000h	32	0x00000000h
计数值寄存器	TMR0_CNTBR	0x0004h	32	0x00000000h
基准值寄存器	TMR0_CMPAR	0x0008h	32	0x0000FFFFh
基准值寄存器	TMR0_CMPBR	0x000ch	32	0x0000FFFFh
基本控制寄存器	TMR0_BCONR	0x0010h	32	0x00000000h
触发选择寄存器	TMR0_HTSSR	（0x40010870h）	32	0x000001FFh
状态标志寄存器	TMR0_STFLR	0x0014h	32	0x00000000h

表 26-1 Timer0 寄存器列表

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。

26.5.1 计数值寄存器（TMR0_CNTmR）（m=A~B）

复位值: 0x00000000h

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”	R
b15~b0	CNTA<B>[15:0]	计数值	当前定时器的计数值	R/W

26.5.2 基准值寄存器（TMR0_CMPmR）（m=A~B）

复位值: 0x0000FFFFh

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”	R
b15~b0	CMPA<B>[15:0]	基准值	设定计数基准值，产生Compare Match事件	R/W

26.5.3 基本控制寄存器（TMR0_BCONR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
HICPB	HCLEB	HSTPB	HSTAB	-	ASYNCLKB	SYNCLKB	SYNSB	CKDIVB[3:0]				-	INTENB	CAPMDB	CSTB
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HICPA	HCLEA	HSTPA	HSTAA	-	ASYNCLKA	SYNCLKA	SYNSA	CKDIVA[3:0]				-	INTENA	CAPMDA	CSTA

位	标记	位名	功能	读写
b31	HICPB	硬件触发捕获输入B	条件：内部硬件触发事件有效 0：条件匹配时，捕获输入无效 1：条件匹配时，捕获输入有效	R/W
b30	HCLEB	硬件触发清零B	条件：内部硬件触发事件有效 0：条件匹配时，定时器清零无效 1：条件匹配时，定时器清零有效	R/W
b29	HSTPB	硬件触发停止B	条件：内部硬件触发事件有效 0：条件匹配时，定时器停止无效 1：条件匹配时，定时器停止有效	R/W
b28	HSTAB	硬件触发启动B	条件：内部硬件触发事件有效 0：条件匹配时，定时器启动无效 1：条件匹配时，定时器启动有效	R/W
b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26	ASYNCLKB	通道B异步计数时钟源选择	0：LRC 1：XTAL32	R/W
b25	SYNCLKB	通道B同步计数时钟源选择	0：PCLK1 1：内部硬件触发事件	R/W
b24	SYNSB	通道B计数方式选择	0：同步计数方式 1：异步计数方式	R/W
b23~b20	CKDIVB[3:0]	通道B计数时钟分频选择	通道B计数时钟分频选择： 0000：时钟源 0001：时钟源/2 0010：时钟源/4 0011：时钟源/8 0100：时钟源/16 0101：时钟源/32 0110：时钟源/64 0111：时钟源/128 1000：时钟源/256 1001：时钟源/512	R/W

			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	硬件触发捕获输入A	条件：内部硬件触发事件有效 0：条件匹配时，捕获输入无效 1：条件匹配时，捕获输入有效	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	通道A计数时钟分频选择： 0000：时钟源 0001：时钟源/2 0010：时钟源/4 0011：时钟源/8 0100：时钟源/16 0101：时钟源/32 0110：时钟源/64 0111：时钟源/128	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	
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（CNTAR）相等时，或者发生捕获输入事件时，该中断无效 1：CMPAR寄存器与计数值（CNTAR）相等时，或者发生捕获输入事件时，该中断使能	R/W
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。

26.5.4 触发选择寄存器（TMR0_HTSSR）

复位值: 0x000001FFh

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1	0：禁止 AOS_COMTRG1 的公共触发事件触发 Timer0 1：允许 AOS_COMTRG1 的公共触发事件触发 Timer0 详见自动运行系统（AOS）章节	R/W
b30	COMEN[0]	公共触发使能0	0：禁止 AOS_COMTRG2 的公共触发事件触发 Timer0 1：允许 AOS_COMTRG2 的公共触发事件触发 Timer0 详见自动运行系统（AOS）章节	R/W
b29~b9	Reserved	-	读出时为“0”,写入时写“0”	R/W
b8~b0	TRGSEL	触发源选择	触发源编号写入，参考【中断控制器（INTC）】章节	R/W

26.5.5 状态标志寄存器（TMR0_STFLR）

复位值: 0x00000000h

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

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	CMFB	计数匹配B	0：CMPBR寄存器的值与计数值（CNTAR）不相等且未发生捕获输入动作 1：CMPBR寄存器的值与计数值（CNTAR）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/W
b0	CMFA	计数匹配A	0：CMPAR寄存器的值与计数值（CNTAR）不相等且未发生捕获输入动作 1：CMPAR寄存器的值与计数值（CNTAR）相等或发生捕获输入动作	R/W

26.6 使用注意事项

1) 在异步计数动作时，需先设定 BCONR.ASYNCLKA<B>位选择异步时钟源,再设定 BCONR.SYNSA<B>位选择异步计数方式，然后再启动 Timer0。
2) 在选择异步计数的情况下，修改计数值(CNTA<B>R)、基准值（CMPA<B>R）、启动位（BCONR.CSTA<B>）、状态位（STFLR.CMFA<B>）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。
3) 在选择异步计数的情况下，连续对计数值(CNTA<B>R)、基准值（CMPA<B>R）、启动位（BCONR.CSTA<B>）、状态位（STFLR.CMFA<B>）进行写动作时，需间隔至少 3 个异步计数时钟。
4) 在选择异步计数的情况下，请将 BCONR.SYNCLKA<B>设定为 0。

27 实时时钟（RTC）

27.1 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 27-1 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
	• 时间戳功能
	• 备份寄存器复位功能
中断	周期中断
	闹钟中断
	入侵事件中断

表 27-1 RTC 的基本规格

27.2 基本框图

RTC 的基本框图如图 27-1 所示。

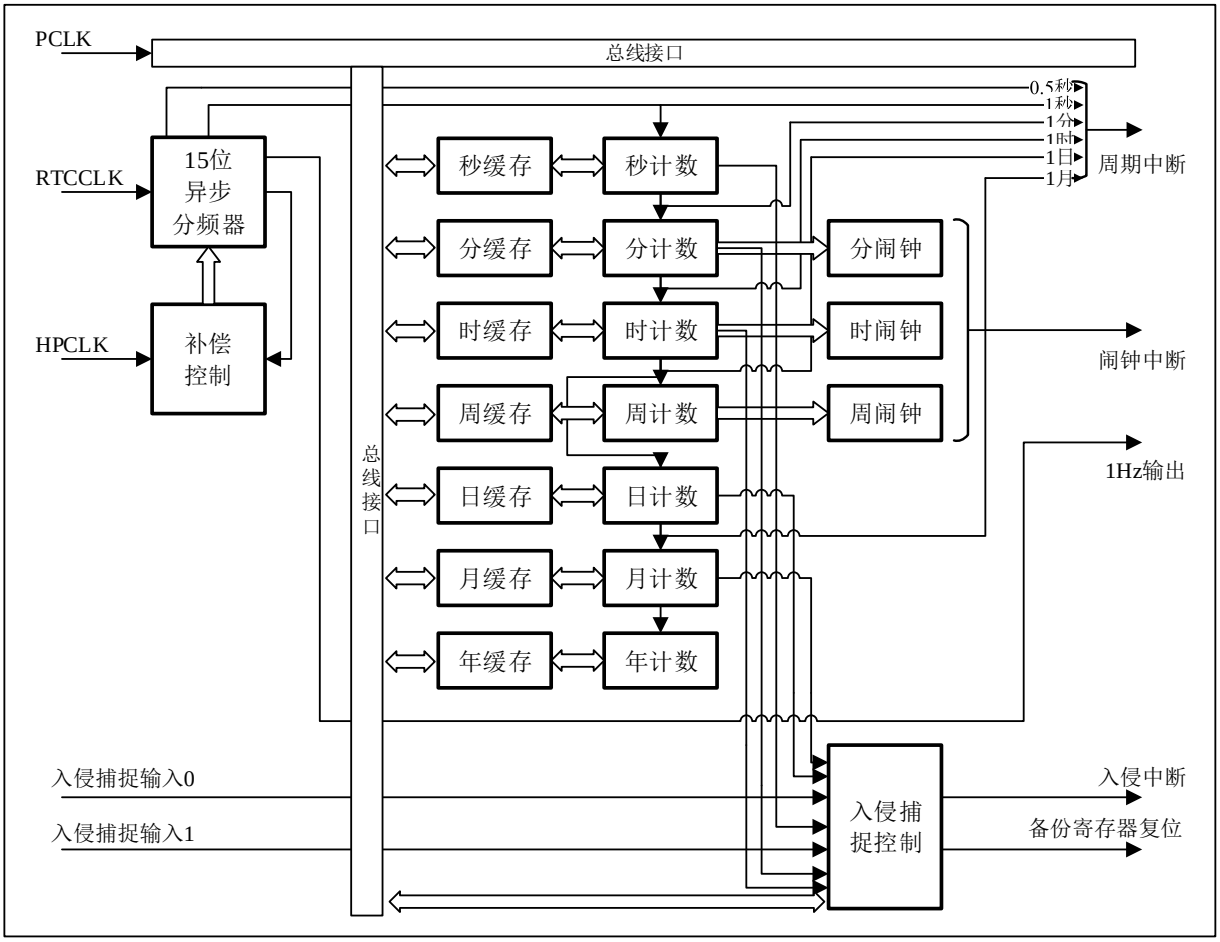


图 27-1 RTC 的基本框图

27.3 功能说明

27.3.1 上电设定

RTC 寄存器值上电复位后不定，上电复位后，必须先设定备份域复位寄存器 PWR_VBATRSTR 后，再设定控制寄存器 RTC_CR0.RESET 位复位所有寄存器。然后设置控制寄存器，日历初始值、闹钟设置，启动 RTC。RTC 启动之后，其他外部各种复位请求都不能复位 RTC，RTC 会一直处于工作状态。可以设定控制寄存器的 RTC_CR1.START 位为“0”停止 RTC 工作。RTC 在进行设定时，必须保证时钟源已稳定。

27.3.2 RTC 计数开始设定

1) 上电后，设定 RTC_CR0.RESET=0，确认 RESET 位为“0”后，设定 RTC_CR0.RESET=1，复位所有寄存器；
2) 设定 RTC_CR1.START=0，停止计数；
3) 设定系统时钟寄存器，打开外部低速振荡器，再设定 RTC_CR3，选择 RTC 计数时钟源；
4) 设定 RTC_CR1，设定时制、周期、1Hz 时钟输出；
5) 设定秒，分，时，周，日，月，年的日历计数寄存器；
6) 需要进行时钟误差补偿时，设定计数时钟误差补偿寄存器 RTC_ERRCRL，RTC_ERRCRH；
7) 清除寄存器 RTC_CR1，RTC_TPSR 中的标志寄存器位，并使能中断；
8) 设定 RTC_CR1.START=1，计数开始。

27.3.3 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。

27.3.4 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，周，日，月，年计数寄存器值；
4) 设定 RTC_CR2.RWREQ=0；
5) 查询直到 RTC_CR2.RWEN=0。

27.3.5 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，周，日，月，年计数寄存器值；
4) 设定 RTC_CR2.RWREQ=0。注意，须在 1 秒内完成所有写操作；
5) 查询直到 RTC_CR2.RWEN=0。

27.3.6 闹钟设定

1) 设定 RTC_CR1.ALME=0，闹钟禁止；
2) 设定 RTC_CR1.ALMIE=1，闹钟中断许可；
3) 分闹钟 RTC_ALMMIN,时闹钟 RTC_ALMHOUR,周闹钟 RTC_ALMWEEK 设定；
4) 设定 RTC_CR1.ALME=1，闹钟许可；
5) 等待闹钟中断；
6) 闹钟发生，RTC_CR1.ALMF=1，进入闹钟中断处理。

27.3.7 时钟误差补偿

由于外部低速晶振在各种温度条件下存在偏差，在需要得到高精度的计数结果时，需

要对误差进行补偿。补偿方法参照 27.5.15 时钟误差补偿寄存器（RTC_ERRCRH、RTC_ERRCRL）。

27.3.8 1Hz 输出

RTC 可输出 1Hz 时钟，提供三种精度输出方式，第一种，无时钟补偿的普通精度 1Hz 输出；第二种，每 32 秒内平均补偿的分布式补偿 1Hz 输出和第三种每秒补偿的均匀式补偿 1Hz 输出。当时钟误差补偿功能有效 RTC_ERRCRH.COMPEN=1 时可选择分布式补偿 1Hz 输出和均匀式补偿 1Hz 输出。其中，

普通精度的 1Hz 输出设定如下：

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。

27.3.9 入侵检测

当入侵检测 RTC_TPCRn.TPEN(n=0,1) 有效时，RTC 的外部管脚检测到有效的入侵事件后可分别触发入侵事件检测功能，时间戳功能和备份寄存器复位功能。

入侵事件检测功能：入侵事件发生时，对应引脚的入侵标志会被置起，当入侵事件连续发生时，上溢标志会被置起。入侵事件 0 和 1 同时发生时，入侵事件 0 标志置起后经过 1 个 RTC 计数时钟后入侵事件 1 标志再被置起，同时上溢标志也被置起。

RTC.TPCRn.TPIE（n=0,1）入侵事件中断使能时，将发生入侵检测中断。

时间戳功能：RTC.TPCRn.TSTPE（n=0,1）时间戳功能有效时，时间戳寄存器将记录入侵事件发生时的秒，分，时，日，月日历时间。

备份寄存器复位：RTC.TPCRn.TPRSTE（n=0,1）备份寄存器复位使能时，发生入侵事件后将复位备份寄存器。备份寄存器包括 32 个 32 位寄存器，用于存储用户 128 字节的备份数据。

注意：

– 在入侵检测有效时，不管时间戳功能，入侵事件中断和备份寄存器复位功能是否有效，事件发生后都将会置起入侵事件标志 RTC_TPSRn.TPFn（n=0,1）。

27.3.10时间戳功能

当入侵检测 RTC_TPCRn.TPEN(n=0,1)有效时，且时间戳记录功能 RTC_TPCRn.TSTPE（n=0,1）有效时，RTC 的外部管脚检测到有效的入侵事件后将触发时间戳功能。时间戳寄存器将记录秒，分，时，日，月日历时间。一组时间戳寄存器分别对应两个入侵事件输入管脚。当管脚在第一次发生入侵事件时，入侵标志置起，时间戳被记录，再次发生入侵事件时，则上溢标志位被置起，但是时间戳寄存器仍然保持第一次的记录值。入侵标志寄存器被清除后，入侵检测发生时，时间戳才会再次被记录。

27.4 中断说明

RTC 支持 3 种中断类型。计时闹钟中断、定周期中断和入侵检测中断。

27.4.1 闹钟中断

闹钟中断 RTC_ALM，在控制寄存器 2（RTC_CR2）的 ALMIE=1 并且控制寄存器 2（RTC_CR2）的 ALME=1 时，若当前日历时间与分闹钟寄存器（RTC_ALMMIN）、时闹钟寄存器（RTC_ALMHOUR）、周闹钟寄存器（RTC_ALMWEEK）相等时，触发闹钟中断。闹钟配置独立的标志寄存器位 RTC_CR2.ALMF，对 RTC_CR2.ALMF 位写“0”清除闹钟标志。

27.4.2 定周期中断

定周期中断 RTC_PRD，控制寄存器 2（RTC_CR2）的 PRDIE=1 时，选择的周期发生后，触发定周期唤醒中断。定周期中断配置独立的标志寄存器 RTC_CR2.PRDF，可通过对 RTC_CR2.PRDF 位写“0”清除定周期标志。

27.4.3 入侵检测中断

当入侵检测有效 RTC_TPCRn.TPEN(n=0,1)，且 RTC.TPCRn.TPIE（n=0,1）入侵事件中断使能时，RTC 的外部管脚检测到有效的入侵事件后，将发生入侵事件中断。由于两组入侵检测共用一个中断，可通过入侵事件检测中断标志位 RTC_TPSR.TPFn（n=0,1）来区分。

27.5 寄存器说明

表 27-2 所示，为 RTC 模块的寄存器列表。

寄存器基地址：0x4004C000

寄存器名	符号	偏移量	位宽	复位值
控制寄存器0	RTC_CR0	0x0000h	8	不定
控制寄存器1	RTC_CR1	0x0004h	8	不定
控制寄存器2	RTC_CR2	0x0008h	8	不定
控制寄存器3	RTC_CR3	0x000Ch	8	不定
秒计数寄存器	RTC_SEC	0x0010h	8	不定
分计数寄存器	RTC_MIN	0x0014h	8	不定
时计数寄存器	RTC_HOUR	0x0018h	8	不定
周计数寄存器	RTC_WEEK	0x001Ch	8	不定
日计数寄存器	RTC_DAY	0x0020h	8	不定
月计数寄存器	RTC_MON	0x0024h	8	不定
年计数寄存器	RTC_YEAR	0x0028h	8	不定
分闹钟寄存器	RTC_ALMMIN	0x002Ch	8	不定
时闹钟寄存器	RTC_ALMHOUR	0x0030h	8	不定
周闹钟寄存器	RTC_ALMWEEK	0x0034h	8	不定
时钟误差补偿寄存器	RTC_ERRCRH	0x0038h	8	不定
时钟误差补偿寄存器	RTC_ERRCRL	0x003Ch	8	不定
入侵控制寄存器0	RTC_TPCR0	0x0040h	8	不定
入侵控制寄存器1	RTC_TPCR1	0x0044h	8	不定
入侵状态寄存器	RTC_TPSR	0x0048h	8	不定
秒时间戳寄存器	RTC_SECTP	0x004Ch	8	不定
分时间戳寄存器	RTC_MINTP	0x0050h	8	不定
时时间戳寄存器	RTC_HOURTP	0x0054h	8	不定
日时间戳寄存器	RTC_DAYTP	0x0058h	8	不定
月时间戳寄存器	RTC_MONTP	0x005Ch	8	不定

表 27-2 寄存器列表

27.5.1 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	R/W

27.5.2 控制寄存器 1（RTC_CR1）

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								START	oneHZSEL	oneHZOE	-	AMPM	PRDS[2:0]		

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计数开始	R/W
b6	oneHZSEL	1Hz输出选择	0：分布式补偿1Hz输出 1：均匀式补偿1Hz输出 注意：RTC_ERRCRH.COMPEN=1时，该位设定有效。	R/W
b5	oneHZOE	1Hz输出许可	0：1Hz输出禁止 1：1Hz输出许可	R/W
b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	AMPM	时制选择	0：12小时时制 1：24小时时制	R/W
b2~0	PRDS[2:0]	周期中断选择	周期选择设定：	R/W

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。

27.5.3 控制寄存器 2（RTC_CR2）

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								ALME	ALMIE	PRDIE	-	ALMF	PRDF	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	闹钟功能使能	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 注意：在ALME=1时有效，闹钟匹配时，一个计数时钟后置“1“。写”0“时清除标志，写”1“无效。	R/W
b2	PRDF	周期标志	0：周期不发生 1：周期发生 注意：设定周期发生后，该位置“1“。写”0“时清除标志，写”1“无效	R/W
b1	RWEN	读出/写入允许	0：读出/写入禁止 1：读出/写入允许 注意：日历寄存器读写允许标志。在读出/写入前请确认该位是否为“1“。日历寄存器包括秒，分，时，周，日，月，年计数寄存器。	R/W
b0	RWREQ	读出/写入请求	0：正常计数模式 1：读出/写入请求 注意：在读出/写入日历寄存器时请将该位置“1”，请求读写，由于计数器在连续计数，请在1秒的时间内完成读出/写入操作并将该位清“0“。	R/W

27.5.4 控制寄存器 3（RTC_CR3）

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								RCKSEL	-	-	LRCEN	-			-

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：选择外部低速发振器XTAL32时钟 1：选择内部低速发振器RTCLRC时钟	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	LRCEN	内部低速发振器使能	0：内部低速发振器RTCLRC停止 1：内部低速发振器RTCLRC工作 注意：低速发振器作为RTC时钟源时，请设定LRCEN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

27.5.5 秒计数寄存器（RTC_SEC）

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									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。

27.5.6 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。

27.5.7 时计数寄存器（RTC_HOUR）

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										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小时时制	AMPM=1	12小时时制	AMPM=0
时间	寄存器表示	时间	寄存器表示
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

27.5.8 日计数寄存器（RTC_DAY）

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										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	HOURD[1:0]	日十位	日十位计数值	R/W
b3~b0	HOURU[3:0]	日个位	日个位计数值	R/W

十进制表示 1~31 日，自动计算闰年和月份。具体表示如下：

月份	日计数表示
2月（普通年）	01~28
2月（闰年）	01~29
4、6、9、11月	01~30
1、3、5、7、8、10、12月	01~31

27.5.9 周计数寄存器（RTC_WEEK）

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													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

27.5.10 月计数寄存器（RTC_MON）

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											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。

27.5.11 年计数寄存器（RTC_YEAR）

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								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。

27.5.12 分闹钟寄存器（RTC_ALMMIN）

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									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[3:0]	分闹钟十位	分闹钟十位匹配值	R/W
b3~b0	ALMMINU[3:0]	分闹钟个位	分闹钟个位匹配值	R/W

请设定十进制 0~59 的 BCD 码。写入其他值，不会发生闹钟匹配。

27.5.13 时闹钟寄存器（RTC_ALMHOUR）

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										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。

27.5.14 周闹钟寄存器（RTC_ALMWEEK）

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									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。

27.5.15 时钟误差补偿寄存器（RTC_ERRCRH、RTC_ERRCRL）

复位值: 不定

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]

复位值: 不定

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	1	0	0	0	0	0	0	0	0	-275.5ppm
	1	0	0	0	0	0	0	0	1	-274.6ppm
	~	~	~	~	~	~	~	~	~	~
	0	0	0	0	0	0	0	0	0	-30.5ppm
	~	~	~	~	~	~	~	~	~	~

	0	0	0	0	1	1	1	1	1	-0.96ppm
	0	0	0	1	0	0	0	0	0	0ppm
	0	0	0	1	0	0	0	0	1	+0.96ppm
	~	~	~	~	~	~	~	~	~	~
	0	0	1	0	0	0	0	0	0	+30.5ppm
	~	~	~	~	~	~	~	~	~	~
	0	1	1	1	1	1	1	1	0	+212.0ppm
	0	1	1	1	1	1	1	1	1	+212.9ppm
0	X	X	X	X	X	X	X	X	X	无效

补偿计算说明：

当默认状态下直接输出 1Hz 时钟，通过测定该时钟的精度，计算补偿目标值。

假设实际测定值为 0.9999888Hz，则：

实际发振频率　　= 32768×0.9999888 ≈32767.63

补偿目标值　= （实际发振频率 – 目标频率）/目标频率×10^6

= （32767.96 – 32768）/32768×10^6

= -11.29ppm

设定值计算：

$$COMP[8:0] = \left(\frac{补偿目标值[ppm] \times 2^{15}}{10^6}\right)_{取2的补码} + 0001.00000B$$

如果补偿目标值为+20.3ppm，计算相应的寄存器值如下：

$COMP[8:0] = (20.3 \times 2^{15}/10^6)_{取2的补码} + 0001.00000B$

$= (0.6651904)_{取2的补码} + 0001.00000B$

= 0000.10101B + 0001.00000B

= 0001.10101B

如果补偿目标值为-20.3ppm，计算相应的寄存器值如下：

$COMP[8:0] = (-20.3 \times 215/106)_{取2的补码} + 0001.00000B$

$= (-0.6651904)_{取2的补码} + 0001.00000B$

= 1111.01011B + 0001.00000B

= 0000.01011B

27.5.16 入侵控制寄存器 0（RTC_TPCR0）

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								TPEN0	TSTPE0	TPIE0	TPRSTE0	TPNF0[1:0]		TPCT0[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	TPEN0	入侵检测使能	0：RTCIC0输入引脚上入侵功能无效	R/W
			1：RTCIC0输入引脚上入侵功能有效	
b6	TSTPE0	时间戳使能	0：时间戳功能禁止	R/W
			1：时间戳功能允许	
b5	TPIE0	入侵事件中断使能	0：入侵事件中断禁止	R/W
			1：入侵事件中断许可	
b4	TPRSTE0	入侵事件复位使能	0：备份寄存器复位禁止	R/W
			1：备份寄存器复位许可	
b3~b2	TPNF0[1:0]	滤波功能	0X：滤波功能无效	R/W
			10：RTCIC0输入引脚检测按照计时时钟进行3次一致性检测滤波	
			11：RTCIC0输入引脚检测按照计时时钟32分频进行3次一致性检测滤波	
b1~b0	TPCT0[1:0]	有效边沿选择	00：RTCIC0输入引脚不检测	R/W
			01：RTCIC0输入引脚检测到上升沿时有效	
			10：RTCIC0输入引脚检测到下降沿时有效	
			11：RTCIC0输入引脚检测到上升沿或下降沿时有效	

27.5.17 入侵控制寄存器 1（RTC_TPCR1）

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								TPEN1	TSTPE1	TPIE1	TPRSTE1	TPNF1[1:0]		TPCT1[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	TPEN1	入侵检测使能	0：RTCIC1输入引脚上入侵功能无效	R/W
			1：RTCIC1输入引脚上入侵功能有效	
b6	TSTPE1	时间戳使能	0：时间戳功能禁止	R/W
			1：时间戳功能允许	
b5	TPIE1	入侵事件中断使能	0：入侵事件中断禁止	R/W
			1：入侵事件中断许可	
b4	TPRSTE1	入侵事件复位使能	0：备份寄存器复位禁止	R/W
			1：备份寄存器复位许可	
b3~b2	TPNF1[1:0]	滤波功能	0X：滤波功能无效	R/W
			10：RTCIC0输入引脚检测按照计时时钟进行3次一致性检测滤波	
			11：RTCIC0输入引脚检测按照计时时钟32分频进行3次一致性检测滤波	
b1~b0	TPCT1[1:0]	有效边沿选择	00：RTCIC1输入引脚不检测	R/W
			01：RTCIC1输入引脚检测到上升沿时有效	
			10：RTCIC1输入引脚检测到下降沿时有效	
			11：RTCIC1输入引脚检测到上升沿或下降沿时有效	

27.5.18 入侵状态寄存器（RTC_TPSR）

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													TPOVF	TPF1	TPF0

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	TPOVF	入侵状态上溢标志	0：复位状态或未发生入侵状态上溢 1：发生入侵上溢 注意：写入“0”后直接清除标志位。	R/W
b1	TPF1	入侵状态标志1	0：RTCIC1输入引脚上未发生入侵事件 1：RTCIC1输入引脚上发生入侵事件 注意：为防止误动作，必须在TPCT1[1:0]=00b，入侵检测边沿无效时才能清除该位。请在设置入侵功能有效前，确认该标志位的状态为0. 写入“0”后直接清除标志位。	R/W
b0	TPF0	入侵状态标志0	0：RTCIC0输入引脚上未发生入侵事件 1：RTCIC0输入引脚上发生入侵事件 注意：为防止误动作，必须在TPCT0[1:0]=00b，入侵检测边沿无效时才能清除该位。请在设置入侵功能有效前，确认该标志位的状态为0. 写入“0”后直接清除标志位。	R/W

27.5.19 秒时间戳寄存器（RTC_SECTP）

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									SECTPD[2:0]			SECTPU[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
b6~b4	SECTPD[2:0]	秒十位	秒时间戳十位计数值	R
b3~b0	SECTPU[3:0]	秒个位	秒时间戳个位计数值	R

27.5.20 分时间戳寄存器（RTC_MINTP）

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									MINTPD[2:0]			MINTPU[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
b6~b4	MINTPD[2:0]	分十位	分时间戳十位计数值	R
b3~b0	MINTPU[3:0]	分个位	分时间戳个位计数值	R

27.5.21 时时间戳寄存器（RTC_HOURTP）

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										HOURTPD[1:0]		HOURTPU[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
b5~b4	HOURTPD[1:0]	时十位	时时间戳十位计数值	R
b3~b0	HOURTPU[3:0]	时个位	时时间戳个位计数值	R

27.5.22 日时间戳寄存器（RTC_DAYTP）

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										DAYTPD[1:0]		DAYTPU[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
b5~b4	HOURTPD[1:0]	日十位	日时间戳十位计数值	R
b3~b0	HOURTPU[3:0]	日个位	日时间戳个位计数值	R

27.5.23 月时间戳寄存器（RTC_MONTP）

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											MONTP[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
b4~b0	MONTP[4:0]	月	月时间戳计数值	R

28 看门狗计数器（WDT/SWDT）

28.1 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的应用程序背离正常的运行而产生的软件故障。

两个看门狗都支持窗口功能。在计数开始前可预设窗口区间，计数值位于窗口区间时，可刷新计数器，计数重新开始。基本特性如表 28-1。

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
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 28-1 看门狗计数器的基本特性

28.2 功能说明

28.2.1 启动看门狗

看门狗计数器的启动方式有两种：硬件启动方式和软件启动方式。

硬件启动方式是指启动时从主闪存区域读取看门狗计数器的设定信息（ICG0 寄存器），计数器自动开始计数；软件启动方式是指设定控制寄存器后，写刷新寄存器完成刷新动作，计数器开始计数。

28.2.2 硬件启动方式

ICG0 寄存器的位 16(WDTAUTS)、位 0(SWDTAUTS)为 0 时，为硬件启动方式。选择硬件启动方式时，WDT_CR 和 SWDT_CR 寄存器的相关设定信息无效。

硬件启动方式时，在复位期间将 ICG0 寄存器里的 WDT/SWDT 相关设定（计数时钟、窗口设定值、计数周期等）载入到 WDT/SWDT 的模块中，复位之后、计数器按照设定自动开始计数。图 28-1 为硬件启动方式的动作例。

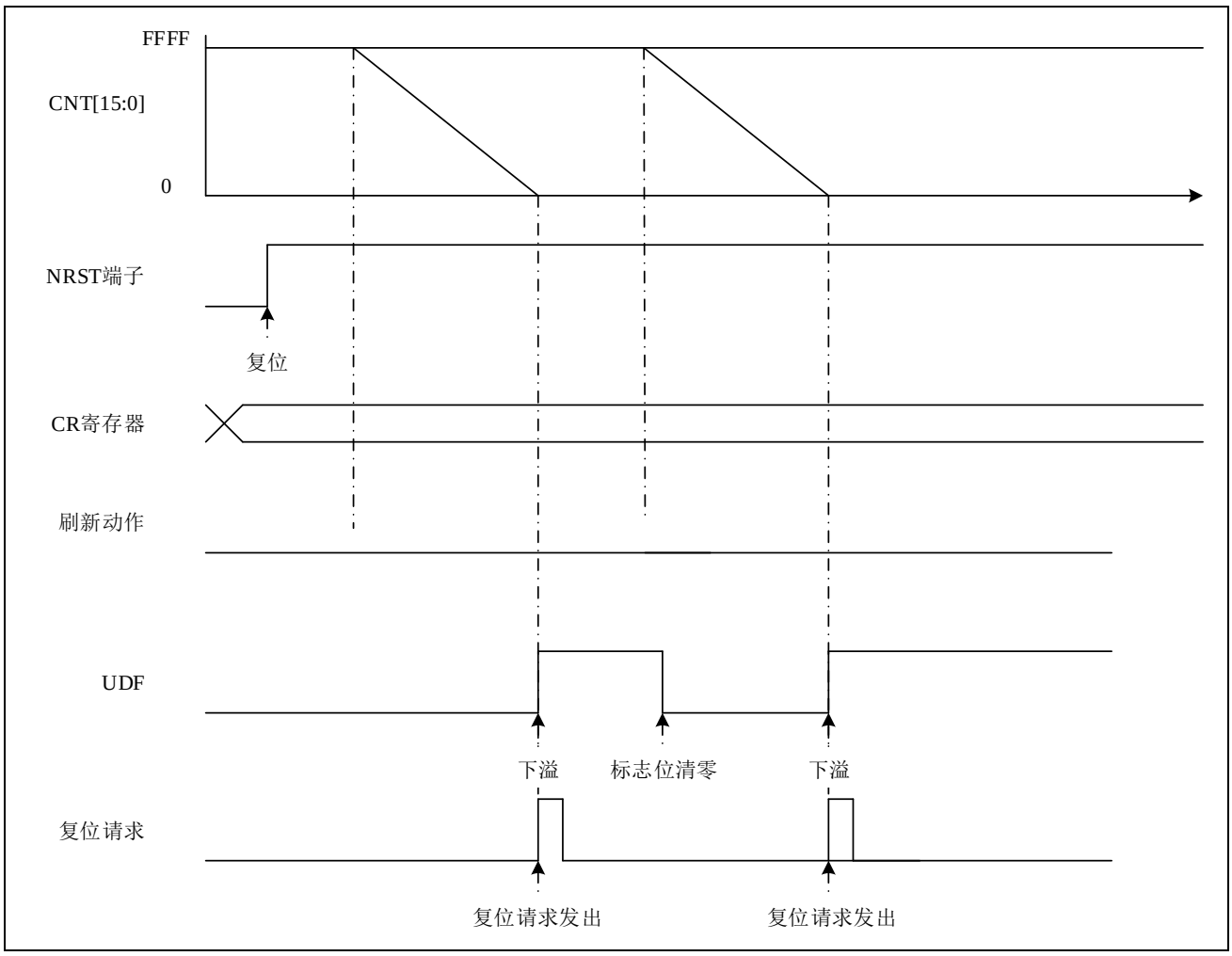


图 28-1 硬件启动例

28.2.3 软件启动方式

ICG0 寄存器的位 16(WDTAUTS) ，位 0(SWDTAUTS)为 1 时，通过设定刷新寄存器的方式启动 WDT/SWDT 为软件启动方式。复位后，设定 WDT_CR/SWDT_CR 寄存器中的计数时钟、窗口设定值、计数周期等，然后执行刷新动作，计数器就开始计数。WDT_CR/SWDT_CR 设定只能 1 次，再次设定写入值无效。图 28-2 为软件启动方式的动作例。

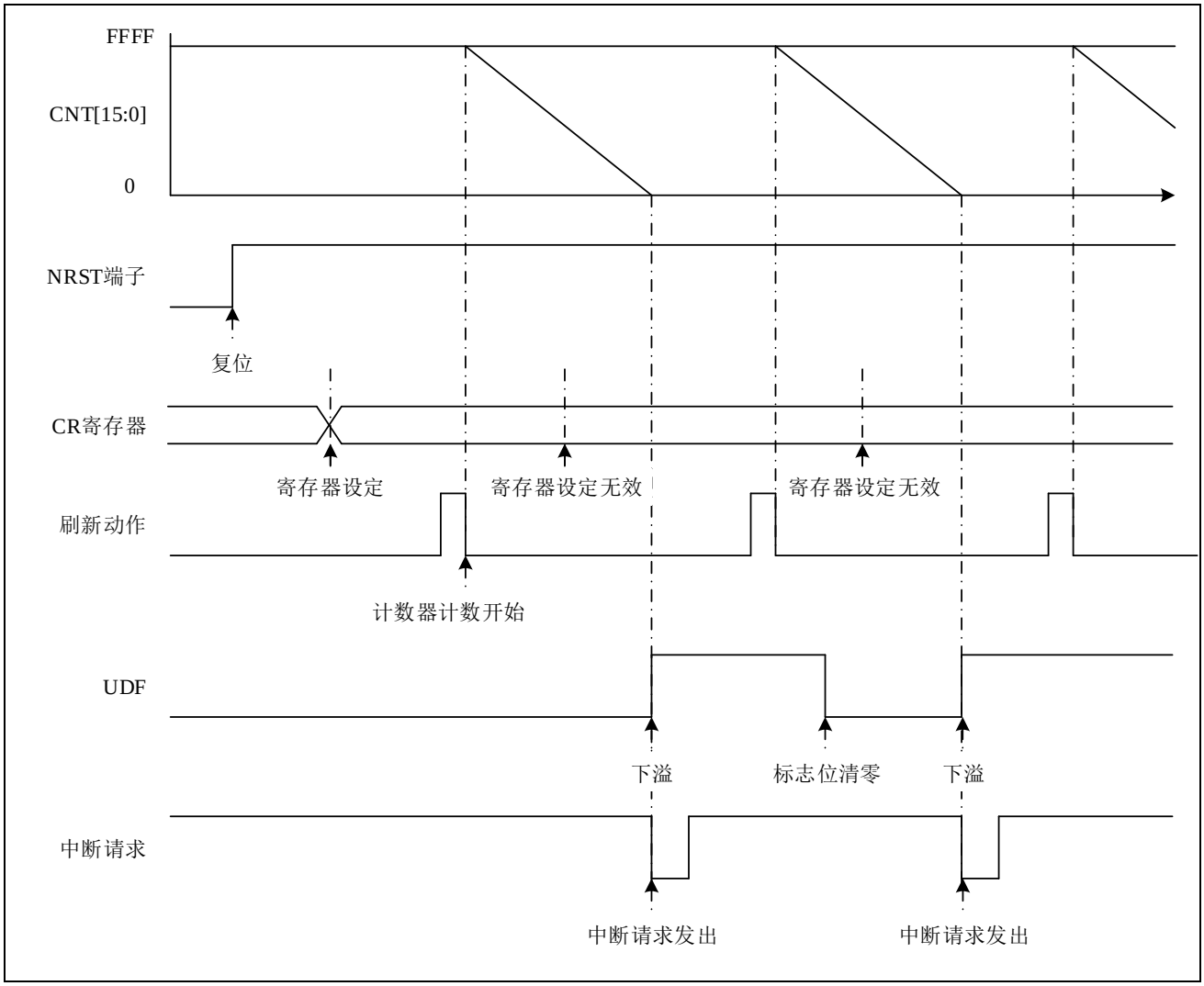


图 28-2 软件启动例

28.2.4 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 28-3 所示动作示例。

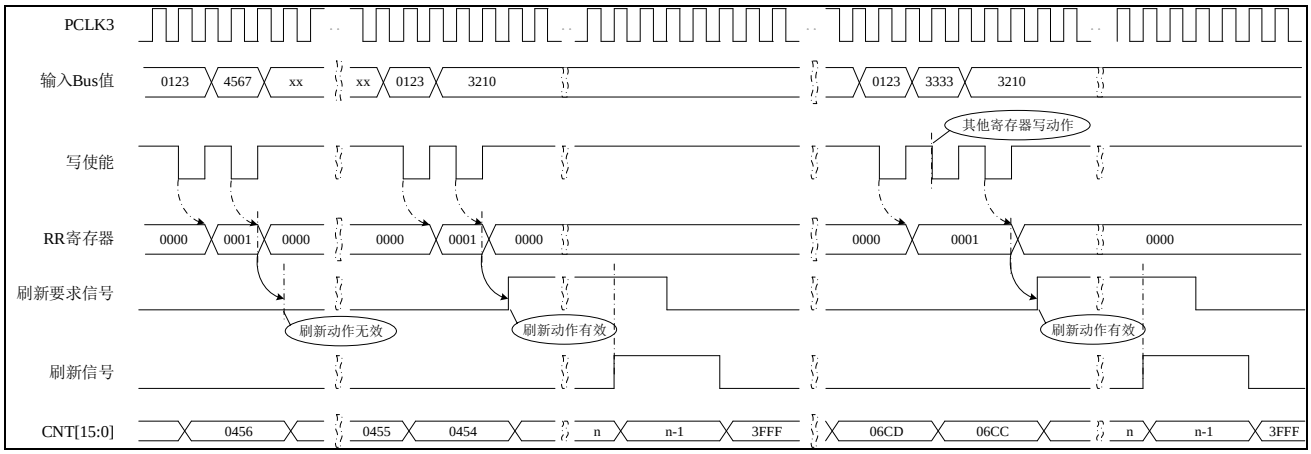


图 28-3 各种刷新动作时序示例（动作确认，刷新要求信号的下降沿等）

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。

28.2.5 标志位

刷新错误标志位和计数下溢标志位在中断请求的情况下会保持。当进入中断后，可以通过查询标志位来确认中断原因。标志位清零：先读“1”再写“0”。

刷新错误或者计数下溢标志位置位时，硬件启动模式看门狗计数不停止；软件启动模式看门狗计数停止。对标志位写“0”时，SWDT 最多需要经过 3 个 SWDTLRC 和 2 个 PCLK3 时间后，寄存器位才能被清零；WDT 最多需要经过 5 个 PCLK3 时间后，寄存器位才能被清零。另，在发生刷新错误或者下溢错误的一定时间内，对标志位读“1”清零无效，这段时间为：1 个计数周期+2 个 SWDTLRC(SWDT 模块)；1 个计数周期+2 个 PCLK3 (WDT 模块) 。

28.2.6 中断复位

WDT/SWDT 在计数器计数下溢或刷新错误时，可以选择产生中断请求或复位请求。硬件启动模式下，通过 ICG0 的 WDTITS/SWDTITS 位，来决定产生中断请求还是复位请求。软件启动模式下，通过设定 WDT_CR/SWDT_CR 寄存器 ITS 位，来决定产生中断请求还是复位请求。

WDT/SWDT 的中断复位产生条件有两种。一种是计数器计数产生下溢；一种是在刷新允许区间之外执行刷新动作，产生刷新错误。

28.2.7 计数下溢

如图 28-4 例，在递减计数到零时产生下溢。

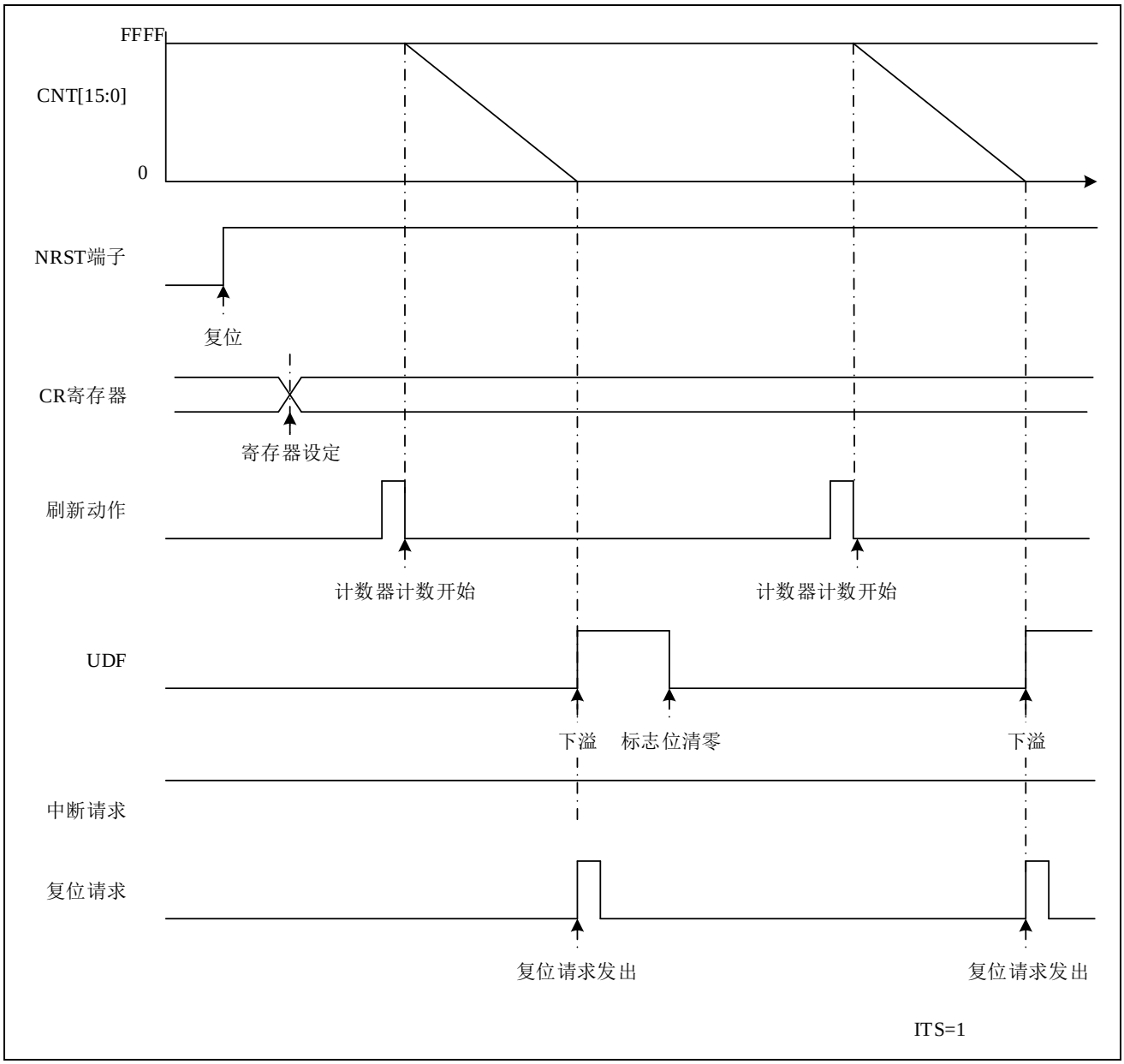


图 28-4 计数器下溢动作例

28.2.8 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 28-5 为刷新动作例。

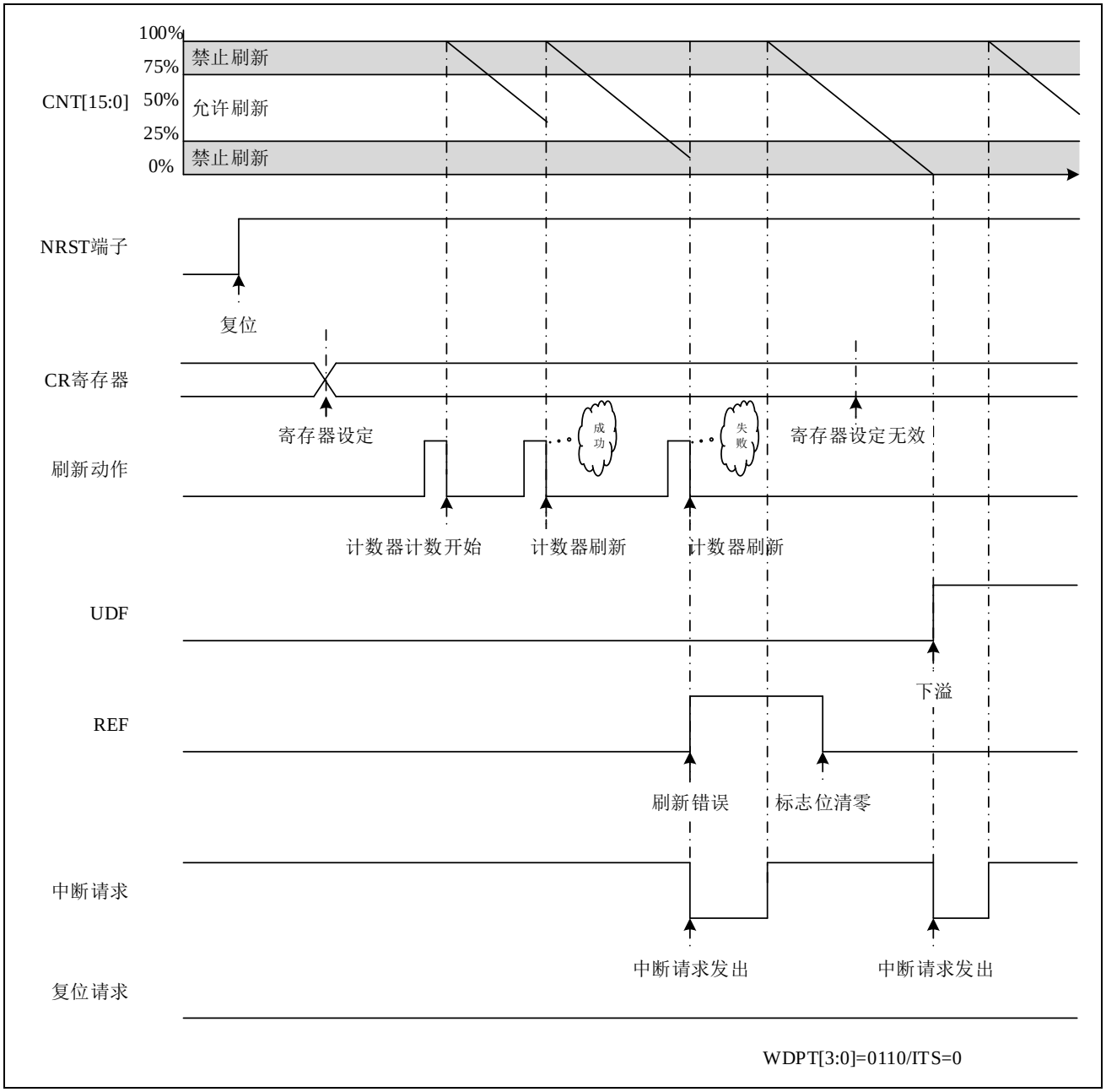


图 28-5 计数器刷新动作例

28.3 寄存器说明

表 28-2 所示，为 WDT 和 SWDT 模块的寄存器列表。

WDT_BASE_ADDR: 0x40049000

SWDT_BASE_ADDR:0x40049400

寄存器名	符号	偏移地址	位宽	复位值
SWDT控制寄存器	SWDT_CR	0x00	32	0x8001_0FF3
SWDT状态寄存器	SWDT_SR	0x04	32	0x0000_0000
SWDT刷新寄存器	SWDT_RR	0x08	32	0x0000_0000
WDT控制寄存器	WDT_CR	0x00	32	0x8001_0FF3
WDT状态寄存器	WDT_SR	0x04	32	0x0000_0000
WDT刷新寄存器	WDT_RR	0x08	32	0x0000_0000

表 28-2 寄存器列表

28.3.1 控制寄存器（SWDT_CR、WDT_CR）

复位值：0x8001_0FF3

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

位	标记	位名	功能	读写
b31	ITS	刷新错误/溢出中断/复位选择	0：中断请求 1：复位请求	R/W
b30~b17	Reserved	-	读出时为“0”,写入时写“0”	R/W
b16	SLPOFF	(S)WDT在低功耗模式下计数禁止	WDT_CR： 0：WDT在sleep模式下计数许可 1：WDT在sleep模式下计数禁止 SWDT_CR： 0：SWDT在sleep/stop模式下计数许可 1：SWDT在sleep/stop模式下计数禁止	R/W
b15~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11~b8	WDPT[3:0]	刷新允许区域计数值百分比	0000：0%~100% 0001：0%~25% 0010：25%~50% 0011：0%~50% 0100：50%~75% 0101：0%~25%,50%~75% 0110：25%~75% 0111：0%~75% 1000：75%~100% 1001：0%~25%,75%~100% 1010：25%~50%,75%~100% 1011：0%~50%,75%~100% 1100：50%~100% 1101：0%~25%,50%~100% 1110：25%~100% 1111：0%~100%	R/W
b7~b4	CKS[3:0]	计数时钟	WDT_CR： 0010：PCLK3/4 0110：PCLK3/64 0111：PCLK3/128 1000：PCLK3/256	R/W

			1001：PCLK3/512 1010：PCLK3/1024 1011：PCLK3/2048 1101：PCLK3/8192 其余值：预留功能 SWDT_CR： 0000：SWDTCLK 0100：SWDTCLK/16 0101：SWDTCLK/32 0110：SWDTCLK/64 0111：SWDTCLK/128 1000：SWDTCLK/256 1011：SWDTCLK/2048 其它值：预留	
b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1~b0	PERI[1:0]	计数周期	00：256 cycle 01：4096 cycle 10：16384 cycle 11：65536 cycle	R/W

28.3.2 状态寄存器（SWDT_SR、WDT_SR）

复位值：0x0000

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

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

位	标记	位名	功能	读写
b31~b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17	REF	刷新错误标志	0：没有刷新错误 1：发生刷新错误 对该位读出1后写入0，该位清零。	R/W
b16	UDF	计数下溢标志	0：没有计数下溢 1：发生计数下溢 对该位读出1后写入0，该位清零。	R/W
b15~b0	CNT[15:0]	计数值	计数器当前计数值	R/W

28.3.3 刷新寄存器（SWDT_RR、WDT_RR）

复位值：0x0000

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

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b0	RF[15:0]	刷新值	依次写入0x0123、0x3210之后，完成刷新动作。 当寄存器写入0x0123h后，读出寄存器为0x00000001;其余情况读出值都是0x00000000。	R/W

28.4 使用注意事项

SWDT 动作时，周边时钟 PCLK3 的动作频率必须大于或等于计数时钟频率的 4 倍，即 PCLK3 频率≥计数时钟频率 x4。

29 通用同步异步收发器（USART）

29.1 简介

本产品搭载通用串行收发器模块（USART）10 个单元。通用串行收发器模块（USART）能够灵活地与外部设备进行全双工数据交换；本 USART 支持通用异步串行通信接口（UART）,时钟同步通信接口，智能卡接口 (ISO/IEC7816-3)和 LIN 通信接口。支持调制解调器操作(CTS/RTS 操作）,多处理器操作。与 Timer0 模块配合支持 UART 接收 TIMEOUT 功能。USART_1 支持通过 RX 线唤醒 STOP 模式功能。

具体功能分配如下：

- UART：全通道支持
- 多处理器通信：全通道支持
- 时钟同步通信：全通道支持
- RX 线唤醒 STOP 模式功能：USART_1 支持
- 小数波特率：USART_1，USART_2，USART_3，USART_4，USART_6，USART_7，USART_8，USART_9 支持
- LIN：USART_5，USART_10 支持
- 智能卡：USART_1，USART_2，USART_3，USART_4，USART_6，USART_7，USART_8，USART_9 支持
- UART 接收超时功能：USART_1，USART_2，USART_6，USART_7 支持

USART 主要特性：

- 支持全双工异步通信，全双工时钟同步通信
- 支持 LIN 总线
- 支持智能卡接口（ISO/IEC7816-3）
- 发送器和接收器具有独立使能位
- 内置双缓冲器
- LSB/MSB 可选
- 支持调制解调器操作 (CTS/RTS)

- 传输标志：发送数据寄存器空，发送数据完成，接收数据寄存器满，接收错误标志，UART 接收超时标志，LIN 唤醒信号检出标志，LIN 间隔段检出标志，LIN 总线错误标志

UART 主要特性：

- 数据长度可编程:8 位/9 位
- 校验功能可配置：奇校验/偶校验/无校验
- 停止位可配置：1 位/2 位
- 时钟源可选：内部时钟源(内部波特率生成器生成的时钟)/外部时钟源(USARTn_CK 管脚输入的时钟)
- 接收错误：校验错误，帧错误，上溢错误
- 支持多个处理器间通信
- 内置数字滤波器
- 支持接收数据 TIMEOUT 功能
- 单元 1 支持停止模式唤醒功能
- 支持全双工/半双工通信方式

时钟同步模式主要特性：

- 数据长度：8 位
- 接收错误：上溢错误
- 时钟源：内部时钟源(内部波特率生成器生成的时钟)/外部时钟源(USARTn_CK 管脚输入的时钟)
- 支持全双工通信方式

智能卡接口的主要特性：

- 数据长度：8 位
- 检测到校验错误时能自动送出错误信号
- 支持数据重发

LIN 的主要特性

- 数据长度：8 位
- 支持唤醒信号的检测

- 支持 10/11 位同步间隔段（BF）的检测
- 支持同步段测量，寄存器记录测量值
- 支持 10/11/13/14 位的间隔段（BF）发送
- 支持总线冲突检测
- 支持回环模式

29.2 USART 系统框图

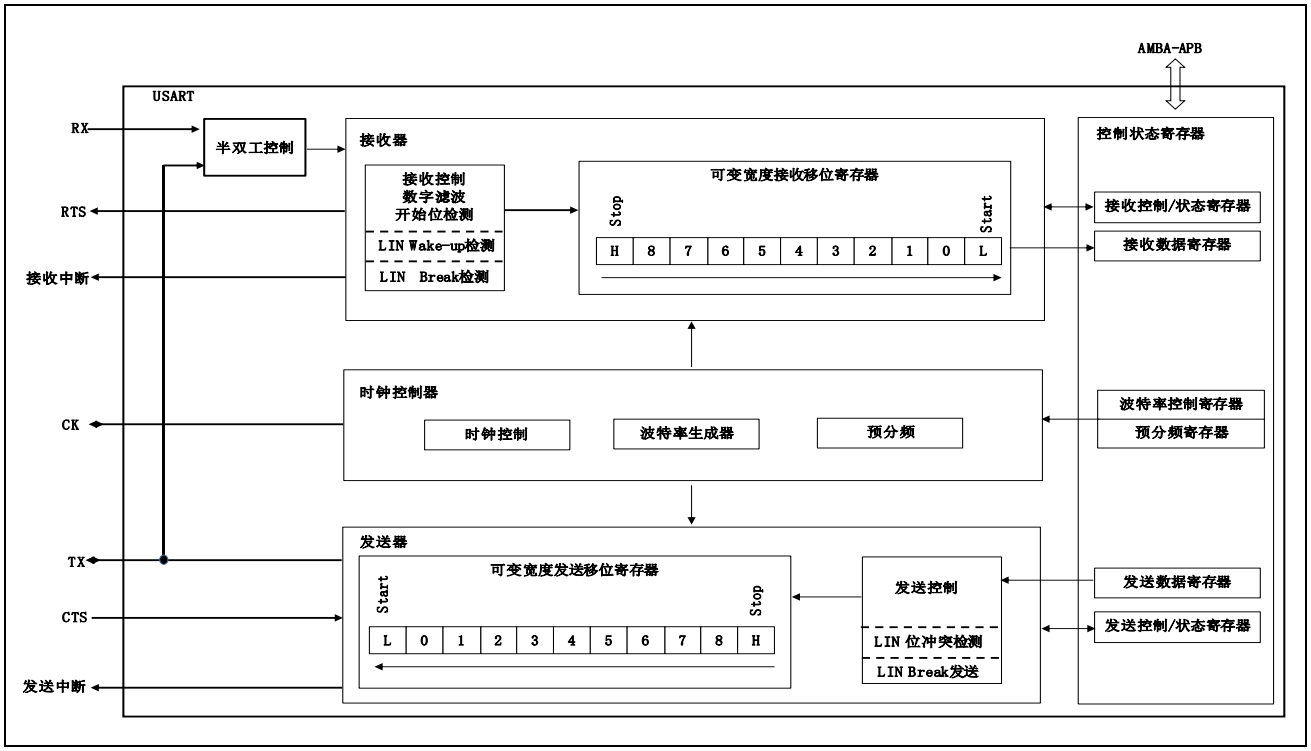


图 29-1 USART 系统框图

29.3 管脚说明

管脚名	方向	功能描述
USARTn_CK	输入输出	时钟
USARTn_TX	输入输出	发送数据管脚 半双工时也作为接收数据管脚
USARTn_RX	输入	接收数据管脚
USARTn_CTS	输入	调制解调器操作管脚
USARTn_RTS	输出	调制解调器操作管脚

n:1~10

表 29-1 USART 管脚说明

29.4 功能说明

本章将对 UART，多处理器通信，智能卡，时钟同步模式和 LIN 的功能详细说明。

29.4.1 UART

29.4.1.1 时钟

UART 可以选择内部波特率生成器生成的时钟(内部时钟源)或 USARTn_CK 管脚输入的时钟(外部时钟源)作为通信的时钟源。

内部时钟源

USARTn_CR2.CLKC[1:0]位设定为 00b 或者 01b 时选择时钟源为内部时钟源即内部波特率生成器生成的时钟。

USARTn_CR2.CLKC[1:0]=00b 时 USARTn_CK 管脚不作为时钟管脚使用，可以作为普通 IO 使用。

USARTn_CR2.CLKC[1:0]=01b 时从 USARTn_CK 管脚输出与通信波特率相同频率的时钟。

内部波特率生成器的时钟源由 USARTn_PR.PSC[1:0]位的设定选择为 PCLK，PCLK/4，PCLK/16，PCLK/64。

外部时钟源

USARTn_CR2.CLKC[1:0]位设定为 10b 或者 11b 时选择时钟源为从 USARTn_CK 管脚输入的外部时钟，输入时钟的频率为波特率的 16 倍(USARTn_CR1.OVER8=0)或者 8 倍(USARTn_CR1.OVER8=1)。

最高波特率

内部时钟源时，内部波特率生成器生成的波特率计算公式为：

$$B = \frac{C}{8 \times (2 - OVER8) \times (DIV_Integer + 1)}$$

B：波特率 单位：MBps

C：USARTn_PR.PSC[1:0]位设定的时钟（PCLK,PCLK/4,PCLK/16,PCLK/64） 单位：MHz

OVER8:USARTn_CR1.OVER8 设定值

DIV_Integer:USARTn_BRR.DIV_Integer 设定值

最高波特率为 PCLK/8(MBps)。

外部时钟源时，外部输入 UART 时钟的最高频率要求为 PCLK(MHz)/4，所以时钟源为外部输入时钟时最高波特率为 PCLK/64(MBps) (USARTn_CR1.OVER8=0 时）或者 PCLK/32 (MBps) (USARTn_CR1.OVER8=1 时）。

需要注意的是，UART 最高通信波特除了以上描述的基于 PCLK 的计算方法外，还需要参考电气特性章节规定的最高通信波特率。

29.4.1.2 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。

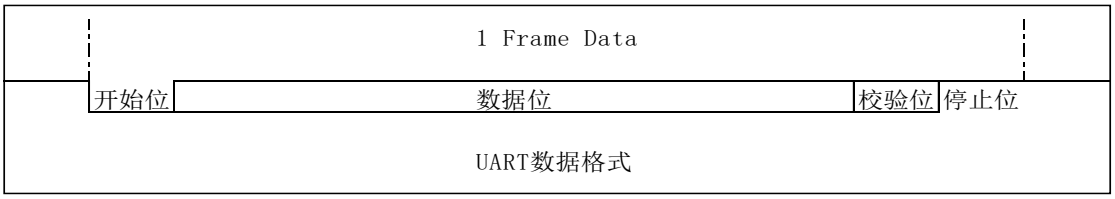


图 29-2 UART 数据格式

开始位

开始位固定有一位的低电平构成。

数据位

数据位可以配置成 8 位或者 9 位。

校验位

校验位可以配置成 1 位偶校验或 1 位奇校验或无校验位。

停止位

停止为固定为高电平，可以配置成 1 位或者 2 位。

29.4.1.3 调制解调器操作

调制解调器操作包括 CTS 功能和 RTS 功能。USARTn_CR3.RTSE=1 时 RTS 功能有效，USARTn_CR3.CTSE=1 时 CTS 功能有效。

CTS 功能

CTS 功能是通过 USARTn_CTS 管脚的输入来控制数据的发送，只有当 USARTn_CTS 管脚输入低电平时才可以发送数据，发送数据过程中如果 USARTn_CTS 输入高电平，正在发送的数据不受影响。

RTS 功能

RTS 功能是指通过 USARTn_RTS 管脚输出低电平，请求对方发送数据。

USARTn_RTS 管脚输出低电平需要满足以下全部条件：

- 接收使能(USARTn_CR1.RE=1)，且不正在接收数据
- USARTn_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

29.4.1.4 发送器

发送器可发送 8 位或 9 位的数据，具体取决于 USARTn_CR1.M 位的设定值。

发送器使能位 (USARTn_CR1.TE) 置 1，写入发送数据后，发送数据在 TX 管脚上串行输出；相应的时钟脉冲可以选择在 USARTn_CK 管脚输出或者不输出。

发送数据的顺序为：开始位->数据位(MSB/LSB)->校验位(有或者无)->停止位。

发送数据寄存器 TDR 和内部的发送移位寄存器组成双缓冲器结构，可以连续发送数据。

通过发送数据寄存器空中断或者 DMA 写入发送数据时，为保证发送的正确性，一次请求只能写入一次数据。

发送数据设定步骤

1. 将 USARTn_CR1 寄存器设定为复位值
2. 设定 UART 所需要使用的管脚
3. 通过 USARTn_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn_CR1, USARTn_CR2，USARTn_CR3 寄存器
5. 设定 USARTn_PR 选择预分频值，USARTn_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. 使能发送器（USARTn_CR1. TE=1），如果需要使用发送数据寄存器空中断，则设置 USARTn_CR1. TXEIE=1
7. 等待发送数据寄存器空，写通信数据到 USARTn_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。

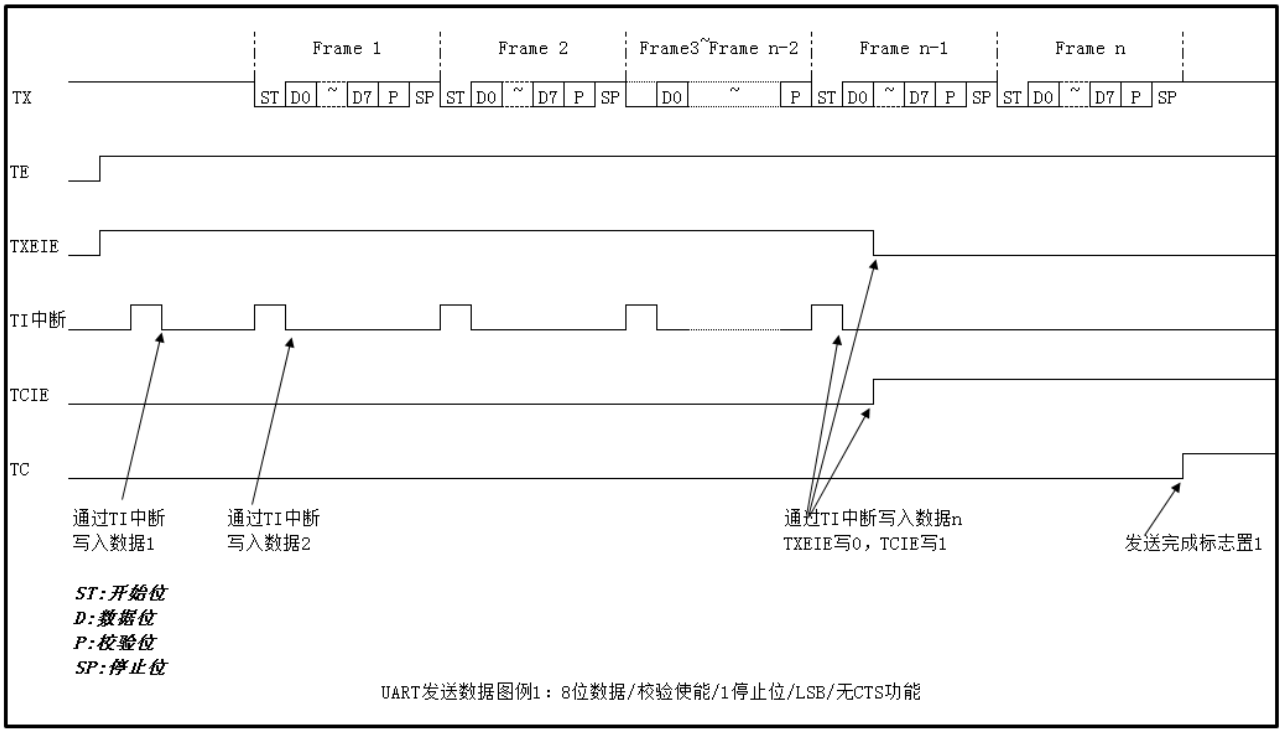


图 29-3 UART 发送数据图例 1

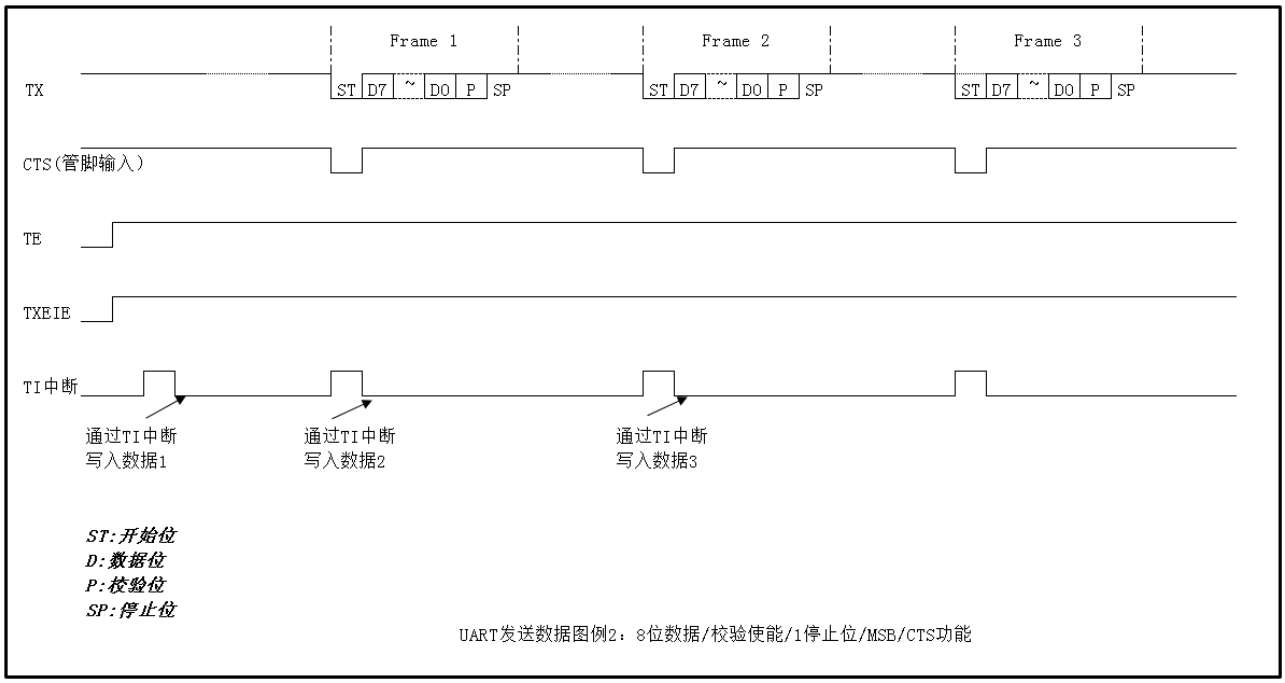


图 29-4 UART 发送数据图例 2

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。

29.4.1.5 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。

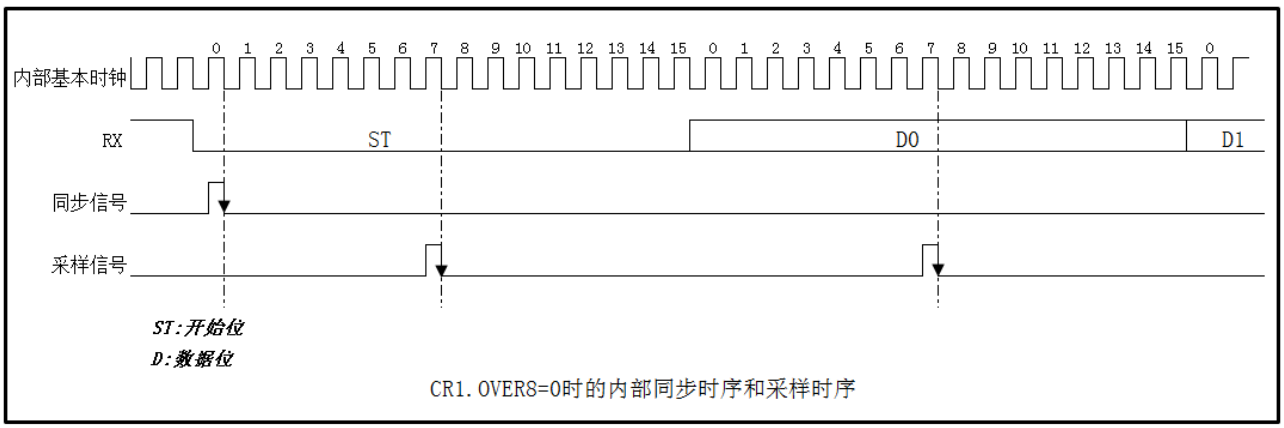


图 29-5 UART 内部同步和采样时序

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
- 是否使用小数波特率

接收容差的计算公式如下：

$$\text{RM_16}[\%] = |0.46 - (FL - 0.5)CFD| \times 100$$
$$\text{RM_8}[\%] = |0.43 - (FL - 0.5)CFD| \times 100$$

RM_16：16位过采样的接收容差

RM_8：8位过采样的接收容差

FL：帧长度（10~13）

CFD：时钟频率偏差

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

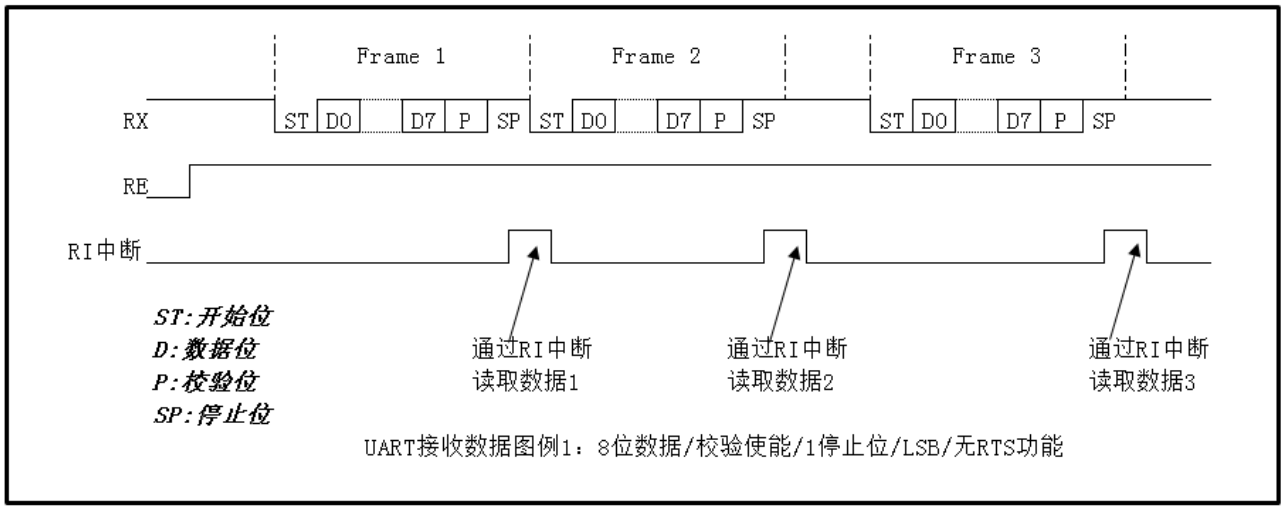


图 29-6 UART 接收数据图例 1

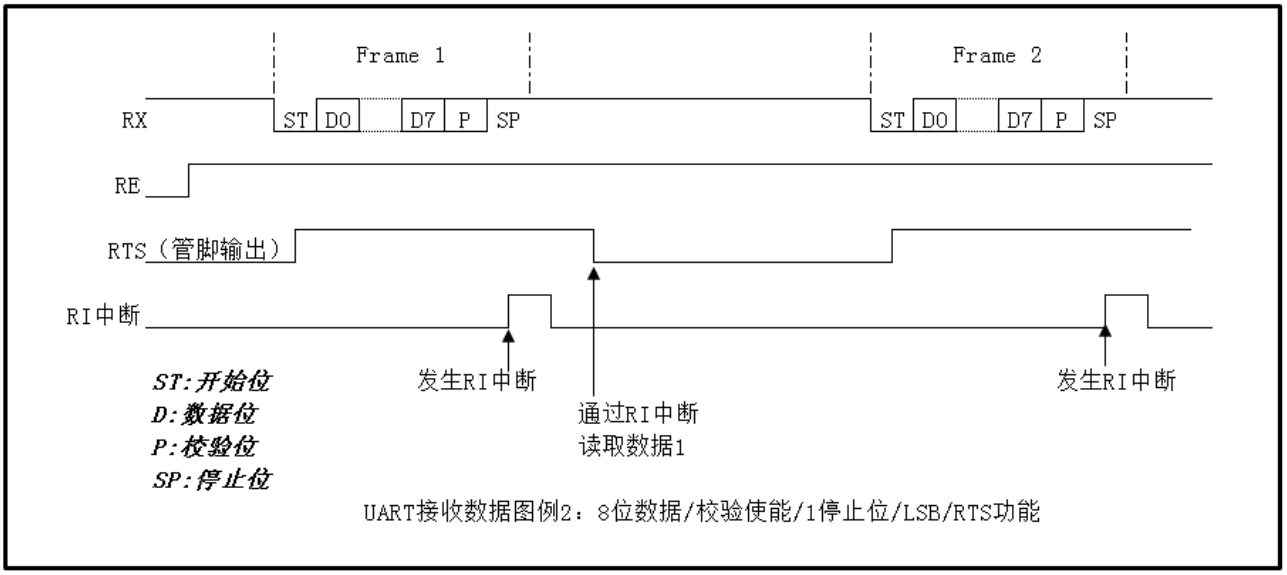


图 29-7 UART 接收数据图例 2

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 EI。

USARTn_CR.RIE=1，未发生任何接收错误，数据从接收移位寄存器传送到接收数据寄存器时 RI 中断发生。

USARTn_CR.RIE=1，接收过程中发生上溢错误，校验错误或者帧错误时 EI 中断发生。

29.4.1.6 UART 接收 TIMEOUT 功能

UART 接收数据停止位被检测时 TIMEOUT 计数器启动，经过设定的 TIMEOUT 时间（设定单位为接收位数）后未检测到下一帧接收数据时，发生 TIMEOUT，如果此时 CR1.RE=1, 则 TIMEOUT 状态位 USARTn_SR.RTOF 置位，如果此时 USARTn_CR1.RE=0，则等待 USARTn_CR1.RE=1 后 TIMEOUT 状态位 USARTn_SR.RTOF 置位。

TIMEOUT 计数器采用 Timer0 模块的计数器，具体对应关系如下：

USART_1：Timer0 Unit1 A 通道

USART_2：Timer0 Unit1 B 通道

USART_6：Timer0 Unit2 A 通道

USART_7：Timer0 Unit2 B 通道

TIMEOUT 功能 Timer0 比较计数器值设定

Timer0 为 16 位计数器，计数时钟最大可以选择 1024 分频，TMR0_CMPA<B>R 值设定计算公式如下：

$$CMPA<B>R = \frac{RTB}{2^{CKDIVA<B>}} - 4$$（计数时钟不分频）

$$CMPA<B>R = \frac{RTB}{2^{CKDIVA<B>}} - 2$$（计数时钟 2 分频）

$$CMPA<B>R = \frac{RTB}{2^{CKDIVA<B>}} - 1$$（计数时钟 4 分频及以上）

CMPA<B>R:TMR0_CMPA<B>R 寄存器值

RTB: Receive Timeout Bits

CKDIRA<B>: TMR0.BCONR.CKDIVA<B>位寄存器值

TIMEOUT 功能设定步骤

1. 将 USARTn_CR1 寄存器设定为复位值
2. 设定 UART 所需要使用的管脚
3. 通过 USARTn_CR2.CLK[1:0] 位选择时钟源（如果选择内部时钟源时需设置 CR2.CLKC[0]=1）
4. 设定 USARTn_CR1, USARTn_CR2，USARTn_CR3 寄存器
5. 设定 USARTn_PR 选择预分频值，USARTn_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. USARTn_CR1.RTOE=1，如果需要使用中断，则设定 USARTn_CR1.RTOIE=1
7. 设置 TMR0.BCONR.CSTA<B>=0
8. 设置 TMR0_CNTA<B>R 为 0，设置 TMR0_CMPA<B>R 寄存器和 TMR0.BCONR.CKDIVA<B>寄存器决定 TIMETOU 时间，
   设置 TMR0.BCONR.HCLEA<B>=1，TMR0.BCONR.HSTAA<B>=1，
   TMR0.BCONR.ASYNCLKA<B>=1，TMR0.BCONR.SYNSA<B>=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<B>=0，
    设置 TMR0.BCONR.CSTA<B>=0，通过写 USARTn_CR1.CRTOF 清除 USARTn_SR.RTOF 状态位。

29.4.1.7 RX 线唤醒停止模式功能

UART 通信空闲时，可以让系统进入停止模式以节省电流消耗，在不改变 UART PORT 设定的情况下，UART 单元 1 可以通过 RX 线来唤醒系统的停止模式。具体步骤如下：

1. UART 通信空闲时，设定 USART_1_WUPI 中断向量以及 INT_WUPEN. RXWUEN 位使能 UART 接收信号线唤醒停止模式功能。
2. 系统进入停止模式。

3. 系统检测到 RX 线下降沿时，从停止模式返回，在 USART_1_WUPI 中断处理程序中关闭该功能。

需要注意的是，当通信方需要唤醒本系统时，需要发送一帧唤醒数据（建议为 0x00），该数据不会被 UART 接收且不置位相关的标志。并且通信方需要经过系统停止模式唤醒需要的时间后再进行 UART 通信数据的传输。

UART RX 线唤醒功能，可以对 RX 线上的噪声进行过滤，详细信息请参考 USART_SYCTLREG 寄存器。

29.4.1.8 UART 半双工通信方式

UART 模式支持单线半双工模式，UART 模式下通过设定 USARTn_CR3.HDSEL=1 启用单线半双工模式。

单线半双工模式时：

- TX 和 RX 线从内部相连接，不再使用 RX 管脚。
- 无数据传输时，TX 管脚处于释放状态。因此使用单线半双工模式时，通过上拉 TX 线来避免无数据传输时的浮空输入。

除此以外，半双工模式与正常 UART 模式通信相似。需要注意的时，发送过程不会被硬件封锁，只要数据在 USARTn_CR1.TE=1 时写入，发送就会进行。因此线路上的冲突必须有软件进行管理。

29.4.1.9 UART 中断和事件

功能名称	使能位（仅中断）	标志位	可否作为事件源
接收错误中断	RIE	ORE,FE,PE	可
接收数据寄存器满中断	RIE	RXNE	可
发送数据寄存器空中断	TXEIE	TXE	可
发送完成中断	TCIE	TC	可
TIMEOUT中断	RTOIE	RTOF	可
RX线唤醒停止模式中断	INT_WUPEN.RXWUEN	-	不可

表 29-2 UART 中断/事件表

29.4.2 多处理器通信

29.4.2.1 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。

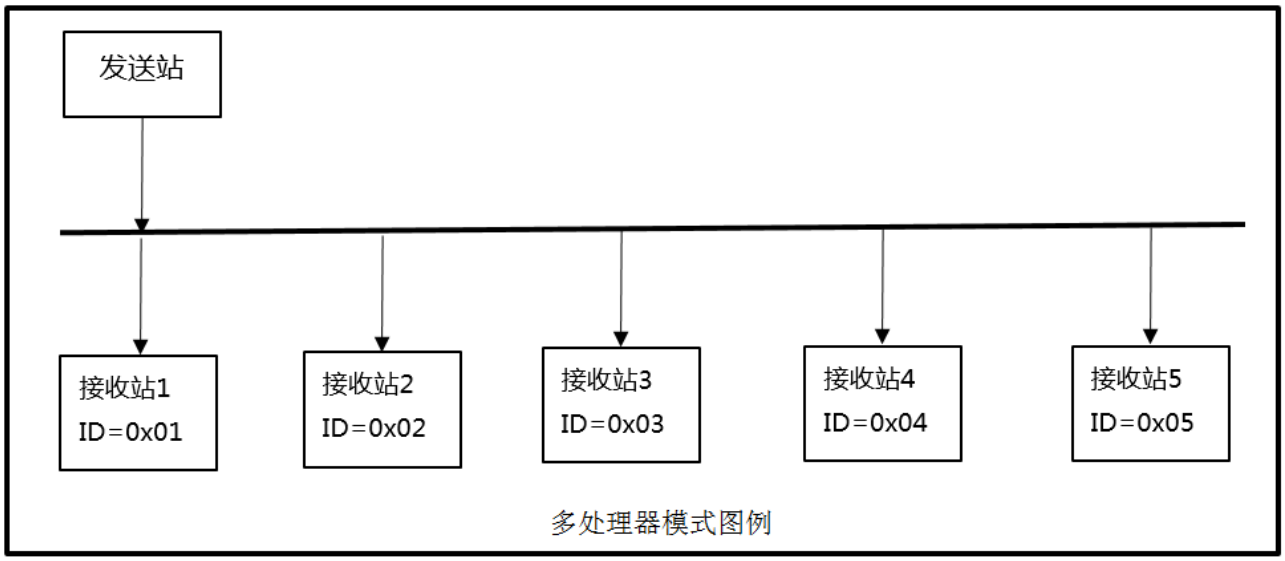


图 29-8 多处理器通信图例

29.4.2.2 数据格式

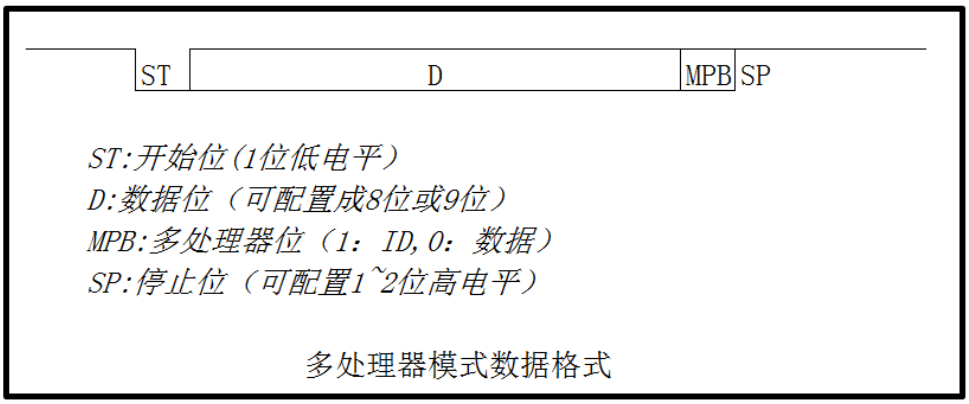


图 29-9 多处理器模式数据格式

29.4.2.3 动作说明

多处理器模式时校验位功能无效，增加了多处理器位功能，其余功能如时钟，中断等与 UART 模式相同。

发送站动作

1. 将 USARTn_CR1 寄存器设定为复位值
2. 设定所需要使用的管脚
3. 通过 USARTn_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn_CR1, USARTn_CR2，USARTn_CR3 寄存器
5. 设定 USARTn_PR 选择预分频值，USARTn_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. 使能发送器（USARTn_CR1. TE=1），如果需要使用发送数据寄存器空中断，则设置 USARTn_CR1. TXEIE=1
7. 等待发送数据寄存器空，设定 USARTn_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。

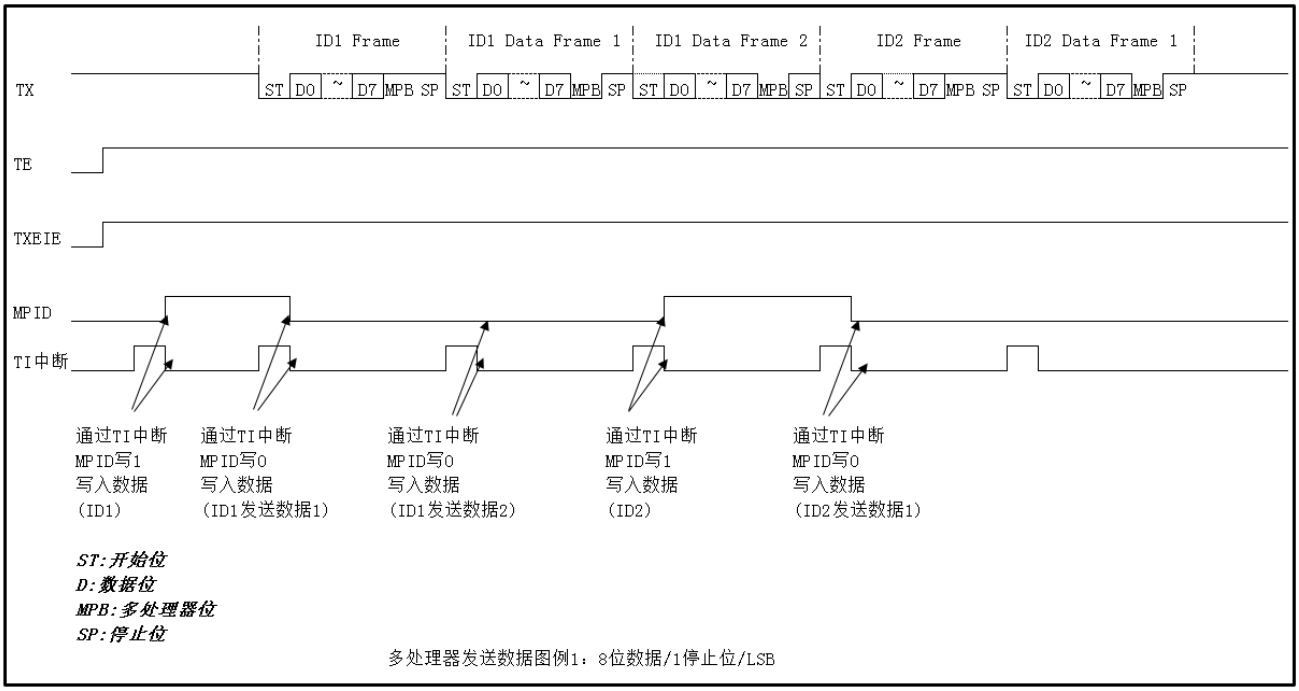


图 29-10 多处理器模式发送数据图例

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

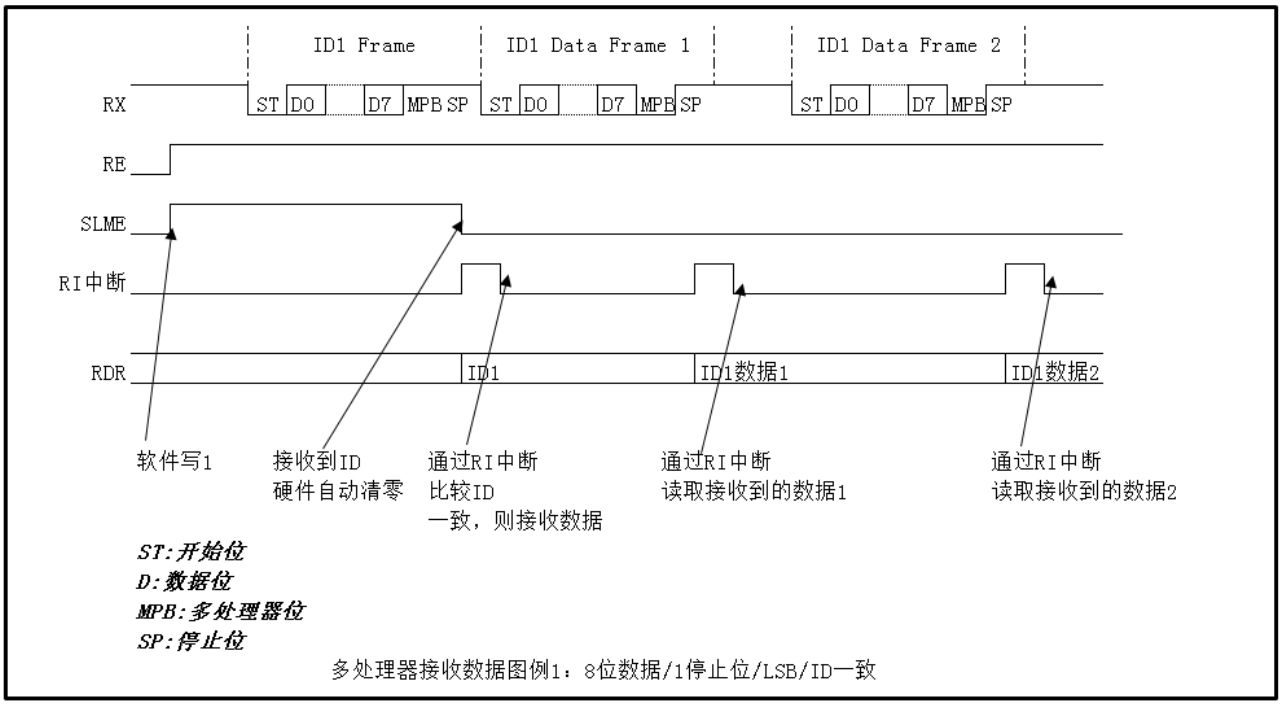


图 29-11　多处理器模式接收数据图例 1

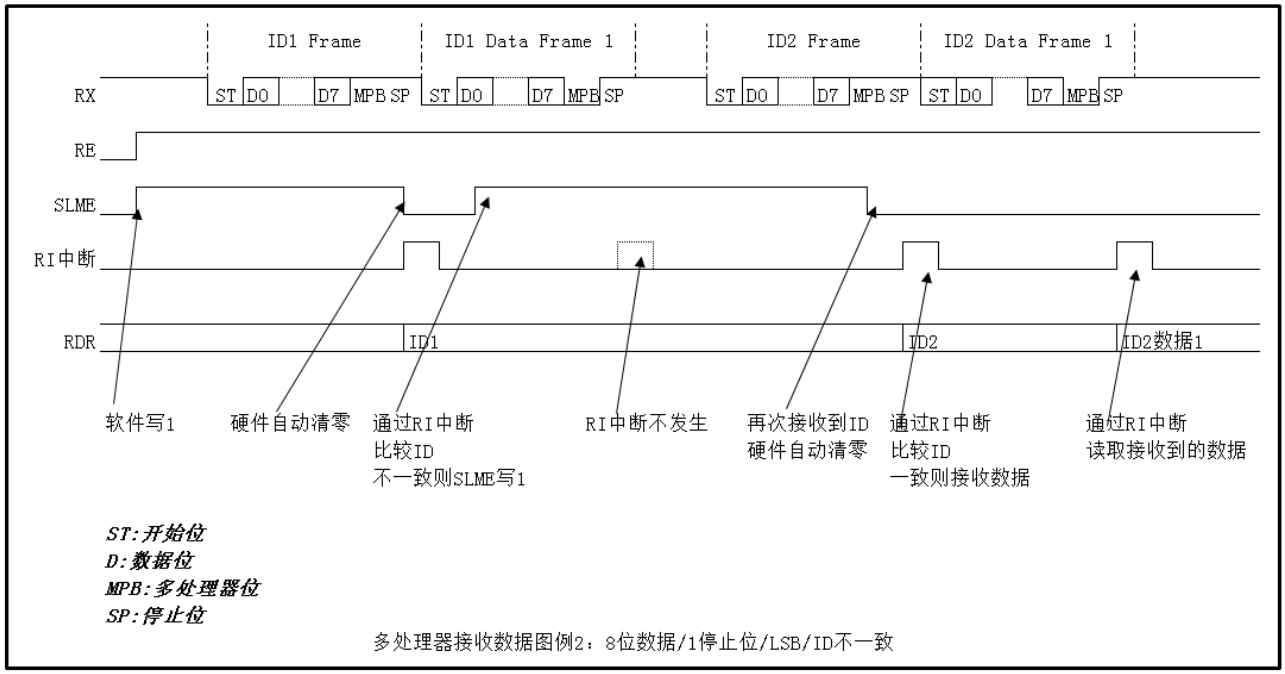


图 29-12 多处理器模式接收数据图例 2

29.4.2.4 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	使能位（仅中断）	标志	可否作为事件源
接收错误中断	RIE	ORE,FE	可
接收数据寄存器满中断	RIE	RXNE	可
发送数据寄存器空中断	TXEIE	TXE	可
发送完成中断	TCIE	TC	可

表 29-3 多处理器模式中断/事件表

29.4.3 UART-LIN

29.4.3.1 功能简介

LIN 是 Local Interconnect Network 的简称，是为降低汽车网络成本的低速（1~20kbps）串行通信协议。

29.4.3.2 LIN 数据格式

LIN 的一帧数据是由开始位+8 位数据+1 停止位组成，以 LSB 方式发送和接收数据。

LIN 总线上的数据行为如下图所示。

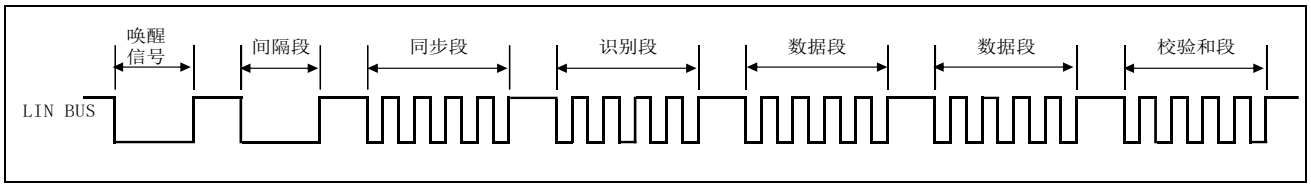


图 29-13 LIN 总线数据行为

29.4.3.3 LIN 发送

唤醒信号帧的发送

通过发送数据 0x80 对应信号帧的发送。

间隔段发送

同步间隔段表示一帧数据的开始，本产品硬件支持 LIN 间隔段的发送。

LIN 间隔段低电平的宽度由寄存器 USARTn_CR2.SBK 寄存器设定。

USARTn_CR2.SBKM=0 时，USARTn_CR2.SBK 写 1，硬件自动送出间隔段。

USARTn_CR2.SBKM=1 时，USARTn_CR2.SBK 写 1 后，写数据 0x00 到发送数据寄存器 USARTn_DR.TDR 开始发送间隔段。

间隔段发送完成后，USARTn_CR2.SBK 自动清零。

同步段发送

通过发送数据 0x55 实现同步段数据的发送

总线错误检测

USARTn_CR2.BEE=1 时，硬件检测到总线上的数据和发送的数据不一致时，置位 USARTn_SR.BE 标志，USARTn_CR2.BEIE=1 时，产生对应的中断。

29.4.3.4 LIN 接收

唤醒信号帧的检测

USARTn_CR2.WKUPE=1 时，硬件自动检测唤醒信号，检测到 RX 线低电平宽度上大于等于 2.5 位数据宽度（19.2Kbps 时为 130uS）时，置位 USARTn_SR.WKUP 标志，USARTn_CR2.WKUPIE=1 时，产生对应的中断。

USARTn_CR2.WKUPE 寄存器仅在系统需要等待唤醒信号时设定为 1，其它情况需要设定为 0。

同步间隔段的检测

本产品硬件支持同步间隔段检测，当检测到通信线低电平宽度大于等于 USARTn_CR2.LBDL 寄存器设定的值，并检测到 break delimeter 时，置位 USARTn_SR.LBD 标志，USARTn_CR2.LBDIE=1 时，产生对应的中断。

需要注意的是，同步间隔段检测时，会置位 USARTn_SR.FE 标志，清除 USARTn_SR.LBD 标志的同时，需要清除 USARTn_SR.FE 标志，并读取 USARTn_DR.RDR 寄存器（读取值为 0x00）。

同步段的检测与波特率测量

当检测到间隔段后，硬件自动测量同步段的频率。测量计数器的时钟由 USARTn_PR.LBMPSC 位设定，计数器的值保存在 USARTn_BTMC 寄存器中，选择的计数器时钟频率除以 USARTn_BTMC 寄存器值即可得到主节点的波特率。

需要注意的是，USARTn_BTMC 寄存器的读取需要在同步段接收完成后再读取。

接收数据设定步骤

1　将 USARTn_CR1 寄存器设定为复位值

2　设定 UART 所需要使用的管脚

3　通过 USARTn_CR2.CLKC[1:0]位选择时钟源

4　设定 USARTn_CR1(RE=1,RIE=0), USARTn_CR2，USARTn_CR3 寄存器

（以下步骤 5~6 为唤醒信号帧的检测，如无必要则跳过）

5　设定 USARTn_CR2.WKUP=1，等待唤醒信号

6　检测到唤醒信号，USARTn_CR2.WKUP=0，等待同步段间隔段

（以下步骤 7 为同步段间隔场检测，如无必要则跳过）

7　检测到同步间隔断后，清除 USARTn_SR.PE/FE/LBD 标志，并确认接收到的数据为 0x00。

（以下步骤 8~9 为同步段频率测量与通信波特率计算，如无必要则跳过）

8　如果需要使用接收中断，则设置 USARTn_CR1. RIE=1

9　检测到同步间隔场，并完成同步段接收后，读取 USARTn_BTMC 寄存器计算波特率，并确认接收到的数据为 0x55

（以下步骤 10~12 为接收识别段，数据以及校验和，过程与 UART 接收数据过程相同）

10 接收识别段

11 接收数据

12 接收校验和

29.4.3.5 LIN 中断和事件

功能名称	使能位（仅中断）	标志位	可否作为事件源
错误中断	RIE	ORE,FE	可
	BEIE	BE	可
唤醒信号/间隔段检测中断	WKUPIE	WKUP	可
	LBD	LDBIE	可
接收数据寄存器满中断	RIE	RXNE	可
发送数据寄存器空中断	TXEIE	TXE	可
发送完成中断	TCIE	TC	可

表 29-4　LIN 中断/事件表

29.4.4 智能卡

29.4.4.1 连接示意图

支持 ISO/IEC 7816-3 规定的智能卡通信协议。下图为智能卡模式的连接示意图。

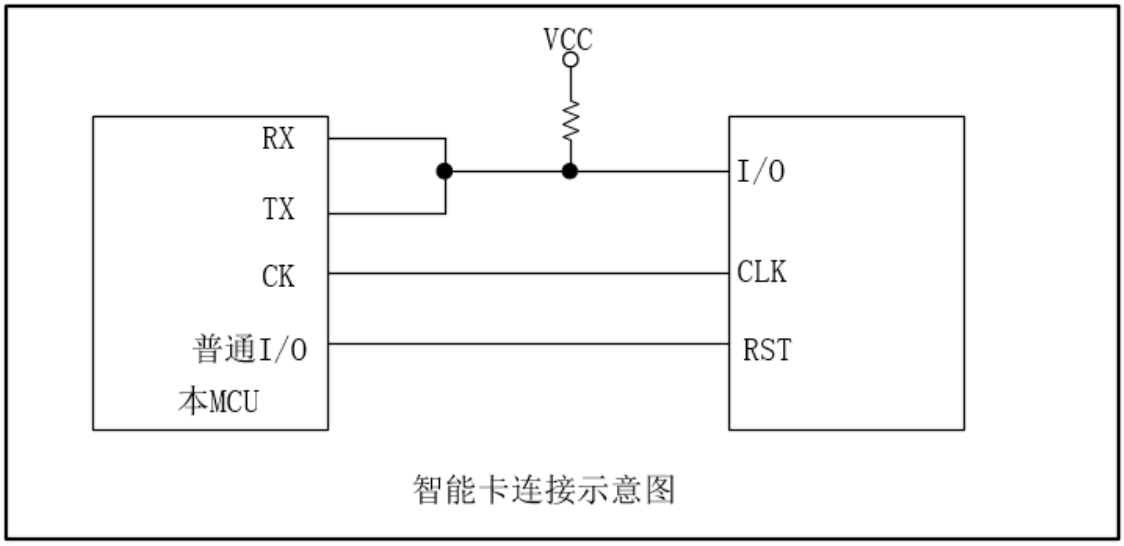


图 29-14 智能卡连接示意图

29.4.4.2 时钟

内部时钟源

智能卡模式时只能使用内部波特率生成器生成的时钟作为时钟源。

一位数据传输的基本时钟数为 USARTn_CR3.BCN[2:0]设定值。

通过设定寄存器 USARTn_CR2.CLKC[1:0]位控制智能卡模式的时钟输出。

最高波特率

内部时钟源时，内部波特率生成器生成的波特率计算公式为：

$$B = \frac{C}{2 \times BCN \times (DIV_Integer + 1)}$$

B：波特率 单位：MBps

C：USARTn_PR.PSC[1:0]位设定的时钟（PCLK,PCLK/4,PCLK/16,PCLK/64） 单位：MHz

DIV_Integer:USARTn_BRR.DIV_Integer 设定值

BCN：USARTn_CR3.BCN 寄存器设定值

当 C 为 PCLK, DIV_Integer=0，BCN=0 时，波特率为最高波特率为 PCLK/64(MBps)。

采样和接收容差

检测到 RX 的下降后，USART 会基于内部基本时钟对接收数据进行时钟同步，从而开始数据接收。接收数据将在数据中央被采样。

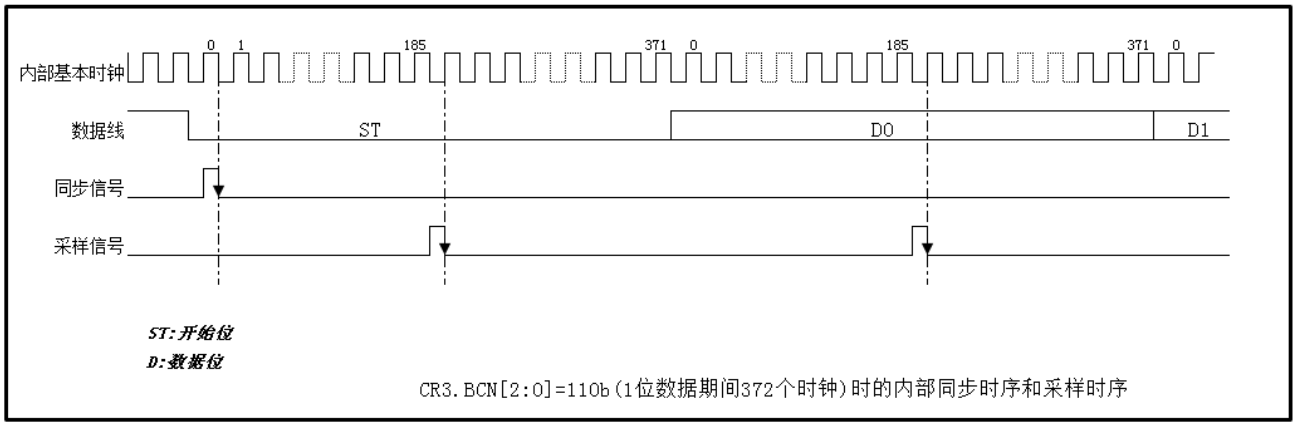


图 29-15　智能卡模式同步时序和采样时序图

接收容差的计算公式如下：

$$RM[\%] = |0.5 \times \left(1 - \frac{1}{BCN}\right) - 9.5CFD| \times 100$$

RM：接收容差

BCN：一位数据传输所需要的时钟数（USARTn.CR3.BCN[2:0]设定值）

CFD：时钟频率偏差

29.4.4.3 数据格式

智能卡模式时一帧数据由起始位，数据位和校验位组成。

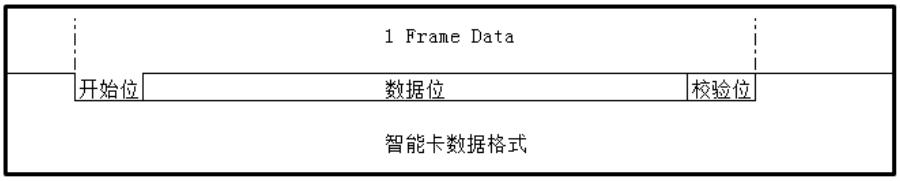


图 29-16　智能卡模式数据格式

起始位

起始位固定有一位低电平构成。

数据位

数据位固定为 8 位数据。

校验位

校验位需配置成 1 位偶校验。

29.4.4.4 智能卡的初始化设定步骤

1. 将 USARTn_CR1 寄存器设定为复位值
2. 设定所需要使用的管脚
3. 状态寄存器确认，USARTn_SR 寄存器设定为复位值
4. 设定 USARTn_CR1, USARTn_CR2，USARTn_CR3 寄存器
5. 设定 USARTn_PR 选择预分频值，USARTn_BRR 寄存器设定通信波特率
6. USARTn_CR2.CLKC[1:0]位设定时钟控制
7. USARTn_CR1 寄存器(TE,RE,RIE,TXEIE 位)设定，除了自测外，TE 和 RE 不要同时设定为 1

由发送模式切换到接收模式，或者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。

29.4.4.5 智能卡模式动作说明

智能卡模式时，TI 中断(发送数据空中断)的标志位为 USARTn_SR.TC 位。USARTn_SR.TC=1 而且 USARTn_CR1.TXEIE=1 时产生 TI 中断。

功能概述

发送数据时两帧数据间 (从校验位结束到下一帧的起始位开始)有 2etu(Elementary Time Unit)以上的保护时间。

发送数据时如果检测到接收方发送的错误信号则经过 2etu 后自动重发数据。

接收数据发生校验错误时，发送 1etu 的低电平即错误信号，错误信号送出的时序为接收开始经过 10.5etu。

发送说明

1. 一帧数据发送完成后，如果检测到接收方发送的错误信号，则 USARTn_SR.FE 置 1（如果 USARTn_CR.RIE=1,则发生错误中断），USARTn_SR.TC 标志不置 1，数据自动重发。USARTn_SR.FE 位必须在接受到下一帧校验位之前清零。
2. 一帧数据无错误发送完成后，USARTn_SR.TC 标志置位，USARTn_CR1.TXEIE=1 时发生 TI 中断。再次写入数据，则可实现连续发送数据。

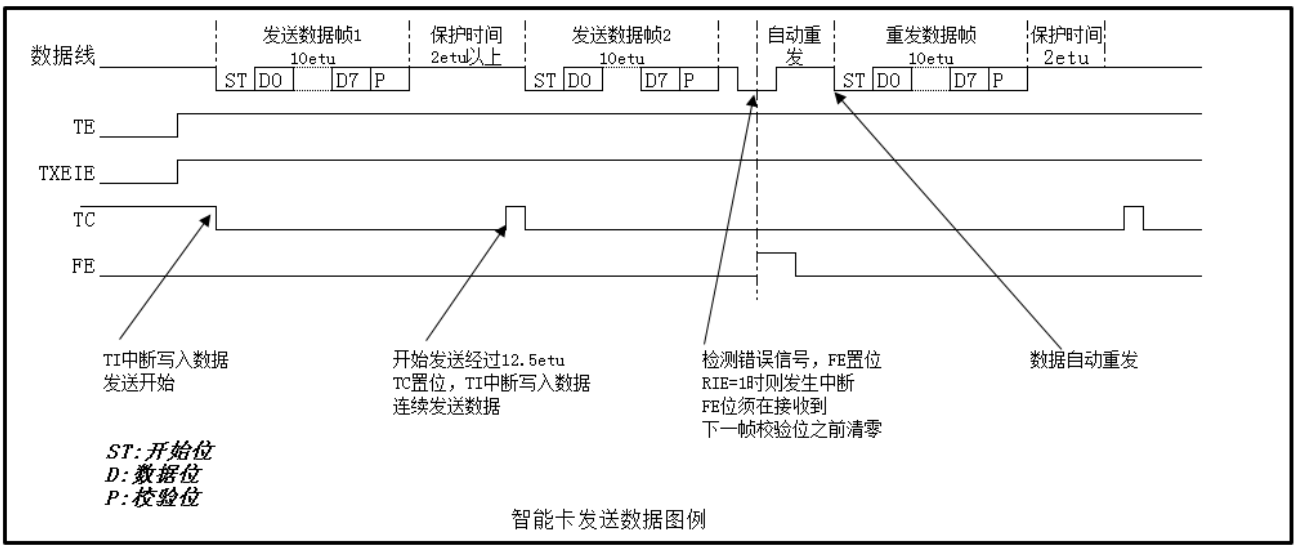


图 29-17 智能卡模式发送数据图例

接收说明

1. 接收数据时，如果检测到校验错误，则 USARTn_SR.PE 置位，中断使能时，发生 EI 中断。USARTn_SR.PE 位需要在接收到下一帧校验位之前清零。
2. 发生校验错误时，会发送 1etu 的低电平，即错误信号，要求发送方重新发送数据。
3. 正常接收数据，可以通过 RI 中断读取接收到的数据，连续接收。
4. 接收数据时，检测上溢错误。

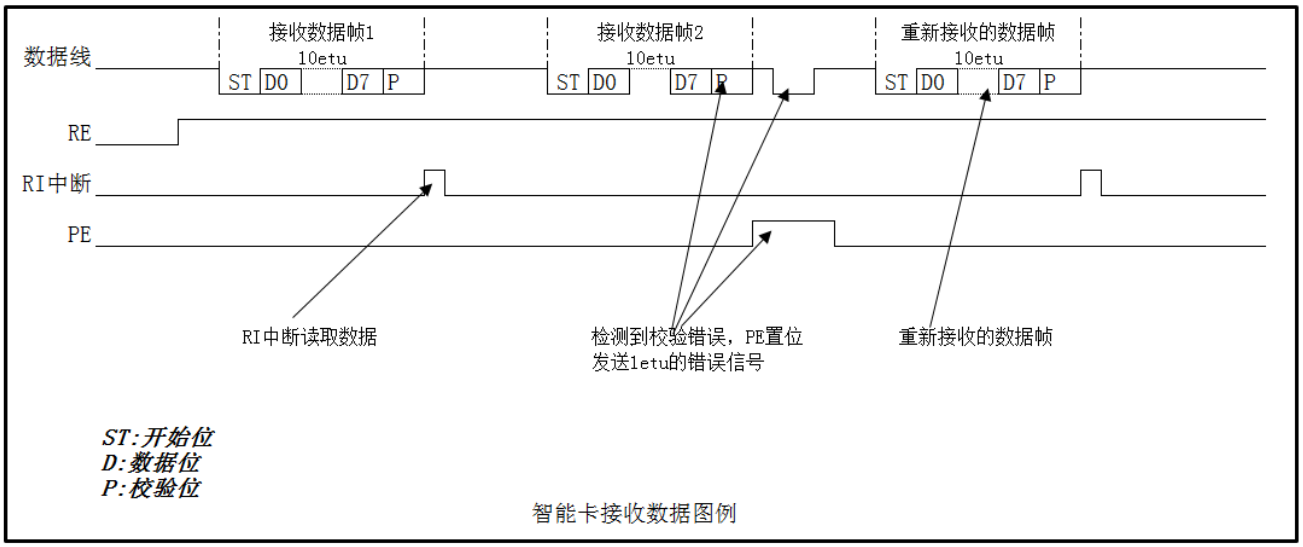


图 29-18 智能卡模式接收数据图例

29.4.4.6 中断和事件

功能名称	使能位（仅中断）	标志位	可否作为事件源
错误中断	RIE	ORE,PE,FE	可
接收数据满中断	RIE	RXNE	可
发送数据空中断	TXEIE	TC	可

表 29-5 智能卡模式中断/事件表

29.4.5 时钟同步模式

29.4.5.1 时钟

时钟同步模式可以选择内部波特率生成器生成的时钟(内部时钟源)或 USARTn_CK 管脚输入的时钟(外部时钟源)作为通信的时钟源。

内部时钟源

同步时钟从 USARTn_CK 管脚输出，一帧数据输出 8 个时钟脉冲，既不发送数据也不接收数据时，时钟输出固定为高电平。

外部时钟源

外部时钟源即从 USARTn_CK 管脚输入时钟作为通信时钟。

最高波特率

内部时钟源时，内部波特率生成器生成的波特率计算公式为：

$$B = \frac{C}{4 \times (DIV_Integer + 1)}$$

B：波特率 单位：MBps

C：USARTn_PR.PSC[1:0]位设定的时钟（PCLK,PCLK/4,PCLK/16,PCLK/64） 单位：MHz

DIV_Integer:USARTn_BRR.DIV_Integer 设定值

内部时钟源时，当 C 为 PCLK, DIV_Integer=1 时，最高波特率为 PCLK/8(MBps)。注意同步模式时 DIV_Integer 禁止设置为 0。

外部时钟源时，外部输入时钟的最大频率要求为 PCLK(MHz)/6，所以最高波特率为 PCLK/6(MBps)。

需要注意的是，同步模式最高通信波特除了以上描述的基于 PCLK 的计算方法外，还需要参考电气特性章节规定的最高通信波特率。

29.4.5.2 数据格式

时钟同步模式一帧数据固定有 8 位组成，一帧数据的发送和接收需要 8 个同步时钟脉冲。发送数据时数据在同步时钟的下降沿送出，接收数据时数据在同步时钟的上升沿被采样。

同步时钟在没有数据传输时固定为高电平，最后一位发送完后，通信线保持最后一位的值。

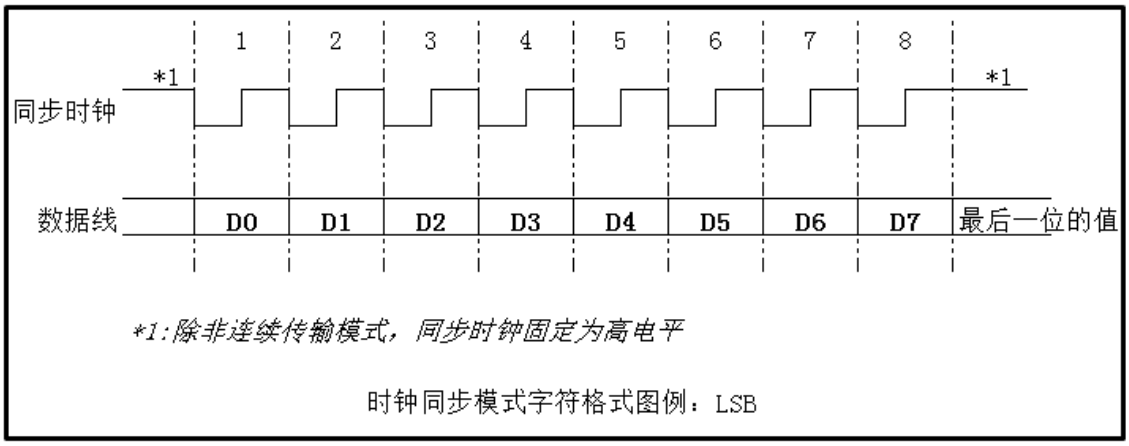


图 29-19 时钟同步模式数据格式

29.4.5.3 调制解调器操作

调制解调器操作包括 CTS 功能和 RTS 功能。USARTn_CR3.RTSE=1 时 RTS 功能有效，USARTn_CR3.CTSE=1 时 CTS 功能有效。

CTS 功能

CTS 功能是通过 USARTn_CTS 管脚的输入来控制数据的发送，只有当 USARTn_CTS 管脚输入低电平时才可以发送数据，发送数据过程中如果 USARTn_CTS 输入高电平，正在发送的数据不受影响。

RTS 功能

RTS 功能是指通过 USARTn_RTS 管脚输出低电平，请求对方发送数据。

USARTn_RTS 管脚输出低电平需要满足以下全部条件：

- 接收使能(USARTn_CR1.RE=1)，且不正在接收数据
- USARTn_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。

29.4.5.4 发送器

发送器使能位 (USARTn_CR1.TE) 置 1 时，发送移位寄存器中的数据在 USARTn_TX 管脚串行输出，相应的时钟脉冲在 USARTn_CK 管脚输出。

发送数据寄存器 USARTn_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
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。

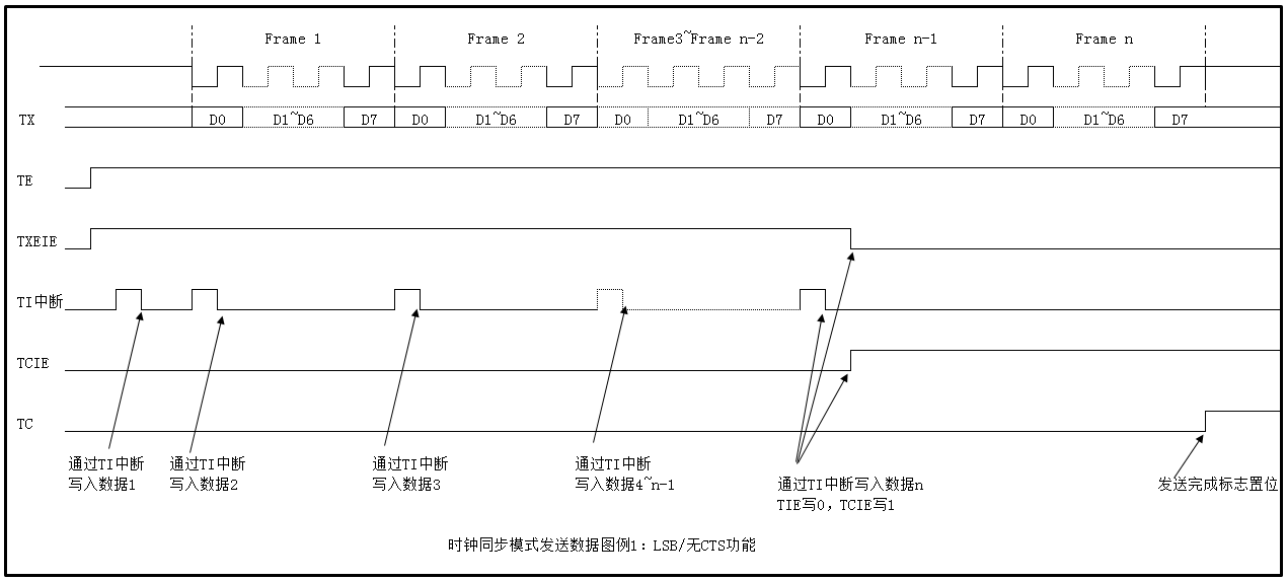


图 29-20 时钟同步模式发送数据图例 1

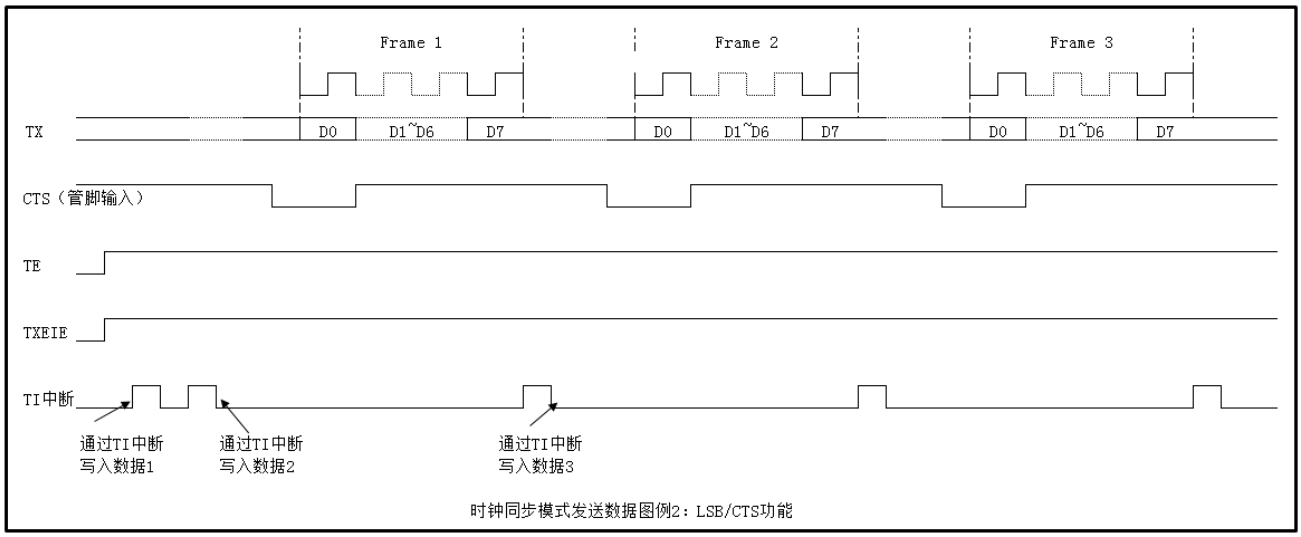


图 29-21 时钟同步模式发送数据图例 2

发送器中断

时钟同步模式发送器支持两种中断，发送数据寄存器空中断 TI 和发送完成中断 TCI。

TXEIE=1，USARTn_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。

29.4.5.5 接收器

接收数据设定步骤

1. 将 USARTn_CR1，USARTn_SR 寄存器设定为复位值

2. 设定所需要使用的管脚
3. 通过 USARTn_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn_CR1, USARTn_CR2，USARTn_CR3 寄存器
5. 设定 USARTn_PR 选择预分频值，USARTn_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. 使能接收器（USARTn_CR1. RE=1），如果需要使用接收中断，则设置 USARTn_CR1. RIE=1

   (使用 RTS 功能时，RE=1 后 USARTn_RTS 输出低电平)
7. 同步于输入同步时钟或者内部生成的同步时钟开始接收数据，接收数据到接收移位寄存器。
   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)

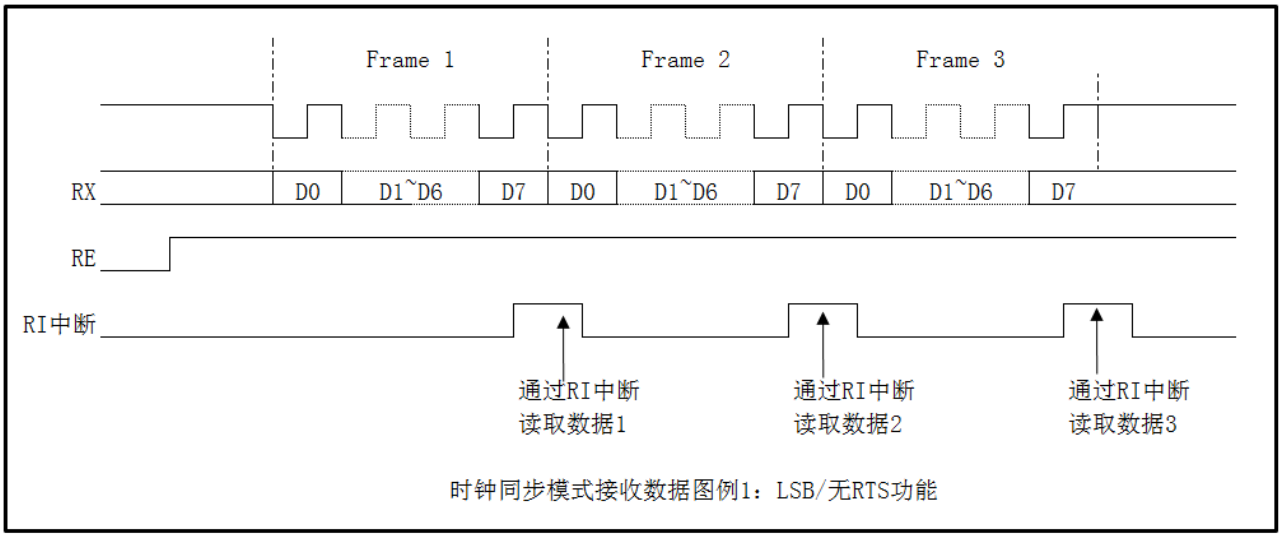


图 29-22 时钟同步模式接收数据图例 1

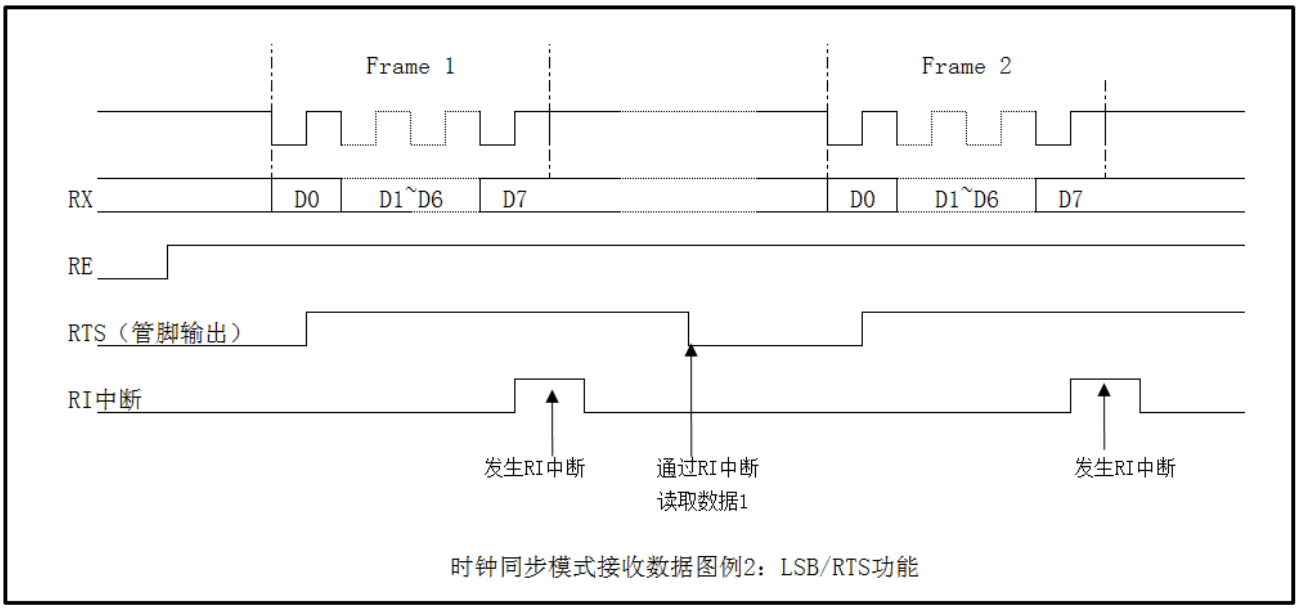


图 29-23 时钟同步模式接收数据图例 2

错误处理

时钟同步模式接收数据时接收错误为上溢错误（USARTn_SR.ORE）。发生接收错误不能再进行数据的接收和发送。可以通过将错误标志清零来重启数据传输，清除方法是写对应的清零寄存器。

上溢错误发生的条件是 USARTn_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 EI。

RIE=1，数据从接收移位寄存器传送到接收数据寄存器时 RI 中断发生。

RIE=1，接收数据发生错误（上溢错误）时 EI 中断发生。

29.4.5.6 同时发送接收数据

USART 时钟同步模式支持全双工动作，同时发送接收数据。同时发送接收数据时需要一个命令把 RE,TE,RIE,TXEIE 写 1，其它设定流程与发送器和接收器相同。

29.4.5.7 时钟同步模式中断和事件

中断名称	使能位(仅中断)	标志位	可否作事件源
接收错误中断	RIE	ORE	可
接收数据寄存器满中断	RIE	RXNE	可
发送数据寄存器空中断	TXEIE	TXE	可
发送完成中断	TCIE	TC	可

表 29-6 时钟同步模式中断/事件表

29.4.6 数字滤波功能

USARTn_CR1.NFE=1 时，内置数字滤波器功能有效。数字滤波器仅在 UART 模式时有效,可以除去接收数据线 RX 上的噪音。

内置数字滤波器可以滤除小于一位数据的 3/16(USARTn_CR1.OVER8=0)宽度或者 3/8 宽度(USARTn_CR1.OVER8=1)的噪音。

如果数字滤波器的时钟停止后再开始时，数字滤波器从时钟停止时保持的状态继续工作。

USARTn_CR.TE=0 且 USARTn_CR.RE=0，将数字滤波器内部的 Flip-Flop 状态复位为 1。

29.4.7 中断

下表给出 USART 整体中断：

中断名称	符号	说明
错误中断	USART_n_EI n=1~10	UART/多处理器/智能卡/时钟同步模式接收错误中断 LIN错误中断
接收数据寄存器满中断	USART_n_RI n=1~10	UART/多处理器/LIN/智能卡/时钟同步模式接收数据寄存器满中断
发送数据寄存器空中断	USART_n_TI n=1~10	UART/多处理器/LIN/智能卡/时钟同步模式发送数据寄存器空中断
发送完成中断	USART_n_TCI n=1~10	UART/多处理器/LIN/智能卡/时钟同步模式发送完成中断
唤醒信号/间隔段检测中断	USART_n_BRKWKPI n=5/10	LIN 唤醒信号/间隔段检测中断
UART接收TIMEOUT中断	USART_n_RTO n=1/2/6/7	UART接收TIMEOUT中断
RX线唤醒停止模式中断	USART_1_WUPI	USART_1 RX线唤醒停止模式中断

表 29-7 USART 整体中断一览表

29.5 寄存器说明

本章将详细描述 USART 模块控制状态等相关寄存器，需要注意的是功能寄存器位只有该通道配置了对应的功能才有效，否则默认为 0。

基地址如下：

USART_1_BASE_ADDR:0x4001CC00

USART_2_BASE_ADDR:0x4001D000

USART_3_BASE_ADDR:0x4001D400

USART_4_BASE_ADDR:0x4001D800

USART_5_BASE_ADDR:0x4001DC00

USART_6_BASE_ADDR:0x40020C00

USART_7_BASE_ADDR:0x40021000

USART_8_BASE_ADDR:0x40021400

USART_9_BASE_ADDR:0x40021800

USART_10_BASE_ADDR:0x40021C00

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 FF00(USART_5/10) 0x0000 FFFF (USART_1/2/3/4/6/7/8/9)
控制寄存器1（USART_CR1）	0x0C	0x8000 0000
控制寄存器2（USART_CR2）	0x10	0x0000 0600
控制寄存器3（USART_CR3）	0x14	0x0000 00000
预分频寄存器（USART_PR）	0x18	0x0000 0000
LIN波特率测量计数寄存器（USART_LBMC）	0x1C	0x0000 0000

表 29-8 USART 寄存器一览表

29.5.1 状态寄存器（USART_SR）

USART Status Register

偏移地址：0x00

复位值：0x000000C0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	MPB
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	LBD	WKUP	RTOF	TXE	TC	RXNE	BE	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/W
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~b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10	LBD	LIN 间隔段检出标志	LIN 间隔段检出标志（仅USART_5/USART_10有效） 0：未检测到间隔段 1：检测到间隔段 LBD置位条件 CR2.WKUPE=0时，检测到RX线低电平宽度上大于等于CR2.LBDL寄存器设定的宽度后，并检测到break delimeter时 清零条件 清零寄存器CR1.CLBD位写入1 注意：RE=0并不能复位LBD位	R
b9	WKUP	LIN Wakeup信号标志	LIN Wakeup信号检出标志（仅USART_5/USART_10有效） 0：未检测到Wakeup信号 1：检测到Wakeup信号 WKUP置位条件 CR2.WKUPE=1时，检测到RX线低电平宽度上大于等于2.5位数据宽度（19.2Kbps时为130uS） 清零条件 清零寄存器CR1.CWKUP位写入1 注意：RE=0并不能复位WKUP位 检测到WKUP信号为1后，表示本LIN节点已经从休眠模式退出，设定寄存器CR2.WKUPE=0后，进行后续间隔场检测等操作。	R
b8	RTOF	UART接收TIMEOUT标志位	UART接收TIMEOUT标志位（仅USART_1/USART_2/ USART_6/ USART_7有效） 0：无UART接收TIMEOUT	R

			1：发生UART接收TIMEOUT RTOF置位条件 ● 从检测到最后一帧数据的STOP位起经过设定时间后没有检测到新的接收数据 RTOF清零条件 ● 清零寄存器CR1.CRTOF位写入 注意：RTOF为由硬件置1，且只有在CR1.RE=1且CR1.RTOE=1时置1。CR1.RE=0时，TIMEOUT功能有效，但RTOF不置1。	
b7	TXE	发送数据寄存器空	发送数据寄存器空标志 TXE位在UART/时钟同步模式/LIN模式时有效。 0：数据未传输到移位寄存器，发送数据寄存器非空 1：数据传输到移位寄存器，发送数据寄存器空 注意：TXE位由硬件置1和清0，数据未传输到移位寄存器时硬件将TXE清0，数据传输到移位寄存器时硬件将TXE置1	R
b6	TC	发送完成标志	发送完成标志位 0：发送数据中 1：发送数据完成 UART模式，时钟同步模式 TC置位条件 ● TE=0发送禁止时 ● 送出一帧数据的最后一位时，发送数据寄存器的值没有被更新 TC清零条件 ● TE=1时，向发送数据寄存器写入发送数据 智能卡模式 TC置位条件 ● TE=0发送禁止时 ● 最后1字节的数据送出后经过特定时间后，FE=0而且发送数据寄存器的值没有被更新。 TC置位的具体时序为：校验位送出后经过2.5位时间 TC清零条件 ● TE=1时，向发送数据寄存器写入发送数据 注意：TE位由0变化为1时TC为保持为1	R
b5	RXNE	接收数据寄存器不为空	接收数据寄存器不为空标志 0：未接收到数据 1：准备好读取接收到的数据 注意：RXNE位由硬件置1和清0，准备好读取接收到的数据时硬件将RXNE置1，读取接收数据后硬件将RXNE清0	R
b4	BE	LIN总线检错误标志	LIN总线检错误标志 0：无总线错误 1：发生总线错误 BE置位条件 ● 发送数据时，检测到总线数据和发送数据不一致,检测位置为每一位的13/16处（16位过采样）	R

			BE清零条件 ● 清零寄存器CR1.CBE位写入1 注意：TE=0并不能复位BE位 BE=1时，正在发送的数据发送完成后，不会开始新的数据发送，需要软件清除BE标志，重启发送动作	
b3	ORE	接收上溢错误	接收上溢错误标志位 0：无接收上溢错误 1：发生接收上溢错误 ORE置位条件 ● 接收数据寄存器未被读取的情况下，又接收到一帧新的数据 ORE清零条件 ● 清零寄存器CR1.CORE位写入1 注意：RE=0并不能复位ORE位 ORE=1之前接收到的数据会被保持，ORE=1时接收到的数据会丢掉 ORE=1后不能继续接收数据，时钟同步模式下也不能发送数据	R
b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	FE	接收帧错误	接收帧错误标志位 0：无接收帧错误 1：发生接收帧错误 UART模式 FE置位条件 ● 接收的数据帧的停止位为低电平，两个停止位的情况只检查第一个停止位 FE清零条件 ●清零寄存器CR1.CFE位写入1 注意：UART模式时，RE=0并不能复位PE位 FE=1时收到的数据会保留但是RI中断不会发生，FE=1后不能继续接收数据 智能卡模式 FE置位条件 ● 采样到低电平错误信号标志 FE清零条件 ●清零寄存器CLR.CFE位写入1 注意：智能卡模式时，RE=0并不能复位FE位	R
b0	PE	接收数据校验	接收数据校验错误标志 0：无接收数据校验错误 1：发生接收数据校验错误 PE置位条件 ● 接收数据发生奇偶校验错误时 PE清零条件 ●清零寄存器CR1.CPE位写入1 注意：RE=0并不能复位PE位 PE=1时收到的数据会保留但是RI中断不会发生，PE=1后不能继续接收数据	R

29.5.2 数据寄存器（USART_DR）

USART Data Register

偏移地址：0x04

复位值：0x000001FF

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
-	-	-	-	-	-	MPID	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/W
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/W
b9	MPID	多处理器模式ID位	多处理器模式时，发送通信数据或者发送ID的选择位 0：发送数据 1：发送ID 注意：MPID位只在多处理器模式时有效,其他模式须设定为复位值	R/W
b8~b0	TDR[8:0]	发送数据寄存器	发送数据寄存器 注意：最高位TDR[8]仅在UART模式而且数据长度设定为9位时有效	R/W

29.5.3 波特率寄存器（USART_BRR）

USART Bit Rate Register

偏移地址：0x08

复位值：0x0000FF00/0x0000FFFF（详见寄存器列表）

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/W
b15~b8	DIV_Integer[7:0]	整数分频寄存器	整数分频寄存器 注意：DIV_Integer[7:0]只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b7	Reserved	-	写入时需写入复位值	R/W
b6~b0	DIV_Fraction[6:0]	小数分频寄存器	小数分频寄存器 注意：DIV_Fraction[6:0]只能在TE=0&RE=0(发送/接收禁止)时设定，而且仅在FBME=1时设定值有效 该寄存器USART_5/USART_10时无效，读出为0	R/W

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 29-9 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 29-10 波特率计算公式（小数波特率有效 FBME=1）

29.5.4 控制寄存器 1（USART_CR1）

USART Control Register 1

偏移地址：0x0C

复位值：0x80000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SBS	NFE	FBME	ML	-	-	-	MS	CLBD	CWKUP	CBE	CRTOF	CORE	-	CFE	CPE
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OVER8	-	-	M	-	PCE	PS	-	TXEIE	TCIE	RIE	SLME	TE	RE	RTOIE	RTOE

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 该寄存器USART_5/USART_10时无效，读出为0	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/W
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	CLBD	LBD清零位	LBD清零位（仅USART_5/USART_10有效） 0:不清零LBD标志	R/W

			1:清零LBD标志 注意：CLBD位写1清除LBD标志，读取时返回0	
b22	CWKUP	WKUP清零位	WKUP清零位（仅USART_5/USART_10有效） 0:不清零WKUP标志 1:清零WKUP标志 注意：CWKUP位写1清除WKUP标志，读取时返回0	R/W
b21	CBE	BE清零位	BE清零位（仅USART_5/USART_10有效） 0:不清零BE标志 1:清零BE标志 注意：CBE位写1清除BE标志，读取时返回0	R/W
b20	CRTOF	RTOF清零位	RTOF清零位（仅USART_1/USART_2/ USART_6/USART_7有效） 0:不清零RTOF标志 1:清零RTOF标志 注意：CRTOF位写1清除RTOF标志，读取时返回0	R/W
b19	CORE	ORE标志清零位	ORE标志清零位 0：不清零ORE标志 1：清零ORE标志 注意：CORE位写1清除ORE标志，读取时返回0	R/W
b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17	CFE	FE标志清零位	FE标志清零位 0:不清零FE标志 1:清零FE标志 注意：CFE位写1清除FE标志，读取时返回0	R/W
b16	CPE	PE标志清零位	PE标志清零位 0:不清零PE标志 1:清零PE标志 注意：CPE位写1清除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/W
b10	PCE	校验使能位	UART模式时，奇偶校验使能位 0：无校验 1：校验	R/W

			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	
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/W
b7	TXEIE	发送数据寄存器空中断使能位	发送数据寄存器空中断使能位 0：TI中断请求无效，TI中断不发生 1：TI中断请求有效，TI中断发生 注意：TE=0时写TXEIE=1，则需要等待TE=1时发生TI中断 TE=1时写TXEIE=1，则需要等待SR.TC=1时才能写入	R/W
b6	TCIE	发送完成中断使能位	发送完成中断使能位 0：TCI中断请求无效，TCI中断不发生 1：TCI中断请求使能，TCI中断发生	R/W
b5	RIE	接收中断使能位	接收中断使能位 0：接收中断请求无效，RI和EI中断不发生 1：接收中断请求有效，RI和EI中断发生	R/W
b4	SLME	静默模式使能位	多处理器操作时，静默模式使能位 0：正常模式 1：静默模式 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（仅USART_1/USART_2/ USART_6/USART_7有效） 0：UART TIMEOUT中断请求无效，RTOI中断不发生 1：UART TIMEOUT中断请求有效，RTOI中断发生	R/W
b0	RTOE	UART TIMEOUT功能使能位	UART TIMEOUT功能使能位（仅USART_1/USART_2/ USART_6/USART_7有效） 0：UART TIMEOUT功能禁止 1：UART TIMEOUT功能使能	R/W

29.5.5 控制寄存器 2（USART_CR2）

USART Control Register 2

偏移地址：0x10

复位值：0x00000600

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	LINEN	STOP	CLKC[1:0]		-	-	WKUPE	SBKL[1:0]		LBDL	LBDIE	BEE	BEIE	WKUPIE	MPE

位	标记	位名	功能	读写
b31~b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17	SBKM	间隔段发送模式	间隔段发送模式选择（仅USART_5/USART_10有效） 0：SBK位写1自动送出间隔段 1：SBK位写1后，写0x00到USARTn_DR.TDR寄存器，开始发送间隔段 注意：时钟同步模式时RE位只能在TE=0&RE=0(发送/接收禁止)时写1	R/W
b16	SBK	间隔段发送使能位	间隔段发送使能位（仅USART_5/USART_10有效） 0：不发送间隔段 1：根据SBKM位设定发送间隔段 SBK SBKM 间隔段发送 0 0 不发送 0 1 不发送 1 0 SBK写1时，自动发送间隔段，发送完成后，SBK自动清零 1 1 写数据0x00到USARTn_DR.TDR，开始发送间隔段，发送完成后，SBK自动清零	R/W
b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14	LINEN	LIN功能使能	UART模式时，LIN功能使能位（仅USART_5/USART_10有效） 0：禁止LIN功能 1：使能LIN功能 注意：LINEN位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b13	STOP	停止位设定位	UART模式时，停止位长度设定位 0：1停止位 1：2停止位 注意：非UART模式时STOP位必须保持复位值 STOP位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b12~b11	CLKC[1:0]	时钟控制位	**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USATRTn_CK管脚，USARTn_CK管脚可以当作普通IO使用	R/W

			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	
b10~b9	Reserved	-	需保持复位值11b,写入时写11b	R/W
b8	WKUPE	LIN唤醒信号检测使能位	LIN唤醒信号检测使能位（仅USART_5/USART_10有效） 0：禁止唤醒信号检测 1：使能唤醒信号检测 注意：仅在系统需要唤醒信号检测时，设置WKUPE为1，检测到唤醒信号后，需要软件清零WKUPE	R/W
b7~b6	SBKL[1:0]	LIN间隔段低电平宽度设定位	LIN间隔段低电平宽度设定位（仅USART_5/USART_10有效） 00b：10位 01b：11位 10b：13位 11b：14位 注意：非UART-LIN模式时SBKL位必须保持复位值 SBKL位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b5	LBDL	LIN间隔段低电平宽度检测阈值设定位	LIN间隔段低电平宽度检测阈值设定位（仅USART_5/USART_10有效） 0：≥10位 1：≥11位 注意：非UART-LIN模式时LBDL位必须保持复位值 LBDL位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b4	LBDIE	LIN间隔段检测中断使能位	LIN间隔段检测中断使能位（仅USART_5/USART_10有效） 0：LIN间隔段检测中断请求无效，不发生中断 1：LIN间隔段检测中断请求无效，发生中断 注意：非UART-LIN模式时LBDIE位必须保持复位值	R/W
b3	BEE	LIN总线错误检测功能使能位	LIN总线错误检测功能使能位（仅USART_5/USART_10有效） 0：禁止LIN总线错误功能检测功能 1：使能LIN总线错误功能检测功能 注意：非UART-LIN模式时BEE位必须保持复位值 BEE位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b2	BEIE	LIN总线错误中断使能位	LIN总线错误中断使能位（仅USART_5/USART_10有效） 0：LIN总线错误中断请求无效，不发生中断	R/W

			1：LIN总线错误中断请求无效，发生中断 注意：非UART-LIN模式时LBDIE位必须保持复位值	
b1	WKUPIE	LIN唤醒信号检测中断使能位	LIN唤醒信号检测中断使能位（仅USART_5/USART_10有效） 0：LIN唤醒信号检测中断请求无效，不发生中断 1：LIN唤醒信号检测中断请求无效，发生中断 注意：非UART-LIN模式WKUPIE位必须保持复位值	R/W
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

29.5.6 控制寄存器 3（USART_CR3）

USART Control Register 3

偏移地址：0x14

复位值：0x00000000

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

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~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	R/W
			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	
			USART_5/USART_10不支持该功能，读出为0	
b20~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9	CTSE	CTS功能使能位	CTS功能使能位	R/W
			0：CTS功能无效	
			1：CTS功能有效	
			注意：CTSE位只能在TE=0&RE=0(发送/接收禁止)时设定	
b8	RTSE	RTS功能使能位	RTS功能使能位	R/W
			0：RTS功能无效	
			1：RTS功能有效	
			注意：RTSE位只能在TE=0&RE=0(发送/接收禁止)时设定	
b7~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5	SCEN	智能卡模式使能位	智能卡模式使能位	R/W
			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 USART_5/USART_10不支持该功能，读出为0	
b4	LOOP	回环模式使能位	回环模式使能位 0：正常模式 1：回环模式 注意：LOOP位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b3	HDSEL	UART单线半双工模式使能位	UART单线半双工模式使能位 0：UART全双工模式 1：UART半双工模式 注意：HDSEL位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b2~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

29.5.7 预分频寄存器（USART_PR）

USART prescaler register

偏移地址：0x18

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	LBMPSC[1:0]		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/W
b3~b2	LBMPSC[1:0]	LIN波特率测量计数器时钟源选择	LIN波特率测量计数器时钟源选择（仅USART_5/USART_10有效） 00：PCLK 01：PCLK/2 10：PCLK/4 11：PCLK/8 注意：LBMPSC[1:0]位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W
b1~b0	PSC[1:0]	预分频器值	内部时钟源时，预分频器分频值选择位 00：PCLK 01：PCLK/4 10：PCLK/16 11：PCLK/64 注意：PSC[1:0]位只能在TE=0&RE=0(发送/接收禁止)时设定	R/W

29.5.8 LIN 波特率测量计数寄存器（USART_LBMC）

USART LIN Baudrate Measuring Counter

偏移地址：0x1C

复位值：0x00000000

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b0	LBMC[15:0]	LIN波特率测量结果计数器	LIN波特率测量结果计数器（仅USART_5/USART_10有效） LIN从节点，用USARTn_PR.LBMPSC寄存器选择的时钟源去测量得到的主节点发送的同步段的频率。 USARTn_PR.LBMPSC选择的计数时钟频率除以LBMC的值，即可得到测量到的LIN通信波特率。 注意：LBMC的值在同步段接收完成后读取才有意义	R

29.5.9 USART1 滤波控制寄存器（USART1_NFC）

USART1 Noise Filting Control Counter

地址：0x4005541C

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	USART1_NFE	USASRT1_NFS[1:0]	

位	标记	位名	功能	读写
b31~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	USART1_NFE	USART1滤波使能寄存器	USART_1滤波使能寄存器 该寄存器用于控制STOP模式下，RX线上的模拟滤波器的开关 0：模拟滤波器关 1：模拟滤波器开，滤波范围请参考USART1_NFS[1:0]的设定	R/W
b1~b0	USART1_NFS	USART1滤波选择寄存器	USART_1滤波选择寄存器 该寄存器用于控制STOP模式下，RX线上的模拟滤波器的滤波范围 00b：滤波宽度档位1 01b：滤波宽度档位2 10b：滤波宽度档位4 11b：滤波宽度档位4 各档位具体值请参考“电气特性 USART1 STOP模式下RX滤波特性”章节。	R/W

29.6 使用注意事项

29.6.1 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。

29.6.2 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。

29.6.3 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

30 集成电路总线（I²C）

30.1 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。

30.2 I²C 系统框图

30.2.1 系统框图

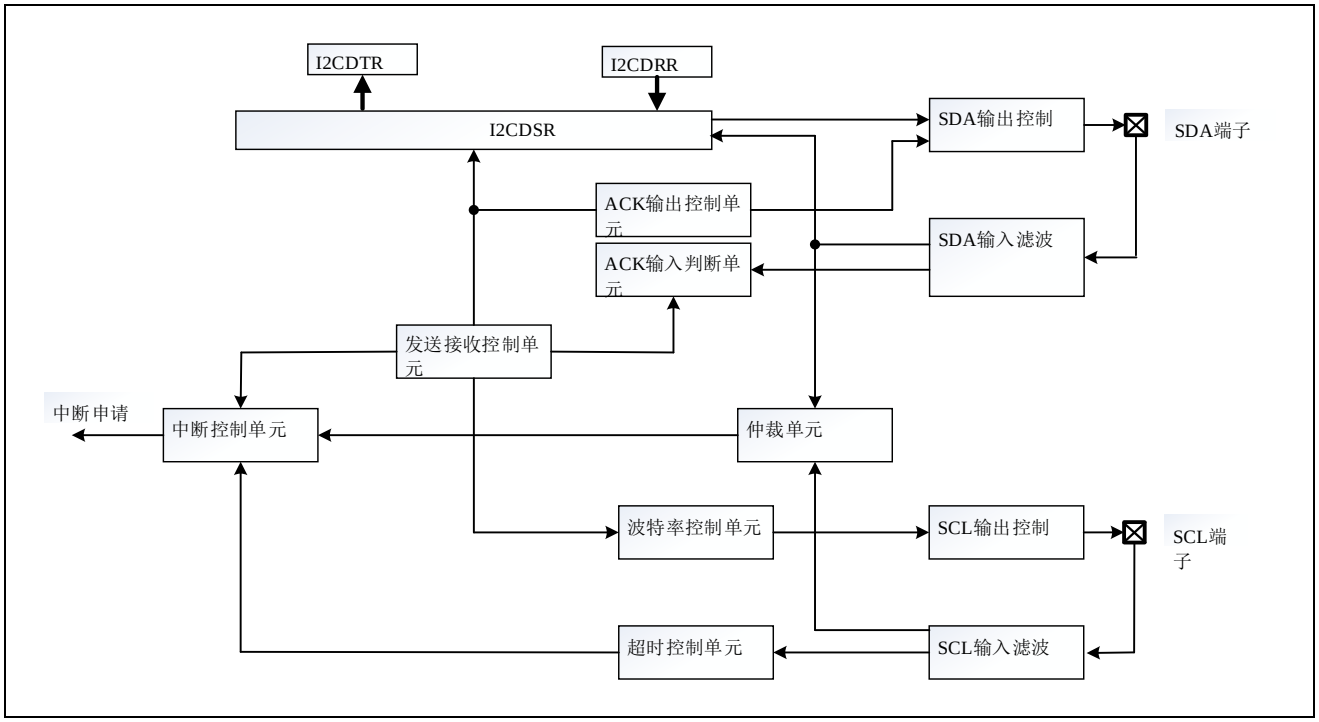


图 30-1 I²C 系统框图

30.2.2 结构图

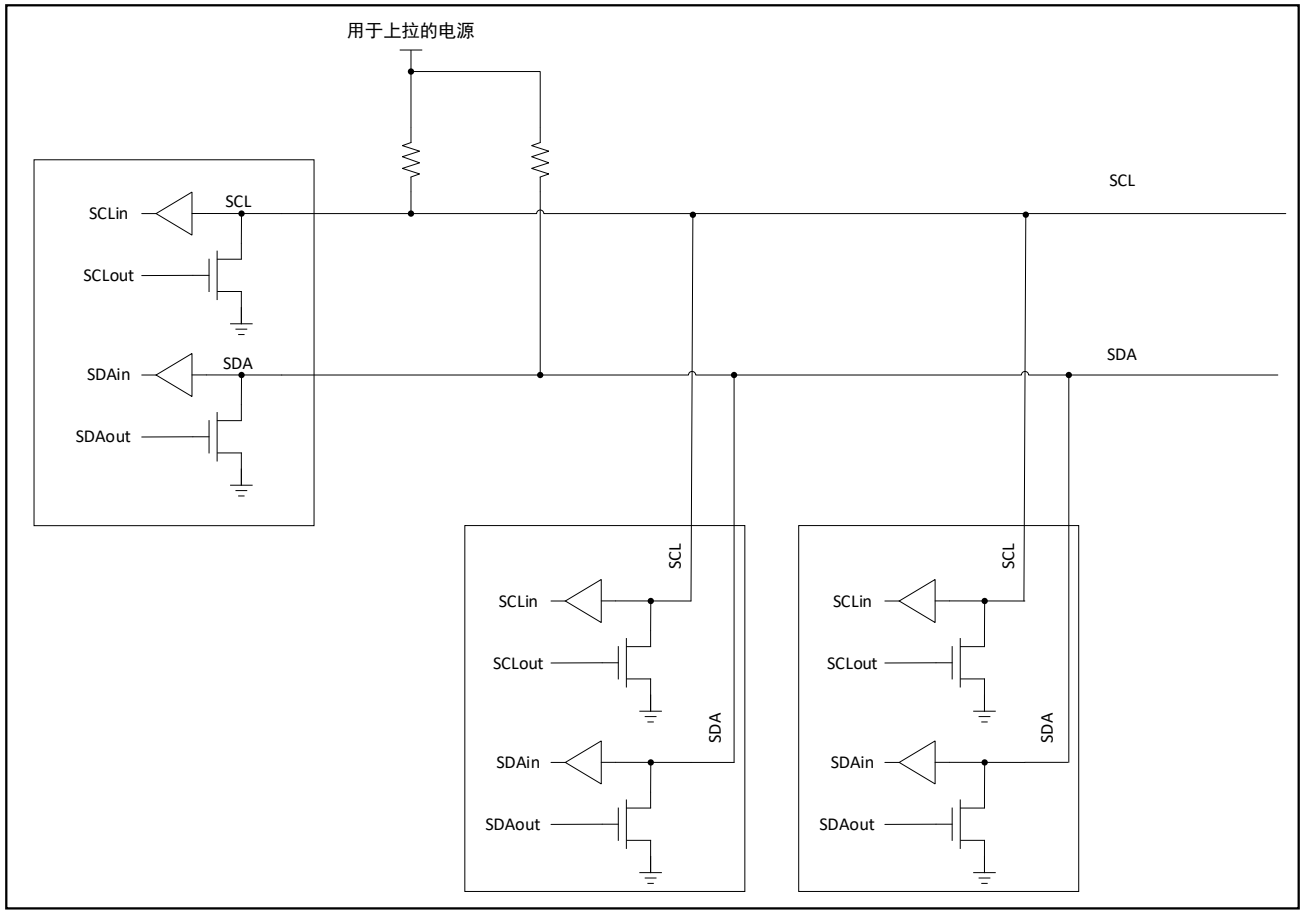


图 30-2 I2C 总线的结构例

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 30-1 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。

30.3 动作说明

本节提供了 I²C 模块功能的描述。

30.3.1 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。

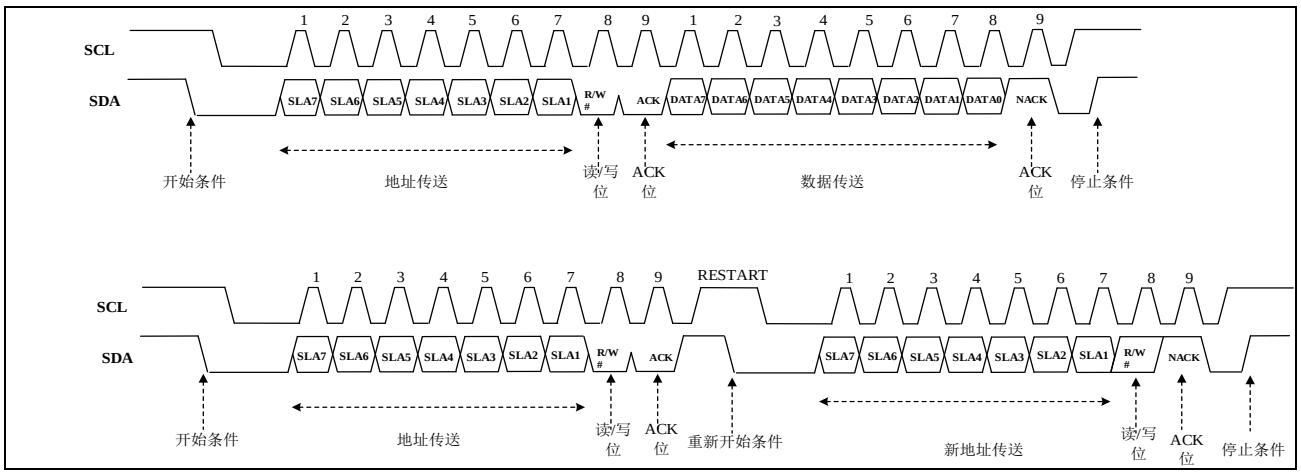


图 30-3 I²C 总线的时序图

30.3.1.1 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。

30.3.1.2 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 30-4 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。

30.3.1.3 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。

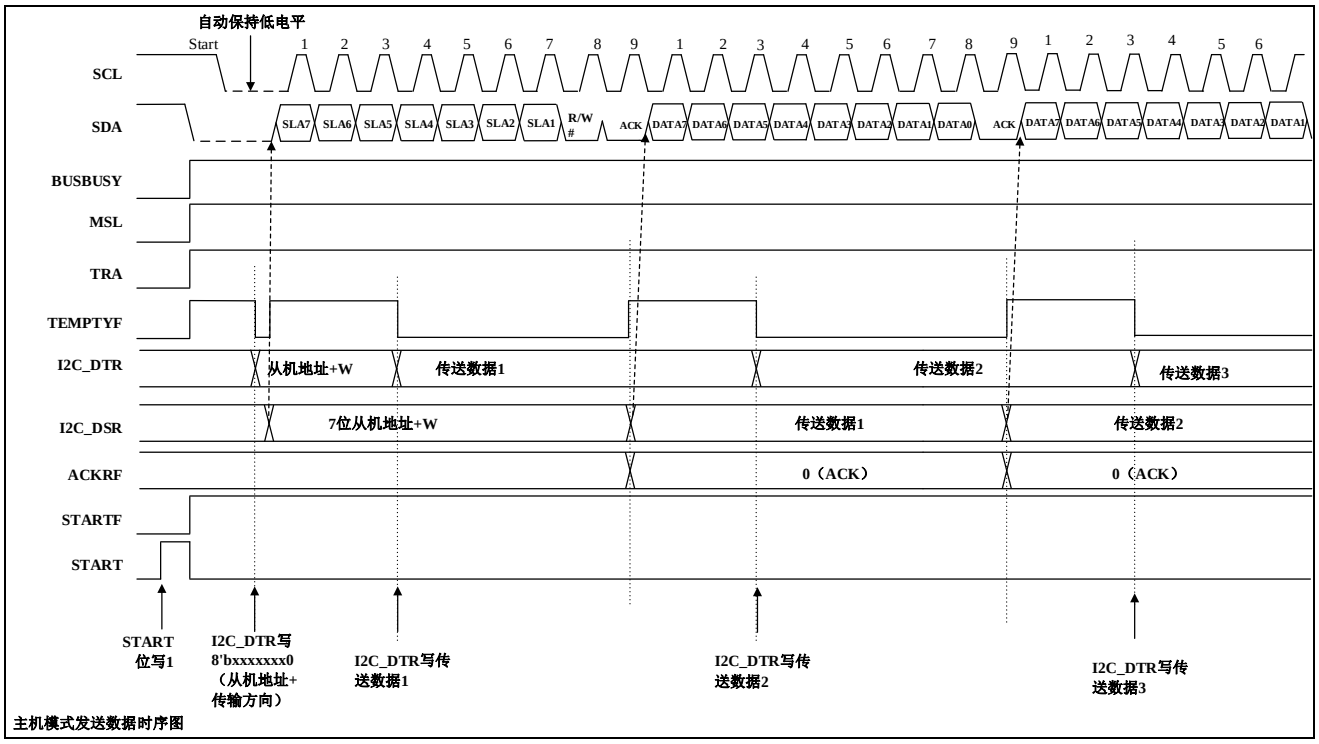


图 30-5　7 位地址格式的主机发送数据时序图（例）

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。

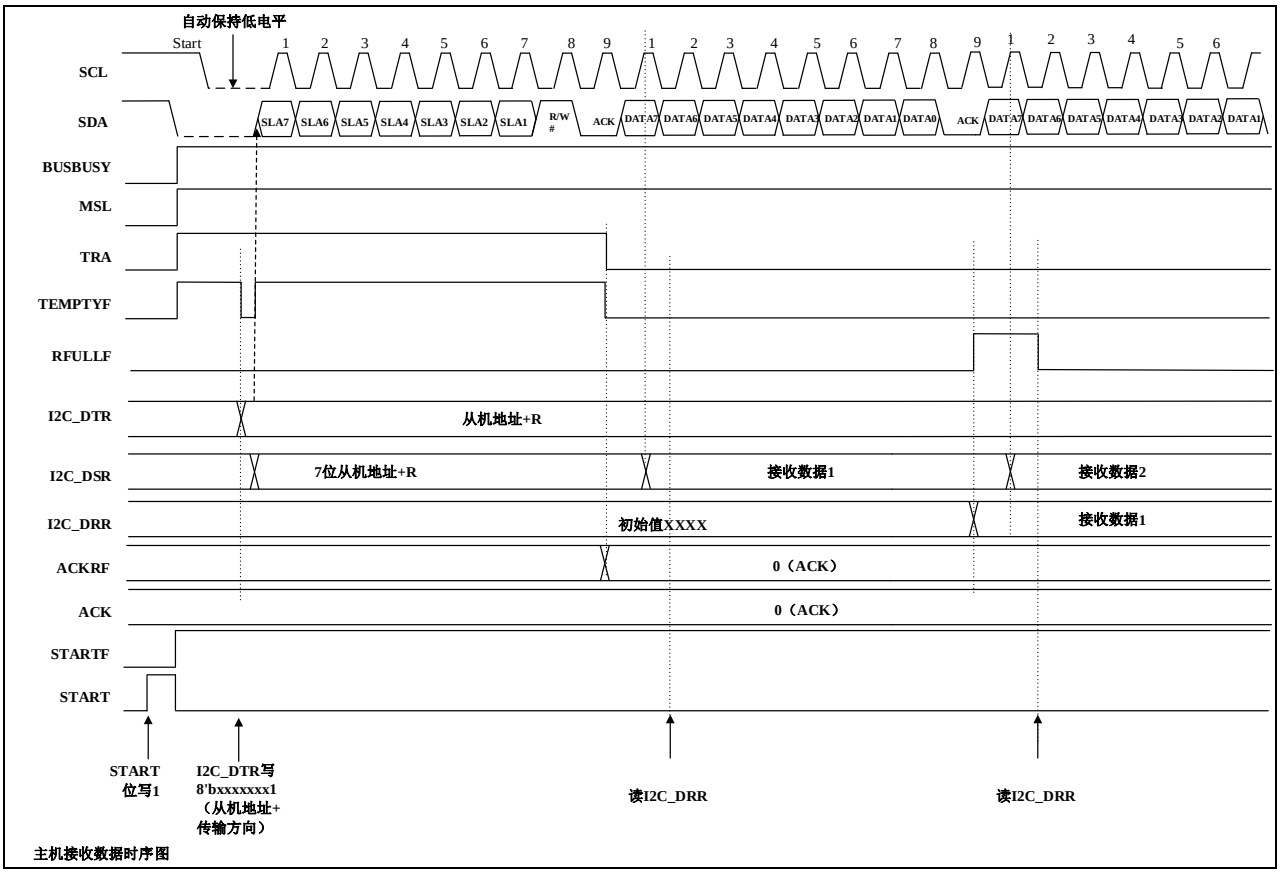


图 30-6　7 位地址格式的主机接收数据的时序图（例）

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。

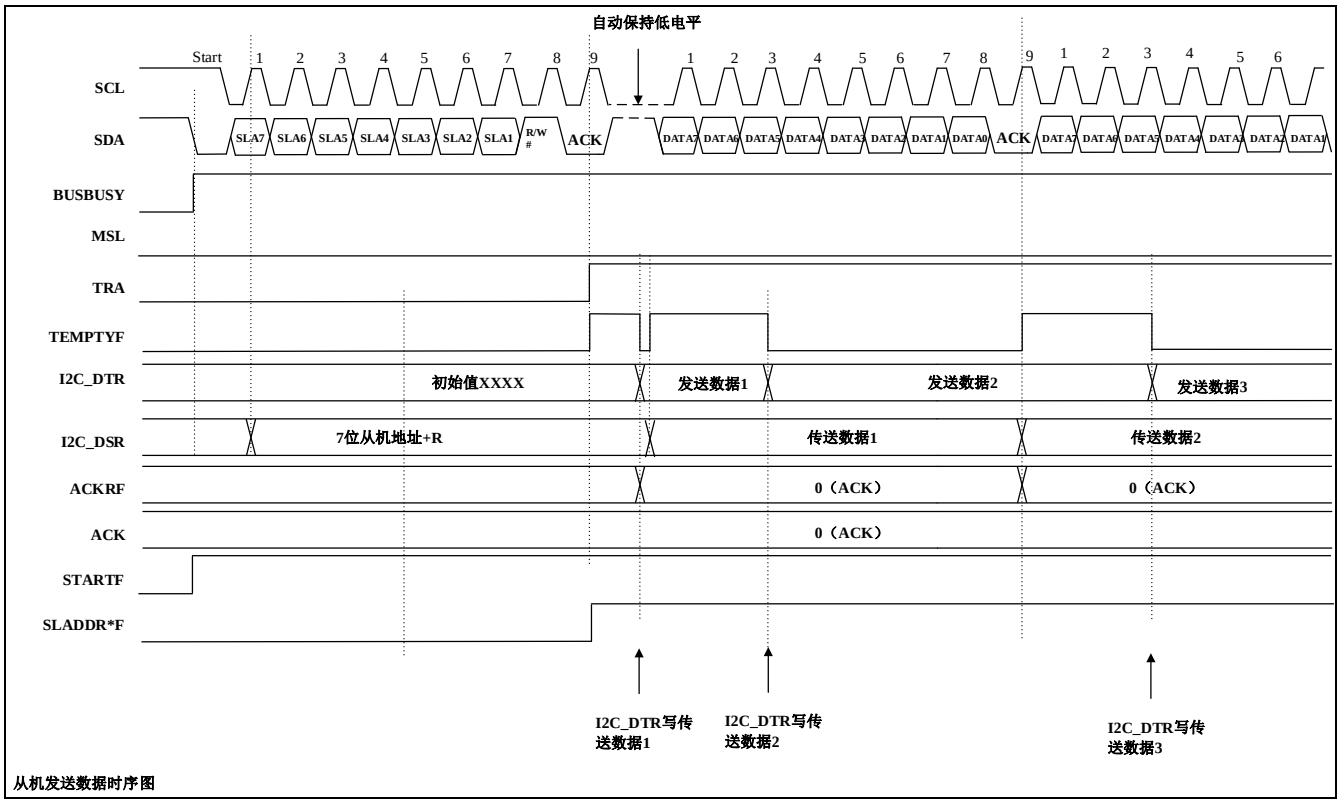


图 30-7　7 位地址格式的从机发送模式时序图（例）

从机接收数据

在从机接收模式中，接收来自主机的 SCL 时钟和数据，接收完数据后返回应答。从机接收数据的运行时序例如下图所示。

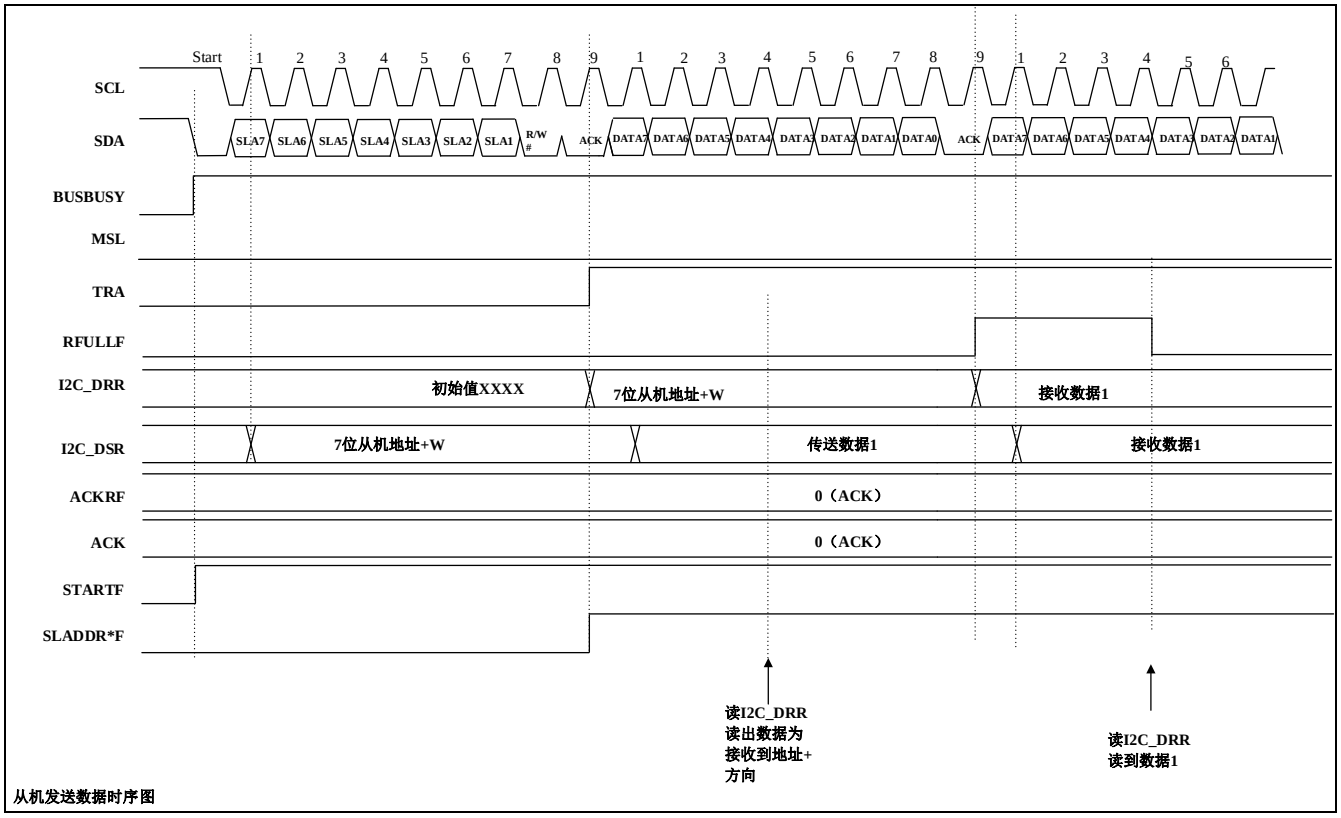


图 30-8　7 位地址格式从机接收模式时序图（例）

30.3.1.4 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。

30.3.1.5 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。

30.3.1.6 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。

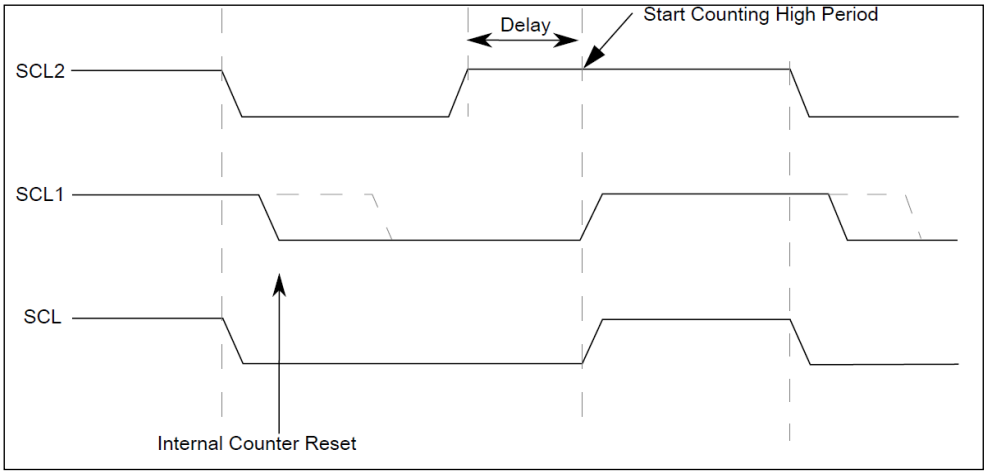


图 30-9 SCL 同步时序

30.3.1.7 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。

30.3.1.8 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。

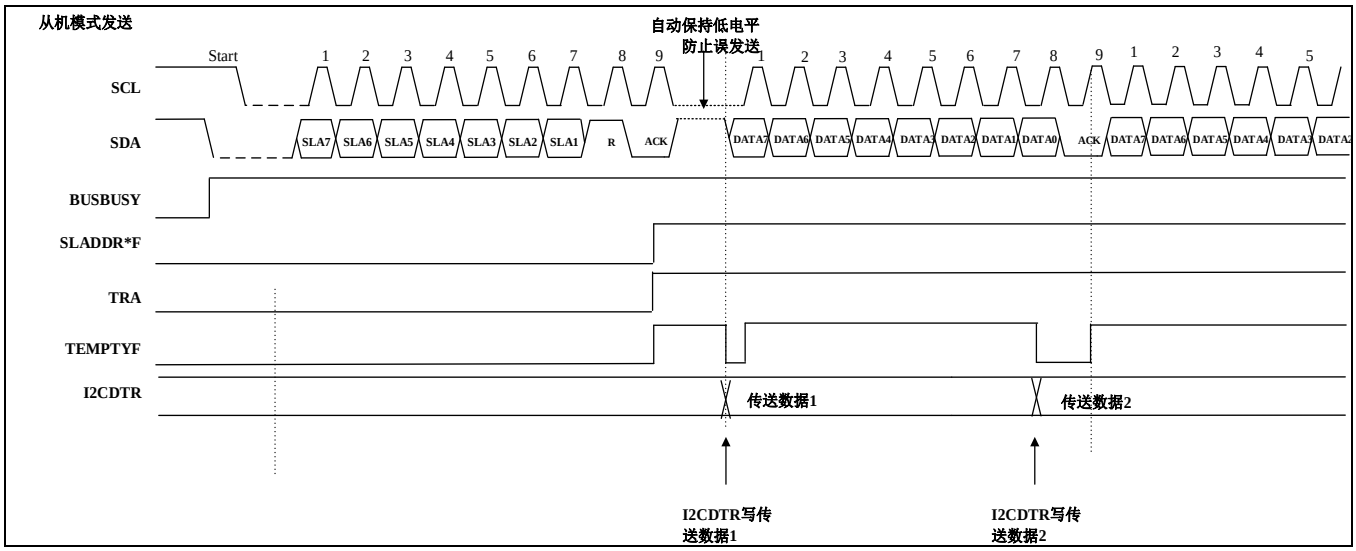


图 30-10 从机发送时序图（1）

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 IC_SR.NACKF 标志或者 TENDF 标志为“1”时，在第 9 个时钟下降后将 SCL 线保持为低电平。此时必须通过读 I2C_DRR 寄存器来结束通信，从而释放 SCL 线。

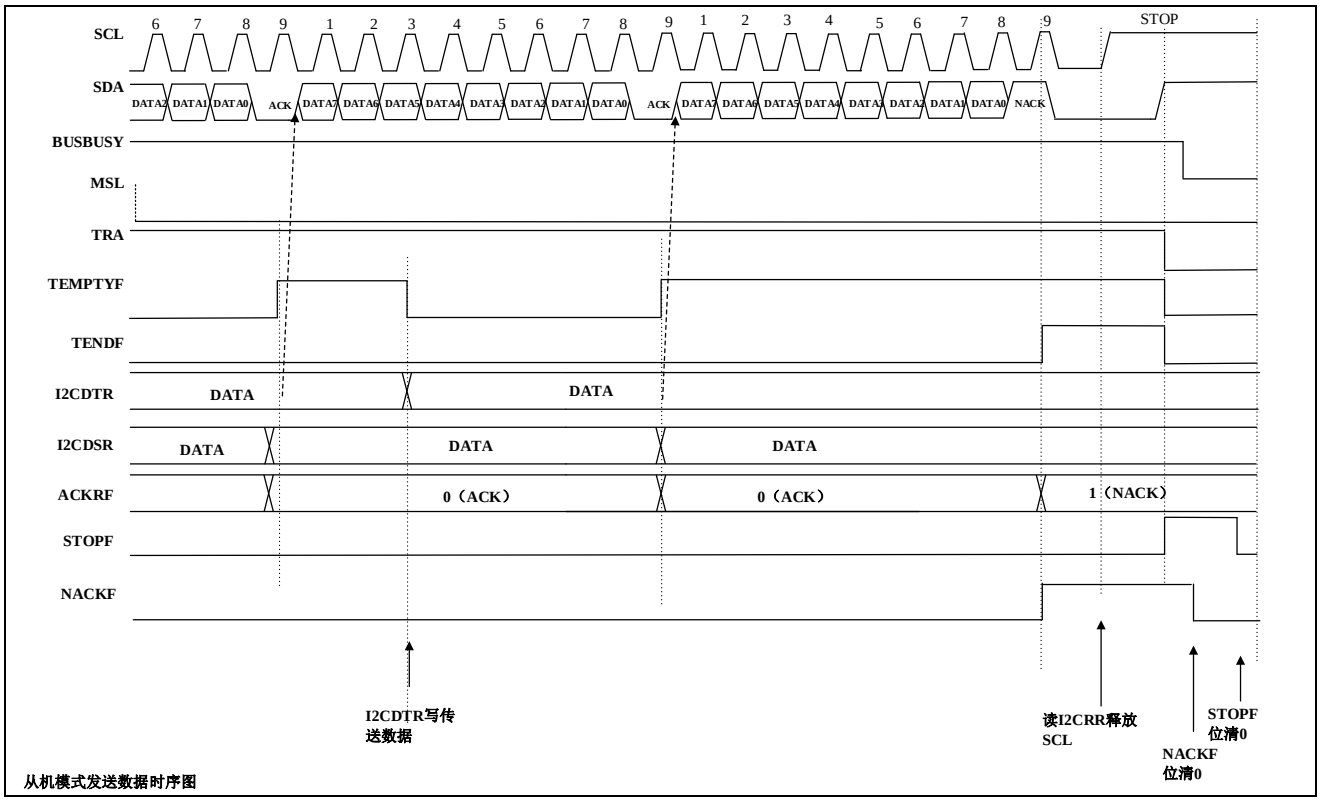
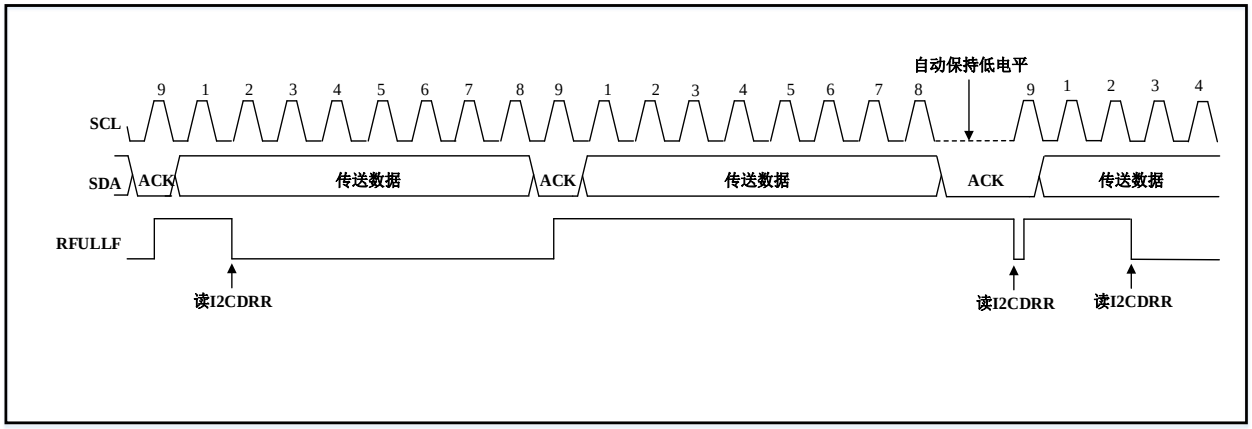


图 30-11　从机发送时序图（2）

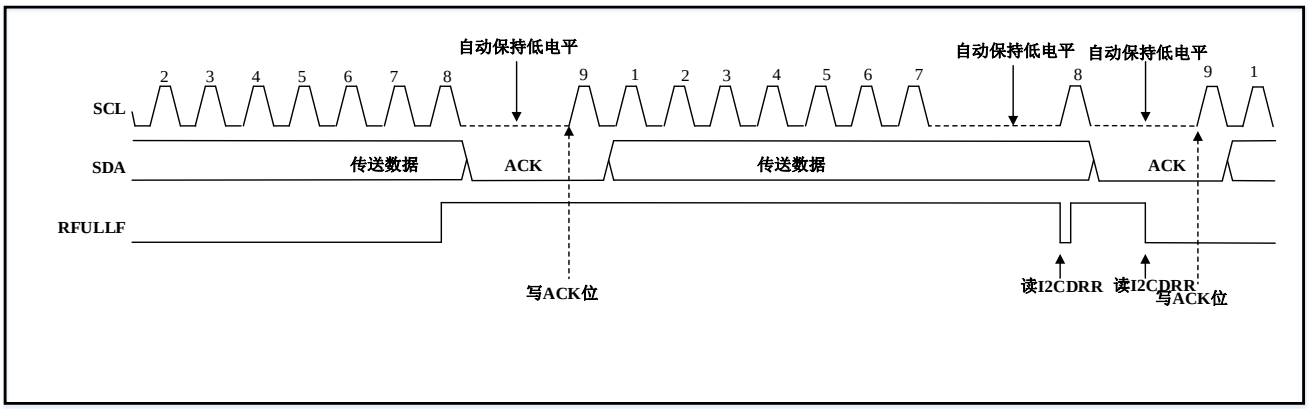
【从机接收模式】

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。



30.3.2 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。

30.3.2.1 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。

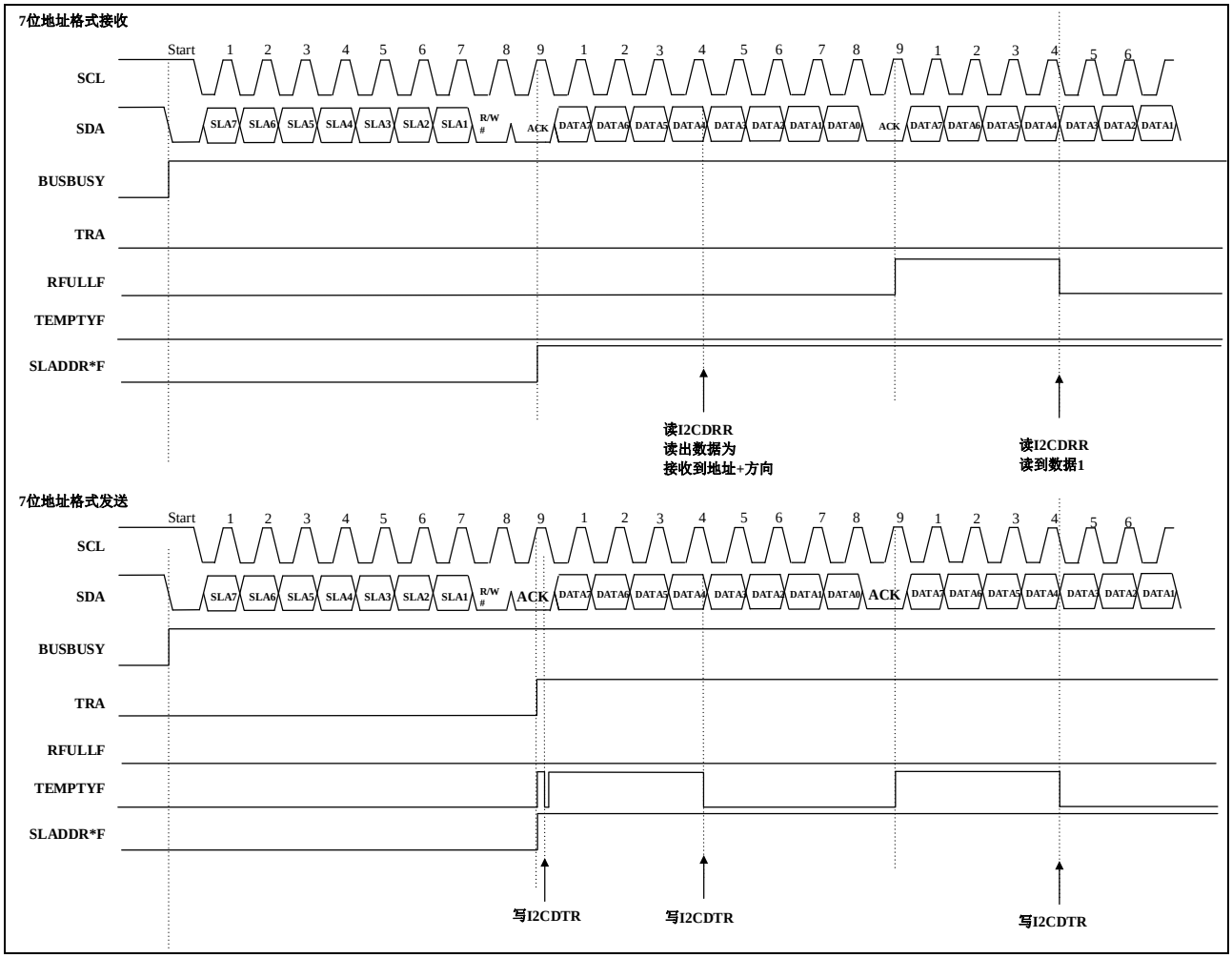


图 30-12　选择 7 位地址格式时的时序

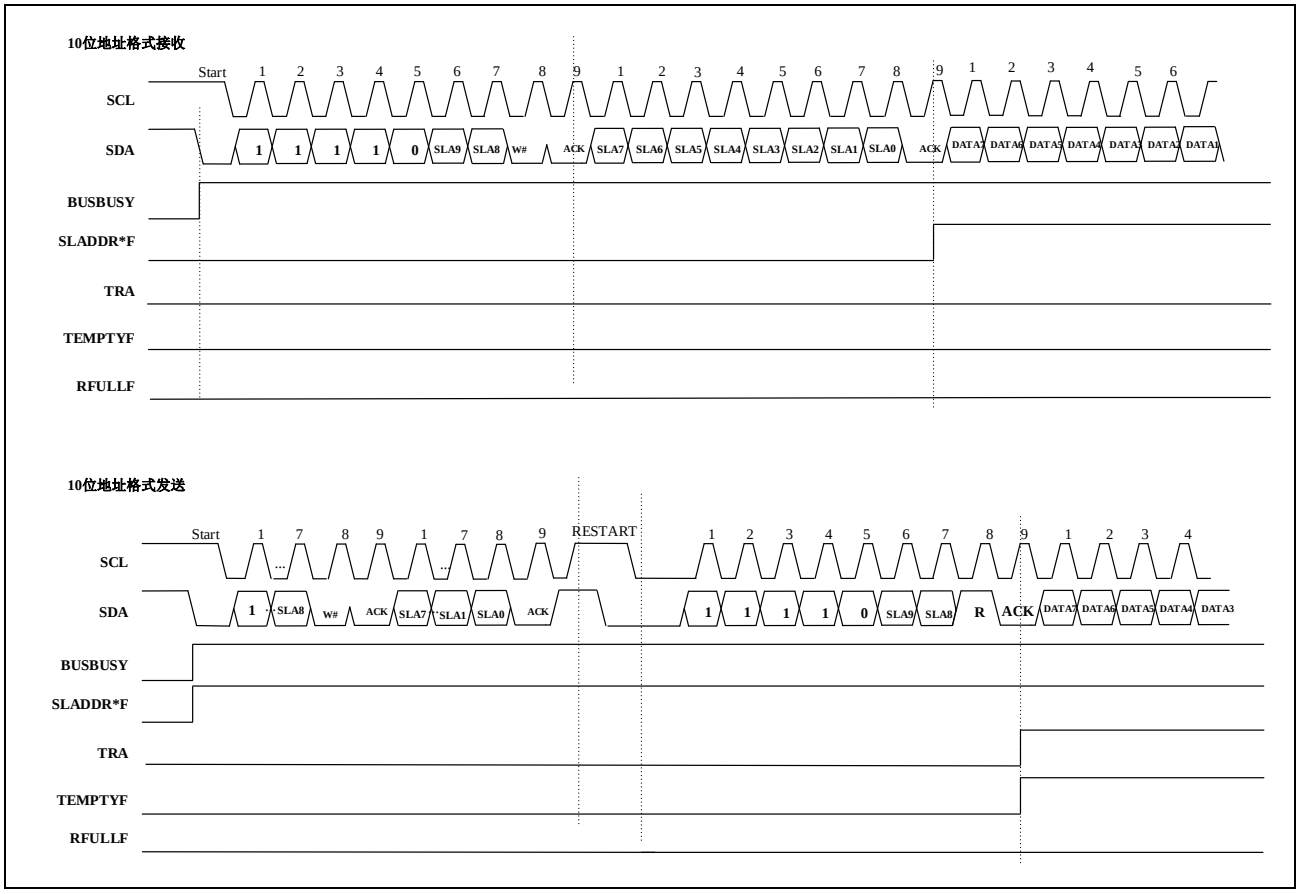


图 30-13　选择 10 位地址格式时的时序

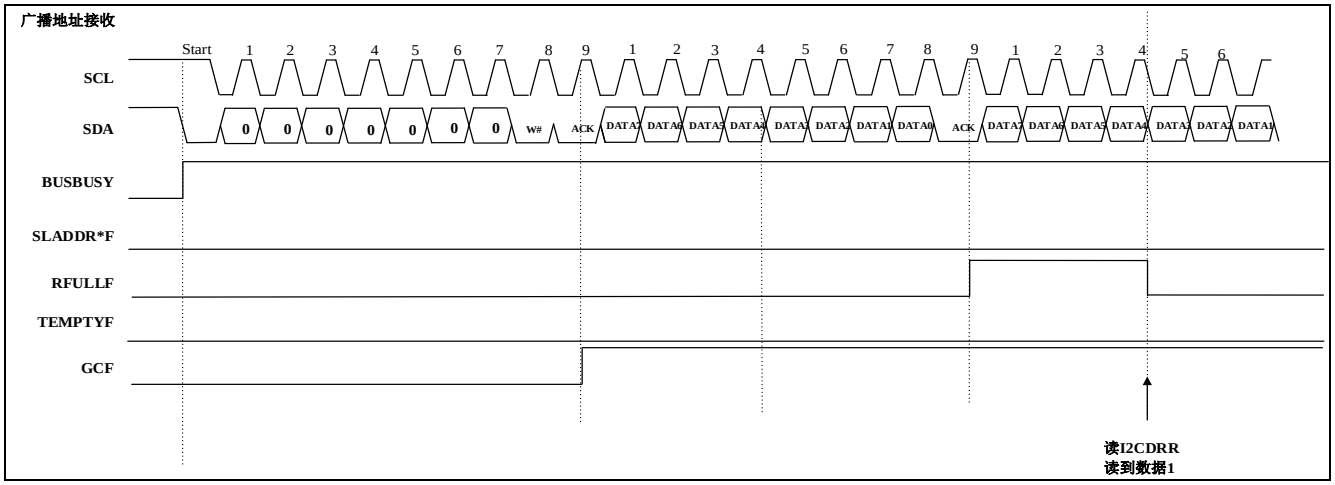
30.3.2.2 广播地址匹配

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。

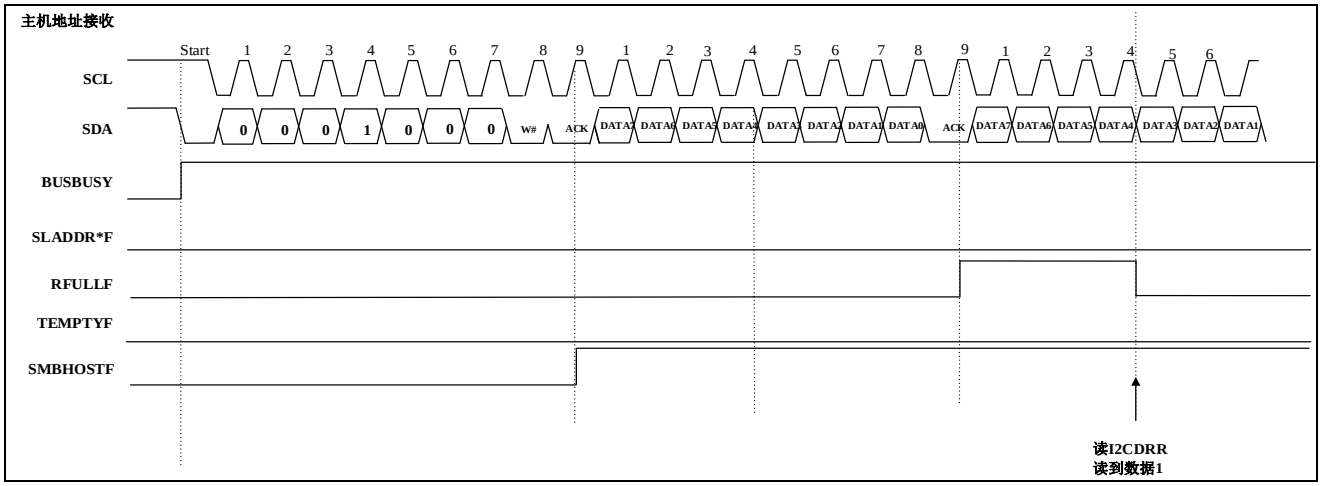


30.3.2.3 SMBus 主机地址匹配

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。

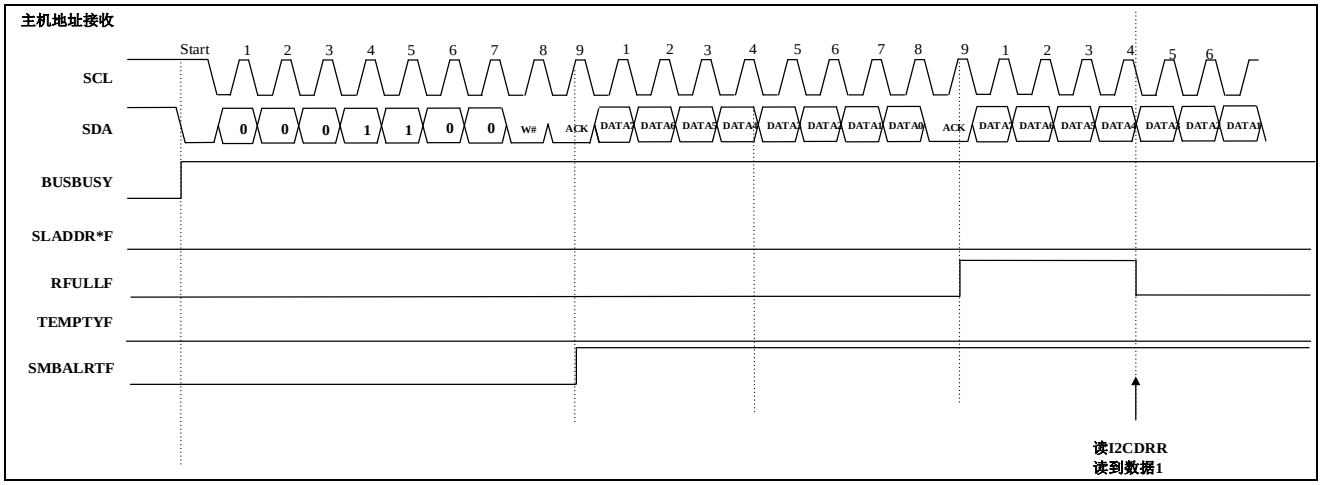


30.3.2.4 SMBus 报警响应地址匹配

本产品具有 SMBus 运行时的报警响应地址检测功能。如果在 I2C_CR1.SMBUS 位为“1”时将 I2C_CR1.SMBARLERTEN 位置“1”，就能在从机接收模式（I2C_CR1.MSL 位 TRA 位为“00b”）中检测 SMBUS 报警响应地址（0001 100b）。

如果检测到 SMBUS 报警响应地址，就在 SCL 时钟的第 9 个时钟的下降沿将 I2C_SR.SMBARLERTF 标志置“1”。

SMBUS 报警响应地址检测后的运行和普通的从机模式运行相同。

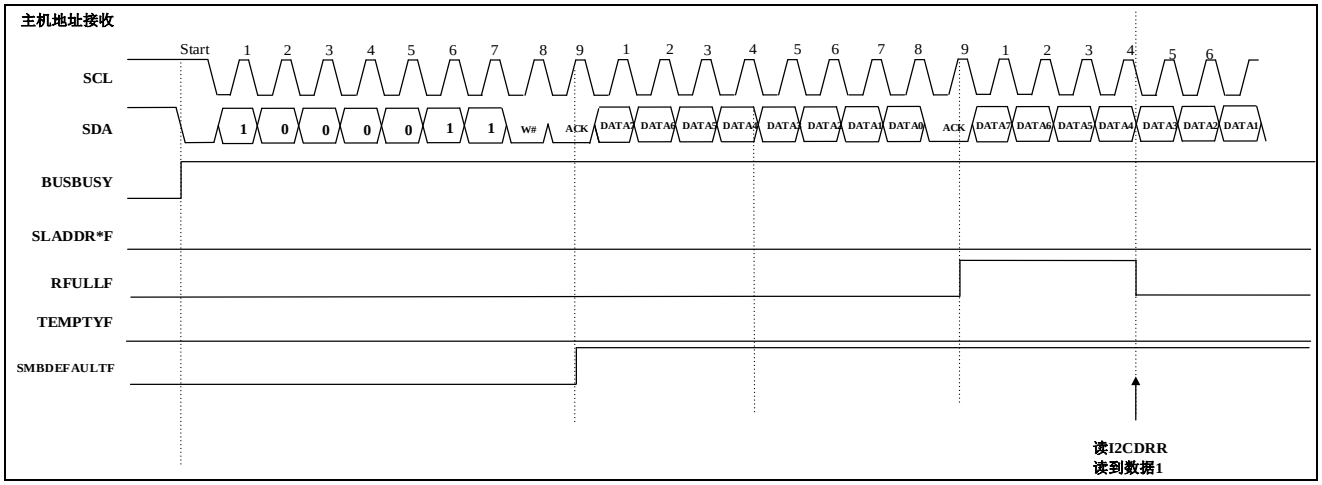


30.3.2.5 SMBus 默认地址匹配

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。



30.3.3 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。

30.3.3.1 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。

30.3.3.2 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。

30.3.4 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。

30.3.5 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。

中断一览表如下图所示。

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图所示。

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

30.3.6 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。

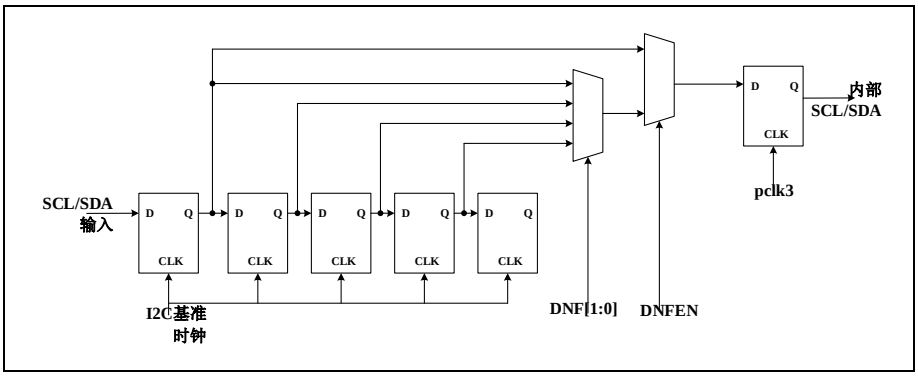


图 30-14 数字滤波电路框图

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。

30.4 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。

30.5 寄存器说明

I2C1 基准地址: 0x4004E000

I2C2 基准地址: 0x4004E400

I2C3 基准地址: 0x4004E800

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_DRR	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 30-2 寄存器一览表

30.5.1 I2C 控制寄存器 1(I2C_CR1)

复位值：0x00000040

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SWRST	-	-	-	-	ACK	STOP	START	RESTART	ENGC	-	SMBHOSTEN	SMBDEFAULTEN	SMBALRTEN	SMBUS	PE

位	标记	位名	功能	读写
b31-16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15	SWRST	软件复位	0：解除复位 1：软件复位 本位与PE位组合，选择内部状态复位或者通信复位	R/W
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	R/W

SWRST	PE	复位内容
1	0	通信复位：I2C内部的全部寄存器及内部状态复位。
1	1	内部状态复位：I2C_SR，I2C_DSR寄存器和内部状态机进行复位

			通信复位	
b7	RESTART	重复起始条件生成位	0：不生成重复起始条件 1：生成重复起始条件 此位可软件置1和清0。 硬件清0条件： 1）检测到开始条件 2）仲裁失败时 3）通信复位	R/W
b6	ENGC	广播呼叫时能	0：广播地址检测无效 1：广播地址检测有效	R/W
b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4	SMBHOSTEN	允许匹配SMBUS主机地址位	0：禁止匹配SMBUS主机地址 1：允许匹配SMBUS主机地址	R/W
b3	SMBDEFAULTEN	允许匹配SMBUS默认地址位	0：禁止匹配SMBUS默认地址 1：允许匹配SMBUS默认地址	R/W
b2	SMBALRTEN	允许匹配SMBUS报警响应地址位	0：禁止SMBUS报警响应地址 1：允许SMBUS报警响应地址	
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

30.5.2 I2C 控制寄存器 1(I2C_CR2)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	SMBALRTIE	SMBHOSTIE	SMBDEFAULTI	GENCALLIE	-	-	-	-

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	TMOUTIE	-	NACKIE	-	-	ARLOIE	-	TEMPTYIE	RFULLIE	-	STOPIE	TENDIE	SLADDR1IE	SLADDR0IE	STARTIE

位	标记	位名	功能	读写
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	SMBALRTIE	SMBUS报警响应地址匹配一致中断允许	0：SMBUS报警响应地址匹配一致中断禁止 1：SMBUS报警响应地址匹配一致中断允许	R/W
b22	SMBHOSTIE	SMBUS主机地址匹配一致中断允许	0：SMBUS主机地址匹配一致中断禁止 1：SMBUS主机地址匹配一致中断允许	R/W
b21	SMBDEFAULTIE	SMBUS默认地址匹配一致中断允许	0：SMBUS默认地址匹配一致中断禁止 1：SMBUS默认地址匹配一致中断允许	R/W
b20	GENCALLIE	广播呼叫地址匹配一致中断允许	0：广播呼叫地址匹配一致中断禁止 1：广播呼叫地址匹配一致中断允许	R/W
b19~b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14	TMOUTIE	超时中断允许	0：超时中断禁止 1：超时中断允许	R/W
b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	NACKIE	NACK中断允许	0：接收到NACK中断禁止 1：接收到NACK中断允许	R/W
b11~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9	ARLOIE	仲裁失败中断允许	0：仲裁失败中断禁止 1：仲裁失败中断允许	R/W
b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	TEMPTYIE	发送数据空中断允许位	0：发送数据空中断禁止 1：发送数据空中断允许	R/W
b6	RFULLIE	接收数据满中断允许位	0：接收数据满中断禁止 1：接收数据满中断允许	R/W
b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4	STOPIE	停止条件中断允许	0：总线检测到停止条件中断禁止 1：总线检测到停止条件中断允许	R/W

b3	TENDIE	发送一帧数据结束中断允许位	0：发送一帧数据结束中断禁止 1：发送一帧数据结束中断允许	R/W
b2	SLADDR1IE	从机地址1匹配一致中断允许	0：从机地址1匹配一致中断禁止 1：从机地址1匹配一致中断允许	R/W
b1	SLADDR0IE	从机地址0匹配一致中断允许	0：从机地址0匹配一致中断禁止 1：从机地址0匹配一致中断允许	R/W
b0	STARTIE	开始条件/重新开始条件中断允许	0：总线检测到开始条件中断禁止 1：总线检测到开始条件中断允许	R/W

30.5.3 I2C 控制寄存器 1(I2C_CR3)

复位值：0x00000006

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	FACKEN	-	-	-	-	HTMOUT	LTMOUT	TMOUTEN

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

30.5.4I2C 控制寄存器 4(I2C_CR4)

复位值：0x00300307

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。	
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

30.5.5 I2C 从机地址寄存器 0(I2C_SLR0)

复位值：0x00001000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRMOD0	-	-	SLADDR0EN	-	-	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

30.5.6 I2C 从机地址寄存器 1(I2C_SLR1)

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRMOD1	-	-	SLADDR1EN	-	-	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

30.5.7 I2C SCL 电平超时控制寄存器(I2C_SLTR)

复位值：0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TOUTHIGH[15:0]															

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

位	标记	位名	功能	读写
b31~b16	TOUTHIGH	SCL高电平超时周期	TOUTHIGH设定SCL高电平超时周期。 SCL高电平超时时间=TOUTHIGH*I2C基准时钟周期	R/W
b15~b0	TOUTLOW	SCL低电平超时周期	TOUTLOW设定SCL低电平超时周期。 SCL低电平超时时间=TOUTLOW*I2C基准时钟周期	R/W

30.5.8 I2C 状态寄存器(I2C_SR)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	SMBALRTF	SMBHOSTF	SMBDEFAULTF	GENCALLF	-	TRA	BUSY	MSL

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	TMOUTF	-	NACKF	-	ACKRF	ARLOF	-	TEMPTYF	RFULLF	-	STOPF	TENDF	SLADDR1F	SLADDR0F	STARTF

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	0：未匹配到广播呼叫地址	R

		标志	1：匹配到广播呼叫地址 置“1”条件： 当接收的从机地址与广播呼叫地址（All“0”）匹配一致时 清“0”条件： GENCALLFCLR写“1” 检测到停止条件 通信复位	
b19	Reserved	-	读出时为“0”,写入时写“0”	R
b18	TRA	发送接收选择位	此位表示选择发送数据还是接收数据。 0：接收数据 1：发送数据 此位可软件置1和清0。 硬件置“1”条件 检测到开始条件 主机模式下，发送的R/W位为0 从机模式下，地址匹配且接收的R/W位为1 硬件清“0”条件 检测到停止条件 主机模式下，发送的R/W位为1 从机模式下，地址匹配且接收的R/W位为0 通信复位	R/W
b17	BUSY	总线忙标志位	0：空闲状态，总线上无通信 1：占有状态，总线正在通信 置“1”条件： 检测到总线上开始条件 清“0”条件： 检测到总线停止条件 通信复位	R
b16	MSL	主从机选择位	此位表示主机还是从机。 0：从机模式 1：主机模式 通过和TRA位的组合，表示I2C的运行模式。 MSL / TRA / I2C运行模式 0 / 0 / 从机接收模式 0 / 1 / 从机发送模式 1 / 0 / 主机接收模式 1 / 1 / 主机发送模式 此位可软件置1和清0。 硬件置“1”条件 START位为1的状态下，检测到开始条件 硬件清“0”条件 1）检测到停止条件	R/W

			2）仲裁失败 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	R

			通信复位	
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	R

			I2C_SLR0. SLADDR0[7:1]匹配时。 当I2C_SLR0.ADDRMOD0位为“1”时，接收10位从机地址的第一个字节地址与11110b+I2C_SLR0. SLADDR0[9:8]匹配一致并且第二个字节地址与I2C_SLR0. SLADDR0[7:0]匹配一致。 清“0”条件： 检测到停止条件 SLADDR0FCLR写“1” 通信复位	
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

30.5.9 I2C 状态清零寄存器(I2C_CLR)

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	SMBALRTFCLR	SMBHOSTFCLR	SMBDEFAULTF	GENCALLFCLR	-	-	-	-

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	TMOUTFCLR	-	NACKFCLR	-	-	ARLOFCLR	-	TEMPTYFCLR	RFULLFCLR	-	STOPFCLR	TENDFCLR	SLADDR1FCLR	SLADDR0FCLR	STARTFCLR

位	标记	位名	功能	读写
b31~b24	Reserved	-	写入时写“0”	W
b23	SMBALRTFCLR	SMBUS报警响应地址匹配一致标志清零位	写“1”清除SMBALRTF标志位	W
b22	SMBHOSTFCLR	SMBUS主机地址匹配一致标志清零位	写“1”清除SMBHOSTF标志位	W
b21	SMBDEFAULTFCLR	SMBUS默认地址匹配一致标志清零位	写“1”清除SMBDEFAULTF标志位	W
b20	GENCALLFCLR	广播呼叫地址匹配一致标志	写“1”清除GENCALLF标志位	W
b19~b15	Reserved	-	写入时写“0”	W
b14	TMOUTFCLR	超时标志位	写“1”清除TMOUTF标志位	W
b13	Reserved	-	写入时写“0”	W
b12	NACKFCLR	NACK标志位	写“1”清除NACKF标志位	W
b11~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1匹配一致标志清零位	写“1”清除SLADDR0F标志位	W
b0	STARTFCLR	开始条件/重新开始条	写“1”清除STARTF标志位	W

	件标志清零位

30.5.10 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。

30.5.11 I2C 数据接收寄存器(I2C_DRR)

复位值：0x00

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

如果接收到 1 帧数据，就能将接收数据从移位寄存器（I2C_DSR）转存到 I2C_DRR 寄存器，进而可以进入到下一个数据接收状态。

I2C_DSR 寄存器和 I2C_DRR 寄存器是双缓冲结构，在 I2C_DSR 寄存器数据接收过程中，如果读取了 I2C_DRR 寄存器的数据，就能进行连续接收数据。

禁止对 I2C_DRR 寄存器写。请在接收数据满中断要求发生时，仅读一次 I2C_DRR 寄存器。

在 I2C_SR.RFULLF 标志位为“1”的状态下，如果不读取 I2C_DRR 寄存器的数据，而立即接收下一个数据，SCL 时钟就在下一次 RFULLF 标志位变“1”的前一个 SCL 时钟自动保持为低电平。

30.5.12 I2C 数据移位寄存器(I2C_DSR)

b7	b6	b5	b4	b3	b2	b1	b0
DSR							

I2C_DSR 寄存器用于发送和接收数据的移位寄存器。I2C_DSR 寄存器不可读也不可写。

在数据发送时，将发送数据从 I2C_DTR 寄存器传送到 I2C_DSR 寄存器，从 SDA 引脚发送数据。在数据接收时，一旦 1 帧数据接收结束，就将数据从 I2C_DSR 寄存器传送到 I2C_DRR 寄存器。

30.5.13 I2C 时钟控制寄存器(I2C_CCR)

复位值：0x00001F1F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	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,CKSDIV=000

波特率=1/{[(SHIGHW+3)+(SLOWW+3)]/ΦI2C+SCL 上升时间+SCL 下降时间}

SDNFE=1,CKSDIV=000

波特率=1/{[(SHIGHW+3+滤波能力)+(SLOWW+3+滤波能力)]/ΦI2C+SCL 上升时间+SCL 下降时间}

DNFE=0,CKSDIV!=000

波特率=1/{[(SHIGHW+2)+(SLOWW+2)]/ΦI2C+SCL 上升时间+SCL 下降时间}

DNFE=1,CKSDIV!=000

波特率=1/{[(SHIGHW+2+滤波能力)+(SLOWW+2+滤波能力)]/ΦI2C+SCL 上升时间+SCL 下降时间}

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。

30.5.14 I2C 滤波控制寄存器(I2C_FLTR)

复位值：0x00000010

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	ANFEN	DNFEN	-	-	DNF[1:0]	

位	标记	位名	功能	读写
b31~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5	ANFEN	模拟滤波功能允许位	0：模拟滤波功能禁止 1：模拟滤波功能允许	R/W
b4	DNFEN	数字滤波功能允许位	0：数字滤波功能禁止 1：数字滤波功能允许	R/W
b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1-b0	DNF[1:0]	数字滤波器滤波能力选择	00：滤波能力1个I2C基准时钟周期 01：滤波能力2个I2C基准时钟周期 10：滤波能力3个I2C基准时钟周期 11：滤波能力4个I2C基准时钟周期	R/W

31 串行外设接口（SPI）

31.1 简介

本产品搭载 6 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	1通道
串行通信功能	• 支持4线式SPI模式和3线式时钟同步运行模式 • 支持全双工和只发送两种通信方式 • 可调整通信时钟SCK的极性和相位
数据格式	• 可选择数据移位顺序:MSB开始/LSB开始 • 可选择数据宽度:4/5/6/7/8/9/10/11/12/13/14/15/16/20/24/32位 • 单次最多可传送或接收4帧宽度为32位的数据
波特率	• 主机模式下可通过内置专用波特率发生器对波特率进行调整，波特率范围为PCLK1的2分频~256分频 • 从机模式下允许的最大波特率为PCLK1的6分频
数据缓冲	• 带有16字节的数据缓冲区域 • 支持双重缓冲
错误监测	• 模式故障错误监测 • 数据过载错误监测 • 数据欠载错误监测 • 奇偶校验错误监测
片选信号控制	• 每个通道配置四根片选信号线 • 可对片选信号和通信时钟的相对时序关系进行调整 • 可对连续两次通信之间的片选信号无效时间进行调整 • 极性可调
主机模式下的传输控制	• 通过将数据写入数据寄存器启动传输 • 通信自动挂起功能

中断	• 接收数据区域已满 • 发送数据区域已空 • SPI错误（模式/过载/欠载/奇偶校验） • SPI空置 • 传输完成（仅为事件源）
低功耗控制	可设置模块停止
其他功能	• SPI初始化功能

表 31-1 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 31.8.3 过载错误。

31.2 SPI 系统框图

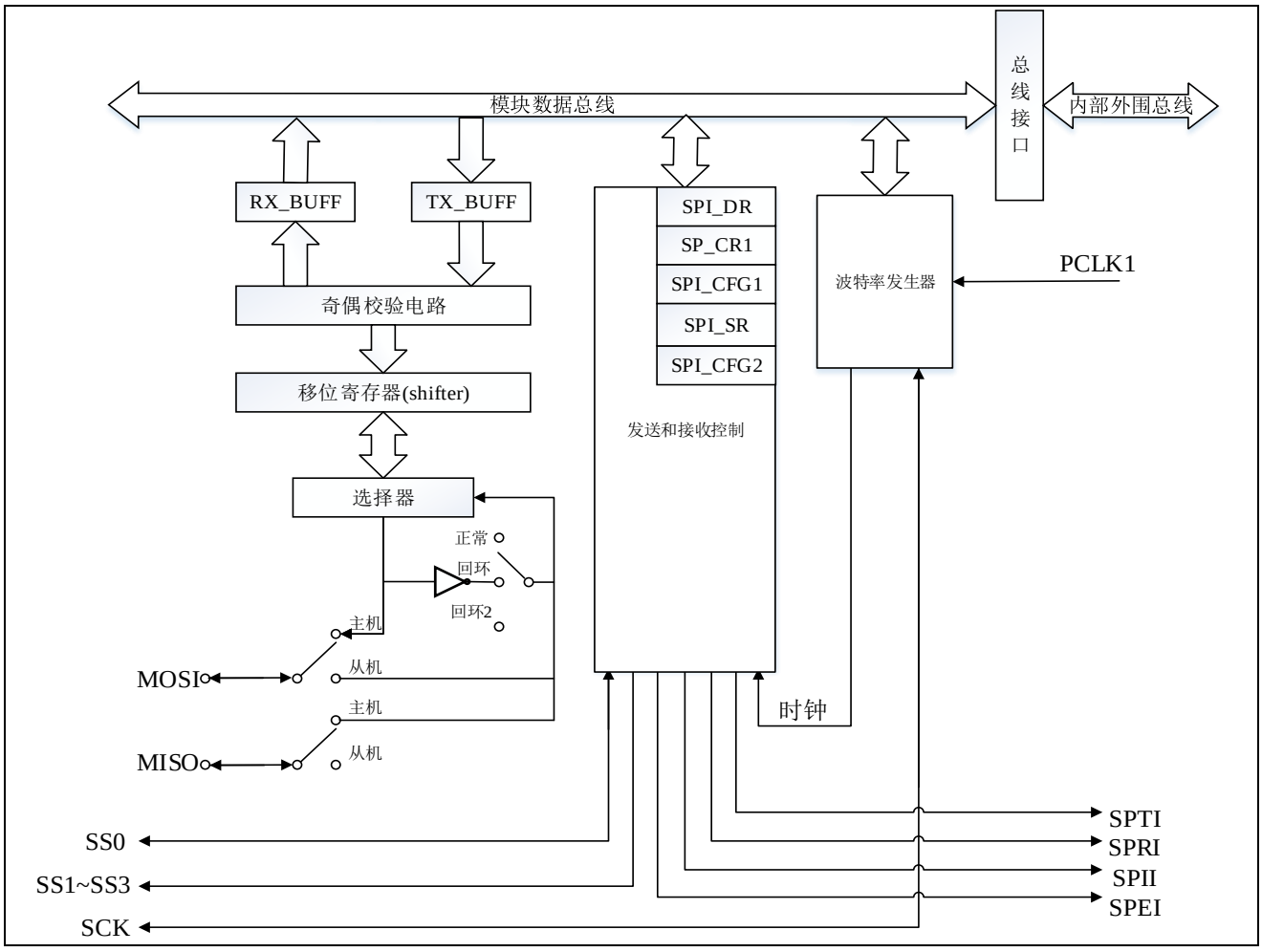


图 31-1 系统框图

31.3 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 31-2 管脚说明

31.4 SPI 动作系统说明

31.4.1 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 31-3 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 31-3 主机模式时 SPI 管脚状态说明

注意：管脚输入类型请设定为 CMOS 输入，输出请设定为高驱动力模式，设定请参照第 11 章节通用 IO 特殊寄存器 PCR。

31.4.2 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 31-4 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 31-4 从机模式时 SPI 管脚状态说明

注意：管脚输入类型请设定为 CMOS 输入，输出请设定为高驱动力模式，设定请参照第 11 章节通用 IO 特殊寄存器 PCR。

31.4.3 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。

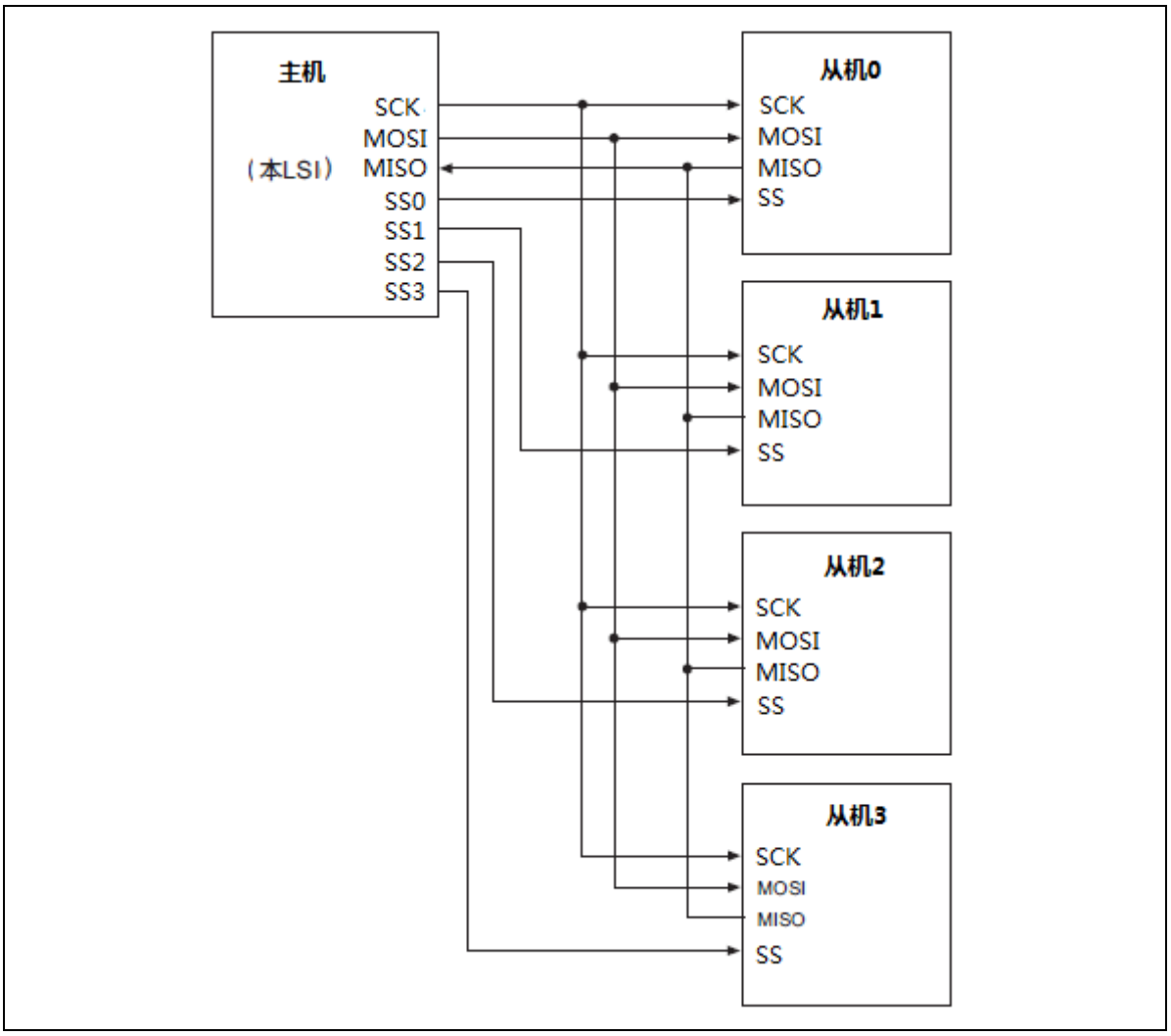


图 31-2 主机模式结构

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。

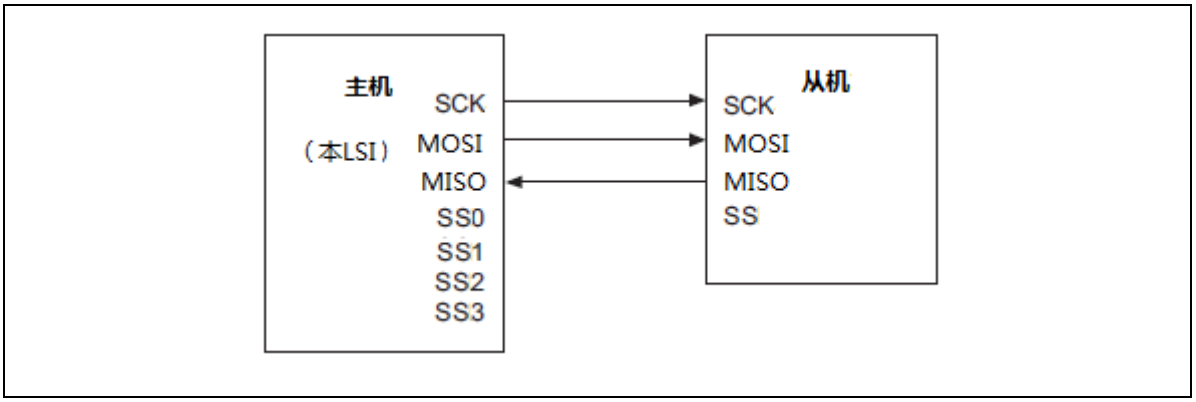


图 31-3　三线式时钟同步运行

31.5 数据通信说明

31.5.1 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。

$$波特率 = \frac{fpck}{2^{N+1}}$$

MBR[2:0]位的设定值	分频比	波特率			
		PCLK1=5MHz	PCLK1=10MHz	PCLK1=20MHz	PCLK1=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 31-5 部分设定值的位速

31.5.2 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 31-4 所示。

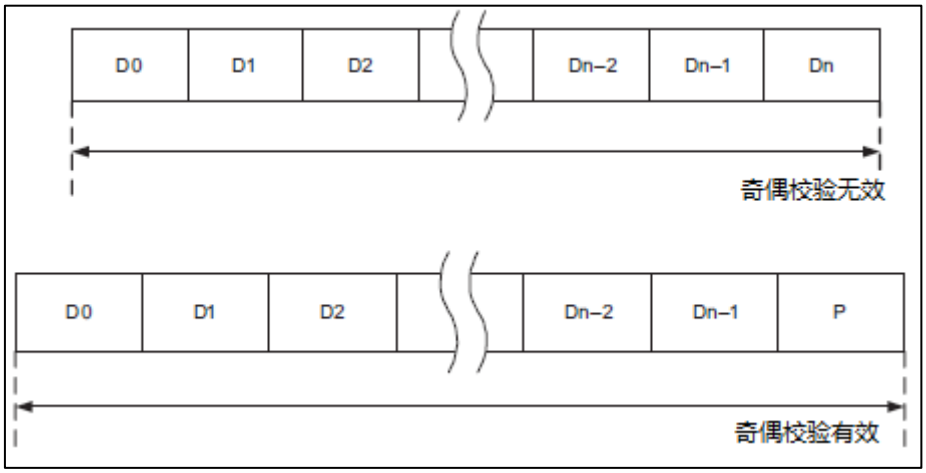


图 31-4 数据格式

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。

31.5.3 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 31-5 是 CPHA=0 时 SPI 的传送时序图。当 SSi 信号的输入电平变为有效电平时，MOSI/MISO 开始更新传输数据。在 SSi 信号变为有效后的第一个 SCK 信号边沿进行第一次数据采样，在此之后，每过一个 SCK 周期对数据进行一次采样。每次采样后的 1/2 个 SCK 周期时 MOSI/MISO 信号上数据进行更新。CPOL 位的设定值不影响 SCK 信号的运行时序，而只影响信号的极性。

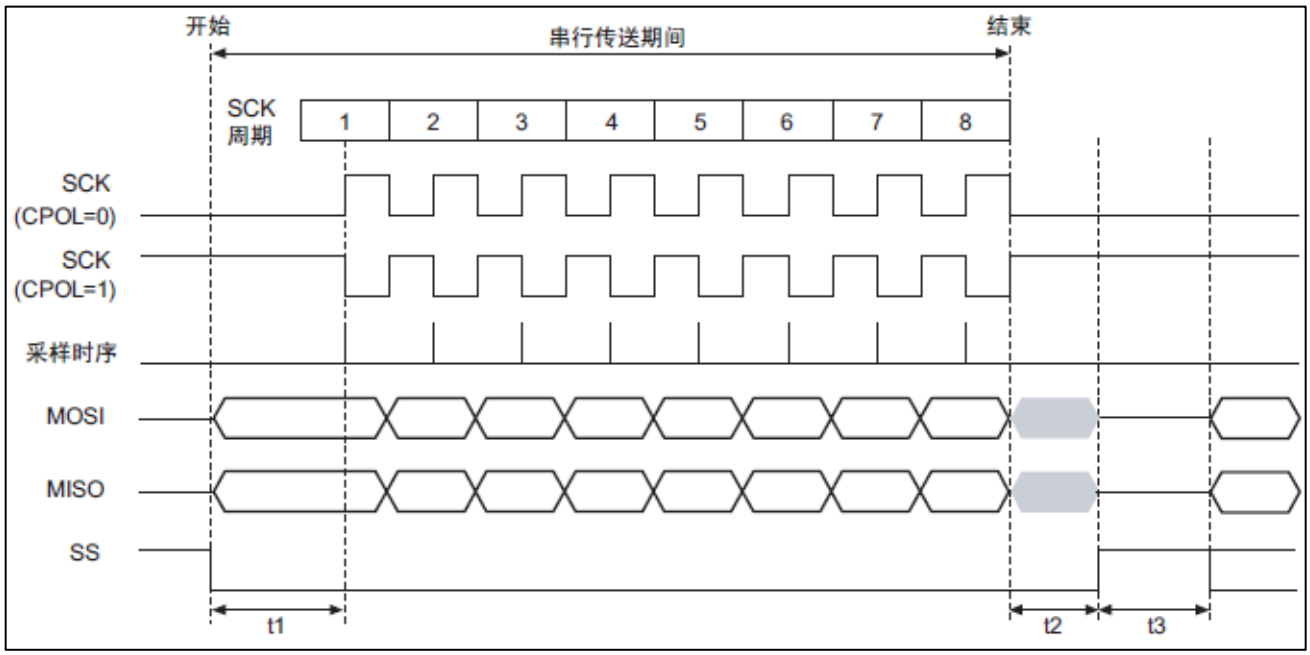


图 31-5 数据传送格式图（CPHA=0）

上图中 t1 表示从 SSi 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 信号变成无效的间隔期间（SS 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 31.6.2 这一章节。

2) CPHA=1 的情况

当 SPI_CFG2.CPHA 位为“1” 时，SPI 在 SCK 的奇数边沿进行数据更新，偶数边沿进行数据采样。图 31-6 是 CPHA=1 时 SPI 的传送时序图。MOSI/MISO 在 SSi 信号变为有效电平后的第一个 SCK 信号边沿时开始传输数据的更新。在此之后，每过一个 SCK 周期对数据进行一次更新。每次更新后的 1/2 个 SCK 周期时对数据进行采

样。SPI_CFG2.CPOL 位的设定值不影响 SCK 信号的运行时序而只影响信号的极性。

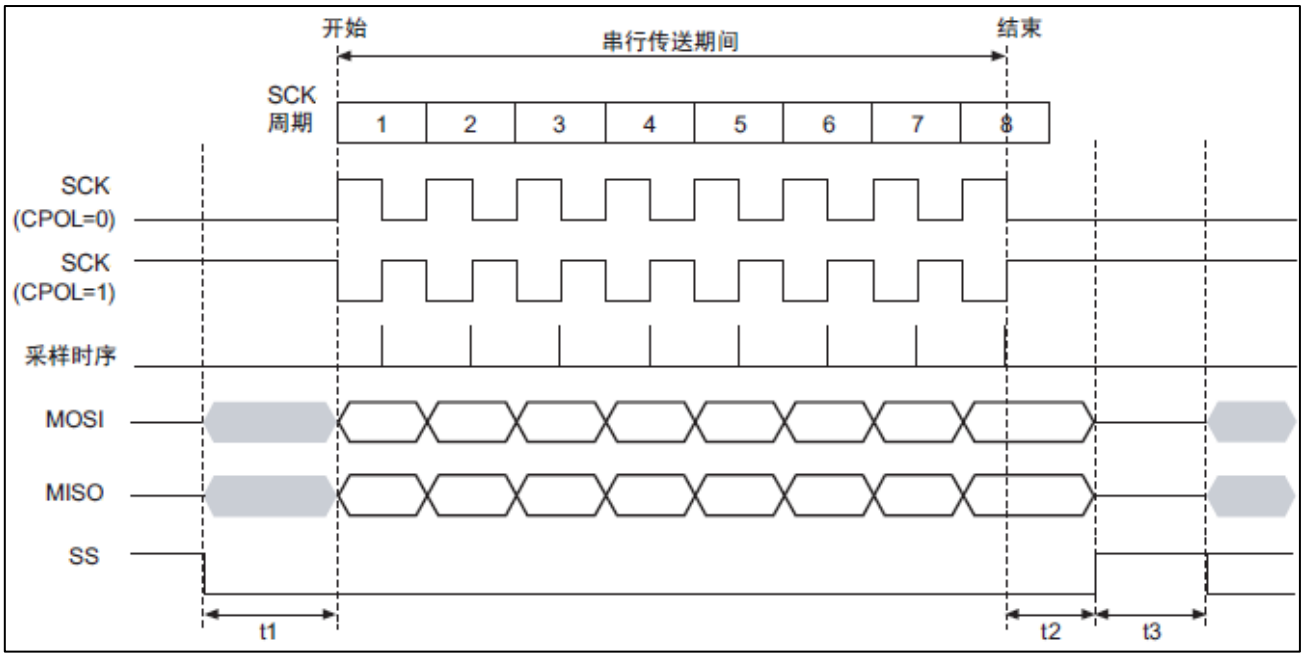


图 31-6　数据传送格式（CPHA=1）

t1、t2、t3 和 CPHA 位为 0 的情况相同。

31.5.4 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 31-7 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。

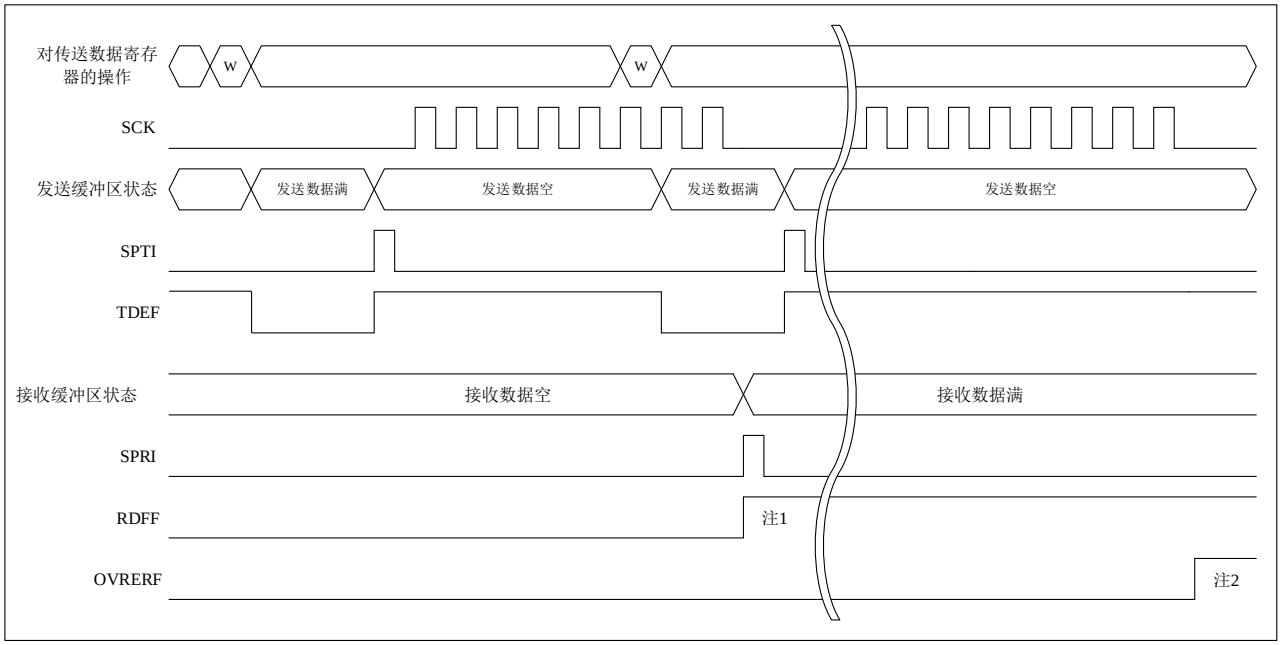


图 31-7 全双工同步串行通信

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 OVRERF 置成 1，本次数据接收无效，接收移位寄存器中的数据将被丢弃。

2) 只发送通信方式

当 SPI_CR1.TXMDS 位为“1” 时，SPI 运行在只发送通信方式。如图 31-8 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。

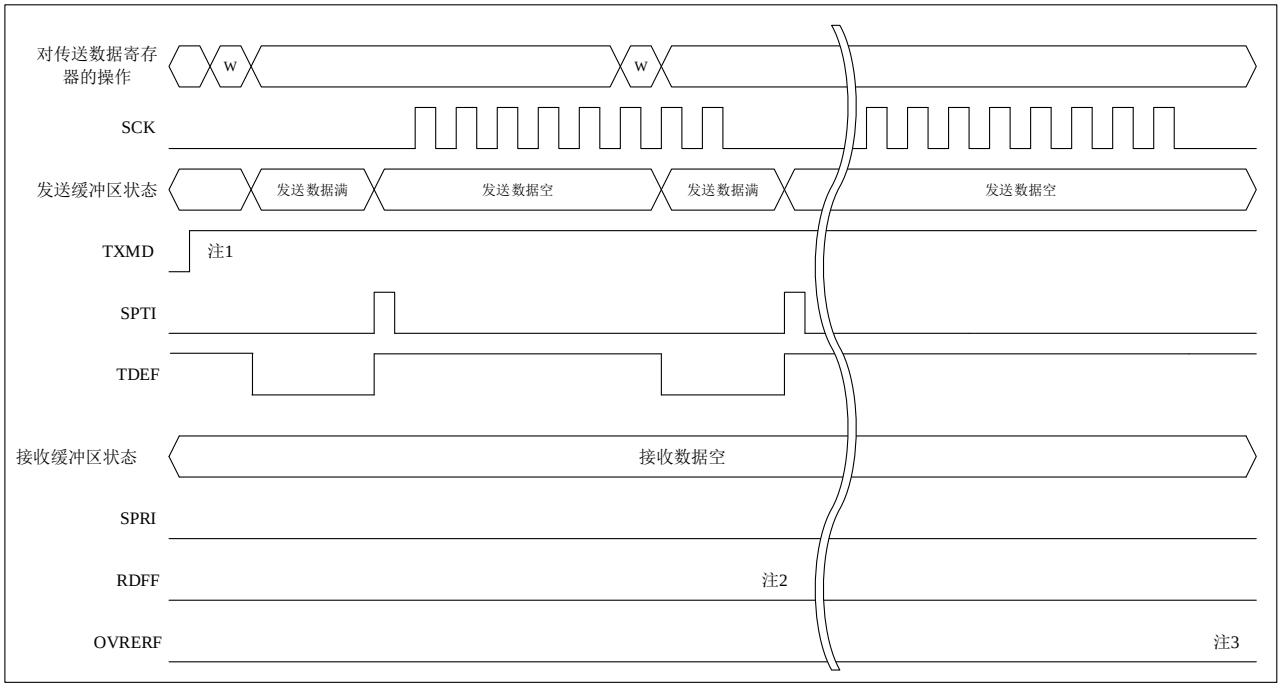


图 31-8 只进行发送通信

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，也不会有数据被接收，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。

31.6 运行说明

31.6.1 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 31-6 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	~PCLK1/2	~PCLK1/6	~PCLK1/2	~PCLK1/6
时钟源	内部波特率发生器	SCK输入	内部波特率发生器	SCK输入
时钟极性	2种	2种	2种	2种
时钟相位	2种	2种	2种	1种（CHPA=1）
开始传送位	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB
传送数据长度	4~32位	4~32位	4~32位	4~32位
SCK延迟控制	有	无	有	无
SS无效延迟控制	有	无	有	无
下次存取延迟控制	有	无	有	无
传送启动方法	通过发送缓冲器空中断请求，写发送缓冲器	SS输入有效或SCK时钟边沿	通过发送缓冲器空中断请求，写发送缓冲器	SCK振荡
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 31-6 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。

31.6.2 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 32.5.3 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，各种迟延时间的设定等。

② 设置通信配置寄存器 2（SPI_CFG2）,包括有 SS 电平设定，数据移位顺序设定，各种延迟的允许位的设定，数据格式及时钟极性相位的设定等。

③ 如需要使用中断，请设置系统的中断寄存器。

④ 如需要使用 DMA，请设置 DMA 的相关寄存器。

⑤ 设定输入输出管脚。

⑥ 设定 SPI 控制寄存器 SPI_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。

31.6.3 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 的接收数据复制到 RX_BUFF。可通过访问 SPI_DR 读取该数据。SPI 在串行传送结束后将 shifter 改为空状态，该状态与 RX_BUFF 的状态无关。

在串行传送期间，如果 SPI 检测到 SS0 输入信号无效，就发生模式故障错误。

最后的采样时序取决于传送数据的位长，从属模式的 SPI 的数据长度取决于 SPI_CFG2.DSIZE[3:0] 位的设定值，SS0 输入信号的极性取决于 SPI_CFG1.SS0PV 位的设定值。有关 SPI 传送格式的详细内容，请参照 32.5.3 传送格式。

注意：

- 当 SPI_CFG2.CPHA 位为“0” 时，将 SS0 输入信号电平从无效变为有效视为开始串行传送的触发信号。由于在从机模式的结构中 SS0 输入信号被固定为有效状态，此时 SPI 将无法正常开始串行传送。因此，在 SS0 输入信号被固定为有效状态的

结构中，要使从机模式的 SPI 正常进行发送和接收，必须将 CPHA 位设置为“1”。如果需要将 CPHA 位设成“0”，则不能固定 SS0 输入信号。

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。

31.6.4 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 32.5.3 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。

31.6.5 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使用 SS0 管脚，因此，在 CPHA 位为“1” 时， 将首个 SCK 边沿视为开始串行传送的触发信号。

如果 SPI 在 shifter 为空的状态下检测到串行传送的开始，就将 shifter 改为满状态，并且不能在串行传送过程中将数据从 TX_BUFF 传送到 shifter。如果在开始串行传送前 shifter 已满，SPI 就保持 shifter 的满状态。

如果 SPI 检测到最后采样时序的 SCK 边沿，则本次串行传输结束。在 RX_BUFF 为空的情况下，串行传送结束后 SPI 将 shifter 的接收数据复制到 RX_BUFF。可通过访问 SPI_DR 读取该数据。SPI 在串行传送结束后将 shifter 改为空状态，该状态与 RX_BUFF 的状态无关。

最后的采样时序取决于传送数据的位长，从属模式的 SPI 的数据长度取决于 SPI_CFG2.DSIZE[3:0] 位的设定值。

2) 时钟同步运行模式时从机的初始化设置

① 设定通信配置寄存器 1（SPI_CFG1），主要包括使用帧数的设定。

② 设置通信配置寄存器 2（SPI_CFG2）,包括有传输速率，数据格式及时钟极性相位的设定等。

③ 如需要使用中断，请设置系统的中断寄存器。

④ 如需要使用 DMA，请设置 DMA 的相关寄存器。

⑤ 设定输入输出管脚。

⑥ 设定 SPI 控制寄存器 SPI_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。

31.6.6 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/OVERRF/UDRERF/PERF）被置成 1。

② 确认 SS0 状态，排除模式故障错误。

③ 将 SPE 清零，停止 SPI 动作。

④ 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。

31.7 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。

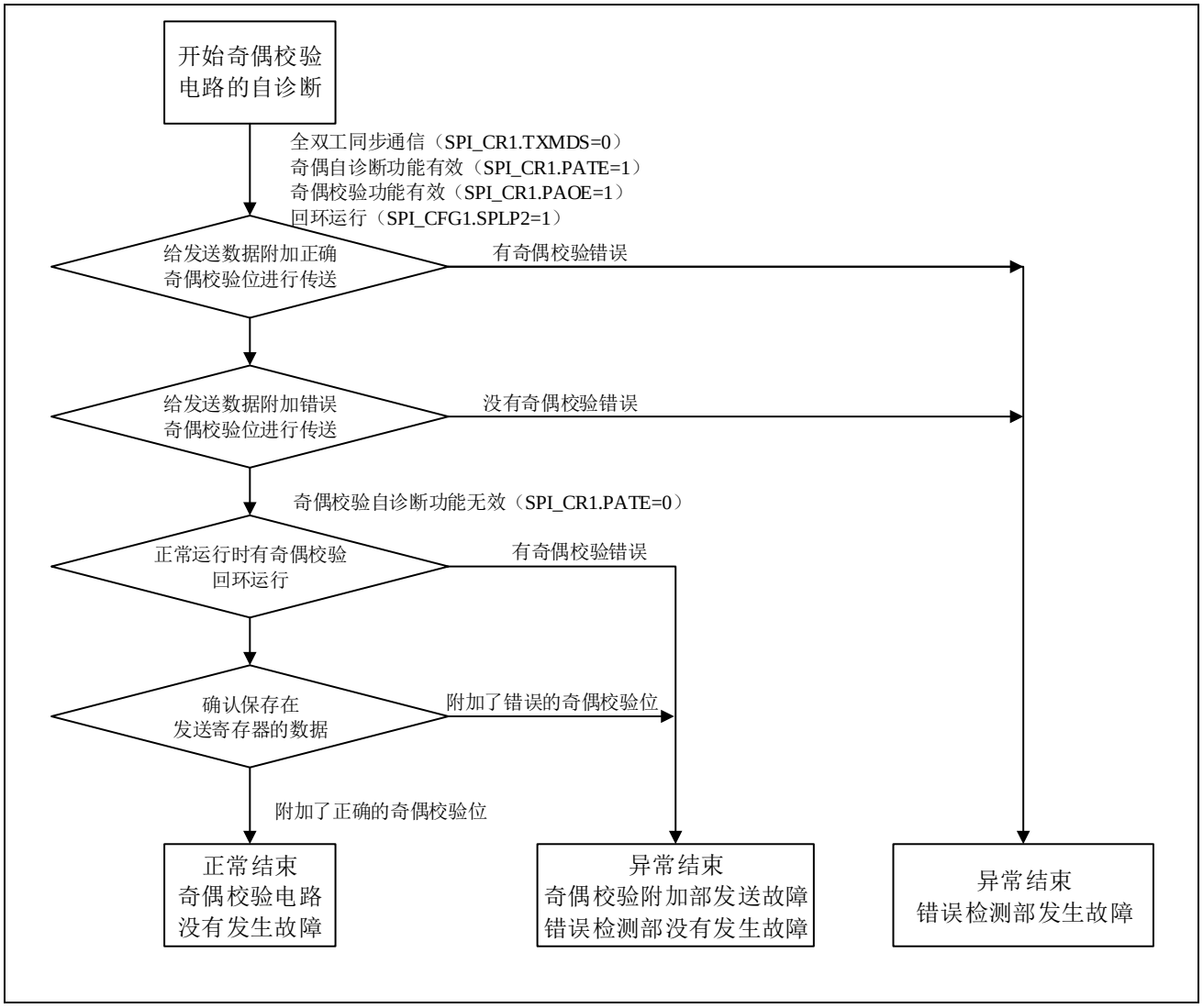


图 31-9 奇偶校验流程

31.8 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 31-7 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	• 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	• 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。	• 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	• 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状态下，接收到错误的奇偶校验位	奇偶校验错误标志有效	奇偶校验错误

表 31-7 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。

31.8.1 欠载错误

MSTR 位为 0 时，SPI 以从机状态运行，如果 SPE 置为 1 后，SS0 管脚接收到有效电平前，传送数据还未准备好，则 SPI 发生欠载错误，SPI_SR.MODFERF 和 SPI_SR.UDRERF 标志将被置为 1。

当检测到欠载错误后，SPI 将停止驱动信号输出，同时把 SPI_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。

31.8.2 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 SSi 有效电平宽度没有达到传输数据所需时间时，发生模式故障，SPI_SR.MODFERF 置 1，SPI_CR1.SPE 置为 0。需要传送时，对 SPI_SR.MODFREF 清零后，再把 SPI_CR1.SPE 置 1。

31.8.3 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”。

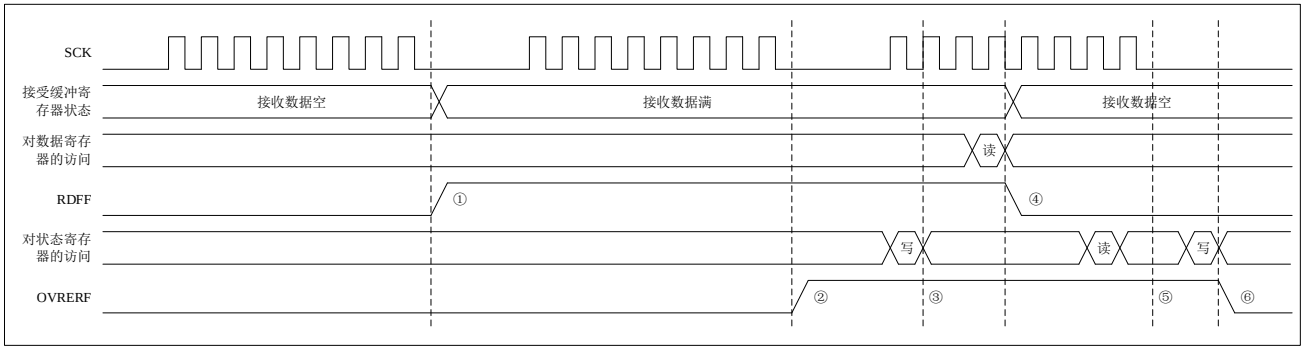


图 31-10 过载错误处理

以下说明在图中的①～⑥所示的时序中标志的运行内容：

① 在接收缓冲器空的状态下结束串行传送，SPI 正常动作，将移位寄存器的数据复制到接收缓冲寄存器中，并将 RDFF 标志置 1。

② 在接收缓冲器满的状态下结束串行传送，SPI 检测到过载错误并且将 OVRERF 标志置 1。SPI 将不会把移位寄存器的数据复制到接收缓冲器。即使 PAE 位为“1” 时，也不检测奇偶校验错误。

③ 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 31-11 图 31-10 和图 31-12。

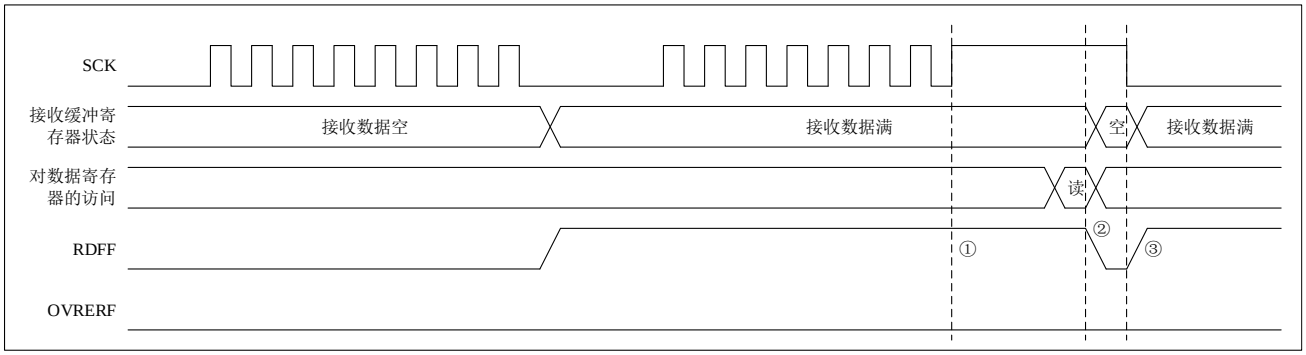


图 31-11　启用时钟自动停止功能时的动作示意图（CPHA=1）

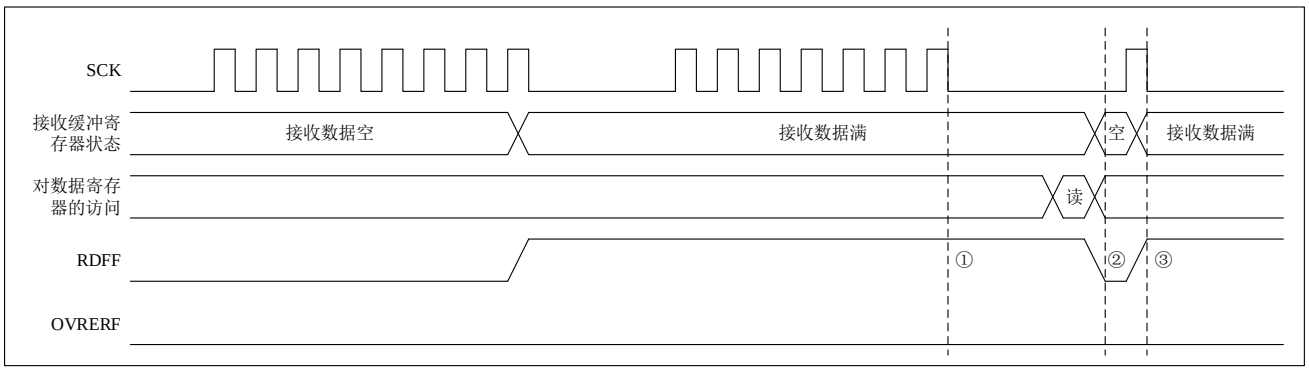


图 31-12　启用时钟自动停止功能时的动作示意图（CPHA=0）

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。

31.8.4 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.OVRERF 位为“1”的状态下，因为 SPI 不会将移位寄存器的数据复制到接收缓冲器，所以不对接收数据进行奇偶校验错误的检测。要

将 PERF 标志清零时，需要在 PERF 标志为 1 的状态下读 SPI_SR 寄存器后再给 PERF 标志写 0。

OVRERF 标志和 PERF 标志的运行实例如下图 31-13 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。

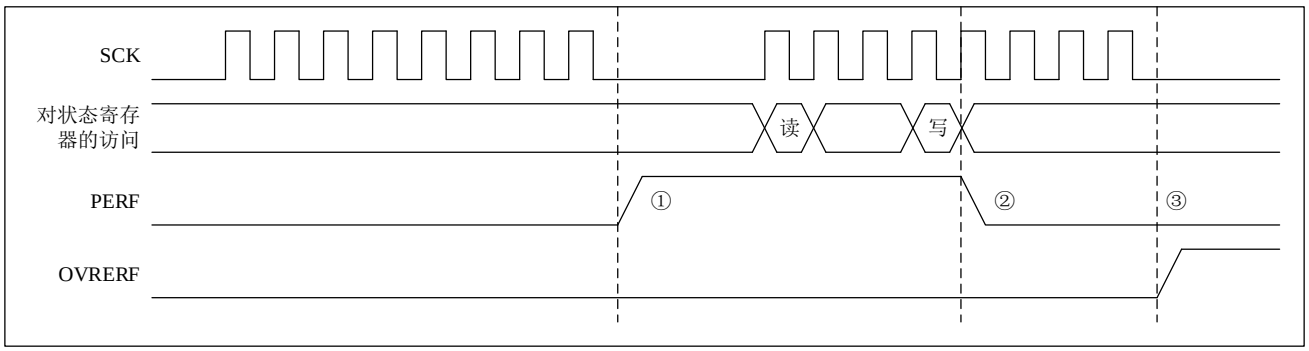


图 31-13 奇偶校验错误

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。

31.9 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。

31.9.1 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。

31.9.2 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。

31.10 中断

SPI 的接收缓冲器满、发送缓冲器空、模式故障(过载、欠载、奇偶校验错误)和 SPI 空闲即可做为中断源也可做为内部触发源，传送完成只可以作为内部触发源。

过载、欠载和奇偶校验错误的中断被集成为 SPI 错误中断 SPEI，所以需要通过标志判断实际发生的中断源。SPI 中断/内部触发源的具体说明如表 31-8 所示。一旦条件成立，就产生相应的中断/内部触发请求。对于接收缓冲器满和发送缓冲器空的中断源，需要通过数据传送改变缓冲器状态来清除。

中断/内部触发源	略称	条件
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
发送缓冲器空	SPTI	在SPI_CR1.TXIE位为“1”的状态下发送缓冲器变空时
SPI错误（过载、欠载、奇偶校验错误）	SPEI	在SPI_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
SPI空闲	SPII	在SPI_CR1.IDIE位为“1”的状态下 主机模式：数据传输完成或者SPI_CR1.SPE由1写为0。 从机模式：SPI_CR1.SPE由1写为0。
传送完成	SPEND	主机模式：数据传输完成。 从机模式：四线式，SSL管脚为无效电平时；三线式，SCK管脚时钟达到数据传输所需时钟数

表 31-8 SPI 中断源说明

31.11 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）】。

31.12 寄存器说明

寄存器一览

寄存器基准地址：SPI1_BASE: 0x4001C000； SPI2_BASE: 0x4001C400

SPI3_BASE: 0x4001C800； SPI4_BASE: 0x40020000

SPI5_BASE: 0x40020400； SPI6_BASE: 0x40020800

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

31.12.1 SPI 数据寄存器（SPI_DR）

复位值：0x0000

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

位	标记	位名	功能	读写
b31~b0	SPD[31:0]	串行数据	SPI数据存储	R/W

31.12.2 SPI 控制寄存器（SPI_CR1）

复位值：0x0000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PAE	PAOE	PATE	MODFE	IDIE	RXIE	TXIE	EIE	CSUSPE	SPE	SPLPBK2	SPLPBK	MSTR	—	TXMDS	SPIMDS

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15	PAE	奇偶校验允许	0：发送数据不附加奇偶校验位，接收数据不进行奇偶校验 1：发送数据附加奇偶校验位，接收数据进行奇偶校验（SPI_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	SPIMDS	SPI模式选择	0：SPI运行（4线式）	R/W
			1：时钟同步运行（3线式）	

31.12.3 SPI 通信配置寄存器 1（SPI_CFG1）

复位值：0x0010

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MIDI[2:0]			-	MSSDL[2:0]				MSSI[2:0]			-	-	-	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	SS3PV	SS2PV	SS1PV	SS0PV	-	SPRDTD	-	-	-	-	FTHLV[1:0]	

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“0”,写入时写“0”	R/W
b30~b28	MIDI[2:0]	主机下次存取数据间隔闲置时间设定位	b30~b28 0 0 0：1 个SCK+2个PCLK1 0 0 1：2 个SCK+2个PCLK1 0 1 0：3 个SCK+2个PCLK1 0 1 1：4 个SCK+2个PCLK1 1 0 0：5 个SCK+2个PCLK1 1 0 1：6 个SCK+2个PCLK1 1 1 0：7 个SCK+2个PCLK1 1 1 1：8 个SCK+2个PCLK1	R/W
b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26~b24	MSSDL[2:0]	主机SS无效延迟设定位	b26~b24 0 0 0：1 个SCK 0 0 1：2 个SCK 0 1 0：3 个SCK 0 1 1：4 个SCK 1 0 0：5 个SCK 1 0 1：6 个SCK 1 1 0：7 个SCK 1 1 1：8 个SCK	R/W
b23	Reserved	-	读出时为“0”,写入时写“0”	R/W
b22~b20	MSSI[2:0]	主机SS闲置时间设定位	b22~b20 0 0 0：1 个SCK 0 0 1：2 个SCK 0 1 0：3 个SCK 0 1 1：4 个SCK 1 0 0：5 个SCK 1 0 1：6 个SCK 1 1 0：7 个SCK 1 1 1：8 个SCK	R/W
b19~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	SPRDTD	数据寄存器读取对象 选择	0：SPI_DR 读接收缓冲器 1：SPI_DR 读发送缓冲器(必须在TDEF=1时才可以读取）	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

31.12.4 SPI 状态寄存器（SPI_SR）

复位值：0x0020

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	RDFF	—	TDEF	UDRERF	PERF	MODFERF	IDLNF	OVRERF

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	RDFF	接收缓冲器满标志	0：SPI_DR寄存器无数据 1：SPI_DR寄存器有数据 硬件置位，清零	R
b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5	TDEF	发送缓冲器空标志	0：发送缓冲器有数据 1：发生缓冲器无数据 硬件置位，清零	R
b4	UDRERF	欠载错误标志	0：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

31.12.5 SPI 通信配置寄存器 2（SPI_CFG2）

复位值：0x0F1D

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MSSIE	MSSDLE	MIDIE	LSBF	DSIZE[3:0]				SSA[2:0]			MBR[2:0]			CPOL	CPHA

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15	MSSIE	SCK延迟允许	0：SCK 延迟为1 个SCK	R/W
			1：SCK 延迟为SSI的设定值	
b14	MSSDLE	SS无效延迟允许	0：SS 无效延迟为1 个SCK	R/W
			1：SS 无效延迟为SSDL的设定值	
b13	MIDIE	SPI下次存取延迟允许	0：下次存取延迟为1 个SCK+2个PCLK1	R/W
			1：下次存取延迟为MIDI的设定值	
b12	LSBF	SPI LSB first 位	0：MSB first	R/W
			1：LSB first	
b11~b8	DSIZE[3：0]	SPI数据长度设定位	b11~b8	R/W
			0 0 0 0：4位	
			0 0 0 1：5位	
			0 0 1 0：6位	
			0 0 1 1：7位	
			0 1 0 0：8位	
			0 1 0 1：9位	
			0 1 1 0：10位	
			0 1 1 1：11位	
			1 0 0 0：12位	
			1 0 0 1：13位	
			1 0 1 0：14位	
			1 0 1 1：15位	
			1 1 0 0：16位	
			1 1 0 1：20位	
			1 1 1 0：24位	
			1 1 1 1：32位	
b7~b5	SSA[2：0]	SS信号有效设定位	b7~b5	R/W
			0 0 0：SS0	
			0 0 1：SS1	
			0 1 0：SS2	
			0 1 1：SS3	

			1 x x：禁止设定	
b4~b2	MBR[2：0]	位速率分频设定位	b4~b2	R/W
			0 0 0：选择PCLK1的2分频	
			0 0 1：选择PCLK1的4分频	
			0 1 0：选择PCLK1的8分频	
			0 1 1：选择PCLK1的16分频	
			1 0 0：选择PCLK1的32分频	
			1 0 1：选择PCLK1的64分频	
			1 1 0：选择PCLK1的128分频	
			1 1 1：选择PCLK1的256分频	
b1	CPOL	SCK极性设定位	0：空闲时的SCK为Low电平	R/W
			1：空闲时的SCK为High电平	
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	R/W
			1：在奇数边沿数据发生变化，在偶数边沿进行数据采样	

32 四线式串行外设接口（QSPI）

32.1 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	• 支持扩展SPI，二线式SPI和四线式SPI等多种协议 • 支持SPI模式0和SPI模式3 • 地址线宽度可选择8位/16位/24位/32位
时序调整	可通过时序调整以支持各种串行闪存
闪存读取	• 支持多种读取方式 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 • 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 32-1 QSPI 主要规格

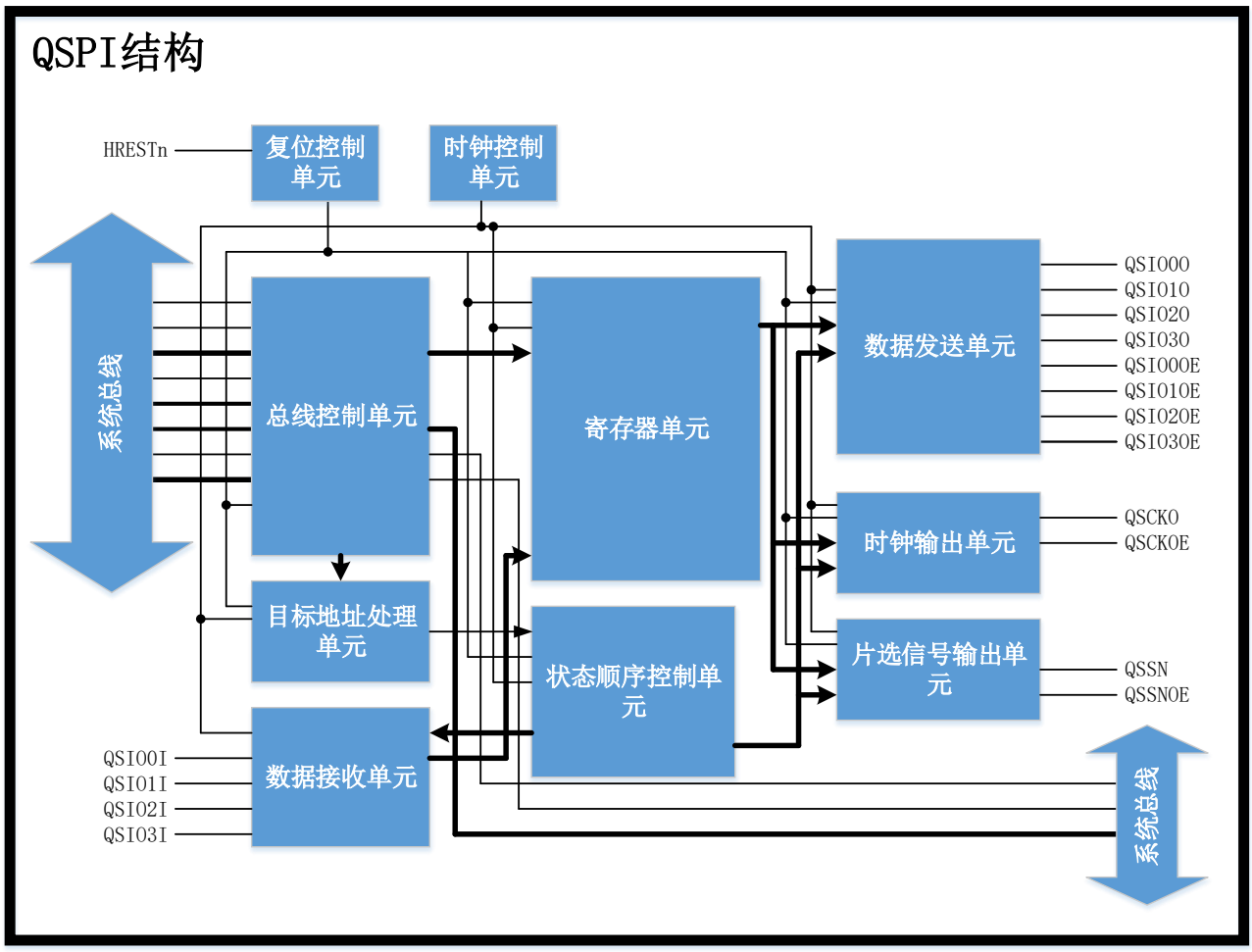


图 32-1 QSPI 的模块构成图

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
QSIO0	输入/输出	数据线0
QSIO1	输入/输出	数据线1
QSIO2	输入/输出	数据线2
QSIO3	输入/输出	数据线3

表 32-2 QSPI 管脚

32.2 内存映射

32.2.1 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。

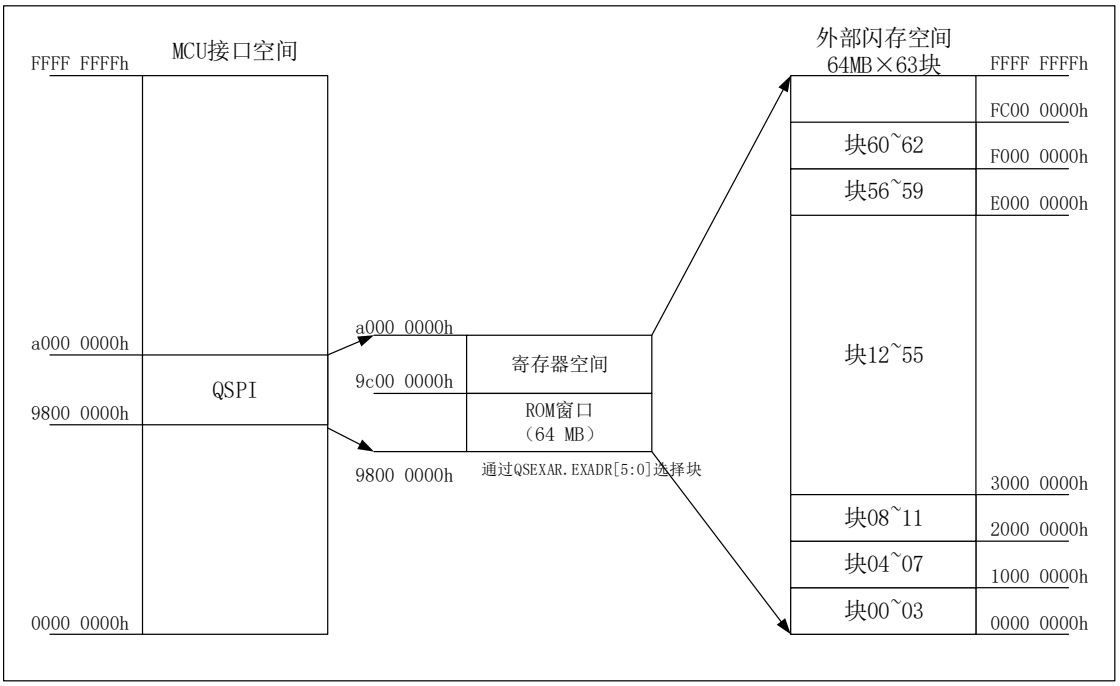


图 32-2 默认区域设定及 AHB 总线空间内存映射关系图

32.2.2 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。

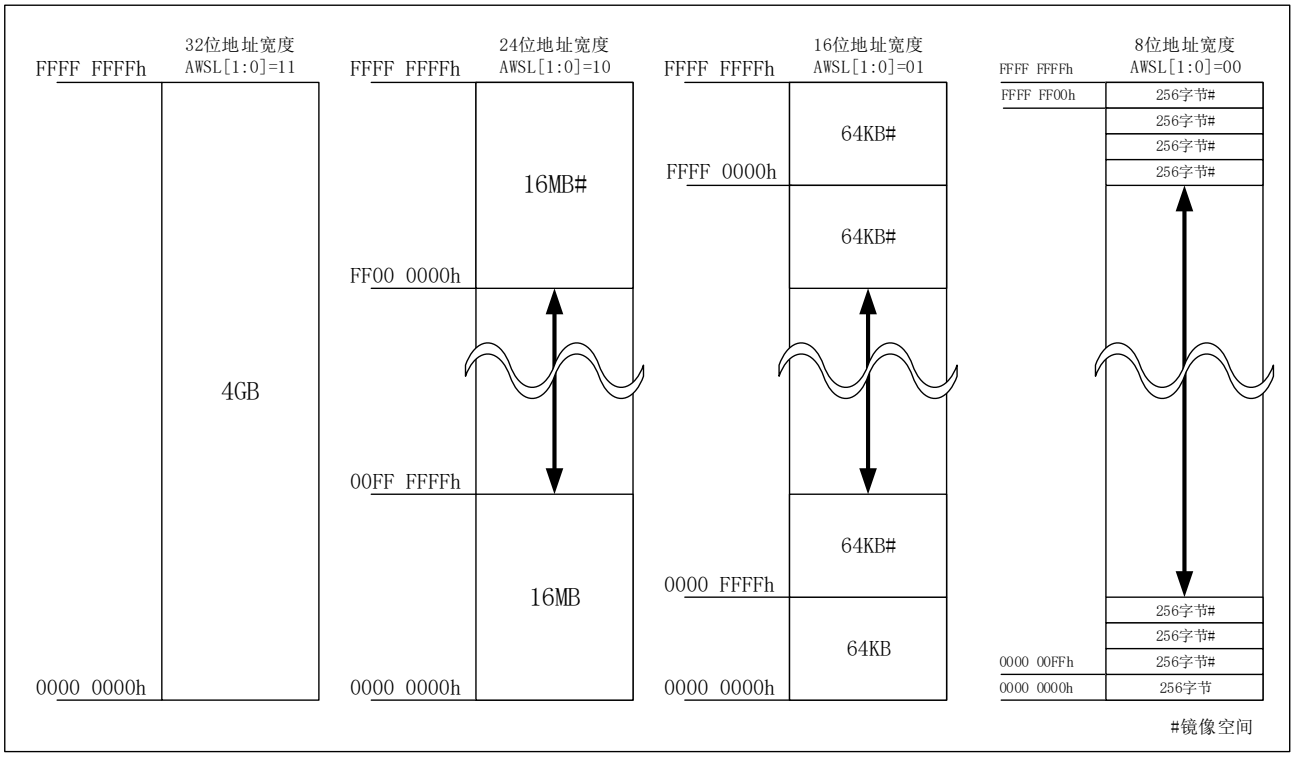


图 32-3 QSPI-ROM 空间内存映像图

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。

32.3 QSPI 总线

32.3.1 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。

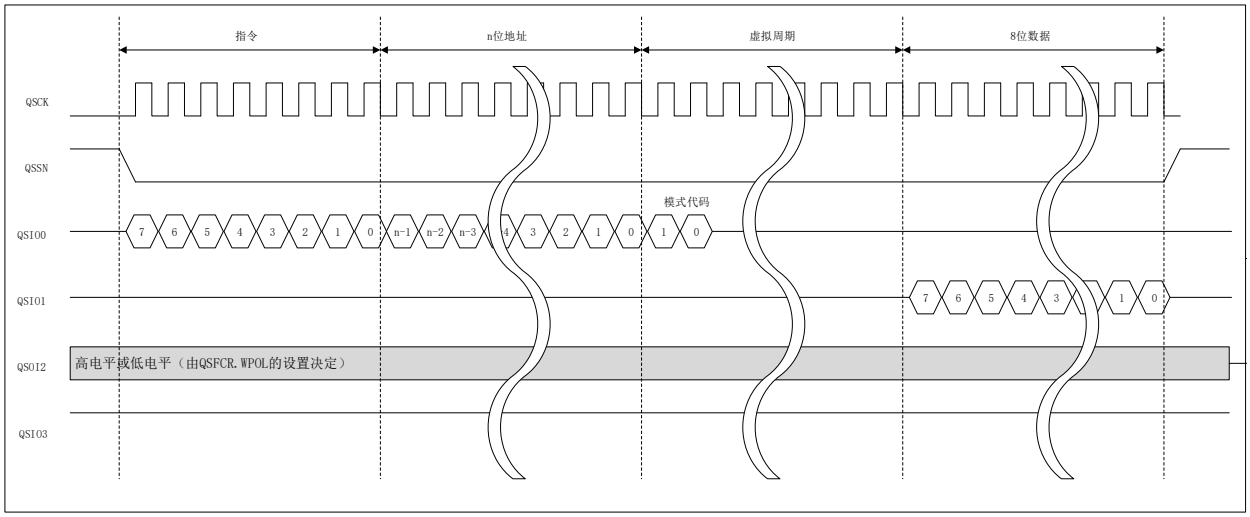


图 32-4 扩展式 SPI 协议动作示意图 1（快速读模式）

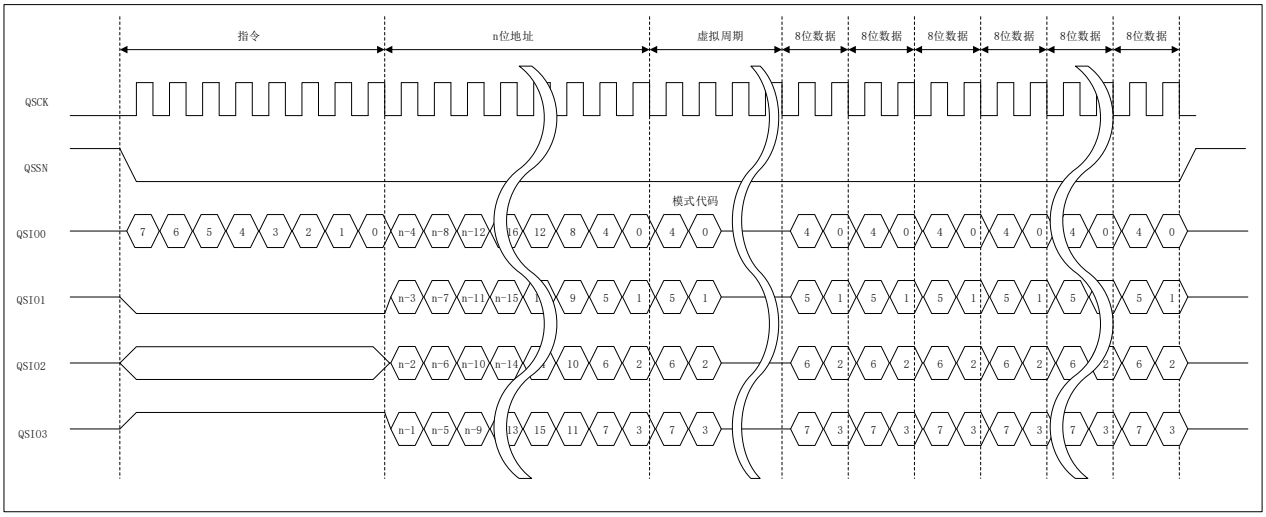


图 32-5 扩展式 SPI 协议动作示意图 2（四线式输入输出快速读模式）

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。

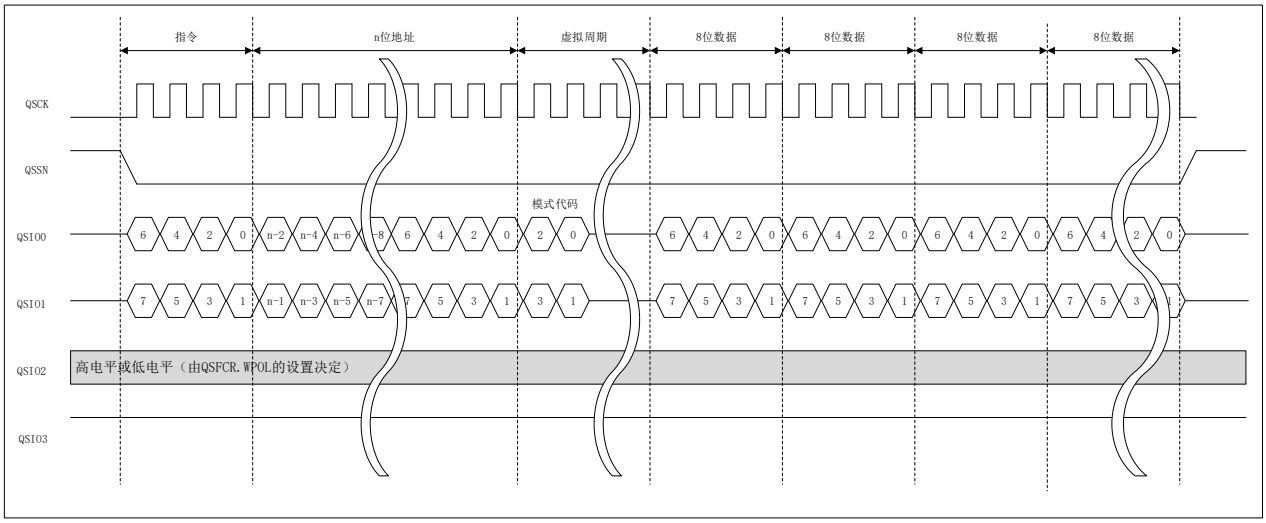


图 32-6　二线式 SPI 协议动作示意图（快速读模式）

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。

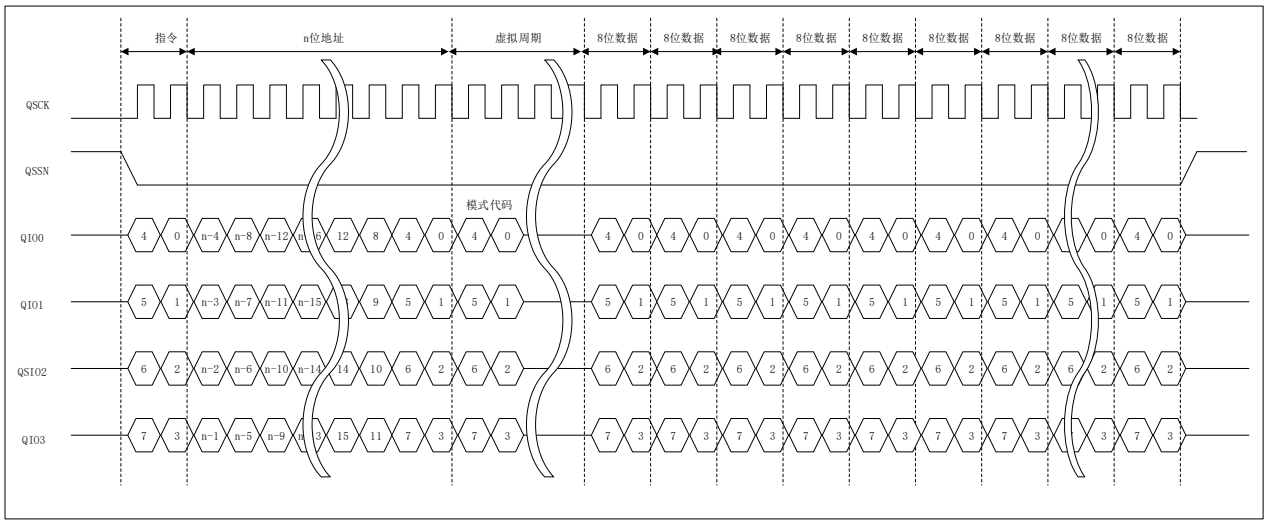


图 32-7　四线式 SPI 协议动作示意图（快速读模式）

32.3.2 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。

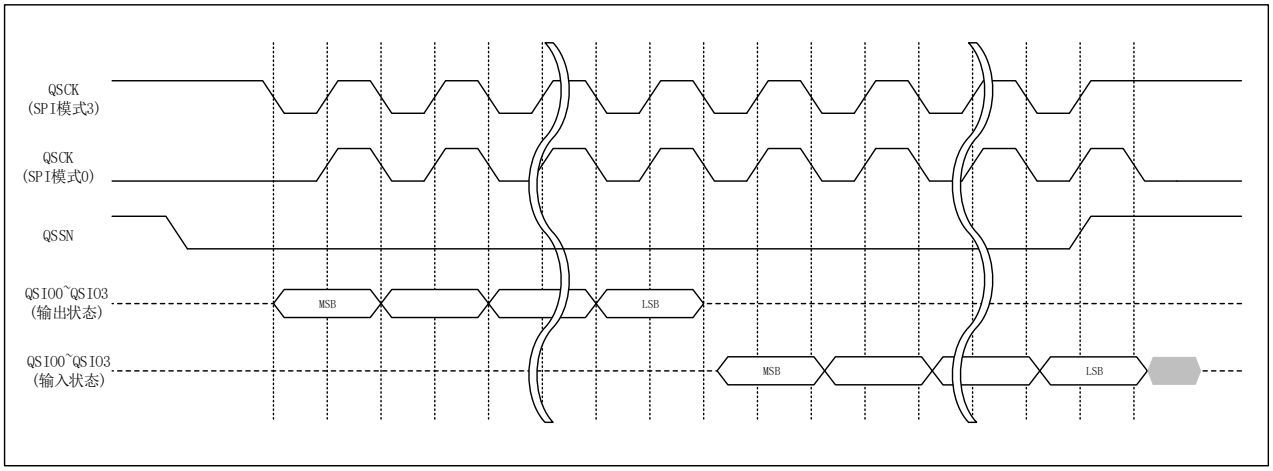


图 32-8 串行接口的基本时序图

32.4 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。

32.4.1 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 (HCLK=200MHz)	DIV[5:0]	分频比	实际动作频率 (HCLK=200MHz)
6'b000000	2	100.00	6'b100000	33	6.06
6'b000001	2	100.00	6'b100001	34	5.88
6'b000010	3	66.67	6'b100010	35	5.71
6'b000011	4	50.00	6'b100011	36	5.56
6'b000100	5	40.00	6'b100100	37	5.41
6'b000101	6	33.33	6'b100101	38	5.26
6'b000110	7	28.57	6'b100110	39	5.13
6'b000111	8	25.00	6'b100111	40	5.00
6'b001000	9	22.22	6'b101000	41	4.88
6'b001001	10	20.00	6'b101001	42	4.76
6'b001010	11	18.18	6'b101010	43	4.65
6'b001011	12	16.67	6'b101011	44	4.55
6'b001100	13	15.38	6'b101100	45	4.44
6'b001101	14	14.29	6'b101101	46	4.35
6'b001110	15	13.33	6'b101110	47	4.26
6'b001111	16	12.50	6'b101111	48	4.17
6'b010000	17	11.76	6'b110000	49	4.08
6'b010001	18	11.11	6'b110001	50	4.00
6'b010010	19	10.53	6'b110010	51	3.92

6'b010011	20	10.00	6'b110011	52	3.85
6'b010100	21	9.52	6'b110100	53	3.77
6'b010101	22	9.09	6'b110101	54	3.70
6'b010110	23	8.70	6'b110110	55	3.64
6'b010111	24	8.33	6'b110111	56	3.57
6'b011000	25	8.00	6'b111000	57	3.51
6'b011001	26	7.69	6'b111001	58	3.45
6'b011010	27	7.41	6'b111010	59	3.39
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 32-3 QSPI 总线基准时钟选择一览表

32.4.2 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。

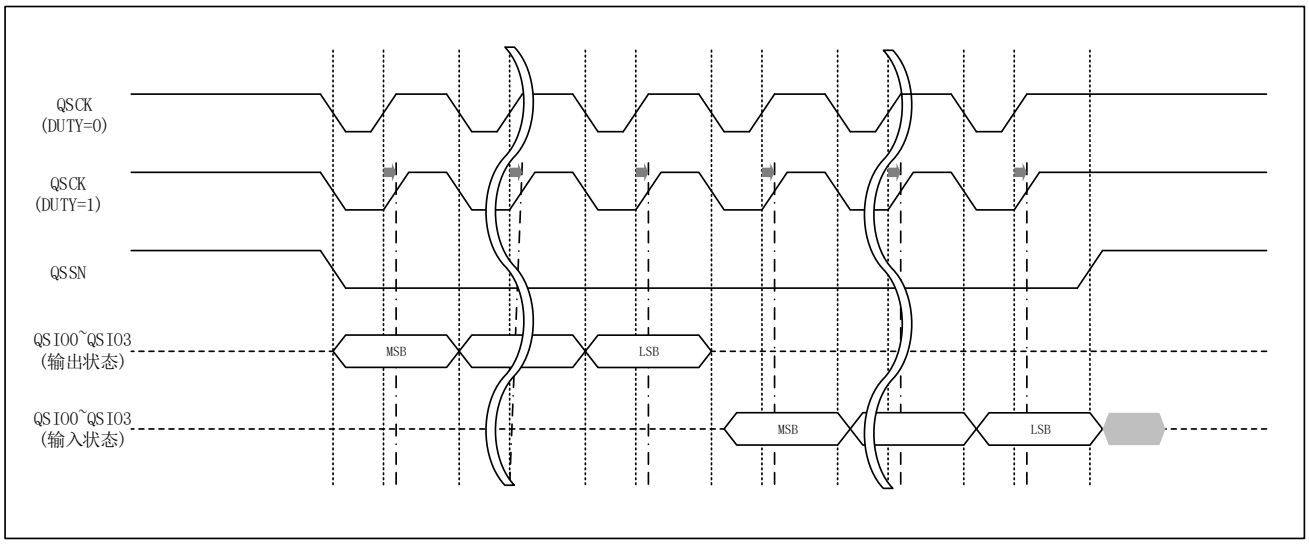


图 32-9 基准时钟选择 HCLK 三分频时输出时钟占空比补正示意图

32.4.3 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。

32.4.4 QSSN 的建立时间

从 QSSN 信号开始输出低电平（即变为有效状态）到 QSCK 信号输出第一个上升沿之

间的时间称为 QSSN 的建立时间，该时间可以通过寄存器设置进行配置以满足外接串行闪存的要求。设置寄存器 QSFCR 的 SSNLD 位来选择 QSSN 的建立时间是 0.5 个或是 1.5 个 QSPI 基准时钟周期。该设置同样可用于配置数据管脚从数据输出许可到 QSCK 信号输出第一个上升沿之间的建立时间，可根据需要合理运用。

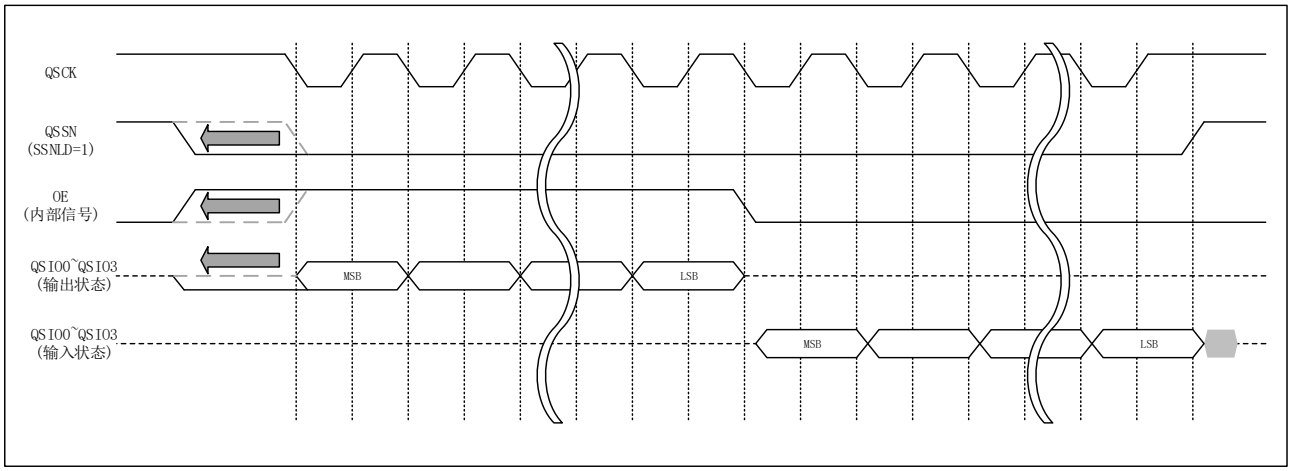


图 32-10 QSSL 建立时间配置示意图

32.4.5 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。

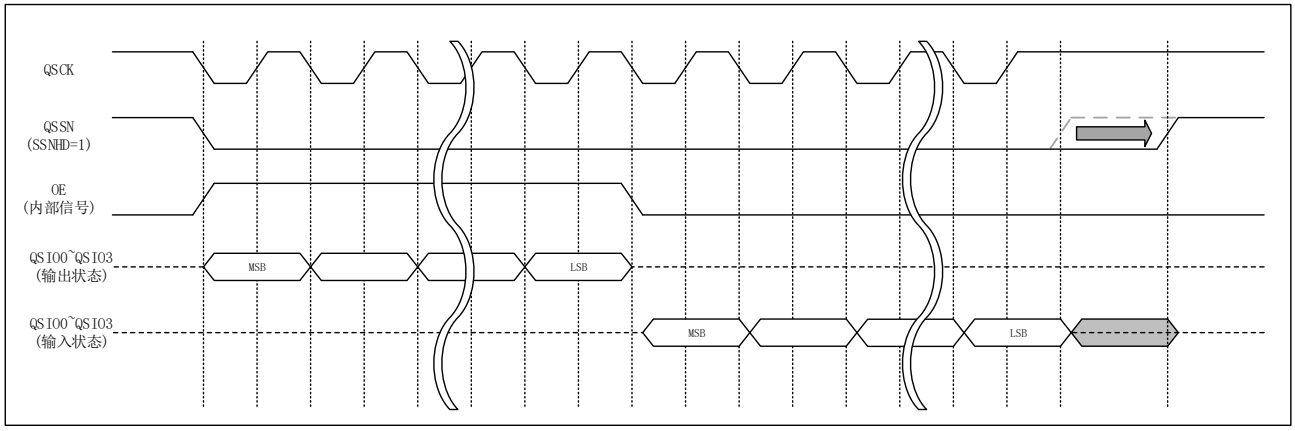


图 32-11 QSSN 保持时间配置示意图

32.4.6 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。

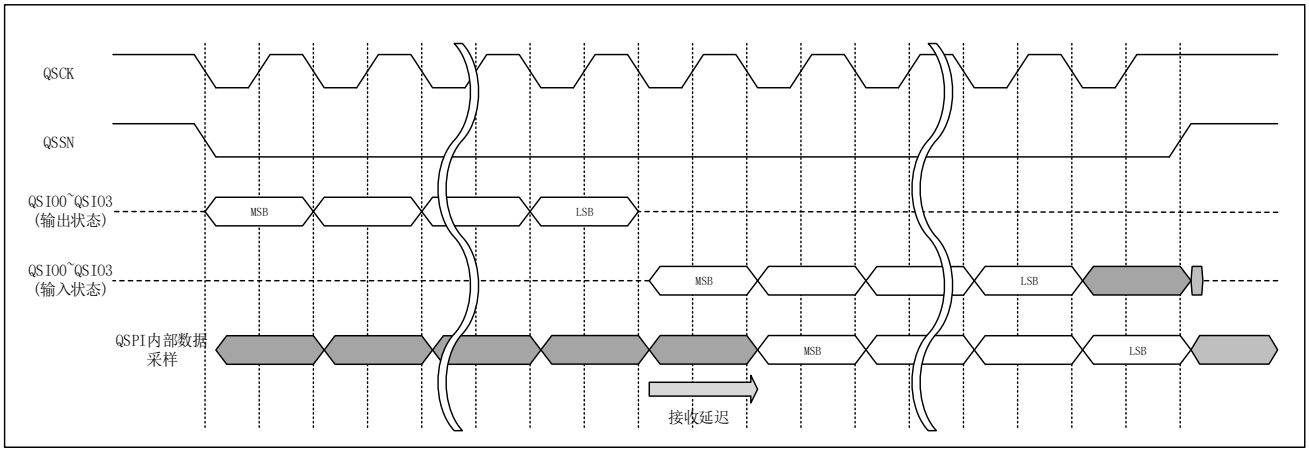


图 32-12 数据接收延迟示意图

32.5 用于 ROM 访问的 SPI 指令介绍

32.5.1 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 32-4 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。

32.5.2 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。

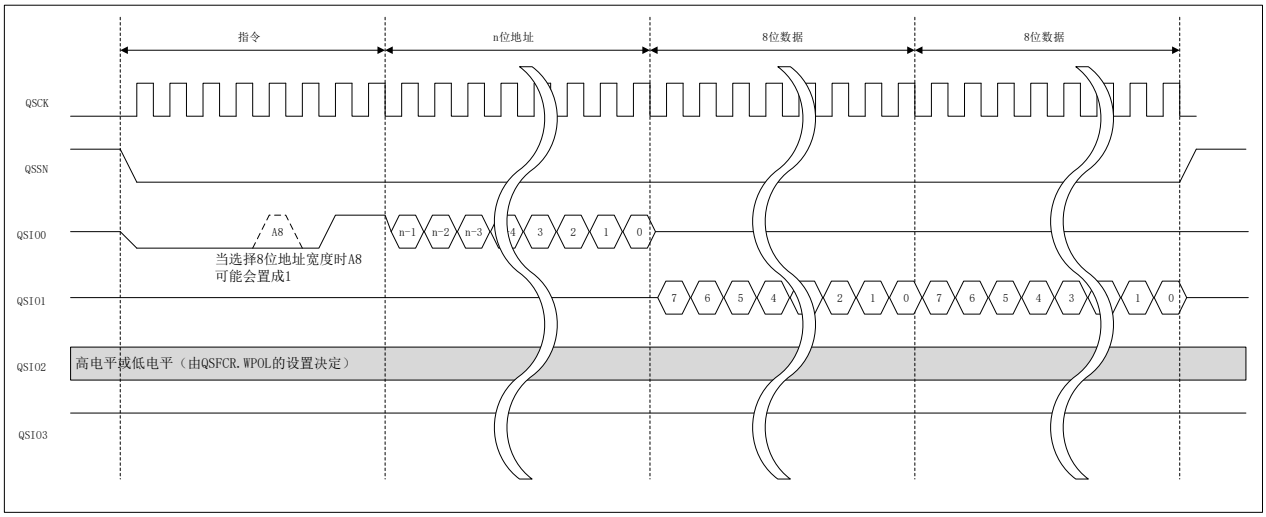


图 32-13 标准读总线周期示意图

32.5.3 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 32.7 XIP 控制。

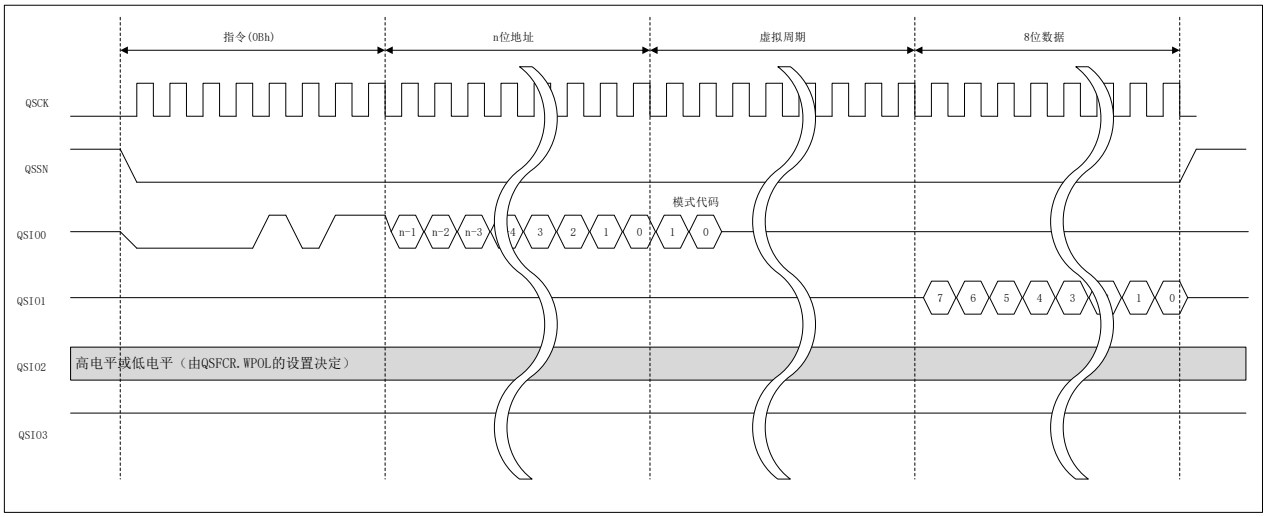


图 32-14　快速读总线周期示意图

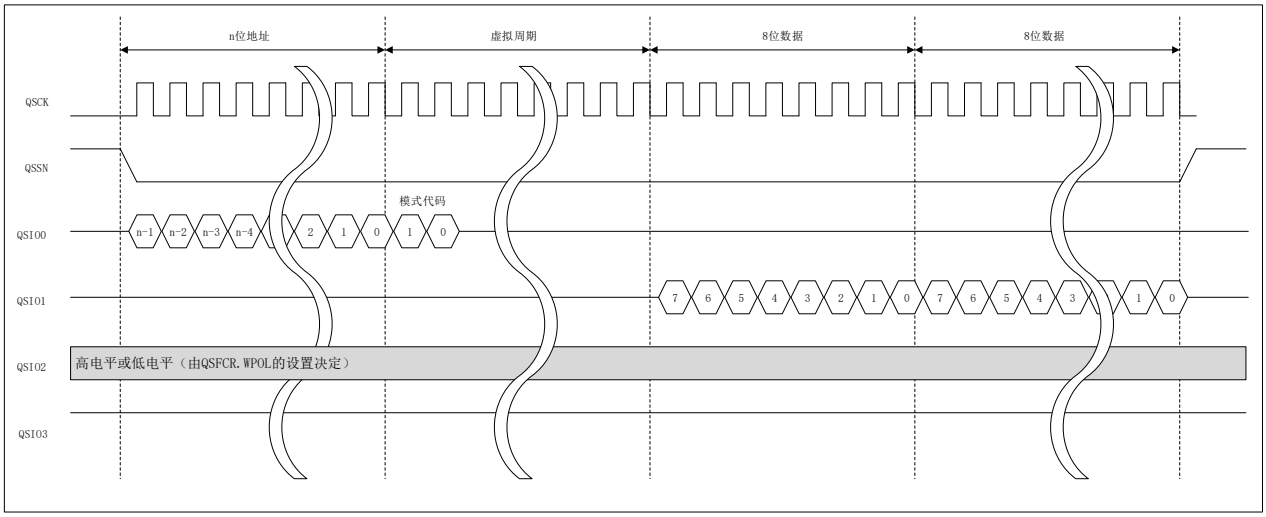


图 32-15　选择 XIP 模式的快速读总线周期示意图

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。

32.5.4 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 32.7 XIP 控制。

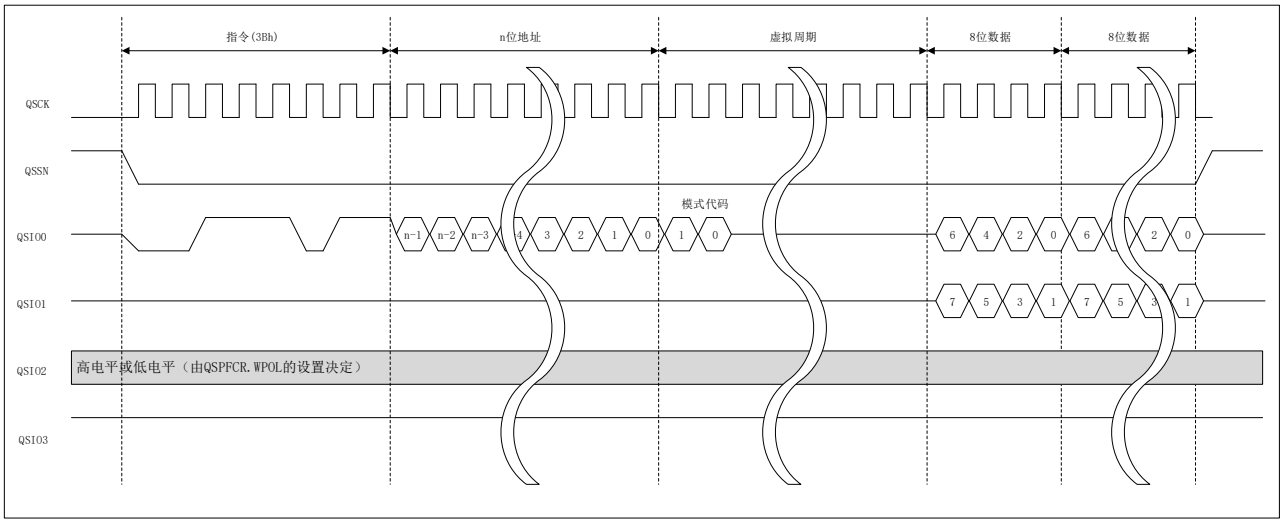


图 32-16 二线式输出快速读总线周期示意图

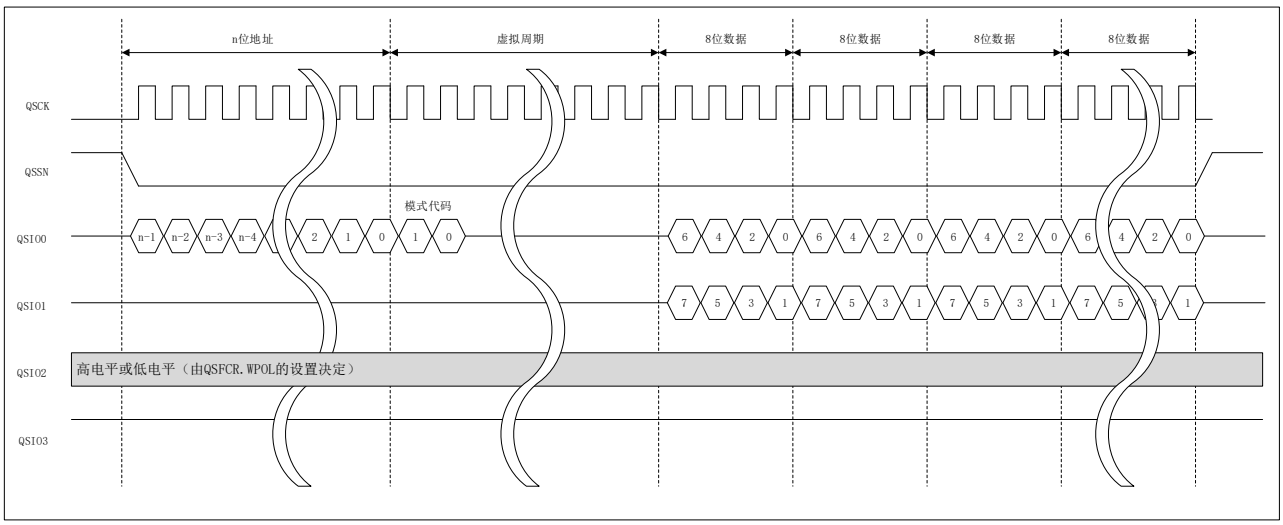


图 32-17　选择 XIP 模式的二线式输出快速读总线周期示意图

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。

32.5.5 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 32.7 XIP 控制。

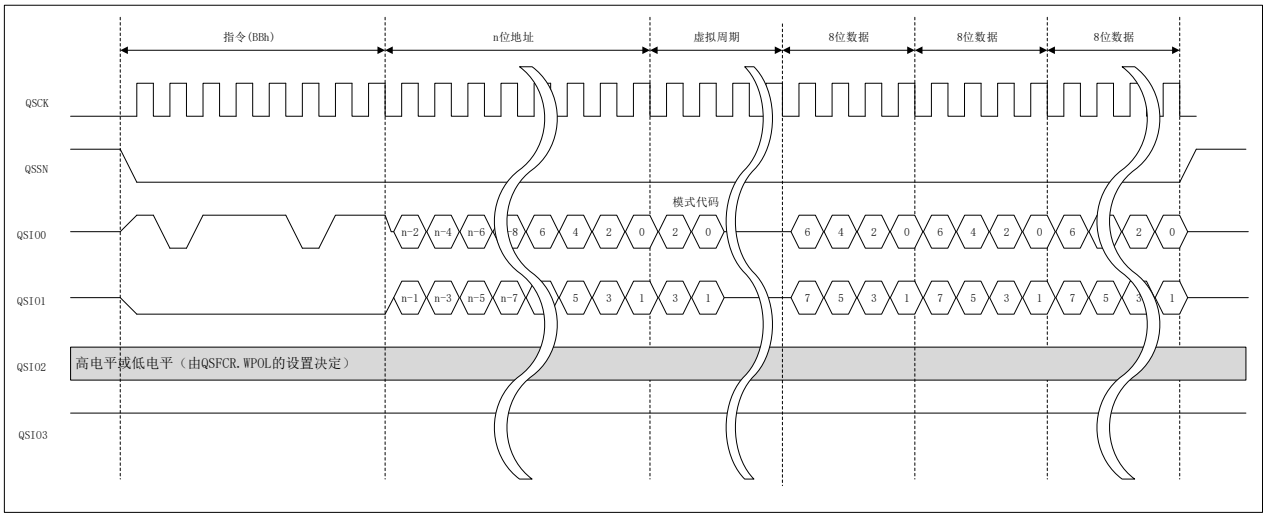


图 32-18　二线式输入输出快速读总线周期示意图

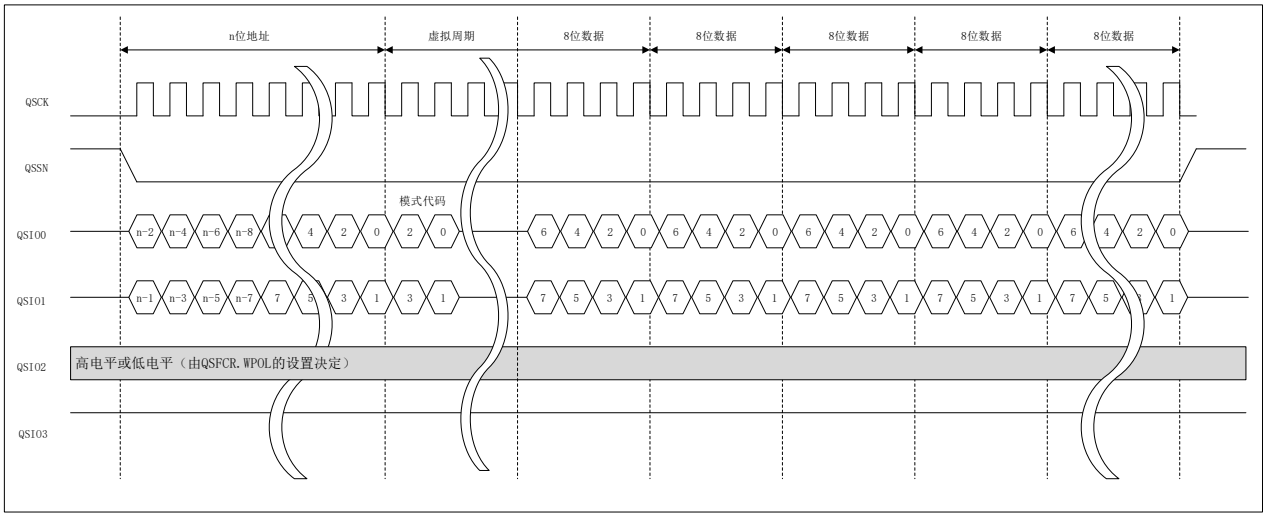


图 32-19　选择 XIP 模式的二线式输入输出快速读总线周期示意图

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。

32.5.6 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 32.7 XIP 控制。

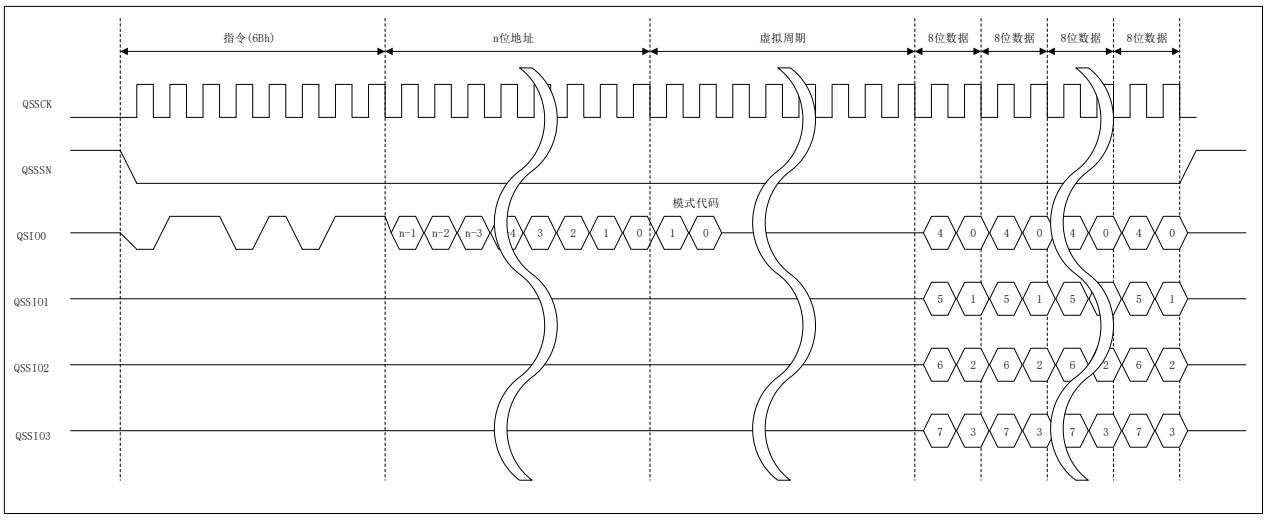


图 32-20 四线式输出快速读总线周期示意图

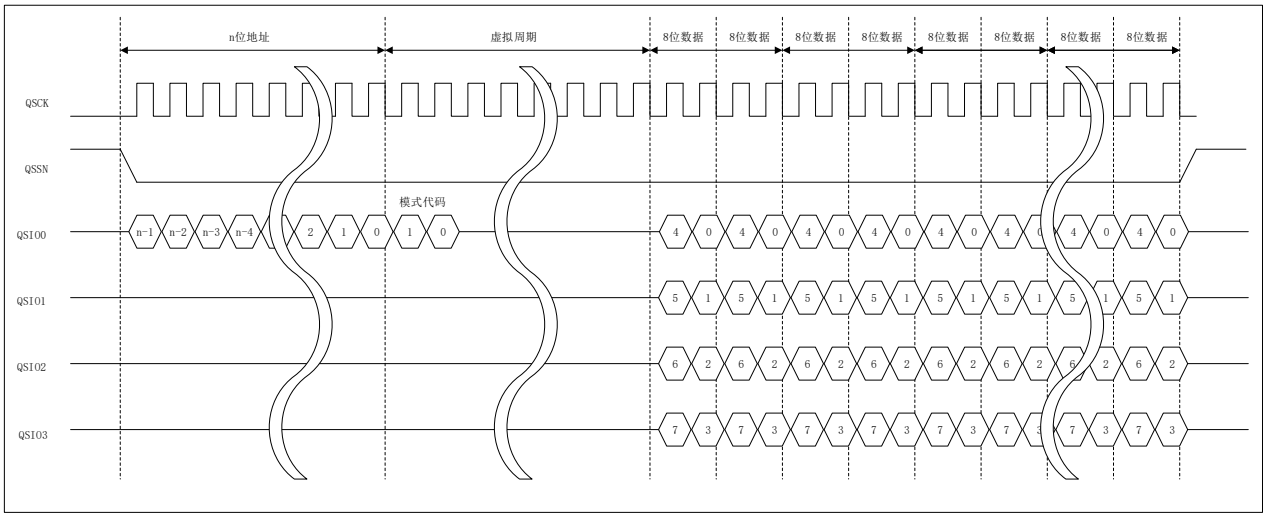


图 32-21　选择 XIP 模式的四线式输出快速读总线周期示意图

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。

32.5.7 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 32.7 XIP 控制。

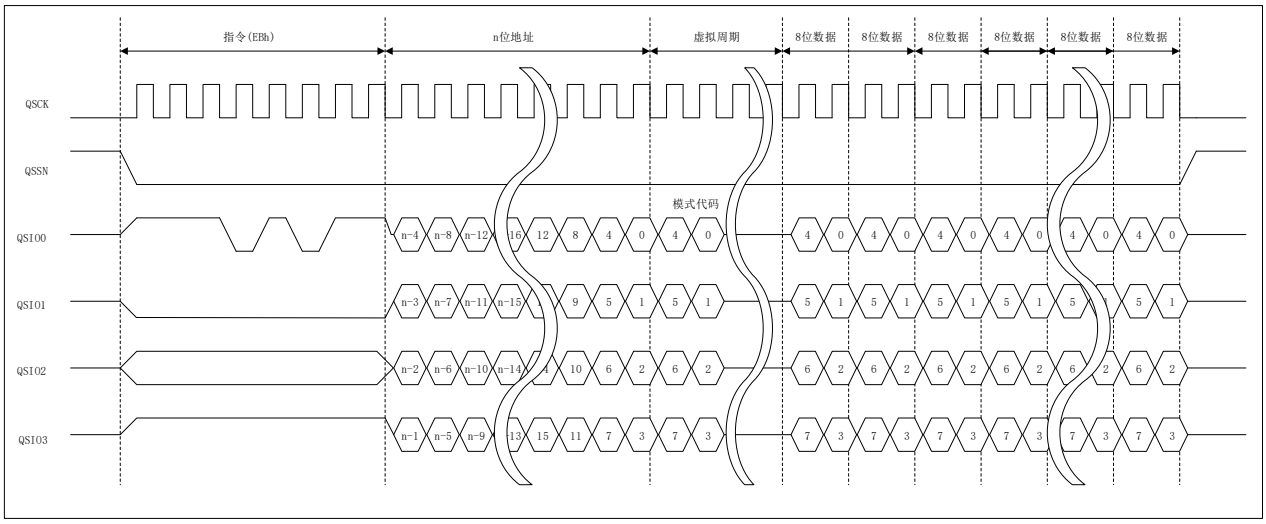


图 32-22　四线式输入输出快速读总线周期示意图

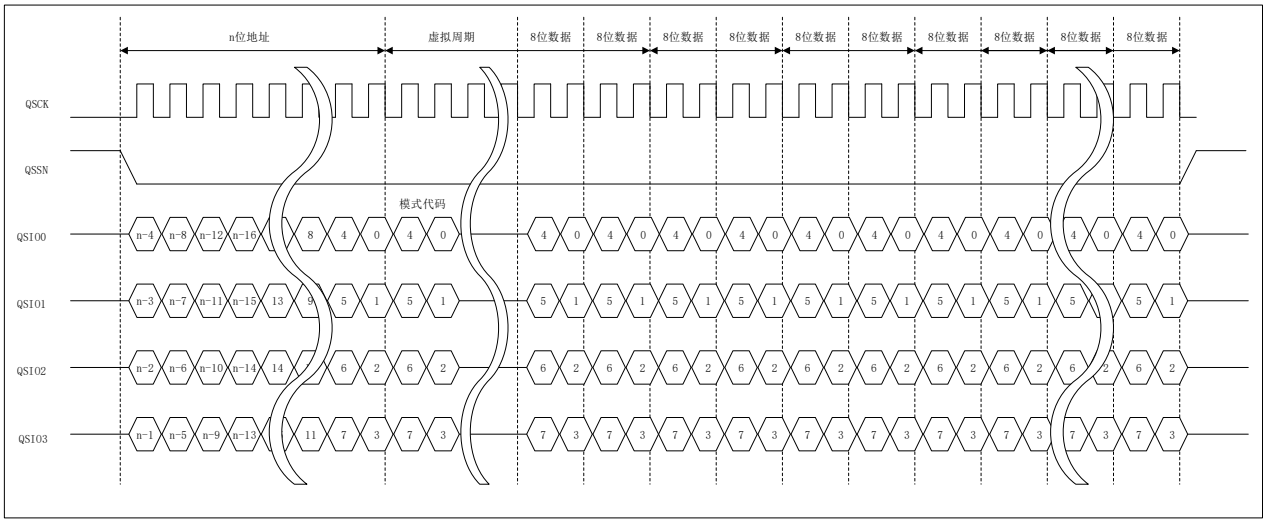


图 32-23　选择 XIP 模式的四线式输入输出快速读总线周期示意图

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。

32.5.8 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）。

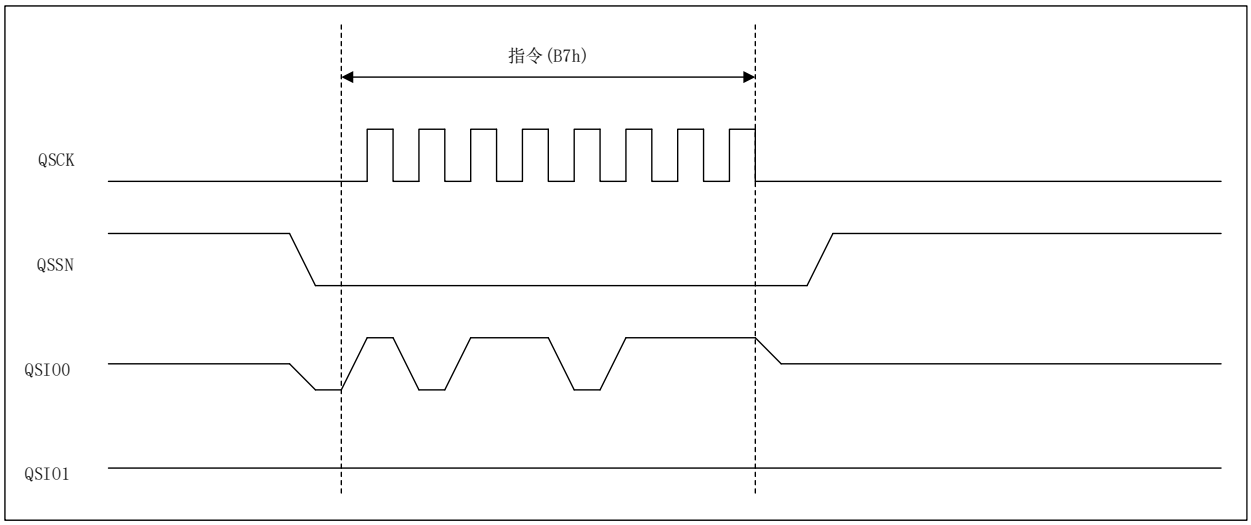


图 32-24 进入 4-Byte 模式指令总线周期示意图

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。

32.5.9 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）。

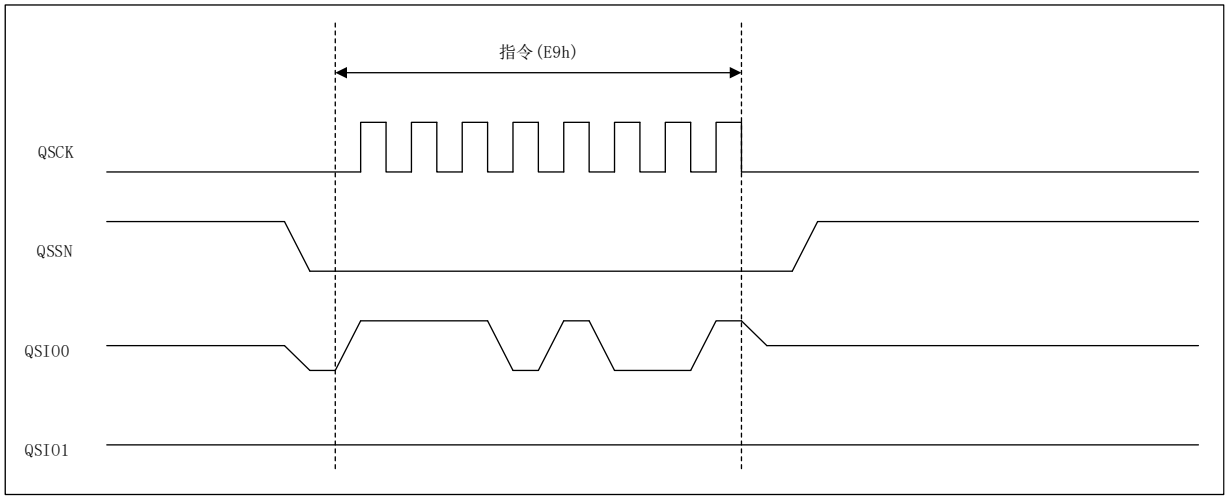


图 32-25 退出 4-Byte 模式指令总线周期示意图

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。

32.5.10 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）。

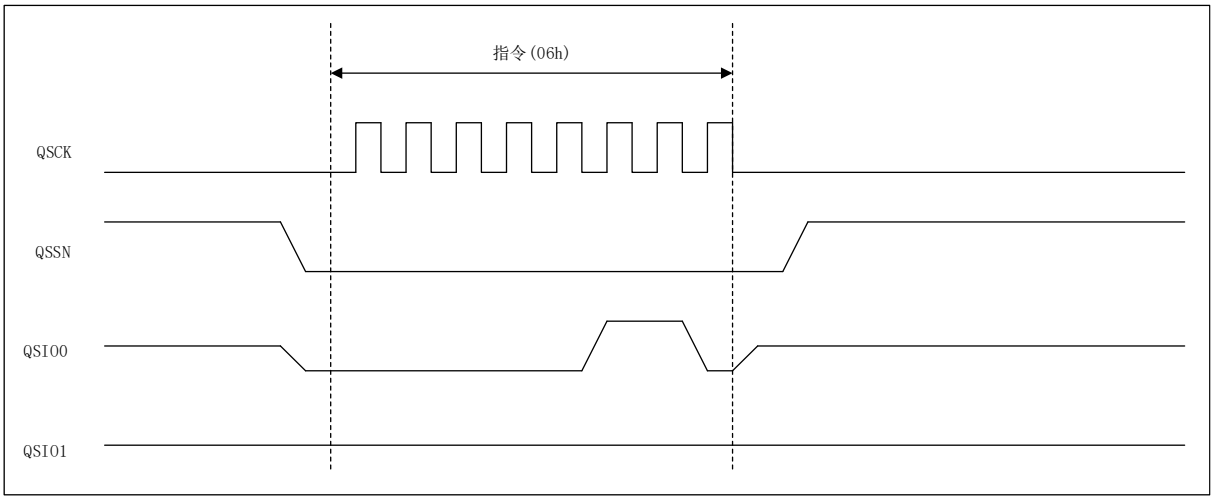


图 32-26 写许可指令总线周期示意图

32.6 QSPI 总线周期的安排

32.6.1 独立转换的单一闪存读取

对于 ROM 的单一读取指令会独立的从芯片内部总线周期一对一的转换为 QSPI 总线周期。当一个 ROM 的读取总线周期被检测到时，QSSN 信号会置为有效状态，从而启动一个 QSPI 总线周期。当接收完串行闪存的数据后，QSSN 信号变成无效状态，该 QSPI 总线周期宣告完成。

当另一个 ROM 的读取总线周期被检测到时，QSSN 信号会在确保无效保持时间已经超过最小无效保持宽度后再度置为有效状态，新一个 QSPI 总线周期开始。

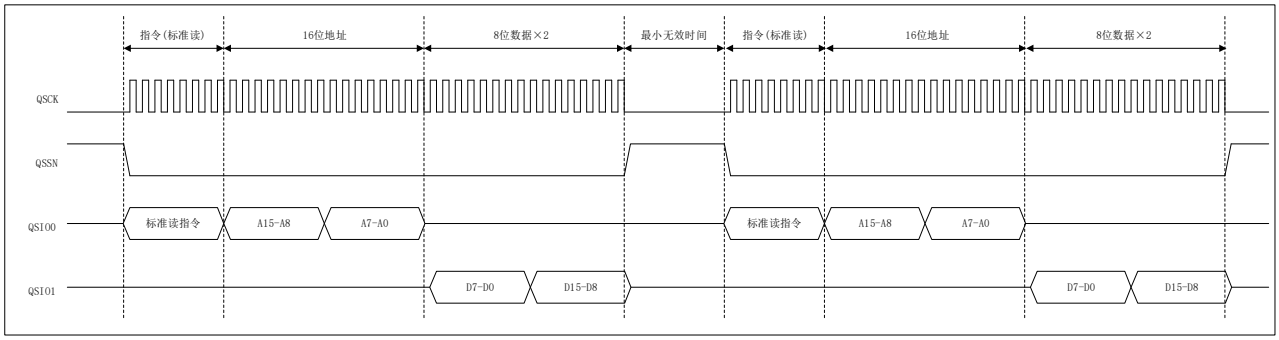


图 32-27 独立转换的单一闪存数据读取操作示意图

32.6.2 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。

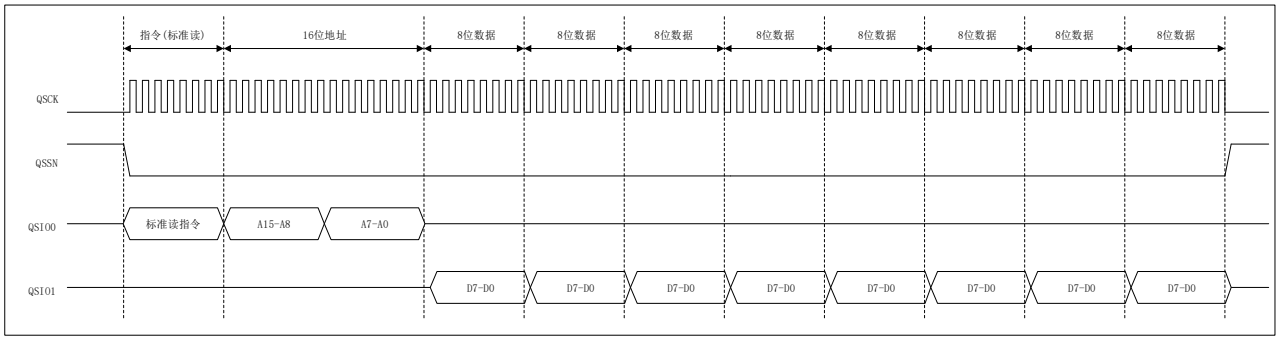


图 32-28　预读取功能有效时数据读取操作示意图

32.6.3 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。

32.6.4 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。

32.6.5 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。

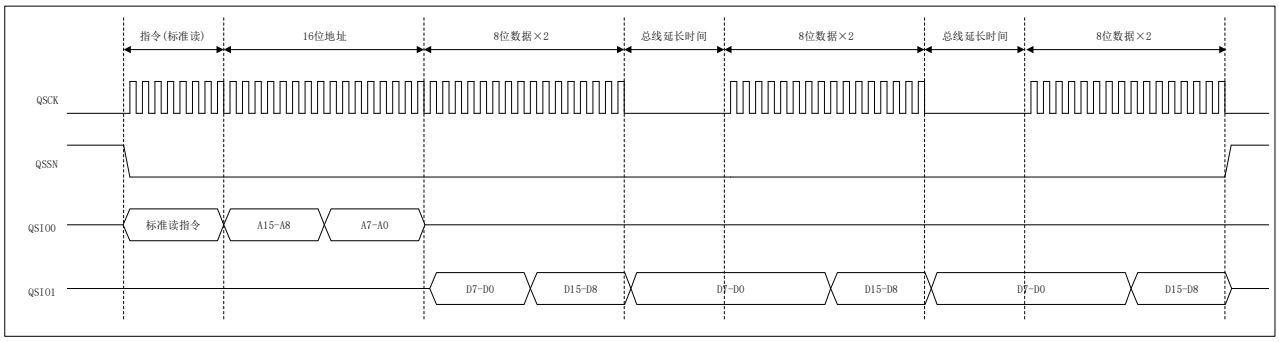


图 32-29 使用 QSPI 总线周期延长功能的数据读取操作示意图

32.7 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 32-30。

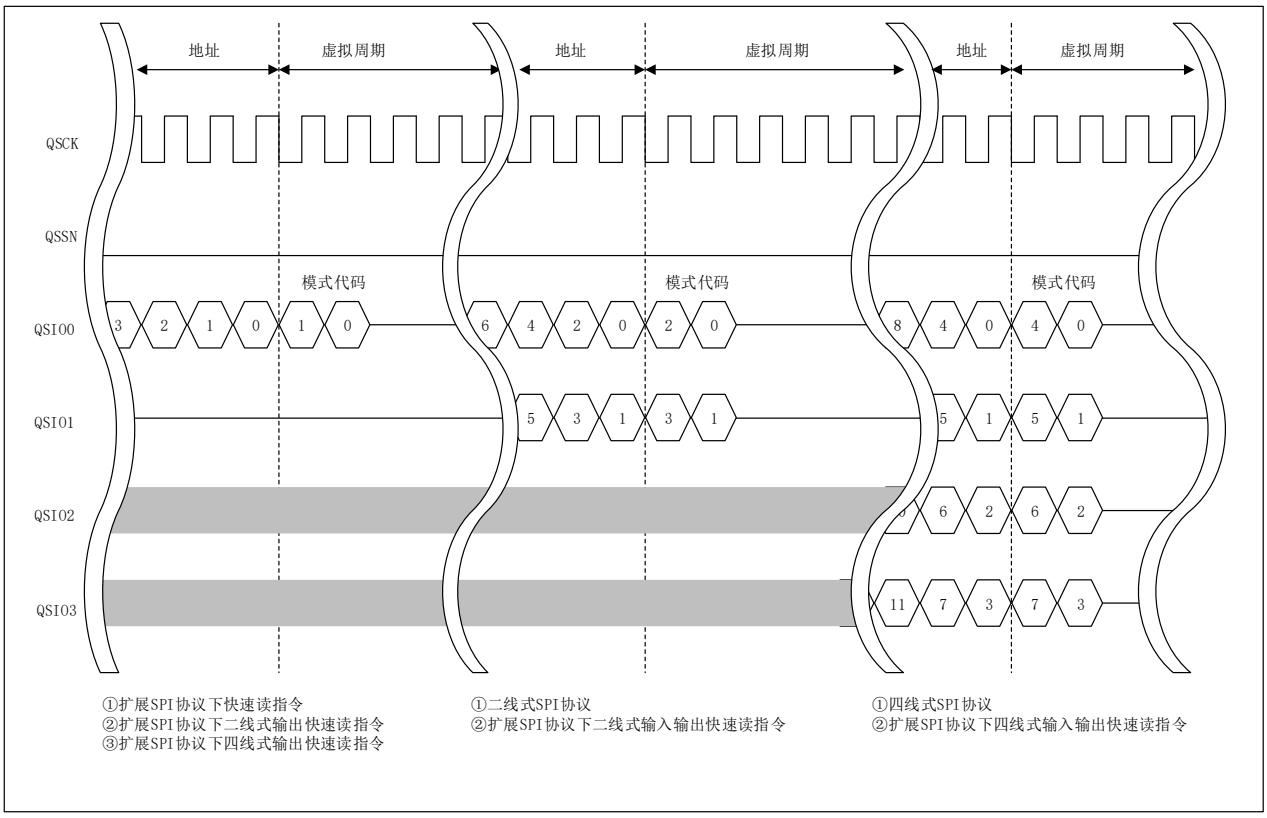


图 32-30 XIP 模式控制示意图

32.7.1 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。

32.7.2 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。

32.8 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	输出状态，输出电平由QSFCR寄存器的WPOL位决定，初始输出为低电平	输出高电平	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 32-5 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。

32.9 直接通信模式

32.9.1 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。

32.9.2 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。

32.9.3 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。

32.10 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）】。

32.11 使用上的注意事项

32.11.1 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。

32.11.2 模块停止信号的设置

QSPI 在系统复位后处于模块停止状态，只有在将模块停止控制寄存器中的 QSPI 模块停止信号清零号才可以对寄存器进行设置。具体请参考【动作模式与低功耗模式】。

32.12 寄存器说明

寄存器基准地址：0x9c000000h

寄存器名	偏移地址	复位值
控制寄存器QSCR	0x0000h	0x003f0000
片选控制寄存器QSCSCR	0x0004h	0x0000000f
格式控制寄存器QSFCR	0x0008h	0x000080b3
状态寄存器QSSR	0x000Ch	0x00008000
直接通信指令寄存器QSDCOM	0x0010h	0xXXXXXXXX
指令代码寄存器QSCCMD	0x0014h	0x00000000
XIP模式代码寄存器QSXCMD	0x0018h	0x000000ff
标志清除寄存器QSSR2（只写）	0x0024h	-
外部地址寄存器QSEXAR	0x0804h	0x00000000

表 32-6 QSPI 寄存器一览表

32.12.1 QSPI 控制寄存器（QSCR）

b31	b30	b29	b28	b27	b26	b25	b24
—	—	—	—	—	—	—	—
b23	b22	b21	b20	b19	b18	b17	b16
—	—	DIV[5:0]					
b15	b14	b13	b12	b11	b10	b9	b8
—	—	DPRSL[1:0]		APRSL[1:0]		IPRSL[1:0]	
b7	b6	b5	b4	b3	b2	b1	b0
SPIMD3	XIPE	DCOME	PFSAE	PFE	MDSEL[2:0]		

位	标记	位名	功能	读写
b31~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	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	R/W

			0 1 1 0 1 1：28个HCLK周期 0 1 1 1 0 0：29个HCLK周期 0 1 1 1 0 1：30个HCLK周期 0 1 1 1 1 0：31个HCLK周期 0 1 1 1 1 1：32个HCLK周期 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	R/W

			1 0：四线式SPI协议 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

32.12.2 QSPI 片选控制寄存器（QSCSCR）

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

位	标记	位名	功能	读写
b31~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5~b4	SSNW[1:0]	QSSN有效时间延长设置	QSPI总线访问后QSSN有效时间延长功能选择 b5 b4 0 0：不延长QSSN有效时间 0 1：将QSSN有效时间延长32个QSCK周期 1 0：将QSSN有效时间延长128个QSCK周期 1 1：将QSSN有效时间无限延长	R/W
b3~b0	SSHW[3:0]	QSSN最小无效时间设置	QSSN信号最小无效时间选择 b3 b2 b1 b0 0 0 0 0：1个QSCK周期 0 0 0 1：2个QSCK周期 0 0 1 0：3个QSCK周期 0 0 1 1：4个QSCK周期 0 1 0 0：5个QSCK周期 0 1 0 1：6个QSCK周期 0 1 1 0：7个QSCK周期 0 1 1 1：8个QSCK周期 1 0 0 0：9个QSCK周期 1 0 0 1：10个QSCK周期 1 0 1 0：11个QSCK周期 1 0 1 1：12个QSCK周期 1 1 0 0：13个QSCK周期 1 1 0 1：14个QSCK周期 1 1 1 0：15个QSCK周期 1 1 1 1：16个QSCK周期	R/W

32.12.3 QSPI 格式控制寄存器（QSFCR）

b31	b30	b29	b28	b27	b26	b25	b24
—	—	—	—	—	—	—	—
b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—	—	—
b15	b14	b13	b12	b11	b10	b9	b8
DUTY	—	—	—	DMCYCN[3:0]			
b7	b6	b5	b4	b3	b2	b1	b0
—	WPOL	SSNLD	SSNHD	—	4BIC	AWSL[1:0]	

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15	DUTY	占空比补正	QSCK输出波形占空比补正 0：不进行占空比补正 1：将QSCK的上升沿滞后0.5个HCLK周期 （当QSCK选择的频率是HCLK的奇数倍时有效）	R/W
b14~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11~b8	DMCYCN[3:0]	虚拟周期设置	使用快速读指令时虚拟周期数量选择 b3 b2 b1 b0 0 0 0 0：3个QSCK周期*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	R/W

			1：比QSCK第一个上升沿提前1.5个QSCK输出QSSN	
b4	SSNHD	QSSN信号释放时间延迟设置	QSSN信号释放时序选择 0：比QSCK最后一个上升沿滞后0.5个QSCK释放QSSN 1：比QSCK最后一个上升沿滞后1.5个QSCK释放QSSN	R/W
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	4BIC	4字节地址读指令代码选择	当地址宽度为4字节时的读指令代码选择 0：不使用4字节地址读指令代码 1：使用4字节地址读指令代码	R/W
b1~b0	AWSL[1:0]	地址宽度选择	串行接口地址宽度选择 b1 b0 0 0：1字节 0 1：2字节 1 0：3字节 1 1：4字节	R/W

*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。

32.12.4 QSPI 状态寄存器（QSSR）

b31	b30	b29	b28	b27	b26	b25	b24
—	—	—	—	—	—	—	—
b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—	—	—
b15	b14	b13	b12	b11	b10	b9	b8
PFAN	PFFUL	—	PFNUM[4:0]				
b7	b6	b5	b4	b3	b2	b1	b0
RAER	XIPF	—	—	—	—	—	BUSY

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15	PFAN	预读取动作状态	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访问错误标志	直接通信模式下发生ROM访问的错误标志位 0：未检测到发生ROM访问 1：检测到发生ROM访问	R/W
b6	XIPF	XIP模式标志	XIP模式状态信号 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

32.12.5 QSPI 指令代码寄存器（QSCCMD）

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

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7~b0	RIC[7:0]	替换指令代码	用于替代默认指令的串行闪存指令代码	R/W

32.12.6 QSPI 直接通信指令寄存器（QSDCOM）

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

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7~b0	DCOM[7:0]	直接通信模式指令	直接通信模式时的接口，通过QSPI总线进行直接通信。对该接口的读写访问会被转化成一个相应的QSPI总线周期。该接口只在直接通信模式下有效，ROM访问模式下该接口禁止访问。	R/W

32.12.7 QSPI XIP 模式代码寄存器（QSXCMD）

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

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7~b0	XIPMC[7:0]	XIP模式代码	用于串行闪存的模式代码。（设置XIP模式）	R/W

32.12.8 QSPI 系统配置寄存器（QSSR2）

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

位	标记	位名	功能	读写
b31~b8	Reserved	-	写入时写“0”	W
b7	RAERCLR	RAER清除	写入1时将QSSR中的RAER位清零	W
b6~b0	Reserved	-	写入时写“0”	W

32.12.9 QSPI 外部扩展地址寄存器（QSEXAR）

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

位	标记	位名	功能	读写
b31~b26	EXADR[5:0]	外部扩展地址代码	QSPI外部地址高6位设置，配合QSPI的ROM访问窗口地址最大可访问64MB×63块的外部ROM空间	R/W
b25~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

33 集成电路内置音频总线模块（I2S）

33.1 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	• 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	• 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	• 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*Fs(Fs为音频采样频率)
支持I2S协议	• I2S Philips标准 • MSB对齐标准 • LSB对齐标准 • PCM标准
数据缓冲	• 带有4字深，32位宽的输入输出FIFO缓冲区域
时钟源	• 可使用内部 I2SCLK(PLLAR/PLLAQ/PLLAP/PLLHR/PLLHQ/PLLHP)；也可由I2S_EXCK引脚上的外部时钟提供
中断	• 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 33-1 I2S 主要特性

33.2 I2S 系统框图

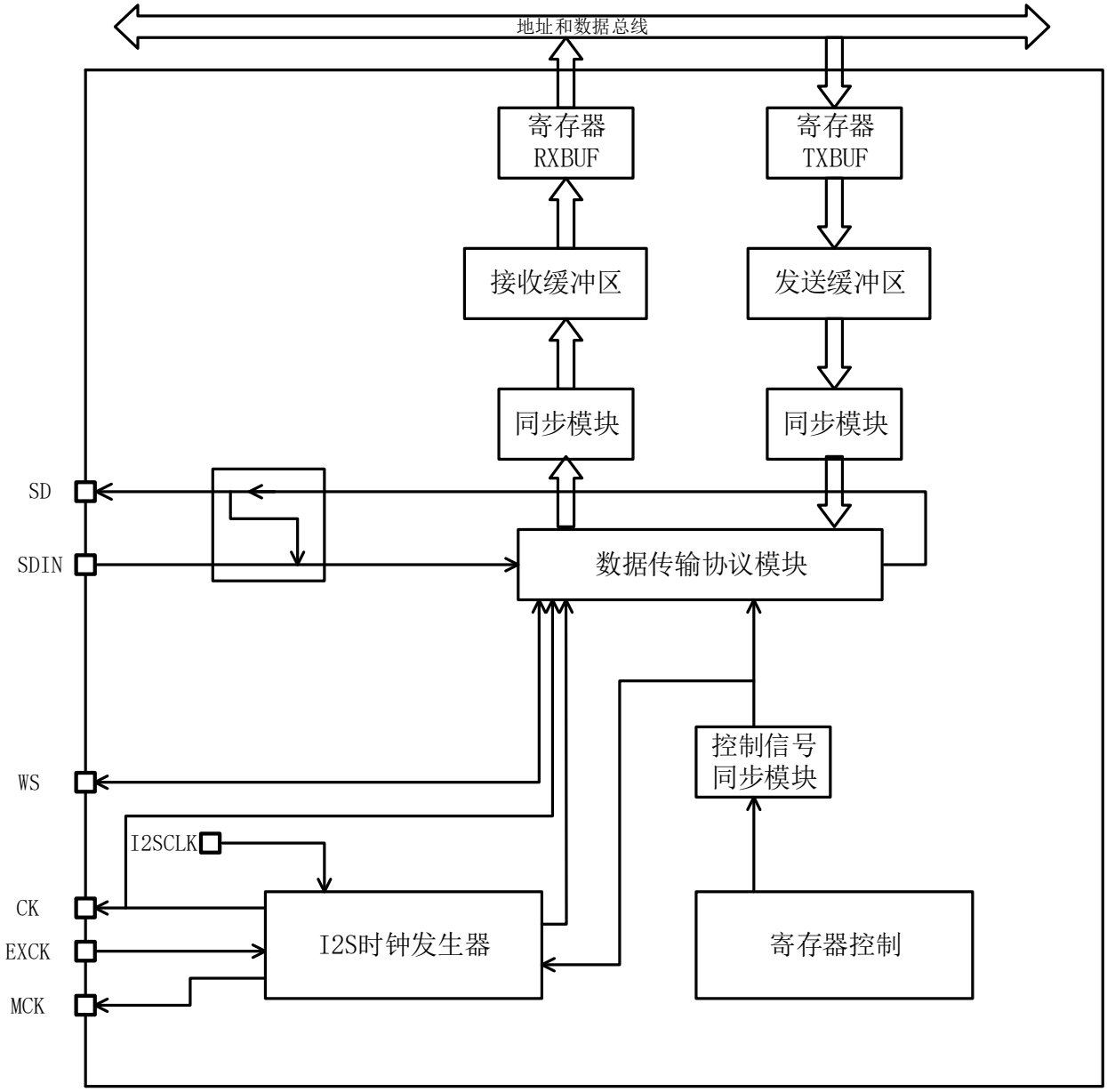


图 33-1 I2S 系统框图

33.3 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 33-2 I2S 管脚说明

33.4 功能说明

本章将对 I2S 的功能进行详细说明。

33.4.1 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×Fs，其中 Fs 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。

33.4.2 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。

33.4.3 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。

33.4.3.1 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。

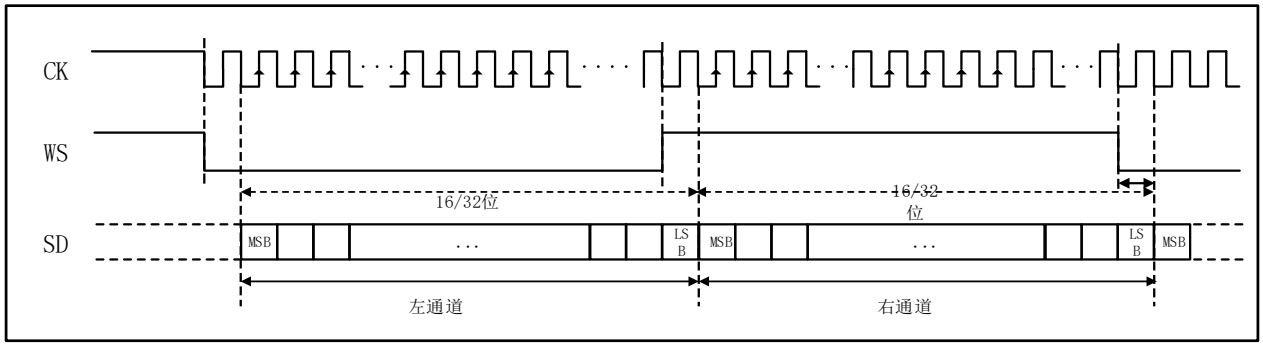


图 33-2 I2S Philips 协议波形（16/32 位全精度）

16 位装载在 16 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXXXX3344，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 0x0000eedd。

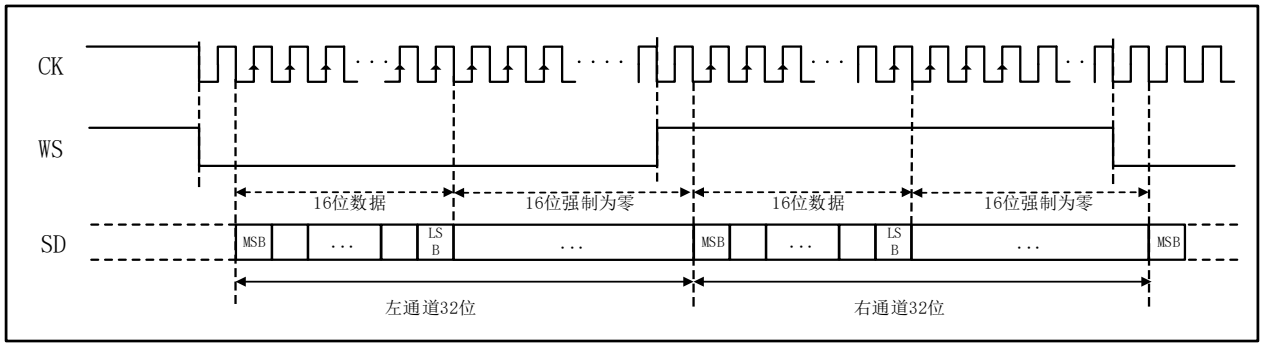


图 33-3 I2S Philips 协议波形（16 位数据封装在 32 位帧中）

16 位装载在 32 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXXXX3344，SD 输出串行数据 0x33440000；接收模式下，SD 输入串行数据 0xeeddXXXX，I2S_RXBUF 寄存器读出数据 0x0000eedd。

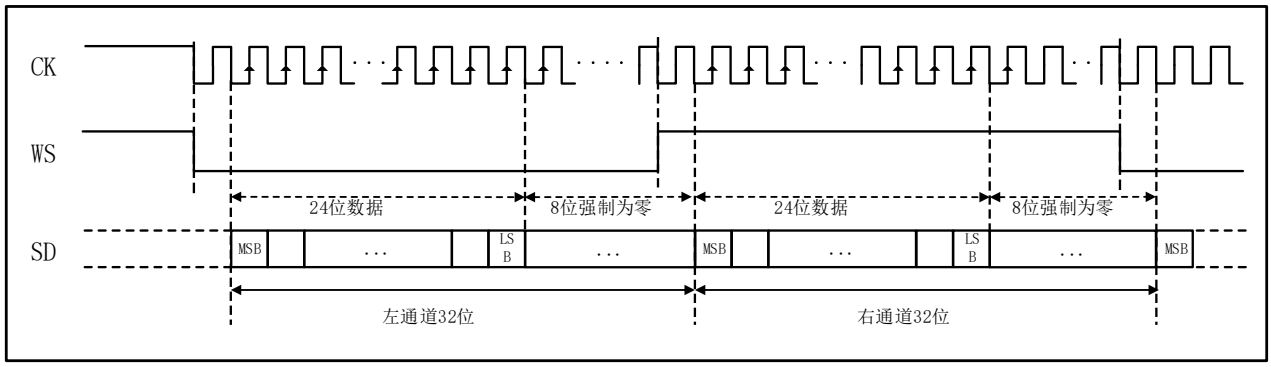


图 33-4 I2S Philips 协议波形（24 位数据封装在 32 位帧中）

24 位装载在 32 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXX223344，SD 输出串行数据 0x22334400；接收模式下，SD 输入串行数据 0xeedd11XX，I2S_RXBUF 寄存器读出数据 0x00eedd11。

33.4.3.2 MSB 对齐标准

此标准同时生成 WS 信号和第一个数据位（即 MSBit）。

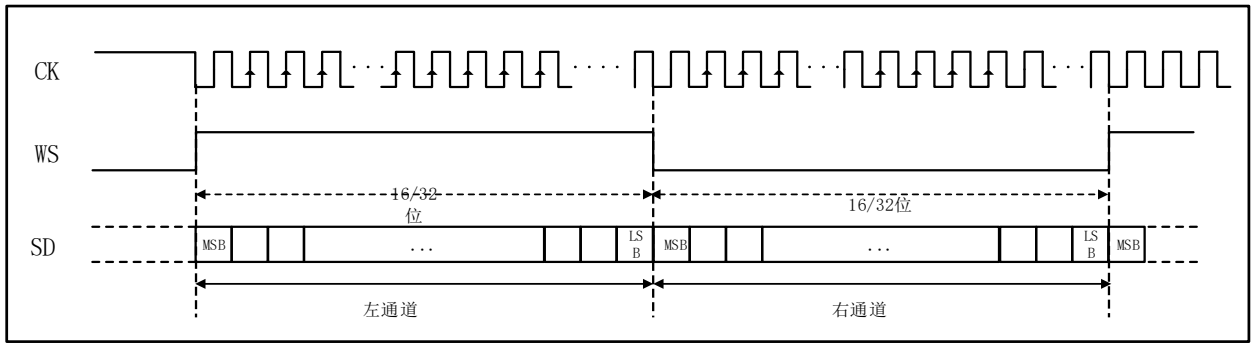


图 33-5 I2S MSB 协议波形（16/32 位全精度）

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 0xXXXX3344，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 0x0000eedd。

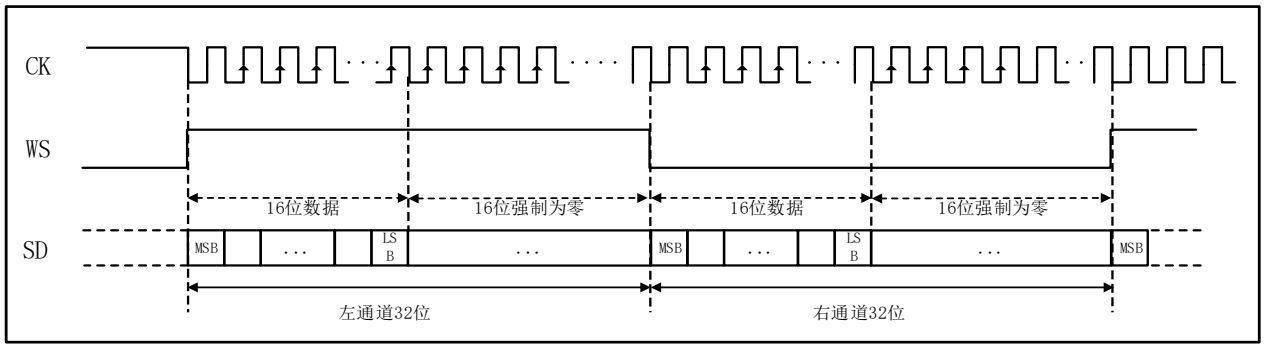


图 33-6 I2S MSB 协议波形（16 位数据封装在 32 位帧中）

16 位装载在 32 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXXXX3344，SD 输出串行数据 0x33440000；接收模式下，SD 输入串行数据 0xeeddXXXX，I2S_RXBUF 寄存器读出数据 0x0000eedd。

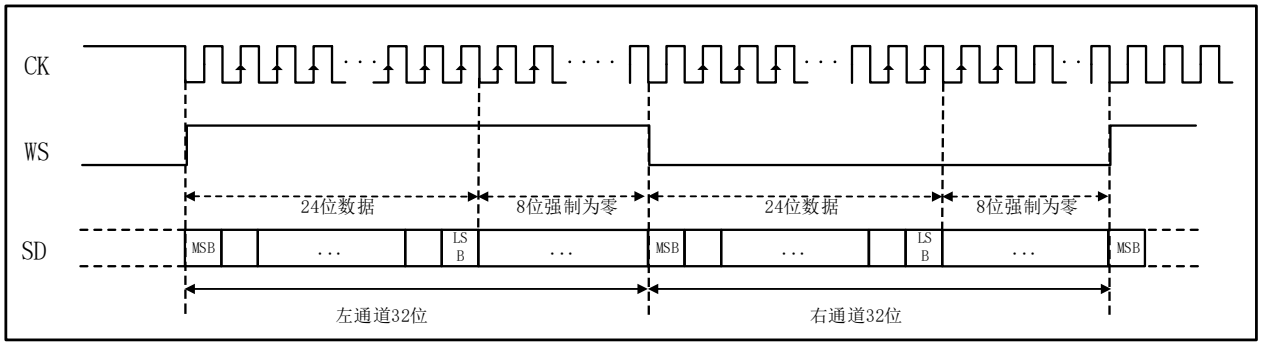


图 33-7 I2S MSB 协议波形（24 位数据封装在 32 位帧中）

24 位装载在 32 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXX223344，SD 输出串行数据 0x22334400；接收模式下，SD 输入串行数据 0xeedd11XX，I2S_RXBUF 寄存器读出数据 0x00eedd11。

33.4.3.3 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）。

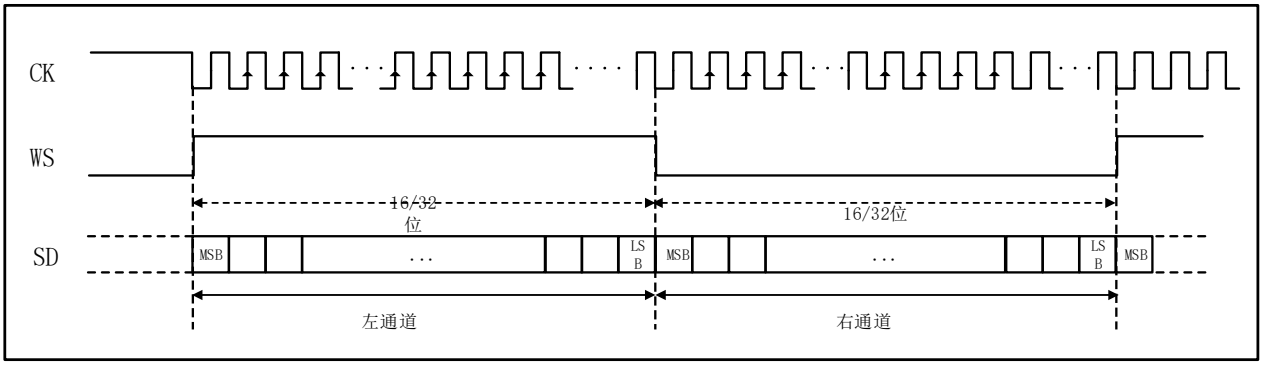


图 33-8 I2S LSB 协议波形（16/32 位全精度）

16 位装载在 16 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXXXX3344，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 0x0000eedd。

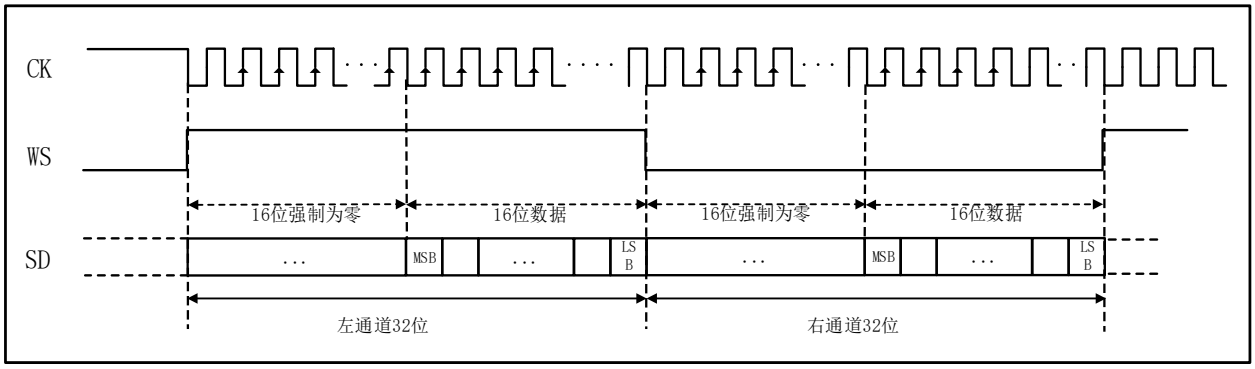


图 33-9 I2S LSB 协议波形（16 位数据封装在 32 位帧中）

16 位装载在 32 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXXXX3344，SD 输出串行数据 0x00003344；接收模式下，SD 输入串行数据 0xXXXX1122，I2S_RXBUF 寄存器读出数据 0x00001122。

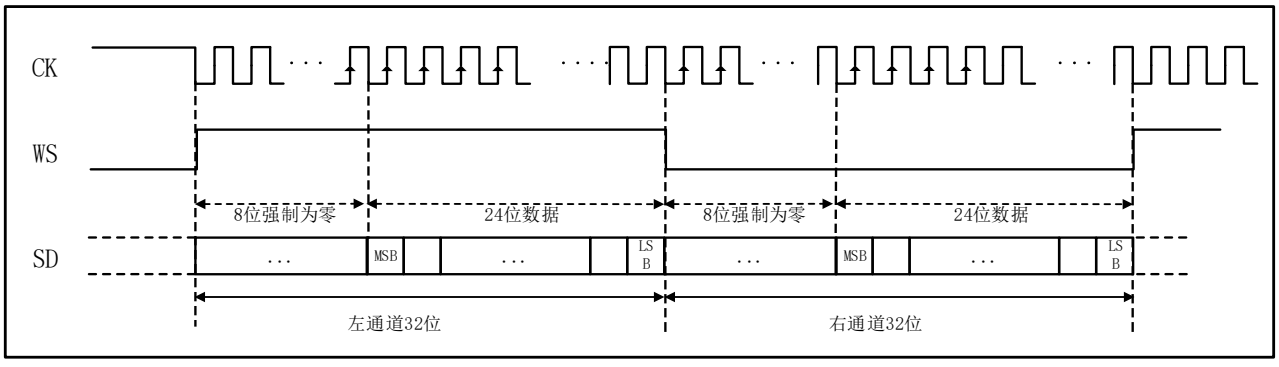


图 33-10 I2S LSB 协议波形（24 位数据封装在 32 位帧中）

24 位装载在 32 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXX223344，SD 输出串行数据 0x00223344；接收模式下，SD 输入串行数据 0xXXdd1122，I2S_RXBUF

寄存器读出数据 0x00dd1122。

33.4.3.4 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。

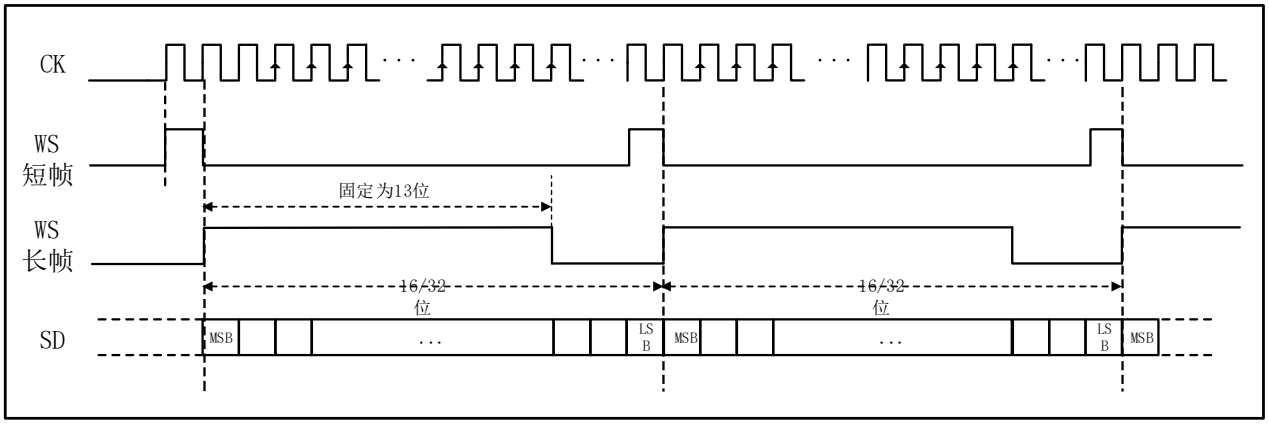


图 33-11 I2S PCM 协议波形（16/32 位全精度）

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 0xXXXX3344，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 0x0000eedd。

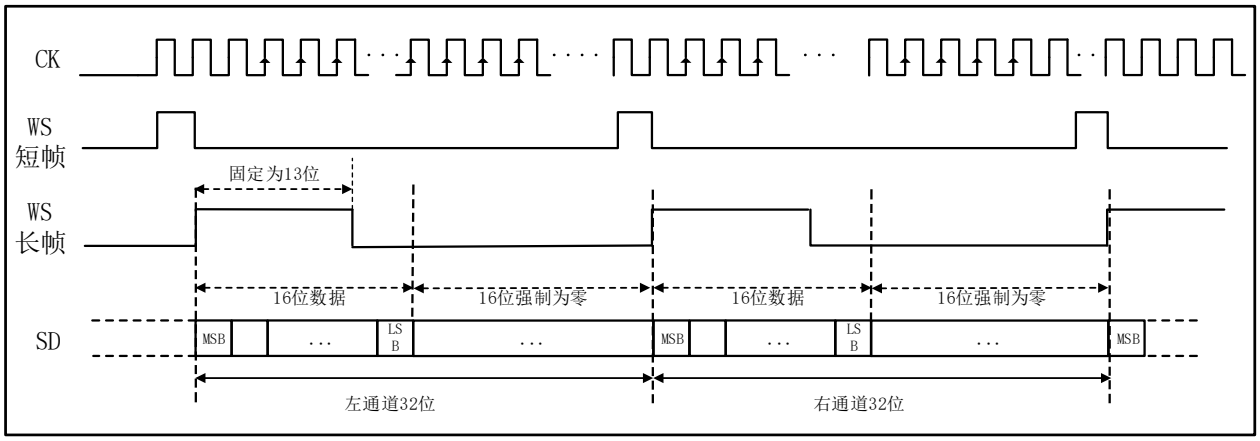


图 33-12 I2S PCM 协议波形（16 位数据封装在 32 位帧中）

16 位装载在 32 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXXXX3344，SD 输出串行数据 0x00003344；接收模式下，SD 输入串行数据 0xeeddXXXX，I2S_RXBUF

寄存器读出数据 0x0000eedd。

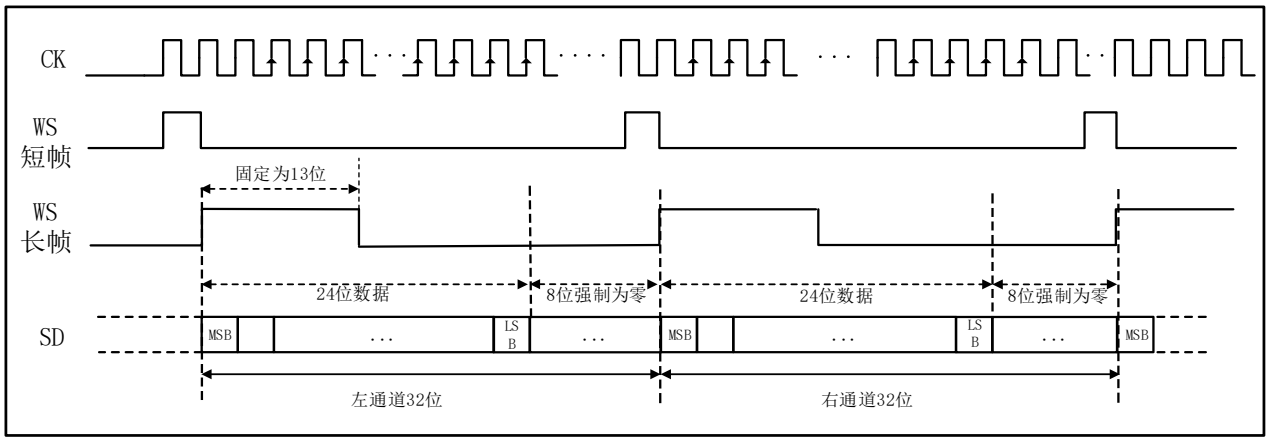


图 33-13 I2S PCM 协议波形（24 位数据封装在 32 位帧中）

24 位装载在 32 位帧，发送模式下，写入到 I2S_TXBUF 寄存器 0xXX223344，SD 输出串行数据 0x00223344；接收模式下，SD 输入串行数据 0xeedd11XX，I2S_RXBUF 寄存器读出数据 0x00eedd11。

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）。

33.4.4 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 =16×2×Fs。如果数据包为 32 位宽，则 I2S 比特率 =32x2xFs。

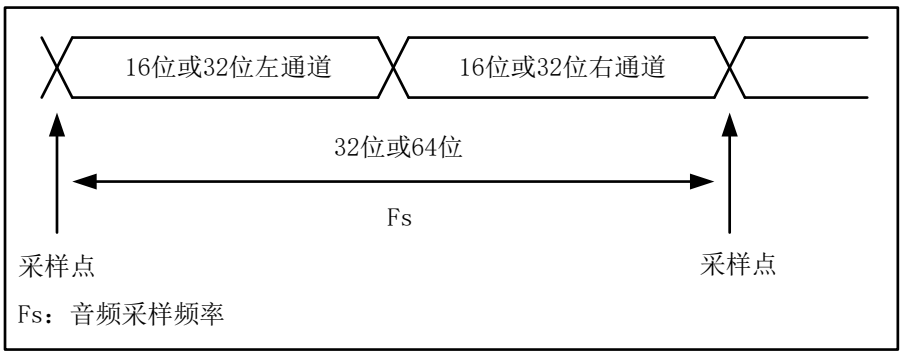


图 33-14 音频采样频率定义

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。

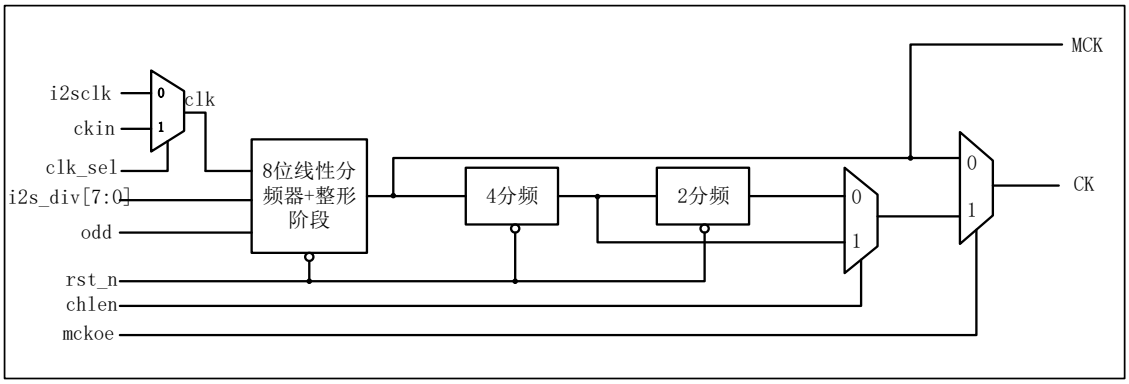


图 33-15 时钟发生器架构

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：

Fs=clk/[(16*2)*((2*I2SDIV+ODD)*8)] (通道帧宽度为 16 位时)

Fs=clk/[(32*2)*((2*I2SDIV+ODD)*4)] (通道帧宽度为 32 位时)

关闭驱动时钟（mckoe 清零）时：

Fs=clk/[(16*2)*(2*I2SDIV+ODD)] (通道帧宽度为 16 位时)

Fs=clk/[(32*2)*(2*I2SDIV+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 33-3 音频频率精度(针对 VCO 输入频率=1MHz)

注意：

- 倍频系数可为 PLLAN/PLLHN，倍频系数为 PLLAN 时输出分频比可为 PLLAP/PLLAQ/PLLAR，倍频系数为 PLLHN 时输出分频比可为 PLLHP/PLLHQ/PLLHR。具体设置参考【时钟控制器（CMU）】的 CMU PLLA 配置寄存器和 CMU PLLH 配置寄存器。

33.4.5 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 33.4.3 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 33.4.3 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。

33.4.6 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 33.4.3 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 33.4.3 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。

33.4.7 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 33-4 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 33-4 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 33-4 I2S 中断请求

33.4.8 使用上的注意事项

33.4.8.1 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。

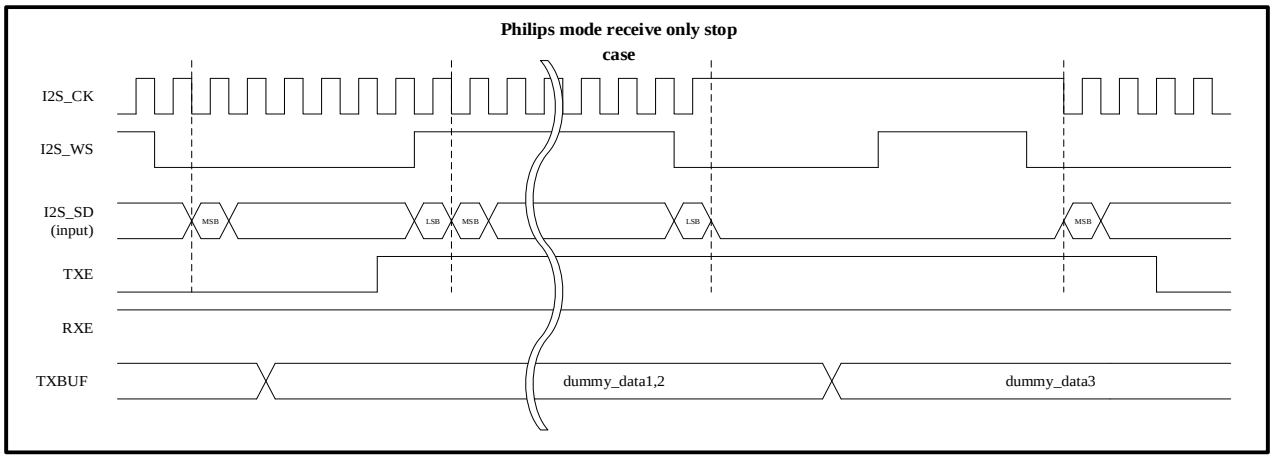


图 33-16 主机只接收暂时停止接收

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。

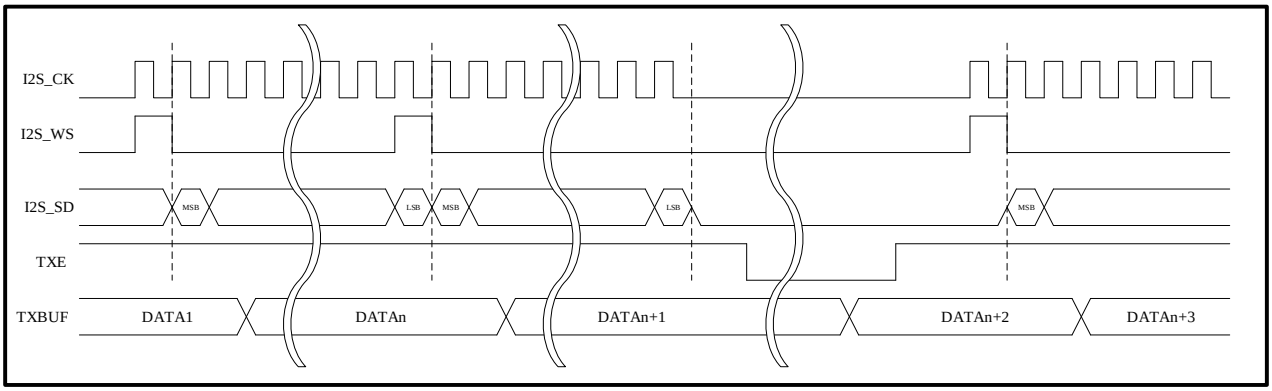


图 33-17 PCM 短帧主机发送暂停后重新发送方式一

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。

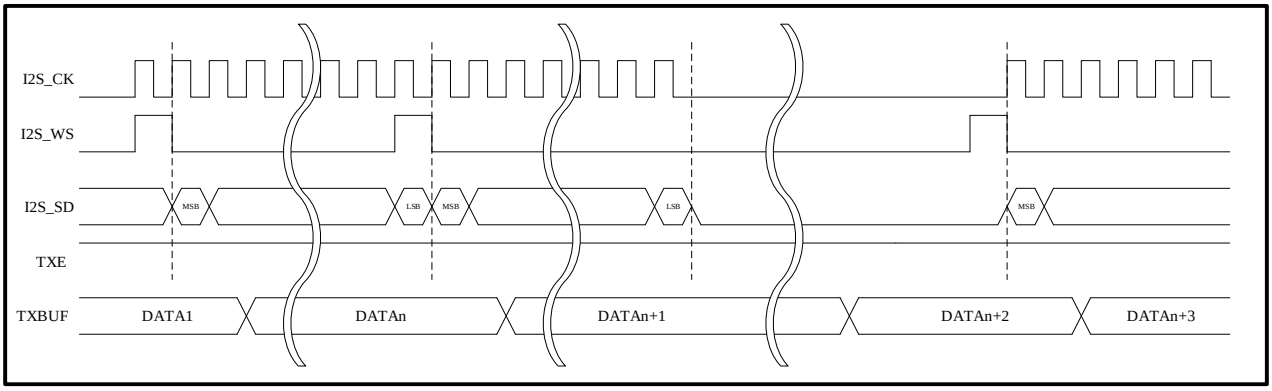


图 33-18 PCM 短帧主机发送暂停后重新发送方式二

33.4.8.2 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。

33.5 寄存器说明

I2S1 基准地址: 0x4001E000

I2S2 基准地址: 0x4001E400

I2S3 基准地址: 0x40022000

I2S4 基准地址: 0x40022400

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 33-5 I2S 寄存器一览表

注：只支持 32 位写寄存器

CMU_BASE_ADDR3 : 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	0x12	16	0xbbbb

注：I2S 主模式时钟源选择 I2SPLL 时使用该寄存器配置时钟源，可配置为 PLLAR/PLLAQ/PLLAP/PLLHR/PLLHQ/PLLHP。

33.5.1 I2S 控制寄存器（I2S_CTRL）

I2S Control Register

偏移地址：0x000

复位值：0x00002200

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	SRST	CLKSEL	DUPLEX	CKOE	LRCKOE	SDOE	I2SPLLSEL	CODECRC	FIFOR

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	RXBIRQWL			-	TXBIRQWL			MCKOE	ODD	WMS	EIE	RXIE	RXE	TXIE	TXE

位	标记	位名	功能	读写
b31~b25	Reserved	-	读出时为“0”,写入时写“0”	R/W
b24	SRST	软件复位	0：解除复位 1：软件复位	R/W
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：解除复位 1：软件复位	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/W
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0/1/2/3/4，因为fifo空间为4	R/W
b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
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0/1/2/3/4，因为fifo空间为4	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

33.5.2 I2S 状态寄存器（I2S_SR）

I2S Status Register

偏移地址：0x004

复位值：0x00000014

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/W
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

33.5.3 I2S 错误状态寄存器（I2S_ER）

I2S Error Status Register

偏移地址：0x008

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	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/W
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 。

33.5.4 I2S 配置寄存器（I2S_CFGR）

I2S Configuration Register

偏移地址：0x00C

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	PCMSYNC	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/W
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

33.5.5 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：0x00000000

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。

33.5.6 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：0x00000000

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。

33.5.7 I2S 分频寄存器（I2S_PR）

I2S Prescaler Register

偏移地址：0x018

复位值：0x00000002

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/W
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

33.5.8 CMU I2S 时钟配置存器(CMU_I2SCKSEL)

CMU_BASE_ADDR3: 0x40054000

偏移地址：0x12

复位值：0xbbbb

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
I2S4CKSEL				I2S3CKSEL				I2S2CKSEL				I2S1CKSEL			

位	标记	位名	功能	读写
b15~b12	I2S4CKSEL	I2S时钟源选择	0000：CMU_SCFGR设定的PCLK1 1000：PLLHQ 1001：PLLHR 1010：PLLAP 1011：PLLAQ 1100：PLLAR 除此以外禁止设定。 注：上记以外设定无时钟	R/W
b11~b8	I2S3CKSEL	I2S时钟源选择	0000：CMU_SCFGR设定的PCLK1 1000：PLLHQ 1001：PLLHR 1010：PLLAP 1011：PLLAQ 1100：PLLAR 除此以外禁止设定。 注：上记以外设定无时钟	R/W
b7~b4	I2S2CKSEL	I2S时钟源选择	0000：CMU_SCFGR设定的PCLK1 1000：PLLHQ 1001：PLLHR 1010：PLLAP 1011：PLLAQ 1100：PLLAR 除此以外禁止设定。 注：上记以外设定无时钟	R/W
b3~b0	I2S1CKSEL	I2S时钟源选择	0000：CMU_SCFGR设定的PCLK1 1000：PLLHQ 1001：PLLHR 1010：PLLAP 1011：PLLAQ 1100：PLLAR 除此以外禁止设定。	R/W

注：上记以外设定无时钟

注：

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

34 USB2.0 高速模块（USBHS）

34.1 USBHS 简介

USB 高速（USBHS）控制器为便携式设备提供了一套 USB 通信解决方案。USBHS 控制器支持主机模式和设备模式，芯片内部集成全速 PHY 且支持 ULPI 接口外接芯片外高速 PHY。主机模式下，USBHS 控制器支持高速（HS, 480Mb/s）、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高速（HS, 480Mb/s）和全速（FS，12Mb/s）收发器。USBHS 控制器支持 USB 2.0 协议所定义的所有四种传输方式（控制传输、批量传输、中断传输和同步传输）。该 USBHS 控制器支持 LPM(Link Power Management)功能。

芯片内部集成全速 PHY 仅支持全速（FS，12Mb/s）和低速（LS，1.5Mb/s）传输，通过 ULPI 接口外接片外高速 PHY 则支持高速（HS, 480Mb/s）、全速（FS，12Mb/s）和低速（LS，1.5Mb/s）传输。

遵循协议如下：

- Universal Serial Bus Revision 2.0 Specification
- USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007
- Errata for USB 2.0 ECN: Link Power Management (LPM) - 7/2007

34.2 USBHS 主要特性

主要分为三类：通用特性、主机模式特性和设备模式特性。

34.2.1 通用特性

- 支持两个 PHY 接口
  - 内置片上 USB2.0 全速 PHY(需要设置 USB_SYCTLREG. USBHS_FSPHYE=1 使能)
  - 连接片外高速 PHY 的 ULPI 接口
- 支持主机模式和设备模式
- 支持 HS/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 8KB 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

34.2.2 主机模式特性

- 主机模式支持 USB2.0 高速（HS, 480Mb/s）、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 16 主机通道（管道）：每个通道都可以动态实现重新配置，可支持任何类型的 USB 传输
- 内置硬件调度器，可：
  - 在周期性硬件队列中存储多达 8 个中断加同步传输请求
  - 在非周期性硬件队列中存储多达 8 个控制加批量传输请求
- 管理一个共享 RX FIFO、一个周期性 TX FIFO 和一个非周期性 TX FIFO，以有效使用 USB 数据 RAM

34.2.3 设备模式特性

- 从机模式支持 USB2.0 高速（HS, 480Mb/s）、全速（FS，12Mb/s）传输。
- 1 个双向控制端点 0
- 15 个 OUT 端点，可以配置为支持批量传输、中断传输或同步传输
- 15 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 16 个发送 FIFOs（每个 IN 端点配备一个发送 FIFO）和一个接收 FIFO（由所有的 OUT 端点共享）
- 支持远程唤醒功能。
- 支持软断开功能
- VBUS PIN 支持 5V 耐压。

34.3 USBHS 系统框图

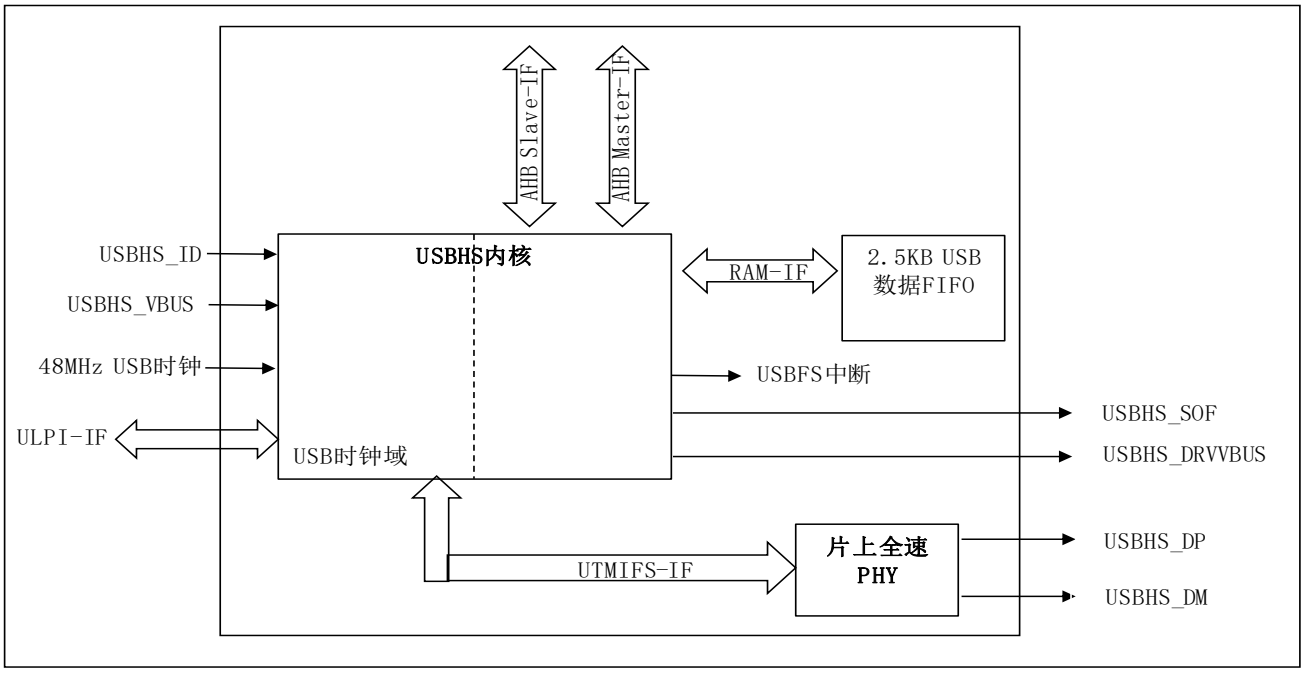


图 34-1 USBHS 系统框图

34.4 USBHS 管脚说明

管脚名	方向	适用模式	对应管脚	功能描述	适用PHY
USBHS_VBUS	输入	设备	PA9	电源端口，5V耐压	片上全速PHY
USBHS_DP	输入/输出	主机、设备	PA12	差分数据D+	片上全速PHY
USBHS_DM	输入/输出	主机、设备	PA11	差分数据D-	片上全速PHY
USBHS_DRVVBUS	输出	主机	PC9、PB8	外部电源芯片使能	片上全速PHY
USBHS_ID	输入	主机	PA10	USB A-B器件识别	片上全速PHY
USBHS_SOF	输出	主机、设备	PA8	SOF输出脉冲	片上全速PHY ULPI PHY
USBHS_ULPI_CLK	输入	主机、设备	PE12、PA5	ULPI Clock	ULPI PHY
USBHS_ULPI_DIR	输入	主机、设备	PC2、PI11	ULPI Direction	ULPI PHY
USBHS_ULPI_NXT	输入	主机、设备	PC3、PH4	ULPI Next	ULPI PHY
USBHS_ULPI_STP	输出	主机、设备	PC0	ULPI Stop	ULPI PHY
USBHS_ULPI_D0	输入/输出	主机、设备	PE13、PA3	ULPI Data 0	ULPI PHY
USBHS_ULPI_D1	输入/输出	主机、设备	PE14、PB0	ULPI Data 1	ULPI PHY
USBHS_ULPI_D2	输入/输出	主机、设备	PE15、PB1	ULPI Data 2	ULPI PHY
USBHS_ULPI_D3	输入/输出	主机、设备	PB10	ULPI Data 3	ULPI PHY
USBHS_ULPI_D4	输入/输出	主机、设备	PB11、PD9	ULPI Data 4	ULPI PHY
USBHS_ULPI_D5	输入/输出	主机、设备	PB12	ULPI Data 5	ULPI PHY
USBHS_ULPI_D6	输入/输出	主机、设备	PB13	ULPI Data 6	ULPI PHY
USBHS_ULPI_D7	输入/输出	主机、设备	PB5、PE11	ULPI Data 7	ULPI PHY

表 34-1 USBHS 管脚说明

由于 USBHS_DP 和 USBHS_DM 管脚和通用 GPIO 复用，在使用 USBHS 片上全速 PHY 时，建议关闭其对应管脚的数字功能，USBHS_DP 和 USBHS_DM 功能为模拟功能，与对应的 PFSR 寄存器设定无关。具体请参考 011.通用 IO（GPIO）章节。另外 USBHS 功能不使用时，USBHS_DP 和 USBHS_DM 管脚对应的数字功能管脚翻转时，会产生额外的电流消耗。

34.5 USBHS 功能说明

34.5.1 USBHS 时钟以及工作模式

USBHS 所使用的时钟有以下三种

- 48MHz 时钟，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 USBHS 模块前，需要在 CMU 模块内配置好 USBHS 时钟。
- 60MHz 时钟，使用 UPLI 接口外接高速 PHY 的输入时钟，仅在外接 PHY 时使用。
- 模块时钟 PCLK1，需要在 CMU 模块内配置好该时钟，该时钟需要大于等于 60MHz。

USBHS 可以作为主机或者设备使用，并且包含一个片上全速 PHY 和一个 ULPI 接口用于外接高速 PHY。

上拉和下拉电阻已经集成在片上全速 PHY 的内部，并且 USBHS 可以根据当前模式和连接状态自动选择。

USBHS 使用片上全速 PHY 时，VCC 电压范围为 3.0~3.6V。

34.5.2 USBHS 模式决定

USBHS 决定当前工作模式有以下两种方法：

方法 1：根据 USBHS_ID 线的状态自动识别，检测到 USBHS_ID 线为高电平时，模块工作在设备模式，检测到 USBHS_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 USBHS_GUSBCFG 的 FDMOD 或者 FHMOD 位为 1，使模块忽略 USBHS_ID 线的电平而强制工作在设备或者主机模式。

34.5.3 USBHS 主机功能

34.5.3.1 主机功能简介

当 USBHS 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。USBHS_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 USBHS_VBUS 可以作为 GPIO 使用。

典型的 USB 主机模式系统构建图如下：

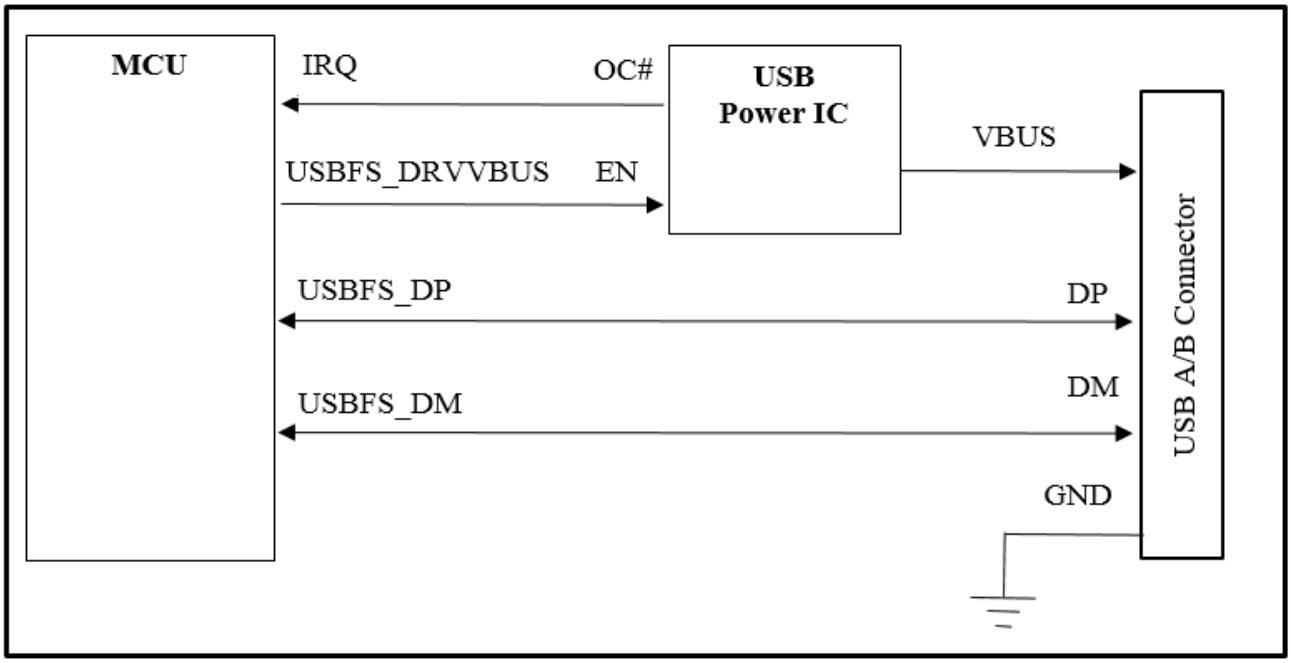


图 34-2 USBHS 主机模式系统构建图

34.5.3.2 主机端口供电

本 MCU 不能输出 5V 以提供 VBUS。为此，必须在微控制器以外添加 USB 电源芯片或基本电源开关（如应用电路板提供 5V 电源）来驱动 5V VBUS 线。外部电源芯片可通过任何 GPIO 输出或者 USBHS_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（USBHS_HPRT 中的 PPWR 位）置 1。

34.5.3.3 主机检测设备连接和断开

USB 设备将在连接后立即被检测到。USBHS 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（USBHS_HPRT 中的 PCDET 位）触发。设备断开事件将触发断开连接检测中断（USBHS_GINTSTS 中的 DISCINT 位）。

34.5.3.4 主机枚举

检测到设备连接后，若又有新的设备连接进来，主机必须通过向新的设备发送 USB 复位和配置命令来启动枚举过程。

应用程序通过将主机端口控制和状态寄存器中的端口复位位（USBHS_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（USBHS_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（USBHS_HPRT 中的 PSPD）读取枚举的设备速度，以及主机已经开始驱动 SOF (HS/FS)或 Keep-alive 令牌 (LS)。此时主机已就绪，可通过对设备发送命令来完成对设备的枚举。

34.5.3.5 主机挂起

应用程序通过将主机端口控制和状态寄存器中的端口挂起位（USBHS_HPRT 中的 PSUSP）置 1 来挂起 USB 总线。USBHS 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（USBHS_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（USBHS_HPRT 中的 PRES 位）自行复位，并通过 USB 自动驱动恢复信号。应用程序必须为恢复窗口定时，然后将端口恢复位清零以退出挂起状态并重新启动 SOF。

如果由主机发起退出挂起状态，则应用程序必须将端口恢复位置 1 以启动主机端口上的恢复信号，为恢复窗口定时并最终将端口恢复位清零。

34.5.3.6 主机通道

USBHS 模块实现了 16 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：HS/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 USBHS 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（USBHS_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。 USBHS_HCINTMSK x 寄存器还提供每个通道各中断源的屏蔽位。

主机模块提供以下状态检查和中断产生功能：

- 传输完成中断，指示应用程序 (AHB) 和 USB 端均已完成数据传输

- 通道因传输完成、USB 事务错误或应用程序发出禁止命令而停止
- 相关的发送 FIFO 为半空或全空状态（IN 端点）
- 接收到 ACK 响应
- 接收到 NAK 响应
- 接收到 STALL 响应
- 由于 CRC 校验失败、超时、位填充错误和错误的 EOP 导致 USB 事务错误
- 串扰错误
- 帧上溢
- 用于数据同步的翻转位出错

34.5.3.7 主机调度器

主机模块内置硬件调度器，可自主对应用程序发出的 USB 事务请求重新排序和管理。每一帧开始时，主机都先执行周期性（同步和中断）事务，然后执行非周期性（控制和批量）事务，以符合 USB 规范对同步和中断传输高优先级的保证。

主机通过请求队列（一个周期性请求队列和一个非周期请求队列）处理 USB 事务。每个请求队列最多可存储 8 个条目。每个条目代表一个应用程序发起但还未得到响应的 USB 事务请求，并存储了执行该 USB 事务所用到的 IN 或 OUT 通道的编号，以及其它相关信息。USB 事务请求在队列中的写入顺序决定了事务在 USB 接口上的执行顺序。

每一帧开始时，主机都先处理周期性请求队列，然后处理非周期性请求队列。如果当前帧结束时，计划在当前帧执行的同步或中断类型的 USB 传输事务请求仍处于挂起状态，则主机将发出未完成周期性传输中断（USBHS_GINTSTS 中的 IPXFR 位）。USBHS 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 USBHS_HPTXSTS 寄存器中的 PTXQSAV 位或 USBHS_HNPTXSTS 寄存器中的 NPTQXSAV 位，确保周期性（非周期性）请求队列中至少有一个可用空间来存储当前请求。

34.5.4 USBHS 设备功能

34.5.4.1 设备功能简介

当 USBHS 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：

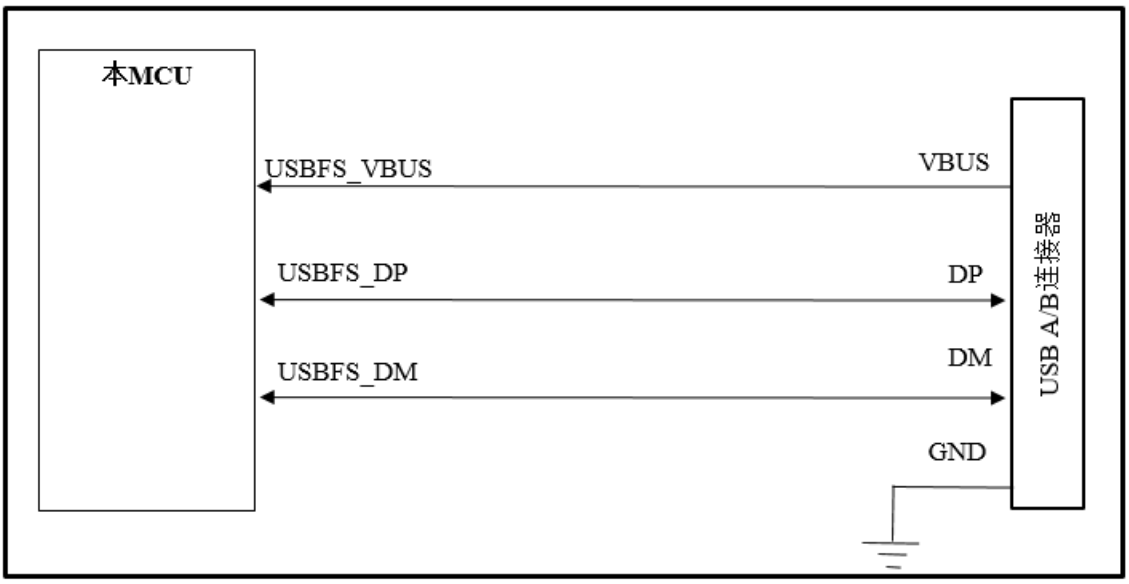


图 34-3 USBHS 设备模式系统构建图

34.5.4.2 设备供电状态

模块检测到 USBHS_VBUS 为高电平时，就会使 USB 设备进入供电状态。然后，USBHS 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（USBHS_GINTSTS 中的 VBUSVINT 位），指示进入供电状态。

此外，USBHS_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），USBHS 将自动断开连接。

供电状态下，USBHS 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（USBHS_GINTST 中的 USBRST）。复位信号结束后，将生成枚举完成中断（USBHS_GINTSTS 中的 ENUMDNE 位），USBHS 随即进入默认状态。

34.5.4.3 设备默认状态

默认状态下，USBHS 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（USBHS_DCFG 中的 DAD 位）。USBHS 随即进入地址状态，并准备好以所配置的 USB 地址对主机事务进行应答。

34.5.4.4 设备挂起状态

USBHS 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（USBHS_GINTSTS 中 ESUSP 位），并在 3ms 后由挂起中断（USBHS_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（USBHS_DSTS 中的 SUSPSTS 位）自动置 1，USBHS 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（USBHS_DCTL 中的 RWUSIG 位）置 1，并在 1ms 到 15ms 内将其清零。但若设备检测到主机发出的恢复信号，将生成恢复中断（USBHS_GINTSTS 中的 WKUPINT 位），设备挂起位自动清零。

34.5.4.5 设备软断开

供电状态可借助软断开功能通过软件退出。将设备控制寄存器中的软断开位（USBHS_DCTL 中的 SDIS 位）置 1 即可移除 DP 上拉电阻，此时尽管没有从主机端口实际拔出 USB 电缆，但主机端仍会发生设备断开检测中断。

34.5.4.6 设备端点

端点类别

USBHS 模块实现了以下 USB 端点：

- 控制端点 0：
  - 双向且仅处理控制消息
  - 使用一组单独的存器来处理 IN 和 OUT 事务
  - 专用控制 (USBHS_DIEPCTL0/USBHS_DOEPCTL0) 寄存器、传输配置 (USBHS_DIEPTSIZ0/USBHS_DIEPTSIZ0) 寄存器和状态中断 (USBHS_DIEPINTx/USBHS_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

- 15 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 (USBHS_DIEPCTLx) 寄存器、传输配置 (USBHS_DIEPTSIZx)寄存器和状态中断(USBHS_DIEPINTx) 寄存器
  - 设备 IN 端点通用中断屏蔽寄存器 (USBHS_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（USBHS_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 (USBHS_GINTSTS/EOPF) 一起触发
- 15 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 (USBHS_DOEPCTLx) 寄存器、传输配置 (USBHS_DOEPTSIZx) 寄存器和状态中断 (USBHS_DOEPINTx) 寄存器
  - 设备 OUT 端点通用中断屏蔽寄存器 (USBHS_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（USBHS_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 (USBHS_GINTSTS/EOPF) 一起触发

端点控制

应用程序可通过设备端点 x IN/OUT 控制寄存器 (DIEPCTLx/DOEPCTLx) 对端点采取以下控制：

- 端点使能/禁止
- 在当前配置下激活端点
- 设置 USB 传输类型（同步、批量和中断）
- 设置支持的数据包大小
- 设置与 IN 端点相关的 Tx-FIFO 编号
- 设置希望收到的或发送时要使用到的 data0/data1 PID（仅限批量/中断传输）
- 设置接收或发送事务时所对应的奇/偶帧（仅限同步传输）

- 可以设置 NAK 位，从而不论此时 FIFO 的状态如何，都对主机的请求回复 NAK
- 可以设置 STALL 位，使得主机对该端点的令牌都被硬件回复 STALL
- 可以将 OUT 端点设置为侦听模式，即对接收到的数据不进行 CRC 检查

端点传输

设备端点 x 传输尺寸寄存器 (DIEPTSIZx/DOEPTSIZx) 允许应用程序对传输尺寸参数进行编程并读取传输状态。必须在端点控制寄存器中的端点使能位置 1 之前完成对此寄存器的设置。使能端点后，这些字段立即变为只读状态，同时 USBHS 模块根据当前传输状态对这些字段进行更新。

可对以下传输参数进行编程：

- 以字节为单位的传输大小
- 构成整个传输的数据包个数

端点状态/状态

设备端点 x 中断寄存器 (DIEPINTx/DOPEPINTx) 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 USBHS_GINTSTS 中 OEPINT 位或 USBHS_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(USBHS_DAINT) 寄存器，以获取设备端点 x 中断寄存器的端点编号。应用程序必须将此寄存器中的相应位清零，才能将 DAINT 和 GINTSTS 寄存器中的相应位清零。

模块提供以下状态检查和中断产生功能：

- 传输完成中断，指示应用程序 AHB 和 USB 端均已完成数据传输
- Setup 阶段已完成（仅针对控制传输类型的 OUT 端点）
- 相关的发送 FIFO 为半空或全空状态（IN 端点）
- NAK 应答已发送到主机（仅针对同步传输的 IN 端点）
- TxFIFO 为空时接收到 IN 令牌（仅针对批量和中断传输类型的 IN 端点）
- 尚未使能端点时接收到 OUT 令牌
- 检测到 babble 错误
- 应用程序关闭端点生效
- 应用程序对端点设置 NAK 生效（仅针对同步传输类型的 IN 端点）

- 接收到 3 个以上连续 setup 数据包（仅针对控制类型的 OUT 端点）
- 检测到超时状况（仅针对控制传输类型的 IN 端点）

34.5.5 USBHS SOF 脉冲管脚输出功能

USBHS 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 USBHS_SOF 管脚输出，输出宽度为 16 个系统时钟周期。

34.5.5.1 主机 SOF

主机模式下，可以在主机帧间隔寄存器 (HFIR) 中对所产生的两个连续 SOF (HS/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（USBHS_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 USBHS 系统控制寄存器 USBHS_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 USBHS_SOF 引脚输出。

此外，SOF 脉冲还可以作为内部事件触发 DMA 传输、TIMER 计数等外部模块工作。

34.5.5.2 设备 SOF

在设备模式下，USB 每次接收到 SOF 令牌时，都将触发帧开始中断（USBHS_GINTSTS 中的 SOF 位）。相应的帧编号可从设备状态寄存器（USBHS_DSTS 中的 FNSOF 位）读取。使用 USBHS 系统控制寄存器 USBHS_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 USBHS_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（USBHS_DCFG 中的 PFIVL 位）。此功能可用于确定该帧的所有同步通信是否完成。

34.5.6 USBHS 功耗控制

不使用 USBHS 模块时，可以通过 CMU 模块停止 USBHS 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（USBHS_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，USBHS 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（USBHS_GCCTL 中的 GATEHCLK 位）
  将时钟门控控制寄存器中的 GATEHCLK 位置 1 时，USBHS 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 USBHS 模块访问时为其提供时钟。

34.5.7 USBHS 动态更新 USBHS_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 USBHS_HFIR 寄存器在当前 SOF 帧内发生更改，则将在下一个帧中对 SOF 周期进行相应修正，具体说明请参见图 34-4。

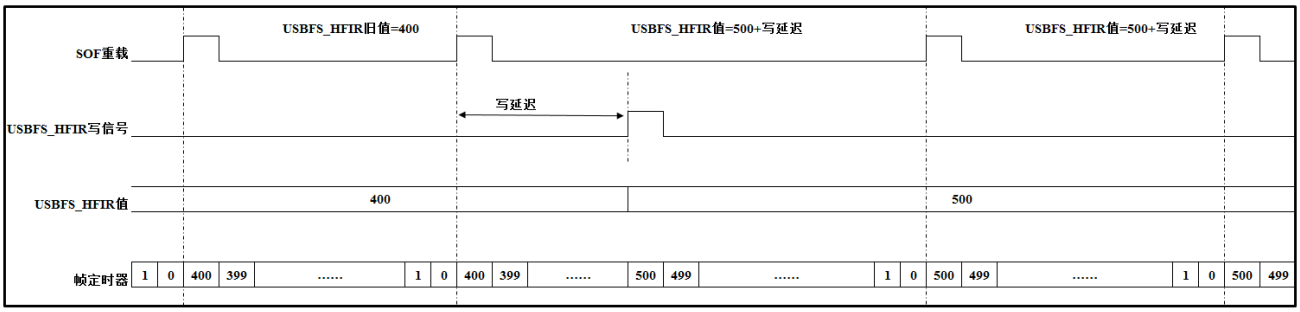


图 34-4 USBHS 动态更新 USBHS_HFIR 寄存器示意图

34.5.8 USBHS 数据 FIFO

USBHS 系统具有 8KB 专用 RAM，采用高效的 FIFO 控制机制。USBHS 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。

34.5.9 USBHS 主机 FIFO 架构

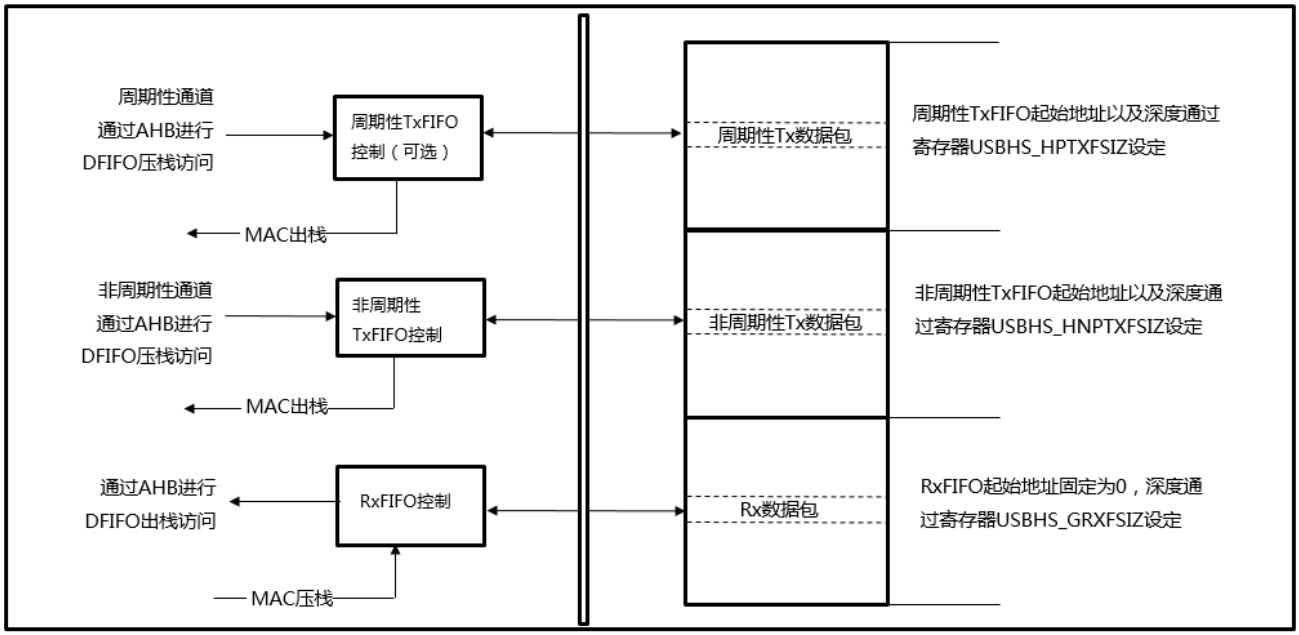


图 34-5 USBHS 主机模式下 FIFO 架构示意图

34.5.9.1 主机 RxFIFO

主机使用一个接收 FIFO 处理所有周期和非周期事务。FIFO 用作接收缓冲区以保存从 USB 接收到的数据（接收到的数据包的数据部分），直至这些数据传输到系统存储器。只要 FIFO 中有空间，来自设备 IN 端点的数据包就接收进来并挨个存储。接收到的每个数据包的状态（包含主机目标通道、字节数、数据 PID 和对所接收数据的校验）也存储在 FIFO 中。 接收 FIFO 的大小在接收 FIFO 大小寄存器 (GRXFSIZ) 中配置。单个接收 FIFO 架构使得 USB 主机高效地填充接收数据缓冲区：

- 所有 IN 配置主机通道共享同一个 RAM 缓冲区（共享 FIFO）
- 对于主机软件驱动的任意序列 IN 令牌，USBHS 模块可将接收 FIFO 填充至限值

只要至少有一个数据包在 RxFIFO 中可供读取，应用程序就会接收 Rx FIFO 非空中断。应用程序从接收状态读取和出栈寄存器中读取数据包信息，最后从 RxFIFO 中读出数据。

34.5.9.2 主机 TxFIFO

主机使用一个发送 FIFO 处理所有非周期（控制和批量）OUT 事务，使用另一个发送 FIFO 处理所有周期（同步和中断 OUT 事务。FIFO 用作发送缓冲区以保存要通过 USB 发送的数据（发送数据包）。周期（非周期）TxFIFO 的大小在主机周期（非周期）发送 FIFO 大小 (HPTXFSIZ/HNPTXFSIZ) 寄存器中配置。

两个 Tx FIFO 按优先级实施操作，周期性通信的优先级较高，因此在 USB 一帧的时间内首先进行周期性通信。帧起始时，内置的主机调度器先处理周期请求队列，再处理非周期请求队列。

两个发送 FIFO 的架构使得 USB 主机能够对周期和非周期发送数据缓冲区分别进行优化管理：

- 配置为支持周期（非周期）OUT 事务的所有主机通道共享同一个 RAM 缓冲区（共享 FIFO）
- 对于主机软件驱动的任意序列 OUT 令牌，USBHS 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，USBHS 模块就会发出周期性 TxFIFO 空中断（USBHS_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（USBHS_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，USBHS 模块就会发出非周期性 TxFIFO 空中断（USBHS_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（USBHS_GAHBCFG 中的 TXFELVL 位）。只要非周期性 TxFIFO 和非周期性请求队列中均存在空闲空间，应用程序便可写入发送数据。可通过读取主机非周期性发送 FIFO 和队列状态寄存器 (HNPTXSTS)来了解二者的可用空间。

34.5.10 USBHS 设备 FIFO 架构

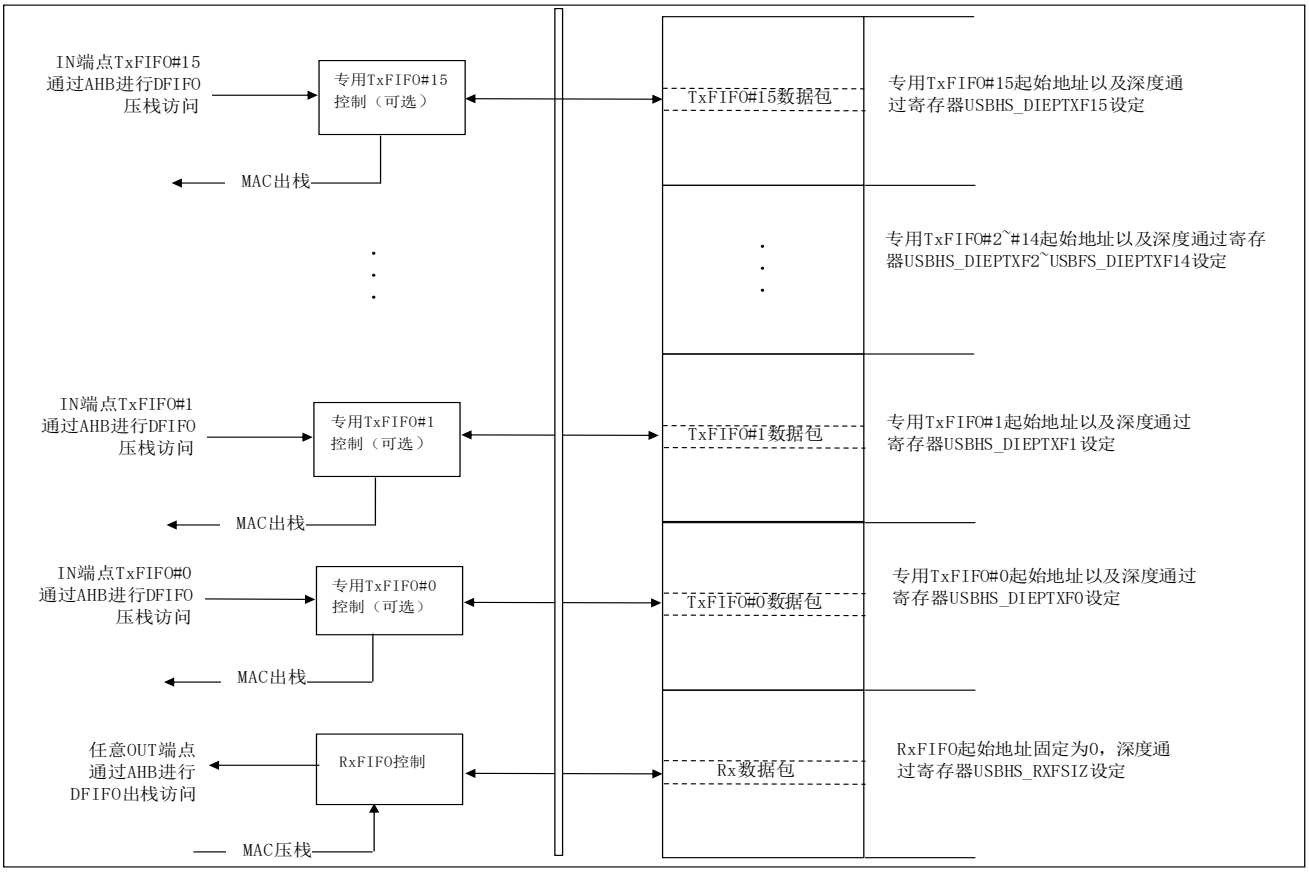


图 34-6 USBHS 设备模式下 FIFO 架构示意图

34.5.10.1 设备 RxFIFO

USBHS 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，USBHS 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（USBHS_GINTSTS 中的 RXFNE 位）。应用程序从接收状态读取和出栈寄存器(GRXSTSP)中读取数据包信息，最后通过读取与端点相关的出栈地址从接收 FIFO 读

出相应数据。

34.5.10.2 设备 TxFIFO

模块为各个 IN 端点提供了专用的 FIFO。应用程序通过非周期发送 FIFO 大小寄存器(USBHS_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。

34.5.11 USBHS FIFO RAM 分配

34.5.11.1 主机模式

接收 FIFO RAM 分配

状态信息随各个接收数据包写入 FIFO。因此，必须至少为接收数据包分配（最大数据包大小/ 4）+ 2 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+2。通常，推荐的空间为（最大数据包/4 + 2）的两倍，这样当上一个数据包向 CPU 传送时，USB 可同时接收后续的数据包。

传输完成状态信息和该端点收到的最后一个数据包会一起被写入 FIFO。所以必须为此分配一个位置。

以 DMA 模式运行时，每个主机通道的 DMA 地址寄存器将存储在 FIFO 中，因此需要在 FIFO 中为每个通道保留一个位置存放其地址寄存器。

发送 FIFO RAM 分配

主机非周期性发送 FIFO 所需的最小 RAM 为所支持的所有非周期性 OUT 通道上传输的最大数据包的大小。

通常，推荐的空间为最大数据包大小的两倍，这样当 USB 正在发送当前数据包的同时，AHB 可以往发送 FIFO 填入下一个数据包。

主机周期性发送 FIFO 所需的最小 RAM 为所支持的所有周期性 OUT 通道上传输的最大数据包的大小。如果至少有一个同步 OUT 端点，则空间必须至少为该通道中最大数据包大小的两倍。

以 DMA 模式运行时，每个主机通道的 DMA 地址寄存器将存储在 FIFO 中，因此需

要在 FIFO 中为每个通道保留一个位置存放其地址寄存器。

34.5.11.2 设备模式

接收 FIFO RAM 分配

应用程序应为 SETUP 数据包分配 RAM：接收 FIFO 中必须保留 13 个位置以在控制端点上接收 SETUP 数据包。USBHS 模块不会向这些为 SETUP 数据包保留的位置写入任何其它数据。将会为全局 OUT NAK 分配一个位置。状态信息随各个接收数据包写入 FIFO。因此，必须至少为接收数据包分配（最大数据包大小/4）+1 的空间。如果使能了多个同步端点，则为接收连续数据包分配的空间必须至少为（最大数据包大小/4）的两倍+1。通常，推荐的空间为（最大数据包/4 + 1）的两倍，这样当上一个数据包向 CPU 传送时，USB 可同时接收后续的数据包。

传输完成状态信息和该端点收到的最后一个数据包会一起被推入 FIFO。通常情况下，推荐为每个 OUT 端点分配一个位置。

发送 FIFO RAM 分配

各个 IN 端点发送 FIFO 所需的最小 RAM 空间为该特定 IN 端点的最大数据包大小。

34.5.12 USBHS 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，USBHS 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 USBHS 模块能够高效填充 8KB RAM 缓冲区且 8KB 发送/接收数据足以满足一个全速帧所能容纳的数据量，因此 USB 系统在一帧之内可以无需应用程序干预达到最大 USB 带宽。

34.5.13 USBHS 中断和事件

USBHS 中断有以下四类，SOTP 模式唤醒中断 USBHS_WKUP、端点 1 OUT 中断 USBHS_EP1_OUT、端点 1 IN 中断 USBHS_EP1_IN、USBHS 全局中断 USBHS_GLB。

USBHS_WKUP 中断

USBHS_WKUP 中断用于 STOP 模式下，通过 USBHS_DP 或者 USBHS_DM 唤醒系统的 STOP 模式，该中断使能位为 INT_WUPEN.USH_WUEN。

使用 USBHS 唤醒系统 STOP 前，需要确保 USBHS 控制器处于挂起状态，并设定寄存器 USBHS_SYCTLREG 中的对应的滤波范围和使能滤波功能。

USBHS_EP1_OUT 中断

USBHS_EP1_OUT 中断是软件需要处理设备模式端点 1 的 OUT 中断，中断标志可以在 USBHS_DEACHINT 中查询。

USBHS_EP1_I N 中断

USBHS_EP1_IN 中断是软件需要处理设备模式端点 1 的 IN 中断，中断标志可以在 USBHS_DEACHINT 中查询

USBHS_GLB 中断

USBHS_GLB 中断是软件需要处理的主要中断，全局中断的标志位可在 USBHS_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	-
VBUSVINT	VBUS有效中断	设备	-
DISCINT	断开连接中断	主机	-
CIDSCHG	连接器ID线状态变化中断	主机或者设备	-
PTXFE	周期性TxFIFO空中断	主机	-
LPMINT	LPM中断	主机或者设备	
HCINT	主机通道中断	主机	-
HPRTINT	主机端口中断	主机	-
DATAFSUSP	数据获取挂起	设备	-
IPXFR/INCOMPISOOUT	未完成周期性传输/未完成	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	设备	-
NPTXFE	非周期性TxFIFO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 34-2 USBHS_GLB 中断事件表

34.6 USBHS 编程模型

34.6.1 USBHS 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 34.5.2 USBHS 模式决定。

本节介绍了 USBHS 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 USBHS_GAHBCFG 寄存器中编程以下字段：
   - 全局中断屏蔽位 GINTMSK = 1
   - RxFIFO 非空（USBHS_GINTSTS 中的 RXFNE 位）
   - 周期性 TxFIFO 空门限
2. 在 USBHS_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 USBHS_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 USBHS_GINTSTS 中的 CMOD 位，软件可确定 USBHS 控制器是在主机模式还是设备模式下工作。

34.6.2 USBHS 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 USBHS_GINTMSK 寄存器中的 HPRTINT 以将对取消屏蔽。
2. 编程 USBHS_HCFG 寄存器以选择全速主机。
3. 将 USBHS_HPRT 中的 PPWR 位编程为 1，给 USB 总线提供 VBUS。
4. 等待 USBHS_HPRT 中的 PCDET 中断。这表示某设备已连接到主机端口。
5. 将 USBHS_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 USBHS_HPRT 中的 PRST 位编程为 0。
8. 等待 USBHS_HPRT 中的 PENCHNG 中断。
9. 读取 USBHS_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 USBHS_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 USBHS_GRXFSIZ 寄存器以选择接收 FIFO 的大小。
13. 编程 USBHS_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 USBHS_HPTXFSIZ 寄存器，以选择用于周期性事务的周期性通信发送 FIFO 的大小和起始地址。

要与设备通信，系统软件必须初始化并使能至少一个通道。

34.6.3 USBHS 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 USBHS_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 USBHS_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 USBHS_GINTSTS 中的 VBUSVINT 中断，表示进入供电状态。

4. 等待 USBHS_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 USBHS_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 USBHS_DSTS 寄存器以确定枚举速度并执行枚举完成时的端点初始化中所列的步骤。

此时，设备已准备好接受 SOF 数据包并在控制端点 0 上执行控制传输。

34.6.4 USBHS DMA 模式

USB 使用 AHB 主接口来获取发送数据包数据（AHB 到 USB）和接收数据更新（USB 到 AHB）。AHB 主接口使用经过编程的 DMA 地址（主机模式下的 HCDMAx 寄存器和设备模式下的 DIEPDMAx/DOEPDMAx 寄存器）来访问数据缓冲区。

34.6.5 USBHS 主机编程模型

34.6.5.1 通道初始化

应用程序必须初始化一个或多个通道，之后才能与所连接的设备通信。

要初始化和使能通道，应用程序必须执行以下步骤：

1. 编程 USBHS_GINTMSK 寄存器以取消对以下位的中断屏蔽：
   - 用于 OUT 事务的非周期性发送 FIFO 为空（在流水线事务级别工作且数据包计数字段编程值大 1 时适用）。
   - 用于 OUT 事务的非周期性发送 FIFO 为半空（在流水线事务级别工作且数据包计数字段编程值大 1 时适用）。
2. 编程 USBHS_HAINTMSK 寄存器以使能所选通道中断。
3. 编程 USBHS_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 USBHS_HCTSIZx 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 USBHS_HCCHARx 寄存器，指定设备的端点特性，例如类型、速度、方向等。（仅当应用程序准备好发送或接收数据包时，才能通过将通道使能位置 1 来使能通道）。

34.6.5.2 通道停止

应用程序可以通过编程 USBHS_HCCHARx 寄存器将 CHDIS 和 CHENA 位置 1 来禁止任何通道。这会使 USBHS 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 USBHS_HCINTx 中的 CHH 中断。USBHS 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 USBHS_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 USBHS_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 USBHS_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。

34.6.5.3 Ping 协议

当 USBHS 主机工作在高速模式下时，如果应用程序要与高速批量或控制（数据和状态阶段）OUT 端点进行通信，则必须使用 ping 协议。

当应用程序接收到 NAK/NYET/TXER 中断时，必须使用 pin 协议。当 USBHS 主机接收到 上述其中一个响应时，它不会在该端点上继续执行任何通信事务，而是丢弃所有已发出或已获取 OUT 请求（从请求队列中），然后清空发送 FIFO 中的相应数据。

这仅在从模式下有效。在从模式下，应用程序可通过以下两种方式发送 ping 令牌：在

使能通道之前，将 HCTSIZx 中的 DOPIN 位置 1，或者在通道已使能之后，对 HCTSIZx 寄存器执行写操作时将 DOPING 位置 1。这将使能 USBHS 主机将 ping 请求写入到请求队列中。应用程序必须等待设备对 ping 令牌的响应（NAK、ACK 或 TXERR 中断），然后才能继续执 行通信事务或发送另一个 ping 令牌。仅当应用程序从设备的 OUT 端点接收到对 ping 令牌的 ACK 响应后，才能继续执行数据通信事务。在 DMA 模式下工作时，对于批量/控制 OUT 事务，应用程序不需要在收到 NAK/NYET 回复时将 HCTSIZx 中的 DOPING 位置 1。USBHS 主机会自动将 HCTSIZx 中的 DOPING 位置 1，然后在批量/控制 OUT 传输时发出 ping 令牌。USBHS 主机会持续发送 ping 令牌，直至收到 ACK 为止，随后自动切换到数据通信事务。

34.6.6 USBHS 设备编程模型

34.6.6.1 USB 复位时的端点端点初始化

1. 为所有 OUT 端点将 NAK 位置 1
   - USBHS_DOEPCTLx 中，SNAK = 1（对于所有 OUT 端点）
2. 取消对以下中断位的屏蔽
   - USBHS_DAINTMSK 中，INEP0=1（控制 0 IN 端点）
   - USBHS_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 USBHS_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 USBHS_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

- USBHS_DOEPTSIZ0 中的 STUPCNT=3（接收最多 3 个连续的 SETUP 数据包）

此时，接收 SETUP 数据包所需的所有初始化工作便已完成。

34.6.6.2 USB 复位时的端点端点初始化

1. 在枚举完成中断（USBHS_GINTSTS 中的 ENUMDNE）中，读取 USBHS_DSTS 寄存器以确定设备的枚举速度。
2. 对 USBHS_DIEPCTL0 中的 MPSIZ 字段进行编程以设置最大数据包大小。该步骤配置控制端点 0。控制端点的最大数据包大小取决于枚举速度。

此时，设备已准备好接收 SOF 数据包并配置为在控制端点 0 执行控制传输。

34.6.6.3 收到 SetAddress 命令时的端点初始化

本节介绍了应用程序在 SETUP 数据包中接收到 SetAddress 命令时必须执行的操作。

1. 使用在 SetAddress 命令中接收到的设备地址来对 USBHS_DCFG 寄存器进行编程
2. 对模块进行编程以发出状态阶段的 IN 数据包

34.6.6.4 收到 SetConfiguration/SetInterface 命令时的端点初始化

本节介绍了应用程序在 SETUP 包中接收 SetConfiguration 或 SetInterface 命令时必须执行的操作。

3. 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。
4. 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。
5. 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。
6. 使用 USBHS_DAINTMSK 寄存器使能有效端点的中断，屏蔽无效端点的中断。
7. 为每个 FIFO 设置数据 FIFO RAM。
8. 配置完所有必需的端点后，应用程序必须对模块进行编程以发送状态阶段的 IN 数据包。

此时，设备模块已可以接收和发送任何类型的数据包。

34.6.6.5 端点激活

本节介绍激活设备端点或者将现有设备端点配置为新类型所需的步骤。

1. 在 USBHS_DIEPCTLx 寄存器（对于 IN 或双向端点）或 USBHS_DOEPCTLx 寄存器（对于 OUT 或双向端点）的以下字段中，对所需端点的特性进行编程。
   - 最大数据包大小
   - USB 活动端点位置 1
   - 端点初始数据同步位（对于中断和批量端点）
   - 端点类型
   - TxFIFO 编号
2. 激活端点后，模块便开始解码发送到该端点的令牌，并在收到的令牌有效的情况下回复有效握手信号。

34.6.6.6 端点停用

本节介绍停用现有端点所需的步骤。

1. 在要停用的端点中，将 USBHS_DIEPCTLx 寄存器（对于 IN 或双向端点）或 USBHS_DOEPCTLx 寄存器(对于 OUT 或双向端点)中的 USB 活动端点位清零。
2. 停用端点后，模块便会忽略发送到该端点的令牌，从而导致 USB 超时。

34.6.7 USBHS 操作模型

34.6.7.1 SETUP 和 OUT 数据传输

本节介绍了数据 OUT 传输和 SETUP 事务期间的内部数据流和应用程序操作步骤。

数据包读取

本节介绍如何从接收 FIFO 读取数据包（OUT 数据和 SETUP 数据包）。

1. 捕获到 RXFNE 中断（USBHS_GINTSTS 寄存器）时，应用程序必须读取接收状态弹出寄存器 (USBHS_GRXSTSP)。
2. 应用程序可以通过写入 RXFNE=0（在 USBHS_GINTMSK 中）来屏蔽 RXFNE 中

断（在 USBHS_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 (USBHS_GINTSTS)。
6. 每次应用程序检测到 USBHS_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(USBHS_DOEPTSIZx) 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 USBHS_DOEPCTLx 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 USBHS_DOEPTSIZx 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(USBHS_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 USBHS_DOEPTSIZx 寄存器进行编程。

- STUPCNT=3

2. 等待 RXFNE 中断 (USBHS_GINTSTS) 并且从接收 FIFO 中读取数据包。
3. STUP 中断的触发(USBHS_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 USBHS_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，USBHS 控制器将生成中断（USBHS_DOEPINTx 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（USBHS_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 (USBHS_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 USBHS_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 USBHS_DCTL 中，SGONAK =1
2. 等待 USBHS_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 USBHS_DCTL 中的 SGONAK 位置 1，则在模块引发 GONAKEFF 中断 (USBHS_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 USBHS_GINTMSK 寄存器中的 GINAKEFFM 位执行写操作来暂时屏蔽此中断。
   - 在 USBHS_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 USBHS_DCTL 中的 SGONAK 位清零。此操作还会清除 GONAKEFF 中断 (USBHS_GINTSTS)。
   - 在 CGONAK 中，USBHS_DCTL=1
5. 如果应用程序在之前已屏蔽此中断，则必须按以下方式取消对该中断的屏蔽：
   - 在 GINTMSK 中，GINAKEFFM=1

将全局 OUT NAK 置 1

应用程序必须使用以下顺序禁止已使能的 OUT 端点。

应用程序编程顺序：

1. 禁止任何 OUT 端点前，应用程序必须在模块中使能全局 OUT NAK 模式。
   - 在 USBHS_DCTL 中，SGONAK=1
2. 等待 GONAKEFF 中断(USBHS_GINTSTS)

3. 通过编程以下字段来禁止 OUT 端点：
   - 在 USBHS_DOEPCTLx 中，EPDIS=1
   - 在 USBHS_DOEPCTLx 中，SNAK=1
4. 等待 EPDISD 中断 (USBHS_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 USBHS_DOEPCTLx 中，EPDIS=0
   - 在 USBHS_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 USBHS_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 × (MPSIZ[EPNUM] + 4 - (MPSIZ[EPNUM] mod 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 位置 1 后，将忽略同步或非同步数据包并且不会将它们写入接收 FIFO，同时非同步 OUT 令牌会收到 NAK 握手应答。
4. 在数据写入接收 FIFO 后，应用程序将从接收 FIFO 中读取数据并将数据写入外部存储器，一次一个数据包，逐个端点过来。
5. 在 AHB 上向外部存储器写入完每个数据包后，端点的传输大小都会自动减去该数据包的大小。
6. 在以下情况时，OUT 端点的 OUT 数据传输完成状态将写入接收 FIFO：
   - 传输大小为 0 并且数据包计数为 0
   - 写入接收 FIFO 的最后一个 OUT 数据包是短数据包
     （数据包大小:0~最大数据包大小-1）
7. 当应用程序弹出此状态条目（OUT 数据传输完成），并生成该端点的传输完成中断，同时清零端点使能位。

应用程序编程顺序：

1. 使用传输大小和相应数据包个数对 USBHS_DOEPTSIZx 寄存器进行编程。
2. 使用端点特性对 USBHS_DOEPCTLx 寄存器进行编程，并将 EPENA 和 CNAK 位置 1。
   - 在 USBHS_DOEPCTLx 中，EPENA=1
   - 在 USBHS_DOEPCTLx 中，CNAK =1
3. 等待 RXFNE 中断（在 USBHS_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 (USBHS_DOEPINTx)，以表示非同步 OUT 数据传输成功完成。
5. 读取 USBHS_DOEPTSIZx 寄存器，以确定有效数据量。

通用同步 OUT 数据传输

本节介绍常规的同步 OUT 数据传输。

应用程序要求：

1. 非同步 OUT 数据传输的所有应用程序要求均适用于同 OUT 数据传输。
2. 对于同步 OUT 数据传输中的传输大小和数据包计数字段，必须始终将其设置为单个帧中可接收的最大数据包大小的数据包数目。同步类型的 OUT 数据传输事务必须在一个帧内完成。
3. 在周期性帧结束（USBHS_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(USBHS_GINTSTS) 之后 SOF(USBHS_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 USBHS_DOEPCTLx 中）= FNSOF [0]（在 USBHS_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 USBHS_DOEPTSIZx 中的 RXDPID 字段。

应用程序编程顺序：

1. 使用传输大小和相应数据包计数对 USBHS_DOEPTSIZx 寄存器进行编程
2. 使用端点特性对 USBHS_DOEPCTLx 寄存器进行编程，并将端点使能位、清除 NAK 位和奇数/偶数帧位置 1。
   - EPENA1
   - CNAK=1
   - EONUM=（0：偶数/1：奇数）
3. 等待 RXFNE 中断（在 USBHS_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 USBHS_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 USBHS_GINTSTS 中的 IISOOXFRM 中断。
6. 读取 USBHS_DOEPTSIZx 寄存器以确定接收的传输大小以及确定帧中接收的数据的有效性。仅当符合以下条件之一时，应用程序才必须将存储器中接收的数据视为有效数据：
   - RXDPID=D0（在 USBHS_DOEPTSIZx 中）并且接收该有效数据的 USB 数据包数量=1
   - RXDPID=D1（在 USBHS_DOEPTSIZx 中）并且接收该有效数据的 USB 数据包数量=2
   - RXDPID=D2（在 USBHS_DOEPTSIZx 中）并且接收该有效数据的 USB 数据包数量=3

接收该有效数据的 USB 数据包数量=应用程序编程的初始数据包个数-模块更新后的剩余数据包个数。

应用程序可将无效数据包丢弃。

不完整的同步 OUT 数据传输

本节介绍了同步 OUT 数据包出现丢包时应用程序编程顺序。

内部数据流：

1. 对于同步 OUT 端点，可能不会始终引发 XFRC 中断（在 USBHS_DOEPINTx 中）。如果模块丢弃同步 OUT 数据包，则在以下情况下，应用程序可能无法检测到 XFRC 中断 (USBHS_DOEPINTx)：
   - 在接收 FIFO 无法容纳完整的 ISO OUT 数据包时，模块将丢弃接收到的 ISO OU 数据
   - 接收到的同步 OUT 数据包存在 CRC 错误
   - 模块接收到的同步 OUT 令牌损坏
   - 应用程序从接收 FIFO 中读取数据的速度非常缓慢
2. 如果模块在所有同步 OU 端点的传输完成前检测到周期性帧结束，将触发未完成同步 OUT 数据中断（USBHS_GINTSTS 中的 IISOOXFRM），指示至少有一个同步 OUT 端点上未触发 XFRC 中断（在 USBHS_DOEPINTx 中）。此时，未完成传输的端点仍保持使能，但在 USB 的该端点上，没有进行中的有效传输。

应用程序编程顺序：

1. 硬件触发 IISOOXFRM 中断 (USBHS_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 (USBHS_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 USBHS_GINTSTS 中）时，应用程序必须读取所有同步 OUT 端点的控制寄存器 (USBHS_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 USBHS_DOEPCTLx 中）= FNSOF [0]（在 USBHS_DSTS 中）
- EPENA=1（在 USBHS_DOEPCTLx 中）

4. 在检测到 SOF 中断（在 USBHS_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 USBHS_DOEPCTLx 中的 EPDIS 位置 1 来禁止端点。
6. 等待 EPDIS 中断（在 USBHS_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 USBHS_DOEPCTL 中），而不是将 USBHS_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 USBHS_DOEPCTLx 中）清零。
4. 如果应用程序由于收到主机的 SetFeature.Endpoint Halt 或 ClearFeature.Endpoint Halt 命令来设置或清除端点的 STALL 状态，则必须在该控制端点上的状态阶段传输前，将 STALL 位置 1 或清零。

34.6.7.2 IN 数据传输

数据包写入

本节介绍在已使能专用发送 FIFO 的情况下应用程序如何将数据包写入端点 FIFO。

1. 应用程序可以选择轮询模式或中断模式。

- 在轮询模式下，应用程序通过读取 USBHS_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 USBHS_DIEPINTx 中），然后读取 USBHS_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 USBHS_DIEPCTLx 寄存器执行读-修改-写操作，以避免在将端点使能位置 1 的同时，修改寄存器中的其它内容。

   如果有足够空间，应用程序可将同一端点的多个数据包写入发送 FIFO。对于周期性 IN 端点，应用程序只能一次写入一个帧内的多个数据包。只有先前一个帧的通信事务传输完成之后，应用程序才会写入下一个帧内要发送的所有数据包。

将 IN 端点 NAK 置 1

内部数据流：

1. 当应用程序将特定端点的 IN NAK 置 1 时，模块将停止端点上的数据发送，而不考虑端点发送 FIFO 中的数据是否可用。
2. 非同步端点收到 IN 令牌，回复 NAK 握手应答。

- 同步端点收到 IN 令牌，返回零长度数据包

3. 模块在 USBHS_DIEPINTx 中触发 INEPNE（IN 端点 NAK 有效）中断以响应 USBHS_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 USBHS_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。

- USBHS_DIEPCTLx 中的 SNAK=1

2. 等待 USBHS_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 USBHS_DIEPCTLx 中的 NAK 状态位 (NAKSTS) 清零。此操作还会清除 INEPNE 中断（在 USBHS_DIEPINTx 中）。
   - 在 USBHS_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。
   - USBHS_DIEPCTLx 中的 SNAK=1
3. 等待 USBHS_DIEPINTx 中的 INEPNE 中断。
4. 将必须禁止的端点的 USBHS_DIEPCTLx 寄存器中的以下位置 1。
   - USBHS_DIEPCTLx 中的 EPDIS=1
   - USBHS_DIEPCTLx 中的 SNAK=1
5. USBHS_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 USBHS_DIEPCTLx 中，EPENA=0
   - 在 USBHS_DIEPCTLx 中，EPDIS=0
6. 应用程序必须为周期性 IN EP 读取 USBHS_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 USBHS_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 USBHS_GRSTCTL 中）=端点发送 FIFO 编号
- TXFFLSH（在 USBHS_GRSTCTL 中）=1

  应用程序必须轮询 USBHS_GRSTCTL 寄存器，直至模块将 TXFFLSH 位清零，这表示 FIFO 清空操作结束。要在该端点上发送新数据，应用程序可以在稍后重新使能该端点。

通用非周期性 IN 数据传输

应用程序要求：

1. 建立 IN 传输前，应用程序必须确保组成一次 IN 传输的每个数据包都可以容纳在单个缓冲区中。
2. 对于 IN 传输，端点传输大小寄存器中的传输大小字段表示本次传输的有效数据量，它由多个最大数据包大小和单个短数据包组成。该短数据包在传输结束时发送。

- 要发送多个最大数据包大小的数据包并在传输结束时外加一个短数据包：

  传输大小[EPNUM] = x × MPSIZ[EPNUM] + sp

  如果 (sp > 0)，数据包计数[EPNUM] = x + 1。

  否则，数据包计数[EPNUM] = x
- 要发送单个零长度数据包：

  传输大小[EPNUM] = 0

  数据包计数[EPNUM] = 1
- 要发送多个最大数据包大小的数据包并在传输结束时外加一个零长度数据包，应用程序必须将传输拆分为两个部分。

  第一部分发送最大数据包大小的数据包，第二部分仅发送零长度数据包。

  第一次传输：传输大小[EPNUM] = x × MPSIZ[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) × MPSIZ[EPNUM]
- 要通过 USB 发送的剩余数据量 = (应用程序编程的初始传输大小 - 已通过 USB 发送的数据量)

内部数据流：

1. 应用程序必须在特定端点的寄存器中设置传输大小和数据包计数字段，并使能该端点来发送数据。
2. 应用程序还必须向该端点的发送 FIFO 写入必需的数据。
3. 应用程序每向发送 FIFO 写入一个数据包，该端点的传输大小便会自动减去该数据包大小。应用程序持续从存储器获取数据来写入发送 FIFO，直到该端点的传输大小变为 0。向 FIFO 写入数据后，“FIFO 中的数据包数”计数会递增（这是一个 3 位计数，由模块在内部进行维护，每个 IN 端点发送 FIFO 对应一个。在 IN 端点 FIFO 中，模块所维护的最大数据包数始终为八个）。对于零长度数据包，每个 FIFO 均另有一个单独的标志，FIFO 中没有任何数据。
4. 当数据写入发送 FIFO 后，模块会在接收到 IN 令牌时将这些数据送出。每个数据包发送出去并收到回复的 ACK 握手信号后，该端点的数据包计数都会递减 1，直到数据包计数变 0 为止。发生超时时，数据包计数不会递减。
5. 对于零长度数据包（由内部零长度标志指示），模块会针对 IN 令牌发出一个零长度数据包，并递减数据包计数字段的值。
6. 如果接收到 IN 令牌的端点对应的 FIFO 中无数据，且该端点的数据包计数字段为零，则模块会针对该端点生成一个“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 USBHS_DIEPTSIZx 寄存器进行编程。
2. 使用端点特性对 USBHS_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 USBHS_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 USBHS_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：

  传输大小[EPNUM] = x × 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) × MPSIZ =< XFERSIZ <= MCNT × MPSIZ
   - PKTCNT = MCNT（在 USBHS_DIEPTSIZx 中）
   - 如果 XFERSIZ < MCNT × MPSIZ，则传输的最后一个数据包为短数据包
   - 请注意：MCNT 位于 USBHS_DIEPTSIZx 中、MPSIZ 位于 USBHS_DIEPCTLx 中、PKTCNT 位于 USBHS_DIEPTSIZx 中、XFERSIZ 位于 USBHS_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 USBHS_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 USBHS_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 USBHS_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 USBHS_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 USBHS_DIEPINTx 中）时如果 USBHS_DIEPINTx 中未产生 ITTXFE 中断，则表示成功完成同步 IN 传输。读取 USBHS_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 USBHS_DIEPINTx 中）时无论是否产生 ITTXFE 中断（在 USBHS_DIEPINTx 中），都表示成功完成中断 IN 传输。读取 USBHS_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 USBHS_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（USBHS_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 USBHS_DIEPINTx 中检测到“TxFIFO 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（USBHS_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 USBHS_DIEPINTx 中的“TxFIFO 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 USBHS_GINTSTS 中）。
2. 硬件触发未完成同步 IN 传输中断（在 USBHS_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 USBHS_DIEPCTLx 寄存器中的下列字段进行编程以禁止端点：
   - USBHS_DIEPCTLx 中的 SNAK=1
   - USBHS_DIEPCTLx 中的 EPDIS=1
6. 硬件触发 USBHS_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 USBHS_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. USBHS_DIEPCTLx 中的 EPDIS=1（当端点已使能时）
   - USBHS_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 USBHS_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 USBHS_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 USBHS_DIEPINTx 的 ITTXFE 中断和 USBHS_DOEPINTx 的 OTEPDIS 中断（当模块已完成传输 SETUP 数据包指定的数据量后）。随后，当应用程序收到此中断时，必须将相应端点控制寄存器中的 STALL 位置 1 并清除此中断。

34.7 寄存器说明

应用程序通过 AHB 从接口对控制和状态寄存器进行读写操作，以此来控制 USBHS 模块。USBHS 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：

- USBHS 系统控制寄存器
- 模块全局寄存器
- 主机模式寄存器
- 设备模式寄存器
- 电源和时钟门控控制寄存器
- 数据 FIFO(DFIFO)访问寄存器

其中 USBHS 系统控制寄存器与其它寄存器基地址不同，该寄存器独立于 USBHS 模块以控制 USBHS 模块的相关设定。

只有模块全局寄存器、电源和时钟门控控制寄存器和数据 FIFO 访问寄存器可以在主机和设备模式下进行访问。当 USBHS 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 USBHS.GINTSTS.NMIS 位反映。当模快从一种角色模式切换到另一种角色模式时，新工作模式下的寄存器必须重新编程为上电复位后的状态。

控制状态寄存器存储器映射

主机和设备模式寄存器占用不同的地址。所有寄存器均在 AHB 时钟域内实现。

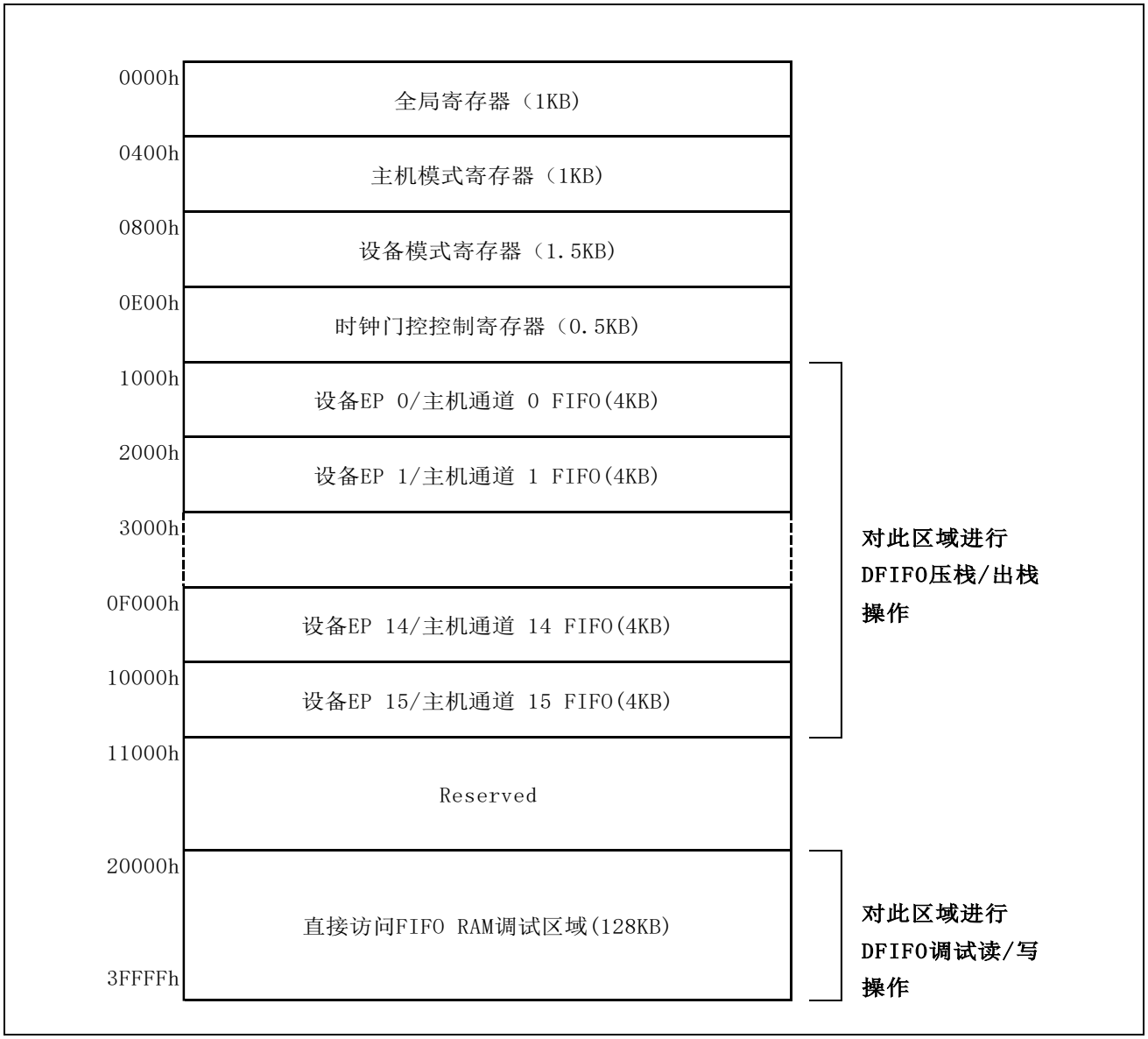


图 34-7 USBHS 控制状态 寄存器存储器映射

USBHS 模块寄存器列表以及基地址请参考表 34-3~表 34-4 USBHS 寄存器一览表。

USB 系统控制寄存器基地址：0x40055400

（USB系统控制寄存器）寄存器名	偏移地址	复位值
USB系统控制寄存器(USBHS_SYCTLREG)	0x00	0x0000 0000

表 34-3 USBHS 系统控制寄存器一览表

USBHS 模块寄存器基地址：0x400C0000

（USBHS全局寄存器）寄存器名	偏移地址	复位值
USBHS VBUS控制寄存器(USBHS_GVBUSCFG)	0x00	0x00XX 0000
USBHS AHB控制寄存器(USBHS_GAHBCFG)	0x08	0x0000 0000
USBHS USB配置寄存器(USBHS_GUSBCFG)	0x0c	0x0000 1410
USBHS复位寄存器(USBHS_GRSTCTL)	0x10	0x8000 0000
USBHS模块中断寄存器(USBHS_GINTSTS)	0x14	0x1400 0020
USBHS中断屏蔽寄存器(USBHS_GINTMSK)	0x18	0x0000 0000
USBHS接收状态调试读取寄存器(USBHS_GRXSTSR)	0x1c	0x0000 0000
USBHS接收状态读取和出栈寄存器(USBHS_GRXSTSP)	0x20	0x0000 0000
USBHS接收FIFO大小寄存器(USBHS_GRXFSIZ)	0x24	0x0000 0800
USBHS 主机非周期性发送FIFO大小寄存器 (USBHS_HNPTXFSIZ)/ 设备端点0发送FIFO大小寄存器(USBHS_DIEPRXF0)	0x28	0x0800 0800
USBHS非周期性发送状态寄存器(USBHS_HNPTXSTS)	0x2c	0x0008 0800
USBHS模块ID寄存器(USBHS_CID)	0x3c	0x1234 5678
USBHS LPM配置寄存器(USBHS_GLPMCFG)	0x54	0x0000 0000
USBHS周期性发送FIFO大小寄存器(USBHS_HPTXFSIZ)	0x100	0x0800 1000
USBHS 设备IN端点n发送FIFO大小寄存器 (USBHS_DIEPTXFx)	0x100+x*4(x=1~15)	0x0800 0100+(x-1)*0x800

（USBHS主机控制和状态寄存器）寄存器名	偏移地址	复位值
USBHS主机配置寄存器(USBHS_HCFG)	0x400	0x0000 0200
USBHS主机帧时间间隔寄存器(USBHS_HFIR)	0x404	0x0000 EA60
USBHS主机帧编号/帧剩余时间间隔寄存器 (USBHS_HFNUM)	0x408	0x0000 3FFF
USBHS主机周期性发送FIFO/队列状态寄存器 (USBHS_HPTXSTS)	0x410	0x0008 0800
USBHS主机全体通道中断寄存器(USBHS_HAINT)	0x414	0x0000 0000
USBHS主机全体通道中断屏蔽寄存器 (USBHS_HAINTMSK)	0x418	0x0000 0000
USBHS主机端口控制和状态寄存器(USBHS_HPRT)	0x440	0x0000 0000
USBHS主机通道x特性寄存器(USBHS_HCCHARx)	0x500+x*0x20(x=0~15)	0x0000 0000
USBHS主机通道x分离控制寄存器(USBHS_HCSPLTx)	0x504+x*0x20(x=0~15)	0x0000 0000
USBHS主机通道x中断寄存器(USBHS_HCINTx)	0x508+x*0x20(x=0~15)	0x0000 0000
USBHS主机通道x中断屏蔽寄存器 (USBHS_HCINTMSKx)	0x50c+x*0x20(x=0~15)	0x0000 0000
USBHS主机通道x传输大小寄存器(USBHS_HCTSIZx)	0x510+x*0x20(x=0~15)	0x0000 0000
USBHS主机通道x DMA地址寄存器(USBHS_HCDMAx)	0x514+x*0x20(x=0~15)	0xXXXX XXXX

（USBHS设备控制和状态寄存器）寄存器名	偏移地址	复位值
USBHS设备配置寄存器(USBHS_DCFG)	0x800	0x0822 0000
USBHS设备控制寄存器(USBHS_DCTL)	0x804	0x0000 0002
USBHS设备状态寄存器(USBHS_DSTS)	0x808	0x0000 0002
USBHS设备IN端点通用中断屏蔽寄存(USBHS_DIEPMSK)	0x810	0x0000 0000
USBHS设备OUT端点通用中断屏蔽寄存器 (USBHS_DOEPMSK)	0x814	0x0000 0000
USBHS设备全体端点中断寄存器(USBHS_DAINT)	0x818	0x0000 0000
USBHS设备全体端点中断屏蔽寄存(USBHS_DAINTMSK)	0x81c	0x0000 0000
USBHS设备阈值控制寄存器（USBHS_DTHRCTL）	0x830	0x0c10 0020
USBHS设备IN端点FIFO空中断屏蔽寄存器 (USBHS_DIEPEMPMSK)	0x834	0x0000 0000
USBHS设备单个端点中断寄存器（USBHS_DEACHINT）	0x838	0x0000 0000
USBHS设备个单端点中断屏蔽寄存器 （USBHS_DEACHINTMSK）	0x83c	0x0000 0000
USBHS设备IN端点1中断屏蔽寄存器 （USBHS_DIEPEACHMSK1）	0x844	0x0000 0000
USBHS设备OUT端点1中断屏蔽寄存器 （USBHS_DOEPEACHMSK1）	0x884	0x0000 0000
USBHS设备IN端点0控制寄存器(USBHS_DIEPCTL0)	0x900	0x0000 8000
USBHS设备IN端点x控制寄存器(USBHS_DIEPCTLx)	0x900+x*0x20(x=1~15)	0x0000 0000
USBHS设备IN端点x中断寄存器(USBHS_DIEPINTx)	0x908+x*0x20(x=0~15)	0x0000 0080
USBHS设备IN端点x传输大小寄存器(USBHS_DIEPSIZx)	0x910+x*0x20(x=0~15)	0x0000 0000
USBHS设备IN端点x DMA地址寄存器 (USBHS_DIEPDMAx)	0x914+x*0x20(x=0~15)	0xXXXX XXXX
USBHS设备IN端点x发送FIFO状态寄存器 (USBHS_DTXFSTSx)	0x918+x*0x20(x=0~15)	0x0000 0800
USBHS设备OUT端点0控制寄存器(USBHS_DOEPCTL0)	0xb00	0x0000 0000
USBHS设备OUT端点x控制寄存器(USBHS_DOEPCTLx)	0xb00+x*0x20(x=1~15)	0x0000 0000
USBHS设备OUT端点x中断寄存器(USBHS_DOEPINTx)	0xb08+x*0x20(x=0~15)	0x0000 0000
USBHS设备OUT端点x传输大小寄存(USBHS_DOEPSIZx)	0xb10+x*0x20(x=0~15)	0x0000 0000

USBHS设备OUT端点x DMA地址寄存器 (USBHS_DOEPDMAx)	0xb14+x*0x20(x=0~15)	0xXXXXXXX

（USBHS电源和时钟没空控制寄存器）寄存器名	偏移地址	复位值
USBHS电源和门控时钟控制急促器(USBHS_PCGCCTL)	0xe00	0x0000 0000

表 34-4 USBHS 系统控制寄存器一览表

34.7.1 USB 系统控制寄存器

34.7.1.1 USB 系统控制寄存器(USB_SYCTLREG)

USB System Control Register

偏移地址：0x00

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	USBHS_NFE	USBHS_NFS[1:0]		-	-	-	-	--	USBHS_NFE	USBHS_NFS[1:0]	

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	USBHS_FSPHYE	USBHS_SOFEN	USBHS_DFB	-	-	-	-	-	-	USBHS_SOFEN	USBHS_DFB

位	标记	位名	功能	读写
b31~b27	Reserved	-	必须保持复位值。	R/W
b26	USBHS_NFE	USBHS滤波使能寄存器	USBHS滤波使能寄存器 该寄存器用于控制STOP模式下，USBHS片上全速PHY DP/DM模拟滤波器的开关 0：模拟滤波器关 1：模拟滤波器开，滤波范围请参考USBHS_NFS[1:0]的设定	R/W
b25~b24	USBHS_NFS	USBHS滤波选择寄存器	USBHS滤波选择寄存器 该寄存器用于控制STOP模式下，USBHS片上全速PHY DP/DM模拟滤波器的滤波范围 00b：滤波宽度档位1 01b：滤波宽度档位2 10b：滤波宽度档位4 11b：滤波宽度档位4 各档位具体值请参考“电气特性 USB片上全速PHY STOP模式下滤波特性”章节。	R/W
b23~b19	Reserved	-	必须保持复位值。	R/W
b18	USBFS_NFE	USBFS滤波使能寄存器	USBFS滤波使能寄存器 该寄存器用于控制STOP模式下，USBHS片上全速PHY DP/DM模拟滤波器的开关 0：模拟滤波器关	R/W

			1：模拟滤波器开，滤波范围请参考USBFS_NFS[1:0]的设定	
b17~b16	USBFS_NFS	USBFS滤波选择寄存器	USBFS滤波选择寄存器 该寄存器用于控制STOP模式下，USBHS片上全速PHY DP/DM模拟滤波器的滤波范围 00b：滤波宽度档位1 01b：滤波宽度档位2 10b：滤波宽度档位4 11b：滤波宽度档位4 各档位具体值请参考“49.电气特性 49.3.34 USB片上全速PHY STOP模式下滤波特性”章节。	R/W
b15~b11	Reserved	-	必须保持复位值。	R/W
b10	USBHS_FSPHYE	USBHS 片上全速PHY使能位	USBHS 片上全速PHY使能位 0：USBHS片上全速PHY挂起 1：USBHS片上全速PHY使能	R/W
b9	USBHS_SOFEN	USBHS SOF脉冲输出使能位	USBHS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	R/W
b8	USBHS_DFB	USBHS VBUS/ID管脚内部去抖滤波器旁路使能位	USBHS 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
b7~b2	Reserved	-	必须保持复位值。	R/W
b1	USBFS_SOFEN	USBFS SOF脉冲输出使能位	USBFS主机发出SOF或者设备成功接收到SOF时，16个系统时钟周期宽度的SOF脉冲从PAD输出使能 0：SOF脉冲不输出 1：SOF脉冲输出 注意：在设备模式和主机模式均可访问。	R/W
b0	USBFS_DFB	USBFS VBUS/ID管脚内部去抖滤波器旁路使能位	USBFS VBUS/ID管脚模块内部去抖动滤波器旁路使能位 0：模块内部去抖动滤波器有效 1：旁路模块内部去抖动滤波器 注意：在设备模式和主机模式均可访问。	R/W

34.7.2 USBHS 全局寄存器

这些寄存器在主机模式和设备模式下都可用，且在这两个模式间切换时无需对其进行重新编程。除非特别说明，否则寄存器描述中的位值以二进制表示。

34.7.2.1 USBHS VBUS 控制寄存器(USBHS_GVBUSCFG)

VBUS Configuration Register

偏移地址：0x00

复位值：0x00XX 0000

该寄存器可用于设定 VBUS 值从而忽略 VBUS 管脚的状态。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								VUBSVAL	VBUSOVEN	Reserved					

位	标记	位名	功能	读写
b31~b8	Reserved	-	必须保持复位值。	R/W
b7	VBUSVAL	VBUS值	VBUS值(VBUS Value) 用于设定USBFS的VBUS值，当设定为1，且VBUSOVEN设定1后对USBFS完成上电。 注意：仅可在设备模式下访问。	R/W
b6	VBUSOVEN	VBUS Override 使能	VBUS Override 使能(VBUS Override 使能) 用于将VBUSVAL设定的值反映到USBHS CORE的状态。仅当该位设置为1，VBUSVAL的值才有效。 注意：仅可在设备模式下访问。	R/W
b5~b0	Reserved	-	必须保持复位值。	R/W

34.7.2.2 USBHS AHB 控制寄存器(USBHS_GAHBCFG)

AHB Configuration Register

偏移地址：0x08

复位值：0x0000 0000

该寄存器可用于在上电后或更改角色模式时对模块进行配置。该寄存器主要包含 AHB 系统相关的配置参数。

应用程序必须在开始任何 AHB 或 USB 事务前对该寄存器进行编程。请勿在初始编程后更改该寄存器。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							PTXFELVL	TXFELVL	Reserved	DMAEN	HBSTLEN[3:0]				GINTMSK

位	标记	位名	功能	读写
b31~b9	Reserved	-	必须保持复位值。	R/W
b8	PTXFELVL	周期性TxFIFO空门限	周期性TxFIFO空门限 (Periodic TxFIFO empty level) 指示何时触发模块中断寄存器中的周期性TxFIFO空中断位（USBHS_GINTSTS中的PTXFE 位）。 0：PTXFE（位于USBHS_GINTSTS）中断指示周期性TxFIFO为半空状态 1：PTXFE（位于USBHS_GINTSTS）中断指示周期性TxFIFO为全空状态 注意：仅可主机模式下访问。	R/W
b7	TXFELVL	设备TxFIFO空门限	设备TxFIFO空门限 (TxFIFO 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（USBHS_DIEPINTx中的TXFE）。 0：TXFE（位于USBHS_DIEPINTx）中断指示IN端点TxFIFO为半空状态 1：TXFE（位于USBHS_DIEPINTx）中断指示IN端点TxFIFO为全空状态 注意：仅可在设备模式下访问。	R/W
b6	Reserved	-	必须保持复位值。	R/W
b5	DMAEN	DMA 使能	DMA 使能 (DMA enable) 0：模块以从模式运行 1：模块以DMA模式运行 注意：在设备模式和主机模式均可访问。	R/W

b4~b1	HBSTLEN	批量长度/类型	批量长度/类型 (Burst length/type) 0000b:单次 0001b: INCR 0011b:INCR4 0101b:INCR8 0111b:INCR16 其它值：保留 注意：在设备模式和主机模式均可访问。	R/W
b0	GINTMSK	全局中断屏蔽	全局中断屏蔽 (Global interrupt mask) 该位用于屏蔽全局中断或对全局中断取消屏蔽。中断状态寄存器由模块进行更新，与此位的设置无关。 0：屏蔽应用程序触发的中断 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。	R/W

34.7.2.3 USBHS USB 配置寄存器 (USBHS_GUSBCFG)

USBHS USB configuration register

偏移地址：0x00C

复位值：0x0000 1410

该寄存器可用于在上电或更改角色模式后对模块进行配置。其中包含与 USB 和 USB-PHY 相关的配置参数。

应用程序必须在开始任何 AHB 或 USB 事务前对该寄存器进行编程。请勿在初始编程后更改该寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved	FDMOD	FHMOD	Reserved			ULPIPD	PTCI	PCCI	Reserved	ULPIEVBUSI	ULPIEVBUSD	ULPICSM	ULPIAR	ULPIFSLS	Reserved

b15	b14	b13	b12	b11	b10	b9	8	b7	b6	b5	b4	b3	b2	b1	b0
PHYLPCS	Reserved	TRDT[3:0]				Reserved			PHYSEL	Reserved			TOCAL[2:0]		

位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R/W
b30	FDMOD	强制设备模式	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USBHS_ID输入引脚的状态。 0：正常模式，取决USBHS_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USBHS_ID输入引脚的状态。 0：正常模式，取决USBHS_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~b26	Reserved	-	必须保持复位值。	R/W
b25	ULPIPD	ULPI接口保护禁止	ULPI接口保护禁止（ULPI Interface protect disable） 该位控制PHY中内置的电路，以保护UPLI接口。该功能所采用的任何上拉或	R/W

			下拉电阻均可禁止。详细信息请参考ULPI规范。 0：使能接口保护电路 1：禁止接口保护电路	
b24	PTCI	通过指示符-	通过指示符（Indicator pass through） 该位控制互补输出是否需要通过内部VBUS有效电平比较器功能，之后方可用于RX CMD中的VBUS状态。详细信息请参考ULPI规范。 0：互补输出信号通过内部VBUS有效电平比较器的验证 1：互补输出信号不通过内部VBUS有效电平比较器的验证	R/W
b23	PCCI	互补指示符 -	互补指示符 (Indicator complement) 该位控制PHY将ExternalVbusIndicator输入信号反转并生成互补输出。有关详细信息，请参见ULPI 规范。 0：PHY 不将ExternalVbusIndicator信号反转 1：PHY 将ExternalVbusIndicator信号反转	R/W
b22	Reserved	-	必须保持复位值。	R/W
b21	ULPIEVBUSI	ULPI外部VBUS指示符-	ULPI外部VBUS指示符 (ULPI external VBUS indicator) 该位指示ULPI PHY使用外部 VBUS 过流指示器。 0：PHY使用内部VBUS有效比较器 1：PHY使用外部VBUS有效比较器	R/W
b20	ULPIEVBUSD	ULPI外部VBUS驱动器	ULPI 外部VBUS驱动器 (ULPI External VBUS Drive) 该位选择ULPI PHY使用内部或外部电源来驱动VBUS上的5 V电压。 0：PHY使用内部电荷泵驱动VBUS（默认） 1：PHY使用外部电源驱动VBUS。	R/W
b19	ULPICSM	ULPI 时钟挂起	ULPI时钟SuspendM (ULPI Clock SuspendM) 该位置位 ULPI PHY 接口控制寄存器中的 ClockSuspendM 位。 该位仅适用于串行模式。 0：PHY 在挂起期间断开内部时钟的电源 1：PHY 不断开内部时钟的电源	R/W
b18	ULPIAR	ULPI时钟自动恢复	ULPI 自动恢复 (ULPI Auto-resume) 该位置位 ULPI PHY 接口控制寄存器中的 AutoResume 位。 0：PHY 不支持自动恢复功能 1：PHY 使用自动恢复功能	R/W
b17	ULFSLS	ULPI FS/LS选择	ULPI FS/LS 选择 (ULPI FS/LS select) 应用程序使用该位为ULPI PHY选择FS/LS串行接口。 该位仅在ULPI PHY上选择了FS串行收发器的情况下有效。 0：ULPI接口 1：ULPI FS/LS串行接口	R/W
b16	Reserved	-	必须保持复位值。	R/W
b15	PHYLPCS	ULPI FS/LS选择	PHY 低功耗时钟选择 (PHY Low-power clock select) 该位用于选择480 MHz或48 MHz（低功耗）PHY模式。在FS和LS模式下，PHY 通常可使用48 MHz时钟运行，从而节约功耗。 0：480 MHz内部PLL时钟 1：48 MHz外部时钟在480 MHz模式下，UTMI 接口以60 MHz或30MHz运	R/W

			行，具体取决于选择的是8位还是16位数据宽度。在48 MHz模式下，UTMI接口在FS和LS模式下以48 MHz运行。	
b14	Reserved	-	必须保持复位值。	R/W
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 × AHB 时钟 + 1 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/W
b6	PHYSEL	USB 2.0高速ULPI PHY或USB片上全速串行收发器选择	USB 2.0高速ULPI PHY或USB片上全速串行收发器选择 (USB 2.0 high-speed ULPI PHY or USB 1.1 full-speed serial transceiver select) 0：USB 2.0高速ULPI PHY 1：USB片上全速串行收发器	R/W
b5~b3	Reserved	-	必须保持复位值。	R/W
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

34.7.2.4 USBHS 复位寄存器 (USBHS_GRSTCTL)

USBHS 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	CSRST

位	标记	位名	功能	读写
b31	AHBIDL	AHB主器件空闲	AHB主器件空闲(AHB master idle) 指示AHB主器件状态机处于空闲情况。 注意：在设备模式和主机模式均可访问。	R
b30	DMAREQ	AHB主器件空闲	DMA请求信号 (DMA request signal) 该位指示DMA请求正在进行中。用于调试。 注意：在设备模式和主机模式均可访问。	R
b30~b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10~b6	TXFNUM	TxFIFO编号	TxFIFO编号(TxFIFO number) 使用TxFIFO刷新位进行FIFO刷新的FIFO编号。只有在模块将TxFIFO刷新位清零后，方可更改此字段。 ● 00000： — 主机模式下刷新非周期性TxFIFO — 设备模式下刷新Tx FIFO 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：USBHS_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/W
b2	FCRST	主机帧计数器复位	主机帧计数器复位(Host frame counter reset) 应用程序对该位执行写操作时，模块中的帧数计数器复位。帧计数器复位后，由模块发送的下一个SOF的帧号为 0。 注意：在设备模式和主机模式均可访问。	R/W
b1	HSRST	HCLK域逻辑软复位	HCLK域逻辑软复位 (HCLK soft reset) 应用程序使用此位来刷新 AHB 时钟域中的控制逻辑。仅复位 AHB 时钟域流水线。 FIFO 不通过此位来刷新。 遵照协议终止 AHB 上的事务后，AHB 时钟域中的所有状态机均复位至空闲状态。 AHB 时钟域状态机所使用的 CSR 控制位清零。 要清零该中断，需要将由 AHB 时钟域状态机生成并用于控制中断状态的状态屏蔽位清零。 由于中断状态位并未清零，因此应用程序可以获取在该位置 1 后所发生的所有模块事件的 状态。 此位为自清零位，模块将在其中所有必要逻辑复位后将该位清零。该过程需要若干个时钟的 时间，具体取决于模块的当前状态。 注意：在设备模式和主机模式均可访问。	R/W

b0	CSRST	模块软复位	模块软复位 (Core soft reset) 按如下所述将HCLK和PCLK域复位： 除以下各位外，将各个中断和所有CSR寄存器位清零： — USBHS_PCGCCTL中的RSTPDMODL位 — USBHS_PCGCCTL中的GAYEHCLK位 — USBHS_PCGCCTL中的PWRCLMP位 — USBHS_PCGCCTL中的STPPCLK位 — USBHS_HCFG 中的FSLSPCS位 — USBHS_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。	R/W

34.7.2.5 USBHS 全局中断状态寄存器 (USBHS_GINTSTS)

USBHS 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 USBHS_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	VBUSVINT	DISCINT	CIDSCHG	LPMINT	PTXFE	HCINT	HPRTINT	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	ENUMDNE	USBRST	USBSUSP	ESUSP	Reserved	Reserved	GONAKEFF	GINAKEFF	NPTXFE	RXFLNE	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	VBUSVINT	VBUS有效中断	VBUS有效中断 (VBUS valid interrupt) 设备模式下，当检测到USBHS_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	检测到断开连接中断 (Disconnect detected interrupt) 当检测到设备断开连接时触发该中断。 通过软件对该位写1清零。 注意：仅可在主机模式下访问。	R/W
b28	CIDSCHG	连接器ID线状态变化中断	连接器ID线状态更改(Connector ID status change) 当连接器ID线状态发生更改时，模块将该位置1。 通过软件对该位写1清零。 注意：在设备模式和主机模式均可访问。	R/W
b27	LPMINT	LPM中断	LPM中断 设备模式 设备收到正确LPM传输且有效应答时该位置1。 主机模式 主机发送LPM后收到有效应答，或者超过设定的重试次数后该位置1。	R/W
b26	PTXFE	周期性TxFIFO空中断	周期性TxFIFO空中断 (Periodic TxFIFO empty interrupt) 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USBHS_GAHBCFG寄存器中的周期性TxFIFO空级别位（USBHS_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	主机通道中断 (Host channels interrupt) 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USBHS_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USBHS_HCINTx 寄存器，以确定引发中断的确切原因。应用程序必须先将USBHS_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	主机端口中断 (Host port interrupt) 模块将该位置1时，指示主机模式下 USBHS 控制器端口的状态发生变化。应用程序必须读取USBHS_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USBHS_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/W
b22	DATAFSUSP	数据获取挂起	数据获取挂起 (Data fetch suspended) 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	R/W

			— 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，USBHS将NAK响应发送到主机。为避免这种情况的发生，应用程序可以检查USBHS_GINTSTS 中的DATAFSUSP中断，该中断可确保在FIFO存满后再将全局 NAK 握手信号清零。或者，应用程序可以在将全局IN NAK握手信号清零时屏蔽“当 FIFO为空时接收到IN令牌中断。 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 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USBHS_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USBHS_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USBHS_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USBHS_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USBHS_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 USBHS_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/W

b15	EOPF	周期性帧结束中断	周期性帧结束中断 (End of periodic frame interrupt) 指示当前帧已达到USBHS_DCFG 寄存器中周期性帧间隔字段（USBHS_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RxFIFO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RxFIFO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USBHS_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/W
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USBHS_DCTL寄存器中由应用程序设置的“将全局OUT NAK置 1”位（USBHS_DCTL中的SGONAK位）已在模块中生效。通过写入USBHS_DCTL 寄存器中的“将全局 OUT NAK 清零”位（USBHS_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USBHS_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（USBHS_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USBHS_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TxFIFO空中断	非周期性TxFIFO空中断(Non-periodic TxFIFO empty interrupt) 当非周期性TxFIFO为半空或全空状态，且非周期性发送请求队列中至少存在可写入一个条目的空间时，将触发该中断。该 FIFO 为半空状态还是全空状态由USBHS_GAHBCFG寄存器中的非周期性TxFIFO空级别位（USBHS_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/W
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： — 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会影响模块运行。 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

34.7.2.6 USBHS 全局中断屏蔽寄存器 (USBHS_GINTMSK)

USBHS 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 (USBHS_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	LPMIM	PTXFEM	HCIM	HPRTIM	Reserved	DATAFSUSPM	IPXFRM/ INCOMPISOOUTM	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	USBSUSPM	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 mak) 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	DISCM	检测到断开连接中断屏蔽	检测到断开连接中断屏蔽 (Disconnect detected interrupt interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在主机模式下访问。	R/W
b28	CIDSCHGM	中断连接器ID线状态变化中断屏蔽	连接器ID线状态更改中断屏蔽(Connector ID status change interrupt mask) 0：屏蔽中断 1：使能中断 注意：在设备模式和主机模式均可访问。	R/W

b27	LPMINTM	LPM中断屏蔽	LPM中断屏蔽(LPM interrupt mask) 0：屏蔽中断 1：使能中断 注意：在设备模式和主机模式均可访问。	R/W
b26	PTXFEM	周期性TxFIFO空中断屏蔽	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/W
b22	DATAFSUSPM	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/ INCOMPISOOUTM	未完成周期性传输中断屏蔽/ 未完成OUT同步传输中断屏蔽	IPXFR:未完成周期性传输中断屏蔽(Incomplete periodic transfer interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在主机模式下访问。 INCOMPISOOUT: 未完成OUT同步传输中断屏蔽(Incomplete isochronous OUT transfer interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W
b20	IISOIXFRM	未完成IN同步传输中断屏蔽	未完成IN同步传输中断屏蔽(Incomplete isochronous IN transfer interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W
b19	OEPIM	OUT端点中断屏蔽	OUT端点中断屏蔽(OUT endpoint interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W

b18	IEPIM	IN端点中断屏蔽	IN端点中断屏蔽(IN endpoint interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W
b17~b16	Reserved	-	必须保持复位值。	R/W
b15	EOPFM	周期性帧结束中断屏蔽	周期性帧结束中断屏蔽(End of periodic frame interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W
b14	ISOODRPM	丢弃同步OUT数据包中断屏蔽	丢弃同步OUT数据包中断屏蔽(Isochronous OUT packet dropped interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W
b13	ENUMDNEM	枚举完成中断屏蔽	枚举完成中断屏蔽(Enumeration done interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W
b12	USBRSTM	USB复位中断屏蔽	USB复位中断屏蔽(USB reset 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
b11	USBSUSPM	USB挂起中断屏蔽	USB挂起中断屏蔽(USB suspend 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
b10	ESUSPM	早期挂起中断屏蔽	早期挂起中断屏蔽(Early suspend 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
b9~b8	Reserved	-	必须保持复位值。	R/W
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) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W
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) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W

b5	NPTXFEM	非周期性TxFIFO空中断屏蔽	非周期性TxFIFO空中断(Non-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
b4	RXFNEM	RxFIFO 非空中断屏蔽	RxFIFO非空中断屏蔽(RxFIFO non-empty 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
b3	SOFM	帧起始中断屏蔽	帧起始中断屏蔽(Start of frame 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
b2	Reserved	-	必须保持复位值。	R/W
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) 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
b0	Reserved	-	必须保持复位值。	R/W

34.7.2.7 USBHS 接收状态调试读取/USBHS 状态读取和出栈寄存器 (USBHS_GRXSTSR/USBHS_GRXSTSP)

USBHS Receive status debug read/USBHS 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（USBHS_GINTSTS 中的 RXFNE 位）置位时，应用程序必须仅弹出接收状态 FIFO。

主机模式：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved											PKTSTS[3:0]				DPID[1]

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DPID[0]	BCNT[11:0]											CHNUM[3:0]			

位	标记	位名	功能	读写
b31~b21	Reserved	-	必须保持复位值。	R/W
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：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved											PKTSTS[3:0]				DPID[1]

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DPID[0]	BCNT[11:0]											EPNUM[3:0]			

位	标记	位名	功能	读写
b31~b21	Reserved	-	必须保持复位值。	R/W
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

34.7.2.8 USBHS 接收 FIFO 大小寄存器 (USBHS_GRXFSIZ)

USBHS Receive FIFO size register

偏移地址：0x024

复位值：0x0000 0800

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[11:0]											

位	标记	位名	功能	读写
b31~b12	Reserved	-	必须保持复位值。	R/W
b11~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2048 上电复位值为最大Rx数据FIFO深度。	R/W

34.7.2.9 USBHS 主机非周期性发送 FIFO 大小寄存器(USBHS_HNPTXFSIZ)/端点 0 发送 FIFO 大小(USBHS_DIEPTXF0)

USBHS Host non-periodic transmit FIFO size register/Device endpoint0 transmit FIFO size register

偏移地址：0x028

复位值：0x08000800

此应用程序可以对必须分配给 TxFIFO 的 RAM 大小进行编程。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
NPTXFD[15:0]/TX0FD[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
NPTXFSA[15:0]/TX0FSA[15:0]															

位	标记	位名	功能	读写
b31~b16	NPTXFD/ TX0FD	非周期性TxFIFO 深度/端点0 TxFIFO深度	主机模式：NPTXFD 非周期性TxFIFO深度(Non-periodic TxFIFO depth) 以32位字为单位。 最小值为16 最大值为2048 设备模式：TX0FD 端点0 TxFIFO深度(Endpoint 0 TxFIFO depth) 以32位字为单位。 最小值为16 最大值为2048	R/W
b15~b0	NPTXFSA/ TX0FSA	非周期性发送RAM起始地址/端点0发送RAM起始地址	主机模式：NPTXFSA 非周期性发送RAM起始地址(Non-periodic transmit RAM start address) 此字段包含非周期性发送FIFO RAM的存储器起始地址。 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。	R/W

34.7.2.10 USBHS 非周期性发送 FIFO/队列状态寄存器 (USBHS_HNPTXSTS)

USBHS Host non-periodic transmit FIFO size register/Device endpoint0 transmit FIFO size register

偏移地址：0x02C

复位值：0x00080800

此只读寄存器包含非周期性 TxFIFO 和非周期性发送请求队列的自由空间信息。

此寄存器仅在主机模式有效，设备模式无效。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved	NPTXQTOP[6:0]							NPTQXSAV[7:0]							

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

位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R/W
b30~b24	NPTXQTOP	非周期性发送请求队列顶部	非周期性发送请求队列顶部 (Top of the non-periodic transmit request queue) 非周期性发送请求队列中MAC目前正在处理的条目。 位30:27：通道/端点编号(Channel/endpoint number) 位26:25： — 00：IN/OUT 令牌 — 01：长度为零的发送数据包 — 11：通道停止命令 位24：结束（所选通道/端点的最后一个条目）(Terminate (last entry for selected channel))	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。 0：非周期性发送请求队列已满 1：1 个位置可用 2：2 个位置可用 n：n 个位置可用 (其中，n范围：0~8) 其它值：保留	R

b15~b0	NPTXFSAV	非周期性TxFIFO可用空间	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。 0：非周期性 TxFIFO 已满 1：1 个字可用 2：2 个字可用 n：n 个字可用（其中，n范围：0~2048） 其它值：保留	R

34.7.2.11 USBHS 模块 ID 寄存器(USBHS_CID)

USBHS 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

34.7.2.12 USBHS LPM 配置寄存器(USBHS_GLPMCFG)

USBHS LPM configuration register

偏移地址：0x054

复位值：0x00000000

该寄存器为可编程 LPM 配置寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved			ENBSEL	LPMRCNTSTS[2:0]			SNDLPM	LPMRCNT[2:0]			LPMCHIDX[3:0]				L1RSMOK

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SLPSTS	LPMRSP[1:0]		L1DSEN	BESLTHRS[3:0]				L1SSEN	REMWAKE	BSEL[3:0]				LPMACK	LPMEN

位	标记	位名	功能	读写
b31~b29	Reserved	-	必须保持复位值。	R/W
b28	ENBSEL	BSEL使能位	BSEL使能位 0:遵循USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007 1：遵循Errata for USB 2.0 ECN: Link Power Management (LPM) - 7/2007	R/W
b27~b25	LPMRCNTSTS[2:0]	LPM重试状态寄存器	LPM重试状态寄存器 记录LPM剩余的重试次数，仅主机模式有效	R
b24	SENDLPM	发送LPM传输	发送LPM传输 0：不发送LPM传输 1：发送LPM传输（EXT 和 LPM 令牌） 当收到设备返回的应答（ACK,STALL或NYET）后该位自动清零。 仅主机模式支持。	R/W
b23~b21	LPMRCNT[2:0]	LPM重试次数寄存器	LPM重试次数寄存器 当收到ERROR应答时，主机发送LPM传输的重试次数,直到收到有效的应答（ACK,STALL或NYET） 仅主机模式支持。	R/W
b20~b17	LPMCHIDX[2:0]	LPM发送通道索引寄存器	LPM发送通道索引寄存器 LPM传输所用主机通道索引，硬件自动插入该通道设定的设备地址以及端点信息。 仅主机模式支持。	R/W
b16	L1RSMOK	L1状态位	L1状态位	R/W

			0：当前状态不能从L1恢复 1：当前状态可以从L1恢复	
b15	SLPSTS	sleep状态位	sleep状态位 设备模式 当设备发送ACK响应LPM传输并经过协议规定的时间后，该位置1，表示进入SLEEP模式。 当总线状态变化，或者断连，或者发送远程唤醒信号时，该位自动清0，退出SLEEP模式。 主机模式 主机发送LPM传输，接收到设备响应的ACK应答后，该位置1，表示进入SLEEP模式。 当主机接收到远程唤醒信号，或者主机发起唤醒，或者主机发起复位时，该位自动清0，退出SLEEP模式。 0：Not in L1 1：In L1	R/W
b14~b13	LPMRSP[1:0]	LPM应答	LPM应答 主机接收或者从机发送的LPM应答 00b：ERROR(无应答) 01b：STALL 00b：NYET 00b：ACK	R/W
b12	L1DSEN	L1 deep sleep使能位	L1 deep sleep使能位 最大限度的降低功耗，应用程序需要将此位设定为1。	R/W
b11~b8	BESLTHRS[3:0]	BSEL阈值设定寄存器	BSEL阈值设定寄存器 设备模式 当设备接收到BSEL值大于等于该寄存器设定时，进入L1。 主机模式 主机发送resume信号的时间。	R/W
b7	L1SSEN	L1 shallow sleep使能位	L1 shallow sleep使能位 最大限度的降低功耗，应用程序需要将此位设定为1。	R/W
b6	REMWAKE	bRemoteWake值	bRemoteWake值 主机发送或者设备接收到的bRemoteWake值	R/W
b5~b2	BSEL[3:0]	BSEL寄存器	BSEL寄存器 设备模式 BSEL bmAttribute值，有效应答LPM传输后自动更新。 主机模式 主机发送LPM传输的BSEL值，或者resume信号长度值。 0000b：125us	R/W

			0001b：150us 0010b：200us 0100b：400us 0101b：500us 0110b：1000us 0111b：2000us 1000b：3000us 1001b：4000us 1010b：5000us 1011b：6000us 1100b：7000us 1101b：8000us 1110b：9000us 1111b：10000us	
b1	LPMACK	LPM应答寄存器	LPM应答寄存器 1：ACK 即使该位写1，但LPM传输有错误时，不发送ACK应答。 存在PID/CRC错误，ERROR应答 bLinkState不等于0001b，STALL应答 存在挂起的数据传输，NYET应答 0：NYET 仅设备模式有效	R/W
b0	LPMEN	LPM使能位	LPM使能位 0：LPM禁止 1：LPM使能	R/W

34.7.2.13 USBHS 主机周期性发送 FIFO 大小寄存器(USBHS_HPTXFSIZ)

USBHS Host periodic transmit FIFO size register

偏移地址：0x100

复位值：0x08001000

此应用程序可以对必须分配给周期 TxFIFO 的 RAM 大小进行编程。

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

位	标记	位名	功能	读写
b31~b16	PTXFD	主机周期性TxFIFO深度	主机周期性TxFIFO深度 (Host periodic TxFIFO depth) 以32位字为单位。 最小值为16	R/W
b15~b0	PTXSA	主机周期性TxFIFO起始地址	主机周期性TxFIFO起始地址(Host periodic TxFIFO start address) 上电复位值是最大RxFIFO深度与最大非周期性TxFIFO深度之和。	R/W

34.7.2.14 USBHS 设备 IN 端点发送 FIFO 大小寄存器 (USBHS_DIEPTXFx)（x = 1..15）

USBHS device IN endpoint transmit FIFO size register

偏移地址：0x104+(x-1)*0x4

复位值：0x08001000+(x-1)*0x800

此应用程序可以对必须分配给设备 TxFIFO 的大小进行编程。

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

位	标记	位名	功能	读写
b31~b16	INEPTXFD	IN端点TxFIFO深度	设备IN端点TxFIFO深度 (Device IN endpoint TxFIFO depth) 以32位字为单位。 最小值为16	R/W
b15~b0	INEPTXSA	IN端点TxFIFOx RAM起始地址	IN端点TxFIFOx RAM起始地址 (IN endpoint FIFOx transmit RAM start address) 此字段包含IN端点发送FIFOx的存储器起始地址。 该地址必须与32位存储器位置对齐。	R/W

34.7.3 USBHS 主机模式寄存器

主机模式寄存器会影响主机模式下的模块操作。在设备模式下不得访问主机模式寄存器，因为产生的结果不明确。

除非特别说明，否则寄存器描述中的位值以二进制表示。

34.7.3.1 USBHS 主机配置寄存器 (USBHS_HCFG)

USBHS Host configuration register

偏移地址：0x400

复位值：0x00000200

此寄存器将在上电后对模块进行配置。请勿在初始化主机后更改此寄存器。

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

位	标记	位名	功能	读写
b31~b3	Reserved	-	必须保持复位值。	R/W
b2	FSLSS	仅支持FS和LS	应用程序使用此位控制模块的枚举速度。使用此位，应用程序可使模块工作为FS主机，即使所连接的设备支持HS通信也是如此。请勿在初始编程后更改此字段。 0：HS/FS/LS，取决于所连接设备支持的最大速度 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

34.7.3.2 USBHS 主机帧时间间隔寄存器 (USBHS_HFIR)

USBHS Host frame interval register

偏移地址：0x404

复位值：0x0000EA60

此寄存器用于存储 USBHS 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/W
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（USBHS_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（USBHS_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

34.7.3.3 USBHS 主机帧编号/帧剩余时间寄存器 (USBHS_HFNUM)

USBHS 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

34.7.3.4 USBHS 主机周期性发送 FIFO/队列状态寄存器(USBHS_HPTXSTS)

USBHS Host periodic transmit FIFO/queue status register

偏移地址：0x410

复位值：0x00080800

此只读寄存器包含周期性 TxFIFO 和周期性发送请求队列的空闲空间信息。

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

位	标记	位名	功能	读写
b31~b24	PTXQTOP	周期性发送请求队列顶部	周期性发送请求队列顶部 (Top of the periodic transmit request queue) 指示周期性Tx请求队列中MAC当前正在处理的项。 该寄存器用于调试。 位 31：奇数/偶数帧 (Odd/Even frame) — 0：以偶数帧发送 — 1：以奇数帧发送 位 30:27：通道/端点编号 (Channel number) 位 26:25：类型 (Type) — 00：输入/输出 — 01：零长度数据包 — 11：禁止通道命令 位 24：结束（所选通道的最后一个条目）(Terminate (last entry for the selected channel))	R
b23~b16	PTXQSAV	周期性发送请求队列可用空间	周期性发送请求队列可用空间 (Periodic transmit request queue space available) 指示可供写入的周期性发送请求队列的空闲位置的数量。该队列既包含IN请求，又包含OUT请求。 0：周期性发送请求队列已满 1：1个位置可用 2：2个位置可用 n：n个位置可用 (其中，n范围：0~8) 其它值：保留	R

b15~b0	PTXFSAVL	周期性发送数据FIFO可用空间	周期性发送数据FIFO可用空间 (Periodic transmit data FIFO space available) 指示可供写入的周期性 TxFIFO 的空闲位置的数量。 以32位字为单位 0：周期性TxFIFO已满 1：1个字可用 2：2个字可用 n：n个字可用（其中，n范围：0~PTXFD） 其它值：保留	R

34.7.3.5 USBHS 主机全体通道中断寄存器 (USBHS_HAINT)

USBHS 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（USBHS_GINTSTS 中的 HCINT 位）中断应用程序。每个通道对应 1 个中断位，最多有 16 个位。

当应用程序通过相应主机通道 x 中断寄存器清零中断时，该寄存器中的位也会清零。

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

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	必须保持复位值。	R/W
b15~b0	HAINT	通道中断	通道中断(Channel interrupt) 每个通道对应一位：通道0对应位0，通道15对应位15。	R/W

34.7.3.6 USBHS 主机全体通道中断屏蔽寄存器(USBHS_HAINTMSK)

USBHS Host all channels interrupt mask register

偏移地址：0x418

复位值：0x0000 0000

主机全体通道中断屏蔽寄存器与主机全体通道中断寄存器结合使用，进而在通道上发生事件时中断应用程序。

每个通道对应 1 个中断屏蔽位，最多有 16 个位。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HAINTM[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	HAINTM	通道中断屏蔽	通道中断屏蔽 (Channel interrupt mask) 0：屏蔽中断 1：使能中断 每个通道对应一位：通道0对应位0，通道15对应位15。	R/W

34.7.3.7 USBHS 主机端口控制和状态寄存器 (USBHS_HPRT)

USBHS 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，USBHS 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（USBHS_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]		PTCTL[3]

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PTCTL[2:0]			PWPR	PLSTS[1:0]		Reserved	PRST	PSUSP	PRES	POCCHNG	POCA	PENCHNG	PENA	PCDET	PCSTS

位	标记	位名	功能	读写
b31~b19	Reserved	-	必须保持复位值。	R/W
b18~b17	PSPD	端口速度	端口速度(Port speed) 指示连接到该端口的设备的速度。 00：高速 01：全速 10：低速 11：保留	R
b16~b13	PTCLT[3:0]	端口测试控制	端口测试控制 (Port test control) 应用程序向该字段写入一个非零值，以将端口置于测试模式，同时端口上会产生对应模式的 信号。 0000：测试模式禁止 0001：Test_J 模式 0010：Test_K 模式 0011：Test_SE0_NAK 模式 0100：Test_Packet 模式 0101：Test_Force_Enable	R/W

			其它值：保留	
b12	PPWR	端口电源	端口电源(Port power) 应用程序使用该字段控制该端口的电源。由于本USBHS内置PHY不具备供电能力，所以此为设置为1时，通过USBHS_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：USBHS_DM 的逻辑电平 位10：USBHS_DP 的逻辑电平	R
b9	Reserved	-	必须保持复位值。	R/W
b8	PRST	端口复位	端口复位(Port reset) 应用程序将该位置1时，会在该端口上启动复位序列。应用程序必须为复位周期定时，并在复位序列完成后将该位清零。 0：端口未处于复位状态 1：端口处于复位状态 应用程序必须将该位置1并最少保持 10 ms，以在端口上启动复位。	R/W
b7	PSUSP	端口挂起	端口挂起(Port suspend) 应用程序将此位置1以将此端口置于挂起模式。只有此位置1时，模块才会停止发送SOF。要停止PHY时钟，应用程序必须将端口时钟停止位置1，这会使能PHY的挂起输入引脚。 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USBHS_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（USBHS_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/W

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（USBHS_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

34.7.3.8 USBHS 主机通道 x 特性寄存器 (USBHS_HCCHARx)（x = 0..15）

USBHS 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]							MC[1:0]		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USBHS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USBHS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(微)帧 1：奇数（微）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	MC	多重计数 (MC)/错误计数(EC)	多重计数 (MC)/错误计数 (EC) (Multi Count (MC)/Error Count (EC)) —当主机通道 x 分离控制寄存器 (USBHS_HCSPLTx) 中的分离使能位 (SPLITEN) 复位 (0) 时， 此字段向主机指示此周期性端点每微帧必须执行的事务数。对于非周期性传输，此字段指定 内部 DMA 引擎更改仲裁之前此通道要获取的数据包数。 00：保留，会产生不明确的结果对该字段的操作 01：1个事务 10：此端点每微帧需要发出2个事务	R/W

			11：此端点每微帧需要发出 3 个事务。 —当 USBHS_HCSPLTx 中的 SPLITEN 位置 (1) 时，此字段指示周期性分离事务发生错误时 要对该事务执行的立即重试次数。此字段至少须置为 01。	
b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 指示选择的传输类型。 00：控制 01：同步 10：批量 11：中断	R/W
b17	LSDEV	低速设备	低速设备 (Low-speed device) 此字段由应用程序置 1，表示此通道正在与一个低速设备进行通信。	R/W
b16	Reserved	-	必须保持复位值。	R/W
b15	EPDIR	端点方向	端点方向(Endpoint direction) 指示通信事务的方向是输入还是输出。 0：输出 1：输入	R/W
b14-b11	EPNUM	端点编号	端点编号 (Endpoint number) 指示要与该主机通道通信的 USB 设备的端点号。	R/W
b10-b0	MPSIZ	最大数据包大小	最大数据包大小 (Maximum packet size) 指示与该主机通道通信的设备端点的最大数据包大小。	R/W

34.7.3.9 USBHS 主机通道 x 分离控制寄存器 (USBHS_HCSPLTx)（x = 0..15）

USBHS host channel-x split control register

偏移地址：0x504 + (通道编号 × 0x20)

复位值：0x0000 0000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XACTPOS[1:0]	HUBADDR[6:0]								PRTADDR[6:0]						

位	标记	位名	功能	读写
b31	SPLITEN	分离使能	分离使能 (Split enable) 应用程序将此位置 1 时，指示允许该通道执行分离通信事务。	R/W
b30~b17	Reserved	-	必须保持复位值。	R/W
b16	COMPLSPLT	执行完全分离	执行完全分离 (Do complete split) 应用程序将此位置1时，可请求主机执行完全分离通信事务。	R/W
b15~b14	XACTPOS	事务位置	事务位置 (Transaction position) 此字段用于决定随各OUT事务发送全部、第一个、中间还是最后一个有效负载。 00：中间。指此事务的中间有效负载（大于188字节） 01：结尾。指此事务的最后一个有效负载（大于188字节）。 10：起始。指此事务的第一个数据有效负载（大于188字节） 11：全部。指此事务的全部数据有效负载（小于或等于188字节）	R/W
b13~b7	HUBADDR	集线器地址	集线器地址 (Hub address) 此字段存储事务转发器的集线器设备地址。	R/W
b6~b0	PRTADDR	端口地址	端口地址 (Port address) 此字段是接收方事务转发器的端口编号。	R/W

34.7.3.10 USBHS 主机通道 x 中断寄存器 (USBHS_HCINTx)（x = 0..15）

USBHS 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（USBHS_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 (USBHS_HAINT) 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 USBHS_HAIN 和 USBHS_GINTSTS 寄存器中的对应位清零。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					DTERR	FRMOR	BBERR	TXERR	NYET	ACK	NAK	STALL	AHBERR	CHH	XFRC

位	标记	位名	功能	读写
b31~b11	Reserved	-	必须保持复位值。	R/W
b10	DTERR	数据切换错误	数据同步错误 (Data toggle error) 应用程序需通过写1清除该位。	R/W
b9	FRMOR	帧溢出错误	帧溢出错误(Frame overrun) 应用程序需通过写1清除该位。	R/W
b8	BBERR	串扰错误	串扰错误(Babble error) 产生串扰事件的典型原因是端点发送了一个数据包，但是数据包长度超过了端点的最大包长。 应用程序需通过写1清除该位。	R/W
b7	TXERR	通信事务错误	通信事务错误 (Transaction error) 指示 USB 上发生下列错误之一： CRC校验失败 超时 位填充错误 错误的EOP 应用程序需通过写1清除该位。	R/W
b6	NYET	收到NYET 响应	收到NYET 响应 (Response received interrupt)	R/W

b5	ACK	收到/发出ACK 响应	收到/发出ACK响应(ACK response received/transmitted interrupt) 应用程序需通过写1清除该位。	R/W
b4	NAK	收到NAK响应	收到NAK响应(NAK response received interrupt) 应用程序需通过写1清除该位。	R/W
b3	STALL	收到STALL响应	收到STALL响应(STALL response received interrupt	R/W
b2	AHBERR	AHB错误	AHB错误 (AHB error) 仅当处于内部DMA模式下且AHB读/写操作期间发生AHB错误时才生成此错误。应用程序可通过读取相应的 DMA 通道地址寄存器来获取错误地址。 应用程序需通过写1清除该位	R/W
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

34.7.3.11 USBHS 主机通道 x 中断屏蔽寄存器 (USBHS_HCINTMSKx)（x = 0..15）

USBHS Host channel-x interrupt mask register

偏移地址：0x50C + (通道编号 × 0x20)

复位值：0x0000 0000

该寄存器用于选择屏蔽主机通道中断。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					DTERRM	FRMORM	BBERRM	TXERRM	NYETM	ACKM	NAKM	STALLM	AHBERRM	CHHM	XFRCM

位	标记	位名	功能	读写
b31~b11	Reserved	-	必须保持复位值。	R/W
b10	DTERRM	数据切换错误中断屏蔽	数据同步错误屏蔽(Data toggle error mask) 0：屏蔽中断 1：使能中断	R/W
b9	FRMORM	帧溢出错误中断屏蔽	帧溢出屏蔽(Frame overrun mask) 0：屏蔽中断 1：使能中断	R/W
b8	BBERRM	串扰错误中断屏蔽	串扰错误屏蔽(Babble error mask) 0：屏蔽中断 1：使能中断	R/W
b7	TXERRM	通信事务错误中断屏蔽	通信事务错误屏蔽(Transaction error mask) 0：屏蔽中断 1：使能中断	R/W
b6	NYETM	NYET响应接收中断屏蔽	NYET 响应接收中断屏蔽 (response received interrupt mask) 0：屏蔽中断 1：使能中断	R/W
b5	ACKM	收到/发出ACK 响应中断屏蔽	ACK响应接收/发送中断屏蔽 (ACK response received/transmitted interrupt mask) 0：屏蔽中断 1：使能中断	R/W
b4	NAKM	收到NAK响应中断屏蔽	NAK响应接收中断屏蔽(NAK response received interrupt mask) 0：屏蔽中断 1：使能中断	R/W

b3	STALLM	收到STALL响应中断屏蔽	收到STALL响应(STALL response received interrupt mask) 0：屏蔽中断 1：使能中断	R/W
b2	AHBERRM	AHBERR中断屏蔽	AHBERR中断屏蔽 (AHB error mask) 0：屏蔽中断 1：使能中断	R/W
b1	CHHM	通道停止中断屏蔽	通道停止中断屏蔽 (Channel halted mask) 0：屏蔽中断 1：使能中断	R/W
b0	XFRCM	传输完成中断屏蔽	传输完成中断屏蔽 (Transfer completed mask) 0：屏蔽中断 1：使能中断	R/W

34.7.3.12 USBHS 主机通道 x 传输大小寄存器 (USBHS_ HCTSIZx)（x = 0..15）

USBHS 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
DOPING	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	DOPING	执行PING操作	执行PING操作 (Do ping)，此位仅用于OUT传输。 将此位置1可指示主机执行PING协议。 注意： 请勿针对IN传输将此位置1。如果针对IN传输将此位置1，将禁止通道。	R/W
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：DATA2 10：DATA1 11：MDATA/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

34.7.3.13 USBHS 主机通道 xDMA 地址寄存器 (USBHS_HCDMAx)（x = 0..15）

USBHS Host channel-x DMA address register

偏移地址：0x514 + (通道编号 × 0x20)

复位值：0xXXXX XXXX

该寄存器用于设定主机 DMA 模式时 DMA 地址。

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

位	标记	位名	功能	读写
b31~b0	DMAADDR	DMA地址	DMA地址(DMA address) 此字段存储主机从设备端点获取数据或往设备端点发送数据所用DMA传输的存储器的地址。每次AHB传输结束，该寄存器都会递增。	R/W

34.7.4 USBHS 设备模式寄存器

设备模式寄存器会影响设备模式下的模块操作。在主机模式下不得访问设备模式寄存器，因为产生的结果不明确。

除非特别说明，否则寄存器描述中的位值以二进制表示。

34.7.4.1 USBHS 设备配置寄存器(USBHS_DCFG)

USBHS Device configuration register

偏移地址：0x800

复位值：0x0822 0000

此寄存器在上电、执行某些控制命令或枚举后，会将模块配置为设备模式。请勿在初始编程后更改该寄存器。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved			PFIVL[1:0]		DAD[6:0]							Reserved	NZLSOHSK	DSPD[1:0]	

位	标记	位名	功能	读写
b31~b13	Reserved	-	必须保持复位值。	R/W
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%（微）帧间隔 01：85%（微）帧间隔 10：90%（微）帧间隔 11：95%（微）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/W

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：高速 01：保留 10：保留 11：全速（USB 1.1收发器时钟为48 MHz）	R/W

34.7.4.2 USBHS 设备控制寄存器(USBHS_DCTL)

USBHS Device control register

偏移地址：0x804

复位值：0x0000 0002

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				POPRGDNE	CGONAK	SGONAK	CGINAK	SGINAK	TCTL[2:0]			GONSTS	GINSTS	SDIS	RWUSIG

位	标记	位名	功能	读写
b31~b12	Reserved	-	必须保持复位值。	R/W
b11	POPRGDNE	上电编程完成	上电编程完成(Power-on programming done) 应用程序使用此位指示寄存器从掉电模式唤醒后已完成编程。	R/W
b10	CGONAK	清零全局OUT NAK	清零全局OUT NAK(Clear global OUT NAK) 对此位执行写操作会将全局OUT NAK清零。	R/W
b9	SGONAK	置位全局OUT NAK	置位全局OUT NAK(Set global OUT NAK) 对此位执行写操作会将全局OUT NAK置 1。 应用程序使用此位在所有OUT端点发送NAK握手信号。 应用程序只有确定模块中断寄存器中全局OUT NAK有效位 （USBHS_GINTSTS中GONAKEFF位）已清零时，才可以将此位置 1。	R/W
b8	CGINAK	清零全局IN NAK	清零全局IN NAK (Clear global IN NAK) 对此位执行写操作会将全局 IN NAK 清零。	R/W
b7	SGINAK	置位全局IN NAK	置位全局IN NAK (Set global IN NAK) 对此字段执行写操作会将全局非周期性IN NAK置 1。应用程序使用此位使所有非周期性IN端点发送NAK握手信号。 应用程序只有确定模块中断寄存器中全局IN NAK有效位（USBHS_GINTSTS中GINAKEFF位）已清零时，才可以将此位置1。	R/W
b6~b4	TCTL	测试控制	测试控制 (Test control) 000：测试模式禁止 001：Test_J 模式 010：Test_K 模式 011：Test_SE0_NAK 模式 100：Test_Packet 模式 101：Test_Force_Enable 其它值：保留	R/W

b3	GONSTS	全局OUT NAK状态	全局OUT NAK状态(Global OUT NAK status) 0：将根据FIFO状态和NAK和STALL位设置发送握手信号。 1：无论RxFIFO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USBHS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。 全速时，软断连的最小时间规定如下： 挂起状态：最小时间为1ms+2.5us 空闲状态：2.5us 非空闲或挂起状态：2.5us	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

34.7.4.3 USBHS 设备状态寄存器(USBHS_DSTS)

USBHS 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(USBHS_DAINT) 寄存器读取发生中断的端点信息。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								LNSTS[1:0]		FNSOF[13:8]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FNSOF[7:0]								Reserved				EERR	ENUMSPD[1:0]		SUSPSTS

位	标记	位名	功能	读写
b31~b24	Reserved	-	必须保持复位值。	R/W
b23~b22	LNSTS	USB总线状态	USB总线状态 LNSTS[1]：D+的逻辑电平 LNSTS[0]：D-的逻辑电平	R
b21~b8	FNSOF	接收SOF的帧编号	接收SOF的帧编号 (Frame number of the received SOF)	R
b7~b4	Reserved	-	必须保持复位值。	R/W
b3	EERR	不定错误	不定错误(Erratic error) 模块将该位置1以报告任何不定错误。 由于不定错误，USBHS控制器会进入挂起状态，并且会USBHS_GINTSTS寄存器的早期挂起位（USBHS_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USBHS控制器通过chirp序列检测速度后被枚举成的速度。 00：高速 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USBHS_DCTL寄存器的远程唤醒信号位（USBHS_DCTL中的RWUSIG位）执行写操作。	R

34.7.4.4 USBHS 设备 IN 端点通用中断屏蔽寄存器(USBHS_DIEPMSK)

USBHS 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 USBHS_DIEPINTx 寄存器配合使用，以便在每个 IN 端点上生成中断。通过对此寄存器的相应位执行写操作，可屏蔽 USBHS_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		NAKM	Reserved				TXFURM	Reserved	INEPNEM	INEPNMM	TTXFEMSK	TOM	Reserved	EPDM	XFRCM

位	标记	位名	功能	读写
b31~b14	Reserved	-	必须保持复位值。	R/W
b13	NAKM	NAK中断屏蔽	NAK中断屏蔽（NAK mask） 0：屏蔽中断 1：使能中断	R/W
b12~b9	Reserved	-	必须保持复位值。	R/W
b8	TXFURM	发送FIFO下溢中断屏蔽	发送FIFO下溢中断屏蔽 (FIFO underrun mask) 0：屏蔽中断 1：使能中断	R/W
b7	Reserved	-	必须保持复位值。	R/W
b6	INEPNEM	IN端点NAK有效中断屏蔽	IN端点NAK有效中断屏蔽 (IN endpoint NAK effective mask) 0：屏蔽中断 1：使能中断	R/W
b5	INEPNMM	EP不匹配时接收到IN令牌中断屏蔽	EP不匹配时接收到IN令牌中断屏蔽 (IN token received with EP mismatch mask) 0：屏蔽中断 1：使能中断	R/W
b4	TTXFEMSK	TxFIFO为空时接收到IN令牌中断屏蔽	TxFIFO为空时接收到IN令牌中断屏蔽 (IN token received when TxFIFO empty mask) 0：屏蔽中断 1：使能中断	R/W

b3	TOM	超时中断屏蔽（非同步端点）	超时中断屏蔽（非同步端点） (Timeout condition mask (Non-isochronous endpoints)) 0：屏蔽中断 1：使能中断	R/W
b2	Reserved	-	必须保持复位值。	R/W
b1	EPDM	端点禁止中断屏蔽	端点禁止中断屏蔽(Endpoint disabled interrupt mask) 0：屏蔽中断 1：使能中断	R/W
b0	XFRCM	传输完成中断屏蔽	传输完成中断屏蔽(Transfer completed interrupt mask) 0：屏蔽中断 1：使能中断	R/W

34.7.4.5 USBHS 设备 OUT 端点通用中断屏蔽寄存器(USBHS_DOEPMSK)

USBHS 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 USBHS_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 USBHS_DOEPINTx 寄存器中的 OUT 端点中断。默认情况下，状态中断都被屏蔽。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved	NYETM	Reserved					OPEM	Reserved	B2BSTUP	Reserved	OTEPDM	STUPM	Reserved	EPDM	XFRCM

位	标记	位名	功能	读写
b31~b15	Reserved	-	必须保持复位值。	R/W
b14	NYETM	NYET中断屏蔽	NYET中断屏蔽（NYET mask） 0：屏蔽中断 1：使能中断	R/W
b13~b9	Reserved	-	必须保持复位值。	R/W
b8	OPEM	OUT数据包错误中断屏蔽	OUT数据包错误中断屏蔽 (OUT packet error mask) 0：屏蔽中断 1：使能中断	R/W
b7	Reserved	-	必须保持复位值。	R/W
b6	B2BSTUP	接收到连续的SETUP数据包中断屏蔽	接收到连续的SETUP数据包中断屏蔽 (Back-to-back SETUP packets received mask) 仅适用于控制OUT端点。 0：屏蔽中断 1：使能中断	R/W
b5	Reserved	-	必须保持复位值。	R/W
b4	OTEPDM	端点禁止时接收到OUT令牌中断屏蔽	端点禁止时接收到OUT令牌中断屏蔽 (OUT token received when endpoint disabled mask) 仅适用于控制 OUT 端点。 0：屏蔽中断 1：使能中断	R/W

b3	STUPM	SETUP阶段完成中断屏蔽	SETUP阶段完成中断屏蔽(SETUP phase done mask) 仅适用于控制端点。 0：屏蔽中断 1：使能中断	R/W
b2	Reserved	-	必须保持复位值。	R/W
b1	EPDM	端点禁止中断屏蔽	端点禁止中断屏蔽(Endpoint disabled interrupt mask) 0：屏蔽中断 1：使能中断	R/W
b0	XFRCM	传输完成中断屏蔽	传输完成中断屏蔽(Transfer completed interrupt mask) 0：屏蔽中断 1：使能中断	R/W

34.7.4.6 USBHS 设备全体端点中断寄存器 (USBHS_DAINT)

USBHS 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，USBHS_DAINT 寄存器将通过 USBHS_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 USBHS_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(USBHS_DIEPINTx/USBHS_DOEPINTx)中的位置 1 和清零时，此寄存器中的相应位也将置 1 和清零。

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

位	标记	位名	功能	读写
b31~b16	OEPINT	OUT端点中断位	OUT端点中断位(OUT endpoint interrupt bits) 每个OUT端点对应一位： OUT端点0对应位16，而OUT端点15对应位31。	R/W
b15~b0	IEPINT	IN端点中断位	IN端点中断位(IN endpoint interrupt bits) 每个IN端点对应一位： IN端点0对应位0，而IN端点15对应位15。	R/W

34.7.4.7 USBHS 设备全体端点中断屏蔽寄存器 (USBHS_DAINTMSK)

USBHS Device all endpoints interrupt mask register

偏移地址：0x81C

复位值：0x0000 0000

USBHS_DAINTMSK 寄存器与设备端点中断寄存器结合使用，在设备端点上发生事件时中断应用程序。但是，与该中断相对应的 USBHS_DAINT 寄存器位仍会置 1。

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

位	标记	位名	功能	读写
b31~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15对应位31。 0：屏蔽中断 1：使能中断	R/W
b15~b0	IEPINTM	IN端点中断屏蔽位	IN端点中断屏蔽位(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

34.7.4.8 USBHS 设备阈值控制寄存器(USBHS_DTHRCTL)

USBHS Device threshold control register

偏移地址：0x0830

复位值：0x0C10 0020

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved				ARPEN	Reserved	RXTHRLEN[8:0]									RXTHREN

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					TXTHRLEN[7:0]									ISOTHREN	NONISOTHREN

位	标记	位名	功能	读写
b31~b28	Reserved	-	必须保持复位值。	R/W
b27	ARPEN	仲裁器驻留使能	仲裁器驻留使能 (Arbiter parking enable) 该位用于控制内部DMA仲裁对IN端点的驻留。当阈值使能且该位置 1 后，DMA会对收到 令牌的IN端点驻留其仲裁。采用这种方式可以避免出现下溢情况。默认情况下使能驻留。	R/W
b26	Reserved	-	必须保持复位值。	R/W
b25~b17	RXTHRLEN	接收阈值长度	接收阈值长度 (Receive threshold length) 该字段以双字为单位指定接收阈值的大小。该字段指定模块在AHB上启动传输之前在USB上接收的数据量。阈值长度最小值为八个双字。推荐RXTHRLEN和设定的AHB突发传输长度相同。	R/W
b16	RXTHREN	接收阈值使能	接收阈值使能 (Receive threshold enable) 该位置1时，模块会使能接收方向的阈值。	R/W
b15~b11	Reserved	-	必须保持复位值。	R/W
b10~b2	TXTHRLEN	发送阈值长度	发送阈值长度 (Transmit threshold length) 该字段以双字为单位指定发送阈值的大小。该字段指定模块在USB上启动传输之前相应端点发送FIFO中的数据量。阈值长度最小值为八个双字。该字段控制同步和非同步IN端点阈值。推荐TXTHRLEN和设定的AHB突发传输长度相同。	R/W
b1	ISOTHREN	ISO IN 端点阈值使能	ISO IN端点阈值使能 (ISO IN endpoint threshold enable) 该位置1时，模块会使能同步IN端点的阈值。	R/W
b0	NONISOTHREN	非同步IN端点阈值使能	非同步IN端点阈值使能 (Nonisochronous IN endpoints enable) 该位置1时，模块会使能非同步IN端点的阈值。	R/W

34.7.4.9 USBHS 设备 IN 端点 FIFO 空中断屏蔽寄存器(USBHS_DIEPEMPMSK)

USBHS Device IN endpoint FIFO empty interrupt mask register

偏移地址：0x834

复位值：0x0000 0000

此寄存器用于控制 IN 端点 FIFO 空中断的生成。

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	必须保持复位值。	R/W
b15~b0	INEPTXFEM	IN EP TxFIFO空中断屏蔽位	IN EP TxFIFO空中断屏蔽位 (IN EP Tx FIFO empty interrupt mask bits) 这些位用作USBHS_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

34.7.4.10 USBHS 设备单个端点中断寄存器(USBHS_DEACHINT)

USBHS Device each endpoint interrupt register

偏移地址：0x0838

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved														OEP1INT	Reserved

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved														IEP1INT	Reserved

位	标记	位名	功能	读写
b31~18	Reserved	-	必须保持复位值。	R/W
b17	OEP1INT	OUT端点1中断位	OUT 端点1中断位 (OUT endpoint 1 interrupt bit)	R/W
b16~b2	Reserved	-	必须保持复位值。	R/W
b1	IEP1INT	IN端点1中断位	IN 端点1中断位 (IN endpoint 1interrupt bit)	R/W
b0	Reserved	-	必须保持复位值。	R/W

34.7.4.11 USBHS 设备单个端点中断屏蔽寄存器(USBHS_DEACHINTMASK)

USBHS Device each endpoint interrupt mask register

偏移地址：0x083C

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved														OEP1INTM	Reserved

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved														IEP1INTM	Reserved

位	标记	位名	功能	读写
b31~18	Reserved	-	必须保持复位值。	R/W
b17	OEP1INTM	OUT端点1中断屏蔽位	OUT 端点1中断屏蔽位 (OUT endpoint 1 interrupt mask bit) 0：屏蔽中断 1：使能中断	R/W
b16~b2	Reserved	-	必须保持复位值。	R/W
b1	IEP1INTM	IN端点1中断屏蔽位	IN 端点1中断屏蔽位 (IN endpoint 1interrupt mask bit) 0：屏蔽中断 1：使能中断	R/W
b0	Reserved	-	必须保持复位值。	R/W

34.7.4.12 USBHS 设备 IN 端 1 中断屏蔽寄存器 (USBHS_DIEPEACHMSK1)

USBHS device each in endpoint-1 interrupt register

偏移地址：0x844

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		NAKM	Reserved				TXFURM	Reserved	INEPNEM	INEPNMM	TTXFEMSK	TOM	Reserved	EPDM	XFRCM

位	标记	位名	功能	读写
b31~14	Reserved	-	必须保持复位值。	R/W
b13	NAKM	NAK中断屏蔽	NAK中断屏蔽 (NAK interrupt mask) 0：屏蔽中断 1：使能中断	R/W
b12~b9	Reserved	-	必须保持复位值。	R/W
b8	TXFURM	发送FIFO下溢中断屏蔽	发送FIFO下溢中断屏蔽 (FIFO underrun mask) 0：屏蔽中断 1：使能中断	R/W
b7	Reserved	-	必须保持复位值。	R/W
b6	INEPNEM	IN端点NAK有效中断屏蔽	IN端点NAK有效中断屏蔽 (IN endpoint NAK effective mask) 0：屏蔽中断 1：使能中断	R/W
b5	INEPNMM	EP不匹配时接收到IN令牌中断屏蔽	EP不匹配时接收到IN令牌中断屏蔽 (IN token received with EP mismatch mask) 0:屏蔽中断 1：使能中断	R/W
b4	TTXFEMSK	TxFIFO为空时接收到IN令牌中断屏蔽	TxFIFO为空时接收到IN令牌中断屏蔽(IN token received when TxFIFO empty mask) 0：屏蔽中断 1：使能中断	R/W
b3	TOM	超时中断屏蔽	超时中断屏蔽（非同步端点）(Timeout condition mask (nonisochronous endpoints)) 0：屏蔽中断 1：使能中断	R/W
b2	AHBERRM	AHBERR中断屏蔽	AHBERR中断屏蔽 (AHB error mask) 0：屏蔽中断 1：使能中断	R/W

b1	EPDM	端点禁止中断屏蔽	端点禁止中断屏蔽 (Endpoint disabled interrupt mask) 0：屏蔽中断 1：使能中断	R/W
b0	XFRCM	传输完成中断屏蔽	传输完成中断屏蔽 (Transfer completed interrupt mask) 0：屏蔽中断 1：使能中断	R/W

34.7.4.13 USBHS 设备 OUT 端点 1 中断屏蔽寄存器 (USBHS_DOEPEACHMSK1)

USBHS device each out endpoint-1 interrupt register

偏移地址：0x884

复位值：0x0000 0000

31	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	NYETM	Reserved					OPEM	Reserved	B2BSTUP	Reserved	OTEPDM	STUPM	Reserved	EPDM	XFRCM

位	标记	位名	功能	读写
b31~b15	Reserved	-	必须保持复位值。	R/W
b14	NYETM	NYET中断屏蔽	NYET中断屏蔽（NYET mask） 0：屏蔽中断 1：使能中断	R/W
b13~b9	Reserved	-	必须保持复位值。	R/W
b8	OPEM	OUT数据包错误中断屏蔽	OUT数据包错误中断屏蔽 (OUT packet error mask) 0：屏蔽中断 1：使能中断	R/W
b7	Reserved	-	必须保持复位值。	R/W
b6	B2BSTUP	接收到连续的SETUP数据包中断屏蔽	接收到连续的SETUP数据包中断屏蔽 (Back-to-back SETUP packets received mask) 仅适用于控制OUT端点。 0：屏蔽中断 1：使能中断	R/W
b5	Reserved	-	必须保持复位值。	R/W
b4	OTEPDM	端点禁止时接收到OUT令牌中断屏蔽	端点禁止时接收到OUT令牌中断屏蔽 (OUT token received when endpoint disabled mask) 仅适用于控制 OUT 端点。 0：屏蔽中断 1：使能中断	R/W
b3	STUPM	SETUP阶段完成中断屏蔽	SETUP阶段完成中断屏蔽(SETUP phase done mask) 仅适用于控制端点。 0：屏蔽中断 1：使能中断	R/W
b2	Reserved	-	必须保持复位值。	R/W

b1	EPDM	端点禁止中断屏蔽	端点禁止中断屏蔽(Endpoint disabled interrupt mask) 0：屏蔽中断 1：使能中断	R/W
b0	XFRCM	传输完成中断屏蔽	传输完成中断屏蔽(Transfer completed interrupt mask) 0：屏蔽中断 1：使能中断	R/W

34.7.4.14 USBHS 设备控制 IN 端点 0 控制寄存器(USBHS_DIEPCTL0)

USBHS 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	TXFNUM[3:0]				STALL	Reserved	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上启动数据发送。 在此端点上触发以下任一中断之前，模块会将此位清零： — SETUP阶段完成 — 端点禁止 — 传输完成	R/W
b30	EPDIS	端点禁止	端点禁止(Endpoint disable) 即使在该端点上的传输完成之前，应用程序也可将此位置1，以停止端点上的数据发送。应用程序必须等到发生端点禁止中断后，才能将端点视为禁止端点。在端点禁止中断位置1前，模块会将此位清零。只有在该端点的端点使能位置1后，应用程序才可将该位置 1。	R/W
b29~b28	Reserved	-	必须保持复位值。	R/W
b27	SNAK	置位NAK位	置位NAK位(Set NAK) 对此位进行写操作会将端点的NAK位置1。 通过此位，应用程序可以控制端点上NAK握手信号的发送。模块也可在端点接收到SETUP数据包后将该端点的此位置1。	R/W
b26	CNAK	清零NAK位	清零NAK位(Clear NAK) 对此位进行写操作会将端点的NAK位清零。	R/W
b25~b22	TXFNUM	TxFIFO编号	TxFIFO编号(TxFIFO number) 该值设置为分配给IN端点0的FIFO编号。只能使用TX-FIFO0。	R/W

b21	STALL	STALL握手	STALL握手(STALL handshake) 应用程序只能将此位置1，端点接收到SETUP令牌时，模块会将此位清零。如果NAK位、全局IN NAK或全局OUT NAK与此位均置 1，则STALL位优先。	R/W
b20	Reserved	-	必须保持复位值。	R/W
b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 硬件设置为‘00’，表示控制类型的端点。	R
b17	NAKSTS	NAK状态	NAK状态(NAK status) 指示以下结果： 0：模块根据FIFO状态回复非NAK握手。 1：模块在此端点上回复NAK握手。 当此位置1时（无论是被应用程序还是被模块），即使TxFIFO中仍有数据可用，模块也会停止发送数据。无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R
b16	Reserved	-	必须保持复位值。	R/W
b15	USBAEP	USB活动端点	USB活动端点(USB active endpoint) 此位总是置1，指示在所有配置和接口中控制端点0始终处于激活状态。	R
b14~b2	Reserved	-	必须保持复位值。	R/W
b1~b0	MPSIZ	最大数据包大小	最大数据包大小(Maximum packet size) 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

34.7.4.15 USBHS 设备 IN 端点 x 控制寄存器(USBHS_DIEPCTLx)(x=1..15)

USBHS Device IN endpoint x control register

偏移地址：0x900 + (端点编号 × 0x20)

复位值：0x0000 0000

应用程序使用此寄存器控制各个逻辑端点（端点 0 除外）的行为。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
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： — 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/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
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 DPID：端点数据PID (Endpoint data PID) 仅适用于中断/批量IN端点。 包含此端点上将要接收或发送的数据包的PID。端点激活后，应用程序必须对要在此端点上接收或发送的首个数据包的PID进行编程。应用程序使用SD0PID寄存器字段对DATA0或DATA1 PID进行编程。 0：DATA0 1：DATA1	R/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

34.7.4.16 USBHS 设备 IN 端点 x 中断寄存器 (USBHS_DIEPINTx) (x=0..15)

USBHS Device IN endpoint x interrupt register

偏移地址：0x908 + (端点编号 × 0x20)

复位值：0x0000 0080

此寄存器指示端点在出现 USB 和 AHB 相关事件时的状态。当模块中断寄存器中的 IN 端点中断位（USBHS_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(USBHS_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 USBHS_DAINT 和 USBHS_GINTSTS 寄存器中的对应位清零。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved		NAK	BERR	PKTDRPSTS	Reserved	Reserved	TXFIFOUDRN	TXFE	INEPNE	Reserved	TTXFE	TOC	Reserved	EPDISD	XFRC

位	标记	位名	功能	读写
b31~b14	Reserved	-	必须保持复位值。	R/W
b13	NAK	NAK中断	NAK中断 (NAK interrupt) 当设备发出或收到NAK时，模块将生成该中断。 如果是同步IN端点，由于Tx FIFO中无数据 可发而发送长度为零的数据包时也会生成该中断。 通过软件写1清零。	R/W
b12	BERR	Babble错误中断	Babble错误中断 (Babble error interrupt) 通过软件写1清零。	R/W
b11	PKTDRPSTS	数据包丢弃状态	数据包丢弃状态 (Packet dropped status) 该位用于向应用程序指示有 ISOC OUT 数据包被丢弃。该位没有相应的中断屏蔽位，也不会生成中断。 通过软件写1清零。	R/W
b10~b9	Reserved	-	必须保持复位值。	R/W
b8	TXFIFOUDRN	发送FIFO下溢	发送FIFO下溢 (TxfifoUndrn) 当模块检测到该端点发送FIFO下溢时，将生成该中断。 相关性：该中断仅在使能了阈值时有效。 通过软件写1清零。	R/W

b7	TXFE	发送FIFO为空	发送FIFO为空 (Transmit FIFO empty) 当此端点的TxFIFO为半空或全空时，此中断被置位。TxFIFO为半空还是全空状态由USBHS_GAHBCFG寄存器中的TxFIFO空白别位（USBHS_GAHBCFG中的TXFELVL位）决定。 通过软件写1清零。	R/W
b6	INEPNE	IN端点NAK有效	INEPNE：IN端点NAK有效 (IN endpoint NAK effective) 当应用程序通过向USBHS_DIEPCTLx 中的CNAK位写入数据来将IN端点NAK清零时，此位可被清零。 该中断指示模块已对（由应用程序或模块）置1的NAK采样，结果已生效。该中断指示由应用程序置1的IN端点NAK位已在模块中起作用。 此中断不保证在USB上发送了NAK握手信号。 STALL位的优先级高于 NAK 位。 软件写1也可将此位清零。	R/W
b5	Reserved	-	必须保持复位值。	R/W
b4	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/W
b1	EPDISD	端点禁止中断	端点禁止中断 (Endpoint disabled interrupt) 此位指示该端点已经由应用程序禁止掉。 通过软件写1清零。	R/W
b0	XFRC	传输完成中断	传输完成中断 (Transfer completed interrupt) 此字段指示在此端点上设置的传输已经在USB和AHB上传输完成。 通过软件写1清零。	R/W

34.7.4.17 USBHS 设备 IN 端点 0 传输大小寄存器(USBHS_DIEPTSIZ0)

USBHS 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（USBHS_DIEPCTL0 中的 EPENA）使能端点 0 后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

非零端点使用端点 1~15 的寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved											PKTCNT[1:0]		Reserved		

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved									XFRSIZ[6:0]						

位	标记	位名	功能	读写
b31~b21	Reserved	-	必须保持复位值。	R/W
b20~b19	PKTCNT	数据包计数	数据包计数 (Packet count) 指示端点0的一次数据传输包含的数据包个数。 每次从TxFIFO读取数据包（最大大小或短数据包）时，此字段将递减。	R/W
b18~b7	Reserved	-	必须保持复位值。	R/W
b6~b0	XFRSIZ	传输大小	传输大小(Transfer size) 指示端点0的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。 每次向TxFIFO写入来自外部存储器的数据包时，模块会使此字段递减。	R/W

34.7.4.18 USBHS 设备 IN 端点 x 传输大小寄存器(USBHS_DIEPTSIZx)(x=1..15)

USBHS 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 USBHS_DIEPCTLx 寄存器中的端点使能位（USBHS_DIEPCTLx 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	MCNT[1:0]		PKTCNT[9:0]										XFRSIZ[18:16]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XFRSIZ[15:0]															

位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R/W
b30~b29	MCNT	多重计数	多重计数 (Multi count) 对于周期性IN端点，此字段指示在USB上每帧必须发送的数据包数。模块使用此字段计算同步IN端点的数据PID。 01：1个数据包 10：2个数据包 11：3 个数据包	R/W
b28~b19	PKTCNT	数据包计数	数据包计数 (Packet count) 指示该端点上的一次数据传输包含的数据包个数。 每次从TxFIFO读取数据包（最大大小或短数据包）时，此字段将递减。	R/W
b18~b0	XFRSIZ	传输大小	传输大小(Transfer size) 此字段包含当前端点的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。 每次向TxFIFO写入来自外部存储器的数据包时，模块会使此字段递减。	R/W

34.7.4.19 USBHS 设备 IN 端点 x DMA 地址寄存器（USBHS_DIEPDMAx)(x=0..15)

USBHS 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

34.7.4.20 USBHS 设备 IN 端点发送 FIFO 状态寄存器（USBHS_ DTXFSTSx)(x=0..15)

USBHS Device IN endpoint transmit FIFO status register

偏移地址：0x918 + (端点编号 × 0x20)

复位值：0x0000 0800

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/W
b15~b0	INEPTFSAV	IN端点TxFIFO可用空间	IN端点TxFIFO可用空间 (IN endpoint TxFIFO space available) 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	R

34.7.4.21 USBHS 设备控制 OUT 端点 0 控制寄存器(USBHS_DOEPCTL0)

USBHS 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/W
b27	SNAK	置位NAK位	置位NAK位(Set NAK) 对此位进行写操作会将端点的NAK位置1。 通过此位，应用程序可以控制端点上NAK握手信号的发送。模块也可在端点接收到SETUP数据包后将该端点的此位置1。	R/W
b26	CNAK	清零NAK位	清零NAK位(Clear NAK) 对此位进行写操作会将端点的NAK位清零。	R/W
b25~b22	Reserved	-	必须保持复位值。	R/W
b21	STALL	STALL握手	STALL握手(STALL handshake) 此端点接收到SETUP令牌时，应用程序只能将此位置1，而模块会将其清零。 如果NAK位、全局OUT NAK与此位同时置1，则STALL位优先。无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R/W
b20	SNPM	监听模式	监听模式 (Snoop mode) 此位用于将端点配置为监听模式。在监听模式下，模块不会在将OUT数据包传输到应用存储区前检查其是否正确。	R/W

b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 硬件设置为‘00’，表示控制类型的端点。	R/W
b17	NAKSTS	NAK状态	NAK状态(NAK status) 指示以下结果： 0：模块根据FIFO状态回复非NAK握手。 1：模块在此端点上回复NAK握手。 当应用程序或模块将此位置1时，即使RxFIFO中存在空间可继续容纳收到的数据包，模块也会停止接收数据。无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R
b16	Reserved	-	必须保持复位值。	R/W
b15	USBAEP	USB活动端点	USB活动端点(USB active endpoint) 此位总是置1，指示在所有配置和接口中控制端点0始终处于激活状态。	R
b14~b2	Reserved	-	必须保持复位值。	R/W
b1~b0	MPSIZ	最大数据包大小	最大数据包大小(Maximum packet size) 控制OUT端点0的最大数据包大小与在控制IN端点0中进行编程的值相同。 00：64 字节 01：32 字节 10：16 字节 11：8 字节	R/W

34.7.4.22 USBHS 设备 OUT 端点 x 控制寄存器(USBHS_DOEPCTLx)(x=1..15)

USBHS 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，USBHS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/W
b21	STALL	STALL握手	STALL握手(STALL handshake) 此端点接收到SETUP令牌时，应用程序只能将此位置1，而模块会将其清零。如果NAK位、全局OUT NAK与此位同时置1，则STALL位优先。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 DPID：端点数据PID (Endpoint data PID) 仅适用于中断/批量OUT端点。 包含此端点上将要接收或发送的数据包的PID。端点激活后，应用程序必须对要在此端点上接收或发送的首个数据包的PID进行编程。应用程序使用SD0PID 寄存器字段对 DATA0或DATA1 PID 进行编程。 0：DATA0 1：DATA1	R/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

34.7.4.23 USBHS 设备 OUT 端点 x 中断寄存器 (USBHS_DOEPINTx) (x=0..15)

USBHS 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 USBHS_GINTSTS 寄存器中的 OUT 端点中断位（USBHS_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 USBHS_DAINT 寄存器，以获取 USBHS_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 USBHS_DAINT 和 USBHS_GINTSTS 寄存器中的对应位清零。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved	NYET	Reserved							B2BSTUP	Reserved	OTEPDIS	STUP	Reserved	EPDISD	XFRC

位	标记	位名	功能	读写
b31~b15	Reserved	-	必须保持复位值。	R/W
b14	NYET	NYET中断	当非同步 OUT 端点回复 NYET 握手信号时，模块将产生此中断。软件写1也可将此位清零。	R/W
b13~b7	Reserved	-	必须保持复位值。	R/W
b6	B2BSTUP	接收到连续的SETUP数据包	接收到连续的SETUP数据包 (Back-to-back SETUP packets received) 仅适用于控制OUT端点。此位指示该端点已接收到三个以上的连续SETUP数据包。软件写1也可将此位清零。	R/W
b5	Reserved	-	必须保持复位值。	R/W
b4	OTEPDIS	端点禁止时接收到OUT令牌	端点禁止时接收到OUT令牌 (OUT token received when endpoint disabled) 仅适用于控制OUT端点。指示在尚未使能端点时接收到OUT令牌，从而产生中断。通过软件写1清零。	R/W
b3	STUP	SETUP阶段完成	SETUP 阶段完成 (SETUP phase done) 仅适用于控制OUT端点。指示控制端点的SETUP阶段已完成，当前控制传输中不再接收到连续的 SETUP数据包。在此中断上，应用程序可以对接收到的SETUP数据包进行解码。 通过软件写1清零。	R/W
b2	Reserved	-	必须保持复位值。	R/W

b1	EPDISD	端点禁止中断	端点禁止中断 (Endpoint disabled interrupt) 此位指示该端点已经由应用程序禁止掉。 通过软件写1清零。	R/W
b0	XFRC	传输完成中断	传输完成中断 (Transfer completed interrupt) 此字段指示在此端点上设置的传输已经在USB和AHB上传输完成。 通过软件写1清零。	R/W

34.7.4.24 USBHS 设备 OUT 端点 0 传输大小寄存器(USBHS_DOEPTSIZ0)

USBHS 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（USBHS_DIEPCTL0 中的 EPENA）使能端点 0 后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

非零端点使用端点 1~15 的寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved	STUPCNT[1:0]		Reserved									PKTCNT	Reserved		

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved									XFRSIZ[6:0]						

位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R/W
b30~b29	STUPCNT	SETUP数据包计数	SETUP数据包计数 (SETUP packet count) 此字段指定端点能连续接收的SETUP数据包数量。 01：1个数据包 10：2个数据包 11：3个数据包	R/W
b28~b20	Reserved	-	必须保持复位值。	R/W
b19	PKTCNT	数据包计数	数据包计数 (Packet count) 一次传输中应该接收到数据包数量。 在端点使能前，软件设置该位，在传输开始后，每当数据包接收到后，该域数值自动减少。	R/W
b18~b7	Reserved	-	必须保持复位值。	R/W
b6~b0	XFRSIZ	传输大小	传输大小(Transfer size) 指示端点0的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。 每次从RxFIFO读取数据包并将其写入外部存储器时，模块会使此字段递减。	R/W

34.7.4.25 USBHS 设备 OUT 端点 x 传输大小寄存器(USBHS_DOEPTSIZx)(x=1..15)

USBHS 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 USBHS_DOEPCTLx 寄存器中的端点使能位（USBHS_DOEPCTLx 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	RXDPID[1:0]/ STUPCNT[1:0]		PKTCNT[9:0]										XFRSIZ[18:16]		

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

位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R/W
b30~b29	RXDPID/ STUPCNT	接收到的数据PID/ SETUP数据包计数	接收到的数据PID (Received data PID) 仅适用于同步 OUT 端点。 这是此端点收到的上一个数据包的 PID。 00：DATA0 01：DATA2 10：DATA1 11：MDATA STUPCNT：SETUP数据包计数 (SETUP packet count) 仅适用于控制OUT端点。 此字段指定端点能连续接收的SETUP数据包数量。 01：1 个数据包 10：2 个数据包 11：3 个数据包	R/W
b28~b19	PKTCNT	数据包计数	数据包计数 (Packet count) 指示该端点上的一次数据传输包含的数据包个数。 每次向RxFIFO写入数据包（最大大小或短数据包）后，此字段将递减。	R/W
b18~b0	XFRSIZ	传输大小	传输大小(Transfer size) 此字段包含当前端点的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这 些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。 每次从RxFIFO读取数据包并将其写入外部存储器时，模块会使此字段递减。	R/W

34.7.4.26 USBHS 设备 OUT 端点 x DMA 地址寄存器（USBHS_DOEPDMAx)(x=0..15)

USBHS Device OUT endpoint x transfer size register

偏移地址：0xB14 + (端点编号 × 0x20)

复位值：0xXXXX 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

34.7.5 USBHS 时钟门控控制寄存器

通过门控时钟控制寄存器控制 HCLK 和 PHY 时钟从而降低功耗。除非特别说明，否则寄存器描述中的位值以二进制表示。

34.7.5.1 USBHS 时钟门控控制寄存器(USBHS_GCCTL)

偏移地址：0xE00

复位值：0x0000 0000

此寄存器在主机模式和设备模式下均可用。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								SUSP	PHYSLEEP	ENL1GTG	Reserved			GATEHCLK	STPPCLK

位	标记	位名	功能	读写
b31~b8	Reserved	-	必须保持复位值。	R/W
b7	SUSP	Deep Sleep状态位	Deep Sleep（L1 suspended）状态位	R
b6	PHYSLEEP	PHY SLEEP模式状态位	PHY SLEEP模式状态位（PHY in sleeep） 该位用于表示PHY处于SLEEP模式状态。	R
b5	ENL1GTG	L1时钟门控	L1时钟门控（Enable Sleep Clock Gating） 该位用于L1模式时时钟门控控制。	R/W
b4~b2	Reserved	-	必须保持复位值。	R/W
b1	GATEHCLK	门控HCLK	门控HCLK (Gate HCLK) 当USB通信挂起或会话无效时，应用程序会将此位置 1，以停止对除AHB总线从接口、主接口和唤醒逻辑之外的模块提供时钟。当USB恢复通信或新会话启动时，应用程序将此位清零。	R/W
b0	STPPCLK	停止PHY时钟	停止PHY时钟(Stop PHY clock) 当USB通信挂起、会话无效或设备断开连接时，应用程序将此位置1以停止PH时钟。当USB恢复通信时，应用程序将此位清零。	R/W

35 USB2.0 全速模块（USBFS）

35.1 USBFS 简介

USB 全速（USBFS）控制器为便携式设备提供了一套 USB 通信解决方案。USBFS 控制器支持主机模式和设备模式，且芯片内部集成全速 PHY。主机模式下，USBFS 控制器支持全速（FS，12Mb/s）和低速（LS，1.5Mb/s）收发器，而设备模式下则仅支持全速（FS，12Mb/s）收发器。USBFS 控制器支持 USB 2.0 协议所定义的所有四种传输方式（控制传输、批量传输、中断传输和同步传输）。该 USBFS 控制器支持 LPM(Link Power Management)功能。

遵循协议如下：

- Universal Serial Bus Revision 2.0 Specification
- USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007
- Errata for USB 2.0 ECN: Link Power Management (LPM) - 7/2007

35.2 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。

35.2.1 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 2.5KB 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

35.2.2 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 16 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

35.2.3 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
- 15 个 OUT 端点，可以配置为支持批量传输、中断传输或同步传输
- 15 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 16 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。

35.3 USBFS 系统框图

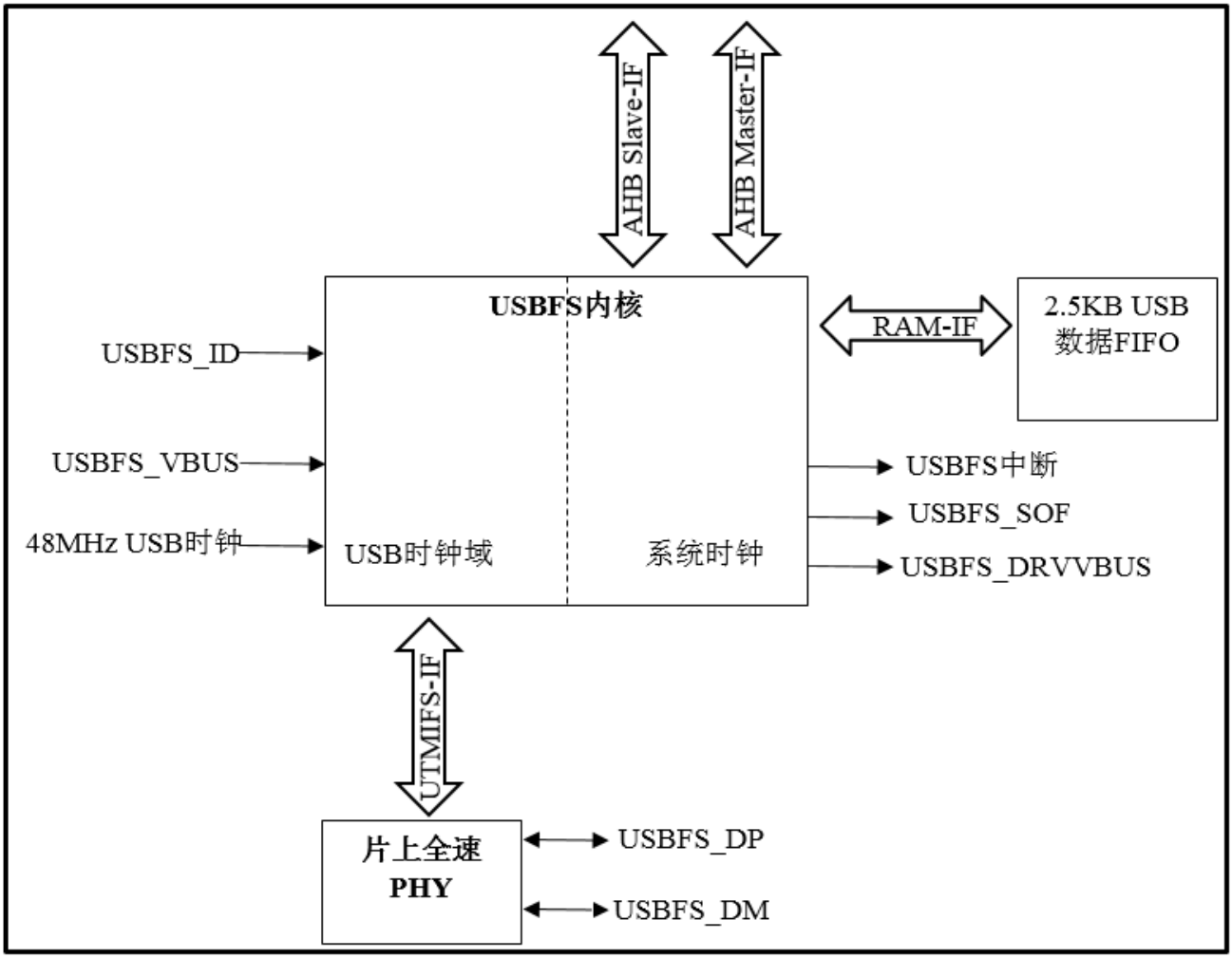


图 35-1 USBFS 系统框图

35.4 USBFS 管脚说明

管脚名	方向	适用模式	对应管脚	功能描述
USBFS_VBUS	输入	设备	PA9	电源端口，5V耐压
USBFS_DP	输入/输出	主机、设备	PA12	差分数据D+信号
USBFS_DM	输入/输出	主机、设备	PA11	差分数据D-信号
USBFS_DRVVBUS	输出	主机	PC9、PB8	外部电源芯片使能信号
USBFS_ID	输入	主机	PA10	USB A-B器件识别信号
USBFS_SOF	输出	主机、设备	PA8	SOF输出脉冲信号

表 35-1 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，USBFS_DP 和 USBFS_DM 功能为模拟功能，与对应的 PFSR 寄存器设定无关。具体请参考 011.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。

35.5 USBFS 功能说明

35.5.1 USBFS 时钟以及工作模式

USBFS 所使用的时钟需要配置为 48MHz，该 48MHz 时钟由内部 PLL 电路产生，PLL 时钟源需选择外部高速振荡器，使用 USBFS 模块前，需要在 CMU 模块内配置好 USBFS 时钟。

USBFS 可以作为主机或者设备使用，并且包含一个片上全速 PHY。

上拉和下拉电阻已经集成在片上全速 PHY 的内部，并且 USBFS 可以根据当前模式和连接状态自动选择。

USBFS 工作时，VCC 电压范围为 3.0~3.6V。

35.5.2 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。

35.5.3 USBFS 主机功能

35.5.3.1 主机功能简介

当 USBFS 工作在主机模式下时，VBUS 为 USB 协议所规定的 5V 电源引脚。内部 PHY 不支持提供 5V 电源，所以需要外接 USB 电源芯片为设备供电。USBFS_DRVVBUS 用来使能外接 USB 电源芯片，外接电源芯片的过电流检出可以通过本 MCU 的外部中断 IRQ 实现。主机模式下 USBFS_VBUS 可以作为 GPIO 使用。

典型的 USB 主机模式系统构建图如下：

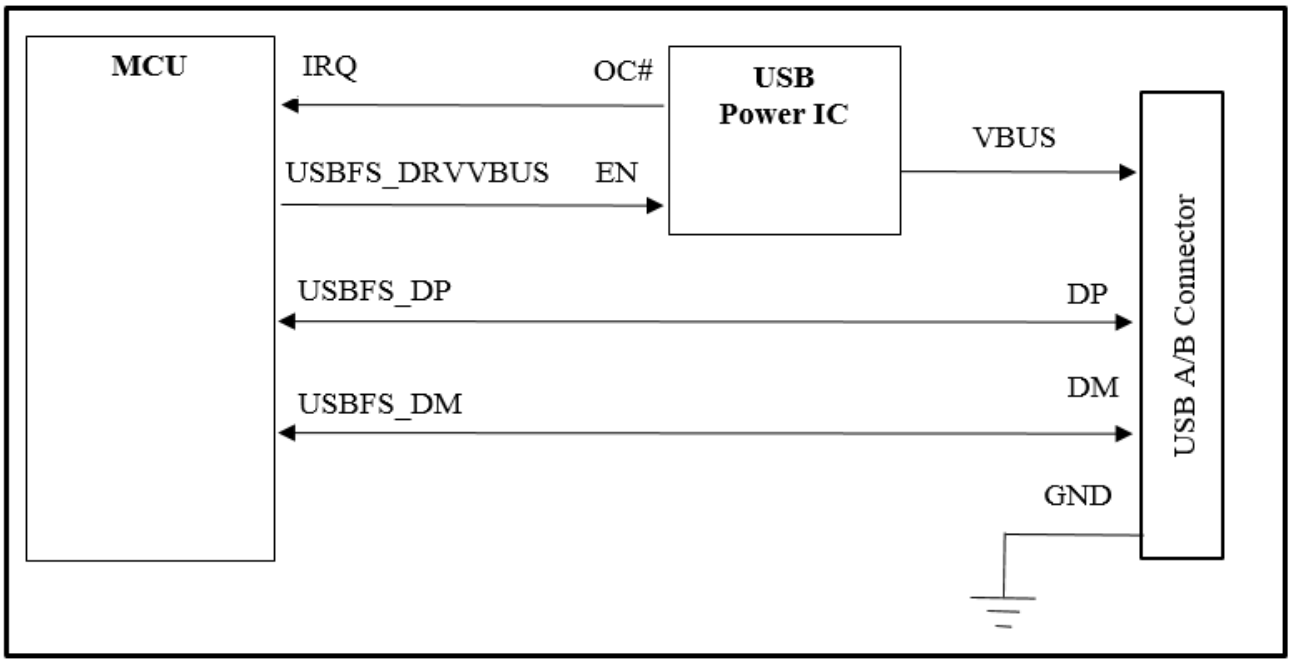


图 35-2 USBFS 主机模式系统构建图

35.5.3.2 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。

35.5.3.3 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）。

35.5.3.4 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。

35.5.3.5 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。

35.5.3.6 主机通道

USBFS 模块实现了 16 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

35.5.3.7 主机调度器

主机模块内置硬件调度器，可自主对应用程序发出的 USB 事务请求重新排序和管理。每一帧开始时，主机都先执行周期性（同步和中断）事务，然后执行非周期性（控制和批量）事务，以符合 USB 规范对同步和中断传输高优先级的保证。

主机通过请求队列（一个周期性请求队列和一个非周期请求队列）处理 USB 事务。每个请求队列最多可存储 8 个条目。每个条目代表一个应用程序发起但还未得到响应的 USB 事务请求，并存储了执行该 USB 事务所用到的 IN 或 OUT 通道的编号，以及其它相关信息。USB 事务请求在队列中的写入顺序决定了事务在 USB 接口上的执行顺序。

每一帧开始时，主机都先处理周期性请求队列，然后处理非周期性请求队列。如果当前帧结束时，计划在当前帧执行的同步或中断类型的 USB 传输事务请求仍处于挂起状态，则主机将发出未完成周期性传输中断（USBFS_GINTSTS 中的 IPXFR 位）。USBFS 模块负责对周期性和非周期性请求队列的管理。周期性发送 FIFO 和队列状态寄存器 (HPTXSTS)与非周期性发送 FIFO 和队列状态寄存器 (HNPTXSTS) 都为只读寄存器，应用程序可使用它们 来读取各请求队列的状态。其中包括：

- 周期性（非周期性）请求队列中当前可用的空闲条目数（最多 8 个）
- 周期性（非周期性）TxFIFO（OUT 事务）中当前可用的空闲空间
- IN/OUT 令牌、主机通道编号和其它状态信息

由于每个请求队列最多可存储 8 个 USB 事务请求，因此应用程序可以把主机 USB 事务请求提前发送给调度器；实际的通信最晚会在调度器处理完已挂起的 8 个周期事务和 8 个非周期事务完成之后出现在 USB 总线上。

要向主机调度器（队列）发出事务请求，应用程序必须读取 USBFS_HPTXSTS 寄存器

中的 PTXQSAV 位或 USBFS_HNPTXSTS 寄存器中的 NPTQXSAV 位，确保周期性（非周期性）请求队列中至少有一个可用空间来存储当前请求。

35.5.4 USBFS 设备功能

35.5.4.1 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：

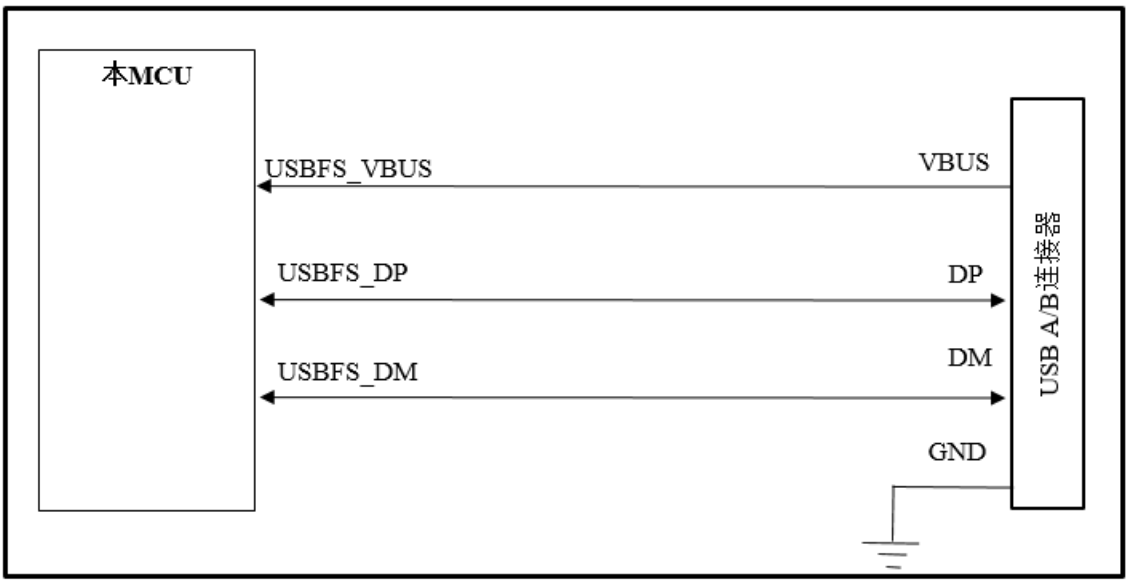


图 35-3 USBFS 设备模式系统构建图

35.5.4.2 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。

35.5.4.3 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。

35.5.4.4 设备挂起状态

USBFS 设备持续监视 USB 活动。在 USB 空闲时间达到 3ms 后，将发出早期挂起中断（USBFS_GINTSTS 中 ESUSP 位），并在 3ms 后由挂起中断（USBFS_GINTSTS 中的 USBSUSP 位）确认设备进入挂起状态。然后，设备状态寄存器中的设备挂起位（USBFS_DSTS 中的 SUSPSTS 位）自动置 1，USBFS 随即进入挂起状态。

可通过设备本身退出挂起状态。这种情况下，应用程序会将设备控制寄存器中的远程唤醒信号位（USBFS_DCTL 中的 RWUSIG 位）置 1，并在 1ms 到 15ms 内将其清零。但若设备检测到主机发出的恢复信号，将生成恢复中断（USBFS_GINTSTS 中的 WKUPINT 位），设备挂起位自动清零。

35.5.4.5 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。

35.5.4.6 设备端点

端点类别

USBFS 模块实现了以下 USB 端点：

- 控制端点 0：
  - 双向且仅处理控制消息
  - 使用一组单独的存器来处理 IN 和 OUT 事务
  - 专用控制 (USBFS_DIEPCTL0/USBFS_DOEPCTL0) 寄存器、传输配置 (USBFS_DIEPTSIZ0/USBFS_DIEPTSIZ0) 寄存器和状态中断 (USBFS_DIEPINTx/USBFS_DOEPINT0)寄存器。控制和传输大小寄存器中可用的

位组与其它端点中稍有不同

- 15 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
- 15 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 传输尺寸寄存器 (DIEPTSIZx/DOEPTSIZx) 允许应用程序对传输尺寸参数进行编程并读取传输状态。必须在端点控制寄存器中的端点使能位置 1 之前完成对此寄存器的设置。使能端点后，这些字段立即变为只读状态，同时 USBFS 模块根据当前传输状态对这些字段进行更新。

可对以下传输参数进行编程：

- 以字节为单位的传输大小
- 构成整个传输的数据包个数

端点状态/状态

设备端点 x 中断寄存器 (DIEPINTx/DOPEPINTx) 指示端点在出现 USB 和 AHB 相关事件时的状态。当模块中断寄存器中的 OUT 端点中断位或 IN 端点中断位（分别为 USBFS_GINTSTS 中 OEPINT 位或 USBFS_GINTSTS 中的 IEPINT 位）置 1 时，应用程序必须读取这些寄存器以获得详细信息。在应用程序读取这些寄存器之前，必须先读取设备全体端点中断(USBFS_DAINT) 寄存器，以获取设备端点 x 中断寄存器的端点编号。应用程序必须将此寄存器中的相应位清零，才能将 DAINT 和 GINTSTS 寄存器中的相应位清零。

模块提供以下状态检查和中断产生功能：

- 传输完成中断，指示应用程序 AHB 和 USB 端均已完成数据传输
- Setup 阶段已完成（仅针对控制传输类型的 OUT 端点）
- 相关的发送 FIFO 为半空或全空状态（IN 端点）
- NAK 应答已发送到主机（仅针对同步传输的 IN 端点）
- TxFIFO 为空时接收到 IN 令牌（仅针对批量和中断传输类型的 IN 端点）
- 尚未使能端点时接收到 OUT 令牌
- 检测到 babble 错误
- 应用程序关闭端点生效
- 应用程序对端点设置 NAK 生效（仅针对同步传输类型的 IN 端点）

- 接收到 3 个以上连续 setup 数据包（仅针对控制类型的 OUT 端点）
- 检测到超时状况（仅针对控制传输类型的 IN 端点）

35.5.5 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。

35.5.5.1 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。

35.5.5.2 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。

35.5.6 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。

35.5.7 USBFS 动态更新 USBFS_HFIR 寄存器

主机模式下，USB 模块具有对帧周期进行动态微调的功能，能够将外部设备与 SOF 帧进行同步。如果 USBFS_HFIR 寄存器在当前 SOF 帧内发生更改，则将在下一个帧中对 SOF 周期进行相应修正，具体说明请参见图 34-4。

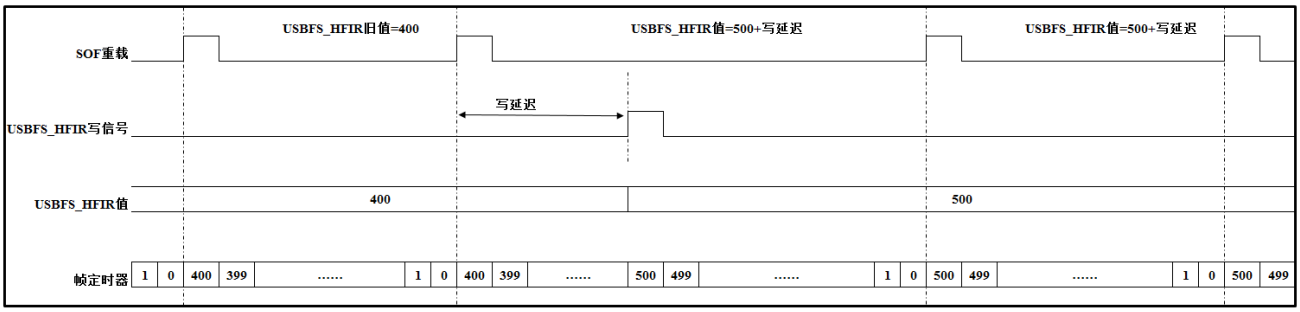


图 35-4 USBFS 动态更新 USBFS_HFIR 寄存器示意图

35.5.8 USBFS 数据 FIFO

USBFS 系统具有 2.5KB 专用 RAM，采用高效的 FIFO 控制机制。USBFS 模块中的数据包 FIFO 控制器模块将 RAM 空间划分为多个 TxFIFO（USB 传输前，应用程序将数据压入其中进行短暂存储）和单个 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。

35.5.9 USBFS 主机 FIFO 架构

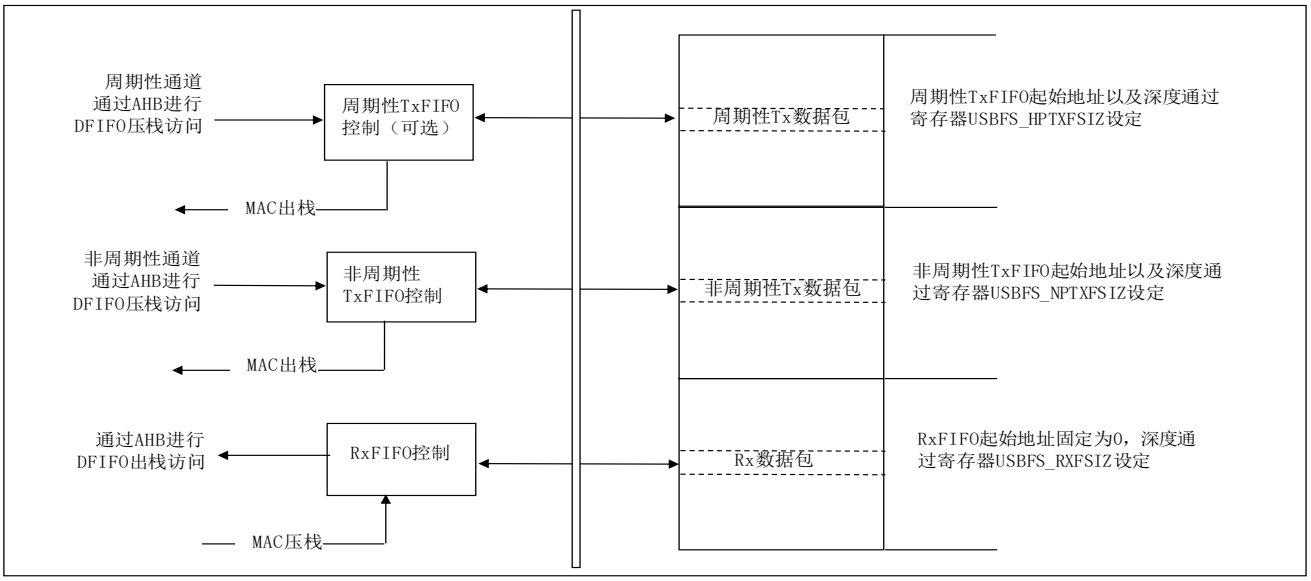


图 35-5 USBFS 主机模式下 FIFO 架构示意图

35.5.9.1 主机 RxFIFO

主机使用一个接收 FIFO 处理所有周期和非周期事务。FIFO 用作接收缓冲区以保存从 USB 接收到的数据（接收到的数据包的数据部分），直至这些数据传输到系统存储器。只要 FIFO 中有空间，来自设备 IN 端点的数据包就接收进来并挨个存储。接收到的每个数据包的状态（包含主机目标通道、字节数、数据 PID 和对所接收数据的校验）也存储在 FIFO 中。 接收 FIFO 的大小在接收 FIFO 大小寄存器 (GRXFSIZ) 中配置。单个接收 FIFO 架构使得 USB 主机高效地填充接收数据缓冲区：

- 所有 IN 配置主机通道共享同一个 RAM 缓冲区（共享 FIFO）
- 对于主机软件驱动的任意序列 IN 令牌，USBFS 模块可将接收 FIFO 填充至限值

只要至少有一个数据包在 RxFIFO 中可供读取，应用程序就会接收 Rx FIFO 非空中断。应用程序从接收状态读取和出栈寄存器中读取数据包信息，最后从 RxFIFO 中读出数据。

35.5.9.2 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。

35.5.10 USBFS 设备 FIFO 架构

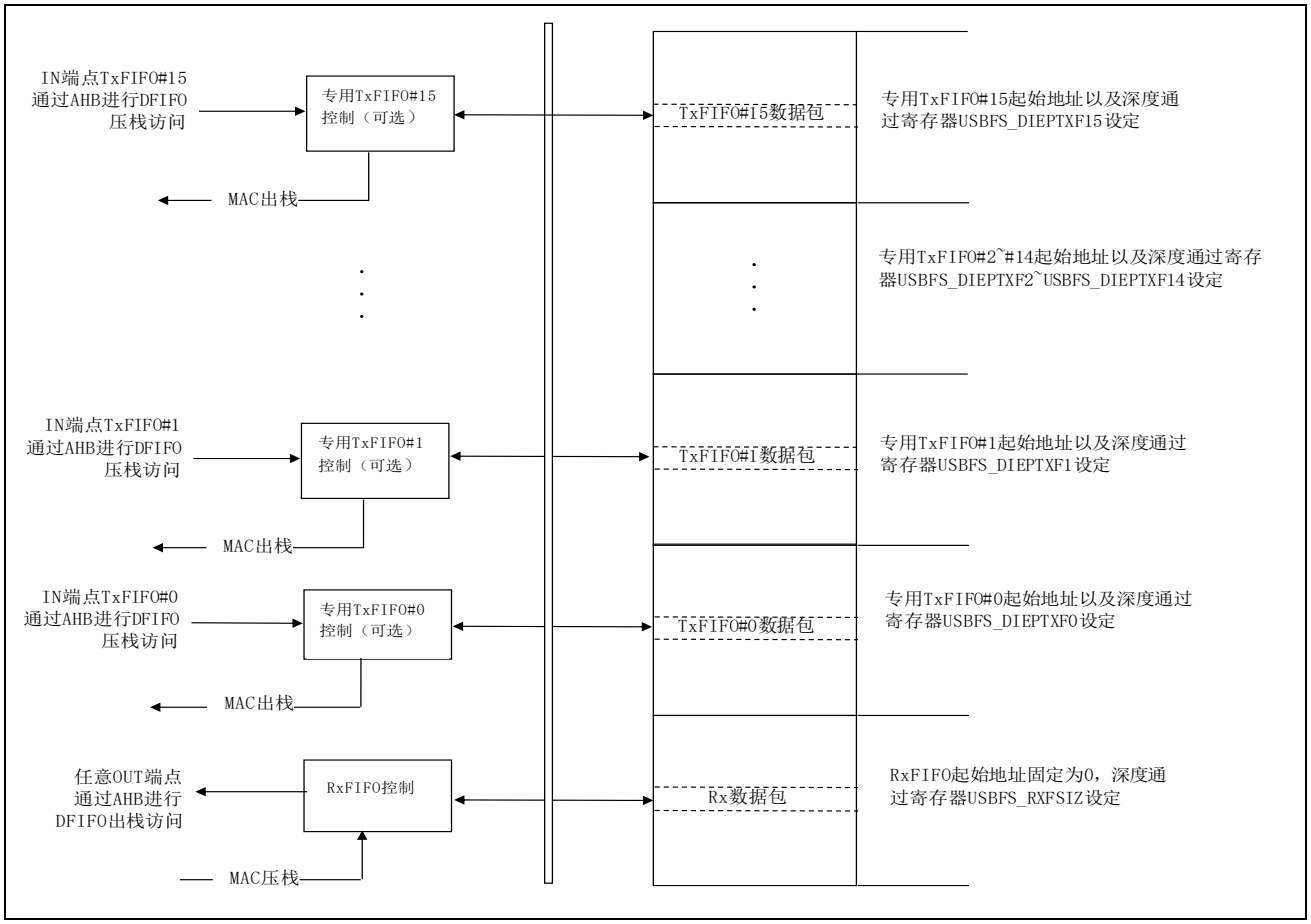


图 35-6 USBFS 设备模式下 FIFO 架构示意图

35.5.10.1 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。

35.5.10.2 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。

35.5.11 USBFS FIFO RAM 分配

35.5.11.1 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。

35.5.11.2 设备模式

接收 FIFO RAM 分配

应用程序应为 SETUP 数据包分配 RAM：接收 FIFO 中必须保留 11 个位置以在控制端点上接收 SETUP 数据包。USBFS 模块不会向这些为 SETUP 数据包保留的位置写入任何其它数据。将会为全局 OUT NAK 分配一个位置。状态信息随各个接收数据包写入 FIFO。因此，必须至少为接收数据包分配（最大数据包大小/4）+1 的空间。如果使能了多个同步端点，则为接收连续数据包分配的空间必须至少为（最大数据包大小/4）的两倍+1。通常，推荐的空间为（最大数据包/4 + 1）的两倍，这样当上一个数据包向 CPU 传送时，USB 可同时接收后续的数据包。

传输完成状态信息和该端点收到的最后一个数据包会一起被推入 FIFO。通常情况下，推荐为每个 OUT 端点分配一个位置。

发送 FIFO RAM 分配

各个 IN 端点发送 FIFO 所需的最小 RAM 空间为该特定 IN 端点的最大数据包大小。

35.5.12 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 2.5KB RAM 缓冲区且 2.5KB 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。

35.5.13 USBFS 中断和事件

USBFS 中断有以下两类，SOTP 模式唤醒中断 USBFS_WKUP 和 USBFS 全局中断 USBFS_GLB。

USBFS_WKUP 中断

USBFS_WKUP 中断用于 STOP 模式下，通过 USBFS_DP 或者 USBFS_DM 唤醒系统的 STOP 模式，该中断使能位为 INT_WUPEN.USF_WUEN。

使用 USBFS 唤醒系统 STOP 前，需要确保 USBFS 控制器处于挂起状态，并设定寄存器 USBFS_SYCTLREG 中的对应的滤波范围和使能滤波功能。

USBFS_GLB 中断

USBFS_GLB 中断是软件需要处理的主要中断，全局中断的标志位可在 USBFS_GINTSTS 寄存器读取。

中断标志	描述	运行模式	内部事件
WKUPINT	恢复/远程唤醒中断	主机或者设备	-
VBUSVINT	VBUS有效中断	设备	-
DISCINT	断开连接中断	主机	-
CIDSCHG	连接器ID线状态变化中断	主机或者设备	-
PTXFE	周期性TxFIFO空中断	主机	-
LPMINT	LPM中断	主机或者设备	
HCINT	主机通道中断	主机	-
HPRTINT	主机端口中断	主机	-
DATAFSUSP	数据获取挂起	设备	-
IPXFR/INCOMPISOOUT	未完成周期性传输/未完成	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	设备	-
NPTXFE	非周期性TxFIFO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 35-2 USBFS_GLB 中断事件表

35.6 USBFS 编程模型

35.6.1 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 35.5.2 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。

35.6.2 USBFS 主机初始化

要将模块作为主机进行初始化，应用程序必须执行以下步骤：

1. 编程 USBFS_GINTMSK 寄存器中的 HPRTINT 以将对取消屏蔽。
2. 编程 USBFS_HCFG 寄存器以选择全速主机。
3. 将 USBFS_HPRT 中的 PPWR 位编程为 1，给 USB 总线提供 VBUS。
4. 等待 USBFS_HPRT 中的 PCDET 中断。这表示某设备已连接到主机端口。
5. 将 USBFS_HPRT 中的 PRST 位编程为 1，在 USB 总线上发出复位信号。

6. 至少等待 10ms，以便完成复位过程。
7. 将 USBFS_HPRT 中的 PRST 位编程为 0。
8. 等待 USBFS_HPRT 中的 PENCHNG 中断。
9. 读取 USBFS_HPRT 中的 PSPD 位以获取枚举速度。
10. 使用所选 PHY 时钟，相应地设置 HFIR 寄存器。
11. 根据步骤 9 中检测到的设备速度编程 USBFS_HCFG 寄存器中的 FSLSPCS 字段。如果 FSLSPCS 发生更改，则必须执行端口复位。
12. 编程 USBFS_GRXFSIZ 寄存器以选择接收 FIFO 的大小。
13. 编程 USBFS_HNPTXFSIZ 寄存器，以选择用于非周期性通信事务的非周期性发送 FIFO 的大小和起始地址。
14. 编程 USBFS_HPTXFSIZ 寄存器，以选择用于周期性事务的周期性通信发送 FIFO 的大小和起始地址。

要与设备通信，系统软件必须初始化并使能至少一个通道。

35.6.3 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。

35.6.4 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。

35.6.5 USBFS 主机编程模型

35.6.5.1 通道初始化

应用程序必须初始化一个或多个通道，之后才能与所连接的设备通信。

要初始化和使能通道，应用程序必须执行以下步骤：

1. 编程 USBFS_GINTMSK 寄存器以取消对以下位的中断屏蔽：
   - 用于 OUT 事务的非周期性发送 FIFO 为空（在流水线事务级别工作且数据包计数字段编程值大 1 时适用）。
   - 用于 OUT 事务的非周期性发送 FIFO 为半空（在流水线事务级别工作且数据包计数字段编程值大 1 时适用）。
2. 编程 USBFS_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_HCTSIZx 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）。

35.6.5.2 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。

35.6.6 USBFS 设备编程模型

35.6.6.1 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。

35.6.6.2 USB 复位时的端点端点初始化

1. 在枚举完成中断（USBFS_GINTSTS 中的 ENUMDNE）中，读取 USBFS_DSTS 寄存器以确定设备的枚举速度。
2. 对 USBFS_DIEPCTL0 中的 MPSIZ 字段进行编程以设置最大数据包大小。该步骤配置控制端点 0。控制端点的最大数据包大小取决于枚举速度。

此时，设备已准备好接收 SOF 数据包并配置为在控制端点 0 执行控制传输。

35.6.6.3 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

35.6.6.4 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

35.6.6.5 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。

35.6.6.6 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。

35.6.7 USBFS 操作模型

35.6.7.1 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_DOEPTSIZx) 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_DOEPCTLx 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_DOEPTSIZx 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_DOEPTSIZx 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_DOEPINTx 中的 B2BSTUP）。

将全局 OUT NAK 置 1

内部数据流：

1. 如果应用程序将全局 OUT NAK（USBFS_DCTL 中的 SGONAK 位）置 1，模块将停止向接收 FIFO 中写入 SETUP 数据包以外的数据。无论接收 FIFO 中可用空间大小如何，设备都会对主机发送的非同步 OUT 令牌回复 NAK，而对同步 OUT 数据包直接予以忽略。
2. 模块将全局 OUT NAK 写入接收 FIFO。应用程序必须为此留出足够空间。
3. 当应用程序从接收 FIFO 中弹出全局 OUT NAK 字时，模块会将 GONAKEFF 中断 (USBFS_GINTSTS) 置 1。
4. 应用程序检测到该中断后，会认为模块处于全局 OUT NAK 模式。应用程序可以通过将 USBFS_DCTL 中的 SGONAK 位清零来清除该中断。

应用程序编程顺序：

1. 要停止接收任何类型的数据到接收 FIFO 中，应用程序必须通过编程以下字段以将全局 OUT NAK 位置 1。
   - 在 USBFS_DCTL 中，SGONAK =1
2. 等待 USBFS_GINTST 中的 GONAKEFF 中断。一旦被触发，该中断表示模块已停止接收 SETUP 数据包以外的任何类型数据。
   - 如果应用程序已将 USBFS_DCTL 中的 SGONAK 位置 1，则在模块引发 GONAKEFF 中断 (USBFS_GINTSTS) 之前，应用程序可以接收有效 OUT 数据包。
3. 应用程序可通过对 USBFS_GINTMSK 寄存器中的 GINAKEFFM 位执行写操作来暂时屏蔽此中断。
   - 在 USBFS_GINTMSK 寄存器中，GINAKEFFM=0
4. 当应用程序准备退出全局 OUT NAK 模式时，必须将 USBFS_DCTL 中的 SGONAK 位清零。此操作还会清除 GONAKEFF 中断 (USBFS_GINTSTS)。

- 在 CGONAK 中，USBFS_DCTL=1

5. 如果应用程序在之前已屏蔽此中断，则必须按以下方式取消对该中断的屏蔽：

- 在 GINTMSK 中，GINAKEFFM=1

将全局 OUT NAK 置 1

应用程序必须使用以下顺序禁止已使能的 OUT 端点。

应用程序编程顺序：

1. 禁止任何 OUT 端点前，应用程序必须在模块中使能全局 OUT NAK 模式。

- 在 USBFS_DCTL 中，SGONAK=1

2. 等待 GONAKEFF 中断(USBFS_GINTSTS)
3. 通过编程以下字段来禁止 OUT 端点：

- 在 USBFS_DOEPCTLx 中，EPDIS=1
- 在 USBFS_DOEPCTLx 中，SNAK=1

4. 等待 EPDISD 中断 (USBFS_DOEPINTx)，该中断表示已完全禁止 OUT 端点。引发 EPDISD 中断时，模块还会将以下位清零：

- 在 USBFS_DOEPCTLx 中，EPDIS=0
- 在 USBFS_DOEPCTLx 中，EPENA=0

5. 应用程序必须将全局 OUT NAK 位清零，以开始从其它未禁止的 OUT 端点接收数据。

- 在 USBFS_DCTL 中，SGONAK=0

通用非同步 OUT 数据传输

本节介绍一种常规非同步 OUT 数据传输（控制、批量或中断）。

应用程序要求：

1. 建立 OUT 传输前，应用程序必须在存储器中分配一个缓冲区，以容纳要作为 OUT 传输的一部分而接收的所有数据。
2. 对于 OUT 传输，端点的传输大小寄存器中的传输大小字段必须是端点的最大数据包大小的倍数（且以字对齐）。

- 传输大小[EPNUM] = n × (MPSIZ[EPNUM] + 4 - (MPSIZ[EPNUM] mod 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 位置 1 后，将忽略同步或非同步数据包并且不会将它们写入接收 FIFO，同时非同步 OUT 令牌会收到 NAK 握手应答。
4. 在数据写入接收 FIFO 后，应用程序将从接收 FIFO 中读取数据并将数据写入外部存储器，一次一个数据包，逐个端点过来。

5. 在 AHB 上向外部存储器写入完每个数据包后，端点的传输大小都会自动减去该数据包的大小。
6. 在以下情况时，OUT 端点的 OUT 数据传输完成状态将写入接收 FIFO：
   - 传输大小为 0 并且数据包计数为 0
   - 写入接收 FIFO 的最后一个 OUT 数据包是短数据包
     （数据包大小:0~最大数据包大小-1）
7. 当应用程序弹出此状态条目（OUT 数据传输完成），并生成该端点的传输完成中断，同时清零端点使能位。

应用程序编程顺序：

1. 使用传输大小和相应数据包个数对 USBFS_DOEPTSIZx 寄存器进行编程。
2. 使用端点特性对 USBFS_DOEPCTLx 寄存器进行编程，并将 EPENA 和 CNAK 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 OUT 数据传输。
2. 对于同步 OUT 数据传输中的传输大小和数据包计数字段，必须始终将其设置为单个帧中可接收的最大数据包大小的数据包数目。同步类型的 OUT 数据传输事务必须在一个帧内完成。
3. 在周期性帧结束（USBFS_GINTSTS 中的 EOPF 中断）之前，应用程序必须从接收 FIFO 中读取所有同步 OUT 数据包（数据条目和状态条目）。

4. 要接收下一帧中的数据，必须在 EOPF(USBFS_GINTSTS) 之后 SOF(USBFS_GINTSTS)之前使能一个同步 OUT 端点。

内部数据流：

1. 同步 OUT 端点的内部数据流与非同步 OUT 端点的基本相同，但稍有差异。
2. 同步 OUT 端点通过将端点使能位置 1 并将 NAK 位清零来使能时，必须相应地将偶数/奇数帧位置 1。仅当符合以下条件时，模块才会在同步 OUT 端点上接收特定帧中的数据：
   - EONUM（在 USBFS_DOEPCTLx 中）= FNSOF [0]（在 USBFS_DSTS 中）
3. 当应用程序从接收 FIFO 中完整地读取一个同步 OUT 数据包（数据和状态）时，模块会根据从接收 FIFO 中读取的最后一个同步 OUT 数据包的数据 PID 更新 USBFS_DOEPTSIZx 中的 RXDPID 字段。

应用程序编程顺序：

1. 使用传输大小和相应数据包计数对 USBFS_DOEPTSIZx 寄存器进行编程
2. 使用端点特性对 USBFS_DOEPCTLx 寄存器进行编程，并将端点使能位、清除 NAK 位和奇数/偶数帧位置 1。
   - EPENA1
   - CNAK=1
   - EONUM=（0：偶数/1：奇数）
3. 等待 RXFNE 中断（在 USBFS_GINTSTS 中）并且从接收 FIFO 中读走数据包。
   - 此步骤可重复多次，具体取决于传输大小。
4. XFRC 中断（在 USBFS_DOEPINTx 中）表示同步 OUT 数据传输完成。该中断不一定意味着存储器中的数据是有效的。
5. 对于同步 OUT 传输，应用程序可能并不总会检测到该中断。相反，应用程序可能检测到 USBFS_GINTSTS 中的 IISOOXFRM 中断。
6. 读取 USBFS_DOEPTSIZx 寄存器以确定接收的传输大小以及确定帧中接收的数据的有效性。仅当符合以下条件之一时，应用程序才必须将存储器中接收的数据视为有效数据：

- RXDPID=D0（在 USBFS_DOEPTSIZx 中）并且接收该有效数据的 USB 数据包数量=1
- RXDPID=D1（在 USBFS_DOEPTSIZx 中）并且接收该有效数据的 USB 数据包数量=2
- RXDPID=D2（在 USBFS_DOEPTSIZx 中）并且接收该有效数据的 USB 数据包数量=3

接收该有效数据的 USB 数据包数量=应用程序编程的初始数据包个数-模块更新后的剩余数据包个数。

应用程序可将无效数据包丢弃。

不完整的同步 OUT 数据传输

本节介绍了同步 OUT 数据包出现丢包时应用程序编程顺序。

内部数据流：

1. 对于同步 OUT 端点，可能不会始终引发 XFRC 中断（在 USBFS_DOEPINTx 中）。如果模块丢弃同步 OUT 数据包，则在以下情况下，应用程序可能无法检测到 XFRC 中断 (USBFS_DOEPINTx)：
   - 在接收 FIFO 无法容纳完整的 ISO OUT 数据包时，模块将丢弃接收到的 ISO OU 数据
   - 接收到的同步 OUT 数据包存在 CRC 错误
   - 模块接收到的同步 OUT 令牌损坏
   - 应用程序从接收 FIFO 中读取数据的速度非常缓慢
2. 如果模块在所有同步 OU 端点的传输完成前检测到周期性帧结束，将触发未完成同步 OUT 数据中断（USBFS_GINTSTS 中的 IISOOXFRM），指示至少有一个同步 OUT 端点上未触发 XFRC 中断（在 USBFS_DOEPINTx 中）。此时，未完成传输的端点仍保持使能，但在 USB 的该端点上，没有进行中的有效传输。

应用程序编程顺序：

1. 硬件触发 IISOOXFRM 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。

35.6.7.2 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空间来写入发送数据包时，应用程序必须首先对端点控制寄存器进行相应写操作，然后再将数据写入数据 FIFO。通常，应用程序必须对 USBFS_DIEPCTLx 寄存器执行读-修改-写操作，以避免在将端点使能位置 1 的同时，修改寄存器中的其它内容。

   如果有足够空间，应用程序可将同一端点的多个数据包写入发送 FIFO。对于周期性 IN 端点，应用程序只能一次写入一个帧内的多个数据包。只有先前一个帧的通信事务传输完成之后，应用程序才会写入下一个帧内要发送的所有数据包。

将 IN 端点 NAK 置 1

内部数据流：

1. 当应用程序将特定端点的 IN NAK 置 1 时，模块将停止端点上的数据发送，而不考虑端点发送 FIFO 中的数据是否可用。
2. 非同步端点收到 IN 令牌，回复 NAK 握手应答。

- 同步端点收到 IN 令牌，返回零长度数据包

3. 模块在 USBFS_DIEPINTx 中触发 INEPNE（IN 端点 NAK 有效）中断以响应 USBFS_DIEPCTLx 中的 SNAK 位。
4. 应用程序检测到该中断后，便会认为端点处于 IN NAK 模式。应用程序可通过将 USBFS_DIEPCTLx 中的 CNAK 位置 1 来清除该中断。

应用程序编程顺序：

1. 要在特定 IN 端点上停止发送任何数据，应用程序必须将 IN NAK 位置 1。要将该位置 1，必须编程以下字段。
- USBFS_DIEPCTLx 中的 SNAK=1
2. 等待 USBFS_DIEPINTx 中的 INEPNE 中断触发。该中断表示模块已在端点上停止发送数据。
3. 在应用程序将 NAK 位置 1 但“NAK 有效”中断尚未触发时，模块可以在端点上发送有效 IN 数据。
4. 应用程序可通过写入 DIEPMSK 中的 INEPNEM 位来临时屏蔽该中断。
- 在 DIEPMSK 中，INEPNEM = 0
5. 要退出端点 NAK 模式，应用程序必须将 USBFS_DIEPCTLx 中的 NAK 状态位 (NAKSTS) 清零。此操作还会清除 INEPNE 中断（在 USBFS_DIEPINTx 中）。
- 在 USBFS_DIEPCTLx 中，CNAK=1
6. 如果应用程序已将该中断屏蔽，则必须按以下方式取消屏蔽：
- 在 DIEPMSK 中，INEPNEM=1

禁止 IN 端点

使用以下顺序来禁止先前已使能的特定 IN 端点。

应用程序编程顺序：

1. 应用程序必须先停止在 AHB 上写入数据，之后才能禁止 IN 端点。
2. 应用程序必须将端点设置为 NAK 模式。
- USBFS_DIEPCTLx 中的 SNAK=1
3. 等待 USBFS_DIEPINTx 中的 INEPNE 中断。

4. 将必须禁止的端点的 USBFS_DIEPCTLx 寄存器中的以下位置 1。
   - USBFS_DIEPCTLx 中的 EPDIS=1
   - USBFS_DIEPCTLx 中的 SNAK=1
5. USBFS_DIEPINTx 中的 EPDISD 中断的触发表示模块已完全禁止指定的端点。在触发中断的同时，模块还会将以下位清零：
   - 在 USBFS_DIEPCTLx 中，EPENA=0
   - 在 USBFS_DIEPCTLx 中，EPDIS=0
6. 应用程序必须为周期性 IN EP 读取 USBFS_DIEPTSIZx 寄存器，以计算端点上有多少数据是在 USB 上发送的。
7. 应用程序必须通过将 USBFS_GRSTCTL 寄存器中的以下字段置 1，来清空端点发送 FIFO 中的数据：
   - TXFNUM（在 USBFS_GRSTCTL 中）=端点发送 FIFO 编号
   - TXFFLSH（在 USBFS_GRSTCTL 中）=1

     应用程序必须轮询 USBFS_GRSTCTL 寄存器，直至模块将 TXFFLSH 位清零，这表示 FIFO 清空操作结束。要在该端点上发送新数据，应用程序可以在稍后重新使能该端点。

通用非周期性 IN 数据传输

应用程序要求：

1. 建立 IN 传输前，应用程序必须确保组成一次 IN 传输的每个数据包都可以容纳在单个缓冲区中。
2. 对于 IN 传输，端点传输大小寄存器中的传输大小字段表示本次传输的有效数据量，它由多个最大数据包大小和单个短数据包组成。该短数据包在传输结束时发送。
   - 要发送多个最大数据包大小的数据包并在传输结束时外加一个短数据包：

     传输大小[EPNUM] = x × MPSIZ[EPNUM] + sp

     如果 (sp > 0)，数据包计数[EPNUM] = x + 1。

     否则，数据包计数[EPNUM] = x
   - 要发送单个零长度数据包：

传输大小[EPNUM] = 0

数据包计数[EPNUM] = 1

- 要发送多个最大数据包大小的数据包并在传输结束时外加一个零长度数据包，应用程序必须将传输拆分为两个部分。

  第一部分发送最大数据包大小的数据包，第二部分仅发送零长度数据包。

  第一次传输：传输大小[EPNUM] = x × MPSIZ[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) × MPSIZ[EPNUM]
- 要通过 USB 发送的剩余数据量 = (应用程序编程的初始传输大小 - 已通过 USB 发送的数据量)

内部数据流：

1. 应用程序必须在特定端点的寄存器中设置传输大小和数据包计数字段，并使能该端点来发送数据。
2. 应用程序还必须向该端点的发送 FIFO 写入必需的数据。
3. 应用程序每向发送 FIFO 写入一个数据包，该端点的传输大小便会自动减去该数据包大小。应用程序持续从存储器获取数据来写入发送 FIFO，直到该端点的传输大小变为 0。向 FIFO 写入数据后，“FIFO 中的数据包数”计数会递增（这是一个 3 位计数，由模块在内部进行维护，每个 IN 端点发送 FIFO 对应一个。在 IN 端点 FIFO 中，模块所维护的最大数据包数始终为八个）。对于零长度数据包，每个 FIFO 均另有一个单独的标志，FIFO 中没有任何数据。
4. 当数据写入发送 FIFO 后，模块会在接收到 IN 令牌时将这些数据送出。每个数据

包发送出去并收到回复的 ACK 握手信号后，该端点的数据包计数都会递减 1，直到数据包计数变 0 为止。发生超时时，数据包计数不会递减。

5. 对于零长度数据包（由内部零长度标志指示），模块会针对 IN 令牌发出一个零长度数据包，并递减数据包计数字段的值。
6. 如果接收到 IN 令牌的端点对应的 FIFO 中无数据，且该端点的数据包计数字段为零，则模块会针对该端点生成一个“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：

传输大小[EPNUM] = x × 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) × MPSIZ =< XFERSIZ <= MCNT × MPSIZ
   - PKTCNT = MCNT（在 USBFS_DIEPTSIZx 中）
   - 如果 XFERSIZ < MCNT × MPSIZ，则传输的最后一个数据包为短数据包
   - 请注意：MCNT 位于 USBFS_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：
   c) 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）。
   d) 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“TxFIFO 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“TxFIFO 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。

35.7 寄存器说明

应用程序通过 AHB 从接口对控制和状态寄存器进行读写操作，以此来控制 USBFS 模块。USBFS 模块所有寄存器为 32 位寄存器，其地址按 32 位对齐，因此只能以 32 位的方式访问。

控制和状态寄存器分为以下几类：

- USBFS 系统控制寄存器
- 模块全局寄存器
- 主机模式寄存器
- 设备模式寄存器
- 电源和时钟门控控制寄存器
- 数据 FIFO(DFIFO)访问寄存器

其中 USBFS 系统控制寄存器与其它寄存器基地址不同，该寄存器独立于 USBFS 模块以控制 USBFS 模块的相关设定。

只有模块全局寄存器、电源和时钟门控控制寄存器和数据 FIFO 访问寄存器可以在主机和设备模式下进行访问。当 USBFS 模块处以一种模式（主机或者设备）下，应用程序不得以另外一种角色模式访问寄存器，比如主机模式时，访问设备模式的寄存器。如果发生了非法访问，将会产生模式不匹配中断并在模块中断寄存器 USBFS.GINTSTS.NMIS 位反映。当模快从一种角色模式切换到另一种角色模式时，新工作模式下的寄存器必须重新编程为上电复位后的状态。

控制状态寄存器存储器映射

主机和设备模式寄存器占用不同的地址。所有寄存器均在 AHB 时钟域内实现。

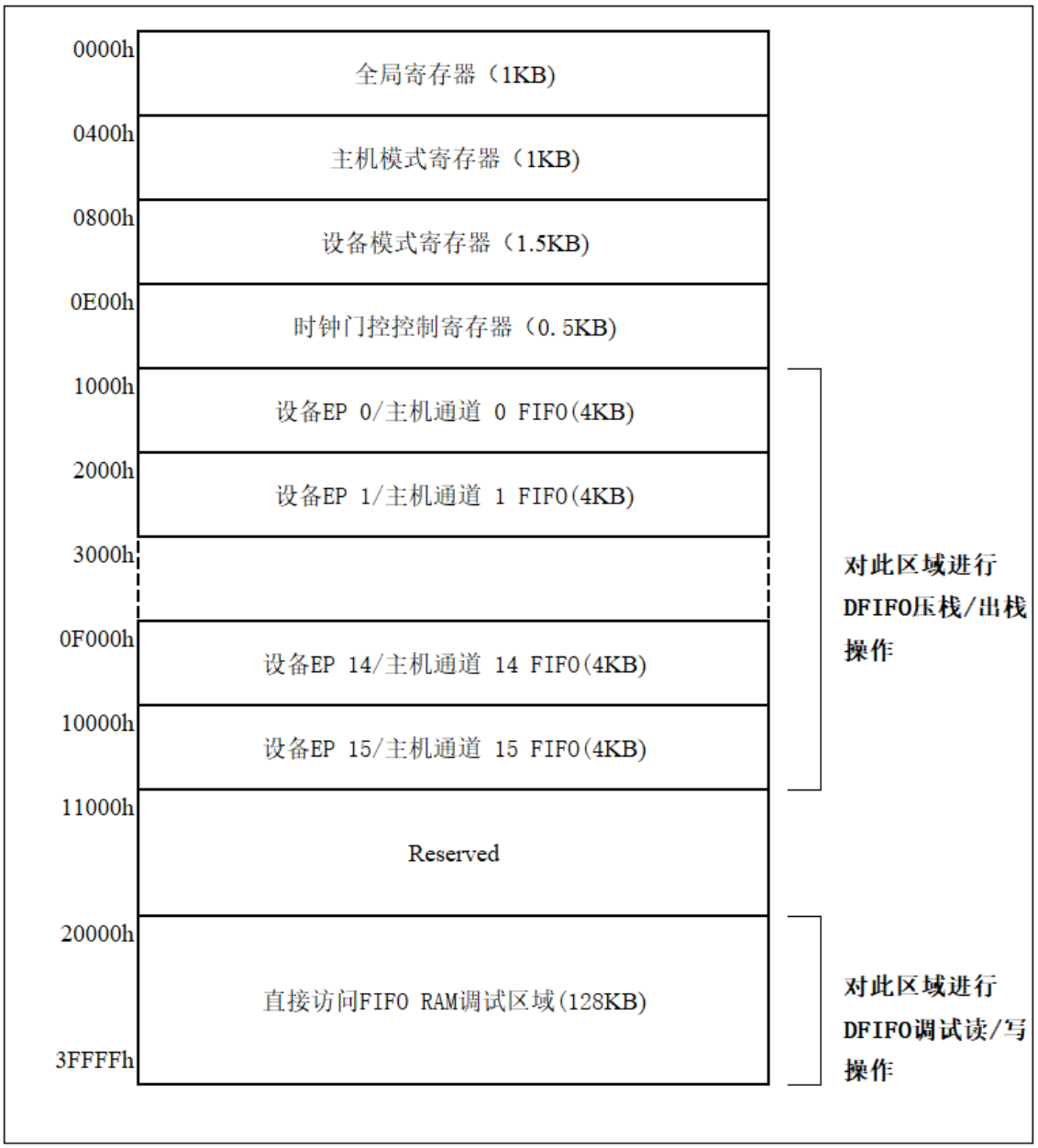


图 35-7 USBFS 控制状态寄存器存储器映射

USBFS 模块寄存器列表以及基地址请参考表 35-3~表 35-4 USBFS 寄存器一览表。

USB 系统控制寄存器基地址：0x40055400

（USBFS系统控制寄存器）寄存器名	偏移地址	复位值
USB系统控制寄存器(USB_SYCTLREG)	0x00	0x0000 0000

表 35-3 USBFS 系统控制寄存器一览表

USBFS 模块寄存器基地址：0x40080000

（USBFS全局寄存器）寄存器名	偏移地址	复位值
USBFS VBUS控制寄存器(USBFS_GVBUSCFG)	0x00	0x00XX 0000
USBFS AHB控制寄存器(USBFS_GAHBCFG)	0x08	0x0000 0000
USBFS USB配置寄存器(USBFS_GUSBCFG)	0x0c	0x0000 1440
USBFS复位寄存器(USBFS_GRSTCTL)	0x10	0x8000 0000
USBFS模块中断寄存器(USBFS_GINTSTS)	0x14	0x1400 0020
USBFS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 0280
USBFS 主机非周期性发送FIFO大小寄存器(USBFS_HNPTXFSIZ)/ 设备端点0发送FIFO大小寄存器(USBFS_DIEPRXF0)	0x28	0x0280 0280
USBFS非周期性发送状态寄存器(USBFS_HNPTXSTS)	0x2c	0x0008 0280
USBFS模块ID寄存器(USBFS_CID)	0x3c	0x1234 5678
USBFS LPM配置寄存器(USBHS_GLPMCFG)	0x54	0x0000 0000
USBFS周期性发送FIFO大小寄存器(USBFS_HPTXFSIZ)	0x100	0x0280 0500
USBFS设备IN端点n发送FIFO大小寄存器(USBFS_DIEPTXFx)	0x100+x*4(x=1~15)	0x0280 0500+(x-1)*0x280

（USBFS主机控制和状态寄存器）寄存器名	偏移地址	复位值
USBFS主机配置寄存器(USBFS_HCFG)	0x400	0x0000 0200
USBFS主机帧时间间隔寄存器(USBFS_HFIR)	0x404	0x0000 EA60
USBFS主机帧编号/帧剩余时间间隔寄存器 (USBFS_HFNUM)	0x408	0x0000 3FFF
USBFS主机周期性发送FIFO/队列状态寄存器 (USBFS_HPTXSTS)	0x410	0x0008 0280
USBFS主机全体通道中断寄存器(USBFS_HAINT)	0x414	0x0000 0000
USBFS主机全体通道中断屏蔽寄存器 (USBFS_HAINTMSK)	0x418	0x0000 0000
USBFS主机端口控制和状态寄存器(USBFS_HPRT)	0x440	0x0000 0000
USBFS主机通道x特性寄存器(USBFS_HCCHARx)	0x500+x*0x20(x=0~15)	0x0000 0000
USBFS主机通道x中断寄存器(USBFS_HCINTx)	0x508+x*0x20(x=0~15)	0x0000 0000
USBFS主机通道x中断屏蔽寄存器(USBFS_HCINTx)	0x50c+x*0x20(x=0~15)	0x0000 0000
USBFS主机通道x传输大小寄存器(USBFS_HCTSIZx)	0x510+x*0x20(x=0~15)	0x0000 0000
USBFS主机通道x DMA地址寄存器(USBFS_HCDMAx)	0x514+x*0x20(x=0~15)	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 0002
USBFS设备状态寄存器(USBFS_DSTS)	0x808	0x0000 0002
USBFS设备IN端点通用中断屏蔽寄存(USBFS_DIEPMSK)	0x810	0x0000 0000
USBFS设备OUT端点通用中断屏蔽寄存器 (USBFS_DOEPMSK)	0x814	0x0000 0000
USBFS设备全体端点中断寄存器(USBFS_DAINT)	0x818	0x0000 0000
USBFS设备全体端点中断屏蔽寄存(USBFS_DAINTMSK)	0x81c	0x0000 0000
USBFS设备IN端点FIFO空中断屏蔽寄存器 (USBFS_DIEPEMPMSK)	0x834	0x0000 0000
USBFS设备IN端点0控制寄存器(USBFS_DIEPCTL0)	0x900	0x0000 8000
USBFS设备IN端点n控制寄存器(USBFS_DIEPCTLx)	0x900+x*0x20(n=1~15)	0x0000 0000
USBFS设备IN端点n中断寄存器(USBFS_DIEPINTx)	0x908+x*0x20(n=0~15)	0x0000 0080
USBFS设备IN端点n传输大小寄存器(USBFS_DIEPSIZx)	0x910+x*0x20(n=0~15)	0x0000 0000
USBFS设备IN端点n DMA地址寄存(USBFS_DIEPDMAx)	0x914+x*0x20(n=0~15)	0x0000 0000
USBFS设备IN端点n发送FIFO状态寄存器 (USBFS_DTXFSTSx)	0x918+x*0x20(n=0~15)	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n控制寄存器(USBFS_DOEPCTLx)	0xb00+x*0x20(n=1~15)	0x0000 0000
USBFS设备OUT端点n中断寄存器(USBFS_DOEPINTx)	0xb08+x*0x20(n=0~15)	0x0000 0000
USBFS设备OUT端点n传输大小寄存(USBFS_DOEPSIZx)	0xb10+x*0x20(n=0~15)	0x0000 0000
USBFS设备OUT端点n DMA地址寄存器 (USBFS_DOEPDMAx)	0xb14+x*0x20(n=0~15)	0xXXXXXXX

（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急促器(USBFS_PCGCCTL)	0xe00	0x0000 0000

表 35-4 USBFS 系统控制寄存器一览表

35.7.1 USB 系统控制寄存器

35.7.1.1 USB 系统控制寄存器(USB_SYCTLREG)

USB System Control Register

偏移地址：0x00

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	USBHS_NFE	USBHS_NFS[1:0]		-	-	-	-	--	USBFS_NFE	USBFS_NFS[1:0]	

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	USBHS_FSPHYE	USBHS_SOFEN	USBHS_DFB	-	-	-	-	-	-	USBFS_SOFEN	USBFS_DFB

位	标记	位名	功能	读写
b31~b27	Reserved	-	必须保持复位值。	R/W
b26	USBHS_NFE	USBHS滤波使能寄存器	USBHS滤波使能寄存器 该寄存器用于控制STOP模式下，USBHS片上全速PHY DP/DM模拟滤波器的开关 0：模拟滤波器关 1：模拟滤波器开，滤波范围请参考USBHS_NFS[1:0]的设定	R/W
b25~b24	USBHS_NFS	USBHS滤波选择寄存器	USBHS滤波选择寄存器 该寄存器用于控制STOP模式下，USBHS片上全速PHY DP/DM模拟滤波器的滤波范围 00b：滤波宽度档位1 01b：滤波宽度档位2 10b：滤波宽度档位4 11b：滤波宽度档位4 各档位具体值请参考“电气特性 USB片上全速PHY STOP模式下滤波特性”章节。	R/W
b23~b19	Reserved	-	必须保持复位值。	R/W
b18	USBFS_NFE	USBFS滤波使能寄存器	USBFS滤波使能寄存器 该寄存器用于控制STOP模式下，USBFS片上全速PHY DP/DM模拟滤波器的开关 0：模拟滤波器关	R/W

			1：模拟滤波器开，滤波范围请参考USBFS_NFS[1:0]的设定	
b17~b16	USBFS_NFS	USBFS滤波选择寄存器	USBFS滤波选择寄存器 该寄存器用于控制STOP模式下，USBFS片上全速PHY DP/DM模拟滤波器的滤波范围 00b：滤波宽度档位1 01b：滤波宽度档位2 10b：滤波宽度档位4 11b：滤波宽度档位4 各档位具体值请参考“49.电气特性 49.3.34 USB片上全速PHY STOP模式下滤波特性”章节。	R/W
b15~b11	Reserved	-	必须保持复位值。	R/W
b10	USBHS_FSPHYE	USBHS 片上全速PHY使能位	USBHS 片上全速PHY使能位 0：USBHS片上全速PHY挂起 1：USBHS片上全速PHY使能	R/W
b9	USBHS_SOFEN	USBHS SOF脉冲输出使能位	USBHS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
b8	USBHS_DFB	USBHS VBUS/ID管脚内部去抖滤波器旁路使能位	USBHS 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
b7~b2	Reserved	-	必须保持复位值。	R/W
b1	USBFS_SOFEN	USBFS SOF脉冲输出使能位	USBFS主机发出SOF或者设备成功接收到SOF时，16个系统时钟周期宽度的SOF脉冲从PAD输出使能 0：SOF脉冲不输出 1：SOF脉冲输出 注意：在设备模式和主机模式均可访问。	R/W
b0	USBFS_DFB	USBFS VBUS/ID管脚内部去抖滤波器旁路使能位	USBFS VBUS/ID管脚模块内部去抖动滤波器旁路使能位 0：模块内部去抖动滤波器有效 1：旁路模块内部去抖动滤波器 注意：在设备模式和主机模式均可访问。	R/W

35.7.2 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。

35.7.2.1 USBFS VBUS 控制寄存器(USBFS_GVBUSCFG)

VBUS Configuration Register

偏移地址：0x00

复位值：0x00XX 0000

该寄存器可用于设定 VBUS 值从而忽略 VBUS 管脚的状态。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								VUBSVAL	VBUSOVEN	Reserved					

位	标记	位名	功能	读写
b31~b8	Reserved	-	必须保持复位值。	R/W
b7	VBUSVAL	VBUS值	VBUS值(VBUS Value) 用于设定USBFS的VBUS值，当设定为1，且VBUSOVEN设定1后对USBFS完成上电。 注意：仅可在设备模式下访问。	R/W
b6	VBUSOVEN	VBUS Override 使能	VBUS Override 使能(VBUS Override 使能) 用于将VBUSVAL设定的值反映到USBFS CORE的状态。仅当该位设置为1，VBUSVAL的值才有效。 注意：仅可在设备模式下访问。	R/W
b5~b0	Reserved	-	必须保持复位值。	R/W

35.7.2.2 USBFS AHB 控制寄存器(USBFS_GAHBCFG)

AHB Configuration Register

偏移地址：0x08

复位值：0x0000 0000

该寄存器可用于在上电后或更改角色模式时对模块进行配置。该寄存器主要包含 AHB 系统相关的配置参数。

应用程序必须在开始任何 AHB 或 USB 事务前对该寄存器进行编程。请勿在初始编程后更改该寄存器。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							PTXFELVL	TXFELVL	Reserved	DMAEN	HBSTLEN[3:0]				GINTMSK

位	标记	位名	功能	读写
b31~b9	Reserved	-	必须保持复位值。	R/W
b8	PTXFELVL	周期性TxFIFO空门限	周期性TxFIFO空门限 (Periodic TxFIFO empty level) 指示何时触发模块中断寄存器中的周期性TxFIFO空中断位（USBFS_GINTSTS中的PTXFE 位）。 0：PTXFE（位于USBFS_GINTSTS）中断指示周期性TxFIFO为半空状态 1：PTXFE（位于USBFS_GINTSTS）中断指示周期性TxFIFO为全空状态 注意：仅可在主机模式下访问。	R/W
b7	TXFELVL	设备TxFIFO空门限	设备TxFIFO空门限 (TxFIFO 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TxFIFO为半空状态 1：TXFE（位于USBFS_DIEPINTx）中断指示IN端点TxFIFO为全空状态 注意：仅可在设备模式下访问。	R/W
b6	Reserved	-	必须保持复位值。	R/W
b5	DMAEN	DMA 使能	DMA 使能 (DMA enable) 0：模块以从模式运行 1：模块以DMA模式运行 注意：在设备模式和主机模式均可访问。	R/W

b4~b1	HBSTLEN	批量长度/类型	批量长度/类型 (Burst length/type) 0000b:单次 0001b: INCR 0011b:INCR4 0101b:INCR8 0111b:INCR16 其它值：保留 注意：在设备模式和主机模式均可访问。	R/W
b0	GINTMSK	全局中断屏蔽	全局中断屏蔽 (Global interrupt mask) 该位用于屏蔽全局中断或对全局中断取消屏蔽。中断状态寄存器由模块进行更新，与此位的设置无关。 0：屏蔽应用程序触发的中断 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。	R/W

35.7.2.3 USBFS USB 配置寄存器 (USBFS_GUSBCFG)

USBFS USB configuration register

偏移地址：0x00C

复位值：0x0000 01440

该寄存器可用于在上电或更改角色模式后对模块进行配置。其中包含与 USB 和 USB-PHY 相关的配置参数。

应用程序必须在开始任何 AHB 或 USB 事务前对该寄存器进行编程。请勿在初始编程后更改该寄存器。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved		TRDT[3:0]				Reserved			PHYSEL	Reserved			TOCAL[2:0]		

位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R/W
b30	FDMOD	强制设备模式	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/W

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 × AHB 时钟 + 1 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/W
b6	PHYSEL	全速串行收发器选择	全速串行收发器选择 (Full Speed serial transceiver select)	R/W
b5~b3	Reserved	-	必须保持复位值。	R/W
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

35.7.2.4 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	CSRST

位	标记	位名	功能	读写
b31	AHBIDL	AHB主器件空闲	AHB主器件空闲(AHB master idle) 指示AHB主器件状态机处于空闲情况。 注意：在设备模式和主机模式均可访问。	R
b30	DMAREQ	AHB主器件空闲	DMA请求信号 (DMA request signal) 该位指示DMA请求正在进行中。用于调试。 注意：在设备模式和主机模式均可访问。	R
b29~b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10~b6	TXFNUM	TxFIFO编号	TxFIFO编号(TxFIFO number) 使用TxFIFO刷新位进行FIFO刷新的FIFO编号。只有在模块将TxFIFO刷新位清零后，方可更改此字段。 ● 00000： — 主机模式下刷新非周期性TxFIFO — 设备模式下刷新Tx FIFO 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/W
b2	FCRST	主机帧计数器复位	主机帧计数器复位(Host frame counter reset) 应用程序对该位执行写操作时，模块中的帧数计数器复位。帧计数器复位后，由模块发送的下一个SOF的帧号为 0。 注意：在设备模式和主机模式均可访问。	R/W
b1	HSRST	HCLK域逻辑软复位	HCLK域逻辑软复位 (HCLK soft reset) 应用程序使用此位来刷新 AHB 时钟域中的控制逻辑。仅复位 AHB 时钟域流水线。 FIFO 不通过此位来刷新。 遵照协议终止 AHB 上的事务后，AHB 时钟域中的所有状态机均复位至空闲状态。 AHB 时钟域状态机所使用的 CSR 控制位清零。 要清零该中断，需要将由 AHB 时钟域状态机生成并用于控制中断状态的状态屏蔽位清零。 由于中断状态位并未清零，因此应用程序可以获取在该位置 1 后所发生的所有模块事件的 状态。 此位为自清零位，模块将在其中所有必要逻辑复位后将该位清零。该过程需要若干个时钟的 时间，具体取决于模块的当前状态。 注意：在设备模式和主机模式均可访问。	R/W

b0	CSRST	模块软复位	模块软复位 (Core soft reset) 按如下所述将HCLK和PCLK域复位： 除以下各位外，将各个中断和所有CSR寄存器位清零： — USBFS_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。	R/W

35.7.2.5 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	VBUSVINT	DISCINT	CIDSCHG	LPMINT	PTXFE	HCINT	HPRTINT	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	ENUMDNE	USBRST	USBSUSP	ESUSP	Reserved	Reserved	GONAKEFF	GINAKEFF	NPTXFE	RXFLNE	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	VBUSVINT	VBUS有效中断	VBUS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	检测到断开连接中断 (Disconnect detected interrupt) 当检测到设备断开连接时触发该中断。 通过软件对该位写1清零。 注意：仅可在主机模式下访问。	R/W
b28	CIDSCHG	连接器ID线状态变化中断	连接器ID线状态更改(Connector ID status change) 当连接器ID线状态发生更改时，模块将该位置1。 通过软件对该位写1清零。 注意：在设备模式和主机模式均可访问。	R/W
b27	LPMINT	LPM中断	LPM中断 设备模式 设备收到正确LPM传输且有效应答时该位置1。 主机模式 主机发送LPM后收到有效应答，或者超过设定的重试次数后该位置1。	R/W
b26	PTXFE	周期性TxFIFO空中断	周期性TxFIFO空中断 (Periodic TxFIFO empty interrupt) 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	主机通道中断 (Host channels interrupt) 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	主机端口中断 (Host port interrupt) 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/W
b22	DATAFSUSP	数据获取挂起	数据获取挂起 (Data fetch suspended) 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	R/W

			— 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。为避免这种情况的发生，应用程序可以检查USBFS_GINTSTS 中的DATAFSUSP中断，该中断可确保在FIFO存满后再将全局 NAK 握手信号清零。或者，应用程序可以在将全局IN NAK握手信号清零时屏蔽“当 FIFO为空时接收到IN令牌中断。 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 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/W

b15	EOPF	周期性帧结束中断	周期性帧结束中断 (End of periodic frame interrupt) 指示当前帧已达到USBFS_DCFG 寄存器中周期性帧间隔字段（USBFS_DCFG 中的PFIVL位）所指定的周期。 通过软件对该位写1清零。 注意：仅可在设备模式下访问。	R/W
b14	ISOODRP	丢弃同步OUT数据包中断	丢弃同步OUT数据包中断(Isochronous OUT packet dropped interrupt) 如果由于RxFIFO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RxFIFO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/W
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TxFIFO空中断	非周期性TxFIFO空中断(Non-periodic TxFIFO empty interrupt) 当非周期性TxFIFO为半空或全空状态，且非周期性发送请求队列中至少存在可写入一个条目的空间时，将触发该中断。该 FIFO 为半空状态还是全空状态由USBFS_GAHBCFG寄存器中的非周期性TxFIFO空级别位（USBFS_GAHBCFG中的TXFELVL位）决定。 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/W
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： — 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会影响模块运行。 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

35.7.2.6 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	LPMINTM	PTXFEM	HCIM	HPRTIM	Reserved	DATAFSUSPM	IPXFRM/INCOMPISOOUTM	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	USBSUSPM	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 mak) 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	DISCM	检测到断开连接中断屏蔽	检测到断开连接中断屏蔽 (Disconnect detected interrupt interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在主机模式下访问。	R/W
b28	CIDSCHGM	中断连接器ID线状态变化中断屏蔽	连接器ID线状态更改中断屏蔽(Connector ID status change interrupt mask) 0：屏蔽中断 1：使能中断 注意：在设备模式和主机模式均可访问。	R/W

b27	LPMINTM	LPM中断屏蔽	LPM中断屏蔽（LPM interrupt mask） 0：屏蔽中断 1：使能中断	R/W
b26	PTXFEM	周期性TxFIFO空中断屏蔽	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/W
b22	DATAFSUSPM	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/ INCOMPISOOUTM	未完成周期性传输中断屏蔽/ 未完成OUT同步传输中断屏蔽	IPXFR:未完成周期性传输中断屏蔽(Incomplete periodic transfer interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在主机模式下访问。 INCOMPISOOUT: 未完成OUT同步传输中断屏蔽(Incomplete isochronous OUT transfer interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W
b20	IISOIXFRM	未完成IN同步传输中断屏蔽	未完成IN同步传输中断屏蔽(Incomplete isochronous IN transfer interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W
b19	OEPIM	OUT端点中断屏蔽	OUT端点中断屏蔽(OUT endpoint interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W

b18	IEPIM	IN端点中断屏蔽	IN端点中断屏蔽(IN endpoint interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W
b17~b16	Reserved	-	必须保持复位值。	R/W
b15	EOPFM	周期性帧结束中断屏蔽	周期性帧结束中断屏蔽(End of periodic frame interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W
b14	ISOODRPM	丢弃同步OUT数据包中断屏蔽	丢弃同步OUT数据包中断屏蔽(Isochronous OUT packet dropped interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W
b13	ENUMDNEM	枚举完成中断屏蔽	枚举完成中断屏蔽(Enumeration done interrupt mask) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W
b12	USBRSTM	USB复位中断屏蔽	USB复位中断屏蔽(USB reset 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
b11	USBSUSPM	USB挂起中断屏蔽	USB挂起中断屏蔽(USB suspend 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
b10	ESUSPM	早期挂起中断屏蔽	早期挂起中断屏蔽(Early suspend 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
b9~b8	Reserved	-	必须保持复位值。	R/W
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) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W
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) 0：屏蔽中断 1：使能中断 注意：仅可在设备模式下访问。	R/W

b5	NPTXFEM	非周期性TxFIFO空中断屏蔽	非周期性TxFIFO空中断(Non-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
b4	RXFNEM	RxFIFO 非空中断屏蔽	RxFIFO非空中断屏蔽(RxFIFO non-empty 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
b3	SOFM	帧起始中断屏蔽	帧起始中断屏蔽(Start of frame 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
b2	Reserved	-	必须保持复位值。	R/W
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) 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
b0	Reserved	-	必须保持复位值。	R/W

35.7.2.7 USBFS 接收状态调试读取/USBFS 状态读取和出栈寄存器(USBFS_GRXSTSR/USBFS_GRXSTSP)

USBFS Receive status debug read/USBFS status read and pop registers

读取的偏移地址：0x01C

出栈的偏移地址：0x020

复位值：0x0000 0000

读取接收状态调试读取寄存器将返回接收 FIFO 顶部的内容。读取接收状态读取和出栈寄存器将额外弹出 RxFIFO 顶部的数据条目。接收状态内容在主机模式和设备模式下的解释不同。

当接收 FIFO 为空时，模块会忽略对该寄存器的读取或出栈操作，并返回值 0x0000 0000。当模块中断寄存器的接收 FIFO 非空位（USBFS_GINTSTS 中的 RXFNE 位）置位时，应用程序必须仅弹出接收状态 FIFO。

主机模式：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved											PKTSTS[3:0]				DPID[1]

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DPID[0]	BCNT[11:0]											CHNUM[3:0]			

位	标记	位名	功能	读写
b31~b21	Reserved	-	必须保持复位值。	R/W
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：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved											PKTSTS[3:0]				DPID[1]

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DPID[0]	BCNT[11:0]											EPNUM[3:0]			

位	标记	位名	功能	读写
b31~b21	Reserved	-	必须保持复位值。	R/W
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

35.7.2.8 USBFS 接收 FIFO 大小寄存器 (USBFS_GRXFSIZ)

USBFS Receive FIFO size register

偏移地址：0x024

复位值：0x0000 0280

此应用程序可以对必须分配给 RxFIFO 的 RAM 大小进行编程。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					RXFD[10:0]										

位	标记	位名	功能	读写
b31~b11	Reserved	-	必须保持复位值。	R/W
b10~b0	RXFD	RxFIFO深度	RXFD：RxFIFO深度 (RxFIFO depth) 以32位字为单位。 最小值为16 最大值为256 上电复位值为最大Rx数据FIFO深度。	R/W

35.7.2.9 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：0x02800280

此应用程序可以对必须分配给 TxFIFO 的 RAM 大小进行编程。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
NPTXFD[15:0]/TX0FD[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
NPTXFSA[15:0]/TX0FSA[15:0]															

位	标记	位名	功能	读写
b31~b16	NPTXFD/ TX0FD	非周期性TxFIFO 深度/端点0 TxFIFO深度	主机模式：NPTXFD 非周期性TxFIFO深度(Non-periodic TxFIFO depth) 以32位字为单位。 最小值为16 最大值为256 设备模式：TX0FD 端点0 TxFIFO深度(Endpoint 0 TxFIFO depth) 以32位字为单位。 最小值为16 最大值为256	R/W
b15~b0	NPTXFSA/ TX0FSA	非周期性发送RAM起始地址/端点0发送RAM起始地址	主机模式：NPTXFSA 非周期性发送RAM起始地址(Non-periodic transmit RAM start address) 此字段包含非周期性发送FIFO RAM的存储器起始地址。 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。	R/W

35.7.2.10 USBFS 非周期性发送 FIFO/队列状态寄存器 (USBFS_HNPTXSTS)

USBFS Host non-periodic transmit FIFO size register/Device endpoint0 transmit FIFO size register

偏移地址：0x02C

复位值：0x00080280

此只读寄存器包含非周期性 TxFIFO 和非周期性发送请求队列的自由空间信息。

此寄存器仅在主机模式有效，设备模式无效。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved	NPTXQTOP[6:0]							NPTQXSAV[7:0]							

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

位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R/W
b30~b24	NPTXQTOP	非周期性发送请求队列顶部	非周期性发送请求队列顶部 (Top of the non-periodic transmit request queue) 非周期性发送请求队列中MAC目前正在处理的条目。 位30:27：通道/端点编号(Channel/endpoint number) 位26:25： — 00：IN/OUT 令牌 — 01：长度为零的发送数据包 — 11：通道停止命令 位24：结束（所选通道/端点的最后一个条目）(Terminate (last entry for selected channel))	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。 0：非周期性发送请求队列已满 1：1 个位置可用 2：2 个位置可用 n：n 个位置可用 (其中，n范围：0~8) 其它值：保留	R

b15~b0	NPTXFSAV	非周期性TxFIFO可用空间	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

35.7.2.11 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

35.7.2.12 USBFS LPM 配置寄存器(USBFS_GLPMCFG)

USBFS LPM configuration register

偏移地址：0x054

复位值：0x00000000

该寄存器为可编程 LPM 配置寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved			ENBSEL	LPMRCNTSTS[2:0]			SNDLPM	LPMRCNT[2:0]			LPMCHIDX[3:0]				L1RSMOK

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SLPSTS	LPMRSP[1:0]		L1DSEN	BESLTHRS[3:0]				L1SSEN	REMWAKE	BSEL[3:0]				LPMACK	LPMEN

位	标记	位名	功能	读写
b31~b29	Reserved	-	必须保持复位值。	R/W
b28	ENBSEL	BSEL使能位	BSEL使能位 0:遵循USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007 1：遵循Errata for USB 2.0 ECN: Link Power Management (LPM) - 7/2007	R/W
b27~b25	LPMRCNTSTS[2:0]	LPM重试状态寄存器	LPM重试状态寄存器 记录LPM剩余的重试次数，仅主机模式有效	R
b24	SENDLPM	发送LPM传输	发送LPM传输 0：不发送LPM传输 1：发送LPM传输（EXT 和 LPM 令牌） 当收到设备返回的应答（ACK,STALL或NYET）后该位自动清零。 仅主机模式支持。	R/W
b23~b21	LPMRCNT[2:0]	LPM重试次数寄存器	LPM重试次数寄存器 当收到ERROR应答时，主机发送LPM传输的重试次数,直到收到有效的应答（ACK,STALL或NYET） 仅主机模式支持。	R/W
b20~b17	LPMCHIDX[2:0]	LPM发送通道索引寄存器	LPM发送通道索引寄存器 LPM传输所用主机通道索引，硬件自动插入该通道设定的设备地址以及端点信息。 仅主机模式支持。	R/W
b16	L1RSMOK	L1状态位	L1状态位 0：当前状态不能从L1恢复	R/W

			1：当前状态可以从L1恢复	
b15	SLPSTS	sleep状态位	sleep状态位 设备模式 当设备发送ACK响应LPM传输并经过协议规定的时间后，该位置1，表示进入SLEEP模式。 当总线状态变化，或者断连，或者发送远程唤醒信号时，该位自动清0，退出SLEEP模式。 主机模式 主机发送LPM传输，接收到设备响应的ACK应答后，该位置1，表示进入SLEEP模式。 当主机接收到远程唤醒信号，或者主机发起唤醒，或者主机发起复位时，该位自动清0，退出SLEEP模式。 0：Not in L1 1：In L1	R/W
b14~b13	LPMRSP[1:0]	LPM应答	LPM应答 主机接收或者从机发送的LPM应答 00b：ERROR(无应答) 01b：STALL 00b：NYET 00b：ACK	R/W
b12	L1DSEN	L1 deep sleep使能位	L1 deep sleep使能位 最大限度的降低功耗，应用程序需要将此位设定为1。	R/W
b11~b8	BESLTHRS[3:0]	BSEL阈值设定寄存器	BSEL阈值设定寄存器 设备模式 当设备接收到BSEL值大于等于该寄存器设定时，进入L1。 主机模式 主机发送resume信号的时间。	R/W
b7	L1SSEN	L1 shallow sleep使能位	L1 shallow sleep使能位 最大限度的降低功耗，应用程序需要将此位设定为1。	R/W
b6	REMWAKE	bRemoteWake值	bRemoteWake值 主机发送或者设备接收到的bRemoteWake值	R/W
b5~b2	BSEL[3:0]	BSEL寄存器	BSEL寄存器 设备模式 BSEL bmAttribute值，有效应答LPM传输后自动更新。 主机模式 主机发送LPM传输的BSEL值，或者resume信号长度值。 0000b：125us 0001b：150us	R/W

			0010b：200us 0100b：400us 0101b：500us 0110b：1000us 0111b：2000us 1000b：3000us 1001b：4000us 1010b：5000us 1011b：6000us 1100b：7000us 1101b：8000us 1110b：9000us 1111b：10000us	
b1	LPMACK	LPM应答寄存器	LPM应答寄存器 1：ACK 即使该位写1，但LPM传输有错误时，不发送ACK应答。 存在PID/CRC错误，ERROR应答 bLinkState不等于0001b，STALL应答 存在挂起的数据传输，NYET应答 0：NYET 仅设备模式有效	R/W
b0	LPMEN	LPM使能位	LPM使能位 0：LPM禁止 1：LPM使能	R/W

35.7.2.13 USBFS 主机周期性发送 FIFO 大小寄存器(USBFS_HPTXFSIZ)

USBFS Host periodic transmit FIFO size register

偏移地址：0x100

复位值：0x02800500

此应用程序可以对必须分配给周期 TxFIFO 的 RAM 大小进行编程。

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

位	标记	位名	功能	读写
b31~b16	PTXFD	主机周期性TxFIFO深度	主机周期性TxFIFO深度 (Host periodic TxFIFO depth) 以32位字为单位。 最小值为16	R/W
b15~b0	PTXSA	主机周期性TxFIFO起始地址	主机周期性TxFIFO起始地址(Host periodic TxFIFO start address) 上电复位值是最大RxFIFO深度与最大非周期性TxFIFO深度之和。	R/W

35.7.2.14 USBFS 设备 IN 端点发送 FIFO 大小寄存器 (USBFS_DIEPTXFx)（x = 1..15）

USBFS device IN endpoint transmit FIFO size register

偏移地址：0x104+(x-1)*0x4

复位值：0x02800500+(x-1)*0x280

此应用程序可以对必须分配给设备 TxFIFO 的大小进行编程。

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

位	标记	位名	功能	读写
b31~b16	INEPTXFD	IN端点TxFIFO深度	设备IN端点TxFIFO深度 (Device IN endpoint TxFIFO depth) 以32位字为单位。 最小值为16	R/W
b15~b0	INEPTXSA	IN端点TxFIFOx RAM起始地址	IN端点TxFIFOx RAM起始地址 (IN endpoint FIFOx transmit RAM start address) 此字段包含IN端点发送FIFOx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

35.7.3 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。

35.7.3.1 USBFS 主机配置寄存器 (USBFS_HCFG)

USBFS Host configuration register

偏移地址：0x400

复位值：0x00000200

此寄存器将在上电后对模块进行配置。请勿在初始化主机后更改此寄存器。

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

位	标记	位名	功能	读写
b31~b3	Reserved	-	必须保持复位值。	R/W
b2	FSLSS	仅支持FS和LS	应用程序使用此位控制模块的枚举速度。使用此位，应用程序可使模块工作为FS主机，即使所连接的设备支持HS通信也是如此。请勿在初始编程后更改此字段。 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

35.7.3.2 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/W
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

35.7.3.3 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

35.7.3.4 USBFS 主机周期性发送 FIFO/队列状态寄存器(USBFS_HPTXSTS)

USBFS Host periodic transmit FIFO/queue status register

偏移地址：0x410

复位值：0x00080280

此只读寄存器包含周期性 TxFIFO 和周期性发送请求队列的空闲空间信息。

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

位	标记	位名	功能	读写
b31~b24	PTXQTOP	周期性发送请求队列顶部	周期性发送请求队列顶部 (Top of the periodic transmit request queue) 指示周期性Tx请求队列中MAC当前正在处理的项。 该寄存器用于调试。 位 31：奇数/偶数帧 (Odd/Even frame) — 0：以偶数帧发送 — 1：以奇数帧发送 位 30:27：通道/端点编号 (Channel number) 位 26:25：类型 (Type) — 00：输入/输出 — 01：零长度数据包 — 11：禁止通道命令 位 24：结束（所选通道的最后一个条目）(Terminate (last entry for the selected channel))	R
b23~b16	PTXQSAV	周期性发送请求队列可用空间	周期性发送请求队列可用空间 (Periodic transmit request queue space available) 指示可供写入的周期性发送请求队列的空闲位置的数量。该队列既包含IN请求，又包含OUT请求。 0：周期性发送请求队列已满 1：1个位置可用 2：2个位置可用 n：n个位置可用 (其中，n范围：0~8) 其它值：保留	R

b15~b0	PTXFSAVL	周期性发送数据FIFO可用空间	周期性发送数据FIFO可用空间 (Periodic transmit data FIFO space available) 指示可供写入的周期性 TxFIFO 的空闲位置的数量。 以32位字为单位 0：周期性TxFIFO已满 1：1个字可用 2：2个字可用 n：n个字可用（其中，n范围：0~PTXFD） 其它值：保留	R

35.7.3.5 USBFS 主机全体通道中断寄存器 (USBFS_HAINT)

USBFS Host all channels interrupt register

偏移地址：0x414

复位值：0x0000 0000

当通道上有事件发生时，主机全体通道中断寄存器会使用模块中断寄存器中的主机通道中断位（USBFS_GINTSTS 中的 HCINT 位）中断应用程序。每个通道对应 1 个中断位，最多有 16 个位。

当应用程序通过相应主机通道 x 中断寄存器清零中断时，该寄存器中的位也会清零。

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

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	必须保持复位值。	R/W
b15~b0	HAINT	通道中断	通道中断(Channel interrupt) 每个通道对应一位：通道0对应位0，通道15对应位15。	R/W

35.7.3.6 USBFS 主机全体通道中断屏蔽寄存器(USBFS_HAINTMSK)

USBFS Host all channels interrupt mask register

偏移地址：0x418

复位值：0x0000 0000

主机全体通道中断屏蔽寄存器与主机全体通道中断寄存器结合使用，进而在通道上发生事件时中断应用程序。

每个通道对应 1 个中断屏蔽位，最多有 16 个位。

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

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	HAINTM	通道中断屏蔽	通道中断屏蔽 (Channel interrupt mask) 0：屏蔽中断 1：使能中断 每个通道对应一位：通道0对应位0，通道15对应位15。	R/W

35.7.3.7 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/W
b18~b17	PSPD	端口速度	端口速度(Port speed) 指示连接到该端口的设备的速度。 00/11：保留 01：全速 10：低速	R
b16~b13	Reserved	-	必须保持复位值。	R/W
b12	PPWR	端口电源	端口电源(Port power) 应用程序使用该字段控制该端口的电源。由于本USBFS内置PHY不具备供电能力，所以此为设置为1时，通过USBFS_DRVVBUS使能外部USB电源芯片供电。 0：掉电 1：通电	R/W
b11~b10	PLSTS	端口线状态	指示 USB 数据线的当前逻辑电平 位11：USBFS_DM 的逻辑电平 位10：USBFS_DP 的逻辑电平	R

b9	Reserved	-	必须保持复位值。	R/W
b8	PRST	端口复位	端口复位(Port reset) 应用程序将该位置1时，会在该端口上启动复位序列。应用程序必须为复位周期定时，并在复位序列完成后将该位清零。 0：端口未处于复位状态 1：端口处于复位状态 应用程序必须将该位置1并最少保持 10 ms，以在端口上启动复位。	R/W
b7	PSUSP	端口挂起	端口挂起(Port suspend) 应用程序将此位置1以将此端口置于挂起模式。只有此位置1时，模块才会停止发送SOF。要停止PHY时钟，应用程序必须将端口时钟停止位置1，这会使能PHY的挂起输入引脚。 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/W
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

35.7.3.8 USBFS 主机通道 x 特性寄存器 (USBFS_HCCHARx)（x = 0..15）

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/W
b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 指示选择的传输类型。 00：控制 01：同步 10：批量 11：中断	R/W

b17	LSDEV	低速设备	低速设备 (Low-speed device) 此字段由应用程序置 1，表示此通道正在与一个低速设备进行通信。	R/W
b16	Reserved	-	必须保持复位值。	R/W
b15	EPDIR	端点方向	端点方向(Endpoint direction) 指示通信事务的方向是输入还是输出。 0：输出 1：输入	R/W
b14-b11	EPNUM	端点编号	端点编号 (Endpoint number) 指示要与该主机通道通信的 USB 设备的端点号。	R/W
b10-b0	MPSIZ	最大数据包大小	最大数据包大小 (Maximum packet size) 指示与该主机通道通信的设备端点的最大数据包大小。	R/W

35.7.3.9 USBFS 主机通道 x 中断寄存器 (USBFS_HCINTx)（x = 0..15）

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_HAINT) 寄存器，以获取主机通道 x 中断寄存器的准确通道编号。应用程序必须将该寄存器中的相应位清零，才能将 USBFS_HAIN 和 USBFS_GINTSTS 寄存器中的对应位清零。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					DTERR	FRMOR	BBERR	TXERR	Reserved	ACK	NAK	STALL	AHBERR	CHH	XFRC

位	标记	位名	功能	读写
b31~b11	Reserved	-	必须保持复位值。	R/W
b10	DTERR	数据切换错误	数据同步错误 (Data toggle error) 应用程序需通过写1清除该位。	R/W
b9	FRMOR	帧溢出错误	帧溢出错误(Frame overrun) 应用程序需通过写1清除该位。	R/W
b8	BBERR	串扰错误	串扰错误(Babble error) 产生串扰事件的典型原因是端点发送了一个数据包，但是数据包长度超过了端点的最大包长。 应用程序需通过写1清除该位。	R/W
b7	TXERR	通信事务错误	通信事务错误 (Transaction error) 指示 USB 上发生下列错误之一： CRC校验失败 超时 位填充错误 错误的EOP 应用程序需通过写1清除该位。	R/W
b6	Reserved	-	必须保持复位值。	R/W

b5	ACK	收到/发出ACK 响应	收到/发出ACK响应(ACK response received/transmitted interrupt) 应用程序需通过写1清除该位。	R/W
b4	NAK	收到NAK响应	收到NAK响应(NAK response received interrupt) 应用程序需通过写1清除该位。	R/W
b3	STALL	收到STALL响应	收到STALL响应(STALL response received interrupt	R/W
b2	AHBERR	AHB错误	AHB错误 (AHB error) 仅当处于内部DMA模式下且AHB读/写操作期间发生AHB错误时才生成此错误。应用程序可通过读取相应的 DMA 通道地址寄存器来获取错误地址。 应用程序需通过写1清除该位	R/W
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

35.7.3.10 USBFS 主机通道 x 中断屏蔽寄存器 (USBFS_HCINTMSKx)（x = 0..15）

USBFS Host channel-x interrupt mask register

偏移地址：0x50C + (通道编号 × 0x20)

复位值：0x0000 0000

该寄存器用于选择屏蔽主机通道中断。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					DTERRM	FRMORM	BBERRM	TXERRM	Reserved	ACKM	NAKM	STALLM	AHBERRM	CHHM	XFRCM

位	标记	位名	功能	读写
b31~b11	Reserved	-	必须保持复位值。	R/W
b10	DTERRM	数据切换错误中断屏蔽	数据同步错误屏蔽(Data toggle error mask) 0：屏蔽中断 1：使能中断	R/W
b9	FRMORM	帧溢出错误中断屏蔽	帧溢出屏蔽(Frame overrun mask) 0：屏蔽中断 1：使能中断	R/W
b8	BBERRM	串扰错误中断屏蔽	串扰错误屏蔽(Babble error mask) 0：屏蔽中断 1：使能中断	R/W
b7	TXERRM	通信事务错误中断屏蔽	通信事务错误屏蔽(Transaction error mask) 0：屏蔽中断 1：使能中断	R/W
b6	Reserved	-	必须保持复位值。	R/W
b5	ACKM	收到/发出ACK 响应中断屏蔽	ACK响应接收/发送中断屏蔽 (ACK response received/transmitted interrupt mask) 0：屏蔽中断 1：使能中断	R/W
b4	NAKM	收到NAK响应中断屏蔽	NAK响应接收中断屏蔽(NAK response received interrupt mask) 0：屏蔽中断 1：使能中断	R/W
b3	STALLM	收到STALL响应中断屏蔽	收到STALL响应(STALL response received interrupt mask) 0：屏蔽中断 1：使能中断	R/W
b2	AHBERRM	AHBERR中断屏蔽	AHBERR中断屏蔽 (AHB error mask)	R/W

			0：屏蔽中断 1：使能中断	
b1	CHHM	通道停止中断屏蔽	通道停止中断屏蔽 (Channel halted mask) 0：屏蔽中断 1：使能中断	R/W
b0	XFRCM	传输完成中断屏蔽	传输完成中断屏蔽 (Transfer completed mask) 0：屏蔽中断 1：使能中断	R/W

35.7.3.11 USBFS 主机通道 x 传输大小寄存器 (USBFS_ HCTSIZx)（x = 0..15）

USBFS Host channel-x transfer size register

偏移地址：0x510 + (通道编号 × 0x20)

复位值：0x0000 0000

该寄存器用于设定主机通道传输大小以及数据 PID。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	DPID[1:0]		PKTCNT[9:0]										XFRSIZ[18:16]		

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

位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R/W
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

35.7.3.12 USBFS 主机通道 xDMA 地址寄存器 (USBFS_HCDMAx)（x = 0..15）

USBFS Host channel-x DMA address register

偏移地址：0x514 + (通道编号 × 0x20)

复位值：0xXXXX XXXX

该寄存器用于设定主机 DMA 模式时 DMA 地址。

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

位	标记	位名	功能	读写
b31~b0	DMAADDR	DMA地址	DMA地址(DMA address) 此字段存储主机从设备端点获取数据或往设备端点发送数据所用DMA传输的存储器的地址。每次AHB传输结束，该寄存器都会递增。	R/W

35.7.4 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。

35.7.4.1 USBFS 设备配置寄存器(USBFS_DCFG)

USBFS Device configuration register

偏移地址：0x800

复位值：0x0820 0000

此寄存器在上电、执行某些控制命令或枚举后，会将模块配置为设备模式。请勿在初始编程后更改该寄存器。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved			PFIVL[1:0]		DAD[6:0]							Reserved	NZLSOHSK	DSPD[1:0]	

位	标记	位名	功能	读写
b31~b13	Reserved	-	必须保持复位值。	R/W
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/W

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

35.7.4.2 USBFS 设备控制寄存器(USBFS_DCTL)

USBFS Device control register

偏移地址：0x804

复位值：0x0000 0002

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				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/W
b11	POPRGDNE	上电编程完成	上电编程完成(Power-on programming done) 应用程序使用此位指示寄存器从掉电模式唤醒后已完成编程。	R/W
b10	CGONAK	清零全局OUT NAK	清零全局OUT NAK(Clear global OUT NAK) 对此位执行写操作会将全局OUT NAK清零。	R/W
b9	SGONAK	置位全局OUT NAK	置位全局OUT NAK(Set global OUT NAK) 对此位执行写操作会将全局OUT NAK置 1。 应用程序使用此位在所有OUT端点发送NAK握手信号。 应用程序只有确定模块中断寄存器中全局OUT NAK有效位（USBFS_GINTSTS中GONAKEFF位）已清零时，才可以将此位置 1。	R/W
b8	CGINAK	清零全局IN NAK	清零全局IN NAK (Clear global IN NAK) 对此位执行写操作会将全局 IN NAK 清零。	R/W
b7	SGINAK	置位全局IN NAK	置位全局IN NAK (Set global IN NAK) 对此字段执行写操作会将全局非周期性IN NAK置 1。应用程序使用此位使所有非周期性IN端点发送NAK握手信号。 应用程序只有确定模块中断寄存器中全局IN NAK有效位（USBFS_GINTSTS中GINAKEFF位）已清零时，才可以将此位置1。	R/W
b6~b4	Reserved	-	必须保持复位值。	R/W
b3	GONSTS	全局OUT NAK状态	全局OUT NAK状态(Global OUT NAK status) 0：将根据FIFO状态和NAK和STALL位设置发送握手信号。 1：无论RxFIFO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。 全速时，软断连的最小时间规定如下： 挂起状态：最小时间为1ms+2.5us 空闲状态：2.5us 非空闲或挂起状态：2.5us	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

35.7.4.3 USBFS 设备状态寄存器(USBFS_DSTS)

USBFS Device status register

偏移地址：0x808

复位值：0x0000 0002

此寄存器指示模块在出现 USB 相关事件时的状态。发生中断时，必须从设备全体中断(USBFS_DAINT) 寄存器读取发生中断的端点信息。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								LNSTS[1:0]		FNSOF[13:8]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FNSOF[7:0]								Reserved				EERR	ENUMSPD[1:0]		SUSPSTS

位	标记	位名	功能	读写
b31~b24	Reserved	-	必须保持复位值。	R/W
b23~b22	LNSTS	USB总线状态	USB总线状态 LNSTS[1]：D+的逻辑电平 LNSTS[0]：D-的逻辑电平	R
b21~b8	FNSOF	接收SOF的帧编号	接收SOF的帧编号 (Frame number of the received SOF)	R
b7~b4	Reserved	-	必须保持复位值。	R/W
b3	EERR	不定错误	不定错误(Erratic error) 模块将该位置1以报告任何不定错误。 由于不定错误，USBFS控制器会进入挂起状态，并且会USBFS_GINTSTS寄存器的早期挂起位（USBFS_GINTSTS中的ESUSP 位）生成一个中断。如果早期挂起中断是由不定错误触发，则应用程序只能执行软断开以恢复通信。	R
b2~b1	ENUMSPD	枚举速度	枚举速度(Enumerated speed) 指示USBFS控制器通过chirp序列检测速度后被枚举成的速度。 01：保留 10：保留 11：全速（PHY 时钟运行频率为 48 MHz） 其它值：保留	R

b0	SUSPSTS	挂起状态	挂起状态(Suspend status) 在设备模式下，只要在USB上检测到挂起状态，该位就会置1。当USB 总线上的空闲状态保持3ms，模块便会进入挂起状态。出现以下情况时，模块会退出挂起状态： — USB 数据线上有活动 — 应用程序对USBFS_DCTL寄存器的远程唤醒信号位（USBFS_DCTL中的RWUSIG位）执行写操作。	R

35.7.4.4 USBFS 设备 IN 端点通用中断屏蔽寄存器(USBFS_DIEPMSK)

USBFS Device IN endpoint common interrupt mask register

偏移地址：0x810

复位值：0x0000 0000

此寄存器与全体端点的各个 USBFS_DIEPINTx 寄存器配合使用，以便在每个 IN 端点上生成中断。通过对此寄存器的相应位执行写操作，可屏蔽 USBFS_DIEPINTx 寄存器中的 IN 端点中断。默认情况下，状态中断都被屏蔽。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved		NAKM	Reserved						INEPNEM	INEPNMM	TTXFEMSK	TOM	Reserved	EPDM	XFRCM

位	标记	位名	功能	读写
b31~b14	Reserved	-	必须保持复位值。	R/W
b13	NAKM	NAK中断屏蔽	NAK中断屏蔽（NAK mask） 0：屏蔽中断 1：使能中断	R/W
b12~b7	Reserved	-	必须保持复位值。	R/W
b6	INEPNEM	IN端点NAK有效中断屏蔽	IN端点NAK有效中断屏蔽 (IN endpoint NAK effective mask) 0：屏蔽中断 1：使能中断	R/W
b5	INEPNMM	EP不匹配时接收到IN令牌中断屏蔽	EP不匹配时接收到IN令牌中断屏蔽 (IN token received with EP mismatch mask) 0：屏蔽中断 1：使能中断	R/W
b4	TTXFEMSK	TxFIFO为空时接收到IN令牌中断屏蔽	TxFIFO为空时接收到IN令牌中断屏蔽 (IN token received when TxFIFO empty mask) 0：屏蔽中断 1：使能中断	R/W
b3	TOM	超时中断屏蔽（非同步端点）	超时中断屏蔽（非同步端点） (Timeout condition mask (Non-isochronous endpoints)) 0：屏蔽中断 1：使能中断	R/W
b2	Reserved	-	必须保持复位值。	R/W

b1	EPDM	端点禁止中断屏蔽	端点禁止中断屏蔽(Endpoint disabled interrupt mask) 0：屏蔽中断 1：使能中断	R/W
b0	XFRCM	传输完成中断屏蔽	传输完成中断屏蔽(Transfer completed interrupt mask) 0：屏蔽中断 1：使能中断	R/W

35.7.4.5 USBFS 设备 OUT 端点通用中断屏蔽寄存器(USBFS_DOEPMSK)

USBFS Device OUT endpoint common interrupt mask register

偏移地址：0x814

复位值：0x0000 0000

此寄存器与全体端点的各个 USBFS_DOEPINTx 寄存器配合使用，以便在每个 OUT 端点上生成中断。通过对此寄存器的相应位执行写操作，可屏蔽 USBFS_DOEPINTx 寄存器中的 OUT 端点中断。默认情况下，状态中断都被屏蔽。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved											OTEPDM	STUPM	Reserved	EPDM	XFRCM

位	标记	位名	功能	读写
b31~b5	Reserved	-	必须保持复位值。	R/W
b4	OTEPDM	端点禁止时接收到OUT令牌中断屏蔽	端点禁止时接收到OUT令牌中断屏蔽 (OUT token received when endpoint disabled mask) 仅适用于控制 OUT 端点。 0：屏蔽中断 1：使能中断	R/W
b3	STUPM	SETUP阶段完成中断屏蔽	SETUP阶段完成中断屏蔽(SETUP phase done mask) 仅适用于控制端点。 0：屏蔽中断 1：使能中断	R/W
b2	Reserved	-	必须保持复位值。	R/W
b1	EPDM	端点禁止中断屏蔽	端点禁止中断屏蔽(Endpoint disabled interrupt mask) 0：屏蔽中断 1：使能中断	R/W
b0	XFRCM	传输完成中断屏蔽	传输完成中断屏蔽(Transfer completed interrupt mask) 0：屏蔽中断 1：使能中断	R/W

35.7.4.6 USBFS 设备全体端点中断寄存器 (USBFS_DAINT)

USBFS Device OUT endpoint common interrupt mask register

偏移地址：0x818

复位值：0x0000 0000

当端点上发生有效事件时，USBFS_DAINT 寄存器将通过 USBFS_GINTSTS 寄存器中的设备OUT端点中断位或设备IN端点中断位（分别为 USBFS_GINTSTS 中的OEPINT或 IEPINT 位）来中断应用程序。每个端点对应一个中断位，OUT 端点和 IN 端点均最多有 16 个中断位。双向端点将使用相应的 IN 和 OUT 中断位。当应用程序将相应设备端点 x 中断寄存器(USBFS_DIEPINTx/USBFS_DOEPINTx)中的位置 1 和清零时，此寄存器中的相应位也将置 1 和清零。

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

位	标记	位名	功能	读写
b31~b16	OEPINT	OUT端点中断位	OUT端点中断位(OUT endpoint interrupt bits) 每个OUT端点对应一位： OUT端点0对应位16，而OUT端点5对应位21。	R/W
b15~b0	IEPINT	IN端点中断位	IN端点中断位(IN endpoint interrupt bits) 每个IN端点对应一位： IN端点0对应位0，而IN端点5对应位5。	R/W

35.7.4.7 USBFS 设备全体端点中断屏蔽寄存器 (USBFS_DAINTMSK)

USBFS Device all endpoints interrupt mask register

偏移地址：0x81C

复位值：0x0000 0000

USBFS_DAINTMSK 寄存器与设备端点中断寄存器结合使用，在设备端点上发生事件时中断应用程序。但是，与该中断相对应的 USBFS_DAINT 寄存器位仍会置 1。

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

位	标记	位名	功能	读写
b31~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~b0	IEPINTM	IN端点中断屏蔽位	IN端点中断屏蔽位(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

35.7.4.8 USBFS 设备 IN 端点 FIFO 空中断屏蔽寄存器(USBFS_DIEPEMPMSK)

USBFS Device IN endpoint FIFO empty interrupt mask register

偏移地址：0x834

复位值：0x0000 0000

此寄存器用于控制 IN 端点 FIFO 空中断的生成。

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	必须保持复位值。	R/W
b15~b0	INEPTXFEM	IN EP TxFIFO空中断屏蔽位	IN EP 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

35.7.4.9 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	TXFNUM[3:0]				STALL	Reserved	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上启动数据发送。 在此端点上触发以下任一中断之前，模块会将此位清零： — SETUP阶段完成 — 端点禁止 — 传输完成	R/W
b30	EPDIS	端点禁止	端点禁止(Endpoint disable) 即使在该端点上的传输完成之前，应用程序也可将此位置1，以停止端点上的数据发送。应用程序必须等到发生端点禁止中断后，才能将端点视为禁止端点。在端点禁止中断位置1前，模块会将此位清零。只有在该端点的端点使能位置1后，应用程序才可将该位置 1。	R/W
b29~b28	Reserved	-	必须保持复位值。	R/W
b27	SNAK	置位NAK位	置位NAK位(Set NAK) 对此位进行写操作会将端点的NAK位置1。 通过此位，应用程序可以控制端点上NAK握手信号的发送。模块也可在端点接收到SETUP数据包后将该端点的此位置1。	R/W
b26	CNAK	清零NAK位	清零NAK位(Clear NAK) 对此位进行写操作会将端点的NAK位清零。	R/W
b25~b22	TXFNUM	TxFIFO编号	TxFIFO编号(TxFIFO number) 该值设置为分配给IN端点0的FIFO编号。只能使用TX-FIFO0。	R/W

b21	STALL	STALL握手	STALL握手(STALL handshake) 应用程序只能将此位置1，端点接收到SETUP令牌时，模块会将此位清零。如果NAK位、全局IN NAK或全局OUT NAK与此位均置 1，则STALL位优先。	R/W
b20	Reserved	-	必须保持复位值。	R/W
b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 硬件设置为‘00’，表示控制类型的端点。	R
b17	NAKSTS	NAK状态	NAK状态(NAK status) 指示以下结果： 0：模块根据FIFO状态回复非NAK握手。 1：模块在此端点上回复NAK握手。 当此位置1时（无论是被应用程序还是被模块），即使TxFIFO中仍有数据可用，模块也会停止发送数据。无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R
b16	Reserved	-	必须保持复位值。	R/W
b15	USBAEP	USB活动端点	USB活动端点(USB active endpoint) 此位总是置1，指示在所有配置和接口中控制端点0始终处于激活状态。	R
b14~b2	Reserved	-	必须保持复位值。	R/W
b1~b0	MPSIZ	最大数据包大小	最大数据包大小(Maximum packet size) 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

35.7.4.10 USBFS 设备 IN 端点 x 控制寄存器(USBFS_DIEPCTLx)(x=1..15)

USBFS Device IN endpoint x control register

偏移地址：0x900 + (端点编号 × 0x20)

复位值：0x0000 0000

应用程序使用此寄存器控制各个逻辑端点（端点 0 除外）的行为。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
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： — 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/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
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 DPID：端点数据PID (Endpoint data PID) 仅适用于中断/批量IN端点。 包含此端点上将要接收或发送的数据包的PID。端点激活后，应用程序必须对要在此端点上接收或发送的首个数据包的PID进行编程。应用程序使用SD0PID寄存器字段对DATA0或DATA1 PID进行编程。 0：DATA0 1：DATA1	R/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

35.7.4.11 USBFS 设备 IN 端点 x 中断寄存器 (USBFS_DIEPINTx) (x=0..15)

USBFS Device IN endpoint x interrupt register

偏移地址：0x908 + (端点编号 × 0x20)

复位值：0x0000 0080

此寄存器指示端点在出现 USB 和 AHB 相关事件时的状态。当模块中断寄存器中的 IN 端点中断位（USBFS_GINTSTS 中的 IEPINT 位）置 1 时，应用程序必须读取此寄存器。在应用程序能够读取此寄存器之前，必须先读取设备全体端点中断(USBFS_DAINT)寄存器，以获取设备端点 x 中断寄存器的准确端点编号。应用程序必须将此寄存器中的相应位清零，才能将 USBFS_DAINT 和 USBFS_GINTSTS 寄存器中的对应位清零。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TXFE	INEPNE	Reserved	TTXFE	TOC	Reserved	EPDISD	XFRC

位	标记	位名	功能	读写
b31~b8	Reserved	-	必须保持复位值。	R/W
b7	TXFE	发送FIFO为空	发送FIFO为空 (Transmit FIFO empty) 当此端点的TxFIFO为半空或全空时，此中断被置位。TxFIFO为半空还是全空状态由USBFS_GAHBCFG寄存器中的TxFIFO空白别位（USBFS_GAHBCFG中的TXFELVL位）决定。	R
b6	INEPNE	IN端点NAK有效	INEPNE：IN端点NAK有效 (IN endpoint NAK effective) 当应用程序通过向USBFS_DIEPCTLx 中的CNAK位写入数据来将IN端点NAK清零时，此位可被清零。 该中断指示模块已对（由应用程序或模块）置1的NAK采样，结果已生效。该中断指示由应用程序置1的IN端点NAK位已在模块中起作用。 此中断不保证在USB上发送了NAK握手信号。 STALL位的优先级高于 NAK 位。 软件写1也可将此位清零。	R/W
b5	Reserved	-	必须保持复位值。	R/W

b4	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/W
b1	EPDISD	端点禁止中断	端点禁止中断 (Endpoint disabled interrupt) 此位指示该端点已经由应用程序禁止掉。 通过软件写1清零。	R/W
b0	XFRC	传输完成中断	传输完成中断 (Transfer completed interrupt) 此字段指示在此端点上设置的传输已经在USB和AHB上传输完成。 通过软件写1清零。	R/W

35.7.4.12 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 1~15 的寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved											PKTCNT[1:0]		Reserved		

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved									XFRSIZ[6:0]						

位	标记	位名	功能	读写
b31~b21	Reserved	-	必须保持复位值。	R/W
b20~b19	PKTCNT	数据包计数	数据包计数 (Packet count) 指示端点0的一次数据传输包含的数据包个数。 每次从TxFIFO读取数据包（最大大小或短数据包）时，此字段将递减。	R/W
b18~b7	Reserved	-	必须保持复位值。	R/W
b6~b0	XFRSIZ	传输大小	传输大小(Transfer size) 指示端点0的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。 每次向TxFIFO写入来自外部存储器的数据包时，模块会使此字段递减。	R/W

35.7.4.13 USBFS 设备 IN 端点 x 传输大小寄存器(USBFS_DIEPTSIZx)(x=1..15)

USBFS Device IN endpoint x transfer size register

偏移地址：0x910 + (端点编号 × 0x20)

复位值：0x0000 0000

在使能该端点之前，应用程序必须修改此寄存器。通过 USBFS_DIEPCTLx 寄存器中的端点使能位（USBFS_DIEPCTLx 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	MCNT[1:0]		PKTCNT[9:0]										XFRSIZ[18:16]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XFRSIZ[15:0]															

位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R/W
b30~b29	MCNT	多重计数	多重计数 (Multi count) 对于周期性IN端点，此字段指示在USB上每帧必须发送的数据包数。模块使用此字段计算同步IN端点的数据PID。 01：1个数据包 10：2个数据包 11：3 个数据包	R/W
b28~b19	PKTCNT	数据包计数	数据包计数 (Packet count) 指示该端点上的一次数据传输包含的数据包个数。 每次从TxFIFO读取数据包（最大大小或短数据包）时，此字段将递减。	R/W
b18~b0	XFRSIZ	传输大小	传输大小(Transfer size) 此字段包含当前端点的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。 每次向TxFIFO写入来自外部存储器的数据包时，模块会使此字段递减。	R/W

35.7.4.14 USBFS 设备 IN 端点 x DMA 地址寄存器（USBFS_DIEPDMAx)(x=0..15)

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

35.7.4.15 USBFS 设备 IN 端点发送 FIFO 状态寄存器（USBFS_ DTXFSTSx)(x=0..15)

USBFS Device IN endpoint transmit FIFO status register

偏移地址：0x918 + (端点编号 × 0x20)

复位值：0x0000 0280

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	必须保持复位值。	R/W
b15~b0	INEPTFSAV	IN端点TxFIFO可用空间	IN端点TxFIFO可用空间 (IN endpoint TxFIFO space available) 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	R

35.7.4.16 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/W
b27	SNAK	置位NAK位	置位NAK位(Set NAK) 对此位进行写操作会将端点的NAK位置1。 通过此位，应用程序可以控制端点上NAK握手信号的发送。模块也可在端点接收到SETUP数据包后将该端点的此位置1。	R/W
b26	CNAK	清零NAK位	清零NAK位(Clear NAK) 对此位进行写操作会将端点的NAK位清零。	R/W
b25~b22	Reserved	-	必须保持复位值。	R/W
b21	STALL	STALL握手	STALL握手(STALL handshake) 此端点接收到SETUP令牌时，应用程序只能将此位置1，而模块会将其清零。 如果NAK位、全局OUT NAK与此位同时置1，则STALL位优先。无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R/W
b20	SNPM	监听模式	监听模式 (Snoop mode) 此位用于将端点配置为监听模式。在监听模式下，模块不会在将OUT数据包传输到应用存储区前检查其是否正确。	R/W

b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 硬件设置为‘00’，表示控制类型的端点。	R/W
b17	NAKSTS	NAK状态	NAK状态(NAK status) 指示以下结果： 0：模块根据FIFO状态回复非NAK握手。 1：模块在此端点上回复NAK握手。 当应用程序或模块将此位置1时，即使RxFIFO中存在空间可继续容纳收到的数据包，模块也会停止接收数据。无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R
b16	Reserved	-	必须保持复位值。	R/W
b15	USBAEP	USB活动端点	USB活动端点(USB active endpoint) 此位总是置1，指示在所有配置和接口中控制端点0始终处于激活状态。	R
b14~b2	Reserved	-	必须保持复位值。	R/W
b1~b0	MPSIZ	最大数据包大小	最大数据包大小(Maximum packet size) 控制OUT端点0的最大数据包大小与在控制IN端点0中进行编程的值相同。 00：64 字节 01：32 字节 10：16 字节 11：8 字节	R/W

35.7.4.17 USBFS 设备 OUT 端点 x 控制寄存器(USBFS_DOEPCTLx)(x=1..15)

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/W
b21	STALL	STALL握手	STALL握手(STALL handshake) 此端点接收到SETUP令牌时，应用程序只能将此位置1，而模块会将其清零。如果NAK位、全局OUT NAK与此位同时置1，则STALL位优先。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 DPID：端点数据PID (Endpoint data PID) 仅适用于中断/批量OUT端点。 包含此端点上将要接收或发送的数据包的PID。端点激活后，应用程序必须对要在此端点上接收或发送的首个数据包的PID进行编程。应用程序使用SD0PID 寄存器字段对 DATA0或DATA1 PID 进行编程。 0：DATA0 1：DATA1	R/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

35.7.4.18 USBFS 设备 OUT 端点 x 中断寄存器 (USBFS_DOEPINTx) (x=0..15)

USBFS Device OUT endpoint x interrupt register

偏移地址：0xb08 + (端点编号 × 0x20)

复位值：0x0000 0080

此寄存器指示端点在出现 USB 和 AHB 相关事件时的状态。当 USBFS_GINTSTS 寄存器中的 OUT 端点中断位（USBFS_GINTSTS 中的 OEPINT 位）置 1 时，应用程序必须读取此寄存器。在应用程序能够读取此寄存器之前，必须先读取 USBFS_DAINT 寄存器，以获取 USBFS_DOEPINTx 寄存器的准确端点编号。应用程序必须将此寄存器中的相应位清零，才能将 USBFS_DAINT 和 USBFS_GINTSTS 寄存器中的对应位清零。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved									B2BSTUP	Reserved	OTEPDIS	STUP	Reserved	EPDISD	XFRC

位	标记	位名	功能	读写
b31~b7	Reserved	-	必须保持复位值。	R/W
b6	B2BSTUP	接收到连续的SETUP数据包	接收到连续的SETUP数据包 (Back-to-back SETUP packets received) 仅适用于控制OUT端点。此位指示该端点已接收到三个以上的连续SETUP数据包。软件写1也可将此位清零。	R/W
b5	Reserved	-	必须保持复位值。	R/W
b4	OTEPDIS	端点禁止时接收到OUT令牌	端点禁止时接收到OUT令牌 (OUT token received when endpoint disabled) 仅适用于控制OUT端点。指示在尚未使能端点时接收到OUT令牌，从而产生中断。通过软件写1清零。	R/W
b3	STUP	SETUP阶段完成	SETUP 阶段完成 (SETUP phase done) 仅适用于控制OUT端点。指示控制端点的SETUP阶段已完成，当前控制传输中不再接收到连续的 SETUP数据包。在此中断上，应用程序可以对接收到的SETUP数据包进行解码。 通过软件写1清零。	R/W
b2	Reserved	-	必须保持复位值。	R/W
b1	EPDISD	端点禁止中断	端点禁止中断 (Endpoint disabled interrupt) 此位指示该端点已经由应用程序禁止掉。 通过软件写1清零。	R/W

b0	XFRC	传输完成中断	传输完成中断 (Transfer completed interrupt) 此字段指示在此端点上设置的传输已经在USB和AHB上传输完成。 通过软件写1清零。	R/W

35.7.4.19 USBFS 设备 OUT 端点 0 传输大小寄存器(USBFS_DOEPTSIZ0)

USBFS Device OUT endpoint 0 transfer size register

偏移地址：0xB10

复位值：0x0000 0000

在使能端点 0 之前，应用程序必须修改此寄存器。通过设备控制端点 0 控制寄存器中的端点使能位（USBFS_DIEPCTL0 中的 EPENA）使能端点 0 后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

非零端点使用端点 1~15 的寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved	STUPCNT[1:0]		Reserved									PKTCNT	Reserved		

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved									XFRSIZ[6:0]						

位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R/W
b30~b29	STUPCNT	SETUP数据包计数	SETUP数据包计数 (SETUP packet count) 此字段指定端点能连续接收的SETUP数据包数量。 01：1个数据包 10：2个数据包 11：3个数据包	R/W
b28~b20	Reserved	-	必须保持复位值。	R/W
b19	PKTCNT	数据包计数	数据包计数 (Packet count) 一次传输中应该接收到数据包数量。 在端点使能前，软件设置该位，在传输开始后，每当数据包接收到后，该域数值自动减少。	R/W
b18~b7	Reserved	-	必须保持复位值。	R/W
b6~b0	XFRSIZ	传输大小	传输大小(Transfer size) 指示端点0的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。 每次从RxFIFO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

35.7.4.20 USBFS 设备 OUT 端点 x 传输大小寄存器(USBFS_DOEPTSIZx)(x=1..15)

USBFS Device OUT endpoint x transfer size register

偏移地址：0xB10 + (端点编号 × 0x20)

复位值：0x0000 0000

在使能该端点之前，应用程序必须修改此寄存器。通过 USBFS_DOEPCTLx 寄存器中的端点使能位（USBFS_DOEPCTLx 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	RXDPID[1:0]/ STUPCNT[1:0]		PKTCNT[9:0]										XFRSIZ[18:16]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XFRSIZ[15:0]															

位	标记	位名	功能	读写
b31	Reserved	-	必须保持复位值。	R/W
b30~b29	RXDPID/ STUPCNT	接收到的数据PID/ SETUP数据包计数	接收到的数据PID (Received data PID) 仅适用于同步 OUT 端点。 这是此端点收到的上一个数据包的 PID。 00：DATA0 01：DATA2 10：DATA1 11：MDATA STUPCNT：SETUP数据包计数 (SETUP packet count) 仅适用于控制OUT端点。 此字段指定端点能连续接收的SETUP数据包数量。 01：1 个数据包 10：2 个数据包 11：3 个数据包	R/W
b28~b19	PKTCNT	数据包计数	数据包计数 (Packet count) 指示该端点上的一次数据传输包含的数据包个数。 每次向RxFIFO写入数据包（最大大小或短数据包）后，此字段将递减。	R/W
b18~b0	XFRSIZ	传输大小	传输大小(Transfer size) 此字段包含当前端点的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这 些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。 每次从RxFIFO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

35.7.4.21 USBFS 设备 OUT 端点 x DMA 地址寄存器（USBFS_DOEPDMAx)(x=0..15)

USBFS Device OUT endpoint x transfer size register

偏移地址：0xB14 + (端点编号 × 0x20)

复位值：0xXXXX 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

35.7.5 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。

35.7.5.1 USBFS 时钟门控控制寄存器(USBFS_GCCTL)

偏移地址：0xE00

复位值：0x0000 0000

此寄存器在主机模式和设备模式下均可用。

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								SUSP	PHYSLEEP	ENL1GTG	Reserved			GATEHCLK	STPPCLK

位	标记	位名	功能	读写
b31~b8	Reserved	-	必须保持复位值。	R/W
b7	SUSP	Deep Sleep状态位	Deep Sleep（L1 suspended）状态位	R
b6	PHYSLEEP	PHY SLEEP模式状态位	PHY SLEEP模式状态位（PHY in sleeep） 该位用于表示PHY处于SLEEP模式状态。	R
b5	ENL1GTG	L1时钟门控	L1时钟门控（Enable Sleep Clock Gating） 该位用于L1模式时时钟门控控制。	R/W
b4~b2	Reserved	-	必须保持复位值。	R/W
b1	GATEHCLK	门控HCLK	门控HCLK (Gate HCLK) 当USB通信挂起或会话无效时，应用程序会将此位置 1，以停止对除AHB总线从接口、主接口和唤醒逻辑之外的模块提供时钟。当USB恢复通信或新会话启动时，应用程序将此位清零。	R/W
b0	STPPCLK	停止PHY时钟	停止PHY时钟(Stop PHY clock) 当USB通信挂起、会话无效或设备断开连接时，应用程序将此位置1以停止PH时钟。当USB恢复通信时，应用程序将此位清零。	R/W

36 CAN FD 控制器（CAN FD）

36.1 简介

在配置 CAN FD 功能的产品中，CAN FD 控制器对应 CAN 控制器通道 2（CAN_2）。具体请参考型号功能对比表。

CAN（Controller Area Network）总线是一种可以在无主机情况下实现微处理器或者设备之间相互通信的总线标准。

CAN FD 控制器遵循 CAN 总线 CAN2.0(2.0A、CAN2.0B）和 CAN FD 协议。

CAN 总线控制器可以处理总线上的数据收发，在本产品中，CAN FD 控制器具有 16 组筛选器。筛选器用于为应用程序选择要接收的消息。

CAN FD 控制器中应用程序可通过 1 个高优先级的主发送缓冲器（Primary Transmit Buffer，以下简称 PTB）和 3 个辅发送缓冲器（Secondary Transmit Buffer，以下简称 STB）将发送数据送至总线，由发送调度器决定邮箱发送顺序。通过 8 个接收缓冲器（Receive Buffer，以下简称 RB）获取总线数据。3 个 STB 以及 8 个 RB 可以理解为一个 3 级 FIFO 和一个 8 级 FIFO，FIFO 完全由硬件控制。

CAN FD 总线控制器同时也可以支持时间触发 CAN 通信（Time-trigger communication）。

CAN FD 主要特性：

- 完全支持 CAN2.0A/CAN2.0B/CAN FD 协议。
- CAN2.0 支持最高通信波特率 1Mbit/s
- 支持 1~1/256 的波特率预分频，灵活配置波特率。
- 8 个接收缓冲器
  - FIFO 方式
  - 错误或者不被接收的数据不会覆盖存储的消息
- 1 个高优先主发送缓冲器 PTB
- 3 个副发送缓冲器 STB
  - FIFO 方式
  - 优先级仲裁方式

- 16 组独立的筛选器
  - 支持 11 位标准 ID 和 29 位扩展 ID
  - 可编程 ID CODE 位以及 MASK 位
- PTB/STB 均支持支持单次发送模式
- 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

36.2 CAN FD 系统框图

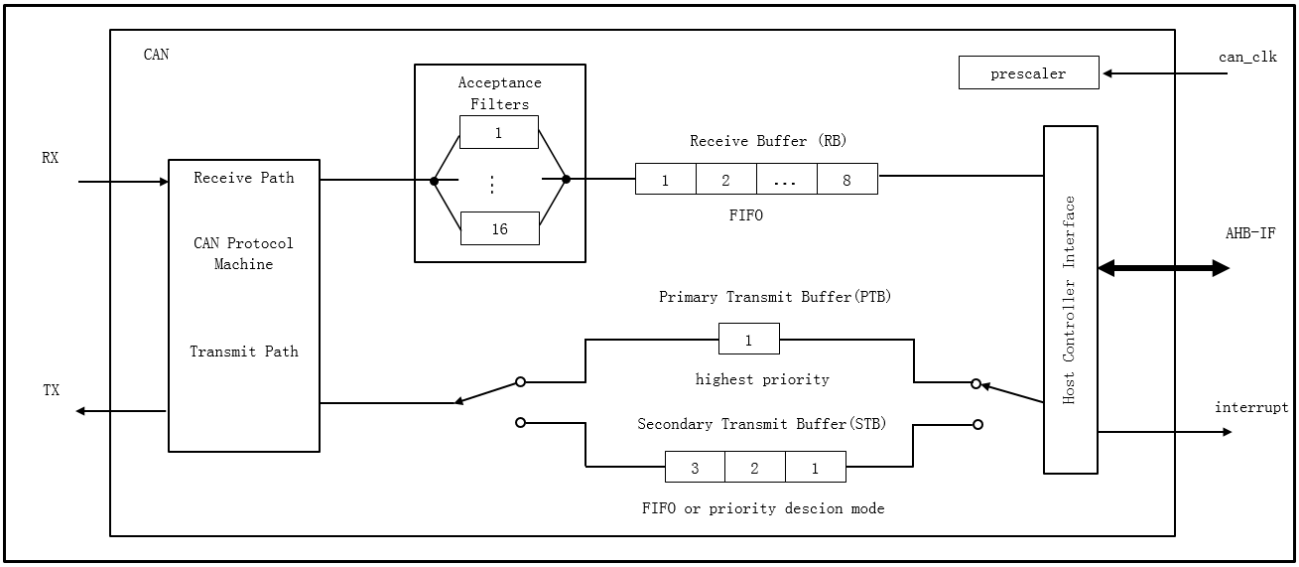


图 36-1 CANFD 系统框图

36.3 管脚说明

管脚名	方向	功能描述
CAN2_RX	输入	CAN接收数据信号
CAN2_TX	输出	CAN发送数据信号
CAN2_TST_SAMPLE	输出	仅用于观测，采样位置（采样点后一周期的通信时钟宽度）
CAN2_TST_CLOCK	输出	仅用于观测，波特率（一位开始前一周期的通信时钟宽度）

表 36-1 CAN 管脚说明

36.4 功能说明

36.4.1 动作模式

CAN FD 控制器存在两个操作模式，复位模式（CAN_CFG_STAT.RESET=1）和动作模式（CAN_CFG_STAT.RESET=0）。模块初始化时，首先应该在复位模式中设定只能在复位模式下操作的寄存器（详见寄存器说明章节），然后退后复位模式，在动作模式中操作其余寄存器。

36.4.2 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 SBT 和 FBT 设定。SBT 寄存器和 FBT 寄存器只能在 CAN_CFG_STAT.RESET=1 即 CAN 软件复位时设定。SBT 寄存器用于 CAN2.0 和 CAN FD 的仲裁段，FBT 寄存器用于 CAN FD 数据段。

FD 通信时，建议通信时钟选取 20MHz/40MHz/80MHz。

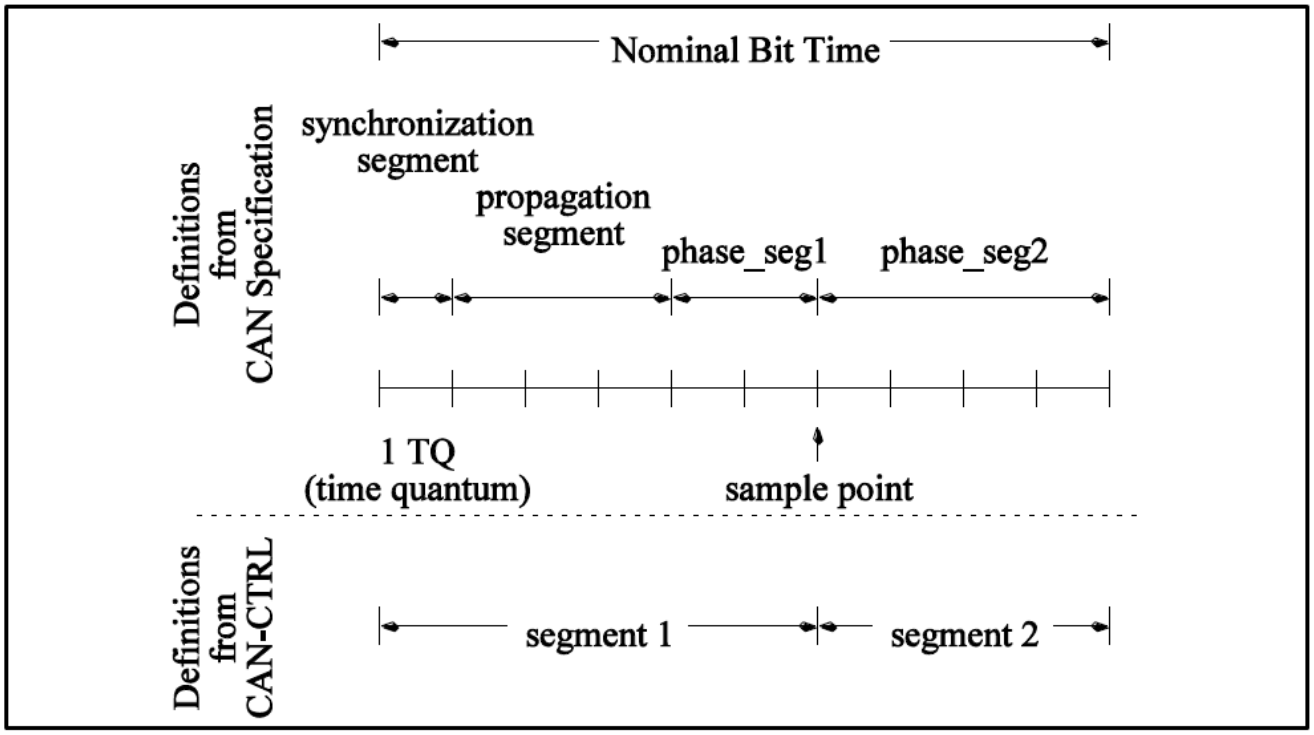


图 36-2 CAN 位时间定义图

TQ 计算方法请参考以下公式，其中 PRESC 通过 SBT 寄存器的 S_PRESC 位或 FBT 寄存器的 F_PRESC 位设定。f_{can_clk}为 CAN 通信时钟频率。

$$S_TQ = \frac{S_PRESC+1}{f_{can_clk}}$$

$$F_TQ = \frac{F_PRESC+1}{f_{can_clk}}$$

位时间计算方法请参考以下公式，其中 S_segment1 和 S_segment2 通过 SBT 寄存器的 S_SEG_1 位和 S_SEG_2 位设定，F_segment1 和 F_segment2 通过 FBT 寄存器的 F_SEG_1 位和 F_SEG_2 位设定。

$$Slow_BT = t_{S_segment1} + t_{S_segment2} = \big((S_SEG_1 + 2) + (S + SEG_2 + 1)\big) \times S_TQ$$

$$Fast_BT = t_{F_segment1} + t_{F_segment2} = \big((F_SEG_1 + 2) + (F + SEG_2 + 1)\big) \times F_TQ$$

位	设定范围			规则
SBT寄存器的S_SEG_1位	[0..63]	CAN2.0 bits	(slow)	SEG_1 ≥ SEG_2 + 1 SEG_2 ≥ SJW
	[0..63]	CAN FD nominal bits	(slow)	
SBT寄存器的S_SEG_2位	[0..7]	CAN2.0 bits	(slow)	
	[0..31]	CAN FD nominal bits	(slow)	

SBT寄存器的S_SJW位	[0..15]	CAN2.0 bits	(slow)	
	[0..15]	CAN FD nominal bits	(slow)	
FBT寄存器的F_SEG_1位	[0..15]	CAN FD data bits	(fast)	SEG_1 ≥ SEG_2 SEG_2 ≥ SJW
FBT寄存器的F_SEG_2位	[0..7]	CAN FD data bits	(fast)	
FBT寄存器的F_SJW位	[0..7]	CAN FD data bits	(fast)	

表 36-2 CAN 位时间设定规则

以下给出 CANFD 的波特率设定推荐，仅供参考。

PSP：Primary Sample Point

SSP：Secondary Sample Point

Seg 1：Segment 1

Seg 2：Segment 2

TDC：Transmitter Delay Compensation

Bit Rate [Mbit/s]	PSP [%]	SSP [%]	Prescaler	Bit Time [TQ]	Seg 1 [TQ]	Seg 2 [TQ]	SJW [TQ]	TDC [CAN 通信时钟]
0.25（仲裁）	80	-	1	80	64	16	16	-
0.5（仲裁）	80	-	1	40	32	8	8	-
0.5	80	禁止	1	40	32	8	8	-
1	80	80	1	20	16	4	4	16
2	80	80	1	10	8	2	2	8
4	80	80	1	5	4	1	1	4
5	75	75	1	4	3	1	1	3

表 36-3 20MHz 通信时钟时波特率设定建议

Bit Rate [Mbit/s]	PSP [%]	SSP [%]	Prescaler	Bit Time [TQ]	Seg 1 [TQ]	Seg 2 [TQ]	SJW [TQ]	TDC [CAN 通信时钟]
0.25（仲裁）	80	-	2	80	64	16	16	-
0.5（仲裁）	80	-	1	80	64	16	16	-
0.5	80	禁止	2	40	32	8	8	-
1	80	80	1	40	32	8	8	32
2	80	80	1	20	16	4	4	16
4	80	80	1	10	8	2	2	8
5	75	75	1	8	6	2	2	6
8	80	80	1	5	4	1	1	4

表 36-4 40MHz 通信时钟时波特率设定建议

Bit Rate [Mbit/s]	PSP [%]	SSP [%]	Prescaler	Bit Time [TQ]	Seg 1 [TQ]	Seg 2 [TQ]	SJW [TQ]	TDC [CAN 通信时钟]
0.25（仲裁）	80	-	4	80	64	16	16	-
0.5（仲裁）	80	-	2	80	64	16	16	-
0.5	80	禁止	4	40	32	8	8	-
1	80	80	2	40	32	8	8	64
2	80	80	2	20	16	4	4	32
4	80	80	1	20	16	4	4	16
5	75	75	1	16	12	4	4	12
8	80	80	1	10	8	2	2	8

表 36-5 80MHz 通信时钟时波特率设定建议

36.4.3 发送缓冲器

CAN_CTRL 提供两种发送缓冲器用于发送数据，主发送数据缓冲器 PTB 和副发送缓冲器 STB。PTB 具有最高的优先级，但只能缓冲一帧数据。STB 优先级比 PTB 低，但可以缓冲 3 帧数据，且 STB 内 3 帧数据可以工作在 FIFO 模式或者优先级仲裁模式。STB 中的 3 帧说数据可以通过 TCMD 寄存器的 TSALL 位设定为 1 全部发送，在 FIFO 模式下，最先写入的数据先发送，在优先级模式下，ID 小的数据先发送。

PTB 中的数据具有最高优先级，所以 PTB 发送能推迟 STB 发送，但是已经赢得仲裁并开始发送的 STB 不能够被 PTB 发送推迟。

PTB 和 STB 可以通过 TBUF 寄存器进行访问。通过 TCMD 寄存器的 TBSEL 位选择 PTB 或者 STB，TBSEL=0，选择 PTB，TBSEL=1，选择 STB。通过 TCTRL 寄存器的 TSNEXT 位选择 STB 中的下一个 SLOT。对应关系如下图所示：

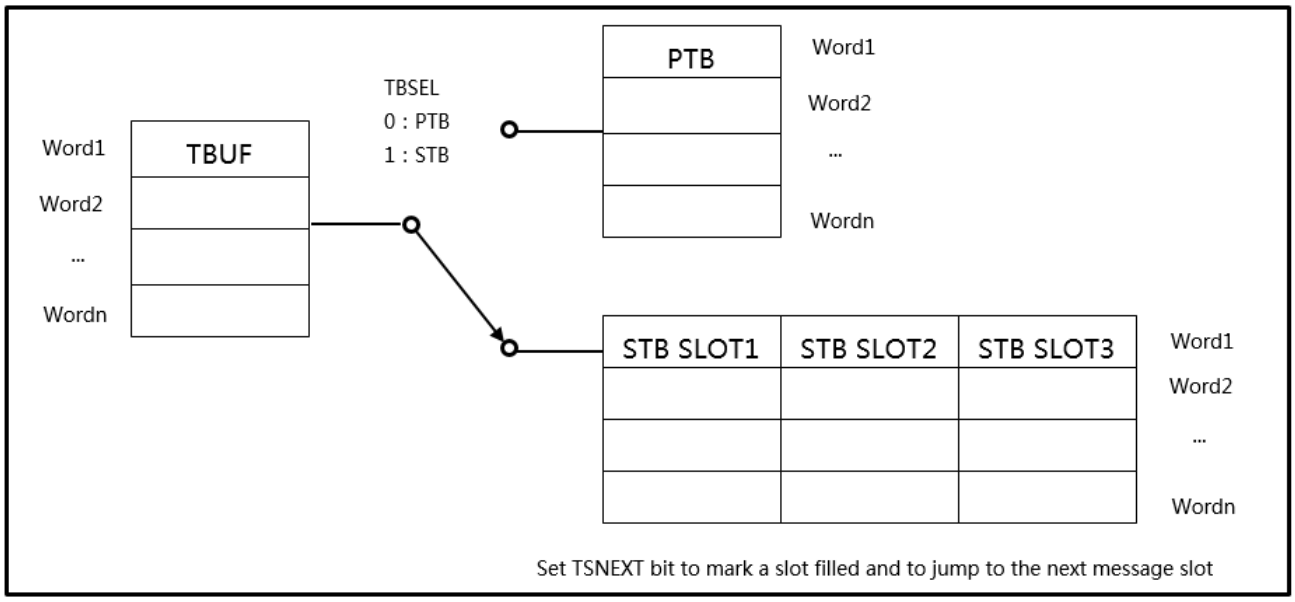


图 36-3 CANFD TBUF 寄存器写发送缓冲器和示意图

36.4.4 接收缓冲器

CAN_CTRL 提供 8 个 SLOT 的接收缓冲器用于存储接收到的数据，该 8 SLOT 的接收缓冲器工作在 FIFO 模式。RB SLOT 通过 RBUF 寄存器来读取接收到的数据，总是最先读取最早接收到的数据，并通过 RCTRL 寄存器的 RREL 设置为 1 释放已经读取的 RB SLOT，并指向下一个 RB SLOT。

通过 RBUF 读取 RB SLOT 示意图如下。

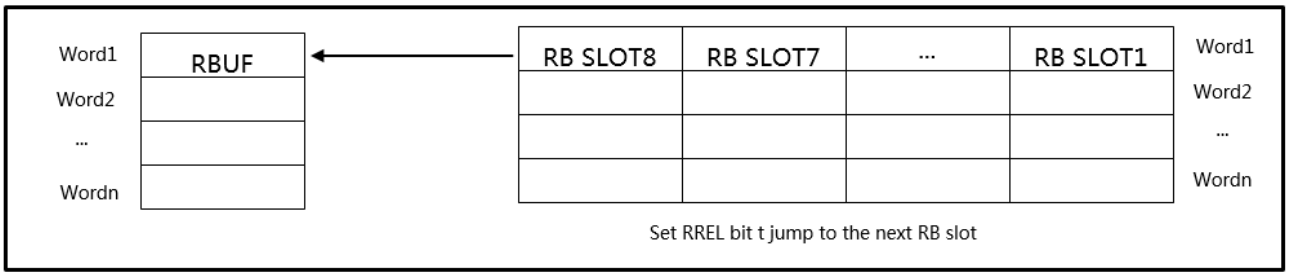


图 36-4 CAN RBUF 寄存器读接收缓冲器示意图

36.4.5 接收筛选寄存器组

CAN_CTRL 提供 16 组 32 位筛选器用于过滤接收到的数据从而降低 CPU 负荷，筛选器可以支持标准格式 11 位 ID 或者扩展格式 29 位 ID。每组筛选器有一个 32 位 ID CODE 寄存器和一个 32 位 ID MASK 寄存器，ID CODE 寄存器用于比较接收到 CAN ID，而 ID MASK 寄存器用于选择比较的 CAN ID 位。对应的 ID MASK 位为 1 时，不比较该位的 ID CODE。

接收到的数据只要通过 16 组筛选器的任意一组，则被接收，接收到的数据存储在 RB 中，否则数据不被接收，也不被存储。

每组筛选器通过 ACFEN 寄存器使能或者禁止。ID CODE 和 ID MASK 通过 ACFCTRL 寄存器的 SELMASK 位设定，SELMASK=0 时，指向 ID CODE，SELMASK=0 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。

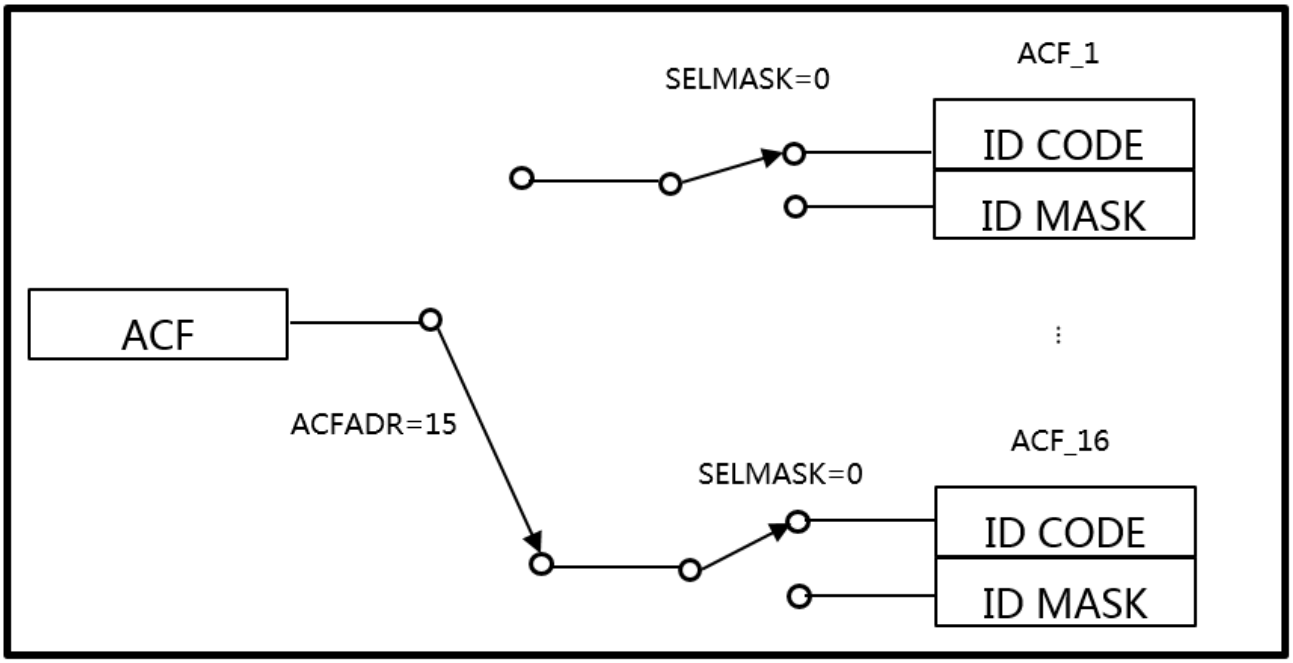


图 36-5 CAN ACF 寄存器访问筛选器组示意图

36.4.6 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. 如果选择的时 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

36.4.7 单次数据发送

不需要使用自动重新发送功能时，可以通过寄存器设定为单次发送模式，CFG_STAT 寄存器的 TPSS 位用于 PTB 的单次发送模式设定，TSSS 位用于 STB 的单次发送模式。数据成功发送时单次发送和正常发送模式时动作相同。但是数据没有成功发送时会出现以下结果：

- TPIF 置位（TPIE=1），对应的 BUF SLOT 数据会被清除。
- 有错误发送时，KOER 更新，BEIF 置位（BEIE=1）。
- 仲裁失败，ALIF 置位（ALIE=1）。

单次发送模式，不能单独依靠 TPIF 来判断发送完成，需要同 BEIF 和 ALIF 一起判断发送是否完成。

36.4.8 取消数据发送

可以通过 TPA 或者 TSA 取消已请求但还没有被执行的数据发送。取消数据发送会出现以下几种情况：

- 仲裁中
  - 节点仲裁失败，则取消数据发送。
  - 节点仲裁成功，则继续发送。
- 数据发送中
  - 成功发送数据且收到 ACK，对应的标志和状态正常置位。数据发送不取消。
  - 成功发送数据但没有收到 ACK，数据发送取消，错误计数器增加。
  - TSALL=1 设定的发送数据，正在发送的 STB SLOT 数据正常发送，没有开始发送的 STB SLOT 被取消。

取消数据发送的结果有以下两种情况。

- TPA 释放 PTB，且使 TPE=0。
- TSA 释放一个 STB SLOT 或者全部 STB SLOT 取决是 TSONE 还是 TSALL 使能的发送。

36.4.9 数据接收

接收筛选器组可以过滤掉不需要的接收数据，减少中断的发生和 RB 的读取，从而降低 CPU 负荷。接收数据设定步骤如下：

1. 设定筛选器组。
2. 设定 RFIE，RAFIE 和 AFWL。
3. 等待 RFIF 或者 RAFIF。
4. 通过 RBUF 从 RB FIFO 中读取最早接收到的数据。
5. 设置 RREL=1，选择下一个 RB SLOT。
6. 重复 4，5 直到通过 RSTAT 确认 RB 为空。

36.4.10错误处理

CAN_CTRL 一方面可以自动处理部分错误，比如自动重发数据或者丢弃接收到含有错误的帧，另一方面通过中断将错误向 CPU 报告。

CAN 节点有以下三种错误状态：

- 错误主动：节点检测到错误时自动发送主动错误标志。
- 错误被动：节点检测到错误时自动发送被动错误标志。
- 节点关闭：关闭状态下此节点不再影响整个 CAN 网络。

CAN_CTRL 提供 TECNT 和 RECNT 两个计数器用于计数错误。TECNT 和 RECNT 计数器按照 CAN 协议规定的规则进行增减。另外提供可编程的 CAN 错误警告 LIMIT 寄存器用于产生错误中断通知 CPU。

CAN 通信过程中有以下 5 种错误类型，错误类型可以通过 EALCAP 寄存器的 KOER 位识别。

- 位错误
- 形式错误
- 填充错误
- 应答错误
- CRC 错误

36.4.11 节点关闭

当发送错误数大于 255 时，CAN 节点自动进入节点关闭状态从而不参与 CAN 通信，直到返回到错误主动状态。可以通过 CFG_STAT 寄存器的 BUSOFF 位确认 CAN 节点关闭状态。BUSOFF 被置位的同时 EIF 中断产生。

CAN 从节点关闭状态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。

36.4.12 仲裁失败位置捕捉

CAN_CTRL 能够精确捕捉到仲裁失败位的位置并反映到 ALC 寄存器中。ALC 寄存器中保存着最近一次仲裁失败位的位置，如果节点赢得仲裁，则 ALC 位不更新。

ALC 值定义如下：

SOF 位后，第一个 ID 数据位 ALC 为 0，第二个 ID 数据位 ALC 为 1，依次类推。因为仲裁只发生在仲裁场内，所以 ALC 的最大值为 31。比如一个标准格式远程帧和一个扩展帧仲裁，扩展帧在 IDE 位失败，则 ALC=12。

36.4.13 回环模式

CAN_CTRL 支持以下两种回环模式：

- 内部回环
- 外部回环

两种回环模式都可以接收自己发出的数据帧，主要用于测试用途。

内部回环模式，模块内部将接收数据线连接到发送数据线，并且发送数据不输出。内部回环模式下，节点会生成自应答信号以避免 ACK 错误。

外部回环模式保持和收发器的连接因此发送的数据仍能出现在 CAN 总线上，在收发器的帮助下，CAN 能收到自己发送的数据。外部回环模式可以通过 RCTRL 寄存器的 SACK 位来决定是否生成自应答信号，SACK=0 时，不生成自应答信号，SACK=1

时，生成自应答信号。

外部回环模式，SACK=0 时，会出现以下两种情况：

- 其它节点也收到本节点发送的数据帧并发送应答信号，该情况下本节点能够成功收发数据。
- 如果没有其它节点返回应答信号，则会产生应答错误，会重新发送数据并增加错误计数器。此时推荐采用单次发送模式。

从回环模式返回到正常模式时，除了清除模式位以外，还需要软件复位 CAN_CTRL。

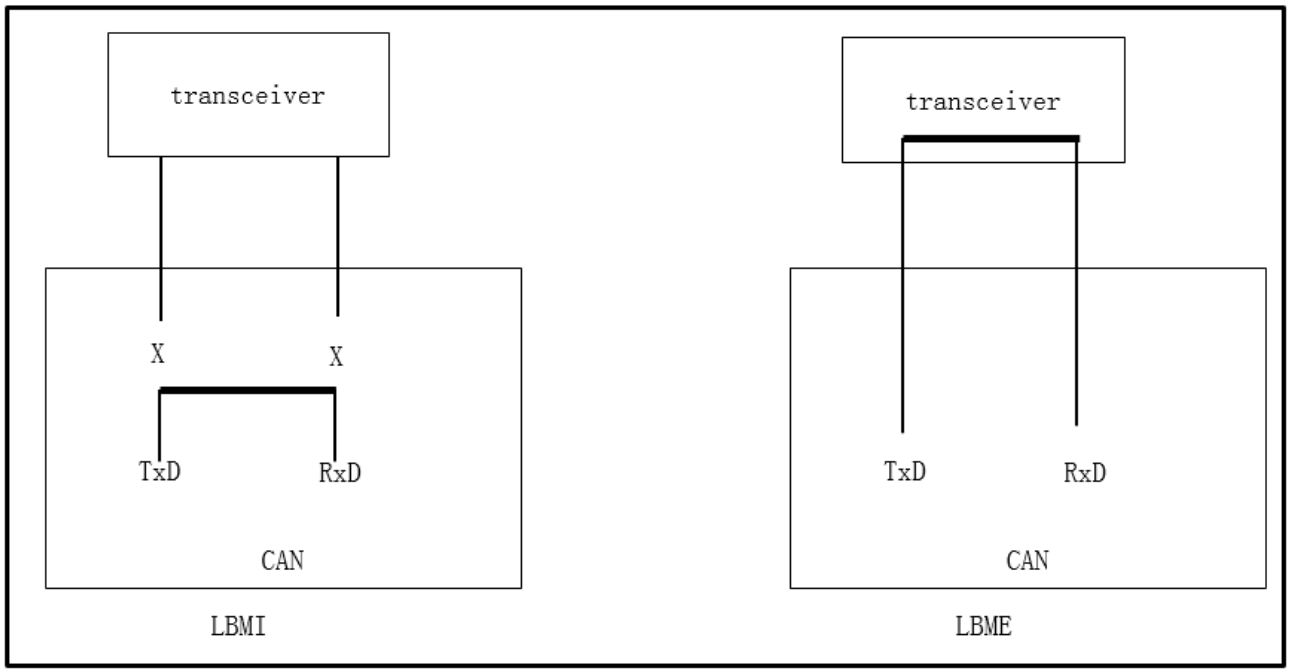


图 36-6 CANFD LBMI 和 LBME 示意图

36.4.14 静默模式

静默模式可以用来监控 CAN 网络数据。在静默模式下，可以从 CAN 总线接收数据，不向总线发送任何数据。将 TCMD 寄存器中的 LOM 置 1，使 CAN 总线控制器进入静默模式，将其清 0 可以离开静默模式。

外部回环模式可以和静默模式组合成外部回环静默模式，此时 CAN 可以认为一个安静的接收者，但在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。

36.4.15 软件复位功能

通过设定寄存器 CFG_STAT 寄存器的 RESET 位为 1,实现软件复位功能，软件复位功能的复位范围如下表所示。

寄存器位名	软件复位	备注	寄存器位名	软件复位	备注
ACFADR	否	-	F_SEG_1	是	只能在软件复位时可写
ACODE	否	只能在软件复位时可写	F_SEG_2	是	只能在软件复位时可写
AE_x	否	-	F_SJW	是	只能在软件复位时可写
AFWL	否	-	KOER	是	-
AIF	是	-	LBME	是	-
ALC	是	-	LBMI	是	-
ALIE	否	-	RACTIVE	是	接收立即停止，并不生成ACK
ALIF	是	-	RAFIE	否	-
AMASK	否	只能在软件复位时可写	RAFIF	是	-
BEIE	否	-	RBALL	是	-
BEIF	是	-	RBUF	是	RB被标记为空，数值不定
BUSOFF	否	通过写1清除	RECNT	否	通过BUSOFF写1清零
EIE	否	-	REF_ID	否	-
EIF	否	-	REF_IDE	否	-
EPASS	否	-	RFIE	否	-
EPIE	否	-	RFIF	是	-
EPIF	是	-	RIE	否	-
EWARN	否	-	RIF	是	-
EWL	是	-	ROIE	否	-
FD_ISO	否	只能在软件复位时可写	ROIF	是	-
F_FRESC	否	只能在软件复位时可写	ROM	否	

寄存器位名	软件复位	备注	寄存器位名	软件复位	备注
ROV	是	-	TSMODE	否	
RREL	是	-	TSNEXT	是	-
RSTAT	是		TSONE	是	-
SACK	是	-	TPIE	否	-
SELMASK	否	-	TPIF	是	-
S_PRESC	否	只能在软件复位时可以写	TPSS	是	-
S_SEG_1	否	只能在软件复位时可以写	TSFF	是	所有STB SLOT被标记为空
S_SEG_2	否	只能在软件复位时可以写	TSIE	否	-
SJW	否	只能在软件复位时可以写	TSIF	是	-
SSPOFF	是	-	TSSS	是	-
TACTIVE	是	发送立即停止	TSSTAT	是	所有STB SLOT被标记为空
TBE	是	-	TTEN	是	-
TBF	否	-	TTIF	是	-
TBPTR	否	-	TTIE	否	-
TBSEL	是	-	TTPTR	否	-
TBUF	是	STB被标记为空，指向PTB	TTTBM	否	-
TDCEN	是	-	TTYPE	否	-
TECNT	否	可通过BUSOFF=1清除	TT_TRIG	否	-
TEIF	是	-	TT_WTRIG	否	-
TPA	是	-	T_PRESC	否	-
TPE	是	-	WTIE	否	-
TSA	是	-	WTIF	是	
TSALL	是	-			

表 36-6 软件复位范围表

36.4.16 向上兼容 CAN-FD 功能

CAN-CTRL 在 CAN FD 功能禁止时即使在包含 CAN FD 网络中收到 CAN FD 的帧，接收器会自动忽略这些帧，不返回 ACK，等到总线空闲时，再发送或者接收下一个 CAN2.0B 的帧。

36.4.17 时间触发 TTCAN

CAN-CTRL 为 ISO11898-4 规定的时间触发通信方式提供部分(lever 1)硬件支持。本章节从以下 5 个部分介绍 TTCAN 功能。

36.4.17.1 TTCAN 模式下的 TBUF 行为

TTTBM=1

TTTBM=1 时，PTB 和 STB SLOT 一样组成 TB SLOT，通过 TBPTR 寄存器指定发送 BUF，其中 TBPTR=0 时，指向 PTB，TBPTR=1 是指向 STB SLOT1，依次类推。主机可以通过 TPE 和 TPF 寄存器来标记发送 BUF SLOT。此时 TBSEL 和 TSNEXT 寄存器无任何意义从而可以被忽略。

TTTBM=1 时，PTB 不具有任何特殊的属性，和 STB SLOT 一样，传送完成标志也采用 TSIF。

TTCAN 模式时，发送 BUF 没有 FIFO 模式和优先级仲裁模式，同时也只有一个选定的 SLOT 可以发送数据。

TTCAN 模式下，传输开始需要采用时间触发方式，TPE，TSONE，TSALL，TPSS 和 TPA 被固定为 0 且被忽略。

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。

36.4.17.2 TTCAN 功能描述

上电后，Time Master 需要根据 ISO 11898-4 协议进行初始化。一个 CAN 网络中，最多可以有 8 个潜在的 Time Master。每一个 Time Master 都具有自己的参考消息 ID（ID

最后 3 位）。这些潜在的 Time Master 根据自己的优先级发送各自的参考消息。

TTEN=1 后，16 位的计数器开始工作，当参考消息被成功接收或者 Time Master 成功发送参考消息时，CAN 控制器将 Sync_Mark 拷贝给 Ref_Mark，Ref_Mark 将 cycle time 设置为 0。成功接收参考消息置位 RIF 标志而成功发送参考消息置位 TPIF 标志或者 TSIF 标志。此时主机需要准备下一个动作的触发条件。

触发条件可以是接收触发。该触发仅触发中断可用于检测期待的消息没有被收到。

触发条件也可以是发送触发。该触发开始发送通过 TTPTR 寄存器指定的 TBUF SLOT 里的数据。如果选定的 TBUF SLOT 被标记为空，则不开始发送，但置位中断标志。

36.4.17.3 TTCAN 时序

CAN_CTRL 支持 ISO11898-4 level 1。包含的一个 16 位计数器工作在 S_PRESC，S_SEG_1，S_SEG_2 定义的位时间下。如果 TTEN=1，则有一个额外的预分频器 T_PRESC。

一帧数据的 SOF 时，计数器的值为 Sync_Mark。如果该帧数据为参考消息，则将 Sync_Mark 拷贝给 Ref_Mark。cycle time 等于计数器的值减去 Ref_Mark。该时间用作为接收消息的时间戳或者发送消息的触发时间基准。

36.4.17.4 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。TTTBM=1 时，只支持时间触发方式。

立即触发

通过写 TT_TRIG 的高位（不在意写入的值），启动触发器。此模式下，TTPTR 选定

的 TBUF SLOT 内的数据会立即发送。TTIF 不置位。

时间触发

时间触发方式仅通过置位 TTIF 标志产生中断，并无其他功能。如果一个节点期待在特定的时间窗口内收到期待的数据，则可以使用时间触发方式。如果 TT_TRIG 值小于实际的 cycle time，则 TEIF 置位且无其它动作。

单次发送触发

单次发送触发方式用于在执行时间窗口内发送数据。此时，忽略 TSSS 位。

通过 TEW 位设定 ISO11898-4 规定的最多 16 个 cycle time 的 Tick，设定范围为 1~16。

如果在规定的发送使能时间窗口内数据没有开始发送，则帧被丢弃。对应的发送 BUF SLOT 被标记为空，并且置位 AIF，对应的发送 BUF 内的数据不会被改写，因为可以通过置位 TPF 再次发送。

如果 TT_TRIG 值小于实际的 cycle time，则 TEIF 置位且无其它动作。

发送开始触发

发送开始触发方式用于仲裁时间窗口内，参与仲裁。TSSS 用于决定是否自动重发或者单次发送模式。如果 TTPTR 寄存器指定的消息没有被成功发送，可以使用发送停止触发来停止该发送。

如果 TT_TRIG 值小于实际的 cycle time，则 TEIF 置位且无其它动作。

发送停止触发

发送停止触发方式用于停止通过发送开始触发方式已经开始的发送。如果发送被停止，则发送帧被舍弃，置位 AIF 并将选定的 TBUF SLOT 标记为空，但 TBUF SLOT 内的数据不会被改写，可以通过置位 TPF 就可以再次发送。

如果 TT_TRIG 值小于实际的 cycle time 则 TEIF 置位且执行停止。

36.4.17.5 TTCAN 触发看门时间

TTCAN 触发看门时间功能类似于看门狗功能，在 TTTBM=1 时使用。用来看门从上次成功接收到参考消息开始的时间。参考消息可以在周期 cycle time 中或者一个事件后被接收，应用程序应该根据具体情况设定合适的看门时间。

如果 cycle count 等于 TT_WTRIG，则置位 WTIF。通过 WTIE 写 0，关闭看门触发。

如果 TT_WTRIG 比实际的 cycle time 小，则 TEIF 置位。

36.4.18 TDC 和 RDC

CANFD 通信时，数据的延迟可能超过一位时间，因此需要补偿。TDC（Transmitter Delay Compensation）和 RDC(Receiver Delay Compensation)用于数据延迟的补偿。其中 TDC 需要软件控制开关，而 RDC 不需要，自动有效。TDC 中采用辅助采样点 SSP（Secondary Sample Point）的方式去补偿数据延迟。如下图：

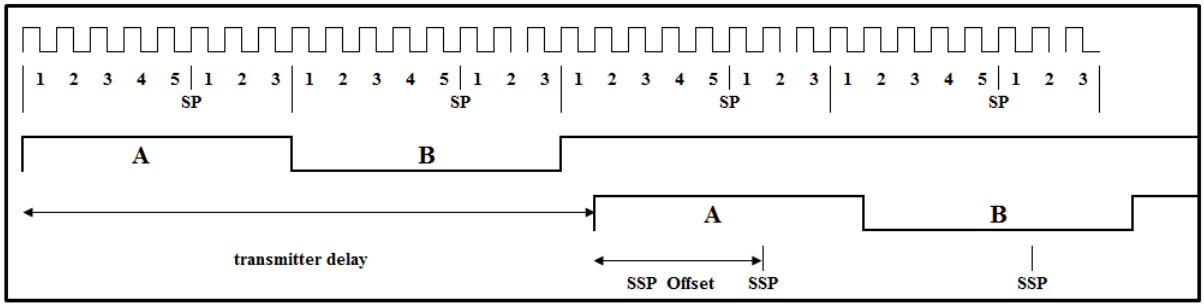


图 36-7 TDC 功能示意图

软件使能 TDC 功能时，本控制器可以自动确定 transmitter delay，通过设定寄存器 TDC.SSPOFF 来设定 SSP Offset。建议 SSP Offset 设定值和$t_{F_segment1}$设定值相同。

36.4.19 中断

符号	中断标志	描述
CAN_2_HOST	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 36-7 CAN 中断表

36.5 寄存器说明

CAN_2_BASE_ADDR: 0x40078000

寄存器名	符号	偏移地址	位宽	复位值
CAN接收BUF寄存器	CAN_RBUF	0x00~0x4F	-	0xXXXX XXXX
CAN发送BUF寄存器	CAN_TBUF	0x50~0x97	-	0xXXXX 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	CAN_RTIE	0xA4	8	0xFE
CAN接收和发送中断标志寄存	CAN_RTIF	0xA5	8	0x00
CAN错误中断使能和标志寄存	CAN_ERRINT	0xA6	8	0x00
CAN警告限定寄存器	CAN_LIMIT	0xA7	8	0x1B
CAN Slow位时序寄存器	CAN_SBT	0xA8	32	0x0102 0203
CAN Fast位时序寄存器	CAN_FBT	0xAC	32	0x0102 0203
CAN错误和仲裁失败捕捉寄存	CAN_EALCAP	0xB0	8	0x00
CAN发送器延迟补偿寄存器	CAN_TDC	0xB1	8	0x00
CAN接收错误计数器寄存器	CAN_RECNT	0xB2	8	0x00
CAN发送错误计数器寄存器	CAN_TECNT	0xB3	8	0x00
CAN筛选器组控制寄存器	CAN_ACFCTRL	0xB4	8	0x00
CAN筛选器组使能寄存器	CAN_ACFEN	0xB6	8	0x01
CAN筛选器组code和mask寄存	CAN_ACF	0xB8	32	0xXXXX 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	0xXXXX XXXX
TTCAN 触发配置寄存器	CAN_TRG_CFG	0xC4	16	0x0000
TTCAN 触发时间寄存器	CAN_TT_TRIG	0xC6	16	0x0000
TTCAN 触发看门时间寄存器	CAN_TT_WTRIG	0xC8	16	0xFFFF

表 36-8 CAN 寄存器一览表

36.5.1 CAN 接收 BUF 寄存器（CAN_RBUF）

CAN Receive Buffer Registers

偏移地址：0x50

复位值：0xXXXX XXXX

RBUF 寄存器指向最早接收到的 CAN 邮箱的 RB SLOT 地址，RBUF 寄存器可以按照任意顺序读取。

KOER 位即为寄存器 EALCAP.KOER，仅在 RBALL=1 时有意义。

TX 位表示在回环模式下接受到自己发送的消息。

CYCLE_TIME 位仅在 TTCAN 模式时有效，表示 SOF 开始时的 cycle time。

CAN_RBUF 仅支持 WORD 访问。

CAN 接收邮箱的数据格式如下：

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
RBUF	ID[7:0]								ID
RBUF+1	-					ID[10:8]			ID
RBUF+2	-								ID
RBUF+3	-								ID
RBUF+4	IDE=0	RTR	FDF	BRS	DLC[3:0]				Control
RBUF+5	KOER[2:0]			TX	-				Status
RBUF+6	CYCLE_TIME[7:0]								TTCAN
RBUF+7	CYCLE_TIME[15:8]								TTCAN
RBUF+8	DATA1								Data
RBUF+9	DATA2								Data
RBUF+10	DATA3								Data
…	…								Data
…	…								Data
…	…								Data
RBUF+70	DATA63								Data
RBUF+71	DATA64								Data

表 36-9 标准格式 CAN 接收邮箱格式

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
RBUF	ID[7:0]								ID
RBUF+1	ID[15:8]								ID
RBUF+2	ID[23:16]								ID
RBUF+3	-			ID[28:24]					ID
RBUF+4	IDE=1	RTR	FDF	BRS	DLC[3:0]				Control
RBUF+5	KOER[2:0]			TX	-				Status
RBUF+6	CYCLE_TIME[7:0]								TTCAN
RBUF+7	CYCLE_TIME[15:8]								TTCAN
RBUF+8	DATA1								Data
RBUF+9	DATA2								Data
RBUF+10	DATA3								Data
…	…								Data
…	…								Data
…	…								Data
RBUF+70	DATA63								Data
RBUF+71	DATA64								Data

表 36-10 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（仅适用于 CAN2.0,CAN FD 时固定为 0）

FDF(CAN FD Frame)

0：CAN2.0 帧

1：CAN FD 帧

BRS(Bit Rate Switch)

0：整帧为低速波特率

1：数据和 CRC 为快速波特率（仅适用于 CAN FD，FDF=0 时固定为 0）

DLC(Data Length Code):

数据长度码，CAN2.0 设定范围为 0~8，对应数据长度为 0Byte~8Byte,CAN FD 设定范围为 0~15,对应数据长度为 0Byte~64Byte

DLC(二进制)	Frame Type	Playload in Bytes
0000~1000	CAN2.0 和 CAN FD	0~8
10001~1111	CAN2.0	8
1001	CAN FD	12
1010	CAN FD	16
1011	CAN FD	20
1100	CAN FD	24
1101	CAN FD	32
1110	CAN FD	48
1111	CAN FD	64

表 36-11 DLC 控制位

状态位含义如下：

KOER：同 EALCAP.KOER

TX：回环模式中接收自己发送的数据时此位置 1

36.5.2 CAN 发送 BUF 寄存器（CAN_TBUF）

CAN Transmit Buffer Registers

偏移地址：0x00

复位值：0xXXXX XXXX

TBUF 寄存器指向下一个空的 CAN 发送 BUF SLOT，TBUF 寄存器可以按照任意顺序读取。通过软件将 TSNEXT 写 1 来标记对应的 TBUF SLOT 已经写入数据，从而指向下一个 TBUF SLOT。

TBUF 只能 WORD 访问。

CAN 发送邮箱的数据格式如下：

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
TBUF	ID[7:0]								ID
TBUF+1	-					ID[10:8]			ID
TBUF+2	-								ID
TBUF+3	-								ID
TBUF+4	IDE=0	RTR	FDF	BRS	DLC[3:0]				Control
TBUF+5	-								-
TBUF+6	-								-
TBUF+7	-								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
…	…								Data
…	…								Data
…	…								Data
TBUF+70	DATA63								Data
TBUF+71	DATA64								Data

表 36-12　标准格式 CAN 发送邮箱格式

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
TBUF	ID[7:0]								ID
TBUF+1	ID[15:8]								ID
TBUF+2	ID[23:16]								ID
TBUF+3	-			ID[28:24]					ID
TBUF+4	IDE=0	RTR	FDF	BRS	DLC[3:0]				Control
TBUF+5	-								-
TBUF+6	-								-
TBUF+7	-								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
…	…								Data
…	…								Data
…	…								Data
TBUF+70	DATA63								Data
TBUF+71	DATA64								Data

表 36-13 扩展格式 CAN 发送邮箱格式

控制位含义请参考 CAN 接收 BUF 寄存器章节说明。

36.5.3 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 times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

36.5.4 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/W
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&TTTBM=1	R/W
b2	TSONE	发送一帧STB数据设定位	发送一帧STB数据设定位（Transmit Secondary ONE frame） 0：不发送 1：发送一帧STB数据 FIFO模式中，发送最早写入的数据，优先级模式里发送最高优先级的数据 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&TTTBM=1	R/W
b1	TSALL	发送所有的STB数据设定位	发送所有的STB数据设定位（Transmit Secondary ALL frame） 0：不发送 1：发送STB中所有的数据 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 – BUSOFF=1 – LOM=1&LBME=0 – TTEN=1&TTTBM=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

36.5.5 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
FD_ISO	TSNEXT	TSMODE	TTTBM	-	-	TSSTAT[1:0]	

位	标记	位名	功能	读写
b7	FD_ISO	CAN FD ISO模式	CAN FD ISO模式(仅能在CAN_CFG_STAT.RESET=1时写入) 0：Bosch CAN FD模式 1：11898-1:2015 CAN FD模式	R/W
b6	TSNEXT	下一个STB SLOT	下一个STB（Transmit buffer Secondary NEXT） 0：无动作 1：当前STB SLOT已填充，指向下一个SLOT 应用程序将TBUF中的数据写完后，通过置位TSNEXT位标识当前STB SLOT已经被填充，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。	R/W
b5	TSMODE	STB发送模式	STB发送模式（Transmit buffer Secondary operation MODE） 0：FIFO模式 1：优先级模式 FIFO模式根据数据帧写入的先后顺序发送。 优先级模式根据ID自动判断，ID越小，优先级越高。 无论何种模式，PTB具有最高的优先级。 注意：TSMODE位只能在STB空时设定。	R/W
b4	TTTBM	TTCAN BUF模式	TTCAN BUF模式（TTCAN Transmit Buffer Mode） TTEN=0时，TTTBM被忽略。 0：TSMODE决定，PTB和STB 1：通过TBPTR和TTPTR设定 TTCAN模式时，只需要接收时间戳功能时，此位可以设置为0，通过TSMODE决定使用PTB还是STB。 注意：TSMODE位只能在STB空时设定。	R/W
b3~b2	Reserved	-	必须保持复位值。	R/W

位	标记	位名	功能	读写
b1~b0	TSSTAT	STB状态	STB状态（Transmission Secondary Status bits）	R
			TTEN=0 或 TTEN=1 &TTTBM=0	
			00：STB空	
			01：STB小于等于半满	
			10：STB大于半满	
			11：STB满	
			TTEN=1 且 TTTBM=1	
			00：PTB和STB空	
			01：PTB和STB非满	
			10：保留	
			11：PTB和STB满	

36.5.6 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	-	RSSTAT[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数据存储所有的数据帧	接收BUF数据存储素所有的数据帧（Receive Buffer stores ALL data frames） 0：正常模式 1：存储所有的数据包括有错误的数据。	R/W
b2	Reserved	-	必须保持复位值。	R/W
b1~b0	RSTAT	接收BUF状态	接收BUF状态（Receive buffer STATus） 00：RBUF空 01：RBUF非空但小于AFWL编程值 10：RBUF大于等于AFWL编程值但未满 11：满（上溢时保持此值）	R

36.5.7 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

36.5.8 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

位	标记	位名	功能	读写
b7	RIF	接收中断标志	接收中断标志（Receive Interrupt Flag） 0：未收到数据帧 1：接收到有效的数据帧或者远程帧 通过应用程序写1清0。	R/W
b6	ROIF	接收上溢中断标志	接收上溢中断标志（Receive Overrun Interrupt Flag） 0：无RB被覆盖（overwrite） 1：RB至少有一个被覆盖 上溢时ROIF和RFIF同时置1。 通过应用程序写1清0。	R/W
b5	RFIF	接收BUF满中断标志	接收BUF满中断标志（RB Full Interrupt Flag） 0：RB FIFO未满 1：RB FIFO满 通过应用程序写1清0。	R/W
b4	RAFIF	接收BUF将满中断标志	接收BUF将满中断标志（RB Almost Full Interrupt Flag） 0：被填充的RB SLOT数目小于AFWL设定值 1：被填充的RB SLOT数目大于等于AFWL设定值 通过应用程序写1清0。	R/W
b3	TPIF	PTB发送中断标志	PTB发送中断标志（Transmission Primary Interrupt Flag） 0：没有PTB发送完成 1：请求的PTB发送成功完成 通过应用程序写1清0。 注意：TTCAN模式时，TPIF无效，仅适用TSIF标志	R/W
b2	TSIF	STB发送中断标志	STB发送中断标志（Transmission Secondary Interrupt Flag） 0：没有STB发送完成 1：请求的STB发送成功完成 通过应用程序写1清0。 注意：TTCAN模式时，TPIF无效，仅使用TSIF标志	R/W

位	标记	位名	功能	读写
b1	EIF	错误中断标志	错误中断标志（Error Interrupt Flag） 0：BUSOFF位未发生变化，或者错误计数器的值与ERROR warning limit设定值的相对关系未发生变化。 1：BUSOFF位发生变化，或者错误计数器的值与ERROR warning limit设定值的相对关系发生变化。比如错误计数器的值从小于设定值变为大于设定值，或者从大于设定值变为小于设定值。 通过应用程序写1清0。	R/W
b0	AIF	取消发送中断标志	取消发送中断标志（Abort Interrupt Flag） 0：未取消发送数据 1：通过TPA和TSA请求的发送消息被成功取消。 通过应用程序写1清0。	R/W

36.5.9 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	错误被动（Error Passive mode active） 0：节点是主动错误节点 1：节点时被动错误节点 通过应用程序写1清0。	R
b5	EPIE	错误被动中断使能	错误被动中断使能（Error Passive Interrupt Enable） 0：禁止 1：使能	R/W
b4	EPIF	错误被动中断标志	错误被动中断标志（Error Passive Interrupt Flag） 0：未发生错误主动到错误被动或者错误被动到错误主动的变化 1：发生错误主动到错误被动或者错误被动到错误主动的变化 通过应用程序写1清0。	R/W
b3	ALIE	仲裁失败中断使能	仲裁失败中断使能（Arbitration Lost Interrupt Enable） 0：禁止 1：使能	R/W
b2	ALIF	仲裁失败中断标志	仲裁失败中断标志（Arbitration Lost Interrupt Flag） 0：仲裁成功 1：仲裁失败 通过应用程序写1清0。	R/W
b1	BEIE	总线错误中断使能	总线错误中断使能（Bus Error Interrupt Enable） 0：禁止 1：使能	R/W
b0	BEIF	总线错误中断标志	总线错误中断标志（Bus Error Interrupt Flag） 0：无总线错误 1：总线错误 通过应用程序写1清0。	R/W

36.5.10 CAN Slow 位时序寄存器（CAN_SBT）

CAN Slow Bit Timing Register

偏移地址：0xA8

复位值：0x0102 0203

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
S_PRESC[7:0]								-	S_SJW[6:0]						
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	S_SEG_2[6:0]							S_SEG_1[7:0]							

位	标记	位名	功能	读写
b31~b24	S_PRESC	预分频设定	预分频设定（S_Prescaler） 此寄存器设定值时将模块通信时钟设定为（S_PRESC+1）分频作为TQ。	R/W
b23	Reserved	-	必须保持复位值。	R/W
b22~b16	S_SJW	再同步补偿宽度时间设定	再同步补偿宽度时间设定（Bit Timing Segment 2） 再同步补偿宽度时间=(S_SJW+1)*TQ	R/W
b15	Reserved	-	必须保持复位值。	R/W
b14~b8	S_SEG_2	位段2时间设定	位段2时间单元设定（Bit Timing Segment 2） 位段2时间=(S_SEG_2+1)*TQ	R/W
b7~b0	S_SEG_1	位段1时间设定	位段1时间单元设定（Bit Timing Segment 1） 位段1时间=(S_SEG_1+2)*TQ	R/W

36.5.11 CAN Fast 位时序寄存器（CAN_FBT）

CAN Fast Bit Timing Register

偏移地址：0xAC

复位值：0x0102 0203

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
F_PRESC[7:0]								--	-	-	-	F_SJW[3:0]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	F_SEG_2[3:0]				-	-	-	F_SEG_1[4:0]				

位	标记	位名	功能	读写
b31~b24	F_PRESC	预分频设定	预分频设定（Prescaler） 此寄存器设定值时将模块通信时钟设定为（F_PRESC+1）分频作为TQ。	R/W
b23~b20	Reserved	-	必须保持复位值。	R/W
b29~b16	F_SJW	再同步补偿宽度时间设定	再同步补偿宽度时间设定（Bit Timing Segment 2） 再同步补偿宽度时间=(F_SJW+1)*TQ	R/W
b15~12	Reserved	-	必须保持复位值。	R/W
b11~b8	F_SEG_2	位段2时间设定	位段2时间单元设定（Bit Timing Segment 2） 位段2时间=(F_SEG_2+1)*TQ	R/W
b7~b5	Reserved	-	必须保持复位值。	R/W
b4~b0	F_SEG_1	位段1时间设定	位段1时间单元设定（Bit Timing Segment 1） 位段1时间=(F_SEG_1+2)*TQ	R/W

36.5.12 CAN 发送器延迟补偿寄存器（CAN_TDC）

CAN Transmitter Delay Compensatin Register

偏移地址：0xb1

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
TDCEN	SSPOFF[6:0]						

位	标记	位名	功能	读写
b7	TDCEN	发送器延迟补偿功能使能位	发送器延迟补偿功能使能位 用于CAN FD BRS=1的情况 0：禁止 1：使能	R/W
b6~b0	SSPOFF	TDC功能辅助采样点位置设定位	TDC功能辅助采样点位置设定位 设定单位为TQ	R/W

36.5.13 CAN 错误和仲裁失败捕捉寄存器（CAN_EALCAP）

CAN Error and Arbitration Lost Capture Register

偏移地址：0xB0

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
KOER[2:0]			ALC[4:0]				

位	标记	位名	功能	读写
b7~b5	KOER	错误类别	错误类别（Kind Of Error） 000：无错误 001：位错误 010：形式错误 011：填充错误 100：应答错误 101：CRC错误 110：其他错误 111：保留 有错误时KOER位更新，正常发送接收时KOER位保持不变。	R
b4~b0	ALC	仲裁失败位置捕捉	仲裁失败位置捕捉（Arbitration Lost Capture） 仲裁失败时ALC记录一帧数据中仲裁失败时的位置。	R/W

36.5.14 CAN 警告限定寄存器（CAN_LIMIT）

CAN Warning Limits Register

偏移地址：0xA7

复位值：0x1B

b7	b6	b5	b4	b3	b2	b1	b0
AFWL[3:0]				EWL[3:0]			

位	标记	位名	功能	读写
b7~b4	AFWL	接收BUF将满Warning Limit	接收BUF将满Warning Limit（receive buffer Almost Full Warning Limit） 设定值范围为1~8。 AFWL=0无意义，当做AFWL=1处理。	R/W
b3~b0	EWL	Error Waring Limit编程值	Error Waring Limit编程值（Programmable Error Warning Limit） Error Waring Limit=（EWL+1）*8。 该寄存器设定值影响EIF标志。	R/W

36.5.15 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

36.5.16 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

36.5.17 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/W
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/W
b3~b0	ACFADR	筛选器地址	筛选器地址（acceptance filter address） ACFADR指向具体的筛选器，通过SELMASK去区分ID和MASK。 0000：指向ACF_1 0001：指向ACF_2 0010：指向ACF_3 0011：指向ACF_4 0100：指向ACF_5 0101：指向ACF_6 0110：指向ACF_7 0111：指向ACF_8 1000：指向ACF_9 1001：指向ACF_10 1010：指向ACF_11 1011：指向ACF_12 1100：指向ACF_13 1101：指向ACF_14 1110：指向ACF_15 1111：指向ACF_16	R/W

36.5.18 CAN 筛选器组使能寄存器（CAN_ACFEN）

CAN Acceptance Filter Enable Register

偏移地址：0xB6

复位值：0x0001

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AE_16	AE_15	AE_14	AE_13	AE_12	AE_11	AE_10	AE_9	AE_8	AE_7	AE_6	AE_5	AE_4	AE_3	AE_2	AE_1

位	标记	位名	功能	读写
b15	AE_16	ACF_16使能	ACF_16使能（Acceptance Filter 16 Enable） 0：禁止 1：使能	R/W
b14	AE_15	ACF_15使能	ACF_15使能（Acceptance Filter 15 Enable） 0：禁止 1：使能	R/W
b13	AE_14	ACF_14使能	ACF_14使能（Acceptance Filter 14 Enable） 0：禁止 1：使能	R/W
b12	AE_13	ACF_13使能	ACF_13使能（Acceptance Filter 13 Enable） 0：禁止 1：使能	R/W
b11	AE_12	ACF_12使能	ACF_12使能（Acceptance Filter 12 Enable） 0：禁止 1：使能	R/W
b10	AE_11	ACF_11使能	ACF_11使能（Acceptance Filter 11 Enable） 0：禁止 1：使能	R/W
b9	AE_10	ACF_10使能	ACF_10使能（Acceptance Filter 10 Enable） 0：禁止 1：使能	R/W
b8	AE_9	ACF_9使能	ACF_9使能（Acceptance Filter 9 Enable） 0：禁止 1：使能	R/W
b7	AE_8	ACF_8使能	ACF_8使能（Acceptance Filter 8 Enable） 0：禁止 1：使能	R/W
b6	AE_7	ACF_7使能	ACF_7使能（Acceptance Filter 7 Enable） 0：禁止 1：使能	R/W
b5	AE_6	ACF_6使能	ACF_6使能（Acceptance Filter 6 Enable） 0：禁止 1：使能	R/W
b4	AE_5	ACF_5使能	ACF_5使能（Acceptance Filter 5 Enable） 0：禁止 1：使能	R/W
b3	AE_4	ACF_4使能	ACF_4使能（Acceptance Filter 4 Enable） 0：禁止 1：使能	R/W
b2	AE_3	ACF_3使能	ACF_3使能（Acceptance Filter 3 Enable） 0：禁止 1：使能	R/W
b1	AE_2	ACF_2使能	ACF_2使能（Acceptance Filter 2 Enable） 0：禁止 1：使能	R/W
b0	AE_1	ACF_1使能	ACF_1使能（Acceptance Filter 1 Enable） 0：禁止 1：使能	R/W

36.5.19 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：0xXXXX XXXX

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	AIDEE	AIDE	ACODE[28:16] or AMASK_28:16]												
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ACODE[15:0] or AMASK[15:0]															

位	标记	位名	功能	读写
b31	Reserved	-	读出值为不定。	R/W
b30	AIDEE	IDE位比较使能	IDE位比较使能（Acceptance mask IDE bit check enable） 尽在SELMASK=1时有效 0：筛选器接收标准格式和扩展格式帧 1：筛选器接收AIDE位定义的标准格式或者扩展格式帧	R/W
b29	AIDE	IDE位MASK	IDE位MASK 0：筛选器仅接收标准格式 1：筛选器仅接收扩展格式	R/W
b28~b0	ACODE/ AMASK	筛选器CODE/ 筛选器MASK	筛选器CODE（acceptance filter code） 通过ACFADR指向具体的筛选器。 SELMASK=0时表示筛选器的CODE。 标准格式时使用位10~位0，扩展格式时使用位29~位0。 筛选器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29~位0。	R/W

36.5.20 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	TBF	设置TB为已填充	设置TB已填充（set TB slot to “Filled”） 0：无操作 1：被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/W
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 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

36.5.21 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） 0：禁止 1：使能	R/W
b6	WTIF	触发看门中断标志	触发看门中断标志（Watch Trigger Interrupt Flag） 当CYCLE COUNT值=TT_WTRIG设定值时且WTIE=1时，WTIF置位。 通过应用程序写1清0。	R/W
b5	TEIF	触发错误中断标志	触发错误中断标志（Trigger Error Interrupt Flag） TT_TTIG设定值小于实际的CYCLE_TIME时，TEIF置位。	R
b4	TTIE	时间触发中断使能	时间触发中断使能（Time Trigger Interrupt Enable） 0：禁止 1：使能	R/W
b3	TTIF	时间触发中断标志	时间触发中断标志（Time Trigger Interrupt Flag） 当CYCLE COUNT值=TT_TRIG设定值时且TTIE=1时，TTIF置位。 如果TT_TRIG没有更新，则TTIF只置位1次，下一个基本CYCLE不置位。 通过应用程序写1清0。	R/W
b2~b1	T_PRESC	TTCAN计数器预分频	TTCAN计数器预分频（TTCAN Timer PRESCaler） 00：SBT寄存器设定的位时间的1分频 01：SBT寄存器设定的位时间的2分频 10：SBT寄存器设定的位时间的4分频 11：SBT寄存器设定的位时间的8分频 注意：T_PRESC可在TTEN=0时进行写操作或者写TTEN=1时同时操作。	R/W
b0	TTEN	TTCAN使能	TTCAN使能（Time Trigger Enable） 0：禁止 1：使能TTCAN，计数器开始计数。	R/W

36.5.22 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位	参考消息的IDE位（REFerence message IDE bit） 0：标准格式 1：扩展格式	R/W
b30~b29	Reserved	-	读出值为不定。	R/W
b28~b0	REF_ID	参考消息的ID位	参考消息的ID位（REFerence message IDentifier） REF_IDE=0：REF_ID[10:0]有效 REF_IDE=1：REF_ID[28:0]有效 REF_ID用于检测参考消息，适用于发送和接收。 检测到参考消息后，当前帧的Sync_Mark则变成Ref_Mark。 REF_ID[2:0]固定为0，并不检查其值，这样最多可以支持8个潜在的time master。 当REF_MSG的最高字节写操作后，则需要等待6个CAN时钟周期以完成REF_MSG向CAN时钟域的传递。	R/W

36.5.23 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/W
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_TRIG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/W
b2~b0	TTPTR	发送触发器TB slot指针	发送触发器TB slot指针（Transmit Trigger TB slot Pointer） 000：指向PTB 001：指向STB SLOT1 010：指向STB SLOT2 011：指向STB SLOT3 其他：设定禁止 如果指向的TB SLOT被标记为空，当到达触发时间后，TEIF置位。	R/W

36.5.24 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

36.5.25 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_TWRIG	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

37 CAN2.0B 控制器（CAN2.0B）

37.1 简介

CAN 2.0B 控制器对应依据产品型号不同对应 CAN 控制器通道 1（CAN_1）或者 CAN 控制器通道 1 和 2（CAN_1/CAN_2）。具体请参考型号功能对比表。

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。

CAN2.0B 控制器遵循 CAN 总线 CAN2.0(2.0A、CAN2.0B）协议。

CAN 总线控制器可以处理总线上的数据收发，在本产品中，CAN2.0B 控制器具有 16 组筛选器。筛选器用于为应用程序选择要接收的消息。

CAN2.0B 控制器中应用程序可通过 1 个高优先级的主发送缓冲器（Primary Transmit Buffer，以下简称 PTB）和 3 个辅发送缓冲器（Secondary Transmit Buffer，以下简称 STB）将发送数据送至总线，由发送调度器决定邮箱发送顺序。通过 8 个接收缓冲器（Receive Buffer，以下简称 RB）获取总线数据。3 个 STB 以及 8 个 RB 可以理解为一个 3 级 FIFO 和一个 8 级 FIFO，FIFO 完全由硬件控制。

CAN2.0B 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。
- CAN2.0B 向上兼容 CAN FD 协议。
- CAN2.0B 支持最高通信波特率 1Mbit/s
- 支持 1~1/256 的波特率预分频，灵活配置波特率。
- 8 个接收缓冲器
  - FIFO 方式
  - 错误或者不被接收的数据不会覆盖存储的消息
- 1 个高优先主发送缓冲器 PTB
- 3 个副发送缓冲器 STB
  - FIFO 方式

- 优先级仲裁方式

- 16 组独立的筛选器
  - 支持 11 位标准 ID 和 29 位扩展 ID
  - 可编程 ID CODE 位以及 MASK 位
- PTB/STB 均支持支持单次发送模式
- 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

37.2 CAN 系统框图

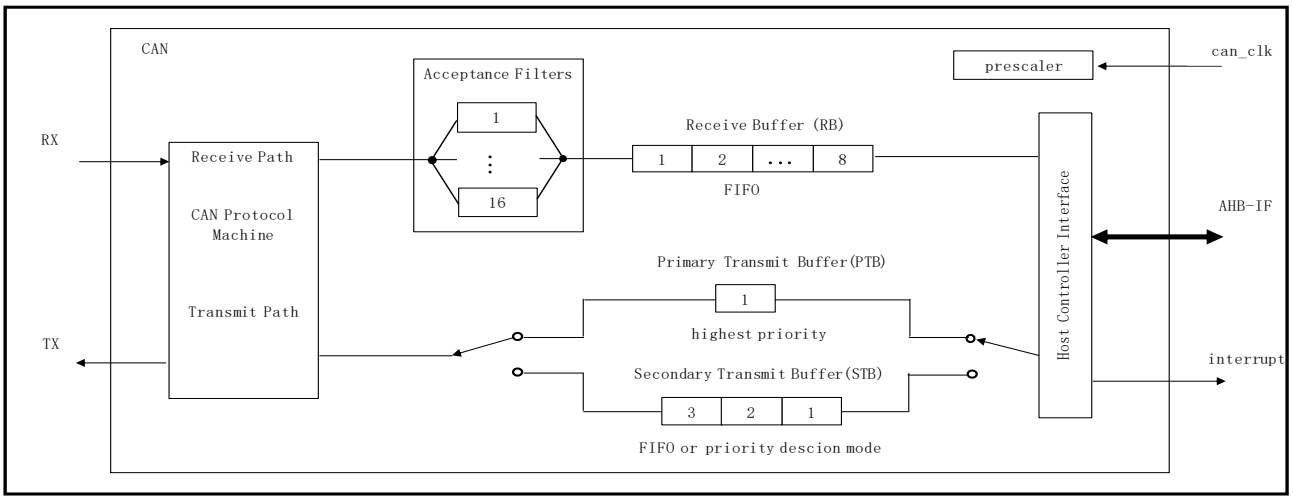


图 37-1 CAN 系统框图

37.3 管脚说明

管脚名	方向	功能描述
CANn_RX	输入	CAN接收数据信号
CANn_TX	输出	CAN发送数据信号
CANn_TST_SAMPLE	输出	仅用于观测，采样位置（采样点后一周期的通信时钟宽度）
CANn_TST_CLOCK	输出	仅用于观测，波特率（一位开始前一周期的通信时钟宽度）

n:1~2

表 37-1 CAN 管脚说明

37.4 功能说明

37.4.1 动作模式

CAN2.0 控制器存在两个操作模式，复位模式（CAN_CFG_STAT.RESET=1）和动作模式（CAN_CFG_STAT.RESET=0）。模块初始化时，首先应该在复位模式中设定只能在复位模式下操作的寄存器（详见寄存器说明章节），然后退后复位模式，在动作模式中操作其余寄存器。

37.4.2 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 SBT 设定。SBT 寄存器只能在 CAN_CFG_STAT.RESET=1 即 CAN 软件复位时设定。

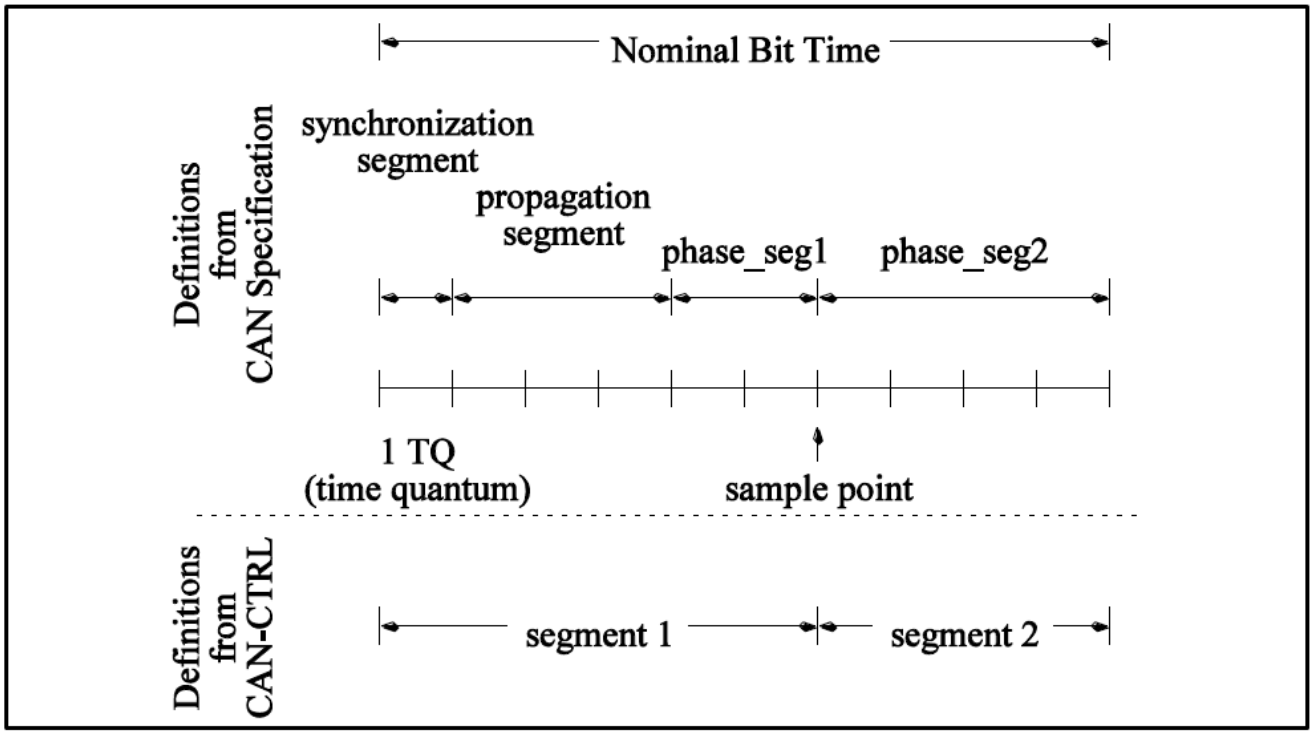


图 37-2 CAN 位时间定义图

TQ 计算方法请参考以下公式，其中 PRESC 通过 SBT 寄存器的 S_PRESC 位设定。

f_{can_clk}为 CAN 通信时钟频率。

$$S_TQ = \frac{S_PRESC+1}{f_{can_clk}}$$

位时间计算方法请参考以下公式，其中 S_segment1 和 S_segment2 通过 SBT 寄存器的 S_SEG_1 位和 S_SEG_2 位设定。

$$Slow_BT = t_{S_segment1} + t_{S_segment2} = \big((S_SEG_1 + 2) + (S + SEG_2 + 1)\big) \times S_TQ$$

位	设定范围	规则
SBT寄存器的S_SEG_1位	[0..63] CAN2.0 bits (slow)	SEG_1 ≥ SEG_2 + 1 SEG_2 ≥ SJW
SBT寄存器的S_SEG_2位	[0..7] CAN2.0 bits (slow))	
SBT寄存器的S_SJW位	[0..15] CAN2.0 bits (slow)	

表 37-2 CAN 位时间设定规则

37.4.3 发送缓冲器

CAN_CTRL 提供两种发送缓冲器用于发送数据，主发送数据缓冲器 PTB 和副发送缓冲器 STB。PTB 具有最高的优先级，但只能缓冲一帧数据。STB 优先级比 PTB 低，但可以缓冲 3 帧数据，且 STB 内 3 帧数据可以工作在 FIFO 模式或者优先级仲裁模式。STB 中的 3 帧说数据可以通过 TCMD 寄存器的 TSALL 位设定为 1 全部发送，在 FIFO 模式下，最先写入的数据先发送，在优先级模式下，ID 小的数据先发送。

PTB 中的数据具有最高优先级，所以 PTB 发送能推迟 STB 发送，但是已经赢得仲裁并开始发送的 STB 不能够被 PTB 发送推迟。

PTB 和 STB 可以通过 TBUF 寄存器进行访问。通过 TCMD 寄存器的 TBSEL 位选择 PTB 或者 STB，TBSEL=0，选择 PTB，TBSEL=1，选择 STB。通过 TCTRL 寄存器的 TSNEXT 位选择 STB 中的下一个 SLOT。对应关系如下图所示：

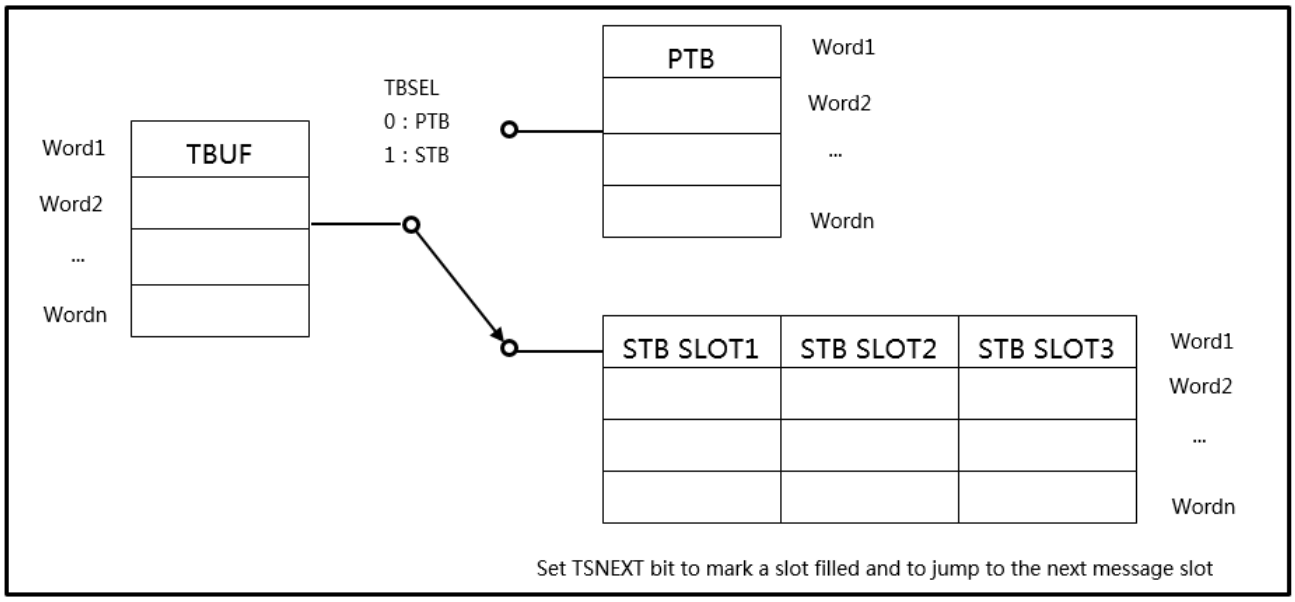


图 37-3 CAN TBUF 寄存器写发送缓冲器和示意图

37.4.4 接收缓冲器

CAN_CTRL 提供 8 个 SLOT 的接收缓冲器用于存储接收到的数据，该 8 SLOT 的接收缓冲器工作在 FIFO 模式。RB SLOT 通过 RBUF 寄存器来读取接收到的数据，总是最先读取最早接收到的数据，并通过 RCTRL 寄存器的 RREL 设置为 1 释放已经读取的 RB SLOT，并指向下一个 RB SLOT。

通过 RBUF 读取 RB SLOT 示意图如下。



图 37-4 CAN RBUF 寄存器读接收缓冲器示意图

37.4.5 接收筛选寄存器组

CAN_CTRL 提供 16 组 32 位筛选器用于过滤接收到的数据从而降低 CPU 负荷，筛选器可以支持标准格式 11 位 ID 或者扩展格式 29 位 ID。每组筛选器有一个 32 位 ID CODE 寄存器和一个 32 位 ID MASK 寄存器，ID CODE 寄存器用于比较接收到 CAN ID，而 ID MASK 寄存器用于选择比较的 CAN ID 位。对应的 ID MASK 位为 1 时，不

比较该位的 ID CODE。

接收到的数据只要通过 16 组筛选器的任意一组，则被接收，接收到的数据存储在 RB 中，否则数据不被接收，也不被存储。

每组筛选器通过 ACFEN 寄存器使能或者禁止。ID CODE 和 ID MASK 通过 ACFCTRL 寄存器的 SELMASK 位设定，SELMASK=0 时，指向 ID CODE，SELMASK=0 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。

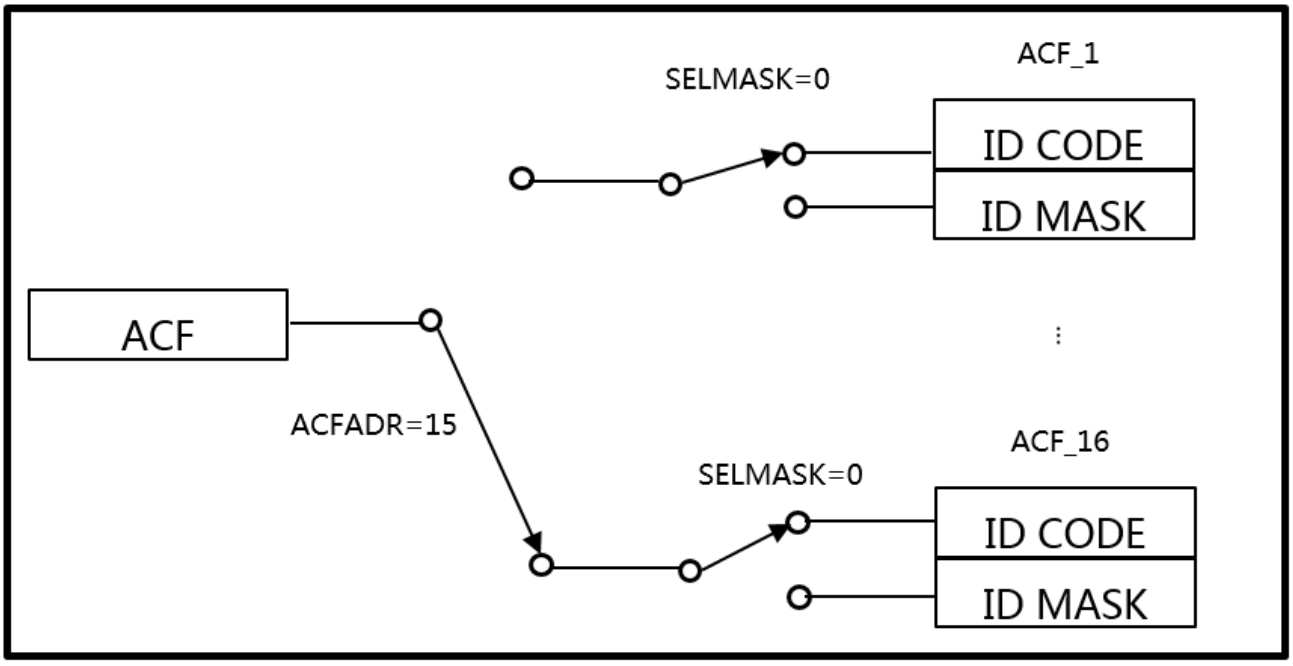


图 37-5 CAN ACF 寄存器访问筛选器组示意图

37.4.6 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. 如果选择的时 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

37.4.7 单次数据发送

不需要使用自动重新发送功能时，可以通过寄存器设定为单次发送模式，CFG_STAT 寄存器的 TPSS 位用于 PTB 的单次发送模式设定，TSSS 位用于 STB 的单次发送模式。数据成功发送时单次发送和正常发送模式时动作相同。但是数据没有成功发送时会出现以下结果：

- TPIF 置位（TPIE=1），对应的 BUF SLOT 数据会被清除。
- 有错误发送时，KOER 更新，BEIF 置位（BEIE=1）。
- 仲裁失败，ALIF 置位（ALIE=1）。

单次发送模式，不能单独依靠 TPIF 来判断发送完成，需要同 BEIF 和 ALIF 一起判断发送是否完成。

37.4.8 取消数据发送

可以通过 TPA 或者 TSA 取消已请求但还没有被执行的数据发送。取消数据发送会出现以下几种情况：

- 仲裁中
  - 节点仲裁失败，则取消数据发送。
  - 节点仲裁成功，则继续发送。
- 数据发送中
  - 成功发送数据且收到 ACK，对应的标志和状态正常置位。数据发送不取消。
  - 成功发送数据但没有收到 ACK，数据发送取消，错误计数器增加。
  - TSALL=1 设定的发送数据，正在发送的 STB SLOT 数据正常发送，没有开始发送的 STB SLOT 被取消。

取消数据发送的结果有以下两种情况。

- TPA 释放 PTB，且使 TPE=0。
- TSA 释放一个 STB SLOT 或者全部 STB SLOT 取决是 TSONE 还是 TSALL 使能的发送。

37.4.9 数据接收

接收筛选器组可以过滤掉不需要的接收数据，减少中断的发生和 RB 的读取，从而降低 CPU 负荷。接收数据设定步骤如下：

1. 设定筛选器组。
2. 设定 RFIE，RAFIE 和 AFWL。
3. 等待 RFIF 或者 RAFIF。
4. 通过 RBUF 从 RB FIFO 中读取最早接收到的数据。
5. 设置 RREL=1，选择下一个 RB SLOT。
6. 重复 4，5 直到通过 RSTAT 确认 RB 为空。

37.4.10错误处理

CAN_CTRL 一方面可以自动处理部分错误，比如自动重发数据或者丢弃接收到含有错误的帧，另一方面通过中断将错误向 CPU 报告。

CAN 节点有以下三种错误状态：

- 错误主动：节点检测到错误时自动发送主动错误标志。
- 错误被动：节点检测到错误时自动发送被动错误标志。
- 节点关闭：关闭状态下此节点不再影响整个 CAN 网络。

CAN_CTRL 提供 TECNT 和 RECNT 两个计数器用于计数错误。TECNT 和 RECNT 计数器按照 CAN 协议规定的规则进行增减。另外提供可编程的 CAN 错误警告 LIMIT 寄存器用于产生错误中断通知 CPU。

CAN 通信过程中有以下 5 种错误类型，错误类型可以通过 EALCAP 寄存器的 KOER 位识别。

- 位错误
- 形式错误

- 填充错误
- 应答错误
- CRC 错误

37.4.11 节点关闭

当发送错误数大于 255 时，CAN 节点自动进入节点关闭状态从而不参与 CAN 通信，直到返回到错误主动状态。可以通过 CFG_STAT 寄存器的 BUSOFF 位确认 CAN 节点关闭状态。BUSOFF 被置位的同时 EIF 中断产生。

CAN 从节点关闭状态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。

37.4.12 仲裁失败位置捕捉

CAN_CTRL 能够精确捕捉到仲裁失败位的位置并反映到 ALC 寄存器中。ALC 寄存器中保存着最近一次仲裁失败位的位置，如果节点赢得仲裁，则 ALC 位不更新。

ALC 值定义如下：

SOF 位后，第一个 ID 数据位 ALC 为 0，第二个 ID 数据位 ALC 为 1，依次类推。因为仲裁只发生在仲裁场内，所以 ALC 的最大值为 31。比如一个标准格式远程帧和一个扩展帧仲裁，扩展帧在 IDE 位失败，则 ALC=12。

37.4.13 回环模式

CAN_CTRL 支持以下两种回环模式：

- 内部回环
- 外部回环

两种回环模式都可以接收自己发出的数据帧，主要用于测试用途。

内部回环模式，模块内部将接收数据线连接到发送数据线，并且发送数据不输出。内部回环模式下，节点会生成自应答信号以避免 ACK 错误。

外部回环模式保持和收发器的连接因此发送的数据仍能出现在 CAN 总线上，在收发器的帮助下，CAN 能收到自己发送的数据。外部回环模式可以通过 RCTRL 寄存器的 SACK 位来决定是否生成自应答信号，SACK=0 时，不生成自应答信号，SACK=1 时，生成自应答信号。

外部回环模式，SACK=0 时，会出现以下两种情况：

- 其它节点也收到本节点发送的数据帧并发送应答信号，该情况下本节点能够成功收发数据。
- 如果没有其它节点返回应答信号，则会产生应答错误，会重新发送数据并增加错误计数器。此时推荐采用单次发送模式。

从回环模式返回到正常模式时，除了清除模式位以外，还需要软件复位 CAN_CTRL。

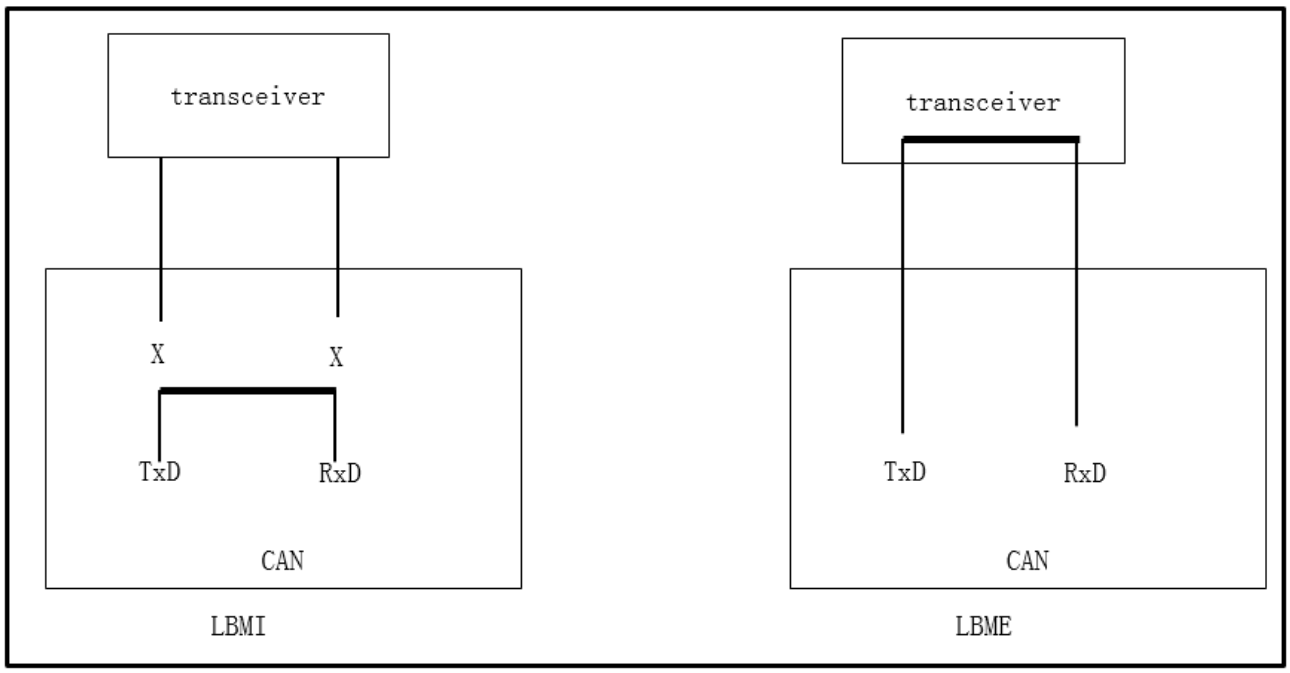


图 37-6 CAN LBMI 和 LBME 示意图

37.4.14 静默模式

静默模式可以用来监控 CAN 网络数据。在静默模式下，可以从 CAN 总线接收数据，不向总线发送任何数据。将 TCMD 寄存器中的 LOM 置 1，使 CAN 总线控制器进入静默模式，将其清 0 可以离开静默模式。

外部回环模式可以和静默模式组合成外部回环静默模式，此时 CAN 可以认为一个安静的接收者，但在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。

37.4.15 软件复位功能

通过设定寄存器 CFG_STAT 寄存器的 RESET 位为 1，实现软件复位功能，软件复位功能的复位范围如下表所示。

寄存器位名	软件复位	备注	寄存器位名	软件复位	备注
ACFADR	否	-	KOER	是	-
ACODE	否	只能在软件复位时可写	LBME	是	-
AE_x	否	-	LBMI	是	-
AFWL	否	-	RACTIVE	是	接收立即停止，并不生成ACK
AIF	是	-	RAFIE	否	-
ALC	是	-	RAFIF	是	-
ALIE	否	-	RBALL	是	-
ALIF	是	-	RBUF	是	RB被标记为空，数值不定
AMASK	否	只能在软件复位时可写	RECNT	否	通过BUSOFF写1清零
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	
EWL	是	-	ROV	是	

寄存器位名	软件复位	备注	寄存器位名	软件复位	备注
RREL	是	-	TSMODE	否	
RSTAT	是		TSNEXT	是	-
SACK	是	-	TSONE	是	-
SELMASK	否	-	TPIE	否	-
S_PRESC	否	只能在软件复位时可以写	TPIF	是	-
S_SEG_1	否	只能在软件复位时可以写	TPSS	是	-
S_SEG_2	否	只能在软件复位时可以写	TSFF	是	所有STB SLOT被标记为空
SJW	否	只能在软件复位时可以写	TSIE	否	-
SSPOFF	是	-	TSIF	是	-
TACTIVE	是	发送立即停止	TSSS	是	-
TBE	是	-	TSSTAT	是	所有STB SLOT被标记为空
TBF	否	-	TTEN	是	-
TBPTR	否	-	TTIF	是	-
TBSEL	是	-	TTIE	否	-
TBUF	是	STB被标记为空，指向PTB	TTPTR	否	-
TDCEN	是	-	TTTBM	否	-
TECNT	否	可通过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	

表 37-3 软件复位范围表

37.4.16 向上兼容 CAN-FD 功能

CAN-CTRL 在 CAN FD 功能禁止时即使在包含 CAN FD 网络中收到 CAN FD 的帧，接收器会自动忽略这些帧，不返回 ACK，等到总线空闲时，再发送或者接收下一个 CAN2.0B 的帧。

37.4.17 时间触发 TTCAN

CAN-CTRL 为 ISO11898-4 规定的时间触发通信方式提供部分(lever 1)硬件支持。本章节从以下 5 个部分介绍 TTCAN 功能。

37.4.17.1 TTCAN 模式下的 TBUF 行为

TTTBM=1

TTTBM=1 时，PTB 和 STB SLOT 一样组成 TB SLOT，通过 TBPTR 寄存器指定发送 BUF，其中 TBPTR=0 时，指向 PTB，TBPTR=1 是指向 STB SLOT1，依次类推。主机可以通过 TPE 和 TPF 寄存器来标记发送 BUF SLOT。此时 TBSEL 和 TSNEXT 寄存器无任何意义从而可以被忽略。

TTTBM=1 时，PTB 不具有任何特殊的属性，和 STB SLOT 一样，传送完成标志也采用 TSIF。

TTCAN 模式时，发送 BUF 没有 FIFO 模式和优先级仲裁模式，同时也只有一个选定的 SLOT 可以发送数据。

TTCAN 模式下，传输开始需要采用时间触发方式，TPE，TSONE，TSALL，TPSS 和 TPA 被固定为 0 且被忽略。

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。

37.4.17.2 TTCAN 功能描述

上电后，Time Master 需要根据 ISO 11898-4 协议进行初始化。一个 CAN 网络中，最多可以有 8 个潜在的 Time Master。每一个 Time Master 都具有自己的参考消息 ID（ID

最后 3 位）。这些潜在的 Time Master 根据自己的优先级发送各自的参考消息。

TTEN=1 后，16 位的计数器开始工作，当参考消息被成功接收或者 Time Master 成功发送参考消息时，CAN 控制器将 Sync_Mark 拷贝给 Ref_Mark，Ref_Mark 将 cycle time 设置为 0。成功接收参考消息置位 RIF 标志而成功发送参考消息置位 TPIF 标志或者 TSIF 标志。此时主机需要准备下一个动作的触发条件。

触发条件可以是接收触发。该触发仅触发中断可用于检测期待的消息没有被收到。

触发条件也可以是发送触发。该触发开始发送通过 TTPTR 寄存器指定的 TBUF SLOT 里的数据。如果选定的 TBUF SLOT 被标记为空，则不开始发送，但置位中断标志。

37.4.17.3 TTCAN 时序

CAN_CTRL 支持 ISO11898-4 level 1。包含的一个 16 位计数器工作在 S_PRESC，S_SEG_1，S_SEG_2 定义的位时间下。如果 TTEN=1，则有一个额外的预分频器 T_PRESC。

一帧数据的 SOF 时，计数器的值为 Sync_Mark。如果该帧数据为参考消息，则将 Sync_Mark 拷贝给 Ref_Mark。cycle time 等于计数器的值减去 Ref_Mark。该时间用作为接收消息的时间戳或者发送消息的触发时间基准。

37.4.17.4 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。TTTBM=1 时，只支持时间触发方式。

立即触发

通过写 TT_TRIG 的高位（不在意写入的值），启动触发器。此模式下，TTPTR 选定

的 TBUF SLOT 内的数据会立即发送。TTIF 不置位。

时间触发

时间触发方式仅通过置位 TTIF 标志产生中断，并无其他功能。如果一个节点期待在特定的时间窗口内收到期待的数据，则可以使用时间触发方式。如果 TT_TRIG 值小于实际的 cycle time，则 TEIF 置位且无其它动作。

单次发送触发

单次发送触发方式用于在执行时间窗口内发送数据。此时，忽略 TSSS 位。

通过 TEW 位设定 ISO11898-4 规定的最多 16 个 cycle time 的 Tick，设定范围为 1~16。

如果在规定的发送使能时间窗口内数据没有开始发送，则帧被丢弃。对应的发送 BUF SLOT 被标记为空，并且置位 AIF，对应的发送 BUF 内的数据不会被改写，因为可以通过置位 TPF 再次发送。

如果 TT_TRIG 值小于实际的 cycle time，则 TEIF 置位且无其它动作。

发送开始触发

发送开始触发方式用于仲裁时间窗口内，参与仲裁。TSSS 用于决定是否自动重发或者单次发送模式。如果 TTPTR 寄存器指定的消息没有被成功发送，可以使用发送停止触发来停止该发送。

如果 TT_TRIG 值小于实际的 cycle time，则 TEIF 置位且无其它动作。

发送停止触发

发送停止触发方式用于停止通过发送开始触发方式已经开始的发送。如果发送被停止，则发送帧被舍弃，置位 AIF 并将选定的 TBUF SLOT 标记为空，但 TBUF SLOT 内的数据不会被改写，可以通过置位 TPF 就可以再次发送。

如果 TT_TRIG 值小于实际的 cycle time 则 TEIF 置位且执行停止。

37.4.17.5 TTCAN 触发看门时间

TTCAN 触发看门时间功能类似于看门狗功能，在 TTTBM=1 时使用。用来看门从上次成功接收到参考消息开始的时间。参考消息可以在周期 cycle time 中或者一个事件后被接收，应用程序应该根据具体情况设定合适的看门时间。

如果 cycle count 等于 TT_WTRIG，则置位 WTIF。通过 WTIE 写 0，关闭看门触发。

如果 TT_WTRIG 比实际的 cycle time 小，则 TEIF 置位。

37.4.18 中断

符号	中断标志	描述
CAN_1_HOST CAN_2_HOST	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 37-4 CAN 中断表

37.5 寄存器说明

CAN_1_BASE_ADDR:0x40009000 CAN_2_BASE_ADDR: 0x40078000

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	-	0xXXXX XXXX
CAN发送BUF寄存器	CAN_TBUF	0x50~0x5F	-	0xXXXX 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	CAN_RTIE	0xA4	8	0xFE
CAN接收和发送中断标志寄存	CAN_RTIF	0xA5	8	0x00
CAN错误中断使能和标志寄存	CAN_ERRINT	0xA6	8	0x00
CAN警告限定寄存器	CAN_LIMIT	0xA7	8	0x1B
CAN Slow位时序寄存器	CAN_SBT	0xA8	32	0x0102 0203
CAN错误和仲裁失败捕捉寄存	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_TECNT	0xB3	8	0x00
CAN筛选器组控制寄存器	CAN_ACFCTRL	0xB4	8	0x00
CAN筛选器组使能寄存器	CAN_ACFEN	0xB6	8	0x01
CAN筛选器组code和mask寄存	CAN_ACF	0xB8	32	0xXXXX 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	0xXXXX XXXX
TTCAN 触发配置寄存器	CAN_TRG_CFG	0xC4	16	0x0000
TTCAN 触发时间寄存器	CAN_TT_TRIG	0xC6	16	0x0000
TTCAN 触发看门时间寄存器	CAN_TT_WTRIG	0xC8	16	0xFFFF

表 37-5 CAN 寄存器一览表

37.5.1 CAN 接收 BUF 寄存器（CAN_RBUF）

CAN Receive Buffer Registers

偏移地址：0x50

复位值：0xXXXX XXXX

RBUF 寄存器指向最早接收到的 CAN 邮箱的 RB SLOT 地址，RBUF 寄存器可以按照任意顺序读取。

KOER 位即为寄存器 EALCAP.KOER，仅在 RBALL=1 时有意义。

TX 位表示在回环模式下接受到自己发送的消息。

CYCLE_TIME 位仅在 TTCAN 模式时有效，表示 SOF 开始时的 cycle time。

CAN_RBUF 仅支持 WORD 访问。

CAN 接收邮箱的数据格式如下：

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	-	-	DLC[3:0]				Control
RBUF+5	KOER[2:0]			TX	-				Status
RBUF+6	CYCLE_TIME[7:0]								TTCAN
RBUF+7	CYCLE_TIME[15:8]								TTCAN
RBUF+8	DATA1								Data
RBUF+9	DATA2								Data
RBUF+10	DATA3								Data
…	…								Data
…	…								Data
…	…								Data
RBUF+14	DATA7								Data
RBUF+15	DATA8								Data

表 37-6 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	-	-	DLC[3:0]				Control
RBUF+5	KOER[2:0]			TX	-				Status
RBUF+6	CYCLE_TIME[7:0]								TTCAN
RBUF+7	CYCLE_TIME[15:8]								TTCAN
RBUF+8	DATA1								Data
RBUF+9	DATA2								Data
RBUF+10	DATA3								Data
…	…								Data
…	…								Data
…	…								Data
RBUF+14	DATA7								Data
RBUF+15	DATA8								Data

表 37-7 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，CAN2.0 设定范围为 0~8，对应数据长度为 0Byte~8Byte

DLC(二进制)	Frame Type	Playload in Bytes
0000~1000	CAN2.0	0~8
10001~1111	CAN2.0	8

表 37-8 DLC 控制位

状态位含义如下：

KOER：同 EALCAP.KOER

TX：回环模式中接收自己发送的数据时此位置 1

37.5.2 CAN 发送 BUF 寄存器（CAN_TBUF）

CAN Transmit Buffer Registers

偏移地址：0x00

复位值：0xXXXX XXXX

TBUF 寄存器指向下一个空的 CAN 发送 BUF SLOT，TBUF 寄存器可以按照任意顺序读取。通过软件将 TSNEXT 写 1 来标记对应的 TBUF SLOT 已经写入数据，从而指向下一个 TBUF SLOT。

TBUF 只能 WORD 访问。

CAN 发送邮箱的数据格式如下：

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	-	-	DLC[3:0]				Control
TBUF+5	-								-
TBUF+6	-								-
TBUF+7	-								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
…	…								Data
…	…								Data
…	…								Data
TBUF+14	DATA7								Data
TBUF+15	DATA8								Data

表 37-9　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	-	-	DLC[3:0]				Control
TBUF+5	-								-
TBUF+6	-								-
TBUF+7	-								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
…	…								Data
…	…								Data
…	…								Data
TBUF+14	DATA7								Data
TBUF+15	DATA8								Data

表 37-10 扩展格式 CAN 发送邮箱格式

控制位含义请参考 CAN 接收 BUF 寄存器章节说明

37.5.3 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 times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

37.5.4 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/W
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&TTTBM=1	R/W
b2	TSONE	发送一帧STB数据设定位	发送一帧STB数据设定位（Transmit Secondary ONE frame） 0：不发送 1：发送一帧STB数据 FIFO模式中，发送最早写入的数据，优先级模式里发送最高优先级的数据 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&TTTBM=1	R/W
b1	TSALL	发送所有的STB数据设定位	发送所有的STB数据设定位（Transmit Secondary ALL frame） 0：不发送 1：发送STB中所有的数据 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 – BUSOFF=1 – LOM=1&LBME=0 – TTEN=1&TTTBM=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

37.5.5 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/W
b6	TSNEXT	下一个STB SLOT	下一个STB（Transmit buffer Secondary NEXT） 0：无动作 1：当前STB SLOT已填充，指向下一个SLOT 应用程序将TBUF中的数据写完后，通过置位TSNEXT位标识当前STB SLOT已经被填充，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。	R/W
b5	TSMODE	STB发送模式	STB发送模式（Transmit buffer Secondary operation MODE） 0：FIFO模式 1：优先级模式 FIFO模式根据数据帧写入的先后顺序发送。 优先级模式根据ID自动判断，ID越小，优先级越高。 无论何种模式，PTB具有最高的优先级。 注意：TSMODE位只能在STB空时设定。	R/W
b4	TTTBM	TTCAN BUF模式	TTCAN BUF模式（TTCAN Transmit Buffer Mode） TTEN=0时，TTTBM被忽略。 0：TSMODE决定，PTB和STB 1：通过TBPTR和TTPTR设定 TTCAN模式时，只需要接收时间戳功能时，此位可以设置为0，通过TSMODE决定使用PTB还是STB。 注意：TSMODE位只能在STB空时设定。	R/W
b3~b2	Reserved	-	必须保持复位值。	R/W

位	标记	位名	功能	读写
b1~b0	TSSTAT	STB状态	STB状态（Transmission Secondary Status bits）	R
			TTEN=0 或 TTEN=1 &TTTBM=0	
			00：STB空	
			01：STB小于等于半满	
			10：STB大于半满	
			11：STB满	
			TTEN=1 且 TTTBM=1	
			00：PTB和STB空	
			01：PTB和STB非满	
			10：保留	
			11：PTB和STB满	

37.5.6 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	-	RSSTAT[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数据存储所有的数据帧	接收BUF数据存储素所有的数据帧（Receive Buffer stores ALL data frames） 0：正常模式 1：存储所有的数据包括有错误的数据。	R/W
b2	Reserved	-	必须保持复位值。	R/W
b1~b0	RSTAT	接收BUF状态	接收BUF状态（Receive buffer STATus） 00：RBUF空 01：RBUF非空但小于AFWL编程值 10：RBUF大于等于AFWL编程值但未满 11：满（上溢时保持此值）	R

37.5.7 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

37.5.8 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

位	标记	位名	功能	读写
b7	RIF	接收中断标志	接收中断标志（Receive Interrupt Flag） 0：未收到数据帧 1：接收到有效的数据帧或者远程帧 通过应用程序写1清0。	R/W
b6	ROIF	接收上溢中断标志	接收上溢中断标志（Receive Overrun Interrupt Flag） 0：无RB被覆盖（overwrite） 1：RB至少有一个被覆盖 上溢时ROIF和RFIF同时置1。 通过应用程序写1清0。	R/W
b5	RFIF	接收BUF满中断标志	接收BUF满中断标志（RB Full Interrupt Flag） 0：RB FIFO未满 1：RB FIFO满 通过应用程序写1清0。	R/W
b4	RAFIF	接收BUF将满中断标志	接收BUF将满中断标志（RB Almost Full Interrupt Flag） 0：被填充的RB SLOT数目小于AFWL设定值 1：被填充的RB SLOT数目大于等于AFWL设定值 通过应用程序写1清0。	R/W
b3	TPIF	PTB发送中断标志	PTB发送中断标志（Transmission Primary Interrupt Flag） 0：没有PTB发送完成 1：请求的PTB发送成功完成 通过应用程序写1清0。 注意：TTCAN模式时，TPIF无效，仅适用TSIF标志	R/W
b2	TSIF	STB发送中断标志	STB发送中断标志（Transmission Secondary Interrupt Flag） 0：没有STB发送完成 1：请求的STB发送成功完成 通过应用程序写1清0。 注意：TTCAN模式时，TPIF无效，仅使用TSIF标志	R/W

位	标记	位名	功能	读写
b1	EIF	错误中断标志	错误中断标志（Error Interrupt Flag） 0：BUSOFF位未发生变化，或者错误计数器的值与ERROR warning limit设定值的相对关系未发生变化。 1：BUSOFF位发生变化，或者错误计数器的值与ERROR warning limit设定值的相对关系发生变化。比如错误计数器的值从小于设定值变为大于设定值，或者从大于设定值变为小于设定值。 通过应用程序写1清0。	R/W
b0	AIF	取消发送中断标志	取消发送中断标志（Abort Interrupt Flag） 0：未取消发送数据 1：通过TPA和TSA请求的发送消息被成功取消。 通过应用程序写1清0。	R/W

37.5.9 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	错误被动（Error Passive mode active） 0：节点是主动错误节点 1：节点时被动错误节点 通过应用程序写1清0。	R
b5	EPIE	错误被动中断使能	错误被动中断使能（Error Passive Interrupt Enable） 0：禁止 1：使能	R/W
b4	EPIF	错误被动中断标志	错误被动中断标志（Error Passive Interrupt Flag） 0：未发生错误主动到错误被动或者错误被动到错误主动的变化 1：发生错误主动到错误被动或者错误被动到错误主动的变化 通过应用程序写1清0。	R/W
b3	ALIE	仲裁失败中断使能	仲裁失败中断使能（Arbitration Lost Interrupt Enable） 0：禁止 1：使能	R/W
b2	ALIF	仲裁失败中断标志	仲裁失败中断标志（Arbitration Lost Interrupt Flag） 0：仲裁成功 1：仲裁失败 通过应用程序写1清0。	R/W
b1	BEIE	总线错误中断使能	总线错误中断使能（Bus Error Interrupt Enable） 0：禁止 1：使能	R/W
b0	BEIF	总线错误中断标志	总线错误中断标志（Bus Error Interrupt Flag） 0：无总线错误 1：总线错误 通过应用程序写1清0。	R/W

37.5.10 CAN Slow 位时序寄存器（CAN_SBT）

CAN Slow Bit Timing Register

偏移地址：0xA8

复位值：0x0102 0203

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
S_PRESC[7:0]								-	S_SJW[6:0]						
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	S_SEG_2[6:0]							S_SEG_1[7:0]							

位	标记	位名	功能	读写
b31~b24	S_PRESC	预分频设定	预分频设定（S_Prescaler） 此寄存器设定值时将模块通信时钟设定为（S_PRESC+1）分频作为TQ。	R/W
b23	Reserved	-	必须保持复位值。	R/W
b22~b16	S_SJW	再同步补偿宽度时间设定	再同步补偿宽度时间设定（Bit Timing Segment 2） 再同步补偿宽度时间=(S_SJW+1)*TQ	R/W
b15	Reserved	-	必须保持复位值。	R/W
b14~b8	S_SEG_2	位段2时间设定	位段2时间单元设定（Bit Timing Segment 2） 位段2时间=(S_SEG_2+1)*TQ	R/W
b7~b0	S_SEG_1	位段1时间设定	位段1时间单元设定（Bit Timing Segment 1） 位段1时间=(S_SEG_1+2)*TQ	R/W

37.5.11 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） 000：无错误 001：位错误 010：形式错误 011：填充错误 100：应答错误 101：CRC错误 110：其他错误 111：保留 有错误时KOER位更新，正常发送接收时KOER位保持不变。	R
b4~b0	ALC	仲裁失败位置捕捉	仲裁失败位置捕捉（Arbitration Lost Capture） 仲裁失败时ALC记录一帧数据中仲裁失败时的位置。	R/W

37.5.12 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1~8。 AFWL=0无意义，当做AFWL=1处理。	R/W
b3~b0	EWL	Error Waring Limit编程值	Error Waring Limit编程值（Programmable Error Warning Limit） Error Waring Limit=（EWL+1）*8。 该寄存器设定值影响EIF标志。	R/W

37.5.13 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

37.5.14 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

37.5.15 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/W
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/W
b3~b0	ACFADR	筛选器地址	筛选器地址（acceptance filter address） ACFADR指向具体的筛选器，通过SELMASK去区分ID和MASK。 0000：指向ACF_1 0001：指向ACF_2 0010：指向ACF_3 0011：指向ACF_4 0100：指向ACF_5 0101：指向ACF_6 0110：指向ACF_7 0111：指向ACF_8 1000：指向ACF_9 1001：指向ACF_10 1010：指向ACF_11 1011：指向ACF_12 1100：指向ACF_13 1101：指向ACF_14 1110：指向ACF_15 1111：指向ACF_16	R/W

37.5.16 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：0x0001

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AE_16	AE_15	AE_14	AE_13	AE_12	AE_11	AE_10	AE_9	AE_8	AE_7	AE_6	AE_5	AE_4	AE_3	AE_2	AE_1

位	标记	位名	功能	读写
b15	AE_16	ACF_16使能	ACF_16使能（Acceptance Filter 16 Enable） 0：禁止 1：使能	R/W
b14	AE_15	ACF_15使能	ACF_15使能（Acceptance Filter 15 Enable） 0：禁止 1：使能	R/W
b13	AE_14	ACF_14使能	ACF_14使能（Acceptance Filter 14 Enable） 0：禁止 1：使能	R/W
b12	AE_13	ACF_13使能	ACF_13使能（Acceptance Filter 13 Enable） 0：禁止 1：使能	R/W
b11	AE_12	ACF_12使能	ACF_12使能（Acceptance Filter 12 Enable） 0：禁止 1：使能	R/W
b10	AE_11	ACF_11使能	ACF_11使能（Acceptance Filter 11 Enable） 0：禁止 1：使能	R/W
b9	AE_10	ACF_10使能	ACF_10使能（Acceptance Filter 10 Enable） 0：禁止 1：使能	R/W
b8	AE_9	ACF_9使能	ACF_9使能（Acceptance Filter 9 Enable） 0：禁止 1：使能	R/W
b7	AE_8	ACF_8使能	ACF_8使能（Acceptance Filter 8 Enable） 0：禁止 1：使能	R/W
b6	AE_7	ACF_7使能	ACF_7使能（Acceptance Filter 7 Enable） 0：禁止 1：使能	R/W
b5	AE_6	ACF_6使能	ACF_6使能（Acceptance Filter 6 Enable） 0：禁止 1：使能	R/W
b4	AE_5	ACF_5使能	ACF_5使能（Acceptance Filter 5 Enable） 0：禁止 1：使能	R/W
b3	AE_4	ACF_4使能	ACF_4使能（Acceptance Filter 4 Enable） 0：禁止 1：使能	R/W
b2	AE_3	ACF_3使能	ACF_3使能（Acceptance Filter 3 Enable） 0：禁止 1：使能	R/W
b1	AE_2	ACF_2使能	ACF_2使能（Acceptance Filter 2 Enable） 0：禁止 1：使能	R/W
b0	AE_1	ACF_1使能	ACF_1使能（Acceptance Filter 1 Enable） 0：禁止 1：使能	R/W

37.5.17 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：0xXXXX XXXX

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	AIDEE	AIDE	ACODE[28:16] or AMASK_28:16]												
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ACODE[15:0] or AMASK[15:0]															

位	标记	位名	功能	读写
b31	Reserved	-	读出值为不定。	R/W
b30	AIDEE	IDE位比较使能	IDE位比较使能（Acceptance mask IDE bit check enable） 尽在SELMASK=1时有效 0：筛选器接收标准格式和扩展格式帧 1：筛选器接收AIDE位定义的标准格式或者扩展格式帧	R/W
b29	AIDE	IDE位MASK	IDE位MASK 0：筛选器仅接收标准格式 1：筛选器仅接收扩展格式	R/W
b28~b0	ACODE/ AMASK	筛选器CODE/ 筛选器MASK	筛选器CODE（acceptance filter code） 通过ACFADR指向具体的筛选器。 SELMASK=0时表示筛选器的CODE。 标准格式时使用位10~位0，扩展格式时使用位29~位0。 筛选器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29~位0。	R/W

37.5.18 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	TBF	设置TB为已填充	设置TB已填充（set TB slot to “Filled”） 0：无操作 1：被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/W
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 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

37.5.19 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） 0：禁止 1：使能	R/W
b6	WTIF	触发看门中断标志	触发看门中断标志（Watch Trigger Interrupt Flag） 当CYCLE COUNT值=TT_WTRIG设定值时且WTIE=1时，WTIF置位。 通过应用程序写1清0。	R/W
b5	TEIF	触发错误中断标志	触发错误中断标志（Trigger Error Interrupt Flag） TT_TTIG设定值小于实际的CYCLE_TIME时，TEIF置位。	R
b4	TTIE	时间触发中断使能	时间触发中断使能（Time Trigger Interrupt Enable） 0：禁止 1：使能	R/W
b3	TTIF	时间触发中断标志	时间触发中断标志（Time Trigger Interrupt Flag） 当CYCLE COUNT值=TT_TRIG设定值时且TTIE=1时，TTIF置位。 如果TT_TRIG没有更新，则TTIF只置位1次，下一个基本CYCLE不置位。 通过应用程序写1清0。	R/W
b2~b1	T_PRESC	TTCAN计数器预分频	TTCAN计数器预分频（TTCAN Timer PRESCaler） 00：SBT寄存器设定的位时间的1分频 01：SBT寄存器设定的位时间的2分频 10：SBT寄存器设定的位时间的4分频 11：SBT寄存器设定的位时间的8分频 注意：T_PRESC可在TTEN=0时进行写操作或者写TTEN=1时同时操作。	R/W
b0	TTEN	TTCAN使能	TTCAN使能（Time Trigger Enable） 0：禁止 1：使能TTCAN，计数器开始计数。	R/W

37.5.20 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位	参考消息的IDE位（REFerence message IDE bit） 0：标准格式 1：扩展格式	R/W
b30~b29	Reserved	-	读出值为不定。	R/W
b28~b0	REF_ID	参考消息的ID位	参考消息的ID位（REFerence message IDentifier） REF_IDE=0：REF_ID[10:0]有效 REF_IDE=1：REF_ID[28:0]有效 REF_ID用于检测参考消息，适用于发送和接收。 检测到参考消息后，当前帧的Sync_Mark则变成Ref_Mark。 REF_ID[2:0]固定为0，并不检查其值，这样最多可以支持8个潜在的time master。 当REF_MSG的最高字节写操作后，则需要等待6个CAN时钟周期以完成REF_MSG向CAN时钟域的传递。	R/W

37.5.21 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/W
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_TRIG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/W
b2~b0	TTPTR	发送触发器TB slot指针	发送触发器TB slot指针（Transmit Trigger TB slot Pointer） 000：指向PTB 001：指向STB SLOT1 010：指向STB SLOT2 011：指向STB SLOT3 其他：设定禁止 如果指向的TB SLOT被标记为空，当到达触发时间后，TEIF置位。	R/W

37.5.22 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

37.5.23 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_TWRIG	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

38 SDIO 控制器（SDIOC）

38.1 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。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。

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

38.2 功能描述

38.2.1 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。

38.2.2 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。

38.2.3 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。

38.2.4 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(PCLK1)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 PCLK1 的频率设定合理的分频系数。

38.2.5 中断和 DMA 启动请求

38.2.5.1 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。

38.2.5.2 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(NORINTSTEN)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)。

38.2.5.3 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 SDIOx_Dy(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。

38.2.5.4 卡的插入(insert)和移除(removal)

SD/MMC/SDIO 设备的插入(insert)和移除(removal)由 SDIOx_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 SDIOx_CD(x=1~2)信号拉高。当有设备插入后，SDIOx_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 SDIOx_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。

38.2.6 主机和设备初始化

38.2.6.1 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 PCLK1 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

38.2.6.2 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。

38.2.6.3 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。

38.2.6.4 SDIO 初始化

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

38.2.7 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 PCLK1 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. 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 卡切换卡至高速模式(high speed mode，fpp<=50MHz)，切换成功后可将 SDIOx_CK(x=1~2)设置为最高 50MHz
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=1~2)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。

38.2.8 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 PCLK1 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 卡切换卡至高速模式(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，首先发送多 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。

38.2.9 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。

38.2.10 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。

38.2.11 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 SDIOx_CD(x=1~2)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 SDIOx_CD(x=1~2)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

38.3 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)。

寄存器列表

英文名称	中文名称	英文缩写	偏移地址
Block Size	数据块长度	BLKSIZE	0x04
Block Count	数据块计数	BLKCNT	0x06
Argument0	参数0	ARG0	0x08
Argument1	参数1	ARG1	0x0A
Transfer Mode	传输模式	TRANSMODE	0x0C
Command	命令	CMD	0x0E
Response0	应答0	RESP0	0x10
Response1	应答1	RESP1	0x12
Response2	应答2	RESP2	0x14
Response3	应答3	RESP3	0x16
Response4	应答4	RESP4	0x18
Response5	应答5	RESP5	0x1A
Response6	应答6	RESP6	0x1C
Response7	应答7	RESP7	0x1E
Buffer Data Port0	数据缓冲0	BUF0	0x20
Buffer Data Port1	数据缓冲1	BUF1	0x22
Present State	主机状态	PSTAT	0x24
Host Control	主机控制	HOSTCON	0x28
Power Control	电源控制	PWRCON	0x29
Block Gap Control	数据块间隔控制	BLKGPCON	0x2A
Clock Control	时钟控制	CLKCON	0x2C
Timeout Control	超时控制	TOOUTCON	0x2E
Software Reset	软件复位	SFTRST	0x2F
Normal Interrupt Status	普通中断状态	NORINTST	0x30
Error Interrupt Status	错误中断状态	ERRINTST	0x32

Normal Interrupt Status Enable	普通中断状态使能	NORINTSTEN	0x34
Error Interrupt Status Enable	错误中断状态使能	ERRINTSTEN	0x36
Normal Interrupt Signal Enable	普通中断信号使能	NORINTSGEN	0x38
Error Interrupt Signal Enable	错误中断信号使能	ERRINTSGEN	0x3A
Auto CMD Error Status	自动命令错误状态	ATCERRST	0x3C
Force Event for Auto CMD Error Status	强制自动命令错误状态控制	FEA	0x50
Force Event for Error Interrupt Status	强制错误状态控制	FEE	0x52

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

英文名称	中文名称	英文缩写	地址
MMC Enable Register	MMC 模式使能寄存器	MMCER	0x40055404

38.3.1 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/W
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

38.3.2 数据块计数存器(BLKCNT)

偏移地址：0x6

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

38.3.3 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

38.3.4 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

38.3.5 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/W
b5	MULB	多数据块	0：当前传输为单数据块传输(single block)	R/W
			1：当前传输为多数据块传输(multi block)	
b4	DDIR	数据传输方向	0：写操作(主机发送数据)	R/W
			1：读操作(主机接收数据)	
b3~b2	ATCEN	自动命令使能	00：不发送自动命令	R/W
			01：多块传输结束后自动发送CMD12	
			10：禁止设定	
			11：禁止设定	
b1	BCE	块计数使能	0：禁用数据块计数使能	R/W
			1：允许数据块计数使能	
b0	Reserved	-	读时读出0，写时请写0	R/W

38.3.6 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/W
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/W
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

38.3.7 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

38.3.8 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

38.3.9 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

38.3.10 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

38.3.11 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

38.3.12 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

38.3.13 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

38.3.14 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

38.3.15 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

38.3.16 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

38.3.17 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/W
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/W
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/W
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

38.3.18 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/W
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/W

38.3.19 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/W
b0	PWON	SDIOC使能	0：禁用SDIOC 1：启用SDIOC	R/W

38.3.20 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/W
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

38.3.21 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					CE	Rsvd	ICE

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PCLK1 0x80：PCLK1的256分频 0x40：PCLK1的128分频 0x20：PCLK1的64分频 0x10：PCLK1的32分频 0x08：PCLK1的16分频 0x04：PCLK1的8分频 0x02：PCLK1的4分频 0x01：PCLK1的2分频 0x00：PCLK1 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

38.3.22 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/W
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PCLK1的时钟周期 0000：PCLK1×2^{13} 0001：PCLK1×2^{14} …… 1110：PCLK1×2^{27} 1111：禁止设定	R/W

38.3.23 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/W
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

38.3.24 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/W
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/W
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

38.3.25 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/W
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/W
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

38.3.26 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/W
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/W
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

38.3.27 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
Reserved							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/W
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/W
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

38.3.28 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/W
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/W
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/W
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

38.3.29 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/W
b8	ACESEN	自动发送命令错误信号使能	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/W
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校验错误信号使能	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

38.3.30 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
Reserved								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/W
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/W
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

38.3.31 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
Reserved								FCMDE	Reserved		FIE	FEBE	FCE	FTOE	FNE

位	标记	位名	功能	读写
b15~b8	Reserved	-	读时读出0，写时请写0	R/W
b7	FCMDE	强制未发送错误	0：无任何效果 1：强制发生ATCERRST.CMDE错误 读该寄存器位无任何效果	R/W
b6~b5	Reserved	-	读时读出0，写时请写0	R/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 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/W

38.3.32 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/W
b8	FACE	强制自动发送命令错误	0：无任何效果 1：强制发生ERRINTST.ACE错误 读该寄存器位无任何效果	R/W
b7	Reserved	-	读时读出0，写时请写0	R/W
b6	FDEBE	强制数据停止位错误	0：无任何效果 1：强制发生ERRINTST.DEBE错误 读该寄存器位无任何效果	R/W
b5	FDCE	强制数据CRC校验错误	0：无任何效果 1：强制发生ERRINTST.DCE错误 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/W
b2	FCEBE	强制命令停止位错误	0：无任何效果 1：强制发生ERRINTST.CEBE错误 读该寄存器位无任何效果	R/W
b1	FCCE	强制命令CRC校验错误	0：无任何效果 1：强制发生ERRINTST.CCE错误 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/W

38.3.33 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												SELMMC2	Reserved	SELMMC1	Reserved

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	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/W
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/W

39 以太网 MAC 控制器（ETHMAC）

39.1 概要

以太网 MAC 控制器（ETHMAC）用于在以太网网络中按照 IEEE802.3-2002 标准发送和接收数据，有多种应用领域，如交换机、网络接口卡等。该 MAC 控制器支持与外部物理层（PHY）相连的两个工业标准接口：介质独立接口（MII）（在 IEEE802.3 规范中定义）和简化介质独立接口（RMII）。

主要遵循以下协议规范：

IEEE802.3-2002，用于以太网 MAC

IEEE1588-2008 标准，用于规定联网时钟同步

AMBA2.0，用于 AHB 主/从端口

RMII 接口规范

39.2 基本特性

39.2.1 基本框图

下图为以太网 MAC 控制器的架构框图。

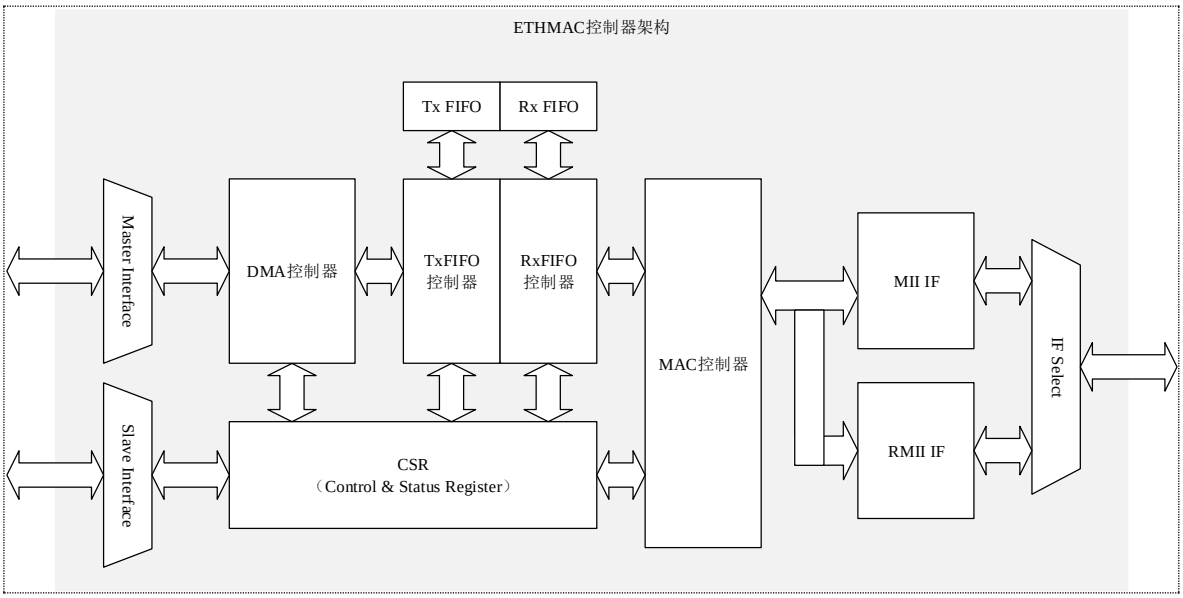


图 39-1 以太网 MAC 控制器架构图

从图中可以看出，以太网 MAC 控制器的内核部分包括 AHB Master 接口、AHB Slave 接口、Rx FIFO、Tx FIFO 和 DMA 控制器、MAC 控制器、接口控制逻辑等。DMA 控制器通过 AHB 主从接口将 MAC 内核和系统存储器相连。

AHB Master 接口用于数据传输，AHB Slave 接口用于该控制器的基本寄存器访问配置等。

在发送数据时，首先将数据从系统存储器由 DMA 控制器送至 Tx FIFO 进行缓冲，经 MAC 控制器的一系列处理（COE 控制、PTP 控制、MMC 控制等）后通过 MII 接口或 RMII 接口送至外部 PHY。

在接收数据时，经过 MII 接口或 RMII 接口接收进来的数据，经 MAC 控制器的一系列处理（COE 控制、PTP 控制、MMC 控制等）后送至 Rx FIFO，DMA 控制器再将 Rx FIFO 的数据传送到系统存储器。

39.2.2 ETH_MAC 特性

ETH_MAC 的基本功能特性如下：

- 支持外部 PHY 接口实现 10/100Mbps 数据传输速率
- 支持 MII 或 RMII 接口与外部快速以太网 PHY 进行通信
- 为应用程序提供单独的发送、接收和控制接口
- 使用 SMI 接口配置和管理最多 32 个 PHY 设备
- 支持检测远程唤醒帧和 AMD Magic Packet™帧
- 支持通过 MII 进行内部回送的 LoopBack 模式
- 支持全双工和半双工操作
  - 支持适用于半双工操作的 CSMA/CD 协议
  - 支持适用于半双工操作的背压流量控制
  - 支持适用于全双工操作的 IEEE802.3x 流量控制
  - 全双工操作时可以将接收的暂停控制帧转发到应用

    注：
    - 该章节中所述“应用”，是指在 MAC 将接收到的帧通过 RxDMA 传送到系统存储器，供上层的网络层、传输层使用；或者将上层网络层、传输层的数据从系统控制器通过 TxDMA 传送到 MAC，上层的网络层、传输层称之为应用。
  - 全双工操作中如果流量控制输入信号消失，将自动发送零时间片暂停帧
- 以太网帧处理
  - 接收以太网帧时，自动去除长度字段小于 1536 的帧的 PAD 和 FCS 字段
  - 接收以太网帧时，自动去除类型帧的 FCS 字段
  - 接收以太网帧时，自动进行接收帧的 CRC 计算
  - 发送以太网帧时，对 SA 进行插入、替换
  - 发送以太网帧时，对 CRC 进行插入、替换
  - 发送以太网帧时，对 VLAN 帧标识进行插入、替换、删除
  - 发送以太网帧时，对小于 60 字节的帧自动生成 PAD 填充
  - 发送以太网帧时，处理冲突帧的自动重新发送
  - 发送以太网帧时，支持可编程帧间隔（40-96 位时间，以 8 为步长）
  - 接收或发送以太网帧时，对帧长度可进行编程，支持高达 16KB 的巨型帧

- 以太网帧地址过滤
  - 支持高达五个 48 位完美（DA）地址过滤器，对每个字节进行掩码操作
  - 支持高达四个 48 位完美（SA）地址过滤器，对每个字节进行掩码操作
  - 支持 64 位 Hash 滤波器，适用于单播和多播目标地址过滤
  - 可传送所有多播地址帧
  - 支持混合模式，可传送所有帧，无需为网络监视进行过滤
  - 传送所有传入数据包时（每次过滤时）均附有一份状态报告
- VLAN 标识符过滤
  - 支持一组 VLAN 标记过滤器
  - 支持 12 位/16 位 VLAN 标记选择
  - 支持 VLAN 标记 Hash 滤波方式
- LAY3 IP 报地址过滤
  - 支持一组 LAY3 IP 报地址过滤器，对每个字节进行掩码操作
  - 支持支持 IPv4 报、IPv6 报的地址过滤选择
- LAY4 TCP/UDP 报端口过滤
  - 支持一组 LAY4 TCP/UDP 报端口过滤器
- 远程唤醒帧过滤模式
  - 支持四组远程唤醒帧过滤器，控制四组掩码操作
- COE 引擎
  - 在接收帧时，可选择丢弃无 TCP/UDP 字段的帧
  - 对接收到的 IPv4 报的 Header Checksum 字段进行校验
  - 对接收到的 TCP 报、UDP 报、ICMP 报的 Checksum 字段进行校验
  - 对发送的 IPv4 报进行 Checksum 计算，将计算结果插入到 Header Checksum 字段
  - 对发送的 TCP 报、UDP 报、ICMP 报进行 Checksum 计算，将计算结果插入到 Checksum 字段
- 支持 MMC 计数器进行网络统计
- FIFO 处理
  - 具有一个可编程阈值的 2KB 发送 FIFO（Tx FIFO）和一个可配置阈值的 2KB 接

收 FIFO（Rx FIFO）

- Rx FIFO 和 Tx FIFO 均支持存储转发模式
- 软件控制刷新 Tx FIFO
- Rx FIFO 在存储转发模式下，可选择过滤所有的错误帧，不将这些错误帧转发给应用
- Rx FIFO 可根据填充（阈值可配置）级别自动生成要发送给 MAC 内核的暂停控制帧或背压信号
- Rx FIFO 中丢失或损坏的帧可进行数据统计
- Rx FIFO 进行多帧存储时，通过在 EOF 传输后向 Rx FIFO 插入接收状态矢量，从而使得 Rx FIFO 无需存储这些帧的接收状态

39.2.3 ETH_PTP 特性

ETH_PTP 的基本功能特性如下：

- 支持接收帧和发送帧进行时间戳快照，插入时间戳
- 支持对系统时间进行粗校准和细校准
- 支持 2 组目标时间设定，系统时间大于目标时间时会触发中断
- 支持 2 组 PPS 输出，可以作为事件输出来控制其它模块的动作

39.2.4 ETH_DMA 特性

ETH_DMA 的基本功能特性如下：

- 支持软件设定 AHB Master 接口的 AHB 突发类型（固定或不确定突发）
- AHB Slave 接口中支持所有 AHB 突发类型
- AHB Master 接口可选择地址对齐突发传输
- 支持 RxDMA 和 TxDMA 引擎单独编程突发长度，以充分利用总线
- 支持对数据缓冲区进行字节对齐寻址
- 描述符特性：
  - 支持环式和链式两种描述符链接方式
  - 采用描述符架构可以在 CPU 几乎不干预的情况传输大型数据块
  - 每个描述符可传输高达 8KB 的数据

- TxDMA 和 RxDMA 均支持存储转发模式
- 报告正常工作和传输错误时的综合状态
- 可编程中断选项，根据不同的应用场景选择不同的中断
- 按帧控制发送/接收完成中断
- 接收引擎和发送引擎间采用循环调度仲裁或固定优先级仲裁

39.3 接口说明

以太网 MAC 控制器支持 MII、RMII、SMI 接口类型，其中用户根据实际需求选择 MII 接口和 RMII 接口中的一种作为数据交换的接口，SMI 接口独立存在，用于和 PHY 间进行基本配置相关的数据交换。

通过对接口控制寄存器（ETH_MAC_IFCONFR）的 IFSEL 位的设定，可选择以太网控制其使用 MII 接口还是 RMII 接口，在对 ETHMAC 的其它特性配置之前，该位必须优先设定。

39.3.1 MII 接口

介质独立接口（MII）定义了 10Mbps 和 100Mbps 的数据传输速率下 MAC 子层与 PHY 之间的互连。图 39-2 所示，是 MII 接口的基本连接图。

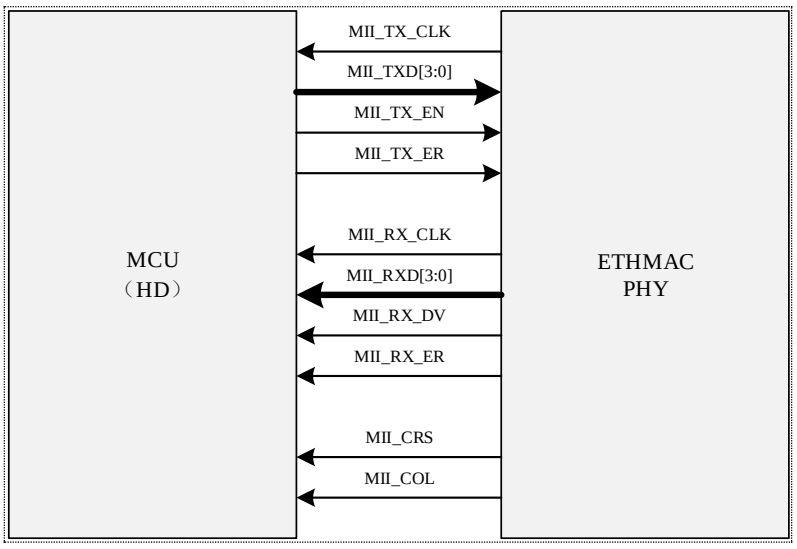


图 39-2 MII 接口连接图

表 39-1 介绍了 MII 接口各信号作用。

接口信号名	功能描述
MII_TX_CLK	发送时钟信号。 该信号提供进行 TX 数据传输时的参考时序。速率为 10Mbps 时为 2.5MHz；速率为 100Mbps 时为 25MHz。
MII_RX_CLK	接收时钟信号。 该信号提供进行 RX 数据传输时的参考时序。速率为 10Mbps 时为 2.5MHz；速率为 100Mbps 时为 25MHz。
MII_TX_EN	发送使能信号。 该信号表示 MAC 当前正对 MII 发送数据。
MII_TX_ER	发送错误信号。
MII_TXD[3:0]	发送数据信号。 该信号是 4 个一组的数据信号，由 MAC 子层同步驱动，在 MII_TX_EN 信号有效时才为有效信号。MII_TXD[0]为最低有效位，MII_TXD[3]为最高有效位。MII_TX_EN 信号无效时，发送数据不会对 PHY 产生任何影响。
MII_RX_DV	接收数据有效信号。 该信号表示 PHY 当前正对 MII 发送数据。为正确地接收帧，该信号必须在时间范围上涵盖要接收的帧，其开始时间不得迟于 SFD 字段出现的时间。
MII_RX_ER	接收错误信号。 该信号必须保持一个或多个发送时钟周期，从而向 MAC 子层指示在帧的某处检测到错误。该错误条件必须通过 RX_DV 信号验证（请参见表 40-3）。
MII_RXD[3:0]	接收数据信号。 该信号是 4 个一组的数据信号，由外部 PHY 同步驱动，在 MII_RX_DV 信号有效时才为有效信号。MII_RXD[0]为最低有效位，MII_RXD[3]为最高有效位。MII_RX_DV 信号无效、MII_RX_ER 信号使能时，特定的 MII_RXD[3:0]值用于传输来自 PHY 的特定信息（请参见表 40-3）。
MII_CRS	载波侦听信号。 当发送或接收介质处于非空闲状态时，由 PHY 使能该信号；发送和接收介质均处于空闲状态时，由 PHY 禁止该信号。PHY 必须确保 MII_CRS 信号在冲突条件下保持有效状态。该信号无需与 TX 和 RX 时钟保持同步。在全双工模式下，该信号没意义。
MII_COL	冲突检测信号。 检测到介质上存在冲突后，PHY 必须立即使能冲突检测信号，并且只要存在冲突条件，冲突检测信号必须保持有效状态。该信号无需与 TX 和 RX 时钟保持同步。在全双工模式下，该信号没意义。

表 39-1 MII 接口信号说明

表 39-2 介绍发送数据时的相关信号编码状态。

MII_TX_EN	MII_TX_ER	MII_TXD[3:0]	状态描述
0	0	0000~1111	正常帧间，无数据发送
0	1	0000~1111	无意义
1	0	0000~1111	正常发送数据
1	1	0000~1111	发送数据出现错误

表 39-2 发送数据时信号状态说明

表 39-3 介绍接收数据时的相关信号编码状态。

MII_RX_DV	MII_RX_ER	MII_RXD[3:0]	状态描述
0	0	0000~1111	正常帧间，无数据接收
0	1	0000	正常帧间
		0001~1101	无意义
		1110	错误载波检测
		1111	无意义
1	0	0000~1111	正常接收数据
1	1	0000~1111	接收数据出现错误

表 39-3 发送数据时信号状态说明

39.3.2 RMII 接口

精简介质独立接口（RMII）规范降低了 10/100Mbps 下微控制器以太网 MAC 控制器与外部 PHY 间的端口数。它所使用的端口数由 MII 的 16 个减少为 7 个，具体如图 39-3 所示。

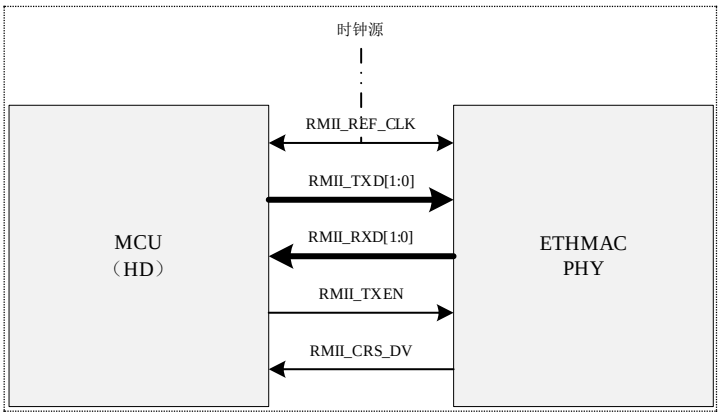


图 39-3 RMII 接口连接图

从连接图可以看出，RMII 接口和 MII 接口主要有以下几个不同点：

1) 发送时钟和接收时钟统一为参考时钟 RMII_REF_CLK，该时钟必须为 50MHz 时钟

2) 数据线由 4 根变为 2 根

3) MII_COL 信号丢弃，MII_RX_DV 和 MII_CRS 信号合并为 RMII_CRS_DV 信号

4) MII_TX_ER、MII_RX_ER 信号丢弃

39.3.3 SMI 接口

SMI 接口可通过以太网 MAC 控制器对 PHY 实现站点管理。MAC 控制器通过时钟和数据 2 根线访问任意的 PHY 寄存器。该接口最多支持访问 32 个 PHY。图 39-4 所示是 SMI 接口的连接图。

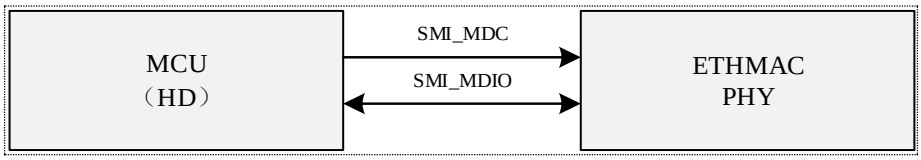


图 39-4 SMI 接口连接图

表 39-4 介绍了 SMI 接口各信号作用。

接口信号名	功能描述
SMI_MDC	周期性时钟，提供以最大频率 2.5MHz 传输数据时的参考时序。MDC 的最短高电平和最短低电平必须均为 160ns。MDC 的最小周期必须为 400 ns。在空闲状态下，SMI 将 MDC 时钟信号驱动为低电平。
SMI_MDIO	数据输入/输出比特流，用于通过 MDC 时钟信号与 PHY 设备进行数据信息传输。

表 39-4 SMI 接口信号说明

MAC 控制器可以从 32 个 PHY 中选择一个 PHY，然后从选中的 PHY 所包含的 32 个寄存器中选择一个寄存器，发送控制数据或接收状态信息。任意给定时间内只能对一个 PHY 中的一个寄存器进行寻址。

SMI 的帧格式如下表 39-5，在与 PHY 进行数据交换的过程中遵循以下格式：

	SMI 帧格式描述							
	Head（32bit）	Start	Operation	PhyAdd	RegAdd	TA	Data（16bit）	Idel
Read	1…1	01	10	xxxxx	xxxxx	Z0	xx…xx	Z
Write	1…1	01	01	xxxxx	xxxxx	10	xx…xx	Z

表 39-5 SMI 帧格式

39.3.4 以太网端口配置

依据上述 MII、RMII、SMI 3 种类型的数据接口，将以太网 MAC 控制器的端口功能分配与各类型接口的对应关系整理如表 39-6 所示：

ETH 端口名	接口功能信号
ETH_MII_RMII_RXCLK	MII_RX_CLK / RMII_REF_CLK
ETH_MII_TXCLK	MII_TX_CLK
ETH_MII_RMII_TXEN	MII_TX_EN / RMII_TX_EN
ETH_MII_TXER	MII_TX_ER
ETH_MII_RMII_TXD1~0	MII_TXD[0] / RMII_TXD[0]
	MII_TXD[1] / RMII_TXD[1]
ETH_MII_TXD3~2	MII_TXD[2]
	MII_TXD[3]
ETH_MII_RMII_RXDV	MII_RX_DV / RMII_CRS_DV
ETH_MII_RXER	MII_RX_ER
ETH_MII_RMII_RXD1~0	MII_RXD[0] / RMII_RXD[0]
	MII_RXD[1] / RMII_RXD[1]
ETH_MII_RXD3~2	MII_RXD[2]
	MII_RXD[3]
ETH_MII_CRS	MII_CRS
ETH_MII_COL	MII_COL
ETH_SMI_MDC	SMI_MDC
ETH_SMI_MDIO	SMI_MDIO

表 39-6 ETHMAC 端口功能分配

39.4 功能说明

以太网 MAC 控制器遵循局域网（LAN）的 IEEE802.3 国际标准，将 CSMA/CD（带有冲突检测的载波侦听多路访问）方式作为基本通信方式，它包括一个带 MII 和 RMII 接口的 MAC802.3（介质访问控制）控制器和一个专用 DMA 控制器。

39.4.1 ETH_MAC 功能

MAC 子层执行以下与数据链路控制步骤相关的功能：

- 数据封装（发送和接收）
  - 组帧（帧边界定界、帧同步）
  - 寻址（处理源地址和目标地址）
  - 错误检测
- 介质访问管理
  - 介质分配（冲突避免）
  - 竞争解决（冲突处理）

MAC 子层主要有两个工作模式：

- 半双工模式：站点使用 CSMA/CD 算法争用物理介质
- 全双工模式：满足以下条件时，无需解决竞争问题（CSMA/CD 算法不是必需的）便可同时发送和接收数据：
  - 物理介质能够支持同步发送和接收
  - 正好有 2 个站点与 LAN 相连
  - 两个站均配置为全双工工作模式

39.4.1.1 MAC 802.3 帧格式

MAC 的数据通信可使用两种帧格式：

- 基本 MAC 帧格式
- 带 VLAN 标识的 MAC 帧格式（对基本 MAC 帧格式的扩展）

图 39-5 所示，是两种帧格式的具体结构。

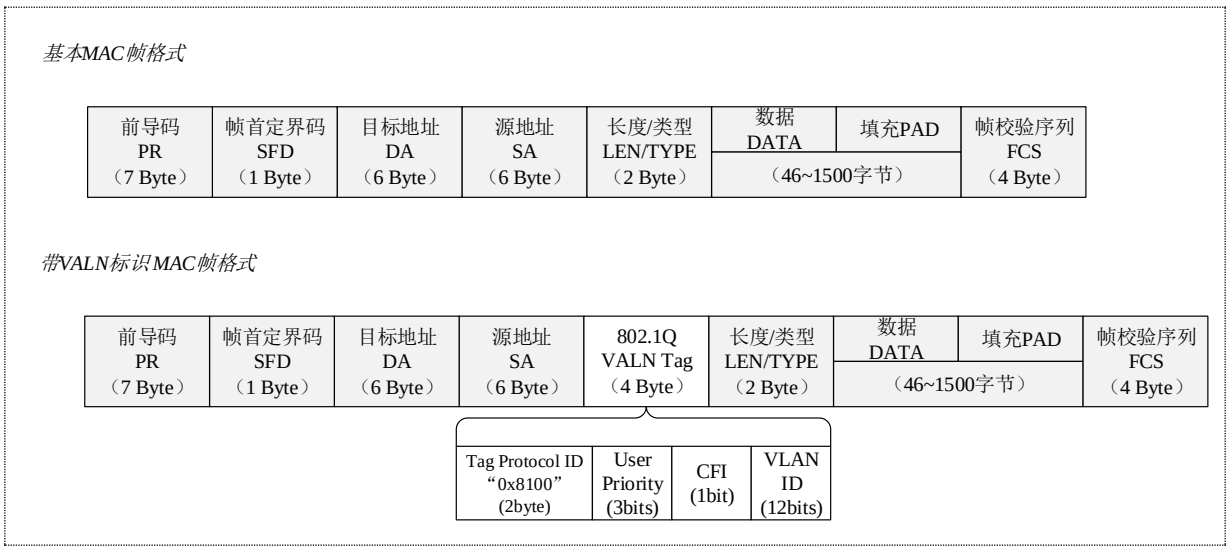


图 39-5 MAC 帧结构

除了帧校验序列 FCS，以太网 MAC 控制器发送每个字节时都按照低位先出的次序进行传输。

CRC 计算包括帧数据的所有字节除去前导码和帧首界定码域。以太网帧 32CRC 生成多项式为 0x04C11DB7，且此多项式用于以太网模块中所有的 32 位 CRC 计算，如下式所示：

$$G(x) = x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1$$

通过以下条件之一定义一个帧为无效 MAC 帧：

- 帧长度与长度/类型字段指定的预期值不一致。如果长度/类型字段包含类型值，则认为帧长度与此字段一致（没有无效帧）
- 帧长度不是字节的整数倍（额外位）
- 根据传入帧计算出的 CRC 值与包含的 FCS 不匹配

39.4.1.2 MAC 帧发送动作

TxDMA 控制发送路径的所有事务。从系统存储器读取的以太网帧由 TxDMA 送入 Tx FIFO，然后传输至 MAC 内核。帧传输结束时，从 MAC 内核获取发送状态并传回 TxDMA。

以太网 MAC 控制器搭载 2KB 的 Tx FIFO。Tx FIFO 填充级别将指示给 DMA，以便 TxDMA 可通过 AHB 接口在所需的系统存储器启动数据传送。来自 AHB Master 接口的数据将送入 Tx FIFO。

检测到 SOF 时，MAC 接收数据并开始向 MII 发送。在应用程序启动发送后，向 MII 发送帧数据所需的时间是可变的，具体取决于 IFG 延迟、发送报头/SFD 的时间以及半双工模式的任意回退延迟等延迟因素。EOF 传输到 MAC 内核后，内核将完成正常的发送，然后将发送的状态返回给 TxDMA。如果在发送过程中发生常规冲突（在半双工模式下），MAC 内核将使发送状态有效，然后接受并丢弃所有后续数据，直至收到下一个 SOF。检测到来自 MAC 的重试请求（在状态中）时，应从 SOF 重新发送同一帧。如果发送期间未连续提供数据，MAC 将发出下溢状态。在帧的正常传输期间，如果 MAC 在未获得前一帧的 EOF 的情况下接收到 SOF，则将忽略该 SOF 并将新的帧视为前一帧的延续。

向 MAC 内核输送数据有两种操作模式：

- 在阈值模式下，只要 FIFO 中的字节数超过配置的阈值（或在超过阈值前写入帧结束），数据就准备好并转发到 MAC 内核。该阈值可使用 ETH_DMA_BUSMODR 的 TTC 位配置。
- 在存储转发模式下，仅当在 FIFO 中存储完整的帧后，才会向 MAC 内核输送帧。如果 Tx FIFO 的大小小于要发送的以太网帧，则在 Tx FIFO 接近填满时向 MAC 内核输送帧。

应用可通过将 FTF 位（ETH_DMA_OPRMODR 寄存器[20]）置 1 来清空 Tx FIFO 的所有内容。此位自行清零，并将 FIFO 指针初始化为默认状态。如果向 MAC 内核传输帧时将该位置 1，则传输将停止，此时 Tx FIFO 被视为空，MAC 发送器将出现下溢事件并且相应的状态字将转发给 TxDMA。

发送基本协议

MAC 控制以太网帧的发送操作。它执行下列功能以满足 IEEE802.3/802.3z 规范。包括：

- 生成报头和 SFD
- 在半双工模式下生成 Jam 阻塞信号
- 控制 Jabber 超时
- 控制半双工模式下的流量（背压）
- 生成发送帧状态
- 包含符合 IEEE1588 的时间戳快照逻辑

当请求发送新的帧时，MAC 将发送报头和 SFD，紧接着发送数据。报头定义为“10101010”样式的 7 个字节，SFD 定义为“10101011”样式的 1 个字节。

冲突窗口定义为 1 个时隙（对于 10/100 Mbps 以太网，为 512 个位时间）。阻塞信号生成仅适用于半双工模式，不适用于全双工模式。在 MII 模式下，如果在开始传输帧到 CRC 字段结束之间的任何时间发生冲突，MAC 将在 MII 上发送 0x55555555 的 32 位 Jam 阻塞信号，通知所有其它站已发生冲突。如果在报头发送阶段发生冲突，MAC 将完成报头和 SFD 的发送，然后发送 Jam 阻塞信号。

MAC 内核使用一个 Jabber 定时器，用于在传输的字节超过 2048 字节时切断以太网帧的发送。在半双工模式下，MAC 使用延迟机制进行流量控制（背压）。当应用请求停止接收帧时，如果已使能发送流量控制，则只要 MAC 检测到接收帧，就会发送一个 32 字节的 Jam 信号。这会导致冲突并使远程站回退。应用程序通过将 ETH_MAC_FLOCTLR 寄存器中的 BPA 位置 1 来请求流量控制。如果应用请求发送帧，则即使激活背压功能，也将按调度计划发送。如果背压功能长时间保持激活（发生的连续冲突事件超过 16 个），则远程站将由于冲突过多而中止发送。如果针对发送帧使能 IEEE1588 时间戳功能，则将 SFD 置于发送 MII 总线时会获取系统时间的时间快照。

调度算法程序

MAC 负责调度 MII 上的帧发送。通过 IFG 设定或截断二进制指数回退算法来保持两个发送帧之间的帧间隔。在满足 IFG 和回退条件后，MAC 使能发送。

IFG 设定可确保两个发送帧之间的空闲期段，即配置的帧间隔（ETH_MAC_CONFIGR 寄存器中的 IFG 位）。如果要发送的帧在配置的 IFG 时间之前到达，则 MII 会等待来自 MAC 的使能信号，然后再开始发送。只要 MII 的载波信号进入无效状态，MAC 就会启动 IFG 计数器。在 IFG 计数值与设定值相等时，MAC 将以全双工模式使能发送。

在半双工模式下，当 IFG 配置为 96 个位时间时，MAC 将遵循 IEEE802.3 规范指定的顺从规则。如果在 IFG 间隔的前三分之二时间内（对于所有 IFG 值都为 64 位时间）检测到载波，MAC 将复位其 IFG 计数器。如 果在 IFG 间隔的后三分之一时间内检测到载波，MAC 将继续执行 IFG 计数并在 IFG 间隔结束后使能发送器。

在半双工模式下，实施截断二进制指数回退算法，具体由 MAC 配置寄存器

（ETH_MAC_CONFIGR）的 BL 位设定控制。

冲突期间重发

在半双工模式下，向 MAC 传输帧时，可能在 MAC 线接口上发生冲突事件。MAC 甚至会在接收到帧结束状态之前就指示重试，然后将使能重新发送并再次将帧从 Tx FIFO 中送出。

当超过 96 个字节送到 MAC 内核后，Tx FIFO 控制器将释放该空间，使 DMA 可送入更多数据。这意味着超过阈值后或 MAC 内核指示延迟冲突事件时，无法重新发送。

TxFIFO刷新

MAC 控制器可通过操作模式寄存器（ETH_DMA_OPRMODR）中的 FTF 位来清空 Tx FIFO。清空操作是立即操作，即使 Tx FIFO 正在向 MAC 内核传输帧，Tx FIFO 和相应的指针也会清零到初始状态。这将导致 MAC 发送器中生成下溢事件，并且帧发送将中止，此时帧的状态将同时标记下溢和清空事件（TDES0 的 FFF 位和 UDE 位）。清空操作期间，没有数据从 TxDMA 传输到 Tx FIFO，TxDMA 根据清空的帧数（包括局部帧），将相应数量的传输发送状态字传输到应用。完全清空的帧的帧清空状态位 (TDES0 的 FFF 位) 将置 1。当 TxDMA 接受所有已清空的帧的状态字后，清空操作完成，FTF 位随后将清零。此时，将接受来自 TxDMA 的新帧。清空操作完成后，将丢弃所有提交的数据，除非数据以 SOF 标记开头。

CRC处理和PAD生成

当从应用接收的字节数少于 60（DA+SA+LT+数据）时，可通过发送描述符 TDES0 的 DPAD 位复位，向发送帧附加零（PAD），使数据长度正好为 46 字节，以满足 IEEE802.3 的最小数据字段要求，也可选择不附加任何填充值。

MAC 还会计算帧检查序列（FCS）字段的循环冗余校验（CRC），根据发送描述符 TDES0 的 DCRC 位和 CRCR 位的设定可选择将校验值插入或替换到正在发送的数据中，具体请参考常规型 Tx 描述符和增强型 Tx 描述符章节。

注：CRC 替换功能使能时，必须使能增强型 Tx 描述符。

如果将 MAC 编程为不将 CRC 值附加到以太网帧的末尾，则不发送计算出的 CRC。但当 TDES0 的 DPAD 位为 0 时，小于 60 字节（DA+SA+LT+数据）的帧会在数据后附加填充 PAD，CRC 结果也将自动附加在填充帧的末尾。

源地址SA处理

从应用过来的数据如果不含源地址字段，MAC 可对该字段进行插入；如果含有源地址字段，MAC 可选择用 MAC 地址寄存器 0~1 中设定的值对该字段进行替换。具体的对 SA 字段处理方式可以通过 MAC 配置寄存器（ETH_MAC_CONFIGR）的 SAIRC 位或描述符 TDES1 的 SAIRC 位进行控制。

注：在通过描述符方式使能该功能时，必须使能增强型 Tx 描述符。

VLAN标识处理

MAC 还可以对从应用来的 VLAN 帧的标识符字段进行插入、替换、删除处理。具体的处理方式可以通过 VLAN 标记发送控制寄存器（ETH_MAC_VTACTLR）的 VLANC 位或描述符 TDES0 的 VLANC 位进行控制。

注：在通过描述符方式使能该功能时，必须使能增强型 Tx 描述符。

发送状态字

在向 MAC 内核传输以太网帧结束时，以及内核完成帧的发送后，发送状态会通过发送描述符提供给应用。发送状态的详细说明参见 TDES0 中的位[17:0]。如果使能 IEEE1588 时间戳功能，将返回特定帧的 64 位时间戳以及发送状态。

发送暂停帧

在全双工模式下，当发送流量控制使能位（ETH_MAC_FLOCTLR 中的 TFE 位）置 1 时，MAC 将生成暂停帧并根据需要发送暂停帧。暂停帧与计算出的 CRC 附加在一起一并发送。可以通过两种方式启动暂停帧的生成。

- 通过将 ETH_MAC_FLOCTLR 寄存器中的 FCA 位置 1 来请求流量控制，则 MAC 将生成并发送单个暂停帧。生成的帧中的暂停时间值为 ETH_MAC_FLOCTLR 中编程的暂停时间值。要在先前发送的暂停帧中指定的时间之前延长或结束暂停时间，用户必须在应用程序中以适当的值编程 ETH_MAC_FLOCTLR 寄存器中的 PAUSET 暂停时间值，然后请求另一次暂停帧发送。
- Rx FIFO 填满时并且应用程序已请求流量控制，则 MAC 将生成并发送暂停帧。生成的帧中的暂停时间值为 ETH_MAC_FLOCTLR 中编程的暂停时间值。如果在此暂停时间结束前，Rx FIFO 在可配置的时隙数（ETH_MAC_FLOCTLR 中的 PLT 位）期间保持填满状态，将发送第二个暂停帧。只要 Rx FIFO 保持填满状态，该

过程将一直重复下去。如果在暂停时间结束之前不再满足此条件，MAC 将发送一个暂停时间为零的暂停帧，向远程端口表明 Rx FIFO 已准备好接收新数据帧。

COE引擎

在网络互联过程中，基本的通信协议（例如 TCP 和 UDP）都实施校验和，并配备校验和字段，这有助于确定通过网络发送的数据的完整性。由于以太网最广泛的用途是通过 IP 数据报封装 TCP 和 UDP，因此以太网 MAC 控制器具有校验和减荷功能（COE 引擎），该功能支持发送路径中的校验和计算、插入以及接收路径中的校验和计算、检测。本节介绍 COE 引擎在发送中的操作，30.4.1.3.6 节介绍 COE 引擎在接收中的操作。

COE 引擎通过完整的帧来计算 TCP、UDP 或 ICMP 的校验和，然后将其插入报头的校验和字段。由于此要求，仅当 Tx FIFO 配置为存储转发模式（ETH_DMA_OPRMODR 寄存器中的 TSF 位置 1）时，才使能此功能。如果内核配置为阈值模式，则将绕过 COE 引擎。

必须确保 Tx FIFO 足够深，使其能存储一个完整的帧，以便将该帧传输到 MAC 内核发送器。如果 FIFO 的深度小于输入以太网帧的大小，则将绕过 TCP/UDP/ICMP 校验和插入功能，仅支持 IPv4 报头校验和插入功能，即使在存储转发模式下也是如此。

发送校验和减荷功能的控制可通过将 CIC 位（TDES0 中的[23:22]）设定来实现每个帧的校验和控制。

- IP 报头校验和

在 IPv4 数据报中，报头字段的完整性由 16 位头校验和字段（IPv4 数据报的第 11 字节和第 12 字节）指示。当以太网帧的类型字段值为 0x0800 且 IP 数据报的版本字段值为 0x4 时，COE 引擎将检测到 IPv4 数据报。计算期间，将忽略输入帧的校验和字段并将其替换为计算出的值。IPv6 报头没有校验和字段，因此，COE 不修改 IPv6 报头字段。

IPv4 报头校验和计算的结果由发送状态中的 IP 报头错误状态位（TDES0 的 IHE 位）指示。只要以太网类型字段的值和 IP 报头版本字段的值不一致，或当以太网帧没有足够的数据（如 IP 报头长度字段所指示）时，此状态位将置 1。

因此，总结下来，在以下情况下会发生 IP 报头错误：

a) 对于 IPv4 数据报：

  - 以太网类型为 0x0800，但 IP 报头版本字段不等于 0x4

- IPv4 报头长度字段指示小于 0x5（20 字节）的值
- 总的帧长度小于 IPv4 报头长度字段给定的值

b) 对于 IPv6 数据报：

- 以太网类型为 0x86DD，但 IP 报头版本字段不等于 0x6
- 帧在 IPv6 报头（40 字节）之前结束，或已完全接收到扩展报头（如扩展报头中相应的报头长度字段中给定）

如果以太网类型字段指示为 IPv4 有效负载，即使 COE 引擎测到 IP 报头错误，也会插入 IPv4 报头校验和。

- TCP/UDP/ICMP 校验和

TCP/UDP/ICMP 校验和功能会对 IPv4 或 IPv6 报头（包括扩展报头）进行处理，并确定封装的有效负载是 TCP、UDP 还是 ICMP。

在下面两种情况下，COE 引擎会无效：

a) 非 TCP、非 UDP 或非 ICMP/ICMPv6 的有效负载。

b) 分段的 IP 帧（IPv4 或 IPv6）、带安全功能（例如，验证报头或封装的安全有效负载）的 IP 帧和带路由报头的 IPv6 帧。

在 COE 引擎有效的情况下，它会计算 TCP、UDP 或 ICMP 有效负载的校验和，然后将其插入报头中相应的字段。它可工作在以下两种方式：

a) TCP、UDP 或 ICMPv6 伪报头并未包含在校验和计算中，并假定其存在于输入帧的校验和字段中。校验和字段包含在校验和计算中，然后替换为最终计算出的校验和。

b) TCP、UDP 或 ICMPv6 伪报头包含在校验和计算中，将忽略校验和字段，并使用最终计算出的值覆盖校验和字段。

注：对于 ICMP-over-IPv4 数据包，由于没有为其定义伪报头，因此在这两种模式下 ICMP 数据包中的校验和字段都必须始终为 0x0000。如果不等于 0x0000，可能向数据包插入不正确的校验和。

此操作的结果由发送描述符中的有效负载错误状态位（TDES0 的 TPCE 位）指示。当检测到下列情况之一时，TPCE 位置 1：

a) 在存储转发模式下，帧已转发到 MAC 发送器，但帧结束未写入 Rx FIFO

b) 在接收到 IP 报头中的有效负载长度字段指示的字节数前，数据包已结束

注：当数据包的长度大于指示的有效负载长度时，将字节作为填充字节忽略，并且不报告错误。检测到第一种类型的错误时，不修改 TCP、UDP 或 ICMP 报头。对于第二种错误类型，计算的校验和仍将插入相应的报头字段。

发送时序

在 RMII 接口时，来自 MII 的每个半字节都在 RMII 上发送，一次发送双位，双位的发送顺序如图 39-6 所示。首先发送位序较低的位（D0 和 D1），再发送位序较高的位（D2 和 D3）。

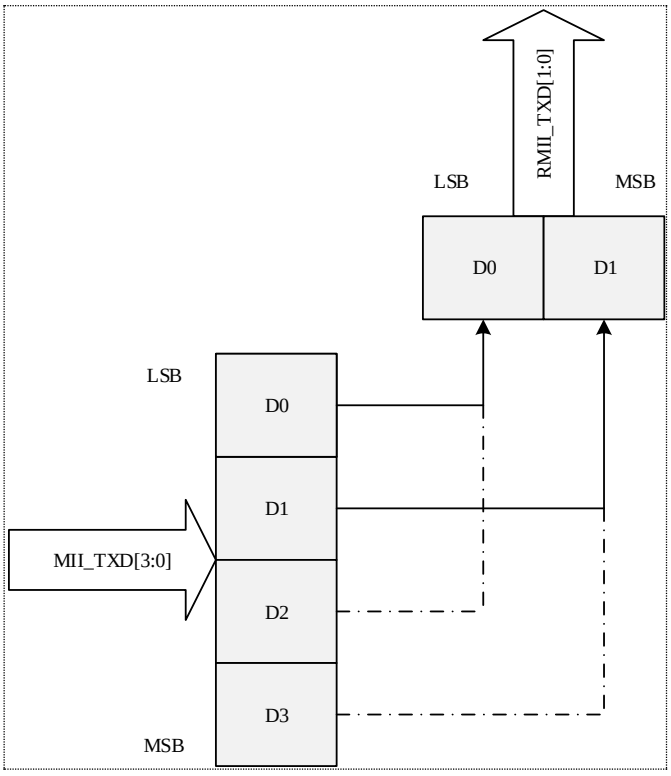


图 39-6 MII/RMII 发送位序

下图 39-7~图 39-9 为基本的发送时序图。

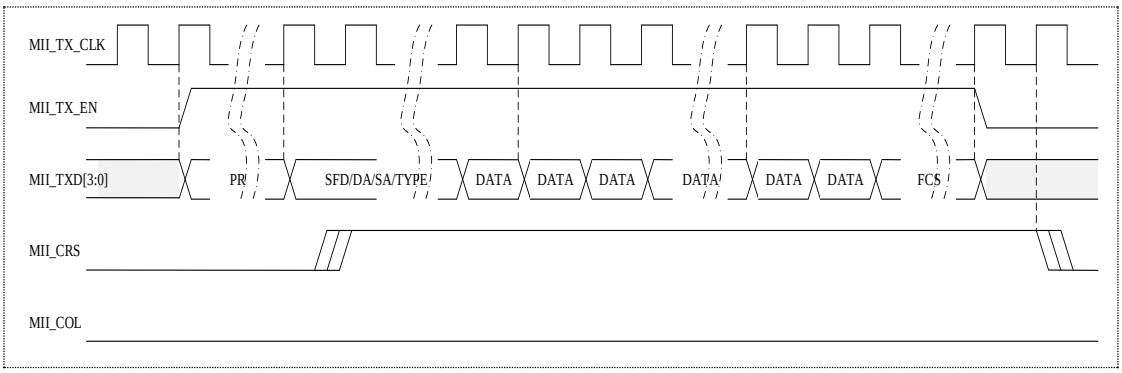


图 39-7 无冲突发送图

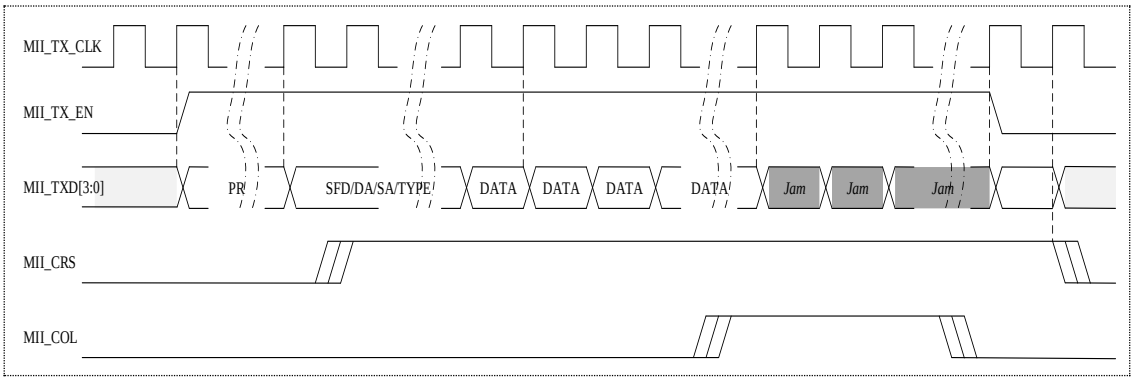


图 39-8 有冲突发送图

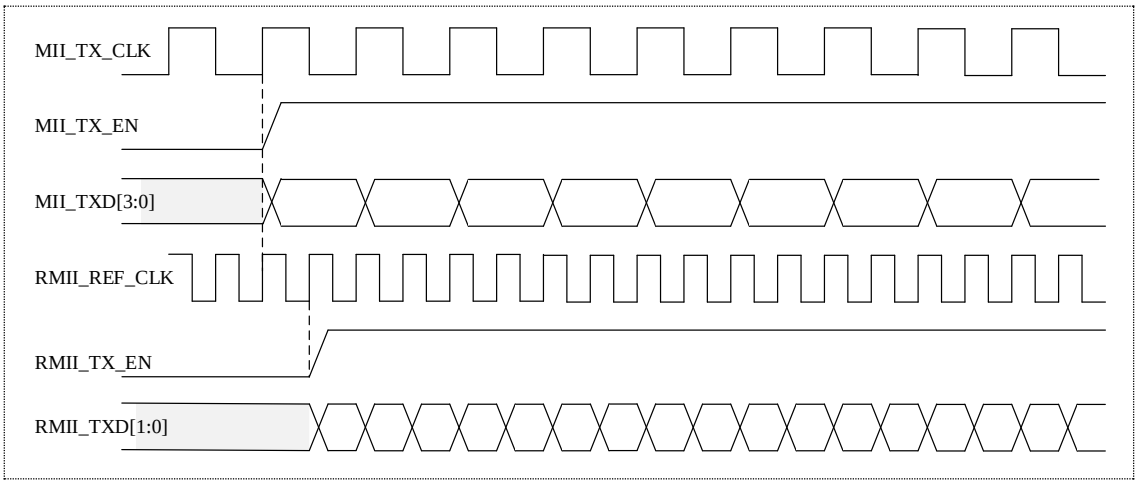


图 39-9 MII/RMII 模式下的发送图

39.4.1.3 MAC 帧接收动作

MAC 控制器接收的帧将送入 Rx FIFO，在接收达到一定数量条件时，就会指示 RxDMA 进行数据传送，RxDMA 可向 AHB 接口发起预配置的突发传输。

以太网 MAC 控制器搭载 2KB 的 Rx FIFO。RxDMA 将 Rx FIFO 中的数据输送到系统存储器有两种操作模式：

在阈值模式下，当 Rx FIFO 接收收到的字节数大于配置的接收阈值时（使用 ETH_DMA_OPRMODR 寄存器中的 RTC 位配置）或完整的数据包时，数据将发起输送，其可用性将通知给 RxDMA。RxDMA 向 AHB 接口发起传输后，数据传输将从 Rx FIFO 持续进行，直到传输完成整个数据包。完成 EOF 帧的传输后，接收状态字将返回并发送到 RxDMA 控制器。在该模式下，某些错误帧不会被丢弃，因为在帧结束时接收到错误状态，而此时已从 Rx FIFO 将数据传送进系统存储器。

在存储转发模式下，当帧完全写入 Rx FIFO 后才可读出帧。在此模式下，将丢弃所有错误帧（如果内核配置为执行此操作），这样只会读出有效帧并将其转发到应用。

接收数据后状态立即可用，因此只要 Rx FIFO 未满，就可以向 Rx FIFO 中存储帧。

当 MAC 在 MII 上检测到 SFD 时，将启动接收操作。MAC 内核将去除报头和 SFD，然后再继续处理帧。检查报头字段以进行过滤，FCS 字段用于验证帧的 CRC。如果帧未通过地址滤波器，则在内核中丢弃该帧。

接收基本协议

MAC 接收数据时，首先去除接收的帧的报头和 SFD。检测到 SFD 后，开始向 Rx FIFO 发送以太网帧数据，从 SFD 后面的第一个字节（目标地址）开始发送。如果使能 IEEE1588 时间戳功能，则在 MII 上检测到任何帧的 SFD 时，都将获取系统时间的快照。除非 MAC 滤出并丢弃帧，否则此时间戳将传递给应用。

如果接收的帧长度/类型字段小于 0x600 并且为 MAC 编程了自动去除 CRC/PAD 选项，则 MAC 将向 Rx FIFO 发送帧数据（数据量不超过长度/类型字段中指定的数量），然后开始丢弃字节（包括 FCS 字段）。如果长度/类型字段大于或等于 0x600，则不管编程的自动 CRC 去除选项的值如何，MAC 都会向 Rx FIFO 发送所有接收到的以太网帧数据。

默认情况下，使能 MAC 看门狗定时器，即超过 2048 个字节（DA+SA+LT+数据+PAD+FCS）的帧会被切断。可通过对 MAC 配置寄存器（ETH_MAC_CONFIGR）中的看门狗禁止 (MWD) 位编程来禁止此功能。但是，即使禁止看门狗定时器，仍将切断大于 16KB 的帧并给出看门狗超时状态。

接收错误处理

如果在从 MAC 接收 EOF 数据之前 Rx FIFO 已满，则将声明上溢并丢弃整个帧，同时帧丢失统计寄存器（ETH_DMA_RFRCNTR）中的上溢计数器将递增。由于上溢，状态将指示这是一个部分帧。如果使能相应功能（ETH_DMA_OPRMODR 中的 FEF 和 FUF 位），Rx FIFO 可过滤错误帧和过小帧。

在阈值模式下，从 Rx FIFO 读取帧 SOF 时，可获取当前帧的基本信息，通过与 Rx 描述符的基本设定比对，则可判定该帧是否为错误帧，并丢弃整个错误帧；在存储转发模式下，则可在数据传送之前过滤并丢弃所有错误帧。

CRC计算和PAD去除

MAC 内核将计算接收的帧（包括目标地址字段到 FCS 字段）的 32 位 CRC，并检查接

收帧中的任何 CRC 错误。不管是否自动去除 CRC/PAD（通过 ETH_MAC_CONFIGR 寄存器 ACS 位控制），MAC 都将接收整个帧来计算所接收帧的 CRC 校验，并将校验的结果通过接收描述符 RDES0 的 CRE 位反馈给应用。

接收状态字

以太网帧接收结束时，MAC 输出接收状态通过 RxDMA 送到接收描述符 RDES0 中以供应用参考查询。

接收暂停帧

在帧发送过程中，MAC 将检测接收暂停帧并暂停帧发送，暂停时间为接收的暂停帧内指定的延迟（仅限全双工模式）。可通过 ETH_MAC_FLOCTLR 中的 RFE 位使能或禁止暂停帧检测功能。使能接收流量控制后，将开始监视接收帧的目标地址是否与控制帧的多播地址（0x0180C2000001）匹配。如果检测到匹配（接收的帧的目标地址与保留的控制帧的目标地址匹配），MAC 将根据 ETH_MAC_FLTCTLR 中的 PCF 位来决定是否将接收的控制帧传输到应用。

MAC 还将对接收到的控制帧的类型、操作码和暂停定时器字段进行解码。如果状态的字节计数指示 64 个字节，并且不存在任何 CRC 错误，则 MAC 发送器将暂停任何数据帧的发送，暂停时间为解码的暂停时间值乘以时隙（对于 10/100 Mbps 模式，均为 64 字节时间）。同时，如果检测到另一个零暂停时间值的暂停帧，MAC 将复位暂停时间并管理新的暂停请求。如果接收到的控制帧与类型字段（0x8808）、操作码（0x00001）以及字节长度（64 字节）均不匹配，或存在 CRC 错误，则 MAC 不会生成暂停。

对于具有多播目标地址的暂停帧，MAC 将根据地址匹配来过滤帧；对于具有单播目标地址的暂停帧，MAC 将根据 DA 字段是否与 MAC 地址寄存器 0 的内容匹配以及 ETH_MAC_FLOCTLR 中的 UNP 位是否置 1（检测具有单播目标地址的暂停帧）来进行过滤。ETH_MAC_FLTCTLR 寄存器中的 PCF 位可对控制帧的过滤以及地址过滤进行控制。

COE引擎

以太网 MAC 控制器支持对接收的以太网帧中的 IPv4 和 IPv6 帧进行检测和处理，以确保数据完整性。通过将 ETH_MAC_CONFIGR 寄存器中的 IPCO 位置 1 来使能接收 COE 引擎。MAC 通过检查所接收的以太网帧的类型字段中是否存在值 0x0800 或

0x86DD，来识别 IPv4 或 IPv6 帧。此识别方法也适用于带 VLAN 标记的帧。

接收 COE 引擎可计算 IPv4 报头校验和，并检查其是否与接收的 IPv4 报头校验和匹配。如果指示的有效负载类型（以太网类型字段）与 IP 报头版本之间有任何不匹配，或接收的帧的字节数小于 IPv4 报头的长度字段中指示的数量（IPv4 或 IPv6 报头中的可用字节数少于 20），IP 报头错误位将置 1。

接收 COE 引擎还可以识别接收的 IP 数据报（IPv4 或 IPv6）中的 TCP、UDP 或 ICMP 有效负载，并正确计算此类有效负载的校验和字段，包括用于校验和计算的 TCP/UDP/ICMPv6 伪报头字节，并检查接收的校验和字段是否与计算的值匹配。此操作的结果由接收描述符 RDES4 中的 IP 有效负载错误位给出。如果 TCP、UDP 或 ICMP 有效负载的长度与 IP 报头中给出的预期有效负载长度不匹配，此状态位也将置 1。

COE 引擎将绕过分段 IP 数据报的有效负载、带安全功能的 IP 数据报、IPv6 路由报头以及除 TCP、UDP 或 ICMP 以外的有效负载。

接收时序

在 RMII 接口时，MAC 内部的每个半字节都从接收自 RMII 的双位发送到 MII，半字节发送顺序如图 39-10 所示。先接收位序较低的位（D0 和 D1），再接收位序较高的位（D2 和 D3）。

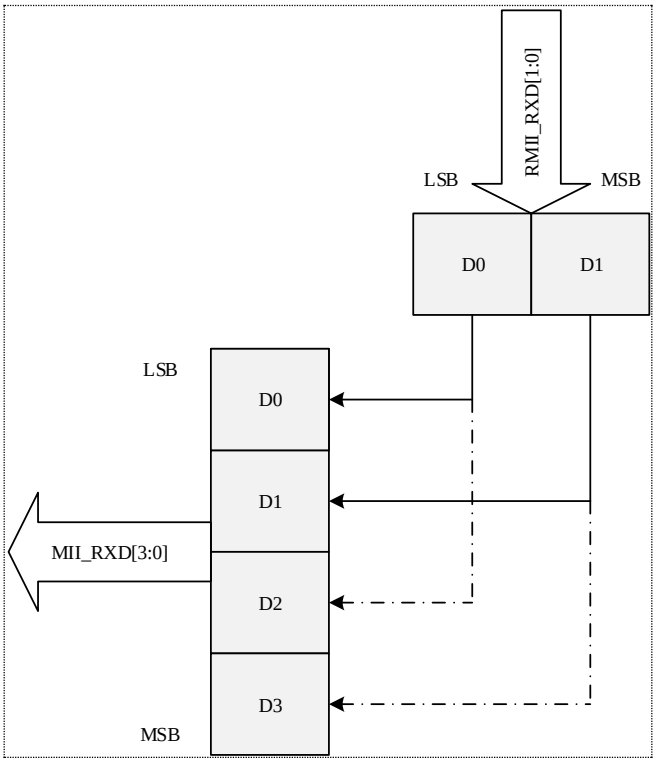


图 39-10 MII/RMII 接收位序

下图 39-11~图 39-13 为基本的发送时序图。

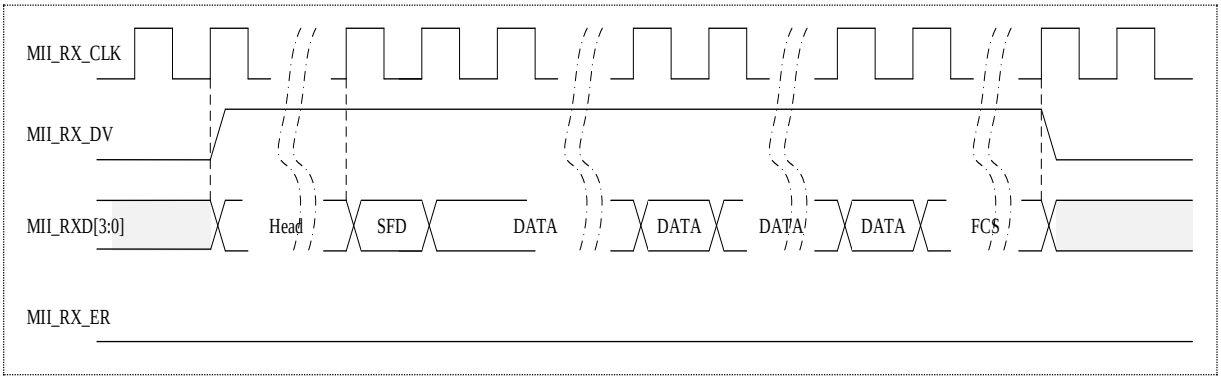


图 39-11 无错误发送图

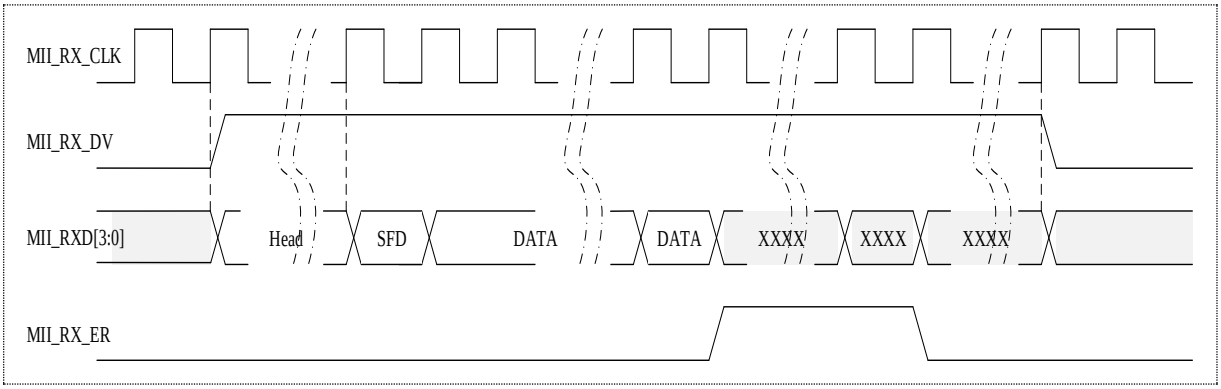


图 39-12 有错误发送图

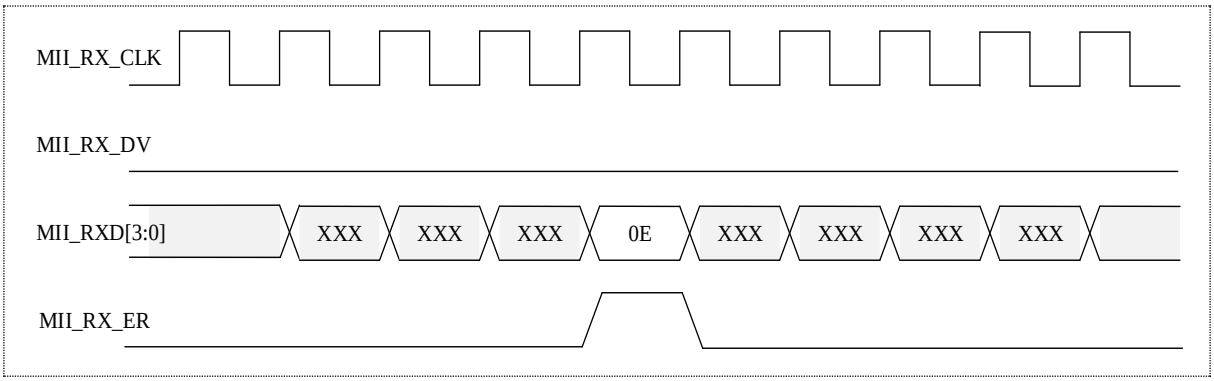


图 39-13 假载波指示下的接收图

39.4.1.4 MAC 帧过滤

MAC 可以对接收到的各种帧进行过滤以选择应用需要的帧，帧过滤方式包括以下内容：

1) MAC 帧的目标地址、源地址过滤
2) VLAN 帧的标识符过滤
3) LAY3 IP 报的源地址、目标地址过滤
4) LAY4 TCP、UDP 报的源端口、目标端口过滤

各帧过滤方式的基本说明分别在下面的章节介绍：

MAC源地址过滤

MAC 可根据接收的帧的源地址字段来执行完美过滤。默认情况下，MAC 将 SA 字段与 MAC 地址寄存器中编程的值进行比较。可通过将 MAC 地址寄存器中的位 30（SA）置 1，来将 MAC 地址寄存器 1~4 配置为 SA 字段进行比较。如果帧过滤控制寄存器（ETH_MAC_FLTCTLR）中的 SAF 位置 1，则 MAC 将丢弃未通过 SA 过滤的帧。否则，SA 过滤的结果将通过接收状态字中的状态位给出（参见接收描述符字 RDES0）。

注：SAF 位置 1 时，对 SA 过滤和 DA 过滤的结果进行与运算，以决定是否需要转发帧。这意味着任何一个过滤未通过都将丢弃帧。两个过滤必须都通过，才能将帧转发到应用。

ETHMAC 控制器将 MAC 地址寄存器 1~4 中地址值的各个字节与接收的相应 SA 字节进行比较时，可以将寄存器中相应的屏蔽字节控制位置 1 来屏蔽该字节，从而实现 SA 的组地址过滤。

对于源地址过滤，可在最终输出时选择反转过滤匹配结果，即在 SA 地址比较匹配时

认定为过滤失败。该功能由帧过滤控制寄存器（ETH_MAC_FLTCTLR）的 SAIF 位控制。

MAC目标地址过滤-单播

MAC 支持多达 5 个用于单播过滤的 MAC 目标地址。如果选择完美过滤（帧过滤控制寄存器中的 HUC 位设为 0），MAC 将接收的单播地址的所有 48 位目标地址与编程的 MAC 地址进行比较来确定是否匹配。默认情况下，始终使能 MAC 地址寄存器 0，其它 MAC 地址寄存器 1~4 则通过单独的使能位进行选择。

ETHMAC 控制器将 MAC 地址寄存器 1~4 的各个字节与接收的相应 DA 字节进行比较时，可以将寄存器中相应的屏蔽字节控制位置 1 来屏蔽该字节，从而实现 DA 的组地址过滤。

MAC目标地址过滤-多播

通过将帧过滤控制寄存器中的 PMF 位置 1，将 MAC 编程为通过所有多播帧。如果 PAM 位复位，MAC 将根据帧过滤寄存器中的 HMC 位执行对多播地址的过滤。在完美过滤模式下，将多播地址与编程的 MAC 地址寄存器 1~4 进行比较。该过滤方式也支持组地址过滤。

MAC目标地址过滤-广播

在默认模式下，MAC 不过滤任何广播帧。但是，如果将帧过滤寄存器中的 DBF 位置 1 来将 MAC 编程为拒绝所有广播帧，则会丢弃任何广播帧。

对于目标地址过滤（单播、多播、广播），也可在最终输出时选择反转过滤匹配结果，即在 DA 地址比较匹配时认定为过滤失败。该功能由帧过滤控制寄存器（ETH_MAC_FLTCTLR）的 DAIF 位控制，同样适用于 Hash 过滤结果。

MAC目标地址Hash过滤

对于单播目标地址和多播目标地址的过滤也支持 Hash 过滤方式。在 Hash 过滤模式（HUC 位/HMC 位置 1）下，MAC 将使用 64 位 Hash 表执行对单播地址或多博地址的不完美过滤，即 Hash 过滤。对于 Hash 过滤，MAC 将使用接收的目标地址的 6 个高 CRC 位来索引 Hash 表的内容。值为 000000 时，选取所选寄存器中的位 0；值为 111111 时，选取散列表寄存器中的位 63。如果相应位（由 6 位 CRC 指示）已置 1，将认为单播帧或多播帧已通过 Hash 过滤，否则认为帧未能通过 Hash 过滤。

注：CRC 是使用下列多项式编码的 32 位值：

$G(x) = x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1$

VLAN标识符过滤

当接收帧为 VLAN 帧时，MAC 可通过设定帧过滤控制寄存器（ETH_MAC_FLTCTLR）的 VTFE 位来使能对 VLAN 标识符字段的过滤。过滤用比较值设置在 VLAN 标记接收过滤寄存器（ETH_MAC_VTAFLTR）的 VLFLT 位，同时可通过设定该寄存器的 VTAL 位来选择对全 VLAN 标识符字段进行比较过滤还是仅比较低 12 位标识符。

对于 VLAN 标识符字段的过滤，也可在最终输出时选择反转过滤匹配结果，即在标识符字段比较匹配时认定为过滤失败。该功能由 VLAN 标记接收过滤寄存器（ETH_MAC_VTAFLTR）的 VTIM 位控制，同样适用于 Hash 过滤结果。

VLAN标识符Hash过滤

对于 VLAN 标识符的过滤也可以选择 Hash 过滤方式。在 Hash 过滤模式（ETH_MAC_VTAFLTR 的 VTHM 位置 1）下，MAC 将使用 16 位 Hash 表执行对 VLAN 标识符的不完美过滤，即 Hash 过滤。对于 Hash 过滤，MAC 将使用 VLAN 标识符的 CRC 位来索引 Hash 表的内容。值为 1 时，标明通过 VLAN 标识符过滤；值为 0 时，标明未通过 VLAN 标识符过滤。

注：CRC 是使用下列多项式编码的 32 位值：

$G(x) = x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1$

L3源地址&目标地址过滤

对接收到的 LAY3 层的 IPv4 报和 IPv6 报也支持源地址和目标地址的过滤。通过 LAY3LAY4 控制寄存器的 L3SAM 位和 L3DAM 位可分别使能源地址过滤与目标地址过滤。源地址或目标地址的比较值在 LAY3 地址寄存器（L3ADDRR0~3）中设定，具体请参考该寄存器说明。

LAY3 层 IP 地址的过滤可以选择高位屏蔽功能，以实现组地址的过滤。源地址和目标地址的高位屏蔽过滤控制分别由 LAY3LAY4 控制寄存器（ETH_MAC_L34CTLR）的 HSBM、HDBM 位设定控制。

对于 LAY3 层 IP 报的地址过滤，也可在最终输出时选择反转过滤匹配结果，即在地址字段比较匹配时认定为过滤失败。该功能由 LAY3LAY4 控制寄存器

（ETH_MAC_L34CTLR）的 L3SAIM、L3DAIM 位控制。

L4源端口&目标端口过滤

对接收到的 LAY4 层的 TCP 报和 UDP 报也支持源端口和目标端口的过滤。通过 LAY3LAY4 控制寄存器的 L4SPM 位和 L4DPM 位可分别使能源端口过滤与目标端口过滤。源端口或目标端口的比较值在 LAY4 端口寄存器（L4PORTR）中设定，具体请参考该寄存器说明。

对于 LAY4 层 TCP 报、UDP 报的端口过滤，也可在最终输出时选择反转过滤匹配结果，即在端口字段比较匹配时认定为过滤失败。该功能由 LAY3LAY4 控制寄存器（ETH_MAC_L34CTLR）的 L4SPIM、L3DPIM 位控制。

39.4.1.5 MAC Loopback 模式

MAC 支持对发送到其接收器的帧进行回送。默认情况下，禁止 MAC 回送功能，可通过编程 MAC 控制寄存器（ETH_MAC_CONFIGR）中的 LM 位使能该功能。

39.4.1.6 MAC 管理计数器（MMC）

MAC 管理计数器 (MMC) 搭载一组寄存器来收集有关已接收帧和已发送帧的统计信息。其中包括一个用于控制各寄存器行为的控制寄存器（ETH_MMC_MMCCTLR）、两个包含当前接收和发送统计状态的 32 位寄存器（ETH_MMC_REVSTSR 和 ETH_MMC_TRSSTSR）、两个包含接收和发送中断控制的 32 位寄存器（ETH_MMC_RITCTLR 和 ETH_MMC_TITCTLR），以及各种帧类型的统计寄存器。这些统计寄存器均可从应用中访问，寄存器章节中详细描述了这些统计寄存器的用途和意义，请参考 ETH_MMC 寄存器章节。

MAC 管理计数器（MMC）针对通过地址过滤的接收帧更新相应的统计寄存器，已丢弃帧的统计信息不会进行更新，除非丢弃的帧为小于 6 字节的矮帧（DA 字节不能完全接收）。

如果帧成功发送，则将发送的帧视为“好帧”。即，如果帧发送过程没有因为以下错误而中止，则认为发送的帧是“好帧”：

- Jabber 超时
- 无载波/载波丢失

- 延迟冲突
- 帧下溢
- 过度延迟
- 过度冲突

如果帧接收成功，则将接收的帧视为“好帧”。即，如果接收帧不存在以下错误，则认为接收帧是“好帧”：

- CRC 错误
- 矮帧（短于 64 字节）
- 对齐错误（仅限 10/100 Mb/s）
- 长度错误（仅限非类型帧）
- 超出范围（仅限非类型帧，超过最大大小）
- MII_RXER 输入错误

注：最大帧大小取决于帧类型，如下：

- *无标识符帧的最大字节数：1518*
- *VLAN 帧的最大字节数：1522*

39.4.1.7 MAC 电源管理（PMT）

ETHMAC 控制器支持电源管理（PMT）机制。可通过远程唤醒帧使能位以及魔术数据包使能位使能 PMT 模块。这些使能位（WKEN 和 MPEN）位于 ETH_MAC_PMTCTLR 寄存器中，可由应用编程。在使能 Power Down 模式（ETH_MAC_PMTCTLR.PWDN=1）时，MAC 将丢弃所有接收到的帧并且不会将这些帧转发给应用。仅当使能相应检测位且接收到对应的魔术数据包或远程唤醒帧时，MAC 控制器才会退出 Power Down 模式。

远程唤醒帧过滤

ETHMAC 控制器有 8 个唤醒帧过滤寄存器。要对每个寄存器执行写操作，需要逐个值加载远程唤醒帧过滤寄存器（ETH_MAC_RTWKFFR）。连续加载远程唤醒帧过滤寄存器八次，便可对远程唤醒帧过滤寄存器加载所需的值。读操作与写操作相同，要读取八个值，必须读唤醒帧过滤寄存器八次后，才能到达最后一个寄存器。每次读/写操作都会将唤醒帧过滤寄存器指针指向下一个过滤寄存器。通过读取 PMT 控制状态寄存器（ETH_MAC_PMTCTLR）的 RTWKPT 位可以知道当前的读写操作在哪个过滤寄存

器上，同时可通过对该寄存器的 RTWKFR 位置 1 复位内部的指针，复位之后可对该远程唤醒帧过滤寄存器重新进行操作。下图 39-14 所示是 ETHMAC 内部的 8 个远程唤醒帧过滤寄存器。

ETH_MAC_RTWKFFR 0	FLT0 Byte Mask							
ETH_MAC_RTWKFFR 1	FLT1 Byte Mask							
ETH_MAC_RTWKFFR 2	FLT2 Byte Mask							
ETH_MAC_RTWKFFR 3	FLT3 Byte Mask							
ETH_MAC_RTWKFFR 4	-	FLT3 Command	-	FLT2 Command	-	FLT1 Command	-	FLT0 Command
ETH_MAC_RTWKFFR 5	FLT3 Offset		FLT2 Offset		FLT1 Offset		FLT0 Offset	
ETH_MAC_RTWKFFR 6	FLT1 CRC16				FLT0 CRC16			
ETH_MAC_RTWKFFR 7	FLT3 CRC16				FLT2 CRC16			

图 39-14　远程唤醒帧过滤寄存器

上图中各部分的功能说明如下：

- 过滤器 i 字节掩码（Byte Mask）

  该寄存器定义过滤器 i 检测帧的哪些字节来确定帧是否为唤醒帧（i=0~3）。MSB（第 31 位）必须为零，位 j[30:0]为字节掩码。如果将字节掩码的位 j（字节数）置 1，则传入帧的过滤器 i 偏移+j 由 CRC 模块处理；否则将忽略过滤器 i 偏移+j。

- 过滤器 i 命令（Command）

  该 4 位命令控制过滤器 i 操作（i=0~3）。Bit3 为地址类型选择，置位时只检测多播帧，反之只检测单播帧；Bit2 和 Bit1 为保留位；Bit0 为过滤器 i 的使能位，置位时使能过滤器 i，反之禁止过滤器 i。

- 过滤器 i 偏移（Offset）

  该寄存器定义过滤器 i 要检测的帧的偏移（i=0~3）。该 8 位模式偏移是要检测的过滤器 i 第一个字节的偏移。允许的最小值为 12，表示帧的第 13 个字节（偏移值 0 表示帧的第一个字节）。

- 过滤器 i CRC-16（CRC16）

该寄存器包含预先写入的 CRC-16 值（i=0~3），用于与帧数据（过滤器 i Offset 且 Byte Mask 对应之后）计算的 CRC-16 值进行比较。

CRC-16 的生成多项式为：$G(x) = x^{16} + x^{15} + x^2 + 1$。

远程唤醒帧检测

当 MAC 处于 Power Down 模式并已将 PMT 控制状态寄存器（ETH_MAC_PMTCTLR）中的远程唤醒位 WKEN 置 1 时，MAC 可在接收到远程唤醒帧后恢复正常工作。PMT 支持四个可编程过滤器，这些过滤器可由应用配置为不同的接收帧模式。如果传入帧通过过滤器的地址过滤，并且过滤器 CRC-16 与传入的检测模式相匹配，则可接收唤醒帧。只需要检查唤醒帧是否存在长度错误、FCS 错误、帧尾错误、MII 错误、冲突，确保其不是矮帧，即使唤醒帧长度超过 512 字节，但只要帧的 CRC 值有效，帧便有效。

ETH_MAC_PMTCTLR 寄存器中的唤醒帧检测状态位（WKFR）会针对每个接收到的远程唤醒帧进行更新。此外，还会生成 PMT 中断（如果已使能）来指示已接收到远程唤醒帧。

魔术数据包检测

魔术数据包是基于 AMD 公司的魔术数据包技术对网络上处于睡眠模式下的器件上电。MAC 接收称为魔术数据包的特定信息包，此信息包的目标地址是网络上的节点。MAC 控制器只对目标地址为本器件或多播地址的魔术数据包进行检查，以确定这些数据包是否满足唤醒要求。对通过地址过滤（单播或多播）的魔术数据包进行检测，确定其是否符合远程唤醒数据格式，即 6 个字节的数据所有位为全“1”的数据包加上重复出现 16 次的 MAC 地址。

应用通过向 ETH_MAC_PMTCTLR 寄存器中的 MPEN 位写入 1 来使能魔术数据包唤醒。PMT 模块持续监测目标地址为对应于指定魔术数据包模式的节点的每个帧。检查每个接收帧的目标地址和源地址字段之后是否为 0xFFFFFFFFFFFF 形式的同步数据流。之后，PMT 模块检查帧是否存在重复 16 次且没有任何断开或中断的 MAC 地址。如果重复 16 次的地址中存在中断，则会再次扫描传入帧中是否存在 0xFFFFFFFFFFFF 形式。这 16 次重复可位于帧中的任何位置，但必须在同步数据流后面（0xFFFFFFFFFFFF）。只要检测到重复 16 次的 MAC 地址，器件就可以接收多播帧。

例如，如果节点 MAC 地址为 0x001122334455，则 MAC 扫描的数据序列为：

目标地址 源地址 ... FFFF FFFF FFFF

0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455

0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455

0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455

0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455

… CRC

时，认为接收到了一个魔术唤醒包。ETH_MAC_PMTCTLR 寄存器中的魔术数据包检测会针对接收到的魔术数据包进行更新。此外，还会生成 PMT 中断（如果已使能）来指示已接收到魔术数据包。

掉电期间系统状态

1) 在 Power Down 模式期间应保持 MAC 接收器使能状态，原因是此时要对魔术数据包/远程唤醒帧进行相关的接收检测，通过将 ETH_MAC_CONFIGR 寄存器中的 RE 位置 1 来使能 MAC 接收器。
2) 在 Power Down 模式期间应保持 MAC 发送器关闭状态。通过将 ETH_MAC_CONFIGR 寄存器中的 TE 位清零来关闭 MAC 发送器。
3) 在 Power Down 模式期间应禁止以太网 DMA，原因是此时无需将魔术数据包/网络唤醒帧复制到系统存储器。通过将 ETH_DMA_OPRMODR 寄存器中的 STT 位和 STR 位清零来关闭 TxDMA 和 RxDMA。

推荐的掉电和唤醒顺序如下：

1. 禁止 TxDMA，并等待所有之前的帧发送完成
2. 通过将 ETH_MAC_CONFIGR 寄存器中的 RE 位和 TE 位清零来禁止 MAC 发送器和 MAC 接收器
3. 等待 RxDMA 清空 Rx FIFO 中的所有帧
4. 禁止 RxDMA
5. 通过将 ETH_MAC_PMTCTLR 寄存器中的 MPEN/WKEN 位置 1 来使能魔术数据包/远程唤醒帧检测
6. 通过将 ETH_MAC_PMTCTLR 寄存器中的 PWDN 位置 1 来使能 MAC 掉电模式

7. 通过将 ETH_MAC_CONFIGR 寄存器中的 RE 位置 1 来使能 MAC 接收器
8. 等待接收魔术数据包和远程唤醒帧
9. 接收到有效唤醒帧时，以太网控制器退出掉电模式
10. 读取 ETH_MAC_PMTCTLR 以清零 PMT 事件标志，使能 MAC 发送器、TxDMA 和 RxDMA

39.4.2 ETH_PTP 功能

IEEE1588 标准定义了一种协议，此协议支持使用网络通信、局域计算和分布式对象等技术实现的测量和控制系统中的精密时钟同步。该协议适用于利用支持多播消息传送的局域网（包括但不限于以太网）进行通信的系统。该协议用于对非均匀系统进行同步，这类系统包含固有精度、分辨率和稳定性都不断变化的时钟。该协议支持亚秒范围的系统级同步精度，并且需要极少的网络和本地时钟计算资源。这种基于消息的协议称为精密时间协议（PTP）， 它通过 UDP/IP 传送。系统或网络归类为主节点和从节点，用于分配时序/时钟信息。该协议通过交换 PTP 报文来同步从节点与主节点，如图 39-15 所示。

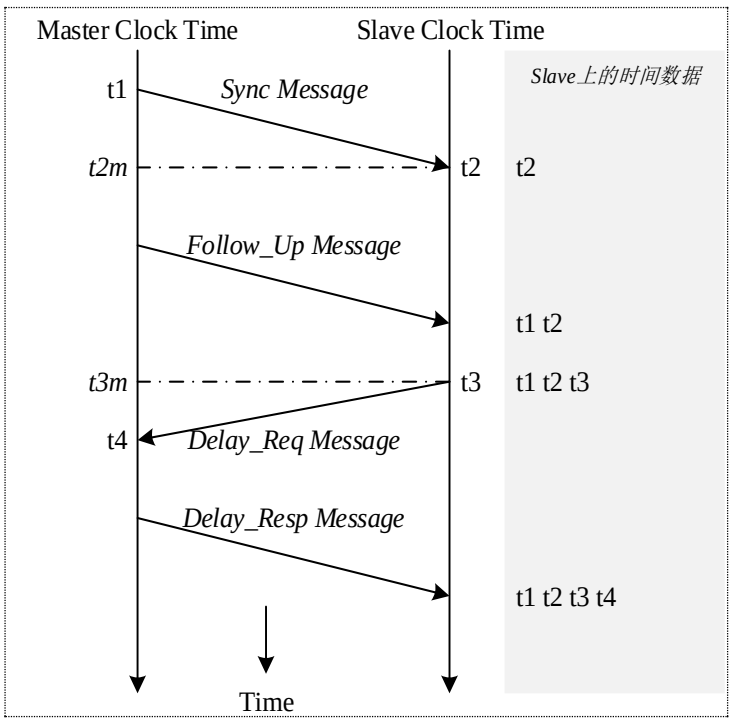


图 39-15 时钟同步图

1) 主节点向其所有节点广播 PTP 同步报文。同步报文包含主节点的参考时间信息。

报文离开主节点系统的时间为 t1。对于以太网端口，必须通过 MII 接口来捕获该时间

2) 从节点接收同步报文并利用其参考时序捕获准确时间 t2

3) 主节点随后会向从节点发送一个包含 t1 信息的 Follow_up 报文以备后用

4) 从节点向主节点发送一个 Delay_Req 报文，标出该帧离开 MII 的准确时间 t3

5) 主节点接收该报文，捕获报文进入其系统的准确时间 t4

6) 主节点将 Delay_Resp 报文中的 t4 信息发送给从节点

7) 从节点使用 t1、t2、t3 和 t4 这四个值来同步其本地参考时序与主节点的参考时序

虽然大多数协议都在 UDP 层之上通过软件实现，但需要硬件完成对通过 MII 接口来捕获特定 PTP 包进入或离开以太网端口时的准确时间。

39.4.2.1 PTP 参考时序源

根据 IEEE1588 规范的定义，要获取时间快照，内核需要一个 64 位格式的参考时间（分成两个 32 位通道，高 32 位表示时间的秒数，低 32 位表示时间的亚秒数）。

PTP 参考时钟输入用于在内部生成参考时间（亦称作系统时间）以及捕获时间戳。该参考时钟的频率必须大于等于时间戳计数器的分辨率。主节点与各从节点之间的同步精度目标约为 100 ns。精度取决于 PTP 参考时钟输入周期、振荡器的特性（漂移）以及同步过程的频率。

由于从 Tx 和 Rx 时钟输入域到 PTP 参考时钟域的同步，时间戳锁存值的不确定度为 1 个参考时钟周期。如果加上分辨率导致的不确定度，则会增加一半的时间戳周期。

39.4.2.2 PTP 报文类型

时间戳功能使能后，可通过时间戳控制寄存器（ETH_PTP_TSPCTLR）的 TSPMTSEL 位来控制对哪一个或哪一类报文进行时间戳快照。具体的对应关系如下表 39-7。

TSPMTSEL[3:0]	时间戳使能的报文类型
00X0	SYNC、Follow_Up、Delay_Req、Delay_Resp
0001	SYNC
0011	Delay_Req
01X0	SYNC、Follow_Up、Delay_Req、Delay_Resp Pdelay_Req、Pdelay_Resp、

	Pdelay_Resp_Follow_Up
0101	SYNC、Pdelay_Req、Pdelay_Resp
0111	Delay_Req、Pdelay_Req、Pdelay_Resp
10XX	SYNC、Delay_Req
11XX	Pdelay_Req、Pdelay_Resp

表 39-7 时间戳快照目标报文

39.4.2.3 PTP 帧发送功能

在 MII 上输出帧的 SFD 时，将捕获时间戳。对于需要捕获时间戳的帧，可对每个发送帧进行标记以指示是否有必要捕获该帧的时间戳。

无需处理发送帧即可识别 PTP 帧，可通过发送描述符中的控制位执行帧控制。

捕获到的时间戳返回给应用的方式与提供帧状态的方式相同。时间戳会随帧的发送状态一同发送回相应的发送描述符内，从而自动将时间戳与特定 PTP 帧相连。64 位时间戳信息会写回 TDES6 和 TDES7 字段，其中 TDES6 会保持时间戳的 32 个最低有效位。

39.4.2.4 PTP 帧接收功能

使能 IEEE1588 时间戳功能时，以太网 MAC 将捕获 MII 上接收到的所有帧的时间戳。MAC 会在完成帧接收过程时提供时间戳。

捕获到的时间戳返回给应用的方式与提供帧状态的方式相同。时间戳会随帧的接收状态一同发送回相应的接收描述符内。64 位时间戳信息会写回 RDES6 和 RDES7 字段，其中 RDES6 会保持时间戳的 32 个最低有效位。

39.4.2.5 PTP 系统时间校准

使用 PTP 输入参考时钟更新 64 位 PTP 时间。该 PTP 时间可用作时钟源，以获取 MII 上发送或接收的以太网帧的快照（时间戳）。可使用粗校准方式或细校准方式对系统时间定时器进行初始化或校准。

使用粗校准方式时，初始值或偏移值会写入时间戳更新寄存器。对于初始化，会将时间戳更新寄存器中的值写入系统时间计数器；对于系统时间校准，会将偏移值（时间戳更新寄存器）加到系统时间中或从系统时间中减去。

使用细校准方式时，从参考时钟频率相对于主时钟（如 IEEE1588 中定义）的偏移会在一段时间内进行校准，而不像粗校准方式中那样，在单个时钟周期内进行校准。校准

时间越长，越有助于保持线性时间，并且不会导致各 PTP 同步消息间隔之间的参考时间发生剧烈变化（或者大型抖动）。在此方法中，会使用一个累加器对基本加数寄存器（ETH_PTP_TSPADDR）中的内容求和，如图 39-16 所示。累加器生成的算数进位将用作使系统时间计数器递增的脉冲。累加器和基本加数寄存器均为 32 位寄存器。此处，累加器用作高精度频率乘法器或除法器。

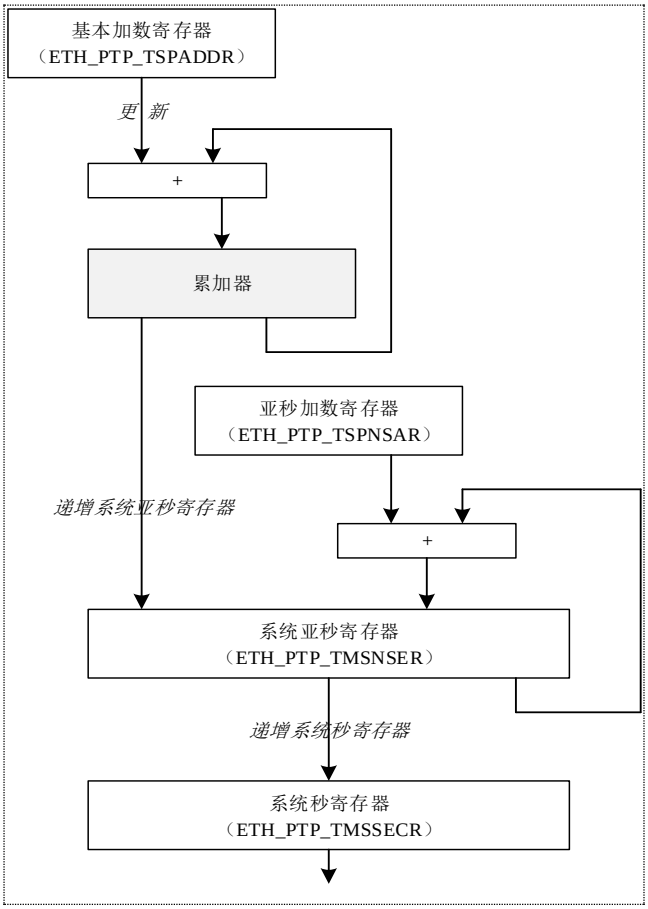


图 39-16 系统时间校准

系统时间更新逻辑需要使用 50MHz 的时钟频率，以达到 20ns 的精度。分频比是指参考时钟频率与所需时钟频率的比率。如果参考时钟频率为 66MHz，则通过计算可知分频比为 66MHz/50MHz=1.32，基本加数寄存器中设置的默认加数值为 2^32/1.32，相当于 0xC1F07C1F；如果参考时钟向下偏移降至 65MHz，则分频比为 65MHz/50MHz=1.3，基本加数寄存器中要设置的值为 2^32/1.3，相当于 0xC4EC4EC4；如果时钟向上偏移升至 67MHz，则必须将基本加数寄存器设置为 0xBF0B7672。当时钟偏移为零时，应编程的默认加法值为 0xC1F07C1F（2^32/1.32）。

用于亚秒加数寄存器递增的值设定为十进制数 43。这可使系统时间的精度达到 20ns（换句话说，增量步长时间为 20ns）。

软件必须根据同步报文对频率偏移进行计算，并相应更新基本加数寄存器。首先，使用基本加数寄存器中的 FreqCompensationValue0 设置从时钟。该值的计算公式如下：

FreqCompensationValue0 = 2^32 / FreqDivisionRatio

如果起初假定 MasterToSlaveDelay 对于连续的同步消息是相同的，则必须应用下述算法。数个同步周期之后，频率会锁定。从时钟随后可确定 MasterToSlaveDelay 的精确值，并使用新值重新与主时钟同步。

该算法如下：

- *在 MasterSyncTime（n）时刻，主时钟向从时钟发送同步报文。从时钟在其本地时钟为 SlaveClockTime（n）时接收到该报文，并用如下公式计算 MasterClockTime（n）：*

  MasterClockTime（n）= MasterSyncTime（n）+ MasterToSlaveDelay（n）

- *当前同步周期的主时钟计数 MasterClockCount（n）的计算公式如下：*

  MasterClockCount（n）= MasterClockTime（n）- MasterClockTime（n-1）

  （假定 MasterToSlaveDelay 对于同步周期 n 和 n-1 是相同的）

- *当前同步周期的从时钟计数 SlaveClockCount（n）的计算公式如下：*

  SlaveClockCount（n）= SlaveClockTime（n）- SlaveClockTime（n-1）

- *当前同步周期的主从时钟计数差值 SlaveClockCount（n）的计算公式如下：*

  ClockDiffCount（n）= MasterClockCount（n）- SlaveClockCount（n）

- *从时钟的分频系数 FreqScaleFactor（n）的计算公式如下：*

  FreqScaleFactor（n）=（MasterClockCount（n）+ ClockDiffCount（n））/ SlaveClockCount（n）

- *加数寄存器的频率补偿值 FreqCompensationValue（n）的计算公式如下：*

  FreqCompensationValue（n）= FreqScaleFactor（n）× FreqCompensationValue（n-1）

理论上，该算法可在一个同步周期内实现锁定，但是由于网络传播延迟和工作条件会不断变化，因此可能需要多个周期。

该算法可进行自校准：如果出于某些原因导致最初通过主时钟设置的从时钟不正确，则该算法会花费更多同步周期将其校准。

39.4.2.6 PTP 系统时间生成初始化

通过将时间戳控制寄存器（ETH_PTP_TSPCTLR）中的 TSPEN 位置 1 来使能时间戳功

能。但之后必须对时间戳计数器进行初始化才能启动时间戳操作。设定顺序如下：

1. 通过将 ETH_MAC_INTMSKR 寄存器中的 TSPIM 位置 1 以屏蔽时间戳触发中断
2. 编程 ETH_PTP_TSPCTLR 寄存器 TSPEN 位以使能时间戳
3. 根据 PTP 时钟频率编程亚秒加数寄存器（ETH_PTP_TAPNSAR）
4. 编程基本加数寄存器（ETH_PTP_TSPADDR）并将时间戳控制寄存器的 TSPADUP 位置 1（基本加数寄存器更新）
5. 轮询时间戳控制寄存器，确认 TSPADUP 位清零
6. 要选择精密校准方式时，将时间戳控制寄存器的 TSPUPSEL 位置 1
7. 用适当的时间值编程时间戳更新秒寄存器和时间戳更新亚秒寄存器
8. 将时间戳控制寄存器 TSPINI 位置 1（时间戳初始化）
9. 用时间戳更新寄存器中写入的值初始化时间戳计数器后，时间戳计数器便开始运行
10. 使能 MAC 接收器和发送器以使时间戳功能正常运行

39.4.2.7 PTP PPS 输出

PTP 模块可以将生成的内部时钟通过 PPS（PPS0）的方式输出到 ETH_PTP_PPS 端口上（连续输出模式），用于网络系统的所有节点的时钟同步，同时有两个 PPS（PPS0 和 PPS1）的输出（单次输出模式），可分别作为事件输出控制其它模块。

在 PPS 连续输出模式时，系统时间发生秒进位（亚秒时间溢出）后，PPS0 通道可生成一系列特定的脉冲或时钟，供外部网络节点时钟同步。具体输出脉冲或输出时钟频率的设定由 PPS 输出控制寄存器（ETH_PTP_PPSCTLR）的 PPSFRE0 位控制。只有 PPS0 通道支持连续输出模式。

在 PPS 单次输出模式时，系统时间若等于目标时间寄存器 0 或目标时间寄存器 1 设定的时间后，在 PPS0 或 PPS1 通道上可分别产生一个脉冲。只有当设定 PPS 输出控制寄存器（ETH_PTP_PPSCTLR）的 TT0SEL 位或 TT1SEL 位后，该功能才会被使能；当设定 PPS 输出控制寄存器（ETH_PTP_PPSCTLR）的 PPSFRE0 位或 PPSFRE1 位后，该功能的才会被激活。在脉冲产生之后，若想再次产生脉冲输出，需重新设定目标时间寄存器 0/1 和 PPS 输出控制寄存器的 PPSFRE0/1 位。

例如，用户根据需求，可将单次输出模式下的 PPS0 或 PPS1 的事件输出与 Timer2 联

动，具体的实现方式请参考 Timer2 章节和 INTC 章节。这种与 Timer2 的联动主要有以下两个作用：

1) 当系统时间等于目标时间时，MAC 提供的触发中断会引起已知延迟，从而导致命令执行时间出现不确定性。在中断触发至命令执行期间可通过 Timer2 的计时功能获得一个准确的中断处理延迟时间。
2) 系统时间的偏差校准。PPS 输出一个固定时间的脉冲，通过 Timer2 的脉宽测量功能对该固定时间脉冲进行检测，并根据 Timer2 的测量结果反馈调整 PPS 的系统时间，实现对 PTP 内部系统时间的校准。

39.4.3 ETH_DMA 功能

ETHMAC 控制器 DMA 具有独立的发送和接收引擎。发送引擎将数据从系统存储器传送到 Tx FIFO，而接收引擎将数据从 Rx FIFO 传送到系统存储器。DMA 可以在 CPU 完全不干预的情况下，通过描述符有效地将数据在 FIFO 和系统存储器之间传送。该 DMA 控制器可经过应用编程，在完成帧发送和接收操作时以及其它正常/错误条件下产生 CPU 中断。因此可以看出，ETHMAC 控制器 DMA 和 MCU 内核之间通过两种数据结构进行通信：

- 控制和状态寄存器
- 描述符列表和数据缓冲区

DMA 既可将 MAC 接收到的数据帧传送系统的数据缓冲区，也可以将系统数据缓冲区中的数据发送给 MAC，描述符中包含指向这些数据缓冲区的指针。DMA 共有两个描述符列表：一个用于接收的 Rx 描述符列表，一个用于发送的 Tx 描述符列表。Rx 描述符列表和 Tx 描述符列表的基地址分别通过描述符列表寄存器（ETH_DMA_RXDLADR 和 ETH_DMA_TXDLADR）设定。该描述符列表是一种前向链表（无论是隐式还是显式），最后一个描述符会指回第一个描述符以构成环形结构。描述符列表位于系统的物理存储空间。

每个描述符最多可指向两个缓冲区，这样便能使用两个物理寻址的缓冲区替代存储器中两个连续的缓冲区。通过配置 Rx 描述符和 Tx 描述符（RDES1[14]和 TDES0[20]）中链接的第二个地址来完成描述符的显式链接。

数据缓冲区同样位于系统的物理存储空间，通常由整个帧或部分帧组成，但不会超过单个帧。数据缓冲区中仅包含数据，数据通信的状态保存在描述符中。检测到帧结束时，DMA 会跳到下一个帧缓冲区，数据链接可实现跨越多个数据缓冲区存储帧，可以使能或禁止数据链接。描述符有两种结构方式：环形结构与链接结构，其实现方式如图 39-17 所示。

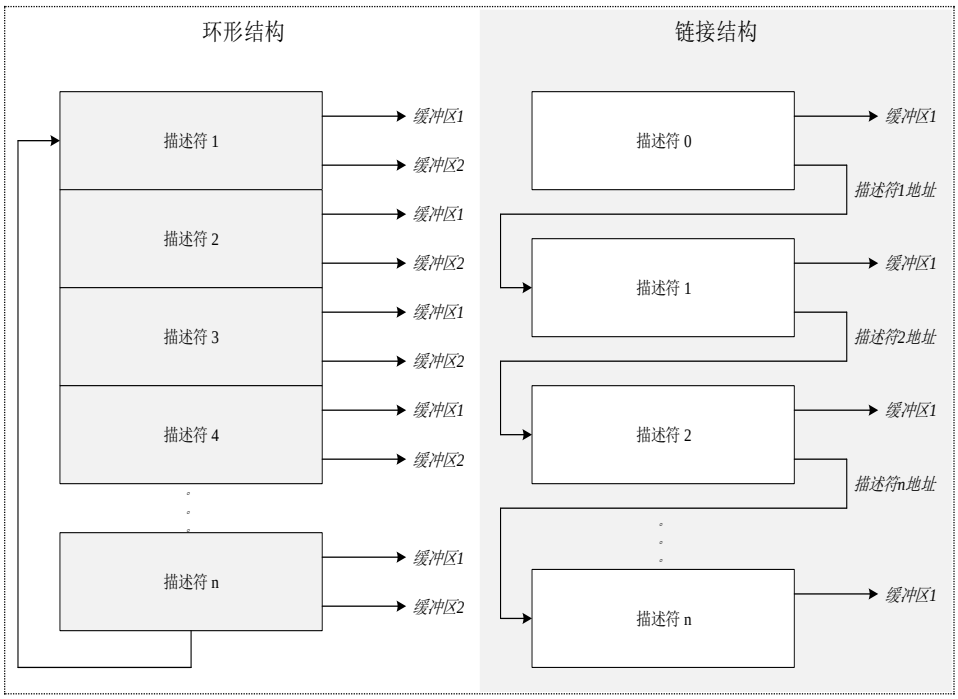


图 39-17 描述符结构

39.4.3.1 DMA 初始化

DMA 的初始化步骤如下：

1. 对 ETH_DMA_BUSMODR 执行写操作以设置总线访问参数
2. 对 ETH_DMA_INTENAR 寄存器执行写操作以屏蔽不必要的中断源
3. 创建描述符列表，对 ETH_DMA_RXDLADR 和 ETH_DMA_TXDLADR 寄存器执行写操作，为 DMA 提供各列表的起始地址
4. 对 MAC 寄存器执行写操作设定所需的过滤选项
5. 对 ETH_MAC_CONFIGR 寄存器执行写操作以配置和使能发送与接收工作模式
6. 对 ETH_DMA_OPRMODR 寄存器执行写操作，将 STT 位和 STR 位置 1 以启动发送和接收
7. 发送与接收引擎进入运行状态，并尝试从相应描述符列表中获取描述符，这两个引

擎随后开始处理接收和发送操作。发送和接收处理过程彼此独立，可单独进行启动或停止

总线突发访问

DMA 会尝试在 AHB Master 接口上执行固定长度的突发传送（当 ETH_DMA_BUSMODR 中的 FBST 位进行相应配置时）。突发的最大长度由 TPBL 或 RPBL 字段（ETH_DMA_BUSMODR [13:8]和[22:17]）指示和限制。对于要读取的 16 个字节，接收和发送描述符始终采用可能的最大突发大小进行访问。

仅当 Tx FIFO 中的空间足以容纳配置的突发或帧结束之前的字节数时（当帧短于配置的突发长度时），TxDMA 才会启动数据传送。TxDMA 会向 AHB Master 接口指示起始地址和所需的传送次数。当 AHB 接口配置为固定长度突发时，将使用 INCR4、INCR8、INCR16 和 SINGLE 的最佳搭配来传送数据。否则（非固定长度突发）会使用 INCR（未定义长度）与 SINGLE 传送数据。

仅当 Rx FIFO 中有足够的数据用于配置的突发时，或者在 Rx FIFO 中检测到帧结束时（当帧短于配置的突发长度时），RxDMA 才会启动数据传送。RxDMA 会向 AHB Master 接口指示起始地址和所需的传送次数。当 AHB 接口配置为固定长度突发时，将使用 INCR4、INCR8、INCR16 和 SINGLE 的最佳搭配来传送数据，如果在 AHB 接口上的固定突发结束前已到达帧结束，将执行空传送以完成固定长度的突发传送。否则（非固定长度突发）会使用 INCR（未定义长度）和 SINGLE 传送数据。

当 AHB 接口配置为地址对齐的节拍（ETH_DMA_BUSMODR.AAL=1）时，两个 DMA 引擎会确保 AHB 启动的第一次突发传送小于或者等于已配置 PBL 的大小。这样，后续的所有节拍都会从与已配置 PBL 对齐的地址开始。由于 AHB 接口多于 INCR16 的传送，DMA 只能对齐节拍最大为 16（PBL>16）的地址。

数据缓冲区对齐

发送和接收数据缓冲区在起始地址对齐方面没有任何限制。缓冲区的起始地址可与四个字节中的任意一个对齐。但是，DMA 始终在地址与总线宽度对齐时启动传输，并且在不需要的字节通道上传输空数据。这通常发生在以太网帧的开始或结束传送期间。

- 缓冲区读操作示例：

  如果发送缓冲区地址为 0x00000FF2，并且需要传送 15 个字节，则 TxDMA 将从地

址 0x00000FF0 读取 5 个全字，但在将数据传送到 Tx FIFO 时，丢弃或忽略额外的字节（前两个字节）。同样地，还将忽略最后一次传送的最后 3 个字节。TxDMA 始终确保向 Tx FIFO 传送的是全 32 位数据，除非是帧结束。

- 缓冲区写操作示例：

  如果接收缓冲区地址为 0x00000FF2，并且需要传送接收到的帧的 16 个字节，则 RxDMA 将从地址 0x00000FF0 开始写入 5 个全 32 位数据。但是，第一次传送的前 2 个字节与第三次传送的最后 2 个字节将包含空数据。

缓冲区大小计算

以太网控制器 DMA 不会更新发送和接收描述符中的缓冲区大小字段，只更新描述符的状态字段。应用必须用驱动程序计算缓冲区的大小。

TxDMA 会向 MAC 内核传送准确的字节数（由 TDES1 中的缓冲区大小字段指示）。如果将描述符标记为第一个描述符（将 TDES0 中的 TFS 位置 1），则 DMA 会将缓冲区的第一次传送标记为帧起始。如果将描述符标记为最后一个描述符（将 TDES0 中的 TLS 位置 1），则 DMA 会将数据缓冲区的最后一次传送标记为帧结束。

RxDMA 持续将数据传送至缓冲区，直到缓冲区已满或接收到帧结束为止。当描述符的 RFS 位置 1 时，如果未将描述符标记为最后一个描述符（RDES0 中的 RLS 位），则该描述符对应的缓冲区会填满，有效数据量由 RDES1 中的缓冲区大小指示；如果将描述符标记为最后一个描述符，则缓冲区不会填满，有效数据量将通过缓冲区大小字段减去数据缓冲指针偏移（RDES0 中的 FRAL 位）得到的结果表示，当数据缓冲区指针与数据总线宽度对齐时，偏移为零。

注：即使当接收缓冲区的起始地址与系统数据总线宽度未对齐时，系统也应分配一个大小与系统总线宽度对齐的接收缓冲区。例如，如果系统分配一个起始地址为 0x1000、大小为 1024 字节 (1 KB) 的接收缓冲区，则软件可将接收描述符中的缓冲区起始地址编程为具有 0x1002 偏移。RxDMA 将帧写入该缓冲区，其中前两个单元（0x1000 和 0x1001）中为空数据。实际帧从单元 0x1002 开始写入。因此，尽管已将缓冲区大小编程为 1024 字节，但由于存在起始偏移地址，因此该缓冲区的实际有用空间为 1022 字节。

DMA仲裁器

以太网控制器 DMA 内的仲裁器会在发送和接收通道对 AHB Master 接口进行的访问之间进行仲裁。可以使用两类仲裁：循环优先级和固定优先级。如果选择循环优先级仲裁（ETH_DMA_BUSMODR.DMAA=0），仲裁器会在发送和接收 DMA 同时请求访问时，按照 ETH_DMA_BUSMODR.TXPR 及 PTAT 位设置的优先方式和比率分配数据总线；如果选择固定优先仲裁（ETH_DMA_BUSMODR.DMAA=1），仲裁器会在发送和接收 DMA 同时请求访问时，按照 ETH_DMA_BUSMODR.TXPR 位设置的优先方式分配数据总线。

39.4.3.2 DMA 错误响应

对于由 DMA 通道发起的任何数据传送，如果从机给出错误响应，则相应 DMA 将停止所有操作并更新状态寄存器（ETH_DMA_DMASTSR）中的错误位和致命总线错误位。此时，该 DMA 控制器只能在软复位或硬复位外设以及重新初始化 DMA 之后才能恢复操作。

39.4.3.3 DMA 发送配置

TxDMA操作

在连续帧传送过程中，TxDMA 根据操作模式寄存器 ETH_DMA_OPRMODR 的 OSF 位的设定不同，对应的第二帧处理时序不同。当 OSF 位置 1 时，发送过程完成第一个帧的传送后，会立即轮询第二个帧的发送描述符列表，如果第二个帧有效，发送过程会在写入第一个帧的状态信息前发送第二帧。这种发送过程无需关闭第一个帧的状态描述符便可同时获取两个帧的方式称之为 OSF 模式，反之为非 OSF 模式。

TxDMA 引擎在 ETH_DMA_OPRMODR.OSF=0 下的操作顺序如下：

1. 在数据缓冲区设置好相应的以太网帧数据后，用户配置发送描述符 (TDES0-TDES3)并将 OWN 位（TDES0[31]）置 1
2. STT 位（ETH_DMA_OPRMODR 寄存器[13]）置 1 后，DMA 立即进入运行状态
3. 在运行状态下，DMA 为需要传送的帧轮询发送描述符列表。轮询启动后，DMA 会以连续的描述符环形顺序或链接顺序持续进行。如果 DMA 检测到标记为 CPU 所有的描述符（TDES0.OWN=0），或者发生错误条件，则会挂起传送并将发送缓冲区不可用位（ETH_DMA_DMASTSR 寄存器 TUS 位）与正常中断汇总使能位

（ETH_DMA_DMASTSR 寄存器 AIS 位）置 1。发送引擎继续执行步骤 9

4. 如果获得的描述符标记为 DMA 所有（TDES0.OWN=1），则 DMA 将根据取得的描述符对发送数据缓冲区地址进行解码
5. DMA 从系统存储器中获取发送数据并传送这些数据
6. 如果以太网帧保存在多个描述符的数据缓冲区中，则 DMA 将关闭中间描述符并获取下一个描述符。重复执行步骤 3、4 和 5，直到完成以太网帧数据的传送
7. 完成帧传送后，如果已为帧使能 IEEE1588 时间戳（按发送状态中的指示），则时间戳值将写入发送描述符（TDES6 和 TDES7），随后状态信息将写入该发送描述符的各状态位（TDES0）。此步骤中会清零 OWN 位，此时该描述符变为由 CPU 所有。
8. 帧发送完成后，在其最后一个描述符位（TDES0.TLS）置 1 时，DMA 动作状态寄存器的 TIS 位将置 1。随后 DMA 引擎返回步骤 3
9. 在挂起状态下，DMA 会在接收到发送轮询要求时尝试重新获取描述符（返回步骤 3），并将下溢中断状态位清零

非 OSF 模式下，TxDMA 的动作流程图如图 39-18 所示。

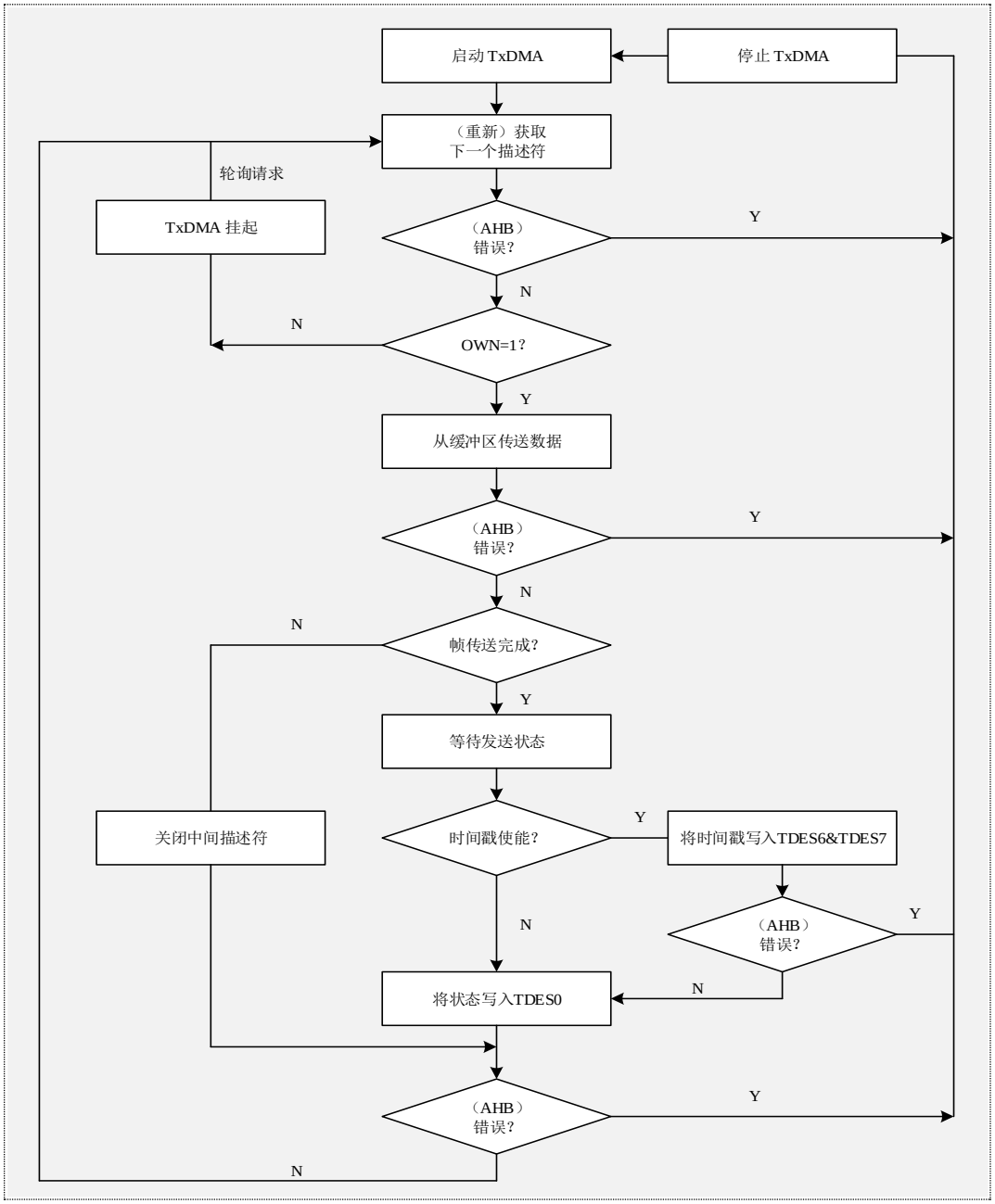


图 39-18 TxDMA 动作流程（非 OSF 模式）

TxDMA 引擎在 ETH_DMA_OPRMODR.OSF=1 下的操作顺序如下：

1. DMA 的操作过程如非 OSF 模式下 TxDMA 的步骤 1~6 中所述
2. 无需关闭前一帧的最后一个描述符，DMA 便可获取下一个描述符
3. 如果 DMA 拥有所需描述符，便会对该描述符中的发送缓冲区地址进行解码。如果 DMA 未拥有描述符，则会进入挂起模式并跳到步骤 7
4. DMA 从系统存储器中取得发送帧并发送该帧，直到帧数据发送结束，如果该帧拆分到多个描述符中，则会同时关闭中间描述符
5. DMA 等待前一个帧的发送状态和时间戳。当状态可用时，如果捕获到时间戳（由

状态位指示），DMA 会将这些时间戳写入 TDES6 和 TDES7。之后，OWN 位清零，DMA 将状态写入相应 TDES0，进而关闭描述符。如果没有为前一个帧使能时间戳，DMA 不会更改 TDES6 和 TDES7 的内容

6. 如果已使能时间戳，发送中断将置 1，DMA 获取下一个描述符，然后继续执行步骤 3（状态正常时）。如果上一个发送状态显示下溢错误，则 DMA 会进入挂起模式（步骤 7）
7. 在挂起模式下，如果 DMA 接收到挂起状态和时间戳，则会将时间戳（如果已为当前帧使能时间戳）写入 TDES6 和 TDES7，随后将状态写入相应的 TDES0。之后，将相关中断置 1 并返回挂起模式
8. 只有在接收到发送轮询要求（ETH_DMA_TXPOLLR 寄存器）后，DMA 才会退出挂起模式并 进入运行状态（转到步骤 1 或步骤 2，具体取决于挂起状态）

OSF 模式下，TxDMA 的动作流程图如图 39-19 所示。

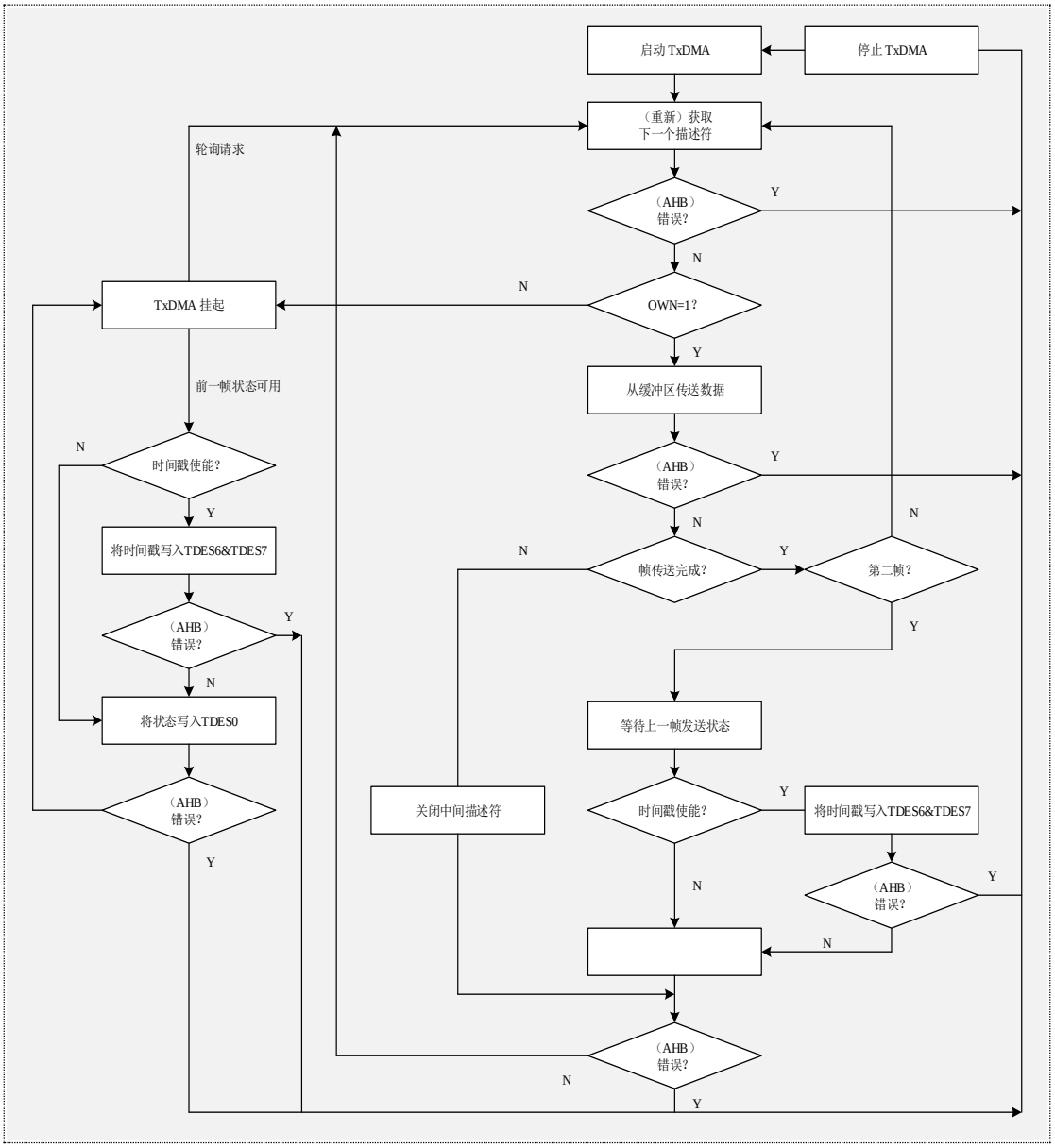


图 39-19　TxDMA 动作流程（OSF 模式）

发送帧处理

系统数据缓冲区须包含完整的以太网帧，不包括报头、PAD 和 FCS 字段， DA 字段、SA 字段和类型/长度字段须包含在有效数据中。如果发送描述符指示 MAC 内核必须禁止插入 CRC 或填充字节，则缓冲区必须具有包括 CRC 字节的完整以太网帧（不包括报头）。帧可以采用数据链接形式，也可以跨越多个缓冲区。帧必须由第一个描述符位（TDES0.TFS）和最后一个描述符位（TDES0.TLS）定界。传送启动时，必须将第一个描述符的 TDES0.TFS 置 1，之后帧数据便从存储器缓冲区传送到 Tx FIFO。同时，如果当前帧的 TDES0.TLS 为 0，发送过程将尝试获取下一个描述符。TDES0.TLS 为 0，则指示中间缓冲区，TDES0.TLS 为 1，则指示帧的最后一个缓冲区。当帧的最后一

个缓冲区完成传送后，DMA 会将最终状态信息写回描述符的（TDES0）相关状态位。此时，如果 TDES0.IOC 位置 1，则发送状态寄存器的 TIS 位将置 1，并获取下一个描述符，然后重复执行该过程。

根据 ETH_DMA_OPRMODR 寄存器 TSF 位的设定，实际的帧传送过程在 Tx FIFO 达到可编程的发送阈值时（ETH_DMA_OPRMODR 寄存器 TTC 位），或者 Tx FIFO 中包含完整的帧时启动传送过程。DMA 完成帧的传送后会释放描述符（清零 TDES0.OWN 位）。

发送轮询挂起

可通过以下任意条件暂停发送轮询：

- DMA 检测到 CPU 所有的描述符（TDES0[31]=0），并且 ETH_DMA_DMASTSR 寄存器的发送缓冲区不可用标志 TUS 置 1。如要恢复，驱动程序必须将描述符的所有权交给 DMA，然后发出轮询查询要求命令
- 检测到由下溢导致的传送错误时，将中止帧的传送。相应的发送描述符（TDES0）位将置 1。

如果 DMA 由于第一个条件而进入挂起状态，则发送状态寄存器的正常中断汇总位与发送缓冲区不可用位（ETH_DMA_DMASTSR 寄存器位 NIS 和 TUS 位）均置 1；如果发生第二个条件，异常中断汇总位与发送下溢位（ETH_DMA_DMASTSR 寄存器位 AIS 位和 UNS 位）都将置 1，并且信息将写入发送描述符，从而导致挂起。两种情况下，发送列表中的位置都会保留。保留的位置是 DMA 关闭的最后一个描述符后面的描述符位置。纠正挂起原因后，驱动程序必须明确发出发送轮询要求命令。

常规型Tx描述符

常规型 Tx 描述符结构由四个 32 位字组成，如表 30.8 所示，分别定义为 TDES0、TDES1、TDES2 和 TDES3。

TDES0	OWN	控制位 [30:26]	TTSE	控制位 [24:18]	TTSS	状态位 [16:0]
TDES1	控制位 [31:29]	字节计数缓冲区2 [28:16]		Reserved		字节计数缓冲区1 [12:0]
TDES2	缓冲地址1					
TDES3	缓冲地址2					

表 39-8 常规型 Tx 描述符

下面分别对 TDES0~TDES3 的具体位做说明。

1） TDES0 的各位功能如下：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
OWN	IOC	TLS	TFS	DCRC	DPAD	TTSE	CRCR	CIC[1:0]		TER	TSAC	VLANC[1:0]		TTSS	IHE

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ETSUM	JTE	FFF	TPCE	LOCE	NCE	TLCE	ECE	VLF	COC[3:0]				EDE	UDE	DEE

位	标记	位名	功能
b31	OWN	所有关系位	0：该描述符为CPU所有 1：该描述符为DMA所有 DMA在完成帧发送时或在描述符分配的缓冲区全部读取完成后将该位清零 属于同一个帧的所有后续描述符置1后，也应该将帧的第一个描述符的OWN位置1
b30	IOC	完成时中断	该位置1时，当前帧发送完毕后，动作状态寄存器的TIS位置1 *注：该位仅在最后一个描述符位（TDES0.TLS）置1的情况下有效*
b29	TLS	最后一个描述符	该位指示此描述符指向的缓冲区为帧的最后一个缓冲区
b28	TFS	第一个描述符	该位指示此描述符包含帧的第一个缓冲区。如果第一个缓冲区的大小为0，则第二个缓冲区将包含帧的帧头；如果第二个缓冲区的大小为0，则下一个描述符将包含帧的帧头
b27	DCRC	无效CRC	该位置1时，MAC不会将循环冗余校验（CRC）附加到所发送帧的末尾 *注：该位仅在第一个描述符位（TDES0.TFS）置1的情况下有效*
b26	DPAD	无效PAD	0：DMA会自动为不足64字节的帧添加补位项和CRC，不管该描述符的DCRC位的设定 1：MAC不会自动为不足64字节的帧添加补位项 *注：该位仅在第一个描述符位（TDES0.TFS）置1的情况下有效*
b25	TTSE	时间戳使能	该位置1时，将针对描述符所描述的发送帧激活IEEE1588硬件时间戳功能 *注：该位仅在第一个描述符位（TDES0.TFS）置1的情况下有效*
b24	CRCR	CRC替换控制	该位置1时，MAC将计算好的CRC值替换即将发送的发送帧的FCS字段 *注：只有当该描述符的TDES0.DCRC位置1时，该位有效*

b23~b22	CIC	Checksum插入控制	这些位控制校验和的计算与插入，如下所示： 00：禁止插入校验和 01：仅使能IP报头校验和的计算与插入 10：使能IP报头校验和以及TCP/UCP/ICMP校验和的计算与插入，但不会在硬件中计算伪报头校验和 11：使能IP报头校验和以及TCP/UDP/ICMP校验和的计算与插入，并在硬件中计算伪报头校验和 *注：该位仅在最后一个描述符位（TDES0.TLS）置1的情况下有效*
b21	TER	环发送结束	该位指示描述符列表已到达其最后一个描述符，DMA会返回描述符列表的首地址，形成一个描述符环
b20	TSAC	第二个地址链接	该位置1时，描述符中的第二个地址是下一个描述符地址，而非第二个缓冲区地址，此时该描述符的TBS2位为无效 *注：TER位优先级高于TSAC位*
b19~b18	VLANC	VLAN插入控制	00：对VLAN帧不做处理 01：在发送前将VLAN帧的标记和类型字段删除 10：将VLAN标记发送控制寄存器（ETH_MAC_VTACTLR）中的VLAN标记值插入到发送帧中 11：用VLAN标记发送控制寄存器（ETH_MAC_VTACTLR）中的VLAN标记值替换原发送帧中的VLAN标识符
b17	TTSS	发送时间戳状态	该位指示已为当前发送帧捕获到时间戳，置1时TDES6和TDES7将保存发送帧捕获到的时间戳值 *注：该位仅在最后一个描述符位（TDES0.TLS）置1的情况下有效*
b16	IHE	IP Header错误	该位置1时指示MAC发送器在IP数据报头中检测到错误 *注：当COE引擎检测出一个IP Header错误时，它仍然会将计算好的Checksum值插入到IPv4 Header Checksum字段*
b15	ETSUM	发送错误汇总	该描述符的B[16]、B[14]、B[13]、B[12]、B[11]、B[10]、B[9]、B[8]、B[2]、B[1]位中有一个被置位，该位也随之被置位
b14	JTE	Jabber超时错误	该位置1时，指示MAC发送器经历了Jabber超时 *注：只有在MAC配置寄存器（ETH_MAC_CONFIGR）的 MJD位未置1时，该位才会置1*
b13	FFF	帧刷新	该位指示DMA已依照CPU发出的软件刷新命令刷新帧
b12	TPCE	有效负载错误	该位指示COE引擎在TCP、UDP或ICMP报有效负载中检测到错误，且不会更新原帧中的Checksum字段
b11	LOCE	载波丢失错误	该位指示帧发送期间丢失载波，即，帧发送期间有一个或多个发送时钟周期的MII_CRS信号无效 *注：该位仅对在MAC处于半双工模式时实现无冲突发送的帧有效*
b10	NCE	无载波错误	该位指示在发送期间未由PHY触发载波侦听信号
b9	TLCE	延迟冲突错误	该位指示帧发送过程因冲突窗口（MII模式下为64个字节时间，含报头）后出现冲突而中止 *注：如果该描述符的UDE错误位置1，则该位无效*
b8	ECE	过度冲突错误	该位指示尝试发送当前帧时因出现16个连续冲突而中止发送 *注：如果MAC配置寄存器（ETH_MAC_CONFIGR）中的DRTY（禁*

			止重试）位置1，则此位会在出现首个冲突后置1并且帧发送过程将中止
b7	VLF	VLAN帧	该位指示所发送的帧为VLAN帧
b6~b3	COC	冲突计数	该位指示发送帧之前出现的冲突个数 *注：过度冲突位（TDES0.ECE）置1时，该计数无效*
b2	EDE	过度延迟错误	该位指示因出现超过24288个位时间的过度延迟而中止发送 *注：该位在MAC配置寄存器（ETH_MAC_CONFIGR）中的延迟检查（DC）位置1时有效*
b1	UDE	下溢错误	该位指示MAC因发送缓冲区的数据未及时到达而中止了帧发送下溢错误表示DMA在帧发送期间遇到发送缓冲区为空的情况
b0	DEE	延迟错误	该位指示MAC在发送前因存在载波而延迟 *注：该位仅在半双工模式下有效*

2） TDES1 的各位功能如下：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SAIRC[2:0]			TBS2												

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved			TBS1												

位	标记	位名	功能
b31~b29	SAIRC	SA插入替换控制	对发送帧的SA字段进行如下处理： 001：将MAC地址寄存器0中的地址值作为SA地址插入到发送帧中 101：将MAC地址寄存器1中的地址值作为SA地址插入到发送帧中 010：将MAC地址寄存器0中的地址值作为SA地址替换发送帧中的SA字段 110：将MAC地址寄存器1中的地址值作为SA地址替换发送帧中的SA字段 其它值：无操作
b28~b16	TBS2	缓冲区2大小	该位以字节为单位指示第二个数据缓冲区的大小 *注：该位在TDES0.TSAC位置1时无效*
b15~b13	Reserved	-	-
b12~b0	TBS1	缓冲区1大小	该位以字节为单位指示第一个数据缓冲区的大小。如果该字段为0，DMA将忽略该缓冲区并使用缓冲区2或下一个描述符，具体取决于该描述符的TSAC的设定

3） TDES2 的各位功能如下：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TBAP1[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TBAP1[15:0]															

位	标记	位名	功能
b31~b0	TBAP1	发送缓冲区1地址	该位向DMA指示数据在存储器中的位置，当软件为DMA提供此描述符（TDES0中的OWN位置1）时，这些位将指示缓冲区1的物理地址

4） TDES3 的各位功能如下：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TBAP2[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TBAP2[15:0]															

位	标记	位名	功能
b31~b0	TBAP2	发送缓冲区2地址/下一个描述符地址	该位向DMA指示数据在存储器中的位置，当软件为DMA提供此描述符（TDES0中的OWN位置1）并且使用描述符环结构时，这些位将指示缓冲区2的物理地址；如果TDES0.TSAC位置1，则该地址包含下一个描述符所在物理寄存器的指针 *注：只有在TDES0.TSAC位置1时，缓冲区地址指针才必须与总线宽度相符*

增强型Tx描述符

当时间戳功能有效（ETH_PTP_TSPCTLR.TSPEN=1）时或者 Checksum Offload 功能有效（ETH_MAC_CONFIGR.IPCO=1）时，应使用增强型 Tx 描述符。

增强型 Tx 描述符结构由 8 个 32 位字组成，如表 30.9 所示，分别定义为 TDES0、TDES1、TDES2、TDES3、TDES4、TDES5、TDES6 和 TDES7。其中 TDES0~TDES3 的功能与常规型相同。

TDES0	OWN	控制位 [30:26]	TTSE	控制位 [24:18]	TTSS	状态位 [16:0]
TDES1	控制位 [31:29]	字节计数缓冲区2 [28:16]			Reserved	字节计数缓冲区1 [12:0]
TDES2	缓冲地址1					
TDES3	缓冲地址2					
TDES4	Reserved					
TDES5	Reserved					
TDES6	时间戳低位时间					
TDES7	时间戳高位时间					

表 39-9 增强型 Tx 描述符

下面分别对 TDES0~TDES7 的具体位做说明。

1) TDES0 的各位功能：同常规型描述符 TDES0 的功能。
2) TDES1 的各位功能：同常规型描述符 TDES1 的功能。
3) TDES2 的各位功能：同常规型描述符 TDES2 的功能。
4) TDES3 的各位功能：同常规型描述符 TDES3 的功能。
5) TDES4、TDES5：Reserved
6) TDES6 的各位功能如下：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TTSL[31:16]															

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

位	标记	位名	功能
b31~b0	TTSL	时间戳地位	DMA将相应发送帧所捕获的时间戳的低32位更新进该字段 *注：在描述符中的最后一个描述符位（TDES0.TLS）置1且发送时间戳状态位（TDES0.TTSS）置1时，该字段才包含时间戳*

7） TDES7 的各位功能如下：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TTSH[31:16]															

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

位	标记	位名	功能
b31~b0	TTSH	时间戳高位	DMA将相应发送帧所捕获的时间戳的高32位更新进该字段 *注：在描述符中的最后一个描述符位（TDES0.TLS）置1且发送时间戳状态位（TDES0.TTSS）置1时，该字段才包含时间戳*

39.4.3.4 DMA 接收配置

RxDMA操作

RxDMA 引擎的操作顺序如下：

1. 用户设置接收描述符（RDES0-RDES3）并将 OWN 位（RDES0[31]）置 1
2. 操作模式寄存器 ETH_DMA_OPRMODR.STR 位置 1 后，DMA 进入运行状态。在运行状态下，DMA 轮询接收描述符列表，尝试获取空闲描述符。如果获得的描述符不空闲（由 CPU 所拥有），则 DMA 进入挂起状态并跳转到步骤 9
3. DMA 将所获取描述符中的接收数据缓冲区地址解码
4. 处理传入帧并将其放入所获取描述符的数据缓冲区
5. 当缓冲区已满或帧传输完成时，接收引擎将获取下一个描述符
6. 如果当前的帧传输完成，DMA 将继续执行第 7 步。如果 DMA 未拥有下一个接收描述符且帧传输尚未完成（尚未传输 EOF 字段），则 DMA 会将 RDES0 中的描述符错误位 DPE 置 1（除非禁止刷新）。DMA 关闭当前描述符（将 OWN 位清零）并通过将最后一个描述符位 (RDES0.RLS) 清零来将其标记为中间描述符（如果禁止刷新，则将其标记为最后一个描述符），随后继续执行第 8 步；如果 DMA 已拥有下一个描述符，但当前的帧传输尚未完成，则 DMA 将关闭当前描述符作为中间值并返回第 4 步
7. 如果已使能 IEEE1588 时间戳功能，DMA 会将时间戳（如果可用）写入当前描述符的 RDES6 和 RDES7。DMA 随后获取所接收帧的状态并将该状态字写入当前描

述符的 RDES0，同时 OWN 位清零且最后一个描述符位置 1

8. 接收引擎检查最新描述符的 OWN 位。如果 CPU 拥有该描述符（OWN 位为 0），则接收缓冲区不可用位（ETH_DMA_DMASTSR.RUS）置 1 且 DMA 接收引擎进入挂起状态（第 9 步）；如果 DMA 拥有该描述符，引擎将返回第 4 步并等待下一个帧
9. 在接收引擎进入挂起状态前，将从 Rx FIFO 中刷新部分帧（可使用 ETH_DMA_OPRMODR 寄存器的 DFRF 位控制刷新）
10. 当收到接收轮询要求命令或者可以从 Rx FIFO 获得下一个帧的起点时，RxDMA 将退出挂起状态。引擎继续执行第 2 步并重新获取下一个描述符

RxDMA 的动作流程图如图 39-20 所示。

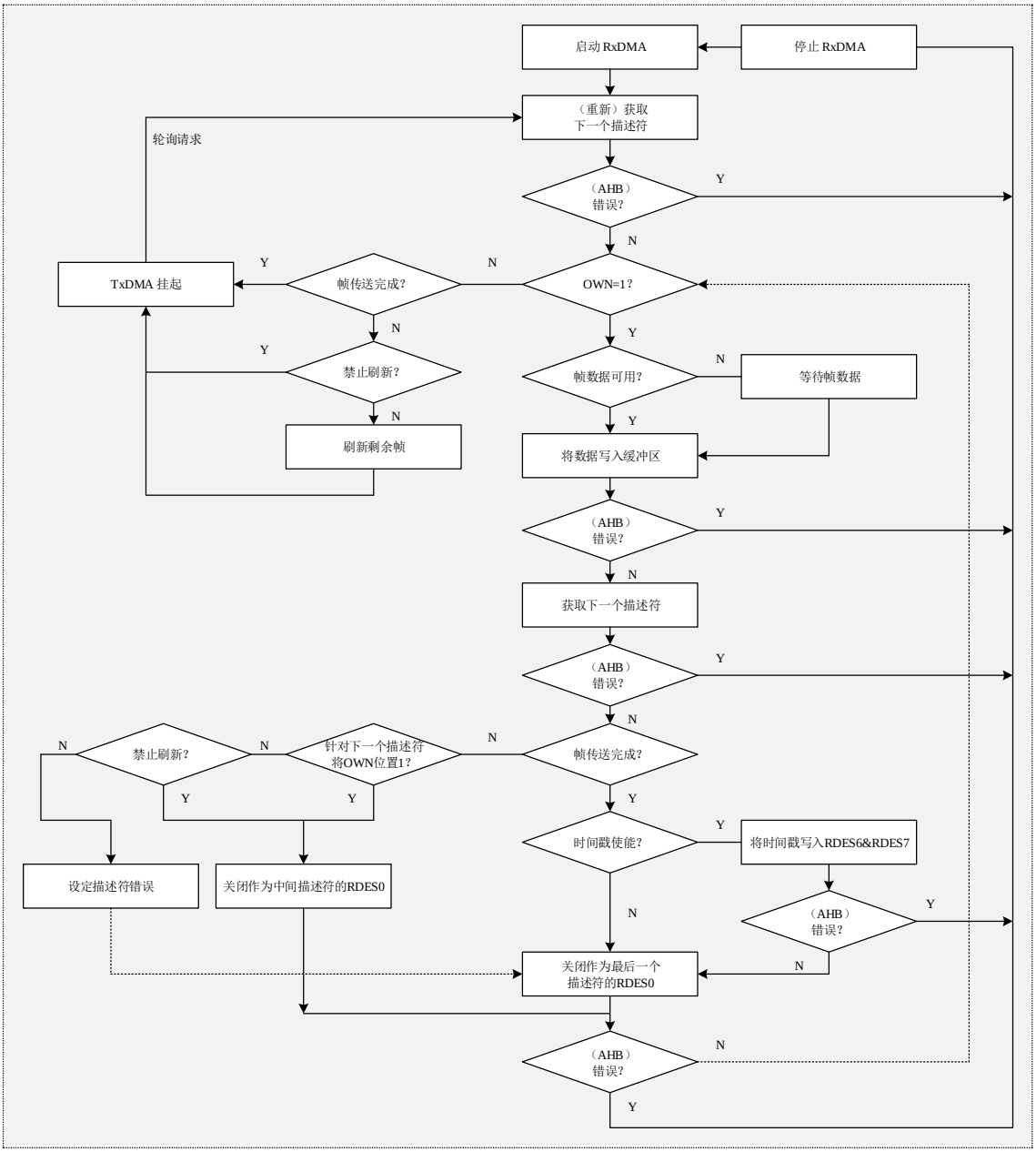


图 39-20 RxDMA 动作流程

RxDMA 的接收状态直到完成时间戳回写并准备将状态回写到描述符时才会确认。如果使能时间戳功能，则当无法获得帧的有效时间戳值时（例如，在写入时间戳前 Rx FIFO 已满），DMA 会向 RDES6 和 RDES7 的所有位写入 1；如果未使能时间戳功能，RDES6 和 RDES7 保持不变。

接收帧获取

接收引擎始终会尝试为即将传入的帧获取一个额外的描述符。只要满足以下任一条件，即尝试获取描述符：

- DMA 进入运行状态后，接收启动/停止位（ETH_DMA_OPRMODR.STR）立即置

1

- 在当前传输的帧结束前，当前描述符的数据缓冲区已满
- MAC 控制器已完成数据帧接收，但当前的接收描述符尚未关闭
- 接收过程因描述符由 CPU 所拥有（RDES0.OWM=0）而挂起，并且接收到新帧
- 已发出接收轮询要求命令

接收帧处理

当帧通过地址过滤且其大小大于或等于为 Rx FIFO 所设置的可配置阈值字节数时，或者在存储转发模式下将整个帧写入 Rx FIFO 时，MAC 才会将接收的帧传输到系统存储器；当帧未通过地址过滤时，将丢弃该帧（除非接收所有位 ETH_MAC_FLTCTLR.RA 置 1）。不足 64 字节的帧由于会发生冲突或过早结束，因而可从 Rx FIFO 中清除。接收 64（可配置阈值）个字节后，DMA 开始将帧数据传输到当前描述符所指定的接收缓冲区中。DMA AHB 接口准备好接收数据传输后，如果 DMA 当前未从存储器获取发送数据，则将第一个描述符位（RDES0.RFS）置 1，以分隔该帧。数据缓冲区已满或者帧的末段已传输到接收缓冲区时，OWN（RDES0）位将复位为 0，此时释放描述符。如果帧包含在单个描述符中，则最后一个描述符位（RDES0.RLS）和第一个描述符位（RDES0.RFS）均置 1。

DMA 获取下一个帧的描述符时，将最后一个描述符位（FDES0.RLS）置 1，并释放前一个帧描述符中的状态位，然后将接收中断位（ETH_DMA_DMASTSR 寄存器 RIS）置 1。该过程将重复执行，直至 DMA 遇到被标记为由 CPU 所有的描述符。这种情况时，接收过程将 ETH_DMA_DMASTSR 寄存器中接收缓冲区不可用位 RUS 置 1，随后进入挂起状态，但其在接收列表中的位置仍然保留。

接收过程挂起

如果在接收过程处于挂起状态时有新的接收帧到达，则 DMA 将重新获取系统存储器中的当前描述符。如果该描述符现在由 DMA 所拥有，接收过程将重新进入运行状态并开始接收帧；如果该描述符仍由主机所拥有，则默认情况下，DMA 将丢弃 Rx FIFO 顶部的当前帧并将丢失帧计数器递增。如果 Rx FIFO 中存储了多个帧，将重复执行该过程。

若将 DMA 工作模式寄存器（ETH_DMA_OPRMODR）的 DFRF 位置 1 后，可避免丢

弃或刷新 Rx FIFO 顶部的帧。在这种情况下，接收过程将动作状态寄存器（ETH_DMA_DMASTSR）的接收缓冲区不可用位 RUS 置 1 并返回到挂起状态。

常规型Rx描述符

常规型 Rx 描述符结构由四个 32 位字组成，如表 39-10 所示，分别定义为 RDES0、RDES1、RDES2 和 RDES3。

RDES0	OWN	状态位 [30:0]				
RDES1	控制位 [31]	Reserved	字节计数缓冲区2 [28:16]	控制位 [15:14]	Reserved	字节计数缓冲区1 [12:0]
RDES2	缓冲地址1					
RDES3	缓冲地址2					

表 39-10 常规型 Rx 描述符

下面分别对 RDES0~RDES3 的具体位做说明。

1） RDES0 的各位功能如下：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
OWN	DAF	FRAL[13:0]													

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ERSUM	DPE	SAF	LEE	OVE	VLAT	RFS	RLS	IPE/TSPA/GF	RLCE	FRAT	WTE	REE	DBE	CRE	DAS/ESA

位	标记	位名	功能
b31	OWN	所有关系位	0：该描述符为CPU所有 1：该描述符为DMA所有 DMA在帧接收完成或此描述符的关联缓冲区已满时将该位清零
b30	DAF	目标地址过滤失败	该位指示帧未能通过MAC内核中的DA过滤
b29~b16	FRAL	帧长统计	该位指示传输到主机存储器中的接收帧字节长度（含CRC） 只有在最后一个描述符位（RDES0.RLS）置1，并且描述符错误位（RDES0.DPE）和上溢错误位（RDES0.OVE）复位时，该字段才有效 当最后一个描述符位和错误汇总位均未置1时，该字段指示为当前帧传输的累计字节数
b15	ERSUM	接收错误汇总	该描述符的B[14]、B[11]、B[7]、B[6]、B[4]、B[3]、B[1]位及REDS4描述符的B[4]、B[3]位中有一个被置位，该位也随之被置位
b14	DPE	描述符错误	该位指示某个帧因超过当前描述符缓冲区的大小而被截断以及DMA未拥有下一个描述符 *注：该位只有在最后一个描述符（RDES0.RLS）置1时才有效*
b13	SAF	源地址过滤失败	该位指示帧的SA字段未能通过MAC内核中的SA过滤

b12	LEE	长度错误	该位指示接收帧的实际长度与长度/类型字段的值不符 *注1：该位仅在帧类型位（RDES0.FRAT）复位后有效* *注2：在CRC错误位（RDES0.CRE）有效时，该位无效*
b11	OVE	上溢错误	该位指示接收帧因缓冲区上溢而损坏
b10	VLAT	VLAN标识符	该位指示描述符所指向的帧是由MAC内核标记的VLAN帧
b9	RFS	第一个描述符	该位指示此描述符包含帧的第一个缓冲区。如果第一个缓冲区的大小为0，则第二个缓冲区将包含帧的帧头；如果第二个缓冲区的大小为0，则下一个描述符将包含帧的帧头
b8	RLS	最后一个描述符	该位指示此描述符指向的缓冲区为帧的最后一个缓冲区
b7	IPE/TSPA/GF	COE错误/时间戳/巨帧	当Checksum Offload功能有效时： 该位指示IPv4或IPv6报头中存在错误 导致此错误的原因可能是： 1）以太网类型字段与IP报头版本字段值不一致 2）IPv4中报头的校验和不匹配 3）以太网帧缺少所需的IP报头字节数 当时间戳功能有效时： 该位指示有一个时间戳快照被记录在了RDES6和RDES7中 *注：该位只在最后一个描述符（RDES0.RLS）置1时有效* 以上两个均无效时： 该位指示MAC接收到一个巨帧
b6	RLCE	延迟冲突错误	该位指示在以半双工模式接收帧时发生了延迟冲突
b5	FRAT	帧类型	0：MAC接收到以太网帧 1：MAC接收到PTP帧 *注：当接收到一个矮帧时，该位无效*
b4	WTE	看门狗错误	该位指示接收看门狗计时器在接收当前帧时超时，且当前帧在看门狗超时后被截断
b3	REE	接收错误	该位指示在帧接收期间，PHY发出RX_DV信号时，因为发出了RX_ER信号
b2	DBE	Dribble位错误	该位指示接收的帧具有非整数倍数的字节（奇数半字节） *注：该位仅在MII模式下有效*
b1	CRE	CRC错误	该位指示接收的帧发生循环冗余校验（CRC）错误 *注：该位只在最后一个描述符（RDES0.RLS）置1有效*
b0	DAS/ESA	地址过滤成功/状态位扩展	在COE功能和PTP功能无效时： 0：接收帧通过了MAC地址寄存器0的DA地址过滤 1：接收帧通过了MAC地址寄存器1~5的DA地址过滤 *注：当该描述符的DAF位置位时，该位无效* 在COE功能或PTP功能有效时： 该位指示RDES4描述符有效 *注：该位只在最后一个描述符（RDES0.RLS）置1时有效*

2） RDES1 的各位功能如下：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DIC	Reserved		RBS2[12:0]												

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RER	RSAC	-	RBS1[12:0]												

位	标记	位名	功能
b31	DIC	完成时禁止中断	该位置1时，在接收帧在由该描述符所指示的缓冲区中结束时禁止引发主机的中断，同时也阻止动作状态寄存器（ETH_DMA_DMASTSR）的RIS位置1 *注：该位仅在最后一个描述符位（RDES0.RLS）置1的情况下有效*
b30~29	Reserved	-	-
b28~b16	RBS2	缓冲区2大小	该位以字节为单位指示第二个数据缓冲区的大小 *注：该位在该描述符的RSAC位置1时无效*
b15	RER	环接收结束	该位指示描述符列表已到达其最后一个描述符，DMA会返回描述符列表的首地址，形成一个描述符环
b14	RSAC	第二个地址链接	该位置1时，描述符中的第二个地址是下一个描述符地址，而非第二个缓冲区地址，此时该描述符的RBS2位为无效 *注：RER位优先级高于RSAC位*
b13	Reserved	-	-
b12~b0	RBS1	缓冲区1大小	该位以字节为单位指示第一个数据缓冲区的大小。如果该字段为0，DMA将忽略该缓冲区并使用缓冲区2或下一个描述符，具体取决于该描述符的RSAC的设定

3） RDES2 的各位功能如下：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RBAP1[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RBAP1[15:0]															

位	标记	位名	功能
b31~b0	RBAP1	接收缓冲区1地址	该位向DMA指示数据在存储器中的位置，当软件为DMA提供此描述符（RDES0中的OWN位置1）时，这些位将指示缓冲区1的物理地址

4） RDES3 的各位功能如下：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RBAP2[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RBAP2[15:0]															

位	标记	位名	功能
b31~b0	RBAP2	接收缓冲区2地址/下一个描述符地址	该位向DMA指示数据在存储器中的位置，当软件为DMA提供此描述符（RDES0中的OWN位置1）并且使用描述符环结构时，这些位将指示缓冲区2的物理地址；如果RDES1.RSAC位置1，则该地址包含下一个描述符所在物理寄存器的指针 *注：只有在RDES1.RSAC位置1时，缓冲区地址指针才必须与总线宽度相符*

增强型Rx描述符

当时间戳功能有效（ETH_PTP_TSPCTLR.TSPEN=1）时或 Checksum Offload 功能有效（ETH_MAC_CONFIGR.IPCO=1）时，应使用增强型 Rx 描述符。

增强型 Tx 描述符结构由 8 个 32 位字组成，如表 39-11 所示，分别定义为 RDES0、RDES1、RDES2、RDES3、RDES4、RDES5、RDES6 和 RDES7。其中 RDES0~RDES3 的功能与常规型相同。

RDES0	OWN	状态位 [30:0]				
RDES1	控制位 [31]	Reserved	字节计数缓冲区2 [28:16]	控制位 [15:14]	Reserved	字节计数缓冲区1 [12:0]
RDES2	缓冲地址1					
RDES3	缓冲地址2					
RDES4	扩展状态位 [31:0]					
RDES5	Reserved					
RDES6	时间戳低位时间					
RDES7	时间戳高位时间					

表 39-11 增强型描述符

下面分别对 RDES0~RDES7 的具体位做说明。

1) RDES0 的各位功能：同常规型描述符 RDES0 的功能。

2) RDES1 的各位功能：同常规型描述符 RDES1 的功能。

3) RDES2 的各位功能：同常规型描述符 RDES2 的功能。

4) RDES3 的各位功能：同常规型描述符 RDES3 的功能。

5) RDES4 的各位功能如下：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved						L4FMS	L3FMS	Reserved							

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	TSPD	PTPV	PTPFT	MTP[3:0]				IPv6 DR	IPv4 DR	IPCB	IPPE	IPHE	IPPT[2:0]		

位	标记	位名	功能
b31~b16	Reserved	-	-
b25	L4FMS	L4过滤成功	该位指示接收帧通过了L4端口过滤
b24	L3FMS	L3过滤成功	该位指示接收帧通过了L3地址过滤
b23~b15	Reserved	-	-
b14	TSPD	时间戳丢弃	该位指示由于Rx FIFO溢出，虽进行了时间戳快照但是丢弃了时间戳
b13	PTPV	PTP版本	该位指示接收到的PTP帧的版本类型 0：IEEE1588v1版本的PTP帧 1：IEEE1588v2版本的PTP帧
b12	PTPFT	PTP帧类型	该位指示接收到的PTP帧类型 0：Over Ethernet类型的PTP帧 1：Over UDP-IPv4或Over UDP-IPv6类型的PTP帧 *注：具体为Over UDP-IPv4还是Over UDP-IPv6可通过该描述符的IPv6FR、IPv4FR位获取*
b11~b8	MTP	报文类型	该位指示接收到的报文类型 0000：未接收到任何 PTP 消息 0001：SYNC（所有时钟类型） 0010：Follow_Up（所有时钟类型） 0011：Delay_Req（所有时钟类型） 0100：Delay_Resp（所有时钟类型） 0101：Pdelay_Req（在点对点透明时钟中） 0110：Pdelay_Resp（在点对点透明时钟中） 0111：Pdelay_Resp_Follow_Up（在点对点透明时钟中） 1000：Announce 1001：Management 1010：Signaling 1111：默认报文类型 其它：保留

b7	IPv6DR	接收到IPv6报	该位指示接收到IPv6报
b6	IPv4DR	接收到IPv4报	该位指示接收到IPv4报
b5	IPCB	Checksum Offload直通	该位表示绕过COE引擎
b4	IPPE	IP有效负载错误	以下情况时，该位置1： 1）当内核计算的16位IP有效负载校验和（即TCP、UDP或ICMP校验和）与接收帧中对应的校验和字段不匹配 2）当TCP、UDP或ICMP段长度与IP报头字段中的有效负载长度值不匹配
b3	IPHE	IP头错误	该位指示由内核计算的16位IPv4报头校验和与接收的校验和字节不匹配，或IP报头版本字段与以太网类型值不一致
b2~b0	IPPT	有效负载类型	该位表示IP数据报中封装的有效负载类型，当IP报头出错或存在分段的IP时，这些位为“000” 000：未知，或未处理IP有效负载 001：UDP 010：TCP 011：ICMP 1xx：保留

6) RDES5：Reserved

7) RDES6 的各位功能如下：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RTSL[31:16]															

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

位	标记	位名	功能
b31~b0	RTSL	时间戳地位	DMA将相应接收帧所捕获的低32位更新进该字段。 *注：只有在描述符中最后一个描述符位（RDES0.RLS）置1时，该字段才包含时间戳*

8) RDES7 的各位功能如下：

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RTSH[31:16]															

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

位	标记	位名	功能
b31~b0	RTSH	时间戳高位	DMA将相应接收帧所捕获的高32位更新进该字段。 *注：只有在描述符中最后一个描述符位（RDES0.RLS）置1时，该字段才包含时间戳*

39.5 中断说明

下图 39-21 是以太网 MAC 控制器中的中断实现方案。从图中可以看出，ETHMAC 将 DMA 的正常事件（NIS）、DMA 的异常事件（AIS）、PMT 事件（PMTIS）、MMC 事件（MMC**IS）、PTP 事件（TSPIS）等通过中断的使能或屏蔽控制汇总成一个中断事件输出至中断模块，所以系统在中断响应时，需要通过读取相应的状态信息来确定是哪种类型的事件产生了中断。

同时，ETHMAC 的 PMT 模块还有一个非屏蔽的中断输出。在 PMT 事件产生时，不管中断屏蔽寄存器（ETH_MAC_INTMSKR）的 PMTIM 位是否置 1，都会产生 PMT 非屏蔽中断。

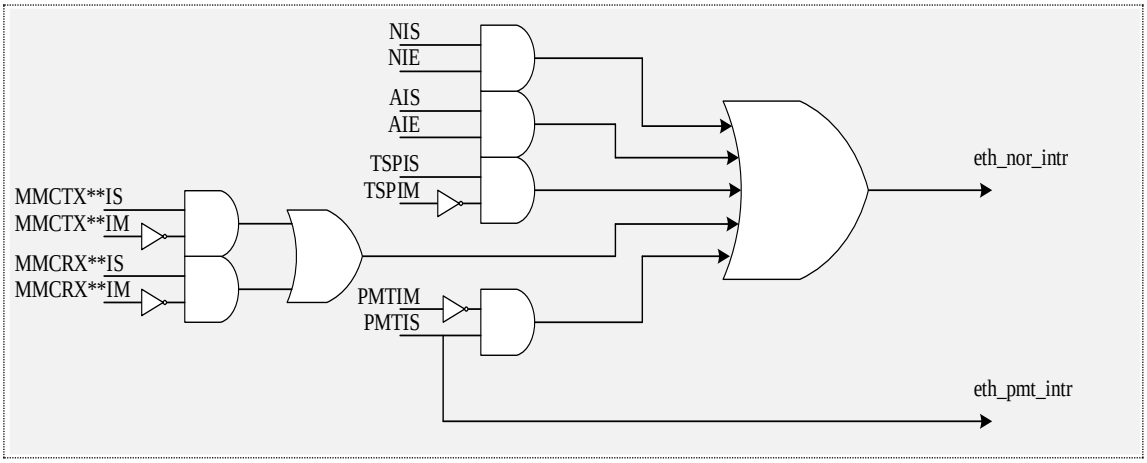


图 39-21 ETHMAC 中断方案

39.5.1 DMA 中断

DMA 在发送数据和接收数据过程中产生的各种正常事件会触发普通中断、各种异常事件会触发异常中断。如图 39-22 所示。

各具体事件的中断控制可通过 DMA 中断使能寄存器（ETH_DMA_INTENAR）的各个位控制。参见 DMA 中断控制寄存器章节。

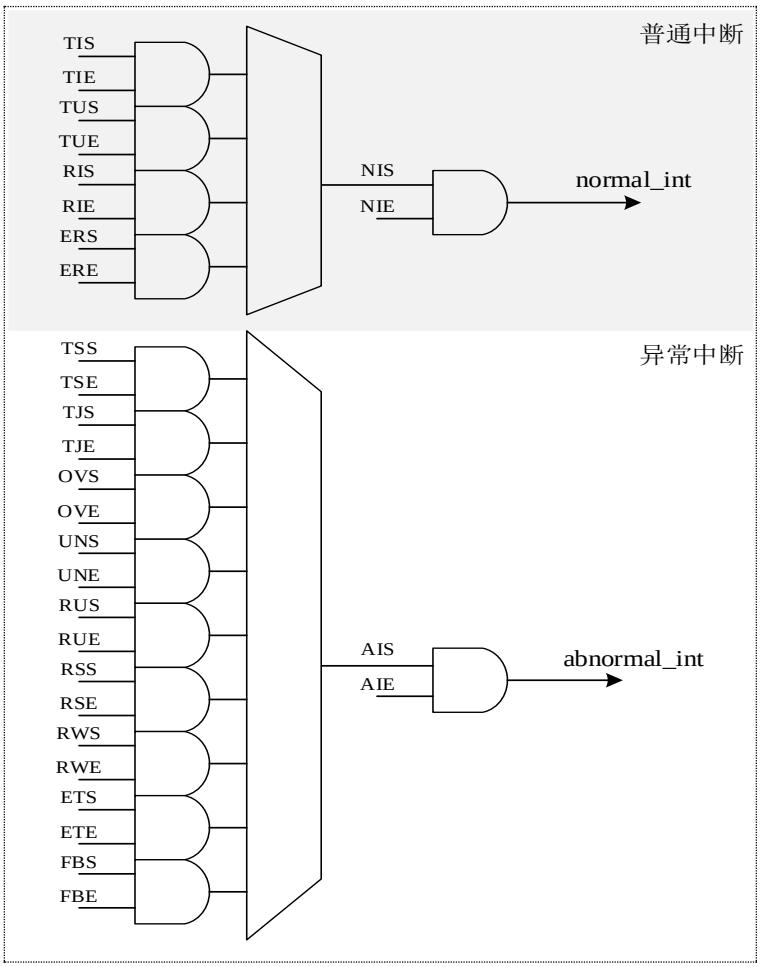


图 39-22 DMA 中断组成

39.5.2 PMT 中断

MAC 在接收到远程唤醒帧和魔术数据包时可触发生成 PMT 中断（可屏蔽中断和非屏蔽中断）。其中可屏蔽中断的生成由 MAC 中断屏蔽寄存器（ETH_MAC_INTMSKR）的 PMTIM 位控制。

39.5.3 PTP 中断

在 PTP 功能有效时，系统时间到达时间戳目标时间寄存器 0~1 中的设定值时可触发生成 PTP 中断。该中断由 PPS 输出控制寄存器（ETH_MAC_PPSCTLR）的 TT0SEL、TT1SEL 位和 MAC 中断屏蔽寄存器（ETH_MAC_INTMSKR）的 TSPIM 位控制。

39.5.4 MMC 中断

在 MMC 中各项发送帧统计数和接收帧统计数溢出时可触发生成 MMC 中断。该中断

由 MMC 的 Rx 中断控制寄存器（ETH_MMC_RITCTLR）和 Tx 中断控制寄存器（ETH_MMC_TITCTLR）的各个位控制。参见 MMC Rx 中断控制寄存器和 Tx 中断控制器章节。

39.6 寄存器说明

表 39-12 所示，为 ETHMAC 模块的寄存器列表。

BASE ADDR: 0x40060000H

寄存器名	符号	偏移量	位宽	复位值
ETH_MAC 接口选择寄存器	ETH_MAC_IFCONFR	（0x40055410h）	32	0x00000000h
ETH_MAC MAC配置寄存器	ETH_MAC_CONFIGR	0x0000h	32	0x00008000h
ETH_MAC MAC流控寄存器	ETH_MAC_FLOCTLR	0x0018h	32	0x00000000h
ETH_MAC MAC状态寄存器	ETH_MAC_MACSTSR	0x0024h	32	0x00000000h
ETH_MAC 中断状态寄存器	ETH_MAC_INTSTSR	0x0038h	32	0x00000000h
ETH_MAC 中断屏蔽寄存器	ETH_MAC_INTMSKR	0x003Ch	32	0x00000000h
ETH_MAC SMI地址寄存器	ETH_MAC_SMIADDR	0x0010h	32	0x00000000h
ETH_MAC SMI数据寄存器	ETH_MAC_SMIDATR	0x0014h	32	0x00000000h
ETH_MAC PMT控制状态寄存器	ETH_MAC_PMTCTLR	0x002Ch	32	0x00000000h
ETH_MAC 远程唤醒帧过滤寄存器	ETH_MAC_RTWKFFR	0x0028h	32	0x00000000h
ETH_MAC MAC帧过滤控制寄存器	ETH_MAC_FLTCTLR	0x0004h	32	0x00000000h
ETH_MAC MAC地址高位寄存器0	ETH_MAC_MACADHR0	0x0040h	32	0x8000FFFFh
ETH_MAC MAC地址地位寄存器0	ETH_MAC_MACADLR0	0x0044h	32	0xFFFFFFFFh
ETH_MAC MAC地址高位寄存器1	ETH_MAC_MACADHR1	0x0048h	32	0x0000FFFFh
ETH_MAC MAC地址地位寄存器1	ETH_MAC_MACADLR1	0x004Ch	32	0xFFFFFFFFh
ETH_MAC MAC地址高位寄存器2	ETH_MAC_MACADHR2	0x0050h	32	0x0000FFFFh
ETH_MAC MAC地址地位寄存器2	ETH_MAC_MACADLR2	0x0054h	32	0xFFFFFFFFh
ETH_MAC MAC地址高位寄存器3	ETH_MAC_MACADHR3	0x0058h	32	0x0000FFFFh
ETH_MAC MAC地址地位寄存器3	ETH_MAC_MACADLR3	0x005Ch	32	0xFFFFFFFFh
ETH_MAC MAC地址高位寄存器4	ETH_MAC_MACADHR4	0x0060h	32	0x0000FFFFh
ETH_MAC MAC地址地位寄存器4	ETH_MAC_MACADLR4	0x0064h	32	0xFFFFFFFFh
ETH_MAC MAC Hash表高位寄存器	ETH_MAC_HASHTHR	0x0008h	32	0x00000000h
ETH_MAC MAC Hash表低位寄存器	ETH_MAC_HASHTLR	0x000Ch	32	0x00000000h
ETH_MAC VLAN标记发送控制寄存器	ETH_MAC_VTACTLR	0x0584h	32	0x00000000h
ETH_MAC VLAN标记接收过滤寄存器	ETH_MAC_VTAFLTR	0x001Ch	32	0x00000000h
ETH_MAC VLAN Hash表寄存器	ETH_MAC_VLAHTBR	0x0588h	32	0x00000000h
ETH_MAC LAY3LAY4控制寄存器	ETH_MAC_L34CTLR	0x0400h	32	0x00000000h
ETH_MAC LAY4端口寄存器	ETH_MAC_L4PORTR	0x0404h	32	0x00000000h
ETH_MAC LAY3地址寄存器0	ETH_MAC_L3ADDRR0	0x0410h	32	0x00000000h

ETH_MAC LAY3地址寄存器1	ETH_MAC_L3ADDRR1	0x0414h	32	0x00000000h
ETH_MAC LAY3地址寄存器2	ETH_MAC_L3ADDRR2	0x0418h	32	0x00000000h
ETH_MAC LAY3地址寄存器3	ETH_MAC_L3ADDRR3	0x041Ch	32	0x00000000h
ETH_PTP 时间戳控制寄存器	ETH_PTP_TSPCTLR	0x0700h	32	0x00002000h
ETH_PTP 时间戳状态寄存器	ETH_PTP_TSPSTSR	0x0728h	32	0x00000000h
ETH_PTP 时间戳基本加数寄存器	ETH_PTP_TSPADDR	0x0718h	32	0x00000000h
ETH_PTP 时间戳亚秒加数寄存器	ETH_PTP_TSPNSAR	0x0704h	32	0x00000000h
ETH_PTP 时间戳系统秒寄存器	ETH_PTP_TMSSECR	0x0708h	32	0x00000000h
ETH_PTP 时间戳系统亚秒寄存器	ETH_PTP_TMSNSER	0x070Ch	32	0x00000000h
ETH_PTP 时间戳更新秒寄存器	ETH_PTP_TMUSECR	0x0710h	32	0x00000000h
ETH_PTP 时间戳更新亚秒寄存器	ETH_PTP_TMUNSER	0x0714h	32	0x00000000h
ETH_PTP 时间戳目标秒寄存器0	ETH_PTP_TMTSECR0	0x071Ch	32	0x00000000h
ETH_PTP 时间戳目标亚秒寄存器0	ETH_PTP_TMTNSER0	0x0720h	32	0x00000000h
ETH_PTP 时间戳目标秒寄存器1	ETH_PTP_TMTSECR1	0x0780h	32	0x00000000h
ETH_PTP 时间戳目标亚秒寄存器1	ETH_PTP_TMTNSER1	0x0784h	32	0x00000000h
ETH_PTP PPS输出控制寄存器	ETH_PTP_PPSCTLR	0x072Ch	32	0x00000000h
ETH_DMA 总线模式寄存器	ETH_DMA_BUSMODR	0x1000h	32	0x00020101h
ETH_DMA 操作模式寄存器	ETH_DMA_OPRMODR	0x1018h	32	0x00000000h
ETH_DMA 动作状态寄存器	ETH_DMA_DMASTSR	0x1014h	32	0x00000000h
ETH_DMA 中断使能寄存器	ETH_DMA_INTENAR	0x101Ch	32	0x00000000h
ETH_DMA 帧丢失统计寄存器	ETH_DMA_RFRCNTR	0x1020h	32	0x00000000h
ETH_DMA 看门狗定时寄存器	ETH_DMA_REVWDTR	0x1024h	32	0x00000000h
ETH_DMA 发送轮询要求寄存器	ETH_DMA_TXPOLLR	0x1004h	32	0x00000000h
ETH_DMA 接收轮询要求寄存器	ETH_DMA_RXPOLLR	0x1008h	32	0x00000000h
ETH_DMA 发送描述符列表地址寄存器	ETH_DMA_TXDLADR	0x1010h	32	0x00000000h
ETH_DMA 接收描述符列表地址寄存器	ETH_DMA_RXDLADR	0x100Ch	32	0x00000000h
ETH_DMA 当前主机发送描述符寄存器	ETH_DMA_CHTXDER	0x1048h	32	0x00000000h
ETH_DMA 当前主机接收描述符寄存器	ETH_DMA_CHRXDER	0x104Ch	32	0x00000000h
ETH_DMA 当前主机发送缓冲区寄存器	ETH_DMA_CHTXBFR	0x1050h	32	0x00000000h
ETH_DMA 当前主机接收缓冲区寄存器	ETH_DMA_CHRXBFR	0x1054h	32	0x00000000h
ETH_MMC MMC控制寄存器	ETH_MMC_MMCCTLR	0x0100h	32	0x00000000h
ETH_MMC Rx统计状态寄存器	ETH_MMC_REVSTSR	0x0104h	32	0x00000000h
ETH_MMC Tx统计状态寄存器	ETH_MMC_TRSSTSR	0x0108h	32	0x00000000h
ETH_MMC Rx中断控制寄存器	ETH_MMC_RITCTLR	0x010Ch	32	0x00000000h
ETH_MMC Tx中断控制寄存器	ETH_MMC_TITCTLR	0x0110h	32	0x00000000h

ETH_MMC Rx单播好帧统计寄存器	ETH_MMC_RXUNGFR	0x01C4h	32	0x00000000h
ETH_MMC Rx多播好帧统计寄存器	ETH_MMC_RXMUGFR	0x0190h	32	0x00000000h
ETH_MMC Rx广播好帧统计寄存器	ETH_MMC_RXBRGFR	0x018Ch	32	0x00000000h
ETH_MMC RxCRC错误帧统计寄存器	ETH_MMC_RXCREFR	0x0194h	32	0x00000000h
ETH_MMC Rx对齐错误帧统计寄存器	ETH_MMC_RXALEFR	0x0198h	32	0x00000000h
ETH_MMC Rx矮帧错误帧统计寄存器	ETH_MMC_RXRUEFR	0x019Ch	32	0x00000000h
ETH_MMC Rx长度错误帧统计寄存器	ETH_MMC_RXLEEFR	0x01C8h	32	0x00000000h
ETH_MMC Rx超范围错误帧统计寄存器	ETH_MMC_RXOREFR	0x01CCh	32	0x00000000h
ETH_MMC Tx单播好帧统计寄存器	ETH_MMC_TXUNGFR	0x0168h	32	0x00000000h
ETH_MMC Tx多播好帧统计寄存器	ETH_MMC_TXMUGFR	0x0120h	32	0x00000000h
ETH_MMC Tx广播好帧统计寄存器	ETH_MMC_TXBRGFR	0x011Ch	32	0x00000000h
ETH_MMC Tx延迟错误帧统计寄存器	ETH_MMC_TXDEEFR	0x0154h	32	0x00000000h
ETH_MMC Tx载波错误帧统计寄存器	ETH_MMC_TXCAEFR	0x0160h	32	0x00000000h
ETH_MMC Tx延迟冲突错误帧统计寄存器	ETH_MMC_TXLCEFR	0x0158h	32	0x00000000h
ETH_MMC Tx过度冲突错误帧统计寄存器	ETH_MMC_TXECEFR	0x015Ch	32	0x00000000h
ETH_MMC Tx过度延迟错误帧统计寄存器	ETH_MMC_TXEDEFR	0x016Ch	32	0x00000000h

表 39-12 ETHMAC 寄存器列表

39.6.1 ETH_MAC 寄存器

39.6.1.1 ETH_MAC 接口选择寄存器（ETH_MAC_IFCONFR）

复位值: 0x00000000h

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										TCK INV	RCK INV	-	-	-	IF SEL

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	TCKINV	TX时钟取反	0：从ETH_MII_TXCLK端口输入的时钟保持原极性 1：从ETH_MII_TXCLK端口输入的时钟极性取反	R/W
b4	RCKINV	RX时钟取反	0：从ETH_MII_RMII_RXCLK端口输入的时钟保持原极性 1：从ETH_MII_RMII_RXCLK端口输入的时钟极性取反	R/W
b3~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	IFSEL	接口选择位	0：MII接口 1：RMII接口	R/W

39.6.1.2 ETH_MAC MAC 配置寄存器（ETH_MAC_CONFIGR）

复位值: 0x00008000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	SAIRC[2:0]			-	-	CST	-	MWD	MJB	-	-	IFG[2:0]			DCRS

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	FES	DO	LM	DM	IPCO	DRTY	-	ACS	BL[1:0]		DC	TE	RE	-	-

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	SAIRC	源地址插入及替换	010：将MAC地址寄存器0中的地址值作为SA地址插入到发送帧中 110：将MAC地址寄存器1中的地址值作为SA地址插入到发送帧中 011：将MAC地址寄存器0中的地址值作为SA地址替换发送帧中的SA字段 111：将MAC地址寄存器1中的地址值作为SA地址替换发送帧中的SA字段 x0x：发送帧中SA字段地址的处理由Tx描述符TDES1的SAIRC位控制	R/W
b27~b26	Reserved	-	读出时为“0”,写入时写“0”	R/W
b25	CST	TYPE帧FCS去除	0：该功能无效 1：在将帧转发到应用之前，去除并丢弃所有TYPE（类型字段大于0x0600）帧的最后4个字节（FCS字段）	R/W
b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	MWD	屏蔽看门狗	0：MAC允许接收的帧不超过2048字节，并会截断超过此限制接收的任何字节 1：MAC禁止接收器上的看门狗定时器，并可接收多达16384字节的帧	R/W
b22	MJB	屏蔽Jabber	0：发送期间应用发送超过2048字节的数据，MAC会截断发送器 1：MAC禁止发送器上的Jabber定时器，并可发送多达16384字节的帧	R/W
b21~b20	Reserved	-	读出时为“0”,写入时写“0”	R/W
b19~b17	IFG	帧间间隙	发送期间帧间的最小间隙 000：96 位时间 001：88 位时间 010：80 位时间 …… 111：40 位时间 *注： 在半双工模式下，最小IFG仅能配置为64位时间（IFG=100），不考虑更低的值*	R/W
b16	DCRS	屏蔽载波侦听	0：半双工模式下帧发送期间的MII_CRS信号有效时，MAC发送器会生成载波侦听错误，甚至中止发送	R/W

			1：半双工模式下帧发送期间的MII_CRS信号被忽略,不会因在此发送期间载波丢失或无载波而生成错误 *注：该位只在半双工模式时有效，全双工模式时无效*	
b15	Reserved	-	读出时为“1”,写入时写“1”	R/W
b14	FES	速度选择	0：10Mbps 1：100Mbps	R/W
b13	DO	屏蔽接收自身包	0：在半双工模式下，MAC接收来自PHY的所有包 1：在半双工模式下，MAC禁止接收帧 *注：该位只在半双工模式时有效，全双工模式时无效*	R/W
b12	LM	Loopback模式	0：Loopback模式无效 1：Loopback模式有效	R/W
b11	DM	全双工模式	0：半双工模式 1：全双工模式	R/W
b10	IPCO	Checksum Offload功能	0：IP Checksum Offload功能无效 1：IP Checksum Offload功能有效	R/W
b9	DRTY	屏蔽重试	0：在MII模式下发生冲突时，MAC会尝试根据该寄存器的BL位的设置进行重试 1：在MII模式下发生冲突时，MAC会忽略当前帧发送并以发送帧状态下过度冲突错误报告帧中止，即MAC仅尝试一次发送 *注：该位只在半双工模式时有效，全双工模式时无效*	R/W
b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	ACS	自动去除PAD/FCS	0：MAC不管接收到的帧的长度，传送所有接收帧到应用 1：MAC在长度字段值小于1536字节时自动去除接收帧上的PAD/FCS字段，在长度字段大于或等于1536字节时不去除PAD/FCS字段	R/W
b6~b5	BL	后退限制	后退限制决定发生冲突后重试期间MAC在重新安排一次发送之前等待的随机整数（r）个时间片延迟（512位时间） 00: k = min（n, 10） 01: k = min（n, 8） 10: k = min（n, 4） 11: k = min（n, 1） 其中n=重新发送尝试的次数。随机整数r的取值范围为：0<= r<2^k *注：该位只在半双工模式时有效，全双工模式时无效*	R/W
b4	DC	延迟检查	0：禁止延迟检查功能。MAC发生延迟，直到CRS信号变为无效信号 1：使能延迟检查功能。当发送状态机在延迟超过24288位时间时，MAC将指示帧中止状态，同时在发送帧状态中将过度延迟错误位置1 *注1：延迟的产生：当发送器准备好发送但因MII上含有有效CRS（载波侦听）信号而被阻止时延迟计数就开始。延迟时间是非累积性的，如果发送器延迟10000位时间，则重新执行一遍发送、冲突、后退，然后在完成后退后复位延迟计数器为0并重新开始计数*	R/W

			注2：该位只在半双工模式时有效，全双工模式时无效	
b3	TE	发送使能	0：发送无效 1：发送使能	R/W
b2	RE	接收使能	0：接收无效 1：接收使能	R/W
b1~b0	-	-	读出时为“0”,写入时写“0”	R/W

39.6.1.3 ETH_MAC MAC 流控寄存器（ETH_MAC_FLOCTLR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PAUSET[15:0]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								DZPQ	-	PLT[1:0]		UNP	RFE	TFE	FCA_BPA

位	标记	位名	功能	读写
b31~b16	PAUSET	暂停时间	该位表示发送控制帧中暂停时间字段要使用的值	R/W
b15~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	DZPQ	屏蔽零时间片帧	0：当流控制信号有效时，正常生成零时间片暂停控制帧 1：当流控制信号有效时，禁止生成零时间片暂停控制帧	R/W
b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5~b4	PLT	暂停阈值下限	配置暂停定时器的阈值，达到该值时会自动重新传输PAUSE帧 00：暂停时间减去4个时隙 01：暂停时间减去28个时隙 10：暂停时间减去144个时隙 11：暂停时间减去256个时隙 *注1：时隙定义为MII接口每发送512位（64字节）所需的时间* *注2：该阈值应始终小于暂停时间。例如，如果暂停时间PAUSET=100H（256个时隙），而PLT=01，则在第一个PAUSE帧发送完成后，第二个PAUSE帧会在第228 (256-28) 个时隙启动时自动发送*	R/W
b3	UNP	单播暂停帧检测	0：MAC仅检测具有802.3x标准中指定的唯一多播地址的暂停帧 1：MAC除了检测具有唯一多播地址的暂停帧外，还会检测具有ETH_MAC_MACADHR0和ETH_MAC_MACADLR0寄存器所指定的站单播地址的暂停帧	R/W
b2	RFE	接收流控制使能	0：禁止暂停帧的解码功能 1：MAC对接收到的暂停帧进行解码，并禁止其在指定的暂停时间内发送	R/W
b1	TFE	发送流控制使能	全双工模式时： 0：禁止MAC传送任何暂停帧 1：使能流控制操作，MAC发送暂停帧 半双工模式时： 0：禁止背压功能 1：使能背压操作	R/W
b0	FCA_BPA	流控制状态	向流控制寄存器写入数据前此位应读为0 全双工模式时： 此位置1时MAC内核启动暂停控制帧。在控制帧发送过程中，此	R/W

	位保持置1以指示帧发送正在进行中。当暂停控制帧发送完成后，MAC会将此位复位为0。此位清零后，才可以对流控制寄存器执行写操作 在半双工模式时： 此位置1时，MAC内核将置位背压功能。在背压操作期间，当MAC接收到新帧时，发送器会开始发送一个导致冲突的Jam信号

39.6.1.4 ETH_MAC MAC 状态寄存器（ETH_MAC_MACSTSR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserced						TFF	TFNE	-	TFWA	TFRS[1:0]		MTP	MTS[1:0]		MTEA

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved						RFFL[1:0]		-	RFRS[1:0]		RFWA	-	MRS[1:0]		MREA

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	TFF	Tx FIFO满	0：Tx FIFO未满 1：Tx FIFO已满，无法再接收发送的帧	R
b24	TFNE	Tx FIFO非空	0：Tx FIFO空 1：Tx FIFO非空，还有未发送的帧	R
b23	Reserved	-	读出时为“0”,写入时写“0”	R/W
b22	TFWA	Tx FIFO写有效	0：Tx FIFO写无效 1：Tx FIFO写控制器有效且正在将数据传输到Tx FIFO	R
b21~b20	TFRS	Tx FIFO读状态	00：空闲态 01：读状态（数据从Tx FIFO往MAC发送器传送） 10：等待来自MAC发送器的Tx Status 11：写入收到的Tx Status或刷新Tx FIFO	R
b19	MTP	MAC发送器暂停	0：MAC发送器处于发送数据状态 1：MAC发送器处于暂停状态（全双工模式时），不安排任何帧发送	R
b18~b17	MTS	MAC发送器状态	00：空闲态 01：等待前一个帧的状态或IFG/回退阶段结束 10：生成并发送暂停控制帧（在全双工模式下） 11：传输要发送的输入帧	R
b16	MTEA	MAC MII发送引擎状态	0：MAC MII发送引擎处于空闲态 1：MAC MII发送引擎正在主动发送数据而未处于空闲状态	R
b15~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9~b8	RFFL	Rx FIFO填充级别	00：Rx FIFO为空 01：Rx FIFO填充级别低于流控制，取消激活阈值 10：Rx FIFO填充级别高于流控制，激活阈值 11：Rx FIFO已满	R
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6~b5	RFRS	Rx FIFO读状态	00：空闲态 01：读取帧数据 10：读取帧状态或时间戳 11：刷新帧数据和状态	R
b4	RFWA	Rx FIFO写有效	0：Rx FIFO写无效	R

			1：Rx FIFO写有效，并正在将收到的帧传输到Rx FIFO	
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~b1	MRS	MAC接收器状态	00：空闲 01：MAC接收器中的FIFO写控制器在工作 10：MAC接收器中的FIFO读控制器在工作 11：MAC接收器中的FIFO读和写控制器在工作	R
b0	MREA	MAC MII接收引擎状态	0：MAC MII接收引擎处于空闲态 1：MAC MII接收引擎正在主动接收数据而未处于空闲状态	R

39.6.1.5 ETH_MAC 中断状态寄存器（ETH_MAC_INTSTSR）

复位值: 0x00000000h

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						TSPIS	-	-	MMC TXIS	MMC RXIS	MMC IS	PMTIS	-	-	-

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	TSPIS	PTP时间戳中断	当系统时间值等于或大于目标时间寄存器中指定的值时，此位置1 *注：该位在读取ETH_PTP_TSPSTSR寄存器的Bit0位时清零*	R
b8~b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6	MMCTXIS	MMC发送统计中断	ETH_MMC_TITCTLR寄存器中生成中断，此位就变为高电平	R
b5	MMCRXIS	MMC接收统计中断	ETH_MMC_RITCTLR寄存器中生成中断，此位就变为高电平	R
b4	MMCIS	MMC统计中断	该寄存器的MMCTXIS或MMCRXIS位有高电平，此位就变为高电平 *注：该位在ETH_MAC_PMTCTLR寄存器的WKFR和MPFR位均变为0时清零*	R
b3	PMTIS	PMT中断	在Power Down模式下接收到魔术数据包或远程唤醒帧，此位就变为高电平	R
b2~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

39.6.1.6 ETH_MAC 中断屏蔽寄存器（ETH_MAC_INTMSKR）

复位值: 0x00000000h

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						TSPIM	Reserved					PMTIM	Reserved		

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	TSPIM	PTP时间戳中断屏蔽	0：PTP时间戳中断不屏蔽 1：PTP时间戳中断屏蔽	R/W
b8~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	PMTIM	PMT中断屏蔽	0：PMT中断不屏蔽 1：PMT中断屏蔽	R/W
b2~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

39.6.1.7 ETH_MAC SMI 地址寄存器（ETH_MAC_SMIADDR）

复位值: 0x00000000h

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
SMIA[4:0]					SMIR[4:0]					SMIC[3:0]				SMIW	SMIB

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~b11	SMIA	物理层地址	指示要访问32个可能的PHY器件中的哪一个	R/W
b10~b6	SMIR	PHY寄存器	指示在所选PHY器件中要选择寄存器	R/W
b5~b2	SMIC	MDC时钟	基于系统频率（PCLK1）决定MDC时钟频率： 选项 系统频率 MDC时钟 0000 60-100MHz 系统频率/42 0001 100-120MHz 系统频率/62 0010 20-35MHz 系统频率/16 0011 35-60MHz 系统频率/26 请不要设定其它值	R/W
b1	SMIW	SMI读写	0：SMI读操作 1：SMI写操作	R/W
b0	SMIB	SMI访问忙	向ETH_MAC_SMIADDR和ETH_MAC_SMIDATR写入前，此位应读取逻辑0 向ETH_MAC_SMIADDR写入过程中，此位也必须复位为0 在PHY寄存器访问过程中，此位由应用程序设为1，指示读或写访问正在进行中 在对PHY进行写操作过程中，ETH_MAC_SMIDATR应始终有效，直到MAC将此位清零；在对PHY进行读操作过程中，ETH_MAC_SMIDATR始终无效，直到MAC将此位清零。此位清零后，才可以向ETH_MAC_SMIADDR写入值	R/W

39.6.1.8 ETH_MAC SMI 数据寄存器（ETH_MAC_SMIDATR）

复位值: 0x00000000h

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
SMID[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	SMID	SMI数据	站点管理读操作之后从PHY中读取的16位数据值，或站点管理写操作之前要写入PHY的16位数据值	R/W

39.6.1.9 ETH_MAC PMT 控制状态寄存器（ETH_MAC_PMTCTLR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RTWKFR	-	-	-	-	RTWKPT[2:0]			Reserved							

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					RTWKTR	GLUB	-	-	WKFR	MPFR	-	-	WKEN	MPEN	PWDN

位	标记	位名	功能	读写
b31	RTWKFR	远程唤醒帧过滤寄存器指针复位	0：远程唤醒帧过滤寄存器指针不复位 1：远程唤醒帧过滤寄存器指针复位 *注：复位完成后该位自动清零*	R/W
b30~b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26~b24	RTWKPT	远程唤醒帧过滤寄存器指针位置	该位表明远程唤醒帧过滤寄存器指针在第几个寄存器上（该值为0~7）	R
b23~b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10	RTWKTR	远程唤醒帧传送	该位置1后，MAC不再接收除魔术数据包和远程唤醒帧以外的包，并将接收到的远程唤醒帧转发给应用 *注1：该位在接收到魔术数据包或远程唤醒帧之后会自动变为0；也可以在接收到魔术数据包或远程唤醒帧之前软件复位* *注2：当该寄存器的GLBU位或WKEN位或MPEN位为1、同时PDWN位为0时，该位必须设定为1* *注3：在PWDN位为1时,该位设定无效*	R/W
b9	GLBU	全局单播	0：将MAC（DAF）地址过滤的任意单播帧视为无效 1：将MAC（DAF）地址过滤的任意单播帧视为唤醒帧	R/W
b8~b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6	WKFR	接收远程唤醒帧	0：未接收到远程唤醒帧 1：接收到远程唤醒帧 *注：该位在读过之后自动清零*	R
b5	MPFR	接收魔术数据包	0：未接收到魔术数据包 1：接收到魔术数据包 *注：该位在读过之后自动清零*	R
b4~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	WKEN	远程唤醒帧使能	0：不接收远程唤醒帧 1：接收远程唤醒帧，触发PMT事件	R/W
b1	MPEN	魔术数据包使能	0：不接收魔术数据包 1：接收魔术数据包，触发PMT事件	R/W
b0	PWDN	Power Down模式使能	0：Active模式 1：Power Down模式 所有接收到的帧都丢弃，只有接收到魔术数据包或远程唤醒帧时，此位会自动清零，同时退出Power Down模式；该位清零	R/W

	后收到的帧会转发给应用 *注：只有WKEN位或MPEN位置1时，才可以将此位置1*

39.6.1.10 ETH_MAC 远程唤醒帧过滤寄存器（ETH_MAC_RTWKFFR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
WKUPFRMFT[31:16]															

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

位	标记	位名	功能	读写
b31~b0	WKUPFRMFT[31:0]	远程唤醒帧过滤值	应用将通过此地址对远程唤醒帧过滤寄存器进行写/读操作。详细信息请参见“远程唤醒帧过滤”章节	R/W

39.6.1.11 ETH_MAC MAC 帧过滤控制寄存器（ETH_MAC_FLTCTLR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RA	Reserved									DNTU	IPFE	-	-	-	VTFE

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					HPF	SAF	SAIF	PCF[1:0]		DBF	PMF	DAIF	HMC	HUC	PR

位	标记	位名	功能	读写
b31	RA	接收所有	0：MAC接收器将通过地址过滤的接收帧传送给应用 1：MAC接收器将所有的接收帧传送到应用，不管它们是否已通过地址过滤	R/W
b30~b22	Reserved	-	读出时为“0”,写入时写“0”	R/W
b21	DNTU	丢弃无TCP/UDP报	0：MAC接收器将所有接收的IP报传送到应用，不管它们是否有TCP/UDP字段 1：MAC接收器丢弃所有无TCP或UDP字段的IP报	R/W
b20	IPFE	L3&L4过滤使能	0：MAC接收器将所有接收的IP报传送到应用，不管它们是否已通过L3&L4过滤 1：MAC接收器丢弃所有与L3&L4设定的过滤值比较不匹配的IP报，只处理通过L3&L4过滤的IP报	R/W
b19~b17	Reserved	-	读出时为“0”,写入时写“0”	R/W
b16	VTFE	VLAN标记过滤使能	0：MAC接收器将所有接收的VLAN帧传送到应用，不管它们是否已通过VLAN标记过滤 1：MAC接收器丢弃与VLAN标记设定的过滤值比较不匹配的VLAN帧	R/W
b15~b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10	HPF	Hash过滤或完美过滤选择	0：若该寄存器的HUC位或HMC位置1，则地址过滤器只传送与Hash过滤器匹配的帧 1：若该寄存器的HUC位或HMC位置1，则地址过滤器传送与完美过滤器或Hash过滤器匹配的帧	R/W
b9	SAF	源地址过滤	0：MAC接收器将所有接收的帧传送到应用，不管它们是否已通过源地址过滤 1：MAC接收器丢弃与源地址设定的过滤值比较不匹配的帧	R/W
b8	SAIF	源地址反转过滤	0：MAC接收器接收到的帧SA字段与设定的过滤值不匹配时，认定为过滤失败 1：MAC接收器接收到的帧SA字段与设定的过滤值匹配时，认定为过滤失败	R/W
b7~b6	PCF	传送控制帧	对控制帧（包括单播暂停帧和多播暂停帧）的转发做如下操作： 00：MAC阻止所有控制帧到达应用 01：MAC将除了暂停控制帧以外的所有控制帧转发到应用，	R/W

			即使这些帧的地址过滤失败 10：MAC将所有控制帧转发到应用，即使这些帧的地址过滤失败 11：MAC转发通过地址过滤的控制帧 *关于暂停控制帧的几点说明：* *注1：在全双工模式时，暂停控制帧的处理由流控寄存器（ETH_MAC_FLOCTLR）的RFE位决定* *注2：在流控寄存器（ETH_MAC_FLOCTLR）的UNP位置1时，接收到的帧的目标地址和MAC地址寄存器0设定的值匹配时，该帧被认为是单播暂停帧* *注3：接收到的帧的TYPE段值为0x8808或OPCODE段值为0x0001时，该帧被认为是暂停控制帧*	
b5	DBF	屏蔽广播帧	0：地址过滤器不对接收到的广播帧进行过滤 1：地址过滤器对所有传入的广播帧进行过滤 *注：该位置1时，其优先级高于其它过滤设定*	R/W
b4	PMF	传送所有多播帧	0：对接收到的带多播目标地址（目标地址字段的第一位是1）的帧进行地址过滤，过滤方式由该寄存器的HMC位决定 1：对接收到的带多播目标地址（目标地址字段的第一位是1）的帧不进行地址过滤	R/W
b3	DAIF	目标地址反转过滤	0：MAC接收器接收到的单播帧或多播帧的目标地址与设定的过滤值不匹配时，认定为过滤失败 1：MAC接收器接收到的单播帧或多播帧的目标地址与设定的过滤值匹配时，认定为过滤失败	R/W
b2	HMC	Hash多播控制	0：MAC对多播帧执行完美目标地址过滤，即将DA字段与设定的过滤值进行比较 1：MAC对多播帧执行Hash目标地址过滤	R/W
b1	HUC	Hash单播控制	0：MAC对单播帧执行完美目标地址过滤，即将DA字段与设定的过滤值进行比较 1：MAC对单播帧执行Hash目标地址过滤	R/W
b0	PR	混合模式	0：地址过滤器进行正常的地址过滤 1：地址过滤器不管目标或源地址，都传送所有传入的帧	R/W

39.6.1.12 ETH_MAC MAC 地址高位寄存器 0（ETH_MAC_MACADHR0）

复位值: 0x8000FFFFh

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
AE0	Reserved														

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRH0[15:0]															

位	标记	位名	功能	读写
b31	AE0	地址使能0	1：使用MAC地址0进行目标地址完美过滤 *注：该位始终为1*	R
b30~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b0	ASDDRH0	高位地址值0	MAC使用此字段过滤所接收的帧以及将MAC地址插入到发送流控制（暂停）帧中	R/W

39.6.1.13 ETH_MAC MAC 地址低位寄存器 0（ETH_MAC_MACADLR0）

复位值: 0xFFFFFFFFh

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ADDRL0[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRL0[15:0]															

位	标记	位名	功能	读写
b31~b0	ADDRL0	低位地址值0	MAC使用此字段过滤所接收的帧以及将MAC地址插入到发送流控制（暂停）帧中	R/W

39.6.1.14 ETH_MAC MAC 地址高位寄存器 1（ETH_MAC_MACADHR1）

复位值: 0x0000FFFFh

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
AE1	SA1	MBC1[5:0]						Reserved							

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRH1[15:0]															

位	标记	位名	功能	读写
b31	AE1	地址使能1	0：忽略MAC地址1的地址过滤 1：使用MAC地址1进行完美过滤	R/W
b30	SA1	地址选择1	0：用MAC地址1对接收帧的DA进行过滤 1：用MAC地址1对接收帧的SA进行过滤	R/W
b29~b24	MBC1	字节屏蔽1	地址过滤时屏蔽对应的字节 b29： 屏蔽ADDRH1[15:8]位 b28： 屏蔽ADDRH1[7:0]位 b27： 屏蔽ADDRL1[31:24]位 b26： 屏蔽ADDRL1[23:16]位 b25： 屏蔽ADDRL1[15:8]位 b24： 屏蔽ADDRL1[7:0]位	R/W
b23~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b0	ASDDRH1	高位地址值1	MAC使用此字段过滤所接收的帧	R/W

39.6.1.15 ETH_MAC MAC 地址低位寄存器 1（ETH_MAC_MACADLR1）

复位值: 0xFFFFFFFFh

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ADDRL1[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRL1[15:0]															

位	标记	位名	功能	读写
b31~b0	ADDRL1	低位地址值1	MAC使用此字段过滤所接收的帧	R/W

39.6.1.16 ETH_MAC MAC 地址高位寄存器 2（ETH_MAC_MACADHR2）

复位值: 0x0000FFFFh

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
AE2	SA2	MBC2[5:0]						Reserved							

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRH2[15:0]															

位	标记	位名	功能	读写
b31	AE2	地址使能2	0：忽略MAC地址2的地址过滤 1：使用MAC地址2进行完美过滤	R/W
b30	SA2	地址选择2	0：用MAC地址2对接收帧的DA进行过滤 1：用MAC地址2对接收帧的SA进行过滤	R/W
b29~b24	MBC2	字节屏蔽2	地址过滤时屏蔽对应的字节 b29： 屏蔽ADDRH2[15:8]位 b28： 屏蔽ADDRH2[7:0]位 b27： 屏蔽ADDRL2[31:24]位 b26： 屏蔽ADDRL2[23:16]位 b25： 屏蔽ADDRL2[15:8]位 b24： 屏蔽ADDRL2[7:0]位	R/W
b23~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b0	ASDDRH2	高位地址值2	MAC使用此字段过滤所接收的帧	R/W

39.6.1.17 ETH_MAC MAC 地址低位寄存器 2（ETH_MAC_MACADLR2）

复位值: 0xFFFFFFFFh

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ADDRL2[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRL2[15:0]															

位	标记	位名	功能	读写
b31~b0	ADDRL2	低位地址值2	MAC使用此字段过滤所接收的帧	R/W

39.6.1.18 ETH_MAC MAC 地址高位寄存器 3（ETH_MAC_MACADHR3）

复位值: 0x0000FFFFh

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
AE3	SA3	MBC3[5:0]						Reserved							

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRH3[15:0]															

位	标记	位名	功能	读写
b31	AE3	地址使能3	0：忽略MAC地址3的地址过滤 1：使用MAC地址3进行完美过滤	R/W
b30	SA3	地址选择3	0：用MAC地址3对接收帧的DA进行过滤 1：用MAC地址3对接收帧的SA进行过滤	R/W
b29~b24	MBC3	字节屏蔽3	地址过滤时屏蔽对应的字节 b29： 屏蔽ADDRH3[15:8]位 b28： 屏蔽ADDRH3[7:0]位 b27： 屏蔽ADDRL3[31:24]位 b26： 屏蔽ADDRL3[23:16]位 b25： 屏蔽ADDRL3[15:8]位 b24： 屏蔽ADDRL3[7:0]位	R/W
b23~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b0	ASDDRH3	高位地址值3	MAC使用此字段过滤所接收的帧	R/W

39.6.1.19 ETH_MAC MAC 地址低位寄存器 3（ETH_MAC_MACADLR3）

复位值: 0xFFFFFFFFh

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ADDRL3[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRL3[15:0]															

位	标记	位名	功能	读写
b31~b0	ADDRL3	低位地址值3	MAC使用此字段过滤所接收的帧	R/W

39.6.1.20 ETH_MAC MAC 地址高位寄存器 4（ETH_MAC_MACADHR4）

复位值: 0x0000FFFFh

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
AE4	SA4	MBC4[5:0]						Reserved							

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRH4[15:0]															

位	标记	位名	功能	读写
b31	AE4	地址使能4	0：忽略MAC地址4的地址过滤 1：使用MAC地址4进行完美过滤	R/W
b30	SA4	地址选择4	0：用MAC地址4对接收帧的DA进行过滤 1：用MAC地址4对接收帧的SA进行过滤	R/W
b29~b24	MBC4	字节屏蔽4	地址过滤时屏蔽对应的字节 b29： 屏蔽ADDRH4[15:8]位 b28： 屏蔽ADDRH4[7:0]位 b27： 屏蔽ADDRL4[31:24]位 b26： 屏蔽ADDRL4[23:16]位 b25： 屏蔽ADDRL4[15:8]位 b24： 屏蔽ADDRL4[7:0]位	R/W
b23~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b0	ASDDRH4	高位地址值4	MAC使用此字段过滤所接收的帧	R/W

39.6.1.21 ETH_MAC MAC 地址低位寄存器 4（ETH_MAC_MACADLR4）

复位值: 0xFFFFFFFFh

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ADDRL4[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRL4[15:0]															

位	标记	位名	功能	读写
b31~b0	ADDRL4	低位地址值4	MAC使用此字段过滤所接收的帧	R/W

39.6.1.22 ETH_MAC MAC Hash 表高位寄存器（ETH_MAC_HASHTHR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
HTH[31:16]															

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

位	标记	位名	功能	读写
b31~b0	HTH	Hash Table高位值	Hash Table的高32位值	R/W

39.6.1.23 ETH_MAC MAC Hash 表低位寄存器（ETH_MAC_HASHTLR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
HTL[31:16]															

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

位	标记	位名	功能	读写
b31~b0	HTL	Hash Table低位值	Hash Table的低32位值	R/W

39.6.1.24 ETH_MAC VLAN 标记发送控制寄存器（ETH_MAC_VTACTLR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved													VLANS	VLANC[1:0]	

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
VLANV[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/W
b18	VLANS	VLAN标记选择	0：在发送VLAN帧时，对标记字段的处理由Tx描述符的TDES0的VLANC控制位设定决定 1：在发送VLAN帧时，对标记字段的处理由该寄存器的VLANC控制位设定决定	R/W
b17~b16	VLANC	VLAN标记控制	在发送VLAN帧时，对标记字段的处理： 00：不做操作，直接发送 01：将VLAN帧的TYPE字段和标记字段删除 10：在插入TYPE字段之后把该寄存器的VLANV位的值插入作为标记字段，不管原帧中是否含有标记字段 11：用该寄存器的VLANV位的值替换原帧中的标记字段	R/W
b15~b0	VLANV	VLAN标记值	用户定义VLAN标记字段值，用于在发送帧时插入或替换标记字段	R/W

39.6.1.25 ETH_MAC VLAN 标记接收过滤寄存器（ETH_MAC_VTAFLTR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved												VTHM	-	VTIM	VTAL

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

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	VTHM	VLAN标记Hash表使能	0：VLAN标记Hash过滤功能无效 1：VLAN标记Hash过滤功能使能	R/W
b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17	VTIM	VLAN标记反转过滤	0：VLAN帧的标记字段与该寄存器VLFLT位设定的过滤值不匹配时，认定为过滤失败 1：VLAN帧的标记字段与该寄存器VLFLT位设定的过滤值匹配时，认定为过滤失败	R/W
b16	VTAL	12BIT VLAN比较	0：VLAN标记字段的全16位与VLFLT设定值比较 1：VLAN标记字段的低12位与VLFLT设定值比较 *注：该位的设定也适用于Hash过滤方式*	R/W
b15~b0	VLFLT	VLAN标记过滤值	此字段包含用于过滤VLAN帧的标记字段的比较值，该位与接收的VLAN帧的第15和第16字节进行比较	R/W

39.6.1.26 ETH_MAC VLAN Hash 表寄存器（ETH_MAC_VLAHTBR）

复位值: 0x00000000h

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
VLHT[15:0]															

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为复位值	R/W
b15~b0	VLHT	VLAN Hash Table	VLAN Hash Table值	R/W

39.6.1.27 ETH_MAC LAY3LAY4 控制寄存器（ETH_MAC_L34CTLR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved										L4 DPIM	L4 DPM	L4 SPIM	L4 SPM	-	L4 PEN

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
L3 HDBM[4:0]					L3 HSBM[4:0]					L3 DAIM	L3 DAM	L3 SAIM	L3 SAM	-	L3 PEN

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	L4DPIM	L4目标端口反转过滤	0：L4的目标端口字段与设定的过滤值不匹配时，认定为过滤失败 1：L4的目标端口字段与设定的过滤值匹配时，认定为过滤失败 *注：当该寄存器的L4DPM位有效时，该位设定才有效*	R/W
b20	L4DPM	L4目标端口过滤	0：忽略L4的目标端口字段的过滤 1：对L4的目标端口字段进行过滤	R/W
b19	L4SPIM	L4源端口反转过滤	0：L4的源端口字段与设定的过滤值不匹配时，认定为过滤失败 1：L4的源端口字段与设定的过滤值匹配时，认定为过滤失败 *注：当该寄存器的L4SPM位有效时，该位设定才有效*	R/W
b18	L4SPM	L4源端口过滤	0：忽略L4的源端口字段的过滤 1：对L4的源端口字段进行过滤	R/W
b17	Reserved	-	读出时为“0”,写入时写“0”	R/W
b16	L4PEN	L4协议使能	0：对L4的TCP协议帧进行端口过滤 1：对L4的UDP协议帧进行端口过滤	R/W
b15~b11	L3HDBM	L3IP目标地址高位屏蔽	对L3目标地址过滤时的位屏蔽设定： IPv4时： 00000：全目标地址过滤 00001：忽略目标地址的最低位LSB[0]的过滤 00010：忽略目标地址的最低位LSB[1:0]的过滤 00011：忽略目标地址的最低位LSB[2:0]的过滤 …… 11111：忽略目标地址的最低位LSB[30:0]的过滤 IPv6时： L3HDBM的Bit2~1作为L3HSBM的Bit7~6 *注：该寄存器的L3DAM有效时，该位才有效*	R/W
b10~b6	L3HSBM	L3IP源地址高位屏蔽	对源地址过滤时的位屏蔽设定： IPv4时： 00000：全源地址过滤 00001：忽略源地址的最低位LSB[0]的过滤	R/W

			00010：忽略源地址的最低位LSB[1:0]的过滤 00011：忽略源地址的最低位LSB[2:0]的过滤 …… 11111：忽略源地址的最低位LSB[30:0]的过滤 IPv6时： 0000000：全源或目标地址过滤 0000001：忽略源或目标地址的最低位LSB[0]的过滤 0000010：忽略源或目标地址的最低位LSB[1:0]的过滤 0000011：忽略源或目标地址的最低位LSB[2:0]的过滤 …… 1111111：忽略源或目标地址的最低位LSB[126:0]的过滤 *注1：该寄存器的L3SAM或L3DAM有效时，该位才有效* *注2：选择IPv6功能，且L3DAM和L3SAM均使能时，源地址和目标地址的过滤控制共用一组过滤地址，且该高位屏蔽控制位也共用*	
b5	L3DAIM	L3目标地址反转过滤	0：L3的IP报的目标地址字段与设定的过滤值不匹配时，认定为过滤失败 1：L3的IP报的目标地址字段与设定的过滤值匹配时，认定为过滤失败 *注：当该寄存器的L3DAM位有效时，该位设定才有效*	R/W
b4	L3DAM	L3目标地址过滤	0：忽略L3的IP报的目标地址字段的过滤 1：对L3的IP报的目标地址字段进行过滤	R/W
b3	L3SAIM	L3源地址反转过滤	0：L3的IP报的源地址字段与设定的过滤值不匹配时，认定为过滤失败 1：L3的IP报的源地址字段与设定的过滤值匹配时，认定为过滤失败 *注：当该寄存器的L3SAM位有效时，该位设定才有效*	R/W
b2	L3SAM	L3源地址过滤	0：忽略L3的IP报的源地址字段的过滤 1：对L3的IP报的源地址字段进行过滤	R/W
b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	L3PEN	L3协议使能	0：对L3的IPv4报进行地址过滤 1：对L3的IPv6报进行地址过滤	R/W

39.6.1.28 ETH_MAC LAY4 端口寄存器 （ETH_MAC_L4PORTR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
L4DPVAL[15:0]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
L4SPVAL[15:0]															

位	标记	位名	功能	读写
b31~b16	L4DPVAL	目标端口过滤值	L4的目标端口过滤值	R/W
b15~b0	L4SPVAL	源端口过滤值	L4的源端口过滤值	R/W

39.6.1.29 ETH_MAC LAY3 地址寄存器 0（ETH_MAC_L3ADDRR0）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
L3ADDR0[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
L3ADDR0[15:0]															

位	标记	位名	功能	读写
b31~b0	L3ADDR0	地址过滤值0	L3的地址过滤值： 1）L3PEN=0&L3SAM=1时，该寄存器为IPv4报的L3的源地址过滤值 2）L3PEN=1&L3SAM=1时，该寄存器为IPv6报的L3的源地址[31:0]过滤值 3）L3PEN=1&L3DAM=1时，该寄存器为IPv6报的L3的目标地址[31:0]位过滤值	R/W

39.6.1.30 ETH_MAC LAY3 地址寄存器 1（ETH_MAC_L3ADDRR1）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
L3ADDR1[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
L3ADDR1[15:0]															

位	标记	位名	功能	读写
b31~b0	L3ADDR1	地址过滤值1	L3的地址过滤值： 1）L3PEN=0&L3DAM=1时，该寄存器为IPv4报的L3的目标地址过滤值 2）L3PEN=1&L3SAM=1时，该寄存器为IPv6报的L3的源地址[63:32]过滤值 3）L3PEN=1&L3DAM=1时，该寄存器为IPv6报的L3的目标地址[63:32]位过滤值	R/W

39.6.1.31 ETH_MAC LAY3 地址寄存器 2（ETH_MAC_L3ADDRR2）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
L3ADDR2[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
L3ADDR2[15:0]															

位	标记	位名	功能	读写
b31~b0	L3ADDR2	地址过滤值2	L3的地址过滤值： 1）L3PEN=1&L3SAM=1时，该寄存器为IPv6报的L3的源地址[95:64]过滤值 2）L3PEN=1&L3DAM=1时，该寄存器为IPv6报的L3的目标地址[95:64]位过滤值	R/W

39.6.1.32 ETH_MAC LAY3 地址寄存器 3（ETH_MAC_L3ADDRR3）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
L3ADDR3[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
L3ADDR3[15:0]															

位	标记	位名	功能	读写
b31~b0	L3ADDR3	地址过滤值3	L3的地址过滤值： 1）L3PEN=1&L3SAM=1时，该寄存器为IPv6报的L3的源地址[127:96]过滤值 2）L3PEN=1&L3DAM=1时，该寄存器为IPv6报的L3的目标地址[127:96]位过滤值	R/W

39.6.2 ETH_PTP 寄存器

39.6.2.1 ETH_PTP 时间戳控制寄存器（ETH_PTP_TSPCTLR）

复位值: 0x00002000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved													TSPADF	TSPMTSEL[3:2]	

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TSPMTSEL[1:0]		TSPOVIPV4	TSPOVIPV6	TSPOVETH	TSPVER	TSPSSR	TSPEALL	-	-	TSPADUP	TSPINT	TSPUP	TSPINI	TSPUPSEL	TSPEN

位	标记	位名	功能	读写
b31~b19	Reserved	-	读出时为“0”,写入时写“0”	R/W
b18	TSPADF	PTP帧过滤	PTP帧直接通过以太网发送时，若该位置1，则对该PTP帧进行目标地址过滤	R/W
b17~b14	TSPMTSEL	PTP报文类型选择	选择要拍摄时间戳快照的报文类型，具体请参见“PTP报文类型”章节	R/W
b13	TSPOVIPV4	接收Over UDP-IPv4封装类型的帧	0：忽略Over UDP-IPv4封装形式的PTP帧 1：接收Over UDP-IPv4封装形式的PTP帧	R/W
b12	TSPOVIPV6	接收Over UDP-IPv6封装类型的帧	0：忽略Over UDP-IPv6封装形式的PTP帧 1：接收Over UDP-IPv6封装形式的PTP帧	R/W
b11	TSPOVETH	接收Over Ethernet封装类型的帧	0：忽略Over Ethernet封装形式的PTP帧 1：接收Over Ethernet封装形式的PTP帧	R/W
b10	TSPVER	PTP报文处理版本	0：用IEEE1588v1版本格式对接收到的PTP帧处理 1：用IEEE1588v2版本格式对接收到的PTP帧处理	R/W
b9	TSPSSR	亚秒计数进制	0：亚秒寄存器的值达到0x7FFFFFFFH，时间戳秒计数器增加1秒，亚秒寄存器归零重新计数 1：亚秒寄存器的值达到0x3B9AC9FFH，时间戳秒寄存器增加1秒，亚秒寄存器归零重新计数 *注1：0x 3B9AC9FF=0d 999999999* *注2：亚秒增量必须根据PTP参考时钟频率和此位的值正确进行编程*	R/W
b8	TSPEALL	使能所有帧	该位置1时，对接收的所有帧都拍摄时间戳快照	R/W
b7~b6	Reserved	-	读出时为“0”,写入时写“0”	R/W
b5	TSPADUP	计数寄存器更新	该位置1时，时间戳加数寄存器（ETH_PTP_TSPADDR）的值会被更新到PTP进行精密校准，更新结束时此位会自动清零 *注：该寄存器位必须在读为0后，才可以将其置1*	R/W
b4	TSPINT	时间戳中断使能	如果系统时间大于在目标时间寄存器中写入的值，将产生时戳中断 *注：产生时间戳触发中断时，此位会清零。*	R/W

b3	TSPUP	时间戳更新	该位置1时，系统时间将通过在时间戳更新秒寄存器（ETH_PTP_TMUSECR）和时间戳更新亚秒寄存器（ETH_PTP_TMUNSER）中指定的值进行更新（加入/减去）。更新结束时此位会自动清零 *注：该寄存器位必须在自己和TSPINI两位均读为0后，才可以将其置1*	R/W
b2	TSPINI	时间戳初始化	该位置1时，系统时间将通过在时间戳更新秒寄存器（ETH_PTP_TMUSECR）和时间戳更新亚秒寄存器（ETH_PTP_TMUNSER）中指定的值进行初始化（覆盖）。初始化结束时此位会自动清零 *注：该寄存器位必须在读为0后，才可以将其置1*	R/W
b1	TSPUPSEL	时间戳校准方式	0：粗校准方式 1：细校准方式	R/W
b0	TSPEN	PTP功能使能	0：时间戳功能无效，MAC不会为发送和接收帧添加时间戳 1：时间戳功能有效，针对发送和接收帧添加时间戳 *注：该位一旦置1，需要对系统时间进行初始化*	R/W

39.6.2.2 ETH_PTP 时间戳状态寄存器（ETH_PTP_TSPSTSR）

复位值: 0x00000000h

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										TSERR1	TSTAR1	TSERR0	-	TSTAR0	TSOVF

位	标记	位名	功能	读写
b31~b6	Reserved	-	读出时为“0”,写入时写“0”	R
b5	TSERR1	目标时间1错误	0：要写入目标时间寄存器1的时间正确 1：要写入目标时间寄存器1的时间已经过去，该写入值错误 *注：该位在读出后自动清零*	R
b4	TSTAR1	目标时间1达到	0：系统时间未达到目标时间寄存器1指示的时间 1：系统时间达到目标时间寄存器1指示的时间	R
b3	TSERR0	目标时间0错误	0：要写入目标时间寄存器0的时间正确 1：要写入目标时间寄存器0的时间已经过去，该写入值错误 *注：该位在读出后自动清零*	R
b2	Reserved	-	读出时为“0”,写入时写“0”	R
b1	TSTAR0	目标时间0达到	0：系统时间未达到目标时间寄存器0指示的时间 1：系统时间达到目标时间寄存器0指示的时间	R
b0	TSOVF	系统时间溢出	0：系统时间未溢出 1：系统时间溢出	R

39.6.2.3 ETH_PTP 时间戳基本加数寄存器（ETH_PTP_TSPADDR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TSPADD[31:16]															

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

位	标记	位名	功能	读写
b31~b0	TSPADD	基本加数值	该寄存器由软件来重新线性调节时钟频率，使其与主时钟频率匹配。在每个时钟周期将该寄存器的值添加到32位累加器中 *注：该寄存器只在细校准模式下有效。*	R/W

39.6.2.4 ETH_PTP 时间戳亚秒加数寄存器（ETH_PTP_TSPNSAR）

复位值: 0x00000000h

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								TSPNSEADD[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	TSPNSEADD	亚秒加数值	该寄存器包含系统亚秒寄存器递增时使用的8位值 在粗校准模式下，每个时钟周期都将该寄存器的值添加到系统时间；在细校准模式下，只在32位累加器溢出时将该寄存器的值添加到系统时间	R/W

39.6.2.5 ETH_PTP 时间戳系统秒寄存器（ETH_PTP_TMSSECR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TSPSYSSEC[31:16]															

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

位	标记	位名	功能	读写
b31~b0	TSPSYSSEC	系统秒时间	该位表示系统时间的秒时间	R

39.6.2.6 ETH_PTP 时间戳系统亚秒寄存器（ETH_PTP_TMSNSER）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	TSPSYSNSEC[30:16]														

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

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“0”,写入时写“0”	R/W
b30~b0	TSPSYSNSEC	系统亚秒时间	该位表示系统时间的亚秒时间	R

39.6.2.7 ETH_PTP 时间戳更新秒寄存器（ETH_PTP_TMUSECR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TSPUPSEC[31:16]															

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

位	标记	位名	功能	读写
b31~b0	TSPUPSEC	系统秒更新值	该位表示要更新进系统时间的秒时间	R/W

39.6.2.8 ETH_PTP 时间戳更新亚秒寄存器（ETH_PTP_TMUNSER）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TSPUPNS	TSPUPNSEC[30:16]														

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

位	标记	位名	功能	读写
b31	TSPUPNS	更新方式	时间更新正负号： 0：从系统时间寄存器的值中减去更新寄存器的值 1：在系统时间寄存器的值中加上更新寄存器的值	R/W
b30~b0	TSPUPNSE	系统亚秒更新值	该位表示要更新进系统时间的亚秒时间	R/W

39.6.2.9 ETH_PTP 时间戳目标秒寄存器 0（ETH_PTP_TMTSECR0）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TSPTAGSEC0[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TSPTAGSEC0[15:0]															

位	标记	位名	功能	读写
b31~b0	TSPTAGSEC0	目标秒时间0	系统时间超过或等于该寄存器和目标亚秒寄存器0的设定值时， 发生中断事件或PPS事件	R/W

39.6.2.10 ETH_PTP 时间戳目标亚秒寄存器 0（ETH_PTP_TMTNSER0）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	TSPTAGNSEC0[30:16]														

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TSPTAGNSEC0[15:0]															

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“0”,写入时写“0”	R/W
b30~b0	TSPTAGNSEC0	目标亚秒时间0	系统时间超过或等于该寄存器和目标秒寄存器0的设定值时，发生中断事件或PPS事件	R/W

39.6.2.11 ETH_PTP 时间戳目标秒寄存器 1（ETH_PTP_TMTSECR1）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TSPTAGSEC1[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TSPTAGSEC1[15:0]															

位	标记	位名	功能	读写
b31~b0	TSPTAGSEC1	目标秒时间1	系统时间超过或等于该寄存器和目标亚秒寄存器1的设定值时，发生中断事件或PPS事件	R/W

39.6.2.12 ETH_PTP 时间戳目标亚秒寄存器 1（ETH_PTP_TMTNSER1）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	TSPTAGNSEC1[30:16]														

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TSPTAGNSEC1[15:0]															

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“0”,写入时写“0”	R/W
b30~b0	TSPTAGNSEC1	目标亚秒时间1	系统时间超过或等于该寄存器和目标秒寄存器1的设定值时，发生中断或PPS事件	R/W

39.6.2.13 ETH_PTP PPS 输出控制寄存器（ETH_PTP_PPSCTLR）

复位值: 0x00000000h

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
-	TT1SEL[1:0]		-	-	PPSFRE1[2:0]			-	TT0SEL[1:0]		PPSOMD	PPSFRE0[3:0]			

位	标记	位名	功能	读写
b31~b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14~b13	TT1SEL	目标时间1功能选择	00：目标寄存器1仅用于中断输出事件 01：无意义，请不要设定 10：目标寄存器1用于中断输出事件和PPS1单次输出事件 11：目标寄存器1仅用于PPS1单次输出事件	R/W
b12~b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10~b8	PPSFRE1	PPS1输出频率	001：当系统时间等于目标秒寄存器1时，产生一次脉冲，之后不再产生脉冲 请不要设定其它值 *注：该位读出时为0*	R/W
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6~b5	TT0SEL	目标时间0功能选择	00：目标寄存器0仅用于中断输出事件 01：无意义，请不要设定 10：目标寄存器0用于中断输出事件和PPS0单次输出事件 11：目标寄存器0仅用于PPS0单次输出事件	R/W
b4	PPSOMD	PPS输出模式	0：连续输出模式 1：单次输出模式 *注：PPS0支持上述两种模式，PPS1仅支持单次输出模式*	R/W
b3~b0	PPSFRE0	PPS0输出频率	连续输出模式时： 当系统时间发生秒进位（亚秒时间溢出）时，PPS0根据下面的设定产生不同形式的输出： 0000：连续输出间隔为1Hz的脉冲 0001：连续输出频率为2Hz的时钟 0010：连续输出频率为4Hz的时钟 0011：连续输出频率为8Hz的时钟 0100：连续输出频率为16Hz的时钟 ..... 1111：连续输出频率为32768Hz的时钟 单次输出模式时： 0001：当系统时间等于目标秒寄存器0时，产生一次脉冲，之后不再产生脉冲 请不要设定其它值	R/W

	注：1）连续输出模式时，时间戳控制寄存器的TSPSSR位须设定为0 *2）单次输出模式时，该位读出时为0*

39.6.3 ETH_DMA 寄存器

39.6.3.1 ETH_DMA 总线模式寄存器（ETH_DMA_BUSMODR）

复位值: 0x00020101h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved				TXPR	MBST	AAL	M8PBL	SPBL	RPBL[5:0]						FBST

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PRAT[1:0]		TPBL[5:0]						DSEN	DSL[4:0]					DMAA	SWR

位	标记	位名	功能	读写
b31~b28	Reserved	-	读出时为“0”,写入时写“0”	R/W
b27	TXPR	传送优先级	0：RxDMA事务优先级高于TxDMA事务 1：TxDMA事务优先级高于RxDMA事务	R/W
b26	MBST	混合突发传送	该位置1且FBST位为0时，AHB Master口将以INCR启动所有长度大于16的突发传输，以SINGLE和INCRx启动长度等于或小于16的突发传输(x=4/8/16)	R/W
b25	AAL	地址对齐	该位置1时: 1）若FBST=1时，AHB接口会生成与起始地址LS位对齐的所有突发 2）若FBST=0时，则第一个突发（访问数据缓冲器的起始地址）不对齐，但后续的突发与地址对齐	R/W
b24	M8PBL	8倍PBL	该位置1时，会将编程的PBL值（RPBL位和TPBL位）乘以8倍，因此，DMA根据PBL值以最大8、16、32、64、128和256个节拍传输数据	R/W
b23	SPBL	独立PBL控制	0：该寄存器的TPBL设定用于TxDMA事务和RxDMA事务 1：该寄存器的TPBL设定只用于TxDMA事务，RPBL设定用于RxDMA事务	R/W
b22~b17	RPBL	Rx可编程突发长度	和该寄存器的TPBL位的意义相同，仅用于RxDMA事务 *注：该位只在该寄存器的SPBL位有效时才有用*	R/W
b16	FBST	固定突发传送	0：DMA Master接口只使用SINGLE和INCR访问类型 1：DMA Master接口使用SINGLE和INCR4、INCR8、INCR16访问类型	R/W
b15~b14	PRAT	优先级比	Rx和Tx对BUS的抢占优先级比 00： 1：1 11： 2：1 10： 3：1 11： 4：1 *注1：该位只在该寄存器的DMAA位为0时有效* *注2：Tx：Rx或Rx：Tx的选择由该寄存器的TXPR位决定*	R/W

b13~b8	TPBL	可编程突发长度	该位指示要在一个TxDMA事务或RxDMA事务中传输的最大节拍数，这是在单个块读/写操作中使用的最大值。DMA每次在主机总线上开始突发传输时，始终尝试按PBL中指定的方式进行突发。允许使用值 1、2、4、8、16和32对PBL进行编程，任何其它值都会产生未定义的行为 *注：PBL值有以下限制：* *1） 可能的最大节拍数(PBL)受Tx FIFO和Rx FIFO大小的限制* *2） 请不要编程超出范围的PBL值*	R/W
b7	DSEN	增强型描述符	增强型描述符格式使能 0：常规型描述符 1：增强型描述符 *注：在PTP功能有效（ETH_PTP_TSPCTLR.TSPEN=1）或校验和减荷功能有效（ETH_MAC_CONFIGR.IPCO=1）时，必须使用增强型描述符*	R/W
b6~b2	DSL	描述符跳跃长度	该位指定两个未链接描述符之间跳过的Word数，地址从当前描述符结束处开始跳到下一个描述符起始处。当DSL值等于零时，在环形模式下，DMA会将描述符表视为连续的	R/W
b1	DMAA	DMA仲裁	0：循环优先级，仲裁方式由该寄存器的PRAT位和TXPR位决定 1：固定优先级，优先方式由该寄存器的TXPR位决定	R/W
b0	SWR	软件复位	该位置1时，DMA控制器复位所有的MAC子系统的内部寄存器和逻辑。在所有内核时钟域完成复位操作后，该位自动清零	R/W

39.6.3.2 ETH_DMA 操作模式寄存器（ETH_DMA_OPRMODR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserevd					DTCOE	RSF	DFRF	-	-	TSF	FTF	-	-	-	TTC[2]

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TTC[1:0]		STT	Reserevd					FEF	FUF	DGF	RTC[1:0]		OSF	STR	-

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	DTCOE	禁止丢弃TCP/IP校验和错误帧	0：如果该寄存器的FEF位置位1，则丢弃所有错误帧 1：如果接收帧中仅存在由接收校验和减荷引擎检测出来的错误，则内核不会丢弃该帧 *注：这类帧在MAC接收到的以太网帧中没有任何错误（包括CRC错误），而仅在封装的有效负载中有错误*	R/W
b25	RSF	接收存储转发	0：Rx FIFO中的帧达到该寄存器的RTC位指定的值时，才进行Rx FIFO的读动作 1：Rx FIFO写入完整帧后，就执行读动作，同时忽略RTC位	R/W
b24	DFRF	禁止刷新接收帧	该位置1时，RxDMA不会因为接收描述符/缓冲区不可用而刷新任何帧，参见“接收过程挂起“章节	R/W
b23~b22	Reserved	-	读出时为“0”,写入时写“0”	R/W
b21	TSF	发送存储转发	0：Tx FIFO中的帧达到该寄存器的TTC位指定的值时，才进行Tx FIFO的读动作，即MAC发送才启动 1：Tx FIFO中有一个完整帧，则MAC发送会启动，同时忽略TTC位 *注：该位只有在已停止传输时才能更改*	
b20	FTF	刷新Tx FIFO	该位置1时，Tx FIFO复位为默认值，此时Tx FIFO中的所有数据均会丢失/刷新。刷新操作结束时该位自动清零 *注：此位清零之前不得对工作模式寄存器执行写操作*	R/W
b19~b17	Reserved	-	读出时为“0”,写入时写“0”	R/W
b16~b14	TTC	发送阈值	该位用于控制Tx FIFO的阈值级别。当Tx FIFO中的帧大小大于阈值时启动DMA发送。此外，长度小于阈值的完整帧会自动传输 000：64 001：128 010：192 011：256 100：40 101：32 110：24 111：16	R/W

			注：该位只有在TSF位为0时才使用	
b13	STT	发送启动停止	0：发送过程在完成发送当前帧的任务之后进入停止状态，并保存发送列表中的下一个描述符位置，该描述符位置在重启发送后会成为当前位置 1：启动发送过程，DMA检查发送列表中的当前位置，即发送描述符地址列表寄存器（ETH_DMA_TXDLADR）设定的地址，或者上一次停止发送时的保留位置尝试获取描述符，来查找待发送的帧 如果当前描述符不属于TxDMA，则发送过程会进入挂起状态；如果该命令在设置ETH_DMA_TXDLADR寄存器之前发出，则TxDMA行为无法预知 *注：启动发送命令只有在传输已停止时才有效；停止发送命令只有在当前帧发送过程结束或发送过程处于挂起状态时才有效*	R/W
b12~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	FEF	转发错误帧	0：Rx FIFO会丢弃带错误状态（CRC错误、冲突错误、巨帧、看门狗超时、上溢）的帧 如果某个帧的起始字节指针已传输到读控制器端（阈值模式下），则不会丢弃该帧；如果总线上未传输帧的起始字节，则Rx FIFO会丢弃此错误帧 1：除矮帧错误帧之外的所有帧都会转发	R/W
b6	FUF	转发过小好帧	0：Rx FIFO丢弃所有不足64字节的帧，除非因接收阈值下限更低（例如RTC=01）而导致此类帧已传输 1：Rx FIFO转发包含PAD字节和CRC的过小帧（无错误但长度不足64字节的帧）	R/W
b5	DGF	丢弃巨帧	0：Rx FIFO不丢弃巨帧 1：Rx FIFO丢弃巨帧	R/W
b4~b3	RTC	接收阈值	该位用于控制Rx FIFO的阈值级别。当Rx FIFO中的帧大小大于阈值时启动DMA传输（请求）。此外，长度小于阈值的完成帧会自动传输 00：64 01：32 10：96 11：128 *注：该位只有在RSF位为0时才使用*	R/W
b2	OSF	处理第二帧	该位置1时会命令DMA处理第二个发送数据帧，即使尚未获得首个帧的状态	R/W
b1	STR	接收启动停止	0：RxDMA会在传输当前帧之后停止操作，并保存接收列表中的下一个描述符位置，该描述符位置在接收过程重启后会成为当前位置 1：启动接收过程，DMA尝试从接收列表中的当前位置，即，接收描述符地址列表寄存器（ETH_DMA_RXDLADR）设定的地	R/W

			址，或者上一次停止接收时的保留位置尝试获取描述符，并处理传入帧 如果RxDMA未占据任何描述符，则接收过程会进入挂起状态； 如果该命令在设置ETH_DMA_RXDLADR寄存器之前发出，则RxDMA行为无法预知 *注：启动接收命令只有在已停止接收时才有效；停止接收命令只有在当前接收过程处于运行（等待接收数据包）或挂起状态时才有效。*	
b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

39.6.3.3 ETH_DMA 动作状态寄存器（ETH_DMA_DMASTSR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	PTPS	PMTS	MMCS	-	EBUS[2:0]			TSTS[2:0]			RSTS[2:0]			NIS

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AIS	ERS	FBS	-	-	ETS	RWS	RSS	RUS	RIS	UNS	OVS	TJS	TUS	TSS	TIS

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	PTPS	PTP事件状态	0：PTP模块无时间戳事件产生 1：PTP模块有时间戳事件发生	R
b28	PMTS	PMT事件状态	0：PMT模块无唤醒事件发生 1：PMT模块有唤醒事件发生	R
b27	MMCS	MMC事件状态	0：MMC模块无帧统计事件发生 1：MMC模块有帧统计事件发生	R
b26	Reserved	-	读出时为“0”,写入时写“0”	R/W
b25~b23	EBUS	错误位状态	该位指示导致总线错误（AHB接口上的错误响应）的错误类型，不会产生中断 000：RxDMA写数据时发生错误 011：TxDMA读数据时发生错误 100：RxDMA写描述符时发生错误 101：TxDMA写描述符时发生错误 110：RxDMA读描述符时发生错误 111：TxDMA读描述符时发生错误 请不要写入其它值 *注：该位仅在该寄存器的FBS位置1时有效*	R
b22~b20	TSTS	发送状态	该位指示TxDMA FSM的状态，不会产生中断 000：停止，发出复位或停止发送命令 001：运行中，正在获取发送传输描述符 010：运行中，正在等待状态 011：运行中，正在读取主机存储器缓冲区中的数据并将其加入发送缓冲区（Tx FIFO）队列 100：时间戳写入 101：Reserved 110：挂起，发送描述符不可用或发送缓冲区下溢 111：运行中，正在关闭发送描述符	R
b19~b17	RSTS	接收状态	该位指示RxDMA FSM的状态，不会产生中断。 000：停止，发出复位或停止接收命令 001：运行中，正在获取接收传输描述符 010：Reserved	R

			011：运行中，正在等待接收数据包 100：挂起，接收描述符不可用 101：运行中，正在关闭接收描述符 110：时间戳写入 111：运行中，将接收数据包的数据从接收缓冲区传输到主机存储器	
b16	NIS	普通中断状态汇总	下列任何一个条件满足，该位就被置1： 1）TIS=1 && ETH_DMA_INTENAR.TIE=1 2）TUS=1 && ETH_DMA_INTENAR.TUE=1 3）RIS=1 && ETH_DMA_INTENAR.RIE=1 4）ERS=1 && ETH_DMA_INTENAR.ERE=1 *注1：只有未屏蔽的中断使能位会影响该普通中断状态汇总位* *注2：每当导致该位置1的对应位被清零时，该位也必须清零（通过向此位写入1）*	R
b15	AIS	异常中断状态汇总	下列任何一个条件满足，该位就被置1： 1）TSS=1 && ETH_DMA_INTENAR.TSE=1 2）TJS=1 && ETH_DMA_INTENAR.TJE=1 3）OVS=1 && ETH_DMA_INTENAR.OVE=1 4）UNS=1 && ETH_DMA_INTENAR.UNE=1 5）RUS=1 && ETH_DMA_INTENAR.RUE=1 6）RSS=1 && ETH_DMA_INTENAR.RSE=1 7）RWS=1 && ETH_DMA_INTENAR.RWE=1 8）ETS=1 && ETH_DMA_INTENAR.ETE=1 9）FBS=1 && ETH_DMA_INTENAR.FBE=1 *注1：只有未屏蔽的中断使能位会影响该异常中断状态汇总位* *注2：每当导致该位置1的对应位被清零时，该位也必须清零（通过向此位写入1）*	R
b14	ERS	提前接收状态	该位指示DMA已填满数据包的首个数据缓冲区 *注：可通过向该位写入1来清零该位，也可在该寄存器的RIS为置位后自动清零*	R
b13	FBS	致命总线错误状态	该位指示发生了总线错误，具体错误类型参见该寄存器的EBUS位，当该位置1后，对应的DMA引擎会禁止其所有的总线访问 *注：可通过向该位写入1来清零该位*	R
b12~b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10	ETS	提前发送状态	该位指示要发送的帧已完全传输到Tx FIFO *注：可通过向该位写入1来清零该位*	R
b9	RWS	接收看门狗状态	当接收到的帧的长度大于2048个字节时，该位置1 *注：可通过向该位写入1来清零该位*	R
b8	RSS	接收停止状态	当接收过程进入停止状态时，该位置1 *注：可通过向该位写入1来清零该位*	R

b7	RUS	接收缓冲不可用状态	该位指示接收列表中的下一个描述符由主机所拥有，DMA无法获取，接收过程进入挂起状态 要恢复处理接收描述符，主机应更改描述符的拥有关系，然后发出接收轮询要求命令 如果未发出接收轮询要求命令，则当接收到下一个识别的输入帧时，接收过程会恢复 仅当上一个接收描述符由DMA所拥有时，该位置1 *注：可通过向该位写入1来清零该位*	R
b6	RIS	接收状态	该位指示帧接收已完成，特定的帧状态信息已发布在描述符中，接收保持运行状态 *注：可通过向该位写入1来清零该位*	R
b5	UNS	发送下溢状态	该位指示在帧发送期间，发送缓冲区发生下溢，发送会进入挂起状态，且描述符中下溢错误TDES0[1]标志位置1 *注：可通过向该位写入1来清零该位*	R
b4	OVS	接收上溢状态	该位指示在帧接收期间，接收缓冲区发生上溢，如果部分帧已传输到应用，则描述符中上溢错误RDES0[11]位置1 *注：可通过向该位写入1来清零该位*	R
b3	TJS	发送Jabber超时状态	该位指示发送Jabber定时器已过期，这意味着发送器过度有效，发送过程会中止并将其置于停止状态，且描述符中发送Jabber超时TDES0[14]标志位置1 *注：可通过向该位写入1来清零该位*	R
b2	TUS	发送缓冲不可用状态	该位指示发送列表中的下一个描述符由主机所拥有，DMA无法获取，发送会进入挂起状态 要恢复处理发送描述符，主机应更改描述符的拥有关系，然后发出发送轮询要求命令 *注：可通过向该位写入1来清零该位*	R
b1	TSS	发送停止状态	当发送过程进入停止状态时，该位置1 *注：可通过向该位写入1来清零该位*	R
b0	TIS	发送中断状态	该位指示帧发送已完成，特定的帧状态信息已发布在描述符中 *注：可通过向该位写入1来清零该位*	R

39.6.3.4 ETH_DMA 中断使能寄存器（ETH_DMA_INTENAR）

复位值: 0x00000000h

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AIE	ERE	FBE	-	-	ETE	RWE	RSE	RUE	RIE	UNE	OVE	TJE	TUE	TSE	TIE

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	NIE	普通中断使能汇总	0：普通事件中断无效 1：普通事件中断有效 *注：普通事件是指动作状态寄存器ETH_DMA_DMASTSR的B0、B2、B6、B14位所指示的事件*	R/W
b15	AIE	异常中断使能汇总	0：异常事件中断无效 1：异常事件中断有效 *注：异常事件是指动作状态寄存器ETH_DMA_DMASTSR的B1、B3、B4、B5、B7、B8、B9、B10、B14位所指示的事件*	R/W
b14	ERE	提前接收中断使能	0：提前接收中断无效 1：提前接收中断使能 *注：该位在NIE位置1后有效*	R/W
b13	FBE	致命总线错误中断使能	0：致命总线错误中断无效 1：致命总线错误中断有效 *注：该位在AIE位置1后有效*	R/W
b12~b11	Reserved	-	读出时为“0”,写入时写“0”	R/W
b10	ETE	提前发送中断使能	0：提前发送中断无效 1：提前发送中断有效 *注：该位在AIE位置1后有效*	R/W
b9	RWE	接收看门狗溢出中断使能	0：接收看门狗溢出中断无效 1：接收看门狗溢出中断有效 *注：该位在AIE位置1后有效*	R/W
b8	RSE	接收停止中断使能	0：接收停止中断无效 1：接收停止中断有效 *注：该位在AIE位置1后有效*	R/W
b7	RUE	接收缓冲不可用中断使能	0：接收缓冲不可用中断无效 1：接收缓冲不可用中断有效 *注：该位在AIE位置1后有效*	R/W
b6	RIE	接收中断使能	0：接收中断无效 1：接收中断有效 *注：该位在NIE位置1后有效*	R/W
b5	UNE	发送下溢中断使能	0：发送下溢中断无效	R/W

			1：发送下溢中断有效 *注：该位在AIE位置1后有效*	
b4	OVE	接收上溢中断使能	0：接收上溢中断无效 1：接收上溢中断有效 *注：该位在AIE位置1后有效*	R/W
b3	TJE	发送Jabber超时中断使能	0：发送Jabber超时中断无效 1：发送Jabber超时中断有效 *注：该位在AIE位置1后有效*	R/W
b2	TUE	发送缓冲不可用中断使能	0：发送缓冲不可用中断无效 1：发送缓冲不可用中断有效 *注：该位在NIE位置1后有效*	R/W
b1	TSE	发送停止中断使能	0：发送停止中断无效 1：发送停止中断有效 *注：该位在AIE位置1后有效*	R/W
b0	TIE	发送中断使能	0：发送中断无效 1：发送中断有效 *注：该位在NIE位置1后有效*	R/W

39.6.3.5 ETH_DMA 帧丢失统计寄存器（ETH_DMA_RFRCNTR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	OVFOVF	OVFCNT[10:0]											UNAOVF

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

位	标记	位名	功能	读写
b31~b29	Reserved	-	读出时为“0”,写入时写“0”	R/W
b28	OVFOVF	上溢丢失帧计数溢出	该寄存器的OVFCNT位统计丢失帧计数器上溢时，该位置1	R
b27~b17	OVFCNT	上溢丢失帧统计	该位指示因Rx FIFO上溢情况以及帧长度过短（不足64字 节的好帧）而丢失的帧数量	R
b16	UNAOVF	缓冲不可用丢失帧计数溢出	该寄存器的UNACNT位统计丢失帧计数器上溢时，该位置1	R
b15~b0	UNACNT	缓冲不可用丢失帧统计	该位指示因缓冲区不可用（无可用的接收描述符）而丢失的帧数量	R

39.6.3.6 ETH_DMA 看门狗定时寄存器（ETH_DMA_REVWDTR）

复位值: 0x00000000h

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								RIWT[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	RIWT	接收看门狗计数	该位指示系统时钟周期数乘以256之后的时间值，即看门狗定时器的设定时间值 看门狗定时器会在RxDMA结束帧传输之后由编程设定的值触发，此时，动作状态寄存器ETH_DMA_DMASTSR的RIS位因相对应的描述符中RDES1.RIS位置1而未被置1，直到看门狗定时器计时结束时，RIS位才置1且看门狗定时器停止。当动作状态寄存器ETH_DMA_DMASTSR的RIS位设为高电平时，看门狗定时器复位	R/W

39.6.3.7 ETH_DMA 发送轮询要求寄存器（ETH_DMA_TXPOLLR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TXPOLL[31:16]															

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

位	标记	位名	功能	读写
b31~b0	TXPOLL	发送轮询要求发起	向该位写入任何值时，DMA都会读取ETH_DMA_CHTXDER寄存器指向的当前描述符，如果该描述符不可用（由主机所有），则TxDMA会返回到挂起状态，并将ETH_DMA_DMASTSR寄存器TUS位进行置位；如果该描述符可用，则发送会继续进行 使用该寄存器来指示TxDMA轮询发送描述符列表，检查当前描述符是否为DMA所有。如果TxDMA处于挂起模式，则发出发送轮询要求命令将其唤醒；如果发送帧中出现下溢错误或TxDMA所拥有的描述符不可用，则TxDMA会进入挂起模式 用户可以随时发出此命令，当该命令开始重新获取主机存储器的当前描述符后，TxDMA即会对其进行复位 *注：该位在读出时始终读出零*	R/W

39.6.3.8 ETH_DMA 接收轮询要求寄存器（ETH_DMA_RXPOLLR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RXPOLL[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RXPOLL[51:0]															

位	标记	位名	功能	读写
b31~b0	RXPOLL	接收轮询要求发起	向该位写入任何值时，DMA都会读取ETH_DMA_CHRXDER寄存器指向的当前描述符，如果该描述符不可用（由主机所有），则RxDMA会返回到挂起状态，且不会将ETH_DMA_DMASTSR寄存器RUS位进行置位；如果该描述符可用，则RxDMA将返回到活动状态 使用该寄存器来指示RxDMA轮询接收描述符列表，检查新描述符，用于将RxDMA从挂起状态唤醒，仅当RxDMA所拥有的描述符不可用时，它才会进入挂起状态 *注：该位在读出时始终读出零*	R/W

39.6.3.9 ETH_DMA 发送描述符列表地址寄存器（ETH_DMA_TXDLADR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TXDLAD[31:16]															

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

位	标记	位名	功能	读写
b31~b0	TXDLAD	发送描述符首地址	该位表示发送描述符列表中的首个描述符的地址 *注：最低2位在读出时始终读出零*	R/W

39.6.3.10 ETH_DMA 接收描述符列表地址寄存器（ETH_DMA_RXDLADR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RXDLAD[31:16]															

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

位	标记	位名	功能	读写
b31~b0	RXDLAD	接收描述符首地址	该位表示接收描述符列表中的首个描述符的地址 *注：最低2位在读出时始终读出零*	R/W

39.6.3.11 ETH_DMA 当前主机发送描述符寄存器（ETH_DMA_CHTXDER）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CHTXDE[31:16]															

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

位	标记	位名	功能	读写
b31~b0	CHTXDE	发送描述符地址指针	该位指向DMA所读取的当前发送描述符的起始地址	R

39.6.3.12 ETH_DMA 当前主机接收描述符寄存器（ETH_DMA_CHRXDER）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CHRXDE[31:16]															

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

位	标记	位名	功能	读写
b31~b0	CHRXDE	接收描述符地址指针	该位指向DMA所读取的当前接收描述符的起始地址	R

39.6.3.13 ETH_DMA 当前主机发送缓冲区寄存器（ETH_DMA_CHTXBFR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CHTXBF[31:16]															

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

位	标记	位名	功能	读写
b31~b0	CHTXBF	发送缓冲区地址指针	该位指向DMA所读取的当前发送缓冲区地址	R

39.6.3.14 ETH_DMA 当前主机接收缓冲区寄存器（ETH_DMA_CHRXBFR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CHRXBF[31:16]															

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

位	标记	位名	功能	读写
b31~b0	CHRXBF	接收缓冲区地址指针	该位指向DMA所读取的当前接收缓冲区地址	R

39.6.4 ETH_MMC 寄存器

39.6.4.1 ETH_MMC MMC 控制寄存器（ETH_MMC_MMCCTLR）

复位值: 0x00000000h

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										MCPSEL	MCPSET	MCF	ROR	COS	CRST

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	MCPSEL	计数器预设值选择	0：将所有MMC统计寄存器的值预设为“预设半值”（7FF0H） 1：将所有MMC统计寄存器的值预设为“预设全值”（FFF0H） *注： 预设全值是指0000H-0010H=FFF0H* *预设半值是指8000H-0010H=7FF0H*	R/W
b4	MCPSET	计数器预设值设定	该位置1后，根据该寄存器的MCPSEL位的设定将所有的MMC统计寄存器预设为全值或半值 *注：该位在置位后1个时钟后自动清零*	R/W
b3	MCF	计数器冻结	该位置1后，所有MMC统计计数器的值被冻结，不再因为接收或发送帧而变化	R/W
b2	ROR	计数器读复位	该位置1后，所有MMC统计计数器在被读取之后便复位为零 *注1：不管该寄存器的MCF位是否有效，该位都可有效* *注2：读取统计计数器的低8位时，即视为读动作*	R/W
b1	COS	计数器计满停止	0：计数器计到FFFFH后，归零，重新计数 1：计数器计到FFFFH后，停止，不再计数	R/W
b0	CRST	计数器复位	置1后，所有MMC统计计数器的值复位 *注：该位在置位后1个时钟后自动清零*	R/W

39.6.4.2 ETH_MMC Rx 统计状态寄存器（ETH_MMC_REVSTSR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved												RXOEIS	RXLEIS	RXUGIS	-

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								RXREIS	RXAEIS	RXCEIS	RXMGIS	RXBGIS	-	-	-

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	RXOEIS	超范围错误帧统计状态	0：接收到的超范围错误帧数未到半值或全值 1：接收到的超范围错误帧数达到半值或全值 *注：该位在读后变为0*	R
b18	RXLEIS	长度错误帧统计状态	0：接收到的长度错误帧数未到半值或全值 1：接收到的长度错误帧数达到半值或全值 *注：该位在读后变为0*	R
b17	RXUGIS	单播好帧统计状态	0：接收到的单播好帧数未到半值或全值 1：接收到的单播好帧数达到半值或全值 *注：该位在读后变为0*	R
b16~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	RXREIS	矮帧错误帧统计状态	0：接收到的矮帧错误帧数未到半值或全值 1：接收到的矮帧错误帧数达到半值或全值 *注：该位在读后变为0*	R
b6	RXAEIS	对齐错误帧统计状态	0：接收到的对齐错误帧数未到半值或全值 1：接收到的对齐错误帧数达到半值或全值 *注：该位在读后变为0*	R
b5	RXCEIS	CRC错误帧统计状态	0：接收到的CRC错误帧数未到半值或全值 1：接收到的CRC错误帧数达到半值或全值 *注：该位在读后变为0*	R
b4	RXMGIS	多播好帧统计状态	0：接收到的多播好帧数未到半值或全值 1：接收到的多播好帧数达到半值或全值 *注：该位在读后变为0*	R
b3	RXBGIS	广播好帧统计状态	0：接收到的广播好帧数未到半值或全值 1：接收到的广播好帧数达到半值或全值 *注：该位在读后变为0*	R
b2~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

39.6.4.3 ETH_MMC Tx 统计状态寄存器（ETH_MMC_TRSSTSR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved									TXEDEIS	TXUGIS	-	TXCAEIS	TXECEIS	TXLCEIS	TXDEEIS

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												TXMGIS	TXBGIS	-	-

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	TXEDEIS	过度延迟错误帧统计状态	0：传送的过度延迟错误帧数未到半值或全值 1：传送的过度延迟错误帧数达到半值或全值 *注：该位在读后变为0*	R
b21	TXUGIS	单播好帧统计状态	0：传送的单播好帧数未到半值或全值 1：传送的单播好帧数达到半值或全值 *注：该位在读后变为0*	R
b20	Reserved	-	读出时为“0”,写入时写“0”	R/W
b19	TXCAEIS	载波错误帧统计状态	0：传送的载波错误帧数未到半值或全值 1：传送的载波错误帧数达到半值或全值 *注：该位在读后变为0*	R
b18	TXECEIS	过度冲突错误帧统计状态	0：传送的过度冲突错误帧数未到半值或全值 1：传送的过度冲突错误帧数达到半值或全值 *注：该位在读后变为0*	R
b17	TXLCEIS	延迟冲突错误帧统计状态	0：传送的延迟冲突错误帧数未到半值或全值 1：传送的延迟冲突错误帧数达到半值或全值 *注：该位在读后变为0*	R
b16	TXDEEIS	延迟错误帧统计状态	0：传送的延迟错误帧数未到半值或全值 1：传送的延迟错误帧数达到半值或全值 *注：该位在读后变为0*	R
b15~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	TXMGIS	多播好帧统计状态	0：传送的多播好帧数未到半值或全值 1：传送的多播好帧数达到半值或全值 *注：该位在读后变为0*	R
b2	TXBGIS	广播好帧统计状态	0：传送的广播好帧数未到半值或全值 1：传送的广播好帧数达到半值或全值 *注：该位在读后变为0*	R
b1~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

39.6.4.4 ETH_MMC Rx 中断控制寄存器（ETH_MMC_RITCTLR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved												RXOEIM	RXLEIM	RXUGIM	-

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								RXREIM	RXAEIM	RXCEIM	RXMGIM	RXBGIM	-	-	-

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	RXOEIM	超范围错误帧中断屏蔽	0：RXOEIS置位时，超范围错误帧MMC统计中断不屏蔽 1：RXOEIS置位时，超范围错误帧MMC统计中断屏蔽	R/W
b18	RXLEIM	长度错误帧中断屏蔽	0：RXLEIS置位时，长度错误帧MMC统计中断不屏蔽 1：RXLEIS置位时，长度错误帧MMC统计中断屏蔽	R/W
b17	RXUGIM	单播好帧中断屏蔽	0：RXUGIS置位时，单播好帧MMC统计中断不屏蔽 1：RXUGIS置位时，单播好帧MMC统计中断屏蔽	R/W
b16~b8	Reserved	-	读出时为“0”,写入时写“0”	R/W
b7	RXREIM	矮帧错误帧中断屏蔽	0：RXREIS置位时，矮帧错误帧MMC统计中断不屏蔽 1：RXREIS置位时，矮帧错误帧MMC统计中断屏蔽	R/W
b6	RXAEIM	对齐错误帧中断屏蔽	0：RXAEIS置位时，对齐错误帧MMC统计中断不屏蔽 1：RXAEIS置位时，对齐错误帧MMC统计中断屏蔽	R/W
b5	RXCEIM	CRC错误帧中断屏蔽	0：RXCEIS置位时，CRC错误帧MMC统计中断不屏蔽 1：RXCEIS置位时，CRC错误帧MMC统计中断屏蔽	R/W
b4	RXMGIM	多播好帧中断屏蔽	0：RXMGIS置位时，多播好帧MMC统计中断不屏蔽 1：RXMGIS置位时，多播好帧MMC统计中断屏蔽	R/W
b3	RXBGIM	广播好帧中断屏蔽	0：RXBGIS置位时，广播好帧MMC统计中断不屏蔽 1：RXBGIS置位时，广播好帧MMC统计中断屏蔽	R/W
b2~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

39.6.4.5 ETH_MMC Tx 中断控制寄存器（ETH_MMC_TITCTLR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved									TXEDEIM	TXUGIM	-	TXCAEIM	TXECEIM	TXLCEIM	TXDEEIM

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												TXMGIM	TXBGIM	-	-

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	TXEDEIM	过度延迟错误帧中断屏蔽	0：TXEDEIS置位时，过度延迟错误帧MMC统计中断不屏蔽 1：TXEDEIS置位时，过度延迟错误帧MMC统计中断屏蔽	R/W
b21	TXUGIM	单播好帧中断屏蔽	0：TXUGIS置位时，单播好帧MMC统计中断不屏蔽 1：TXUGIS置位时，单播好帧MMC统计中断屏蔽	R/W
b20	Reserved	-	读出时为“0”,写入时写“0”	R/W
b19	TXCAEIM	载波错误帧中断屏蔽	0：TXCAEIS置位时，载波错误帧MMC统计中断不屏蔽 1：TXCAEIS置位时，载波错误帧MMC统计中断屏蔽	R/W
b18	TXECEIM	过度冲突错误帧中断屏蔽	0：TXECEIS置位时，过度冲突错误帧MMC统计中断不屏蔽 1：TXECEIS置位时，过度冲突错误帧MMC统计中断屏蔽	R/W
b17	TXLCEIM	延迟冲突错误帧中断屏蔽	0：TXLCEIS置位时，延迟冲突错误帧MMC统计中断不屏蔽 1：TXLCEIS置位时，延迟冲突错误帧MMC统计中断屏蔽	R/W
b16	TXDEEIM	延迟错误帧中断屏蔽	0：TXDEEIS置位时，延迟错误帧MMC统计中断不屏蔽 1：TXDEEIS置位时，延迟错误帧MMC统计中断屏蔽	R/W
b15~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	TXMGIM	多播好帧中断屏蔽	0：TXMGIS置位时，多播好帧MMC统计中断不屏蔽 1：TXMGIS置位时，多播好帧MMC统计中断屏蔽	R/W
b2	TXBGIM	广播好帧中断屏蔽	0：TXBGIS置位时，广播好帧MMC统计中断不屏蔽 1：TXBGIS置位时，广播好帧MMC统计中断屏蔽	R/W
b1~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

39.6.4.6 ETH_MMC Rx 单播好帧统计寄存器（ETH_MMC_RXUNGFR）

复位值: 0x00000000h

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
RXUNGCNT[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	RXUNGCNT	单播好帧数	该位表明接收到的单播好帧的数量	R

39.6.4.7 ETH_MMC Rx 多播好帧统计寄存器（ETH_MMC_RXMUGFR）

复位值: 0x00000000h

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
RXMUGCNT[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	RXMUGCNT	多播好帧数	该位表明接收到的多播好帧的数量	R

39.6.4.8 ETH_MMC Rx 广播好帧统计寄存器（ETH_MMC_RXBRGFR）

复位值: 0x00000000h

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
RXBRGCNT[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	RXBRGCNT	广播好帧数	该位表明接收到的广播好帧的数量	R

39.6.4.9 ETH_MMC RxCRC 错误帧统计寄存器（ETH_MMC_RXCREFR）

复位值: 0x00000000h

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
RXCRECNT[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	RXCRECNT	CRC错误帧数	该位表明接收到的CRC错误帧的数量	R

39.6.4.10 ETH_MMC Rx 对齐错误帧统计寄存器（ETH_MMC_RXALEFR）

复位值: 0x00000000h

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
RXALECNT[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	RXALECNT	对齐错误帧数	该位表明接收到的对齐错误帧的数量	R

39.6.4.11 ETH_MMC Rx 矮帧错误帧统计寄存器（ETH_MMC_RXRUEFR）

复位值: 0x00000000h

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
RXRUECNT[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	RXRUECNT	矮帧错误帧数	该位表明接收到的矮帧错误帧的数量	R

39.6.4.12 ETH_MMC Rx 长度错误帧统计寄存器（ETH_MMC_RXLEEFR）

复位值: 0x00000000h

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
RXLEECNT[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	RXLEECNT	长度错误帧数	该位表明接收到的长度错误帧的数量	R

39.6.4.13 ETH_MMC Rx 超范围错误帧统计寄存器（ETH_MMC_RXOREFR）

复位值: 0x00000000h

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
RXORECNT[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	RXORECNT	超范围错误帧数	该位表明接收到的超范围错误帧的数量	R

39.6.4.14 ETH_MMC Tx 单播好帧统计寄存器（ETH_MMC_TXUNGFR）

复位值: 0x00000000h

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
TXUNGCNT[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	TXUNGCNT	单播好帧数	该位表明已发送的单播好帧的数量	R

39.6.4.15 ETH_MMC Tx 多播好帧统计寄存器（ETH_MMC_TXMUGFR）

复位值: 0x00000000h

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
TXMUGCNT[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	TXMUGCNT	多播好帧数	该位表明已发送的多播好帧的数量	R

39.6.4.16 ETH_MMC Tx 广播好帧统计寄存器（ETH_MMC_TXBRGFR）

复位值: 0x00000000h

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
TXBRGCNT[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	TXBRGCNT	广播好帧数	该位表明已发送的广播好帧的数量	R

39.6.4.17 ETH_MMC Tx 延迟错误帧统计寄存器（ETH_MMC_TXDEEFR）

复位值: 0x00000000h

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
TXDEECNT[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	TXDEECNT	延迟错误帧数	该位表明已发送的延迟错误帧的数量	R

39.6.4.18 ETH_MMC Tx 载波错误帧统计寄存器（ETH_MMC_TXCAEFR）

复位值: 0x00000000h

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
TXCAECNT[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	TXCAECNT	载波错误帧数	该位表明已发送的载波错误帧的数量	R

39.6.4.19 ETH_MMC Tx 延迟冲突错误帧统计寄存器（ETH_MMC_TXLCEFR）

复位值: 0x00000000h

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
TXLCECNT[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	TXLCECNT	延迟冲突错误帧数	该位表明已发送的延迟冲突错误帧的数量	R

39.6.4.20 ETH_MMC Tx 过度冲突错误帧统计寄存器（ETH_MMC_TXECEFR）

复位值: 0x00000000h

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
TXECECNT[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	TXECECNT	过度冲突错误帧数	该位表明已发送的过度冲突错误帧的数量	R

39.6.4.21 ETH_MMC Tx 过度延迟错误帧统计寄存器（ETH_MMC_TXEDEFR）

复位值: 0x00000000h

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
TXEDECNT[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	TXEDECNT	过度延迟错误帧数	该位表明已发送的过度延迟错误帧的数量	R

40 外部存储器控制器（EXMC）

40.1 概要

外部存储器控制器 EXMC（External Memory Controller）是一个用来访问各种片外存储器，实现数据交换的独立模块。EXMC 通过配置可以把内部的 AMBA 协议接口转换为各种类型的专用片外存储器通信协议接口，包括 SRAM，PSRAM、NOR Flash，NAND Flash 和 SDRAM 等。EXMC 内部划分为多个子模块，每个子模块支持特定的存储器类型，用户可以通过对子模块的寄存器配置来控制外部对应类型的存储器。

40.2 基本特性

40.2.1 功能列表

EXMC 基本的功能列表如下所示。

支持的外部存储器类型	• SRAM
	• PSRAM
	• NOR Flash
	• NAND Flash
	• SDRAM
接口协议	• 支持 AMBA 协议与各种外部存储器的接口转换
接口时序	• 各种类型存储器的接口时序可编程
总线位宽	• 支持 8 位、16 位、32 位 MEM 总线宽度
总线复用	• NOR Flash 和 PSRAM 支持地址线和数据线复用
自动分割	• AMBA 总线位宽和外部存储器位宽不匹配时，支持自动分割及字节选择控制
中断类型	• NFC ECC 计算完成中断
	• NFC ECC 错误中断
	• NFC 设备访问结束中断

表 40-1 EXMC 的基本功能

40.2.2 控制器架构

图 40-1 为基本的 EXMC 架构框图。各种类型的外部存储器控制器独立生成对应协议的接口送至端口 MUX 逻辑。端口 MUX 逻辑将各种外部存储器的地址、数据、控制信号等共享在相同的端口上后再从芯片的端口上输出。因此 EXMC 控制器一次只能访问一个外部器件。

注：架构图中将 SRAM/PSRAM/NOR Flash 控制器定义为 SMC（Static Memory Controller）、SDRAM 控制器定义为 DMC（Dynamic Memory Controller）、将 NAND Flash 控制器定义为 NFC（NAND Flash Memory Controller），后面提到 SMC、DMC、NFC 时，与对应

的控制器有相同的含义。

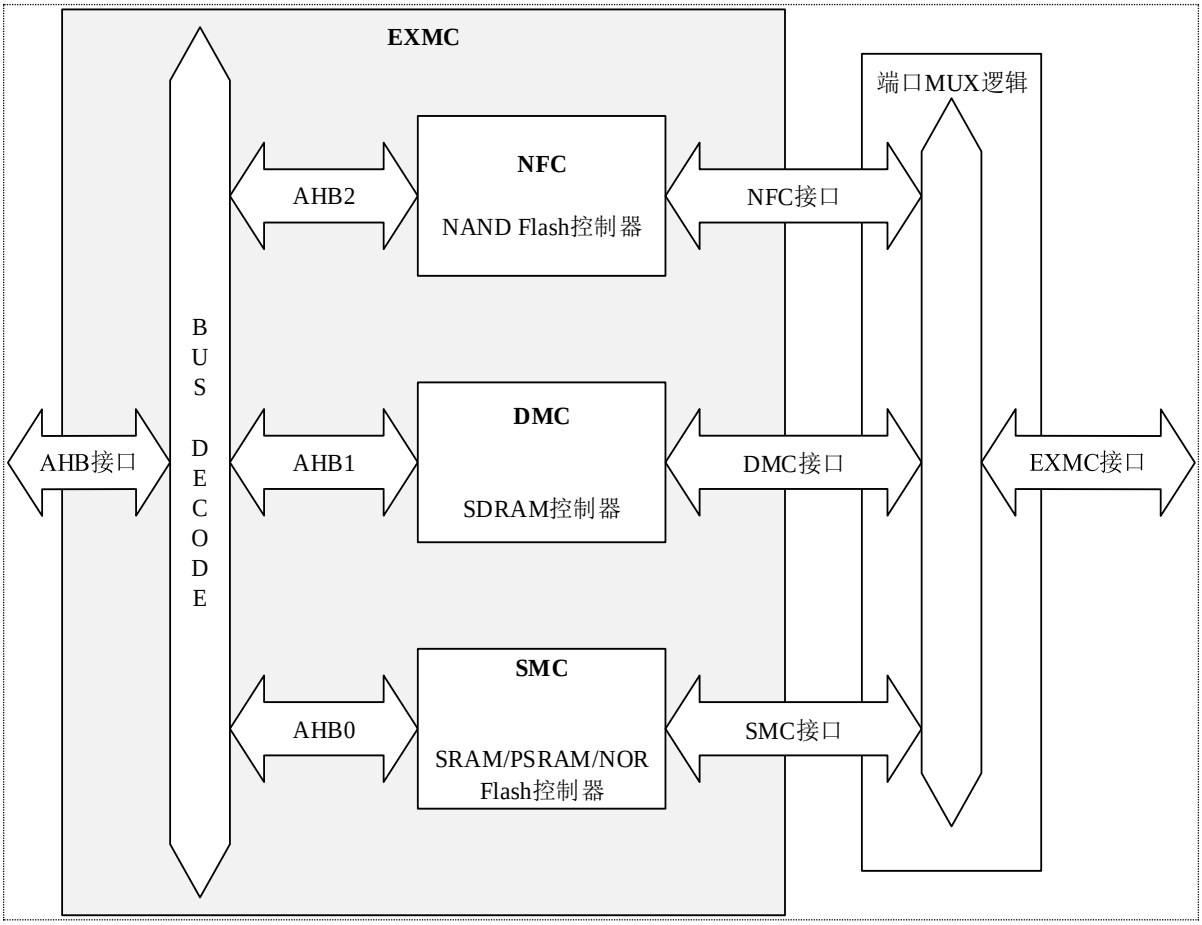


图 40-1 EXMC 架构图

40.2.3 基本访问规范

EXMC 是 AHB 总线协议和外部存储器协议的转换接口。在 SMC 和 DMC 的数据传输过程中，AHB 数据宽度和存储器数据宽度可能不相同。为了保证数据传输的一致性，读写访问需要遵从以下规范：

a) AHB 访问宽度等于存储器宽度，数据正常访问

b) AHB 访问宽度大于存储器宽度，自动将 AHB 访问分割成几个连续的存储器数据宽度的传输

c) AHB 访问宽度小于存储器宽度，通过字节选择控制信号 SMC_BLS[3:0]或 DMC_DQM[3:0]来访问对应的字节

AHB 访问宽度和各种数据位宽的存储器之间的具体的访问方式参见下表 40-2。

R/W	AHB 访问宽度	外部存储器位宽	注释
R	8	16	
	16		
	32		转换成 2 次 EXMC 读操作
	8	32	
	16		
	32		
W	8	16	使用字节控制信号 BLS[0]或 DQM[1]
	16		
	32		转换成 2 次 EXMC 写操作
	8	32	使用字节控制信号 BLS[0]或 DQM[3:1]
	16		使用字节控制信号 BLS[1:0]或 DQM[3:2]
	32		

表 40-2 AHB 访问宽度和存储器位宽对应访问方式表

注：上表仅适用于 SMC 和 DMC 的数据访问

40.2.4 地址映射

本产品中定义了总大小为 1GB 的外部存储器访问区间，用于不同类型的外部存储器和内部数据交换，包括 EXMC 和 QSPI 等。该 1GB 的空间按图 40-2 的方案分配，SRAM/PSRAM/NOR Flash（SMC）的数据访问映射 512MB 的地址空间、SDRAM（DMC）的数据访问映射 128MB 的地址空间、NAND Flash（NFC）的控制和数据访问映射 1MB 的地址空间、QSPI 的控制和数据访问映射 128MB 的地址空间。

注：地址空间为 0x98000000~0x9FFFFFFF 的访问控制由 QSPI 模块实现，具体请参考 QSPI 章节。

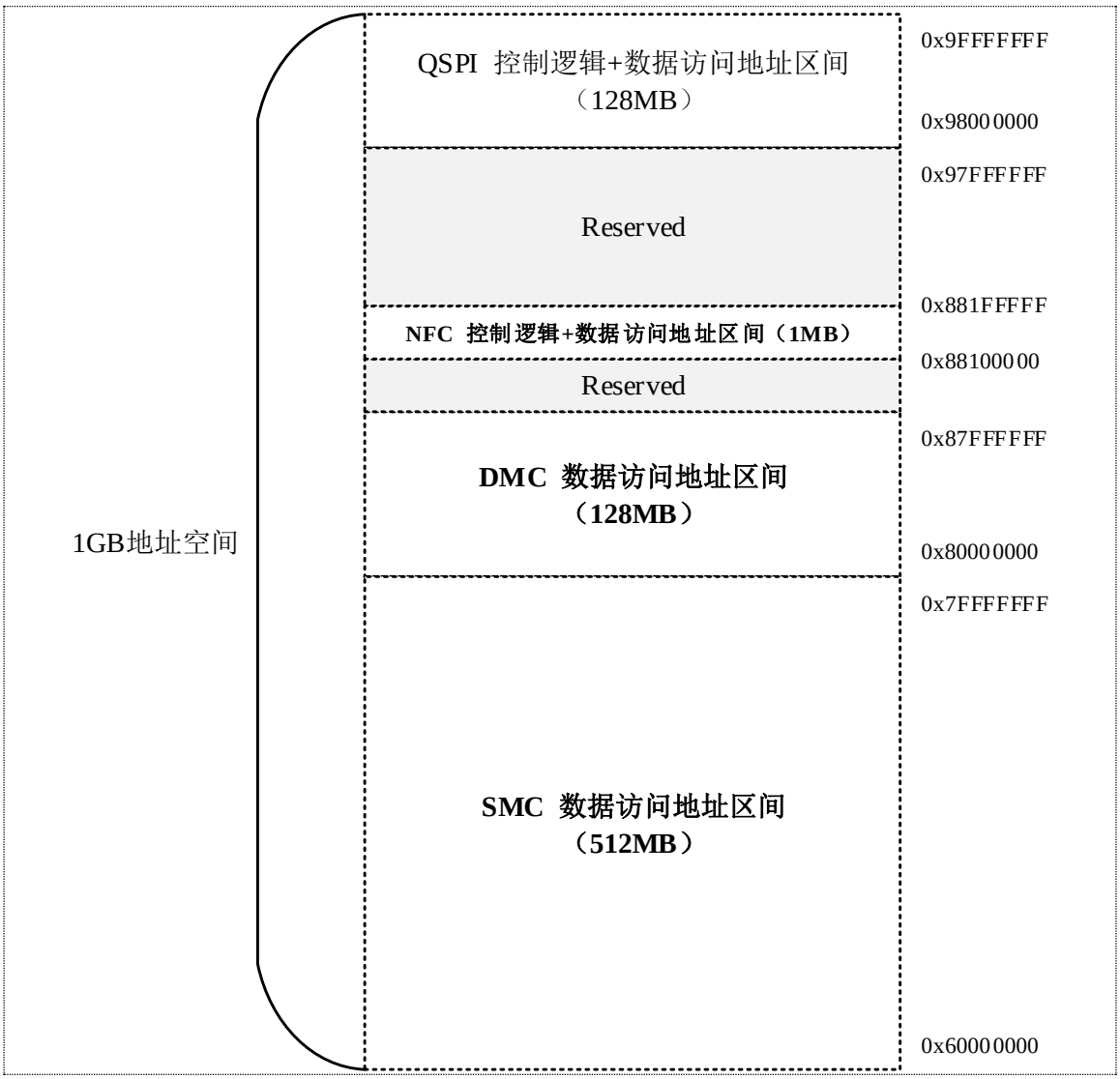


图 40-2 外部空间地址分配

SMC 的地址空间可划分为 4 个大小可编程的 Chip，通过片选控制寄存器（SMC_CSCR0~1）的设定可以划分各个 Chip 的空间大小。各个 Chip 对应的外部存储器对象可以独立地配置，并对应有不同的 CS（Chip Select）信号输出至外部端口。图 40-3 是 SMC 的 Chip Select 划分，当 CPU 访问的地址空间在某一个 SMC 的 CS 空间时，其对应的 SMC_CS 信号输出就变为有效电平。

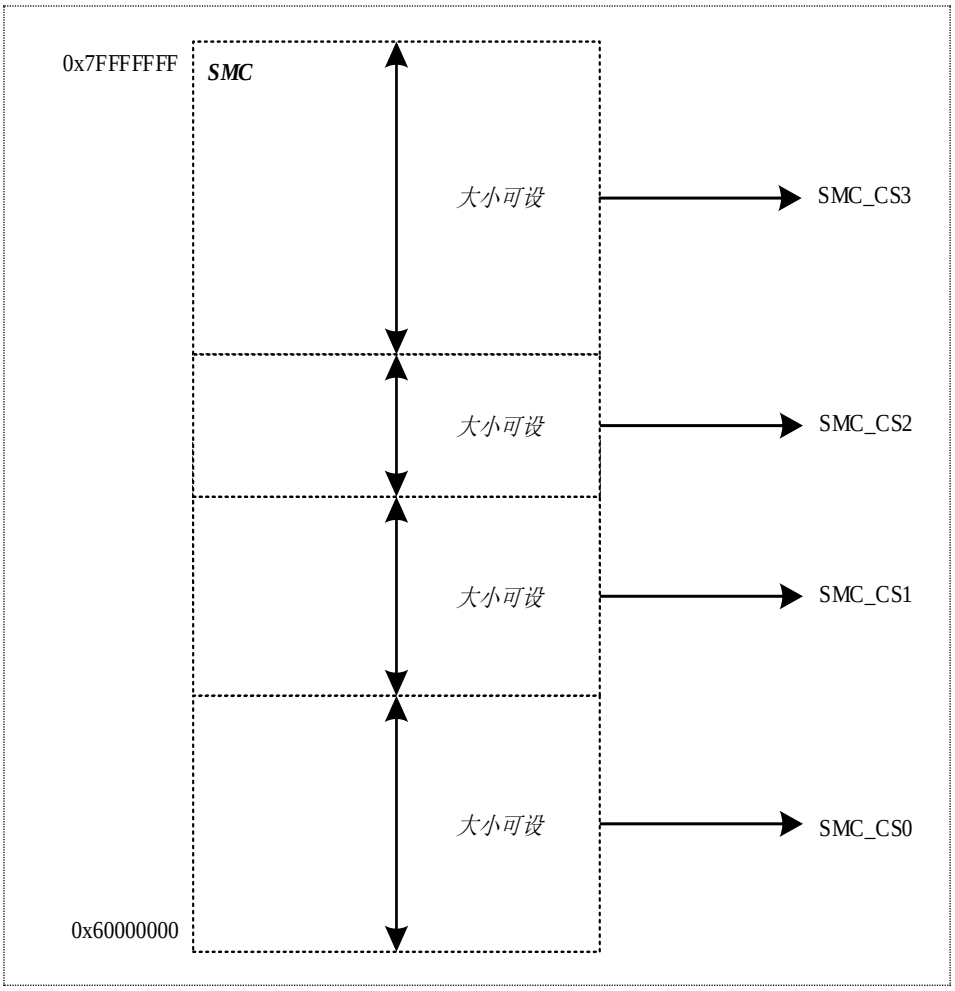


图 40-3 SMC 的地址空间划分

DMC 的地址空间可划分为 4 个大小可编程的 Chip，通过片选控制寄存器（DMC_CSCR0~3）的设定可以划分各个 Chip 的空间大小。各个 Chip 对应的外部存储器对象可以独立地配置，并对应有不同的 CS（Chip Select）信号输出至外部端口。图 40-4 是 DMC 的 Chip Select 划分，当 CPU 访问的地址空间在某一个 DMC 的 CS 空间时，其对应的 DMC_CS 信号输出就变为有效电平。

注：SMC 和 DMC 的地址设定方法，参考表 40-18 的设定例。

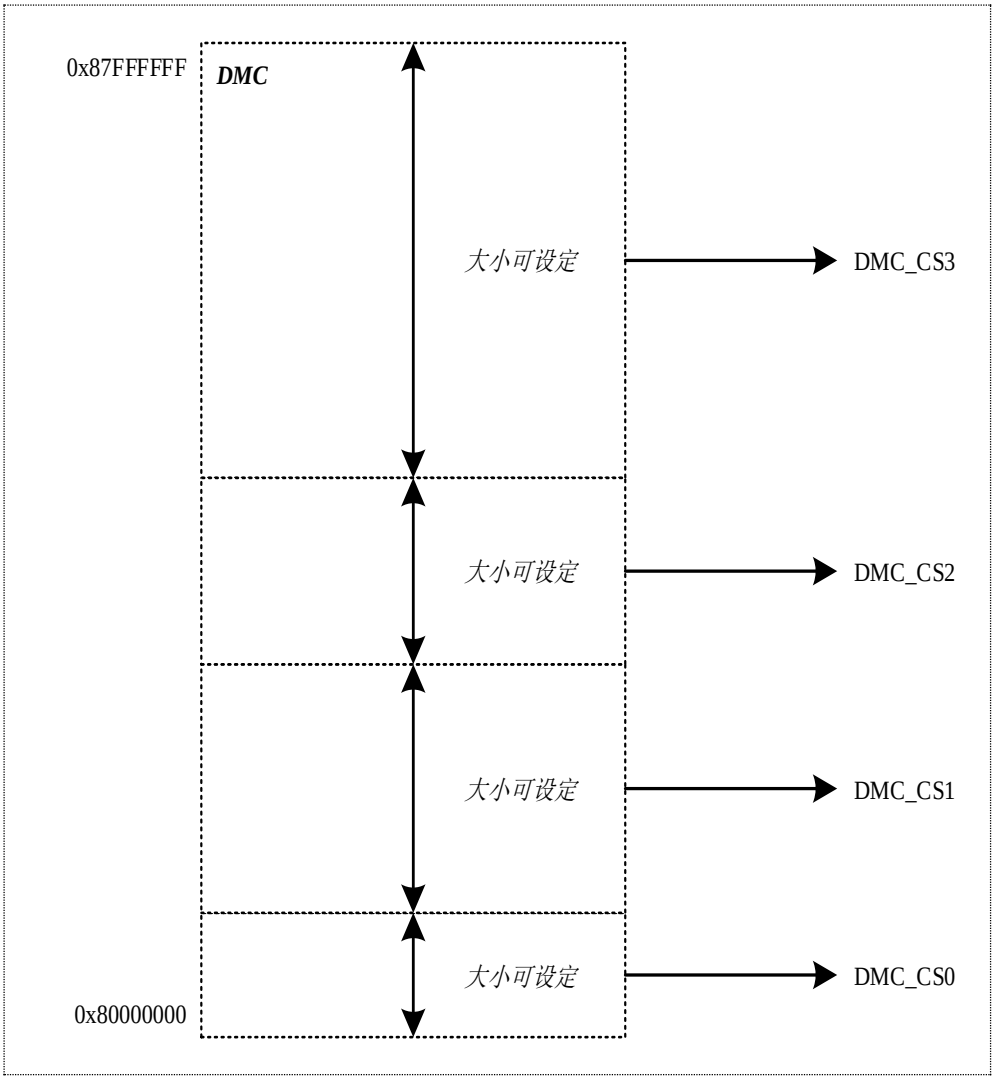


图 40-4 DMC 的地址空间划分

NFC 的地址空间分为数据寄存器空间和控制寄存器空间。其中数据寄存器占用 32KB 空间，控制寄存器占用 32KB 空间，共计 64KB，剩余地址为 NFC 预留地址。下图 40-5 表示了 NFC 的地址空间划分。

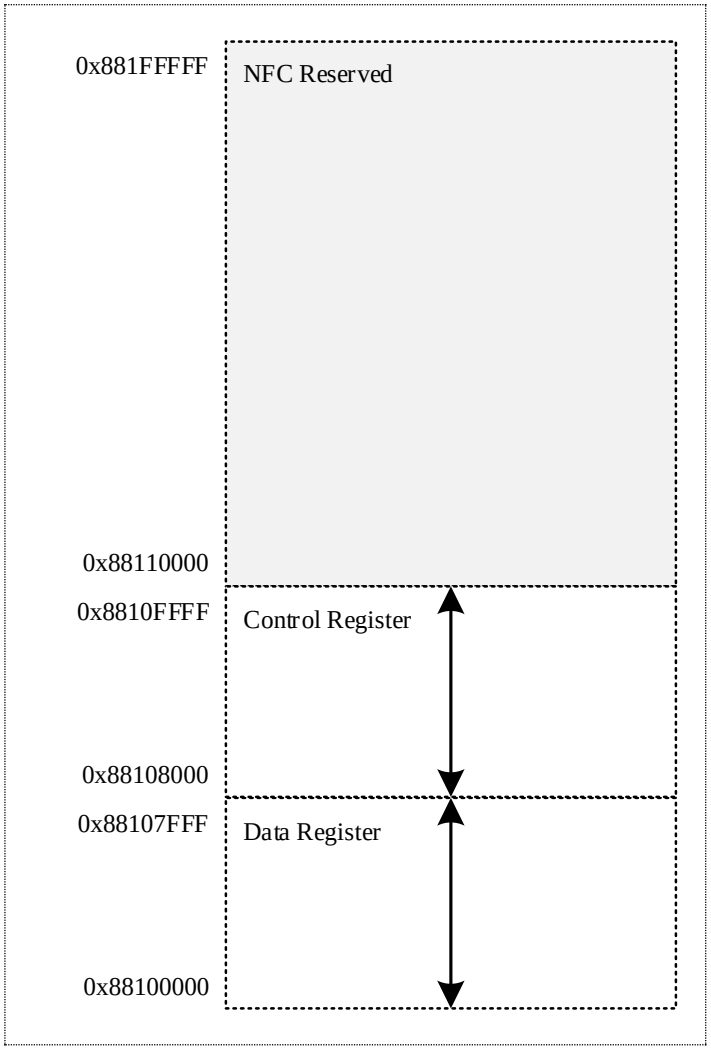


图 40-5 NFC 的地址空间划分

40.2.5 协议接口

SMC、DMC、NFC 对应不同类型的存储器，各自有不同类型的协议接口信号。EXMC 通过该接口协议实现和外部存储器的数据交换。以下为各种类型存储器所需要的协议接口说明。

40.2.5.1 SMC 协议接口

下表为 SRAM/PSRAM/NOR Flash 控制（SMC）访问所需的协议接口。

协议接口名	方向	有效电平	功能描述
SMC_CLK	out	-	SMC 的时钟输出
SMC_ADD[29:0]	out	-	SMC 的地址输出
SMC_DATA[31:0]	inout	-	SMC 的数据
SMC_WE	out	L	SMC 的写使能
SMC_OE	out	L	SMC 的输出使能
SMC_CS[3:0]	out	L	SMC 的片选信号
SMC_BLS[3:0]	out	L	SMC 的字节选通信号
SMC_ADV	out	L	SMC 的地址锁存信号
SMC_CRE	out	H	SMC 的配置寄存器模式信号
SMC_WAIT	in	L	SMC 的输入等待信号
SMC_BAA	out	L	SMC 的地址提示信号

表 40-3 SMC 协议接口

40.2.5.2 DMC 协议接口

下表为 SDRAM 控制（DMC）访问所需的协议接口。

协议接口名	方向	有效电平	功能描述
DMC_CLK	out	-	DMC 的时钟输出
DMC_ADD[15:0]	out	-	DMC 的地址输出
DMC_DATA[31:0]	inout	-	DMC 的数据
DMC_CKE	out	H	DMC 的 CLK 输出使能
DMC_AP	out	H	DMC 的自动充电信号
DMC_WE	out	L	DMC 的写使能
DMC_RAS	out	L	DMC 的行地址选通信号
DMC_CAS	out	L	DMC 的列地址选通信号
DMC_BA[1:0]	out	-	DMC 的 Bank 地址信号
DMC_CS[3:0]	out	L	DMC 的片选信号
DMC_DQM[3:0]	out	L	DMC 的字节选通信号

表 40-4 DMC 协议接口

40.2.5.3 NFC 协议接口

下表为 NAND Flash 控制（NFC）访问所需的协议接口。

协议接口名	方向	有效电平	功能描述
NFC_CLE	out	H	NFC 的命令锁存信号
NFC_ALE	out	H	NFC 的地址锁存信号
NFC_DATA[15:0]	inout	-	NFC 的地址和数据
NFC_CE[7:0]	out	L	NFC 的片选信号
NFC_WE	out	L	NFC 的写使能
NFC_RE	out	L	NFC 的读使能
NFC_WP	out	L	NFC 的写保护信号
NFC_RB[7:0]	in	L	NFC 的输入忙信号

表 40-5 NFC 协议接口

基于上述三种类型外部存储器的接口信号表，将芯片的 EXMC 端口功能配置如下表。

引脚名	方向	有效电平	功能描述		
			SMC	DMC	NFC
EXMC_CLK	O	H/L	SMC_CLK	DMC_CLK	-
EXMC_ADD29~0	O	H/L	SMC_ADD[29:18]	-	-
	O	H/L	SMC_ADD[17:16]	DMC_BA[1:0]	-
	O	H/L	SMC_ADD[15:0]	DMC_ADD[15:0]	-
EXMC_DATA31~0	IO	H/L	SMC_DATA[31:16]	DMC_DATA[31:16]	-
	IO	H/L	SMC_DATA[15:0]	DMC_DATA[15:0]	NFC_DATA[15:0]
EXMC_WE	O	L	SMC_WE	DMC_WE	NFC_WE
EXMC_CE7~0	O	L	SMC_BLS[3:0]	DMC_DQM[3:0]	NFC_CE[7:4]
	O	L	SMC_CS[3:0]	DMC_CS[3:0]	NFC_CE[3:0]
EXMC_OE	O	L	SMC_OE	DMC_RAS	NFC_RE
EXMC_BAA	O	L	SMC_BAA	DMC_CAS	NFC_WP
EXMC_ADV	O	L	SMC_ADV	-	-
EXMC_ALE	O	H	SMC_CRE	DMC_CKE	NFC_ALE
EXMC_CLE	O	H	-	DMC_AP	NFC_CLE
EXMC_RB7~0	I	L	-	-	NFC_RB[7:1]
	I	L	SMC_WAIT	-	NFC_RB[0]

表 40-6 EXMC 端口功能分配

40.3 功能说明

40.3.1 SMC-SRAM/PSRAM/NOR Flash 控制器

40.3.1.1 SRAM/PSRAM/NOR Flash 介绍

SRAM/NOR Flash 都是具有静态存取功能的存储器，不需要刷新电路即能保存它内部存储的数据。其地址线的宽度决定了存储单元的数量，每一个存储单元上对应存储器存储宽度大小的空间用于存储数据，因此 SRAM/PSRAM/NOR Flash 存储器的容量大小是存储器数据宽度 x2^地址线宽度。

PSRAM 是假静态随机存储器，由于它具有 SRAM 的接口，因此能像 SRAM 一样，与系统内部通过 EXMC 的 SMC 模块实现数据交换。

40.3.1.2 SMC 基本功能

SMC 基本功能特性如下：

- 支持 16 位、32 位外部存储器数据带宽
- AHB 字、半字、字节访问
- 为每个存储器 Chip 提供独立的片选控制
- 字节选择信号输出
- 地址线、数据线复用
- 可编程的协议时序参数
- 可编程速率的自动刷新动作（PSRAM 时使用）
- 具有 2 个 47 位的命令 FIFO
- 具有 4 个 36 位的写数据 FIFO
- 具有 4 个 32 位的读数据 FIFO
- 低功耗管理

40.3.1.3 SMC 初始设定

SMC 在上电复位之后的状态图及状态之间的切换如图 40-6 所示。

初始复位时 SMC 为 Low Power 状态；通过状态控制寄存器（SMC_STCR0 和 SMC_STCR1）的设定，SMC 可以在 Ready 状态和 Low Power 状态之间进行切换，具体请参考【SMC 低功耗管理】章节。

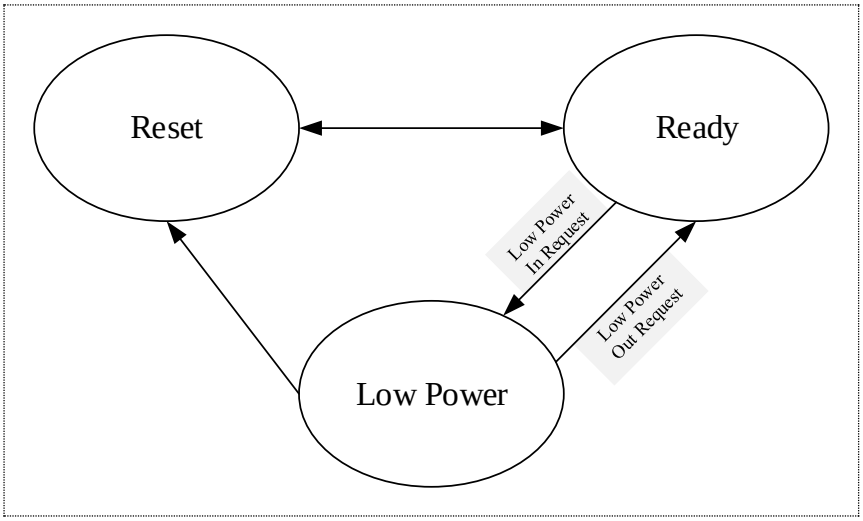


图 40-6 SMC 状态图

SMC 在和外部 SRAM、PSRAM、NOR Flash 等进行通信之前，必须进行初始设定，配置好相关的参数，以保证能进行正确的数据传输。具体的设定顺序参考下图 40-7。

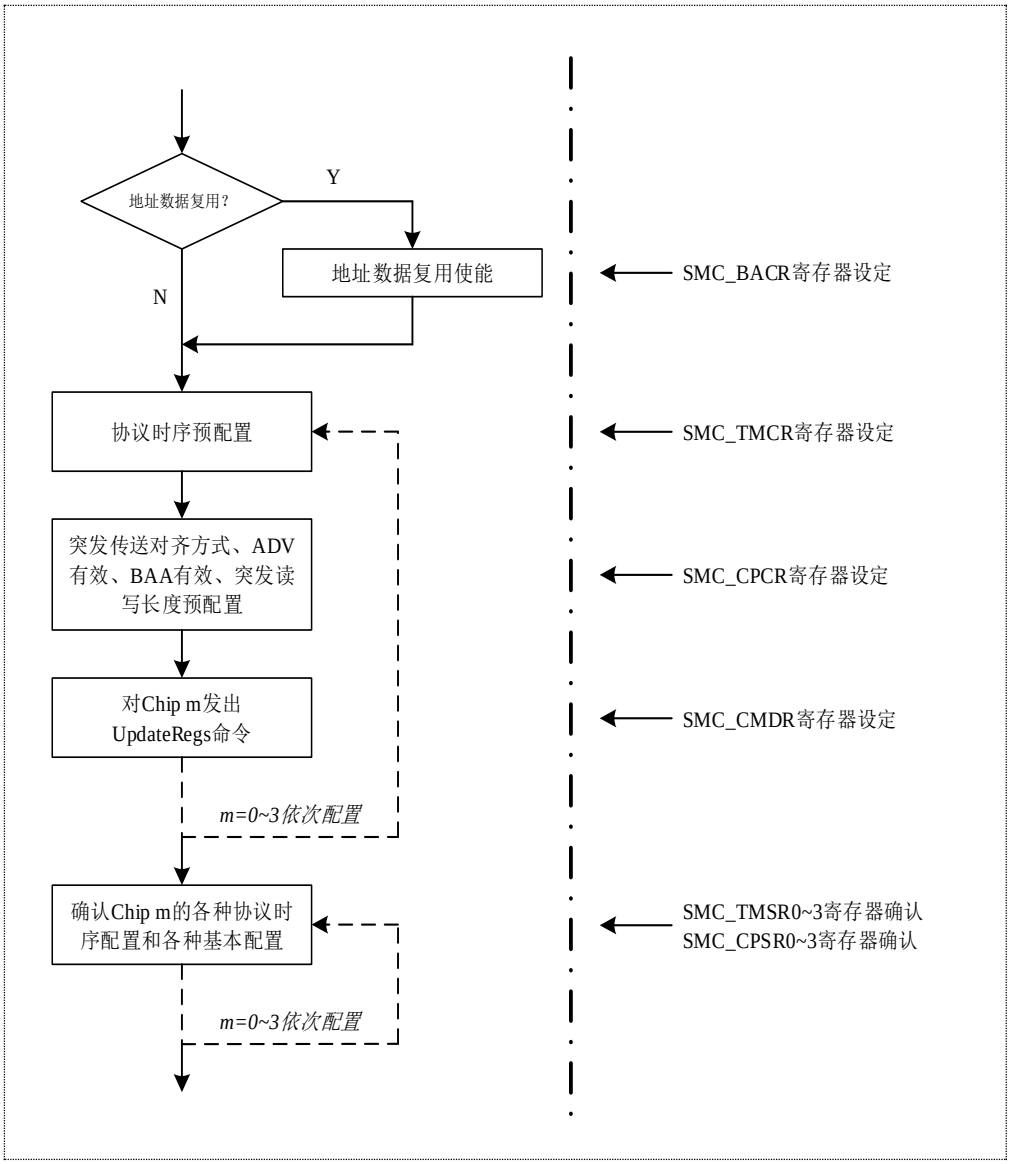


图 40-7 SMC 初始设定流程

40.3.1.4 SMC 访问动作

访问方式

SMC 对外部存储器的读写，可以选择同步访问方式或者异步访问方式。

同步访问方式是对应同步访问类型的外部存储器，SMC 在读写控制信号等输出时，时钟信号也同时输出，外部存储器根据 SMC 输出的时钟与 SMC 进行数据传输，即 SMC_CLK 有效；异步访问方式是对应异步访问类型的外部存储器，SMC 在读写控制信号等输出时，时钟信号不输出（输出始终为低），外部存储器依据其它控制信号实现数据访问。

在同步访问方式时，SMC 和外部存储器间通过 SMC_WAIT 信号来控制读写动作。在

SMC_WAIT 信号输入低电平时，SMC 处于等待状态，不执行任何操作，等到 SMC_WAIT 变为高电平后 SMC 才正常的进行读写访问。

通过设定 CHIP 配置寄存器（SMC_CPCR）的 RSYN、WSYN 位，可选择同步读写方式或异步读写方式。

访问时序

SMC 支持多种类型的存储器的访问，可以把 AHB 的单次或突发读写操作转换成存储器的读写操作。下面列出几种控制器与外部存储器进行数据读写的时序例：

a. 单次读动作

b. 地址数据线复用的单次读动作

c. 单次写动作

d. 地址数据线复用的单次写动作

e. 突发读动作

f. 突发写动作

a) 表 40-7 和图 40-8~图 40-11 是单次读动作的基本时序图和设定例。

基本设定	MW	RSYN	RBL	WSYN	WBL	BAA	ADV	BLS
	<set>	<set>	b000	-	-	-	b0	b0
时序设定	t_rc	t_wc	t_ceoe	t_wp	t_pc	t_tr	- 表示不关注	
	b0011	-	b001	-	-	-	<set> 表示用户设定值	

表 40-7 单次读动作基本设定例

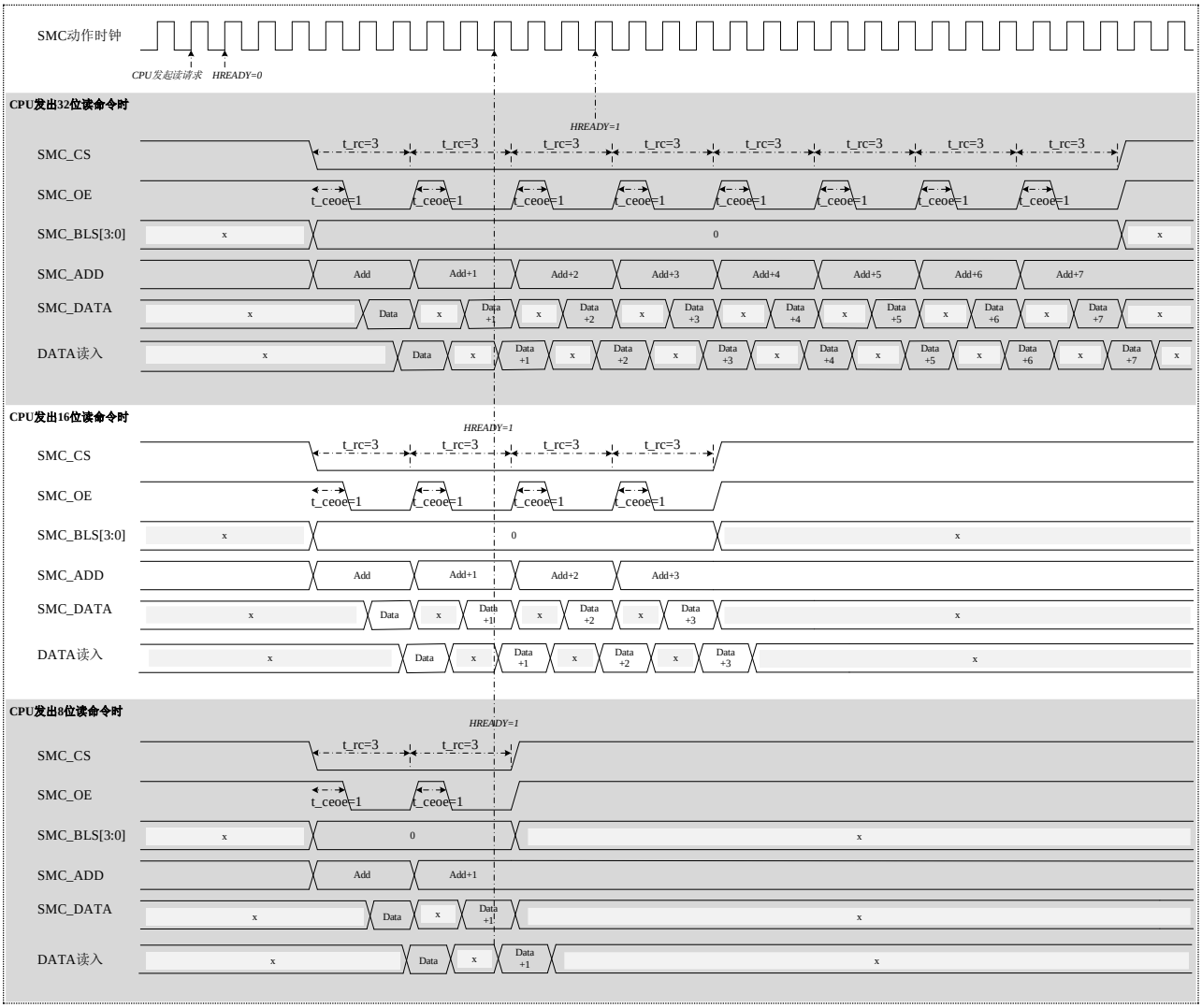


图 40-8　单次读动作基本时序（异步方式（RSYN=0）&16 位位宽（MW=01））

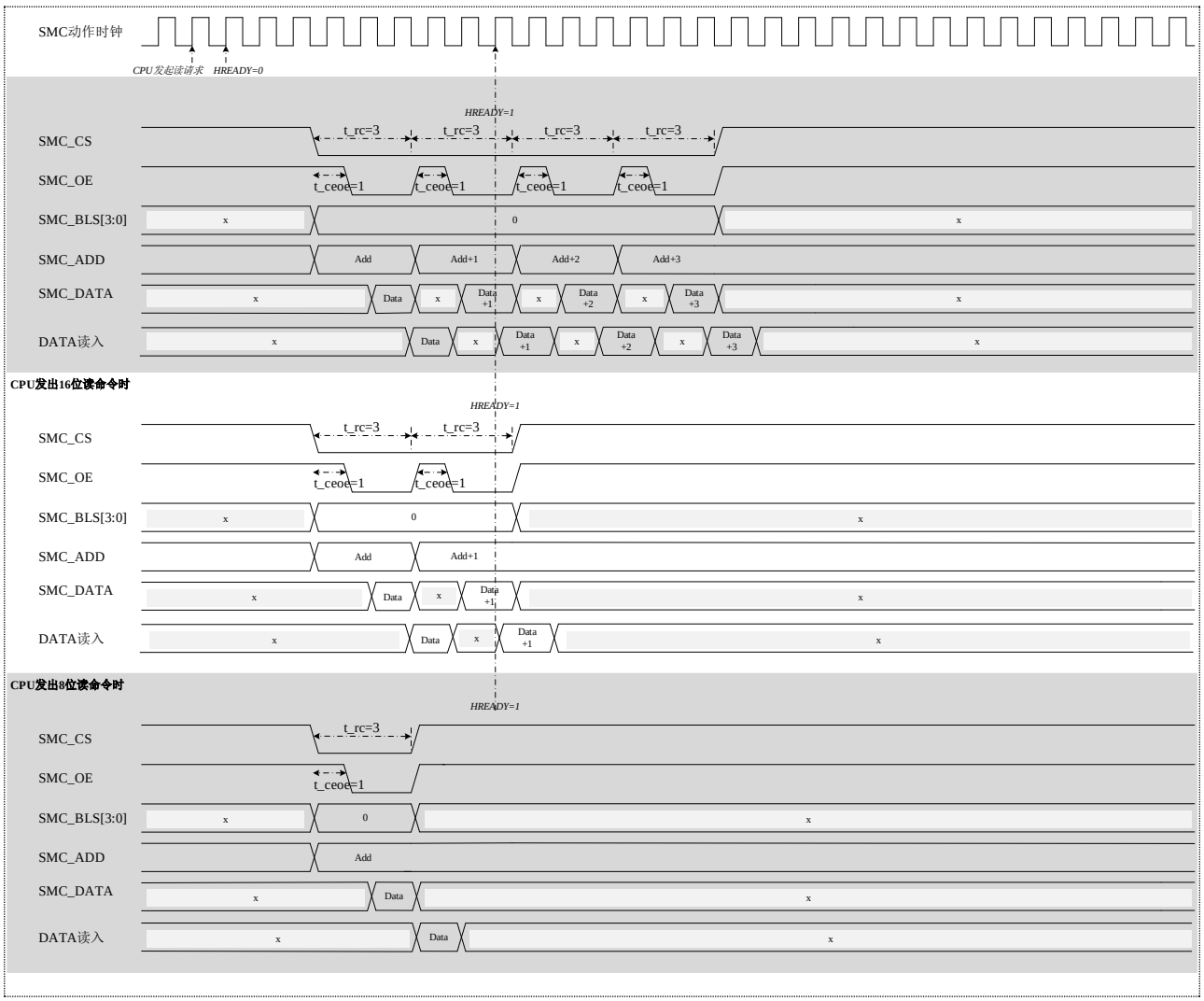


图 40-9　单次读动作基本时序（异步方式（RSYN=0）&32 位位宽（MW=10））

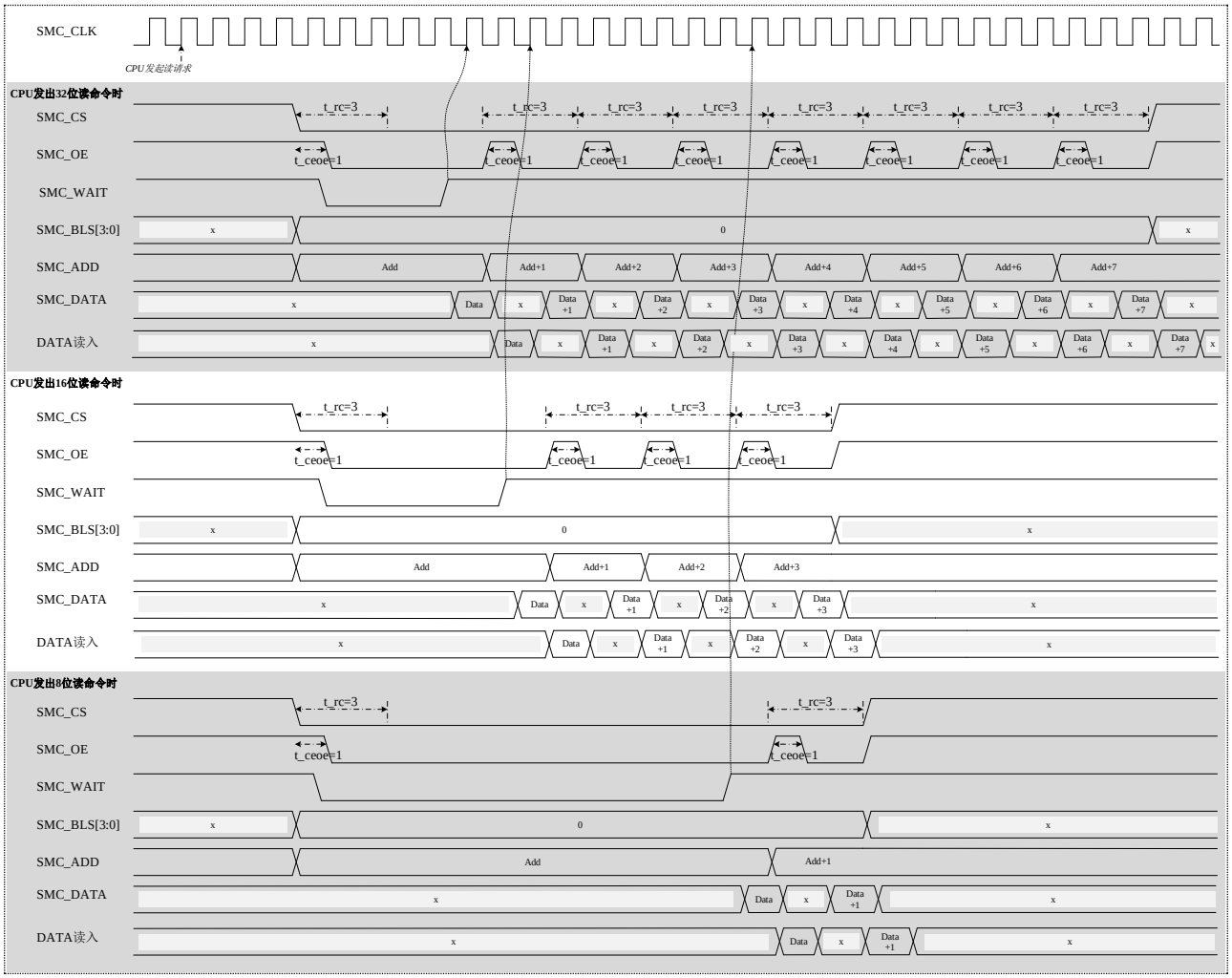


图 40-10　单次读动作基本时序（同步方式（RSYN=1）&16 位位宽（MW=01））

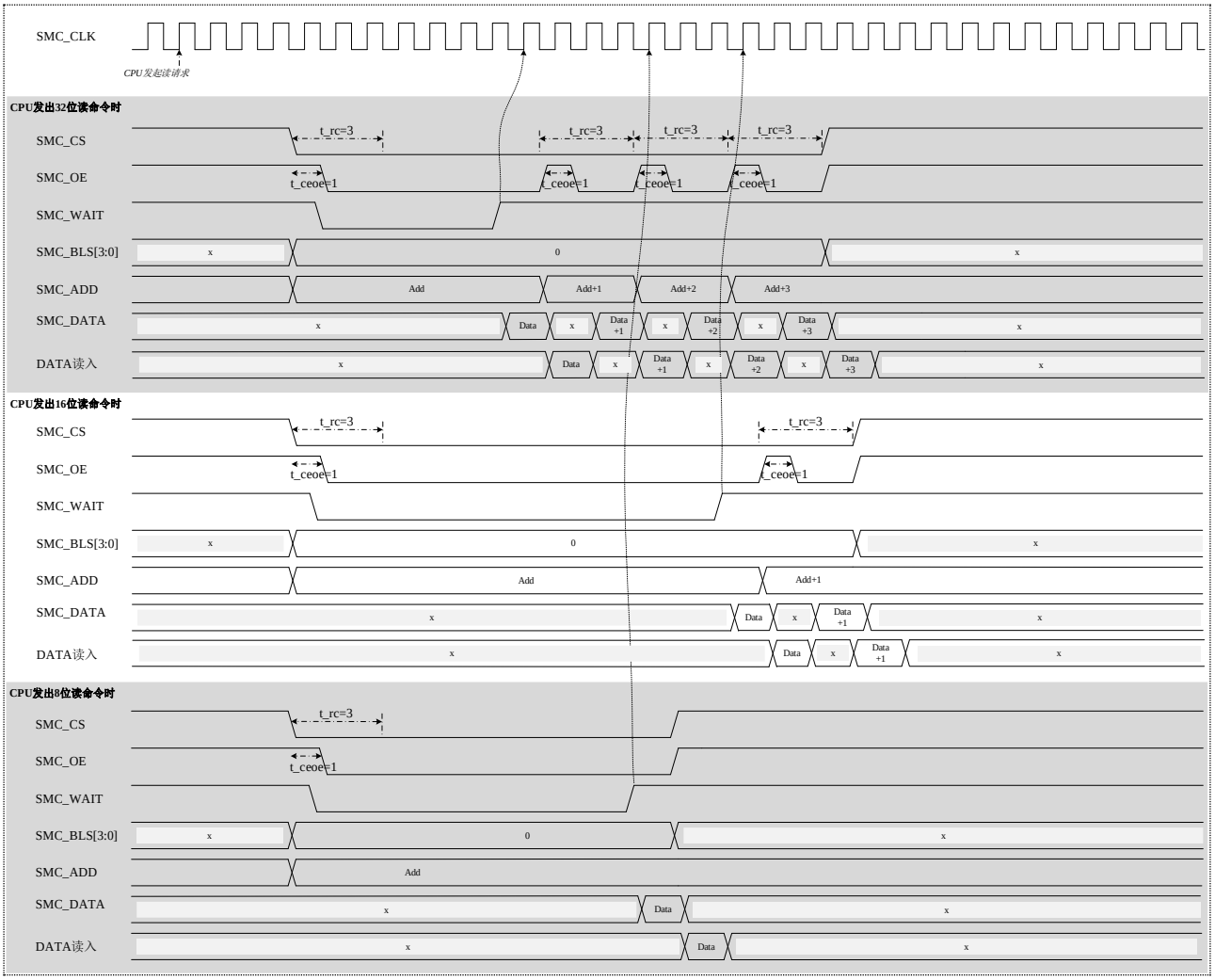


图 40-11　单次读动作基本时序（同步方式（RSYN=1）&32 位位宽（MW=10））

b) 表 40-8 和图 40-12 是地址数据线复用的单次读动作的基本时序图和设定例。

基本设定	MW	RSYN	RBL	WSYN	WBL	BAA	ADV	BLS
	<set>	<set>	b000	-	-	-	b1	b0
时序设定	t_rc	t_wc	t_ceoe	t_wp	t_pc	t_tr	- 表示不关注	
	b0110	-	b011	-	-	-	<set> 表示用户设定值	

表 40-8 地址数据线复用单次读动作基本设定例

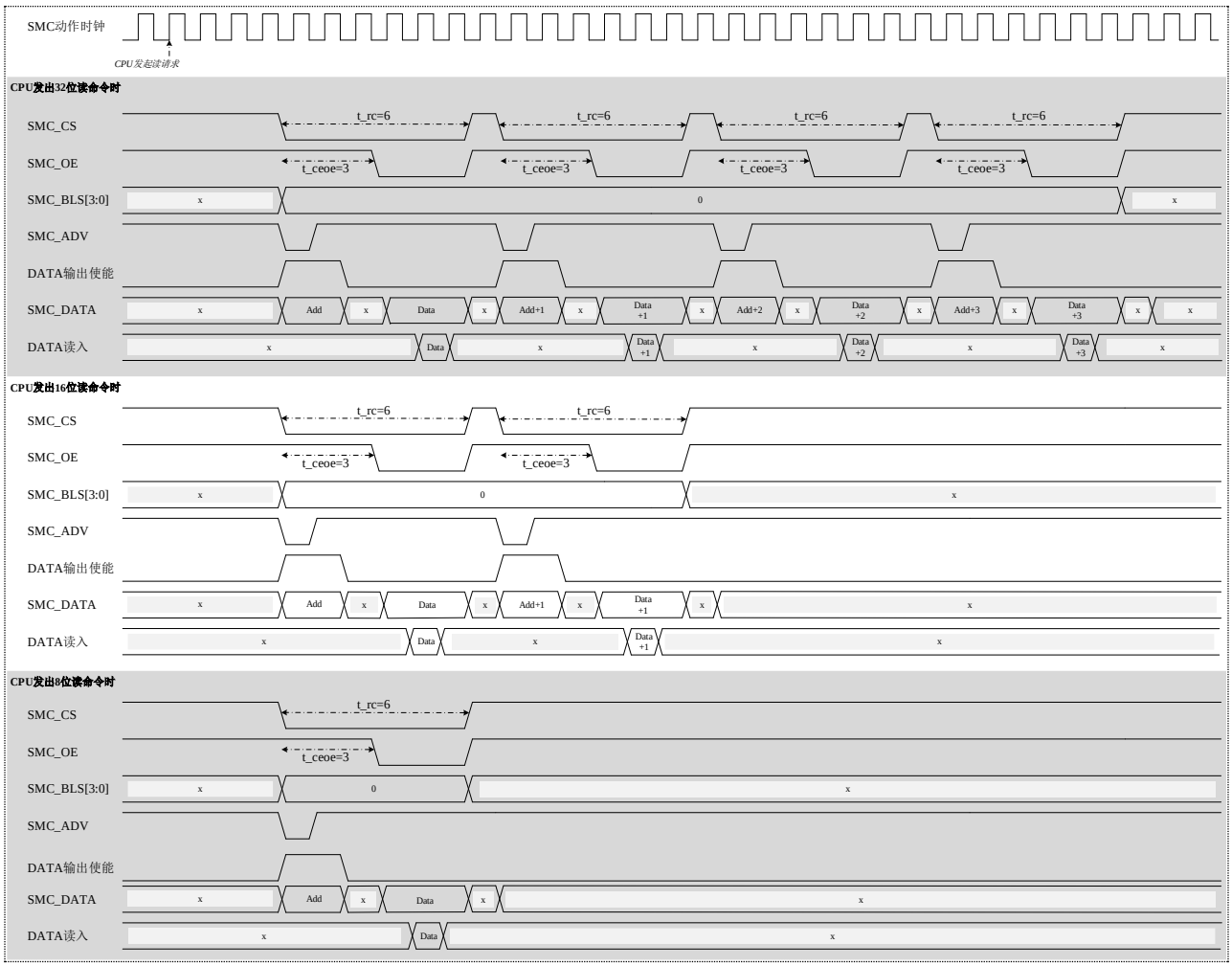


图 40-12 地址数据线复用单次读动作基本时序（异步方式（RSYN=0）&32 位位宽（MW=10））

c) 表 40-9 和图 40-13~图 40-16 是单次写动作的基本时序图和设定例。

基本设定	MW	RSYN	RBL	WSYN	WBL	BAA	ADV	BLS
	<set>	-	-	<set>	b000	-	b0	<set>
时序设定	t_rc	t_wc	t_ceoe	t_wp	t_pc	t_tr	- 表示不关注	
	-	b0100	-	b010	-	-	<set> 表示用户设定值	

表 40-9 单次写动作基本设定例

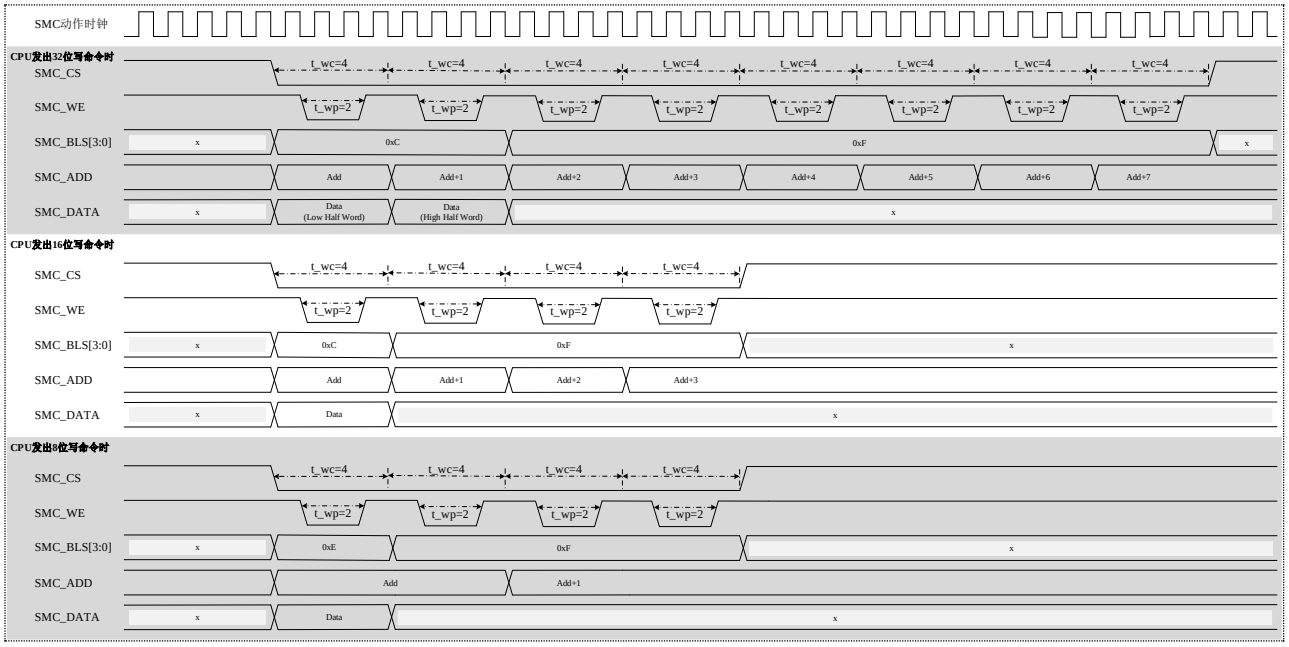


图 40-13 单次写动作基本时序（异步方式（WSYN=0）&16 位位宽（MW=01）&BLS=0）

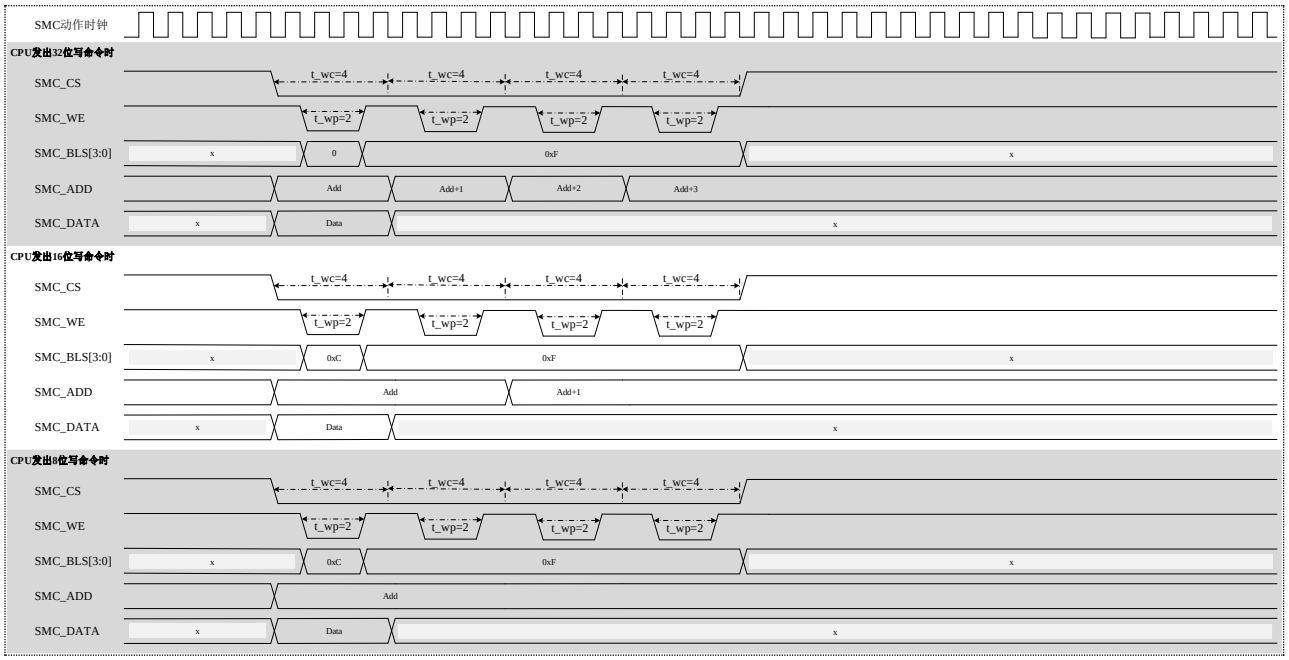


图 40-14 单次写动作基本时序（异步方式（WSYN=0）&32 位位宽（MW=10）&BLS=1）

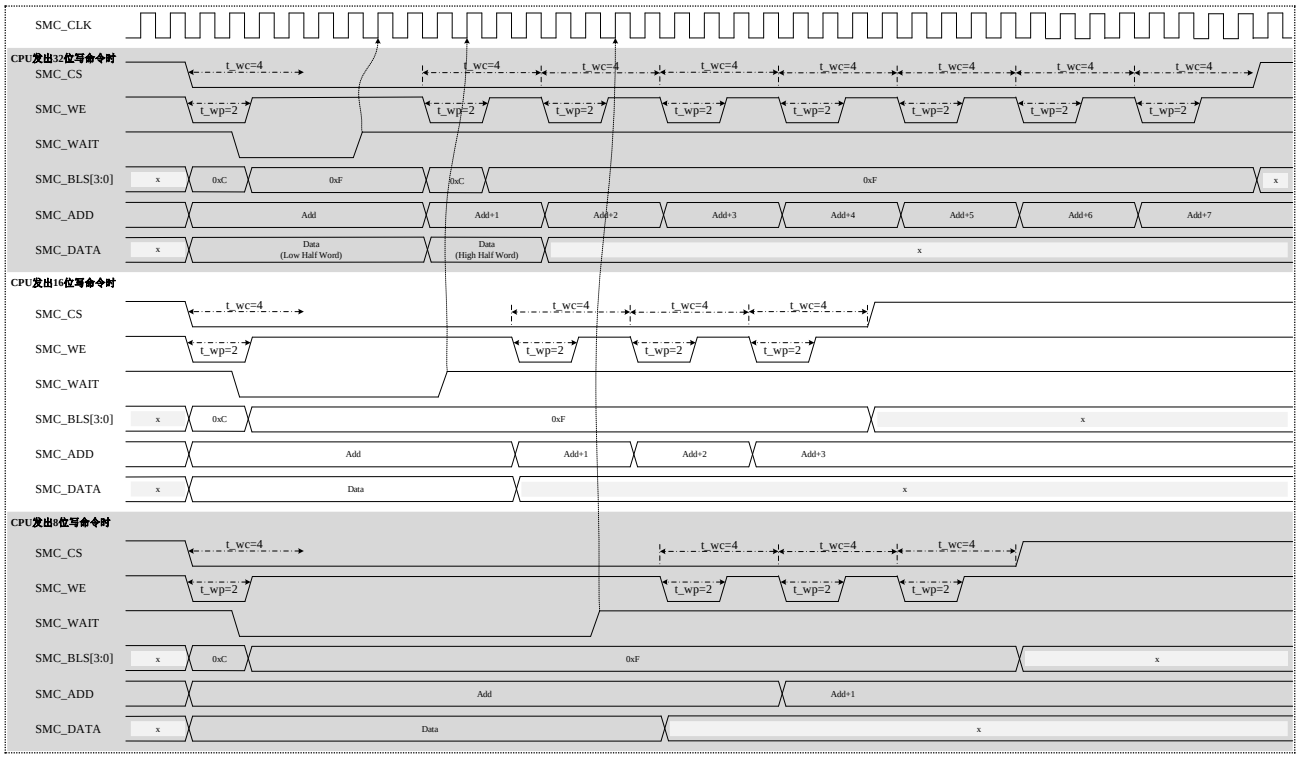


图 40-15　单次写动作基本时序（同步方式（WSYN=1）&16 位位宽（MW=01）&BLS=1）

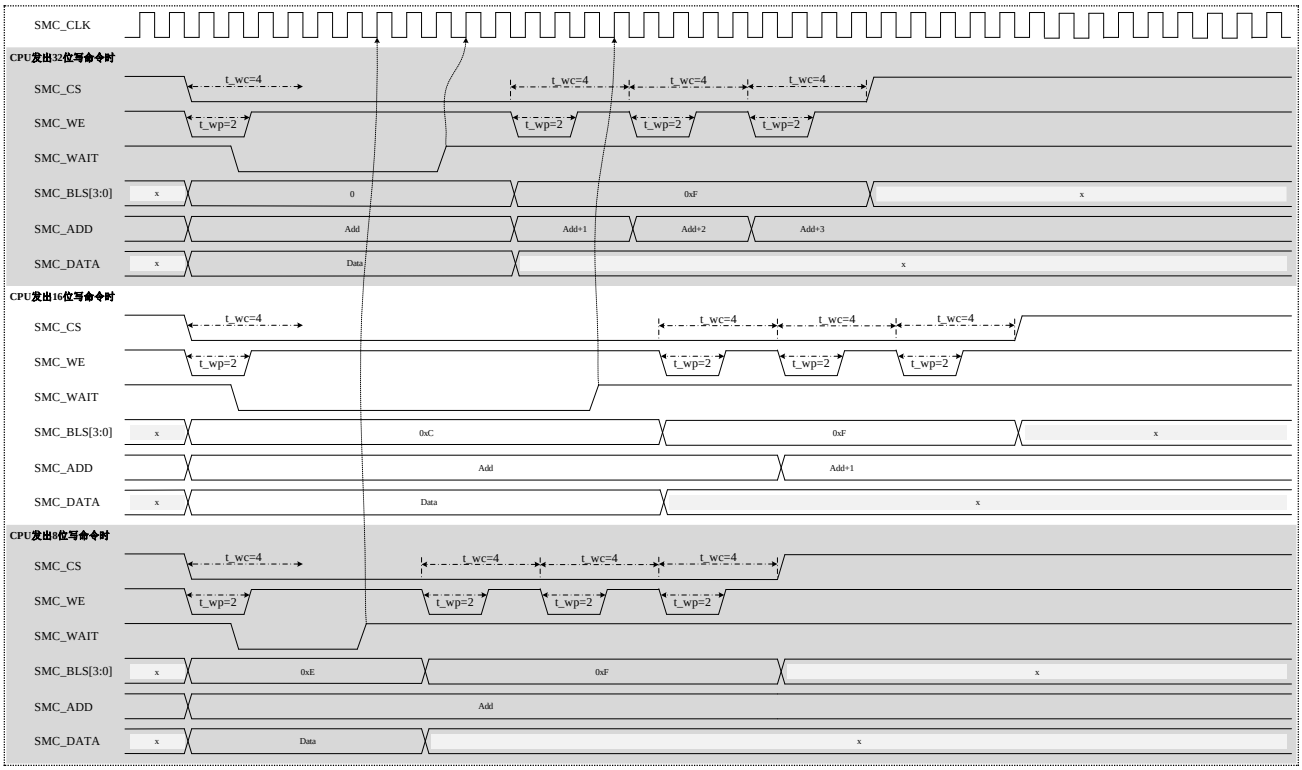


图 40-16　单次写动作基本时序（同步方式（WSYN=1）&32 位位宽（MW=10）&BLS=0）

d) 表 40-10 和图 40-17 是地址数据线复用的单次写动作的基本时序图和设定例。

基本设定	MW	RSYN	RBL	WSYN	WBL	BAA	ADV	BLS
	<set>	-	-	<set>	b000	b0	b1	b1
时序设定	t_rc	t_wc	t_ceoe	t_wp	t_pc	t_tr	- 表示不关注	
	-	b0101	-	b010	-	-	<set> 表示用户设定值	

表 40-10 地址数据线复用单次写动作基本设定例

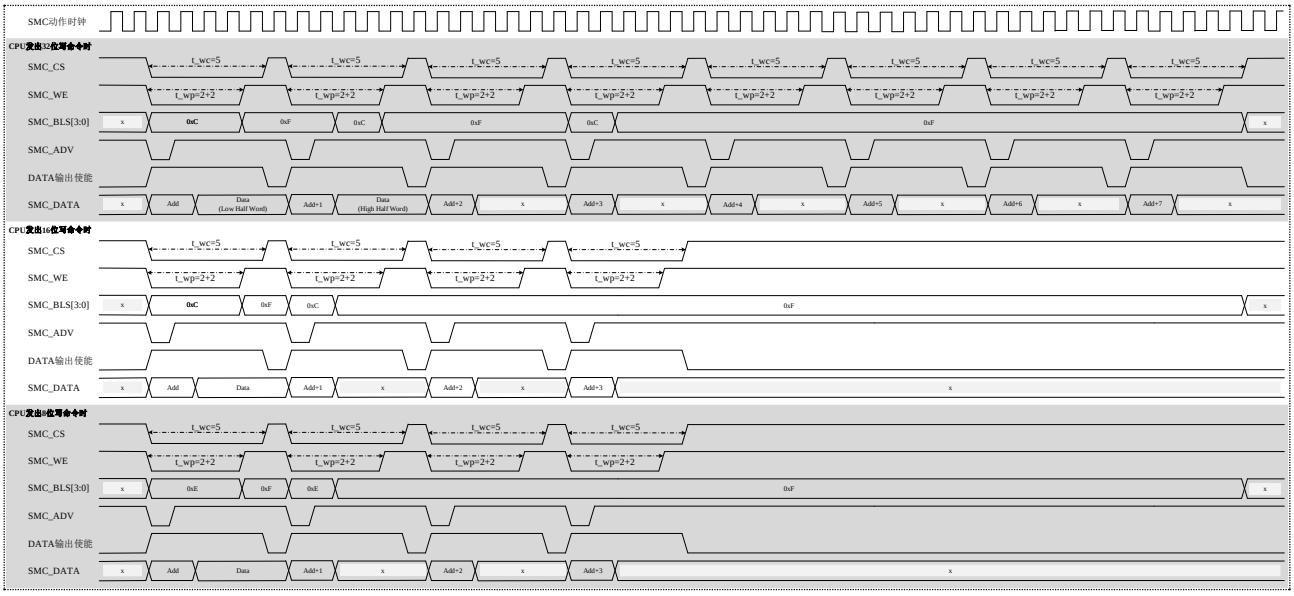


图 40-17 地址数据线复用单次写动作基本时序（异步方式（WSYN=0）&16 位位宽（MW=01））

e) 表 40-11 和图 40-18 是突发读动作的基本时序图和设定例。

基本设定	MW	RSYN	RBL	WSYN	WBL	BAA	ADV	BLS
	<set>	<set>	<set>	-	-	-	-	b0
时序设定	t_rc	t_wc	t_ceoe	t_wp	t_pc	t_tr	- 表示不关注	
	b011	-	b001	-	-	-	<set> 表示用户设定值	

表 40-11 突发读动作基本设定例

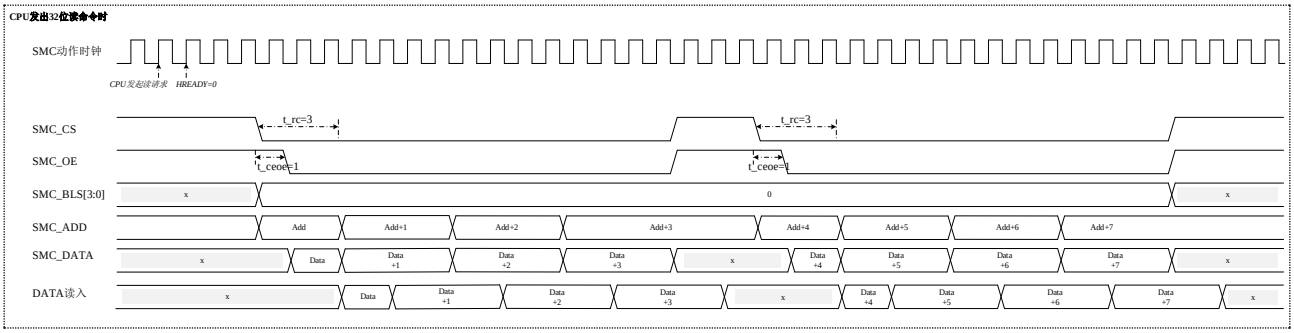


图 40-18 突发读动作基本时序（异步方式（RSYN=0）&32 位位宽（MW=10）&RBL=010）

f) 表 40-12 和图 40-19 是突发写动作的基本时序图和设定例。

基本设定	MW	RSYN	RBL	WSYN	WBL	BAA	ADV	BLS
	<set>	-	-	<set>	<set>	-	-	b1
时序设定	t_rc	t_wc	t_ceoe	t_wp	t_pc	t_tr	- 表示不关注	
	-	b0011	-	b001	-	-	<set> 表示用户设定值	

表 40-12 突发写动作基本设定例

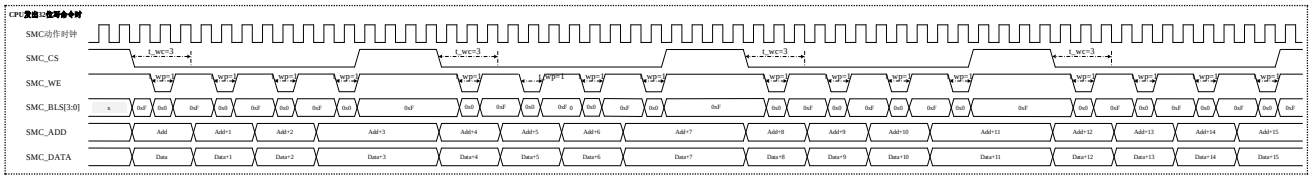


图 40-19 突发写动作基本时序（异步方式（WSYN=0）&32 位位宽（MW=10）&WBL=011）

FIFO管理

SMC 中含有 3 类 FIFO。分别是命令 FIFO、写数据 FIFO、读数据 FIFO。

命令 FIFO：2 个命令 FIFO，用于缓存 2 次 SMC 命令及对应的命令地址、CS 等信息。

写数据 FIFO：4 个写数据 FIFO，用于缓存 AHB 的写数据。

读数据 FIFO：4 个读数据 FIFO，用于缓存从 SRAM/PSRAM/NOR Flash 读入的数据。

40.3.1.5 SMC 低功耗管理

SMC 处于 Low Power 状态时，如果将状态控制寄存器（SMC_STCR1）的 LPWOR 位设定为 1，则 SMC 从 Low Power 状态转为 Ready 状态，SMC 开始正常工作；如果对 SMC 长时间无操作，可将状态控制寄存器（SMC_STCR0）的 LPWIR 位设定为 1，使 SMC 进入 Low Power 状态，进入 Low Power 状态后，SMC 的内部动作停止，功耗会降低。

40.3.2 DMC-SDRAM 控制器

40.3.2.1 SDRAM 介绍

同步动态随机存储器（SDRAM）是基于外部同步时钟刷新的动态随机存储器（DRAM），它的同步时钟由 DMC 的 DMC_CLK 引脚提供。

Bank特性

SDRAM 内部可以分为多个 Bank（EXMC 支持最大 4 个 Bank 的存储器），允许设备以交错的方式进行访问，以获取更大的并发性和数据传输量。每个 Bank 是一个存储器矩阵，矩阵由行和列构成，行列的选择定位一个存储单元，存储单元上对应存储器存储宽度的空间，因此 SDRAM 存储器的每个 Bank 的大小可以认为是存储器数据宽度 x 行数 x 列数。用户可以通过片选控制寄存器（DMC_CSCR）的相关设定（行地址长度、列地址长度等）使 DMC 和外部 SDRAM 的参数匹配，实现与不同 SDRAM 的通信。

行激活

行激活是将行地址所在的 Bank 使能，完整的行地址由 2 Bit 的 Bank 地址（DMC_BA[1:0]）和 16 Bit 的行地址（DMC_ADD[15:0]）组成。行激活会将所选行的所有比特信息读入读写放大器，一旦行激活完成，读写操作就可以顺利的进行。

行激活需要一定的时间，这个时间间隔叫做行列延时，它是行寻址到列寻址的最小时间间隔。DMC 的行列延时（t_rcd），是包含 SDRAM 行列延时的最小时钟周期数，它代表了行使能到 SDRAM 读写间的最小等待时间，可通过时序配置寄存器（DMC_TMCR_t_rcd）来设定该时间。在这段时间中，由于 SDRAM 控制器对 Bank 的操作是独立进行的，用户可向其它的 Bank 地址发出控制命令。图 40-20 是行激活到读写操作的时序图。

注：行激活在 EXMC 章节的时序图中都表示为“Active”。

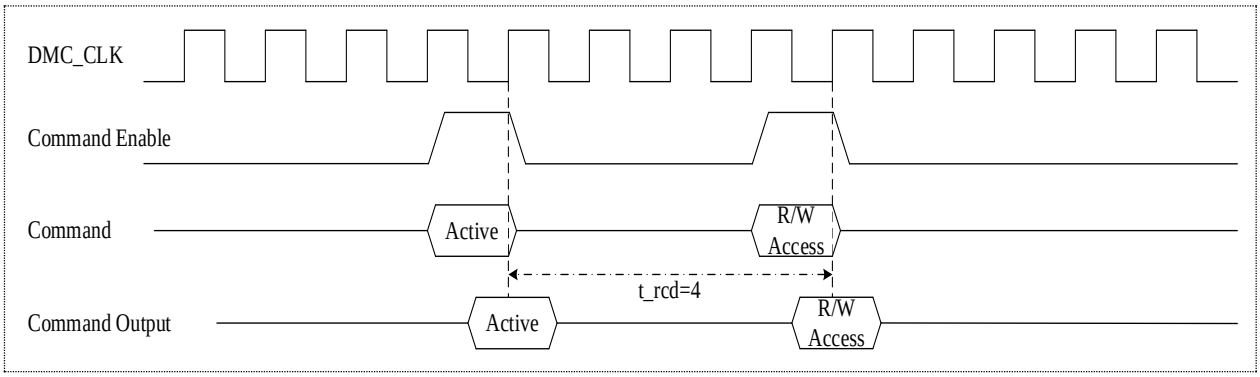


图 40-20 行激活到读写操作时序

预充电

如果 DMC 在访问外部数据时需要进行行切换，就需要将该 Bank 原来有效（工作）的行关闭，重新发送行列地址。Bank 关闭现有工作行，准备打开新行的操作就是预充电。预充电动作可以通过命令寄存器（DMC_CMDR）的预充电命令（PrechargeAll）触发，或者在读写换行时触发。

在突发读或突发写命令中，A10/A8（由 DMC_CPCR.APBS 设定）位变为“H”，在读写操作完成后自动附加一个预充电动作，实现自动预充电。

行预充电延时（t_rp）代表 SDRAM 行切换的最小时间，它是预充电完成到下一次行使能命令的最小时间间隔，可通过时序配置寄存器（DMC_TMCR_t_rp）来设定该时间。

下图 40-21、图 40-22 是两种不同方式的预充电的时序图。

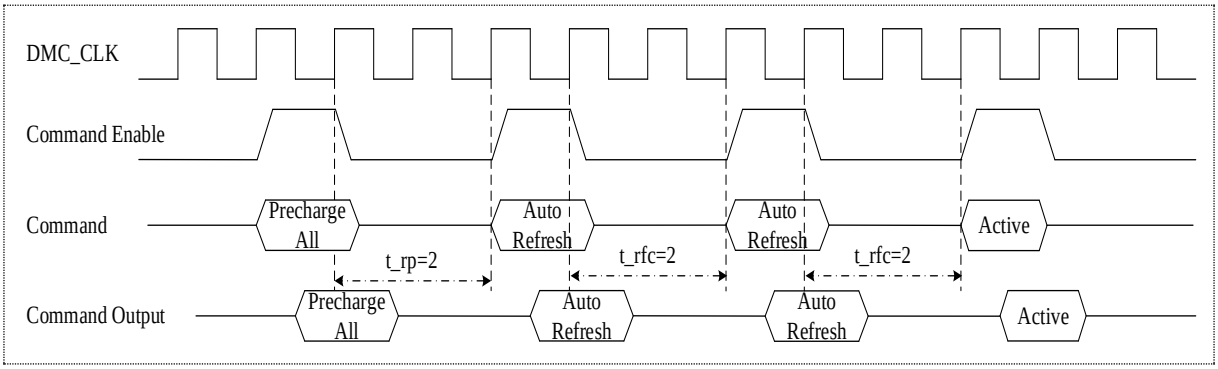


图 40-21 命令方式预充电

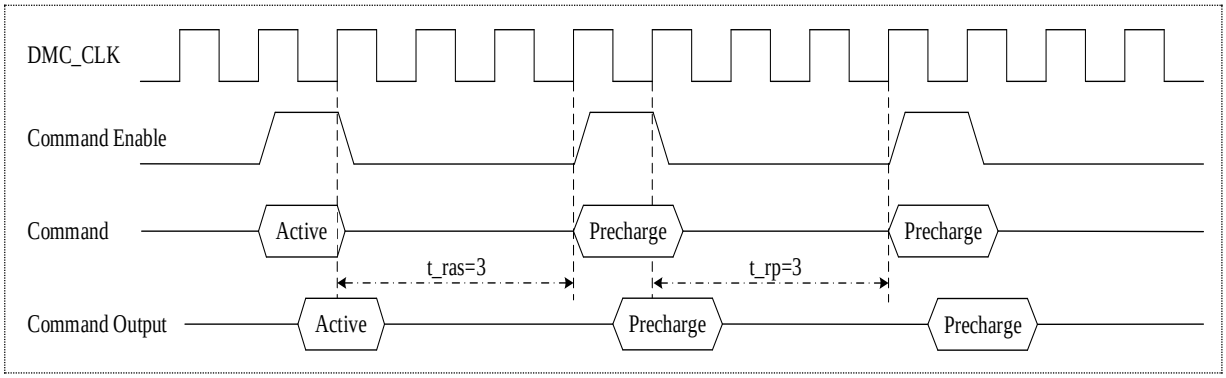


图 40-22 读写后自动预充电

刷新动作

由于动态存储器的存储单元存在漏电现象，为了保持每个存储单元数据的正确性，必须保证对所有的存储单元能定期实现刷新。EXMC 支持两种刷新方式，自动刷新方式（Auto Refresh，简称 AR）和自刷新方式（Self Refresh，简称 SR）。

自动刷新（AR）是 EXMC 周期性的提供刷新命令，刷新间隔由刷新时间寄存器（DMC_RFTR）决定。刷新对象为所有 Bank（所有 Bank 的对应行同时刷新），且只有在所有 Bank 都空闲并且未处于低功耗状态时才启动自动刷新操作。自动刷新执行期间只能输入空操作，其它操作指令只能等待而无法执行。自动刷新执行完毕后所有 Bank 都进入空闲状态。

自刷新（SR）主要用于休眠模式低功耗状态下的数据保存。在 AR 命令时 DMC_CKE 处于无效状态就进入了自刷新方式。SR 时 SDRAM 不再依靠 DMC_CLK 时钟工作，而是根据 SDRAM 内部的时钟进行刷新操作，在 SR 期间除了 DMC_CKE 之外的所有外部信号都是无效的（无需外部提供刷新指令）。DMC 在从 Pause 状态变为 Low Power 状态（通过发出 Sleep 命令）时，会自动发出自刷新指令。图 40-23 是自刷新动作的进入和退出时序图。

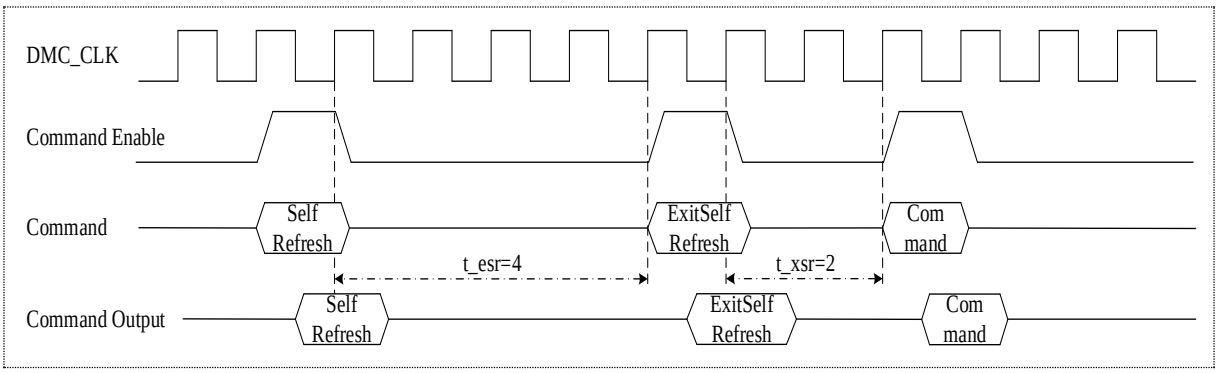


图 40-23 自刷新动作的进入和退出时序

命令真值表

对 SDRAM 进行控制和访问的各种命令都是由 DMC 的 SDRAM 协议接口发出，其具体的命令列表如下表所示。

DMC_CS	DMC_RAS	DMC_CAS	DMC_WE	DMC_BA	DMC_ADD		命令
					ADD[xx]	ADD[10]/[8]	
0	0	0	0	0	Config		MdRegConfig 命令
0	0	0	1	-	-	-	SelfRefresh 命令（DMC_CKE=0）
							AutoRefresh 命令（DMC_CKE=1）
0	0	1	0	-	-	1	PrechargeAll 命令
				Bank	-	0	指定 Bank 的 Prechagre 命令
0	0	1	1	Bank	Row	Row	行激活命令
0	1	0	0	Bank	Col	1	写动作时 Precharge 命令
				Bank	Col	0	写动作命令
0	1	0	1	Bank	Col	1	读动作时 Precharge 命令
				Bank	Col	0	读动作命令
0	1	1	0	-	-	-	中止突发传送
0	1	1	1	-	-	-	无操作
1	-	-	-	-	-	-	命令禁止

表 40-13 DMC 的命令真值表

40.3.2.2 DMC 基本功能

DMC 基本功能特性如下：

- 支持 16 位、32 位外部存储器数据带宽
- 支持多达 16 位行地址、12 位列地址、2 位内部 Bank 地址
- AHB 字、半字、字节访问
- 为每个存储器 Chip 提供独立的片选控制
- 每个存储器 Chip 大小可独立配置
- 字节选择信号输出
- 自动进行行和 Bank 边界管理
- 可编程的协议时序参数
- 可编程速率的自动刷新动作
- 具有 2 个 41 位的命令 FIFO
- 具有 10 个 36 位的写数据 FIFO
- 具有 10 个 32 位的读数据 FIFO
- 低功耗管理

40.3.2.3 DMC 初始设定

DMC 在上电复位之后的状态图及状态之间切换图如图 40-24 所示。初始复位时，DMC 进入 Config 状态，此时若通过状态控制寄存器（DMC_STCR）发出 Go 命令，之后无操作时，DMC 自动进入 Low Power 状态；正常动作期间，DMC 的状态切换可通过状态控制寄存器（DMC_STCR）的设定来控制，具体请参考【DMC 低功耗管理】章节。

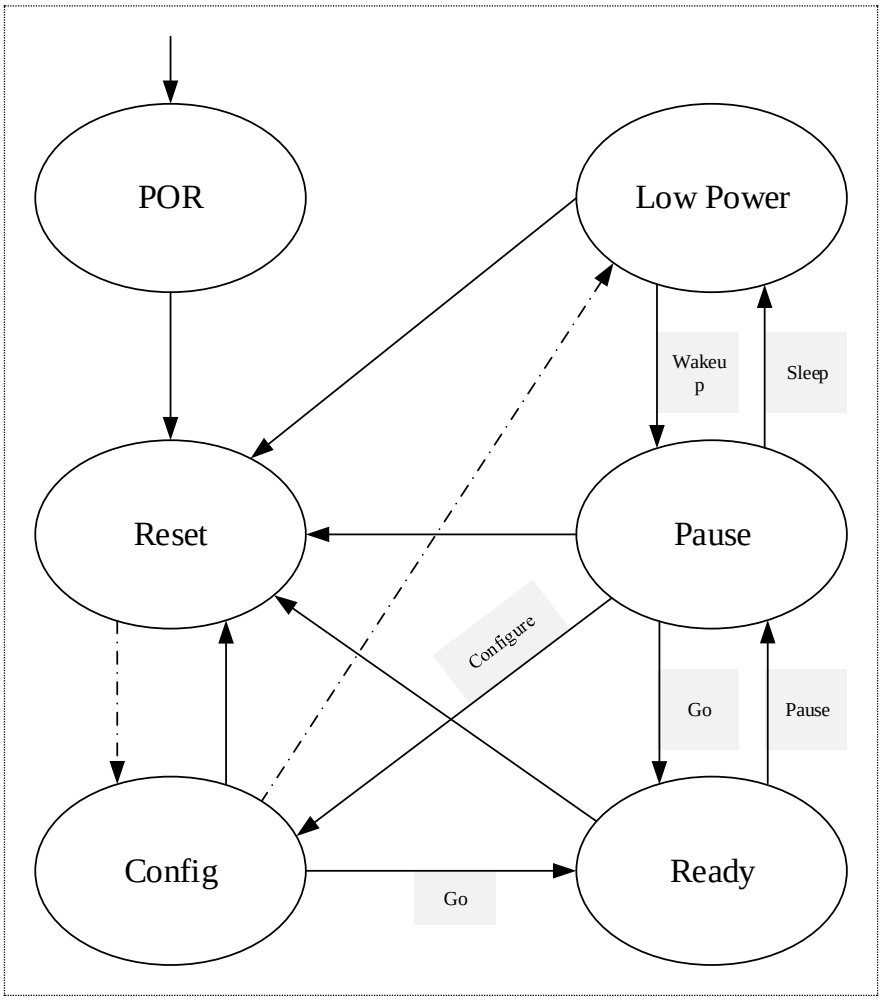


图 40-24 DMC 状态图

DMC 在和外部 SDRAM 进行通信之前，必须进行初始设定，配置好相关的参数，以保证能进行正确的数据交换。具体的设定顺序参考下图 40-25。

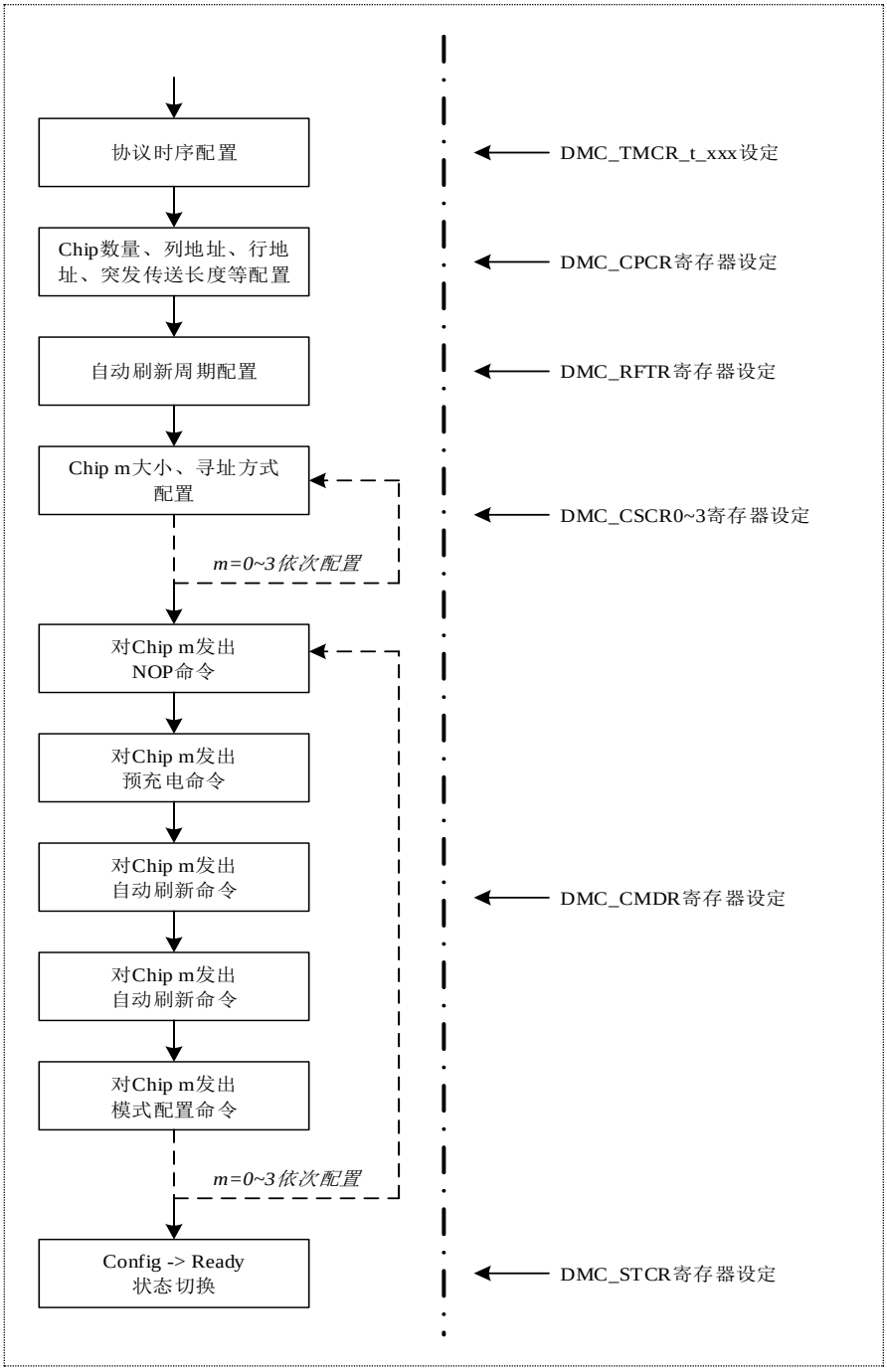


图 40-25 DMC 初始设定流程

40.3.2.4 DMC 访问动作

访问时序

DMC 可以把 AHB 的单次或突发读写操作转换成存储器的读写操作。与外部 SDRAM 存储器进行数据读写的方式有下面几种：

a. 单次读动作

b. 单次写动作

c. 突发读动作

d. 突发写动作

a) 图 40-26 是单次读动作的基本时序图。

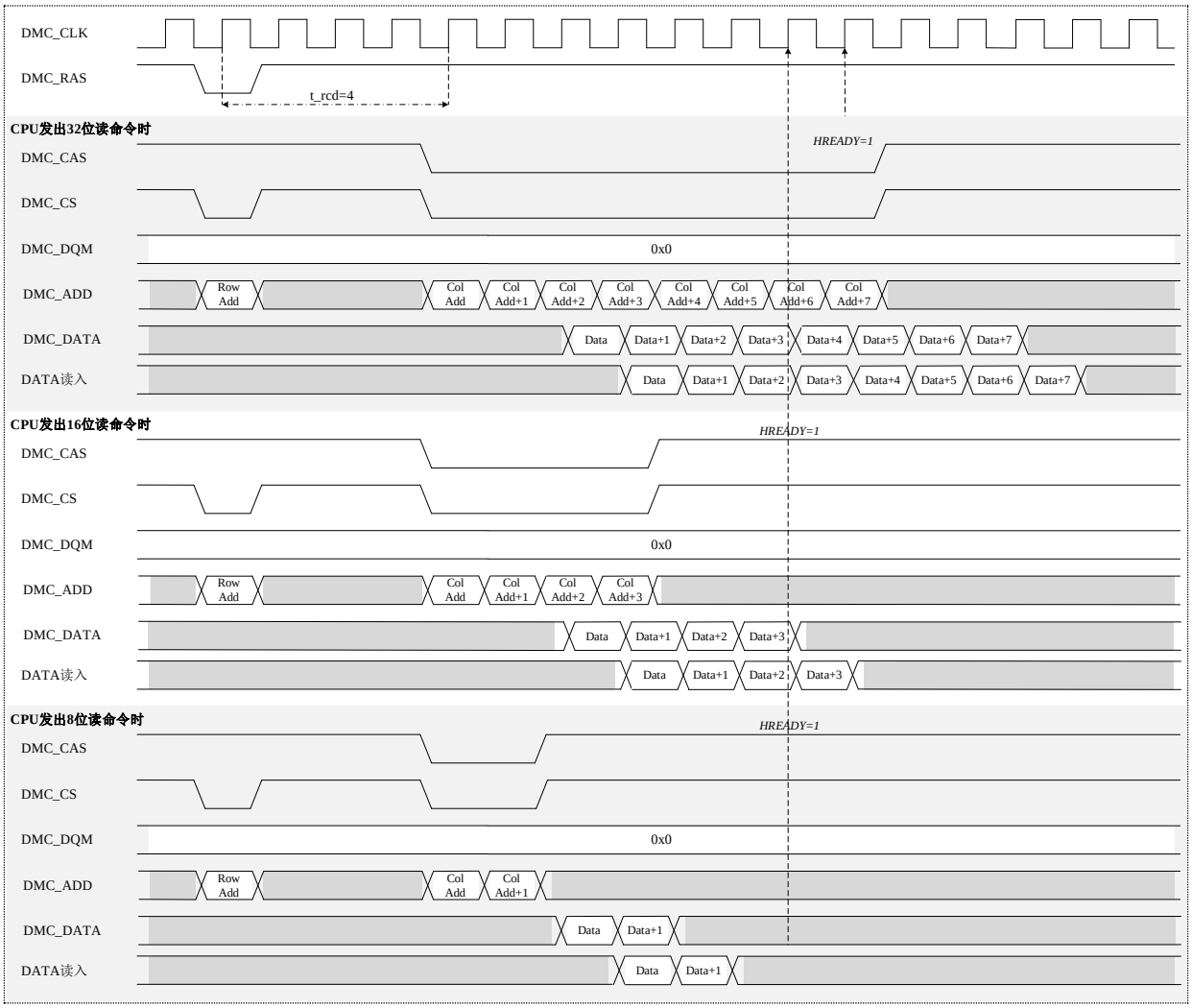


图 40-26 单次读动作基本时序（16 位位宽（DMCMW=00））

b) 图 40-27 是单次写动作的基本时序图。

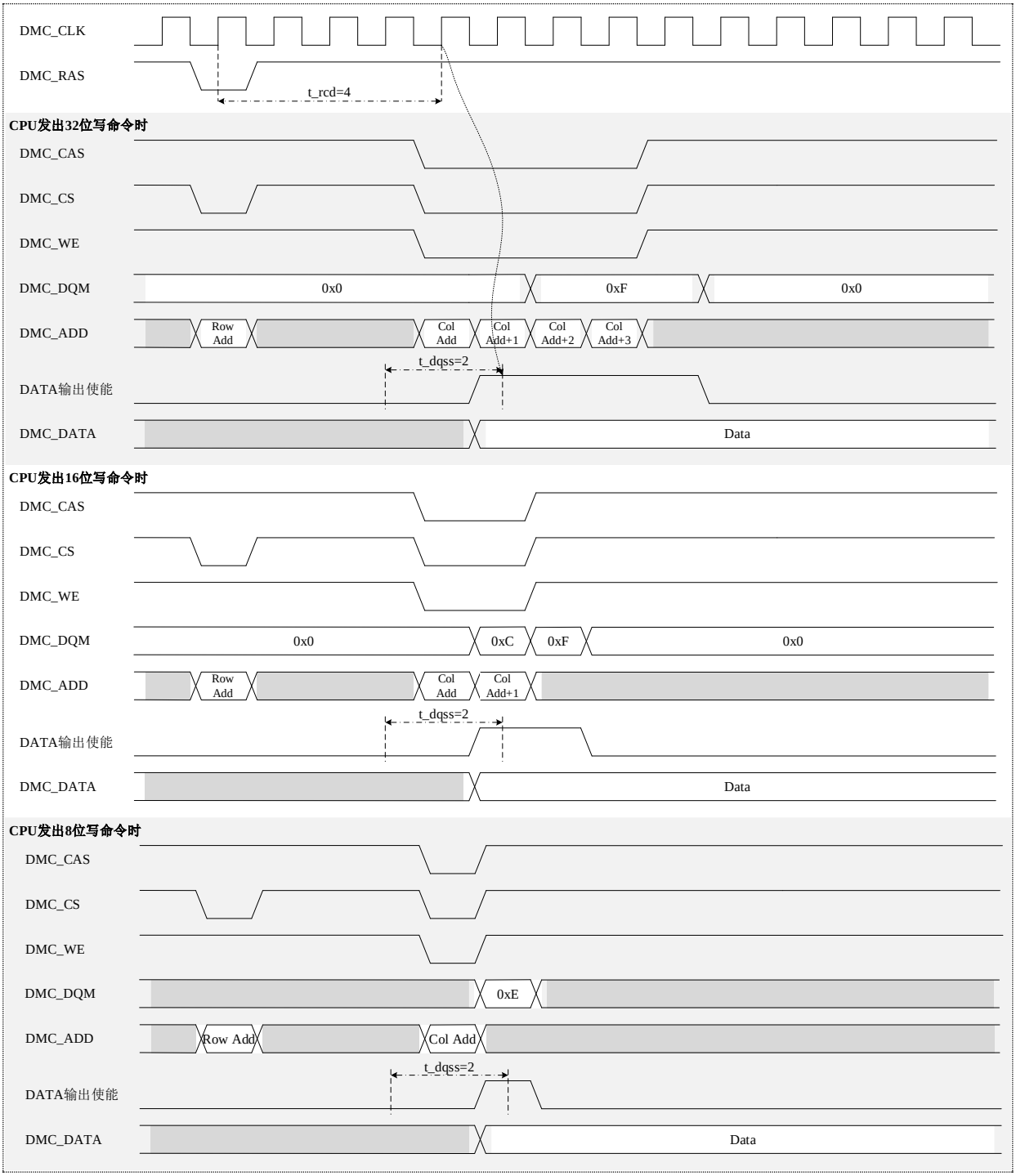


图 40-27 单次写动作基本时序（32 位位宽（DMCMW=01））

c) 图 40-28 是突发读动作的基本时序图。

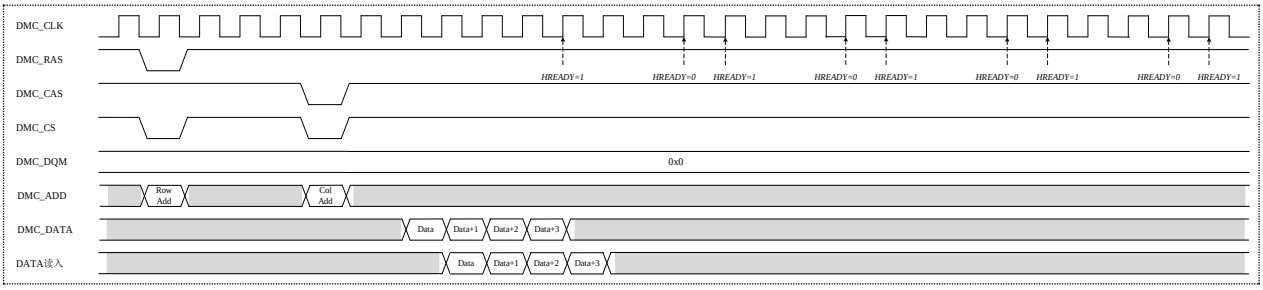


图 40-28 突发读动作基本时序（32 位位宽（DMCMW=10）&BURST=010）

d) 图 40-29 是突发写动作的基本时序图。

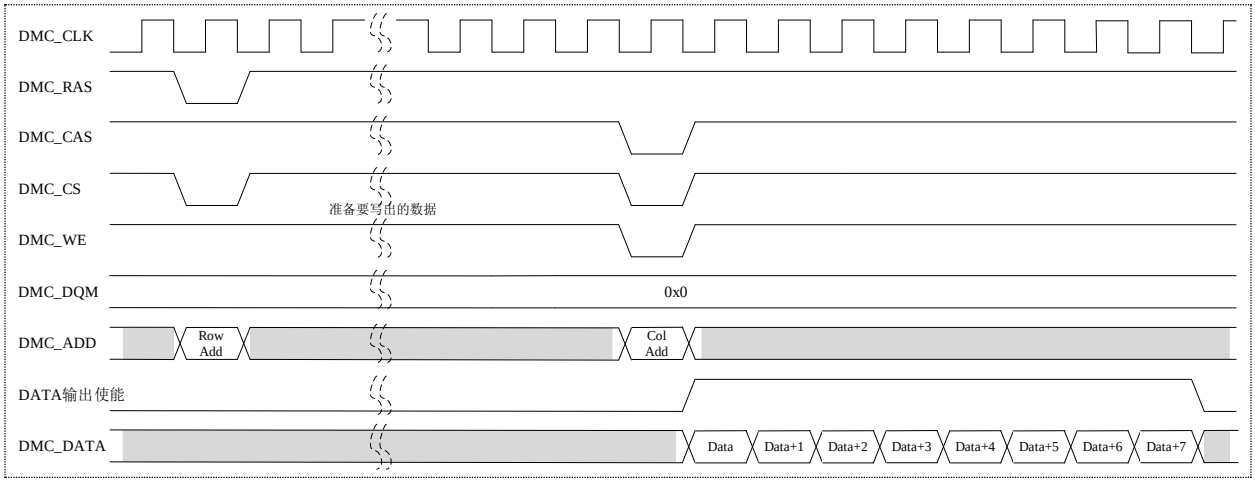


图 40-29 突发写动作基本时序（32 位位宽（DMCMW=10）&BURST=011）

FIFO管理

DMC 中含有 3 类 FIFO。分别是命令 FIFO、写数据 FIFO、读数据 FIFO。

命令 FIFO：2 个命令 FIFO，用于缓存 2 次 DMC 命令及对应的命令地址、CS 等信息。

写数据 FIFO：10 个写数据 FIFO，用于缓存 AHB 的写数据。

读数据 FIFO：10 个读数据 FIFO，用于缓存从 SDRAM 读入的数据。

40.3.2.5 DMC 低功耗管理

DMC 内部可以在 Ready 状态和 Low Power 状态之间切换，实现 DMC 内部的功耗管理；同时可以控制空闲状态时 DMC 的输出时钟或时钟使能，以实现对外部 SDRAM 存储器的功耗管理。

Low Power状态控制

DMC 处于 Config 状态时，通过状态控制寄存器（DMC_STCR）依次发出 Go 命令、Pause 命令和 Sleep 命令后，DMC 就进入 Low Power 状态。进入 Low Power 状态后，

DMC 的内部动作停止，功耗会降低。在 Low Power 状态时，通过状态控制寄存器（DMC_STCR）依次发出 Wakeup 命令和 Go 命令后，DMC 就从 Low Power 状态退出，进入 Ready 状态开始正常动作。

DMC_CLK输出控制

CHIP 配置寄存器（DMC_CPCR）的 CKSTOP 位设定为 1 时，DMC 在对外部 SDRAM 无操作时，DMC_CLK 端口的时钟输出会自动停止；当有新的操作时，DMC_CLK 的时钟会重新开启，动作完成之后，DMC_CLK 又自动停止，从而实现了在空闲状态时对时钟 DMC_CLK 的输出控制，进而达到控制 DMC 及 SDRAM 功耗的目的。

DMC_CKE输出控制

CHIP 配置寄存器（DMC_CPCR）的 CKEDIS 位设定为 1 时，当 DMC 对外部 SDRAM 执行一次操作后，经过 DMC_CPCR.CKEDISPRD[5:0]的设定周期后，DMC 变为空闲状态，DMC_CKE 的端口输出就变为无效；当有新的操作时，DMC_CKE 会重新变为有效，动作完成之后，DMC_CKE 又变为无效，从而实现了在空闲状态时对时钟输出使能 DMC_CKE 的控制，进而达到控制 SDRAM 功耗的目的。图 40-30 是 DMC_CKE 端口输出控制的时序图。

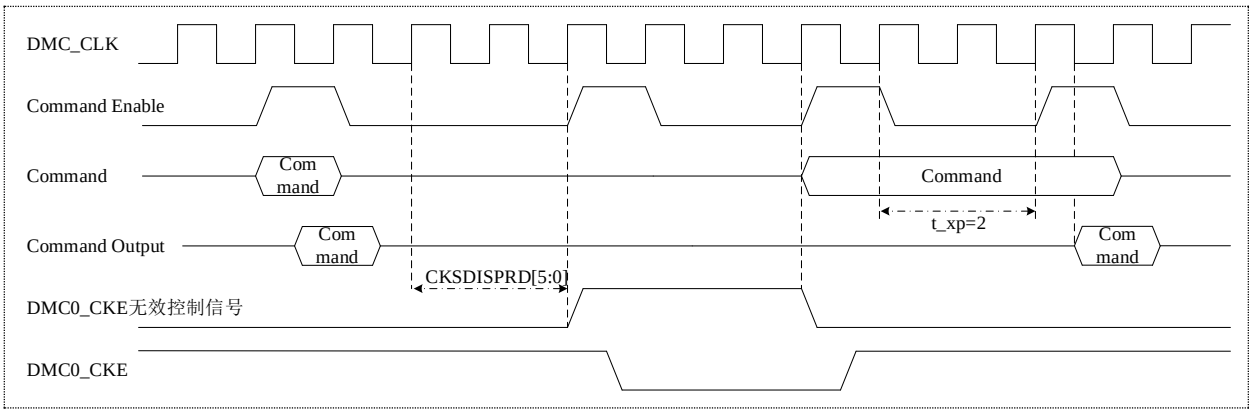


图 40-30 DMC_CKE 端口输出控制时序

40.3.3 NFC-NAND Flash 控制器

40.3.3.1 NAND Flash 介绍

NAND Flash 内存是 Flash 内存的一种，其内部采用非线性宏单元模式，为固态大容量内存的实现提供了廉价有效的解决方案。NAND Flash 存储器具有容量较大，改写速度快等优点，适用于大量数据的存储，因而在业界得到了越来越广泛的应用，如嵌入式产品中包括数码相机、MP3 随身听记忆卡、体积小巧的 U 盘等。

40.3.3.2 NFC 基本功能

NFC 基本功能特性如下：

- 支持 ONFI 协议，能够简单的实现对 SLC 和 MLC NAND Flash 设备的访问
- 支持页大小 2KB/4KB/8KB 的设备
- 支持 8bit 和 16bit 位宽的设备
- 支持每 512byte 为单位的 1bit 汉明码 ECC 纠错
- 能够计算每 512byte 为单位的 4bit BCH 码，并在 ECC 错误时给出伴随式
- 访问时序可通过寄存器配置

40.3.3.3 NFC 访问命令

下表列出了 NFC 支持的 ONFI 访问命令。其中“必须”表示所有 ONFI 设备必须支持的命令。“可选”表示某些 ONFI 设备能够支持的命令

命令	必须(M)/可选(O)	第一周期(command)	第二周期(confirm)
Read	M	00h	30h
Copyback read	O	00h	35h
Change read column	M	05h	E0h
Change read column enhanced	O	06h	E0h
Read cache random	O	00h	31h
Read cache sequential	O	31h	
Read cache end	O	3Fh	
Block erase	M	60h	D0h
Block erase interleaved	O	60h	D1h
Read status	M	70h	
Read status enhanced	O	78h	
Page program	M	80h	10h
Page program interleaved	O	80h	11h
Page cache program	O	80h	15h
Copyback program	O	85h	10h
Copyback program interleaved	O	85h	11h
Change write column	M	85h	
Change row address	O	85h	
Read ID	M	90h	
Read parameter page	M	ECh	
Read Unique ID	O	EDh	
Get Features	O	EEh	
Set Features	O	EFh	
Reset LUN	O	FAh	

表 40-14 ONFI 访问命令

40.3.3.4 NFC 访问动作

Page Read

1) 向命令寄存器写入 0x00000000
2) 向索引寄存器写入 NAND Flash 地址
3) 向命令寄存器写入 0x000000E0，或者对数据寄存器执行一次读操作，此时 NAND Flash 器件将拉低 RB 信号，表示此时器件正在准备数据。当 RB 信号恢复为高时，表示数据准备完毕。因此，如果使用命令寄存器写 0x000000E0 的方式，需等待至 RB 信号为高后读取数据寄存器，使用对数据寄存器执行一次读操作时，控制器将在总线上插入等待直到 RB 为高，并返回第一个数据
4) 通过读取数据寄存器得到整个 page 的数据
5) 用户不需要读取该 page 全部数据。当读完所需数据后，向命令寄存器写入 0x00000023，控制器将自动读取剩余数据并开始计算 ECC。当 ECC 计算完成时将产生中断并将标志位置位，将并将 ECC 结果写入 ECC 结果寄存器中。如果用户选择 ECC 功能无效，将省略此步骤
6) 向命令寄存器写入 0x000000FE 结束当次读操作，并将 CE 置为无效

Page Write

1) 向命令寄存器写入 0x00000080
2) 向索引寄存器写入 NAND Flash 地址
3) 依次向数据寄存器写入数据，当 ECC 有效时，写入数据的同时控制器将计算 ECC
4) 写完全部数据后，向命令寄存器写入 0x00000010，NAND Flash 器件将开始 programming，同时拉低 RB。当 ECC 有效时，如果用户写入的数据不足 1page，控制器将自动用 FF 补足剩余的地址并计算 ECC 值。并在写完数据后将 ECC 值自动写入 NAND Flash 器件的 spare area 中
5) programming 结束后，NAND Flash 器件将 RB 恢复为高。向命令寄存器写入 0x000000FE 结束此次写操作并将 CE 置为无效

Block Erase

1) 向命令寄存器写入 0x81000M60，其中 M 为 BANK 号
2) 向命令寄存器写入 0x40000MAB，其中 M 为 BANK 号，AB 为 NAND Flash row

address 的最低位的字节

3) 向命令寄存器写入 0x40000MWX，其中 M 为 BANK 号，WX 为 NAND Flash row address 的次低位的字节

4) 向命令寄存器写入 0x40000MYZ，其中 M 为 BANK 号，YZ 为 NAND Flash row address 的最高位的字节

5) 向命令寄存器写入 0x00000MD0，其中 M 为 BANK 号

6) 等待 NAND Flash 器件将 RB 恢复为高

Reset

1) 向命令寄存器写入 0x00000MFF，其中 M 为 BANK 号

2) 等待 NAND Flash 器件将 RB 恢复为高

Read ID

1) 向命令寄存器写入 0x81000M90，其中 M 为 BANK 号

2) 向命令寄存器写入 0x40000MAB，其中 M 为 BANK 号，AB 为 ID 的地址

3) 读取数据寄存器得到 ID

4) 向命令寄存器写入 0x000000FE 将 CE 置为无效

Read Status

1) 向命令寄存器写入 0x81000M70，其中 M 为 BANK 号

2) 读取数据寄存器，寄存器值的最低位字节为状态值

3) 向命令寄存器写入 0x000000FE 将 CE 置为无效

40.4 中断说明

NFC 有 3 类中断事件，由中断使能寄存器（NFC_IENR）控制是否发生相应中断，用户能够通过查询中断状态寄存器（NFC_ISTR）判断是否发生中断事件，并通过写寄存器清除中断事件。NFC 的 3 类中断事件如下：

1) ECC 计算结束时发生中断
2) ECC 校验错误时发生中断
3) 各个 RB 信号产生上升沿，即对应设备就绪时发生中断

40.5 寄存器说明

SMC：0x88000000h DMC：0x88000400h NFC：0x88100000h

寄存器名		符号	偏移量	位宽	复位值
SMC	使能寄存器	SMC_ENAR	（0x4005540Ch）	32	0x00000000h
	状态寄存器	SMC_STSR	0x0000h	32	0x00000001h
	命令寄存器	SMC_CMDR	0x0010h	32	-
	状态控制寄存器 0	SMC_STCR0	0x0008h	32	-
	状态控制寄存器 1	SMC_STCR1	0x000ch	32	-
	刷新时间寄存器	SMC_RFTR	0x0020h	32	0x00000000h
	基本控制寄存器	SMC_BACR	0x0200h	32	0x00000300h
	片选控制寄存器	SMC_CSCR0	0x0208h	32	0xFFFFFFFFh
	片选控制寄存器	SMC_CSCR1	0x020Ch	32	0x00000000h
	CHIP 配置寄存器	SMC_CPCR	0x0018h	32	-
	CHIP 状态寄存器 0	SMC_CPSR0	0x0104h	32	0x00FF0A00h
	CHIP 状态寄存器 1	SMC_CPSR1	0x0124h	32	0x00FF0A00h
	CHIP 状态寄存器 2	SMC_CPSR2	0x0144h	32	0x00FF0A00h
	CHIP 状态寄存器 3	SMC_CPSR3	0x0164h	32	0x00FF0A00h
	时序配置寄存器	SMC_TMCR	0x0014h	32	-
	时序状态寄存器 0	SMC_TMSR0	0x0100h	32	0x001263CCh
	时序状态寄存器 1	SMC_TMSR1	0x0120h	32	0x001263CCh
	时序状态寄存器 2	SMC_TMSR2	0x0140h	32	0x001263CCh
	时序状态寄存器 3	SMC_TMSR3	0x0160h	32	0x001263CCh
DMC	使能寄存器	DMC_ENAR	（0x4005540Ch）	32	0x00000000h
	状态寄存器	DMC_STSR	0x0000h	32	0x00000700h
	命令寄存器	DMC_CMDR	0x0008h	32	-
	状态控制寄存器	DMC_STCR	0x0004h	32	-
	刷新时间寄存器	DMC_RFTR	0x0010h	32	0x00000A60h
	基本控制寄存器	DMC_BACR	0x0300h	32	0x00000300h
	片选控制寄存器 0	DMC_CSCR0	0x0200h	32	0x0000FF00h
	片选控制寄存器 1	DMC_CSCR1	0x0204h	32	0x0000FF00h
	片选控制寄存器 2	DMC_CSCR2	0x0208h	32	0x0000FF00h
	片选控制寄存器 3	DMC_CSCR3	0x020ch	32	0x0000FF00h
	CHIP 配置寄存器	DMC_CPCR	0x000ch	32	0x00020040h

	时序配置寄存器	DMC_TMCR_t_casl	0x0014h	32	0x00000003h
	时序配置寄存器	DMC_TMCR_t_dqss	0x0018h	32	0x00000001h
	时序配置寄存器	DMC_TMCR_t_mrd	0x001ch	32	0x00000002h
	时序配置寄存器	DMC_TMCR_t_ras	0x0020h	32	0x00000007h
	时序配置寄存器	DMC_TMCR_t_rc	0x0024h	32	0x0000000Bh
	时序配置寄存器	DMC_TMCR_t_rcd	0x0028h	32	0x00000035h
	时序配置寄存器	DMC_TMCR_t_rfc	0x002ch	32	0x00001012h
	时序配置寄存器	DMC_TMCR_t_rp	0x0030h	32	0x00000035h
	时序配置寄存器	DMC_TMCR_t_rrd	0x0034h	32	0x00000002h
	时序配置寄存器	DMC_TMCR_t_wr	0x0038h	32	0x00000003h
	时序配置寄存器	DMC_TMCR_t_wtr	0x003ch	32	0x00000002h
	时序配置寄存器	DMC_TMCR_t_xp	0x0040h	32	0x00000001h
	时序配置寄存器	DMC_TMCR_t_xsr	0x0044h	32	0x0000000Ah
	时序配置寄存器	DMC_TMCR_t_esr	0x0048h	32	0x00000014h
NFC	使能寄存器	NFC_ENAR	（0x4005540Ch）	32	0x00000000h
	状态寄存器	NFC_STSR	（0x40055428h）	32	0x00000000h
	控制寄存器	NFC_STCR	（0x40055408h）	32	0x00000000h
	数据寄存器	NFC_DATR	0x0000~0x7FFFh	32	不定
	命令寄存器	NFC_CMDR	0x8000h	32	0x00000000h
	索引寄存器 0	NFC_IDXR0	0x8004h	32	0x00000000h
	索引寄存器 1	NFC_IDXR1	0x8048h	32	0x00000000h
	基本配置寄存器	NFC_BACR	0x8054h	32	0x00002187h
	中断使能寄存器	NFC_IENR	0x8030h	32	0x00000000h
	中断状态寄存器	NFC_ISTR	0x8034h	32	0x00000000h
	中断结果寄存器	NFC_IRSR	0x8038h	32	0x00000000h
	时序配置寄存器 0	NFC_TMCR0	0x804Ch	32	0x03030202h
	时序配置寄存器 1	NFC_TMCR1	0x8050h	32	0x28080303h
	时序配置寄存器 2	NFC_TMCR2	0x805Ch	32	0x03050D03h
	ECC 校验寄存器	NFC_ECCR	0x8060~0x817Fh	32	不定

表 40-15 EXMC 寄存器列表

40.5.1 SMC-SRAM/PSRAM/NOR Flash 寄存器

40.5.1.1 使能寄存器（SMC_ENAR）

复位值: 0x00000000h

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved														SMCEN	-

位	标记	位名	功能	读写
b31~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	SMCEN	SMC使能	0：SMC模块的功能端口无效	R/W
			1：SMC模块的功能端口有效	
b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

40.5.1.2 状态寄存器（SMC_STSR）

复位值: 0x00000001h

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

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

位	标记	位名	功能	读写
b31~b1	Reserved	-	读出时为复位值	R
b0	STATUS	SMC当前状态	0：Ready	R
			1：Low Power	

40.5.1.3 命令寄存器（SMC_CMDR）

复位值: -

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved						CMDCHIP[2:0]			CMD[1:0]		CRES	CMDADD[19:0]			

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

位	标记	位名	功能	读写
b31~b26	Reserved	-	写入时写“0”	W
b25~b23	CMDCHIP[1:0]	CS对象设定	CMD命令的Chip对象 000：Chip 0 001：Chip 1 010：Chip 2 011：Chip 3 请不要设定其它值	W
b22~b21	CMD[1:0]	命令输入	00：无效 01：MdRegConfig 10：UpdateRegs 11：MdRegConfig & UpdateRegs	W
b20	CRES	CRE端口设定	0：SMC_CRE输出低电平 1：在发出MdregConfig命令时，SMC_CRE端口输出高电平	W
b19~b0	CMDADD[19:0]	地址对象设定	当CMD=01或11时，CMDADD[19:0]表示要访问的外部MEM地址	W

40.5.1.4 状态控制寄存器（SMC_STCR0）

复位值: -

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved													LPWIR	-	-

位	标记	位名	功能	读写
b31~b3	Reserved	-	写入时写“0”	W
b2	LPWIR	Low Power状态进入请求	0：无请求 1：Low Power状态进入	W
b1~b0	Reserved	-	写入时写“0”	W

40.5.1.5 状态控制寄存器（SMC_STCR1）

复位值: -

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved													LPWOR	-	-

位	标记	位名	功能	读写
b31~b3	Reserved	-	写入时写“0”	W
b2	LPWOR	Low Power状态退出请求	0：无请求 1：Low Power状态退出	W
b1~b0	Reserved	-	写入时写“0”	W

40.5.1.6 刷新时间寄存器（SMC_RFTR）

复位值: 0x00000000h

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

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

位	标记	位名	功能	读写
b31~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3~b0	REFPRD[3:0]	刷新周期	刷新周期值设定	R/W

40.5.1.7 基本控制寄存器（SMC_BACR）

复位值: 0x00000300h

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CKSEL[1:0]		Reserved									MUXMD	Reserved			

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b14	CKSEL[1:0]	采样时钟选择	00：读取锁存SMC_DATA、SMC_WAIT端口数据时，使用内部EXCLK 01：读取锁存SMC_DATA、SMC_WAIT端口数据时，使用内部EXCLK取反做时钟 10：读取锁存SMC_DATA、SMC_WAIT端口数据时，使用EXMC_CLK端口状态反馈做时钟 11：设定禁止	R/W
b13~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9~b8	Reserved	-	读出时为“1”,写入时写“1”	R/W
b7~b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4	MUXMD	SMC地址数据复用	0：SMC地址和数据不复用 1：SMC地址和数据复用	R/W
b3~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

40.5.1.8 片选控制寄存器（SMC_CSCR0）

复位值: 0xFFFFFFFFh

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ADDMSK3[7:0]								ADDMSK2[7:0]							

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDMSK1[7:0]								ADDMSK0[7:0]							

位	标记	位名	功能	读写
b31~b24	ADDMSK3[7:0]	CS3空间屏蔽地址	CS3空间屏蔽地址设定	R/W
b23~b16	ADDMSK2[7:0]	CS2空间屏蔽地址	CS2空间屏蔽地址设定	R/W
b15~b8	ADDMSK1[7:0]	CS1空间屏蔽地址	CS1空间屏蔽地址设定	R/W
b7~b0	ADDMSK0[7:0]	CS0空间屏蔽地址	CS0空间屏蔽地址设定	R/W

40.5.1.9 片选控制寄存器（SMC_CSCR1）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ADDMAT3[7:0]								ADDMAT2[7:0]							

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDMAT1[7:0]								ADDMAT0[7:0]							

位	标记	位名	功能	读写
b31~b24	ADDMAT3[7:0]	CS3空间匹配地址	CS3空间匹配地址设定	R/W
b23~b16	ADDMAT2[7:0]	CS2空间匹配地址	CS2空间匹配地址设定	R/W
b15~b8	ADDMAT1[7:0]	CS1空间匹配地址	CS1空间匹配地址设定	R/W
b7~b0	ADDMAT0[7:0]	CS0空间匹配地址	CS0空间匹配地址设定	R/W

40.5.1.10 CHIP 配置寄存器（SMC_CPCR）

复位值: -

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-			BLSS	ADVS	BAAS	MW[1:0]		WBL[2:0]			WSYN	RBL[2:0]			RSYN

位	标记	位名	功能	读写
b31~b16	Reserved	-	写入时写“0”	W
b15~b13	Reserved	-	写入时写“0”	W
b12	BLSS	BLS端口设定	0：BLS的端口状态和CS端口的状态变化同步 1：BLS的端口状态和WE端口的状态变化同步	W
b11	ADVS	ADV端口设定	0：ADV端口功能无效 1：地址数据线复用功能有效时，ADV端口功能有效	W
b10	BAAS	BAA端口设定	0：BAA端口功能无效 1：BAA端口功能有效	W
b9~b8	MW[1:0]	存储器位宽选择	01：16位位宽 10：32位位宽 请不要设定其它值	W
b7~b5	WBL[2:0]	突发写数据长度	000：1次写传送 001：4次连续写传送 010：8次连续写传送 011：16次连续写传送 100：32次连续写传送 101：连续写，无次数限制 请不要设定其它值	W
b4	WSYN	写同步	0：异步写使能 1：同步写使能	W
b3~b1	RBL[2:0]	突发读数据长度	000：1次读传送 001：4次连续读传送 010：8次连续读传送 011：16次连续读传送 100：32次连续读传送 101：连续读，无次数限制 请不要设定其它值	W
b0	RSYN	读同步	0：异步读使能 1：同步读使能	W

40.5.1.11 CHIP 状态寄存器（SMC_CPSR0~3）

复位值: 0xFF000A00h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ADDMAT[7:0]								ADDMSK[7:0]							

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-			BLSS	ADVS	BAAS	MW[1:0]		WBL[2:0]			WSYN	RBL[2:0]			RSYN

位	标记	位名	功能	读写
b31~b24	ADDMAT[7:0]	CS空间匹配地址	匹配地址读出	R
b23~b16	ADDMSK[7:0]	CS空间屏蔽地址	屏蔽地址读出	R
b15~b13	-	-	读出为0	R
b12	BLSS	SMC_BLS状态	SMC_BLS状态读出	R
b11	ADVS	SMC_ADV状态	SMC_ADV状态读出	R
b10	BAAS	SMC_BAA状态	SMC_BAA状态读出	R
b9~b8	MW[1:0]	存储器位宽状态	存储器位宽状态读出	R
b7~b5	WBL[2:0]	写数据长度状态	写数据长度状态读出	R
b4	WSYN	写动作方式状态	写动作方式状态读出	R
b3~b1	RBL[2:0]	读数据长度状态	读数据长度状态读出	R
b0	RSYN	读动作方式状态	读动作方式状态读出	R

40.5.1.12 时序配置寄存器（SMC_TMCR）

复位值: -

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved									t_tr[2:0]			-	t_pc[2:0]		

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	t_wp[2:0]			-	t_ceoe[2:0]			t_wc[3:0]				t_rc[3:0]			

位	标记	位名	功能	读写
b31~b23	Reserved	-	写入时写“0”	W
b22~b20	t_tr[2:0]	t_tr周期设定	t_tr设定	W
b19	Reserved	-	写入时写“0”	W
b18~b16	t_pc[2:0]	t_pc周期设定	t_pc设定	W
b15	Reserved	-	写入时写“0”	W
b14~b12	t_wp[2:0]	t_wp周期设定	t_wp设定	W
b11	Reserved	-	写入时写“0”	W
b10~b8	t_ceoe[2:0]	t_ceoe周期设定	t_ceoe设定	W
b7~b4	t_wc[3:0]	t_wc周期设定	t_wc设定	W
b3~b0	t_rc[3:0]	t_rc周期设定	t_rc设定	W

40.5.1.13 时序状态寄存器（SMC_TMSR0~3）

复位值: 0x001263CCh

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved									t_tr[2:0]			-	t_pc[2:0]		

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	t_wp[2:0]			-	t_ceoe[2:0]			t_wc[3:0]				t_rc[3:0]			

位	标记	位名	功能	读写
b31~b23	Reserved	-	读出时为“0”	R
b22~b20	t_tr[2:0]	t_tr周期状态	t_tr值读出	R
b19	Reserved	-	读出时为“0”	R
b18~b16	t_pc[2:0]	t_pc周期状态	t_pc值读出	R
b15	Reserved	-	读出时为“0”	R
b14~b12	t_wp[2:0]	t_wp周期状态	t_wp值读出	R
b11	Reserved	-	读出时为“0”	R
b10~b8	t_ceoe[2:0]	t_ceoe周期状态	t_ceoe值读出	R
b7~b4	t_wc[3:0]	t_wc周期状态	t_wc值读出	R
b3~b0	t_rc[3:0]	t_rc周期状态	t_rc值读出	R

40.5.2 DMC-SDRAM 寄存器

40.5.2.1 使能寄存器（DMC_ENAR）

复位值: 0x00000000h

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															DMCEN

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	DMCEN	DMC使能	0：DMC模块的功能端口无效	R/W
			1：DMC模块的功能端口有效	

40.5.2.2 状态寄存器（DMC_STSR）

复位值: 0x00000700h

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												MEMW[1:0]		STATUS[1:0]	

位	标记	位名	功能	读写
b31~b4	Reserved	-	读出时为复位值	R
b3~b2	MEMW[1:0]	DMC MEM位宽	00：16位	R
			01：32位	
			读出其它值无效	
			注：该位读出值与DMC_BACR.DMCMW[1:0]的设定值相同	
b1~b0	STATUS[1:0]	DMC当前状态	00：Config	R
			01：Ready	
			10：Paused	
			11：Low Power	

40.5.2.3 命令寄存器（DMC_CMDR）

复位值: -

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved										CMDCHIP[1:0]		CMD[1:0]		CMDBA[1:0]	

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	CMDADD[13:0]													

位	标记	位名	功能	读写
b31~b22	Reserved	-	写入时写“0”	W
b21~b20	CMDCHIP[1:0]	CS对象设定	CMD命令的Chip对象 00：Chip 0 01：Chip 1 10：Chip 2 11：Chip 3	W
b19~b18	CMD[1:0]	命令输入	00：PrechargeAll 01：AutoRefresh 10：MdRegConfig 11：NOP	W
b17~b16	CMDBA[1:0]	BANK对象设定	CMD（ModeConfig）命令时指定的Bank对象 00：Bank 0 01：Bank 1 10：Bank 2 11：Bank 3	W
b15~b14	Reserved	-	写入时写“0”	W
b13~b0	CMDADD[13:0]	地址对象设定	CMD（ModeConfig）命令时指定的Address对象	W

40.5.2.4 状态控制寄存器（DMC_STCR）

复位值: -

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved													STCTL[2:0]		

位	标记	位名	功能	读写
b31~b3	Reserved	-	写入时写“0”	W
b2~b0	STCTL[2:0]	状态切换控制	000：Go	W
			001：Sleep	
			010：Wakeup	
			011：Pause	
			100：Configure	
			请不要设定其它值	

40.5.2.5 刷新时间寄存器（DMC_RFTR）

复位值: 0x00000A60h

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
-	REFPRD[14:0]														

位	标记	位名	功能	读写
b31~b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14~b0	REFPRD[14:0]	刷新周期	刷新周期值设定	R/W

40.5.2.6 基本控制寄存器（DMC_BACR）

复位值: 0x00000300h

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CKSEL[1:0]		Reserved												DMCMW[1:0]	

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b14	CKSEL[1:0]	采样时钟选择	00：读取锁存DMC_DATA端口数据时，使用内部EXCLK 01：读取锁存DMC_DATA端口数据时，使用内部EXCLK取反做时钟 10：读取锁存DMC_DATA端口数据时，使用EXMC_CLK端口状态反馈做时钟 11：设定禁止	R/W
b13~b10	Reserved	-	读出时为“0”,写入时写“0”	R/W
b9~b8	Reserved	-	读出时为“1”,写入时写“1”	R/W
b7~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1~b0	DMCMW[1:0]	存储器位宽选择	00：16位位宽 01：32位位宽 请不要设定其它值	R/W

40.5.2.7 片选控制寄存器（DMC_CSCR0~3）

复位值: 0x0000FF00h

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

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

位	标记	位名	功能	读写
b31~b17	Reserved	-	读出时为“0”	R
b16	BRCx	地址译码方式	0：Row、Bank、Column方式 1：Bank、Row、Column方式 注：x=0~3	R/W
b15~b8	ADDMATx[7:0]	地址匹配	要匹配的地址 注：x=0~3	R/W
b7~b0	ADDMSKx[7:0]	地址屏蔽	要屏蔽的对应地址 注：x=0~3	R/W

注：以 ADDMAT0 和 ADDMSK0 为例，说明 DMC 的地址空间的对应方式。其它 CS 空间的设定和 CS0 设定类似。

ADDMAT0[7:0]	ADDMSK0[7:0]	对应外部地址	CS0 空间大小
86h	FFh	86000000h~86FFFFFFh	16Mb
	FEh	86000000h~87FFFFFFh	32Mb
	其它设定禁止	-	-
84h	FFh	84000000h~84FFFFFFh	16Mb
	FEh	84000000h~85FFFFFFh	32Mb
	FCh	84000000h~87FFFFFFh	64Mb
	其它设定禁止	-	-
82h	FFh	82000000h~82FFFFFFh	16Mb
	FEh	82000000h~83FFFFFFh	32Mb
	其它设定禁止	-	-
80h	FFh	80000000h~80FFFFFFh	16Mb
	FEh	80000000h~81FFFFFFh	32Mb
	FCh	80000000h~83FFFFFFh	64Mb
	F8h	80000000h~87FFFFFFh	128Mb
	其它设定禁止	-	-

表 40-16 CS0 设定和访问地址对应关系表

40.5.2.8 CHIP 配置寄存器（DMC_CPCR）

复位值: 0x00020040h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	ACTCP[1:0]		-	-	-	-	-	BURST[2:0]		

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CKEDISPRD[5:0]						CKSTOP	CKEDIS	APBS	ROWBS[2:0]			-	COLBS[2:0]		

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~b24	ACTCP[1:0]	自动刷新Chip数目	00：Chip 0 自动刷新有效 01：Chip 0~1自动刷新有效 10：Chip 0~2自动刷新有效 11：Chip 0~3自动刷新有效	R/W
b23~b19	Reserved	-	读出时为“0”,写入时写“0”	R/W
b18~b16	BURST[2:0]	突发传输长度	000：1次传送 001：2次连续传送 010：4次连续传送 011：8次连续传送 100：16连续次传送 请不要设定其它值 注：该位的设定须和外部Memory的Mode Register的对应位设定一致	R/W
b15~b10	CKEDISPRD[5:0]	时钟输出关闭周期	时钟输出关闭的等待周期设定	R/W
b9	CKSTOP	时钟停止设定	0：MEM时钟正常输出 1：无操作时，MEM时钟停止	R/W
b8	CKEDIS	CKE输出设定	0：CKE输出有效（H） 1：命令FIFO空时，经过CKEDISPRD设定的周期后，CKE输出变为无效（L）	R/W
b7	APBS	自动预充电位	0：地址线的第10位 1：地址线的第8位	R/W
b6~b4	ROWBS[2:0]	行地址宽度	000：行地址宽度为11位 001：行地址宽度为12位 010：行地址宽度为13位 011：行地址宽度为14位 100：行地址宽度为15位 101：行地址宽度为16位 请不要设定其它值	R/W
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~b0	COLBS[2:0]	列地址宽度	000：列地址宽度为8位	R/W

	001：列地址宽度为9位 010：列地址宽度为10位 011：列地址宽度为11位 100：列地址宽度为12位 请不要设定其它值

40.5.2.9 时序配置寄存器（DMC_TMCR_t_casl）

复位值: 0x00000003h

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													t_casl[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	t_casl[2:0]	t_casl周期设定	t_casl设定（CAS latency time）	R/W

40.5.2.10 时序配置寄存器（DMC_TMCR_t_dqss）

复位值: 0x00000001h

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														t_dqss	

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~b0	t_dqss[1:0]	t_dqss周期设定	t_dqss设定	R/W

40.5.2.11 时序配置寄存器（DMC_TMCR_t_mrd）

复位值: 0x00000002h

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									t_mrd[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	t_mrd[6:0]	t_mrd周期设定	t_mrd设定（mode register command time）	R/W

40.5.2.12 时序配置寄存器（DMC_TMCR_t_ras）

复位值: 0x00000007h

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												t_ras[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	t_ras[3:0]	t_ras周期设定	t_ras设定（RAS to precharge delay time）	R/W

40.5.2.13 时序配置寄存器（DMC_TMCR_t_rc）

复位值: 0x0000000Bh

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												t_rc[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	t_rc[3:0]	t_rc周期设定	t_rc设定（Active bank x to Active bank x delay time）	R/W

40.5.2.14 时序配置寄存器（DMC_TMCR_t_rcd）

复位值: 0x00000035h

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									t_rcd_p[2:0]			-	t_rcd_b[2:0]		

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	t_rcd_p[2:0]	附加t_rcd周期设定	附加t_rcd设定	R/W
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~b0	t_rcd_b[2:0]	基础t_rcd周期设定	基础t_rcd设定 t_rcd（RAS to CAS minimum delay time）定义如下： 当t_rcd_b-t_rcd_p>=3时，t_rcd=t_rcd_b；当t_rcd_b-t_rcd_p<3时，t_rcd=t_rcd_p+3	R/W

40.5.2.15 时序配置寄存器（DMC_TMCR_t_rfc）

复位值: 0x00001012h

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
-			t_rfc_p[4:0]					-			t_rfc_b[4:0]				

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~b8	t_rfc_p[4:0]	附加t_rfc周期设定	附加t_rfc设定	R/W
b7~b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4~b0	t_rfc_b[4:0]	基础t_rfc周期设定	基础t_rfc设定 t_rfc（autorefresh command time）定义如下： 当t_rfc_b-t_rfc_p>=3时，t_rfc=t_rfc_b；当t_rfc_b-t_rfc_p<3时，t_rfc=t_rfc_p+3	R/W

40.5.2.16 时序配置寄存器（DMC_TMCR_t_rp）

复位值: 0x00000035h

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									t_rp_p[2:0]			-	t_rp_b[2:0]		

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	t_rp_p[2:0]	附加t_rp周期设定	附加t_rp设定	R/W
b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2~b0	t_rp_b[2:0]	基础t_rp周期设定	基础t_rp设定 t_rp（precharge to RAS delay time）定义如下： 当t_rp_b-t_rp_p>=3时，t_rp=t_rp_b；当t_rp_b-t_rp_p<3时，t_rp=t_rp_p+3	R/W

40.5.2.17 时序配置寄存器（DMC_TMCR_t_rrd）

复位值: 0x00000002h

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												t_rrd[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	t_rrd[3:0]	t_rrd周期设定	t_rrd设定（Active bank x to Active bank y delay time）	R/W

40.5.2.18 时序配置寄存器（DMC_TMCR_t_wr）

复位值: 0x00000003h

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													t_wr[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	t_wr[2:0]	t_wr周期设定	t_wr设定（write to precharge delay time）	R/W

40.5.2.19 时序配置寄存器（DMC_TMCR_t_wtr）

复位值: 0x00000002h

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													t_wtr[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	t_wtr[2:0]	t_wtr周期设定	t_wtr设定（write to read delay time）	R/W

40.5.2.20 时序配置寄存器（DMC_TMCR_t_xp）

复位值: 0x00000001h

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								t_xp[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	t_xp[7:0]	t_xp周期设定	t_xp设定（exit power-down command time）	R/W

40.5.2.21 时序配置寄存器（DMC_TMCR_t_xsr）

复位值: 0x0000000Ah

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								t_xsr[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	t_xsr[7:0]	t_xsr周期设定	t_xsr设定（exit self-refresh command time）	R/W

40.5.2.22 时序配置寄存器（DMC_TMCR_t_esr）

复位值: 0x00000014h

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								t_esr[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	t_esr[7:0]	t_esr周期设定	t_esr设定（self-refresh command time）	R/W

40.5.3 NFC-NAND Flash 寄存器

40.5.3.1 使能寄存器（NFC_ENAR）

复位值: 0x00000000h

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													NFCEN	-	-

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	NFCEN	NFC使能	0：NFC模块的功能端口无效	R/W
			1：NFC模块的功能端口有效	
b1~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

40.5.3.2 状态寄存器（NFC_STSR）

复位值: 0x00000000h

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							PECC	CHIP_BUSY[7:0]							

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	PECC	ECC状态	0：NFC没有进行ECC计算	R
			1：NFC正在进行ECC计算	
b7~b0	CHIP_BUSY	设备状态	表示BANK0~7设备的状态	R
			0:设备正忙	
			1:设备就绪	

40.5.3.3 控制寄存器（NFC_STCR）

复位值: 0x00000000h

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															OPENP

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	OPENP	open page使能	0：不允许NFC在复位后自动发送一个读命令 1：允许NFC在复位后自动发送一个读命令	R/W

40.5.3.4 数据寄存器（NFC_DATR）

复位值: 0xXXXXXXXXh

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	数据	数据寄存器用于读写Flash设备上的数据。读写数据的地址由索引寄存器写入的地址决定	R/W

40.5.3.5 命令寄存器（NFC_CMDR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ARG[23:8]															

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

位	标记	位名	功能	读写
b31~b8	ARG	参数	用于对命令进行调整，详见表 40-18	R/W
b7~b0	CMD	命令	发送至设备的命令，详见表 40-18	R/W

下表列出了所有命令和参数的设定方法：

命令（CMD）	参数（ARG）	功能
00h	000000h	Read 命令
10h	000000h	Page program confirm
11h	000000h	用于 two-plane page program 第一个 page 结束后插入的一个 confirm
15h	000000h	Page cache program 中的第二个命令
23h	000000h	读取 ECC 状态。写入此命令后 NFC 将从 NAND Flash 的额外区域读取 ECC 数据并开始 ECC 计算
60h	810000h~810007h	Block erase 命令，参数用于指示操作对象 BANK0~BANK7
70h	810000h~810007h	Read status 命令，参数用于指示操作对象 BANK0~BANK7
80h	000000h	Page program 命令
81h	000000h	用于 two-plane page program 第二个 page 开始写入
90h	000000h~000007h	Read ID 命令，参数用于指示操作对象 BANK0~BANK7
D0h	810000h~810007h	Block erase confirm，参数用于指示操作对象 BANK0~BANK7
D1h	810000h~810007h	Block erase interleaved confirm，参数用于指示操作对象 BANK0~BANK7
E0h	000000h	开始读数据。用于指示 NFC 开始读取数据，写入该命令后，NFC 将开放 1 个 page 用于数据读取，此时 CHIP_BUSY 将置位，等待 CHIP_BUSY 复位后可以读取数据寄存器获得数据
FEh	000000h	读操作完成后写入该命令指示 NFC 将片选信号 CE 置为无效
FFh	000000h~000007h	Reset 命令，参数用于指示操作对象 BANK0~BANK7
05h	810000h~810007h	Change read column 命令的第一周期
06h	810000h~810007h	Change read column enhanced 命令的第一周期

31h	830000h~830007h	Read cache sequential 命令的第一周期和 Read cache random 命令的第二周期
32h	810000h~810007h	Copyback read interleaved 命令
35h	810000h~810007h	Copyback read 命令
3Fh	830000h~830007h	Read cache end
78h	800000h~800007h	Read status enhanced 命令
85h	810000h~810007h	Copyback program interleaved 和 Copyback program 命令
85h	840000h~840007h	Change write column 和 Change row address
E0h	850000h~850007h	Change read column 和 Change read column enhanced 命令的第二周期
ECh	830000h~830007h	Read parameter page 命令
EDh	830000h~830007h	Read Unique ID 命令
EEh	830000h~830007h	Get Features 命令
EFh	830000h~830007h	Set Features 命令
FAh	820000h~820007h	Reset LUN 命令
FCh	820000h~820007h	Asynchronous reset 命令

表 40-17 命令参数列表

40.5.3.6 索引寄存器（NFC_IDXR0）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IDX0[31:16]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
IDX0[15:0]															

位	标记	位名	功能	读写
b31~b0	IDX0	索引	组成访问地址的低32位，高8位由IDXR1寄存器组成	R/W

40.5.3.7 索引寄存器（NFC_IDXR1）

复位值: 0x00000000h

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								IDX1[7:0]							

位	标记	位名	功能	读写
b31~b8	Reserved	-	读出时为“0”	R/W
b7~b0	IDX1	索引	组成访问地址的高8位，低32位由IDXR0寄存器组成	R/W

IDXR0 和 IDXR1 共同组成了访问 NAND Flash 设备时的 40 位地址。寄存器写入的值和实际设备的地址关系取决于设备的容量。下表列出了访问各个容量设备时的地址构成：

容量	2K page			4K page			8K page		
	page	column	bank	page	column	bank	page	column	bank
512Mbit	26:12	11:0	29:27	26:13	12:0	29:27	26:14	13:0	29:27
1Gbit	27:12	11:0	30:28	27:13	12:0	30:28	27:14	13:0	30:28
2Gbit	28:12	11:0	31:29	28:13	12:0	31:29	28:14	13:0	31:29
4Gbit	29:12	11:0	32:30	29:13	12:0	32:30	29:14	13:0	32:30
8Gbit	30:12	11:0	33:31	30:13	12:0	33:31	30:14	13:0	33:31
16Gbit	31:12	11:0	34:32	31:13	12:0	34:32	31:14	13:0	34:32
32Gbit	32:12	11:0	35:33	32:13	12:0	35:33	32:14	13:0	35:33
64Gbit	33:12	11:0	36:34	33:13	12:0	36:34	33:14	13:0	36:34

表 40-18 索引寄存器值与 MEM 地址对应表

40.5.3.8 基本配置寄存器（NFC_BACR）

复位值: 0x00002187h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								SCS[7:0]							

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	RAC	ECCM[1:0]		WP	PAGE[1:0]		BANK[1:0]		-	-	16BIT	SIZE[2:0]		

位	标记	位名	功能	读写
b31~b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23~b16	SCS	spare column size	设置spare column size的大小，以word为单位	R/W
b15~b14	Reserved	-	读出时为“0”,写入时写“0”	R/W
b13	RAC	row address cycle	0：2 row address cycles 1：3 row address cycles	R/W
b12~b11	ECCM	ECC模式	00：1bitECC模式 01：4bitECC模式 其它：设定禁止	R/W
b10	WP	写保护	0：将设备置为写保护状态 1：将设备置为未写保护状态	R/W
b9~b8	PAGE	page size大小	00：设定禁止 01：2K 10：4K 11：8K	R/W
b7~b6	BANK	bank(CE)数	00：1bank 01：2bank 10：4bank 11：8bank	R/W
b5~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	16BIT	数据位宽	0：8bit 1：16bit	R/W
b2~b0	SIZE	设备容量	000：16Gbit 001：32Gbit 010：64Gbit 011：512Mbit 100：1Gbit 101：2Gbit 110：4Gbit 111：8Gbit	R/W

40.5.3.9 中断使能寄存器（NFC_IENR）

复位值: 0x00000000h

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
RBEN[7:0]								ECC DIS	ECCE EN	-	ECCC EN	-	-	ECCE CEN	ECCE UEN

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b8	RBEN	就绪中断使能	0：当访问设备就绪(RB信号置位)时不允许产生中断 1：当访问设备就绪(RB信号置位)时允许产生中断	R/W
b7	ECCDIS	ECC禁用	0：启用ECC功能 1：禁用ECC功能	R/W
b6	ECCEEN	ECC错误中断使能	0：ECC错误发生时不产生中断 1：ECC错误发生时产生中断	R/W
b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4	ECCCEN	ECC计算完成中断使能	0：ECC计算完成时不产生中断 1：ECC计算完成时产生中断	R/W
b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	ECCECEN	可纠正ECC错误中断使能	0：1bitECC模式下发生可纠正ECC错误时不产生中断 1：1bitECC模式下发生可纠正ECC错误时产生中断	R/W
b0	ECCEUEN	不可纠正ECC错误中断使能	0：1bitECC模式下发生不可纠正ECC错误时不产生中断 1：1bitECC模式下发生不可纠正ECC错误时产生中断	R/W

40.5.3.10 中断状态寄存器（NFC_ISTR）

复位值: 0x00000000h

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
RBST[7:0]								-	ECCEST	-	ECCCST	-	-	ECCECST	ECCEUST

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b8	RBST	就绪中断状态	当访问设备就绪(RB信号置位)时置位 注：写入0时该位复位，写入1时无任何效果	R/W
b7	Reserved	-	读出时为“0”,写入时写“0”	R/W
b6	ECCEST	ECC错误状态	4bit ECC错误发生时置位 注：写入0时该位复位，写入1时无任何效果	R/W
b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4	ECCCST	ECC计算完成状态	ECC计算完成时置位 注：写入0时该位复位，写入1时无任何效果	R/W
b3~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	ECCECST	可纠正ECC错误状态	1bitECC模式下发生可纠正ECC错误时置位 注：写入0时该位复位，写入1时无任何效果	R/W
b0	ECCEUST	不可纠正ECC错误状态	1bitECC模式下发生不可纠正ECC错误时置位 注：写入0时该位复位，写入1时无任何效果	R/W

40.5.3.11 中断结果寄存器（NFC_IRSR）

复位值: 0x00000000h

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
RBRS[7:0]								-	ECCERS	-	ECCCRS	-	-	ECCECRS	ECCEURS

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b8	RBRS	就绪中断	该位为NFC_ISTR.RBST和NFC_IENR.RBEN的与逻辑结果	R
b7	Reserved	-	读出时为“0” ,写入时写“0”	R/W
b6	ECCERS	ECC错误中断	该位为NFC_ISTR.ECCEST和NFC_IENR.ECCEEN的与逻辑结果	R
b5	Reserved	-	读出时为“0” ,写入时写“0”	R/W
b4	ECCCRS	ECC计算完成中断	该位为NFC_ISTR.ECCCST和NFC_IENR.ECCCEN的与逻辑结果	R
b3~b2	Reserved	-	读出时为“0” ,写入时写“0”	R/W
b1	ECCECRS	可纠正ECC错误中断	该位为NFC_ISTR.ECCECST和NFC_IENR.ECCECEN的与逻辑结果	R
b0	ECCEURS	不可纠正ECC错误中断	该位为NFC_ISTR.ECCEUST和NFC_IENR.ECCEUEN的与逻辑结果	R

40.5.3.12 时序配置寄存器（NFC_TMCR0）

复位值: 0x03030202h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TH[7:0]								TRP[7:0]							

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

位	标记	位名	功能	读写
b31~b24	TH	-	设置CLE/ALE/CE在WE/RE变化后的保持时间	R/W
b23~b16	TRP	-	设置RE的有效(低电平)脉冲宽度	R/W
b15~b8	TWP	-	设置WE的有效(低电平)脉冲宽度	R/W
b7~b0	TS	-	设置CLE/ALE/CE在WE/RE变化前的建立时间	R/W

40.5.3.13 时序配置寄存器（NFC_TMCR1）

复位值: 0x28080303h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TWB[7:0]								TRR[7:0]							

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

位	标记	位名	功能	读写
b31~b24	TWB	-	设置WE上升沿到RB下降沿的时间	R/W
b23~b16	TRR	-	设置RB上升沿到RE下降沿的时间	R/W
b15~b8	TRH	-	设置WE的无效(高电平)脉冲宽度	R/W
b7~b0	TWH	-	设置WE的无效(高电平)脉冲宽度	R/W

时序配置寄存器 0 和时序配置寄存器 1 设置的参数如下图所示：

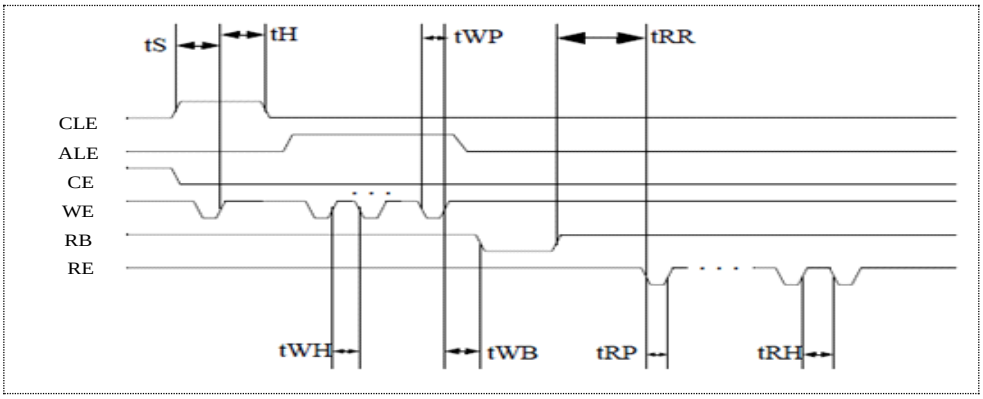


图 40-31 NFC 时序控制图

40.5.3.14 时序配置寄存器（NFC_TMCR2）

复位值: 0x03050D03h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TADL[7:0]								TRTW[7:0]							

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

位	标记	位名	功能	读写
b31~b24	TADL	-	设置ALE到读写数据的时间	R/W
b23~b16	TRTW	-	设置RE上升沿到WE下降沿的时间	R/W
b15~b8	TWTR	-	设置WE上升沿到RE下降沿的时间	R/W
b7~b0	TCCS	-	设置Change read column和Change write column命令后的延迟时间	R/W

40.5.3.15 ECC 校验寄存器（NFC_ECCR）

1) 在 1bitECC 模式下，ECCR0~15（Add Offset：0x8060~0x809C）作为 ECC 结果寄存器：

复位值: 0x00000000h

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
-	-	ME	SE	ERRLOC[11:0]											

位	标记	位名	功能	读写
b31~b14	Reserved	-	读出时为“0” ,写入时写“0”	R/W
b13	ME	多位错误	0：未检测到多位ECC错误 1：检测到多位ECC错误	R
b12	SE	一位错误	0：未检测到一位ECC错误 1：检测到一位ECC错误	R
b11~b0	ERRLOC	错误位置	表示错误的位置，只有当SE置位时有意义 2:0位表示错误byte中的bit位置： 000代表bit0错误，111代表bit7错误 11:3位表示错误byte在512byte中的位置： 000h表示第1个byte，1FFh代表第512个byte	R

2) 在 4bitECC 模式下，ECCR7~71（Add Offset：0x807C~0x817C）作为 ECC 状态寄存器和伴随式寄存器，如下表所示：

寄存器名	符号	偏移量	位宽	复位值
ECC状态寄存器	ECC_STAT	0x807C	32	0x00000000h
伴随式寄存器0	ECC_SYND0	0x8080~0x808C	32	0x00000000h
伴随式寄存器1	ECC_SYND1	0x8090~0x809C	32	0x00000000h
伴随式寄存器2	ECC_SYND2	0x80A0~0x80AC	32	0x00000000h
伴随式寄存器3	ECC_SYND3	0x80B0~0x80BC	32	0x00000000h
伴随式寄存器4	ECC_SYND4	0x80C0~0x80CC	32	0x00000000h
伴随式寄存器5	ECC_SYND5	0x80D0~0x80DC	32	0x00000000h
伴随式寄存器6	ECC_SYND6	0x80E0~0x80EC	32	0x00000000h
伴随式寄存器7	ECC_SYND7	0x80F0~0x80FC	32	0x00000000h

伴随式寄存器8	ECC_SYND8	0x8100~0x810C	32	0x00000000h
伴随式寄存器9	ECC_SYND9	0x8110~0x811C	32	0x00000000h
伴随式寄存器10	ECC_SYND10	0x8120~0x812C	32	0x00000000h
伴随式寄存器11	ECC_SYND11	0x8130~0x813C	32	0x00000000h
伴随式寄存器12	ECC_SYND12	0x8140~0x814C	32	0x00000000h
伴随式寄存器13	ECC_SYND13	0x8150~0x815C	32	0x00000000h
伴随式寄存器14	ECC_SYND14	0x8160~0x816C	32	0x00000000h
伴随式寄存器15	ECC_SYND15	0x8170~0x817C	32	0x00000000h

表 40-19 4bitECC 模式时的 ECCR 寄存器列表

ECC 状态寄存器（ECC_STAT）

复位值: 0x00000000h

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
ECCSEC[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
b15~0	ECCSEC	错误位置	0：对应section未出现ECC错误 1：对应section出现ECC错误	R

ECC 伴随式寄存器（ECC_SYND）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved			S2[12:0]												

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved			S1[12:0]												

位	标记	位名	功能	读写
b31~b29	Reserved	-	读出时为“0” ,写入时写“0”	R/W
b28~16	S2	S2	α2伴随式系数	R

b15~b13	Reserved	-	读出时为“0”，写入时写“0”	R/W
b12~0	S1	S1	α 伴随式系数	R

ECC 伴随式寄存器（ECC_SYND）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved			S4[12:0]												

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved			S3[12:0]												

位	标记	位名	功能	读写
b31~b29	Reserved	-	读出时为“0”，写入时写“0”	R/W
b28~16	S4	S4	α 4伴随式系数	R
b15~b13	Reserved	-	读出时为“0”，写入时写“0”	R/W
b12~0	S3	S3	α 3伴随式系数	R

ECC 伴随式寄存器（ECC_SYND）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved			S6[12:0]												

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved			S5[12:0]												

位	标记	位名	功能	读写
b31~b29	Reserved	-	读出时为“0”，写入时写“0”	R/W
b28~16	S6	S6	α 6伴随式系数	R
b15~b13	Reserved	-	读出时为“0”，写入时写“0”	R/W
b12~0	S5	S5	α 5伴随式系数	R

ECC 伴随式寄存器（ECC_SYND）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved			S8[12:0]												

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved			S7[12:0]												

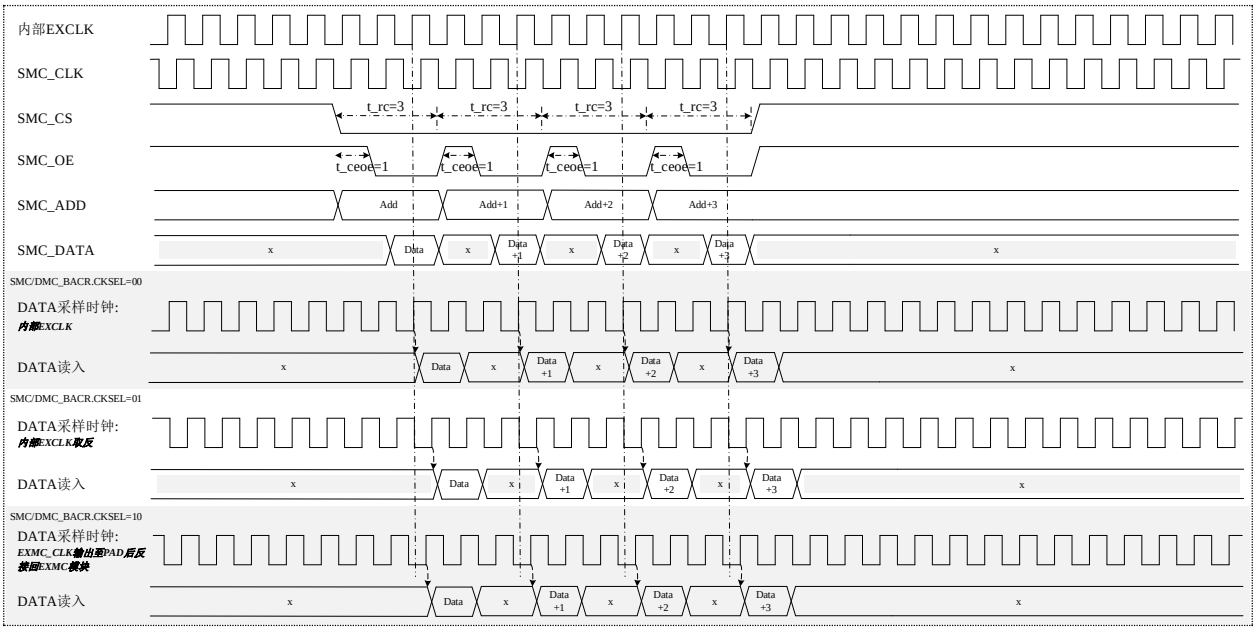
位	标记	位名	功能	读写
b31~b29	Reserved	-	读出时为“0” ,写入时写“0”	R/W
b28~16	S8	S8	α8伴随式系数	R
b15~b13	Reserved	-	读出时为“0” ,写入时写“0”	R/W
b12~0	S7	S7	α7伴随式系数	R

40.6 使用注意事项

1) SMC 的相关寄存器只能在 Ready 状态下设定；DMC 的相关寄存器只能在 Config 或 Low Power 状态下设定。

2) 在使用 SMC、DMC 模块对外部 Memory 进行读访问时，Memory Device 在接收到控制器发出的读命令后将数据传送给控制器，控制器对传送回的数据进行采样的时钟可以通过 SMC_BACR.CKSEL 或 DMC_BACR.CKSEL 进行选择，具体设定请参考寄存器章节。选择不同的采样时钟，可以改善通信路径的时间裕量。

下图是同步模式下的 SMC 访问例。



41 数字视频接口（DVP）

41.1 简介

数字摄像头接口（DVP）是一个同步并行接口，可采集从外部 CMOS 摄像头模块传入的 8 位、10 位、12 位或 14 位高速数据流。支持软件同步和硬件同步。支持对数据流的采集频率控制和窗口裁剪控制。支持单色或原始拜尔格式/YCbCr4:2:2/RGB565 逐行视频和压缩数据(JPEG)等不同格式的数据流采集。

41.2 系统框图

DVP 基本的功能及特性如下表所示。

基本功能	• 8位、10位、12位、14位并行接口
	• 单帧模式和连续模式
	• 软件、硬件行同步和帧同步
	• 帧采集频率控制
	• 窗口裁剪
	• FIFO控制
数据格式	• 单色或原始拜尔格式
	• YCbCr4:2:2逐行视频
	• RGB565逐行视频
	• JPEG压缩数据
中断事件	• 帧开始中断和事件
	• 帧结束中断和事件
	• 行开始中断和事件
	• 行结束中断和事件
	• 软件同步错误中断和事件
	• FIFO溢出错误中断和事件

表 41-1 DVP 基本功能及特性

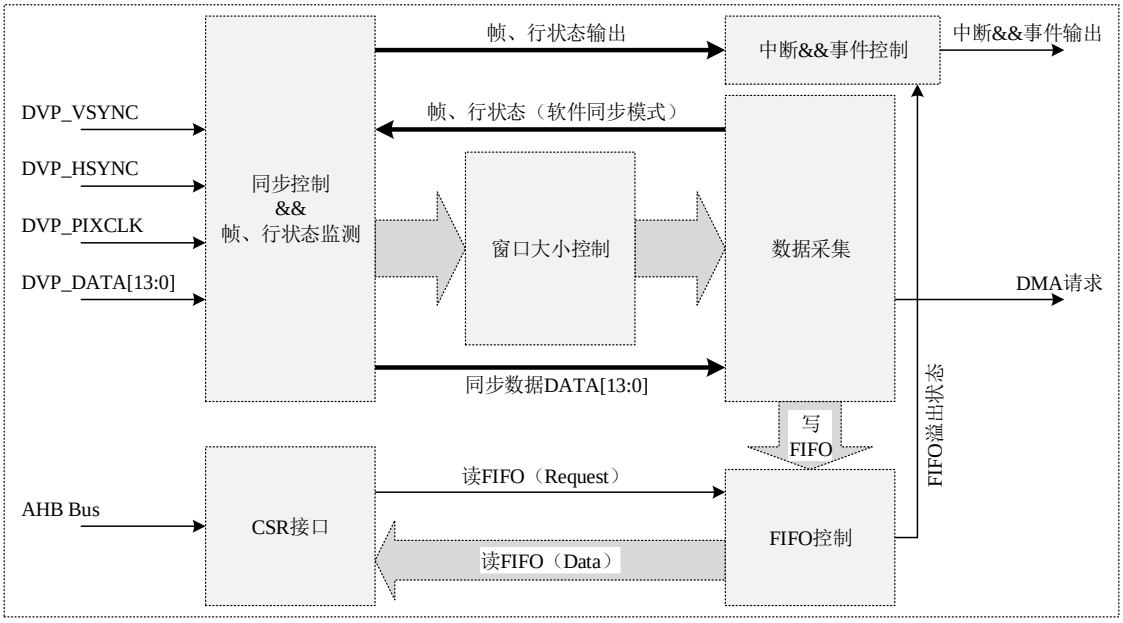


图 41-1 DVP 基本框图

下表所示为 DVP 的输入端口列表。

端口名	方向	功能
DVP_VSYNC	in	帧同步输入端口
DVP_HSYNC	in	行同步输入端口
DVP_PIXCLK	in	时钟输入端口
DVP_DATA[13:0]	in	数据输入端口

表 41-2 DVP 端口列表

41.3 功能说明

41.3.1 视频数据格式

41.3.1.1 单色格式

- 每个像素 8 位。

  下表显示了数据的存储方式。

字节地址	31:24	23:16	15:8	7:0
0x00	n+3	n+2	n+1	n
0x04	n+7	n+6	n+5	n+4

表 41-3 单色格式视频数据存储方式

41.3.1.2 YCbCr 格式

- 一个缓冲区交替存储 Y、Cb 和 Cr：CbYCrYCbYCr。。。
- DVP 模块支持 YCbCr4:2:2 格式数据传送，像素分量包括 Y（亮度）、Cb 和 Cr（蓝色色度和红色色度）。每个分量都采用 8 位进行编码，亮度和色度（交替）存储在一起。

  下表显示了数据的存储方式。

字节地址	31:24	23:16	15:8	7:0
0x00	Y n+1	Cr n	Y n	Cb n
0x04	Y n+3	Cr n+2	Y n+2	Cb n+2

表 41-4 YCbCr 格式视频数据存储方式

41.3.1.3 RGB565 格式

- 一个缓冲区交替存储 RGB 信号，BGRBGRBGR。。。
- DVP 模块支持 16BPP（每个像素 16 位）：RGB565（每 32 位表示 2 个像素），不支持 24BPP（托盘化格式）和灰度格式。

  下表显示了数据的存储方式。

字节地址	31:27	26:21	20:16	15:11	10:5	4:0
0x00	R n+1	G n+1	B n+1	R n	G n	B n
0x04	R n+3	G n+3	B n+3	R n+3	G n+3	B n+3

表 41-5 RGB565 格式视频数据存储方式

41.3.1.4 JPEG 格式

- JPEG 图像不按行和帧存储，DVP_VSYNC 信号用于启动采集过程，DVP_HSYNC 用作数据使能信号。
- DVP_HSYNC 有效区间包含的字节数可能不是 4 的倍数。
- DVP 模块支持 JPEG 格式数据传输（DVP_CTR.JPEGEN=1）。

41.3.2 并口存储格式

DVP 接口由 11、13、15、17 个输入信号组成。仅支持从模式。根据 DVP_CTR 寄存器中 BITSEL[1:0]位的设置，可以采集 8 位、10 位、12 位或 14 位数据。

摄像头模块输出的数据 DVP_DATA 与像素时钟 DVP_PIXCLK 保持同步，并根据像素时钟的极性在像素时钟上升沿/下降沿发生变化；DVP_VSYNC 信号指示帧的开始/结束；DVP_HSYNC 信号指示行的开始/结束。

DVP_VSYNC 或 DVP_HSYNC 有效时，表示同步区间，数据传输无效；DVP_VSYNC 和 DVP_HSYNC 无效时，表示数据区间，数据传输有效。如下图 41-2 所示信号波形图，DVP_DATA 在 DVP_PIXCLK 的上升沿发生变化（DVP 模块通过设定，在 DVP_PIXCLK 的下降沿采集数据 DVP_DATA），DVP_VSYNC 或 DVP_HSYNC 在高电平时为同步区间，数据传输无效。

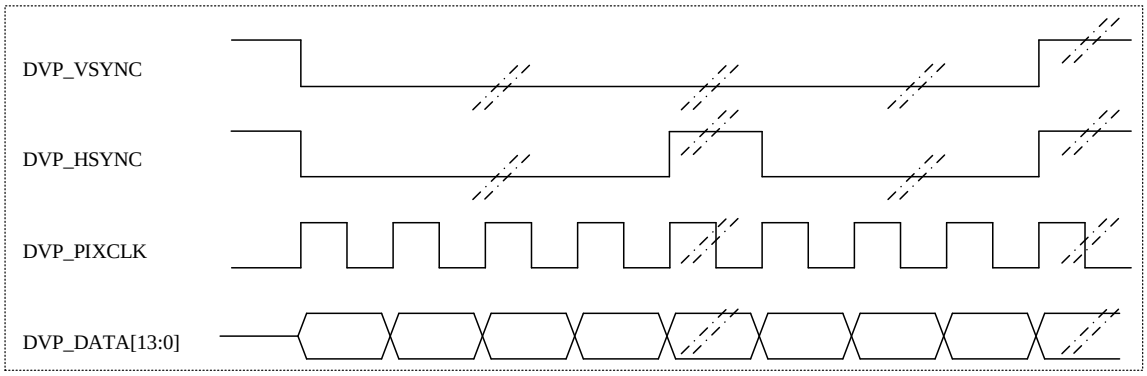


图 41-2 DVP 信号波形示意图

41.3.2.1 8Bit 数据

当 DVP_CTR.BITSEL[1:0]=00 时，DVP 接口将采集其输入 DVP_DATA[7:0]的 8 位数据，并将其存储为 8 位数据，忽略 DVP_DATA[13:8]的输入。在此情况下，每四个像素时钟周期会生成一个 32 位数据字。下表列举了采集到的数据字节在两个 32 位字中的位置排布。

字节地址	31:24	23:16	15:8	7:0
0x00	Dn+3[7:0]	Dn+2[7:0]	Dn+1[7:0]	Dn[7:0]
0x04	Dn+7[7:0]	Dn+6[7:0]	Dn+5[7:0]	Dn+4[7:0]

表 41-6 8bit 数据 DVP 存储方式

41.3.2.2 10Bit 数据

当 DVP_CTR.BITSEL[1:0]=01 时，DVP 接口将采集其输入 DVP_DATA[9:0]的 10 位数据，并将其存储到 16 位字的 10 个最低有效位，忽略 DVP_DATA[13:10]的输入。此时 DVP_DTR 寄存器中的其余最高有效位（Bit15:10）将清零。在此情况下，每两个像素时钟周期会生成一个 32 位数据字，如下表所示。

字节地址	31:26	25:16	15:10	9:0
0x00	0	Dn+1[9:0]	0	Dn[9:0]
0x04	0	Dn+3[9:0]	0	Dn+2[9:0]

表 41-7 10bit 数据 DVP 存储方式

41.3.2.3 12Bit 数据

当 DVP_CTR.BITSEL[1:0]=10 时，DVP 接口将采集其输入 DVP_DATA[11:0]的 12 位数据，并将其存储到 16 位字的 12 个最低有效位，忽略 DVP_DATA[13:12]的输入。此时 DVP_DTR 寄存器中的其余最高有效位（Bit15:12）将清零。在此情况下，每两个像素时钟周期会生成一个 32 位数据字，如下表所示。

字节地址	31:28	27:16	15:12	11:0
0x00	0	Dn+1[11:0]	0	Dn[11:0]
0x04	0	Dn+3[11:0]	0	Dn+2[11:0]

表 41-8 12bit 数据 DVP 存储方式

41.3.2.4 14Bit 数据

当 DVP_CTR.BITSEL[1:0]=11 时，DVP 接口将采集其输入 DVP_DATA[13:0]的 14 位数据，并将其存储到 16 位字的 14 个最低有效位。此时 DVP_DTR 寄存器中的其余最高有效位（Bit15:14）将清零。在此情况下，每两个像素时钟周期会生成一个 32 位数据字，如下表所示。

字节地址	31:30	29:16	15:14	13:0
0x00	0	Dn+1[13:0]	0	Dn[13:0]
0x04	0	Dn+3[13:0]	0	Dn+2[13:0]

表 41-9 14bit 数据 DVP 存储方式

41.3.3 模式选择

41.3.3.1 单帧模式

单帧模式下（DVP_CTR.CAPMD=1）只采集单个帧。在 DVP_CTR.CAPEN 位置 1 后，DVP 接口将等待系统检测到帧开始，然后对数据进行采集。收到完整的一帧数据后，将自动清零 DVP_CTR 中的 CAPEN 位，摄像头接口不再接收数据。如果使能帧结束中断，将生成对应的中断或事件(DVP_FRAMEND)。如下图所示。

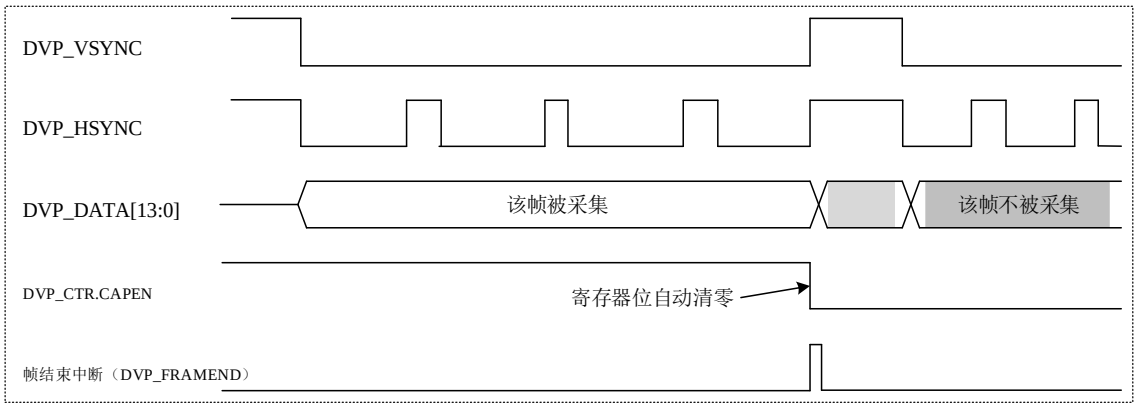


图 41-3 单帧模式数据采集动作

41.3.3.2 连续模式

连续模式下（DVP_CTR.CAPMD=0）会连续采集图像帧。在 DVP_CTR.CAPEN 位置 1 后，DVP 接口将等待系统检测到帧开始，然后对数据进行采集。该过程一直持续到 DVP_CTR 中的 CAPEN 位清零。CPU 对 DVP_CTR.CAPEN 位发出软件清零动作后，数据采集过程持续到当前帧结束，然后 DVP_CTR.CAPEN 位才被清零。如下图所示。

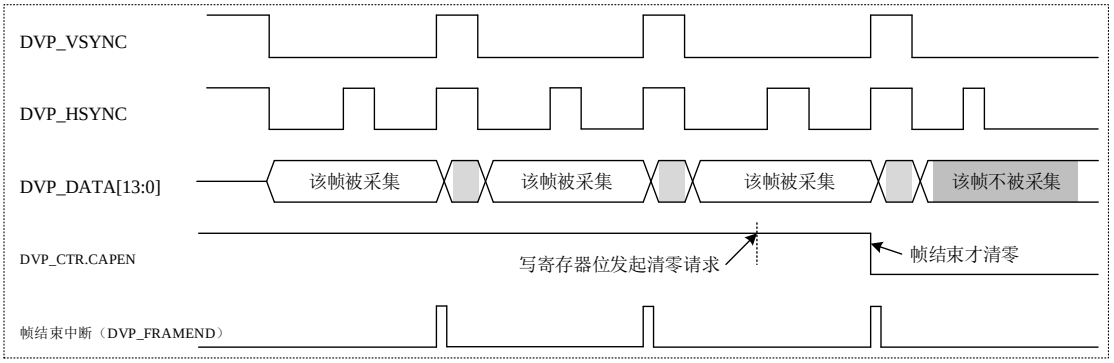


图 41-4 连续模式数据采集动作

在连续模式下，可以通过设定 DVP_CTR.CAPFRC[1:0]位来控制图像帧采集频率，以实现降低带宽的目的。例如当 DVP_CTR.CAPFRC[1:0]=01 时，摄像头接口对输入数据隔

帧进行采样，即每隔一帧采集一次图像数据。如下图所示。

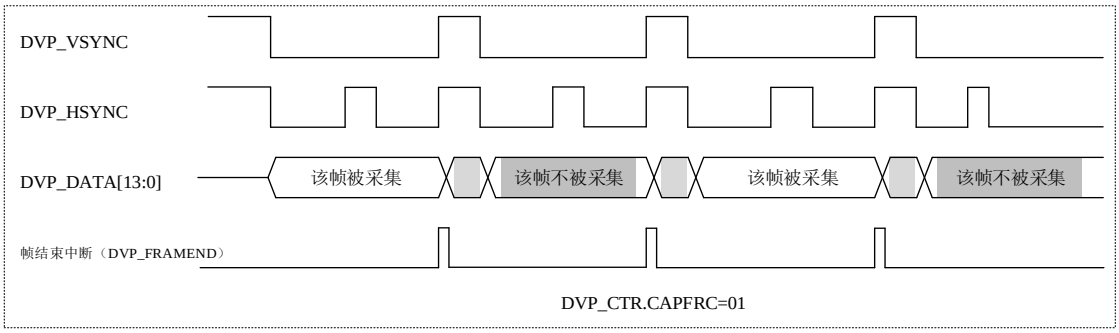


图 41-5　帧采集频率控制

41.3.4 同步控制

41.3.4.1 软件同步

软件同步模式下，将使用数据流中嵌入的 32 位同步码来同步数据流。这些同步码使用数据流中不再使用的 0x00/0xFF。同步码共有 4 种类型，均采用 0xFF0000XY 的格式，DVP 接口通过采集到的一个 32 位字数据判断帧的同步状态。

四个软件同步码分别表示以下事件：

- 帧开始 FS（DVP_SSYNDR.FSDAT[7:0]）
- 行开始 LS（DVP_SSYNDR.LSDAT[7:0]）
- 行结束 LE（DVP_SSYNDR.LEDAT[7:0]）
- 帧结束 FE（DVP_SSYNDR.FEDAT[7:0]）

同步码格式 0xFF0000XY 中的 XY 由上述 4 组寄存器位决定。例如，当采集到的数据中 XY 和 DVP_SSYNDR.FSDAT[7:0]相同时，代表一帧数据传送开始。

4 个同步码还具有位屏蔽功能，可以仅使用同步码中未被屏蔽的位进行比较，各同步码的屏蔽位由 DVP_SSYNMR 中的对应位设定。屏蔽功能有效时，可以选择一个位用于同步码的比较，来检测帧/行起始和结束。这意味着可以多个同步码表示帧/行的起始和结束，只要未被屏蔽的位相同即可。例如当 DVP_SSYNDR.FSDAT[7:0]设定为 0xA5 时、DVP_SSYNMR.FSMSK[7:0]设定为 0xF0 时，只需要比较数据码的高 4 位来检测是否是 FS 信号。

根据软件同步机理，DVP 只支持 8 位数据格式接口（DVP_CTR.BITSEL[1:0]=00）和全帧采集（DVP_CTR.CAPFRC[1:0]=00），其它数据宽度格式或隔帧采集时不适用，会产生无法预知的结果。JPEG 格式数据传输也不适用（DVP_CTR.JPEGEN=0）。

41.3.4.2 硬件同步

硬件同步模式下将使用两个端口同步信号(DVP_VSYNC/DVP_HSYNC)。摄像头接口模块根据 DVP_VSYNC/DVP_HSYNC 的电平变化判断帧/行起始和帧/行结束。

在硬件同步模式时启用数据流采集功能（DVP_CTR.CAPEN=1），在 DVP_VSYNC 端口上采集到无效边沿后，数据传输便可以开始执行，由 DMA 将连续帧传输到多个连续的缓冲区或一个具有循环特性的缓冲区。每一帧结束时都将激活帧传送结束中断（当

帧传送结束中断 DVP_FRAMEND 使能时）。

41.3.5 窗口裁剪

摄像头接口可以使用裁剪功能从收到的图像中选择一个矩形窗口。窗口的起始坐标和尺寸大小（像素时钟数表示水平尺寸、行数表示垂直尺寸）由窗口裁剪偏移寄存器（DVP_CPSFTR）和窗口裁剪尺寸寄存器（DVP_CPSZER）指定。如图 41-6 所示。

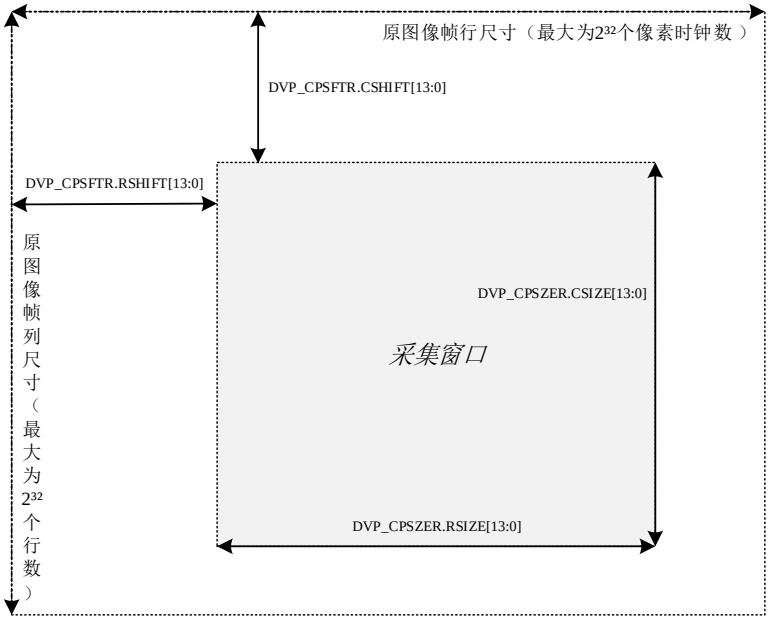


图 41-6 DVP 窗口设定图

裁剪窗口的起点坐标指定为某一行号（从 0 开始）和某个像素时钟数（从 0 开始），窗口大小则指定为行数和像素时钟数。如果在窗口裁剪尺寸寄存器（DVP_CPSZER）中指定的行数完成之前，DVP_VSYNC 信号已有效，那么图像采集将停止，并且在中断使能时生成帧传送结束中断（DVP_FRAMEND）。如图 41-7，是窗口裁剪数据采集的示意图。

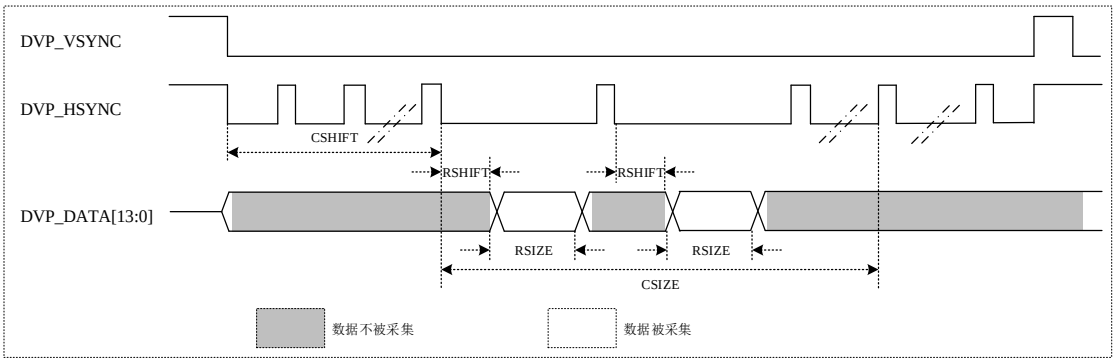


图 41-7 窗口裁剪数据采集图

JPEG 格式数据传输（DVP_CTR.JPEGEN=1）时，窗口裁剪功能不适用，需将窗口裁剪功能无效。

41.3.6 FIFO 控制

为了对 AHB 上的数据传输速度加以管理，在 DVP 模块中搭载了 8 个字深度的 FIFO 并配有一个 FIFO 控制器。每次从摄像头接口 FIFO 读取数据时读指针递增、每次向摄像头接口 FIFO 写入数据时写指针递增。如果数据写入速度超过了 AHB 接口能够承受的速率，FIFO 会产生溢出，数据会被覆盖。此时若使能 FIFO 溢出错误中断，将产生相应的中断。

如果发生软件同步错误事件，FIFO 将复位，DVP 接口将等待新的数据帧开始。

41.3.7 DMA 控制

当 DVP_CTR.CAPEN=1 时，DMA 接口就被激活。摄像头接口每次在数据寄存器（DVP_DTR）中收到一个完整的 32 位数据块时，就可以触发一个 DMA 读请求。DMA 通过 AHB 接口对 DMA 数据传输寄存器（DVP_DMR）访问时，即从 FIFO 中读出对应的缓存数据。该 DMA 读请求是事件列表的第 90 号事件（参见 INTC 模块的中断事件请求信号章节），DMA 的相关设定参考 DMA 章节。

在 JPEG 数据传输时，行中包含的字节数可能不是 4 的倍数，DVP 接口在检测到行结束并且尚未凑成 32 位字时，将使用“0”进行填充，并触发一个 DMA 读请求。

41.4 中断及事件说明

DVP 接口含有 3 类中断和事件输出，分别是帧传送状态中断和事件、软件同步错误中断和事件、FIFO 溢出错误中断和事件。

41.4.1 帧传送状态中断及事件

在图像帧正常传送期间的各个状态时，分别可产生相应的中断。如下表所示。

中断/事件名	中断/事件符号	中断使能位
帧传送开始	DVP_FRAMSTA	IER.FSIEN
行传送开始	DVP_LINESTA	IER.LSIEN
行传送结束	DVP_LINEEND	IER.LEIEN
帧传送结束	DVP_FRAMEND	IER.FEIEN

表 41-10 帧传送中断

41.4.2 软件同步错误中断及事件

在软件同步模式时，若接收到的同步码出现表 41-11 中所示的错误序列时，对应的状态位（DVP_STR.SQUERF）会被置 1，此时若设定相应的中断使能（DVP_IER.SQUERIEN）有效，则产生软件同步错误（DVP_SQUERR）中断。

FS	LS	LE	FE
0	0	0	1
0	0	1	x
0	1	x	x
1	0	1	x
1	1	0	1

表 41-11 同步码错误序列

上表中“1”表示接收到该同步序列，“0”表示未接收到该同步序列，“x”表示该同步序列是否被接收到不影响序列错误判断。例如，当 FS：LS：LE：FE=1101 时，表明在接收到帧开始、行开始序列之后，又接收到了帧结束序列，行结束序列未接收到，此时产生软件同步序列错误事件。

产生同步码错误时，内部的帧采集控制逻辑会复位，不再接收当前帧的数据，直至DVP_STR.SQUERF 被清零之后，开始接收并监测下一帧的帧开始序列。

41.4.3 FIFO 溢出错误中断及事件

在 FIFO 发生溢出时，对应的状态位（DVP_STR.FIFOERF）会被置 1，此时若设定相应的中断使能（DVP_IER.FIFOERIEN）有效，则产生 FIFO 溢出错误（DVP_FIFOERR）中断。

41.5 寄存器说明

表 41-12 所示，为 DVP 模块的寄存器列表。

BASE ADDR: 0x40055800H

寄存器名	符号	偏移量	位宽	复位值
控制寄存器	DVP_CTR	0x0000h	32	0x00000000h
数据寄存器	DVP_DTR	0x0004h	32	0x00000000h
状态寄存器	DVP_STR	0x0008h	32	0x00000000h
中断寄存器	DVP_IER	0x000Ch	32	0x00000000h
DMA数据传输寄存器	DVP_DMR	0x0010h	32	0x00000000h
软件同步数据寄存器	DVP_SSYNDR	0x0020h	32	0x00000000h
软件同步屏蔽寄存器	DVP_SSYNMR	0x0024h	32	0xFFFFFFFFh
窗口裁剪偏移寄存器	DVP_CPSFTR	0x0028h	32	0x00000000h
窗口裁剪尺寸寄存器	DVP_CPSZER	0x002Ch	32	0x00000000h

表 41-12 DVP 寄存器列表

41.5.1 控制寄存器（DVP_CTR）

复位值: 0x0000000h

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
-	DVP EN	-	-	BITSEL[1:0]		CAPFRC[1:0]		VSYNC SEL	HSYNC SEL	PIXCK SEL	SW SYNC	JPEG EN	CROP EN	CAP MD	CAP EN

位	标记	位名	功能	读写
b31~b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14	DVPEN	DVP模块使能	0：DVP模块无效 1：DVP模块使能	R/W
b13~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11~b10	BITSEL[1:0]	数据宽度选择	00：8位数据宽度 01：10位数据宽度 10：12位数据宽度 11：14位数据宽度	R/W
b9~b8	CAPFRC[1:0]	帧采集频率选择	00：全帧采集（采集所有帧） 01：每隔1帧采集一次（带宽降低50%） 10：每隔3帧采集一次（带宽降低75%） 11：全帧采集（采集所有帧）	R/W
b7	VSYNCSEL	帧同步电平选择	0：DVP_VSYNC为低电平时，同步功能有效 1：DVP_VSYNC为高电平时，同步功能有效	R/W
b6	HSYNCSEL	行同步电平选择	0：DVP_HSYNC为低电平时，同步功能有效 1：DVP_HSYNC为高电平时，同步功能有效	R/W
b5	PIXCKSEL	像素时钟选择	0：DVP_PIXCLK下降沿时，采集数据 1：DVP_PIXCLK上升沿时，采集数据	R/W
b4	SWSYNC	软件同步使能	0：软件同步无效 1：软件同步有效	R/W
b3	JPEGEN	JPEG传送使能	0：非JPEG格式数据传送 1：JPEG格式数据传送	R/W
b2	CROPEN	窗口裁剪使能	0：窗口裁剪功能无效 1：窗口裁剪功能有效	R/W
b1	CAPMD	采集模式选择	0：连续模式 1：单帧模式	R/W
b0	CAPEN	采集功能使能	0：禁止采集 1：使能采集 注：单帧模式下一帧数据结束后该位自动清零；连续模式下，该位写零后，到当前帧结束后，清零才生效。	R/W

41.5.2 数据寄存器（DVP_DTR）

复位值: 0x00000000h

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]	采集数据	当前采集到的一组数据	R

41.5.3 状态寄存器（DVP_STR）

复位值: 0x00000000h

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										FIFO ERF	SQU ERF	FEF	LEF	LSF	FSF

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	FIFOERF	FIFO溢出错误标志	0：未发生FIFO溢出错误 1：发生FIFO溢出错误	R/W
b4	SQUERF	软件同步错误标志	0：未发生软件同步错误 1：发生软件同步错误	R/W
b3	FEF	帧传送结束标志	0：未监测到帧传送结束 1：检测到帧传送结束	R/W
b2	LEF	行传送结束标志	0：未监测到行传送结束 1：检测到行传送结束	R/W
b1	LSF	行传送开始标志	0：未监测到行传送开始 1：检测到行传送开始	R/W
b0	FSF	帧传送开始标志	0：未监测到帧传送开始 1：检测到帧传送开始	R/W

41.5.4 中断寄存器（DVP_IER）

复位值: 0x00000000h

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										FIFO ERIEN	SQU ERIEN	FE IEN	LE IEN	LS IEN	FS IEN

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	FIFOERIEN	FIFO溢出错误中断使能	0：FIFO溢出错误中断无效 1：FIFO溢出错误中断使能	R/W
b4	SQUERIEN	软件同步错误中断使能	0：软件同步错误中断无效 1：软件同步错误中断使能	R/W
b3	FEIEN	帧传送结束中断使能	0：帧传送结束中断无效 1：帧传送结束中断使能	R/W
b2	LEIEN	行传送结束中断使能	0：行传送结束中断无效 1：行传送结束中断使能	R/W
b1	LSIEN	行传送开始中断使能	0：行传送开始中断无效 1：行传送开始中断使能	R/W
b0	FSIEN	帧传送开始中断使能	0：帧传送开始中断无效 1：帧传送开始中断使能	R/W

41.5.5 DMA 数据传输寄存器（DVP_DMR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DMR[31:16]															

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

位	标记	位名	功能	读写
b31~b0	DMR[31:0]	DMA传输值	从该寄存器读取当前读FIFO中的数据	R

41.5.6 软件同步数据寄存器（DVP_SSYNDR）

复位值: 0x00000000h

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

位	标记	位名	功能	读写
b31~b24	FEDAT[7:0]	帧结束同步码	软件同步模式时的帧结束同步码	R/W
b23~b16	LEDAT[7:0]	行结束同步码	软件同步模式时的行结束同步码	R/W
b15~b8	LSDAT[7:0]	行开始同步码	软件同步模式时的行开始同步码	R/W
b7~b0	FSDAT[7:0]	帧开始同步码	软件同步模式时的帧开始同步码	R/W

41.5.7 软件同步屏蔽寄存器（DVP_SSYNMR）

复位值: 0xFFFFFFFFh

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

位	标记	位名	功能	读写
b31~b24	FEMSK[7:0]	帧结束屏蔽码	软件同步模式时的帧结束屏蔽码	R/W
b23~b16	LEMSK[7:0]	行结束屏蔽码	软件同步模式时的行结束屏蔽码	R/W
b15~b8	LSMSK[7:0]	行开始屏蔽码	软件同步模式时的行开始屏蔽码	R/W
b7~b0	FSMSK[7:0]	帧开始屏蔽码	软件同步模式时的帧开始屏蔽码	R/W

41.5.8 窗口裁剪偏移寄存器（DVP_CPSFTR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	CSHIFT[13:0]													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	RSHIFT[13:0]													

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~b16	CSHIFT[13:0]	列偏移数	表示窗口开始的行数	R/W
b15~b14	Reserved	-	读出时为“0”,写入时写“0”	R/W
b13~b0	RSHIFT[13:0]	行偏移数	表示窗口开始的列数（时钟像素数） 注：在软件同步模式时，行偏移数以4的倍数进行偏移，b1~b0可读写，但功能设定无效。	R/W

41.5.9 窗口裁剪尺寸寄存器（DVP_CPSZER）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	CSIZE[13:0]													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	RSIZE[13:0]													

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~b16	CSIZE[13:0]	列尺寸数	表示窗口每一列上的尺寸大小，即窗口的行数	R/W
b15~b14	Reserved	-	读出时为“0”,写入时写“0”	R/W
b13~b0	RSIZE[13:0]	行尺寸数	表示窗口每一行上的尺寸大小（时钟像素数），即窗口的列数 注：窗口大小列数需设定为4的倍数，b1~b0设定无效，固定为0。	R/W

42 加密协处理模块（CPM）

42.1 简介

加密协处理模块（CPM）包括 AES 加解密算法处理器，HASH 安全散列算法，TRNG 真随机数发生器三个子模块。

AES 加解密算法处理器遵循美国国家标准技术研究所（NIST）在 2000 年 10 月 2 日正式宣布的新的数据加密标准，分组长度固定为 128 位，而密钥长度支持 128/192/256 位。

HASH 安全散列算法是 SHA-2 版本的 SHA-256（Secure Hash Algorithm），符合美国国家标准和技术局发布的国家标准“FIPS PUB 180-3”，可以对长度不超过 2^64 位的消息产生 256 位的消息摘要输出。支持 HMAC（密钥散列消息认证码）应用，使用 SHA-256 散列函数对消息进行认证。

TRNG 真随机数发生器是以连续模拟噪声为基础的随机数发生器，提供 64bit 随机数。

42.2 加解密算法处理器（AES）

42.2.1 算法简介

AES（The Advanced Encryption Standard）是美国国家标准技术研究所（NIST）在 2000 年 10 月 2 日正式宣布的新的数据加密标准。

AES 的分组长度固定为 128 位，而密钥长度支持 128、192 和 256 位。对于加密来说，其输入是一个明文分组和一个密钥，输出是一个密文分组；对解密而言，输入是一个密文分组和一个密钥，而输出是一个明文分组。此过程如图 42-1 所示：

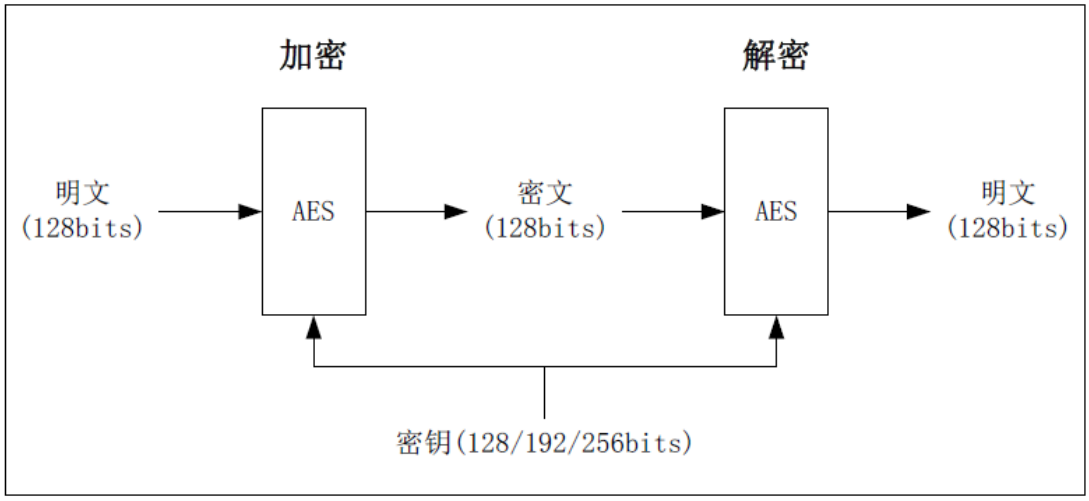


图 42-1 AES 加解密流程示意图

AES 算法处理的基本单位是字节，128 位信息被分成 16 个字节，按顺序复制到一个 4×4 的矩阵中，称为状态（state），AES 的所有变换都是基于状态矩阵的变换，该矩阵上保存着计算的中间结果。

AES 是一个密钥迭代分组密码，包含了轮变换对状态的重复作用。AES 的轮变换由四个操作组成：SubBytes、ShiftRows、MixColumns、AddRoundKey。其中，SubBytes 包括求每个字节在 GF(2^8)中的模逆元和一个仿射变换；ShiftRows 是一个字节换位，它将状态中的行按照不同的偏移量进行循环移位；MixColumns 对状态各列进行线性变换；AddRoundKey，状态中的各字节与轮密钥进行逐位异或操作。AES 的加密流程如图 42-2 所示：

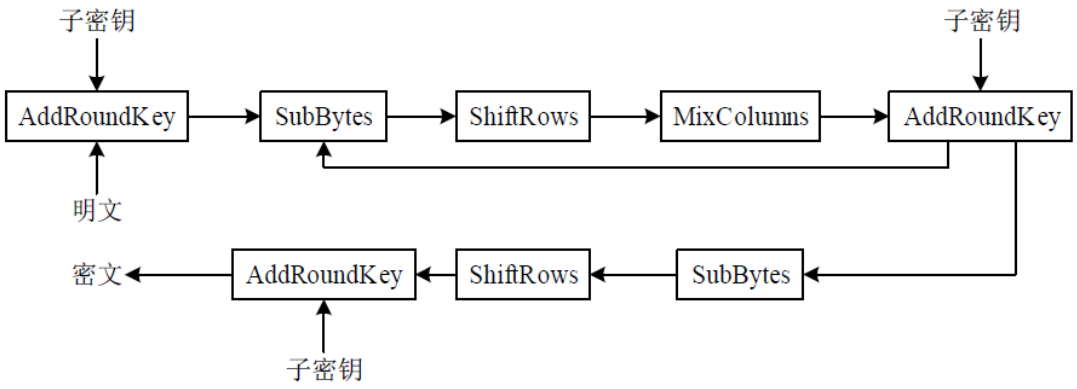


图 42-2 AES 加密流程图

图中所用子密钥需要由初始密钥扩展而来，且密钥的扩展过程和加密过程是同步进行的。

由于明文固定为 128 位，加密过程运行的轮数就取决于密钥的长度。比如，密钥为 128 位时，运行轮数为 10 轮；密钥为 192 位时，运行轮数为 12 轮；密钥为 256 位时，运行轮数为 14 轮。除了最后一轮缺少 MixColumns 变换，其余各轮均进行完整的轮变换操作。

解密流程与加密流程有所区别，首先必须完成所有密钥的扩展，解密过程从扩展的最后一轮子密钥往回使用；然后是轮变换的四个操作变成了相应的逆运算：InvSubBytes、InvShiftRows、InvMixColumns、AddRoundKey。InvSubBytes 中的模逆运算仍然保持，但仿射变换改为逆变换；InvShiftRows 和 InvMixColumns 变成相应的逆变换；AddRoundKey 保持不变。

直接解密流程的轮变换对四个操作的调用顺序为：InvShiftRows、InvSubBytes、AddRoundKey、InvMixColumns，与加密流程的调用顺序不一致，但使用的密钥与加密流程一致；等价解密流程的轮变换对四个操作的调用顺序为：InvSubBytes 、InvShiftRows、InvMixColumns、AddRoundKey，与加密流程的调用顺序完全一致，只是每一轮的子密钥需要进行 InvMixColumns 运算。

关于详细的算法表述，可以参见标准《FIPS PUB 197》

42.2.2 AES 模块功能描述

- 执行 AES 算法标准的加密流程和解密流程，其执行结果完全符合《FIPS PUB 197》对算法原理的描述；
- 支持 128、192 和 256 位密钥。

42.2.3 加密操作流程

AES 的加密操作流程例如下：

1) 将待加密的 128 位数据写入数据寄存器（AES_DR）中。
2) 将加密密钥写入密钥寄存器（AES_KR）中。
3) 根据密钥长度设置 AES_CR.KEYSIZE
4) 将 AES_CR.MODE 设置为 0，启动加密模式。
5) 向控制寄存器中的 AES_CR.START 写入 1，启动模块进行运算。

   3）、4）和 5）可同时进行。
6) 等待 AES_CR.START 的值恢复为 0，模块运算结束。
7) 读取数据寄存器（AES_DR），获得 128 位密文。

42.2.4 解密操作流程

AES 的解密操作流程例如下：

1) 将待解密的 128 位数据写入数据寄存器（AES_DR）中。
2) 将解密密钥写入密钥寄存器（AES_KR）中。
3) 根据密钥长度设置 AES_CR.KEYSIZE
4) 将 AES_CR.MODE 设置为 1，启动解密模式。
5) 向控制寄存器中的 AES_CR.START 写入 1，启动模块进行运算。

   3）、4）和 5）可同时进行。
6) 等待 AES_CR.START 的值恢复为 0，模块运算结束。
7) 读取数据寄存器（AES_DR），获得 128 位明文。

42.2.5 数据示例

128 位明文：

0xFFEEDDCCBBAA99887766554433221100

128 位密钥：

0x0F0E0D0C0B0A09080706050403020100

128 位密文：

0x5AC5B47080B7CDD830047B6AD8E0C469

加密前

寄存器	值（密钥）	寄存器	值（明文）
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC

加密后

寄存器	值（密钥）	寄存器	值（密文）
Key0	0x03020100	Data0	0xD8E0C469
Key1	0x07060504	Data1	0x30047B6A
Key2	0x0B0A0908	Data2	0x80B7CDD8
Key3	0x0F0E0D0C	Data3	0x5AC5B470

表 42-1 128 位操作寄存器示例

128 位明文：

0xFFEEDDCCBBAA99887766554433221100

192 位密钥：

0x17161514131211100F0E0D0C0B0A09080706050403020100

128 位密文：

0x5AC5B47080B7CDD830047B6AD8E0C469

加密前

寄存器	值（密钥）	寄存器	值（明文）
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC
Key4	0x13121110		
Key5	0x17161514		

加密后

寄存器	值（密钥）	寄存器	值（密文）
Key0	0x03020100	Data0	0xA47CA9DD
Key1	0x07060504	Data1	0xE0DF4C86
Key2	0x0B0A0908	Data2	0xA070AF6E
Key3	0x0F0E0D0C	Data3	0x91710DEC
Key4	0x13121110		
Key5	0x17161514		

表 42-2 192 位操作寄存器示例

128 位明文：

0xFFEEDDCCBBAA99887766554433221100

256 位密钥：

0x1F1E1D1C1B1A19181716151413121110F0E0D0C0B0A09080706050403020100

128 位密文：

0x5AC5B47080B7CDD830047B6AD8E0C469

加密前

寄存器	值（密钥）	寄存器	值（明文）
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC
Key4	0x13121110		
Key5	0x17161514		
Key6	0x1B1A1918		
Key7	0x1F1E1D1C		

加密后

寄存器	值（密钥）	寄存器	值（密文）
Key0	0x03020100	Data0	0xCAB7A28E
Key1	0x07060504	Data1	0xBF456751
Key2	0x0B0A0908	Data2	0x9049FCEA
Key3	0x0F0E0D0C	Data3	0x8960494B
Key4	0x13121110		
Key5	0x17161514		
Key6	0x1B1A1918		
Key7	0x1F1E1D1C		

表 42-3　256 位操作寄存器示例

42.2.6 运行时间说明

AES 模块从启动一次运算（AES_CR.START 写入 1）到该次运算结束（AES_CR.START 恢复到 0）所需时间如下表所示：

	128位密钥	192位密钥	256位密钥
加密	220 cycles	260 cycles	300 cycles
解密	290 cycles	332 cycles	398 cycles

表 42-4 AES 加解密运行时间

42.2.7 操作注意事项

1) 在 AES 加解密过程中，数据寄存器会改变，如果下次运算操作的数据就是本次运算的结果，就无需重新写入数据。
2) 支持 128、192 和 256 位密钥，128 位密钥写入偏移地址 0x20~0x2C，192 位密钥写入偏移地址 0x20~0x34，256 位密钥写入偏移地址 0x20~0x3C。
3) 判断模块运算结束的方法：不断读取 AES_CR.START，如果其值变为 0，则表示运算结束。

42.2.8 寄存器说明

寄存器基地址: 0x40008000

寄存器名	符号	偏移量	位宽	复位值
AES控制寄存器	AES_CR	0x0000	32	0x00000000h
AES数据寄存器0~3	AES_DR0~3	0x0010~0x001C	32	0x00000000h
AES密钥寄存器0~7	AES_KR0~7	0x0020~0x003C	32	0x00000000h

表 42-5 寄存器列表

42.2.8.1 AES 控制寄存器（AES_CR）

复位值：0x00000000h

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											KEYSIZE		Reserved	MODE	START

位	标记	位名	功能	读写
b31~b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4~b3	KEYSIZE	密钥长度选择	2’b00：密钥长度为128位 2’b01：密钥长度为192位 2’b10：密钥长度为256位 2’b11：密钥长度为128位	R/W
b2	Reserved	-	读出时为“0”,写入时写“0”	-
b1	MODE	加密解密模式选择	0：加密运算 1：解密运算	R/W
b0	START	启动	0：本模块运算结束或未被启动 1：启动本模块进行运算	R/W

说明：

1. AES_CR.START 位的操作方法是：软件对本位写入 1 后，本模块将启动运行，本次运行结束后本模块硬件会自动将本位清 0，软件查询到本位为 0 即表示本次运行完成。

2. 对本寄存器的写入操作只能在本模块不处于运算状态时（即 AES_CR.START＝0 时）才能进行，否则硬件将自动忽略写操作。读操作则不受此限制。

42.2.8.2 AES 数据寄存器（AES_DRx）（x=0~3）

复位值：0x00000000h

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	加密数据	存放AES算法的128比特明文/密文	R/W

说明：

1. 数据寄存器由 4 个 32 位的寄存器组成 128 位数据，用于在模块运算前存放需要被加密的明文或者需要被解密的密文，并且运算完成后存放加密后的密文或者解密后的明文。

加密运算		解密运算	
运算前	运算后	运算前	运算后
128位明文	128位密文	128位密文	128位明文

四个 32 位寄存器连接在一起组成一个 128 位的数据，读写操作时需要分别对四个寄存器进行操作。数据寄存器对应的操作顺序如下：

数据举例：0xFFEEDDCCBBAA99887766554433221100

偏移地址	寄存器名称	填入数据
0x10	AES_DR0	0x33221100
0x14	AES_DR1	0x77665544
0x18	AES_DR2	0xBBAA9988
0x1C	AES_DR3	0xFFEEDDCC

2. 对于本寄存器的写入只能在本模块没有处于运算状态时（即 AES_CR.START＝0 时）才能进行，否则硬件将自动忽略对本寄存器的写操作。

3. 对于本寄存器的读取只能在本模块没有处于运算状态时（即 AES_CR.START＝0 时）才能进行，否则对本寄存器的读取将得到全 0。

42.2.8.3 AES 密钥寄存器（AES_KRx）（x=0~7）

复位值：0x00000000h

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

位	标记	位名	功能	读写
b31~b0	Key	密钥	存放AES算法的128/192/256位密钥	R/W

说明：

1. 密钥寄存器由 8 个 32 位的寄存器组成，存放输入的初始密钥。写操作时需要分别对 8 个 32 位的寄存器进行操作。对应的操作顺序如下：

   128 位密钥举例：0x0F0E0D0C_0B0A0908_07060504_03020100

偏移地址	寄存器名称	填入数据
0x20	AES_KR0	0x03020100
0x24	AES_KR1	0x07060504
0x28	AES_KR2	0x0B0A0908
0x2C	AES_KR3	0x0F0E0D0C

   192 位密钥举例：0x17161514_13121110_0F0E0D0C_0B0A0908_07060504_03020100

偏移地址	寄存器名称	填入数据
0x20	AES_KR0	0x03020100
0x24	AES_KR1	0x07060504
0x28	AES_KR2	0x0B0A0908
0x2C	AES_KR3	0x0F0E0D0C
0x30	AES_KR4	0x13121110
0x34	AES_KR5	0x17161514

256 位密钥举例：

0x1F1E1D1C_1B1A1918_17161514_13121110_0F0E0D0C_0B0A0908_07060504_03020100

偏移地址	寄存器名称	填入数据
0x20	AES_KR0	0x03020100
0x24	AES_KR1	0x07060504
0x28	AES_KR2	0x0B0A0908
0x2C	AES_KR3	0x0F0E0D0C
0x30	AES_KR4	0x13121110
0x34	AES_KR5	0x17161514
0x38	AES_KR6	0x1B1A1918
0x3C	AES_KR7	0x1F1E1D1C

2. 对于本寄存器的写入只能在本模块没有处于运算状态时（即 AES_CR.START = 0 时）才能进行，否则硬件将自动忽略对本寄存器的写操作。
3. 对于本寄存器的读取只能在本模块没有处于运算状态时（即 AES_CR.START = 0 时）才能进行，否则对本寄存器的读取将得到全 0。

42.3 安全散列算法（HASH）

42.3.1 算法简介

安全散列算法的步骤如下：

首先填充消息使其长度恰好为一个比 512 的倍数仅小 64 位的数。填充方法是附一个 1 在消息后面，后接所要求的多个 0，然后在其后附上 64 位的消息长度(填充前)，使消息长度恰好是 512 位的整数倍。

其次将 A、B、C、D、E、F、G、H 8 个 32 位变量，用十六进制初始化。然后开始算法的主循环，一次处理 512 位消息，循环次数是消息中 512 位分组的数目。

主循环一共进行 64 次操作，此操作称为压缩函数。每次操作都包含移位、循环移位、逻辑运算、模 232 加等，运算的过程见下图 42-3。最后的输出由 A、B、C、D、E、F、G、H 级联而成。其中 Wt 为由 512 位消息得到的第 t 步所用的临时值，Kt 为第 t 步所用的常数值，t（0≤t≤63）是 64 步循环中的一步。

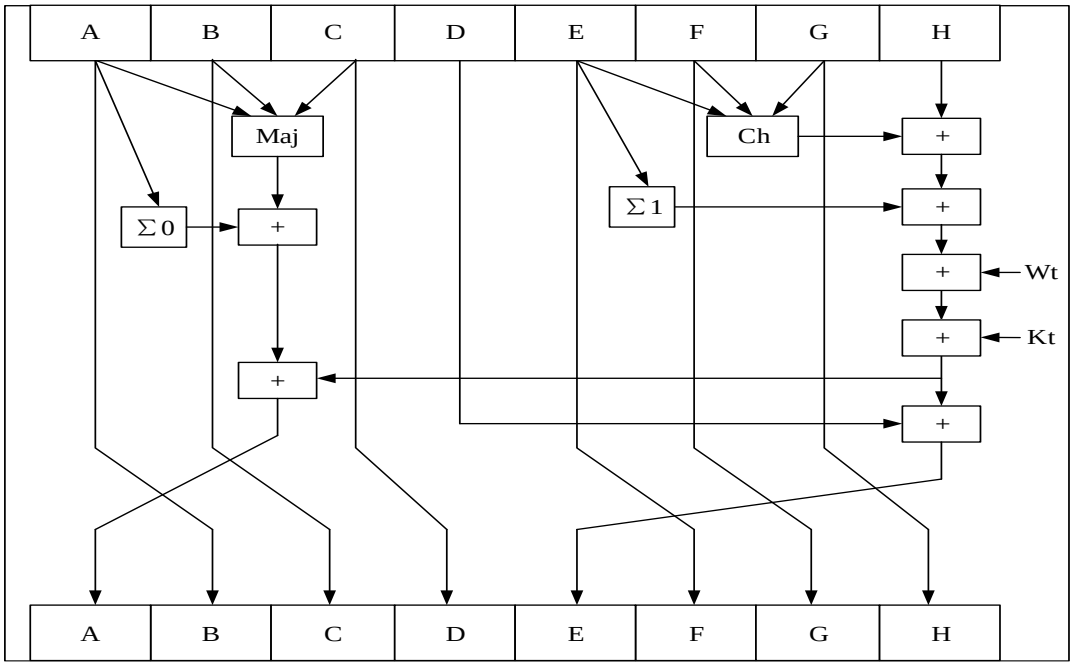


图 42-3 HASH 算法流程图

HASH 模块集成了硬件 HMAC 功能；也可以通过软件方式根据 HMAC 算法直接调用 HASH 运算实现 HMAC 功能。

42.3.2 操作流程

HSAH 模块的操作流程如下：

软件操作流程：

1) 软件将原始数据按照算法规则进行填充，并且将填充后的消息按 512 位进行分组。
2) 将数据写入数据寄存器 HASH_DR 中。
3) 如果本次运算为消息分组的第一组数据，设置 HASH_CR.FST_GRP=1。
4) 设置 HASH_CR.START=1，启动本模块进行运算。

注：以上的 3）和 4）可以同时进行

5) 判断本模块本次运算是否完成，通过以下方法：

不断读取 HASH_CR.START 直到读到该位为 0 即表示运算完成；或者不断读取 HASH_CR.BUSY 直到读到该位为 0 即表示运算完成；又或者设置 HASH 中断，通过中断方式判断运算结束。

6) 如果本次运算不是消息分组的最后一组数据，返回到 2）。
7) 如果本次运算为消息分组的最后一组数据，读取摘要寄存器 HASH_HR 获得本次运算的结果。如需再次进行运算则返回到步骤 1）。

使用 DMA 操作流程：

1) 将需要处理的消息按照 512 比特进行分组，最后一组消息需要按照规则预先填充为 512 比特的块。
2) 设置内部触发事件寄存器的 HASH 运算启动方式为 DMA 传送完成和数据块传送完成；设置内部触发事件寄存器的 DMA 传送开始方式为 HASH 运算完成（或者其他内部触发事件）；
3) 设置 HASH_CR.MODE=2’b00，HASH 运算方式；设置 HASH_CR.FST_GRP=1，并设置 HASH_CR.START=1，启动 HASH 运算。如果选择其他内部触发事件启动 DMA，则需要等待触发事件发生后再开始 HASH 运算。
4) 等待 HASH_CR.CYC_END=1，HASH 运算完成，并清除该位。
5) 读取摘要寄存器 HASH_HR 的结果。

注意，当只有一组运算数据时，请同时设置 HASH_CR.FST_GRP=1 和 HASH_CR.KMSG_END=1.

42.3.3 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≥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 mod 512（后 64bit 留做原始消息长度的表示）；如果 l%512≥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) Example1：

      原始消息为字符串“abcde”，用二进制位串表示其 ASCII 码为："01100001 01100010 01100011 01100100 01100101"，添加长度的步骤如下：

      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) Example2：

原始消息为字符串“abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopq”。每个字符都可通过其 ASCII 码转换为 8bit，所以消息的长度为 l = 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

42.3.4 HMAC 运算

HMAC（Keyed-Hashing for Message Authentication）以不可逆方式将正在处理的消息与用户所选的密钥进行绑定，从而用于消息验证。基本而言，HMAC 是通过加入共享密钥的方式由两个嵌套的底层 HASH 运算组成：

HMAC（message）= Hash[((key XOR opad) | Hash((key XOR ipad) | message)]

其中：

ipad: 64 字节(SHA-256) 0x36。

opad:64 字节(SHA-256) 0x5C。

| :表示连接运算符。

42.3.4.1 HMAC 运算过程

1) 设置 HASH_CR.MODE=2’b01，选择 HMAC 工作模式；
2) 如果使用的密钥长度超过 64 字节，则需要设置 HASH_CR.LKEY=1（在此情况下，HMAC 将使用规范规定的密钥替代给定密钥）；
3) 将密钥写入 HASH_DR 寄存器，直到所有的密钥分组运算完成；
4) 将消息写入 HASH_DR 寄存器，直到所有的消息分组运算完成；
5) 读取 HASH_HR 寄存器的最终散列运算结果。

HMAC 运算过程中的数据加载，可以选择软件操作流程和 DMA 操作流程的方式，具体举例如下：

42.3.4.2 HMAC 软件操作流程

1) 设置模式 HASH_CR.MODE=2’b01，HMAC 工作模式；设置 HASH_CR.LKEY=1,选择长密钥方式；
2) HASH_DR 中写入密钥分组数据；
3) 如果为第一组数据，则设置 HASH_CR.FST_GRP=1。设置 HASH_CR.START=1 启动密钥运算；
4) 等待 HASH_CR.BUSY=0 运算完成；
5) 如果不是最后一组数据，则重复 2)~4)；
6) 如果是最后一组数据，则设置 HASH_CR.KMSG_END=1，并设置

HASH_CR.START=1 启动最后一组密钥运算。等待 HASH_CR.CYC_END=1 后，密钥运算完成，并将该位清除。

7) HASH_DR 中写入消息分组数据；
8) 如果为第一组消息数据，则设置 HASH_CR.FST_GRP=1。设置 HASH_CR.START=1 启动消息运算；
9) 等待 HASH_CR.BUSY=0 运算完成；
10) 如果不是最后一组消息数据，则重复操作 7)~9)；
11) 如果是最后一组消息，则设置 HASH_CR.KMSG_END=1，并设置 HASH_CR.START=1 启动最后一组消息运算。等待 HASH_CR.CYC_END=1 后，所有 HMAC 运算完成，并将该位清除；
12) 读取 HASH_HR 中的最终消息认证结果。

42.3.4.3 HMAC 使用 DMA 操作流程

1) 设置 DMA 为数据块传送方式，数据块的大小为 16 个字，不管是密钥还是消息都需要按照 16 个字的大小分组，未满 16 个字时，请按照 HASH 消息填充规则进行填充，配置 DMA 数据块源地址为密钥地址；
2) 设置内部触发事件寄存器的 HASH 运算启动方式为 DMA 传送完成和数据块传送完成；设置内部触发事件寄存器的 DMA 传送开始方式为 HASH 运算完成（或者其他内部触发事件）；
3) 设置 HASH_CR.MODE=2’b01，HMAC 运算模式；根据密钥长度配置 HASH_CR.LKEY 位；设置 HASH_CR.FST_GRP=1，HASH_CR.START=1 启动密钥运算；如果选择其他内部触发事件，则需要等待触发事件后开始启动运算；
4) 等待 HASH_CR.CYC_END=1 密钥运算完成后，将该位清除；
5) 停止 DMA，将 DMA 数据块源地址指向消息地址；
6) 设置 HASH_CR.FST_GRP=1，HASH_CR.START=1 启动消息运算；或等待其他触发事件后启动消息运算。
7) 等待 HASH_CR.CYC_END=1 消息运算完成后，将该位清除；
8) 读取 HASH_HR 寄存器中最终的消息认证结果。

42.3.4.4 HMAC 运算举例

密钥和消息三组 HMAC 运算结果如下：

密钥	
字符	abcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcd
16 进制（第一组）	6162636461626364616263646162636461626364616263646162636461626364616263646162636461626364616263646162636461626364
字符	abcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcd
16 进制（第二组）	6162636461626364616263646162636461626364616263646162636461626364616263646162636461626364616263646162636461626364
字符	abcd
16 进制（第三组）	6162636480000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000420
消息	
字符	abcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcd
16 进制（第一组）	6162636461626364616263646162636461626364616263646162636461626364616263646162636461626364616263646162636461626364
字符	abcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcd
16 进制（第二组）	6162636461626364616263646162636461626364616263646162636461626364616263646162636461626364616263646162636461626364
字符	abcd
16 进制（第三组）	6162636480000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000420
消息认证	
16 进制	1a1957893345e9d85873dbc359495f33954053387b7f55f928148a468910796b

42.3.4.5 注意事项

1) 当填充后的密钥和消息只有一组数据，即 512 位时，启动运算前请同时设置 HASH_CR.FST_GRT=1 和 HASH_CR.KMSG_END=1；
2) 关于密钥填充，当密钥小于 64 字节时，未满 64 字节部分请填充 0x00；当密钥大于 64 字节时，未满 64 字节部分请按照 HASH 填充规则进行填充。

42.3.5 中断说明

HASH 中断标记 HASH_INT。不管是在 HASH 运算模式还是在 HMAC 运算模式下，在设置 HASH_CR.HEIE=1 中断许可条件下，每组数据运算完成后将产生一次中断请求。HASH_CR.HCIE=1 中断许可条件下，所有的密钥或消息运算完成后将产生一次中断请求。HEIE 和 HCIE 同时许可时，任何一个条件满足都将产生中断请求。另外，HEIE 使能后，在模块复位后或者运算完成后，直接设置 HASH_CR.FST_GRP 且 HASH_CR.START=1 时，也将产生一次中断用于数据加载。

42.3.6 硬件触发事件选择

通过配置内部硬件触发事件选择寄存器 HASH_ITRGSELA，HASH_ITRGSELB 在数据传送完成后启动 HASH 运算，通常与 DMA 配合使用。例如，当选择 DMA_2 传送数据和启动 HASH 运算时，首先，配置 DMA2_TRGSELx(x=0~7)选择 HASH_INT 启动 DMA 传送；其次，配置 HASH_ITRGSELA 选择 DMA_BTCx(x=0~7)，表示完成一个数据块(512bit)传输并启动 HASH 运算，配置 HASH_ITRGSELB 选择 DMA_TCx(x=0~7)，表示完成所有数据块传输。当启动 HASH 后，DMA 和 HASH 相互触发，直到所有数据块完成传送和运算。

42.3.7 寄存器说明

BASE ADDR: 0x40008400

寄存器名	符号	偏移量	位宽	复位值
HASH控制寄存器	HASH_CR	0x0000	32	0x00000000h
HASH摘要寄存器7	HASH_HR7	0x0010	32	0x00000000h
HASH摘要寄存器6	HASH_HR6	0x0014	32	0x00000000h
HASH摘要寄存器5	HASH_HR5	0x0018	32	0x00000000h
HASH摘要寄存器4	HASH_HR4	0x001C	32	0x00000000h
HASH摘要寄存器3	HASH_HR3	0x0020	32	0x00000000h
HASH摘要寄存器2	HASH_HR2	0x0024	32	0x00000000h
HASH摘要寄存器1	HASH_HR1	0x0028	32	0x00000000h
HASH摘要寄存器0	HASH_HR0	0x002C	32	0x00000000h
HASH数据寄存器15	HASH_DR15	0x0040	32	0x00000000h
HASH数据寄存器14	HASH_DR14	0x0044	32	0x00000000h
HASH数据寄存器13	HASH_DR13	0x0048	32	0x00000000h
HASH数据寄存器12	HASH_DR12	0x004C	32	0x00000000h
HASH数据寄存器11	HASH_DR11	0x0050	32	0x00000000h
HASH数据寄存器10	HASH_DR10	0x0054	32	0x00000000h
HASH数据寄存器9	HASH_DR9	0x0058	32	0x00000000h
HASH数据寄存器8	HASH_DR8	0x005C	32	0x00000000h
HASH数据寄存器7	HASH_DR7	0x0060	32	0x00000000h
HASH数据寄存器6	HASH_DR6	0x0064	32	0x00000000h
HASH数据寄存器5	HASH_DR5	0x0068	32	0x00000000h
HASH数据寄存器4	HASH_DR4	0x006C	32	0x00000000h
HASH数据寄存器3	HASH_DR3	0x0070	32	0x00000000h
HASH数据寄存器2	HASH_DR2	0x0074	32	0x00000000h
HASH数据寄存器1	HASH_DR1	0x0078	32	0x00000000h
HASH数据寄存器0	HASH_DR0	0x007c	32	0x00000000h

BASE ADDR: 0x40010800

寄存器名	符号	偏移量	位宽	复位值
硬件事件触发选择寄存器A	HASH_ITRGSELA	0x007C	32	0x000001FFh
硬件事件触发选择寄存器B	HASH_ITRGSELB	0x0078	32	0x000001FFh

42.3.7.1 HASH 控制寄存器（HASH_CR）

复位值：0x00000000h

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
HEIE	HCIE	-	-	-	HMAC_END	CYC_END	BUSY	-	LKEY	MODE[1:0]		-	KMSG_END	FST_GRP	START

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”，写入时请写“0”	R/W
b15	HEIE	每组数据运算完成中断许可	0：每组运算完成中断禁止 1：每组运算完成中断许可	R/W
b14	HCIE	全部密钥或消息运算完成中断许可	0：全部密钥或消息运算完成中断禁止 1：全部密钥或消息运算完成中断许可	R/W
b14~b11	Reserved	-	读出时为“0”，写入时请写“0”	R/W
b10	HMAC_END	HMAC运算完成标志	0：HMAC未运算完成 1：HMAC运算完成 当标志位为“1”后，写“0”清除。注意，软件在确认标志位后请及时清“0”，否则标志位将保持为“1”。	R/W
b9	CYC_END	密钥或消息运算完成标志	0：密钥或消息运算未完成 1：密钥或消息运算完成 当标志位为“1”后，写“0”清除。注意，软件在确认标志位后请及时清“0”，否则标志位将保持为“1”。	R/W
b8	BUSY	转换标志	0：HASH运算处于空闲状态 1：HASH运算处理中	R
b7	Reserved	-	读出时为“0”，写入时请写“0”	R/W
b6	LKEY	长密钥选择	0：短密钥（≤64字节） 1：长密钥（>64字节） 在 HMAC 模式中，该位在短密钥（≤64 字节）或长密钥（> 64 字节）之间进行选择。在MODE=2’b01，HMAC模式时有效。	R/W
b5~b4	MODE	工作模式	00：SHA-256工作模式 01：HMAC工作模式 1X：设定禁止	R/W
b3	Reserved	-	读出时为“0”，写入时请写“0”	R/W
b2	KMSG_END	密钥或消息分组的最后一组	0：密钥或消息非最后一组 1：密钥或消息最后一组 当最后一组密钥或消息转换完成后，该位自动清“0”	
b1	FST_GRP	密钥或消息分组的第一组	0：密钥或消息非第一组	R/W

			1：密钥或消息第一组 当第一组密钥或消息转换完成后，该位自动清“0”	
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。

42.3.7.2 HASH 摘要寄存器（HASH_HR）

位数 ： 256 位

偏移地址 ： 10’h010 – hash[255:224]

10’h014 – hash[223:192]

10’h018 – hash[191:160]

10’h01C – hash[159:128]

10’h020 – hash[127:96]

10’h024 – hash[95:64]

10’h028 – hash[63:32]

10’h02C – hash[31:0]

复位值 ： 0x00000000h（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。

42.3.7.3 HASH 数据寄存器（HASH_DR）

位数 ： 512 位

偏移地址 ： 10’h040 – data[511: 480]

10’h044 – data[479: 448]

10’h048 – data[447: 416]

10’h04C – data[415: 384]

……

10’h070 – data[127: 96]

10’h074 – data[95: 64]

10’h078 – data[63: 32]

10’h07C – data[31: 0]

复位值 ： 0x00000000h（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。

42.3.7.4 HASH 硬件事件触发选择寄存器 A（HASH_ITRGSELA）

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

偏移地址：0x7C

复位值: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
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~b9	Reserved	-	读时读出 0，写时请写 0	R
b8~b0	ITRGSEL[8:0]	触发源选择	写入要选择的外围电路事件的编号	R/W

42.3.7.5 HASH 硬件事件触发选择寄存器 B（HASH_ITRGSELB）

寄存器说明：当最后一组数据写入 HASH_DR 后，通过该寄存器选择硬件事件触发源，以通知 HASH 进行最后一次运算。注意，请勿选择 DMA_TCx(x=0~7)以外的其他触发信号。

偏移地址：0x78

复位值: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
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~b9	Reserved	-	读时读出 0，写时请写 0	R
b8~b0	ITRGSEL[8:0]	触发源选择	写入要选择的外围电路事件的编号	R/W

42.4 真随机数发生器（TRNG）

42.4.1 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 42-4 所示。其中随机数发生器是一个模拟随机数发振器电路，用于得到随机性的噪声；算法模块捕捉随机噪声并把结果保存至数据模块、并通过总线输出；控制模块控制 TRNG 的模式和启动。

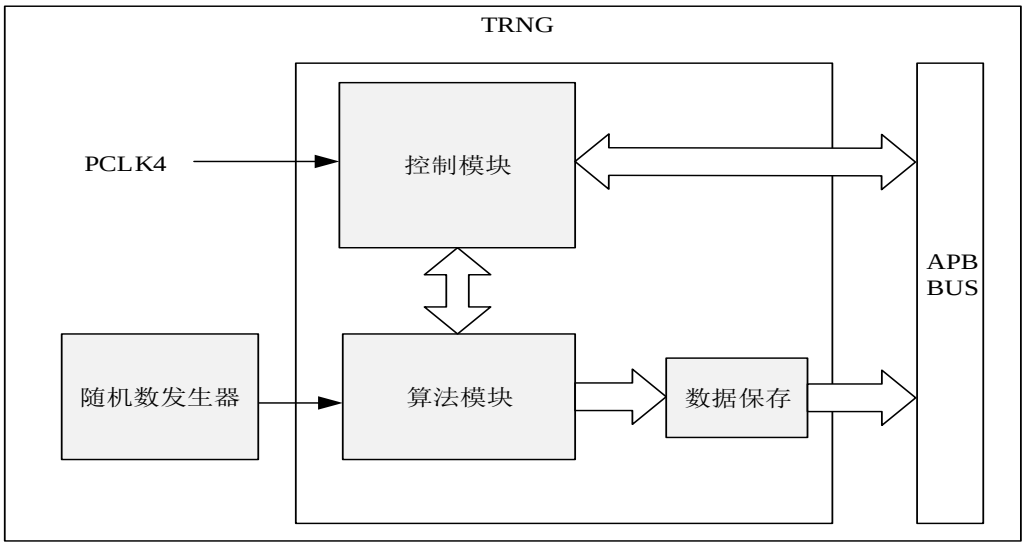


图 42-4 TRNG 系统框图

42.4.2 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）。

42.4.3 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。

42.4.4 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。

42.4.5 寄存器说明

BASE ADDR: 0x40042000

寄存器名	符号	偏移量	位宽	复位值
TRNG控制寄存器	TRNG_CR	0x0000h	32	0x00000000h
TRNG模式寄存器	TRNG_MR	0x0004h	32	0x00000012h
TRNG数据寄存器0	TRNG_DR0	0x000Ch	32	0x08000000h
TRNG数据寄存器1	TRNG_DR1	0x0010h	32	0x08000200h

42.4.5.1 TRNG 控制寄存器（TRNG_CR）

复位值：0x00000000h

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

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

位	标记	位名	功能	读写
b31~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	RUN	随机数运算开始	0：随机数运算停止 1：随机数运算开始 软件写”1”，产生新的64位随机数；运行完毕后，硬件清零。	R/W
b0	EN	模拟发振器使能	0：关闭模拟随机数发生器电路 1：打开模拟随机数发生器电路	R/W

42.4.5.2 TRNG 模式寄存器（TRNG_MR）

复位值：0x00000012h

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

位	标记	位名	功能	读写
b31~b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4~2	CNT[2:0]	移位次数控制位	捕捉随机噪声时，移位次数控制位 011：移位32次 100：移位64次 101：移位128次 110：移位256次 000~010，111：功能保留位	R/W
b1	Reserved	-	读出时为“1”,写入时写“1”	R/W
b0	LOAD	装载控制位	随机数生成前，数据寄存器是否从随机数发生器装载新的初始值 0：不装载新的初始值 1：装载新的初始值	R/W

42.4.5.3 TRNG 数据寄存器（TRNG_DR）

复位值：DR0：0x08000000h

DR1：0x08000200h

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

43 CRC 运算（CRC）

43.1 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。

43.2 功能框图

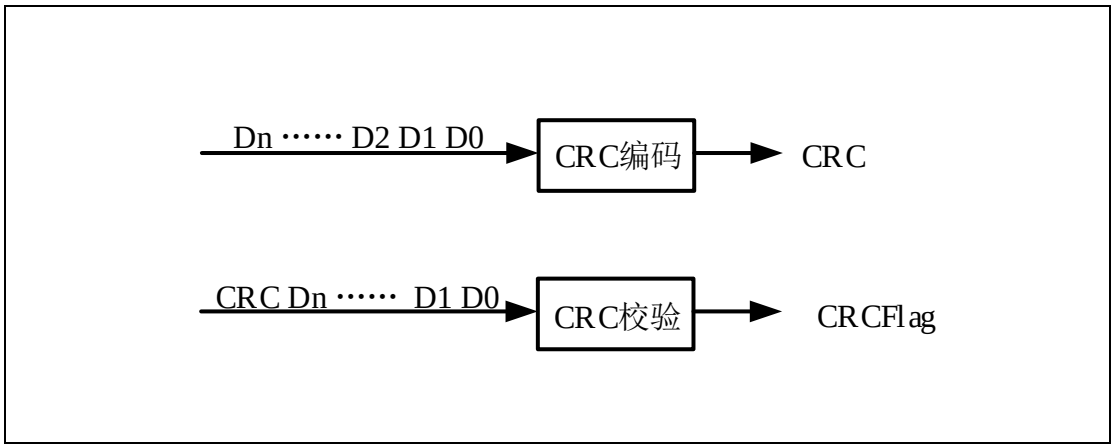


图 43-1 CRC 应用示意图

43.3 功能说明

本模块 CRC 算法遵从 ISO/IEC13239 的定义，分别采用 32 位和 16 位的 CRC。CRC32 的生成多项式为 $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X+1$，32 位初值为”0xFFFFFFFF”。CRC16 的生成多项式为 $X^{16}+X^{12}+X^{5}+1$,16 位初值为”0xFFFF”。

本模块功能包括：

- CRC 编码和 CRC 校验；
- 3 种位宽访问方式 8 位，16 位，32 位：

  8 位位宽下输入数据示例为 0x00,0x11,0x22,0x33,0x44,0x55,0x66,0x77;

  16 位位宽下输入数据示例为 0x1100,0x3322,0x5544,0x7766;

  32 位位宽下输入数据示例为 0x33221100,0x77665544;

43.3.1 CRC16 编码模式

CRC 编码就是对原始数据编码以计算其 CRC 值，操作流程如下：

1. 向 CRC_CR.CR 写入 1’b0,选择 CRC16。
2. 向 CRC_RESLT[15:0]写入 0xFFFF，初始化 CRC 计算。
3. 将待编码的原始数据按 8 位/16 位/32 位组织方式，依次写入 CRC_DAT 寄存器。

   注：按 8 位组织方式写 CRC 值，应先写低位，后写高位。
4. 读取 CRC_RESLT[15:0]，即可获得 16 位 CRC 编码值。

43.3.2 CRC16 校验模式

校验模式可以校验已编码的数据是否被篡改，操作流程如下所示：

1. 向 CRC_CR.CR 写入 1’b0,选择 CRC16。
2. 向 CRC_RESLT[15:0]写入 0xFFFF，初始化 CRC 计算。
3. 将已编码的数据按 8 位/16 位/32 位组织方式，依次写入 CRC_DAT 寄存器。

   注：按 8 位组织方式写 CRC 值，应先写低位，后写高位。
4. 校验码以 16 位位宽将数据写入 CRC_DAT 寄存器。
5. 读取 CRC_CR.FLG 寄存器，为 1 则表示校验成功，为 0 则表示校验失败。

43.3.3 CRC32 编码模式

CRC 编码就是对原始数据编码以计算其 CRC 值，操作流程如下：

1. 向 CRC_CR.CR 写入 1’b1,选择 CRC32。
2. 向 CRC_RESLT[31:0]写入 0xFFFF_FFFF，初始化 CRC 计算。
3. 将待编码的原始数据按 8 位/16 位/32 位组织方式，依次写入 CRC_DAT 寄存器。

   注：按 8 位组织方式写 CRC 值，应先写低位，后写高位。
4. 读取 CRC_RESLT[31:0]，即可获得 16 位 CRC 编码值。

43.3.4 CRC32 校验模式

校验模式可以校验已编码的数据是否被篡改，操作流程如下所示：

1. 向 CRC_CR.CR 写入 1’b1,选择 CRC32。
2. 向 CRC_RESLT[31:0]写入 0xFFFF_FFFF，初始化 CRC 计算。
3. 将已编码的数据按 8 位/16 位/32 位组织方式，依次写入 CRC_DAT 寄存器。

   注：按 8 位组织方式写 CRC 值，应先写低位，后写高位。
4. 校验码按 8 位/16 位/32 位组织方式，写入 CRC_DAT 寄存器。
5. 读取 CRC_CR.FLG 寄存器，为 1 则表示校验成功，为 0 则表示校验失败。

43.4 寄存器说明

表 43-1 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_0001
CRC结果寄存器	CRC_RESTLT	0x04	32	0x0000_0000
CRC数据寄存器	CRC_DAT	0x80~0xFF	32	0x0000_0000

表 43-1 CRC 寄存器列表

43.4.1 控制寄存器（CRC_CR）

复位值：0x0001

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	FLAG	CRC

位	标记	位名	功能	读写
b30~b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	FLAG	校验结果位	0：当前校验错误 1：当前校验正确	R
b0	CR	运算控制位	0：CRC16 1：CRC32	R/W

43.4.2 结果寄存器（CRC_RESLT）

复位值：0x0000

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

位	标记	位名	功能	读写
b31~b0	RESULT[31:0]	结果位	选择CRC16时，取RESULT[15:0]; 选择CRC32时，取RESULT[31:0];	R/W

43.4.3 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(0x80~0xFF)，对该范围内任何一个地址进行操作，都会认为对本寄存器进行操作。该寄存器读出为全0。	R/W

44 数据计算单元（DCU）

44.1 概要

数据计算单元(Data Computing Unit)是一个不借助于 CPU 的简单处理数据的模块。每个 DCU 单元具有 3 个数据寄存器，能够进行 2 个数据的加减和比较大小以及窗口比较功能。还可以通过定时器触发为数模转换模块(DAC)提供连续变化的数字量以产生三角波和锯齿波输出。本产品搭载 8 个 DCU 单元，每个单元均可独立完成自身功能。

功能概要：

- 可进行 4 种数据处理：2 个数据的加法，减法，比较以及 3 个数据窗口比较
- 加法和减法对 DATA0 和 DATA1 寄存器中的数据进行运算，结果存放在 DATA0 中
- 加法和减法可以选择写寄存器后计算或由其他外围电路事件触发进行运算
- 加法和减法运算可自动将结果进行一次减半操作，并将减半的结果和加减运算的结果分别放入 2 个数据寄存器内供其他模块使用
- 比较模式可以对 DATA0 和 DATA1 寄存器之间，以及 DATA0 和 DATA2 寄存器之间的 2 个数据进行比较，可以分别选择当满足大于，小于，等于时产生中断和标志位
- 比较模式可以用来进行窗口比较，即分别设置 DATA1 和 DATA2 为窗口的上下限，根据 DATA0 和 DATA1 及 DATA0 和 DATA2 的比较结果判断 DATA0 在窗口内或窗口外
- 能够由其他外围电路事件触发进行运算，并根据运算结果产生多种中断和事件信号。当其他具有硬件触发启动功能的外围电路选择 DCU 作为触发源时，由 DCU 产生事件信号来启动外围电路开始动作
- 在三角波输出模式下，通过定时器定时触发产生连续递增和递减的数字量给数模转换模块(DAC)，后者能够将这些数字量通过数模转换转换为三角波输出
- 在递增锯齿波输出模式下，通过定时器定时触发产生连续递增的数字量给数模转换模块(DAC)，后者能够将这些数字量通过数模转换转换为递增的锯齿波输出
- 在递减锯齿波输出模式下，通过定时器定时触发产生连续递减的数字量给数模转换模块(DAC)，后者能够将这些数字量通过数模转换转换为递减的锯齿波输出

44.2 功能描述

44.2.1 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(DATA0＋DATA1)/2 的运算，DATA0＋DATA1 的结果存入 DATA0，而(DATA0＋DATA1)/2 的结果存入 DATA2。当 DATA0＋DATA1 结果超过 0xFF(8bit 模式)或 0xFFFF(16bit 模式)或 0xFFFFFFFF(32bit 模式)时产生标志位并产生中断。

加法模式应用举例：

1. 控制寄存器 DCU_CTL 选择加法模式，数据宽度为 16bit
2. 中断条件选择寄存器 DCU_INTSEL 选择运算条件
3. 分别在 DATA0 和 DATA1 中写入 0xFF00 和 0x55，此时计算结果为 0xFF55，结果保存在 DATA0 中
4. DATA1 中继续写入 0xFF，此时计算结果溢出，产生结果标志。读取标志寄存器 DCU_FLAG 的得到结果标志
5. 写标志复位寄存器 DCU_FLAGCLR 清除标志位

44.2.2 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(DATA0－DATA1)/2 的运算，DATA0－DATA1 的结果存入 DATA0，而(DATA0-DATA1)/2 的结果存入 DATA2。当 DATA0－DATA1 结果小于 0x0(8bit，16bit，32bit 模式)时产生标志位并产生中断。

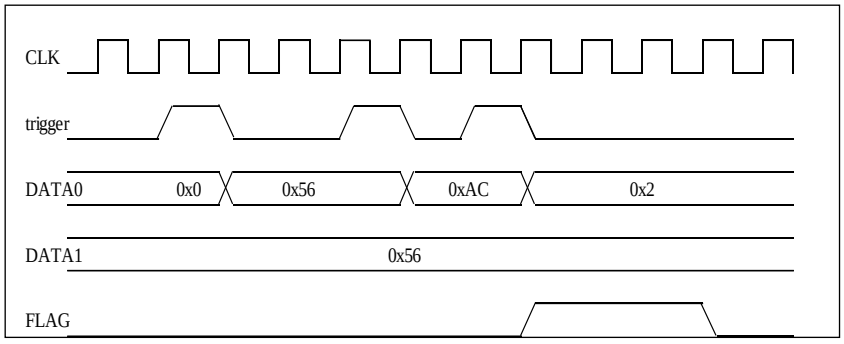
44.2.3 硬件触发启动模式

DCU 能够根据外围电路产生的事件触发启动运算。使用硬件触发启动模式时，需要先将功能时钟控制 0 寄存器(FCG0)的外围电路触发功能使能位置为有效。每个 DCU 单元可独立选择由其它外围电路送来的触发启动信号，选择启动信号时，在触发源选择寄存器(DCU_TRGSEL)中写入要选择的外围电路启动源的编号。当该外围电路事件发

生时，会将事件信号输入 DCU 并触发 DCU 启动开始运算。硬件触发启动模式包括触发加模式和触发减模式。在触发加模式时，每发生一次事件触发，DCU 将启动并进行一次(DATA0＋DATA1)/2 的运算，DATA0＋DATA1 的结果存入 DATA0，而(DATA0＋DATA1)/2 的结果存入 DATA2。当 DATA0＋DATA1 的结果超过 0xFF(8bit 模式)或 0xFFFF(16bit 模式)或 0xFFFFFFFF(32bit 模式)时产生标志位并产生中断。在触发减模式时，每发生一次事件触发，DCU 将启动并进行一次(DATA0－DATA1)/2 的运算，DATA0－DATA1 的结果存入 DATA0，而(DATA0-DATA1)/2 的结果存入 DATA2。当 DATA0－DATA1 的结果小于 0x0(8bit，16bit，32bit 模式)时产生标志位并产生中断。

硬件触发启动模式应用举例：

1. 控制寄存器 DCU_CTL 选择触发加模式，数据宽度为 8bit
2. 中断条件选择寄存器 DCU_INTSEL 选择运算条件
3. 分别在 DATA0 和 DATA1 中写入 0x00 和 0x56
4. 在触发源选择寄存器 DCU_TRGSEL 中写入事件编号
5. 使选中的外围电路动作并产生事件，DCU 由该事件触发并进行一次加法运算，结果为 0x56 并保存在 DATA0 中
6. 在连续 3 次触发后，计算结果溢出，产生结果标志。读取标志寄存器 DCU_FLAG 得到结果标志
7. 写标志复位寄存器 DCU_FLAGCLR 清除标志位



44.2.4 比较模式

比较 DATA0 和 DATA1 以及 DATA0 和 DATA2 的大小，可选择当 DATA0 大于 DATA1，DATA0 小于 DATA1，DATA0 等于 DATA1 时以及当 DATA0 大于 DATA2，DATA0 小于 DATA2，DATA0 等于 DATA2 时产生标志位并产生中断。比较模式下可以选择数据

开始比较的条件，写 DATA0 后比较或写任何数据寄存器后比较。

比较模式应用举例：

1. 控制寄存器 DCU_CTL 选择比较模式，数据宽度为 16bit。在写入 DATA0 后开始比较。
2. 中断条件选择寄存器 DCU_INTSEL 选择 DATA0＞DATA1 时产生标志
3. 在 DATA1 和 DATA0 中分别写入 0xBBBB 和 0xAAAA，此时由于不满足 DATA0＞DATA1 因此不产生标志。
4. 在 DATA1 写入 0x8888，此时虽然满足 DATA0＞DATA1，但由于设置为写入 DATA0 后开始比较，因此也不产生标志。
5. DATA0 写入 0x9999 后满足标志产生条件，产生标志，
6. 写标志复位寄存器 DCU_FLAGCLR 清除标志位。

44.2.5 中断和事件信号输出

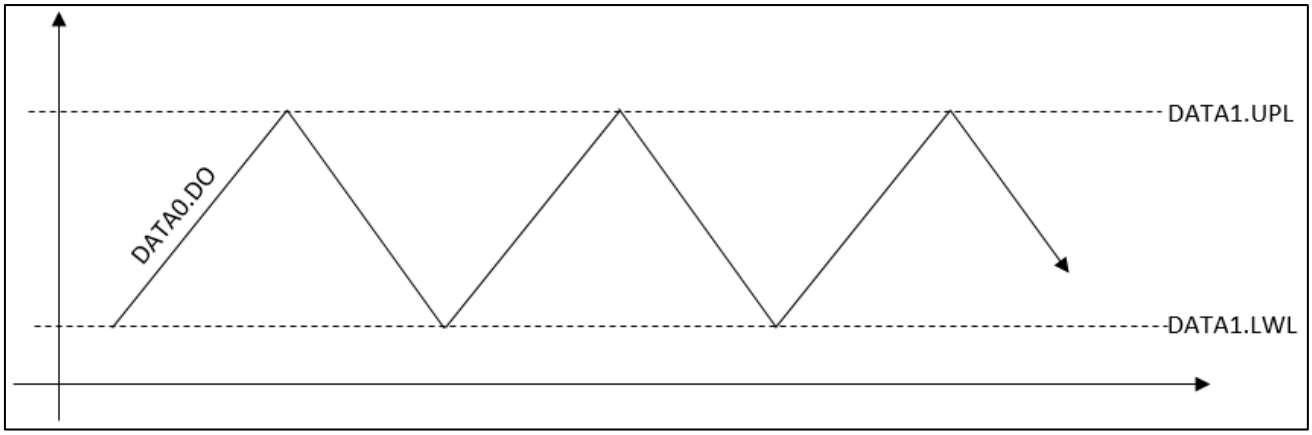
DCU 具有多种中断和用于触发启动其他外围电路的事件输出供用户选择。中断和事件输出的控制由中断和事件寄存器(DCU_INTEVT)控制，当需要输出事件信号时，用户需要将中断和事件寄存器(DCU_INTEVT)的相应控制位置为有效。每个 DCU 单元各输出一个 DCU 事件信号，在事件列表中分别为 DCU1~DCU8。当需要对应事件在发生时产生中断，用户需要将中断和事件寄存器(DCU_INTEVT)的相应控制位置为有效的同时，将控制寄存器(DCU_CTL)的 INTEN 位置 1。每个 DCU 单元各输出一个 DCU 中断信号，在中断列表中分别为 DCU1~DCU8。

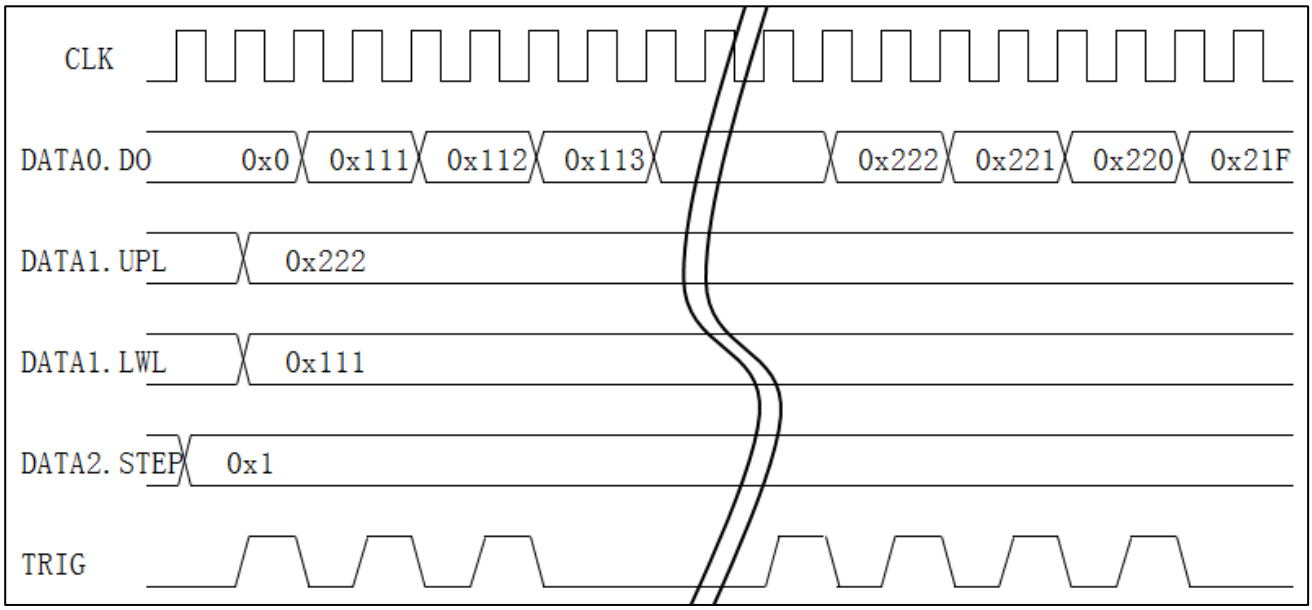
44.2.6 三角波输出模式

三角波输出模式是 DCU 通过定时器作为触发源，为数模转换模块(DAC)提供一个连续变化的数字量，DAC 将这些数字量转化为三角波输出。三角波输出模式通过 DCU 控制寄存器(DCU_CTL)设定，设定成三角波输出模式后，数据寄存器 DATA0~2 的功能将发生变化，数据寄存器 DATA0 将作为输出给 DAC 的数字量的数据寄存器并复位。DATA0 变为只读寄存器,DATA.DO(b0~b11)为有效位。DATA1 变为振幅设定寄存器，DATA1.UPL(b16~b27)和 DATA1.LWL(b0~b11)为有效位。DATA2 变为步长设定寄存器，DATA2.STEP(b0~b11)为有效位。仅 DCU1~4 具有三角波输出模式。

使用三角波输出模式步骤如下

1. 将 DCU 控制寄存器(DCU_CTL)的 MODE 设为 1000b
2. 设定三角波振幅的上限和下限。上限通过 DATA1.UPL[11:0]位设定，下限通过 DATA1.LWL[11:0]设定
3. 设定三角波递增和递减的步长值，通过 DATA2.STEP[11:0]设定
4. 通过触发源选择寄存器(DCU_TRGSEL)选择合适的定时器事件或软件触发作为触发源
5. 配置选定的定时器，使其在设定的时间间隔上产生触发时间
6. 启动定时器，第一次产生触发事件时，DATA1.LWL 的值加载到 DATA0.DO 并输出给 DAC，以后每次产生触发事件时，DATA0.DO 将根据 DATA2.STEP 设定的步长值递增
7. 当 DATA0.DO 递增至 DATA1.UPL 设定的上限时，将自动根据 DATA2.STEP 设定的步长值开始递减。当递减至 DATA1.LWL 设定的下限后重新开始递增
8. 如果需要在三角波到达峰点或谷点时发生中断，请先将 DCU 控制寄存器中的中断使能位(INTEN)置为无效，再设置 DATA0~DATA2 的值，在启动定时器开始产生触发事件前清除一次 DCU 标志位后，将 DCU 控制寄存器中的中断使能位(INTEN)置为有效后，再启动定时器
9. 将 DCU 控制寄存器(DCU_CTL)的 MODE 设为 0000b 以退出三角波输出模式


DATA1.UPL
DATA0.DO
DATA1.LWL


CLK
DATA0.DO
0x0
0x111
0x112
0x113
0x222
0x221
0x220
0x21F
DATA1.UPL
0x222
DATA1.LWL
0x111
DATA2.STEP
0x1
TRIG

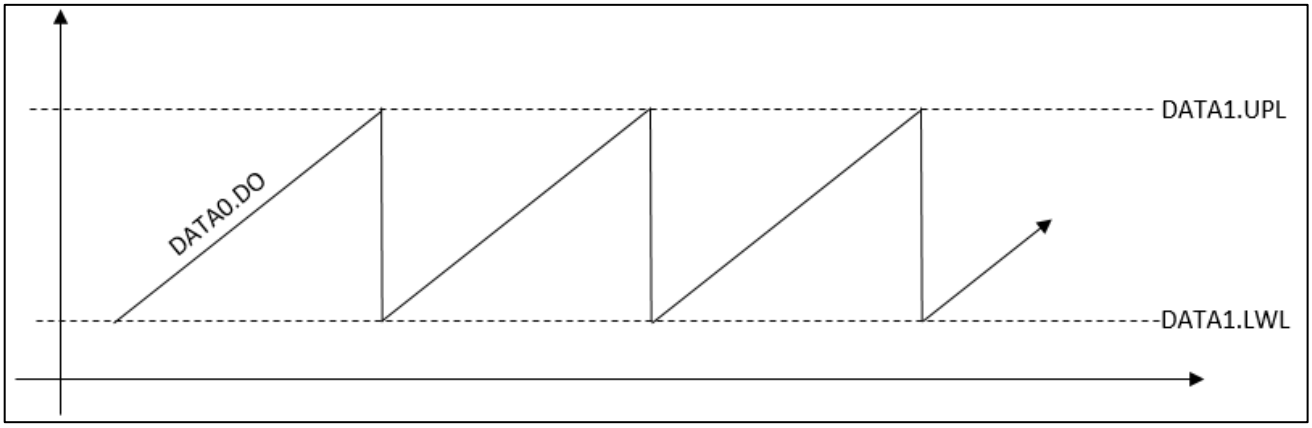
44.2.7 递增锯齿波输出模式

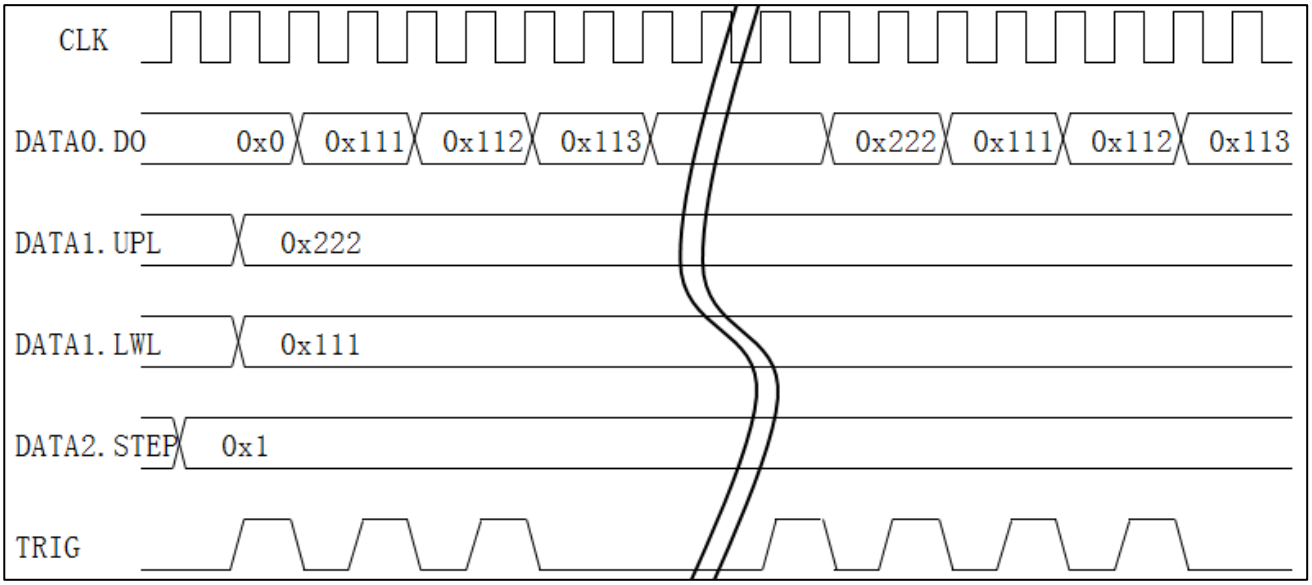
递增锯齿波输出模式是 DCU 通过定时器作为触发源，为数模转换模块(DAC)提供一个连续变化的数字量，DAC 将这些数字量转化为锯齿波输出。递增锯齿波输出模式通过 DCU 控制寄存器(DCU_CTL)设定，设定成递增锯齿波输出模式后，数据寄存器 DATA0~2 的功能将发生变化，数据寄存器 DATA0 将作为输出给 DAC 的数字量的数据寄存器并复位。DATA0 变为只读寄存器,DATA.DO(b0~b11)为有效位。DATA1 变为振幅设定寄存器，DATA1.UPL(b16~b27)和 DATA1.LWL(b0~b11)为有效位。DATA2 变为步长设定寄存器，DATA2.STEP(b0~b11)为有效位。仅 DCU1~4 具有递增锯齿波输出模式。

使用递增锯齿波输出模式步骤如下

1. 将 DCU 控制寄存器(DCU_CTL)的 MODE 设为 1001b
2. 设定锯齿波振幅的上限和下限。上限通过 DATA1.UPL[11:0]位设定，下限通过 DATA1.LWL[11:0]设定
3. 设定锯齿波递增的步长值，通过 DATA2.STEP[11:0]设定
4. 通过触发源选择寄存器(DCU_TRGSEL)选择合适的定时器事件或软件触发作为触发源
5. 配置选定的定时器，使其在设定的时间间隔上产生触发时间
6. 启动定时器，第一次产生触发事件时，DATA1.LWL 的值加载到 DATA0.DO 并输出给 DAC，以后每次产生触发事件时，DATA0.DO 将根据 DATA2.STEP 设定的步长值递增
7. 当 DATA0.DO 递增至 DATA1.UPL 设定的上限时，将自动重新加载 DATA1.LWL 到 DATA0.DO，使其从 DATA1.LWL 设定的下限重新开始递增
8. 如果需要在锯齿波重载时发生中断，请先将 DCU 控制寄存器中的中断使能位(INTEN)置为无效，再设置 DATA0~DATA2 的值，在启动定时器开始产生触发事件前清除一次 DCU 标志位后，将 DCU 控制寄存器中的中断使能位(INTEN)置为有效后，再启动定时器
9. 将 DCU 控制寄存器(DCU_CTL)的 MODE 设为 0000b 以退出锯齿波输出模式

10. 将 DCU 控制寄存器(DCU_CTL)的 MODE 设为 0000b 以退出锯齿波输出模式



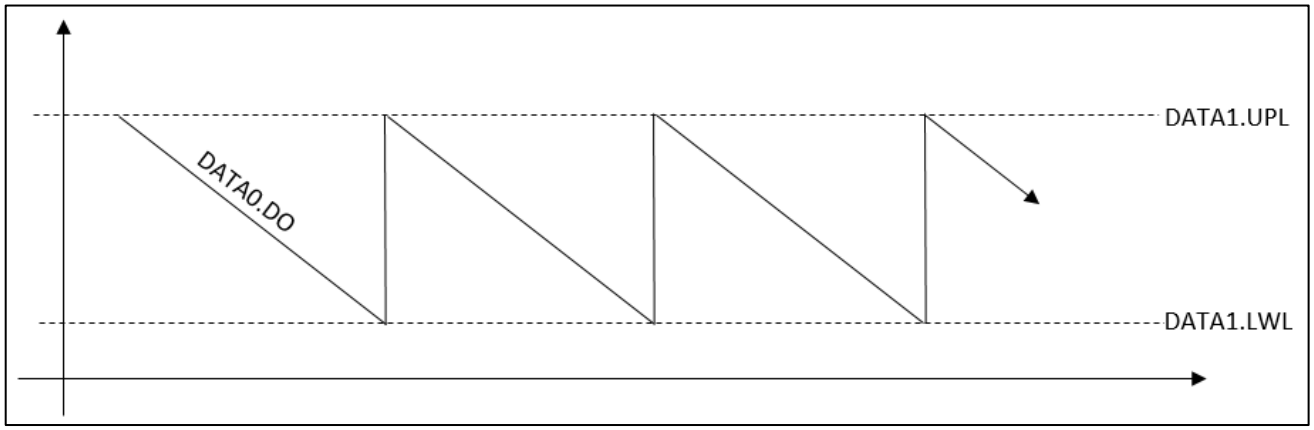


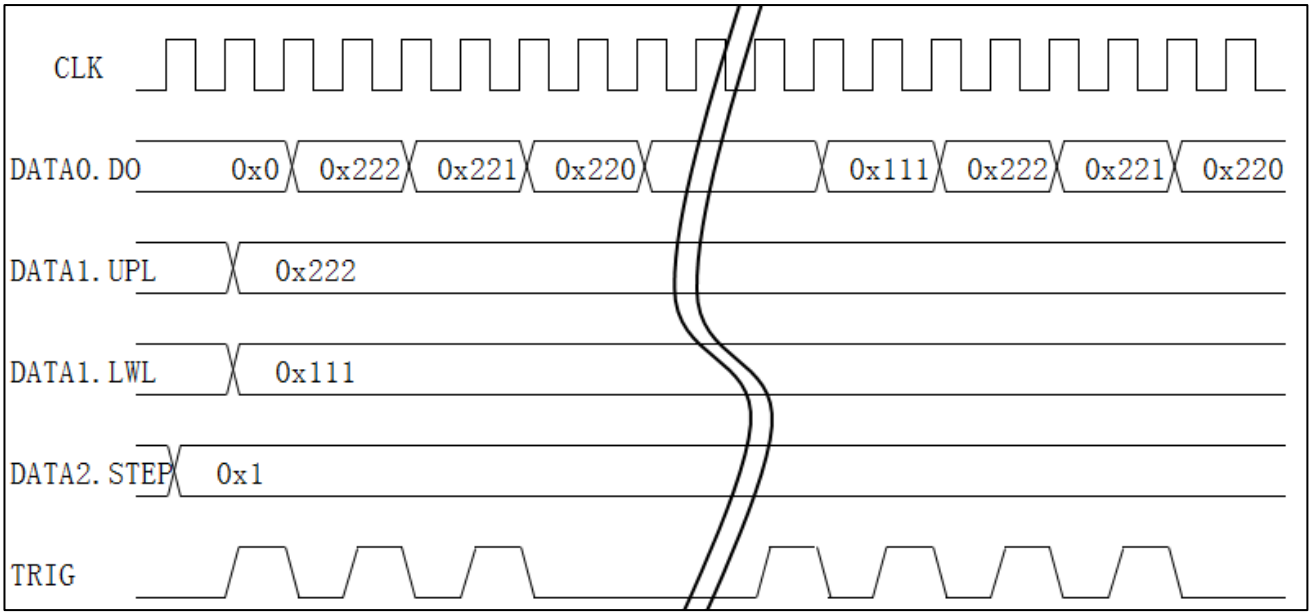
44.2.8 递减锯齿波输出模式

递减锯齿波输出模式是 DCU 通过定时器作为触发源，为数模转换模块(DAC)提供一个连续变化的数字量，DAC 将这些数字量转化为锯齿波输出。递减锯齿波输出模式通过 DCU 控制寄存器(DCU_CTL)设定，设定成递减锯齿波输出模式后，数据寄存器 DATA0~2 的功能将发生变化，数据寄存器 DATA0 将作为输出给 DAC 的数字量的数据寄存器并复位。DATA0 变为只读寄存器,DATA.DO(b0~b11)为有效位。DATA1 变为振幅设定寄存器，DATA1.UPL(b16~b27)和 DATA1.LWL(b0~b11)为有效位。DATA2 变为步长设定寄存器，DATA2.STEP(b0~b11)为有效位。仅 DCU1~4 具有递减锯齿波输出模式。

使用递减锯齿波输出模式步骤如下

1. 将 DCU 控制寄存器(DCU_CTL)的 MODE 设为 1010b，
2. 设定锯齿波振幅的上限和下限。上限通过 DATA1.UPL[11:0]位设定，下限通过 DATA1.LWL[11:0]设定
3. 设定锯齿波递减的步长值，通过 DATA2.STEP[11:0]设定
4. 通过触发源选择寄存器(DCU_TRGSEL)选择合适的定时器事件或软件触发作为触发源
5. 配置选定的定时器，使其在设定的时间间隔上产生触发时间
6. 启动定时器，第一次产生触发事件时，DATA1.UPL 的值加载到 DATA0.DO 并输出给 DAC，以后每次产生触发事件时，DATA0.DO 将根据 DATA2.STEP 设定的步长值递增
7. 当 DATA0.DO 递减至 DATA1.LWL 设定的下限时，将自动重新加载 DATA1.UPL 到 DATA0.DO，使其从 DATA1.UPL 设定的下限重新开始递减
8. 如果需要在锯齿波重载时发生中断，请先将 DCU 控制寄存器中的中断使能位(INTEN)置为无效，再设置 DATA0~DATA2 的值，在启动定时器开始产生触发事件前清除一次 DCU 标志位后，将 DCU 控制寄存器中的中断使能位(INTEN)置为有效后，再启动定时器
9. 将 DCU 控制寄存器(DCU_CTL)的 MODE 设为 0000b 以退出锯齿波输出模式




CLK
DATA0. DO
0x0
0x222
0x221
0x220
0x111
0x222
0x221
0x220
DATA1. UPL
0x222
DATA1. LWL
0x111
DATA2. STEP
0x1
TRIG

44.3 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	0x40056000
DCU1 标志寄存器	DCU1_FLAG	DCU 的结果标识	0x40056004
DCU1 数据寄存器 0	DCU1_DATA0	存放运算数据	0x40056008
DCU1 数据寄存器 1	DCU1_DATA1	存放运算数据	0x4005600C
DCU1 数据寄存器 2	DCU1_DATA2	存放运算数据	0x40056010
DCU1 标志复位寄存器	DCU1_FLAGCLR	清除 DCU 的结果标识	0x40056014
DCU1 中断和事件寄存器	DCU1_INTEVT	DCU 中断和事件的条件	0x40056018
DCU1 触发源选择寄存器*	DCU1_TRGSEL	选择硬件触发启动模式的触发源	0x40010804

*:与 DCU5 共用同一个寄存器

单元 2

名称	英文缩写	说明	地址
DCU2 控制寄存器	DCU2_CTL	配置 DCU 的动作模式	0x40056400
DCU2 标志寄存器	DCU2_FLAG	DCU 的结果标识	0x40056404
DCU2 数据寄存器 0	DCU2_DATA0	存放运算数据	0x40056408
DCU2 数据寄存器 1	DCU2_DATA1	存放运算数据	0x4005640C
DCU2 数据寄存器 2	DCU2_DATA2	存放运算数据	0x40056410
DCU2 标志复位寄存器	DCU2_FLAGCLR	清除 DCU 的结果标识	0x40056414
DCU2 中断和事件寄存器	DCU2_INTEVT	DCU 中断和事件的条件	0x40056418

DCU2 触发源选择寄存器*	DCU2_TRGSEL	选择硬件触发启动模式的触发源	0x40010808

*:与 DCU6 共用同一个寄存器

单元 3

名称	英文缩写	说明	地址
DCU3 控制寄存器	DCU3_CTL	配置 DCU 的动作模式	0x40056800
DCU3 标志寄存器	DCU3_FLAG	DCU 的结果标识	0x40056804
DCU3 数据寄存器 0	DCU3_DATA0	存放运算数据	0x40056808
DCU3 数据寄存器 1	DCU3_DATA1	存放运算数据	0x4005680C
DCU3 数据寄存器 2	DCU3_DATA2	存放运算数据	0x40056810
DCU3 标志复位寄存器	DCU3_FLAGCLR	清除 DCU 的结果标识	0x40056814
DCU3 中断和事件寄存器	DCU3_INTEVT	DCU 中断和事件的条件	0x40056818
DCU3 触发源选择寄存器*	DCU3_TRGSEL	选择硬件触发启动模式的触发源	0x4001080C

*:与 DCU7 共用同一个寄存器

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	0x40056C00
DCU4 标志寄存器	DCU4_FLAG	DCU 的结果标识	0x40056C04
DCU4 数据寄存器 0	DCU4_DATA0	存放运算数据	0x40056C08
DCU4 数据寄存器 1	DCU4_DATA1	存放运算数据	0x40056C0C
DCU4 数据寄存器 2	DCU4_DATA2	存放运算数据	0x40056C10
DCU4 标志复位寄存器	DCU4_FLAGCLR	清除 DCU 的结果标识	0x40056C14

DCU4 中断和事件寄存器	DCU4_INTEVT	DCU 中断和事件的条件	0x40056C18
DCU4 触发源选择寄存器*	DCU4_TRGSEL	选择硬件触发启动模式的触发源	0x40010810

*:与 DCU8 共用同一个寄存器

单元 5

名称	英文缩写	说明	地址
DCU5 控制寄存器	DCU5_CTL	配置 DCU 的动作模式	0x40057000
DCU5 标志寄存器	DCU5_FLAG	DCU 的结果标识	0x40057004
DCU5 数据寄存器 0	DCU5_DATA0	存放运算数据	0x40057008
DCU5 数据寄存器 1	DCU5_DATA1	存放运算数据	0x4005700C
DCU5 数据寄存器 2	DCU5_DATA2	存放运算数据	0x40057010
DCU5 标志复位寄存器	DCU5_FLAGCLR	清除 DCU 的结果标识	0x40057014
DCU5 中断和事件寄存器	DCU5_INTEVT	DCU 中断和事件的条件	0x40057018
DCU5 触发源选择寄存器*	DCU1_TRGSEL	选择硬件触发启动模式的触发源	0x40010804

*:与 DCU1 共用同一个寄存器

单元 6

名称	英文缩写	说明	地址
DCU6 控制寄存器	DCU6_CTL	配置 DCU 的动作模式	0x40057400
DCU6 标志寄存器	DCU6_FLAG	DCU 的结果标识	0x40057404
DCU6 数据寄存器 0	DCU6_DATA0	存放运算数据	0x40057408
DCU6 数据寄存器 1	DCU6_DATA1	存放运算数据	0x4005740C
DCU6 数据寄存器 2	DCU6_DATA2	存放运算数据	0x40057410

DCU6 标志复位寄存器	DCU6_FLAGCLR	清除 DCU 的结果标识	0x40057414
DCU6 中断和事件寄存器	DCU6_INTEVT	DCU 中断和事件的条件	0x40057418
DCU6 触发源选择寄存器*	DCU6_TRGSEL	选择硬件触发启动模式的触发源	0x40010808

*:与 DCU2 共用同一个寄存器

单元 7

名称	英文缩写	说明	地址
DCU7 控制寄存器	DCU7_CTL	配置 DCU 的动作模式	0x40057800
DCU7 标志寄存器	DCU7_FLAG	DCU 的结果标识	0x40057804
DCU7 数据寄存器 0	DCU7_DATA0	存放运算数据	0x40057808
DCU7 数据寄存器 1	DCU7_DATA1	存放运算数据	0x4005780C
DCU7 数据寄存器 2	DCU7_DATA2	存放运算数据	0x40057810
DCU7 标志复位寄存器	DCU7_FLAGCLR	清除 DCU 的结果标识	0x40057814
DCU7 中断和事件寄存器	DCU7_INTEVT	DCU 中断和事件的条件	0x40057818
DCU7 触发源选择寄存器*	DCU7_TRGSEL	选择硬件触发启动模式的触发源	0x4001080C

*:与 DCU3 共用同一个寄存器

单元 8

名称	英文缩写	说明	地址
DCU8 控制寄存器	DCU8_CTL	配置 DCU 的动作模式	0x40057C00
DCU8 标志寄存器	DCU8_FLAG	DCU 的结果标识	0x40057C04
DCU8 数据寄存器 0	DCU8_DATA0	存放运算数据	0x40057C08
DCU8 数据寄存器 1	DCU8_DATA1	存放运算数据	0x40057C0C

DCU8 数据寄存器 2	DCU8_DATA2	存放运算数据	0x40057C10
DCU8 标志复位寄存器	DCU8_FLAGCLR	清除 DCU 的结果标识	0x40057C14
DCU8 中断和事件寄存器	DCU8_INTEVT	DCU 中断和事件的条件	0x40057C18
DCU8 触发源选择寄存器*	DCU8_TRGSEL	选择硬件触发启动模式的触发源	0x40010810

*:与 DCU4 共用同一个寄存器

44.3.1 DCU 控制寄存器(DCUx_CTL)(x=1~8)

寄存器说明：该寄存器用来配置 DCU 的动作模式

复位值：0x80000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							COMPTRG	Reserved		DATASIZE[1:0]		MODE[3:0]			

位	标记	位名	功能	读写
b31	INTEN	中断使能	0：不允许产生中断 1：允许产生中断	R/W
b30~b9	Reserved	-	读时读出0，写时请写0	R/W
b8	COMPTRG	比较模式触发比较的时机	0：写入DATA0后比较 1：写入DATA0或DATA1或DATA2后比较	R/W
b7~b6	Reserved	-	读时读出0，写时请写0	R/W
b5~b4	DATASIZE[1:0]	加/减法模式，比较模式数据大小	00：8bit 01：16bit 10：32bit	R/W
b3~b0	MODE[2:0]	动作模式	0000：DCU无效 0001：加法模式，运算在DATA1寄存器写入数据后进行 0010：减法模式，运算在DATA1寄存器写入数据后进行 0011：硬件触发加法模式，由其他外围电路触发启动进行加法运算 0100：硬件触发减法模式，由其他外围电路触发启动进行减法运算 0101：比较模式 1000*：三角波输出模式 1001*：递增锯齿波输出模式 1010*：递减锯齿波输出模式 其它：设定禁止	R/W

*:仅 DCU1/DCU2/DCU3/DCU4 有此功能

44.3.2 DCU 标志寄存器(DCUx_FLAG)(x=1~4)

寄存器说明：该寄存器产生 DCU 的结果标识

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				FLAT_TOP	FLAG_BTM	FLAG_RLD	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~b12	Reserved	-	读时读出0，写时请写0	R/W
b11	FLAT_TOP	三角波峰点	三角波输出模式下，当DATA0[11:0]的值达到最大时置位，DCU标志复位寄存器DCU_FLAGCLR的CLR_TOP位写1时清除	R
b10	FLAG_BTM	三角波谷点	三角波输出模式下，当DATA0[11:0]的值达到最小时置位，DCU标志复位寄存器DCU_FLAGCLR的CLR_TOP位写1时清除	R
b9	FLAG_RLD	锯齿波重载	递增锯齿波输出模式下，当DATA0[11:0]的值达到最大时置位； 递减锯齿波输出模式下，当DATA0[11:0]的值达到最小时置位。 DCU标志复位寄存器DCU_FLAGCLR的CLR_RLD位写1时清除	R
b8~b7	Reserved	-	读时读出0，写时请写0	R/W
b6	FLAG_GT1	大于标志位1	比较模式下，当DATA0＞DATA1时置位，DCU标志复位寄存器DCU_FLAGCLR的CLR_GT1位写1时清除	R
b5	FLAG_EQ1	等于标志位1	比较模式下，当DATA0＝DATA1时置位，DCU标志复位寄存器DCU_FLAGCLR的CLR_EQ1位写1时清除	R
b4	FLAG_LS1	小于标志位1	比较模式下，当DATA0＜DATA1时置位，DCU标志复位寄存器DCU_FLAGCLR的CLR_LS1位写1时清除	R
b3	FLAG_GT2	大于标志位2	比较模式下，当DATA0＞DATA2时置位，DCU标志复位寄存器DCU_FLAGCLR的CLR_GT2位写1时清除	R
b2	FLAG_EQ2	等于标志位2	比较模式下，当DATA0＝DATA2时置位，DCU标志复位寄存器DCU_FLAGCLR的CLR_EQ2位写1时清除	R
b1	FLAG_LS2	小于标志位2	比较模式下，当DATA0＜DATA2时置位，DCU标志复位寄存器DCU_FLAGCLR的CLR_LS2位写1时清除	R
b0	FLAG_OP	运算标志位	加法，减法和触发加，触发减模式时，当加法产生上溢或减法产生下溢时置位，DCU标志复位寄存器DCU_FLAGCLR的CLR_OP位写1时清除	R

44.3.3 DCU 标志寄存器(DCUx_FLAG)(x=5~8)

寄存器说明：该寄存器产生 DCU 的结果标识

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved									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/W
b6	FLAG_GT1	大于标志位1	比较模式下，当DATA0＞DATA1时置位，DCU标志复位寄存器DCU_FLAGCLR的CLR_GT1位写1时清除	R
b5	FLAG_EQ1	等于标志位1	比较模式下，当DATA0＝DATA1时置位，DCU标志复位寄存器DCU_FLAGCLR的CLR_EQ1位写1时清除	R
b4	FLAG_LS1	小于标志位1	比较模式下，当DATA0＜DATA1时置位，DCU标志复位寄存器DCU_FLAGCLR的CLR_LS1位写1时清除	R
b3	FLAG_GT2	大于标志位2	比较模式下，当DATA0＞DATA2时置位，DCU标志复位寄存器DCU_FLAGCLR的CLR_GT2位写1时清除	R
b2	FLAG_EQ2	等于标志位2	比较模式下，当DATA0＝DATA2时置位，DCU标志复位寄存器DCU_FLAGCLR的CLR_EQ2位写1时清除	R
b1	FLAG_LS2	小于标志位2	比较模式下，当DATA0＜DATA2时置位，DCU标志复位寄存器DCU_FLAGCLR的CLR_LS2位写1时清除	R
b0	FLAG_OP	运算标志位	加法，减法和触发加，触发减模式时，当加法产生上溢或减法产生下溢时置位，DCU标志复位寄存器DCU_FLAGCLR的CLR_OP位写1时清除	R

44.3.4 DCU 数据寄存器(DCUx_DATAy)(x=1~8,y=0,1,2)

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
三角波输出模式*	输出数据寄存器	振幅设置寄存器	步长寄存器
递增锯齿波输出模式*	输出数据寄存器	振幅设置寄存器	步长寄存器
递减锯齿波输出模式*	输出数据寄存器	振幅设置寄存器	步长寄存器

*:仅 DCU1/DCU2/DCU3/DCU4 有此功能

复位值：0x00000000

加/减法模式，触发加/减法模式，比较模式下 DCUx_DATAy (x=1~8,y=0,1,2)用来存放运算数据。

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

三角波输出模式 DCUx_DATA0(x=1~4)作为输出数据寄存器，b0~b11 为有效位

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				DO[11:0]											

位	标记	位名	功能	读写
b31~b12	Reserved	-	读时读出0，写时请写0	R/W
b11~b0	DO[11:0]	输出数据	输出到DAC的数据	R/W

三角波输出模式 DCUx_DATA1(x=1~4)作为振幅设定寄存器，b16~b27 及 b0~b11 为有效位

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

位	标记	位名	功能	读写
b31~b28	Reserved	-	读时读出0，写时请写0	R/W
b27~b16	UPL[11:0]	振幅上限	设定三角波振幅的上限	R/W
b15~b12	Reserved	-	读时读出0，写时请写0	R/W
b11~b0	LWL[11:0]	振幅下限	设定三角波振幅的下限	R/W

三角波输出模式 DCUx_DATA2(x=1~4)作为步长寄存器，b0~b11 为有效位

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				STEP [11:0]											

位	标记	位名	功能	读写
b31~b12	Reserved	-	读时读出0，写时请写0	R
b11~b0	STEP[11:0]	步长设置	设定三角波递增和递减的步长值	R/W

递增/递减锯齿波输出模式 DCUx_DATA0(x=1~4)作为输出数据寄存器，b0~b11 为有效位

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				DO[11:0]											

位	标记	位名	功能	读写
b31~b12	Reserved	-	读时读出0，写时请写0	R/W
b11~b0	DO[11:0]	输出数据	输出到DAC的数据	R

递增/递减锯齿波输出模式 DCUx_DATA1(x=1~4)作为振幅设置寄存器，b16~b27 及 b0~b11 为有效位

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

位	标记	位名	功能	读写
b31~b28	Reserved	-	读时读出0，写时请写0	R/W
b27~b16	UPL[11:0]	振幅上限	设定锯齿波振幅的上限	R/W
b15~b12	Reserved	-	读时读出0，写时请写0	R/W
b11~b0	LWL[11:0]	振幅下限	设定锯齿波振幅的下限	R/W

递增/递减锯齿波输出模式 DCUx_DATA2(x=1~4)作为步长寄存器，b0~b11 为有效位

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				STEP [11:0]											

位	标记	位名	功能	读写
b31~b12	Reserved	-	读时读出0，写时请写0	R/W
b11~b0	STEP[11:0]	步长设置	设定锯齿波递增或递减的步长值	R/W

44.3.5 DCU 标志复位寄存器(DCUx_FLAGCLR) (x=1~4)

寄存器说明：该寄存器用于清除 DCU 的结果标识

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				CLR_TOP	CLR_BTM	CLR_RLD	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~b12	Reserved	-	读时读出0，写时请写0	R/W
b11	CLR_TOP	清除三角波峰点标志位	写1时清除DCU_FLAG的FLAG_TOP位，写0无任何效果 读该寄存器位无任何效果	R/W
b10	CLR_BTM	清除三角波谷点标志位	写1时清除DCU_FLAG的FLAG_BTM位，写0无任何效果 读该寄存器位无任何效果	R/W
b9	CLR_RLD	清除锯齿波重载标志位	写1时清除DCU_FLAG的FLAG_RLD位，写0无任何效果 读该寄存器位无任何效果	R/W
b8~b7	Reserved	-	读时读出0，写时请写0	R/W
b6	CLR_GT1	清除大于标志位1	写1时清除DCU_FLAG的FLAG_GT1位，写0无任何效果 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/W

44.3.6 DCU 标志复位寄存器(DCUx_FLAGCLR) (x=5~8)

寄存器说明：该寄存器用于清除 DCU 的结果标识

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved									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/W
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 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/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 读该寄存器位无任何效果	R/W

44.3.7 DCU 中断和事件寄存器(DCUx_INTEVTSEL) (x=1~4)

寄存器说明：该寄存器能够选择 DCU 在何种条件下产生中断和输出事件信号

复位值：0x00000000

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

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				SEL_TOP	SEL_BTM	SEL_RLD	SEL_WIN[1:0]		SEL_GT1	SEL_EQ1	SEL_LS1	SEL_GT2	SEL_EQ2	SEL_LS2	SEL_OP

位	标记	位名	功能	读写
b31~b12	Reserved	-	读时读出0，写时请写0	R/W
b11	SEL_TOP	三角波峰点条件选择	1：在三角波输出模式下当DATA0[11:0]达到最大时产生中断和输出事件信号 0：在三角波和输出模式下当DATA0[11:0]达到最大时不产生中断和输出事件信号	R/W
b10	SEL_BTM	三角波谷点条件选择	1：在三角波输出模式下当DATA0[11:0]达到最小值时产生中断和输出事件信号 0：在三角波输出模式下当DATA0[11:0]达到最小值时不产生中断和输出事件信号	R/W
b9	SEL_RLD	锯齿波重载条件选择	1：在递增锯齿波输出模式下当DATA0[11:0]达到最大值时产生中断和输出事件信号；在递减锯齿波输出模式下当DATA0[11:0]达到最小值时产生中断和输出事件信号 0：在递增/递减锯齿波输出模式下不产生中断和输出事件信号	R/W
b8~b7	SEL_WIN[1:0]	窗口比较条件选择	比较模式下满足INT_WIN设定的窗口比较条件时产生中断和输出事件信号，在INT_WIN设定有效时，满足其他比较条件时不会产生中断和输出事件信号 00：不产生窗口比较中断和输出事件信号，此设定下由该寄存器的b1~b6选择其他中断和事件信号产生条件 01：当DATA0数据在窗口内时产生中断和输出事件信号，即DATA2≤DATA0≤DATA1 10：当DATA0数据在窗口外时产生中断和输出事件信号，即DATA0＞DATA1或DATA0＜DATA2 11：比较模式下不产生任何中断和事件信号	R/W
b6	SEL_GT1	大于条件选择1	1：比较模式下并且INT_WIN=00时，当DATA0＞DATA1时产生中断和输出事件信号 0：比较模式下并且INT_WIN=00时，当DATA0＞DATA1时不产生中断和输出事件信号 INT_WIN≠00时该位无效	R/W

b5	SEL _EQ1	等于条件选择1	1：比较模式下并且INT_WIN=00时，当DATA0＝DATA1时产生中断和输出事件信号 0：比较模式下并且INT_WIN=00时，当DATA0＝DATA1时不产生中断和输出事件信号 INT_WIN≠00时该位无效	R/W
b4	SEL_LS1	小于条件选择1	1：比较模式下并且INT_WIN=00时，当DATA0＜DATA1时产生中断和输出事件信号 0：比较模式下并且INT_WIN=00时，当DATA0＜DATA1时不产生中断和输出事件信号 INT_WIN≠00时该位无效	R/W
b3	SEL_GT2	大于条件选择2	1：比较模式下并且INT_WIN=00时，当DATA0＞DATA2时产生中断和输出事件信号 0：比较模式下并且INT_WIN=00时，当DATA0＞DATA2时不产生中断和输出事件信号 INT_WIN≠00时该位无效	R/W
b2	SEL_EQ2	等于条件选择2	1：比较模式下并且INT_WIN=00时，当DATA0＝DATA2时产生中断和输出事件信号 0：比较模式下并且INT_WIN=00时，当DATA0＝DATA2时不产生中断和输出事件信号 INT_WIN≠00时该位无效	R/W
b1	SEL_LS2	小于条件选择2	1：比较模式下并且INT_WIN=00时，当DATA0＜DATA2时产生中断和输出事件信号 0：比较模式下并且INT_WIN=00时，当DATA0＜DATA2时不产生中断和输出事件信号 INT_WIN≠00时该位无效	R/W
b0	SEL_OP	运算条件选择	1：加法和减法模式下当运算结果上溢或下溢时产生中断和输出事件信号 0：加法和减法模式下当运算结果上溢或下溢时不产生中断和输出事件信号	R/W

44.3.8 DCU 中断和事件寄存器(DCUx_INTEVTSEL) (x=5~8)

寄存器说明：该寄存器能够选择 DCU 在何种条件下产生中断和输出事件信号

复位值：0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							SEL_WIN[1:0]		SEL _GT1	SEL _EQ1	SEL _LS1	SEL _GT2	SEL _EQ2	SEL _LS2	SEL _OP

位	标记	位名	功能	读写
b31~b9	Reserved	-	读时读出0，写时请写0	R/W
b8~b7	SEL_WIN[1:0]	窗口比较条件选择	比较模式下满足INT_WIN设定的窗口比较条件时产生中断和输出事件信号，在INT_WIN设定有效时，满足其他比较条件时不会产生中断和输出事件信号 00：不产生窗口比较中断和输出事件信号，此设定下由该寄存器的b1~b6选择其他中断和事件信号产生条件 01：当DATA0数据在窗口内时产生中断和输出事件信号，即DATA2≤DATA0≤DATA1 10：当DATA0数据在窗口外时产生中断和输出事件信号，即DATA0＞DATA1或DATA0＜DATA2 11：比较模式下不产生任何中断和事件信号	R/W
b6	SEL _GT1	大于条件选择1	1：比较模式下并且INT_WIN=00时，当DATA0＞DATA1时产生中断和输出事件信号 0：比较模式下并且INT_WIN=00时，当DATA0＞DATA1时不产生中断和输出事件信号 INT_WIN≠00时该位无效	R/W
b5	SEL _EQ1	等于条件选择1	1：比较模式下并且INT_WIN=00时，当DATA0＝DATA1时产生中断和输出事件信号 0：比较模式下并且INT_WIN=00时，当DATA0＝DATA1时不产生中断和输出事件信号 INT_WIN≠00时该位无效	R/W
b4	SEL_LS1	小于条件选择1	1：比较模式下并且INT_WIN=00时，当DATA0＜DATA1时产生中断和输出事件信号 0：比较模式下并且INT_WIN=00时，当DATA0＜DATA1时不产生中断和输出事件信号 INT_WIN≠00时该位无效	R/W
b3	SEL_GT2	大于条件选择2	1：比较模式下并且INT_WIN=00时，当DATA0＞DATA2时产生中断和输出事件信号 0：比较模式下并且INT_WIN=00时，当DATA0＞DATA2时不产生中断	R/W

			和输出事件信号 INT_WIN≠00时该位无效	
b2	SEL_EQ2	等于条件选择2	1：比较模式下并且INT_WIN=00时，当DATA0＝DATA2时产生中断和输出事件信号 0：比较模式下并且INT_WIN=00时，当DATA0＝DATA2时不产生中断和输出事件信号 INT_WIN≠00时该位无效	R/W
b1	SEL_LS2	小于条件选择2	1：比较模式下并且INT_WIN=00时，当DATA0＜DATA2时产生中断和输出事件信号 0：比较模式下并且INT_WIN=00时，当DATA0＜DATA2时不产生中断和输出事件信号 INT_WIN≠00时该位无效	R/W
b0	SEL_OP	运算条件选择	1：加法和减法模式下当运算结果上溢或下溢时产生中断和输出事件信号 0：加法和减法模式下当运算结果上溢或下溢时不产生中断和输出事件信号	R/W

44.3.9 DCU 触发源选择寄存器(DCUx_TRGSEL)(x=1~4)

寄存器说明：在 DCU 选择硬件触发启动模式，三角波输出模式或递增/递减锯齿波输出模式（仅限 DCU1~4）后，将要产生触发的事件的编号写入该寄存器中，当编号对应的外围电路事件发生后，DCU 将由该事件触发启动并进行运算。DCU1 将和 DCU5 共用寄存器 DCU1_TRGSEL，即当 DCU1_TRGSEL 写入事件编号，该编号事件发生时，会同时触发 DCU1 和 DCU5；DCU2 将和 DCU6 共用寄存器 DCU2_TRGSEL，即当 DCU2_TRGSEL 写入事件编号，该编号事件发生时，会同时触发 DCU2 和 DCU6；DCU3 将和 DCU7 共用寄存器 DCU3_TRGSEL，即当 DCU3_TRGSEL 写入事件编号，该编号事件发生时，会同时触发 DCU3 和 DCU7；DCU4 将和 DCU8 共用寄存器 DCU4_TRGSEL，即当 DCU4_TRGSEL 写入事件编号，该编号事件发生时，会同时触发 DCU4 和 DCU8。

复位值：0x000001FF

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

位	标记	位名	功能	读写
b31	COMEN[1]	公共触发使能	0：禁止AOS_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 详见自动运行系统(AOS)章节	R/W
b30	COMEN[0]	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 详见自动运行系统(AOS)章节	R/W
b29~b9	Reserved	-	读时读出0，写时请写0	R/W
b8~b0	TRGSEL[8:0]	触发源选择	写入要选择的外设电路事件的编号	R/W

45 数学运算单元（MAU）

45.1 简介

数学运算单元（MAU）是一个内含开方运算和正弦运算两种运算类型的硬件加速运算模块，支持定点数的开方和正弦运算。正弦函数支持 360°/2^12 运算精度。

45.2 功能说明

45.2.1 开方运算

开方运算支持 32 位定点数输入，运算结果为 17 位定点数输出。根据输入数据的大小不同，运算执行周期会不同。

45.2.1.1 操作流程

中断无效（MAU_CSR.INTEN=0）时，开方运算的操作流程如下；中断有效时，用户根据该流程自定义加入中断处理过程：

1) 在数据输入寄存器 0（MAU_DTR0）中写入被开方数据
2) 将控制状态寄存器的启动位置 1（MAU_CSR.START=1）
3) 运算开始后，控制状态寄存器的状态位（MAU_CSR.BUSY）会置位，直到运算完成，该位自动清零。期待 BUSY 位清零，表明运算完成
4) 从结果输出寄存器 0（MAU_RTR0）中读取运算结果
5) 重复以上流程 1）~4）

关于流程 3），从 MAU_CSR.START 置位后，BUSY 位开始变高，在持续 8~16 个运算时钟周期后（根据输入数据的大小运算时间不等）自动变为 0。用户在使用开方运算模块时可根据此位的变化灵活判断当前的运算状态。

45.2.1.2 结果处理

开放运算的运算结果是 17 位数据输出，其最低位结果采用四舍五入方式得到。例如，当输入数据为 0x0000000CH 时，其结果为 0x00003H；当输入数据为 0x0000000DH 时，其结果为 0x00004H。

45.2.1.3 定点数处理

当输入数据为定点小数，且需要保持输入输出数据的格式统一时，用户可通过控制状态寄存器的移位控制位（MAU_CSR.SHIFT）来对输出数据进行相应的处理。

例如，当采用 Q14.2 格式进行定点数运算时，可设定 MAU_CSR.SHIFT=1，使输出结果也遵循 Q14.2 格式。定点数 6.25（Q14.2 格式为 0x00000019H）在移位之前的输出结果位 5（0x00000005H），移位 1 位之后的结果为 2.5（Q14.2 格式为 0x0000000AH）。

45.2.2 正弦运算

正弦运算支持 12 位数据输入，运算结果为 16 位有符号数输出。其中 12 位数据输入表示要运算的直角坐标系角度；16 位数据输出中，最高位表示结果的正负。

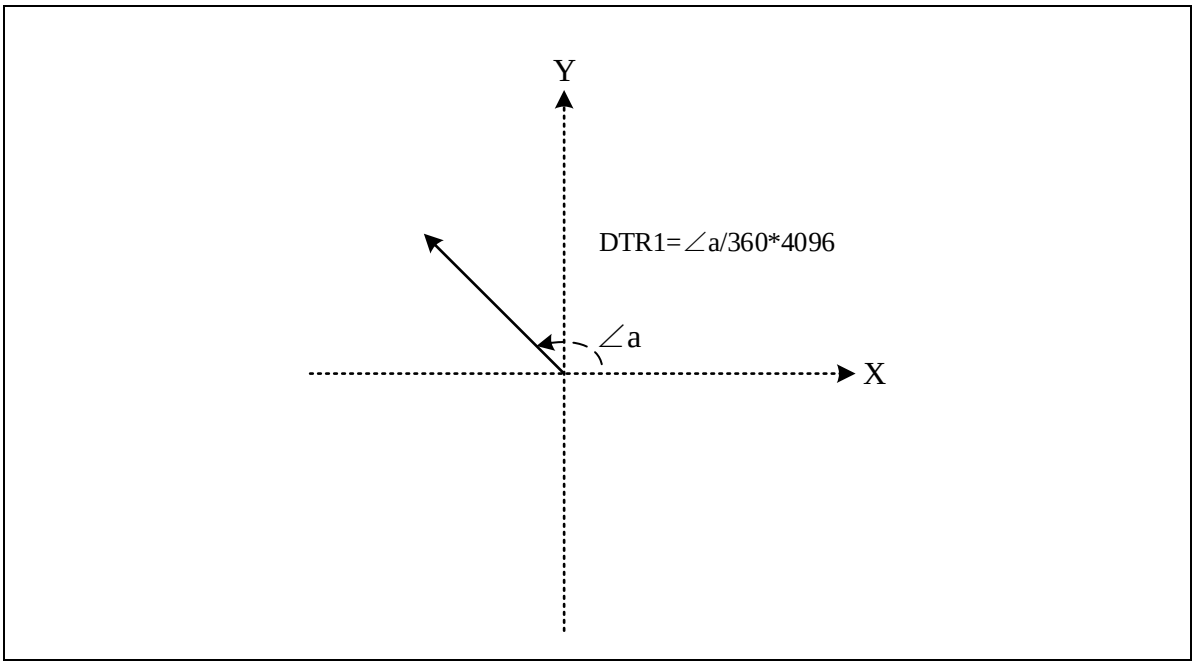


图 45-1　正弦运算角度值示意图

45.2.2.1 操作流程

正弦运算的操作流程如下：

1) 在数据输入寄存器 1（MAU_DTR1）中写入角度值
2) 在 1 个周期之后从结果输出寄存器 1（MAU_RTR1）中读取运算结果
3) 重复以上流程 1）~2）

45.3 中断及事件说明

45.3.1 中断输出

在每一次开方运算完成之后，都会有相应的运算完成事件产生。若使能中断控制位（MAU_CSR.INTEN=1），则对应的中断请求信号（MAU_SQRT）也会产生。

45.3.2 事件输出

在每一次开方运算完成之后，都会有相应的运算完成事件产生，相应的事件请求信号（MAU_SQRT）也会输出，可以用于选择触发其它模块。

45.4 寄存器说明

表 45-1 所示，为 MAU 模块的寄存器列表。

BASE ADDR: 0x40055000H

寄存器名	符号	偏移量	位宽	复位值
控制状态寄存器	MAU_CSR	0x0000h	32	0x00000000h
数据输入寄存器0	MAU_DTR0	0x0004h	32	0x00000000h
结果输出寄存器0	MAU_RTR0	0x000Ch	32	0x00000000h
数据输入寄存器1	MAU_DTR1	0x0010h	32	0x00000000h
结果输出寄存器1	MAU_RTR1	0x0014h	32	0x00000000h

表 45-1 MAU 寄存器列表

45.4.1 控制状态寄存器（MAU_CSR）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	SHIFT[4:0]					-	-	-	-	BUSY	-	INTEN	START

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~b8	SHIFT[4:0]	开方结果移位	00000：运算结果不移位输出 00001：运算结果左移1位后输出 00010：运算结果左移2位后输出 … 10000：运算结果左移16位后输出 请不要设定其它值	R/W
b7~b4	Reserved	-	读出时为“0”,写入时写“0”	R/W
b3	BUSY	开方运算状态	0：开方运算未开始或运算完成 1：正在进行开方运算	R/W
b2	Reserved	-	读出时为“0”,写入时写“0”	R/W
b1	INTEN	开方运算中断使能	0：开方运算完成后不产生中断 1：开方运算完成后产生中断	R/W
b0	START	开方运算开始	0：该位写零无效 1：开方运算开始 注：该位在写入1时内部产生一次START请求，在读出时始终为0	R/W

45.4.2 数据输入寄存器 0（MAU_DTR0）

复位值: 0x00000000h

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

位	标记	位名	功能	读写
b31~b0	SQRT_DIN[31:0]	被开方数	被开方数输入	R/W

45.4.3 结果输出寄存器 0（MAU_RTR0）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															SQRT_DOUT[16]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SQRT_DOUT[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/W
b16~b0	SQRT_DOUT[16:0]	开方结果	开方结果输出	R/W

45.4.4 数据输入寄存器 1（MAU_DTR1）

复位值: 0x00000000h

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

位	标记	位名	功能	读写
b31~b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11~b0	SIN_DIN[11:0]	角度设定	直角坐标系角度输入	R/W

45.4.5 结果输出寄存器 1（MAU_RTR1）

复位值: 0x00000000h

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b0	SIN_DOUT[15:0]	正弦运算结果	正弦运算结果输出	R/W

46 滤波数学加速器（FMAC）

46.1 简介

滤波数学加速器（FMAC）是一个 FIR 滤波计算的硬件加速模块。该模块可进行最大 16 阶，且阶数可配置的 FIR 数字滤波。内置 16x16 bit 乘法器、32+5bit 加法器，用户可自定义输出数据精度。本系列产品搭载 4 个 FMAC 模块。

46.2 基本框图

该模块的基本特性如下：

- 最大 16 阶 FIR 滤波，阶数可配置
- 滤波系数可配置
- 16 位有符号数输入、输出数据，16 位有符号数滤波系数
- 16x16 bit 乘法器
- 37bit 加法器
- 提供中断信号和状态查询
- AHB slave 接口配置参数和输入输出数据

其基本框图如下：

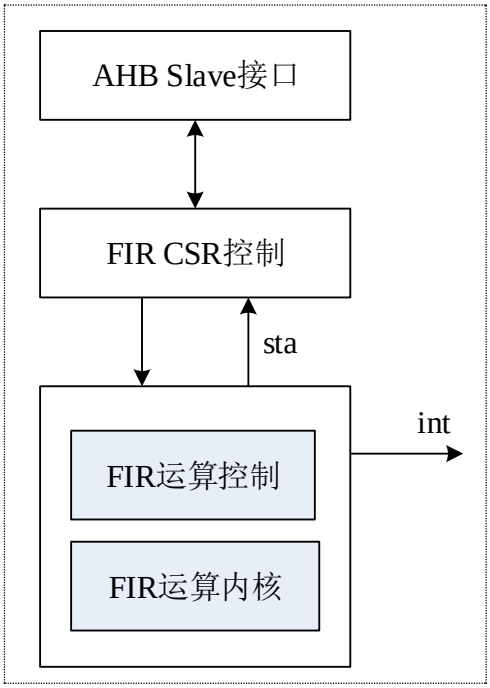


图 46-1 FIR 基本框图

在选择 16 阶 FIR 滤波时，FIR 运算控制和运算内核需要 17 个时钟周期处理一个数据，在 150MHz 的工作时钟频率下，可以处理最大 8.8MHz 的输入数据流。不同的阶数，处理所需的时间不同。阶数越小，处理速度越快。

46.3 操作流程

本模块需要按照以下步骤来使用：

1) 将使能位（FMAC_ENR.FIREN）置为 1；
2) 配置滤波器阶数（FMAC_CTR.STAGE_NUM[4:0]）及输出有效结果移位数（FMAC_CTR.SHIFT[4:0]）；
3) 如果需要使用中断，则配置中断使能位（FMAC_IER.INTEN）为 1；
4) 写入滤波器系数（有 N 阶，就只需配置 FMAC_COR0~FMAC_CORN）；
5) 写入待处理数据（FMAC_DTR）；
6) 等待完成中断或者查询完成状态（FMAC_STR.READY）为 1；
7) 读取结果（FAMC_RTR0 或 FAMC_RTR1）；
8) 重复 5）~7），直到所有数据处理完。

46.4 模块使能

模块使能位（FMAC_ENR.FIREN）的作用有两个，作用一是使能整个模块，只有其为 1 时，模块才允许工作；作用二是模块的软复位作用，当某种阶数和系数的配置使用完以后，需要换另一种阶数和系数的配置，那么需要将 FIREN 置为 0，将前面计算残留的信息清零。再按照操作流程来操作。

FIREN 清 0 时，不会将配置参数清零，只会将内部寄存器和中间结果清零。

46.5 系数归一化

如果输出结果只取 16 位，那么在计算过程中以及计算结果，总共截去的位数多达 20 位，所以如果输入的参与计算的数据太小，则输出结果有可能为 0，或者很小的数（损失了很大的精度）。所以滤波器系数有必要做归一化，即将其中的最大值（如果是正数）归一化成 32767（如果是负数，归一化成-32768），其他系数乘以相同的比例。

例如，有 5 个系数为 10，13，35，96，42，要将其归一化。首先找出其中的最大值，

为 96，然后用 32767 除以该最大值，32767/96=341.3，取值 341，即为归一化的倍数，其他系数都要乘以该倍数，即为归一化以后的数。

所以归一化以后的系数为：3410，4433，11935，32736，14322。

46.6 中断及事件说明

46.6.1 中断输出

在每一次运算完成之后，都会有相应的运算完成事件产生。若使能中断控制位（FMAC_IER.INTEN=1），则对应的中断请求信号（FMAC_m_FIR，m=1~4）也会产生。

46.6.2 事件输出

在每一次运算完成之后，都会有相应的运算完成事件产生，相应的事件请求信号（FMAC_m_FIR，m=1~4）也会输出，可以用于选择触发其它模块。

46.7 寄存器说明

表 46-1 所示，为 FMAC 模块的寄存器列表。

BASE ADDR: 0x40058000H（U1）、0x40058400H（U2）、

0x40058800H（U3）、0x40058C00H（U4）

寄存器名	符号	偏移量	位宽	复位值
模块使能寄存器	FMAC_ENR	0x0000h	32	0x00000000h
基本控制寄存器	FMAC_CTR	0x0004h	32	0x00000010h
中断控制寄存器	FMAC_IER	0x0008h	32	0x00000000h
数据输入寄存器	FMAC_DTR	0x000Ch	32	0x00000000h
滤波系数寄存器0	FMAC_COR0	0x0020h	32	0x00000000h
滤波系数寄存器1	FMAC_COR1	0x0024h	32	0x00000000h
。。。	。。。	。。。	。。。	。。。
滤波系数寄存器16	FMAC_COR16	0x0060h	32	0x00000000h
结果输出寄存器0	FMAC_RTR0	0x0010h	32	0x00000000h
结果输出寄存器1	FMAC_RTR1	0x0014h	32	0x00000000h
运算状态寄存器1	FMAC_STR	0x0018h	32	0x00000000h

表 46-1 FMAC 寄存器列表

46.7.1 模块使能寄存器（FMAC_ENR）

复位值: 0x00000000h

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

位	标记	位名	功能	读写
b31~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	FMACEN	模块使能	0：不使能 1：使能，即允许工作 注：该信号作为本模块的软复位信号使用，即当需要改变滤波阶数和滤波系数的时候，需要先将该位信号置为0，再将该位置为1	R/W

46.7.2 基本控制寄存器（FMAC_CTR）

复位值: 0x00000000h

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

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

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~b8	SHIFT[4:0]	滤波结果移位	滤波结果右移位数，控制MAU_RTR1输出的结果 00000：不移位，即选取运算结果的[31:0]给MAU_RTR1 00001：右移1位，即选取运算结果的[32:1]给MAU_RTR1 00010：右移2位，即选取运算结果的[33:2]给MAU_RTR1 …… 10100：右移20位，即选取运算结果的[35:20]给MAU_RTR1，高位移入符号位 10101：右移21位，即选取运算结果的[36:21]给MAU_RTR1，高位移入符号位 其他值：不移位	R/W
b7~b5	Reserved	-	读出时为“0”,写入时写“0”	R/W
b4~b0	STAGE_NUM[4:0]	滤波阶数设定	设定滤波器的滤波阶数	R/W

46.7.3 中断控制寄存器（FMAC_IER）

复位值: 0x00000000h

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															INTEN

位	标记	位名	功能	读写
b31~b1	Reserved	-	读出时为“0”,写入时写“0”	R/W
b0	INTEN	中断使能	0：运算完成后不产生中断 1：开方运算完成后产生中断	R/W

46.7.4 数据输入寄存器（FMAC_DTR）

复位值: 0x00000000h

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b0	FMAC_DIN[15:0]	数据输入	FIR数据输入	R/W

46.7.5 滤波系数寄存器（FMAC_COR0~16）

复位值: 0x00000000h

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15~b0	FMAC_CIN[15:0]	滤波系数	滤波系数输入	R/W

46.7.6 结果输出寄存器 0（FMAC_RTR0）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
FMAC_DOUT0[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FMAC_DOUT0[15:0]															

位	标记	位名	功能	读写
b31~b0	FMAC_DOUT0[31:0]	结果输出	累加器结果的高32位结果输出	R/W

46.7.7 结果输出寄存器 1（FMAC_RTR1）

复位值: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
FMAC_DOUT1[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FMAC_DOUT1[15:0]															

位	标记	位名	功能	读写
b31~b0	FMAC_DOUT1[31:0]	结果输出	累加器结果的低32位结果输出	R/W

46.7.8 运算状态寄存器（FMAC_STR）

复位值: 0x00000000h

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

位	标记	位名	功能	读写
b31	READY	运算完成位	0：计算未完成，结果不可用 1：计算完成，结果可用 注：该寄存器读之后自动清零	R/W
b30~b0	Reserved	-	读出时为“0”,写入时写“0”	R/W

47 调试控制器（DBGC）

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

47.1 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，支持嵌入式跟踪宏单元（ETM）。利用这些调试功能，可以在取指（指令断点）或访问数据（数据断点）时停止内核。内核停止时，可以查询内核的内部状态和系统的外部状态。查询完成后，将恢复内核和系统并恢复程序执行。

提供两个调试接口：

- 串行调试跟踪接口 SWD
- 并行调试跟踪接口 JTAG

47.2 DBGC 系统框图

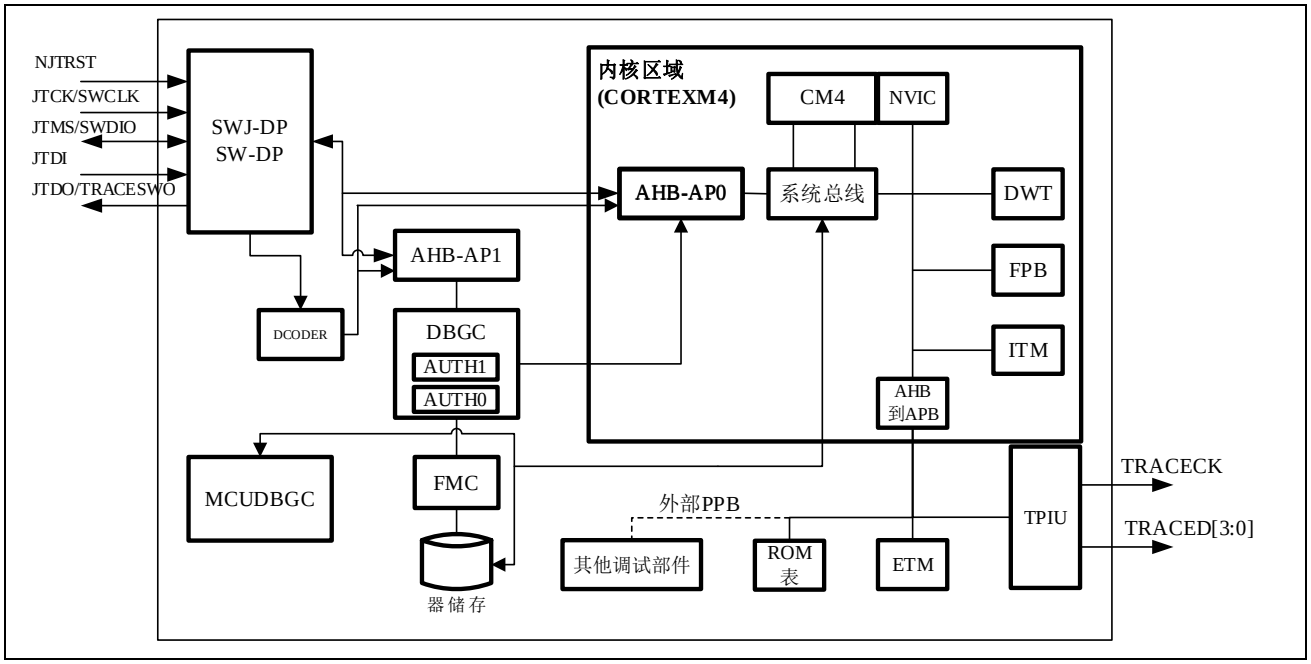


图 47-1 调试控制系统

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
- AHP-AP：AHB 访问端口
- ITM：指令跟踪单元
- ETM：嵌入式跟踪宏单元
- FPB：Flash 指令断点
- DWT：数据断点触发
- TPIU：跟踪端口单元接口（大封装上提供，其中会映射相应引脚）
- 灵活调试引脚分配

注意：

- 有关 ARM Cortex™-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》。

47.3 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。

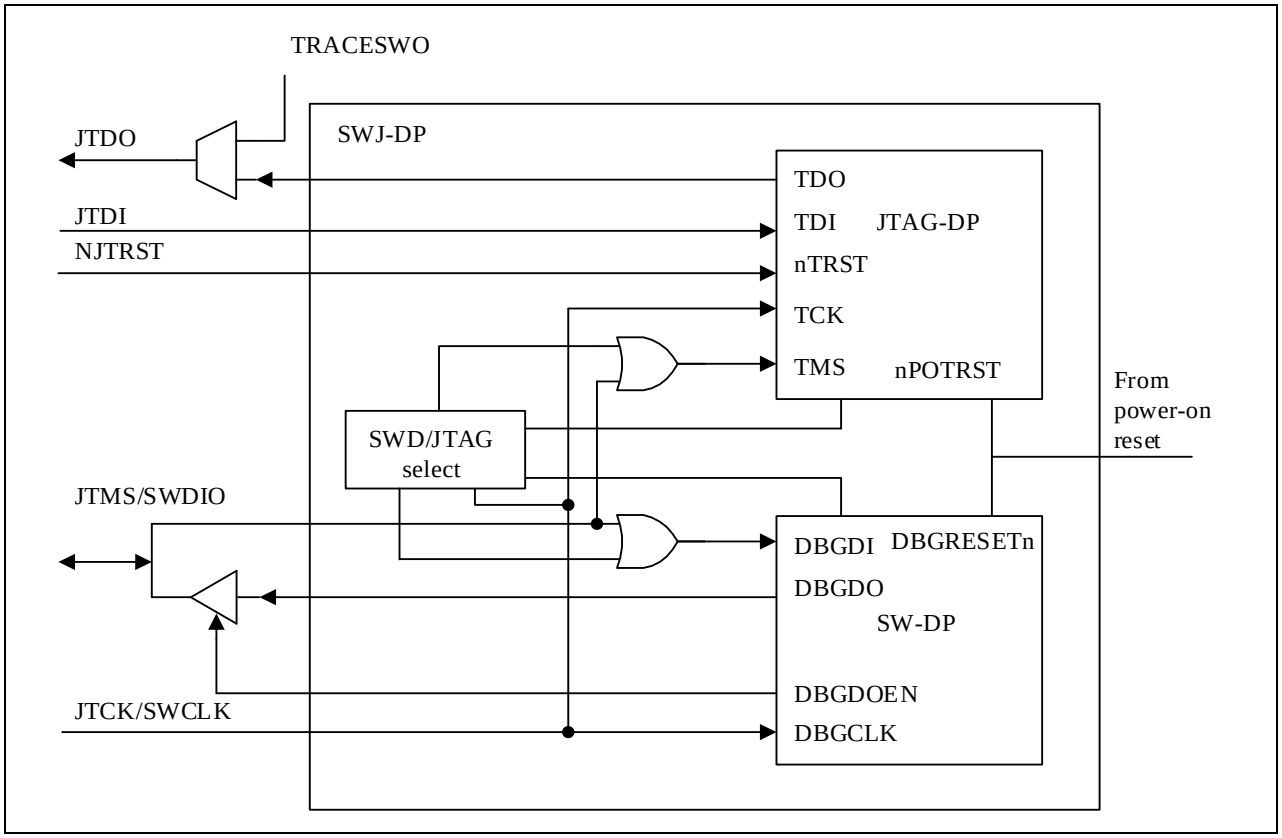


图 47-2 调试控制系统

47.3.1 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

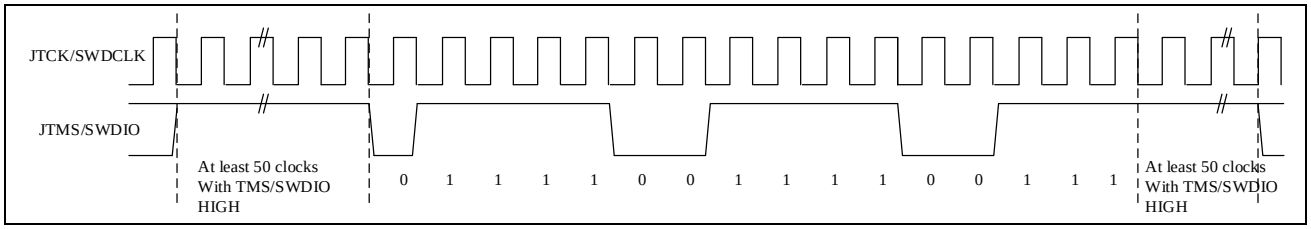


图 47-3 JTAG-DP 到 SW-DP 切换时序

47.4 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。

47.4.1 SWJ 调试端口引脚

MCU 的 5 个普通 I/O 口可用作 SWJ-DP 接口引脚。

表 47-1 SWJ 调试端口引脚

SWJ-DP 引脚名称	JTAG调试端口		SW调试端口	
	类型	说明	类型	调试分配
JTMS/SWDIO	I	JTAG模式选择	I/O	串行线数据输入/输出
JTCK/SWCLK	I	JTAG时钟	I	串行时钟
JTDI	I	JTAG数据输入	-	-
JTDO/TRACESWO	O	JTAG数据输出	-	TRACESWO(如果使能异步跟踪)
NJTRST	I	JTAG复位	-	-

47.4.2 灵活的 SWJ-DP 引脚分配

复位（上电或引脚复位）后，会将用于 SWJ-DP 的全部 5 个引脚指定为专用引脚，可供调试工具立即使用（请注意，除非明确编程，否则不分配跟踪输出）。但是，MCU 可以禁止部分或全部 SWJ-DP 端口，进而释放相关引脚以用作 GPIO。有关如何禁止 SWJ-DP 端口引脚的更多详细信息，请参见：通用 IO 特殊控制寄存器 PSPCR。

表 47-2 灵活的 SWJ-DP 引脚分配

可用的调试端口	分配的SWJ IO引脚				
	JTMS/ SWDIO	JTCK/ SWCLK	JTDI	JTDO	NJTRST
全部SWJ（JTAG-DP+SW-DP）-复位状态	√	√	√	√	√
禁止JTAG-DP和使能SW-DP	√	√	可释放	可释放	可释放
禁止JTGA-DP和禁止SW-DP	可释放	可释放	可释放	可释放	可释放

47.4.3 JTAG 引脚上的内部上拉

根据 JTAG IEEE 标准，必须确保 JTAG 输入引脚不悬空，因为这些引脚直接连接到 MCU 内部用于控制调试功能。还必须特别注意 JTCK/SWCLK 引脚，该引脚直接用于调试控制时钟功能。为避免 IO 电平浮空，MCU 在 JTAG 引脚上内置有内部上拉电阻：

- NJTRST：内部上拉
- JTDI：内部上拉
- JTMS/SWDIO：内部上拉
- JTCK/SWCLK：内部上拉
- JTDO：内部上拉

在未连接调试器状态下，用户软件可以通过设定 GPIO 特殊控制寄存器释放 JTAG IO 作为普通的 I/O 口使用。由于芯片内部上拉为<100K 欧的弱上拉，建议采用外部 10K 欧的外部上拉。

47.4.4 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)

47.5 寄存器说明

寄存器描述如下：

基地址：0xE0042000

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	0x0000003B	CPU/调试IDE*
TRACE端口控制寄存器	MCUTRACECTL	0x0024	32	0x00000000	CPU/调试IDE*
外设调试暂停寄存器	MCUSTPCTL2	0x0028	32	0x00000000	CPU/调试IDE*

注意：

- 寄存器位于 PPB 区域，只能在特权模式下 CPU 才能进行访问。

47.5.1 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	CDBGPWRUPACK	调试器上电反馈	0：无反馈 1：调试上电反馈	R/W
b0	CDBGPWRUPREQ	调试器上电请求	0：无上电请求 1：上电请求	R/W

47.5.2 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 003B

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	M22 STP	M21 STP	M20 STP	M19 STP	M18 STP	M17 STP	M16 STP
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
M15 STP	M14 STP	M13 STP	M12 STP	M11 STP	M10 STP	M09 STP	M08 STP	M07 STP	M06 STP	PVD2 STP	PVD1 STP	PVD0 STP	RTC STP	WDT STP	SWD TSTP

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“0”,写入时写“0”	R/W
b30	Reserved	-	读出时为“0”,写入时写“0”	R/W
b29	Reserved	-	读出时为“0”,写入时写“0”	R/W
b28	Reserved	-	读出时为“0”,写入时写“0”	R/W
b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26	Reserved	-	读出时为“0”,写入时写“0”	R/W
b25	Reserved	-	读出时为“0”,写入时写“0”	R/W
b24	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	M22STP	TMR6_8计数暂停信号	0：即使内核停止，计数器仍然计数 1：内核停止时，计数器暂停计数	R/W
b21	M21STP	TMR6_7计数暂停信号	0：即使内核停止，计数器仍然计数 1：内核停止时，计数器暂停计数	R/W
b20	M20STP	TMR6_6计数暂停信号	0：即使内核停止，计数器仍然计数 1：内核停止时，计数器暂停计数	R/W
b19	M19STP	TMR6_5计数暂停信号	0：即使内核停止，计数器仍然计数 1：内核停止时，计数器暂停计数	R/W
b18	M18STP	TMR6_4计数暂停信号	0：即使内核停止，计数器仍然计数 1：内核停止时，计数器暂停计数	R/W
b17	M17STP	TMR6_3计数暂停信号	0：即使内核停止，计数器仍然计数 1：内核停止时，计数器暂停计数	R/W
b16	M16STP	TMR6_2计数暂停信号	0：即使内核停止，计数器仍然计数 1：内核停止时，计数器暂停计数	R/W
b15	M15STP	TMR6_1计数暂停信号	0：即使内核停止，计数器仍然计数 1：内核停止时，计数器暂停计数	R/W
b14	M14STP	TMR4_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
b13	M13STP	TMR4_2计数暂停信号	0：即使内核停止，计数器仍然计数	R/W

			1：内核停止时，计数器暂停计数	
b12	M12STP	TMR4_1计数暂停信号	0：即使内核停止，计数器仍然计数	R/W
			1：内核停止时，计数器暂停计数	
b11	M11STP	TMR2_4计数暂停信号	0：即使内核停止，计数器仍然计数	R/W
			1：内核停止时，计数器暂停计数	
b10	M10STP	TMR2_3计数暂停信号	0：即使内核停止，计数器仍然计数	R/W
			1：内核停止时，计数器暂停计数	
b9	M09STP	TMR2_2计数暂停信号	0：即使内核停止，计数器仍然计数	R/W
			1：内核停止时，计数器暂停计数	
b8	M08STP	TMR2_1计数暂停信号	0：即使内核停止，计数器仍然计数	R/W
			1：内核停止时，计数器暂停计数	
b7	M07STP	TMR0_2计数暂停信号	0：即使内核停止，计数器仍然计数	R/W
			1：内核停止时，计数器暂停计数	
b6	M06STP	TMR0_1计数暂停信号	0：即使内核停止，计数器仍然计数	R/W
			1：内核停止时，计数器暂停计数	
b5	PVD2STP	PVD2暂停信号	0：即使内核停止，仍然产生PVD2中断申请或复位	R/W
			1：内核停止时，屏蔽PVD2中断申请或复位	
b4	PVD1STP	PVD1暂停信号	0：即使内核停止，仍然产生PVD1中断申请或复位	R/W
			1：内核停止时，屏蔽PVD1中断申请或复位	
b3	PVD0STP	PVD0暂停信号	0：即使内核停止，仍然产生PVD0中断申请或复位	R/W
			1：内核停止时，屏蔽PVD0中断申请或复位	
b2	RTCSTP	RTC计数暂停信号	0：即使内核停止，RTC计数器仍然计数	R/W
			1：内核停止时，RTC计数器暂停计数	
b1	WDTSTP	WDT计数暂停信号	0：即使内核停止，WDT计数器仍然计数	R/W
			1：内核停止时，WDT计数器暂停计数	
b0	SWDTSTP	SWDT计数暂停信号	0：即使内核停止，SWDT计数器仍然计数	R/W
			1：内核停止时，SWDT计数器暂停计数	

47.5.3 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	TRACE_MODE	

位	标记	位名	功能	读写
b31~b3	Reserved	-	读出时为“0”,写入时写“0”	R/W
b2	TRACE_IOEN	TRACE引脚输出控制	0：同步跟踪引脚输出禁止	R/W
			1：同步跟踪引脚输出许可	
b1~b0	TRACE_MODE	TRACED输出引脚控制	00：异步跟踪	R/W
			01：同步跟踪1位TRACED[0]	
			10：同步跟踪2位TRACED[1:0]	
			11：同步跟踪4位TRACED[3:0]	

47.5.4 外设调试暂停寄存器 2（MCUSTPCTL2）

当 CPU 处于调试状态时，周边模块暂停控制。

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	M43 STP	M42 STP	M41 STP	M40 STP	M39 STP	M38 STP	M37 STP	M36 STP	M35 STP	M34 STP	M33 STP	M32 STP

位	标记	位名	功能	读写
b31	Reserved	-	读出时为“0”,写入时写“0”	R/W
b30	Reserved	-	读出时为“0”,写入时写“0”	R/W
b29	Reserved	-	读出时为“0”,写入时写“0”	R/W
b28	Reserved	-	读出时为“0”,写入时写“0”	R/W
b27	Reserved	-	读出时为“0”,写入时写“0”	R/W
b26	Reserved	-	读出时为“0”,写入时写“0”	R/W
b25	Reserved	-	读出时为“0”,写入时写“0”	R/W
b24	Reserved	-	读出时为“0”,写入时写“0”	R/W
b23	Reserved	-	读出时为“0”,写入时写“0”	R/W
b22	Reserved	-	读出时为“0”,写入时写“0”	R/W
b21	Reserved	-	读出时为“0”,写入时写“0”	R/W
b20	Reserved	-	读出时为“0”,写入时写“0”	R/W
b19	Reserved	-	读出时为“0”,写入时写“0”	R/W
b18	Reserved	-	读出时为“0”,写入时写“0”	R/W
b17	Reserved	-	读出时为“0”,写入时写“0”	R/W
b16	Reserved	-	读出时为“0”,写入时写“0”	R/W
b15	Reserved	-	读出时为“0”,写入时写“0”	R/W
b14	Reserved	-	读出时为“0”,写入时写“0”	R/W
b13	Reserved	-	读出时为“0”,写入时写“0”	R/W
b12	Reserved	-	读出时为“0”,写入时写“0”	R/W
b11	M43STP	TMRA_12计数暂停信号	0：即使内核停止，计数器仍然计数 1：内核停止时，计数器暂停计数	R/W
b10	M42STP	TMRA_11计数暂停信号	0：即使内核停止，计数器仍然计数 1：内核停止时，计数器暂停计数	R/W
b9	M41STP	TMRA_10计数暂停信号	0：即使内核停止，计数器仍然计数 1：内核停止时，计数器暂停计数	R/W
b8	M40STP	TMRA_9计数暂停信号	0：即使内核停止，计数器仍然计数 1：内核停止时，计数器暂停计数	R/W
b7	M39STP	TMRA_8计数暂停信号	0：即使内核停止，计数器仍然计数	R/W

			1：内核停止时，计数器暂停计数	
b6	M38STP	TMRA_7计数暂停信号	0：即使内核停止，计数器仍然计数	R/W
			1：内核停止时，计数器暂停计数	
b5	M37STP	TMRA_6计数暂停信号	0：即使内核停止，计数器仍然计数	R/W
			1：内核停止时，计数器暂停计数	
b4	M36STP	TMRA_5计数暂停信号	0：即使内核停止，计数器仍然计数	R/W
			1：内核停止时，计数器暂停计数	
b3	M35STP	TMRA_4计数暂停信号	0：即使内核停止，计数器仍然计数	R/W
			1：内核停止时，计数器暂停计数	
b2	M34STP	TMRA_3计数暂停信号	0：即使内核停止，计数器仍然计数	R/W
			1：内核停止时，计数器暂停计数	
b1	M33STP	TMRA_2计数暂停信号	0：即使内核停止，计数器仍然计数	R/W
			1：内核停止时，计数器暂停计数	
b0	M32STP	TMRA_1计数暂停信号	0：即使内核停止，计数器仍然计数	R/W
			1：内核停止时，计数器暂停计数	

47.6 SW 调试端口

47.6.1 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）。

47.7 TPIU（跟踪端口接口单元）

47.7.1 简介

TPIU 是 ITM 和 ETM 与片上跟踪数据之间的桥接器。

输出数据流封装成跟踪源 ID，然后被跟踪端口分析器 (TPA) 捕获。

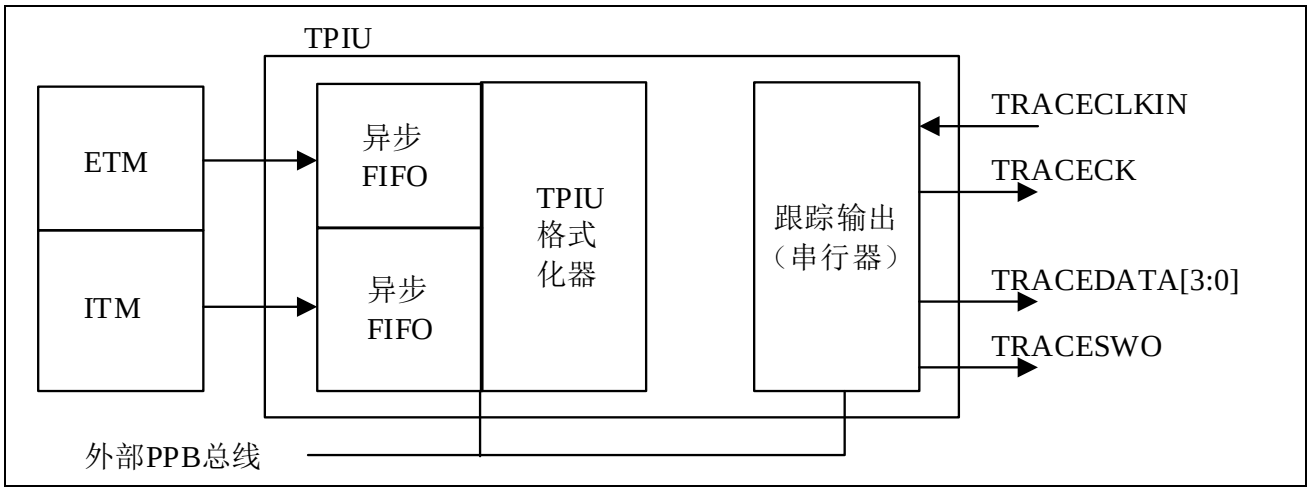


图 47-4 TPIU 框图

47.7.2 TRACE 引脚分配

- 异步模式

  异步模式需要 1 个额外引脚，并且适用于所有封装。仅在使用串线行模式时异步模式才可用（在 JTAG 模式下不可用）。

TPIU引脚名称	跟踪异步模式	
	类型	说明
TRACESWO	输出	TRACE异步数据输出

- 同步模式

  同步模式需要 2 到 5 个额外引脚，具体取决于所跟踪数据的长度，并且仅适用于较大型的封装。此外，同步模式在 JTAG 模式和串行模式下均可用，并可提供比异步跟踪更高的带宽输出能力。

TPIU引脚名称	跟踪同步模式	
	类型	说明
TRACECK	输出	TRACE时钟
TRACED[3:0]	输出	TRACE同步数据输出，可以是1、2或4.

TPIU TRACE 引脚分配

默认情况下，不分配这些引脚。可通过将 MCU 调试组件配置寄存器（MCUTRACECTL）中的 TRACE_IOEN 和 TRACE_MODE 位来配置这些引脚。必须由调试主机或 CPU 来完成此配置。

此外，要分配的引脚数目取决于跟踪配置（异步跟踪或同步跟踪）。

- 异步模式：需要 1 个额外引脚
- 同步模式：需要 5 个额外引脚
  - TRACECK
  - TRACED[0]（如果端口数据长度配置为 1，2 或 4）
  - TRACED[1]（如果端口数据长度配置为 2 或 4）
  - TRACED[2]（如果端口数据长度配置 4）
  - TRACED[3]（如果端口数据长度配置 4）

要分配 TRACE 引脚，调试主机必须对 MCU 调试配置寄存器（MCUTRACECTL）的位 TRACE_IOEN 和 TRACE_MODE[1:0]进行编程。默认情况下不分配 TRACE 引脚。此寄存器映射到外部 PPB 总线上，通过上电（而非引脚复位）复位。可在引脚复位状态下通过调试器写入该寄存器。

TPIU引脚用途	分配的TRACE IO引脚					
	JTDO/ TRACESWO	TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]
无跟踪（默认状态） TRACE_IOEN =0 TRACE_MODE=XX	释放*	释放	释放	释放	释放	释放
异步跟踪 TRACE_IOEN =1 TRACE_MODE=00	TRACESWO	释放	释放	释放	释放	释放
同步跟踪1位 TRACE_IOEN =1 TRACE_MODE=01	释放*	TRACECK	TRACED[0]	释放	释放	释放
同步跟踪2位 TRACE_IOEN =1 TRACE_MODE=10	释放*	TRACECK	TRACED[0]	TRACED[1]	释放	释放
同步跟踪4位 TRACE_IOEN =1 TRACE_MODE=11	释放*	TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]

注意：

- 使用串行模式时，释放此引脚。但使用 JTAG 时，此引脚分配给 TDO。

47.7.3 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。

47.7.4 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）。

47.7.5 TPIU 配置示例

- 将调试异常和监视控制寄存器 (DEMCR)中的位 TRCENA 置 1
- 将所需值写入 TPIU 当前端口大小寄存器（对于 1 位端口大小，默认值为 0x1）
- 将 0x102 写入 TPIU 格式化器和刷新控制寄存器（默认值）
- 写入 TPIU 选择引脚协议以选择同步模式或异步模式。示例：0x2 表示异步 NRZ 模式（类似于 USART）
- 将 0x00 写入 MCUTRACECTL 控制寄存器（位 TRACE_IOEN），为异步模式分配 TRACE I/O
- 此时发送 TPIU 同步数据包 (FF_FF_FF_7F)
- 配置 ITM 并对 ITM 激励寄存器进行写操作以输出值

版本信息 & 联系方式

版本	日期	修订内容摘要
Rev1.0	2020/11/18	初版发布。
Rev1.01	2020/12/31	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。



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

