XHSC 小华半导体
XIAOHUA SEMICONDUCTOR

HC32F072 系列

32 位 ARM® Cortex®-M0+ 微控制器

参考手册

Rev1.80　2026 年 08 月

声　明

★ 小华半导体有限公司（以下简称：“XHSC”）保留随时更改、更正、增强、修改小华半导体产品和/或本文档的权利，恕不另行通知。用户可在下单前获取最新相关信息。XHSC 产品依据购销基本合同中载明的销售条款和条件进行销售。

★ 客户应针对您的应用选择合适的 XHSC 产品，并设计、验证和测试您的应用，以确保您的应用满足相应标准以及任何安全、安保或其它要求。客户应对此独自承担全部责任。

★ XHSC 在此确认未以明示或暗示方式授予任何知识产权许可。

★ XHSC 产品的转售，若其条款与此处规定不同，XHSC 对此类产品的任何保修承诺无效。

★ 任何带有“®”或“™”标识的图形或字样是 XHSC 的商标。所有其他在 XHSC 产品上显示的产品或服务名称均为其各自所有者的财产。

★ 重要提示：在使用本产品前，请务必查阅官方发布的最新勘误表，充分知悉产品已知修正信息，规避使用风险。

★ 本通知中的信息取代并替换先前版本中的信息。

©2026 小华半导体有限公司 保留所有权利

目 录

声 明 **2**
目 录 **3**
表索引 **26**
图索引 **29**
简介（Overview） **36**
1 系统结构 **37**
1.1 概述 37
1.2 系统地址划分 38
1.3 存储器和模块地址分配 39
2 工作模式 **41**
2.1 运行模式 43
2.2 休眠模式 44
2.3 深度休眠模式 45
3 系统控制器（SYSCTRL） **47**
3.1 系统时钟介绍 47
3.1.1 内部高速 RC 时钟 RCH 48
3.1.2 内部低速 RC 时钟 RCL 48
3.1.3 外部低速晶振时钟 XTL 49
3.1.4 外部高速晶振时钟 XTH 49
3.1.5 锁相环时钟 PLL 49
3.1.6 内部高速时钟 RCH48M 49
3.1.7 时钟启动过程 50
3.2 系统时钟切换 50
3.2.1 标准的时钟切换流程 50
3.2.2 RCH 不同振荡频率间切换流程 51
3.2.3 从其它时钟切换到 XTL 示例 52
3.2.4 从其它时钟切换到 XTH 示例 52
3.2.5 从其它时钟切换到 RCL 示例 53
3.2.6 从其它时钟切换到 RCH 示例 53
3.2.7 PLL 与 RCH 相互切换示例，参考时钟为 RCH 53
3.2.8 PLL 与 XTH 相互切换示例，参考时钟为 XTH 54
3.3 时钟校准模块 56
3.4 中断唤醒控制 56

3.4.1 从深度休眠模式唤醒后执行中断服务程序的方法 57
3.4.2 从深度休眠模式唤醒后不执行中断服务程序的方法 57
3.4.3 使用退出休眠特性 58
3.5 寄存器 59
3.5.1 系统控制寄存器 0（SYSCTRL0） 60
3.5.2 系统控制寄存器 1（SYSCTRL1） 62
3.5.3 系统控制寄存器 2（SYSCTRL2） 64
3.5.4 RCH 控制寄存器（RCH_CR） 65
3.5.5 XTH 控制寄存器（XTH_CR） 66
3.5.6 RCL 控制寄存器（RCL_CR） 67
3.5.7 XTL 控制寄存器（XTL_CR） 68
3.5.8 PLL 控制寄存器（PLL_CR） 69
3.5.9 外围模块时钟控制寄存器（PERI_CLKEN0） 71
3.5.10 外围模块时钟控制寄存器（PERI_CLKEN1） 73
3.5.11 定时器时钟倍频控制（OVCK_CR） 74
3.5.12 RC48M 控制寄存器（RC48M_CR） 74
4 复位控制器（RESET） 75
4.1 复位控制器介绍 75
4.1.1 上电下电复位 POR 75
4.1.2 外部复位管脚复位 76
4.1.3 WDT 复位 76
4.1.4 PCA 复位 76
4.1.5 LVD 低电压复位 76
4.1.6 Cortex-M0+ SYSRESETREQ 复位 76
4.1.7 Cortex-M0+ LOCKUP 复位 76
4.2 寄存器 77
4.2.1 复位标识寄存器（RESET_FLAG） 77
4.2.2 外围模块复位控制寄存器（PERI_RESET0） 79
4.2.3 外围模块复位控制寄存器（PERI_RESET1） 81
5 中断控制器（NVIC） 82
5.1 概述 82
5.2 中断优先级 82
5.3 中断向量表 83
5.4 中断输入和挂起行为 84
5.5 中断等待 86

5.6 中断源 87
5.7 中断结构图 88
5.8 寄存器 90
5.8.1 中断使能设置寄存器（NVIC_ISER） 91
5.8.2 中断使能清除寄存器（NVIC_ICER） 91
5.8.3 中断挂起状态设置寄存器（NVIC_ISPR） 92
5.8.4 中断挂起状态清除寄存器（NVIC_ICPR） 92
5.8.5 中断优先级寄存器（NVIC_IPR0） 93
5.8.6 中断优先级寄存器（NVIC_IPR1） 93
5.8.7 中断优先级寄存器（NVIC_IPR2） 94
5.8.8 中断优先级寄存器（NVIC_IPR3） 94
5.8.9 中断优先级寄存器（NVIC_IPR4） 95
5.8.10 中断优先级寄存器（NVIC_IPR5） 95
5.8.11 中断优先级寄存器（NVIC_IPR6） 96
5.8.12 中断优先级寄存器（NVIC_IPR7） 96
5.8.13 中断屏蔽特殊寄存器（PRIMASK） 97
5.9 软件基本操作 98
5.9.1 外部中断使能 98
5.9.2 NVIC 中断使能和清除使能 98
5.9.3 NVIC 中断挂起和清除挂起 98
5.9.4 NVIC 中断优先级 98
5.9.5 NVIC 中断屏蔽 98
6 端口控制器（GPIO） 100
6.1 端口控制器简介 100
6.2 端口控制器主要特性 100
6.3 端口控制器功能描述 101
6.3.1 端口配置功能 101
6.3.2 端口的写入 103
6.3.3 端口的读取 104
6.3.4 端口数字复用功能 105
6.3.5 端口中断功能 107
6.4 端口配置操作流程 108
6.4.1 端口复用配置为模拟端口操作流程 108
6.4.2 端口复用配置为数字通用端口操作流程 108
6.4.3 端口复用配置为数字功能端口操作流程 108

6.4.4 端口复用配置为调试测试端口操作流程 108
6.4.5 端口复用配置为红外输出信号操作流程 108
6.4.6 端口高电平中断操作流程 108
6.4.7 端口低电平中断操作流程 109
6.4.8 端口上升沿中断操作流程 109
6.4.9 端口下降沿中断操作流程 109
6.4.10 端口上拉使能配置操作流程 109
6.4.11 端口下拉使能配置操作流程 109
6.4.12 端口增强驱动配置操作流程 110
6.4.13 端口开漏输出配置操作流程 110
6.4.14 端口位置位操作流程 110
6.4.15 端口位清零操作流程 110
6.4.16 端口位置位清零操作流程 110
6.5 端口控制器寄存器描述 111
6.5.1 端口通用寄存器 117
6.5.2 端口辅助寄存器 126
6.5.3 端口 PA 功能选择寄存器 133
6.5.4 端口 PB 功能选择寄存器 149
6.5.5 端口 PC 功能选择寄存器 165
6.5.6 端口 PD 功能选择寄存器 176
6.5.7 端口 PE 功能选择寄存器 185
6.5.8 端口 PF 功能选择寄存器 194
7 I2C 总线（I2C） 200
7.1 简介 200
7.2 主要特性 200
7.3 协议描述 201
7.3.1 I2C 总线上数据传输 201
7.3.2 I2C 总线上的应答 202
7.3.3 I2C 总线上的仲裁 203
7.4 功能描述 204
7.4.1 串行时钟发生器 205
7.4.2 输入滤波器 205
7.4.3 地址比较器 205
7.4.4 应答标志位 205
7.4.5 中断产生器 206

7.4.6 工作模式 206
7.4.7 状态码表述 211
7.5 编程示例 213
7.5.1 主机发送示例 213
7.5.2 主机接收示例 213
7.5.3 从机接收示例 214
7.5.4 从机发送示例 215
7.6 寄存器描述 216
7.6.1 I2C 波特率计数器使能寄存器(I2Cx_TMRUN) 217
7.6.2 I2C 波特率计数器配置寄存器(I2Cx_TM) 217
7.6.3 I2C 配置寄存器(I2Cx_CR) 218
7.6.4 I2C 数据寄存器(I2Cx_DATA) 219
7.6.5 I2C 地址寄存器(I2Cx_ADDR) 219
7.6.6 I2C 状态寄存器(I2Cx_STAT) 220
8 串行外设接口（SPI） 221
8.1 SPI 简介 221
8.2 SPI 主要特性 221
8.3 SPI 功能描述 222
8.3.1 SPI 主机模式 222
8.3.2 SPI 从机模式 223
8.3.3 SPI 数据帧格式 224
8.3.4 SPI 状态标志及中断 226
8.3.5 SPI 多机系统配置说明 226
8.3.6 SPI 管脚配置说明 228
8.4 SPI 编程示例 229
8.4.1 SPI 主机发送示例 229
8.4.2 SPI 主机接收示例 229
8.4.3 SPI 从机发送示例 230
8.4.4 SPI 从机接收示例 231
8.5 SPI 寄存器描述 232
8.5.1 SPI 配置寄存器(SPIx_CR) 233
8.5.2 SPI 片选配置寄存器(SPIx_SSN) 234
8.5.3 SPI 状态寄存器(SPIx_STAT) 235
8.5.4 SPI 数据寄存器(SPIx_DATA) 236
8.5.5 SPI 配置寄存器 2(SPIx_CR2) 237

8.5.6 SPI 中断清除寄存器 2(SPIx_ICLR)....238
9 时钟校准模块（CLKTRIM）....239
9.1 CLKTRIM 简介....239
9.2 CLKTRIM 主要特性....239
9.3 CLKTRIM 功能描述....239
9.3.1 CLKTRIM 校准模式....239
9.3.2 CLKTRIM 监测模式....242
9.4 CLKTRIM 寄存器描述....245
9.4.1 配置寄存器(CLKTRIM_CR)....246
9.4.2 参考计数器初值配置寄存器(CLKTRIM_REFCON)....247
9.4.3 参考计数器值寄存器(CLKTRIM_REFCNT)....247
9.4.4 校准计数器值寄存器(CLKTRIM_CALCNT)....248
9.4.5 中断标志位寄存器(CLKTRIM_IFR)....248
9.4.6 中断标志位清除寄存器(CLKTRIM_ICLR)....249
9.4.7 校准计数器溢出值配置寄存器(CLKTRIM_CALCON)....249
10 硬件除法器模块（HDIV）....250
10.1 HDIV 简介....250
10.2 HDIV 主要特性....250
10.3 HDIV 功能描述....251
10.3.1 HDIV 操作流程....251
10.4 HDIV 寄存器描述....252
10.4.1 被除数寄存器(HDIV_DIVIDEND)....253
10.4.2 除数寄存器(HDIV_DIVISOR)....253
10.4.3 商寄存器(HDIV_QUOTIENT)....254
10.4.4 余数寄存器(HDIV_REMAINDER)....254
10.4.5 符号寄存器(HDIV_SIGN)....255
10.4.6 状态寄存器(HDIV_STAT)....255
11 FLASH 控制器（FLASH）....256
11.1 概述....256
11.2 FLASH 容量划分....256
11.3 功能描述....256
11.3.1 页擦除（Sector Erase）....257
11.3.2 全片擦除（Chip Erase）....257
11.3.3 写操作（Program）....258
11.3.4 读操作（Read）....260

11.4 擦写时间 260
11.5 读等待周期 261
11.6 擦写保护 261
11.6.1 擦写保护位 261
11.6.2 PC 地址擦写保护 261
11.7 寄存器写保护 262
11.8 寄存器 263
11.8.1 TNVS 参数寄存器（FLASH_TNVS） 264
11.8.2 TPGS 参数寄存器（FLASH_TPGS） 264
11.8.3 TPROG 参数寄存器（FLASH_TPROG） 265
11.8.4 TSERASE 寄存器（FLASH_TSERASE） 265
11.8.5 TMERASE 参数寄存器（FLASH_TMERASE） 266
11.8.6 TPRCV 参数寄存器（FLASH_TPRCV） 266
11.8.7 TSRCV 参数寄存器（FLASH_TSRCV） 267
11.8.8 TMRCV 参数寄存器（FLASH_TMRCV） 267
11.8.9 CR 寄存器（FLASH_CR） 268
11.8.10 IFR 寄存器（FLASH_IFR） 269
11.8.11 ICLR 寄存器（FLASH_ICLR） 269
11.8.12 BYPASS 寄存器（FLASH_BYPASS） 270
11.8.13 SLOCK0 寄存器（FLASH_SLOCK0） 270
11.8.14 SLOCK1 寄存器（FLASH_SLOCK1） 271
12 RAM 控制器（RAM） 272
12.1 概述 272
12.2 功能描述 272
12.2.1 RAM 地址范围 272
12.2.2 读写位宽 272
12.2.3 奇偶校验 272
12.3 寄存器 273
12.3.1 控制寄存器（RAM_CR） 274
12.3.2 奇偶校验出错地址寄存器（RAM_ERRADDR） 274
12.3.3 出错中断标志寄存器（RAM_IFR） 275
12.3.4 出错中断标志清除寄存器（RAM_ICLR） 275
13 DMA 控制器（DMAC） 276
13.1 DMAC 简介 276
13.2 DMAC 主要特性 276

13.3 功能框图 277
13.4 功能描述 278
13.4.1 DMAC 传输模式概述 278
13.4.2 DMA 软件 Block 传输模式 278
13.4.3 DMA 软件 Burst 传输模式 279
13.4.4 DMA 硬件 Block 传输模式 280
13.4.5 DMA 硬件 Burst 传输模式 280
13.4.6 DMA 传输暂停 282
13.4.7 DMA 其它配置 282
13.4.8 DMA 中断 283
13.5 寄存器 284
13.5.1 DMAC_CONF 285
13.5.2 DMAC_CONFA0、DMAC_CONFA1 286
13.5.3 DMAC_CONFB0、DMAC_CONFB1 288
13.5.4 DMAC_SRCADR0、DMAC_SRCADR1 290
13.5.5 DMAC_DSTADR0、DMAC_DSTADR1 290
14 通用定时器（TIM0/1/2/3） 291
14.1 通用定时器简介 291
14.1.1 基本特性(TIM0/1/2) 291
14.1.2 基本特性(TIM3) 292
14.2 Timer 功能描述 293
14.2.1 定时器时钟 293
14.2.2 定时计数器 293
14.2.3 定时器预除频 293
14.2.4 模式 0 计数定时器功能 294
14.2.5 模式 1 脉宽测量 PWC 297
14.2.6 模式 2/3 比较捕获模式 301
14.2.7 模式 2/3 从模式 323
14.2.8 正交编码计数功能 324
14.2.9 Timer 触发 ADC 327
14.2.10 刹车控制 328
14.2.11 Timer 互联 328
14.2.12 GATE 输入互联 329
14.2.13 ETR 输入互联 329
14.2.14 CHx 捕获输入互联 329

14.2.15 DMA 330
14.3 Timer 寄存器描述 332
14.4 模式 0 定时器寄存器描述 333
14.4.1 16 位模式重载寄存器（TIMx_ARR） 333
14.4.2 16 位模式计数寄存器（TIMx_CNT） 333
14.4.3 32 位模式计数寄存器（TIMx_CNT32） 334
14.4.4 控制寄存器（TIMx_M0CR） 335
14.4.5 中断标志寄存器（TIMx_IFR） 337
14.4.6 中断标志清除寄存器（TIMx_ICLR） 337
14.4.7 死区时间寄存器（TIMx_DTR） 338
14.5 脉冲宽度测量 PWC 寄存器描述 339
14.5.1 16 位模式计数寄存器（TIMx_CNT） 339
14.5.2 控制寄存器（TIMx_M1CR） 340
14.5.3 中断标志寄存器（TIMx_IFR） 342
14.5.4 中断标志清除寄存器（TIMx_ICLR） 342
14.5.5 主从模式控制寄存器（TIMx_MSCR） 343
14.5.6 输出控制滤波（TIMx_FLTR） 344
14.5.7 控制寄存器（TIMx_CR0） 345
14.5.8 比较捕获寄存 (TIMx_CCR0A) 346
14.6 模式 2,3 寄存器描述 347
14.6.1 16 位模式重载寄存器（TIMx_ARR） 347
14.6.2 16 位模式计数寄存器（TIMx_CNT） 347
14.6.3 控制寄存器（TIMx_M23CR） 348
14.6.4 中断标志寄存器（TIMx_IFR） 351
14.6.5 中断标志清除寄存器（TIMx_ICLR） 353
14.6.6 主从模式控制寄存器（TIMx_MSCR） 354
14.6.7 输出控制/输入滤波（TIMx_FLTR） 356
14.6.8 ADC 触发控制寄存器（TIMx_ADTR） 359
14.6.9 通道 0 控制寄存器（TIMx_CRCH0） 360
14.6.10 通道 1/2 控制寄存器（TIM3_CRCH1/2）（仅 TIM3 存在） 362
14.6.11 死区时间寄存器（TIMx_DTR） 364
14.6.12 重复周期设置值（TIMx_RCR） 365
14.6.13 通道 0 比较捕获寄存器 (TIMx_CCR0A/B) 365
14.6.14 通道 1/2 比较捕获寄存器 (TIM3_CCR1/2 A/B) （仅 TIM3 存在） 366
15 可编程计数阵列（PCA） 367

15.1 PCA 简介 367
15.2 PCA 功能描述 367
15.2.1 PCA 定时/计数器 368
15.2.2 PCA 捕获功能 370
15.2.3 PCA 比较功能 371
15.3 PCA 模块与其他模块互连及控制 376
15.4 PCA 寄存器描述 377
15.4.1 控制寄存器（PCA_CCON） 378
15.4.2 模式寄存器（PCA_CMOD） 379
15.4.3 计数寄存器（PCA_CNT） 380
15.4.4 中断清除寄存器（PCA_ICLR） 380
15.4.5 比较捕获模式寄存器（PCA_CCAPM0~4） 381
15.4.6 比较捕获数据寄存器高 8 位（PCA_CCAP0~4H） 382
15.4.7 比较捕获数据寄存器低 8 位（PCA_CCAP0~4L） 382
15.4.8 比较捕获 16 位寄存器（PCA_CCAP0~4） 383
15.4.9 比较高速输出标志寄存器（PCA_CCAPO） 383
15.4.10 周期寄存器（PCA_CARR） 384
15.4.11 增强 PWM 控制（PCA_EPWM） 384
16 高级定时器（TIM4/5/6） 385
16.1 Advanced Timer 简介 385
16.2 Advanced Timer 功能描述 387
16.2.1 基本动作 387
16.2.2 时钟源选择 389
16.2.3 计数方向 390
16.2.4 数字滤波 390
16.2.5 软件同步 391
16.2.6 硬件同步 391
16.2.7 缓存功能 394
16.2.8 通用 PWM 输出 395
16.2.9 正交编码计数 400
16.2.10 周期间隔响应 404
16.2.11 保护机制 405
16.2.12 中断说明 405
16.2.13 DMA 406
16.2.14 刹车保护 406

16.2.15 内部互连 408
16.3 寄存器描述 411
16.3.1 通用计数基准值寄存器（TIMx_CNTER） 413
16.3.2 通用周期基准值寄存器（TIMx_PERAR） 413
16.3.3 通用周期缓存寄存器（TIMx_PERBR） 414
16.3.4 通用比较基准值寄存器（TIMx_GCMAR-GCMDR） 414
16.3.5 专用比较基准值寄存器（TIMx_SCMAR-SCMBR） 415
16.3.6 死区时间基准值寄存器（TIMx_DTUAR- DTDAR） 415
16.3.7 通用控制寄存器（TIMx_GCONR） 416
16.3.8 中断控制寄存器（TIMx_ICONR） 417
16.3.9 端口控制寄存器（TIMx_PCONR） 419
16.3.10 缓存控制寄存器（TIMx_BCONR） 422
16.3.11 死区控制寄存器（TIMx_DCONR） 423
16.3.12 滤波控制寄存器（TIMx_FCONR） 424
16.3.13 有效周期寄存器（TIMx_VPERR） 425
16.3.14 状态标志寄存器（TIMx_STFLR） 426
16.3.15 硬件启动事件选择寄存器（TIMx_HSTAR） 428
16.3.16 硬件停止事件选择寄存器（TIMx_HSTPR） 430
16.3.17 硬件清零事件选择寄存器（TIMx_HCELR） 432
16.3.18 硬件捕获 A 事件选择寄存器（TIMx_HCPAR） 434
16.3.19 硬件捕获 B 事件选择寄存器（TIMx_HCPBR） 436
16.3.20 硬件递加事件选择寄存器（TIMx_HCUPR） 438
16.3.21 硬件递减事件选择寄存器（TIMx_HCDOR） 440
16.3.22 软件同步启动寄存器（TIMx_SSTAR） 442
16.3.23 软件同步停止寄存器（TIMx_SSTPR） 443
16.3.24 软件同步清零寄存器（TIMx_SCLRR） 444
16.3.25 中断标志寄存器（TIMx_IFR） 445
16.3.26 中断标志清除寄存器（TIMx_ICLR） 446
16.3.27 展频及中断触发选择（TIMx_CR） 447
16.3.28 AOS 选择控制寄存器（TIMx_AOSSR） 449
16.3.29 AOS 选择控制寄存器标志清除（TIMx_AOSCL） 450
16.3.30 端口刹车控制寄存器（TIMx_PTBKS） 451
16.3.31 端口触发控制寄存器（TIMx_TTRIG） 452
16.3.32 AOS 触发控制寄存器（TIMx_ITRIG） 453
16.3.33 端口刹车极性控制寄存器（TIMx_PTBKP） 454

17 看门狗定时器（WDT）..... 455
17.1 WDT 简介..... 455
17.2 WDT 功能描述..... 456
17.2.1 WDT 溢出后产生中断..... 456
17.2.2 WDT 溢出后产生复位..... 456
17.3 WDT 寄存器描述..... 457
17.3.1 WDT 清除控制寄存器（WDT_RST）..... 458
17.3.2 WDT_CON 寄存器..... 459
18 通用同步异步收发器（UART）..... 460
18.1 简介..... 460
18.2 主要特性..... 460
18.3 功能描述..... 461
18.3.1 工作模式..... 461
18.3.2 波特率生成..... 465
18.3.3 帧错误检测..... 469
18.3.4 多机通讯..... 469
18.3.5 DMAC 硬件握手..... 469
18.3.6 硬件流控..... 470
18.3.7 收发端缓存..... 472
18.4 寄存器..... 473
18.4.1 数据寄存器（UARTx_SBUF）..... 474
18.4.2 控制寄存器（UARTx_SCON）..... 475
18.4.3 地址寄存器（UARTx_SADDR）..... 477
18.4.4 地址掩码寄存器（UARTx_SADEN）..... 477
18.4.5 标志位寄存器（UARTx_ISR）..... 478
18.4.6 标志位清除寄存器（UARTx_ICR）..... 478
18.4.7 波特率寄存器（UARTx_SCNT）..... 479
19 低功耗同步异步收发器（LPUART）..... 480
19.1 简介..... 480
19.2 主要特性..... 481
19.3 功能描述..... 481
19.3.1 配置时钟和传输时钟..... 481
19.3.2 工作模式..... 482
19.3.3 波特率生成..... 486
19.3.4 帧错误检测..... 489

19.3.5 多机通讯 489
19.3.6 DMAC 硬件握手 490
19.3.7 硬件流控 490
19.3.8 收发端缓存 492
19.4 寄存器 494
19.4.1 数据寄存器（LPUARTx_SBUF） 495
19.4.2 控制寄存器（LPUARTx_SCON） 496
19.4.3 地址寄存器（LPUARTx_SADDR） 498
19.4.4 地址掩码寄存器（LPUARTx_SADEN） 498
19.4.5 标志位寄存器（LPUARTx_ISR） 499
19.4.6 标志位清除寄存器（LPUARTx_ICR） 499
19.4.7 波特率寄存器（LPUARTx_SCNT） 500
20 循环冗余校验（CRC） 501
20.1 概述 501
20.2 主要特性 501
20.3 功能描述 501
20.3.1 工作模式 501
20.3.2 编码方式 501
20.3.3 写入位宽 501
20.4 编程示例 502
20.4.1 CRC-16 编码模式 502
20.4.2 CRC-16 检验模式 502
20.4.3 CRC-32 编码模式 502
20.4.4 CRC-32 检验模式 502
20.5 寄存器描述 503
20.5.1 寄存器列表 503
20.5.2 控制寄存器（CRC_CR） 504
20.5.3 结果寄存器（CRC_RESULT） 504
20.5.4 数据寄存器（CRC_DATA） 505
21 真随机数发生器（TRNG） 506
21.1 概述 506
21.2 功能框图 506
21.3 功能描述 506
21.4 寄存器 507
21.4.1 控制寄存器（TRNG_CR） 508

21.4.2 模式寄存器（TRNG_MODE）.....509
21.4.3 数据寄存器 0（TRNG_DATA0）.....510
21.4.4 数据寄存器 1（TRNG_DATA1）.....510
21.5 软件基本操作.....511
21.5.1 生成 64bits 真随机数的操作流程（上电第一次）.....511
21.5.2 生成 64bits 真随机数的操作流程（非上电第一次生成）.....512
22 高级加密标准模块（AES）.....513
22.1 功能定义.....513
22.1.1 AES 算法简述.....513
22.1.2 AES 模块功能描述.....514
22.2 模块寄存器说明.....515
22.2.1 控制寄存器（AES_CR）.....516
22.2.2 数据寄存器（AES_Data）.....517
22.2.3 密钥寄存器（AES_Key）.....518
22.3 异常机制.....520
22.4 本模块操作说明.....520
22.4.1 IP 操作的共同点.....520
22.4.2 加密操作流程.....520
22.4.3 解密操作流程.....520
22.4.4 数据示例.....521
22.5 运行时间说明.....524
23 Crystal-less USB 时钟校准器 （CTS）.....525
23.1 48M 时钟校准.....525
23.1.1 系统特性.....525
23.1.2 框图.....526
23.1.3 基准参考源同步输入.....526
23.1.4 频率误差测量.....526
23.1.5 频率误差微调.....527
23.1.6 CTS 初始化和配置.....528
23.2 其他振荡校准.....528
23.2.1 系统特性.....528
23.2.2 框图.....529
23.2.3 校准说明.....529
23.3 定时器功能说明.....529
23.3.1 框图.....529

23.3.2 定时器时钟选择 530
23.3.3 定时器 530
23.4 CTS 寄存器 531
23.4.1 CTS 控制寄存器（CTS_CR） 532
23.4.2 CTS 配置寄存器（CTS_CFGR） 533
23.4.3 CTS 中断和状态寄存器（CTS_ISR） 535
23.4.4 CTS 中断状态清除寄存器（CTS_ICR） 536
24 音频接口（I2S） 537
24.1 功能说明 537
24.1.1 管脚复用 538
24.1.2 I2S 全双工 538
24.2 I2S 音频协议说明 539
24.2.1 I2S Philips 标准 540
24.2.2 MSB 对齐标准 542
24.2.3 LSB 对齐标准 544
24.2.4 PCM 标准 546
24.3 I2S 时钟发生器 548
24.4 I2S 工作模式 552
24.4.1 I2S 主模式 552
24.4.2 I2S 从模式 554
24.5 I2S 状态标志 555
24.5.1 忙标志 (BSY) 555
24.5.2 发送缓冲区为空 (TXE_L TXE_R) 556
24.5.3 接收缓冲区非空 (RXNE_L RXNE_R) 556
24.5.4 下溢标志 (UDR) 556
24.5.5 上溢标志 (OVR) 556
24.5.6 帧错误标志 (FRE) 556
24.6 I2S 中断 DMA 557
24.7 I2S 寄存器 558
24.7.1 I2Sx 控制寄存器（I2Sx_CR） 559
24.7.2 I2Sx 状态寄存器（I2Sx_SR） 560
24.7.3 I2Sx 配置寄存器（I2Sx_CFGR） 562
24.7.4 I2Sx 预分频寄存器（I2Sx_PR） 564
24.7.5 I2Sx 中断状态清除寄存器（I2Sx_ICR） 565
24.7.6 I2Sx 数据寄存器 L（I2Sx_DRL） 566

24.7.7 I2Sx 数据寄存器 R（I2Sx_DRR） 566
25 USB2.0 全速模块（USBFS） 567
25.1 USBFS 简介 567
25.2 USBFS 主要特性 567
25.2.1 通用特性 567
25.2.2 设备模式特性 567
25.3 USBFS 系统框图 568
25.4 USBFS 管脚说明 568
25.5 USBFS 功能说明 569
25.5.1 USBFS 时钟以及工作模式 569
25.5.2 USBFS 设备功能 569
25.5.3 USBFS 功耗控制 572
25.5.4 USBFS 数据 FIFO 573
25.5.5 USBFS 设备 FIFO 架构 573
25.5.6 USBFS FIFO RAM 分配 574
25.5.7 USBFS 系统性能 575
25.5.8 USBFS 中断和事件 575
25.6 USBFS 编程模型 576
25.6.1 USBFS 模块初始化 576
25.6.2 USBFS 设备初始化 576
25.6.3 USBFS 设备软断开操作 577
25.6.4 USB 复位时的端点初始化 577
25.6.5 USB 枚举完成时的端点初始化 578
25.6.6 收到 SetAddress 命令时的端点初始化 578
25.6.7 收到 SetConfiguration/SetInterface 命令时的端点初始化 578
25.6.8 端点激活(Active) 578
25.6.9 端点不激活(Deactive) 579
25.6.10 禁止(Disable)输出端点 579
25.6.11 暂停(Stall)非同步输出端点 579
25.6.12 全局输出端点 NAK 模式 580
25.6.13 输出端点传输停止 580
25.6.14 配置中断输入端点为周期或非周期端点（共享 FIFO 模式） 580
25.6.15 全局非周期输入端点 NAK 模式（共享 FIFO 模式） 580
25.6.16 发生输入端点 NAK 581
25.6.17 禁止非周期输入端点（共享 FIFO 模式） 581

25.6.18 禁止周期输入端点（共享 FIFO 模式）.....581
25.6.19 非周期输入数据传输超时（共享 FIFO 模式）.....582
25.6.20 非同步输入端点暂停（共享 FIFO 模式）.....582
25.6.21 输入端点传输停止（共享 FIFO 模式）.....583
25.6.22 端点不匹配（共享 FIFO 模式）.....583
25.6.23 控制传输.....583
25.6.24 输出数据传输.....588
25.6.25 非周期批量输入数据传输无阈值.....588
25.6.26 非周期批量输入数据传输有阈值.....589
25.6.27 非同步输出数据传输无阈值.....589
25.6.28 非同步输出数据传输有阈值.....589
25.6.29 同步输出数据传输有阈值.....589
25.6.30 未完成同步输出数据传输.....589
25.6.31 周期输入数据传输无阈值.....590
25.6.32 周期输入数据传输有阈值.....591
25.7 寄存器说明.....592
25.7.1 USBFS 全局寄存器.....594
25.7.2 USBFS 设备模式寄存器.....611
25.7.3 USBFS 时钟门控控制寄存器.....644
26 控制器局域网（CAN）.....645
26.1 简介.....645
26.2 CAN 系统框图.....646
26.3 管脚说明.....646
26.4 功能说明.....647
26.4.1 波特率设定.....647
26.4.2 发送缓冲器.....648
26.4.3 接收缓冲器.....649
26.4.4 接收筛选寄存器组.....649
26.4.5 数据发送.....650
26.4.6 单次数据发送.....650
26.4.7 取消数据发送.....651
26.4.8 数据接收.....651
26.4.9 错误处理.....651
26.4.10 节点关闭.....652
26.4.11 仲裁失败位置捕捉.....652

26.4.12 回环模式 652
26.4.13 静默模式 653
26.4.14 软件复位功能 654
26.4.15 向上兼容 CAN-FD 功能 655
26.4.16 时间触发 TTCAN 655
26.4.17 中断 658
26.5 寄存器说明 659
26.5.1 CAN 接收 BUF 寄存器（CAN_RBUF） 661
26.5.2 CAN 发送 BUF 寄存器（CAN_TBUF） 663
26.5.3 CAN 配置和状态寄存器（CAN_CFG_STAT） 665
26.5.4 CAN 命令寄存器（CAN_TCMD） 666
26.5.5 CAN 发送控制寄存器（CAN_TCTRL） 668
26.5.6 CAN 接收控制寄存器（CAN_RCTRL） 670
26.5.7 CAN 接收和发送中断使能寄存器（CAN_RTIE） 671
26.5.8 CAN 接收和发送中断状态寄存器（CAN_RTIF） 672
26.5.9 CAN 错误中断使能和标志寄存器（CAN_ERRINT） 674
26.5.10 CAN 位时序寄存器（CAN_BT） 675
26.5.11 CAN 错误和仲裁失败捕捉寄存器（CAN_EALCAP） 676
26.5.12 CAN 警告限定寄存器（CAN_LIMIT） 676
26.5.13 CAN 接收错误计数器寄存器（CAN_RECNT） 677
26.5.14 CAN 发送错误计数器寄存器（CAN_TECNT） 677
26.5.15 CAN 筛选器组控制寄存器（CAN_ACFCTRL） 678
26.5.16 CAN 筛选器组使能寄存器（CAN_ACFEN） 679
26.5.17 CAN 筛选器组 code 和 mask 寄存器（CAN_ACF） 680
26.5.18 TTCAN TB slot 指针寄存器（CAN_TBSLOT） 681
26.5.19 TTCAN 时间触发配置寄存器（CAN_TTCFG） 682
26.5.20 TTCAN 参考消息寄存器（CAN_REF_MSG） 683
26.5.21 TTCAN 触发配置寄存器（CAN_TRG_CFG） 684
26.5.22 TTCAN 触发时间寄存器（CAN_TRG_TRIG） 685
26.5.23 TTCAN 触发看门时间寄存器（TRG_WTRIG） 685
26.6 使用注意事项 686
26.6.1 CAN 总线抗干扰措施 686
26.6.2 CAN 控制器噪声制约 686
27 模数转换器（ADC） 687
27.1 模块简介 687

27.2 ADC 框图 687
27.3 转换时序及转换速度 688
27.4 单次转换模式 688
27.5 扫描转换模式 690
27.5.1 顺序扫描转换模式 690
27.5.2 插队扫描转换模式 693
27.5.3 扫描转换触发 DMA 读取 696
27.6 连续转换累加模式 696
27.7 ADC 转换外部触发源 699
27.8 ADC 转换结果比较 699
27.9 ADC 中断 700
27.10 使用温度传感器测量环境温度 701
27.11 ADC 模块寄存器 703
27.11.1 ADC 基本配置寄存器 0（ADC_CR0） 705
27.11.2 ADC 基本配置寄存器 1（ADC_CR1） 708
27.11.3 ADC 顺序扫描转换通道配置寄存器 0（ADC_SQR0） 710
27.11.4 ADC 顺序扫描转换通道配置寄存器 1（ADC_SQR1） 710
27.11.5 ADC 顺序扫描转换通道配置寄存器 2（ADC_SQR2） 712
27.11.6 ADC 插队扫描转换通道配置寄存器（ADC_JQR） 713
27.11.7 ADC 顺序扫描转换通道 x 转换结果（ADC_SqrResult0 - 15） 713
27.11.8 ADC 插队扫描转换通道 x 转换结果（ADC_JqrResult0 - 3） 714
27.11.9 ADC 转换结果（ADC_Result） 714
27.11.10 ADC 转换结果累加值（ADC_ResultAcc） 715
27.11.11 ADC 比较上阈值（ADC_HT） 715
27.11.12 ADC 比较下阈值（ADC_LT） 716
27.11.13 ADC 中断标志寄存器（ADC_IFR） 717
27.11.14 ADC 中断清除寄存器（ADC_ICR） 718
27.11.15 ADC 单次转换或顺序扫描转换外部中断触发源配置寄存器（ADC_ExtTrigger0） 719
27.11.16 ADC 插队扫描转换外部中断触发源配置寄存器（ADC_ExtTrigger1） 721
27.11.17 ADC 单次转换启动控制寄存器（ADC_SglStart） 723
27.11.18 ADC 顺序扫描转换启动控制寄存器（ADC_SqrStart） 723
27.11.19 ADC 插队扫描转换启动控制寄存器（ADC_JqrStart） 724
27.11.20 ADC 一直转换启动停止控制寄存器（ADC_allStart） 724
28 数模转换器（DAC） 725
28.1 DAC 特性 725

28.2 功能说明 726
28.2.1 框图 726
28.2.2 DAC 输出缓冲器使能 727
28.2.3 DAC 通道使能 727
28.2.4 DAC 输出电压 727
28.2.5 DAC 触发选择 728
28.2.6 单模式功能说明 729
28.2.7 双模式功能说明 730
28.2.8 噪音生成 735
28.2.9 三角波生成 735
28.2.10 DMA 请求 736
28.3 DAC 寄存器 738
28.3.1 DAC 控制寄存器（DAC_CR0） 739
28.3.2 DAC 软件触发寄存器（DAC_SWTRIGR） 742
28.3.3 DAC0 12 位右对齐数据保持寄存器（DAC_DHR12R0） 742
28.3.4 DAC0 12 位左对齐数据保持寄存器（DAC_DHR12L0） 743
28.3.5 DAC0 8 位右对齐数据保持寄存器（DAC_DHR8R0） 743
28.3.6 DAC1 12 位右对齐数据保持寄存器（DAC_DHR12R1） 744
28.3.7 DAC1 12 位左对齐数据保持寄存器（DAC_DHR12L1） 744
28.3.8 DAC1 8 位右对齐数据保持寄存器（DAC_DHR8R1） 745
28.3.9 双 DAC 12 位右对齐数据保持寄存器（DAC_DHR12RD） 745
28.3.10 双 DAC 12 位左对齐数据保持寄存器（DAC_DHR12LD） 746
28.3.11 双 DAC 8 位右对齐数据保持寄存器（DAC_DHR8RD） 746
28.3.12 DAC0 数据输出寄存器（DAC_DOR0） 747
28.3.13 DAC1 数据输出寄存器（DAC_DOR1） 747
28.3.14 DAC 状态寄存器（DAC_SR） 748
28.3.15 DAC 触发选择寄存器（DAC_ETRS） 749
29 模拟比较器（VC） 750
29.1 模拟电压比较器 VC 简介 750
29.2 电压比较器框架图 751
29.3 建立/响应时间 752
29.4 滤波时间 752
29.5 迟滞功能 752
29.6 VC 寄存器 753
29.6.1 VC 配置寄存器（VC_CR） 754

29.6.2 VC0 配置寄存器（VC0_CR） 756
29.6.3 VC1 配置寄存器（VC1_CR） 758
29.6.4 VC0 输出配置寄存器（VC0_OUT_CFG） 760
29.6.5 VC1 输出配置寄存器（VC1_ OUT_CFG） 761
29.6.6 VC 中断寄存器（VC_IFR） 762
29.6.7 VC2 配置寄存器（VC2_CR） 763
29.6.8 VC2 输出配置寄存器（VC2_ OUT_CFG） 765
30 低电压检测器（LVD） 767
30.1 LVD 简介 767
30.2 LVD 框图 767
30.3 迟滞功能 768
30.4 数字滤波 768
30.5 配置示例 769
30.5.1 LVD 配置为低电压复位 769
30.5.2 LVD 配置为电压变化中断 769
30.6 LVD 寄存器 770
30.6.1 LVD 配置寄存器（LVD_CR） 771
30.6.2 LVD 中断寄存器（LVD_IFR） 773
31 运算放大器（OPA） 774
31.1 OPA 特性 774
31.2 OPA 功能描述 774
31.2.1 OPA0/1/2 （通用 OPA） 774
31.2.2 OPA3/4 --- (可复用为 DAC buffer*2) 775
31.3 OPA 寄存器 776
31.3.1 OPA 输出使能寄存器（OPA_CR0） 777
31.3.2 OPA 控制寄存器（OPA_CR1） 778
31.3.3 OPA 配置寄存器（OPA_CR2） 779
32 模拟其它寄存器 780
32.1 BGR 配置寄存器（BGR_CR） 780
33 SWD 调试接口 781
33.1 SWD 调试附加功能 781
33.2 ARM® 参考文档 782
33.3 调试端口引脚 782
33.3.1 SWD 端口引脚 782
33.3.2 SW-DP 引脚分配 782

33.3.3 SWD 引脚上的内部上拉 783
33.4 SWD 端口 783
33.4.1 SWD 协议简介 783
33.4.2 SWD 协议序列 783
33.4.3 SW-DP 状态机（复位、空闲状态、ID 代码） 784
33.4.4 DP 和 AP 读/写访问 784
33.4.5 SW-DP 寄存器 785
33.4.6 SW-AP 寄存器 785
33.5 内核调试 786
33.6 BPU（断点单元） 786
33.6.1 BPU 功能 787
33.7 DWT（数据观察点） 787
33.7.1 DWT 功能 787
33.7.2 DWT 程序计数器采样寄存器 787
33.8 MCU 调试组件（DBG） 787
33.8.1 对低功耗模式的调试支持 787
33.8.2 对定时器、看门狗的调试支持 787
33.9 调试模式模块工作状态控制（DEBUG_ACTIVE） 788
34 器件电子签名 790
34.1 产品唯一身份标识（UID）寄存器（80 位） 790
34.2 产品型号寄存器 790
34.3 FLASH 容量寄存器 791
34.4 RAM 容量寄存器 791
34.5 管脚数量寄存器 791
35 附录 A SysTick 定时器 792
35.1 SysTick 定时器简介 792
35.2 设置 SysTick 792
35.3 SysTick 寄存器 793
35.3.1 SysTick 控制和状态寄存器（CTRL） 793
35.3.2 SysTick 重载寄存器（LOAD） 793
35.3.3 SysTick 当前值寄存器（VAL） 793
35.3.4 SysTick 校准值寄存器（CALIB） 794
36 附录 B 文档约定 795
36.1 寄存器相关缩写词列表 795
36.2 词汇表 795

版本修订记录 796

表索引

表 1-1 地址划分表 39
表 2-1 运行模式下可运行模块图 43
表 2-2 休眠模式下可运行模块图 44
表 2-3 深度休眠模式下可运行模块图 46
表 5-1 Cortex-M0+ 处理器中断一览 82
表 5-2 外部中断与 NVIC 中断输入对应关系 87
表 6-1 端口状态真值表 103
表 6-2 端口复用表 105
表 7-1 I2C 时钟信号波特率 205
表 7-2 I2C 状态码表述 212
表 7-3 寄存器列表 216
表 8-1 SPI 管脚配置说明表 228
表 8-2 SPI 寄存器列表 232
表 8-3 主机模式波特率选择 233
表 9-1 寄存器列表 245
表 10-1 寄存器列表 252
表 11-1 FLASH 容量划分 256
表 11-2 不同频率下 FLASH 擦写时间参数 260
表 12-1 RAM 地址映射 272
表 12-2 寄存器基地址 273
表 14-1 Timer 寄存器列表 332
表 15-1 PCA 比较捕获功能模块设置 376
表 15-2 PCA 寄存器列表 377
表 16-1 Advanced Timer 基本特性 385
表 16-2 Advanced Timer 端口列表 385
表 16-3 AOS 源选择 409
表 16-4 端口触发选择 410
表 16-5 Advanced Timer 寄存器列表 411
表 17-1 WDT 寄存器列表 457
表 18-1 Mode0/1/2/3 数据结构 461
表 18-2 B8 数据含义 461
表 18-3 PCLK=4MHz 波特率计算表 466
表 18-4 PCLK=8MHz 波特率计算表 466

表 18-5 PCLK=16MHz 波特率计算表 467
表 18-6 PCLK=24MHz 波特率计算表 467
表 18-7 PCLK=32MHz 波特率计算表 468
表 18-8 PCLK=48MHz 波特率计算表 468
表 19-1 Mode0/1/2/3 数据结构 482
表 19-2 B8 数据含义 482
表 19-3 SCL 为 4MHz 波特率计算表 486
表 19-4 SCLK 为 8MHz 波特率计算表 487
表 19-5 SCLK 为 16MHz 波特率计算表 487
表 19-6 SCLK 为 24MHz 波特率计算表 488
表 19-7 SCLK 为 32MHz 波特率计算表 488
表 19-8 SCLK 为 48MHz 波特率计算表 489
表 22-1 寄存器列表 515
表 22-2 128 位操作寄存器示例 521
表 22-3 192 位操作寄存器示例 522
表 22-4 192 位操作寄存器示例 523
表 22-5 AES 加解密运行时间 524
表 23-1 CTS 寄存器 531
表 24-1 I2Sx 寄存器 558
表 25-1 USBFS 管脚说明 568
表 25-2 USBFS 中断事件表 575
表 25-3 USBFS 寄存器一览表 592
表 26-1 CAN 管脚说明 646
表 26-2 CAN 位时间设定规则 648
表 26-3 软件复位范围表 654
表 26-4 CAN 中断表 658
表 26-5 CAN 寄存器一览表 659
表 26-6 CAN 寄存器 BYTE/HALFWORD/WORD 访问排布表 660
表 26-7 标准格式 CAN 接收邮箱格式 661
表 26-8 扩展格式 CAN 接收邮箱格式 662
表 26-9 标准格式 CAN 发送邮箱格式 663
表 26-10 扩展格式 CAN 发送邮箱格式 664
表 27-1 ADC 寄存器 703
表 28-1 DAC 寄存器 738
表 29-1 VC 寄存器 753

表 30-1 LVD 寄存器 770
表 31-1 OPA 寄存器 776

图索引

图 1-1 系统架构示意图 37
图 1-2 地址区域划分示意图 38
图 2-1 控制模式框图 41
图 3-1 时钟控制模块框图 48
图 3-2 晶振时钟启动示意图 50
图 3-3 时钟切换示意图 55
图 3-4 时钟校准原理图 56
图 4-1 复位来源示意图 75
图 5-1 只使用了高两位的优先级寄存器 82
图 5-2 中断向量表 83
图 5-3 中断激活和挂起状态 84
图 5-4 中断挂起状态被清除然后被重新确认 84
图 5-5 中断退出时若中断请求保持高电平就会引起中断处理的再次执行 85
图 5-6 中断处理中产生的中断挂起也可以被确认 85
图 5-7 中断结构图 88
图 6-1 端口电路图 101
图 6-2 AHB/FASTIO 总线端口随系统时钟的变化 103
图 6-3 读取端口引脚数据同步图 104
图 7-1 I2C 传输协议 201
图 7-2 START 和 STOP 条件 201
图 7-3 I2C 总线上位传输 202
图 7-4 I2C 总线上应答信号 202
图 7-5 I2C 总线上的仲裁 203
图 7-6 I2C 功能模块图 204
图 7-7 主发送模式数据同步图 206
图 7-8 I2C 主机发送状态图 207
图 7-9 主接收模式数据同步图 208
图 7-10 I2C 主机接收状态图 208
图 7-11 从接收模式数据同步图 209
图 7-12 从机接收状态图 209
图 7-13 从发送模式数据同步图 210
图 7-14 I2C 从机发送状态图 210
图 7-15 I2C 广播呼叫状态图 211

图 8-1 从机接收示意图 223
图 8-2 从机发送示意图 223
图 8-3 主机模式帧格式 224
图 8-4 从机 CPHA 为 0 时数据帧格式 224
图 8-5 从机 CHPA 为 1 时数据帧格式 225
图 8-6 SPI 多主机/多从机系统的示意图 227
图 9-1 时钟源选择示意图 240
图 9-2 时钟校准模块硬件示意图 240
图 9-3 时钟校准波形示意图 241
图 9-4 CLKTRIM 时钟选择 242
图 9-5 时钟监控波形示意图 244
图 13-1 功能框图 277
图 14-1 TIM0/1/2 框图 292
图 14-2 TIM3 框图 293
图 14-3 自由计数框图 294
图 14-4 重载计数波形 295
图 14-5 16 位重载计数波形 295
图 14-6 32 位自由计数波形 295
图 14-7 自由计数时序图 296
图 14-8 重载计数时序图（预分频设置为 2） 296
图 14-9 PWC 测量框图 298
图 14-10 高电平脉冲宽度测量 298
图 14-11 下降沿到下降沿周期测量 299
图 14-12 上升沿到上升沿周期测量 299
图 14-13 上升沿到上升沿周期测量单次模式 300
图 14-14 计数器框图 301
图 14-15 无预分频的向上计数 302
图 14-16 带预分频的上计数 303
图 14-17 不带预分频的下计数 303
图 14-18 带预分频的下计数 304
图 14-19 带预分频的上下计数 304
图 14-20 边沿对齐计时器波形(DIR =1) 304
图 14-21 边沿对齐计时器波形(DIR =0) 305
图 14-22 中心对齐计数器波形 305
图 14-23 重复计数器产生更新时序 306

图 14-24 三角波模式下缓存使能.....306
图 14-25 三角波模式下缓存无效.....307
图 14-26 锯齿波模式下上计数缓存使能.....307
图 14-27 锯齿波模式下上计数缓存无效.....308
图 14-28 锯齿波模式下计数缓存使能.....308
图 14-29 锯齿波模式下计数缓存无效.....309
图 14-30 锯齿波模式下计数比较缓存使能.....309
图 14-31 OCREF 输出框图.....310
图 14-32 锯齿波计数单点比较 OCREF 输出波形（OCMx=111）.....311
图 14-33 三角波计数单点比较 OCREF 输出波形（OCMx=111）.....311
图 14-34 锯齿波计数双点比较 OCREF 输出（OCMx=111）.....312
图 14-35 三角波计数双点比较 OCREF 输出（OCMx=111）.....312
图 14-36 独立 PWM 输出框图.....313
图 14-37 CCPx=0 时 PWM 输出波形.....313
图 14-38 CCPx=1 时 PWM 输出波形.....313
图 14-39 互补 PWM 输出框图.....314
图 14-40 互补 PWM 输出波形图.....314
图 14-41 互补 PWM 输出波形图.....315
图 14-42 死区时间.....315
图 14-43 三角波模式单脉冲计数.....316
图 14-44 锯齿波上计数单脉冲模式.....316
图 14-45 锯齿波下计数单脉冲模式.....317
图 14-46 中断示意图.....318
图 14-47 捕获功能框图.....318
图 14-48 捕获时序图.....319
图 14-49 CHA 端口选择.....319
图 14-50 CHB 端口选择.....320
图 14-51 从模式示意图.....323
图 14-52 门控功能.....323
图 14-53 触发和复位功能.....324
图 14-54 编码计数.....326
图 14-55 ADC 触发.....327
图 15-1 PCA 整体框图.....367
图 15-2 PCA 计数器框图.....369
图 15-3 PCA 捕获功能框图.....371

图 15-4 PCA 比较功能框图.....372
图 15-5 PCA 16 位 PWM 功能框图.....373
图 15-6 PCA WDT 功能框图.....374
图 15-7 PCA PWM 功能框图.....375
图 15-8 PCA PWM 输出波形.....376
图 16-1 Advanced Timer 框图.....386
图 16-2 锯齿波波形（递加计数）.....387
图 16-3 三角波波形.....387
图 16-4 比较输出动作.....388
图 16-5 捕获输入动作.....389
图 16-6 捕获输入端口的滤波功能.....390
图 16-7 软件同步动作.....391
图 16-8 硬件同步动作.....393
图 16-9 单缓存方式比较输出时序.....394
图 16-10 PWM 展频输出示意图.....395
图 16-11 CHA 输出 PWM 波.....396
图 16-12 三角波 A 模式时软件设定 GCMBR 互补 PWM 波输出.....396
图 16-13 三角波 B 模式时硬件设定 GCMBR 互补 PWM 波输出（对称死区）.....397
图 16-14 6 相 PWM 波.....398
图 16-15 三角波 A 模式时带死区时间三相互补 PWM 波输出.....399
图 16-16 位置模式时基本计数动作.....400
图 16-17 位置模式时相位差计数动作设定(1 倍).....401
图 16-18 位置模式时相位差计数动作设定(2 倍).....401
图 16-19 位置模式时相位差计数动作设定(4 倍).....401
图 16-20 位置模式时方向计数动作.....402
图 16-21 公转模式时 Z 相计数动作.....402
图 16-22 公转模式时位置计数器输出计数动作.....403
图 16-23 公转模式时 Z 相计数和位置计数器输出混合计数动作.....403
图 16-24 公转计数模式-混合计数 Z 相屏蔽动作例 1.....404
图 16-25 公转计数模式-混合计数 Z 相屏蔽动作例 2.....404
图 16-26 周期间隔有效请求信号动作.....405
图 16-27 端口刹车与软件刹车示意图.....407
图 16-28 输出同高同低刹车示意图.....407
图 16-29 VC 刹车控制示意图.....408
图 16-30 Timer4/5/6 中断选择.....408

图 17-1 WDT 整体框图 455
图 18-1 结构框图 460
图 18-2 Mode0 发送数据 462
图 18-3 Mode0 接收数据 462
图 18-4 Mode1 发送数据 463
图 18-5 Mode1 接收数据 463
图 18-6 Mode2 发送数据 463
图 18-7 Mode2 接收数据 464
图 18-8 Mode3 发送数据 464
图 18-9 Mode3 接收数据 464
图 18-10 UART 硬件流控 470
图 18-11 nRTS 硬件流控信号 471
图 18-12 nCTS 硬件流控信号 471
图 18-13 接收缓存 472
图 18-14 发送缓存 472
图 19-1 结构框图 480
图 19-2 Mode0 发送数据 483
图 19-3 Mode0 接收数据 483
图 19-4 Mode1 发送数据 484
图 19-5 Mode1 接收数据 484
图 19-6 Mode2 发送数据 484
图 19-7 Mode2 接收数据 484
图 19-8 Mode3 发送数据 485
图 19-9 Mode3 接收数据 485
图 19-10 LPUART 硬件流控 491
图 19-11 nRTS 硬件流控信号 491
图 19-12 nCTS 硬件流控信号 492
图 19-13 接收缓存 492
图 19-14 发送缓存 493
图 21-1 TRNG 数据流 506
图 22-1 AES 的加解密示意图 513
图 22-2 AES 的加密流程图 514
图 23-1 CTS 计数器 527
图 24-1 功能框图 537
图 24-2 Philip 协议波形（16/32 位全精度） 540

图 24-3 Philip 协议波形（24 位帧） 540
图 24-4 24 位数据发送写操作 541
图 24-5 24 位数据模式接收读操作 541
图 24-6 Philip 协议 16 位扩展到 32 位数据帧 541
图 24-7 16 位数据读写操作 542
图 24-8 MSB 协议波形（16/32 位全精度） 542
图 24-9 MSB 协议波形（24 位帧） 543
图 24-10 MSB 协议 16 位扩展到 32 位数据帧 543
图 24-11 LSB 协议波形（16/32 位全精度） 544
图 24-12 LSB 协议波形（24 位帧） 544
图 24-13 LSB 24 位数据发送写操作 545
图 24-14 LSB 24 位数据模式接收读操作 545
图 24-15 LSB 协议 16 位扩展到 32 位数据帧 545
图 24-16 16 位数据读写操作 546
图 24-17 PCM 协议波形（16 位） 546
图 24-18 PCM 协议波形（16 位数据扩展到 32 位） 547
图 24-19 音频采样频率定义 548
图 24-20 时钟产生 548
图 25-1 USBFS 系统框图 568
图 25-2 USBFS 设备模式系统构建图 569
图 25-3 USBFS 设备模式下 FIFO 架构示意图 573
图 26-1 CAN 系统框图 646
图 26-2 CAN 位时间定义图 647
图 26-3 CAN TBUF 寄存器写发送缓冲器和示意图 648
图 26-4 CAN RBUF 寄存器读接收缓冲器示意图 649
图 26-5 CAN ACF 寄存器访问筛选器组示意图 650
图 26-6 CAN LBMI 和 LBME 示意图 653
图 27-1 ADC 示意框图 687
图 27-2 ADC 转换时序图 688
图 27-3 ADC 顺序扫描转换过程示例 691
图 27-4 ADC 插队扫描转换过程示例 693
图 27-5 ADC 顺序扫描过程中进行插队扫描示例 696
图 27-6 ADC 连续转换累加过程示例 697
图 27-7 ADC 单次转换或顺序扫描转换外部触发源示意图 699
图 27-8 ADC 插队扫描转换外部触发源示意图 699

图 27-9 ADC 转换结果 700
图 28-1 触发不使能时 DAC 转换时序图 729
图 28-2 LFSR 产生波形的 DAC 转换（使用软件触发） 735
图 28-3 生成 DAC 三角波 736
图 28-4 生成三角波波形的 DAC 转换（使能软件触发） 736
图 29-1 VC 框架图 751
图 29-2 VC 滤波响应时间 752
图 29-3 VC 迟滞功能 752
图 30-1 LVD 框图 767
图 33-1 调试支持框图 781

简介（Overview）

HC32F072 系列是一款宽电压工作范围的通用 MCU。集成 12 位 1Msps 高精度 SARADC，2 个 12 位 DAC 以及集成了比较器、运放、内置高性能 PWM 定时器、多路 UART、SPI、I2C、I2S、USB、CAN 等丰富的通讯外设，内建 AES、TRNG 等信息安全模块，具有高整合度、高抗干扰、高可靠性的特点。本产品内核采用 Cortex-M0+ 内核，配合成熟的 Keil & IAR 调试开发软件，支持 C 语言及汇编语言，汇编指令。

通用 MCU 典型应用

可广泛应用于各类市场应用：如人机交互、手持设备、游戏外设、打印机、可视对讲等智能家居应用。

关于本手册

本手册主要介绍芯片的功能、操作事项和使用方法。关于芯片的规格，请参阅对应的“数据手册”。

1 系统结构

1.1 概述

本产品系统由以下部分组成：

- 2 个 AHB 总线 Master：
  - Cortex-M0+
  - DMA 控制器
- 4 个 AHB 总线 Slaves：
  - FLASH 存储器
  - SRAM 存储器
  - AHB0，AHB to APB Bridge，包含所有 APB 接口外设
  - AHB1，包含所有 AHB 接口外设

整个系统总线结构采用多层次 AHB-lite 总线互连实现。如下图所示：

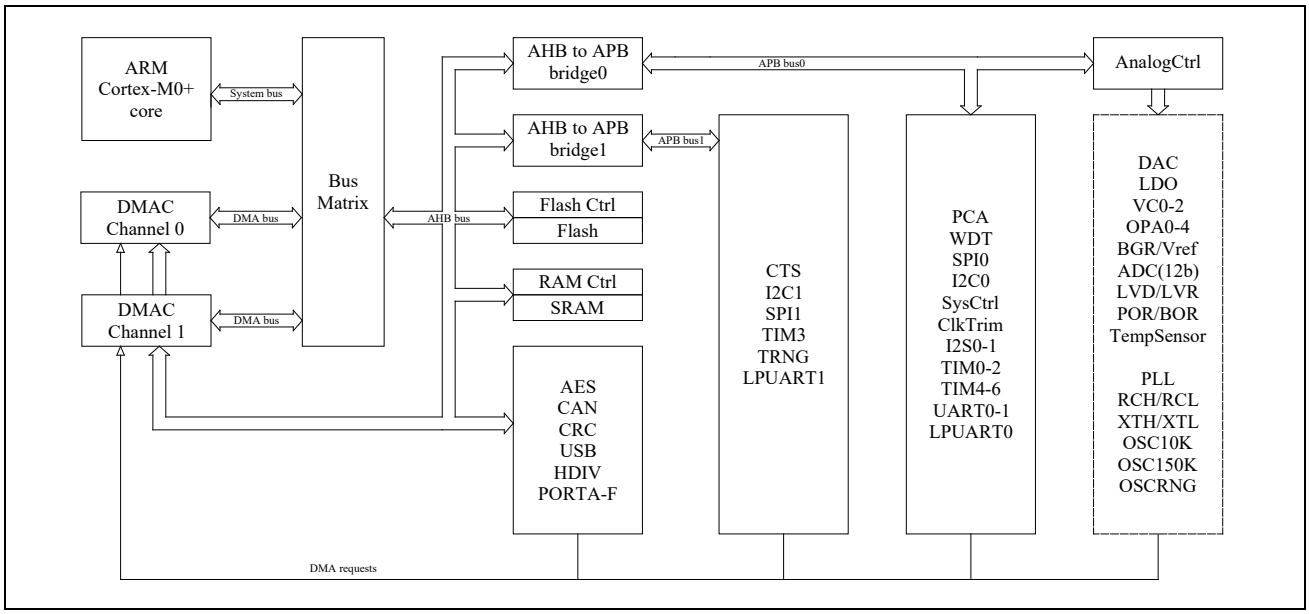


图 1-1 系统架构示意图

1.2 系统地址划分

整个系列系统的地址区域划分，如下图所示：

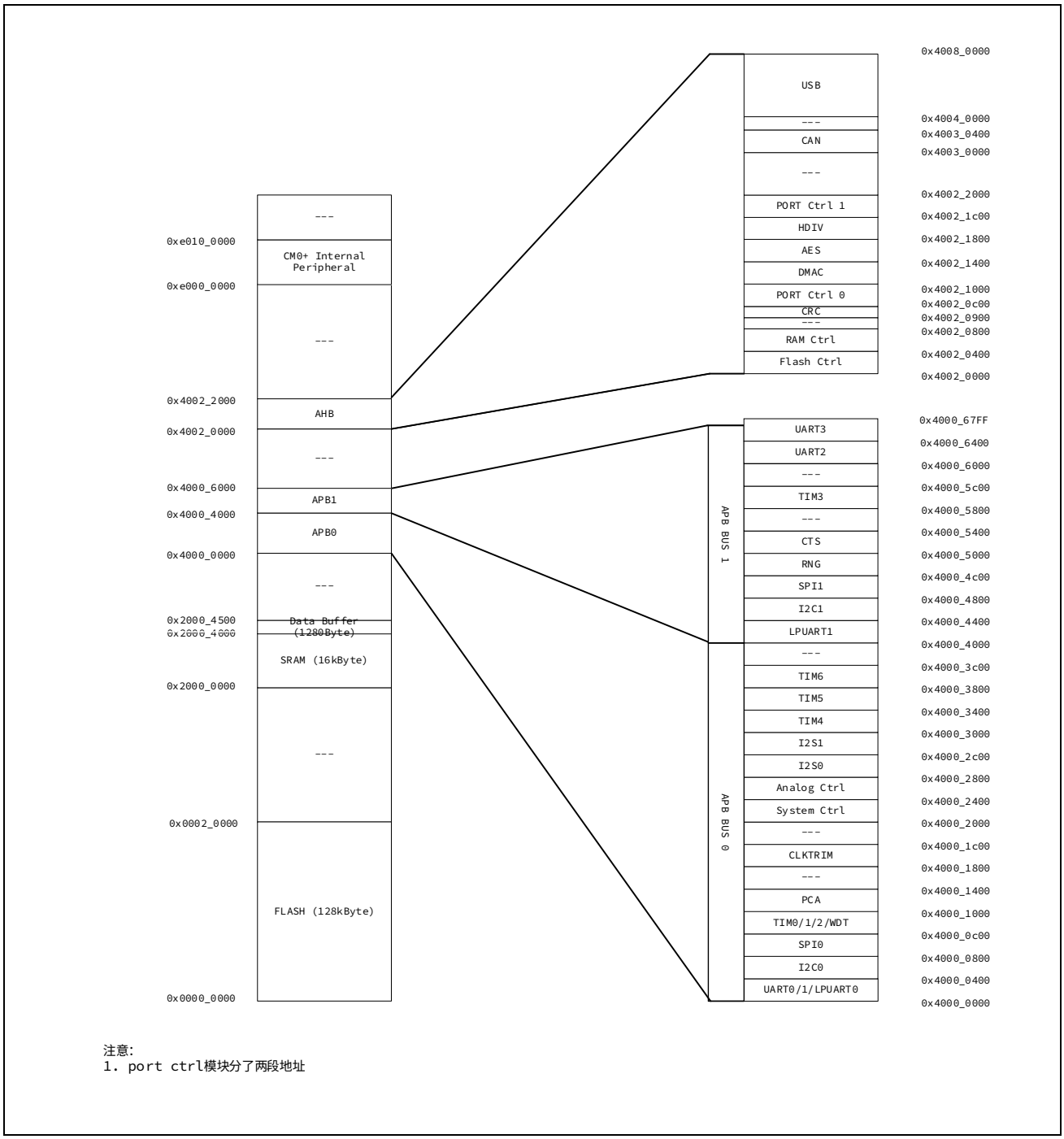


图 1-2 地址区域划分示意图

1.3 存储器和模块地址分配

表 1-1 地址划分表

Boundary Address	Size	Memory Area	Description
0x0000_0000 – 0x0000_FFFF	128KByte	FLASH	
0x0001_0000 – 0x1FFF_FFFF	-	Reserved	
0x2000_0000 – 0x2000_1FFF	16KByte	SRAM	
0x2000_2000 – 0x3FFF_FFFF	-	Reserved	
0x4000_0000 – 0x4000_00FF	256Byte	UART0	
0x4000_0100 – 0x4000_01FF	256Byte	UART1	
0x4000_0200 – 0x4000_02FF	256Byte	LPUART0	
0x4000_0300 – 0x4000_03FF	-	Reserved	
0x4000_0400 – 0x4000_07FF	1KByte	I2C0	
0x4000_0800 – 0x4000_0BFF	1KByte	SPI0	
0x4000_0C00 – 0x4000_0CFF	256Byte	TIM0	
0x4000_0D00 – 0x4000_0DFF	256Byte	TIM1	
0x4000_0E00 – 0x4000_0EFF	256Byte	TIM2	
0x4000_0F00 – 0x4000_0F7F	-	Reserved	
0x4000_0F80 – 0x4000_0FFF	128Byte	WDT	
0x4000_1000 – 0x4000_13FF	1KByte	PCA	
0x4000_1400 – 0x4000_17FF	-	Reserved	
0x4000_1800 – 0x4000_1BFF	1KByte	CLKTRIM	
0x4000_1C00 – 0x4000_1FFF	-	Reserved	
0x4000_2000 – 0x4000_23FF	1KByte	SYSCTRL	
0x4000_2400 – 0x4000_27FF	1KByte	ANALOGCTRL	
0x4000_2800 – 0x4000_2BFF	1KByte	I2S0	
0x4000_2C00 – 0x4000_2FFF	1KByte	I2S1	
0x4000_3000 – 0x4000_33FF	1KByte	TIM4	
0x4000_3400 – 0x4000_37FF	1KByte	TIM5	
0x4000_3800 – 0x4000_3BFF	1KByte	TIM6	
0x4000_3C00– 0x4000_3FFF	-	Reserved	
0x4000_4000– 0x4000_43FF	1KByte	LPUART1	
0x4000_4400– 0x4000_47FF	1KByte	I2C1	
0x4000_4800– 0x4000_4BFF	1KByte	SPI1	
0x4000_4C00– 0x4000_4FFF	1KByte	TRNG	
0x4000_5000– 0x4000_53FF	1KByte	CTS	
0x4000_5400– 0x4000_57FF	-	Reserved	
0x4000_5800– 0x4000_5BFF	1KByte	TIM3	
0x4000_5C00– 0x4000_5FFF	-	Reserved	

Boundary Address	Size	Memory Area	Description
0x4000_6000 – 0x4000_63FF	1KByte	UART2	
0x4000_6400 – 0x4000_67FF	1KByte	UART3	
0x4000_6800 - 0x4001_FFFF	-	Reserved	
0x4002_0000 - 0x4002_03FF	1KByte	FLASH CTRL	
0x4002_0400 - 0x4002_07FF	1KByte	RAM CTRL	
0x4002_0800 - 0x4002_08FF	256Byte	Reserved	
0x4002_0900 - 0x4002_0BFF	768Byte	CRC	
0x4002_0C00 - 0x4002_0FFF	1KByte	PORT CTRL	
0x4002_1000 - 0x4002_13FF	1KByte	DMAC	
0x4002_1400 - 0x4002_17FF	1KByte	AES	
0x4002_1800 - 0x4002_1BFF	1KByte	HDIV	
0x4002_1C00 - 0x4002_1FFF	1KByte	PORT CTRL1	
0x4003_0000 - 0x4003_03FF	1KByte	CAN	
0x4004_0000 - 0x4007_FFFF	256K Byte	USB	

2 工作模式

本产品的电源管理模块负责管理本产品各种工作模式之间的切换，以及控制各工作模式下的各功能模块的工作状态。本产品的工作电压（VCC）为 1.8V ~ 5.5V。

本产品有如下几个工作模式：

1) 运行模式：CPU 运行，周边功能模块运行。
2) 休眠模式：CPU 停止运行，周边功能模块运行。
3) 深度休眠模式：CPU 停止运行，高速时钟停止运行。

从运行模式，通过执行软件程序，可进入其他低功耗模式。从其他各种低功耗模式，通过中断触发，可回到运行模式。

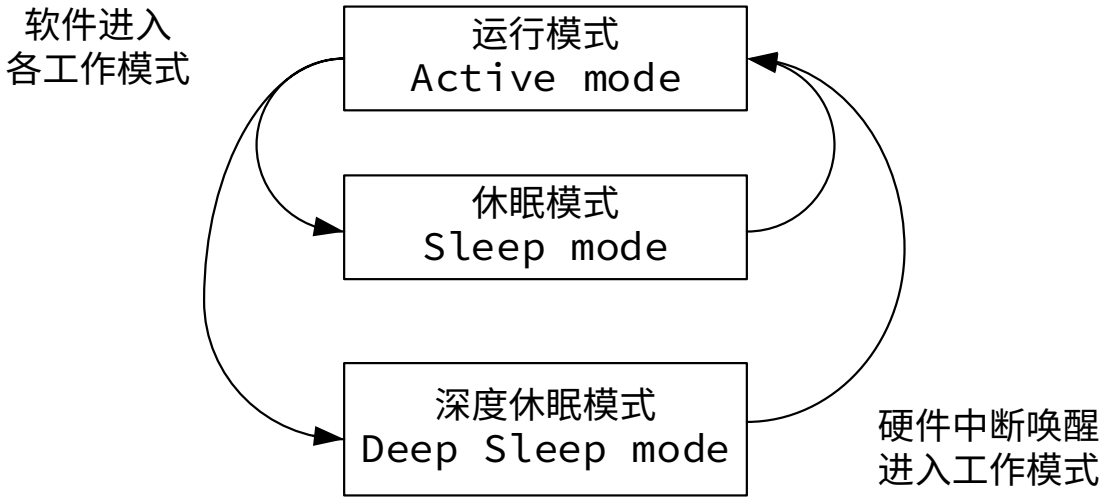


图 2-1 控制模式框图

在各模式下，CPU 可响应所有的中断类型：

中断向量编号	中断来源	运行模式	休眠模式	深度休眠模式
[0]	GPIO_PA	✓	✓	✓
[1]	GPIO_PB	✓	✓	✓
[2]	GPIO_PC/GPIO_PE	✓	✓	✓
[3]	GPIO_PD/GPIO_PF	✓	✓	✓
[4]	DMAC	✓	✓	
[5]	TIM3	✓	✓	
[6]	UART0	✓	✓	
[7]	UART1	✓	✓	
[8]	LPUART0	✓	✓	✓
[9]	LPUART1	✓	✓	✓
[10]	SPI0/I2S0	✓	✓	

中断向量编号	中断来源	运行模式	休眠模式	深度休眠模式
[11]	SPI1/I2S1	✓	✓	
[12]	I2C0	✓	✓	
[13]	I2C1	✓	✓	
[14]	TIM0	✓	✓	
[15]	TIM1	✓	✓	
[16]	TIM2	✓	✓	
[17]	-			
[18]	TIM4	✓	✓	
[19]	TIM5	✓	✓	
[20]	TIM6	✓	✓	
[21]	PCA	✓	✓	
[22]	WDT	✓	✓	✓
[23]	-			
[24]	ADC/DAC	✓	✓	
[25]	-			
[26]	VC0/VC1/VC2/LVD	✓	✓	✓
[27]	USB	✓	✓	
[28]	CAN	✓	✓	
[29]	-			
[30]	FLASH/RAM	✓	✓	
[31]	CLKTRIM/CTS	✓	✓	✓

在各模式下，本产品可响应所有的复位类型：

	复位源	运行模式	休眠模式	深度休眠模式
[0]	上电掉电复位 POR	✓	✓	✓
[1]	外部 Reset Pin 复位	✓	✓	✓
[2]	LVD 复位	✓	✓	✓
[3]	WDT 复位	✓	✓	✓
[4]	PCA 复位	✓	✓	
[5]	Cortex-M0+ LOCKUP 硬件复位	✓		
[6]	Cortex-M0+ SYSRESETREQ 软件复位	✓		

2.1 运行模式

本产品运行模式（Active Mode）：

在系统在电源上电复位后，或从各低功耗唤醒后，微控制器 MCU 处于运行状态。当 CPU 不需继续运行时，可以利用多种低功耗模式来节能，例如等待某个外部事件时。用户需要根据最低能耗、最快速启动时间、可用的唤醒源等条件，选定一个最佳的低功耗模式。

表 2-1 运行模式下可运行模块图

运行模式(Active Mode)		
Cortex-M0+	SWD	XTH
FLASH	CAN	RCH
RAM	USB	PLL
DMAC	UART0-3	ADC
TIM0-3	SPI0-1	DAC
TIM4-6	I2C0-1	TRNG
PCA	I2S0-1	OPA0-4
HDIV	CRC	LVD
AES	XTL	RESET
LPUART0-1	RCL	POR/BOR
GPIO	CLKTRIM	VC0-2
		WDT
		CTS

几种降低运行模式下芯片功耗的方法：

1) 在运行模式下，通过对预分频寄存器（SYSCTRL0.HCLK_PRS, SYSCTRL0. PCLK_PRS）进行编程，可以降低任意一个系统时钟(HCLK，PCLK)的速度。进入休眠模式前，也可以利用预分频器来降低外设的时钟。
2) 在运行模式下，关闭不使用外设的时钟（PERIx_CLKx）来减少功耗。
3) 在运行模式下，关闭不使用外设的时钟（PERIx_CLKx）来减少功耗，并让系统进入休眠模式下更多地减少功耗，并在执行 WFI 指令前关闭不使用外设的时钟（PERIx_CLKEN.x）。
4) 使用低功耗模式代替休眠模式，因为本产品的唤醒时间极短（~4us），亦可满足系统的实时响应的需求。

2.2 休眠模式

本产品休眠模式（Sleep Mode）

使用 WFI 指令可以进入休眠模式，休眠模式下，CPU 停止运行，但时钟模块、系统时钟、NVIC 中断处理以及周边的功能模块仍都可以工作。

系统进入休眠状态，不会改变端口状态，在进入休眠前根据需要更改 IO 的状态为休眠下的状态。

- 如何进入休眠模式：

  通过执行 WFI 指令进入休眠状态。根据 Cortex-M0+系统控制寄存器中的 SLEEPONEXIT 位的值，有两种选项可用于选择休眠模式进入机制：

  SLEEP-NOW：如果 SLEEPONEXIT 位被清除，当 WFI 或 WFE 被执行时，微控制器立即进入休眠模式。

  SLEEP-ON-EXIT：如果 SLEEPONEXIT 位被置位，系统从最低优先级的中断处理程序中退出时，微控制器就立即进入休眠模式。

- 如何退出休眠模式：

  如果执行 WFI 指令进入休眠模式，任意一个高优先级嵌套向量中断控制器响应的外设中断都能将系统从休眠模式唤醒。

使用注意：

1) SLEEP-ON-EXIT 该位置 1，执行完中断自动进入 sleep，程序不需要写 __wfi();
2) SLEEP-ON-EXIT 该位清 0，main()执行 __wfi() 后进入 sleeping，中断触发且执行完中断程序返回 main() 后，执行 WFI 指令后进入 sleeping。等待后续中断触发。
3) SLEEP-ON-EXIT 位不影响 __wfi() 指令的执行。SLEEP-ON-EXIT =0：main() 执行 wfi() 后进入 sleeping，中断触发且执行完中断程序返回 main() 后，继续往下执行；
4) 若在中断中进入 sleep，只有优先级高于此中断的中断才能唤醒，先执行高优先级，再执行低优先级；优先级低于或等于此中断的中断不能唤醒。

表 2-2 休眠模式下可运行模块图

休眠模式(Sleep Mode)		
Cortex-M0+	SWD	XTH
FLASH	CAN	RCH
RAM	USB	PLL
DMAC	UART0-3	ADC
TIM0-3	SPI0-1	DAC
TIM4-6	I2C0-1	TRNG
PCA	I2S0-1	OPA0-4
HDIV	CRC	LVD
AES	XTL	RESET

休眠模式(Sleep Mode)		
LPUART0-1	RCL	POR/BOR
GPIO	CLKTRIM	VC0-2
		WDT
		CTS

注：灰色虚框内的模块在当前状态下不工作。

2.3 深度休眠模式

本产品深度休眠模式 （Deep Sleep Mode）

使用 SLEEPDEEP 配合 WFI 指令可以进入深度休眠模式，在深度休眠模式下，CPU 停止运行，高速时钟关闭，低速时钟可配置是否运行，部分低功耗的周边模块可配置是否允许，NVIC 中断处理仍可以工作。

- 系统从高速时钟进入深度休眠模式，高速时钟自动关闭，低速时钟保持进入深度休眠前的状态。
- 系统从低速时钟进入深度休眠模式，由于低速时钟不会自动关闭，保持运行，进入休眠模式。只有ARM Cortex-M0+不运行，其他模块都运行。
- 系统时钟切换时，所有时钟都不会自动关闭，需要根据功耗及系统需求软件关闭打开相应的时钟。
- 系统进入深度休眠状态，不会改变端口状态，在进入休眠前根据需要更改 IO 的状态为休眠下的状态。

如何进入深度休眠模式：

首先设置 Cortex-M0+ 系统控制寄存器中的 SLEEPDEEP 位，通过执行 WFI 指令进入休眠状态。根据 Cortex™-M0+ 系统控制寄存器中的 SLEEPONEXIT 位的值，有两种选项可用于选择深度休眠模式进入机制：

SLEEP-NOW：如果 SLEEPONEXIT 位被清除，当 WFI 或 WFE 被执行时，微控制器立即进入休眠模式。

SLEEP-ON-EXIT：如果 SLEEPONEXIT 位被置位，系统从最低优先级的中断处理程序中退出时，微控制器就立即进入休眠模式。

如何退出深度休眠模式：

如果执行 WFI 指令进入休眠模式，任意一个被嵌套向量中断控制器响应的外设中断（Deep Sleep 下可运行的 周边模块中断）都能将系统从休眠模式唤醒。

唤醒设置参考 3.4 中断唤醒控制 WIC。

表 2-3 深度休眠模式下可运行模块图

深度休眠模式(DeepSleep Mode)		
Cortex-M0+	SWD	XTH
FLASH	CAN	RCH
RAM	USB	PLL
DMAC	UART0-3	ADC
TIM0-3	SPI0-1	DAC
TIM4-6	I2C0-1	TRNG
PCA	I2S0-1	OPA0-4
HDIV	CRC	LVD
AES	XTL	RESET
LPUART0-1	RCL	POR/BOR
GPIO	CLKTRIM	VC0-2
		WDT
		CTS

注：灰色虚框内的模块在当前状态下不工作。

系统控制寄存器(Cortex-M0+ 内核系统控制寄存器)

地址: 0xE000ED10

复位值：0x0000 0000

位	标记	功能描述	读写
31:5	RESERVED	保留	
4	SEVONPEND	设置为1时，每次新的中断挂起都会产生一个事件，如果使用了WFE 休眠，它可用于唤醒处理器	RW
3	RESERVED	保留	
2	SLEEPDEEP	设为1时，执行 WFI 进入深度休眠，本产品进入 Deep sleep 模式 设置为0时，执行 WFI 进入休眠，本产品进入 sleep/Idle 模式	RW
1	SLEEPONEXIT	设为1时，当退出异常处理并返回程序线程时，处理器自动进入休眠模式(WFI) 设置为0时，该特性就会被自动禁止	RW
0	RESERVED	保留	

进入深度休眠后，唤醒后系统时钟有两种选择，默认使用进入深度休眠的时钟，配置寄存器 SYSCTRL0.wakeup_byRCH 为 1 后不管进入深度休眠前是什么时钟，唤醒后都使用内部高速时钟 RCH。如果使用外部晶体振荡这样设置可以加速唤醒系统。

3 系统控制器（SYSCTRL）

3.1 系统时钟介绍

时钟控制模块主要控制系统时钟以及外设时钟，可以配置不同的时钟源作为系统时钟、可以配置不同的系统时钟分频、可以启动或禁用外设时钟。为了确保振荡器精度，内部时钟均具有校准功能。

本产品支持以下五个不同的时钟源作为系统时钟：

- 内部高速 RC 时钟 RCH（输出频率为 4~24MHz）
- 内部低速 RC 时钟 RCL（38.4K 与 32.8K 可配置）
- 外部高速晶振时钟 XTH
- 外部低速晶振时钟 XTL
- 锁相环时钟 PLL
  - 注 1：切换系统时钟的时钟源时，请严格按照操作步骤进行切换，详见 5.2 章节。
  - 注 2：XTL 可以不接晶振，直接从 PC14 管脚输入 32.768kHz 的时钟信号。XTH 可以不接晶振，直接从 PF00 管脚输入 4~32MHz 的时钟信号。

本产品还包含以下三个辅助时钟：

- 内部低速 10K 时钟，仅供看门狗和 CLKTRIM 模块使用。
- 内部 150K 时钟，仅供 LVD 和 VC 模块使用。
- 内部 48M 时钟，仅供 USB 模块使用，可以使用 CTS 模块自动校准。

下图为本产品的时钟架构。

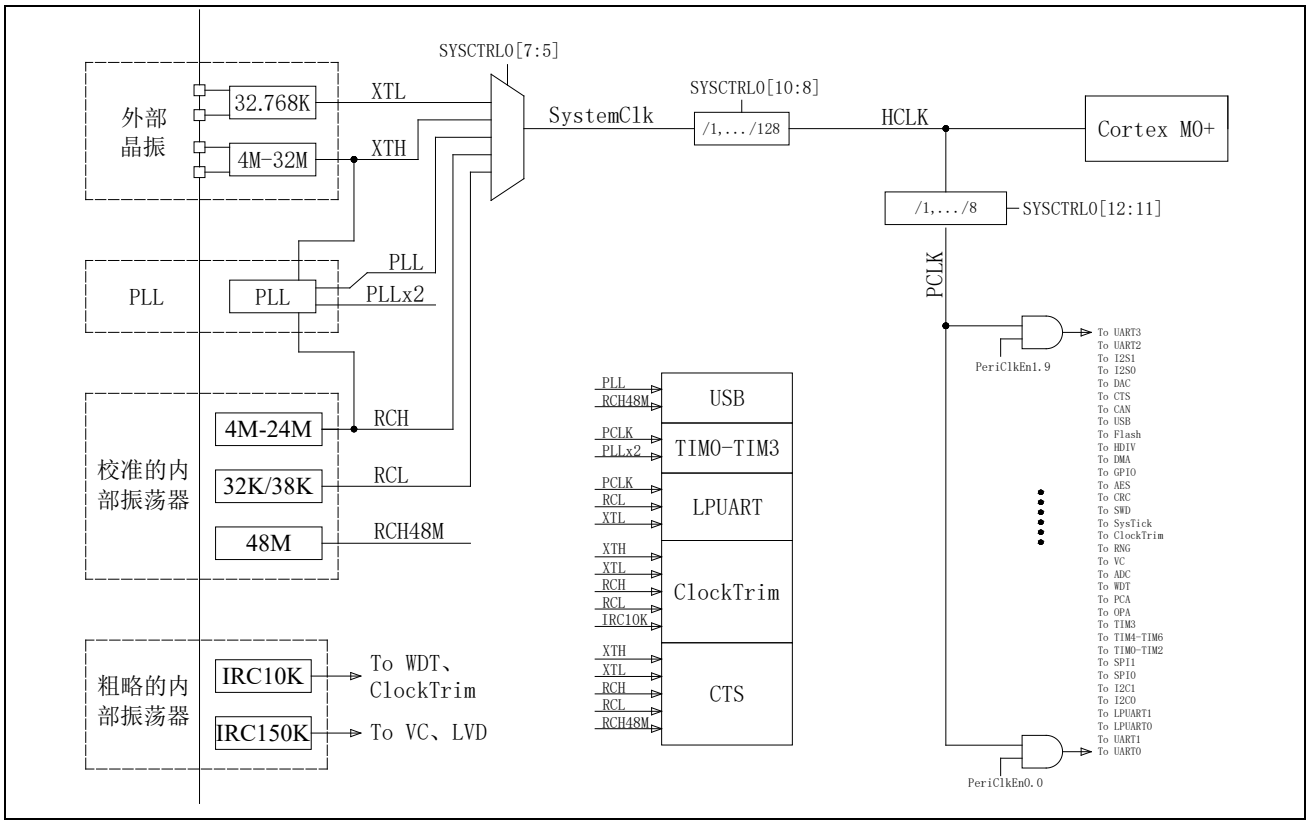


图 3-1 时钟控制模块框图

3.1.1 内部高速 RC 时钟 RCH

芯片上电或复位后的默认时钟源为频率为 4MHz 的内部高速时钟，频率为 4MHz 的内部高速时钟。当系统进入 Deep Sleep，此高速时钟会自动关闭。

更改寄存器 RCH_CR[10:0]的数值即可调整 RCH 的输出频率。寄存器数值每增加 1 则 RCH 的输出频率增加约 0.2%，总调整范围为 4~24MHz。

出厂时已预调好的 5 个频率 4MHz、8MHz、16MHz、22.12MHz、24MHz；如需其它频率请手动调整该寄存器的数值。

更改 RCH 输出频率需要按照特定的更改时序，详见系统时钟切换章节。

内部高速时钟从启动到稳定仅需 4us。为了在深度休眠模式下能快速响应中断，建议进入深度休眠模式前将系统时钟切换为 RCH。

3.1.2 内部低速 RC 时钟 RCL

内部低速时钟可以通过寄存器 RCL_CR[9:0]来选择其输出频率，可供选择的频率为 38.4KHz、32.768KHz。当系统进入 Deep Sleep，此低速时钟不会自动关闭，超低功耗外设模块可以选择 RCL 作为其时钟。

3.1.3 外部低速晶振时钟 XTL

外部低速晶振时钟需外接一个 32.768KHz 的低功耗晶振，具有超高精度以及超低功耗。当系统进入 Deep Sleep，此低速时钟不会自动关闭。超低功耗模式下工作的外设模块可以选择 XTL 作为其时钟。

XTL 也可以不接晶振，直接从 PC14 管脚输入 32.768KHz 的时钟信号。从 PC14 输入时钟信号的方法为：配置 PC14 引脚为 GPIO 输入；设置 SYSCTRL1. EXTL_EN 为 1。

注意：

- 晶体及其匹配器件需符合数据手册电气特性中低速外部时钟 XTL 的相关要求。

3.1.4 外部高速晶振时钟 XTH

外部高速晶振时钟需外接一个 4~32MHz 的高速晶振。当系统进入 Deep Sleep，此高速时钟会自动关闭。

XTH 也可以不接晶振，直接从 PF00 管脚输入 4~32MHz 的时钟信号。从 PF00 输入时钟信号的方法为：配置 PF00 引脚为 GPIO 输入；设置 SYSCTRL1. EXTH_EN 为 1。

注意：

- 晶体及其匹配器件需符合数据手册电气特性中高速外部时钟 XTH 的相关要求。
- 为保证 XTH 工作的高可靠性，强烈建议使用外部有源晶振。

3.1.5 锁相环时钟 PLL

内建 PLL 支持 8~48MHz 的时钟输出。PLL 的参考时钟源为：RCH、XTH 晶振时钟、PF00 管脚输入时钟。

PLL 在深度休眠模式时具有约几十微安的功耗，建议在进入深度休眠模式前配置系统时钟为 RCH/XTH 并关闭 PLL 及 BGR，唤醒后重新配置系统时钟为 PLL。

3.1.6 内部高速时钟 RCH48M

内部高速时钟 RCH48M 可以用作 crystall-less USB 的时钟，使用模块 CTS 可以自动校准。不可做为系统时钟。

3.1.7 时钟启动过程

上述五种时钟源都需要启动稳定时间，下图以外部 XTH 为例说明时钟的启动稳定过程。

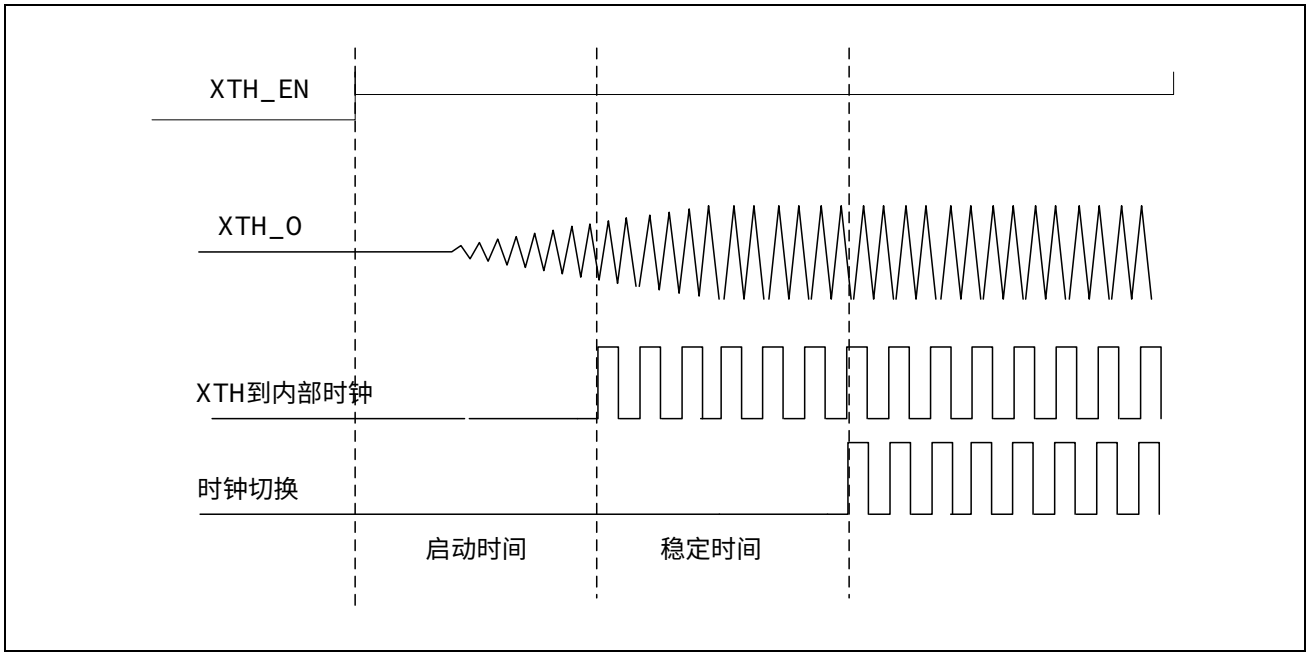


图 3-2 晶振时钟启动示意图

3.2 系统时钟切换

时钟源的切换是由寄存器 SYSCTRL0[7:0] 来控制。系统时钟的时钟源可通过 SYSCTRL0[7:5]在 RCH、RCL、XTH、XTL、PLL 之间进行切换。切换时 RCH、RCL、XTH、XTL 这四个时钟源中的任意两者可以相互切换；PLL 只可与 RCH、XTH 这两个时钟进行切换。时钟切换操作必须按照下文所描述的 7 种时钟切换流程进行，否则可能出现异常。

时钟切换时需要同步配置 FLASH_CR. WAIT。时钟频率不大于 24MHz 则应设置 FLASH_CR.WAIT 为 0；时钟频率大于 24MHz 则应设置 FLASH_CR. WAIT 为 1；时钟频率大于 48MHz 则应设置 FLASH_CR. WAIT 为 2。

注：设置 *FLASH_CR. WAIT* 的值，需要先向 *FLASH_BYPASS* 寄存器依次写入 *0x5A5A*、*0xA5A5* 然后再对 *FLASH_CR. WAIT* 进行赋值，详见 *FLASH* 控制器章节。

3.2.1 标准的时钟切换流程

操作流程如下：

Step1：如新时钟源需要外部引脚，则将该引脚设置为适当的模式。

注：接外部晶振时需要模拟引脚；接外部时钟输入时需要 *GPIO* 输入并使能外部时钟输入。

Step2：配置新时钟源的的振荡参数。

Step3：使能新时钟源的振荡器。

Step4：根据当前时钟源和新时钟源两者中较高的频率，按 Flash 控制器章节流程配置 FLASH_CR. WAIT。

Step5：等待新时钟源输出稳定的频率。

Step6：配置 SYSCTRL0. CLKSW，选择系统时钟的来源为新时钟源。

Step7：根据新时钟源的频率，按 Flash 控制器章节流程配置 FLASH_CR. WAIT。

Step8：关闭不再使用的时钟源。

3.2.2 RCH 不同振荡频率间切换流程

RCH 不同振荡频率间切换具有两种方案。

方案 1：

Step1：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step2：设置 SYSCTRL0.HCLK_PRS 为 0x7。

Step3：向上或向下逐级调整 RCH 的输出频率，4M -> 8M -> 16M -> 24M/22.12M 或 24M/22.12M -> 16M -> 8M -> 4M。

Step4：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step5：设置 SYSCTRL0.HCLK_PRS 为 0x0。

从 4M 切换到 24M 的示例代码如下所示：

```
M0P_SystemCtrl->SYSCTRL2 = 0X5A5A;
M0P_SystemCtrl->SYSCTRL2 = 0XA5A5;
M0P_SystemCtrl->SYSCTRL0_f.HCLK_PRS = 7;
M0P_SystemCtrl->RCH_CR =   *((uint16 *)( 0X00100C08 ) );   //4M
M0P_SystemCtrl->RCH_CR =   *((uint16 *)( 0X00100C06 ) );   //8M
M0P_SystemCtrl->RCH_CR =   *((uint16 *)( 0X00100C04 ) );   //16M
M0P_SystemCtrl->RCH_CR =   *((uint16 *)( 0X00100C00 ) );   //24M
M0P_SystemCtrl->SYSCTRL2 = 0X5A5A;
M0P_SystemCtrl->SYSCTRL2 = 0XA5A5;
M0P_SystemCtrl->SYSCTRL0_f.HCLK_PRS = 0
```

方案 2：

Step1：将系统时钟切换为 RCL，参见 5.2.5 从其它时钟切换到 RCL 示例。

Step2：将系统时钟切换为 RCH，参见 5.2.6 从其它时钟切换到 RCH 示例。

3.2.3 从其它时钟切换到 XTL 示例

操作流程如下：

Step1：设置 PCADS. 14 及 PCADS. 15 为 1，配置 PC14/PC15 引脚为模拟端口。

Step2：根据晶振特性，配置 XTL_CR[5:0]。

Step3：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step4：设置 SYSCTRL0. XTL_EN 为 1，使能晶振振荡电路。

Step5：查询等待 XTL_CR. Stable 标志变为 1，晶振输出稳定时钟。

Step6：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step7：设置 SYSCTRL0. CLKSW 为 3，将系统时钟切换为 XTL。

Step8：按 Flash 控制器章节流程设置 FLASH_CR. WAIT 为 0。

Step9：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step10：设置 SYSCTRL0.xxx_EN 为 0，关闭原时钟。

3.2.4 从其它时钟切换到 XTH 示例

Step1：设置 PFADS. 0 及 PFADS. 1 为 1，配置 PF00/PF01 引脚为模拟端口。

Step2：根据晶振特性，配置 XTH_CR[3:0]。

Step3：设置 XTH_CR. Startup 为 3，选择最长的晶振稳定时间。

Step4：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step5：设置 SYSCTRL0. XTH_EN 为 1，使能晶振振荡电路。

Step6：根据当前时钟和 XTH 两者中较高的频率，按 Flash 控制器章节流程配置 FLASH_CR.WAIT。

Step7：查询等待 XTH_CR. Stable 标志变为 1，软件延时 10ms 以上，晶振输出稳定时钟。

Step8：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step9：设置 SYSCTRL0. CLKSW 为 1 后，将系统时钟切换为 XTH。

Step10：根据 XTH 的频率，按 Flash 控制器章节流程配置 FLASH_CR. WAIT。

Step11：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step12：设置 SYSCTRL0.xxx_EN 为 0，关闭原时钟。

3.2.5 从其它时钟切换到 RCL 示例

操作流程如下：

Step1：配置 RCL_CR.TRIM 及 RCL_CR. Startup。

Step2：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3：设置 SYSCTRL0. RCL_EN 为 1，使能 RCL 振荡电路。

Step4：查询等待 RCL_CR. Stable 标志变为 1，RCL 输出稳定时钟。

Step5：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6：设置 SYSCTRL0. CLKSW 为 2，将系统时钟切换为 RCL。

Step7：按 Flash 控制器章节流程设置 FLASH_CR. WAIT 为 0。

Step8：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step9：设置 SYSCTRL0.xxx_EN 为 0，关闭原时钟。

3.2.6 从其它时钟切换到 RCH 示例

操作流程如下：

Step1：配置 RCH_CR.TRIM。

Step2：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3：设置 SYSCTRL0. RCH_EN 为 1，使能 RCH 振荡电路。

Step4：查询等待 RCH_CR. Stable 标志变为 1，RCH 输出稳定时钟。

Step5：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6：设置 SYSCTRL0. CLKSW 为 0，将系统时钟切换为 RCH。

Step7：按 Flash 控制器章节流程设置 FLASH_CR. WAIT 为 0。

Step8：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step9：设置 SYSCTRL0.xxx_EN 为 0，关闭原时钟。

3.2.7 PLL 与 RCH 相互切换示例，参考时钟为 RCH

从 RCH 切换到 PLL 操作流程如下：

Step1：设置 BGR_CR 为 0x01 并延时 20us，等待 BGR 启动并稳定。

Step2：设置 PLL_CR.REFSEL 为 3，选择 PLL 时钟源为 RCH。

Step3：根据 RCH 频率，配置 PLL_CR.FRSEL。

Step4：根据 PLL 输出频率，配置 PLL_CR.DIVN 及 PLL_CR.FOSC。

Step5：设置 PLL_CR. Startup 为 7，选择最长的 PLL 稳定时间。

Step6：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step7：设置 SYSCTRL0.PLL_EN 为 1，使能 PLL 振荡电路。

Step8：根据 PLL 输出频率，按 Flash 控制器章节流程配置 FLASH_CR. WAIT。

Step9：查询等待 PLL_CR. Stable 标志变为 1，PLL 输出稳定时钟。

Step10：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step11：设置 SYSCTRL0. CLKSW 为 4，将系统时钟切换为 PLL。

警告：切换过程中不可以更改 RCH_CR.TRIM。

从 PLL 切换到 RCH 操作流程如下：

Step1：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step2：设置 SYSCTRL0. CLKSW 为 0，将系统时钟切换为 RCH。

Step3：按 Flash 控制器章节流程设置 FLASH_CR. WAIT 为 0。

Step4：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step5：设置 SYSCTRL0.PLL_EN 为 0，关闭 PLL。

Step6：设置 BGR_CR 为 0x01 为 0，关闭 BGR。

警告：切换过程中不可以更改 RCH_CR.TRIM。

3.2.8 PLL 与 XTH 相互切换示例，参考时钟为 XTH

从 RCH 切换到 PLL 操作流程如下：

Step1：设置 BGR_CR 为 0x01 并延时 20us，等待 BGR 启动并稳定。

Step2：设置 PLL_CR.REFSEL 为 0，选择 PLL 时钟源为 XTH。

Step3：根据 XTH 频率，配置 PLL_CR.FRSEL。

Step4：根据 PLL 输出频率，配置 PLL_CR.DIVN 及 PLL_CR.FOSC。

Step5：设置 PLL_CR. Startup 为 7，选择最长的 PLL 稳定时间。

Step6：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step7：设置 SYSCTRL0.PLL_EN 为 1，使能 PLL 振荡电路。

Step8：根据 PLL 输出频率，按 Flash 控制器章节流程配置 FLASH_CR. WAIT。

Step9：查询等待 PLL_CR. Stable 标志变为 1，PLL 输出稳定时钟。

Step10：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step11：设置 SYSCTRL0. CLKSW 为 4，将系统时钟切换为 PLL。

从 PLL 切换到 XTH 操作流程如下：

Step1：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step2：设置 SYSCTRL0. CLKSW 为 1，将系统时钟切换为 XTH。

Step3：根据 XTH 的频率，按 Flash 控制器章节流程配置 FLASH_CR. WAIT。

Step4：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step5：设置 SYSCTRL0.PLL_EN 为 0，关闭 PLL。

Step6：设置 BGR_CR 为 0x01 为 0，关闭 BGR。

下图为时钟切换时序图：

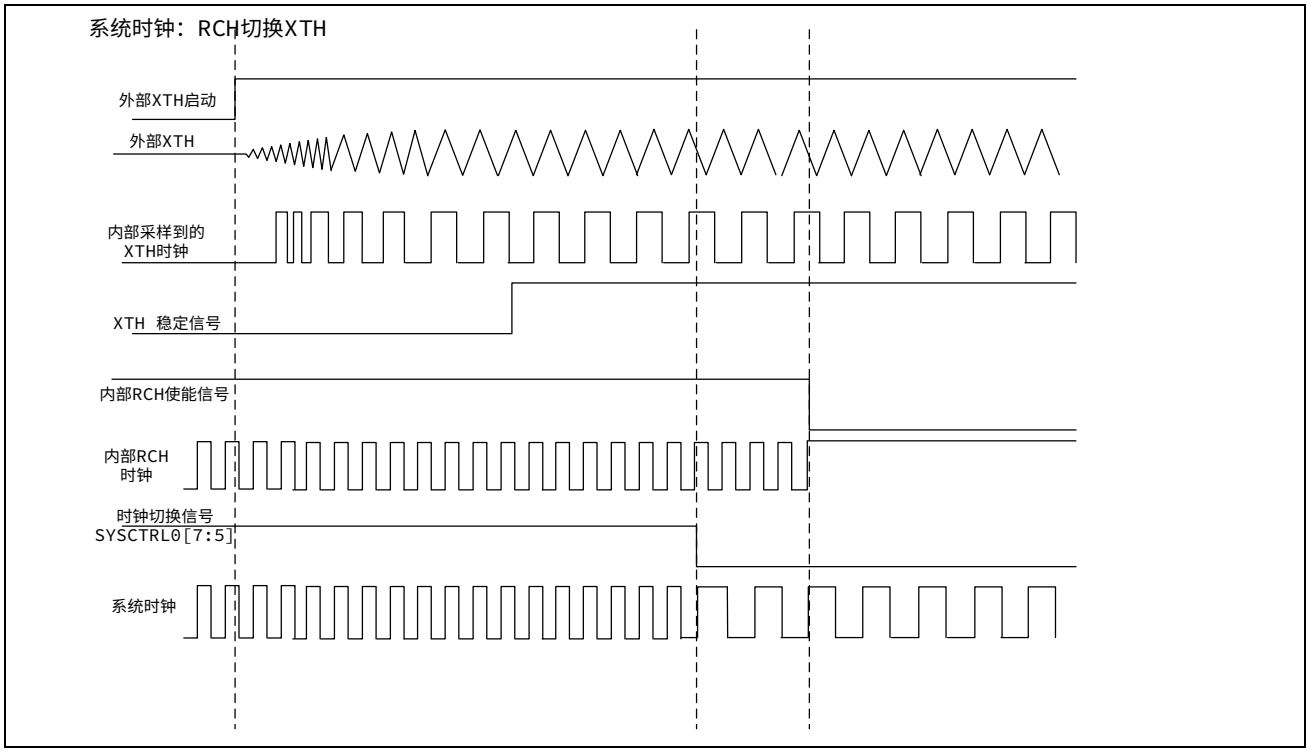


图 3-3 时钟切换示意图

3.3 时钟校准模块

本产品内嵌时钟校准电路，如下图所示，系统时钟的 5 个源均可以相互校准，当选择好参考时钟以及被校准时钟后，设置寄存器 REFCNT 值，置位 cali.start 启动时钟校准电路，此时两个 32 位计数器（递增、递减）同时工作，当递减计数器等于 0 时，cali.finish 被置位，表明校准结束，此时软件可以读取 CALCNT 值，这样就很容易得到参考时钟与被校准时钟之间的频率关系。

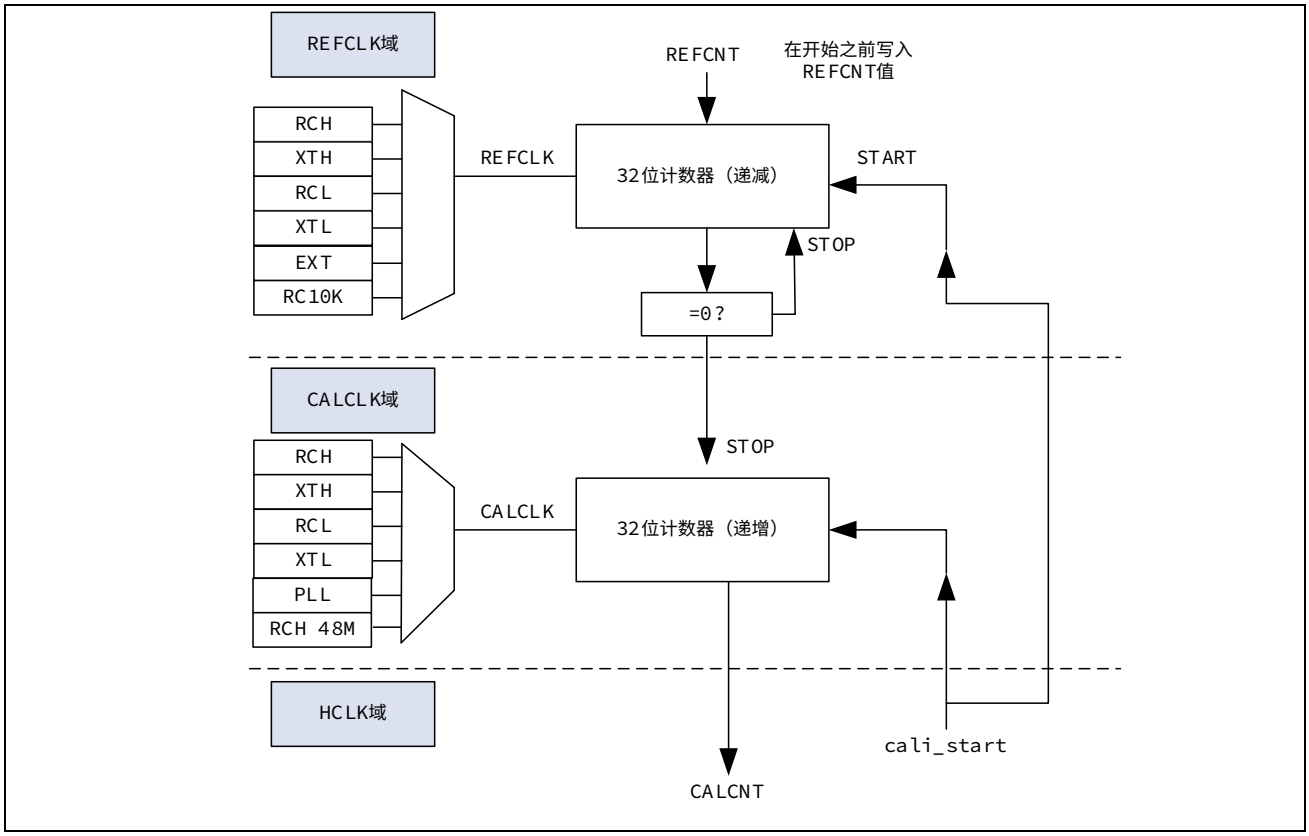


图 3-4 时钟校准原理图

3.4 中断唤醒控制

当处理器执行 WFI 指令进入休眠状态时，会停止执行指令。在休眠状态下发生了中断请求（更高优先级）且需要处理时，处理器就会被唤醒。

休眠状态下的处理器收到中断请求时的行为如下表所示：

PRIMASK 状态	WFI 行为	唤醒	ISR 执行
0	IRQ 优先级>当前等级	Y	Y
0	IRQ 优先级≦当前等级	N	N
1	IRQ 优先级>当前等级	Y	N
1	IRQ 优先级≦当前等级	N	N

3.4.1 从深度休眠模式唤醒后执行中断服务程序的方法

1. 使能需要唤醒处理器的模块所对应的 NVIC
2. 使能需要唤醒处理器的模块所对应的中断
3. 设置 SCB->SCR.SLEEPDEEP 为 1
4. 执行 WFI 指令以进入深度休眠模式
5. 系统进入深度休眠模式等待中断唤醒，唤醒后执行中断服务程序

例程：

```
SCB->SCR |= 0x00000004u;
while(1)
{
__asm("WFI");
}
```

3.4.2 从深度休眠模式唤醒后不执行中断服务程序的方法

1. 使能需要唤醒处理器的模块所对应的 *NVIC*
2. 使能需要唤醒处理器的模块所对应的中断
3. 设置 *PRIMASK* 为 *1*
4. 设置 *SCB->SCR.SLEEPDEEP* 为 *1*
5. 执行 *WFI* 指令以进入深度休眠模式
6. 系统进入深度休眠模式等待中断唤醒，唤醒后执行下一条指令
7. 清除中断标志，清除中断挂起状态
8. 执行用户定义的操作

例程：

```
__asm("CPSID I");     //Set PRIMASK
SCB->SCR |= 0x00000004u;
while(1)
{
                __asm("WFI");
                MOP_TIMx->ICLR = 0x00;          //Clear Int Flag
                NVIC_ClearPendingIRQ(TIMERLP_IRQn ); //Clear Pending Flag
        ...     //执行用户定义的操作
}
```

3.4.3 使用退出休眠特性

退出休眠（sleep-on-exit）非常适合中断驱动的应用程序。当该特性使能时，只要完成异常处理并且返回到了线程模式，处理器就会进入休眠模式。利用退出休眠特性，处理器可以尽可能多的处于休眠模式。

Cortex-M0 利用退出休眠特性进入休眠，这种情况同执行完异常退出后立即执行 WFI 的效果差不多。不过，为了下次进入异常时，不用再进行压栈操作，处理器不会执行出栈的过程。

1. 使能需要唤醒处理器的模块所对应的 *NVIC*
2. 使能需要唤醒处理器的模块所对应的中断
3. 设置 *SCB->SCR.SLEEPDEEP* 为 *1*
4. 设置 *SCB->SCR.SLEEPONEXIT* 为 *1*
5. 执行 *WFI* 指令以进入深度休眠模式
6. 系统进入深度休眠模式等待中断唤醒，唤醒后执行中断服务子程序
7. 退出中断服务时自动进入休眠模式

例程：

```
SCB->SCR |= 0x00000004u;
SCB->SCR |= 0x00000002u;
while(1)
{
__asm("WFI");
}
```

3.5 寄存器

基地址 0x40002000

表 3-1 系统控制寄存器

寄存器	偏移地址	描述
SYSCTRL0	0x000	系统控制寄存器0
SYSCTRL1	0x004	系统控制寄存器1
SYSCTRL2	0x008	系统控制寄存器2
RCH_CR	0x00C	RCH 控制寄存器
XTH_CR	0x010	XTH 控制寄存器
RCL_CR	0x014	RCL 控制寄存器
XTL_CR	0x018	XTL 控制寄存器
PLL_CR	0x03C	PLL 控制寄存器
PERI_CLKEN0	0x020	外围模块时钟控制寄存器
PERI_CLKEN1	0x024	外围模块时钟控制寄存器
OVCK_CR	0x054	定时器时钟倍频控制
RC48M_CR	0x058	RC48M 控制寄存器

3.5.1 系统控制寄存器 0（SYSCTRL0）

偏移地址：0x000

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WakeupByRCH	Reserved		PclkPrs		HclkPrs			CLKSW			PLLEN	XTL EN	RCL EN	XTHEN	RCHEN
RW			RW		RW			RW			RW	RW	RW	RW	RW

位	标记	功能描述
31:16	Reserved	保留
15	WakeupByRCH	1：从 Deep Sleep 唤醒后，system clock来源为RCH，原时钟继续使能。 0：从 Deep Sleep 唤醒后，不改变system clock来源。
14:13	Reserved	保留
12:11	PclkPrs	PCLK时钟来源选择 00： HCLK 01： HCLK/2 10： HCLK/4 11： HCLK/8
10:8	HclkPrs	HCLK时钟来源选择 000： SystemClk 001： SystemClk/2 010： SystemClk/4 011： SystemClk/8 100： SystemClk/16 101： SystemClk/32 110： SystemClk/64 111： SystemClk/128
7:5	CLKSW	SystemClk时钟来源选择 000：内部高速时钟 RCH 001：外部高速晶振 XTH 010：内部低速时钟 RCL 011：外部低速晶振 XTL 100：内部 PLL
4	PLLEN	PLL使能控制 0： 关闭 1： 使能

		注：需要 BGR 使能并稳定后才可以使能 PLL。
3	XTLEN	外部低速晶振 XTL 使能控制 0： 关闭 1： 使能 注：需要将PC14、PC15设置成模拟端口。
2	RCLEN	内部低速时钟 RCL 使能控制 0： 关闭 1： 使能
1	XTHEN	外部高速晶振 XTH 使能控制 0： 关闭 1： 使能 注：当系统进入DeepSleep，此高速时钟会自动关闭
0	RCHEN	内部高速时钟 RCH 使能信号。 0： 关闭 1： 使能 注：当系统进入DeepSleep，此高速时钟会自动关闭。

注意：

- 每次改写 SYSCTRL0,SYSCTRL1 的值,均需要先对 SYSCTRL2 依次写入 0x5A5、0xA5A5。这样的步骤可有效防止对 SYSCTRL0，SYSCTRL1 寄存器的误操作。

3.5.2 系统控制寄存器 1（SYSCTRL1）

偏移地址：0x004

复位值: 0x00000008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		USB48MSEL	Reserved				SWD_USE_IO	Res.	LOCKUP_EN	Reserved		XTL_ALWAYS_ON	EXTL_EN	EXTH_EN	Res.
		RW					RW		RW			RW	RW	RW	

位	标记	功能描述
31:14	Reserved	保留位
13	USB48MSEL	USB时钟选择 0: RCH48M; 1: PLL倍频时钟
12:9	Res.	保留位
8	SWD_USE_IO	SWD 端口功能配置 0：SWD 端口 1：GPIO 端口
7	Res.	保留位
6	LOCKUP_EN	Cortex-M0+ LockUp 功能配置 0：关闭 1：使能 注：使能后，CPU读到无效指令时会复位MCU，可增强系统可靠性。
5:4	Res.	保留位
3	XTL_ALWAYS_ON	XTL 高级使能控制 1：SYSCTRL0.XTL_EN 只可置位。 0：SYSCTRL0.XTL_EN 可置位可清零。
2	EXTL_EN	外部 XTL 时钟输入控制 1：XTL 输出时钟从 PC14 输入。 0：XTL 输出时钟由晶振产生。 注：使用PC14 输入时钟时，需设置SYSCTRL0.XTL_EN为1；当使用PC14外部输入时钟时PC15管脚禁止作为GPIO使用，PC15端口需要配置为模拟状态。
1	EXTH_EN	外部 XTH 输入控制 1：XTH 输出时钟从 PF00输入。 0：XTH 输出时钟由晶振产生。 注：使用PF00 输入时钟时，需设置SYSCTRL0.XTH_EN为1；当使用PF00外部输入时钟时

		PF01管脚禁止作为GPIO使用，PF01端口需要配置为模拟状态。
0	Reserved	保留位

注意：

- 每次改写 SYSCTRL0,SYSCTRL1 的值,均需要先对 SYSCTRL2 依次写入 0x5A5、0xA5A5。这样的步骤可有效防止对 SYSCTRL0，SYSCTRL1 寄存器的误操作。

3.5.3 系统控制寄存器 2（SYSCTRL2）

偏移地址：0x008

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSCTRL2															
WO															

位	标记	功能描述
31:16	Reserved	保留位
15:0	SYSCTRL2	寄存器 SYSCTL0，SYSCTRL1 保护系列控制寄存器，对 SYSCTRL2 先写 0x5A5A，再写 0xA5A5，启动对于寄存器 SYSCTL0，SYSCTRL1 的写操作，只要对寄存器 SYSCTL0，SYSCTRL1 写操作了，这个保护位自动回复保护状态，需要重新写入系列打开保护。

3.5.4 RCH 控制寄存器（RCH_CR）

偏移地址：0x00C

复位值: 0x00000126

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				Stable	TRIM										
				RO	RW										

位	标记	功能描述
31:12	Reserved	保留位
11	stable	RCH 时钟稳定标志位。 1：代表 RCH 已经稳定，可以被内部电路使用。 0：代表 RCH 未稳定，不可以被内部电路使用。
10:0	TRIM	时钟频率调整，更改该寄存器的数值即可调整RCH的输出频率。寄存器数值每增加1则RCH的输出频率增加约0.2%，总调整范围为4~24MHz，Flash中已保存了5组频率的校准值，将Flash内的校准值读出并写入RCH_CR.TRIM即可获得精准的频率。 24M校准值地址：0x00100C00 - 0x00100C01 22.12M校准值地址：0x00100C02 - 0x00100C03 16M校准值地址：0x00100C04 - 0x00100C05 8M校准值地址：0x00100C06 - 0x00100C07 4M校准值地址：0x00100C08 - 0x00100C09

3.5.5 XTH 控制寄存器（XTH_CR）

偏移地址：0x010

复位值: 0x00000022

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									Stable	Startup		xth_fsel		Driver	
									RO	RW		RW		RW	

位	标记	功能描述
31:7	Reserved	保留位
6	stable	外部高速时钟 XTH 稳定标志位。 1：代表 XTH 已经稳定，可以被内部电路使用。 0：代表 XTH 未稳定，不可以被内部电路使用。 注意：为了增加系统可靠性，在查询到该标志后，需要软件延时 10ms 以上，才可将系统时钟切换为 XTH。
5:4	Startup	外部高速时钟 XTH 稳定时间选择 00: 256个周期; 01: 1024个周期; 10: 4096个周期; 11: 16384个周期; 注：强烈建议将 XTH 的稳定时间设置为 11。如果 XTH 稳定时间不足，在进行时钟切换时或从深度休眠唤醒时，系统不能稳定工作。
3:2	xth_fsel	外部晶振工作频率选择 11： 24M~32M 10： 12M~24M 01： 8M~16M 00： 4M~8M
1:0	Driver	外部晶振驱动能力选择 11：最强驱动能力 10：默认驱动能力（推荐值） 01：弱驱动能力 00：最弱驱动能力 注：需要根据晶振特性、负载电容以及电路板的寄生参数选择适当的驱动能力。驱动能力越大则功耗越大；驱动能力越弱则功耗越小。

3.5.6 RCL 控制寄存器（RCL_CR）

偏移地址：0x014

复位值: 0x0000033Fh

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			Stable	Startup		TRIM									
			RO	RW		RW									

位	标记	功能描述
31:13	Reserved	保留位
12	stable	内部低速时钟 RCL 稳定标志位。 1：代表 RCL 已经稳定，可以被内部电路使用。 0：代表 RCL 未稳定，不可以被内部电路使用。
11:10	Startup	内部低速时钟 RCL 稳定时间选择 11: 256个周期; 10: 64个周期; 01: 16个周期; 00: 4个周期;
9:0	TRIM	内部低速时钟频率调整，Flash中保存了2组频率的校准值。 将Flash内的校准值读出并写入RCL_CR.TRIM即可获得精准的频率。 38.4K校准值地址： 0x00100C20 - 0x00100C21 32.768K校准值地址：0x00100C22 - 0x00100C23

3.5.7 XTL 控制寄存器（XTL_CR）

偏移地址：0x018

复位值: 0x00000021

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									Stable	Startup		Amp_sel		Driver	
									RO	RW		RW		RW	

位	标记	功能描述
31:7	Reserved	
6	Stable	外部低速晶振 XTL 稳定标志位。 1：代表 XTL 已经稳定，可以被内部电路使用。 0：代表 XTL 未稳定，不可以被内部电路使用。
5:4	Startup	外部低速晶振 XTL 稳定时间选择 00: 256 个周期; 01: 1024 个周期; 10: 4096 个周期; 11: 16384 个周期;
3:2	amp_sel	XTL晶振振荡幅度的调整。 11: 最大振幅 10: 较大振幅（推荐值） 01: 正常振幅 00: 最小振幅
1:0	Driver	XTL晶振驱动能力选择 11: 最强驱动能力 10: 较强驱动能力 01: 一般驱动能力 （推荐值） 00: 最弱驱动能力 注：需要根据晶振特性、负载电容以及电路板的寄生参数选择适当的驱动能力。驱动能力越大则功耗越大；驱动能力越弱则功耗越小。

3.5.8 PLL 控制寄存器（PLL_CR）

偏移地址：0x03C

复位值: 0x000010B0F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													Stable	Startup	
													RO	RW	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Stableopt	FRSEL		LFSEL		IBSEL		DVIN				FOSC			REFSEL	
RW	RW		RW		RW		RW				RW			RW	

位	标记	功能描述
31:19	Reserved	保留
18	Stable	PLL 稳定标志 0: PLL未稳定 1:PLL已经稳定
17:15	Startup	PLL 稳定时间选择 000: 128个PLL周期 001: 256个PLL周期 010: 512个PLL周期 011: 1024个PLL周期 100: 2048个PLL周期 101: 4096个PLL周期 110: 8192个PLL周期 111: 16384个PLL周期
14:13	FRSEL	PLL 输入频率选择，根据PLL输入时钟的频率进行如下配置 00：4M~6M 01：6M~12M 10：12M~20M 11：20M~24M
12:11	LFSEL	PLL 滤波器控制位， 请保持默认值
10:9	IBSEL	PLL 偏置电流选择， 请保持默认值
8:5	DIVN	PLL 输出时钟的倍频系数, PLL输出频率与输入频率的倍数 输出频率 = DIVN*输入频率， DIVN 允许范围0X2~0XC
4:2	FOSC	PLL 输出频率范围选择，根据PLL输出时钟频率进行如下配置 000：8M~12M

		001：12M~18M 010：18M~24M 011：24M~36M 1xx：36M~48M
1:0	REFSEL	输入时钟选择 x0：XTH晶振生成的时钟 01：XTH从管脚PF00输入的时钟。（详见3.1.4） 11：RCH时钟

3.5.9 外围模块时钟控制寄存器（PERI_CLKEN0）

复位值: 0x8080_0000

偏移地址：0x020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLASH	HDIV	DMA	GPIO	AES	CRC	SWD	TICK	Res.		Trim	Res.		TRNG	VC	ADC
RW	RW	RW	RW	RW	RW	RW	RW			RW			RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDT	PCA	OPA	Res.	TIM3	ADVTIM	Res.	BASETIM	SPI1	SPI0	I2C1	I2C0	LPUART1	LPUART0	UART1	UART0
RW	RW	RW		RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31	FLASH	FLASH 控制器模块时钟使能。关闭后 FLASH 配置寄存器不可写，FLASH 中的程序仍然可以运行。 1：使能；0：关闭
30	HDIV	HDIV 模块时钟使能。 1：使能；0：关闭
29	DMA	DMAC 模块时钟使能。 1：使能；0：关闭
28	GPIO	GPIO 模块时钟使能。 1：使能；0：关闭
27	AES	AES 模块时钟使能。 1：使能；0：关闭
26	CRC	CRC 模块时钟使能。 1：使能；0：关闭
25	SWD	SWD 模块时钟使能。 1：使能；0：关闭
24	TICK	SysTick 定时器参考时钟使能。 1：使能；0：关闭
23:22	Res.	保留位
21	TRIM	CLKTRIM 模块时钟使能。 1：使能；0：关闭
20:19	Res.	保留位
18	TRNG	TRNG 模块时钟使能。 1：使能；0：关闭
17	VC	VC、LVD，模块时钟使能。 1：使能；0：关闭

16	ADC	ADC、BGR 模块时钟使能。 1：使能；0：关闭
15	WDT	WDT 模块时钟使能。 1：使能；0：关闭
14	PCA	PCA 模块时钟使能。 1：使能；0：关闭
13	OPA	OPA 模块时钟使能。 1：使能；0：关闭
12	Res.	保留位
11	TIM3	TIM3 模块时钟使能。 1：使能；0：关闭
10	ADVTIM	TIM456 模块时钟使能。 1：使能；0：关闭
9	Res.	保留位
8	BASETIM	TIM012 模块时钟使能。 1：使能；0：关闭
7	SPI1	SPI1 模块时钟使能。 1：使能；0：关闭
6	SPI0	SPI0 模块时钟使能。 1：使能；0：关闭
5	I2C1	I2C1 模块时钟使能。 1：使能；0：关闭
4	I2C0	I2C0 模块时钟使能。 1：使能；0：关闭
3	LPUART1	LPUART1 模块时钟使能。 1：使能；0：关闭
2	LPUART0	LPUART0 模块时钟使能。 1：使能；0：关闭
1	UART1	UART1 模块时钟使能。 1：使能；0：关闭
0	UART0	UART0 模块时钟使能。 1：使能；0：关闭

3.5.10 外围模块时钟控制寄存器（PERI_CLKEN1）

复位值: 0x0000_0000

偏移地址：0x024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						UART3	UART2	RES	I2S1	I2S0	RES	DAC	CTS	CAN	USB
						R/W	R/W		R/W	R/W		R/W	R/W	R/W	R/W

位	标记	功能描述
31:10	Reseved	保留位
9	UART3	UART3 模块时钟使能。 1：使能；0：关闭
8	UART2	UART2 模块时钟使能。 1：使能；0：关闭
7	RES.	保留位
6	I2S1	I2S1 模块时钟使能。 1：使能；0：关闭
5	I2S0	I2S0 模块时钟使能。 1：使能；0：关闭
4	RES.	保留位
3	DAC	DAC 模块时钟使能。 1：使能；0：关闭
2	CTS	CTS模块时钟使能。 1：使能；0：关闭
1	CAN	CAN 模块时钟使能。 1：使能；0：关闭
0	USB	USB 模块时钟使能。 1：使能；0：关闭

3.5.11 定时器时钟倍频控制（OVCK_CR）

复位值 0x00000

偏移地址：0x054

b31	b30	b29	b28	b27	b26	b25	b24	b23	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															OVCK
															RW

位	标记	功能描述
31:1	Reserved	保留位
0	OVCK	定时器使用PLL时钟时，定时器使用时钟选择控制。 0：定时器使用系统同频时钟；1：定时器使用系统时钟2倍频 定时器切换时钟（更改OVCK时）需要关闭相应定时器的Peri_clken

3.5.12 RC48M 控制寄存器（RC48M_CR）

偏移地址：0x058

复位值: 0x0280

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			EN	Stable	Res.	TRIM									
			RW	R		R/W									

位	标记	功能描述
31:13	Reseved	保留位
12	EN	RCH48M模块使能。 1：使能；0：关闭
11	Stable	RCH48M时钟稳定标志 0：稳定；1未稳定
10	RES.	保留位
9:0	TRIM	RCH48M 校准值

4 复位控制器（RESET）

4.1 复位控制器介绍

本产品具有 7 个复位信号来源，每个复位信号都可以让 CPU 重新运行，绝大多数寄存器会被复位到复位值，程序会从复位向量处开始执行。

- 数字区域上电掉电复位 POR
- 外部 Reset PAD，低电平为复位信号
- WDT 复位
- PCA 复位
- LVD 低电压复位
- Cortex-M0+ SYSRESETREQ 软件复位
- Cortex-M0+ LOCKUP 硬件复位

每个复位源由相应的复位标志进行指示。复位标志均由硬件置位，需要用户软件清零。芯片复位时，如果查询到 Reset_flag. POR15V 或 Reset_flag. POR5V 为 1 则为**上电复位**。**上电复位**时用户程序应当将寄存器 Reset_flag 清零，则下一次复位时可通过 Reset_flag 的相关比特判断复位来源。

下图描述各区域的复位来源。

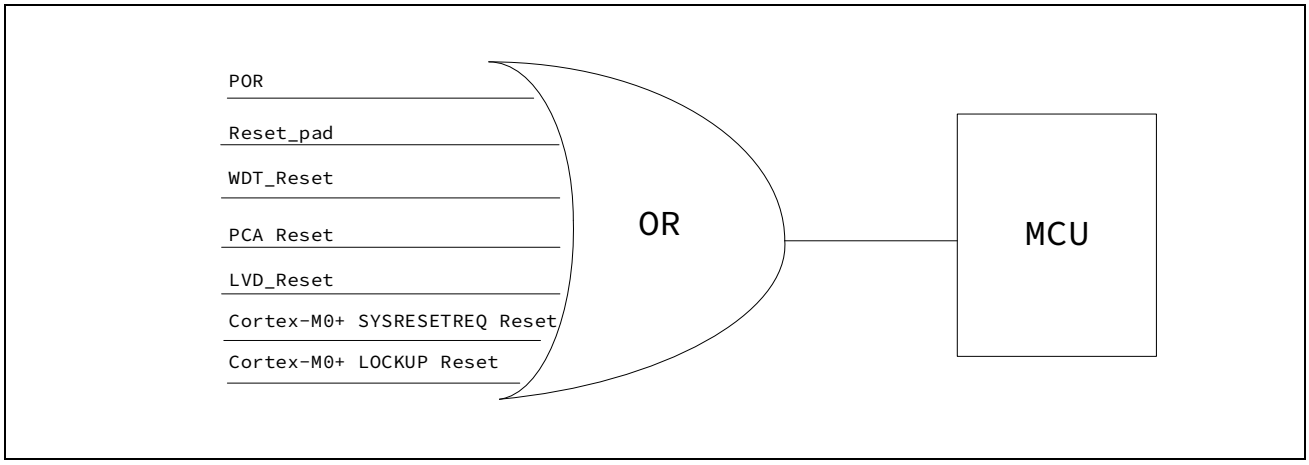


图 4-1 复位来源示意图

4.1.1 上电下电复位 POR

本产品有两个供电区域：VCC 区域、VCAP 区域。所有的模拟模块及 IO 工作于 VCC 区域；其它模块工作于 VCAP 区域。

VCC 区域上电时，当 VCC 电压低于 POR 阈值电压时（典型值为 1.65V），会产生 POR5V 信号；VCC 区域下电时，当 VCC 电压低于 BOR 阈值电压时（典型值为 1.5V），会产生 POR5V 信号。

VCAP 区域上电时，当 VCAP 电压低于 POR 阈值电压时，会产生 POR15V 信号；VCAP 区域下电时，当 VCAP 电压低于 BOR 阈值电压时，会产生 POR15V 信号。

POR5V 信号和 POR15V 信号均会将芯片的寄存器复位到初始化状态。

4.1.2 外部复位管脚复位

当外部复位引脚检测到低电平时会产生一个系统复位。该复位引脚已内置上拉电阻，并集成了一个毛刺过滤电路。毛刺过滤电路会过滤小于 20uS（典型值）的毛刺信号，因此，加到复位引脚上的低电平信号必须大于 20uS，才能确保芯片可靠复位。

4.1.3 WDT 复位

看门狗复位，请参看 WDT 一章说明。

4.1.4 PCA 复位

PCA 复位，请参看 PCA 一章说明。

4.1.5 LVD 低电压复位

LVD 复位，请参考 LVD 一章说明。

4.1.6 Cortex-M0+ SYSRESETREQ 复位

Cortex-M0+ 软件复位

4.1.7 Cortex-M0+ LOCKUP 复位

当 Cortex-M0+ 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 CORE 区域。

4.2 寄存器

4.2.1 复位标识寄存器（RESET_FLAG）

复位值: 00000000_00000000_00000000_xxxxxx11b

地址：0x4000201C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RSTB	sysreq	lockup	PCA	WDT	LVD	Por15	Por5v
								RW0	RW0	RW0	RW0	RW0	RW0	RW0	RW0

位	标记	功能描述
31:8	Reserved	保留位
7	RSTB	RESETB 端口复位标志，需要软件初始化及清除，上电状态不定 1：发生端口复位 0：无端口复位发生 写0清除，写1无效
6	Sysreq	Cotrex-M0+ CPU 软件复位标志，需要软件初始化及清除，上电状态不定 1：发生 Cotrex-M0+ CPU 软件复位 0：无 Cotrex-M0+ CPU 软件复位发生 写0清除，写1无效
5	Lockup	Cotrex-M0+ CPU Lockup 复位标志，需要软件初始化及清除，上电状态不定 1：发生 Cotrex-M0+ CPU Lockup 复位 0：无 Cotrex-M0+ CPU Lockup 复位发生 写0清除，写1无效
4	PCA	PCA 复位标志，需要软件初始化及清除，上电状态不定 1：发生 PCA 复位 0：无 PCA 复位发生 写0清除，写1无效
3	WDT	WDT 复位标志，需要软件初始化及清除，上电状态不定 1：发生 WDT 复位 0：无 WDT 复位发生 写0清除，写1无效
2	LVD	LVD 复位标志，需要软件初始化及清除，上电状态不定 1：发生 LVD 复位 0：无 LVD 复位发生 写0清除，写1无效

1	POR15V	VCAP域复位标志 1：VCAP域发生复位 0：VCAP域无复位发生 写 0 清除，写 1 无效
0	POR5V	VCC电源域复位标志 1：VCC电源域发生复位 0：VCC电源域无复位发生 写0清除，写1无效

4.2.2 外围模块复位控制寄存器（PERI_RESET0）

复位值: 0x7F7F6FFF

地址：0x40002028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	HDIV	DMA	GPIO	AES	CRC	SWD	TICK	Res.		Trim	Res.		TRNG	VC	ADC
	RW	RW	RW	RW	RW	RW	RW			RW			RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PCA	OPA	Res.	TIM3	ADVTIM	Res.	BASETIM	SPI1	SPI0	I2C1	I2C0	LPUART1	LPUART0	UART1	UART0
	RW	RW		RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31	Res.	保留位
30	HDIV	HDIV 模块复位使能。 1：正常工作；0：模块处于复位状态
29	DMA	DMAC 模块复位使能。 1：正常工作；0：模块处于复位状态
28	GPIO	GPIO 模块复位使能。 1：正常工作；0：模块处于复位状态
27	AES	AES 模块复位使能。 1：正常工作；0：模块处于复位状态
26	CRC	CRC 模块复位使能。 1：正常工作；0：模块处于复位状态
25	SWD	SWD 模块复位使能。 1：正常工作；0：模块处于复位状态
24	TICK	SYSTICK 模块复位使能。 1：正常工作；0：模块处于复位状态
23:22	Res.	保留位
21	TRIM	CLKTRIM 模块复位使能。 1：正常工作；0：模块处于复位状态
20:19	Res.	保留位
18	TRNG	TRNG 模块复位使能。 1：正常工作；0：模块处于复位状态
17	VC	VC，LVD，模块复位使能。 1：正常工作；0：模块处于复位状态
16	ADC	ADC 模块复位使能。 1：正常工作；0：模块处于复位状态

15	Res.	保留位
14	PCA	PCA 模块复位使能。 1：正常工作；0：模块处于复位状态
13	OPA	OPA 模块复位使能。 1：正常工作；0：模块处于复位状态
12	Res.	保留位
11	TIM3	TIM3 模块复位使能。 1：正常工作；0：模块处于复位状态
10	ADVTIM	TIM456 模块复位使能。 1：正常工作；0：模块处于复位状态
9	Res.	保留位
8	BASETIM	TIM012 模块复位使能。 1：正常工作；0：模块处于复位状态
7	SPI1	SPI1 模块复位使能。 1：正常工作；0：模块处于复位状态
6	SPI0	SPI0 模块复位使能。 1：正常工作；0：模块处于复位状态
5	I2C1	I2C1 模块复位使能。 1：正常工作；0：模块处于复位状态
4	I2C0	I2C0 模块复位使能。 1：正常工作；0：模块处于复位状态
3	LPUART1	LPUART1 模块复位使能。 1：正常工作；0：模块处于复位状态
2	LPUART0	LPUART0 模块复位使能。 1：正常工作；0：模块处于复位状态
1	UART1	UART1 模块复位使能。 1：正常工作；0：模块处于复位状态
0	UART0	UART0 模块复位使能。 1：正常工作；0：模块处于复位状态

4.2.3 外围模块复位控制寄存器（PERI_RESET1）

复位值: 0x0000_03FF

地址：0x4000202C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						UART3	UART2	RES.	I2S1	I2S0	RES.	DAC	CTS	CAN	USB
						R/W	R/W		R/W	R/W		R/W	R/W	R/W	R/W

位	标记	功能描述
31:10	Reseved	保留位
9	UART3	UART3 模块复位使能。 1：正常工作；0：模块处于复位状态
8	UART2	UART2 模块复位使能。 1：正常工作；0：模块处于复位状态
7	RES.	保留位
6	I2S1	I2S1 模块复位使能。 1：正常工作；0：模块处于复位状态
5	I2S0	I2S0 模块复位使能。 1：正常工作；0：模块处于复位状态
4	RES.	保留位
3	DAC	DAC 模块复位使能。 1：正常工作；0：模块处于复位状态
2	CTS	CTS模块复位使能。 1：正常工作；0：模块处于复位状态
1	CAN	CAN 模块复位使能。 1：正常工作；0：模块处于复位状态
0	USB	USB 模块复位使能。 1：正常工作；0：模块处于复位状态

5 中断控制器（NVIC）

5.1 概述

Cortex-M0+ 处理器内置了嵌套向量中断控制器（NVIC），支持最多 32 个中断请求（IRQ）输入，以及 1 个不可屏蔽中断（NMI）输入（在本产品系统中并未使用）。另外，处理器还支持多个内部异常。

每个异常源都有一个单独的异常编号，每种异常类型都有对应的优先级，有些异常的优先级是固定的，而有些则是可编程的。具体如下表所示：

表 5-1 Cortex-M0+ 处理器中断一览

异常编号	异常类型	优先级	描述
1	复位	-3（最高）	复位
2	NMI	-2	不可屏蔽中断（在本系统中没有使用）
3	硬件错误	-1	错误处理异常
4-10	保留	NA	…
11	SVC	可编程	通过SVC指令调用管理程序
12-13	保留	NA	…
14	PendSV	可编程	系统服务的可挂起请求
15	SysTick	可编程	SysTick定时器
16	中断#0	可编程	外部中断#0
17	中断#1	可编程	外部中断#1
…	…	…	…
47	中断#31	可编程	外部中断#31

本章节只对处理器的 32 个外部中断请求（中断#0 到中断#31）做详细介绍，处理器内部异常的具体情况可参考其他相关文档。同时，本章节只讨论处理器内核中 NVIC 的中断处理机制，外设模块自身的中断产生机制不在这里展开讨论。

5.2 中断优先级

每一个外部中断都对应一个优先级寄存器，每个优先级都是 2 位宽，并且使用中断优先级寄存器的最高两位，每个寄存器占 1 个字节（8 位）。在这个设定下，可以使用的优先级为 0x00（最高）、0x40、0x80 和 0xc0（最低）。

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
已使用		未使用，读出为 0					

图 5-1 只使用了高两位的优先级寄存器

如果处理器已经在运行另外一个中断处理，而新中断的优先级大于正在执行的，这时就会发生抢占。正在运行的中断处理会被暂停，转而执行新的中断，这个过程通常被称为中断嵌套。新的中断执行完毕后，之前的中断处理会继续执行，并且在其结束后返回到程序线程中。

如果处理器正在运行的另外一个中断处理的优先级相同或者更高，新的中断将会等待并且进入挂起状态。挂起的中断将会一直等到当前中断等级改变，例如，当前运行的中断处理完成返回后，当前优先级降低到了比挂起中断还要小。

如果两个中断同时发生，并且它们的优先级相同，中断编号较小的中断将会首先执行。例如，如果中断#0 和中断#1 使能且具有相同的优先级，在它们同时被触发时，中断#0 会首先执行。

5.3 中断向量表

当 Cortex-M0+处理器要处理中断服务请求时，它需要首先确定异常处理的起始地址，所需的信息叫做向量表，如图 5-2 所示。向量表存储在存储器空间的开始位置，包含了系统中可用异常（中断）的异常（中断）向量，以及主栈指针（MSP）的初始值。



图 5-2 中断向量表

其中，中断向量的存储顺序同中断编号一致，由于每个向量都是 1 个字（4 字节），中断向量的地址为中断编号乘 4，每个中断向量都是中断处理的起始地址。

5.4 中断输入和挂起行为

Cortex-M0+ 处理器的 NVIC 模块中，每一个中断输入都对应着一个挂起状态寄存器，且每个寄存器只有 1 位，用于保存中断请求，而不管这个请求有没有得到确认。当处理器开始处理这个中断时，硬件将会自动清除挂起状态位。

本系统的外设使用电平触发中断输出，当中断事件发生时，由于外设连接到了 NVIC 上，中断信号会得到确认。在处理器执行中断服务并且清除外设的中断信号以前，该信号会保持高电平。在 NVIC 内部，当检测到有中断发生时，该中断的挂起状态会被置位，当处理器接收该中断并且开始执行中断服务程序后，挂起状态就会被清除。该过程如图 5-3 所示：

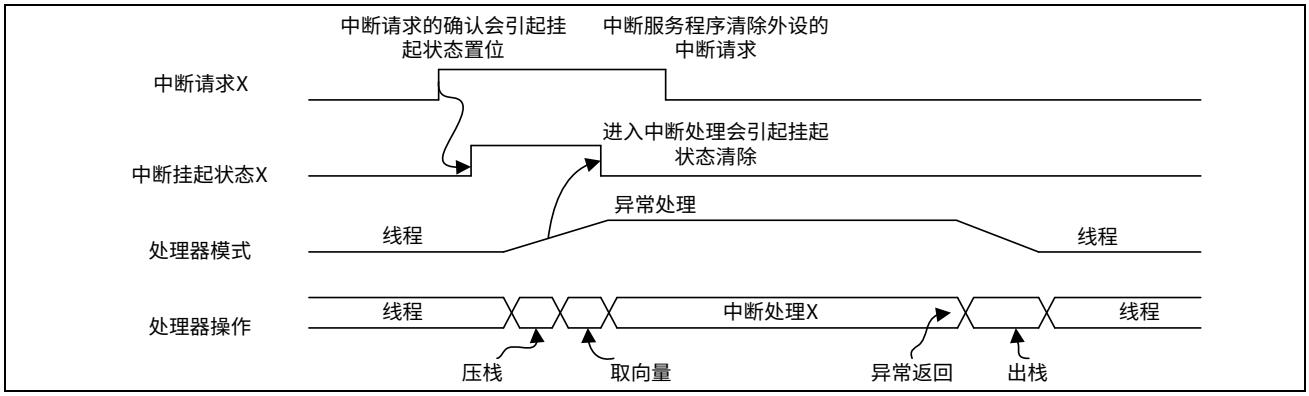


图 5-3 中断激活和挂起状态

如果中断请求没有立即执行，并且在确认之前被软件清除了，这样处理器会忽略掉本次请求，并且不会执行中断处理。可以通过写 NVIC_ICPR 寄存器来清除中断挂起状态，这种处理在设置外设时非常有用，因为在设置以前，该外设可能已经产生了一个中断请求。

如果在软件清除挂起状态时，外设仍然保持着中断请求，挂起状态还会立即生成。该过程如图 5-4 所示：

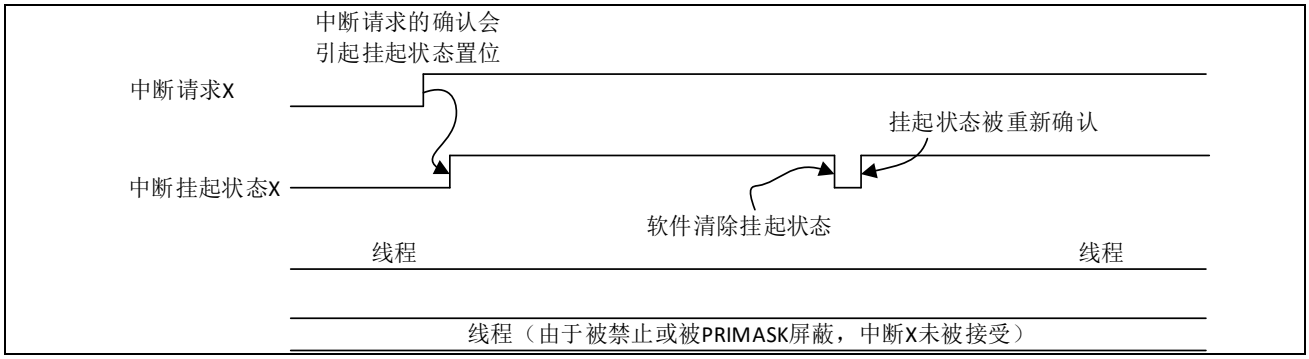


图 5-4 中断挂起状态被清除然后被重新确认

如果外设产生的中断请求在异常处理时没有被清除，异常返回后挂起状态就会被又一次激活，这样中断服务程序会再次执行。该过程如图 5-5 所示：

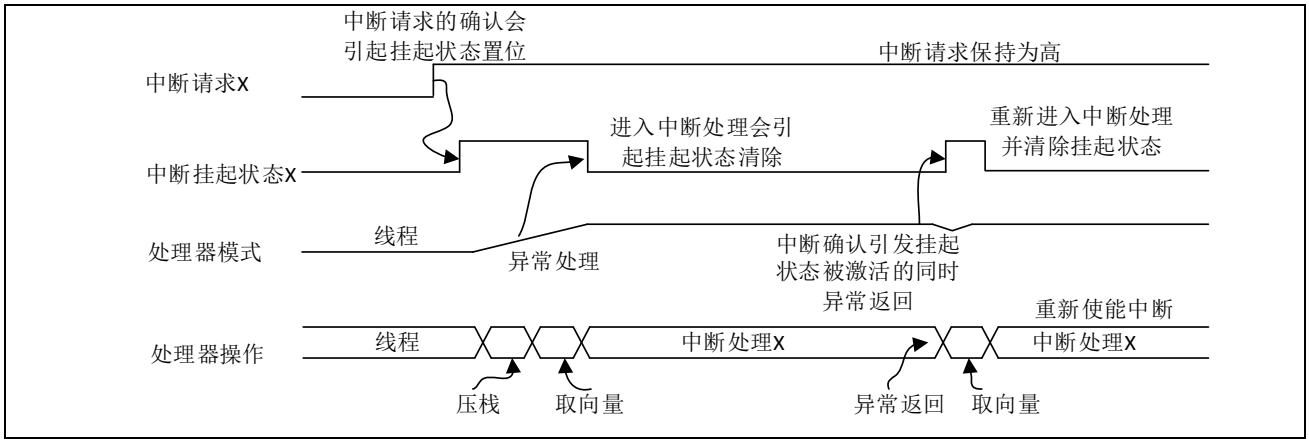


图 5-5 中断退出时若中断请求保持高电平就会引起中断处理的再次执行

如果在终端服务程序执行的过程中产生外设中断请求，该请求会被当作新的中断请求，并且在本次中断退出后，还会引起中断服务程序的再次执行。该过程如图 5-6 所示：

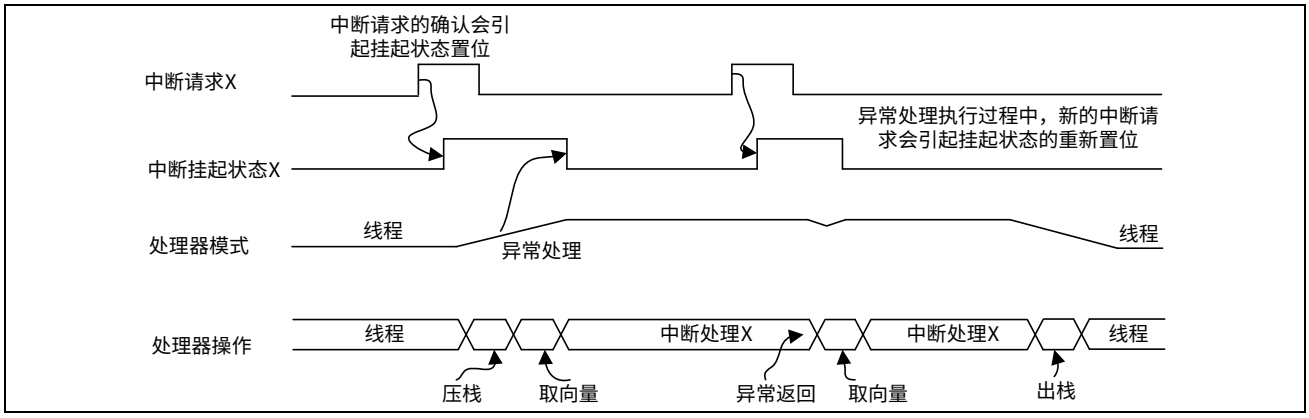


图 5-6 中断处理中产生的中断挂起也可以被确认

5.5 中断等待

通常情况下，NVIC 的中断等待时间为 16 个周期。这个等待时间从中断确认的处理器时钟周期开始，一直到中断处理开始执行结束。计算中断等待需要具备以下前提：

- 该中断使能并且没有被 PRIMASK 或是其他正在执行的异常处理所屏蔽。
- 存储器系统没有任何等待状态，在中断处理、压栈、取向量或者中断处理开始时的取指都会用到总线传输，如果存储器系统需要等待，那么发生总线传输时产生的等待状态则可能使得中断延迟。

下面的几种情况可能会导致不同的中断等待：

- 中断的末尾连锁，如果中断返回时产生了另外一个中断请求，处理器就会跳过出栈和压栈过程，这样就减少了中断等待时间。
- 延迟到达，如果中断发生时，另外一个低优先级的中断正在进行压栈处理，由于延迟到达机制的存在，高优先级的中断会首先执行，这样也会导致高优先级中断的等待时间减小。

5.6 中断源

因为 Cortex-M0+ 处理器的 NVIC 支持最多 32 个外部中断，而在本系统中，外部中断源大于 32 个，所以部分外部中断复用在同一个 NVIC 中断输入上，其中 NMI（不可屏蔽中断）并没有使用。本系统所有外部中断源和 NVIC 中断输入的对应关系如下表所示：

表 5-2 外部中断与 NVIC 中断输入对应关系

NVIC中断输入	外部中断源	Active模式	Sleep模式	DeepSleep模式
中断#0	PORTA	✓	✓	✓
中断#1	PORTB	✓	✓	✓
中断#2	PORTC/PORTE	✓	✓	✓
中断#3	PORTD/PORTF	✓	✓	✓
中断#4	DMAC	✓	✓	-
中断#5	TIM3	✓	✓	-
中断#6	UART0/2	✓	✓	-
中断#7	UART1/3	✓	✓	-
中断#8	LPUART0	✓	✓	✓
中断#9	LPUART1	✓	✓	✓
中断#10	SPI0/I2S0	✓	✓	-
中断#11	SPI1/I2S1	✓	✓	-
中断#12	I2C0	✓	✓	-
中断#13	I2C1	✓	✓	-
中断#14	TIM0	✓	✓	-
中断#15	TIM1	✓	✓	-
中断#16	TIM2	✓	✓	-
中断#17	-			
中断#18	TIM4	✓	✓	-
中断#19	TIM5	✓	✓	-
中断#20	TIM6	✓	✓	-
中断#21	PCA	✓	✓	-
中断#22	WDT	✓	✓	✓
中断#23	-			
中断#24	ADC/DAC	✓	✓	-
中断#25	-			
中断#26	VC0/VC1/VC2/LVD	✓	✓	✓
中断#27	USB	✓	✓	
中断#28	CAN	✓	✓	
中断#29	-			
中断#30	FLASH/RAM	✓	✓	-

NVIC中断输入	外部中断源	Active模式	Sleep模式	DeepSleep模式
中断#31	CLKTRIM/CTS	✓	✓	✓

注意：

- 由于某些模块中断被复用于同一个 IRQ 中断源，当 CPU 进入该中断操作时，必须先判断是哪个模块产生的中断，再进行相对应的中断操作。

5.7 中断结构图

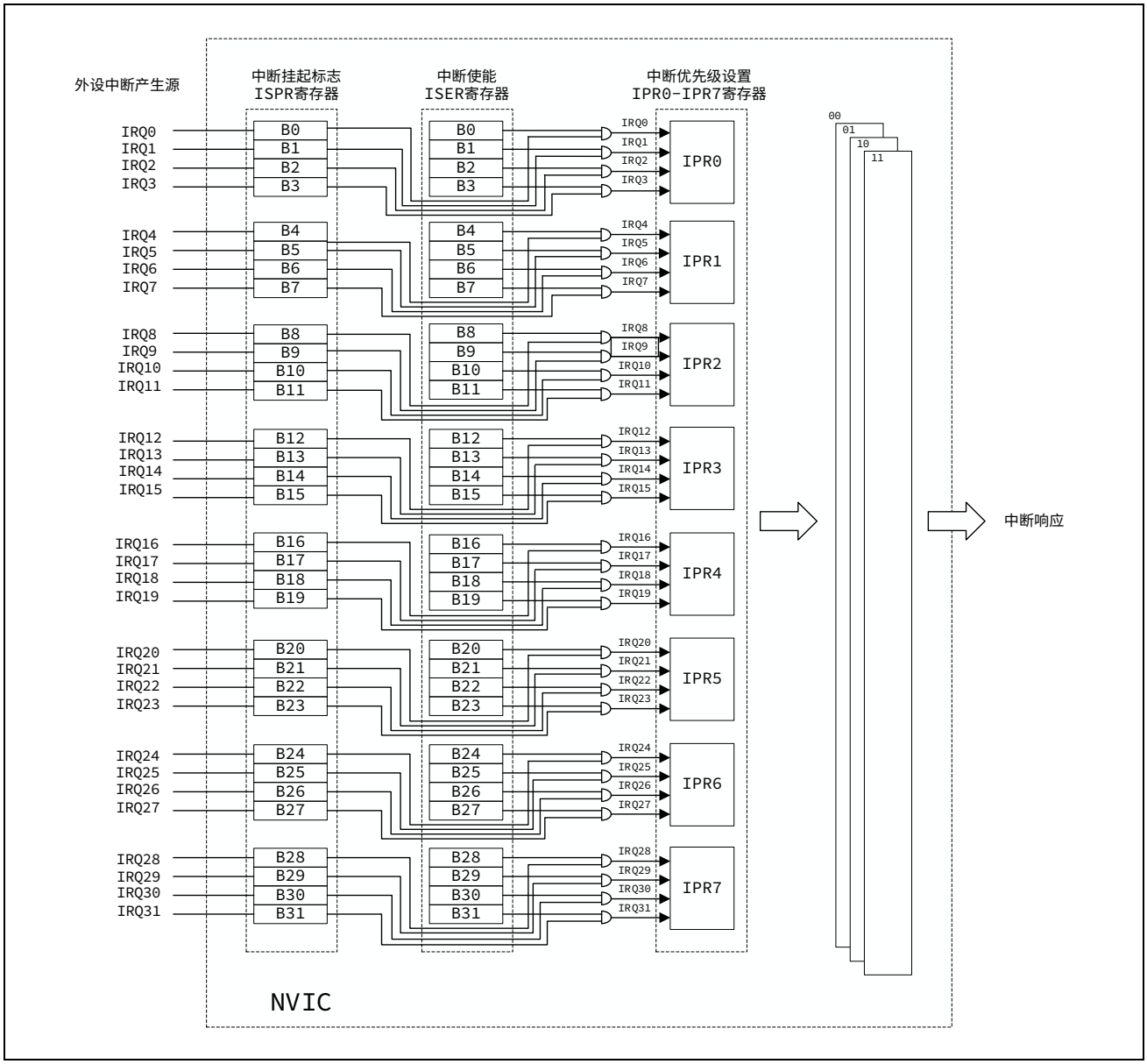


图 5-7 中断结构图

本系统的中断结构框图如图 5-7 所示。需要注意几点：

- 外设中断源各自的中断使能没有在图中标出，这里只包括外设中断产生之后的中断信号逻辑框图。
- IRQ 有多于 1 个外设中断输入复用，必须分别读取这些外设的中断标志位来判定是哪个外设的中断。
- 如果外设中断源有高电平产生，无论 NVIC 中断使能寄存器 SCS_SETENA 置位与否，中断挂起寄存器 SCS_SEPEND 都会被置位，表示相应的外设中断源有中断产生。
- 只有中断使能寄存器 NVIC_ISER 置位，相应的中断 IRQ 才会给处理器相应，执行相应的中断程序。
- 在中断程序中必须清除外设中断源高电平中断信号，中断挂起寄存器 NVIC_ISPR 由硬件自动清除。
- 中断优先级寄存器 NVIC_IPR0- NVIC_IPR7 设置了 32 个中断源的优先级，00 优先级最高，11 优先级最低。当优先级相同时，优先级由中断编号决定，编号越小优先级越高。

5.8 寄存器

基地址：0xE000 E000

寄存器	偏移地址	描述
NVIC_ISER	0x100	中断请求使能寄存器
NVIC_ICER	0x180	中断请求清除使能寄存器
NVIC_ISPR	0x200	中断设置挂起寄存器
NVIC_ICPR	0x280	中断清除挂起寄存器
NVIC_IPR0	0x400	中断#0-中断#3优先级寄存器
NVIC_IPR1	0x404	中断#4-中断#7优先级寄存器
NVIC_IPR2	0x408	中断#8-中断#11优先级寄存器
NVIC_IPR3	0x40C	中断#12-中断#15优先级寄存器
NVIC_IPR4	0x410	中断#16-中断#19优先级寄存器
NVIC_IPR5	0x414	中断#20-中断#23优先级寄存器
NVIC_IPR6	0x418	中断#24-中断#27优先级寄存器
NVIC_IPR7	0x41C	中断#28-中断#31优先级寄存器
PRIMASK	-	中断屏蔽特殊寄存器

5.8.1 中断使能设置寄存器（NVIC_ISER）

偏移地址：0x100

复位值：0x0000 0000

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

位	标记	功能描述
31:0	SETENA [31:0]	设置使能中断#0到中断#31；写“1”置位，写“0”无效 [0]:IRQ0 [1]:IRQ1 [2]:IRQ2 ..... [31]:IRQ31

5.8.2 中断使能清除寄存器（NVIC_ICER）

偏移地址：0x180

复位值：0x0000 0000

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

位	标记	描述
31:0	CLRENA	清除使能中断#0到中断#31；写“1”清零，写“0”无效

5.8.3 中断挂起状态设置寄存器（NVIC_ISPR）

偏移地址：0x200

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SETPEND[31:16]															
RW															

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

位	标记	功能描述
31:0	SETPEND	设置中断#0到中断#31的挂起状态；写“1”置位，写“0”无效 [0]:IRQ0 [1]:IRQ1 [2]:IRQ2 …… [31]:IRQ31

5.8.4 中断挂起状态清除寄存器（NVIC_ICPR）

偏移地址：0x280

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLRPEND[31:16]															
RW															

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

位	标记	描述
31:0	CLRPEND	清除中断#0到中断#31的挂起状态；写“1”清零，写“0”无效 [0]:IRQ0 [1]:IRQ1 [2]:IRQ2 …… [31]:IRQ31

5.8.5 中断优先级寄存器（NVIC_IPR0）

偏移地址：0x400

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR0[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR0[15:0]]															
RW															

位	标记	功能描述
31:0	IPR0[31:0]	中断#0 到中断#3 的优先级； [31:30]：中断#3 的优先级 [23:22]：中断#2 的优先级 [15:14]：中断#1 的优先级 [7:6]： 中断#0 的优先级 其中，00 优先级最高，11 优先级最低

5.8.6 中断优先级寄存器（NVIC_IPR1）

偏移地址：0x404

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR1[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR1[15:0]]															
RW															

位	标记	功能描述
31:0	IPR1[31:0]	中断#4 到中断#7 的优先级； [31:30]：中断#7 的优先级 [23:22]：中断#6 的优先级 [15:14]：中断#5 的优先级 [7:6]： 中断#4 的优先级 其中，00 优先级最高，11 优先级最低

5.8.7 中断优先级寄存器（NVIC_IPR2）

偏移地址：0x408

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR2[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR2[15:0]]															
RW															

位	标记	功能描述
31:0	IPR2[31:0]	中断#8 到中断#11 的优先级； [31:30]：中断#11 的优先级 [23:22]：中断#10 的优先级 [15:14]：中断#9 的优先级 [7:6]： 中断#8 的优先级 其中，00 优先级最高，11 优先级最低

5.8.8 中断优先级寄存器（NVIC_IPR3）

偏移地址：0x40C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR3[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR3[15:0]]															
RW															

位	标记	功能描述
31:0	IPR3[31:0]	中断#12 到中断#15 的优先级； [31:30]：中断#15 的优先级 [23:22]：中断#14 的优先级 [15:14]：中断#13 的优先级 [7:6]： 中断#12 的优先级 其中，00 优先级最高，11 优先级最低

5.8.9 中断优先级寄存器（NVIC_IPR4）

偏移地址：0x410

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR4[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR4[15:0]]															
RW															

位	标记	功能描述
31:0	IPR4[31:0]	中断#16 到中断#19 的优先级； [31:30]：中断#19 的优先级 [23:22]：中断#18 的优先级 [15:14]：中断#17 的优先级 [7:6]： 中断#16 的优先级 其中，00 优先级最高，11 优先级最低

5.8.10 中断优先级寄存器（NVIC_IPR5）

偏移地址：0x414

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR5[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR5[15:0]]															
RW															

位	标记	功能描述
31:0	IPR5[31:0]	中断#20 到中断#23 的优先级； [31:30]：中断#23 的优先级 [23:22]：中断#22 的优先级 [15:14]：中断#21 的优先级 [7:6]： 中断#20 的优先级 其中，00 优先级最高，11 优先级最低

5.8.11 中断优先级寄存器（NVIC_IPR6）

偏移地址：0x418

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR6[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR6[15:0]]															
RW															

位	标记	功能描述
31:0	IPR6[31:0]	中断#24 到中断#27 的优先级； [31:30]：中断#27 的优先级 [23:22]：中断#26 的优先级 [15:14]：中断#25 的优先级 [7:6]： 中断#24 的优先级 其中，00 优先级最高，11 优先级最低

5.8.12 中断优先级寄存器（NVIC_IPR7）

偏移地址：0x41C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR7[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR7[15:0]]															
RW															

位	标记	功能描述
31:0	IPR7[31:0]	中断#28 到中断#31 的优先级； [31:30]：中断#31 的优先级 [23:22]：中断#30 的优先级 [15:14]：中断#29 的优先级 [7:6]： 中断#28 的优先级 其中，00 优先级最高，11 优先级最低

5.8.13 中断屏蔽特殊寄存器（PRIMASK）

偏移地址：---

复位值：0x0000 0000

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

位	符号	描述
31:1	Reserved	
0	PRIMASK	置位后，除了 NMI 和硬件错误异常外的其他中断都会被屏蔽掉 清零后，所有异常和中断都不会被屏蔽掉 该特殊寄存器需要通过 MSR 和 MRS 特殊寄存器操作指令访问，也可以用改变处理器状态指令 CPS 访问。在处理对时间敏感的应用时，需要操作 PRIMASK 寄存器。

5.9 软件基本操作

5.9.1 外部中断使能

在每一个外设模块内部都有各自的中断使能寄存器，在需要进行中断操作时，必须首先打开外设自己的中断使能。该使能位的操作没有在本章节讨论，请参考外设模块各自的章节描述。

5.9.2 NVIC 中断使能和清除使能

Cortex-M0+ 处理器支持最多 32 个中断源，每个中断源都对应有一个中断使能位和清零使能位。这样就有了 32 位的中断使能寄存器 NVIC_ISER 和 32 位的清零使能寄存器 NVIC_ICER。如果想使能某一个中断，则对 NVIC_ISER 寄存器的相应位置 1。如果想清零某一个中断，则对 NVIC_ICER 寄存器的相应位置 1。

注意，这里提到的中断使能仅仅是针对处理器 NVIC 而言的，每个外设的中断生成与否，是由外设的中断控制寄存器决定，与 NVIC_ISER 和 NVIC_ICER 无关。

5.9.3 NVIC 中断挂起和清除挂起

如果一个中断发生了，却无法立即处理，这个中断请求将会被挂起。挂起状态保存在一个寄存器中，如果处理器的当前优先级还没有降低到可以处理挂起的请求，并且没有手动清除挂起状态，该状态将会一直保持合法。

当处理器开始进入中断处理，硬件会自动引起挂起状态的清除。

可以通过操作中断设置挂起 NVIC_ISPR 和中断清除挂起 NVIC_ICPR 这两个寄存器来访问或修改中断挂起状态。中断挂起状态寄存器允许使用软件来触发中断。

5.9.4 NVIC 中断优先级

设置 NVIC_IPR0- NVIC_IPR7 寄存器决定 SCS_IRQ0- SCS_IRQ32 的优先级。中断优先级寄存器的编程应该在中断使能之前，其通常是在程序开始时完成的。应该避免在中断使能之后改变中断优先级，这种情况的结果不可预知，并且不被 Cortex-M0+ 处理器支持。

5.9.5 NVIC 中断屏蔽

有些对时间敏感的应用，需要在一段较短的时间内禁止所有中断，可以利用中断屏蔽寄存器 SCS_PRIMASK 实现。特殊寄存器 SCS_PRIMASK 只有 1 位有效，并且在复位后默认为 0。该寄存器为 0 时，所有的中断和异常都处于允许状态；而设为 1 后，只有 NMI（本系统不支持）和硬件错误异常处于使能。实际上，当 SCS_PRIMASK 设置为 1 后，处理器的当前优先级就降到了 0（可和值的最高优先级）。

可以通过多种方法编程 SCS_PRIMASK 寄存器，使用汇编语言，可以利用 MS R 指令在设置和清除 SCS_PRIMASK 寄存器。若使用 C 语言以及 CMSIS 设备驱动库，用户可以使用以下函数来设置和清除 PRIMASK。

void __enable_irq(void); //清除 PRIMASK

void __disable_irq(void); //设置 PRIMASK

6 端口控制器（GPIO）

6.1 端口控制器简介

本系列有 86 个数字通用输入输出端口 PA[15:0], PB[15:0], PC[15:0], PD[15:0], PE[15:0], PF[11:9], PF[7:0]。所有封装未封出的 GPIO 也应配置为输入并使能上拉。

模拟模块 ADC/VC/LVD 的输入输出信号，各功能模块（如 SPI，UART，I2C，Timer 等）的输入输出信号和数字通用输入输出端口复用。

每个端口都可以配置成内部上拉（pull up）/下拉（pull down）的输入，高阻输入（floating input），推挽输出（CMOS output），开漏输出（open drain output），两档驱动能力输出。数字端口被配置成模拟端口后，数字功能被隔离，不能输出数字“1”和“0”，CPU 读取端口输入值寄存器的结果为“0”。

每个数字端口被配置为输入时，都可以提供外部中断，中断类型可以配置成高电平触发、低电平触发、上升沿触发、下降沿触发 4 种，查询 Px_STAT[n]的中断标志位即可知道相应的中断触发端口。另外，每个数字端口的中断都可以把芯片从休眠模式/深度休眠模式唤醒到工作模式。

芯片复位后端口为高阻输入（floating input），目的是防止芯片被异常复位时，对外部器件产生异常动作。但为了避免高阻输入而产生的漏电，用户要在芯片启动之后对端口进行相应的配置（配置成内部上拉/下拉输入或者输出）。

6.2 端口控制器主要特性

端口控制器支持以下特性：

- 端口输入值/输出值寄存器支持 FAST IO/AHB 总线读写
- 其他寄存器支持 AHB 总线接口读写
- 模拟功能引脚/数字通用引脚/数字功能引脚复用
- 支持上拉/下拉/两档驱动/开漏输出功能选择
- 支持工作模式/休眠模式/深度休眠模式下中断
- 支持高电平/低电平/上升沿/下降沿触发中断
- 支持位置位，位清零，位置位清零功能

注：*FAST IO* 介绍请参考 *ARM Cortex-M0+_IntegrationAndImplementationManual.pdf*。

6.3 端口控制器功能描述

6.3.1 端口配置功能

每一个端口都可以根据系统需求通过配置寄存器（**PxADS**）把端口配置成模拟端口或数字端口。当PxADS 为‘1’时，端口配置成模拟端口，当 PxADS 为‘0’时，端口配置成数字端口。

端口电路结构如下图所示：

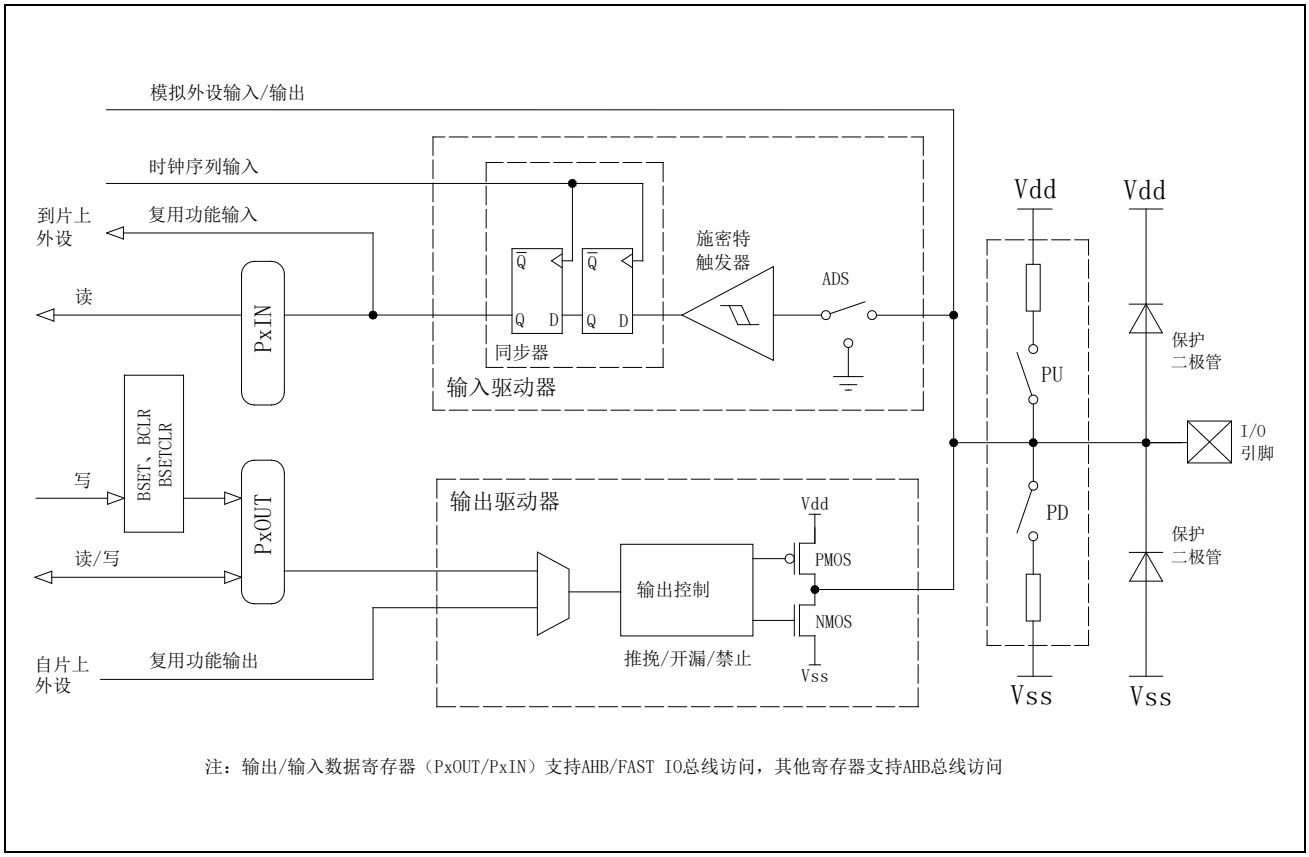


图 6-1 端口电路图

端口配置为数字端口时，可以接受数字通用输入输出信号（设置 Px_SEL 寄存器为‘0’），也可以通过配置寄存器 Px_SEL 接受各功能模块（如 SPI，UART，I2C，Timer 等）的输入输出信号，详见端口数字复用功能模块。

还可以配置相应的寄存器实现以下特性：

1) 内部上拉（PxPU）/下拉（PxPD）

上拉寄存器（PxPU）、下拉寄存器（PxPD）分别对应端口上拉使能和端口下拉使能，当对应位为‘1’时，设置对应位引脚上拉/下拉使能，为‘0’时，禁止对应位引脚上拉/下拉。

2) 两档驱动输出（PxDR）

能够通过 PxDR 寄存器改变驱动能力，PxDR 为‘1’时为低驱动能力，PxDR 为‘0’时为高驱动能力。

3) 开漏输出（PxOD）

通过 PxOD 寄存器设置管脚输出状态。当 PxOD 为‘1’时，端口开漏输出使能，为‘0’时，端口推挽输出使能。开漏引脚不连接外部的上拉电阻时，只能输出低电平，如果需要同时具备输出高电平的功能时，则需要上拉电阻。

4) 方向选择（PxDIR）

用于设置端口引脚的方向。PxDIR 为‘0’的时候端口为输出，PxDIR 为‘1’的时候端口为输入。

5) 输入电平状态 （PxIN）

通过读取 PxIN 寄存器能够获得同步后的管脚电平，PxIN 为‘1’时为高电平，PxIN 为‘0’时为低电平。可通过 AHB/FAST IO 总线访问。

6) 输出高低电平选择（PxOUT）

当端口引脚配置为输出时，如果 PxOUT 为‘1’，端口引脚输出为高电平，如果 PxOUT 为‘0’，则输出低电平。可通过 AHB/FAST IO 总线访问。

7) 位置位（PxBSET）、位清零（PxBCLR）、位置位清零（PxBSETCLR）

位置位与位清零适用于在不改变其他位的值的情况下，对用户想要改变的位进行设置，得到相应的值。位置位是将对应的值设置为‘1’，位清零是将对应的值设置为‘0’。

注意：

- 以上特性在配置为模拟端口时无效。

端口的状态和寄存器配置的关系如下表：

表 6-1 端口状态真值表

IO 状态	IO 方向	PxADS	PxDIR	PxOUT	PxIN	PxBSET	PxBCLR	PxBSETCLR	PxPU	PxPD	PxOD	PxDR	Px_SEL
模拟	输入/输出	1	W	W	0	W	W	W	W	W	W	W	W
浮空	输入	0	1	W	X	W	W	W	0	0	W	W	0
下拉	输入	0	1	W	0	W	W	W	0	1	W	W	0
上拉	输入	0	1	W	1	W	W	W	1	0	W	W	0
上拉	输入	0	1	W	1	W	W	W	1	1	W	W	0
1	输入	0	1	W	1	W	W	W	W	W	W	W	0
0	输入	0	1	W	0	W	W	W	W	W	W	W	0
1	输出	0	0	1	1	0	0	0	W	W	0	W	0
0	输出	0	0	0	0	0	0	0	W	W	0	W	0
1	输出	0	0	W	1	1	0	0	W	W	0	W	0
0	输出	0	0	W	0	0	1	0	W	W	0	W	0
(SET) 1	输出	0	0	W	(SET) 1	0	0	1	W	W	0	W	0
(CLR) 0	输出	0	0	W	(CLR) 0	0	0	1	W	W	0	W	0
0	输出	0	0	0	0	0	0	0	W	W	1	W	0
Z	输出	0	0	1	X	0	0	0	0	0	1	W	0
0	输出	0	0	1	0	0	0	0	0	1	1	W	0
1	输出	0	0	1	1	0	0	0	1	0	1	W	0
1	输出	0	0	1	1	0	0	0	1	1	1	W	0
注：	0 - Logic low		1 - Logic high		W - Whatever 0 or 1		X - unknow state		Z - high impedance				

6.3.2 端口的写入

端口输入值/输出值寄存器（PxIN/PxOUT）支持 AHB 总线和 FAST IO 总线的读写（通过寄存器 GPIO_CTRL2.ahb_sel 位控制），对于其他的寄存器则只支持 AHB 总线的读写。对于这两种不同的总线，系统时钟（HCLK）对这两种总线的处理周期并不相同。下面两幅图为对于两种总线端口翻转的最快时序：

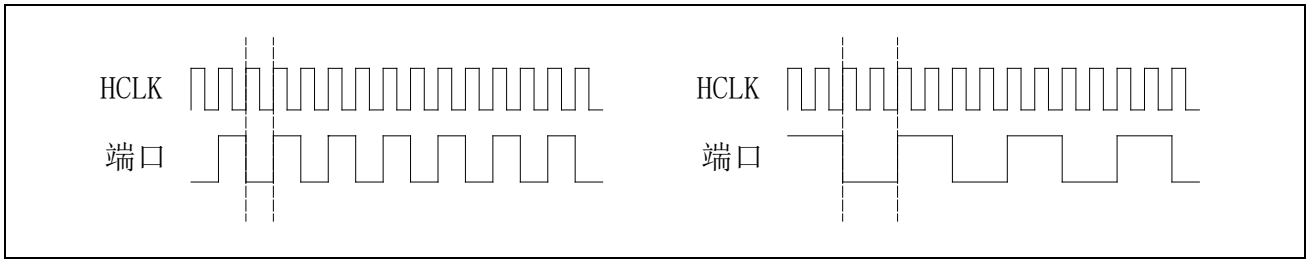


图 6-2 AHB/FASTIO 总线端口随系统时钟的变化

（左为 FAST IO，右为 AHB）

对于 AHB 总线，每两个 HCLK 周期，IO 翻转一次，而对于 FAST IO 总线，每个 HCLK 周期，IO 翻转一次。

6.3.3 端口的读取

每一个端口可以通过读取 PxIN 寄存器来获得端口引脚电平。如图 6-1 所示，PxIN 寄存器的各个位与其前面的锁存器组成了一个同步器从而避免了系统时钟状态发生改变时在短时间内引脚电平变化而造成的信号不稳定，但是同时也引入了延迟。读取端口引脚数据的同步图如下：

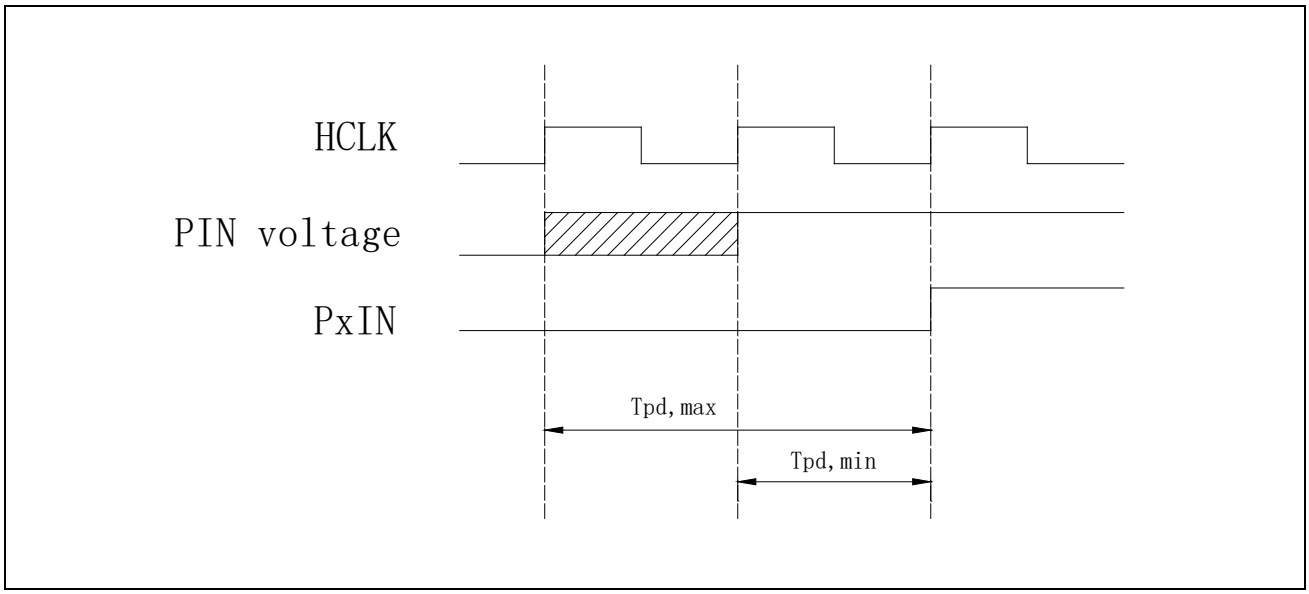


图 6-3 读取端口引脚数据同步图

在系统时钟上升沿之后的时钟时期，管脚电平信号会锁存在内部寄存器，如阴影部分所示，在下一次系统时钟上升沿之后，稳定的管脚电平信号能够被读取。之后再系统时钟上升沿时，数据锁存到 PxIN 寄存器中。信号装换延迟 Tpd 为 1-2 个系统时钟。

注意：

- 未连接引脚的处理：
  在使用过程中如果有未接引脚，为避免在其他数字输入使能模式时引脚没有确定的电平造成悬空消耗电流，推荐赋予引脚一个确定的电平。

6.3.4 端口数字复用功能

端口数字复用是端口控制器的主要功能之一。通过配置寄存器，可以灵活的把端口配置成测试调试端口/数字通用端口/数字功能端口。

如下表所示：Px_SEL 寄存器用于数字通用端口/数字功能端口切换，每一个端口都可以独立配置成系统所需的功能端口。调试端口的配置信息请参考相关章节。

表 6-2 端口复用表

PxSEL							
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
PA00	UART1_CTS	LPUART1_TXD	TIM0_ETR	VC0_OUT	TIM1_CHA	TIM3_ETR	TIM0_CHA
PA01	UART1_RTS	LPUART1_RXD	TIM0_CHB	TIM1_ETR	TIM1_CHB	HCLK_OUT	SPI1_MOSI
PA02	UART1_TXD	TIM0_CHA	VC1_OUT	TIM1_CHA	TIM2_CHA	PCLK_OUT	SPI1_MISO
PA03	UART1_RXD	TIM0_GATE	TIM1_CHB	TIM2_CHB	SPI1_CS	TIM3_CH1A	TIM5_CHA
PA04	SPI0_CS	UART1_TXD	PCA_CH4	TIM2_ETR	TIM5_CHA	LVD_OUT	TIM3_CH2B
PA05	SPI0_SCK	TIM0_ETR	PCA_ECI	TIM0_CHA	TIM5_CHB	XTL_OUT	XTH_OUT
PA06	SPI0_MISO	PCA_CH0	TIM3_BK	TIM1_CHA	VC0_OUT	TIM3_GATE	LPUART0_CTS
PA07	SPI0_MOSI	PCA_CH1	HCLK_OUT	TIM3_CH0B	TIM2_CHA	VC1_OUT	TIM4_CHB
PA08	UART0_TXD	TIM3_CH0A	CRS_SYNC	CAN_STBY	TIM1_GATE	TIM4_CHA	TIM3_BK
PA09	UART0_TXD	TIM3_CH1A	TIM0_BK	I2C0_SCL		HCLK_OUT	TIM5_CHA
PA10	UART0_RXD	TIM3_CH2A	TIM2_BK	I2C0_SDA	TIM2_GATE	PCLK_OUT	TIM6_CHA
PA11	UART0_CTS	TIM3_GATE	I2C1_SCL	CAN_RX	VC0_OUT	SPI0_MISO	TIM4_CHB
PA12	UART0_RTS	TIM3_ETR	I2C1_SDA	CAN_TX	VC1_OUT	SPI0_MOSI	
PA13	IR_OUT	UART0_RXD	LVD_OUT	TIM3_ETR			VC2_OUT
PA14	UART1_TXD	UART0_TXD	TIM3_CH2A	LVD_OUT	RCH_OUT	RCL_OUT	PLL_OUT
PA15	SPI0_CS	UART1_RXD	LPUART1_RTS	TIM0_ETR	TIM0_CHA	TIM3_CH1A	
PB00	PCA_CH2	TIM3_CH1B	LPUART0_TXD	TIM5_CHB	RCH_OUT	RCL_OUT	PLL_OUT
PB01	PCA_CH3	PCLK_OUT	TIM3_CH2B	TIM6_CHB	LPUART0_RTS	VC2_OUT	TCLK_OUT
PB02		PCA_ECI	LPUART1_TXD	TIM4_CHA	TIM1_BK	TIM0_BK	TIM2_BK
PB03	SPI0_SCK	TIM0_CHB	TIM1_GATE	TIM3_CH0A		XTL_OUT	XTH_OUT
PB04	SPI0_MISO	PCA_CH0	TIM2_BK	UART0_CTS	TIM2_GATE	TIM3_CH0B	
PB05	SPI0_MOSI		TIM1_BK	PCA_CH1			UART0_RTS
PB06	I2C0_SCL	UART0_TXD	TIM1_CHB	TIM0_CHA		TIM3_CH0A	
PB07	I2C0_SDA	UART0_RXD	TIM2_CHB	LPUART1_CTS	TIM0_CHB		
PB08	I2C0_SCL	TIM1_CHA	CAN_RX	TIM2_CHA	TIM0_GATE	TIM3_CH2A	UART0_TXD
PB09	I2C0_SDA	IR_OUT	SPI1_CS	TIM2_CHA	CAN_TX	TIM2_CHB	UART0_RXD
PB10	I2C1_SCL	SPI1_SCK	TIM1_CHA	LPUART0_TXD	TIM3_CH1A	LPUART1_RTS	UART1_RTS
PB11	I2C1_SDA	TIM1_CHB	LPUART0_RXD	TIM2_GATE	TIM6_CHA	LPUART1_CTS	UART1_CTS
PB12	SPI1_CS	TIM3_BK	LPUART0_TXD	TIM0_BK		LPUART0_RTS	TIM6_CHA

PxSEL							
0	1	2	3	4	5	6	7
PB13	SPI1_SCK	I2C1_SCL	TIM3_CH0B	LPUART0_CTS	TIM1_CHA	TIM1_GATE	TIM6_CHB
PB14	SPI1_MISO	I2C1_SDA	TIM3_CH1B	TIM0_CHA		LPUART0_RTS	TIM1_BK
PB15	SPI1_MOSI	TIM3_CH2B	TIM0_CHB	TIM0_GATE			LPUART1_RXD
PC00			UART1_CTS	UART2_RTS	I2S0_MCK		
PC01		TIM5_CHB	UART1_RTS		I2S0_SD	UART2_CTS	
PC02	SPI1_MISO			UART2_RXD			
PC03	SPI1_MOSI				UART2_TXD		
PC04	LPUART0_TXD	TIM2_ETR	IR_OUT	VC2_OUT	I2S0_WS		
PC05	LPUART0_RXD	TIM6_CHB	PCA_CH4		I2S0_SDIN		
PC06	PCA_CH0	TIM4_CHA	TIM2_CHA		I2S1_SCK	UART3_RXD	
PC07	PCA_CH1	TIM5_CHA	TIM2_CHB		I2S1_MCK	UART3_TXD	
PC08	PCA_CH2	TIM6_CHA	TIM2_ETR		I2S1_SD	UART3_CTS	
PC09	PCA_CH3	TIM4_CHB	TIM1_ETR		I2S1_WS	UART3_RTS	
PC10	LPUART1_TXD	LPUART0_TXD	PCA_CH2				
PC11	LPUART1_RXD	LPUART0_RXD	PCA_CH3				
PC12	LPUART0_TXD	LPUART1_TXD	PCA_CH4				
PC13			TIM3_CH1B		I2S0_SCK		
PC14							
PC15							
PD00	CAN_RX	SPI1_CS					
PD01	CAN_TX	SPI1_SCK					
PD02	PCA_ECI	LPUART0_RTS	TIM1_ETR				
PD03	UART1_CTS	SPI1_MISO		I2S1_SCK			
PD04	UART1_RTS	SPI1_MOSI		I2S1_MCK			
PD05	UART1_TXD		CAN_STBY	I2S1_SD			
PD06	UART1_RXD			I2S1_WS			
PD07	UART1_TXD			I2S1_SDIN			
PD08	LPUART0_TXD	I2S0_SCK					
PD09	LPUART0_RXD	I2S0_MCK					
PD10	LPUART0_TXD	I2S0_SD					
PD11	LPUART0_CTS	I2S0_WS					
PD12	LPUART0_RTS	UART2_RTS					
PD13	UART2_RXD	I2S0_SDIN					
PD14	UART2_TXD						
PD15	CRS_SYNC	UART2_CTS					
PE00	TIM1_CHA						
PE01	TIM2_CHA						

PxSEL							
0	1	2	3	4	5	6	7
PE02	PCA_ECI						
PE03	PCA_CH0						
PE04	PCA_CH1						
PE05	PCA_CH2						
PE06	PCA_CH3						
PE07	TIM3_ETR						
PE08	TIM3_CH0B						
PE09	TIM3_CH0A						
PE10	TIM3_CH1B						
PE11	TIM3_CH1A						
PE12	TIM3_CH2B	SPI0_CS	UART3_CTS				
PE13	TIM3_CH2A	SPI0_SCK	UART3_RTS				
PE14	TIM3_CH0B	SPI0_MISO	UART3_RXD				
PE15	TIM3_BK	SPI0_MOSI	UART3_TXD				
PF00	I2C0_SDA	CRS_SYNC	UART1_TXD				
PF01	I2C0_SCL		UART1_RXD				
PF02							
PF03							
PF04							
PF05							
PF06	I2C1_SCL	LPUART1_CTS	UART0_CTS				
PF07	I2C1_SDA	LPUART1_RTS	UART0_RTS				
PF09	TIM0_CHA						
PF10	TIM0_CHB						
PF11							

6.3.5 端口中断功能

每一个数字通用端口都可以由外部信号源产生中断，外部信号源可以是高电平/低电平/上升沿/下降沿 4 种类型的信号，分别对应的中断使能寄存器为高电平中断使能寄存器(PxHIE)/低电平中断使能寄存器(PxLIE)/上升沿中断使能寄存器(PxRIE)/下降沿中断使能寄存器(PxFIE)。当中断触发时，通过查询中断状态寄存器(Px_STAT)就可以判断是哪一个端口触发了中断，通过清零中断清除寄存器(Px_ICLR)就可以清除对应的中断状态标志位。

6.4 端口配置操作流程

6.4.1 端口复用配置为模拟端口操作流程

a) 设置寄存器 PxADS[n]为 1

6.4.2 端口复用配置为数字通用端口操作流程

a) 设置寄存器 PxADS[n]为 0
b) 设置寄存器 Px_SEL 为 0
c) 设置寄存器 PxDIR[n]为 1：端口方向为输入，CPU 可以读取端口的状态 PxIN[n]。
d) 设置寄存器 PxDIR[n]为 0：端口方向为输出
e) 设置寄存器 PxOUT[n]为 1：端口输出高电平
f) 设置寄存器 PxOUT[n]为 0：端口输出低电平

6.4.3 端口复用配置为数字功能端口操作流程

a) 设置寄存器 PxADS[n]为 0
b) 设置寄存器 Px_SEL 为 1~7(根据系统需求，参考端口复用表)
c) 设置寄存器 PxDIR[n](根据系统需求)
d) 设置寄存器 PxPU[n]/PxPD[n]/PxOD[n](根据系统需求)

6.4.4 端口复用配置为调试测试端口操作流程

参考测试调试相关章节。

6.4.5 端口复用配置为红外输出信号操作流程

端口 PA13,PB09,PC04 可以把频率为 38K 的内部时钟信号调制成红外信号输出。

a) 设置寄存器 PAADS[13]/PBADS[9]/PCADS[4]为 0
b) 设置寄存器 PA13_SEL = 1/PB09_SEL = 2/PC04_SEL = 3
c) 设置寄存器 PADIR[13] = 0/PBDIR[9] = 0/PCDIR[4] = 0：端口方向为输出
d) 设置寄存器 GPIO_CTRL1 的 bit14 选择红外信号输出极性
e) 设置寄存器 PAOUT[13]/PBOUT[9]/PCOUT[4]门控红外信号的输出

6.4.6 端口高电平中断操作流程

a) 设置寄存器 PxADS[n]为 0
b) 设置寄存器 Px_SEL 为 0
c) 设置寄存器 PxDIR[n]为 1
d) 设置寄存器 Px_HIE[n]为 1

e) 中断触发后读取中断状态寄存器 Px_STAT[n]

f) 设置寄存器 Px_ICLR[n]为 0 清除中断状态寄存器 Px_STAT[n]

6.4.7 端口低电平中断操作流程

a) 设置寄存器 PxADS[n]为 0

b) 设置寄存器 Px_SEL 为 0

c) 设置寄存器 PxDIR[n]为 1

d) 设置寄存器 Px_ LIE[n]为 1

e) 中断触发后读取中断状态寄存器 Px_STAT[n]

f) 设置寄存器 Px_ICLR[n]为 0 清除中断状态寄存器 Px_STAT[n]

6.4.8 端口上升沿中断操作流程

a) 设置寄存器 PxADS[n]为 0

b) 设置寄存器 Px_SEL 为 0

c) 设置寄存器 PxDIR[n]为 1

d) 设置寄存器 Px_ RIE[n]为 1

e) 中断触发后读取中断状态寄存器 Px_STAT[n]

f) 设置寄存器 Px_ICLR[n]为 0 清除中断状态寄存器 Px_STAT[n]

6.4.9 端口下降沿中断操作流程

a) 设置寄存器 PxADS[n]为 0

b) 设置寄存器 Px_SEL 为 0

c) 设置寄存器 PxDIR[n]为 1

d) 设置寄存器 Px_ FIE[n]为 1

e) 中断触发后读取中断状态寄存器 Px_STAT[n]

f) 设置寄存器 Px_ICLR[n]为 0 清除中断状态寄存器 Px_STAT[n]

6.4.10 端口上拉使能配置操作流程

a) 设置寄存器 PxPU[n]为 1

6.4.11 端口下拉使能配置操作流程

a) 设置寄存器 PxPU[n]为 0

b) 设置寄存器 PxPD[n]为 1

注：当 PxPU[n]，PxPD[n]同时置 1 时，PxPU[n]优先级高，PxPD[n]无效。

6.4.12 端口增强驱动配置操作流程

a) 设置寄存器 PxDR[n]为 0

6.4.13 端口开漏输出配置操作流程

a) 设置寄存器 PxOD[n]为 1

6.4.14 端口位置位操作流程

a) 设置寄存器 PxBSET[n]为 1

6.4.15 端口位清零操作流程

a) 设置寄存器 PxBCLR[n]为 1

6.4.16 端口位置位清零操作流程

a) 设置寄存器 PxBSETCLR[n]为 1

6.5 端口控制器寄存器描述

寄存器列表 1　　基地址 1：0x40020C00

偏移量	寄存器名称	访问	寄存器描述
0x00	PA00_SEL	RW	端口 PA00 功能配置寄存器
0x04	PA01_SEL	RW	端口 PA01 功能配置寄存器
0x08	PA02_SEL	RW	端口 PA02 功能配置寄存器
0x0c	PA03_SEL	RW	端口 PA03 功能配置寄存器
0x10	PA04_SEL	RW	端口 PA04 功能配置寄存器
0x14	PA05_SEL	RW	端口 PA05 功能配置寄存器
0x18	PA06_SEL	RW	端口 PA06 功能配置寄存器
0x1c	PA07_SEL	RW	端口 PA07 功能配置寄存器
0x20	PA08_SEL	RW	端口 PA08 功能配置寄存器
0x24	PA09_SEL	RW	端口 PA09 功能配置寄存器
0x28	PA10_SEL	RW	端口 PA10 功能配置寄存器
0x2c	PA11_SEL	RW	端口 PA11 功能配置寄存器
0x30	PA12_SEL	RW	端口 PA12 功能配置寄存器
0x34	PA13_SEL	RW	端口 PA13 功能配置寄存器
0x38	PA14_SEL	RW	端口 PA14 功能配置寄存器
0x3c	PA15_SEL	RW	端口 PA15 功能配置寄存器
0x40	PB00_SEL	RW	端口 PB00 功能配置寄存器
0x44	PB01_SEL	RW	端口 PB01 功能配置寄存器
0x48	PB02_SEL	RW	端口 PB02 功能配置寄存器
0x4c	PB03_SEL	RW	端口 PB03 功能配置寄存器
0x50	PB04_SEL	RW	端口 PB04 功能配置寄存器
0x54	PB05_SEL	RW	端口 PB05 功能配置寄存器
0x58	PB06_SEL	RW	端口 PB06 功能配置寄存器
0x5c	PB07_SEL	RW	端口 PB07 功能配置寄存器
0x60	PB08_SEL	RW	端口 PB08 功能配置寄存器
0x64	PB09_SEL	RW	端口 PB09 功能配置寄存器
0x68	PB10_SEL	RW	端口 PB10 功能配置寄存器
0x6c	PB11_SEL	RW	端口 PB11 功能配置寄存器
0x70	PB12_SEL	RW	端口 PB12 功能配置寄存器
0x74	PB13_SEL	RW	端口 PB13 功能配置寄存器
0x78	PB14_SEL	RW	端口 PB14 功能配置寄存器
0x7c	PB15_SEL	RW	端口 PB15 功能配置寄存器
0x80	PC00_SEL	RW	端口 PC00 功能配置寄存器
0x84	PC01_SEL	RW	端口 PC01 功能配置寄存器
0x88	PC02_SEL	RW	端口 PC02 功能配置寄存器

偏移量	寄存器名称	访问	寄存器描述
0x8c	PC03_SEL	RW	端口 PC03 功能配置寄存器
0x90	PC04_SEL	RW	端口 PC04 功能配置寄存器
0x94	PC05_SEL	RW	端口 PC05 功能配置寄存器
0x98	PC06_SEL	RW	端口 PC06 功能配置寄存器
0x9c	PC07_SEL	RW	端口 PC07 功能配置寄存器
0xa0	PC08_SEL	RW	端口 PC08 功能配置寄存器
0xa4	PC09_SEL	RW	端口 PC09 功能配置寄存器
0xa8	PC10_SEL	RW	端口 PC10 功能配置寄存器
0xac	PC11_SEL	RW	端口 PC11 功能配置寄存器
0xb0	PC12_SEL	RW	端口 PC12 功能配置寄存器
0xb4	PC13_SEL	RW	端口 PC13 功能配置寄存器
0xb8	PC14_SEL	RW	端口 PC14 功能配置寄存器
0xbc	PC15_SEL	RW	端口 PC15 功能配置寄存器
0xc0	PD00_SEL	RW	端口 PD00 功能配置寄存器
0xc4	PD01_SEL	RW	端口 PD01 功能配置寄存器
0xc8	PD02_SEL	RW	端口 PD02 功能配置寄存器
0xcc	PD03_SEL	RW	端口 PD03 功能配置寄存器
0xd0	PD04_SEL	RW	端口 PD04 功能配置寄存器
0xd4	PD05_SEL	RW	端口 PD05 功能配置寄存器
0xd8	PD06_SEL	RW	端口 PD06 功能配置寄存器
0xdc	PD07_SEL	RW	端口 PD07 功能配置寄存器
0xe0	PD08_SEL	RW	端口 PD08 功能配置寄存器
0xe4	PD09_SEL	RW	端口 PD09 功能配置寄存器
0xe8	PD10_SEL	RW	端口 PD10 功能配置寄存器
0xec	PD11_SEL	RW	端口 PD11 功能配置寄存器
0xf0	PD12_SEL	RW	端口 PD12 功能配置寄存器
0xf4	PD13_SEL	RW	端口 PD13 功能配置寄存器
0xf8	PD14_SEL	RW	端口 PD14 功能配置寄存器
0xfc	PD15_SEL	RW	端口 PD15 功能配置寄存器
0x100	PADIR	RW	端口 PA 输入输出配置寄存器
0x104	PAIN	RO	端口 PA 输入值寄存器
0x108	PAOUT	RW	端口 PA 输出值配置寄存器
0x10c	PAADS	RW	端口 PA 数模配置寄存器
0x110	PABSET	RW	端口 PA 位置位寄存器
0x114	PABCLR	RW	端口 PA 位清零寄存器
0x118	PABSETCLR	RW	端口 PA 位置位清零寄存器
0x11c	PADR	RW	端口 PA 驱动能力配置寄存器
0x120	PAPU	RW	端口 PA 上拉使能配置寄存器

偏移量	寄存器名称	访问	寄存器描述
0x124	PAPD	RW	端口 PA 下拉使能配置寄存器
0x12c	PAOD	RW	端口 PA 开漏输出配置寄存器
0x130	PAHIE	RW	端口 PA 高电平中断使能配置寄存器
0x134	PALIE	RW	端口 PA 低电平中断使能配置寄存器
0x138	PARIE	RW	端口 PA 上升沿中断使能配置寄存器
0x13c	PAFIE	RW	端口 PA 下降沿中断使能配置寄存器
0x200	PA_STAT	RO	端口 PA 中断状态寄存器
0x210	PA_ICLR	RW	端口 PA 中断清除寄存器
0x140	PBDIR	RW	端口 PB 输入输出配置寄存器
0x144	PBIN	RO	端口 PB 输入值寄存器
0x148	PBOUT	RW	端口 PB 输出值配置寄存器
0x14c	PBADS	RW	端口 PB 数模配置寄存器
0x150	PBBSET	RW	端口 PB 位置位寄存器
0x154	PBBCLR	RW	端口 PB 位清零寄存器
0x158	PBBSETCLR	RW	端口 PB 位置位清零寄存器
0x15c	PBDR	RW	端口 PB 驱动能力配置寄存器
0x160	PBPU	RW	端口 PB 上拉使能配置寄存器
0x164	PBPD	RW	端口 PB 下拉使能配置寄存器
0x16c	PBOD	RW	端口 PB 开漏输出配置寄存器
0x170	PBHIE	RW	端口 PB 高电平中断使能配置寄存器
0x174	PBLIE	RW	端口 PB 低电平中断使能配置寄存器
0x178	PBRIE	RW	端口 PB 上升沿中断使能配置寄存器
0x17c	PBFIE	RW	端口 PB 下降沿中断使能配置寄存器
0x240	PB_STAT	RO	端口 PB 中断状态寄存器
0x250	PB_ICLR	RW	端口 PB 中断清除寄存器
0x180	PCDIR	RW	端口 PC 输入输出配置寄存器
0x184	PCIN	RO	端口 PC 输入值寄存器
0x188	PCOUT	RW	端口 PC 输出值配置寄存器
0x18c	PCADS	RW	端口 PC 数模配置寄存器
0x190	PCBSET	RW	端口 PC 位置位寄存器
0x194	PCBCLR	RW	端口 PC 位清零寄存器
0x198	PCBSETCLR	RW	端口 PC 位置位清零寄存器
0x19c	PCDR	RW	端口 PC 驱动能力配置寄存器
0x1a0	PCPU	RW	端口 PC 上拉使能配置寄存器
0x1a4	PCPD	RW	端口 PC 下拉使能配置寄存器
0x1ac	PCOD	RW	端口 PC 开漏输出配置寄存器
0x1b0	PCHIE	RW	端口 PC 高电平中断使能配置寄存器
0x1b4	PCLIE	RW	端口 PC 低电平中断使能配置寄存器

偏移量	寄存器名称	访问	寄存器描述
0x1b8	PCRIE	RW	端口 PC 上升沿中断使能配置寄存器
0x1bc	PCFIE	RW	端口 PC 下降沿中断使能配置寄存器
0x280	PC_STAT	RO	端口 PC 中断状态寄存器
0x290	PC_ICLR	RW	端口 PC 中断清除寄存器
0x1c0	PDDIR	RW	端口 PD 输入输出配置寄存器
0x1c4	PDIN	RO	端口 PD 输入值寄存器
0x1c8	PDOUT	RW	端口 PD 输出值配置寄存器
0x1cc	PDADS	RW	端口 PD 数模配置寄存器
0x1d0	PDBSET	RW	端口 PD 位置位寄存器
0x1d4	PDBCLR	RW	端口 PD 位清零寄存器
0x1d8	PDBSETCLR	RW	端口 PD 位置位清零寄存器
0x1dc	PDDR	RW	端口 PD 驱动能力配置寄存器
0x1e0	PDPU	RW	端口 PD 上拉使能配置寄存器
0x1e4	PDPD	RW	端口 PD 下拉使能配置寄存器
0x1ec	PDOD	RW	端口 PD 开漏输出配置寄存器
0x1f0	PDHIE	RW	端口 PD 高电平中断使能配置寄存器
0x1f4	PDLIE	RW	端口 PD 低电平中断使能配置寄存器
0x1f8	PDRIE	RW	端口 PD 上升沿中断使能配置寄存器
0x1fc	PDFIE	RW	端口 PD 下降沿中断使能配置寄存器
0x2c0	PD_STAT	RO	端口 PD 中断状态寄存器
0x2d0	PD_ICLR	RW	端口 PD 中断清除寄存器
0x304	GPIO_CTRL1	RW	端口辅助功能配置寄存器 1
0x308	GPIO_CTRL2	RW	端口辅助功能配置寄存器 2
0x30c	GPIO_TIMGS	RW	端口辅助功能定时器门控选择
0x310	GPIO_TIMES	RW	端口辅助功能定时器 ETR 选择
0x314	GPIO_TIMCPS	RW	端口辅助功能定时器捕获输入选择
0x318	GPIO_PCAS	RW	端口辅助功能 PCA 捕获选择
0x1000	PE00_SEL	RW	端口 PE00 功能配置寄存器
0x1004	PE01_SEL	RW	端口 PE01 功能配置寄存器
0x1008	PE02_SEL	RW	端口 PE02 功能配置寄存器
0x100c	PE03_SEL	RW	端口 PE03 功能配置寄存器
0x1010	PE04_SEL	RW	端口 PE04 功能配置寄存器
0x1014	PE05_SEL	RW	端口 PE05 功能配置寄存器
0x1018	PE06_SEL	RW	端口 PE06 功能配置寄存器
0x101c	PE07_SEL	RW	端口 PE07 功能配置寄存器
0x1020	PE08_SEL	RW	端口 PE08 功能配置寄存器
0x1024	PE09_SEL	RW	端口 PE09 功能配置寄存器
0x1028	PE10_SEL	RW	端口 PE10 功能配置寄存器

偏移量	寄存器名称	访问	寄存器描述
0x102c	PE11_SEL	RW	端口 PE11 功能配置寄存器
0x1030	PE12_SEL	RW	端口 PE12 功能配置寄存器
0x1034	PE13_SEL	RW	端口 PE13 功能配置寄存器
0x1038	PE14_SEL	RW	端口 PE14 功能配置寄存器
0x103c	PE15_SEL	RW	端口 PE15 功能配置寄存器
0x1040	PF00_SEL	RW	端口 PF00 功能配置寄存器
0x1044	PF01_SEL	RW	端口 PF01 功能配置寄存器
0x1048	PF02_SEL	RW	端口 PF02 功能配置寄存器
0x104c	PF03_SEL	RW	端口 PF03 功能配置寄存器
0x1050	PF04_SEL	RW	端口 PF04 功能配置寄存器
0x1054	PF05_SEL	RW	端口 PF05 功能配置寄存器
0x1058	PF06_SEL	RW	端口 PF06 功能配置寄存器
0x105c	PF07_SEL	RW	端口 PF07 功能配置寄存器
0x1064	PF09_SEL	RW	端口 PF09 功能配置寄存器
0x1068	PF10_SEL	RW	端口 PF10 功能配置寄存器
0x106c	PF11_SEL	RW	端口 PF11 功能配置寄存器
0x1100	PEDIR	RW	端口 PE 输入输出配置寄存器
0x1104	PEIN	RO	端口 PE 输入值寄存器
0x1108	PEOUT	RW	端口 PE 输出值配置寄存器
0x110c	PEADS	RW	端口 PE 数模配置寄存器
0x1110	PEBSET	RW	端口 PE 位置位寄存器
0x1114	PEBCLR	RW	端口 PE 位清零寄存器
0x1118	PEBSETCLR	RW	端口 PE 位置位清零寄存器
0x111c	PEDR	RW	端口 PE 驱动能力配置寄存器
0x1120	PEPU	RW	端口 PE 上拉使能配置寄存器
0x1124	PEPD	RW	端口 PE 下拉使能配置寄存器
0x112c	PEOD	RW	端口 PE 开漏输出配置寄存器
0x1130	PEHIE	RW	端口 PE 高电平中断使能配置寄存器
0x1134	PELIE	RW	端口 PE 低电平中断使能配置寄存器
0x1138	PERIE	RW	端口 PE 上升沿中断使能配置寄存器
0x113c	PEFIE	RW	端口 PE 下降沿中断使能配置寄存器
0x1200	PE_STAT	RO	端口 PE 中断状态寄存器
0x1210	PE_ICLR	RW	端口 PE 中断清除寄存器
0x1140	PFDIR	RW	端口 PF 输入输出配置寄存器
0x1144	PFIN	RO	端口 PF 输入值寄存器
0x1148	PFOUT	RW	端口 PF 输出值配置寄存器
0x114c	PFADS	RW	端口 PF 数模配置寄存器
0x1150	PFBSET	RW	端口 PF 位置位寄存器

偏移量	寄存器名称	访问	寄存器描述
0x1154	PFBCLR	RW	端口 PF 位清零寄存器
0x1158	PFBSETCLR	RW	端口 PF 位置位清零寄存器
0x115c	PFDR	RW	端口 PF 驱动能力配置寄存器
0x1160	PFPU	RW	端口 PF 上拉使能配置寄存器
0x1164	PFPD	RW	端口 PF 下拉使能配置寄存器
0x116c	PFOD	RW	端口 PF 开漏输出配置寄存器
0x1170	PFHIE	RW	端口 PF 高电平中断使能配置寄存器
0x1174	PFLIE	RW	端口 PF 低电平中断使能配置寄存器
0x1178	PFRIE	RW	端口 PF 上升沿中断使能配置寄存器
0x117c	PFFIE	RW	端口 PF 下降沿中断使能配置寄存器
0x1240	PF_STAT	RO	端口 PF 中断状态寄存器
0x1250	PF_ICLR	RW	端口 PF 中断清除寄存器

6.5.1 端口通用寄存器

6.5.1.1 端口 Px 输入输出配置寄存器(PxDIR) (x = A,B,C,D,E,F)

地址偏移量：

0x100(PA),0x140(PB),0x180(PC),0x1C0(PD),0x1100(PE),0x1140(PF)

复位值：0xffff ffff

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxDIR[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxDIR	端口Px输入输出配置寄存器（对应Px15-Px00） 1：配置成输入　　0：配置成输出 注：每个bit对应一个端口，例如：PxDIR[15]对应端口Px15

6.5.1.2 端口 Px 输入值寄存器(PxIN) (x = A,B,C,D,E,F)

地址偏移量：

0x104(PA),0x144(PB),0x184(PC),0x1C4(PD),0x1104(PE),0x1144(PF)

复位值：NA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxIN[15:0]															
ro															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxIN	端口Px输入值寄存器（对应Px15-Px00） 1：输入为高电平 0：输入为低电平

6.5.1.3 端口 Px 输出值配置寄存器(PxOUT) (x = A,B,C,D,E,F)

地址偏移量：

0x108(PA),0x148(PB),0x188(PC),0x1C8(PD),0x1108(PE),0x1148(PF)

复位值：NA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxOUT[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxOUT	端口 Px 输出值配置寄存器（对应 Px15-Px00） 1：输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0：输出低电平。

6.5.1.4 端口 Px 数模配置寄存器(PxADS) (x = A,B,C,D,E,F)

地址偏移量：

0x10C(PA),0x14C(PB),0x18C(PC),0x1CC(PD),0x110C(PE),0x114C(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxADS[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxADS	端口Px数模配置寄存器（对应Px15-Px00） 1：配置为模拟端口 0：配置为数字端口

6.5.1.5 端口 Px 位置位寄存器(PxBSET) (x = A,B,C,D,E,F)

地址偏移量：

0x110(PA),0x150(PB),0x190(PC),0x1D0(PD),0x1110(PE),0x1150(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxBSET[15:0]															
W1															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxBSET	端口Px位置位寄存器（对应Px15-Px00） 1：置位 0：保持

6.5.1.6 端口 Px 位清零寄存器(PxBCLR) (x = A,B,C,D,E,F)

地址偏移量：

0x114(PA),0x154(PB),0x194(PC),0x1D4(PD),0x1114(PE),0x1154(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxBCLR[15:0]															
W1															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxBCLR	端口Px位清零寄存器（对应Px15-Px00） 1：清零 0：保持

6.5.1.7 端口 Px 位置位清零寄存器(PxBSETCLR) (x = A,B,C,D,E,F)

地址偏移量：

0x118(PA),0x158(PB),0x198(PC),0x1D8(PD),0x1118(PE),0x1158(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PxBSET[15:0]															
W1															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxBCLR[15:0]															
W1															

位	标记	功能描述
31:16	PxBSET	端口Px位置位寄存器 1：置位 0：保持
15:0	PxBCLR	端口Px位清零寄存器 1：清零 0：保持

注意：

- PxBSET 和 PxBCLR 相同位同时置 1 时，PxBCLR 具有高优先级。即该端口被清零。

6.5.1.8 端口 Px 上拉使能配置寄存器(PxPU) (x = A,B,C,D,E,F)

地址偏移量：

0x120(PA),0x160(PB),0x1A0(PC),0x1E0(PD),0x1120(PE),0x1160(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	标记	功能描述
31:16	Reserved	保留
15:0	PxPU	端口Px上拉使能配置寄存器 1：使能 0：禁止

6.5.1.9 端口 Px 驱动能力配置寄存器(PxDR) (x = A,B,C,D,E,F)

地址偏移量：

0x11C(PA),0x15C(PB),0x19C(PC),0x1DC(PD),0x111C(PE),0x115C(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxDR[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxDR	端口Px驱动能力配置寄存器 1：低驱动能力 0：高驱动能力

6.5.1.10 端口 Px 下拉使能配置寄存器(PxPD) (x = A,B,C,D,E,F)

地址偏移量：

0x124(PA),0x164(PB),0x1A4(PC),0x1E4(PD),0x1124(PE),0x1164(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxPD[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxPD	端口Px下拉使能配置寄存器 1：使能 0：禁止

6.5.1.11 端口 Px 开漏输出配置寄存器(PxOD) (x = A,B,C,D,E,F)

地址偏移量：

0x12C(PA),0x16C(PB),0x1AC(PC),0x1EC(PD),0x112C(PE),0x116C(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxOD[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxOD	端口Px开漏输出配置寄存器 1：将端口输出方式设置为开漏输出 0：将端口输出方式设置为推挽输出

6.5.1.12 端口 Px 高电平中断使能配置寄存器(PxHIE) (x = A,B,C,D,E,F)

地址偏移量：

0x130(PA),0x170(PB),0x1B0(PC),0x1F0(PD),0x1130(PE),0x1170(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxHIE[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxHIE	端口Px高电平中断使能配置寄存器 1：使能 0：禁止

6.5.1.13 端口 Px 低电平中断使能配置寄存器(PxLIE) (x = A,B,C,D,E,F)

地址偏移量：

0x134(PA),0x174(PB),0x1B4(PC),0x1F4(PD),0x1134(PE),0x1174(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxLIE[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxLIE	端口Px低电平中断使能配置寄存器 1：使能 0：禁止

6.5.1.14 端口 Px 上升沿中断使能配置寄存器(PxRIE) (x = A,B,C,D,E,F)

地址偏移量：

0x138(PA),0x178(PB),0x1B8(PC),0x1F8(PD),0x1138(PE),0x1178(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxRIE[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxRIE	端口Px上升沿中断使能配置寄存器 1：使能 0：禁止

6.5.1.15 端口 Px 下降沿中断使能配置寄存器(PxFIE) (x = A,B,C,D,E,F)

地址偏移量：

0x13C(PA),0x17C(PB),0x1BC(PC),0x1FC(PD),0x113C(PE),0x117C(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxFIE[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxFIE	端口Px下降沿中断使能配置寄存器 1：使能 0：禁止

6.5.1.16 端口 Px 中断状态寄存器(Px_STAT) (x = A,B,C,D,E,F)

地址偏移量：

0x200(PA),0x240(PB),0x280(PC),0x2C0(PD),0x1200(PE),0x1240(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Px_STAT[15:0]															
ro															

位	标记	功能描述
31:16	Reserved	保留
15:0	Px_STAT	端口Px中断状态寄存器 1：中断触发 0：无中断触发

6.5.1.17 端口 Px 中断清除寄存器(Px_ICLR) (x = A,B,C,D,E,F)

地址偏移量：

0x210(PA),0x250(PB),0x290(PC),0x2D0(PD),0x1210(PE),0x1250(PF)

复位值：0xffff ffff

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Px_ICLR[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	Px_ICLR	端口Px中断清除寄存器 1：保留中断标志位 0：清除中断标志位

6.5.2 端口辅助寄存器

6.5.2.1 端口辅助功能配置寄存器 1(GPIO_CTRL1)

地址偏移量：0x304

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	ir_pol	hclk_en	pclk_en	hclk_sel		pclk_sel		ssn0_sel				ext_clk_sel			
	rw	rw	rw	rw		rw		rw				rw			

位	标记	功能描述
31:15	Reserved	保留
14	ir_pol	IR输出极性选择. 0 – 正向输出 1 – 反向输出
13	hclk_en	HCLK输出控制 0 – 输出0 1 – 输出HCLK
12	pclk_en	PCLK输出门控. 0 – 输出0 1 – 输出PCLK
11:10	hclk_sel	HCLK输出分频选择. 00：HCLK 01：HCLK/2 10：HCLK/4 11：HCLK/8
9:8	pclk_sel	PCLK输出分频选择. 00：PCLK 01：PCLK/2 10：PCLK/4 11：PCLK/8
7:4	ssn0_sel	SPI0 SSN信号来源选择. 0000: 高电平 1000: PB01 0001: PA03 1001:PB02 0010: PA04 1010:PB05 0011: PA06 1011:PB06 0100: PA08 1100:PB09 0101: PA09 1101:PB10 0110: PA12 1110:PB12 0111: PA15 1111:PB14
3:0	ext_clk_sel	外部时钟信号来源选择. 0000: 高电平 1000: PB01 0001: PA03 1001: PB02

		0010: PA04 0011: PA06 0100: PA08 0101: PA09 0110: PA12 0111: PA15	1010: 1011: 1100: 1101: 1110: 1111:	PB05 PB06 PB09 PB10 PB12 PB14

6.5.2.2 端口辅助功能配置寄存器 2(GPIO_CTRL2)

地址偏移量：0x308

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ahb_sel	Res							tclk_div		tclk_sel		ssn1_sel			
rw								rw		rw		rw			

位	标记	功能描述
31:4	Reserved	保留
15	ahb_sel	AHB总线访问模式 0 – 使用FASTIO总线访问PxIN,PxOUT寄存器 1 -- 使用AHB总线访问PxIN,PxOUT寄存器
14:8	Reserved	保留
7:6	tclk_div	TCLK输出分频选择. 00: TCLK 01: TCLK/2 10: TCLK/4 11: TCLK/8
5:4	tclk_sel	tclk输出信号源选择. 0 0 RCH48M 0 1 RCH 1 0 XTH 1 1 PLL
3:0	ssn1_sel	SPI1 SSN信号来源选择. 0000 高电平 0001 PA03 0010 PA04 0011 PA06 0100 PA08 0101 PA09 0110 PA12 0111 PA15 1000 PB01 1001 PB02 1010 PB05 1011 PB06 1100 PB09 1101 PB10 1110 PB12 1111 PB14

6.5.2.3 端口辅助功能定时器门控选择(GPIO_TIMGS)

地址偏移量：0x30C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TIM3_G			TIM2_G			TIM1_G			TIM0_G		
				R/W			R/W			R/W			R/W		

位	标记	功能描述
31:12	Reserved	保留
11:9	TIM3_G	Timer3 定时器GATE输入选择，选择见下表
8:6	TIM2_G	Timer2 定时器GATE输入选择，选择见下表
5:3	TIM1_G	Timer1 定时器GATE输入选择，选择见下表
2:0	TIM0_G	Timer0 定时器GATE输入选择，选择见下表

	TIM0_g	TIM1_g	TIM2_g	TIM3_g
000	PX_SEL	PX_SEL	PX_SEL	PX_SEL
001	UART0_RXD	LPUART0_RXD	UART0_RXD	UART0_RXD
010	UART1_RXD	LPUART1_RXD	UART1_RXD	UART1_RXD
011	VC0_OUT	VC0_OUT	VC0_OUT	UART2
100	VC1_OUT	VC1_OUT	VC1_OUT	UART3
101	PA03	PA08	PA10	VC0_OUT
110	PB08	PB03	PB04	PA06
111	PB15	PB13	PB11	PA11

6.5.2.4 端口辅助功能定时器 ETR 选择(GPIO_TIMES)

地址偏移量：0x310

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TIM3_E			TIM2_E			TIM1_E			TIM0_E		
				R/W			R/W			R/W			R/W		

位	标记	功能描述
31:12	Reserved	保留
11:9	TIM3_E	Timer3 定时器ETR输入选择，选择见下表
8:6	TIM2_E	Timer2 定时器ETR输入选择，选择见下表
5:3	TIM1_E	Timer1 定时器ETR输入选择，选择见下表
2:0	TIM0_E	Timer0 定时器ETR输入选择，选择见下表

	TIM0_e	TIM1_e	TIM2_e	TIM3_e
000	PX_SEL	PX_SEL	PX_SEL	PX_SEL
001	LPUART0_RXD	UART0_RXD	LPUART0_RXD	UART0_RXD
010	LPUART1_RXD	UART1_RXD	LPUART1_RXD	UART1_RXD
011	VC0_OUT	VC1_OUT	VC0_OUT	VC1_OUT
100	LVD_OUT	LVD_OUT	Reserved	Reserved
101	PA00	PA01	PA04	PA00
110	PA05	PC09	PC04	PA12
111	PA15	PD02	PC08	PA13

6.5.2.5 端口辅助功能定时器捕获输入选择(GPIO_TIMCPS)

地址偏移量：0x314

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TIM3_CB			TIM3_CA			TIM2_CA			TIM1_CA			TIM0_CA		
	R/W			R/W			R/W			R/W			R/W		

位	标记	功能描述
31:15	Reserved	保留
14:12	TIM3_CB	Timer3 定时器CH0B输入选择，选择见下表
11:9	TIM3_CA	Timer3 定时器CH0A输入选择，选择见下表
8:6	TIM2_CA	Timer2 定时器CHA输入选择，选择见下表
5:3	TIM1_CA	Timer1 定时器CHA输入选择，选择见下表
2:0	TIM0_CA	Timer0 定时器CHA输入选择，选择见下表

	TIM0_CHA	TIM1_CHA	TIM2_CHA	TIM3_CH0A	TIM3_CH0B
000	PX_SEL	PX_SEL	PX_SEL	PX_SEL	PX_SEL
001	UART0_RXD	UART1_RXD	LPUART0_RXD	LPUART1_RXD	UART0_RXD
010	PA00	PA00	VC0_OUT	LPUART0_RXD	UART1_RXD
011	PA02	PA02	PA02	Reserved	Reserved
100	PA05	PA06	PA07	VC0_OUT	VC1_OUT
101	PA15	PB08	PB08	PA08	PA07
110	PB06	PB10	PB09	PB03	PB04
111	PB14	PB13	PC06	PB06	PB13

6.5.2.6 端口辅助功能 PCA 捕获选择(GPIO_PCAS)

地址偏移量：0x318

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										PCA_CH0			PCA_ECI		
										R/W			R/W		

位	标记	功能描述
31:6	Reserved	保留
5:3	PCA_ECI	PCA ECI时钟输入选择，选择见下表
2:0	PCA_CH0	PCA CH0 捕获口输入选择，选择见下表

	pca_eci	pca_ch0
000	PX_SEL	PX_SEL
001	Reserved	Reserved
010	LVD_OUT	Reserved
011	VC0_OUT	LVD_OUT
100	VC1_OUT	VC1_OUT
101	PA05	PA06
110	PB02	PB04
111	PD02	PC06

6.5.3 端口 PA 功能选择寄存器

6.5.3.1 端口 PA00 功能配置寄存器(PA00_SEL)

地址偏移量：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA00_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA00_sel	端口PA00功能选择. 000 ---- GPIO PA00 001 ---- UART1_CTS UART1模块CTS信号 010 ---- LPUART1_TXD LPUART1模块TXD信号 011 ---- TIM0_ETR Timer0模块外部时钟输入信号 100 ---- VC0_OUT VC0模块输出/反向输出信号 101 ---- TIM1_CHA Timer1模块通道A信号 110 ---- TIM3_ETR Timer3模块外部时钟输入信号 111 ---- TIM0_CHA Timer0模块通道A信号

6.5.3.2 端口 PA01 功能配置寄存器(PA01_SEL)

地址偏移量：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA01_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA01_sel	端口PA01功能选择. 000 ---- GPIO PA01 001 ---- UART1_RTS UART1模块RTS信号 010 ---- LPUART1_RXD LPUART1模块RXD信号 011 ---- TIM0_CHB Timer0模块通道B信号 100 ---- TIM1_ETR Timer1模块外部时钟输入信号 101 ---- TIM1_CHB Timer1模块通道B信号 110 ---- HCLK_OUT AHB总线时钟输出信号 111 ---- SPI1_MOSI SPI1模块主机输出从机输入数据信号

6.5.3.3 端口 PA02 功能配置寄存器(PA02_SEL)

地址偏移量：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA02_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA02_sel	端口PA02功能选择. 000 ---- GPIO PA02 001 ---- UART1_TXD UART1模块TXD信号 010 ---- TIM0_CHA Timer0模块通道A信号 011 ---- VC1_OUT VC1模块输出/反向输出信号 100 ---- TIM1_CHA Timer1模块通道A信号 101 ---- TIM2_CHA Timer2模块通道A信号 110 ---- PCLK_OUT APB总线时钟输出信号 111 ---- SPI1_MISO SPI1模块主机输入从机输出数据信号

6.5.3.4 端口 PA03 功能配置寄存器(PA03_SEL)

地址偏移量：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA03_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA03_sel	端口PA03功能选择. 000 ---- GPIO PA03 001 ---- UART1_RXD UART1模块RXD信号 010 ---- TIM0_GATE Timer0模块门控信号 011 ---- TIM1_CHB Timer1模块通道B信号 100 ---- TIM2_CHB Timer2模块通道B信号 101 ---- SPI1_CS SPI1模块主机模式片选信号 110 ---- TIM3_CH1A Timer3模块通道1A信号 111 ---- TIM5_CHA Timer6模块通道1A信号

6.5.3.5 端口 PA04 功能配置寄存器(PA04_SEL)

地址偏移量：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA04_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA04_sel	端口PA04功能选择. 000 ---- GPIO PA04 001 ---- SPI0_CS SPI0模块主机模式片选信号 010 ---- UART1_TXD UART1模块TXD信号 011 ---- PCA_CH4 PCA模块通道4捕获/比较信号 100 ---- TIM2_ETR Timer2模块外部时钟输入信号 101 ---- TIM5_CHA Timer6模块通道1A信号 110 ---- LVD_OUT LVD模块输出信号 111 ---- TIM3_CH2B Timer3模块通道2B信号

6.5.3.6 端口 PA05 功能配置寄存器(PA05_SEL)

地址偏移量：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA05_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA05_sel	端口PA05功能选择. 000 ---- GPIO PA05 001 ---- SPI0_SCK SPI0模块时钟信号 010 ---- TIM0_ETR Timer0模块外部时钟输入信号 011 ---- PCA_ECI PCA模块外部时钟输入信号 100 ---- TIM0_CHA Timer0模块通道A信号 101 ---- TIM5_CHB Timer6模块通道1B信号 110 ---- XTL_OUT 外部32K晶振输出信号 111 ---- XTH_OUT 外部32M晶振输出信号

6.5.3.7 端口 PA06 功能配置寄存器(PA06_SEL)

地址偏移量：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA06_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA06_sel	端口PA06功能选择. 000 ---- GPIO PA06 001 ---- SPI0_MISO SPI0模块主机输入从机输出数据信号 010 ---- PCA_CH0 PCA模块通道0捕获/比较信号 011 ---- TIM3_BK Timer3模块刹车信号 100 ---- TIM1_CHA Timer1模块通道A信号 101 ---- VC0_OUT VC0模块输出/反向输出信号 110 ---- TIM3_GATE Timer3模块门控信号 111 ---- LPUART0_CTS LPUART0模块CTS信号

6.5.3.8 端口 PA07 功能配置寄存器(PA07_SEL)

地址偏移量：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA07_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA07_sel	端口PA07功能选择. 000 ---- GPIO PA07 001 ---- SPI0_MOSI SPI0模块主机输出从机输入数据信号 010 ---- PCA_CH1 PCA模块通道1捕获/比较信号 011 ---- HCLK_OUT AHB总线时钟输出信号 100 ---- TIM3_CH0B Timer3模块通道0B信号 101 ---- TIM2_CHA Timer2模块通道A信号 110 ---- VC1_OUT VC1模块输出/反向输出信号 111 ---- TIM4_CHB Timer6模块通道0B信号

6.5.3.9 端口 PA08 功能配置寄存器(PA08_SEL)

地址偏移量：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA08_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA08_sel	PA08_sel: 端口PA08功能选择. 000 ---- GPIO PA08 001 ---- UART0_TXD UART0模块TXD信号 010 ---- TIM3_CH0A Timer3模块通道0A信号 011 ---- CRS_SYNC TBD 100 ---- CAN_STBY TBD 101 ---- TIM1_GATE Timer1模块门控信号 110 ---- TIM4_CHA Timer6模块通道0A信号 111 ---- TIM3_BK Timer3模块刹车信号

6.5.3.10 端口 PA09 功能配置寄存器(PA09_SEL)

地址偏移量：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA09_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA09_sel	端口PA09功能选择. 000 ---- GPIO PA09 001 ---- UART0_TXD UART0模块TXD信号 010 ---- TIM3_CH1A Timer3模块通道1A信号 011 ---- TIM0_BK Timer0模块刹车信号 100 ---- I2C0_SCL I2C0模块时钟信号 101 ---- Reserved Reserved 110 ---- HCLK_OUT AHB总线时钟输出信号 111 ---- TIM5_CHA Timer6模块通道1A信号

6.5.3.11 端口 PA10 功能配置寄存器(PA10_SEL)

地址偏移量：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA10_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA10_sel	PA10_sel: 端口PA10功能选择. 000 ---- GPIO PA10 001 ---- UART0_RXD UART0模块RXD信号 010 ---- TIM3_CH2A Timer3模块通道2A信号 011 ---- TIM2_BK Timer2模块刹车信号 100 ---- I2C0_SDA I2C0模块数据信号 101 ---- TIM2_GATE Timer2模块门控信号 110 ---- PCLK_OUT APB总线时钟输出信号 111 ---- TIM6_CHA Timer6模块通道2A信号

6.5.3.12 端口 PA11 功能配置寄存器(PA11_SEL)

地址偏移量：0x2C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA11_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA11_sel	PA11_sel: 端口PA11功能选择. 000 ---- GPIO PA11 001 ---- UART0_CTS UART0模块CTS信号 010 ---- TIM3_GATE Timer3模块门控信号 011 ---- I2C1_SCL I2C1模块时钟信号 100 ---- CAN_RX CAN模块接收信号 101 ---- VC0_OUT VC0模块输出/反向输出信号 110 ---- SPI0_MISO SPI0模块主机输入从机输出数据信号 111 ---- TIM4_CHB Timer6模块通道0B信号

6.5.3.13 端口 PA12 功能配置寄存器(PA12_SEL)

地址偏移量：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA12_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA12_sel	PA12_sel: 端口PA12功能选择. 000 ---- GPIO PA12 001 ---- UART0_RTS UART0模块RTS信号 010 ---- TIM3_ETR Timer3模块外部时钟输入信号 011 ---- I2C1_SDA I2C1模块数据信号 100 ---- CAN_TX CAN模块发送信号 101 ---- VC1_OUT VC1模块输出/反向输出信号 110 ---- SPI0_MOSI SPI0模块主机输出从机输入数据信号 111 ---- Reserved Reserved

6.5.3.14 端口 PA13 功能配置寄存器(PA13_SEL)

地址偏移量：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA13_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA13_sel	PA13_sel: 端口PA13功能选择. 000 ---- GPIO PA13 001 ---- IR_OUT 红外输出信号 010 ---- UART0_RXD UART0模块RXD信号 011 ---- LVD_OUT LVD模块输出信号 100 ---- TIM3_ETR Timer3模块外部时钟输入信号 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- VC2_OUT VC2模块输出/反向输出信号

6.5.3.15 端口 PA14 功能配置寄存器(PA14_SEL)

地址偏移量：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA14_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA14_sel	PA14_sel: 端口PA14功能选择. 000 ---- GPIO PA14 001 ---- UART1_TXD UART1模块TXD信号 010 ---- UART0_TXD UART0模块TXD信号 011 ---- TIM3_CH2A Timer3模块通道2A信号 100 ---- LVD_OUT LVD模块输出信号 101 ---- RCH_OUT 内部24M RC时钟输出信号 110 ---- RCL_OUT 内部38K RC时钟输出信号 111 ---- PLL_OUT 内部PLL时钟输出信号

6.5.3.16 端口 PA15 功能配置寄存器(PA15_SEL)

地址偏移量：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA15_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA15_sel	PA15_sel: 端口PA15功能选择. 000 ---- GPIO PA15 001 ---- SPI0_CS SPI0模块主机模式片选信号 010 ---- UART1_RXD UART1模块RXD信号 011 ---- LPUART1_RTS LPUART1模块RTS信号 100 ---- TIM0_ETR Timer0模块外部时钟输入信号 101 ---- TIM0_CHA Timer0模块通道A信号 110 ---- TIM3_CH1A Timer3模块通道1A信号 111 ---- Reserved Reserved

6.5.4 端口 PB 功能选择寄存器

6.5.4.1 端口 PB00 功能配置寄存器(PB00_SEL)

地址偏移量：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB00_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB00_sel	端口PB00功能选择. 000 ---- GPIO PB00 001 ---- PCA_CH2 PCA模块通道2捕获/比较信号 010 ---- TIM3_CH1B Timer3模块通道1B信号 011 ---- LPUART0_TXD LPUART0模块TXD信号 100 ---- TIM5_CHB Timer6模块通道1B信号 101 ---- RCH_OUT 内部24M RC时钟输出信号 110 ---- RCL_OUT 内部38K RC时钟输出信号 111 ---- PLL_OUT 内部PLL时钟输出信号

6.5.4.2 端口 PB01 功能配置寄存器(PB01_SEL)

地址偏移量：0x44

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB01_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB01_sel	PB01_sel: 端口PB01功能选择. 000 ---- GPIO PB01 001 ---- PCA_CH3 PCA模块通道3捕获/比较信号 010 ---- PCLK_OUT APB总线时钟输出信号 011 ---- TIM3_CH2B Timer3模块通道2B信号 100 ---- TIM6_CHB Timer6模块通道2B信号 101 ---- LPUART0_RTS LPUART0模块RTS信号 110 ---- VC2_OUT VC2模块输出/反向输出信号 111 ---- TCLK_OUT 内部时钟输出信号

6.5.4.3 端口 PB02 功能配置寄存器(PB02_SEL)

地址偏移量：0x48

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB02_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB02_sel	PB02_sel: 端口PB02功能选择. 000 ---- GPIO PB02 001 ---- Reserved Reserved 010 ---- PCA_ECI PCA模块外部时钟输入信号 011 ---- LPUART1_TXD LPUART1模块TXD信号 100 ---- TIM4_CHA Timer6模块通道0A信号 101 ---- TIM1_BK Timer1模块刹车信号 110 ---- TIM0_BK Timer0模块刹车信号 111 ---- TIM2_BK Timer2模块刹车信号

6.5.4.4 端口 PB03 功能配置寄存器(PB03_SEL)

地址偏移量：0x4C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB03_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB03_sel	PB03_sel: 端口PB03功能选择. 000 ---- GPIO PB03 001 ---- SPI0_SCK SPI0模块时钟信号 010 ---- TIM0_CHB Timer0模块通道B信号 011 ---- TIM1_GATE Timer1模块门控信号 100 ---- TIM3_CH0A Timer3模块通道0A信号 101 ---- Reserved Reserved 110 ---- XTL_OUT 外部32K晶振输出信号 111 ---- XTH_OUT 外部32M晶振输出信号

6.5.4.5 端口 PB04 功能配置寄存器(PB04_SEL)

地址偏移量：0x50

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB04_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB04_sel	PB04_sel: 端口PB04功能选择. 000 ---- GPIO PB04 001 ---- SPI0_MISO SPI0模块主机输入从机输出数据信号 010 ---- PCA_CH0 PCA模块通道0捕获/比较信号 011 ---- TIM2_BK Timer2模块刹车信号 100 ---- UART0_CTS UART0模块CTS信号 101 ---- TIM2_GATE Timer2模块门控信号 110 ---- TIM3_CH0B Timer3模块通道0B信号 111 ---- Reserved Reserved

6.5.4.6 端口 PB05 功能配置寄存器(PB05_SEL)

地址偏移量：0x54

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB05_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB05_sel	PB05_sel: 端口PB05功能选择. 000 ---- GPIO PB05 001 ---- SPI0_MOSI SPI0模块主机输出从机输入数据信号 010 ---- Reserved Reserved 011 ---- TIM1_BK Timer1模块刹车信号 100 ---- PCA_CH1 PCA模块通道1捕获/比较信号 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- UART0_RTS UART0模块RTS信号

6.5.4.7 端口 PB06 功能配置寄存器(PB06_SEL)

地址偏移量：0x58

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB06_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB06_sel	PB06_sel: 端口PB06功能选择. 000 ---- GPIO PB06 001 ---- I2C0_SCL I2C0模块时钟信号 010 ---- UART0_TXD UART0模块TXD信号 011 ---- TIM1_CHB Timer1模块通道B信号 100 ---- TIM0_CHA Timer0模块通道A信号 101 ---- Reserved Reserved 110 ---- TIM3_CH0A Timer3模块通道0A信号 111 ---- Reserved Reserved

6.5.4.8 端口 PB07 功能配置寄存器(PB07_SEL)

地址偏移量：0x5C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB07_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB07_sel	PB07_sel: 端口PB07功能选择. 000 ---- GPIO PB07 001 ---- I2C0_SDA I2C0模块数据信号 010 ---- UART0_RXD UART0模块RXD信号 011 ---- TIM2_CHB Timer2模块通道B信号 100 ---- LPUART1_CTS LPUART1模块CTS信号 101 ---- TIM0_CHB Timer0模块通道B信号 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.4.9 端口 PB08 功能配置寄存器(PB08_SEL)

地址偏移量：0x60

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB08_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB08_sel	PB08_sel: 端口PB08功能选择. 000 ---- GPIO PB08 001 ---- I2C0_SCL I2C0模块时钟信号 010 ---- TIM1_CHA Timer1模块通道A信号 011 ---- CAN_RX CAN模块接收信号 100 ---- TIM2_CHA Timer2模块通道A信号 101 ---- TIM0_GATE Timer0模块门控信号 110 ---- TIM3_CH2A Timer3模块通道2A信号 111 ---- UART0_TXD UART0模块TXD信号

6.5.4.10 端口 PB09 功能配置寄存器(PB09_SEL)

地址偏移量：0x64

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB09_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB09_sel	PB09_sel: 端口PB09功能选择. 000 ---- GPIO PB09 001 ---- I2C0_SDA I2C0模块数据信号 010 ---- IR_OUT 红外输出信号 011 ---- SPI1_CS SPI1模块主机模式片选信号 100 ---- TIM2_CHA Timer2模块通道A信号 101 ---- CAN_TX CAN模块发送信号 110 ---- TIM2_CHB Timer2模块通道B信号 111 ---- UART0_RXD UART0模块RXD信号

6.5.4.11 端口 PB10 功能配置寄存器(PB10_SEL)

地址偏移量：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB10_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB10_sel	PB10_sel: 端口PB10功能选择. 000 ---- GPIO PB10 001 ---- I2C1_SCL I2C1模块时钟信号 010 ---- SPI1_SCK SPI1模块时钟信号 011 ---- TIM1_CHA Timer1模块通道A信号 100 ---- LPUART0_TXD LPUART0模块TXD信号 101 ---- TIM3_CH1A Timer3模块通道1A信号 110 ---- LPUART1_RTS LPUART1模块RTS信号 111 ---- UART1_RTS UART1模块RTS信号

6.5.4.12 端口 PB11 功能配置寄存器(PB11_SEL)

地址偏移量：0x6C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB11_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB11_sel	PB11_sel: 端口PB11功能选择. 000 ---- GPIO PB11 001 ---- I2C1_SDA I2C1模块数据信号 010 ---- TIM1_CHB Timer1模块通道B信号 011 ---- LPUART0_RXD LPUART0模块RXD信号 100 ---- TIM2_GATE Timer2模块门控信号 101 ---- TIM6_CHA Timer6模块通道2A信号 110 ---- LPUART1_CTS LPUART1模块CTS信号 111 ---- UART1_CTS UART1模块CTS信号

6.5.4.13 端口 PB12 功能配置寄存器(PB12_SEL)

地址偏移量：0x70

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB12_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB12_sel	PB12_sel: 端口PB12功能选择. 000 ---- GPIO PB12 001 ---- SPI1_CS SPI1模块主机模式片选信号 010 ---- TIM3_BK Timer3模块刹车信号 011 ---- LPUART0_TXD LPUART0模块TXD信号 100 ---- TIM0_BK Timer0模块刹车信号 101 ---- Reserved Reserved 110 ---- LPUART0_RTS LPUART0模块RTS信号 111 ---- TIM6_CHA Timer6模块通道2A信号

6.5.4.14 端口 PB13 功能配置寄存器(PB13_SEL)

地址偏移量：0x74

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB13_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB13_sel	PB13_sel: 端口PB13功能选择. 000 ---- GPIO PB13 001 ---- SPI1_SCK SPI1模块时钟信号 010 ---- I2C1_SCL I2C1模块时钟信号 011 ---- TIM3_CH0B Timer3模块通道0B信号 100 ---- LPUART0_CTS LPUART0模块CTS信号 101 ---- TIM1_CHA Timer1模块通道A信号 110 ---- TIM1_GATE Timer1模块门控信号 111 ---- TIM6_CHB Timer6模块通道2B信号

6.5.4.15 端口 PB14 功能配置寄存器(PB14_SEL)

地址偏移量：0x78

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB14_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB14_sel	PB14_sel: 端口PB14功能选择. 000 ---- GPIO PB14 001 ---- SPI1_MISO SPI1模块主机输入从机输出数据信号 010 ---- I2C1_SDA I2C1模块数据信号 011 ---- TIM3_CH1B Timer3模块通道1B信号 100 ---- TIM0_CHA Timer0模块通道A信号 101 ---- Reserved Reserved 110 ---- LPUART0_RTS LPUART0模块RTS信号 111 ---- TIM1_BK Timer1模块刹车信号

6.5.4.16 端口 PB15 功能配置寄存器(PB15_SEL)

地址偏移量：0x7C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB15_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB15_sel	PB15_sel: 端口PB15功能选择. 000 ---- GPIO PB15 001 ---- SPI1_MOSI SPI1模块主机输出从机输入数据信号 010 ---- TIM3_CH2B Timer3模块通道2B信号 011 ---- TIM0_CHB Timer0模块通道B信号 100 ---- TIM0_GATE Timer0模块门控信号 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- LPUART1_RXD LPUART1模块RXD信号

6.5.5 端口 PC 功能选择寄存器

6.5.5.1 端口 PC00 功能配置寄存器(PC00_SEL)

地址偏移量：0x80

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PC00_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PC00_sel	PC00_sel: 端口PC00功能选择. 000 ---- GPIO PC00 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- UART1_CTS UART1模块CTS信号 100 ---- UART2_RTS UART2模块RTS信号 101 ---- I2S0_MCK I2S0模块MCK信号 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.5.2 端口 PC01 功能配置寄存器(PC01_SEL)

地址偏移量：0x84

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PC01_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PC01_sel	PC01_sel: 端口PC01功能选择. 000 ---- GPIO PC01 001 ---- Reserved Reserved 010 ---- TIM5_CHB Timer6模块通道1B信号 011 ---- UART1_RTS UART1模块RTS信号 100 ---- Reserved Reserved 101 ---- I2S0_SD I2S0模块SD信号 110 ---- UART2_CTS UART2模块CTS信号 111 ---- Reserved Reserved

6.5.5.3 端口 PC02 功能配置寄存器(PC02_SEL)

地址偏移量：0x88

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PC02_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PC02_sel	PC02_sel: 端口PC02功能选择. 000 ---- GPIO PC02 001 ---- SPI1_MISO SPI1模块主机输入从机输出数据信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- UART2_RXD UART2模块RXD信号 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.5.4 端口 PC03 功能配置寄存器(PC03_SEL)

地址偏移量：0x8C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														PC03_sel	
														rw	

位	标记	功能描述
31:3	Reserved	保留
2:0	PC03_sel	PC03_sel: 端口PC03功能选择. 000 ---- GPIO PC03 001 ---- SPI1_MOSI SPI1模块主机输出从机输入数据信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- UART2_TXD UART2模块TXD信号 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.5.5 端口 PC04 功能配置寄存器(PC04_SEL)

地址偏移量：0x90

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														PC04_sel	
														rw	

位	标记	功能描述
31:3	Reserved	保留
2:0	PC04_sel	PC04_sel: 端口PC04功能选择. 000 ---- GPIO PC04 001 ---- LPUART0_TXD LPUART0模块TXD信号 010 ---- TIM2_ETR Timer2模块外部时钟输入信号 011 ---- IR_OUT 红外输出信号 100 ---- VC2_OUT VC2模块输出/反向输出信号 101 ---- I2S0_WS I2S0模块WS信号 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.5.6 端口 PC05 功能配置寄存器(PC05_SEL)

地址偏移量：0x94

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PC05_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PC05_sel	PC05_sel: 端口PC05功能选择. 000 ---- GPIO PC05 001 ---- LPUART0_RXD LPUART0模块RXD信号 010 ---- TIM6_CHB Timer6模块通道2B信号 011 ---- PCA_CH4 PCA模块通道4捕获/比较信号 100 ---- Reserved Reserved 101 ---- I2S0_SDIN I2S0模块SDIN信号 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.5.7 端口 PC06 功能配置寄存器(PC06_SEL)

地址偏移量：0x98

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PC06_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PC06_sel	PC06_sel: 端口PC06功能选择. 000 ---- GPIO PC06 001 ---- PCA_CH0 PCA模块通道0捕获/比较信号 010 ---- TIM4_CHA Timer6模块通道0A信号 011 ---- TIM2_CHA Timer2模块通道A信号 100 ---- Reserved Reserved 101 ---- I2S1_SCK I2S1模块SCK信号 110 ---- UART3_RXD UART3模块RXD信号 111 ---- Reserved Reserved

6.5.5.8 端口 PC07 功能配置寄存器(PC07_SEL)

地址偏移量：0x9C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PC07_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PC07_sel	PC07_sel: 端口PC07功能选择. 000 ---- GPIO PC07 001 ---- PCA_CH1 PCA模块通道1捕获/比较信号 010 ---- TIM5_CHA Timer6模块通道1A信号 011 ---- TIM2_CHB Timer2模块通道B信号 100 ---- Reserved Reserved 101 ---- I2S1_MCK I2S1模块MCK信号 110 ---- UART3_TXD UART3模块TXD信号 111 ---- Reserved Reserved

6.5.5.9 端口 PC08 功能配置寄存器(PC08_SEL)

地址偏移量：0xA0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PC08_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PC08_sel	PC08_sel: 端口PC08功能选择. 000 ---- GPIO PC08 001 ---- PCA_CH2 PCA模块通道2捕获/比较信号 010 ---- TIM6_CHA Timer6模块通道2A信号 011 ---- TIM2_ETR Timer2模块外部时钟输入信号 100 ---- Reserved Reserved 101 ---- I2S1_SD I2S1模块SD信号 110 ---- UART3_CTS UART3模块CTS信号 111 ---- Reserved Reserved

6.5.5.10 端口 PC09 功能配置寄存器(PC09_SEL)

地址偏移量：0xA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PC09_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PC09_sel	PC09_sel: 端口PC09功能选择. 000 ---- GPIO PC09 001 ---- PCA_CH3 PCA模块通道3捕获/比较信号 010 ---- TIM4_CHB Timer6模块通道0B信号 011 ---- TIM1_ETR Timer1模块外部时钟输入信号 100 ---- Reserved Reserved 101 ---- I2S1_WS I2S1模块WS信号 110 ---- UART3_RTS UART3模块RTS信号 111 ---- Reserved Reserved

6.5.5.11 端口 PC10 功能配置寄存器(PC10_SEL)

地址偏移量：0xA8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PC10_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PC10_sel	PC10_sel: 端口PC10功能选择. 000 ---- GPIO PC10 001 ---- LPUART1_TXD LPUART1模块TXD信号 010 ---- LPUART0_TXD LPUART0模块TXD信号 011 ---- PCA_CH2 PCA模块通道2捕获/比较信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.5.12 端口 PC11 功能配置寄存器(PC11_SEL)

地址偏移量：0xAC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PC11_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PC11_sel	PC11_sel: 端口PC11功能选择. 000 ---- GPIO PC11 001 ---- LPUART1_RXD LPUART1模块RXD信号 010 ---- LPUART0_RXD LPUART0模块RXD信号 011 ---- PCA_CH3 PCA模块通道3捕获/比较信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.5.13 端口 PC12 功能配置寄存器(PC12_SEL)

地址偏移量：0xB0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PC12_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PC12_sel	PC12_sel: 端口PC12功能选择. 000 ---- GPIO PC12 001 ---- LPUART0_TXD LPUART0模块RXD信号 010 ---- LPUART1_TXD LPUART1模块TXD信号 011 ---- PCA_CH4 PCA模块通道4捕获/比较信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.5.14 端口 PC13 功能配置寄存器(PC13_SEL)

地址偏移量：0xB4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														PC13_sel	
														rw	

位	标记	功能描述
31:3	Reserved	保留
2:0	PC13_sel	PC13_sel: 端口PC13功能选择. 000 ---- GPIO PC13 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- TIM3_CH1B Timer3模块通道1B信号 100 ---- Reserved Reserved 101 ---- I2S0_SCK I2S0模块SCK信号 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.5.15 端口 PC14 功能配置寄存器(PC14_SEL)

地址偏移量：0xB8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														PC14_sel	
														rw	

位	标记	功能描述
31:3	Reserved	保留
2:0	PC14_sel	PC14_sel: 端口PC14功能选择. 000 ---- GPIO PC14 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.5.16 端口 PC15 功能配置寄存器(PC15_SEL)

地址偏移量：0xBC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PC15_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PC15_sel	PC15_sel: 端口PC15功能选择. 000 ---- GPIO PC15 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.6 端口 PD 功能选择寄存器

6.5.6.1 端口 PD00 功能配置寄存器(PD00_SEL)

地址偏移量：0xC0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PD00_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PD00_sel	端口PD00功能选择. 000 ---- GPIO PD00 001 ---- CAN_RX CAN模块接收信号 010 ---- SPI1_CS SPI1模块主机模式片选信号 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.6.2 端口 PD01 功能配置寄存器(PD01_SEL)

地址偏移量：0xC4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD01_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD01_sel	端口PD01功能选择. 000 ---- GPIO PD01 001 ---- CAN_TX CAN模块发送信号 010 ---- SPI1_SCK SPI1模块时钟信号 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.6.3 端口 PD02 功能配置寄存器(PD02_SEL)

地址偏移量：0xC8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD02_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD02_sel	PD02_sel: 端口PD02功能选择. 000 ---- GPIO PD02 001 ---- PCA_ECI PCA模块外部时钟输入信号 010 ---- LPUART0_RTS LPUART0模块RTS信号 011 ---- TIM1_ETR Timer1模块外部时钟输入信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.6.4 端口 PD03 功能配置寄存器(PD03_SEL)

地址偏移量：0xCC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PD03_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PD03_sel	端口PD03功能选择. 000 ---- GPIO PD03 001 ---- UART1_CTS UART1模块CTS信号 010 ---- SPI1_MISO SPI1模块主机输入从机输出数据信号 011 ---- Reserved Reserved 100 ---- I2S1_SCK I2S1模块SCK信号 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.6.5 端口 PD04 功能配置寄存器(PD04_SEL)

地址偏移量：0xD0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PD04_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PD04_sel	端口PD04功能选择. 000 ---- GPIO PD04 001 ---- UART1_RTS UART1模块RTS信号 010 ---- SPI1_MOSI SPI1模块主机输出从机输入数据信号 011 ---- Reserved Reserved 100 ---- I2S1_MCK I2S1模块MCK信号 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.6.6 端口 PD05 功能配置寄存器(PD05_SEL)

地址偏移量：0xD4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PD05_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PD05_sel	端口PD05功能选择. 000 ---- GPIO PD05 001 ---- UART1_TXD UART1模块TXD信号 010 ---- Reserved Reserved 011 ---- CAN_STBY CAN模块STBY信号 100 ---- I2S1_SD I2S1模块SD信号 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.6.7 端口 PD06 功能配置寄存器(PD06_SEL)

地址偏移量：0xD8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PD06_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PD06_sel	PD06_sel: 端口PD06功能选择. 000 ---- GPIO PD06 001 ---- UART1_RXD UART1模块RXD信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- I2S1_WS I2S1模块WS信号 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.6.8 端口 PD07 功能配置寄存器(PD07_SEL)

地址偏移量：0xDC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PD07_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PD07_sel	PD07_sel: 端口PD07功能选择. 000 ---- GPIO PD07 001 ---- UART1_TXD UART1模块TXD信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- I2S1_SDIN I2S1模块SDIN信号 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.6.9 端口 PD08 功能配置寄存器(PD08_SEL)

地址偏移量：0xE0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PD08_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PD08_sel	PD08_sel: 端口PD08功能选择. 000 ---- GPIO PD08 001 ---- LPUART0_TXD LPUART0模块TXD信号 010 ---- I2S0_SCK I2S0模块SCK信号 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.6.10 端口 PD09 功能配置寄存器(PD09_SEL)

地址偏移量：0xE4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD09_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD09_sel	PD09_sel: 端口PD09功能选择. 000 ---- GPIO PD09 001 ---- LPUART0_RXD LPUART0模块RXD信号 010 ---- I2S0_MCK I2S0模块MCK信号 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.6.11 端口 PD10 功能配置寄存器(PD10_SEL)

地址偏移量：0xE8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD10_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD10_sel	PD10_sel: 端口PD10功能选择. 000 ---- GPIO PD10 001 ---- LPUART0_TXD LPUART0模块TXD信号 010 ---- I2S0_SD I2S0模块SD信号 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.6.12 端口 PD11 功能配置寄存器(PD11_SEL)

地址偏移量：0xEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														PD11_sel	
														rw	

位	标记	功能描述
31:3	Reserved	保留
2:0	PD11_sel	PD11_sel: 端口PD11功能选择. 000 ---- GPIO PD11 001 ---- LPUART0_CTS LPUART0模块CTS信号 010 ---- I2S0_WS I2S0模块WS信号 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.6.13 端口 PD12 功能配置寄存器(PD12_SEL)

地址偏移量：0xF0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														PD12_sel	
														rw	

位	标记	功能描述
31:3	Reserved	保留
2:0	PD12_sel	PD12_sel: 端口PD12功能选择. 000 ---- GPIO PD12 001 ---- LPUART0_RTS LPUART0模块RTS信号 010 ---- UART2_RTS UART2模块RTS信号 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.6.14 端口 PD13 功能配置寄存器(PD13_SEL)

地址偏移量：0xF4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD13_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD13_sel	PD13_sel: 端口PD13功能选择. 000 ---- GPIO PD13 001 ---- UART2_RXD UART2模块RXD信号 010 ---- I2S0_SDIN I2S0模块SDIN信号 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.6.15 端口 PD14 功能配置寄存器(PD14_SEL)

地址偏移量：0xF8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD14_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD14_sel	PD14_sel: 端口PD14功能选择. 000 ---- GPIO PD14 001 ---- UART2_TXD UART2模块TXD信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.6.16 端口 PD15 功能配置寄存器(PD15_SEL)

地址偏移量：0xFC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PD15_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PD15_sel	PD15_sel: 端口PD15功能选择. 000 ---- GPIO PD15 001 ---- CRS_SYNC 010 ---- UART2_CTS UART2模块CTS信号 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.7 端口 PE 功能选择寄存器

6.5.7.1 端口 PE00 功能配置寄存器(PE00_SEL)

地址偏移量：0x1000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PE00_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PE00_sel	端口PE00功能选择. 000 ---- GPIO PE00 001 ---- TIM1_CHA Timer1模块通道A信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.7.2 端口 PE01 功能配置寄存器(PE01_SEL)

地址偏移量：0x1004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE01_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE01_sel	端口PE01功能选择. 000 ---- GPIO PE01 001 ---- TIM2_CHA Timer2模块通道A信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.7.3 端口 PE02 功能配置寄存器(PE02_SEL)

地址偏移量：0x1008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved-															PE02_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE02_sel	端口PE02功能选择. 000 ---- GPIO PE02 001 ---- PCA_ECI PCA模块外部时钟输入信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.7.4 端口 PE03 功能配置寄存器(PE03_SEL)

地址偏移量：0x100C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE03_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE03_sel	端口PE03功能选择. 000 ---- GPIO PE03 001 ---- PCA_CH0 PCA模块通道0捕获/比较信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.7.5 端口 PE04 功能配置寄存器(PE04_SEL)

地址偏移量：0x1010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE04_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE04_sel	端口PE04功能选择. 000 ---- GPIO PE04 001 ---- PCA_CH1 PCA模块通道1捕获/比较信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.7.6 端口 PE05 功能配置寄存器(PE05_SEL)

地址偏移量：0x1014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE05_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE05_sel	端口PE05功能选择. 000 ---- GPIO PE05 001 ---- PCA_CH2 PCA模块通道2捕获/比较信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.7.7 端口 PE06 功能配置寄存器(PE06_SEL)

地址偏移量：0x1018

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE06_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE06_sel	端口PE06功能选择. 000 ---- GPIO PE06 001 ---- PCA_CH3 PCA模块通道3捕获/比较信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.7.8 端口 PE07 功能配置寄存器(PE07_SEL)

地址偏移量：0x101C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PE07_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PE07_sel	端口PE07功能选择. 000 ---- GPIO PE07 001 ---- TIM3_ETR Timer3模块外部时钟输入信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.7.9 端口 PE08 功能配置寄存器(PE08_SEL)

地址偏移量：0x1020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PE08_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PE08_sel	PE08_sel: 端口PE08功能选择. 000 ---- GPIO PE08 001 ---- TIM3_CH0B Timer3模块通道0B信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.7.10 端口 PE09 功能配置寄存器(PE09_SEL)

地址偏移量：0x1024

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE09_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE09_sel	端口PE09功能选择. 000 ---- GPIO PE09 001 ---- TIM3_CH0A Timer3模块通道0A信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.7.11 端口 PE10 功能配置寄存器(PE10_SEL)

地址偏移量：0x1028

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE10_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE10_sel	PE10_sel: 端口PE10功能选择. 000 ---- GPIO PE10 001 ---- TIM3_CH1B Timer3模块通道1B信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.7.12 端口 PE11 功能配置寄存器(PE11_SEL)

地址偏移量：0x102C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE11_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE11_sel	PE11_sel: 端口PE11功能选择. 000 ---- GPIO PE11 001 ---- TIM3_CH1A Timer3模块通道1A信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.7.13 端口 PE12 功能配置寄存器(PE12_SEL)

地址偏移量：0x1030

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE12_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE12_sel	PE12_sel: 端口PE12功能选择. 000 ---- GPIO PE12 001 ---- TIM3_CH2B Timer3模块通道2B信号 010 ---- SPI0_CS SPI0模块主机模式片选信号 011 ---- UART3_CTS UART3模块CTS信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.7.14 端口 PE13 功能配置寄存器(PE13_SEL)

地址偏移量：0x1034

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PE13_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PE13_sel	PE13_sel: 端口PE13功能选择. 000 ---- GPIO PE13 001 ---- TIM3_CH2A Timer3模块通道2A信号 010 ---- SPI0_SCK SPI0模块时钟信号 011 ---- UART3_RTS UART3模块RTS信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.7.15 端口 PE14 功能配置寄存器(PE14_SEL)

地址偏移量：0x1038

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PE14_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PE14_sel	PE14_sel: 端口PE14功能选择. 000 ---- GPIO PE14 001 ---- TIM3_CH0B Timer3模块通道0B信号 010 ---- SPI0_MISO SPI0模块主机输入从机输出数据信号 011 ---- UART3_RXD UART3模块RXD信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.7.16 端口 PE15 功能配置寄存器(PE15_SEL)

地址偏移量：0x103C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PE15_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PE15_sel	PE15_sel: 端口PE15功能选择. 000 ---- GPIO PE15 001 ---- TIM3_BK Timer3模块刹车信号 010 ---- SPI0_MOSI SPI0模块主机输出从机输入数据信号 011 ---- UART3_TXD UART3模块TXD信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.8 端口 PF 功能选择寄存器

6.5.8.1 端口 PF00 功能配置寄存器(PF00_SEL)

地址偏移量：0x1040

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PF00_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PF00_sel	端口PF00功能选择. 000 ---- GPIO PF00 001 ---- I2C0_SDA I2C0模块数据信号 010 ---- CRS_SYNC 011 ---- UART1_TXD UART1模块TXD信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.8.2 端口 PF01 功能配置寄存器(PF01_SEL)

地址偏移量：0x1044

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PF01_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PF01_sel	PF01_sel: 端口PF01功能选择. 000 ---- GPIO PF01 001 ---- I2C0_SCL I2C0模块时钟信号 010 ---- TIM4_CHB Timer6模块通道0B信号 011 ---- UART1_RXD UART1模块RXD信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.8.3 端口 PF02 功能配置寄存器(PF02_SEL)

地址偏移量：0x1048

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PF02_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PF02_sel	PF02_sel: 端口PF02功能选择. 000 ---- GPIO PF02 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.8.4 端口 PF03 功能配置寄存器(PF03_SEL)

地址偏移量：0x104C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PF03_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PF03_sel	PF03_sel: 端口PF03功能选择. 000 ---- GPIO PF03 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.8.5 端口 PF04 功能配置寄存器(PF04_SEL)

地址偏移量：0x1050

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PF04_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PF04_sel	PF04_sel: 端口PF04功能选择. 000 ---- GPIO PF04 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.8.6 端口 PF05 功能配置寄存器(PF05_SEL)

地址偏移量：0x1054

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PF05_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PF05_sel	PF05_sel: 端口PF05功能选择. 000 ---- GPIO PF05 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.8.7 端口 PF06 功能配置寄存器(PF06_SEL)

地址偏移量：0x1058

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PF06_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PF06_sel	PF06_sel: 端口PF06功能选择. 000 ---- GPIO PF06 001 ---- I2C1_SCL I2C1模块时钟信号 010 ---- LPUART1_CTS LPUART1模块CTS信号 011 ---- UART0_CTS UART0模块CTS信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.8.8 端口 PF07 功能配置寄存器(PF07_SEL)

地址偏移量：0x105C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PF07_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PF07_sel	PF07_sel: 端口PF07功能选择. 000 ---- GPIO PF07 001 ---- I2C1_SDA I2C1模块数据信号 010 ---- LPUART1_RTS LPUART1模块RTS信号 011 ---- UART0_RTS UART0模块RTS信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.8.9 端口 PF09 功能配置寄存器(PF09_SEL)

地址偏移量：0x1064

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PF09_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PF09_sel	PF09_sel: 端口PF09功能选择. 000 ---- GPIO PF09 001 ---- TIM0_CHA Timer0模块通道A信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.8.10 端口 PF10 功能配置寄存器(PF10_SEL)

地址偏移量：0x1068

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PF10_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PF10_sel	PF10_sel: 端口PF10功能选择. 000 ---- GPIO PF10 001 ---- TIM0_CHB Timer0模块通道B信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

6.5.8.11 端口 PF11 功能配置寄存器(PF11_SEL)

地址偏移量：0x106C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PF11_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PF11_sel	PF11_sel: 端口PF11功能选择. 000 ---- GPIO PF11 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

7 I2C 总线（I2C）

7.1 简介

I2C 是双线双向的同步串行总线，它利用一根时钟线和一根数据线在连接总线的两个器件之间进行信息的传递，为设备之间数据交换提供了一种简单高效的方法。每个连接到总线上的器件都有唯一的地址，任何器件既可以作为主机也可以作为从机，但同一时刻只允许有一个主机。I2C 标准是一个具有冲突检测机制和仲裁机制的真正意义上的多主机总线，它能在多个主机同时请求控制总线时利用仲裁机制避免数据冲突并保护数据。

I2C 总线控制器，能满足 I2C 总线的各种规格并支持所有与 I2C 总线通信的传输模式。I2C 逻辑能自主地处理字节的传输。它能保持跟踪串行传送，而且还有一个状态寄存器（I2Cx_STAT)能反映 I2C 总线控制器和 I2C 总线的状态。

7.2 主要特性

I2C 控制器支持以下特性：

- 支持主机发送/接收，从机发送/接收四种工作模式
- 支持标准(100Kbps) / 快速(400Kbps) / 高速(1Mbps) 三种工作速率
- 支持 7 位寻址功能
- 支持噪声过滤功能
- 支持广播地址
- 支持中断状态查询功能

7.3 协议描述

I2C 总线使用连接设备的"SCL"(串行时钟总线)和"SDA"(串行数据总线)来传送信息。主机在 SCL 线上输出串行时钟信号，数据在 SDA 线上进行传输，每传输一个字节（最高位 MSB 开始传输），后面跟随一个应答位。一个 SCL 时钟脉冲传输一个数据位。

7.3.1 I2C 总线上数据传输

通常标准 I2C 传输协议包含四个部分：起始(S)或重复起始信号(Sr)，从机地址及读写位，传输数据，停止信号(P)。

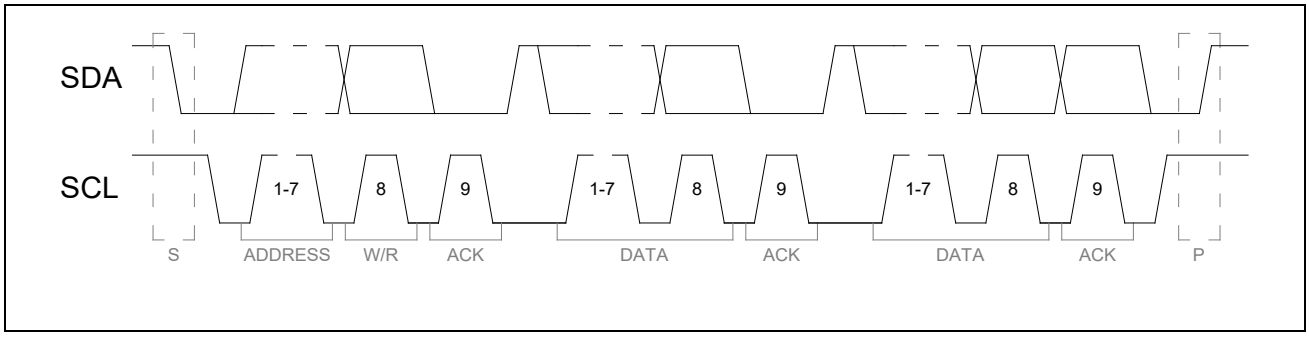


图 7-1 I2C 传输协议

- 起始信号、重复起始信号、停止信号
  - 当总线处于空闲状态下（SCL 和 SDA 线同时为高），SDA 线上出现由高到低的信号，表明总线上产生了起始信号。
  - 当两个起始信号之间没有停止信号时，即产生了重复起始信号。主机采用这种方法与另一个从机或相同的从机以不同传输方向进行通信（例如：从写入设备到从设备读出）而不释放总线。
  - 当 SCL 线为高时，SDA 线上出现由低到高的信号，被定义为停止信号。主机向总线发出停止信号结束数据传送。

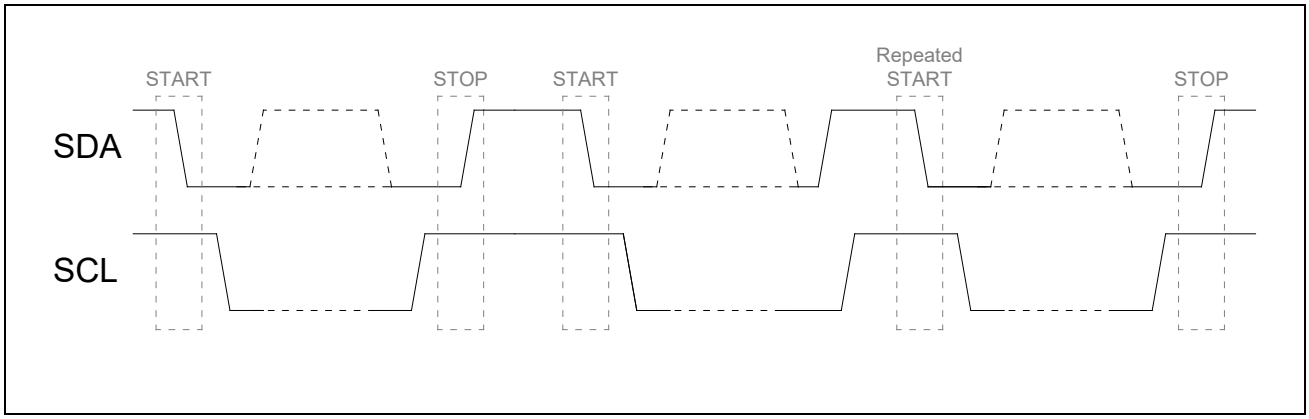


图 7-2 START 和 STOP 条件

- 从机地址及读写位

- 当起始信号产生后，主机立即传输数据的第一字节：7 位从机地址 + 读写位，读写位控制从机的数据传输方向（0：写；1：读）。被主机寻址的从机会通过在第 9 个 SCL 时钟周期将 SDA 置为低电平作为应答。

■ 传输数据

- 数据传输过程中，一个 SCL 时钟脉冲传输一个数据位，且 SDA 线只有在 SCL 为低时才可以改变。

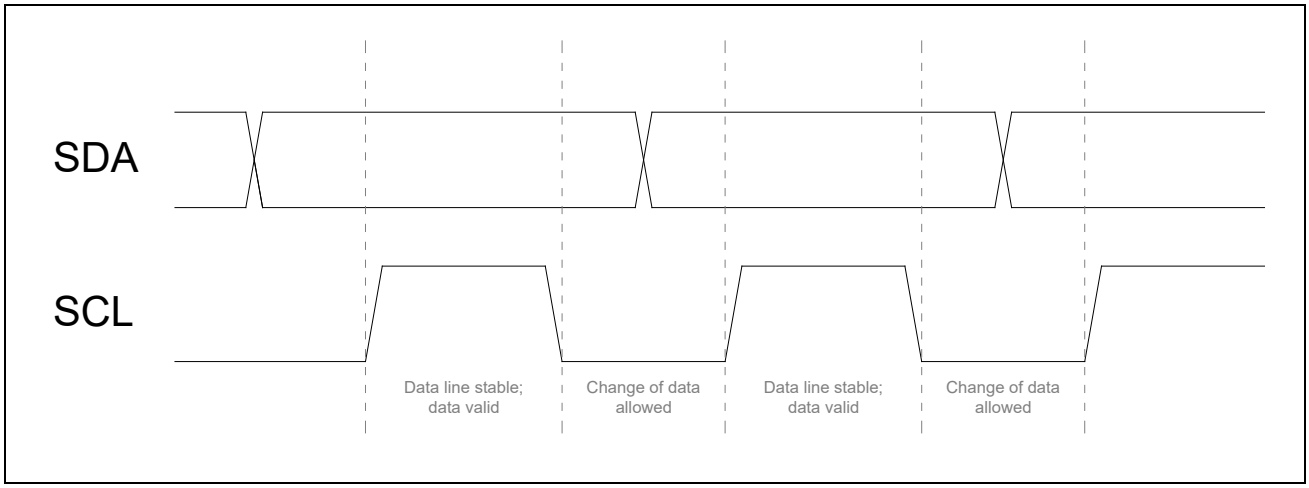


图 7-3 I2C 总线上位传输

7.3.2 I2C 总线上的应答

每传输一个字节，后面跟随一个应答位。通过将 SDA 线拉低，来允许接收端回应发送端。ACK 为一个低电平信号，当时钟信号为高时，SDA 保持低电平则表明接收端已成功接收到发送端的数据。

当主机作为发送器件时，如果从机上产生无响应信号(NACK)，主机可以产生停止信号来退出数据传输，或者产生重复起始信号开始新一轮的数据传输。当主机作为接收器件时，发生无响应信号(NACK)，从机释放 SDA 线，使主机产生停止信号或重复起始信号。

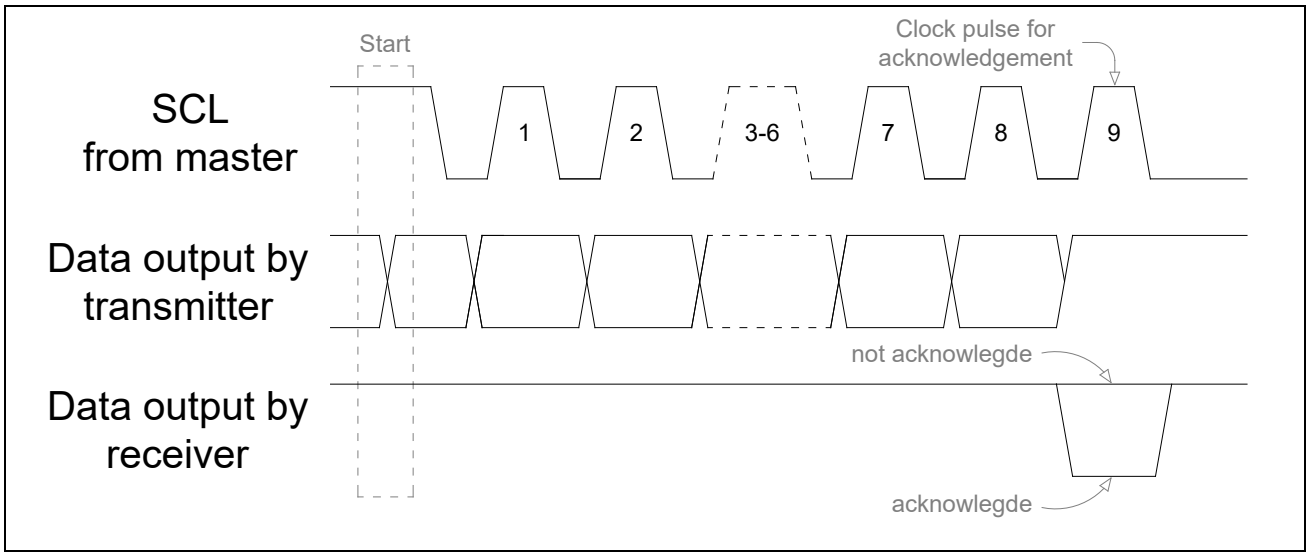


图 7-4 I2C 总线上应答信号

7.3.3 I2C 总线上的仲裁

I2C 总线上的仲裁分为两个部分：SCL 线上的同步和 SDA 线上的仲裁

- SCL 线上的同步（时钟同步）
  - 由于 I2C 总线具有线“与”的逻辑功能，SCL 线上只要有一个节点发送低电平，总线上就表现低电平。当所有的节点都发送高电平时，总线才能表现为高电平。所以，时钟低电平的时间由时钟电平期最长的器件决定，而时钟的高电平时间由时钟高电平期最短的器件决定。由于 I2C 这种特性，当多个主机同时发送时钟信号时，在总线上表示的是统一的时钟信号。如果从机希望主机降低传送速度可以通过将 SCL 主动拉低延长其低电平时间来通知主机，当主机在准备下一次传送时发现 SCL 的电平被拉低时进行等待，直到从机完成操作并释放 SCL 线的控制权。
- SDA 线上的仲裁
  - SDA 线上的仲裁也是由于 I2C 总线具有线“与”的逻辑功能。主机在发送数据后，通过比较总线上的数据来决定是否退出竞争。丢失仲裁的主机立即切换到未被寻址的从机状态，以确保自身能被仲裁胜利的主机寻址到。仲裁失败的主机继续输出时钟脉冲（在 SCL 上），直到发送完当前的串行字节。通过这种原理可以保证 I2C 总线在多个主机企图控制总线时保证数据的不丢失。

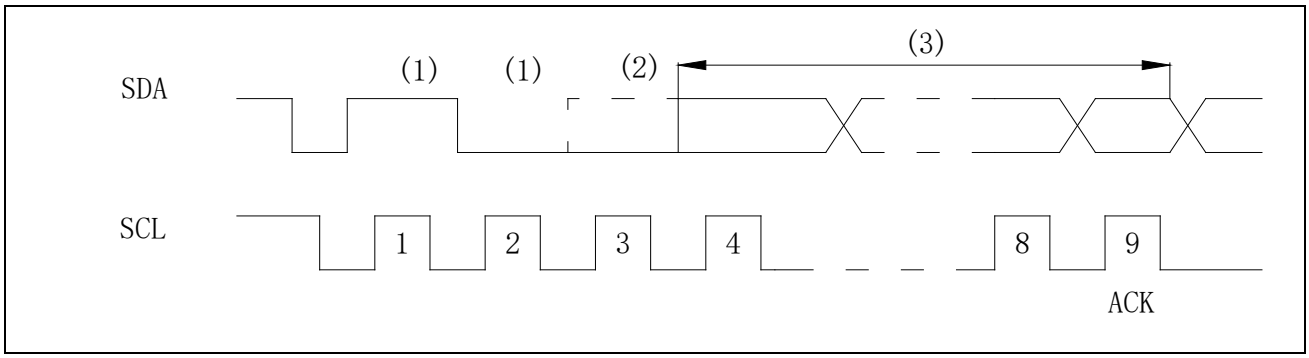


图 7-5 I2C 总线上的仲裁

a) 另一器件发送串行数据；

b) 另一器件通过拉低 SDA 先撤消了该 I2C 主机发送的一个逻辑 1（虚线）。仲裁丢失，I2C 进入从接收模式；

c) 此时 I2C 处于从接收模式，但仍产生时钟脉冲，直至发送完当前字节。I2C 将不为下个字节的传输产生时钟脉冲。一旦赢得仲裁，SDA 上的数据传输由新的主机来启动。

7.4 功能描述

I2C 总线使用双线在连接到总线"SCL"（串行时钟线）和"SDA"（串行数据线）的设备间传送信息。滤波逻辑可以过滤数据总线上的毛刺来保护数据的完整性。由于只有无方向端口，I2C 组件需要使用到引脚的漏端开路缓冲器。每个连接到总线的设备都能使用软件通过特定地址寻址。I2C 标准是一个具有冲突检测机制和仲裁机制的真正意义上的多主机总线。它能防止两个或者多个主机在同时开始传输数据时发生数据冲突。I2C 总线状态能在状态寄存器中进行查询。

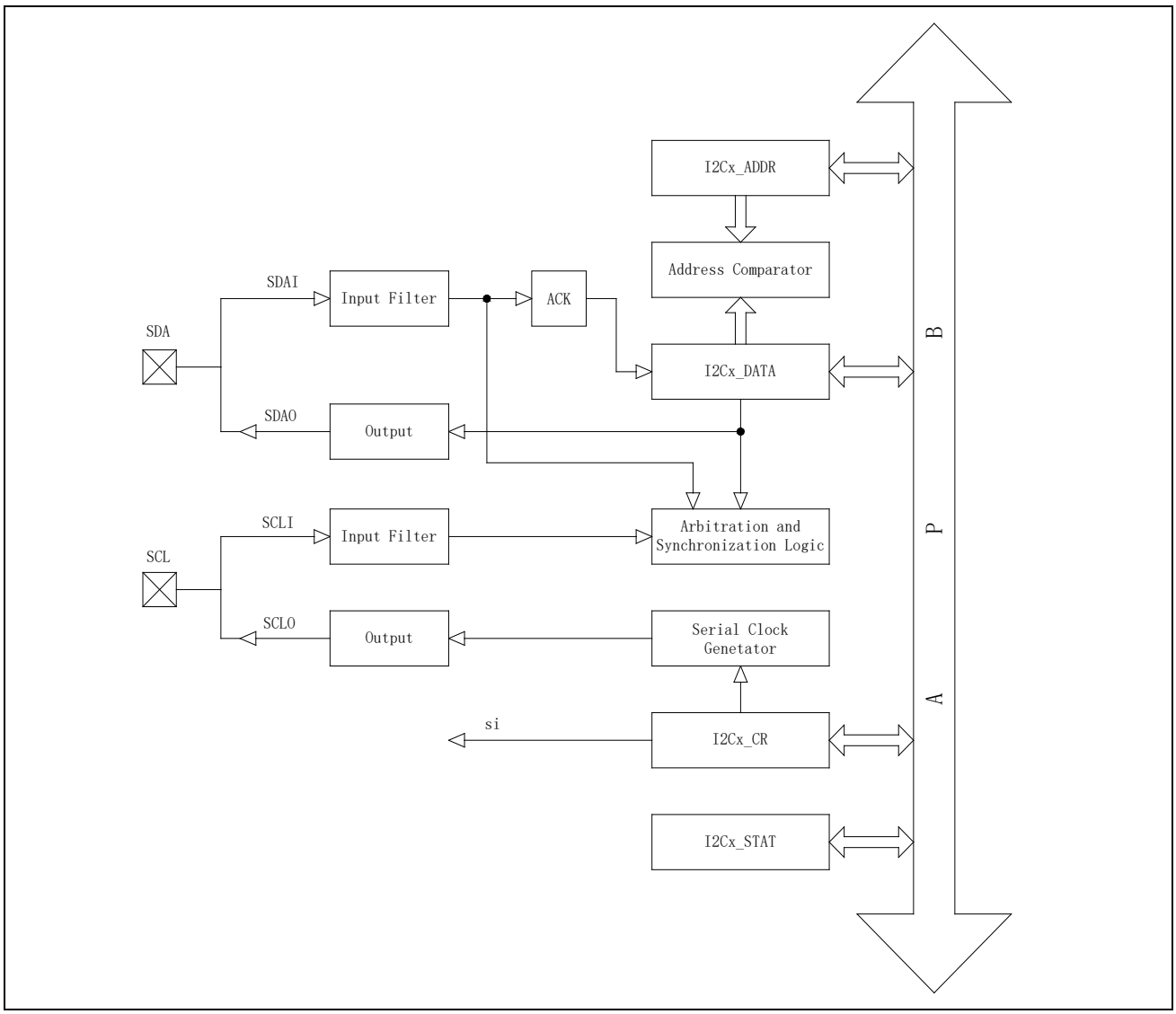


图 7-6 I2C 功能模块图

7.4.1 串行时钟发生器

串行时钟发生器采用一个 8 位的计数器作为波特率发生器，SCL 信号和 PCLK 信号的频率关系为 Fscl = Fpclk / 8 / (I2Cx_TM.tm+1)，其中 I2Cx_TM.tm 应大于 0。

下表列出了 PCLK 频率与 I2Cx_TM.tm 组合时，SCL 信号的输出频率值。

表 7-1 I2C 时钟信号波特率

PCLK (KHz)	I2Cx_TM.tm						
	1	2	3	4	5	6	7
1000	62	41	31	25	20	17	15
2000	125	83	62	50	41	35	31
4000	250	166	125	100	83	71	62
6000	375	250	187	150	125	107	93
8000	500	333	250	200	166	142	125
10000	625	416	312	250	208	178	156
12000	750	500	375	300	250	214	187
14000	875	583	437	350	291	250	218
16000	1000	666	500	400	333	285	250

7.4.2 输入滤波器

输入信号与 PCLK 同步，低于 PCLK 周期的尖锋脉冲信号会被滤除。

当本模块作为主机时，如果 I2C_TM 的值小于等于 9，应设置 I2C_CR.H1M 为 1；如果 I2C_TM 的值大于 9，应设置 I2C_CR.H1M 为 0。

当本模块作为从机时，如果 PCLK 与 SCL 频率比值小于等于 30 时，应设置 I2C_CR.H1M 为 1；如果 PCLK 与 SCL 频率比值大于 30 时，应设置 I2C_CR.H1M 为 0。

7.4.3 地址比较器

I2C 比较器将自己的从机地址与接收到的 7 位从机地址做比较。它可使用"I2Cx_ADDR"寄存器对自己的从机地址进行编程。并且会根据"I2Cx_ADDR"寄存器的"i2cadr"位与首次接收到的 8 位字节或广播地址（0x00）相比较。如果任何一者相同，"I2Cx_CR"寄存器的"si"位会被置 1 并产生一个中断请求。

7.4.4 应答标志位

"I2Cx_CR"寄存器的"aa"标志位为应答标志位。当"aa"位为 1 时，I2C 模块收到数据后回应答位，当"aa"位为 0 时，I2C 模块收到数据后回非应答位。

7.4.5 中断产生器

"I2Cx_CR"寄存器的"si"标志位为中断标志位。每当状态寄存器（I2Cx_STAT）的值产生变化时（除变为0xF8 外），"si"标志位都会被置 1。当产生中断时，通过查询状态寄存器（I2Cx_STAT）可获知 I2C 总线的状态，以确定中断的实际来源。为了进行下一步的操作，"si"标志位必须通过软件清零。

7.4.6 工作模式

I2C 组件可实现 8 位的双向数据传输，传输速率在标准模式下可达到 100Kbps 而在高速模式下可达400Kbps，在超高速模式下可达 1Mbps，并且可以在四种模式下工作：主机发送模式、主机接收模式、从机接收模式、从机发送模式。 还有一种特殊模式广播呼叫模式，其操作方式与从机接收模式类似。

- 主机发送模式

  主机发送多个字节到从机，主机产生时钟，所以需要在 I2Cx_TM 中填入设定值。主机发送模式中需要将 I2Cx_CR.sta 置 1。当总线空闲时，主机发起一个起始位 START。若成功 I2Cx_CR.si 被置1。接下来把从机地址和写位（SLA+W）写入 I2Cx_DATA 中，清零"si"位后，总线上发出 SLA+W。主机发出 SLA+W 收到从机应答位 ACK 后，"si"被置 1。接下来根据用户定义格式发送数据。所有数据发送完后，将 I2Cx_CR.sto 置 1，清零"si"位后发出 STOP 信号，也可以发送重复起始信号进行新一轮数据传输。

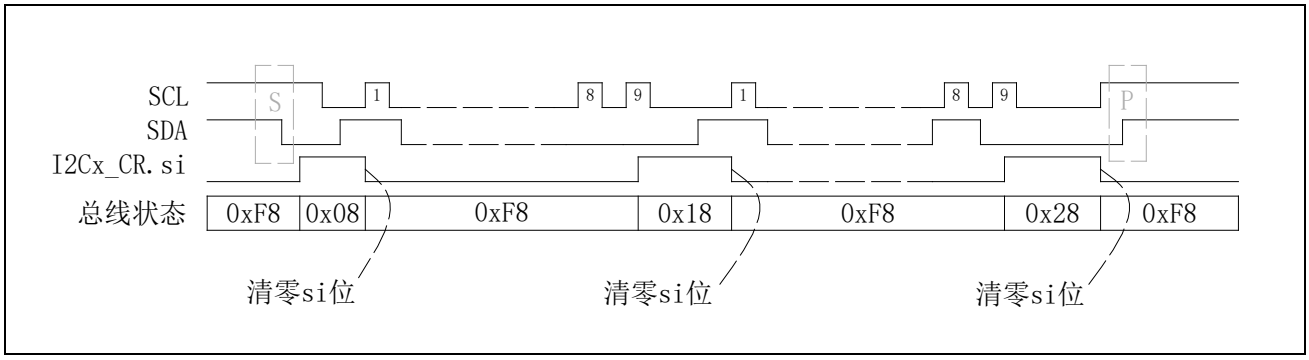


图 7-7 主发送模式数据同步图

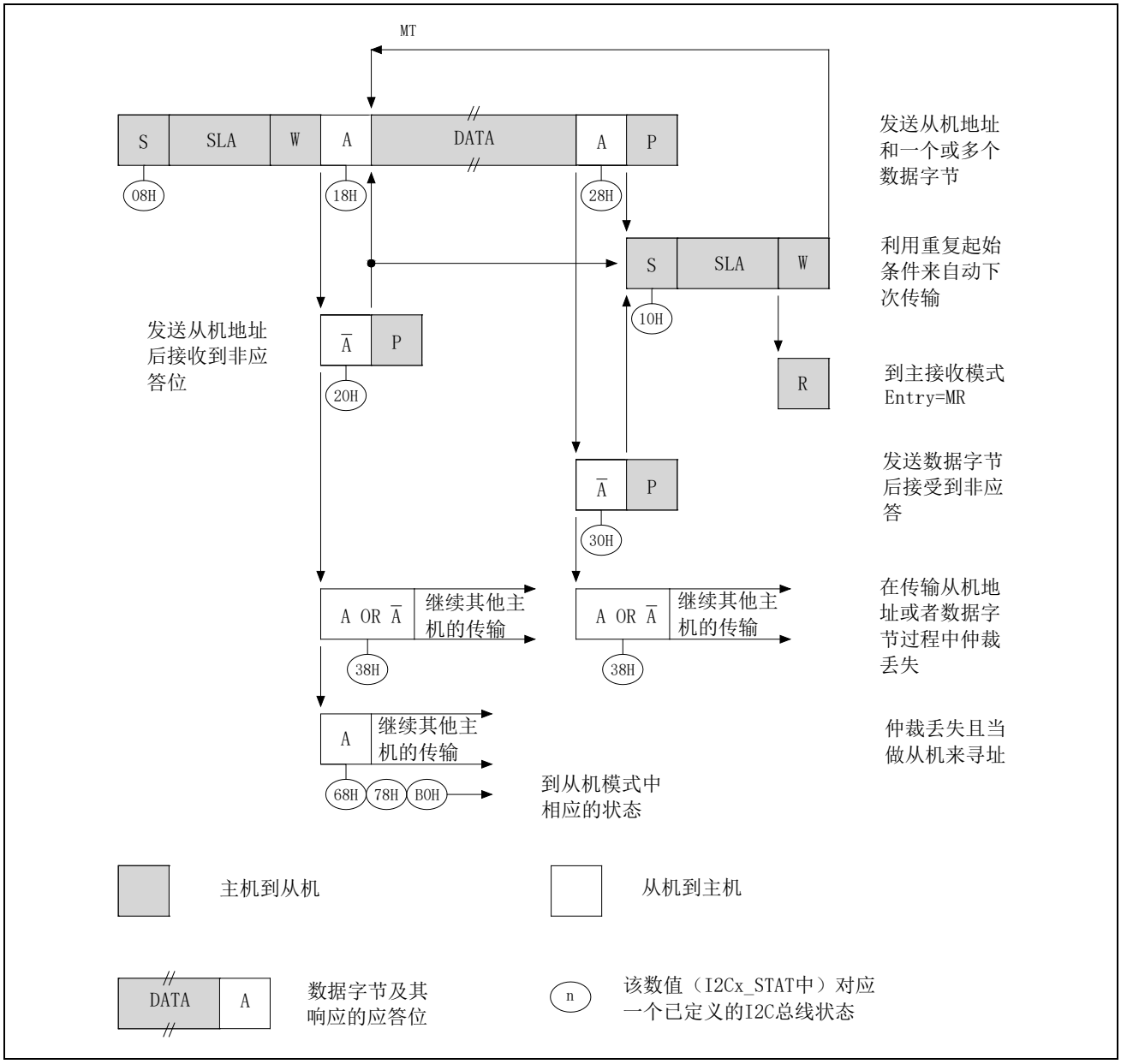


图 7-8 I2C 主机发送状态图

- 主机接收模式

  主机接收模式，由从机传输数据。初始化设置与主机发送模式相同，主机发送起始位以后，I2Cx_DATA 应写入从机地址和“读位”(SLA+R)。收到从机应答位 ACK 后 I2Cx_CR.si 被置 1。"si" 清 0 后开始接收从机数据，若 I2Cx_CR.aa 为 1，主机收到数据后回应答位；若为 0 主机收到数据后不回应答 NACK。然后主机可以发停止信号或重复起始信号开始下一轮的数据传输。

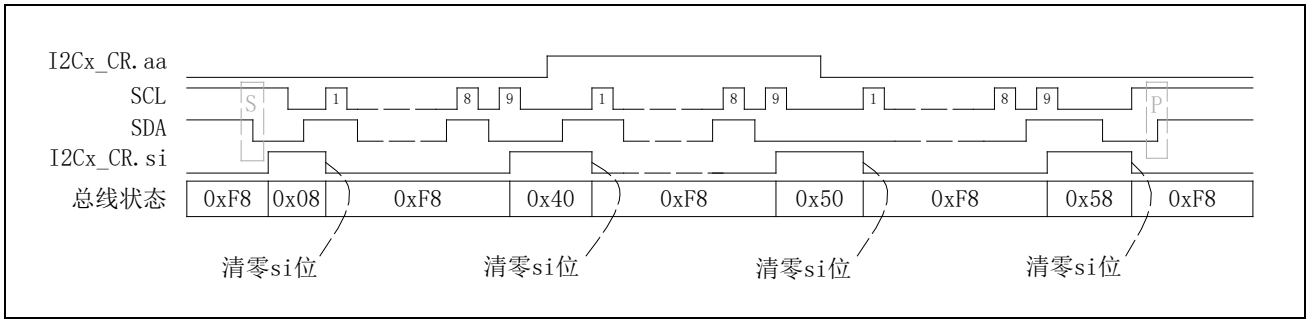


图 7-9　主接收模式数据同步图

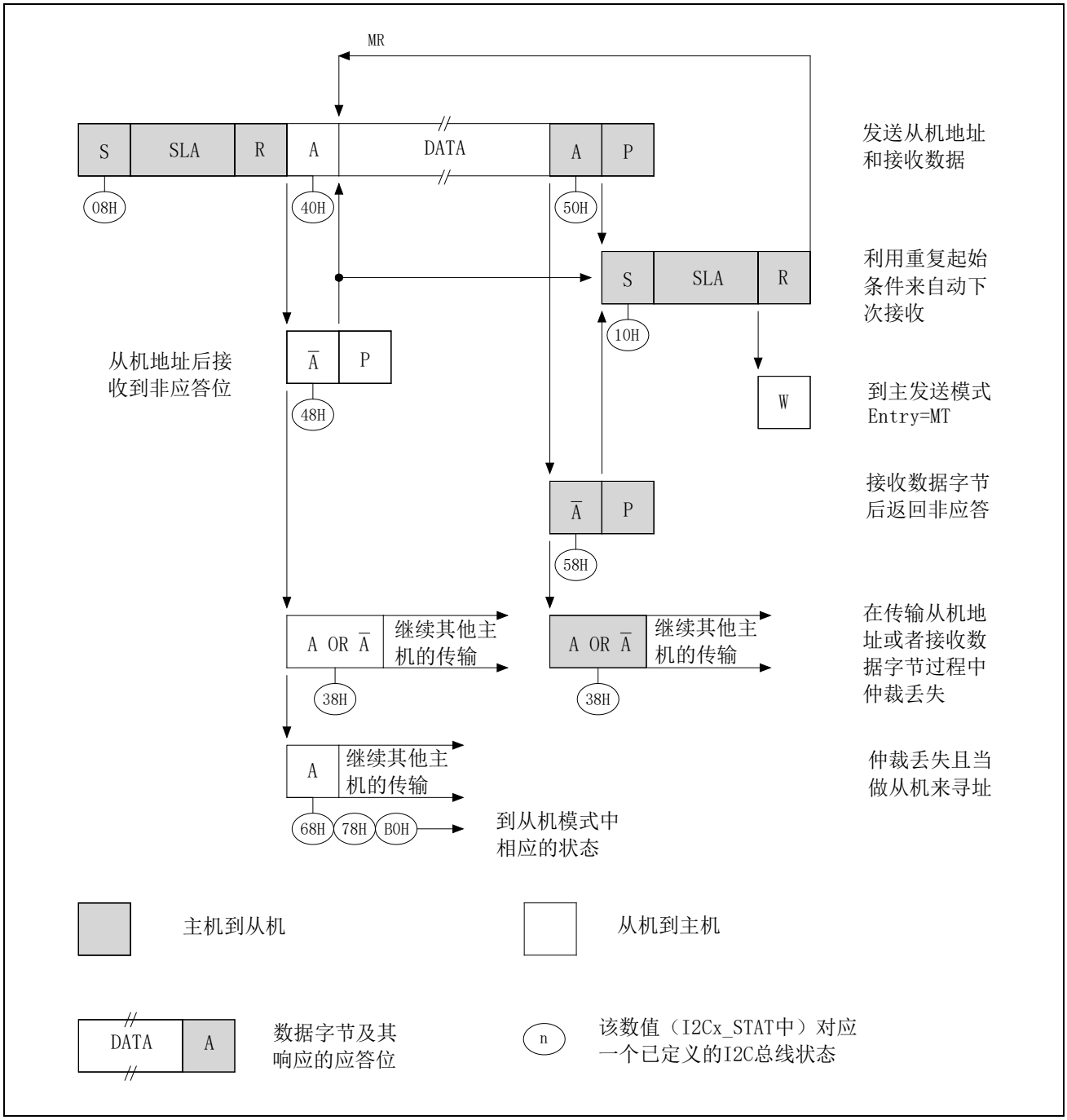


图 7-10　I2C 主机接收状态图

■ 从机接收模式

在从机接收模式下，从机接收主机发来的数据。在传输开始前，I2Cx_ADDR 应写入从机地址，I2Cx_CR.aa 置 1 用以响应主机的寻址。上述初始化后，从机进入空闲模式，等待“写”信号（SLA+W）。若主机仲裁失败，也会直接进入从机接收模式。

当从机被“写”信号 SLA+W 寻址到后，需要清零"si"位，以便从主机接收数据。如果在传输过程中 I2Cx_CR.aa=0，从机将在下一字节返回无应答位 NACK，从机也将转为未寻址从机，与主机联系终止，不再接收数据，且 I2C_DATA 保持之前接收到的数据。从机地址识别可通过置位"aa"来恢复，这意味着"aa"位可临时将 I2C 模块从 I2C 总线上分离出来。

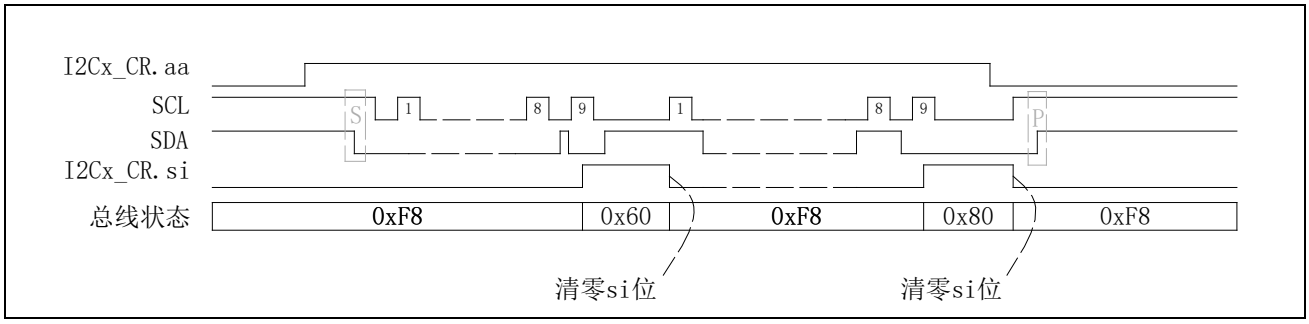


图 7-11 从接收模式数据同步图

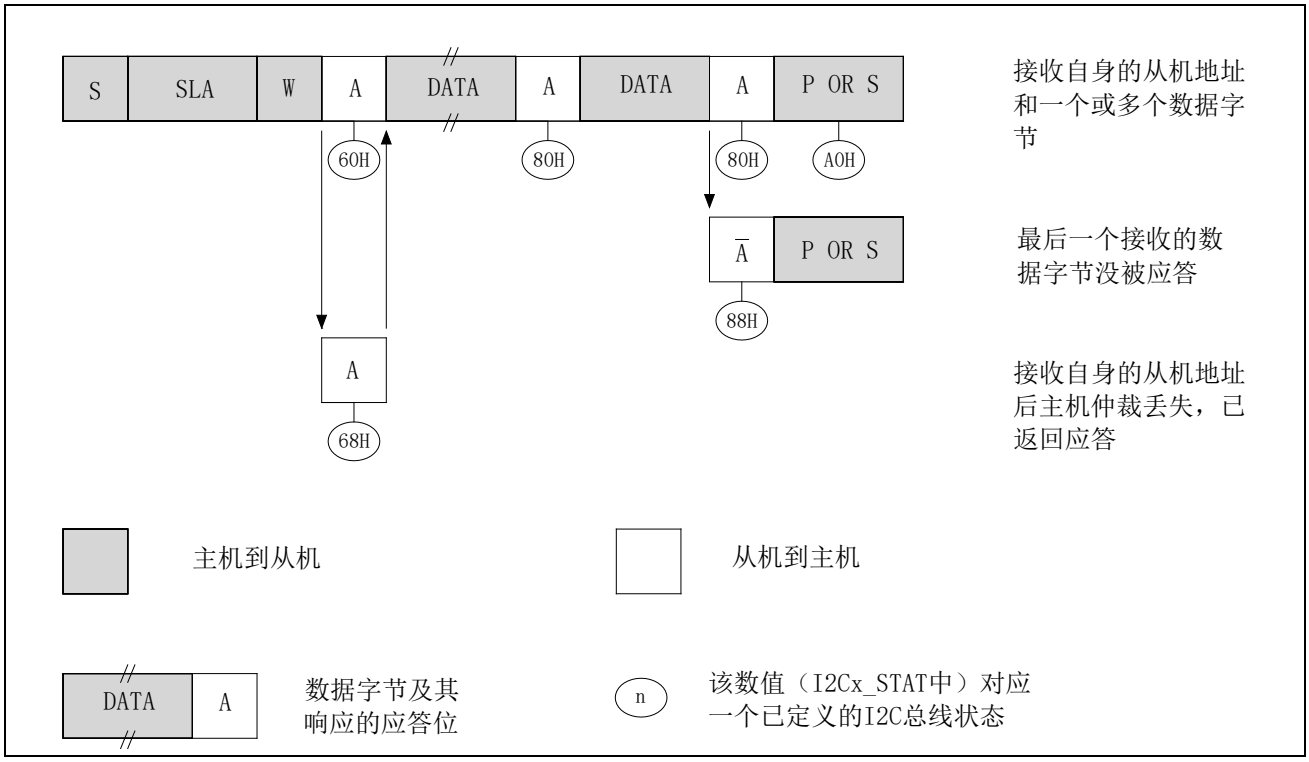


图 7-12 从机接收状态图

■ 从机发送模式

从机发送模式，数据由从机发送给主机。当初始化 I2Cx_ADDR 及 I2Cx_CR.aa 值后，器件等待直到自身地址被“读”信号(SLA+R)寻址。若主机仲裁失败，也可进入从机发送模式。

当从机被“读”信号 SLA+R 寻址，需要将"si"清零用以向主机发送数据。通常主机接收每字节数据后会返回应答位。

如果传输过程中 I2Cx_CR.aa 清零 ，从机将发送最后一个字节数据，并在接下去的传输中发送全 1 数据，并将自身变为未寻址从机。

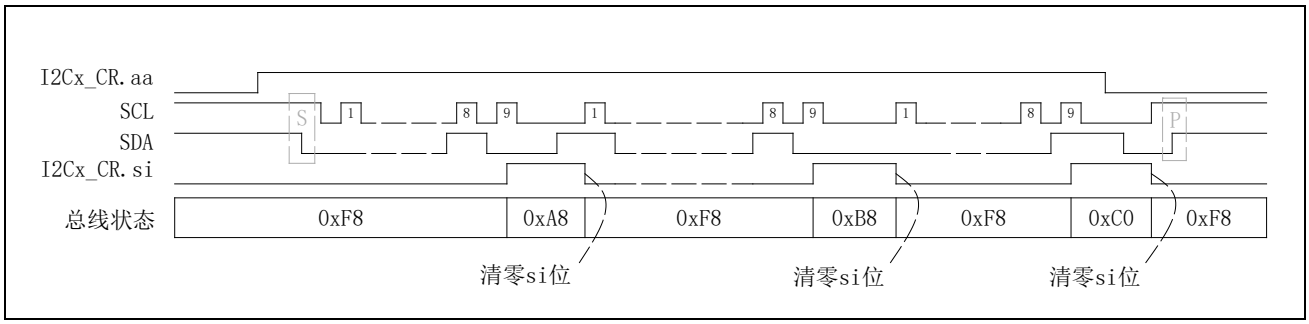


图 7-13 从发送模式数据同步图

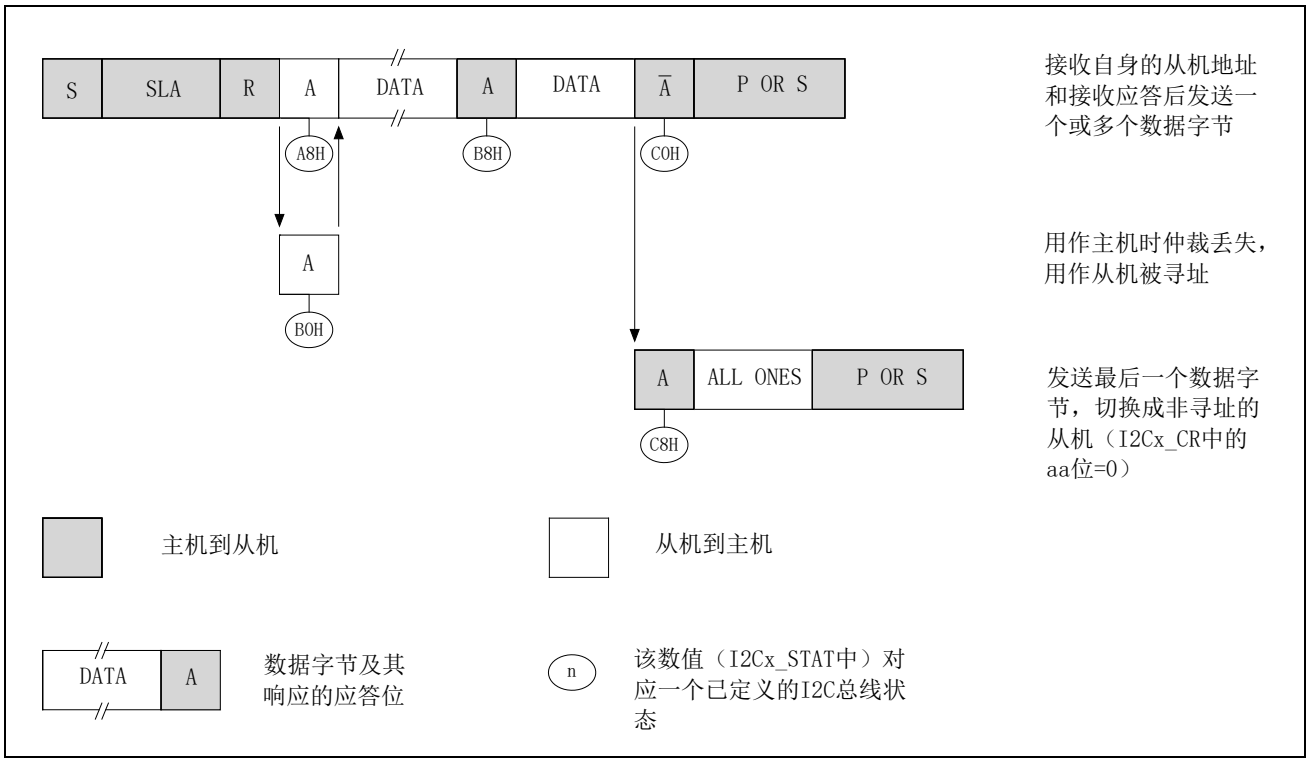


图 7-14 I2C 从机发送状态图

■ 广播呼叫模式

广播呼叫模式是一种特殊的从机接收模式， 寻址方式为 0x00，从机地址和读写都为 0。当 I2Cx_ADDR.GC 及 I2Cx_CR.aa 都为置 1，使能接收广播呼叫模式。在该模式下 I2Cx_STAT 值与普通从机接收模式 I2Cx_STAT 值不同。仲裁失败也可能进入广播呼叫模式。

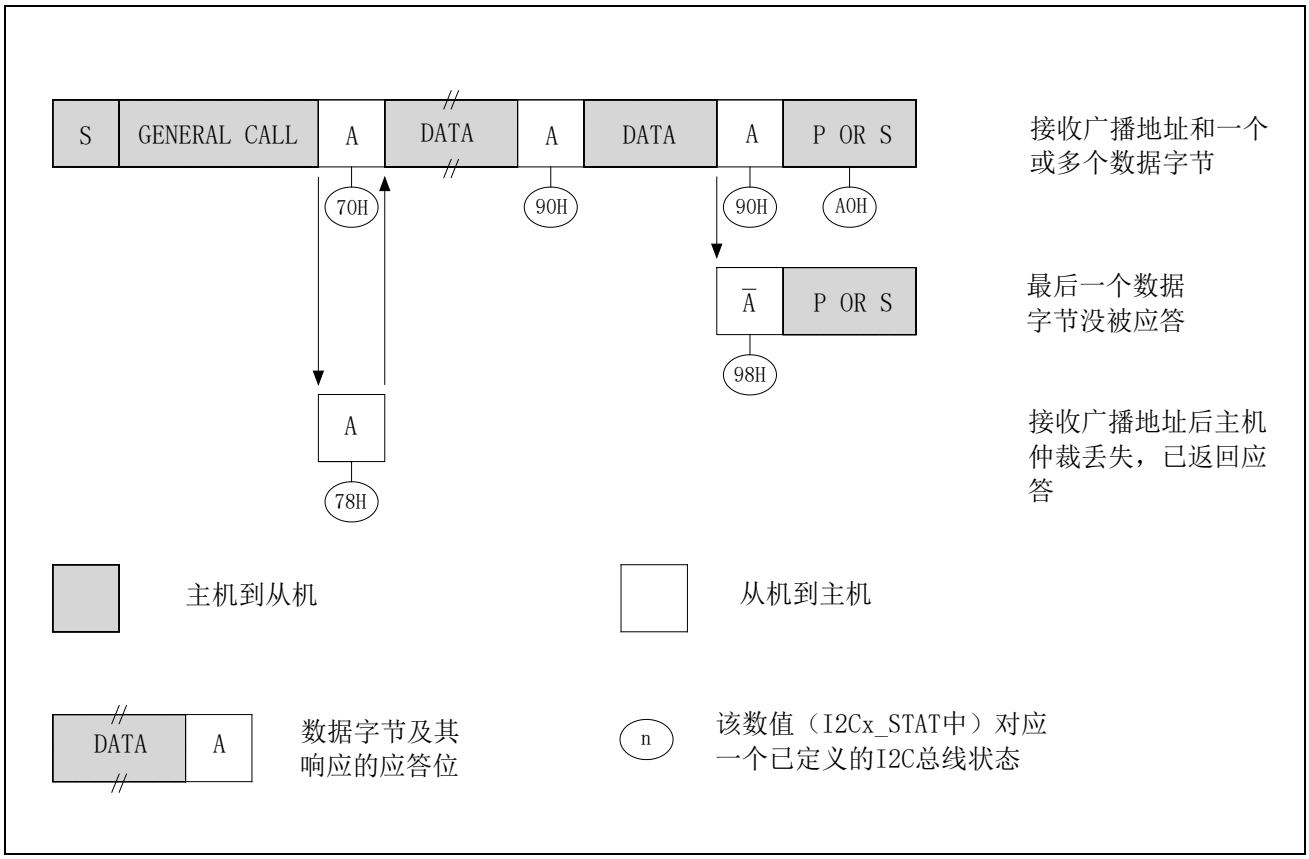


图 7-15 I2C 广播呼叫状态图

7.4.7 状态码表述

I2C 状态寄存器中有两种特殊的状态：F8H 和 00H。

F8H：这个状态码表示没有任何可用的相关信息，因为串行中断标志"si"还没有置位。这种情况在其它状态和 I2C 模块还未开始执行串行传输之间出现。

00H：该状态代码表示在 I2C 串行传输过程中出现了总线错误。当格式帧的非法位置上出现了起始或停止条件时总线错误产生。这些非法位置是指在串行传输过程中的地址字节、数据字节或应答位。当外部干扰影响到内部 I2C 模块信号时也会产生总线错误。总线错误出现时"si"置位。

表 7-2 I2C 状态码表述

状态代码	描述
主发送模式	
08H	已发送起始条件
10H	已发送重复起始条件
18H	已发送 SLA+W，已接收 ACK
20H	已发送 SLA+W，已接收非 ACK
28H	已发送 I2Cx_DATA 中的数据，已接收 ACK
30H	已发送 I2Cx_DATA 中的数据，已接收非 ACK
38H	在 SLA+读写或写数据字节时丢失仲裁
主接收模式	
08H	已发送起始条件
10H	已发送重复起始条件
38H	在非 ACK 中丢失仲裁
40H	已发送 SLA+R，已接收 ACK
48H	已发送 SLA+R，已接收非 ACK
50H	已接收数据字节，ACK 已返回
58H	已接收数据字节，非 ACK 已返回
从接收模式	
60H	已接收自身的 SLA+W，已返回 ACK
68H	主控时在 SLA+读写丢失仲裁，已接收自身的 SLA+W，已返回 ACK
80H	前一次寻址使用自身从地址，已接收数据字节，已返回 ACK
88H	前一次寻址使用自身从地址，已接收数据字节，已返回非 ACK
A0H	静态寻址时，接收到停止条件或重复起始条件
从发送模式	
A8H	已接收自身的 SLA+R，已返回 ACK
B0H	当主机时丢失仲裁，已接收自身 SLA+R，已返回 ACK
B8H	已发送数据，已接收 ACK
C0H	已发送数据字节，已接收非 ACK
C8H	装入的数据字节已被发送，已接收 ACK
广播呼叫模式	
70H	已接收广播地址（0x00），已返回 ACK
78H	主控时在 SLA+读写丢失仲裁，已接收广播地址，已返回 ACK
90H	前一次寻址使用广播地址，已接收数据字节，已返回 ACK
98H	前一次寻址使用广播地址，已接收数据字节，已返回非 ACK
A0H	静态寻址时，接收到停止条件或重复起始条件
其余杂项状态	
F8H	无可用的相关状态信息，si=0
00H	传输过程出现总线错误，或外部干扰使 I2C 进入未定义的状态

7.5 编程示例

7.5.1 主机发送示例

Step1：设置 PERI_CLKEN0.I2Cx 为 1，使能 I2Cx 模块时钟。

Step2：向 PERI_RESET.I2Cx 依次写入 0、1，复位 I2Cx 模块。

Step3:将 I2C 的 SCL、SDA 使用的端口配置为开漏模式，根据 I2C 的模式需要配置合适的端口方向(如配置端口方向为输出，在 DIR 相应位清零之前，需要先初始化相应的输出寄存器为 1)，最后将 SCL、SDA 映射到相应的管脚。

Step4：配置 I2Cx_TM，使 SCL 的时钟速率符合应用需求。

Step5：设置 I2Cx_TMRUN 为 1，使能 SCL 时钟发生器。

Step6：设置 I2Cx_CR.ens 为 1，使能 I2C 模块。

Step7：设置 I2Cx_CR.sta 为 1，总线尝试发送 Start 信号。

Step8：等待 I2Cx_CR.si 变为 1，Start 信号已发送到总线上。

Step9：查询 I2Cx_STAT，如果该寄存器值为 0x08 或 0x10，继续执行下一步骤，否则进行出错处理。

Step10：向 I2Cx_DATA 中写入 SLA+W，设置 I2Cx_CR.sta 为 0，设置 I2Cx_CR.si 为 0，发送 SLA+W。

Step11：等待 I2Cx_CR.si 变为 1，SLA+W 已发送到总线上。

Step12：查询 I2Cx_STAT，如果该寄存器值为 0x18，继续执行下一步骤。否则进行出错处理。

Step13：向 I2Cx_DATA 写入待发送的数据，设置 I2Cx_CR.si 为 0，发送数据。

Step14：等待 I2Cx_CR.si 变为 1，数据已发送到总线上。

Step15：查询 I2Cx_STAT，如果该寄存器值为 0x28，继续执行下一步骤。否则进行出错处理。

Step16：如待发送的数据未完成，则跳转到 Step13 继续执行。

Step17：设置 I2Cx_CR.sto 为 1，设置 I2Cx_CR.si 为 0，总线尝试发送 Stop 信号。

7.5.2 主机接收示例

Step1：设置 PERI_CLKEN0.I2Cx 为 1，使能 I2Cx 模块时钟。

Step2：向 PERI_RESET.I2Cx 依次写入 0、1，复位 I2Cx 模块。

Step3:将 I2C 的 SCL、SDA 使用的端口配置为开漏模式，根据 I2C 的模式需要配置合适的端口方向(如配置端口方向为输出，在 DIR 相应位清零之前，需要先初始化相应的输出寄存器为 1)，最后将 SCL、SDA 映射到相应的管脚。

Step4：配置 I2Cx_TM，使 SCL 的时钟速率符合应用需求。

Step5：设置 I2Cx_TMRUN 为 1，使能 SCL 时钟发生器。

Step6：设置 I2Cx_CR.ens 为 1，使能 I2C 模块。

Step7：设置 I2Cx_CR.sta 为 1，总线尝试发送 Start 信号。

Step8：等待 I2Cx_CR.si 变为 1，Start 信号已发送到总线上。

Step9：查询 I2Cx_STAT，如果寄存器值为 0x08 或 0x10，继续执行下一步骤，否则进行出错处理。

Step10：向 I2Cx_DATA 写入 SLA+R，设置 I2Cx_CR.sta 为 0，设置 I2Cx_CR.si 为 0，发送 SLA+R。

Step11：等待 I2Cx_CR.si 变为 1，SLA+R 已发送到总线上。

Step12：查询 I2Cx_STAT，如果寄存器值为 0x40，继续执行下一步骤，否则进行出错处理。

Step13：设置 I2Cx_CR.aa 为 1，使能应答标志。

Step12：设置 I2Cx_CR.si 为 0，从机发送数据，主机根据 I2Cx_CR.aa 发送 ACK 或 NACK。

Step13：等待 I2Cx_CR.si 变为 1，从 I2Cx_DATA 读取已接收到的数据。

Step14：查询到 I2Cx_STAT，如果该寄存器值为 0x50 或 0x58，继续执行下一步骤，否则进行出错处理。

Step15：如果待接收的数据只差最后一个字节，设置 I2Cx_CR.aa 为 0，使能非应答标志。

Step16：如待接收的数据未完成，则跳转到 Step14 继续执行。

Step17：设置 I2Cx_CR.sto 为 1，设置 I2Cx_CR.si 为 0，总线尝试发送 Stop 信号。

7.5.3 从机接收示例

Step1：设置 PERI_CLKEN0.I2Cx 为 1，使能 I2Cx 模块时钟。

Step2：向 PERI_RESET.I2Cx 依次写入 0、1，复位 I2Cx 模块。

Step3:将 I2C 的 SCL、SDA 使用的端口配置为开漏模式，根据 I2C 的模式需要配置合适的端口方向(如配置端口方向为输出，在 DIR 相应位清零之前，需要先初始化相应的输出寄存器为 1)，最后将 SCL、SDA 映射到相应的管脚。

Step4：设置 I2Cx_CR.ens 为 1，使能 I2C 模块。

Step5：配置 I2Cx_ADDR 为从机地址。

Step6：设置 I2Cx_CR.aa 为 1，使能应答标志。

Step7：等待 I2Cx_CR.si 变为 1，被 SLA+W 寻址。

Step8：查询 I2Cx_STAT，如果该寄存器值为 0x60，继续执行下一步骤，否则进行出错处理。

Step9：设置 I2Cx_CR.si 为 0，主机发送数据，从机根据 I2Cx_CR.aa 返回 ACK 或 NACK。

Step10：等待 I2Cx_CR.si 变为 1，从 I2Cx_DATA 中读取已接收到的数据。

Step11：查询 I2Cx_STAT，如果该寄存器值为 0x80，继续执行下一步骤，否则进行出错处理。

Step12：如待接收的数据未完成，则跳转到 Step9 继续执行。

Step13：设置 I2Cx_CR.aa 为 0，设置 I2Cx_CR.si 为 0。

7.5.4 从机发送示例

Step1：设置 PERI_CLKEN0.I2Cx 为 1，使能 I2Cx 模块时钟。

Step2：向 PERI_RESET.I2Cx 依次写入 0、1，复位 I2Cx 模块。

Step3:将 I2C 的 SCL、SDA 使用的端口配置为开漏模式，根据 I2C 的模式需要配置合适的端口方向(如配置端口方向为输出，在 DIR 相应位清零之前，需要先初始化相应的输出寄存器为 1)，最后将 SCL、SDA 映射到相应的管脚。

Step4：设置 I2Cx_CR.ens 为 1，使能 I2C 模块。

Step5：配置 I2Cx_ADDR 为从机地址。

Step6：设置 I2Cx_CR.aa 为 1，使能应答标志。

Step7：等待 I2Cx_CR.si 变为 1，被 SLA+R 寻址。

Step8：查询 I2Cx_STAT，如果该寄存器的值为 0xA8，继续执行下一步骤，否则进行出错处理。

Step9：向 I2Cx_DATA 写入待发送的数据，设置 I2Cx_CR.si 为 0，发送数据。

Step10：等待 I2Cx_CR.si 变为 1，数据已发送到总线上。

Step11：查询 I2Cx_STAT，如果该寄存器的值为 0xB8 或 0xC0 时，继续执行下一步骤，否则进行出错处理。

Step12：如待发送的数据未完成，则跳转到 Step9 继续执行。

Step13：设置 I2Cx_CR.aa 为 0，设置 I2Cx_CR.si 为 0。

7.6 寄存器描述

寄存器列表

I2C0 基地址：0x40000400

I2C1 基地址：0x40004400

表 7-3 寄存器列表

偏移量	寄存器名称	访问	寄存器描述
0x00	I2Cx_TMRUN	RW	I2C 波特率计数器使能寄存器.
0x04	I2Cx_TM	RW	I2C 波特率计数器配置寄存器.
0x08	I2Cx_CR	RW	I2C 配置寄存器.
0x0c	I2Cx_DATA	RW	I2C 数据寄存器.
0x10	I2Cx_ADDR	RW	I2C 地址寄存器.
0x14	I2Cx_STAT	RO	I2C 状态寄存器.

7.6.1 I2C 波特率计数器使能寄存器(I2Cx_TMRUN)

地址偏移量：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															tme
															RW

位	标记	功能描述
31：1	Reserved	
0	tme	波特率计数器使能. 0 – 禁止 1 – 使能

7.6.2 I2C 波特率计数器配置寄存器(I2Cx_TM)

地址偏移量：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								tm							
								RW							

位	标记	功能描述
31:8	Reserved	
7:0	tm	tm:波特率计数器配置值. Fscl = Fpclk / 8 / (tm+1) ，其中tm >0

7.6.3 I2C 配置寄存器(I2Cx_CR)

地址偏移量：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									ens	sta	sto	si	aa	Res	h1m
									RW	RW	RW	RW	RW		RW

位	标记	功能描述
31:7	Reserved	
6	ens	I2C 模块使能控制 0 – 禁止 1 – 使能
5	sta	I2C总线控制 0 – 无功能 1 – 向总线发送START
4	sto	I2C总线控制 0 – 无功能 1 – 向总线发送STOP
3	si	I2C 中断标志 读出1，已发生I2C中断 写入0，I2C进行一下步操作
2	aa	应答控制位 0 – 发送NAK 1 – 发送ACK
1	Reserved	
0	h1m	I2C 滤波参数配置 0 – 高级滤波，更高的抗干扰性能 1 – 简单滤波，更快的通信速率 注：详见【输入滤波器】章节。

7.6.4 I2C 数据寄存器(I2Cx_DATA)

地址偏移量：0x0c

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								i2cdat							
								RW							

位	标记	功能描述
31:8	Reserved	
7:0	i2cdat	I2C 数据寄存器 在 I2C 发送模式下，写入待发送的数据 在 I2C 接收模式下，读出收到的数据

7.6.5 I2C 地址寄存器(I2Cx_ADDR)

地址偏移量：0x10

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								i2cadr							GC
								RW							RW

位	标记	功能描述
31:8	Reserved	
7:1	i2cadr	I2C 从机模式地址.
0	GC	广播地址应答使能 0 – 禁止 1 – 使能

7.6.6 I2C 状态寄存器(I2Cx_STAT)

地址偏移量：0x14

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								i2csta							
								RO							

位	标记	功能描述
31:8	Reserved	
7:0	i2csta	I2C 状态寄存器 状态值的具体定义详见【状态码表述】章节

8 串行外设接口（SPI）

8.1 SPI 简介

SPI 接口是工作于全双工模式下的同步串行数据通信接口，使用 4 个引脚进行通信：MISO、MOSI、SCK、CS/SSN。当 SPI 作为主机时，输出 CS 和 SCK 信号以控制通信过程。当 SPI 作为从机时，在 SSN 和 SCK 信号的控制下进行通信。

8.2 SPI 主要特性

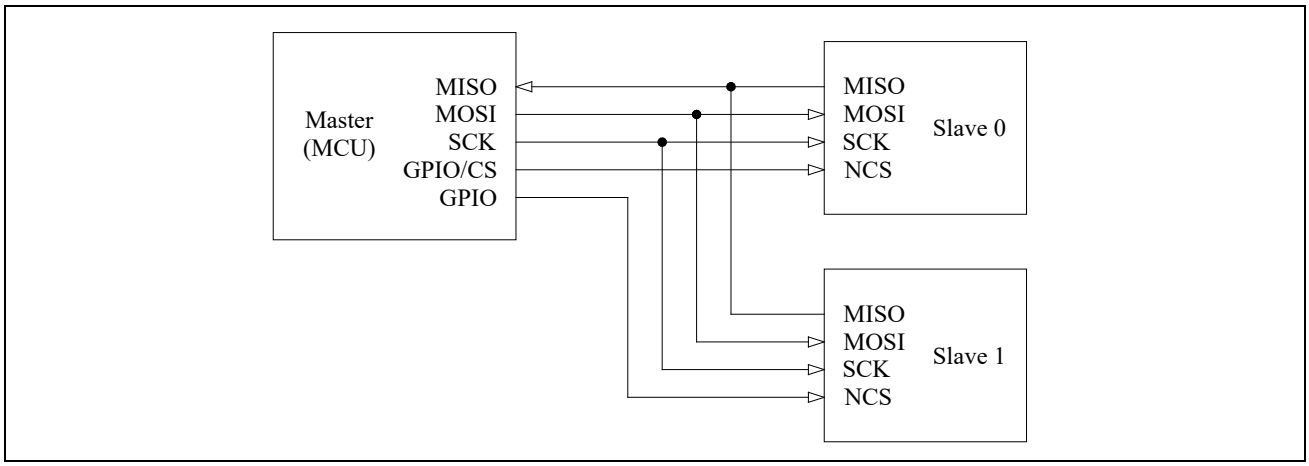
- 支持 SPI 主机模式、SPI 从机模式
- 支持标准四线全双工通信
- 支持配置串行时钟极性和相位
- 主机模式支持 7 种通信速率
- 主机模式最大分频系数为 PCLK/2
- 从机模式最大分频系数为 PCLK/4
- 帧长度固定为 8 比特，优先传送 MSB
- 支持 DMA 软件/硬件访问

8.3 SPI 功能描述

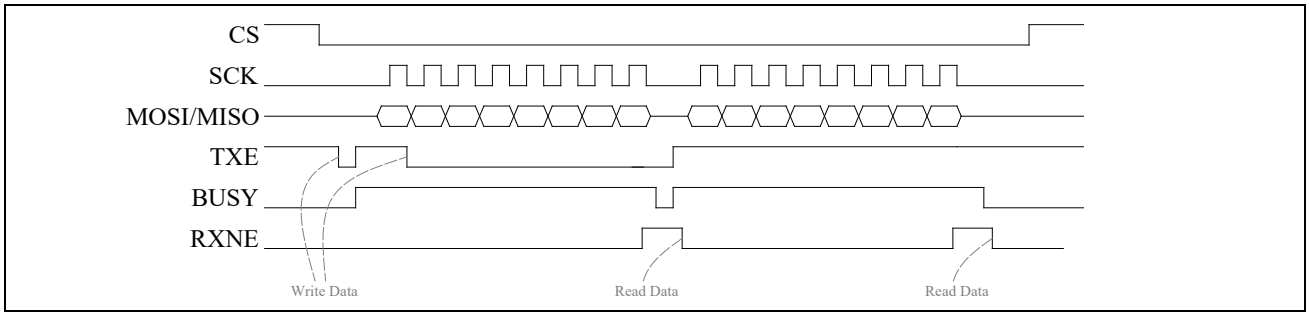
8.3.1 SPI 主机模式

每个数据帧的长度固定为 8 比特，发送数据的第一位固定为 MSB。设置 SPIx_CR.mstr 为 1，则 SPI 接口工作于主机模式。将数据写入 SPIx_Data 寄存器，即可启动 SPI 传输，SCK 引脚会自动产生串行时钟；在串行时钟边沿，移位寄存器中的数据被发送到 MOSI 引脚，MISO 引脚的数据被接收到移位寄存器中。SCK 引脚输出时钟的频率由 SPIx_CR[spr2:spr0]进行控制，其输出频率的范围为 PCLK/2~PCLK/128；CS 引脚的输出电平由 SPIx_SSN.ssn 进行控制，GPIO 引脚的输出电平由 GPIO 相关寄存器控制。

主机模式的典型应用框图如下所示。

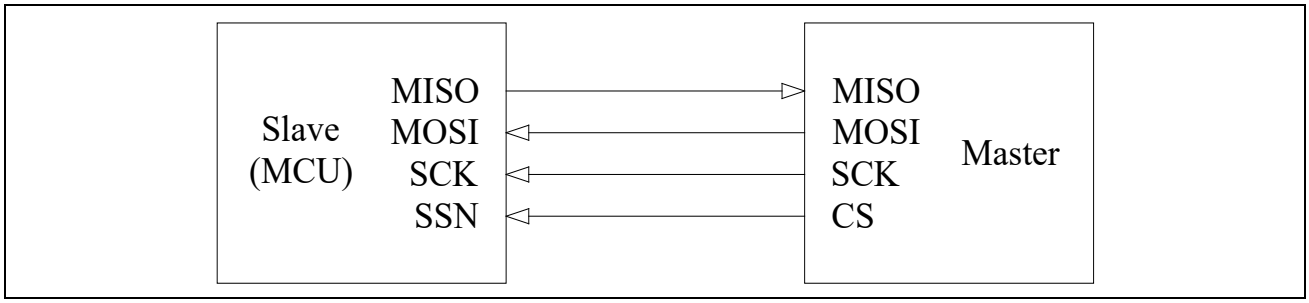


主机模式的通信示意如下图所示，其中 CPOL=0，CPHA=0。



8.3.2 SPI 从机模式

每个数据帧的长度固定为 8 比特，接收数据的第一位固定为 MSB。设置 SPIx_CR.mstr 为 0，则 SPI 接口工作于从机模式。该模式下，SCK 引脚作为输入脚且串行时钟来自于外部主机；SSN 引脚作为输入脚且片选信号来自外部主机或固定为低电平。SSN 引脚详见 GPIO 章节辅助寄存器。从机模式的典型应用框图如下所示。



当 SPI 从机从 SPI 主机接收到一字节数据时，RXNE 位会被置高；用户程序应尽快读取收到的数据。从机模式接收数据的通信时序如下所示，其中 CPOL=0，CPHA=0。

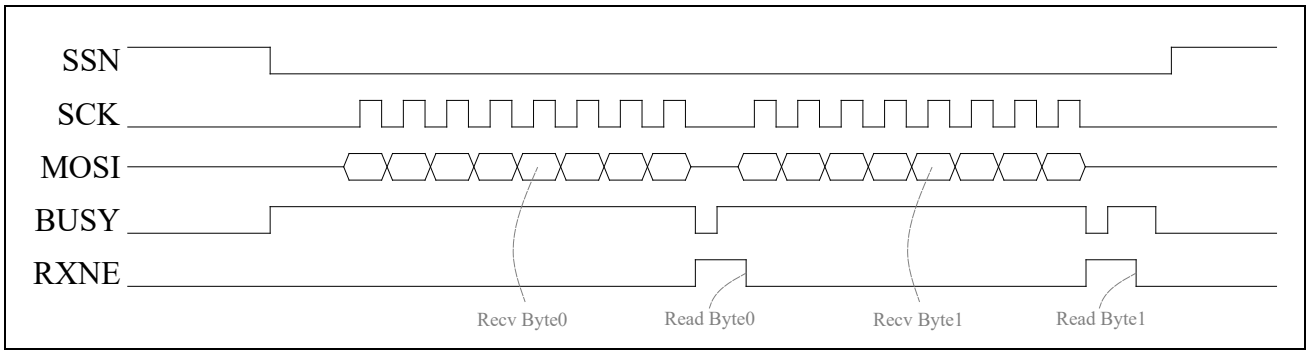


图 8-1 从机接收示意图

当 SPI 从机需要发送数据到主机时，在主机拉低 NSS 之前应依次进行如下五步操作：设置 PERI_RESET0.SPIx 位为 0、设置 PERI_RESET0.SPIx 位为 1、配置 SPI 通信参数、向 SPIx_DATA 寄存器中写入待发送的第一字节数据；在 NSS 拉低后，每当查询到 TXE 标志为 1 时，应尽快向 SPIx_DATA 寄存器中写入待发送的后续数据。从机模式发送数据的通信时序如下所示，其中 CPOL=0，CPHA=0。

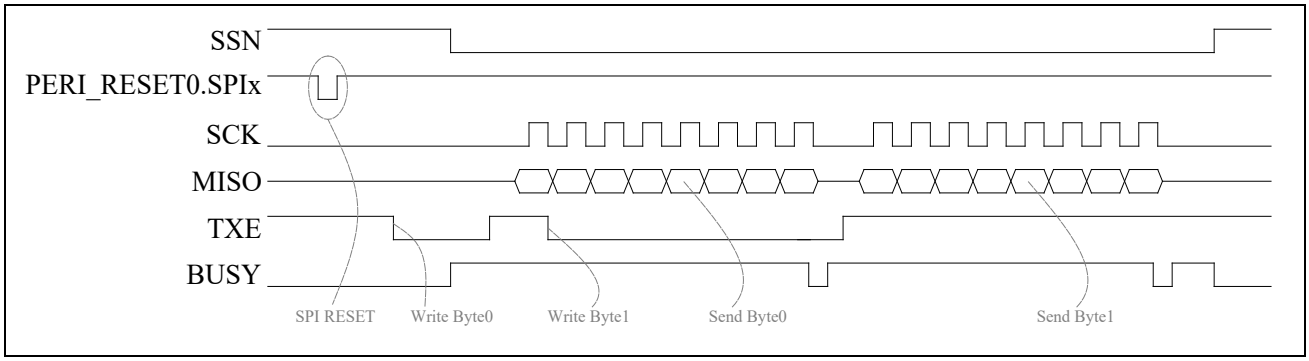


图 8-2 从机发送示意图

8.3.3 SPI 数据帧格式

SPI 接口帧格式取决于时钟极性位 CPOL 和时钟相位位 CPHA 的配置。

当 CPOL 为 0 时，SCK 线空闲状态为低电平。当 CPOL 为 1 时，SCK 线空闲状态为高电平。当 CPHA 为 0 时，数据会在第一个 SCK 时钟转换信号跳变时被采样。当 CPHA 为 1 时，数据会在第二个 SCK 时钟信号跳变时被采样。

SPI 接口主机帧格式如下图所示。

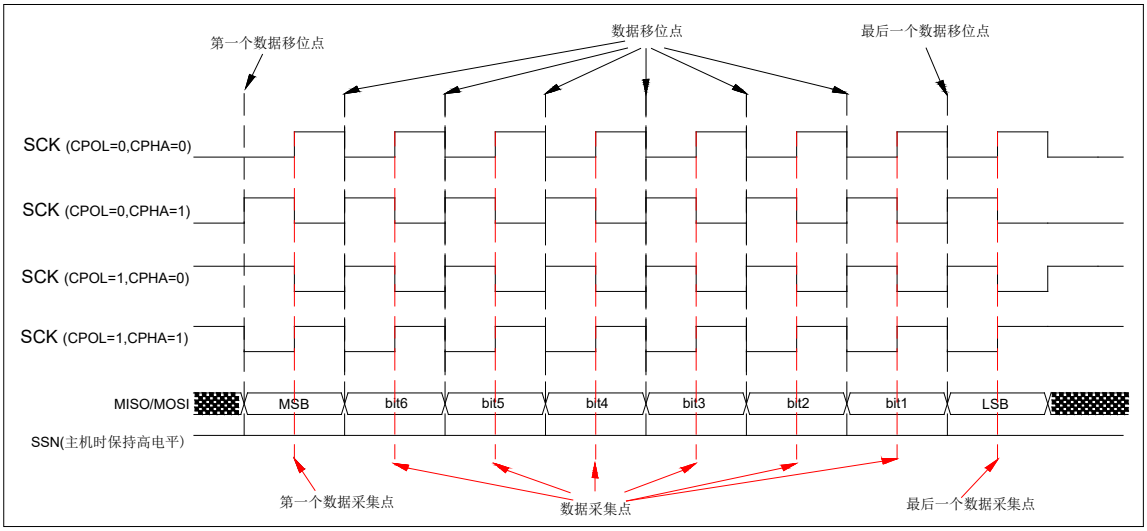


图 8-3 主机模式帧格式

SPI 接口从机帧格式如下图所示。

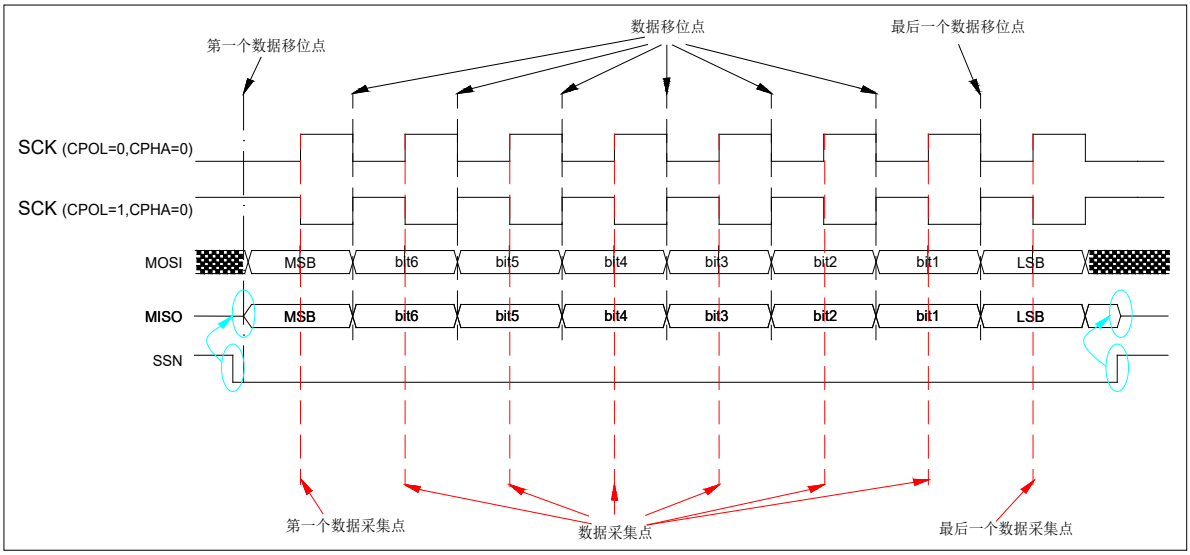


图 8-4 从机 CPHA 为 0 时数据帧格式

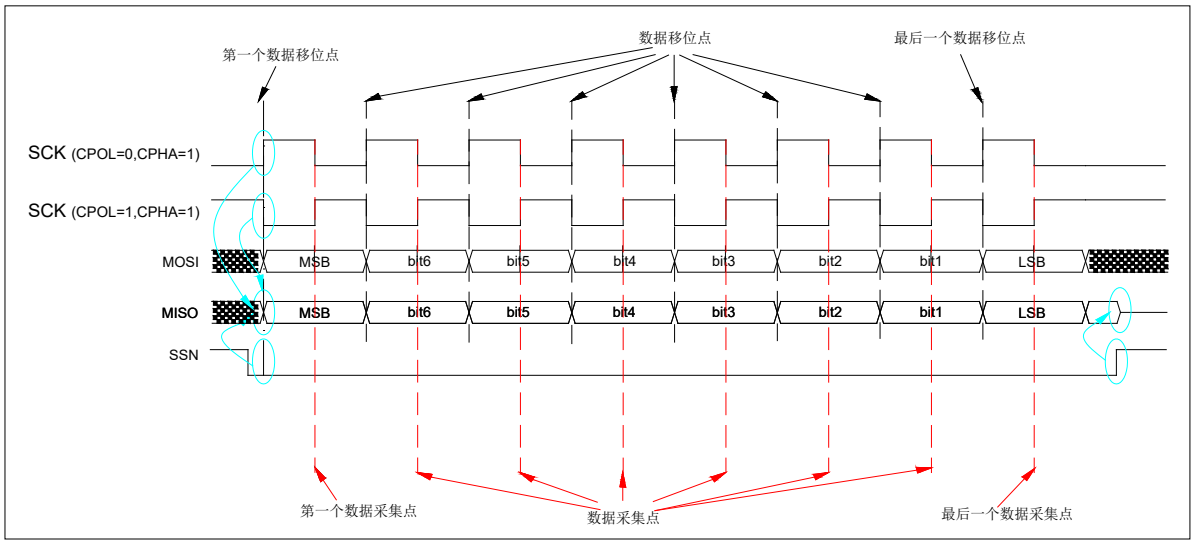


图 8-5　从机 CHPA 为 1 时数据帧格式

8.3.4 SPI 状态标志及中断

SPI 在工作中会产生如下四种状态标志，其产生条件及清除方法如下所示。

- 当发送缓冲器（SPIx_Data）里的数据被转移到发送移位寄存器后，SPIx_STAT.txe 会被硬件置位，说明发送缓冲器为空，可以写入下一笔数据。向 SPIx_DATA 写入数据即可清除该标志位。
- 当接收移位寄存器里的数据被转移到接收缓冲器(SPIx_Data)时，SPIx_STAT.rxne 会被硬件置位，说明接收缓冲器非空，需要用户尽快读取该数据。从 SPIx_DATA 读出数据即可清除该标志位。
- 当 SPI 工作于主机模式且外部 SSN 输入为低电平，SPIx_STAT.mdf 会被硬件置位，说明有其它 SPI 主机在占用总线。当 SSN 输入为高电平时，SPIx_STAT.mdf 会被硬件自动清除。
- 当 SPI 工作于从机模式,如果正在进行数据传输时 SSN 管脚被拉高,则 SPIx_STAT.sserr 会被置位。设置 SPIx_CR.spen 为 0 可以清除该标志。

如果允许 SPI 中断（SPIx_CR2.int_en=1），以下三种情况均可产生中断：

- SPI 发送缓冲器为空，即 SPIx_STAT.txe 为 1
- SPI 接收缓冲器非空，即 SPIx_STAT.rxne 为 1
- SPI 的主机模式错误，即 SPIx_STAT.mdf 为 1

在中断服务程序中需要对 SPIx_ICLR 写入 0x00 以清除内部中断标志。

8.3.5 SPI 多机系统配置说明

- 当 SPI 模块作为主机且工作于单主机系统时，可通过 SPI_CS 管脚或 GPIO 管脚控制从机。当选择 SPI_CS 管脚作为从机的片选信号时，设置 SPIx_SSN.ssn 为 0 即可选中从机，设置 SPIx_SSN.ssn 为 1 即可释放从机。当选择 GPIO 管脚作为从机的片选信号时，设置 GPIOx_OUT 寄存器相应的比特为 0 即可选中从机，设置 GPIOx_OUT 寄存器相应的比特为 1 即可释放从机。
- 当 SPI 模块从为从机时，根据需要配置 SPI_SSN 的来源（详见 GPIO 端口辅助控制器）。当 SSN 为低时，即可选中本从机以进行通信；当 SSN 为高时，则本机处于未选中态。
- 当 SPI 模式工作于多主机多从机时，所有的从机片选信号都通过 GPIO 管脚连接，主机还必须通过 GPIO 管脚和其他主机的 SSN 信号相连，来监测总线是否被占用。下图所示的 Master0 需要进行通信的操作方法为：等待 Master0.SSN 变为高；从 GPIO0 输出低以通知 Master1 释放 SPI 总线；从 GPIO2 输出低以选中 Slave1；与 Slave1 进行通信；从 GPIO2 输出高以释放 Slave1；从 GPIO0 输出高以释放 Master1。

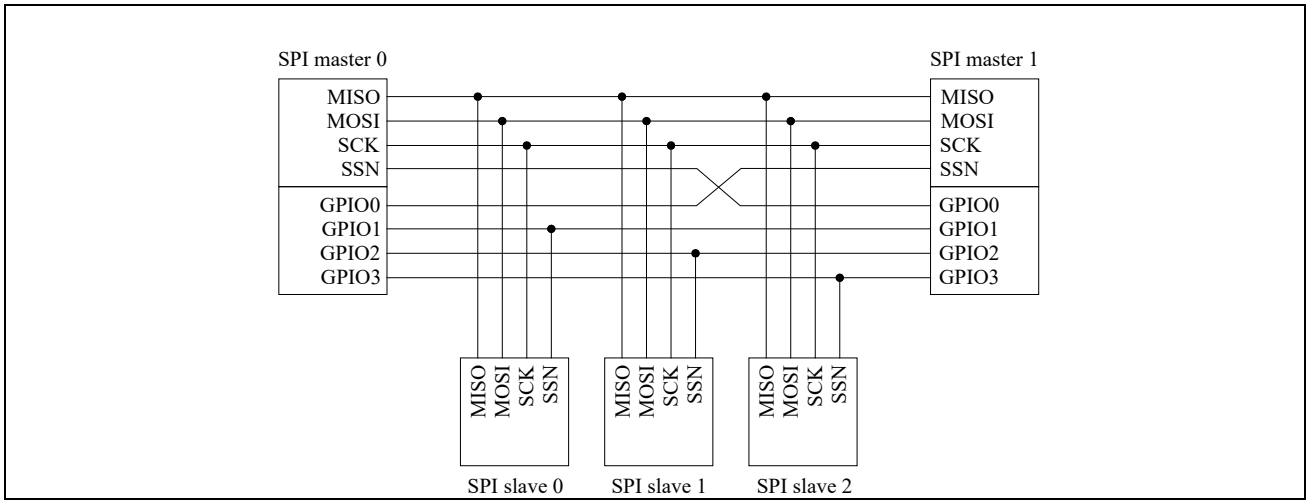


图 8-6　SPI 多主机/多从机系统的示意图

8.3.6 SPI 管脚配置说明

SPI 可以在一些特殊的管脚配置下保持部分或全部的功能。

具体情况如下表（“√”代表该管脚被配置并使用，空白代表该管脚未被配置）：

表 8-1 SPI 管脚配置说明表

	SPI_CS(主)/ SPI_SSN(从)	SCK	MOSI	MISO	功能说明
主机模式	√	√	√	√	一般配置 全部主机功能正常
		√	√	√	全部主机功能正常
	√	√	√		主机发送功能正常
	√	√		√	主机接收功能正常
		√	√		主机发送功能正常
		√		√	主机接收功能正常
从机模式	√	√	√	√	一般配置 全部从机功能正常
	√	√	√		从机接收功能正常
	√	√		√	从机发送功能正常
	√ 固定低电平	√	√	√	全部从机功能正常
	√ 固定低电平	√	√		从机接收功能正常
	√ 固定低电平	√		√	从机发送功能正常

注意：

- 表中未列举的情况暂不支持。
- 主机模式下，即使不使用 SPIx_CS 片选输出，发送数据前需要设置 SPIx.SSN 为 1，发送数据后需要设置 SPIx.SSN 为 0。
- 从机模式且片选输入固定为低电平时，为维持正常功能，必须满足 SPIx_CR.cpha=1。

8.4 SPI 编程示例

8.4.1 SPI 主机发送示例

Step1：按 GPIO 章节管脚数字复用功能的相关描述，将 CS/SCK/MISO/MOSI 映射到需要的管脚；并配置 CS/SCK/MOSI 管脚为输出模式，配置 MISO 管脚为输入模式。

Step2：设置 SPIx_CR.mstr 为 1，使 SPI 工作于主机模式。

Step3：配置 SPIx_CR[spr2:spr0]，使 SCK 输出的时钟速率符合应用需求。

Step4：配置 SPIx_CR.cpol 及 SPIx_CR.cpha，使数据帧格式符合应用需求。

Step5：设置 SPIx_CR.spen 为 1，使能 SPI 接口。

Step6：设置 SPIx_SSN.ssn 为 0，使 CS 管脚输出低电平以选中从机。

Step7：当查询到 SPIx_STAT.txe 为 1 时，将待发送的数据尽快写入 SPIx_DATA。

Step8：如待发送的数据未完成，则跳转到 Step7 继续执行。

Step9：查询等待 SPIx_STAT.txe 变为 1，最后一个字节数据已开始发送。

Step10：查询等待 SPIx_STAT.busy 变为 0，SPI 总线空闲。

Step11：设置 SPIx_SSN.ssn 为 1，使 CS 管脚输出高电平以释放从机。

注意：

- 可使用 GPIO 代替 CS 实现片选输出，多用于多机通信系统。
- 在传输过程必需设置 SPIx_SSN.ssn 为 0，传输完成后必需设置 SPIx_SSN.ssn 为 1。

8.4.2 SPI 主机接收示例

Step1：按 GPIO 章节管脚数字复用功能的相关描述，将 CS/SCK/MISO/MOSI 映射到需要的管脚；并配置 CS/SCK/MOSI 管脚为输出模式，配置 MISO 管脚为输入模式。

Step2：设置 SPIx_CR.mstr 为 1，使 SPI 工作于主机模式。

Step3：配置 SPIx_CR[spr2:spr0]，使 SCK 输出的时钟速率符合应用需求。

Step4：配置 SPIx_CR.cpol 及 SPIx_CR.cpha，使数据帧格式符合应用需求。

Step5：设置 SPIx_CR.spen 为 1，使能 SPI 接口。

Step6：设置 SPIx_SSN.ssn 为 0，使 CS 管脚输出低电平以选中从机。

Step7：向 SPIx_DATA 写入任意数据以触发主机发送 SCK。

Step8：查询等待 SPIx_STAT.rxne 变为 1，已收到从机发送的数据。

Step9：从 SPIx_DATA 读出收到的数据。

Step10：如果待接收的数据未完成，则跳转到 Step7 继续执行。

Step11：设置 SPIx_SSN.ssn 为 1，使 CS 管脚输出高电平以释放从机。

注意：

- 可使用 GPIO 代替 CS 实现片选输出，多用于多机通信系统。
- 在传输过程必需设置 SPIx_SSN.ssn 为 0，传输完成后必需设置 SPIx_SSN.ssn 为 1。

8.4.3 SPI 从机发送示例

Step1：按 GPIO 章节管脚数字复用功能的相关描述，将 SSN/SCK/MISO/MOSI 映射到需要的管脚；并配置 SSN/SCK/MOSI 管脚为输入模式，配置 MISO 管脚为输出模式。SSN 管脚来源详见 GPIO 端口辅助控制器。

Step2：设置 PERI_RESET0.SPIx 为 0，使 SPI 模块处于复位状态。

Step3：设置 PERI_RESET0.SPIx 为 1，使 SPI 模块处于工作状态。

Step4：设置 SPIx_CR.mstr 为 0，使 SPI 工作于从机模式。

Step5：配置 SPIx_CR.cpol 及 SPIx_CR.cpha，使数据帧格式符合应用需求。

Step6：设置 SPIx_CR.spen 为 1，使能 SPI 接口。

Step7：将待发送的第一个字节数据写入 SPIx_DATA。

Step8：查询等待 SSN 管脚拉低，主机选中 SPI 从机。

Step9：当查询到 SPIx_STAT.txe 为 1 时，将待发送的数据尽快写入 SPIx_DATA。

Step10：当查询到 SSN 管脚为低且待发送的数据尚未完成，跳转到 Step9。

Step11：查询等待 SSN 管脚拉高，主机释放 SPI 从机。

注意：

- 每当 SPI 从机需要发送数据时，均需从 Step2 开始执行初始化操作及发送操作。

8.4.4 SPI 从机接收示例

Step1：按 GPIO 章节管脚数字复用功能的相关描述，将 SSN/SCK/MISO/MOSI 映射到需要的管脚；并配置 SSN/SCK/MOSI 管脚为输入模式，配置 MISO 管脚为输出模式。SSN 管脚来源详见 GPIO 端口辅助控制器。

Step2：设置 SPIx_CR.mstr 为 0，使 SPI 工作于从机模式。

Step3：配置 SPIx_CR.cpol 及 SPIx_CR.cpha，使数据帧格式符合应用需求。

Step4：设置 SPIx_CR.spen 为 1，使能 SPI 接口。

Step5：查询等待 SSN 管脚拉低，主机选中 SPI 从机。

Step6：查询等待 SPIx_STAT.rxne 变为 1，已收到主机发送的数据。

Step7：从 SPIx_DATA 读出收到的数据。

Step8：如果待接收的数据未完成，则跳转到 Step6 继续执行。

Step9：查询等待 SSN 管脚拉高，主机释放 SPI 从机。

8.5 SPI 寄存器描述

寄存器列表

SPI0 基地址：0x40000800

SPI1 基地址：0x40004800

表 8-2 SPI 寄存器列表

偏移量	寄存器名称	访问	寄存器描述
0x00	SPIx_CR	RW	SPIx 配置寄存器
0x04	SPIx_SSN	RW	SPIx 片选配置寄存器
0x08	SPIx_STAT	RO	SPIx 状态寄存器
0x0c	SPIx_DATA	RW	SPIx 数据寄存器
0x10	SPIx_CR2	RW	SPIx 配置寄存器 2
0x14	SPIx_ICLR	WO	SPIx 中断清除寄存器

8.5.1 SPI 配置寄存器(SPIx_CR)

地址偏移量：0x00

复位值：0x0000 0014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								spr2	spen	Res	mstr	cpol	cpha	spr1	spr0
								RW	RW		RW	RW	RW	RW	RW

位	标记	功能描述
31:8	Reserved	
7	spr2	波特率选择位2 参考spr0.
6	spen	SPI模块使能控制 0 – 禁止 1 – 使能
5	Reserved	
4	mstr	SPI工作模式配置 0 – 从机模式 1 – 主机模式
3	cpol	SCK线空闲状态配置 0 – 低电平 1 – 高电平
2	cpha	时钟相位配置 0 – 第一边沿 1 – 第二边沿
1	spr1	波特率选择位1 参考spr0
0	spr0	波特率选择位0

表 8-3 主机模式波特率选择

spr2	spr1	spr0	SCK Rate
0	0	0	PCLK /2
0	0	1	PCLK /4
0	1	0	PCLK /8
0	1	1	PCLK /16
1	0	0	PCLK /32
1	0	1	PCLK /64
1	1	0	PCLK /128
1	1	1	Reserved

8.5.2 SPI 片选配置寄存器(SPIx_SSN)

地址偏移量：0x04

复位值：0x000000FF

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

位	标记	功能描述
31:1	Reserved	
0	ssn	主机模式时SPI_CS 输出电平配置 0：SPI_CS 端口输出低电平 1：SPI_CS 端口输出高电平

8.5.3 SPI 状态寄存器(SPIx_STAT)

地址偏移量：0x08

复位值：0x00000004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								spif	Res.	sserr	mdf	busy	txe	rxne	Res.
								RO		RO	RO	RO	RO	RO	

位	标记	功能描述
31:8	Reserved	
7	spif	接收完成标志 1：已从SPI总线上接收到一个字节 0：正在接收中
6	Reserved	
5	sserr	从机模式SSN错误标志
4	mdf	主机模式时，冲突标志 1：SSN管脚电平为低 0：SSN管脚电平为高
3	busy	SPI总线传输状态标志 1：SPI总线正在传输数据 0：SPI总线空闲
2	txe	发送缓冲器状态标志 1：发送缓冲器空 0：发送缓冲器非空
1	rxne	接收缓冲器状态标志 1：接收缓冲器非空 0：接收缓冲器空
0	Reserved	

8.5.4 SPI 数据寄存器(SPIx_DATA)

地址偏移量：0x0c

复位值：0x00000000

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

位	标记	功能描述
31:8	Reserved	
7:0	spdat	数据寄存器 在发送模式，向这个寄存器写入待发送的字节； 在接收模式，从这个寄存器读取接收到的字节；

8.5.5 SPI 配置寄存器 2(SPIx_CR2)

地址偏移量：0x10

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									rxneie	txeie	hdma_tx	hdma_rx	int_en	Reserved	
									RW	RW	RW	RW	RW		

位	标记	功能描述
31:7	Reserved	
6	rxneie	接收缓冲器非空中断使能 0 – 禁止 1 – 使能
5	txeie	发送缓冲器空中断使能 0 – 禁止 1 – 使能
4	hdma_tx	DMA硬件访问发送使能. 0 – 禁止 1 – 使能
3	hdma_rx	DMA硬件访问接收使能. 0 – 禁止 1 – 使能
2	int_en	SPI中断使能. 0 – 禁止 1 – 使能
1:0	Reserved	请保持这两位的值均为1。

8.5.6 SPI 中断清除寄存器 2(SPIx_ICLR)

地址偏移量：0x14

复位值：0x0000 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															int_clr
															R1W0

位	标记	功能描述
31:1	Reserved	
0	int_clr	SPI中断清除 0 – 清除 1 – 保持

9 时钟校准模块（CLKTRIM）

9.1 CLKTRIM 简介

CLKTRIM(Clock Trimming)模块是一个专门用来校准/监测时钟的电路。在校准模式下选择精准的时钟源来校准不精准的时钟源，反复校准，调节不精准时钟源的参数，直到被校准时钟源的频率达到精度要求。校准模式下计数值会有一定的误差，但是在允许的精度误差范围内。在监测模式下选择稳定的时钟源来监测系统工作时钟，在设定的监测周期下，监测系统工作时钟是否有失效的情况发生并产生中断。在校准模式和监控模式下，所需的时钟源都必须初始化和使能，具体的配置过程请参考【系统控制器（SYSCTRL）】章节。

9.2 CLKTRIM 主要特性

CLKTRIM 支持以下特性：

- 校准模式
- 监测模式
- 32 位参考时钟计数器可加载初值
- 32 位待校准时钟计数器可配置溢出值
- 6 种参考时钟源
- 6 种待校准时钟源
- 支持中断方式

9.3 CLKTRIM 功能描述

9.3.1 CLKTRIM 校准模式

9.3.1.1 时钟校准原理

在校准一个频率不精准，但是有参数设置可以调节频率的时钟（待校准时钟 CAL_CLK）时，需要一个精准的时钟作为参考时钟(REF_CLK)。设定校准时间 Ttrim，待校准时钟频率 Fcal，参考时钟频率 Fref，使用待校准时钟和参考时钟同时计数，在校准时间结束时停止计数。读取待校准时钟计数器的值 M，参考时钟计数器的值 N，

得到等式 Ttrim = M/Fcal = N/Fref

推导出 Fcal = Fref * M / N

由公式判断出，参考时钟的频率误差率越小，待校准时钟的误差率就越小。

计算出待校准时钟频率后，如果误差率超出系统要求的范围，可以调节待校准时钟的参数后，再次重复上面的步骤校准，直到待校准时钟频率的误差率满足系统的要求。

9.3.1.2 时钟校准模块硬件结构

时间校准模块的待校准时钟有 6 个时钟源，参考时钟有 6 个时钟源，如图 9-1 所示：

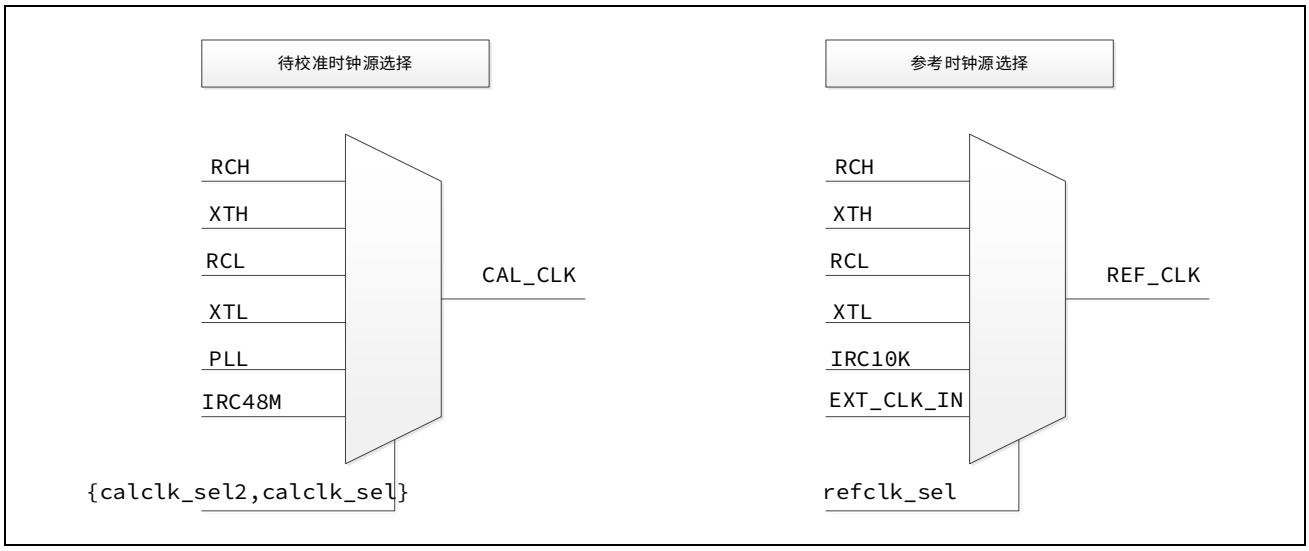


图 9-1 时钟源选择示意图

待校准时钟的选择由寄存器 CLKTRIM_CR.calclk_sel2,CLKTRIM_CR.calclk_sel 来配置。

参考时钟的选择由寄存器 CLKTRIM_CR.refclk_sel 来配置。

如图 9-2 所示，时钟校准模块有 2 个 32 位计数器，1 个是以参考时钟为时钟，可配置初值的减计数器,初值由寄存器 CLKTRIM_REFCON .rcntval 来配置,计数器值可以从寄存器 CLKTRIM_REFCNT 读出，当计数器减计数到 0 时，停止计数并产生中断。1 个是以待校准时钟为时钟，可配置溢出值的加计数器，溢出值由寄存器 CLKTRIM_CALCON. ccntval 来配置，计数器值可以从寄存器 CLKTRIM_CALCNT 读出，当计数器加计数到溢出值时，停止计数并产生中断。

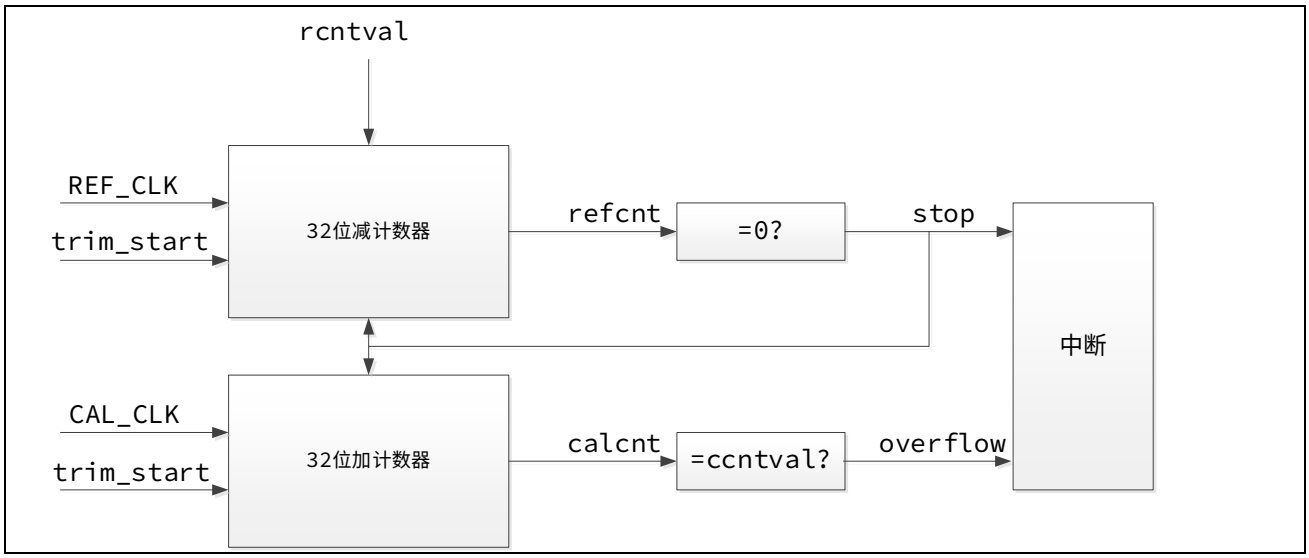


图 9-2 时钟校准模块硬件示意图

9.3.1.3 时钟校准软件流程

1. 设置 CLKTRIM_CR .refclk_sel 寄存器选择参考时钟。
2. 设置 CLKTRIM_CR .calclk_sel 寄存器选择被校准时钟。
3. 设置 CLKTRIM_REFCON .rcntval 寄存器为校准时间。
4. 设置 CLKTRIM_CR .IE 寄存器使能中断。
5. 设置 CLKTRIM_CR .trim_start 寄存器开始校准。
6. 参考时钟计数器和待校准时钟计数器开始计数。
7. 当参考时钟计数器从初始值减计数到 0 时，CLKTRIM_IFR.stop 置 1，触发中断。
8. 中断服务子程序判断 CLKTRIM_IFR.stop 为 1，读取寄存器 CLKTRIM_REFCNT 和 CLKTRIM_CALCNT 的值，
9. 清零 CLKTRIM_CR .trim_start 寄存器结束校准。
10. 根据时钟校准原理章节提到的公式计算出待校准时钟的频率，

    其中 M = 寄存器 CLKTRIM_CALCNT 的值，

    N = 寄存器 CLKTRIM_REFCON .rcntval 的值

    当待校准时钟的频率不满足误差率要求时，调整待校准时钟参数，

    重新执行步骤 5~10，直到待校准时钟满足误差率要求。

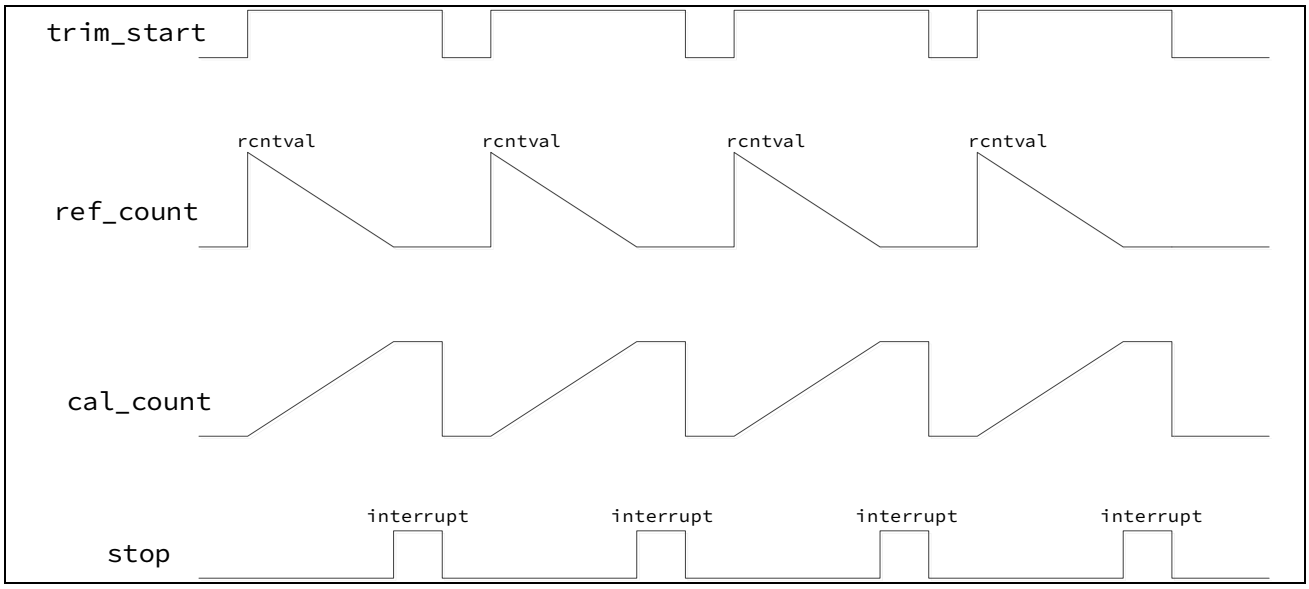


图 9-3 时钟校准波形示意图

注意：

- 校准模式在校准过程中有可能因为校准时间设置过长，发生待校准时钟计数器在 CLKTRIM_IFR.stop 置 1 之前溢出的情况，CLKTRIM_IFR. calcnt_of 置 1，触发中断。中断服务子程序发现 CLKTRIM_IFR. calcnt_of 置 1 时，清零 CLKTRIM_CR .trim_start 寄存器结束校准。

这种情况下校准是无法正确进行的，必须调整校准时间，重新校准。

具体步骤是：

1． 设置 CLKTRIM_REFCON .rcntval 寄存器调整校准时间。

2． 设置 CLKTRIM_CR .trim_start 寄存器重新开始校准。

9.3.2 CLKTRIM 监测模式

9.3.2.1 时钟监测原理

为了监测系统工作时钟（被监测时钟 CAL_CLK）是否正常工作时，需要一个稳定的时钟作为参考时钟(REF_CLK)。设定监测时间 Ttrim，被监测时钟频率 Fcal，参考时钟频率 Fref，被监测时钟计数器溢出值 CALVAL，使用被监测时钟和参考时钟同时计数，在监测时间结束时停止计数，判断被监测时钟计数器是否处于溢出状态。如果被监测时钟计数器已溢出，表示在监测时间内被监测时钟工作正常，继续进行下一次监测。如果被监测时钟计数器没有溢出，表示在监测时间内被监测时钟工作异常，可能被监测时钟已经停止或者频率大幅变化，产生被监测时钟工作异常中断，由硬件切换系统工作时钟。

9.3.2.2 时钟监测硬件结构

时间校准模块的被监测时钟有 3 个时钟源，参考时钟有 6 个时钟源，如下图所示：

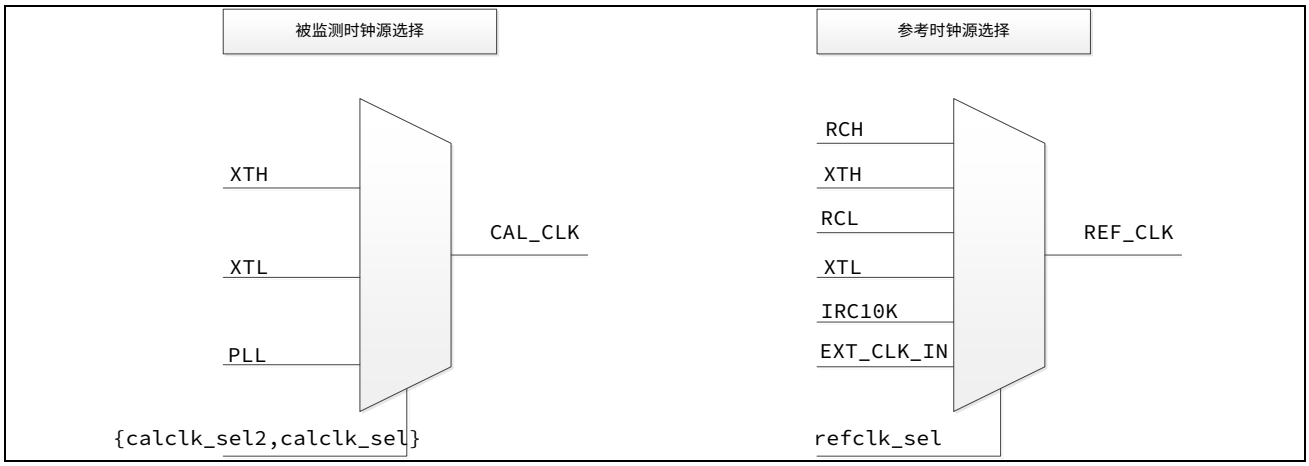


图 9-4 CLKTRIM 时钟选择

被监测时钟的选择由寄存器 CLKTRIM_CR.calclk_sel2,CLKTRIM_CR.calclk_sel 来配置，参考时钟的选择由寄存器 CLKTRIM_CR.refclk_sel 来配置。

9.3.2.3 时钟监测软件流程

1. 设置 CLKTRIM_CR .refclk_sel 寄存器选择参考时钟。
2. 设置 CLKTRIM_CR .calclk_sel 寄存器选择被监控时钟。
3. 设置 CLKTRIM_REFCON .rcntval 寄存器为监控间隔时间。
4. 设置 CLKTRIM_CALCON. ccntval 寄存器为被监控时钟计数器溢出时间。
5. 设置 CLKTRIM_CR .mon_en 寄存器使能监控功能。
6. 设置 CLKTRIM_CR .IE 寄存器使能中断。
7. 设置 CLKTRIM_CR .trim_start 寄存器开始监控。
8. 参考时钟计数器和被监控时钟计数器开始计数。
9. 当参考时钟计数器计数到达监控间隔时间时，CLKTRIM_IFR.stop 置 1，同时判断被监控时钟计数器是否溢出，即 CLKTRIM_IFR. calcnt_of 是否置 1。如果溢出，即 CLKTRIM_IFR. calcnt_of 为 1，表示被监控时钟工作正常。如果没有溢出，即 CLKTRIM_IFR. calcnt_of 为 0，表示被监控时钟失效，CLKTRIM_IFR .xtl_fault/xth_fault 置 1，触发中断。
10. 处理中断服务子程序，清除中断标志位 CLKTRIM_IFR .xtl_fault/xth_fault，清零 CLKTRIM_CR .trim_start 寄存器结束监测。

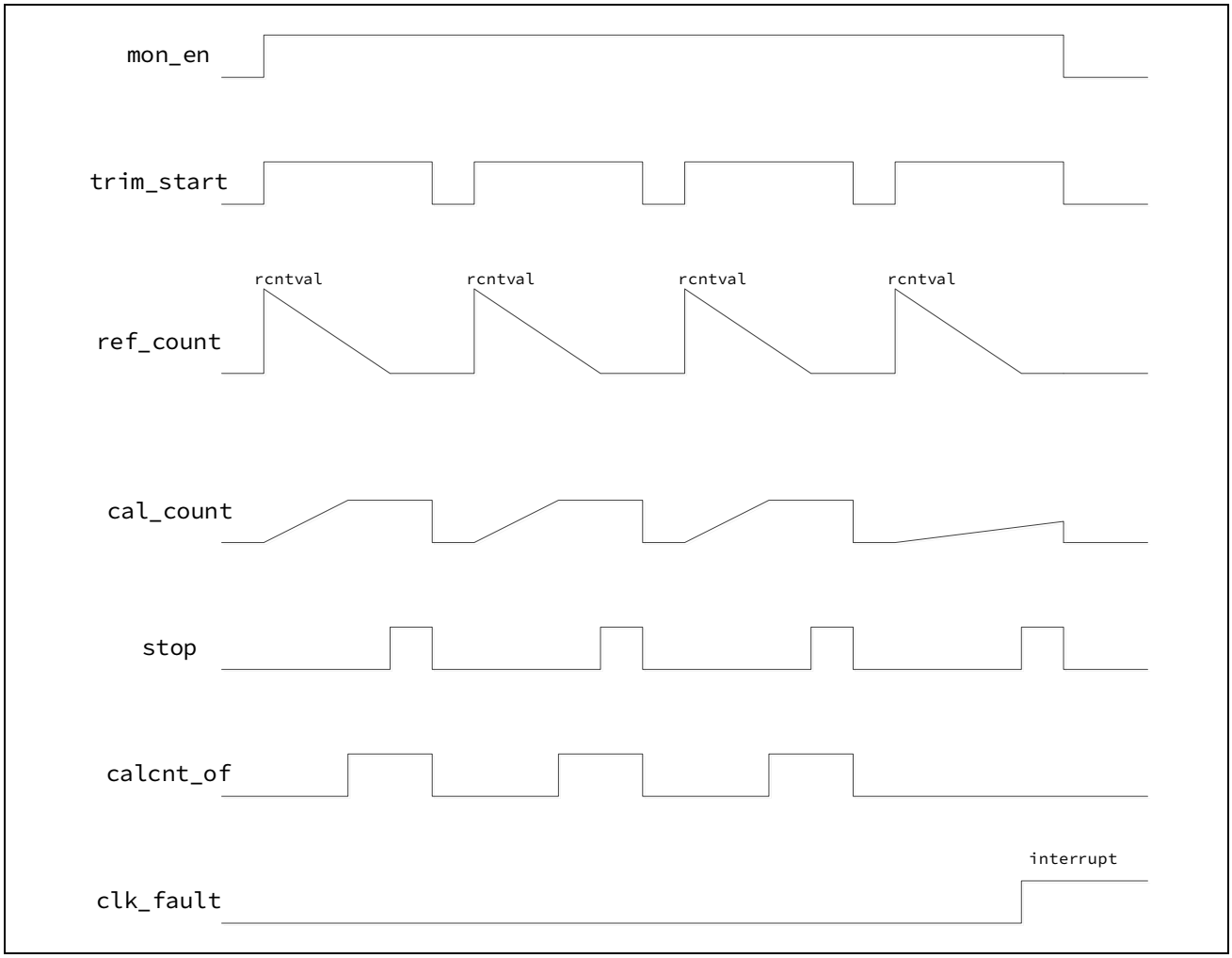


图 9-5　时钟监控波形示意图

9.4 CLKTRIM 寄存器描述

寄存器列表

基地址：0x40001800

表 9-1 寄存器列表

偏移量	寄存器名称	访问	寄存器描述
0x00	CLKTRIM_CR	RW	配置寄存器
0x04	CLKTRIM_REFCON	RW	参考计数器初值配置寄存器
0x08	CLKTRIM_REFCNT	RO	参考计数器值寄存器
0x0c	CLKTRIM_CALCNT	RO	校准计数器值寄存器
0x10	CLKTRIM_IFR	RO	中断标志位寄存器
0x14	CLKTRIM_ICLR	RW	中断标志位清除寄存器
0x18	CLKTRIM_CALCON	RW	校准计数器溢出值配置寄存器

9.4.1 配置寄存器(CLKTRIM_CR)

地址偏移量：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							calclk_sel2	IE	mon_en	calclk_sel		refclk_sel			trim_start
							RW	RW	RW	RW		RW			RW

位	标记	功能描述
31:9	Reserved	
8	calclk_sel2	待校准/监测时钟选择高位寄存器
7	IE	中断使能寄存器 0 – 禁止 1 – 使能
6	mon_en	监测模式使能寄存器 0 – 禁止 1 – 使能
5:4	calclk_sel	待校准/监测时钟选择低位寄存器 calclk_sel2，calclk_sel 000 RCH 001 XTH 010 RCL 011 XTL 100 PLL 101 IRC48M
3:1	refclk_sel	参考时钟选择寄存器 000 ---- RCH 001 ---- XTH 010 ---- RCL 011 ---- XTL 100 ---- IRC10K 101 ---- EXT_CLK_IN
0	trim_start	校准/监测开始寄存器 0 – 停止 1 – 开始

9.4.2 参考计数器初值配置寄存器(CLKTRIM_REFCON)

地址偏移量：0x04

复位值：0x00000000

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

位	标记	功能描述
31:0	rcntval	参考计数器初始值

9.4.3 参考计数器值寄存器(CLKTRIM_REFCNT)

地址偏移量：0x08

复位值：0x00000000

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

位	标记	功能描述
31:0	refcnt	参考计数器值

9.4.4 校准计数器值寄存器(CLKTRIM_CALCNT)

地址偏移量：0x0c

复位值：0x00000000

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

位	标记	功能描述
31:0	calcnt	校准计数器值

9.4.5 中断标志位寄存器(CLKTRIM_IFR)

地址偏移量：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											pll_fault	xth_fault	xtl_fault	calcnt_of	stop
											RO	RO	RO	RO	RO

位	标记	功能描述
31:5	Reserved	
4	pll_fault	PLL失效标志. CLKTRIM_ICLR.pll_fault_clr写零清除此标志位
3	xth_fault	XTH 失效标志. CLKTRIM_ICLR.xth_fault_clr写零清除此标志位
2	xtl_fault	XTL 失效标志. CLKTRIM_ICLR.xtl_fault_clr写零清除此标志位
1	calcnt_of	校准计数器溢出标志. CLKTRIM_CR.start写零清除此标志位
0	stop	参考计数器停止标志. CLKTRIM_CR.start写零清除此标志位

9.4.6 中断标志位清除寄存器(CLKTRIM_ICLR)

地址偏移量：0x14

复位值：0x1111 1111

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											pll_fault_clr	xth_fault_clr	xtl_fault_clr	Reserved	
											R1W0	R1W0	R1W0		

位	标记	功能描述
31:5	Reserved	
4	pll_fault_clr	清除 PLL 失效标志，写零清除。
3	xth_fault_clr	清除 XTH 失效标志，写零清除。
2	xtl_fault_clr	清除 XTL 失效标志，写零清除。
1:0	Reserved	

9.4.7 校准计数器溢出值配置寄存器(CLKTRIM_CALCON)

地址偏移量：0x18

复位值：0xffff ffff

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

位	标记	功能描述
31:0	ccntval	校准计数器溢出值

10 硬件除法器模块（HDIV）

10.1HDIV 简介

HDIV（Hardware Divider）是一个 32 位有符号/无符号整数硬件除法器。

10.2HDIV 主要特性

HDIV 硬件除法器支持以下特性：

- 可配置有符号/无符号整数除法计算
- 32 位被除数，16 位除数
- 输出 32 位商和 32 位余数
- 除数为零警告标志位，除法运算结束标志位
- 10 个时钟周期完成一次除法运算
- 写除数寄存器触发除法运算开始
- 读商寄存器/余数寄存器时自动等待计算结束

10.3HDIV 功能描述

10.3.1 HDIV 操作流程

1. 在系统控制器里打开硬件除法器的时钟使能寄存器。
2. 配置寄存器 HDIV_SIGN 设置有符号/无符号除法运算。
3. 配置寄存器 HDIV_DIVIDEND 设置被除数。
4. 配置寄存器 HDIV_DIVISOR 设置除数。
5. 除法运算开始，查询寄存器 HDIV_STAT 运算结束标志位 div_end，div_end 为 1 标志运算结束。读寄存器 HDIV_QUOTIENT 得到商，读寄存器 HDIV_REMAINDER 得到余数。
6. 当除数为零时，除法运算立即结束，运算结果保持上一次运算的结果，同时除数为零警告标志位 div_zero 被置起。
7. 在除法运算结束之前，读寄存器 HDIV_QUOTIENT/HDIV_REMAINDER 时，CPU 将被保持直到运算结束。

举例：计算一个无符号除法，被除数为 1917887483(0x7250A3FB)，除数为 9597(0x257D)

步骤一，配置寄存器 HDIV_SIGN 为 0，即无符号除法运算

步骤二，配置寄存器 HDIV_DIVIDEND 为 0x7250A3FB，即设置被除数

步骤三，配置寄存器 HDIV_DIVISOR 为 0x257D，即设置除数，计算开始

步骤四，查询寄存器 HDIV_STAT 运算结束标志位 div_end，div_end 为 1 标志，运算结束。

读寄存器 HDIV_QUOTIENT 得到商 199842(0x30CA2)

读寄存器 HDIV_REMAINDER 得到余数 3809(0xEE1)

10.4HDIV 寄存器描述

寄存器列表　　基地址：0x40021800

表 10-1　寄存器列表

偏移量	寄存器名称	访问	寄存器描述
0x00	HDIV_DIVIDEND	RW	被除数寄存器.
0x04	HDIV_DIVISOR	RW	除数寄存器.
0x08	HDIV_QUOTIENT	RO	商寄存器.
0x0c	HDIV_REMAINDER	RO	余数寄存器.
0x10	HDIV_SIGN	RW	符号寄存器
0x14	HDIV_STAT	RO	状态寄存器

10.4.1 被除数寄存器(HDIV_DIVIDEND)

地址偏移量：0x00

复位值：0x0000 0000

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

位	标记	功能描述
31:0	DIVIDEND	被除数值寄存器

10.4.2 除数寄存器(HDIV_DIVISOR)

地址偏移量：0x04

复位值：0x00000001

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

位	标记	功能描述
31:16	Reserved	
15:0	DIVISOR	除数值寄存器（写此寄存器自动触发除法运算）

10.4.3 商寄存器(HDIV_QUOTIENT)

地址偏移量：0x08

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
QUOTIENT[31:16]															
RO															

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

位	标记	功能描述
31:0	QUOTIENT	商结果寄存器

10.4.4 余数寄存器(HDIV_REMAINDER)

地址偏移量：0x0c

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REMAINDER[31:16]															
RO															

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

位	标记	功能描述
31:0	REMAINDER	余数结果寄存器

10.4.5 符号寄存器(HDIV_SIGN)

地址偏移量：0x10

复位值：0x00000000

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

位	标记	功能描述
31:1	Reserved	
0	sign	符号选择寄存器。 0 --- 无符号除法运算 1 --- 有符号除法运算

10.4.6 状态寄存器(HDIV_STAT)

地址偏移量：0x14

复位值：0x00000001

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

位	标记	功能描述
31:2	Reserved	
1	div_zero	除数为零警告标志位。 0 --- 除数不为零 1 --- 除数为零
0	div_end	除法运算结束标志位。 0 --- 运算进行中 1 --- 运算结束

11 FLASH 控制器（FLASH）

11.1概述

本系统包含一块 16k/32k/64k/128k 字节（Byte）容量的 FLASH 存储器，共划分为 32/64/128/256 个页（Sector），每个页（Sector）的容量为 512 字节（Byte）。FLASH 控制器支持对 FLASH 存储器的擦除、编程以及读操作。本控制器还支持对 FLASH 存储器擦写保护，以及控制寄存器的写保护。

11.2FLASH 容量划分

表 11-1 FLASH 容量划分

地址	序号		地址	序号
0x0E00 – 0x0FFF	Sector7	.....	0x1FE00 – 0x1FFFF	Sector255
0x0C00 – 0x0DFF	Sector6	.....	0x1FC00 – 0x1FDFF	Sector254
0x0A00 – 0x0BFF	Sector5	.....	0x1FA00 – 0x1FBFF	Sector253
0x0800 – 0x09FF	Sector4	.....	0x1F800 – 0x1F9FF	Sector252
0x0600 – 0x07FF	Sector3	.....	0x1F600 – 0x1F7FF	Sector251
0x0400 – 0x05FF	Sector2	.....	0x1F400 – 0x1F5FF	Sector250
0x0200 – 0x03FF	Sector1	.....	0x1F200 – 0x1F3FF	Sector249
0x0000 – 0x01FF	Sector0	.....	0x1F000 – 0x1F1FF	Sector248

注意：

- 16KByte FLASH 由 Sector0-32 组成；
- 32KByte FLASH 由 Sector0-63 组成；
- 64KByte FLASH 由 Sector0-127 组成；
- 128KByte FLASH 由 Sector0-255 组成。

11.3功能描述

本控制器支持对 FLASH 的 Byte（8 位）、half-word（16 位）和 word（32 位）三种位宽的数据读写操作。注意，Byte 操作的目标地址必须按字节对齐，half-word 操作的目标地址必须按半字对齐（地址最低位为 1’b0），word 操作的地址必须按字对齐（地址最低两位为 2’b00）。如果目标地址没有按照位宽对齐，则该操作无效，并且 CPU 会进入 Hard Fault 出错中断。

本控制器采用高安全性的硬件设计，具有 FLASH 操作源防御功能：仅当 FLASH 操作函数的地址位于 0~32K，才能正确执行 FLASH 擦写操作。FLASH 地址的 0~32K 具有更高的安全性，重要函数须放置此区域内；例如重要程序入口、中断入口函数、高安全算法模块、UID、AES、真随机数的算法配合，组成高安全认证系统。

11.3.1 页擦除（Sector Erase）

页擦除每次可以擦除用户指定的一个页（Sector）。擦除操作完成后，页（Sector）内的数据均为 0xFF。如果该擦除操作是从 FLASH 内执行，则 CPU 会停止取指，硬件自动等待该操作完成（FLASH_CR. BUSY 变为 0）；如果该擦除操作是从 RAM 内执行，则 CPU 不会停止取指，用户软件应等待该操作完成（FLASH_CR. BUSY 变为 0）。

Sector 擦除操作步骤如下：

Step1：配置 FLASH 擦写参数，详见 11.4 章节。

Step2：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3：配置 FLASH_CR. OP 为 2，设置 Flash 操作模式为 Sector 擦除。

Step4：检查 FLASH_CR. OP 是否为 2，如不为 2 则跳转到 Step2。

Step5：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6：设置 FLASH_SLOCK 相应的比特为 1，去除该 Sector 的擦写保护。

Step7：检查 FLASH_SLOCK 相应的比特是否为 1，如不为 1 则跳转到 Step5。

Step8：对待擦除的 Sector 内的任意地址进行写入任意数据，触发 Sector 擦除。

例：*（(unsigned char *) 0x00000200）= 0x00。

Step9：等待 FLASH_CR. BUSY 变为 0，Sector 擦除操作完成。

Step10：如需擦除其它 Sector，重复 Step5 – Step9。

注意：

- 对 FLASH 进行页擦除的代码所在的地址必须小于 32768。

11.3.2 全片擦除（Chip Erase）

全片擦除可以一次性擦除全部的页（Sector）。擦除操作完成后，所有页（Sector）内的数据均为 0xFF。如果该擦除操作是从 FLASH 内执行，则该操作会被禁止。因为该操作会擦除当前 PC 所在的程序段。如发生这种情况，出错标志位会被置起；如果该擦除操作是从 RAM 内执行，则 CPU 不会停止取指，用户软件应等待该操作完成（FLASH_CR. BUSY 变为 0）。

Chip 擦除操作步骤如下：

Step1：配置 FLASH 擦写参数，详见 11.4 章节。

Step2：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3：配置 FLASH_CR. OP 为 3，设置 Flash 操作模式为 Chip 擦除。

Step4：检查 FLASH_CR. OP 是否为 3，如不为 3 则跳转到 Step2。

Step5：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6：设置 FLASH_SLOCK0 为 0xFFFFFFFF，去除所有 Sector 的擦写保护。

Step7：检查 FLASH_SLOCK0 是否为 0xFFFFFFFF，如不为 0xFFFFFFFF 则跳转到 则跳转到 Step5。

Step8：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step9：设置 FLASH_SLOCK1 为 0xFFFFFFFF，去除所有 Sector 的擦写保护。

Step10：检查 FLASH_SLOCK1 是否为 0xFFFFFFFF，如不为 0xFFFFFFFF 则跳转到 则跳转到 Step8。

Step11：对待擦除的 Chip 内的任意地址进行写操作，触发 Chip 擦除。

例：*（(unsigned char *）0x00000000）= 0x00。

Step12：等待 FLASH_CR. BUSY 变为 0，Chip 擦除操作完成。

11.3.3 写操作（Program）

写操作只能将 FLASH 内的比特数据由 1 写成 0，故写入数据前应确保待写入的地址内的数据为 0xFF。支持写入 3 种数据长度：Byte（8bits）、Half-word（16bits）、Word（32bits）,写入的数据以小端模式存放在 FLASH 中，即低地址存放数据的低字节。如果写操作是从 FLASH 内执行，则 CPU 会停止取指，硬件自动等待该操作完成（FLASH_CR. BUSY 变为 0）；如果该写操作是从 RAM 内执行，则 CPU 不会停止取指，用户软件应等待该操作完成（FLASH_CR. BUSY 变为 0）。

Byte 写操作步骤如下：

Step1：配置 FLASH 擦写参数，详见 11.4 章节。

Step2：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3：配置 FLASH_CR. OP 为 1，设置 Flash 操作模式为写入。

Step4：检查 FLASH_CR. OP 是否为 1，如不为 1 则跳转到 Step2。

Step5：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6：设置 FLASH_SLOCK 相应的比特为 1，去除擦写保护。

Step7：检查 FLASH_SLOCK 相应的比特是否为 1，如不为 1 则跳转到 Step5。

Step8：对待写入的目标地址进行 Byte 写操作，触发写入操作。

例：*（(unsigned char *）0x00001231）= 0x5A。

Step9：等待 FLASH_CR. BUSY 变为 0，写入操作完成。

Step10：如需写 Byte 到其它地址，重复 Step8 – Step9。

Half-word 写操作步骤如下：

Step1：配置 FLASH 擦写时间，详见 11.4 章节。

Step2：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3：配置 FLASH_CR. OP 为 1，设置 Flash 操作模式为写入。

Step4：检查 FLASH_CR. OP 是否为 1，如不为 1 则跳转到 Step2。

Step5：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6：设置 FLASH_SLOCK 相应的比特为 1，去除擦写保护。

Step7：检查 FLASH_SLOCK 相应的比特是否为 1，如不为 1 则跳转到 Step5。

Step8：对待写入的目标地址进行 Half-word 写操作，触发写入操作。

例：*（(unsigned short int *）0x00001232）= 0xABCD。

Step9：等待 FLASH_CR. BUSY 变为 0，写入操作完成。

Step10：如需写 Half-word 到其它地址，重复 Step8 - Step9。

Word 写操作步骤如下：

Step1：配置 FLASH 擦写参数，详见 11.4 章节。

Step2：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3：配置 FLASH_CR. OP 为 1，设置 Flash 操作模式为写入。

Step4：检查 FLASH_CR. OP 是否为 1，如不为 1 则跳转到 Step2。

Step5：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6：设置 FLASH_SLOCK 相应的比特为 1，去除擦写保护。

Step7：检查 FLASH_SLOCK 相应的比特是否为 1，如不为 1 则跳转到 Step5。

Step8：对待写入的目标地址进行 Word 写操作，触发写入操作。

例：*（(unsigned int *）0x00001234）= 0x55667788。

Step9：等待 FLASH_CR. BUSY 变为 0，写入操作完成。

Step10：如需写 Word 到其它地址，重复 Step8 - Step9。

注意：

- 对 FLASH 进行写操作的代码所在的地址必须小于 32768。

11.3.4 读操作（Read）

支持读出 3 种数据长度：Byte（8bits）、Half-word（16bits）、Word（32bits）,读出的数据为小端模式，即低地址存放数据的低字节。读操作无需操作步骤，任何时刻都可以读出 FLASH 内的数据。

例：

Byte 读操作：　　*temp = *（(*unsigned char* *) *0x00001231*）

Half-word 读操作：　　*temp = *（(*unsigned short int* *) *0x00001232*）

Word 读操作：　　*temp = *（(*unsigned int* *) *0x00001234*）

11.4擦写时间

FLASH 存储器对擦除和编程操作的控制信号具有严格的时间要求，控制信号的时序不合格会造成擦除操作和编程操作失败。上电时默认装载了 HCLK 为 4MHz 时的擦写参数；如果对 Flash 进行擦写时的 HCLK 频率不为 4MHz，则用户程序应装载 HCLK 频率所对应的擦写参数。对 FLASH 进行操作时，支持的最高 HCLK 为 48MHz。

擦写时序参数相关的寄存器为：FLASH_TNVS、FLASH_TPGS、FLASH_TPROG、FLASH_TSERASE、FLASH_TMERASE、FLASH_TPRCV、FLASH_TSRCV、FLASH_TMRCV。如果 HCLK 由默认 4MHz 升高为 8MHz，则上述 FLASH_Tx 寄存器的值应该设为默认值的 2 倍，即保持当前 Tsysclk*FLASH_Tx 的结果与默认值相等即可。

以下表格为不同频率下相应的 FLASH 擦写时间参数：

表 11-2 不同频率下 FLASH 擦写时间参数

Parameter	4M	8M	16M	24M	32M	48M
TNVS	0x20	0x40	0x80	0xC0	0x100	0x180
TPGS	0x17	0x2E	0x5C	0x8A	0xB8	0xFF
TPROG	0x1B	0x36	0x6C	0xA2	0xD8	0x144
TSERASE	0x4650	0x8CA0	0x11940	0x1A5E0	0x23280	0x34BC0
TMERASE	0x222E0	0x445C0	0x88B80	0xCD140	0x111700	0x19A280
TPRCV	0x18	0x30	0x60	0x90	0xC0	0x120
TSRCV	0xF0	0x1E0	0x3C0	0x5A0	0x780	0xB40
TMRCV	0x3E8	0x7D0	0xFA0	0x1770	0x1F40	0x2EE0

配置系统频率为 8MHz 时擦写参数的操作步骤如下：

Step1：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step2：向 FLASH_TNVS 寄存器写入 0x40，如读出的该寄器值不为 0x40，则跳转到上一步。

Step3：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step4：向 FLASH_TPGS 寄存器写入 0x2E，如读出的该寄器值不为 0x2E，则跳转到上一步。

Step5：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6：向 FLASH_TPROG 寄存器写入 0x36，如读出的该寄器值不为 0x36，则跳转到上一步。

Step7：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step8：向 FLASH_TSERASE 寄存器写入 0x8CA0，如读出的该寄器值不为 0x8CA0，则跳转到上一步。

Step9：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step10：向 FLASH_TMERASE 寄存器写入 0x445C0，如读出的该寄器值不为 0x445C0，则跳转到上一步。

Step11：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step12：向 FLASH_TPRCV 寄存器写入 0x30，如读出的该寄器值不为 0x30，则跳转到上一步。

Step13：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step14：向 FLASH_TSRCV 寄存器写入 0x1E0，如读出的该寄器值不为 0x1E0，则跳转到上一步。

Step15：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step16：向 FLASH_TMRCV 寄存器写入 0x7D0，如读出的该寄器值不为 0x7D0，则跳转到上一步。

11.5读等待周期

本设备内置的 FLASH 支持的最快取指频率为 24MHz。当 HCLK 频率超过 24MHz 小于 48MHz 时，必须为 FLASH 的读取时间插入等待周期，即设置 FLASH_CR.WAIT 为 1。当插入等待周期时，FLASH 每两个周期才会完成一次读取操作。

11.6擦写保护

11.6.1 擦写保护位

整个 16k/32k/64k128k 字节 FLASH 存储器被划分为 32/64/128/256 个页，每 4 个页共用一个擦写保护位。当页被保护时，对该页进行的擦写操作均无效并产生报警标志位和中断信号。当 FLASH 存储器中的任意页被保护时，对该 FLASH 的全片擦写无效，并产生报警标志位和中断信号。

11.6.2 PC 地址擦写保护

CPU 在 FLASH 中运行程序时，如果当前 PC 指针正好落在待擦写的页地址范围之内，那么该擦写操作无效并产生报警标志位和中断信号。

11.7寄存器写保护

本模块的重要控制器屏蔽普通的写操作，必须用写序列方式才能修改。

需要通过写序列方式才能更改的寄存器如下所示：

FLASH_TNVS、FLASH_TPGS、FLASH_TPROG、FLASH_TSERASE、FLASH_TMERASE、FLASH_TPRCV、FLASH_TSRCV、FLASH_TMRCV、FLASH_CR、FLASH_SLOCK0、FLASH_SLOCK1。

不需要通过写序列方式即可更改的寄存器如下所示：

FLASH_ICLR、FLASH_BYPASS。

通过写序列方式修改寄存器值的具体操作步骤如下所示：

Step1：向 FLASH_BYPASS 寄存器写入 0x5A5A。

Step2：向 FLASH_BYPASS 寄存器写入 0xA5A5。

Step3：对待修改的寄存器写入目标值。

Step4：验证待修改的寄存器当前值是否与目标相同，如不则跳转到 Step1。

Step5：执行其他操作。

注意：

- 写 0x5a5a、0xa5a5、写目标寄存器，这三步写操作之间不可插入任何写操作（写 ROM、RAM、REG），否则无法改写目标寄存器的数值。如改写失败，需要重新进行这三步操作。

11.8寄存器

基地址：0x4002 0000

寄存器	偏移地址	描述
FLASH_TNVS	0x00	Tnvs 时间参数
FLASH_TPGS	0x04	Tpgs 时间参数
FLASH_TPROG	0x08	Tprog 时间参数
FLASH_TSERASE	0x0C	Tserase 时间参数
FLASH_TMERASE	0x10	Tmerase 时间参数
FLASH_TPRCV	0x14	Tprcv 时间参数
FLASH_TSRCV	0x18	Tsrcv 时间参数
FLASH_TMRCV	0x1C	Tmrcv时间参数
FLASH_CR	0x20	控制寄存器
FLASH_IFR	0x24	中断标志寄存器
FLASH_ICLR	0x28	中断标志清除寄存器
FLASH_BYPASS	0x2C	0x5a5a-0xa5a5 Bypass序列寄存器
FLASH_SLOCK0	0x30	Sector0-127擦写保护寄存器
FLASH_SLOCK1	0x34	Sector128-255擦写保护寄存器

11.8.1 TNVS 参数寄存器（FLASH_TNVS）

偏移地址：0x00

复位值：0x0000 0020

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

位	标记	功能描述
31:9	Reserved	
8:0	TNVS	计算公式：TNVS = 8*HCLK，HCLK的单位为MHz。 4MHz示例：TNVS = 8*4 = 32。

11.8.2 TPGS 参数寄存器（FLASH_TPGS）

偏移地址：0x04

复位值：0x0000 0017

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

位	标记	功能描述
31:8	Reserved	
7:0	TPGS	计算公式：TPGS = 5.75*HCLK，HCLK的单位为MHz。 4MHz示例：TPGS = 5.75*4 = 23。 注：当计算所得数值大于 0xFF 时，则 TPGS 应赋值 0xFF。

11.8.3 TPROG 参数寄存器（FLASH_TPROG）

偏移地址：0x08

复位值：0x0000 001B

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

位	标记	功能描述
31:9	Reserved	
8:0	TPROG	计算公式：TPROG = 6.75*HCLK，HCLK的单位为MHz。 4MHz示例：TPROG = 6.75*4 = 27。

11.8.4 TSERASE 寄存器（FLASH_TSERASE）

偏移地址：0x0C

复位值：0x0000 4650

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

位	标记	功能描述
31:18	Reserved	
17:0	TSERASE	计算公式：TSERASE = 4500*HCLK，HCLK的单位为MHz。 4MHz示例：TSERASE = 4500*4 = 18000。

11.8.5 TMERASE 参数寄存器（FLASH_TMERASE）

偏移地址：0x10

复位值：0x000222E0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											TMERASE				
											RW				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMERASE															
RW															

位	标记	功能描述
31:21	Reserved	
20:0	TMERASE	计算公式：TMERASE = 35000*HCLK，HCLK的单位为MHz。 4MHz示例：TMERASE = 35000*4 = 140000。

11.8.6 TPRCV 参数寄存器（FLASH_TPRCV）

偏移地址：0x14

复位值：0x0000 0018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TPRCV											
				RW											

位	标记	功能描述
31:12	Reserved	
11:0	TPRCV	计算公式：TPRCV = 6*HCLK，HCLK的单位为MHz。 4MHz示例：TPRCV = 6*4 = 24。

11.8.7 TSRCV 参数寄存器（FLASH_TSRCV）

偏移地址：0x18

复位值：0x0000 00F0

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

位	标记	功能描述
31:12	Reserved	
11:0	TSRCV	计算公式：TSRCV = 60*HCLK，HCLK的单位为MHz。 4MHz示例：TSRCV = 60*4 = 240。

11.8.8 TMRCV 参数寄存器（FLASH_TMRCV）

偏移地址：0x1C

复位值：0x0000 03E8

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

位	标记	功能描述
31:14	Reserved	
13:0	TMRCV	计算公式：TMRCV = 250*HCLK，HCLK的单位为MHz。 4MHz示例：TMRCV = 250*4 = 1000。

11.8.9 CR 寄存器（FLASH_CR）

偏移地址：0x20

复位值：0x0000 0200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						DPSTB_EN	Reserved		IE		BUSY	WAIT		OP	
						RW			RW		RO	RW		RW	

位	标记	功能描述
31:10	Reserved	
9	DPSTB_EN	FLASH dpstb使能Mask位； 0：当系统进入deepsleep模式，FLASH不进入低功耗模式； 1：当系统进入deepsleep模式，FLASH进入低功耗模式；
8:7	Reserved	
6:5	IE	IE[6]：FLASH 擦写被保护地址中断使能；0：不使能；1：使能 IE[5]：FLASH 擦写 PC 值中断使能；0：不使能；1：使能
4	BUSY	空闲/忙标志位；0：空闲状态；1：忙状态；
3:2	WAIT	读FLASH 周期； 0~24MHz: 00/11，1个周期； 24~48MHz：01：2个周期； 48~72MHz：10:3个周期；（本系列产品最高时钟为48MHz）
1:0	OP	FLASH 操作； 00：读（read）；01：写（program）；10：页擦除（sector erase）；11：全片擦除（chip erase）

11.8.10 IFR 寄存器（FLASH_IFR）

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														IF1	IF0
														RO	RO

位	标记	描述
31:2	Reserved	
1	IF1	擦写保护报警中断标志位
0	IF0	擦写 PC 地址报警中断标志位

11.8.11 ICLR 寄存器（FLASH_ICLR）

偏移地址：0x28

复位值：0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														ICLR1	ICLR0
														R1W0	R1W0

位	标记	功能描述
31:4	Reserved	
3:2	Reserved	写无效，读为0x3
1	ICLR1	清除保护报警中断标志位；写0清除；写1无效；
0	ICLR0	清除 PC 地址报警中断标志位；写0清除；写1无效；

11.8.12 BYPASS 寄存器（FLASH_BYPASS）

偏移地址：0x2C

复位值：0x0000 0000

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

位	标记	描述
31:16	Reserved	
15:0	BYSEQ	在修改本模块寄存器之前，必须对 BYSEQ[15:0] 寄存器写入 0x5a5a-0xa5a5 序列。每次写入该 Bypass 序列后，只可以修改一次寄存器。如需再次修改寄存器，必须再次输入 0x5a5a-0xa5a5 序列。

11.8.13 SLOCK0 寄存器（FLASH_SLOCK0）

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOCK0[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLOCK0[15:0]															
RW															

位	标记	描述
31:0	SLOCK	Sector 擦写保护位；0：不允许擦写；1：允许擦写 Bit[0] 对应：Sector0-1-2-3 Bit [1] 对应：Sector4-5-6-7 Bit [2] 对应：Sector8-9-10-11 Bit [3] 对应：Sector12-13-14-15 …… Bit [31] 对应：Sector124-125-126-127

11.8.14 SLOCK1 寄存器（FLASH_SLOCK1）

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOCK1[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLOCK1[15:0]															
RW															

位	标记	描述
31:0	SLOCK	Sector 擦写保护位；0：不允许擦写；1：允许擦写 Bit [0] 对应：Sector128-129-130-131 Bit [1] 对应：Sector132-133-134-135 Bit [2] 对应：Sector136-137-138-139 Bit [3] 对应：Sector140-141-142-143 …… Bit [31] 对应：Sector252-253-254-255

12 RAM 控制器（RAM）

12.1概述

本系统中包含一块容量为 2k/4k/8k/16k 字节（Byte）的 SRAM，支持字节（8 位）、半字（16 位 bits）、字（32 位）三种读写操作。可在系统时钟频率下进行读写操作，无须等待周期。此外，本控制器还支持奇偶校验，可对每个字节（Byte）的 SRAM 数据进行奇偶校验，并产生奇偶校验出错中断。

12.2功能描述

12.2.1 RAM 地址范围

RAM 在系统映射中的地址范围如下表所示：

表 12-1 RAM 地址映射

地址范围	大小	Memory类型
0x2000_0000 – 0x2000_3FFF	16KByte	SRAM

12.2.2 读写位宽

本控制器支持 字节（8 位）、半字（16 位）、字（32 位） 三种位宽的读写操作。字节操作的地址必须按字节对齐，半字操作的目标地址必须按半字对齐（地址最低位为 1’b0），字操作的地址必须按字对齐（地址最低两位为 2’b00）。如果读写操作的目标地址没有按照位宽规定对齐，该操作无效，并且系统会产生 Hard Fault 出错中断。

12.2.3 奇偶校验

本控制器支持 SRAM 数据的奇偶校验。当对 SRAM 写数据时，对每个字节的数据做奇偶校验，并把 1bit 校验值和 8bits 数据一起存入 SRAM 中。当对 SRAM 读数据时，控制器会读取 8bits 数据和 1bit 校验值，并做奇偶校验，如果校验出错，则置位奇偶校验出错标志位，在中断使能情况下，会产生出错中断。

注意：

- 在奇偶校验使能时，在读取 SRAM 数据前，必须初始化 SRAM，否则可能会误触发奇偶校验报警标志位或者中断。

12.3寄存器

基地址：0x4002 0400

表 12-2 寄存器基地址

寄存器	偏移地址	描述
RAM_CR	0x00	控制寄存器
RAM_ERRADDR	0x04	出错地址寄存器
RAM_IFR	0x08	出错中断标志寄存器
RAM_ICLR	0x0C	出错中断标志清除寄存器

12.3.1 控制寄存器（RAM_CR）

偏移地址：0x00

复位值：0x0000 0000

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

位	标记	功能描述
31:2	Reserved	保留
1	IE	出错报警中断使能信号；1：使能报警中断，0：关闭报警中断；
0	Reserved	保留

12.3.2 奇偶校验出错地址寄存器（RAM_ERRADDR）

偏移地址：0x04

复位值：0x0000 0000

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

位	标记	功能描述
31:14	Reserved	
13:0	ERRADDR	奇偶校验出错Byte地址；中断标志位清除后，该地址同时被清除；

12.3.3 出错中断标志寄存器（RAM_IFR）

偏移地址：0x08

复位值：0x0000 0000

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

位	标记	功能描述
31:1	Reserved	
0	ERR	奇偶校验出错标志位

12.3.4 出错中断标志清除寄存器（RAM_ICLR）

偏移地址：0x0C

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															ERR CLR
															R1W 0

位	标记	功能描述
31:1	Reserved	
0	ERRCLR	出错中断标志清除位；写1：无效，写0：清零

13 DMA 控制器（DMAC）

13.1DMAC 简介

直接存储器存取(DMA)用来提供在外设和存储器之间或者存储器和存储器之间的高速数据传输；传输过程中无须 CPU 参与；故 CPU 可同步进行其它操作。DMAC 具有 2 条独立的 DMA 通道，每个通道专门用来管理来自于外设或存储器访问的请求。还有一个仲裁器来协调各个 DMA 请求的优先权。

13.2DMAC 主要特性

- 2 条独立 DMA 通道，支持优先级配置
- 3 种数据传输宽度：8-bit、16-bit、32-bit
- 4 种传输模式：软件 Block、软件 Burst、硬件 Block、硬件 Burst
- 2 种来源地址类型：外设、存储器
- 2 种目标地址类型：外设、存储器
- 2 种地址变化模式：固定、自增
- 传输地址寻址范围：0x00000000 ~ 0xFFFFFFFF
- 待传输数据块数量可配置：1~65536
- 待传输数据块大小可配置：1~16
- 支持地址及传输数量重装载功能

13.3功能框图

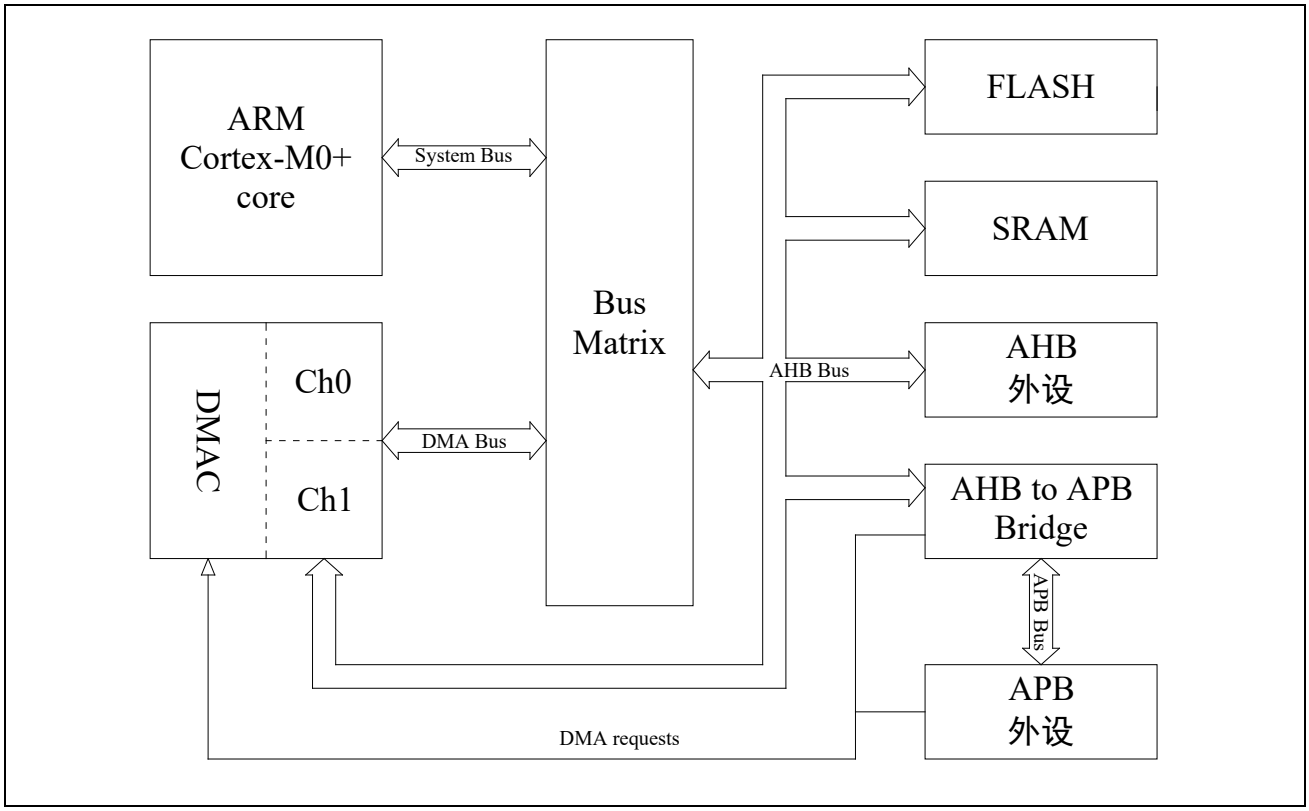


图 13-1 功能框图

- DMAC

  DMAC 具有 2 条 DMA 通道，当通道之间发生冲突时，由优先级控制器进行仲裁。
- Bus Matrix

  CPU 和 DMAC 均连接到总线矩阵上。当 CPU 释放 AHB 总线时，DMAC 可通过 Bus Matrix 实现对 AHB 总线设备及 APB 总线设备进行访问。CPU 访问总线的优先级高于 DMAC。
- DMA requests

  如果外设支持 DMA request，则该 DMA 通道可以配置为硬件触发否则只能配置为软件触发。

13.4功能描述

13.4.1 DMAC 传输模式概述

本 DMA 控制器支持 4 种传输模式，软件 Block 传输模式、软件 Burst 传输模式、硬件 Block 传输模式、硬件 Burst 传输模式。

4 种传输模式对比如下方表格所示：

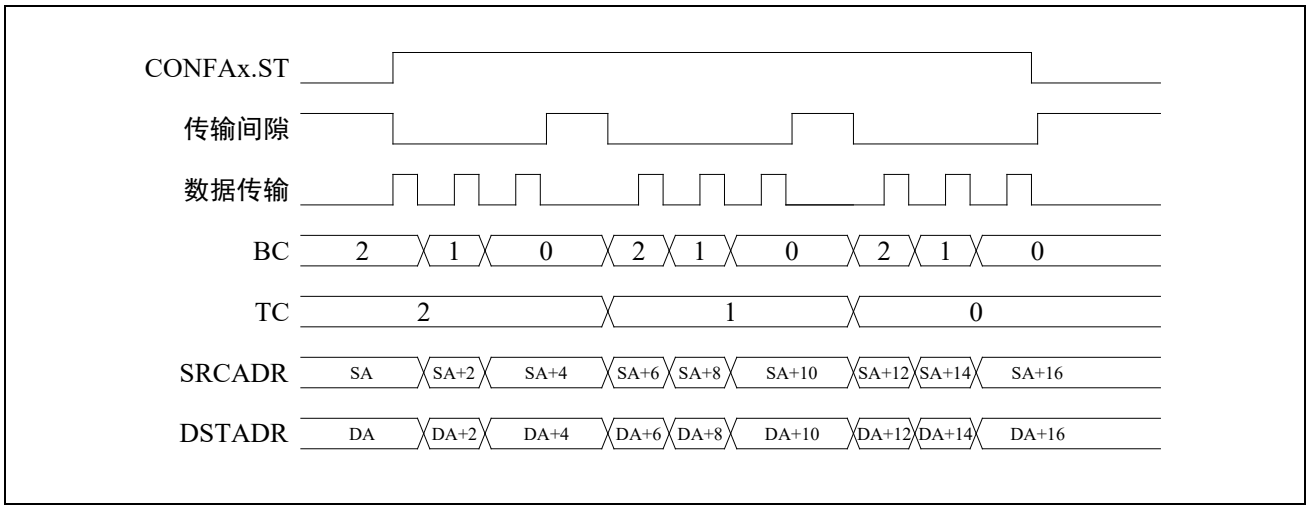
对比条目	软件 Block 传输	软件 Burst 传输	硬件 Block 传输	硬件 Burst 传输
是否可以被更高优先级 DMA 通道或 CPU 打断	可以	不可以	可以	不可以
触发传输启动的条件	写 DMAC 寄存器		外设中断标志	
触发一次传输的数据量	(BC+1) * (TC+1)		BC+1	(BC+1) * (TC+1)
配置后传输的总数据量	(BC+1) * (TC+1)			
主要应用场景	• 存储器到存储器 • 存储器到无 DMA 请求的外设		• 存储器到有 DMA 请求的外设 • 有 DMA 请求的外设到存储器	• 存储器到存储器 • 存储器到无 DMA 请求的外设

13.4.2 DMA 软件 Block 传输模式

当配置 DMAC_CONFAx.TRI_SEL=0，DMAC_CONFBx.MODE=0 时，DMAC 工作于软件 Block 传输模式。

当用户代码向 DMAC_CONFAx.ST 写入 1，则触发 DAMC 启动软件 Block 传输模式；DMA 传输完成 (BC+1) * (TC+1)个数据后停止。每传输完成 BC+1 个数据之后，DMAC 插入一个传输间隙，优先级仲裁器在该间隙进行优先级仲裁。如在该间隙发现 CPU 或更高优先 DMA 通道请求总线，则总线占有权转移给 CPU 或更高优先 DMA 通道；当 CPU 或更高优先 DMA 通道释放总线后，未完成的数据传输将继续完成。

软件 Block 传输示意如下所示，其中 SA 代表源地址，DA 代表目标地址；数据块大小为 3（BC=2），数据块数量为 3（TC=2），数据宽度为 16bit，地址自增。

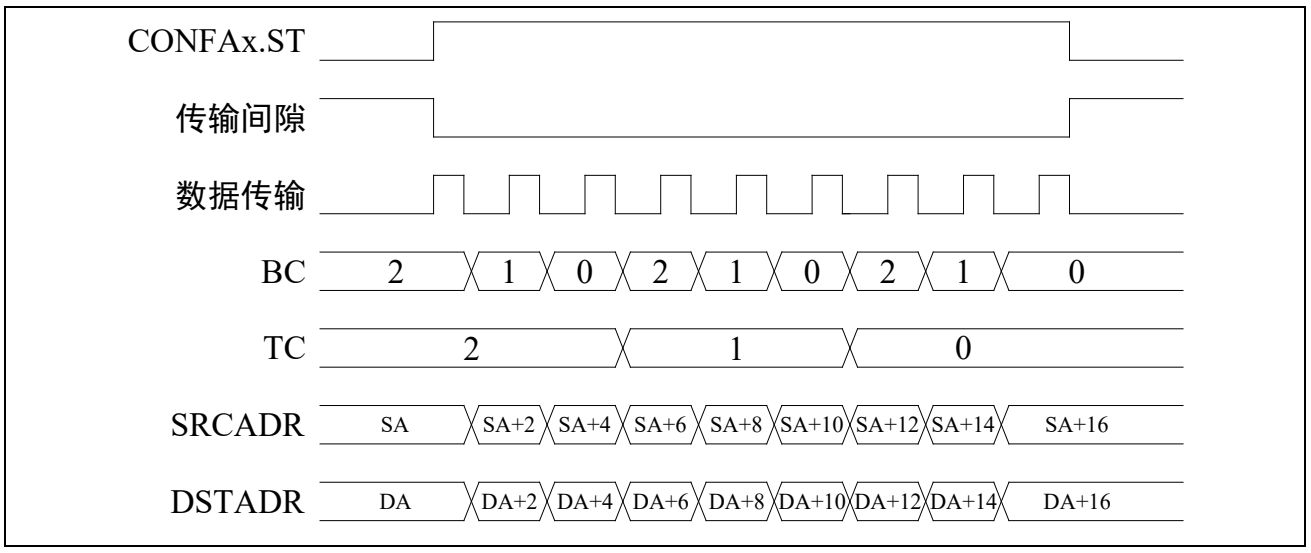


13.4.3 DMA 软件 Burst 传输模式

当配置 DMAC_CONFAx.TRI_SEL=0，DMAC_CONFBx.MODE=1 时，DMAC 工作于软件 Burst 传输模式。

当用户代码向 DMAC_CONFAx.ST 写入 1，则触发 DMAC 启动软件 Burst 传输模式；DMA 传输完成 (BC+1) * (TC+1)个数据后停止。该传输没有传输间隙，所有数据传输完成之前，DMA 将一直占据总线， CPU 或更高优先 DMA 通道只能等待该操作完成后才能访问总线。建议每次不要传输太多的数据，否则将可能造成 CPU 在一段时间内完全不能工作。

软件 Burst 传输示意如下所示，其中 SA 代表源地址，DA 代表目标地址；数据块大小为 3（BC=2），数据块数量为 3（TC=2），数据宽度为 16bit，地址自增。



13.4.4 DMA 硬件 Block 传输模式

当配置 DMAC_CONFAx.TRI_SEL=非零，DMAC_CONFBx.MODE=0 时，DMAC 工作于硬件 Block 传输模式。

每当外设 DMA request 信号出现一次，则触发 DMAC 启动一次硬件 Block 传输，传输(BC+1)个数据。需要出现(TC+1)次外设 DMA request 信号，才能完成所有 DMA 数据的传输。每完成 BC+1 个数据之后，DMAC 插入一个传输间隙，优先级仲裁器在该间隙进行优先级仲裁。如在该间隙发现 CPU 或更高优先 DMA 通道也在请求总线，则总线占有权转移给 CPU 或更高优先 DMA 通道；当 CPU 或更高优先 DMA 通道释放总线后，未完成的数据传输将继续完成。

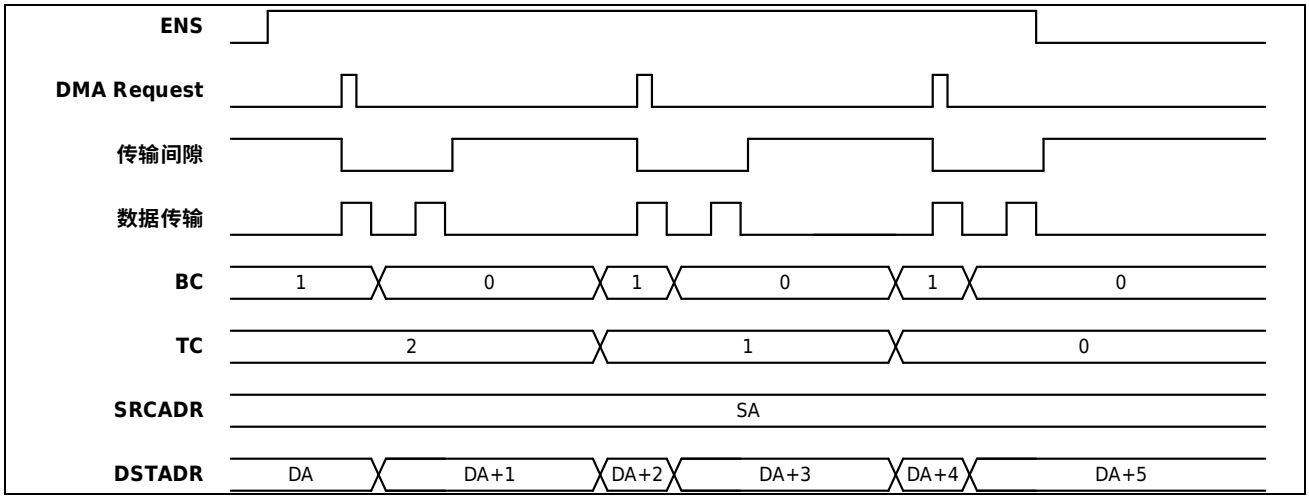
注意：

- 在该模式下，一般将 BC 设为 0，以实现从外设获取一个数据或提供一个数据给外设。

DMA 支持的硬件 request 信号如下所示：

- LPUARTx / UARTx 接收 Buf 非空、发送 Buf 空
- SPIx 接收 Buf 非空、发送 Buf 空
- I2Sx 接收 Buf 非空、发送 Buf 空
- TIMx 捕获完成、比较匹配
- ADC 插队转换完成
- ADC 顺序转换完成
- DAC 转换完成

硬件 Block 传输示意如下所示，其中 SA 代表源地址，DA 代表目标地址；数据块大小为 2（BC=1），数据块数量为 3（TC=2），数据宽度为 8bit，目标地址自增，源地址固定，传输完成自动禁止。



13.4.5 DMA 硬件 Burst 传输模式

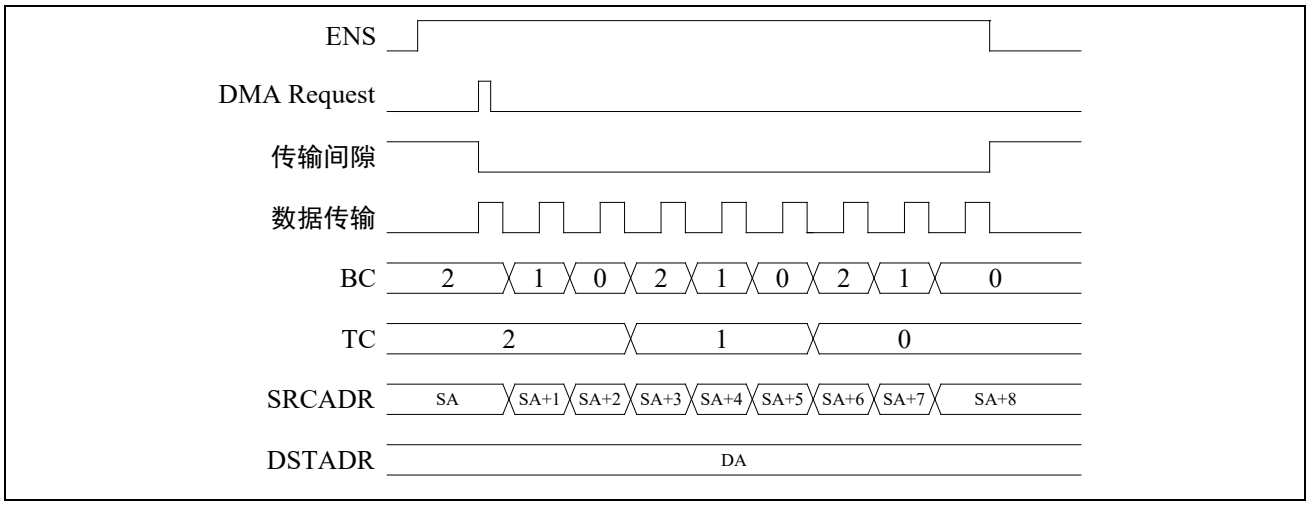
当配置 DMAC_CONFAx.TRI_SEL=非零，DMAC_CONFBx.MODE=1 时，DMAC 工作于硬件 Burst 传输模式。

当外设 DMA request 信号出现时，则触发 DMAC 启动硬件 Burst 传输模式，传输完成(BC+1) * (TC+1) 个数据后停止。该传输没有传输间隙，所有数据传输完成之前，DMA 将一直占据总线， CPU 或更高优先 DMA 通道只能等待该操作完成后才能访问总线。建议每次不要传输太多的数据，否则将可能造成 CPU 在一段时间内完全不能工作。

DMAC 支持的硬件 request 信号如下所示：

- LPUARTx / UARTx 接收 Buf 非空、发送 Buf 空
- SPIx 接收 Buf 非空、发送 Buf 空
- I2Sx 接收 Buf 非空、发送 Buf 空
- TIMx 捕获完成、比较匹配
- ADC 插队转换完成
- ADC 顺序转换完成
- DAC 转换完成

硬件 Burst 传输示意如下所示，其中 SA 代表源地址，DA 代表目标地址；数据块大小为 3（BC=2），数据块数量为 3（TC=2），数据宽度为 8bit，源地址自增，目标地址固定，传输完成自动禁止。



13.4.6 DMA 传输暂停

当 DMAC 工作于 Block 传输模式时，配置 DMAC_CONF.HALT 为非零值时，所有正在传输的通道都将在传输间隔时进入暂停状态，不再进行数据传输。配置 DMAC_CONF.HALT 为 0 后，DMAC 需要收到下一次触发信号才会继续刚才未完成的传输。配置 DMAC_CONFAx.PAS 为 1 时，x 相应的正在传输的通道将在传输间隔进入暂停状态，不再进行数据传输。配置 DMAC_CONFAx.PAS 为 0 后，DMAC 需要收到下一次触发信号才会继续刚才未完成的传输。

当 DMAC 工作于 Burst 传输模式且没有进行数据传输时，配置 DMAC_CONF.HALT 为非零值，则 DMA 通道进入暂停状态，不再进行数据传输。配置 DMAC_CONF.HALT 为 0 后，DMAC 收到下一次触发信号时会启动一次新的完整传输。当 DMAC 工作于 Burst 传输模式且没有进行数据传输时，配置 DMAC_CONFAx.PAS 为 1 时，则 DMA 通道进入暂停状态，不再进行数据传输。配置 DMAC_CONFAx.PAS 为 0 后，DMAC 收到下一次触发信号时才会启动一次新的完整传输。

当 DMAC 工作于 Burst 传输模式且正在进行数据传输时，向 DMAC_CONF.HALT 写入非 0 值或向 DMAC_CONFAx.PAS 写入 1 均不能暂停正在进行的传输；当前传输完成后 DMA 通道也不会进入暂停态。

13.4.7 DMA 其它配置

13.4.7.1 数据宽度

配置 DMAC_CONFBx.WIDTH，可以配置每个待传送的数据的宽度为 8bit、16bit、32bit，注意该位宽应与 DMA 的源地址与目标地址的位宽完全一致。

13.4.7.2 数据块大小

配置 DMAC_CONFAx.BC，可以配置每个数据块待传送的数据个数为 1~16。注意，BC 的值应当与源地址和目标地址所提供的连续操作次数相匹配。

如果源地址或目标地址一次只能提供或接收一个数据则 BC 只能配置为 0。例如 UART / LPUART / SPI 每次只能向 DMA 提供一个数据，UART / LPUART / SPI / DAC 每次只能从 DMA 接收一个数据，所以 BC 只能配置为 0。

如果源地址和目标地址均能提供或接收多个数据，则 BC 可以配置为更大的值，以加快传输速度。例如从 FLASH 传输到 RAM 或从 FLASH 传输到 CRC，源地址和目标地址都可以连续操作，故可以配置 BC 为 15 以实现快速传输。

13.4.7.3 数据块个数

配置 DMAC_CONFAx.TC，可以配置每次 DMA 启动后，传输出的数据块的个数为 1~65536。

13.4.7.4 通道优先级

当 DMAC_CONF.PRIO=0 时，DMAC 工作在固定优先模式下。此时 CH0 比 CH1 具有更高的优先级，只有在 CH0 不进行数据传输时，CH1 才能进行数据传输。

当 DMAC_CONF.PRIO=1 时，DMAC 工作在循环优先模式下。此时 CH0 与 CH1 轮流具有更高的优先级，两个通道将轮流占有总线以进行数据传输。

13.4.7.5 自动重装载

如果使能自动重装载功能则数据块大小、数据块个数、来源地址、目标地址在传输完成后均自动装载为上一次配置的数值，不用每次进行配置。

13.4.8 DMA 中断

DMA 支持两种类形的中断：传输出错和传输完成。当中断触发时，通过查询状态寄存器 DMAC_CONFBx.STAT 就可以判断是什么条件触发了中断，通过对寄存器 DMAC_CONFBx.STAT 清零就可以清除内置的中断状态标志位。

5 种触发 DMA 中断的条件如下：

- 传输地址超出寻址范围
- 外设请求停止 DMA
- 访问传输来源地址出错
- 访问传输目的地址出错
- 传输完成

13.5寄存器

DMAC 基地址：0x4002 1000

寄存器	偏移地址	控制通道	描述
DMAC_CONF	0x00	ALL	所有通道配置寄存器
DMAC_CONFA0	0x10	CH0	通道0配置A寄存器
DMAC_CONFB0	0x14	CH0	通道0配置B寄存器
DMAC_SRCADR0	0x18	CH0	通道0传输源地址寄存器
DMAC_DSTADR0	0x1C	CH0	通道0传输目的地址寄存器
DMAC_CONFA1	0x20	CH1	通道1配置A寄存器
DMAC_CONFB1	0x24	CH1	通道1配置B寄存器
DMAC_SRCADR1	0x28	CH1	通道1传输源地址寄存器
DMAC_DSTADR1	0x2C	CH1	通道1传输目的地址寄存器

13.5.1 DMAC_CONF

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN	Res		PRIO	HALT				Reserved							
RW	R		RW	RW											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

位	标记	功能描述
31	EN	DMA控制器使能 1：使能DMA控制器 0：禁止DMA控制器 注意：当禁止DMA控制器时，将强制停止正在进行的传输，造成不可预知的结果。
30:29	Reserved	
28	PRIO	DMA通道优先级配置 1：循环优先级模式，CH0和CH1轮流获取总线访问权 0：固定优先级模式，CH0优先级高于CH1优先级
27:24	HALT	DMA全部通道传输暂停控制 0000：恢复所有通道数据传输 xxxx：暂停所有通道数据传输
23:0	Reserved	

13.5.2 DMAC_CONFA0、DMAC_CONFA1

偏移地址：0x10（DMAC_CONFA0）

0x20（DMAC_CONFA1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ENS	PAS	ST	TRI_SEL							Reserved		BC			
RW	RW	RW	RW									RW			

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

位	标记	功能描述
31	ENS	DMA 通道使能控制 1：使能当前DMA通道；若CONFBx.MSK为0，则传输完成时该位自动清零 0：禁止当前DMA通道 注意：当禁止DMA通道时，将强制停止正在进行的传输，造成不可预知的结果。
30	PAS	DMA 通道数据传输暂停控制 1：暂停DMA通道数据传输 0：恢复DMA通道数据传输
29	ST	DMA 通道软件触发控制 1：触发 DMA 通道启动软件 Block / Burst 传输，传输完成时该位自动清零 0：停止 DMA 通道软件 Block / Burst 传输
28:22	TRI_SEL	DMA 通道触发源配置 0x00：软件触发 0x40：SPI0 接收 Buf 非空 0x41：SPI0 发送 Buf 空 0x42：SPI1 接收 Buf 非空 0x43：SPI1 发送 Buf 空 0x44：ADC 插队转换完成 0x45：ADC 顺序转换完成 0x46：Reserved 0x47：Reserved 0x48：UART0 接收 Buf 非空 0x49：UART0 发送 Buf 空 0x4A：UART1 接收 Buf 非空 0x4B：UART1 发送 Buf 空 0x4C：LPUART0 接收 Buf 非空 0x4D：LPUART0 发送 Buf 空 0x4E：LPUART1 接收 Buf 非空 0x4F：LPUART1 发送 Buf 空 0x50：DAC0 转换完成 0x51：DAC1 转换完成

		0x52：TIM0 通道 A，发生捕获事件或比较事件 0x53：TIM0 通道 B，发生捕获事件或比较事件 0x54：TIM1 通道 A，发生捕获事件或比较事件 0x55：TIM1 通道 B，发生捕获事件或比较事件 0x56：TIM2 通道 A，发生捕获事件或比较事件 0x57：TIM2 通道 B，发生捕获事件或比较事件 0x58：TIM3 通道 A，发生捕获事件或比较事件 0x59：TIM3 通道 B，发生捕获事件或比较事件 0x5A：TIM4 通道 A，发生捕获事件或比较事件 0x5B：TIM4 通道 B，发生捕获事件或比较事件 0x5C：TIM5 通道 A，发生捕获事件或比较事件 0x5D：TIM5 通道 B，发生捕获事件或比较事件 0x5E：TIM6 通道 A，发生捕获事件或比较事件 0x5F：TIM6 通道 B，发生捕获事件或比较事件 0x60：UART2 接收 Buf 非空 0x61：UART2 发送 Buf 空 0x62：UART3 接收 Buf 非空 0x63：UART3 发送 Buf 空 0x64：I2S0 左声道发送 Buf 空或接收 Buf 非空 0x65：I2S0 右声道发送 Buf 空或接收 Buf 非空 0x66：I2S1 左声道发送 Buf 空或接收 Buf 非空 0x67：I2S1 右声道发送 Buf 空或接收 Buf 非空
21:20	Reserved	Reserved
19:16	BC	配置 DMA 通道待传输的数据块的大小为 BC+1 注意：当源址址或目标地址每隔一段时间才能提供或接收一个数据时，BC 值只可设为 0
15:0	TC	配置 DMA 通道待传输的数据块的数量为 TC+1 传输过程中，每传输完成一个数据块，该寄存器值自减 1 注意：DMA 传输的总数据量为(TC + 1) * (BC + 1)

13.5.3 DMAC_CONFB0、DMAC_CONFB1

偏移地址：0x14 （DMAC_CONFB0）

0x24 （DMAC_CONFB1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		MODE		WIDTH		FS	FD	RC	RS	RD	ERR_IE	FIS_IE	STAT		
		RW		RW		RW	RW	RW	RW	RW	RW	RW	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															MSK
															RW

位	标记	功能描述
31:30	Reserved	
29:28	MODE	DMA通道传输模式配置 00：Block 传输 01：Burst 传输 10：Reserved 11：Reserved
27:26	WIDTH	DMA通道传输数据宽度配置 00：8bit 01：16bit 10：32bit 11：Reserved
25	FS	DMA通道源地址自增使能 1：源地址在传输过程中保持不变 0：每传输完成一个数据，源地址进行自增； 数据宽度为 8bit / 16bit / 32bit 所对应的自增量分别为 1 / 2 / 4
24	FD	DMA通道目标地址自增使能 1：目标地址在传输过程中保持不变 0：每传输完成一个数据，目标地址进行自增； 数据宽度为8bit / 16bit / 32bit所对应的自增量分别为1 / 2 / 4
23	RC	DMA通道BC和TC重载使能 1：使能 BC/TC 重载，传输完成后 BC/TC 的值自动恢复为写入的初值 0：禁止 BC/TC 重载，传输完成后 BC/TC 的值均为 0
22	RS	DMA通道源地址重载使能 1：使能源地址重载，传输完成后源地址的值为写入的初值 0：禁止源地址重载，传输完成后源地址的值不确定
21	RD	DMA通道目标地址重载使能 1：使能目标地址重载，传输完成后目标地址的值自动恢复为写入的初值 0：禁止目标地址重载，传输完成后目标地址的值不确定

20	ERR_IE	DMA通道传输错误中断使能 1：传输错误时，产生中断 0：传输错误时，禁止中断 注意：在中断服务程序中，需要对 STAT 写入 0x00 以清除中断状态
19	FIS_IE	DMA 通道传输完成中断使能 1：传输完成时，产生中断 0：传输完成时，禁止中断 注意：在中断服务程序中，需要对 STAT 写入 0x00 以清除中断状态
18:16	STAT	DMA通道当前传输状态 000：初始值 001：传输错误，传输地址超出寻址范围 010：传输错误，外设请求停止 DMA 011：传输错误，访问传输来源地址出错 100：传输错误，访问传输目的地址出错 101：传输完成 110：保留 111：传输暂停
15:1	Reserved	
0	MSK	DMA 通道传输完成时自动禁能配置 1：DMA 通道传输完成时，CONFAx.ENS 保持不变 0：DMA 通道传输完成时，CONFAx.ENS 自动清零

13.5.4 DMAC_SRCADR0、DMAC_SRCADR1

偏移地址：0x18（DMAC_SRCADR0）

0x28（DMAC_SRCADR1）

复位值：0x0000 0000

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

位	标记	功能描述
31:0	SRCADR	DMA 通道源地址配置 可通过 CONFBx.RS 配置其自动重载 注意：地址对齐方式应与数据宽度相匹配，否则会造成地址访问出错

13.5.5 DMAC_DSTADR0、DMAC_DSTADR1

偏移地址：0x1C（DMAC_DSTADR0）

0x2C（DMAC_DSTADR1）

复位值：0x0000 0000

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

位	符号	描述
31:0	DSTADR	DMA 通道目标地址配置 可通过 CONFBx.RD 配置其自动重载 注意：地址对齐方式应与数据宽度相匹配，否则会造成地址访问出错

14 通用定时器（TIM0/1/2/3）

14.1通用定时器简介

TIM0/1/2 是分别由 1 个计数单元和 2 个比较单元组成的定时器。每个定时器支持 2 个独立 PWM 输出或 1 对互补 PWM 输出和 1 个独立 PWM 输出。支持 2 个捕获输入。TIM0/1/2 可以组成 3 对互补 PWM 输出。

TIM3 是由 1 个计数单元和 6 个比较单元组成的定时器，支持 6 个独立 PWM 输出或 3 对互补 PWM 输出。支持 6 个捕获输入。

使用定时器预分频、系统预分频器和系统时钟选择，可灵活的调整脉冲宽度和波形周期；可以方便的测量脉冲宽度。

14.1.1 基本特性(TIM0/1/2)

- 2 路独立 PWM 输出 CHA,CHB,
- 1 路互补 PWM 输出（CHA,CHB）+1 路独立 PWM 输出（gate）
- 1 路互补 PWM 输出（CHA,CHB）+1 路捕获功能(gate)
- 最多 2 路捕获输入
- 脉冲宽度测量
- 死区控制
- 刹车控制
- 边沿对齐、对称中心对齐与非对称中心对齐 PWM 输出
- 正交编码计数功能
- 单脉冲模式
- 外部计数功能
- DMA 触发
- 系统时钟选择 PLL 时，定时器时钟可以配置为系统时钟的 2 倍频

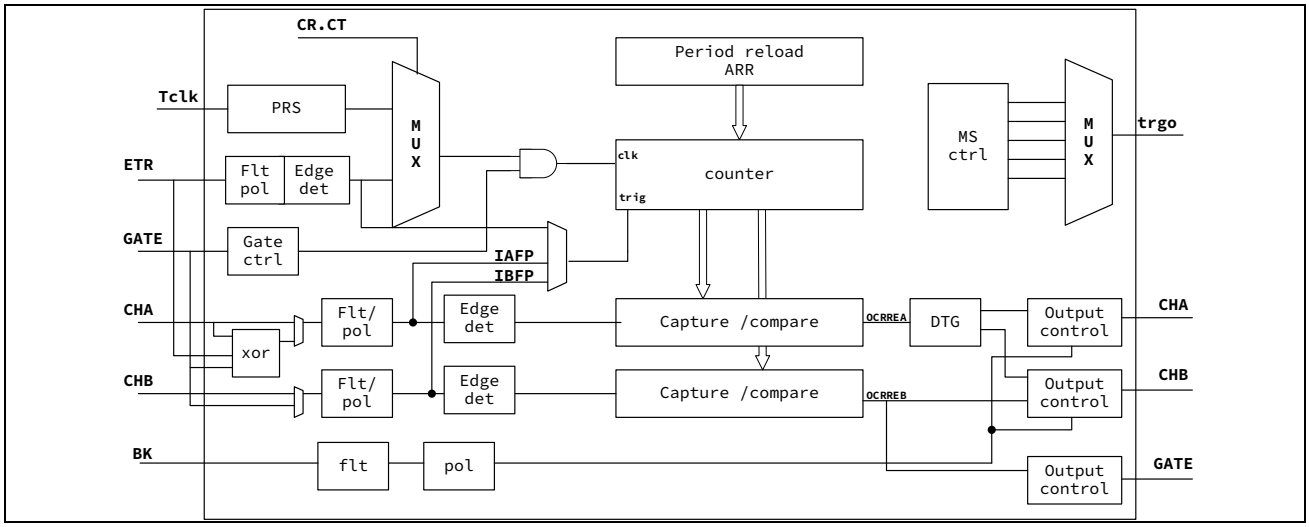


图 14-1 TIM0/1/2 框图

14.1.2 基本特性(TIM3)

- 6 路独立 PWM 输出 CH0A,CH0B, CH1A,CH1B, CH2A,CH2B
- 3 路互补 PWM 输出（CHxA,CHxB）+1 路独立 PWM 输出（gate）
- 3 路互补 PWM 输出（CHxA,CHxB）+1 路捕获功能(gate)
- 最多 6 路捕获输入
- 脉冲宽度测量
- 死区控制
- 刹车控制
- 边沿对齐、对称中心对齐与非对称中心对齐 PWM 输出
- 正交编码计数功能
- 单脉冲模式
- 外部计数功能
- DMA 触发
- 系统时钟选择 PLL 时，定时器时钟可以配置为系统时钟的 2 倍频

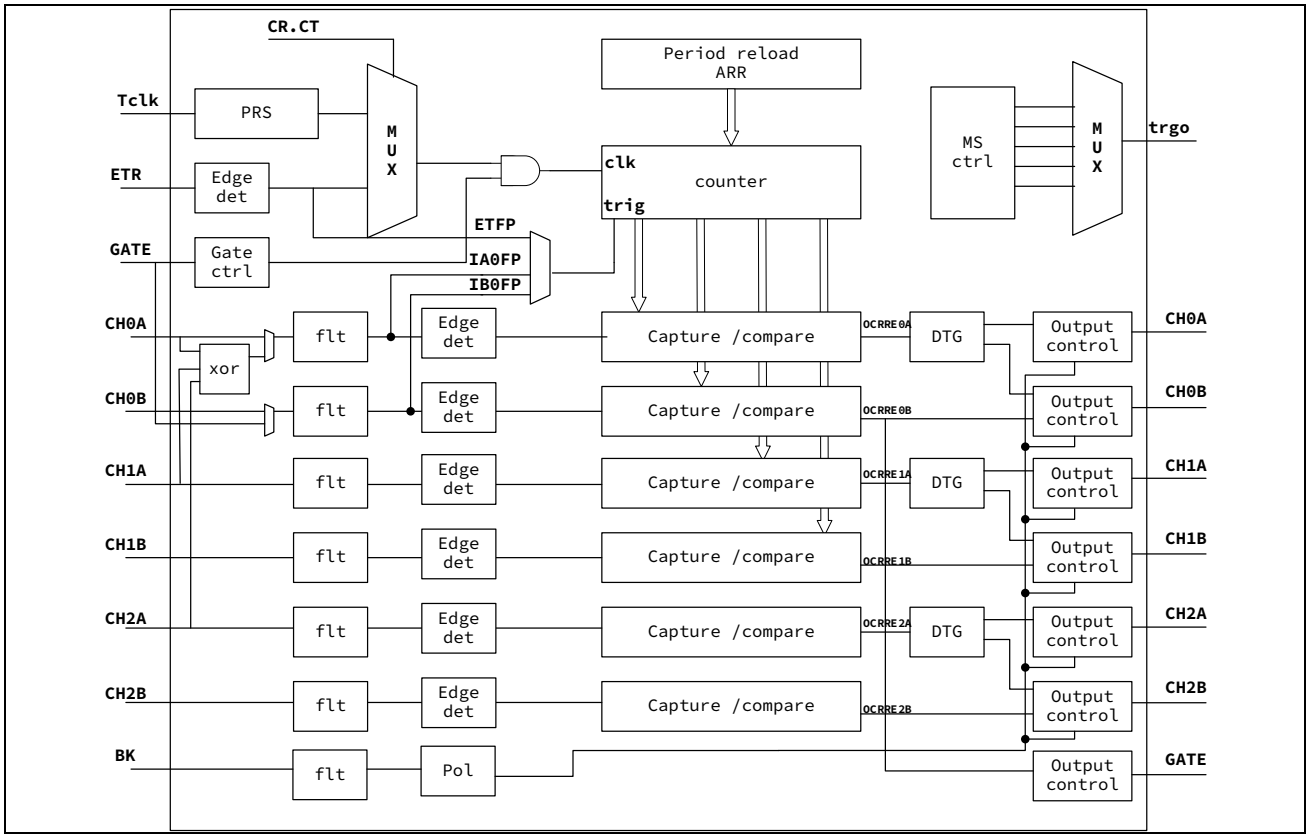


图 14-2 TIM3 框图

14.2Timer 功能描述

14.2.1 定时器时钟

定时器使用 PCLK 作为定时器时钟，下文定时器时钟使用 TCLK 标示。

14.2.2 定时计数器

可编程通用定时器的主要模块是一个 16 位计数器及其相关的自动重载寄存器。计数器可递增计数、递减计数或交替进行递增和递减计数。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

14.2.3 定时器预除频

使用 TCLK 做为 Timer 定时器时钟时，可以使用预分频。预分频设置如下：

PRS	000	001	010	011	100	101	110	111
分频比	1	2	4	8	16	32	64	256

预分频器没有预装载缓存，因此对预分频进行更改，预分频将会立刻生效。

14.2.4 模式 0 计数定时器功能

这种模式下，计数器向上计数。计数器支持两种计数模式，重载模式与自由计数模式，可以选择外部时钟 ETR 计数或系统时钟计数。门控 Gate 可以控制计数屏蔽。计数到最大值溢出产生中断。翻转输出端口 CHA,CHB，这种模式下 CHA,CHB 反相。

重载模式计数范围从 ARR 向上计数到 0xFFFF 溢出，然后从 ARR 再开始计数，计数周期为 0Xffff-ARR+1；自由计数模式从设置计数值计数到 0xFFFFFFFF 后溢出，溢出后计数值从 0x0 重新开始计数。

计数器可用通过外部门控功能控制计数器是否计数。控制关系如下：

M0CR.GATE	M0CR.GATEP	端口 GATE 输入	M0CR.CTEN	计数器
x	x	x	0	不计数
0	x	x	1	计数
1	0	高电平	1	计数
1	0	低电平	1	不计数
1	1	高电平	1	不计数
1	1	低电平	1	计数

14.2.4.1 功能框图

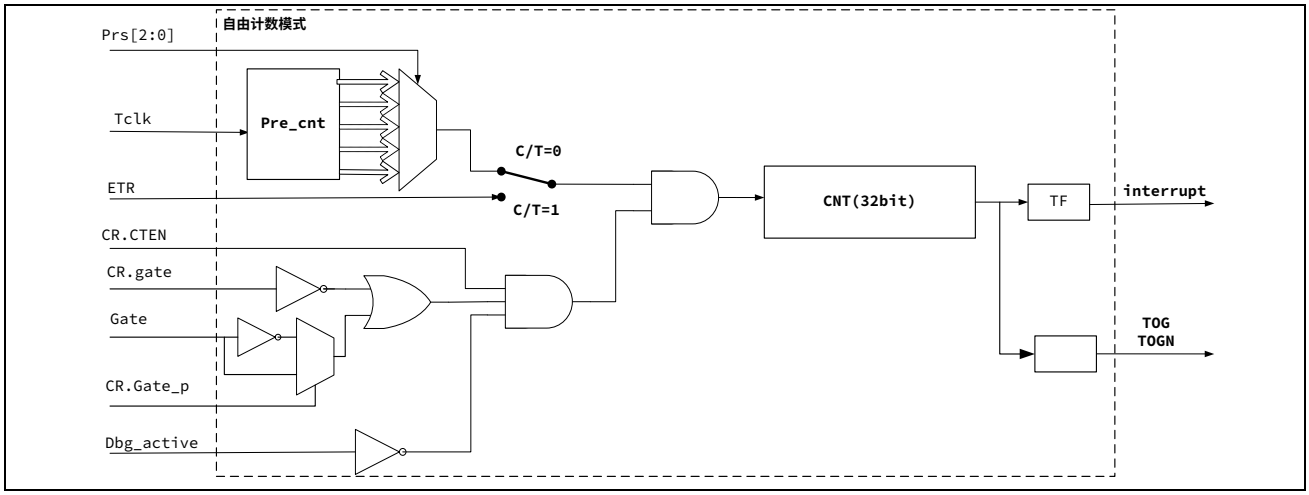


图 14-3 自由计数框图

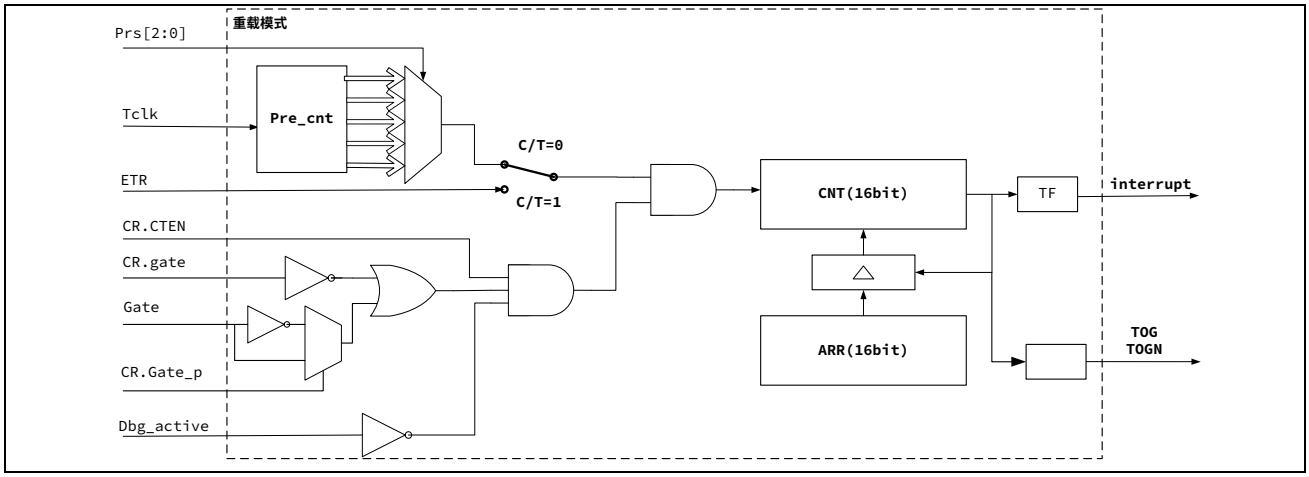


图 14-4　重载计数波形

14.2.4.2 计数波形

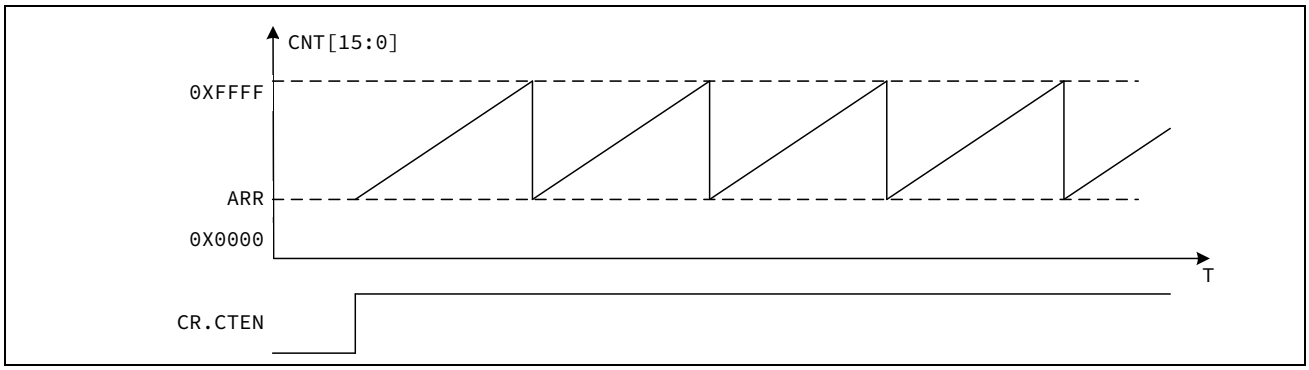


图 14-5　16 位重载计数波形

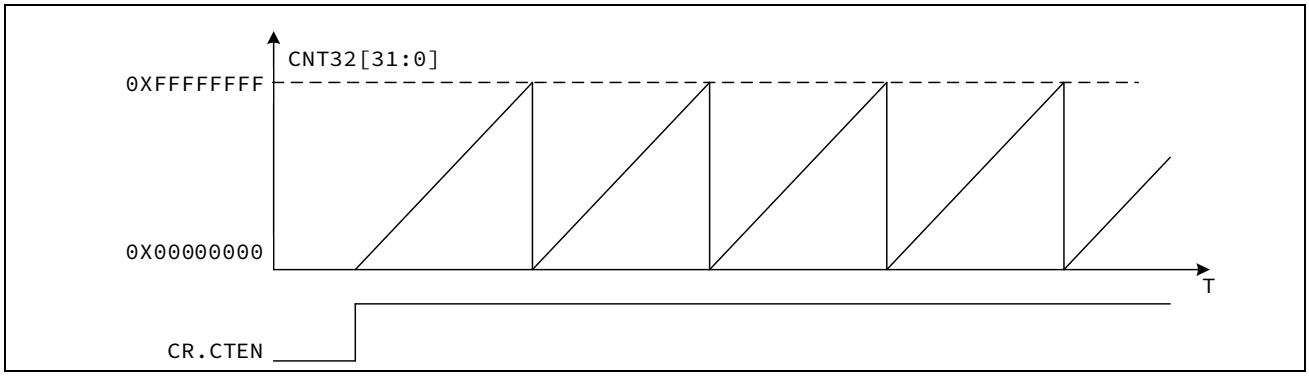


图 14-6　32 位自由计数波形

14.2.4.3 计数功能

计数功能用于测定某个事件发生的次数。在计数功能中，计数器在每个相应的输入时钟的下降沿累加一次。输入信号被内部的 tclk 采样，因此外部输入时钟频率不能超过系统的 tclk 时钟。计数到最大值会溢出并且产生中断。中断标志需要软件清除。

14.2.4.4 定时功能

定时功能用于产生间隔定时。在定时功能中，定时器有预除频，定时器在每个预除频的一个时钟累加一次，计数到最大值会溢出并且产生中断。中断标志需要软件清除。

14.2.4.5 时序图

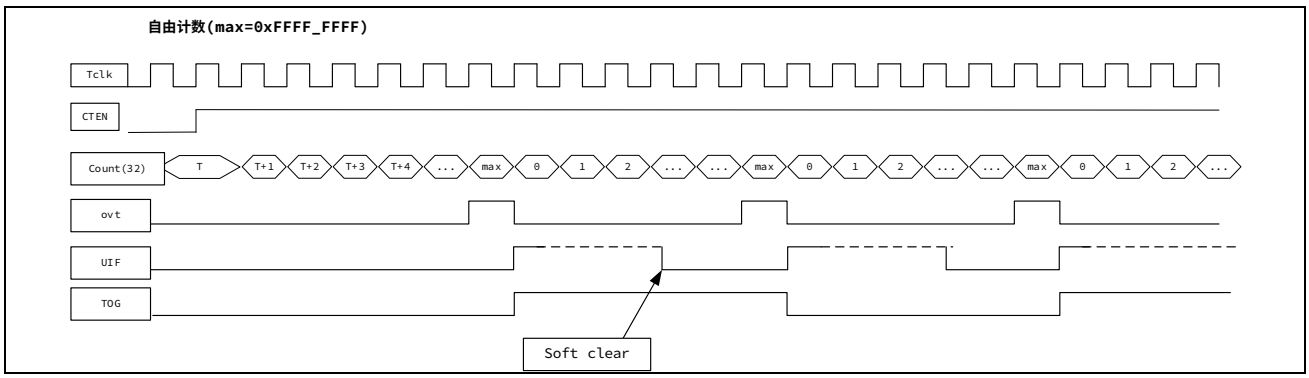


图 14-7 自由计数时序图

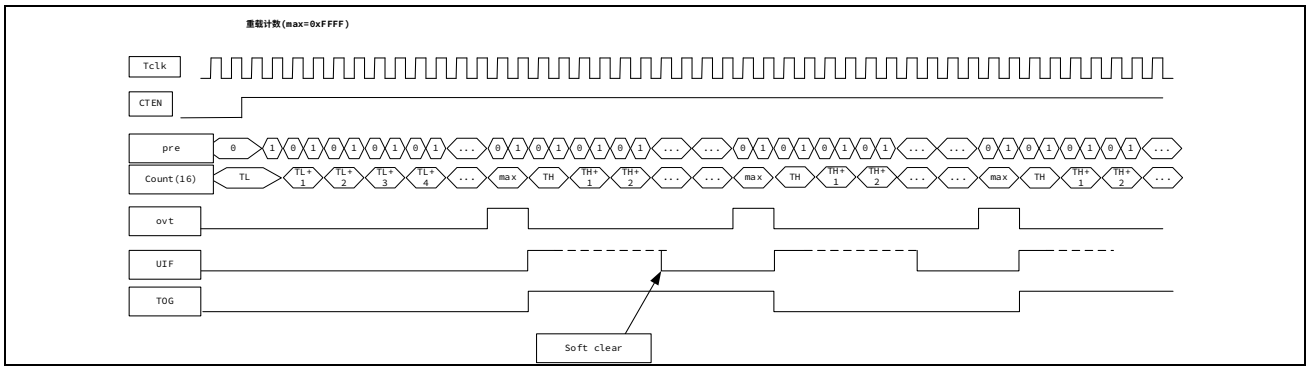


图 14-8 重载计数时序图（预分频设置为 2）

14.2.4.6 Buzzer 功能

通过定时器的翻转输出功能可以实现驱动 Buzzer 的功能。使用 toggle 输出需要使能 DTR.MOE 控制位。设置 CR.TOG_EN 为 0 可以同时设置端口 CHA,CHB 输出为 0。在计数时钟为 4M 情况下 Buzzer 输出不同频率的 timer 重载模式配置如下：

Buzzer 频率	计数器周期	计数器计数值	计数器重载值	CNTL 初始值	CNTH 重载值
1000Hz	0.5ms	2000	63536	0XF830	0XF830
2000Hz	0.25ms	1000	64536	0XFC18	0XFC18
4000Hz	0.125ms	500	65036	0XFE0C	0XFE0C

14.2.4.7 设置示例

重载定时器设置

1. 设置定时器模式 M0CR.MODE=0
2. 设置装载值 ARR
3. 设置计数器初值 CNT
4. 清除中断标志
5. 使能中断 M0CR.UIE
6. 使能重载模式 M0CR.MD
7. 开启定时器 M0CR.CTEN

门控外部时钟自由计数设置

1. 设置定时器模式 M0CR.MODE=0
2. 设置计数器初值 CNT
3. 使能门控功能 M0CR.GATE
4. 选择门控有效电平 M0CR.GATEP
5. 清除中断标志
6. 使能中断 M0CR.UIE
7. 使能外部时钟模式 M0CR.CT
8. 开启定时器 M0CR.CTEN

BUZZER 输出控制

1. 根据输出频率设置合适的 ARR 值
2. 设置定时器为重载模式，参考重载定时器设置
3. 使能输出使能 DTR.MOE
4. 启动另外一个定时器控制 M0CR.TOGEN 实现频率的间隔输出。

14.2.5 模式 1 脉宽测量 PWC

这种模式下可以自动测量输入脉冲的高电平低电平或者周期宽度。

第一个有效边沿计数器初始化为 0x0001，第二个有效边沿将停止计数，并将当前计数值存入 CMAR，并且产生捕获中断 CAF，如果计数器发生溢出，会产生溢出标志。设置溢出中断使能会产生溢出中断。

M1CR.edg1st	0	0	1	1
M1CR.edg2nd	0	1	0	1
脉宽测量	上沿~上沿 周期宽度	上沿~下沿 高电平宽度	下沿~上沿 低电平宽度	下沿~下沿 周期宽度

周期测量时，会间隔一个周期测量一个周期。

14.2.5.1 PWC 功能框图

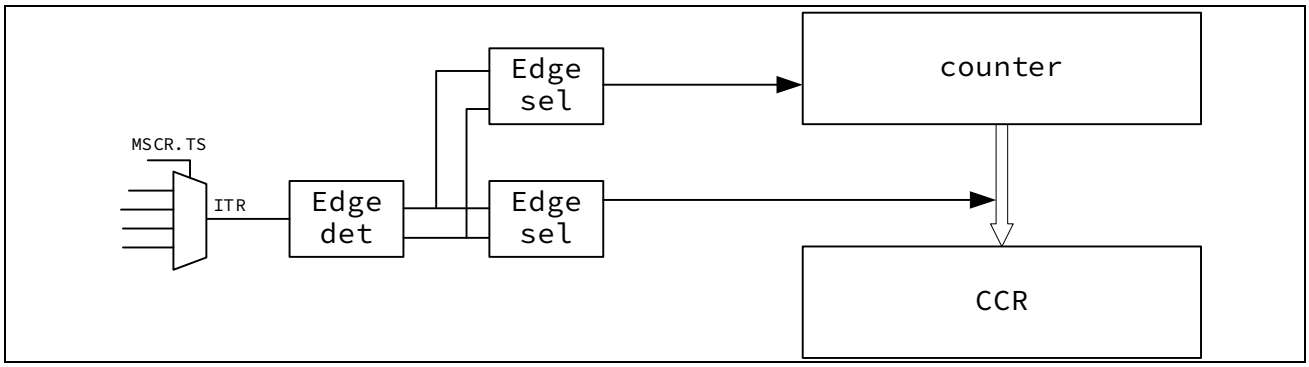


图 14-9 PWC 测量框图

MSCR.TS	触发选择 000：端口ETR的滤波相位选择后的信号ETFP ； 001：内部互联信号 ITR0 010：内部互联信号 ITR1; 011：内部互联信号 ITR2; 100：内部互联信号 ITR3; 101：无效 110：端口CH0A的滤波后的信号IAFP（极性选择在脉宽测量模式下无效） 111：端口CH0B的滤波后的信号IBFP（极性选择在脉宽测量模式下无效）

14.2.5.2 PWC 波形测量时序图

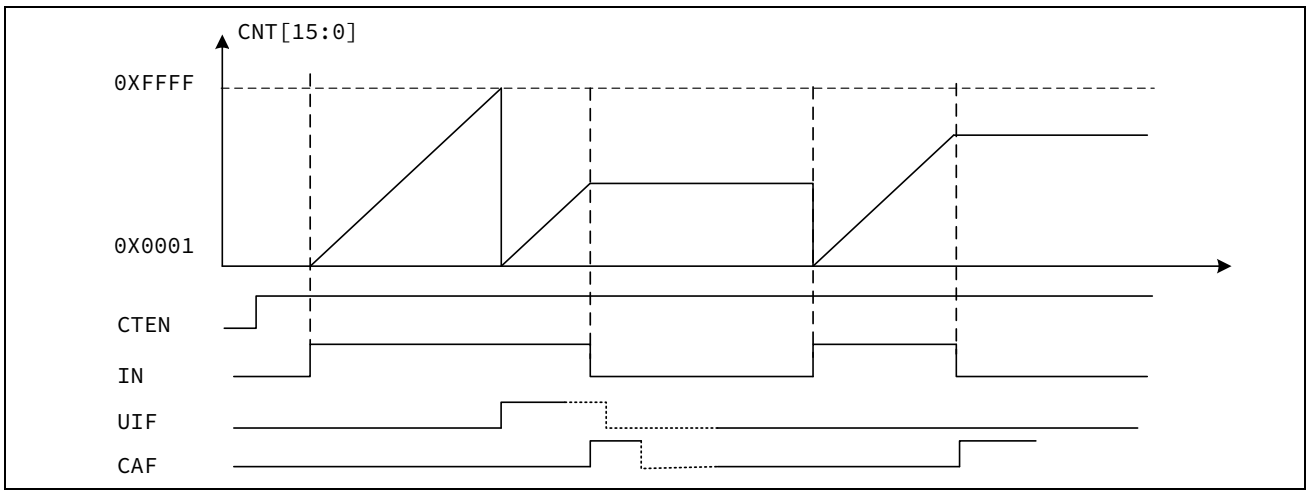


图 14-10 高电平脉冲宽度测量

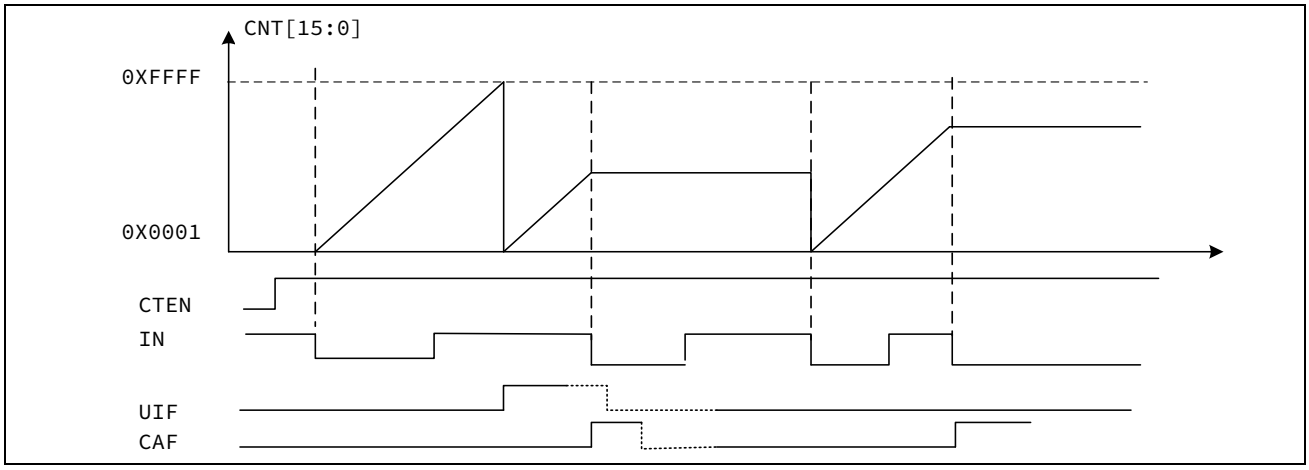


图 14-11　下降沿到下降沿周期测量

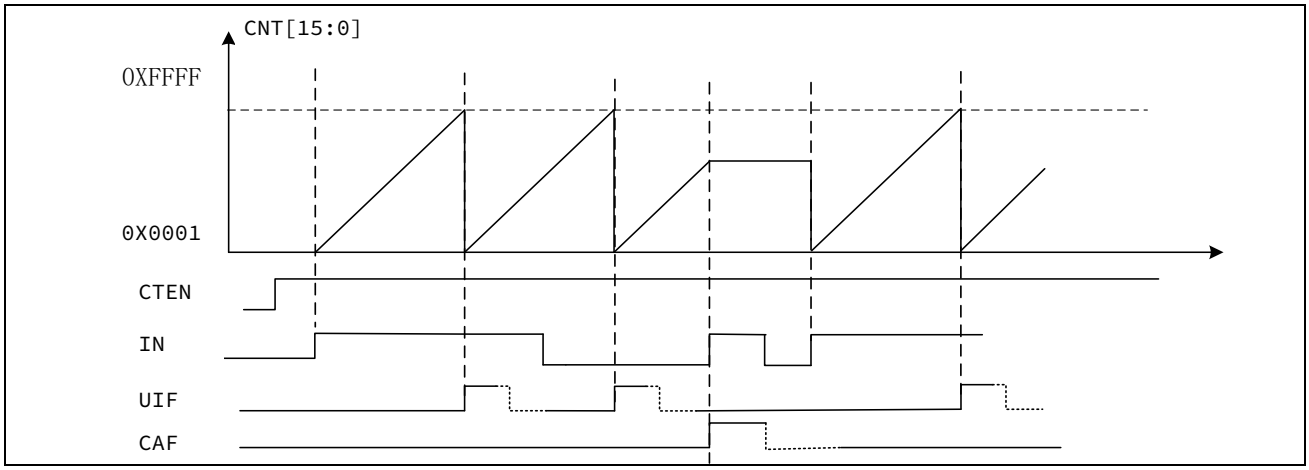


图 14-12　上升沿到上升沿周期测量

通过寄存 MSCR.TS 选择测量信号源。

000 ETFP：　ETR 外部输输入滤波后的相位选择信号，可选择外部滤波与输入反向

001 ITR0：　Timer 内部互联信号 0，其他 timer 的 TRGO 输出

010 ITR1：　Timer 内部互联信号 1，其他 timer 的 TRGO 输出

011 ITR2：　Timer 内部互联信号 2，其他 timer 的 TRGO 输出

100 ITR3：　Timer 内部互联信号 3，其他 timer 的 TRGO 输出

101 IA0ED：无效

110 IAFP：　CH0A 外部输输入滤波后的相位选择信号，可选择外部滤波与输入反向

111 IBFP：　CH0B 外部输输入滤波后的相位选择信号，可选择外部滤波与输入反向

	ITR0	ITR1	ITR2	ITR3
Timer0	-	TIM1_TRGO	TIM2_TRGO	TIM3_TRGO
Timer1	TIM0_TRGO	-	TIM2_TRGO	TIM3_TRGO
Timer2	TIM0_TRGO	TIM1_TRGO	-	TIM3_TRGO
Timer3	TIM0_TRGO	TIM1_TRGO	TIM2_TRGO	-

注：关于 *TRGO* 输出参考寄存器描述。

14.2.5.3 PWC 单次触发模式

设置 M1CR.ONESHOT=1 可以设置 PWC 单次测量，测量完成后 CTEN 将被清除。

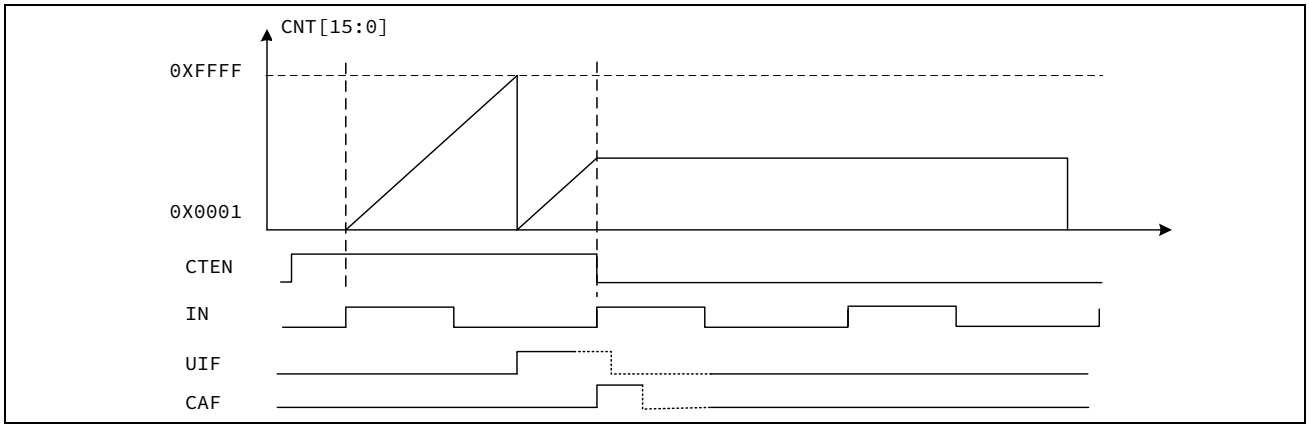


图 14-13　上升沿到上升沿周期测量单次模式

14.2.5.4 设置示例

脉冲低电平测量设置

1. 设置为脉冲测量模式 M1CR.MODE=1
2. 设置 MSCR.TS 选择测量的信号
3. 设置 M1CR.edg2dn=0,M1CR.edg1st=1 选择测量低电平
4. 清除中断标志
5. 使能溢出中断 M1CR.UIE
6. 使能测量结束中断 CR0.CIEA
7. 使能定时器 M1CR.CTEN
8. 中断服务程序中读取 CCR0A 及溢出次数并清除中断标志
9. 等待下次测量

脉冲高电平单次测量设置

1. 设置为脉冲测量模式 M1CR.MODE=1
2. 设置 MSCR.TS 选择测量的信号
3. 设置脉冲单次测量模式 M1CR.ONESHOT=1
4. 设置 M1CR.edg2dn=1, M1CR.edg1st=0 选择测量低电平

5. 清除中断标志
6. 使能溢出中断 M1CR.UIE
7. 使能测量结束中断 CR0.CIEA
8. 使能定时器 M1CR.CTEN
9. 中断服务程序中读取 CCR0A 及溢出次数并清除中断标志
10. 测量结束

14.2.6 模式 2/3 比较捕获模式

14.2.6.1 计数器

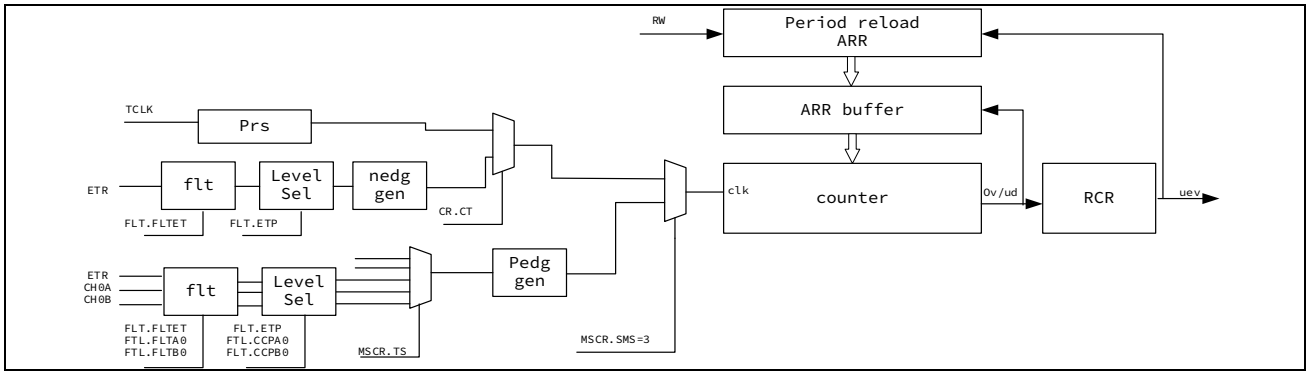


图 14-14 计数器框图

计数器主要部分是一个 16 位计数器与相关的自动装载寄存器。这个计数器可以向上计数（模式 2），向下计数（模式 2）或向上向下双向计数（模式 3）。计数器的时钟可以由预分频器 PRS 分频得到，也可以选择 ETR 输入外部时钟或者通过 MSCR.TS 选择的外部输入信号和内部互联信号。

- 计数器基本单元包括：
- 计数器寄存器 CNT
- 预分频寄存器 CR.PRS
- 自动装载寄存器 ARR
- 重复次数寄存器 RCR
- 时钟选择控制寄存器 FLT,CR0,MSCR,CR

自动装载寄存器具有缓存功能，计数器产生事件更新后重载值从缓存寄存器更新到计数器。当计数器停止状态或者缓存功能关闭状态，自动装载寄存器立刻更新到缓存寄存器。当定时器处于运行状态并行缓存功能有效时，写入到自动装载寄存器的值不会立刻更新到缓存寄存器，当事件更新后才有自动装载寄存器更新到缓存寄存器。

时钟选择及门控功能，触发功能，复位功能参考模式 2/3 从模式章节。

14.2.6.2 计数器波形

模式 2 为锯齿波计数波形，通过设置 CR.DIR 可以更改计数方向。

设置 CR.DIR 为 0 时，计数器为递增计数模式，这种模式下，计数器从 0 计数到自动重载值（TIMx_ARR），然后重新 从 0 开始计数并生成计数器上溢事件。如果使用重复计数器，则当递增计数的重复次数达到重复计数器寄存器中编程的次数加一次(TIMx_RCR+1) 后，将生成更新事件 (UEV)。否则，将在每次计数器上溢时产生更新事件。

将 TIMx_CR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_IFR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 自动重载缓存值将以 ARR 寄存器值进行更新
- 比较缓存值将以比较寄存器 CCRxy 进行更新

以下图示显示 ARR=0X2C 时不同计数方向的计数器波形

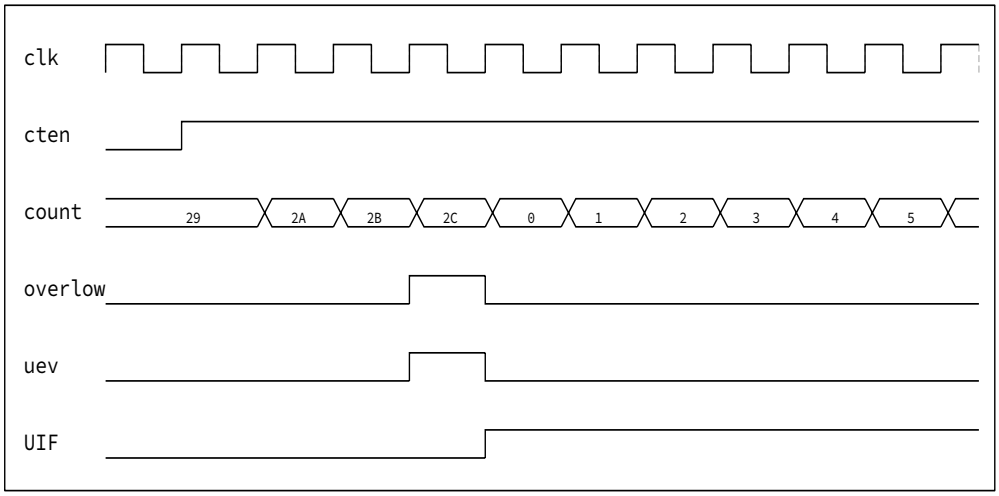


图 14-15 无预分频的向上计数

计数溢出周期时钟个数为 ARR+1，

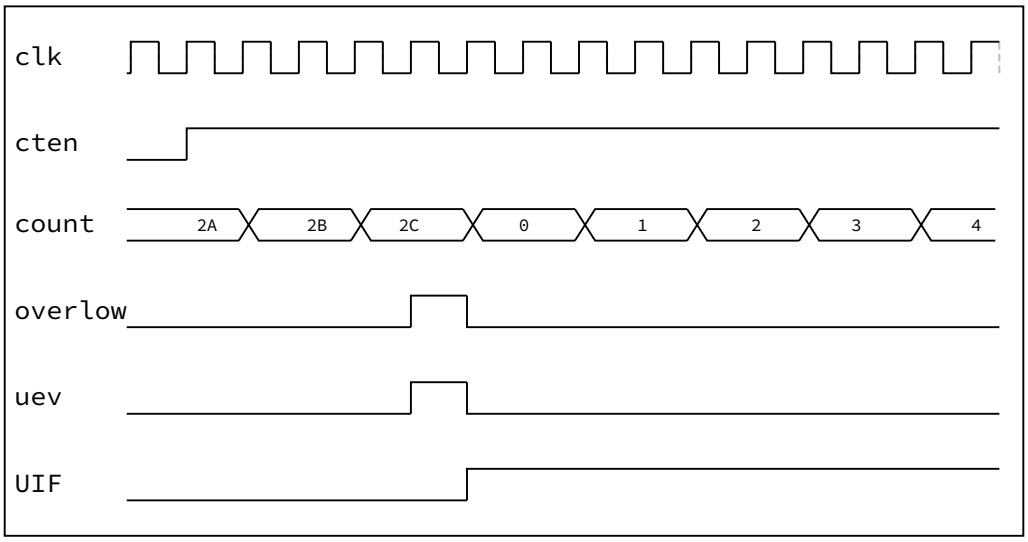


图 14-16　带预分频的上计数

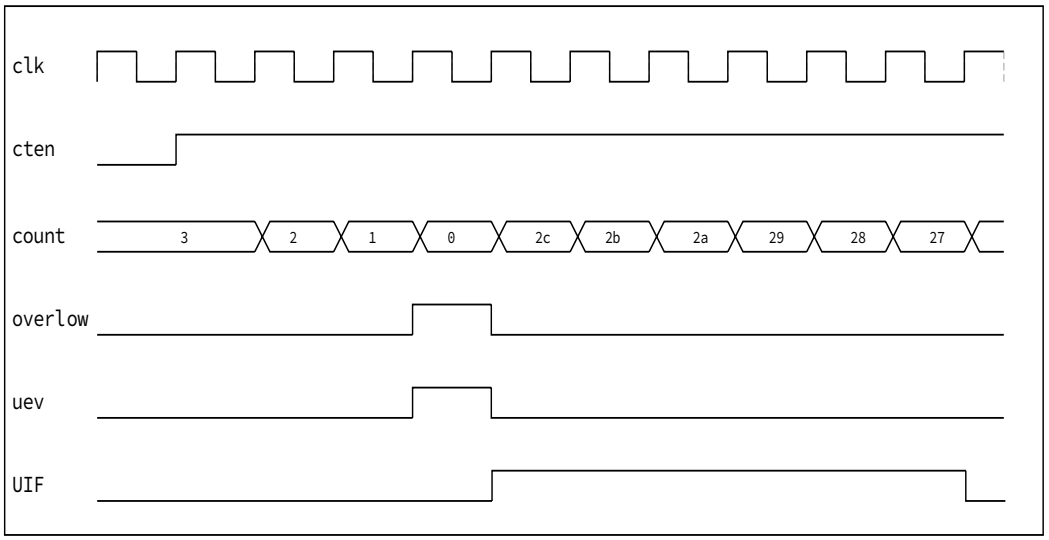


图 14-17　不带预分频的下计数

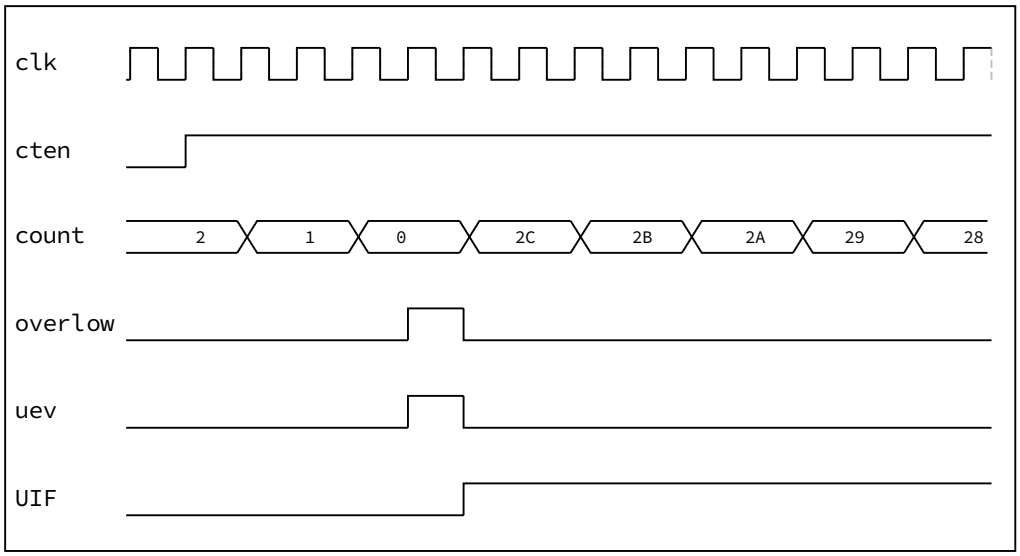


图 14-18　带预分频的下计数

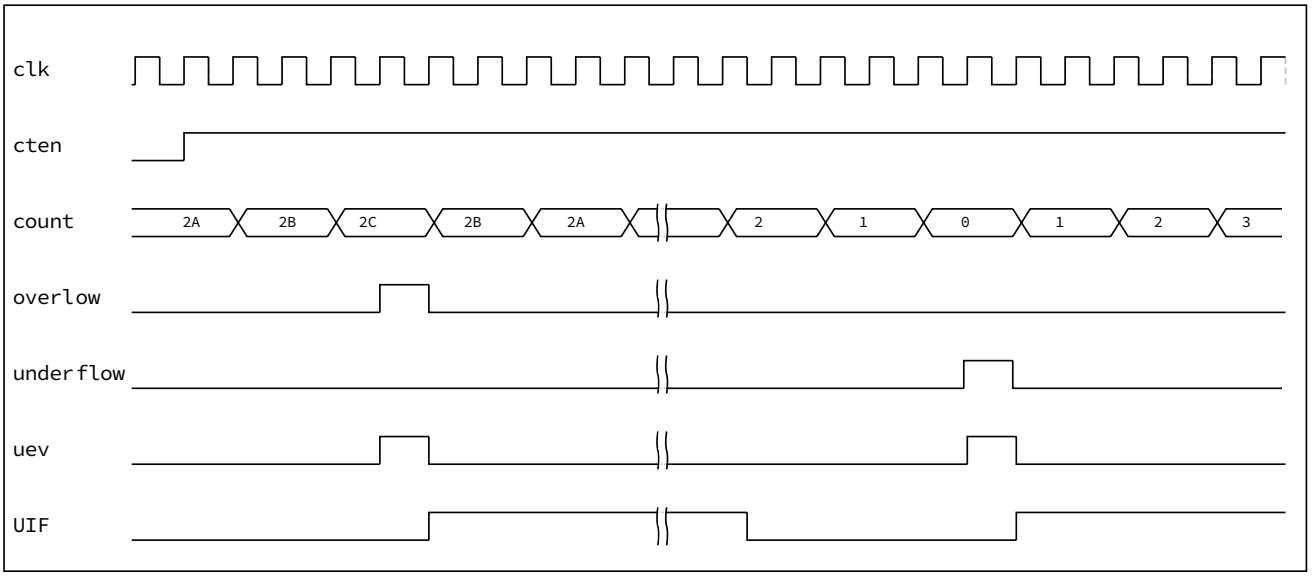


图 14-19　带预分频的上下计数

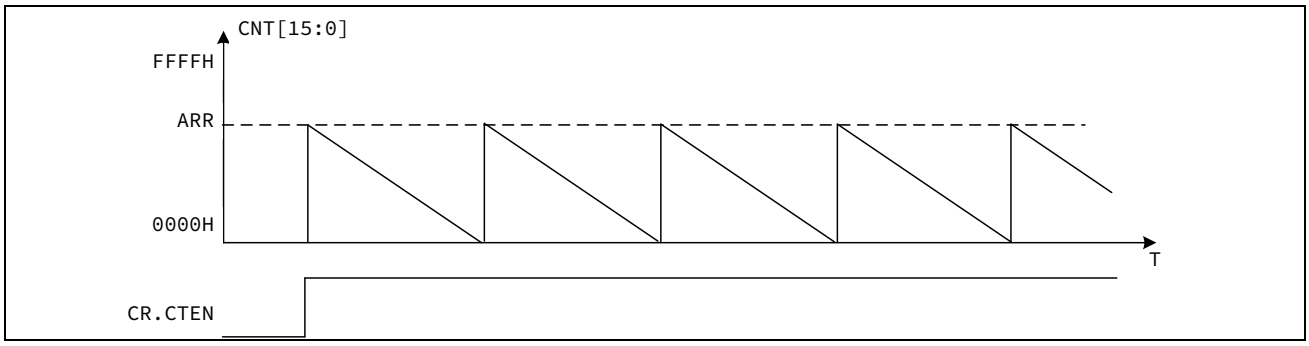


图 14-20　边沿对齐计时器波形(DIR =1)

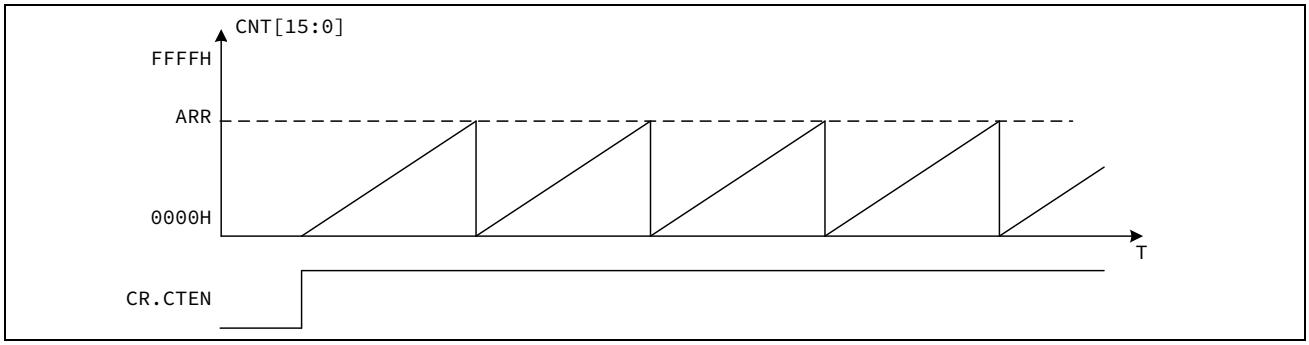


图 14-21 边沿对齐计时器波形(DIR =0)

模式 3 为三角波计数波形，计数方向控制位只读，不可以更改计数方向。

中心对齐（三角波）模式下 CR.DIR 方向位是只读的。写值无效。从其他模式切换到中心对齐模式 DIR 自动清 0。软件事件更新和从模式外部触发复位模式 DIR 自动清零。

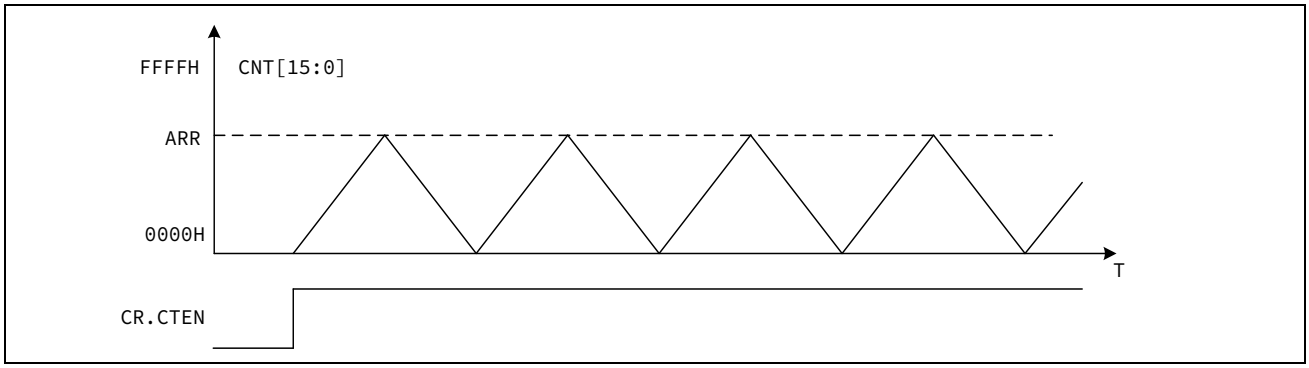


图 14-22 中心对齐计数器波形

14.2.6.3 重复计数

重复计数器使用计数器的溢出进行向下计数。计数到 0 时，即计数器发生重复寄存器设置的值加一次溢出时。当缓存寄存器使能时，周期重载寄存器更新到周期缓存寄存器。比较模式下比较寄存的值更新到比较缓存寄存中。

重复计数器在下面条件成立时递减

- 上计数模式下每次计数器溢出时
- 下计数模式下每次计数器下溢时
- 三角波模式下每次上溢出和每次下溢出时

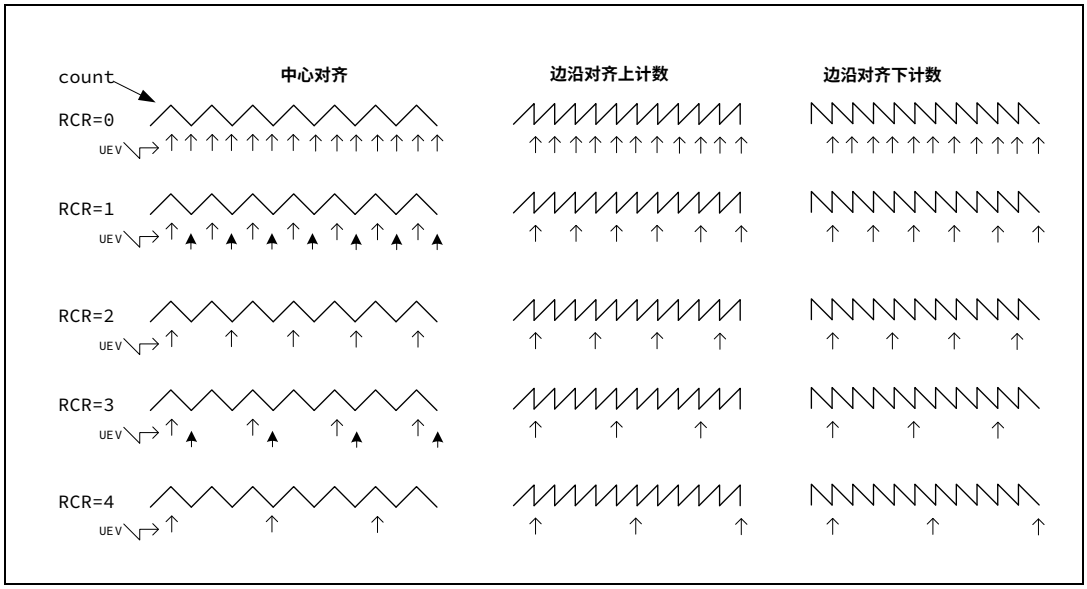


图 14-23 重复计数器产生更新时序

除了上下溢出通过重复计数器可以产生事件更新 UEV 外，还可以通过写寄存器 CR.UG 产生软件及从模式复位事件更新 UEV；这时需要配置 CR.URS。

14.2.6.4 数据缓存

自动重载数据 ARR 与比较寄存器都可以配置缓存功能，当缓存功能有效时，当发生 UEV 事件更新时，写入的周期值 ARR 与比较值 CCR 才会生效。

自动重载值在不同计数模式下的更新时序图如下：

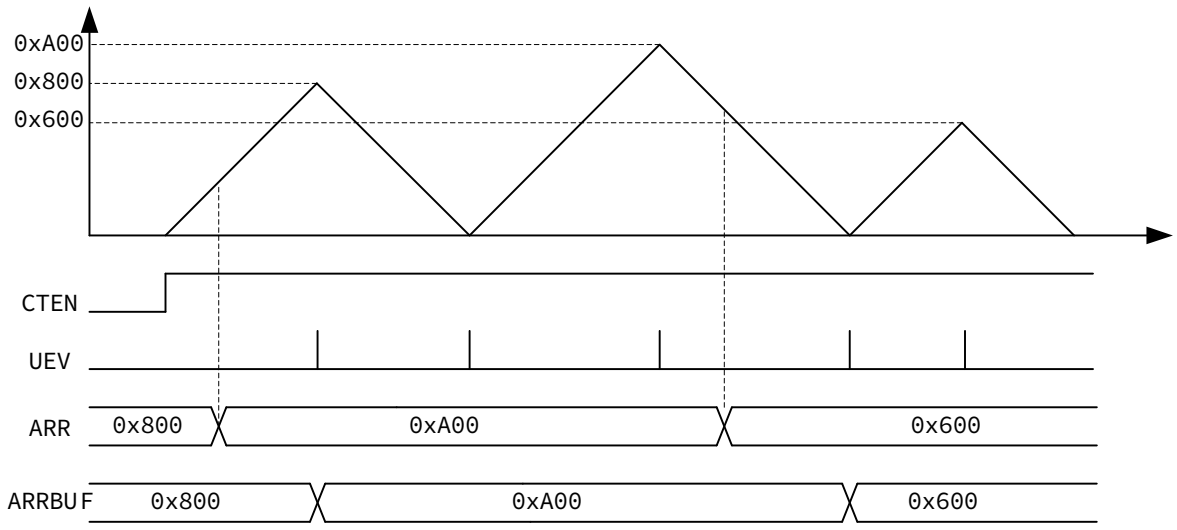


图 14-24 三角波模式下缓存使能

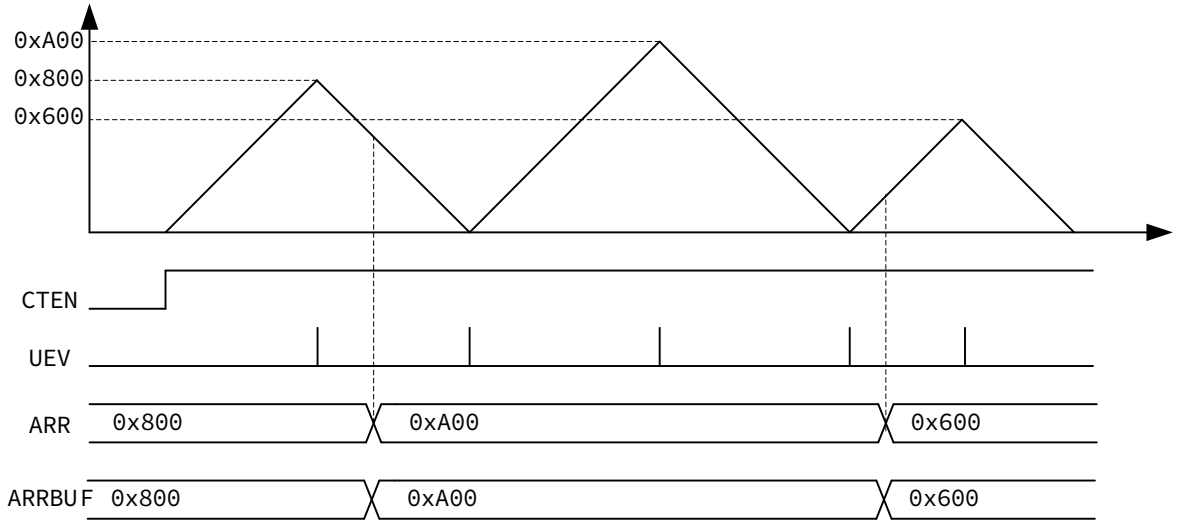


图 14-25　三角波模式下缓存无效

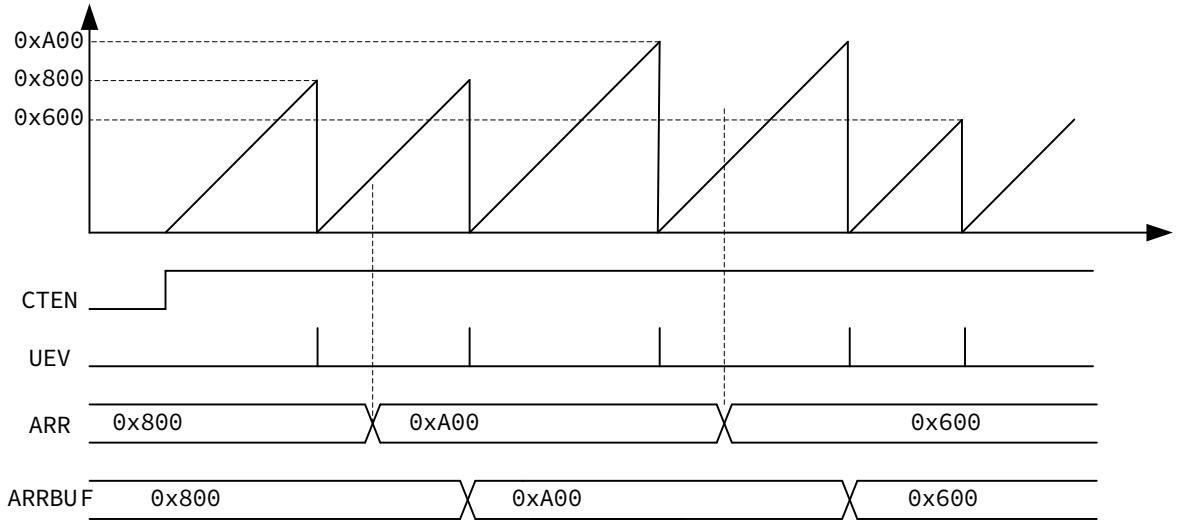


图 14-26　锯齿波模式下上计数缓存使能

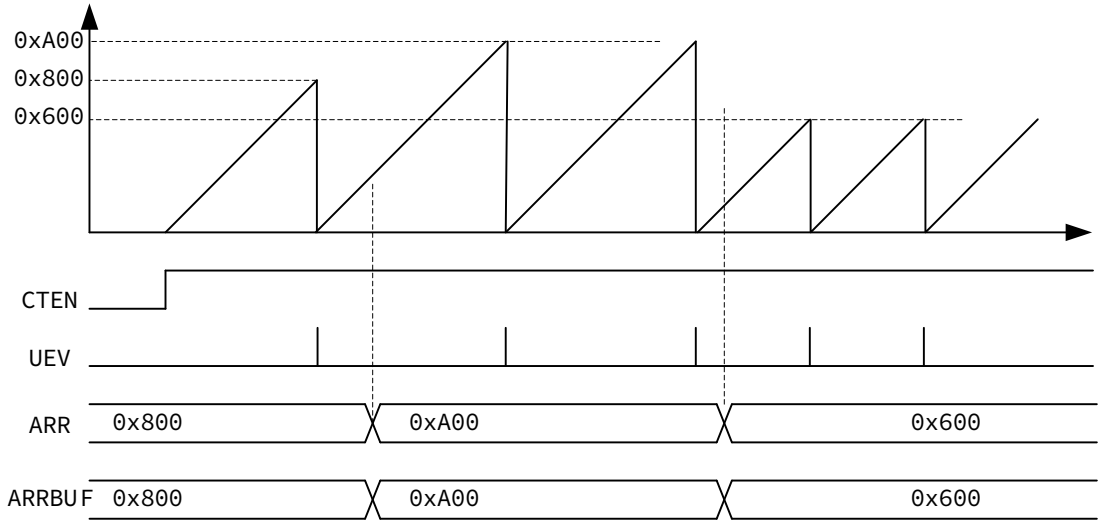


图 14-27　锯齿波模式下上计数缓存无效

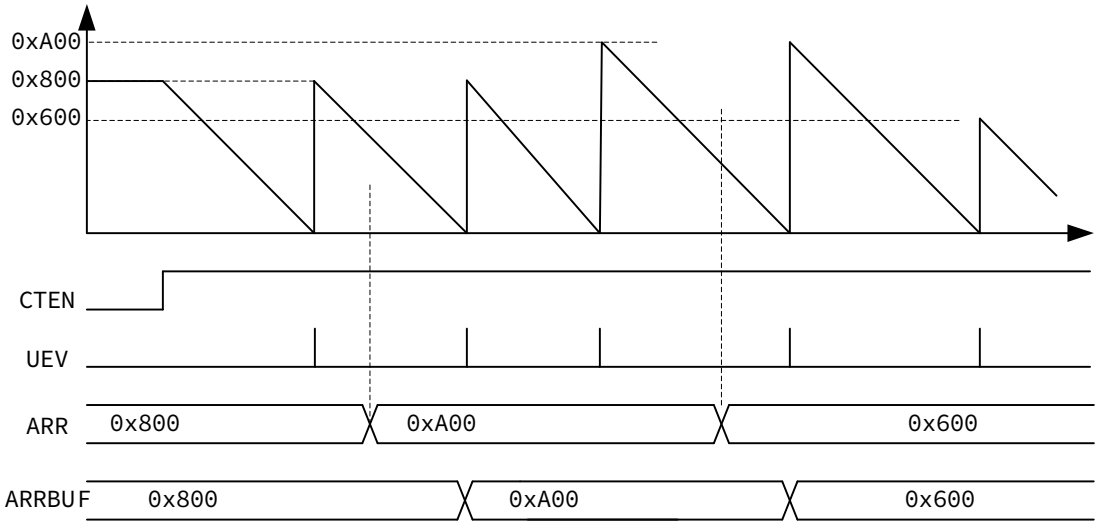


图 14-28　锯齿波模式下计数缓存使能

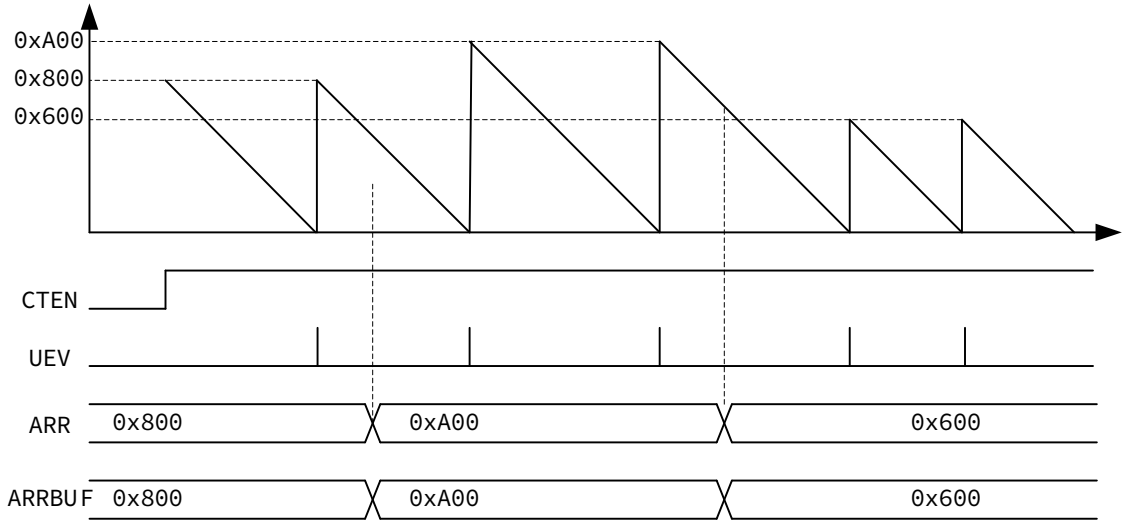


图 14-29　锯齿波模式下计数缓存无效

在三角波模式与锯齿波上计数模式时，如果缓存不使能，更改的 ARR 时，当前计数器的值要小于要更改的 ARR 周期值，否则当前周期会计数到 0XFFFF。

比较缓存与周期缓存更新状态一致，这里不一一列出时序图。

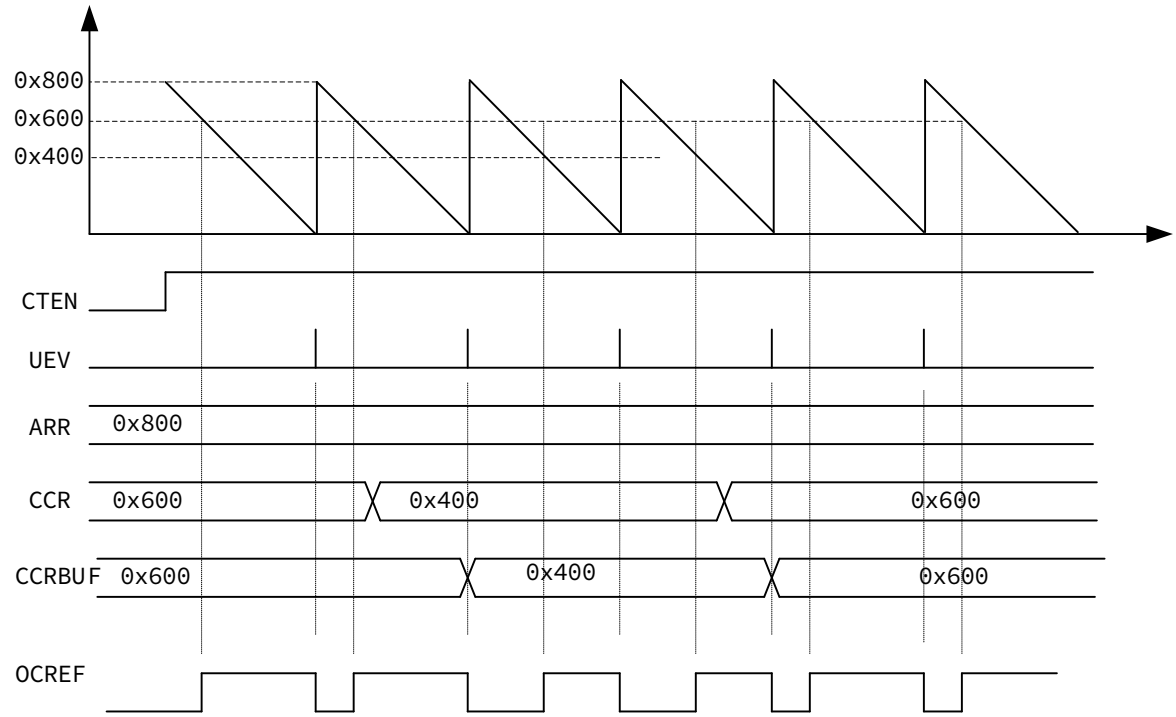


图 14-30　锯齿波模式下计数比较缓存使能

14.2.6.5 比较输出 OCREF

比较输出 OCREFA 可以配置为单点比较，使用比较寄存器 CCRA 控制 OCREFA 的输出；OCREFA 的比较输出也可以配置为双点比较，使用比较寄存器 CCRA,CCRB 一起控制 OCREFA 的比较输出。

OCREFB 的比较输出只能使用单点比较，使用比较寄存器 CCRB 控制 OCREFB 的比较输出。

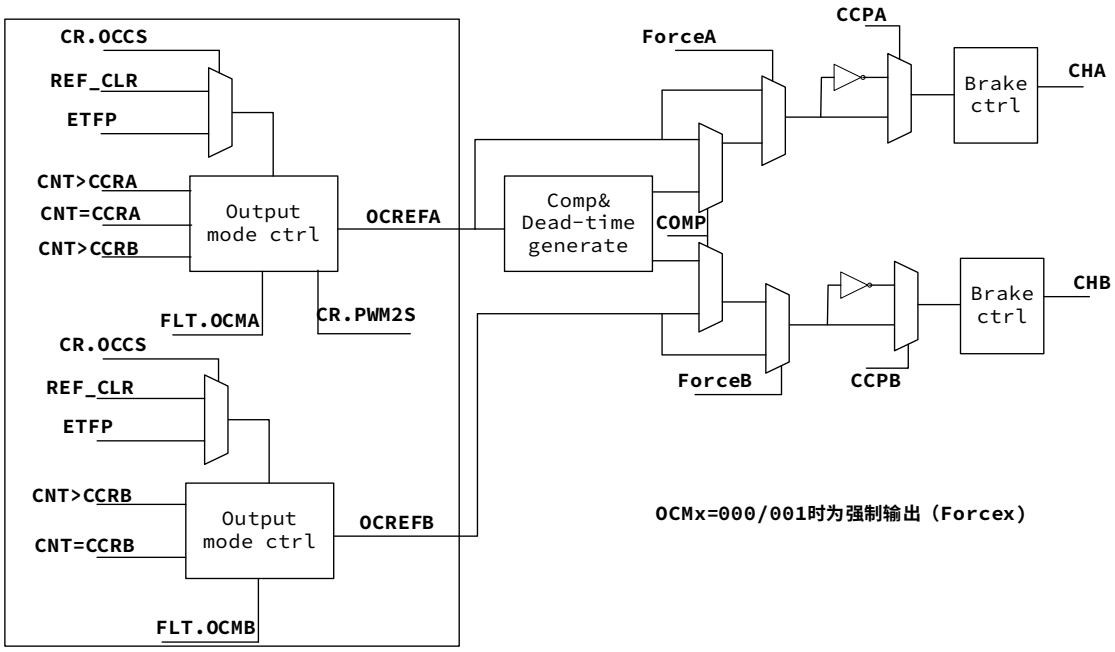


图 14-31 OCREF 输出框图

OCREF 输出使用 OCMx 选择

000：强制为 0

001：强制为 1

010：比较匹配时强制为 0

011：比较匹配时强制为 1

100：比较匹配时翻转

101：比较匹配时输出一个计数周期的高电平

110：PWM 模式 1

单点比较：

上计数时 CNT<CCRxy 输出高，下计数时 CNT>CCRxy 输出为低电平

双点比较：

1）锯齿波上计数 CCRxA<CNT≤CCRxB 输出为低电平

2）锯齿波下计数 CCRxA<CNT≤CCRxB 输出为高电平

3）三角波上计数 CNT<CCRxA 输出高，下计数 CNT>CCRxB 为低电平

111：PWM 模式 2

单点比较：

上计数时 CNT<CCRxy 输出低，下计数时 CNT>CCRxy 输出为高　电平

双点比较：

1）锯齿波上计数 CCRxA≤CNT<CCRxB 输出为高电平

2）锯齿波下计数 CCRxA≤CNT<CCRxB 输出为低电平

3）三角波上计数 CNT<CCRxA 输出低，下计数 CNT>CCRxB 为高电平

注：强制输出有高优先级，当强制输出有效时，互补输出控制无效。

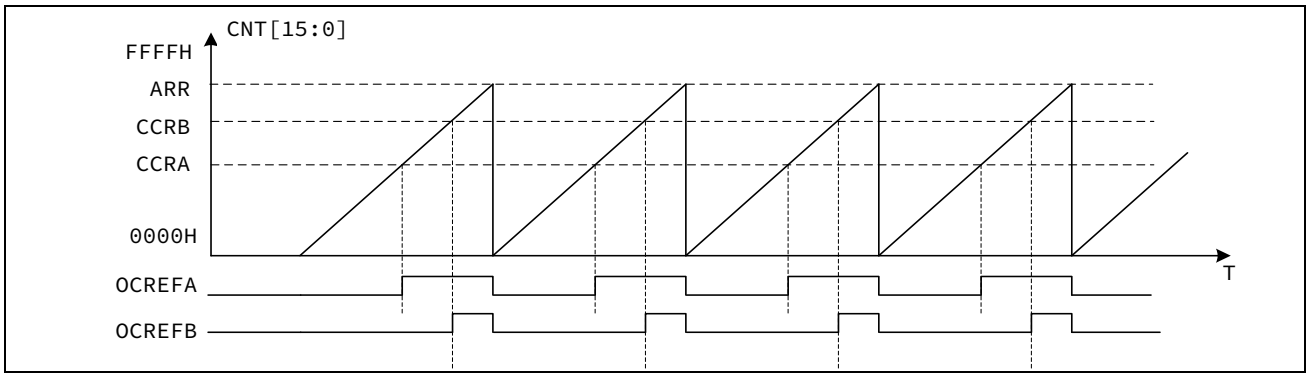


图 14-32　锯齿波计数单点比较 OCREF 输出波形（OCMx=111）

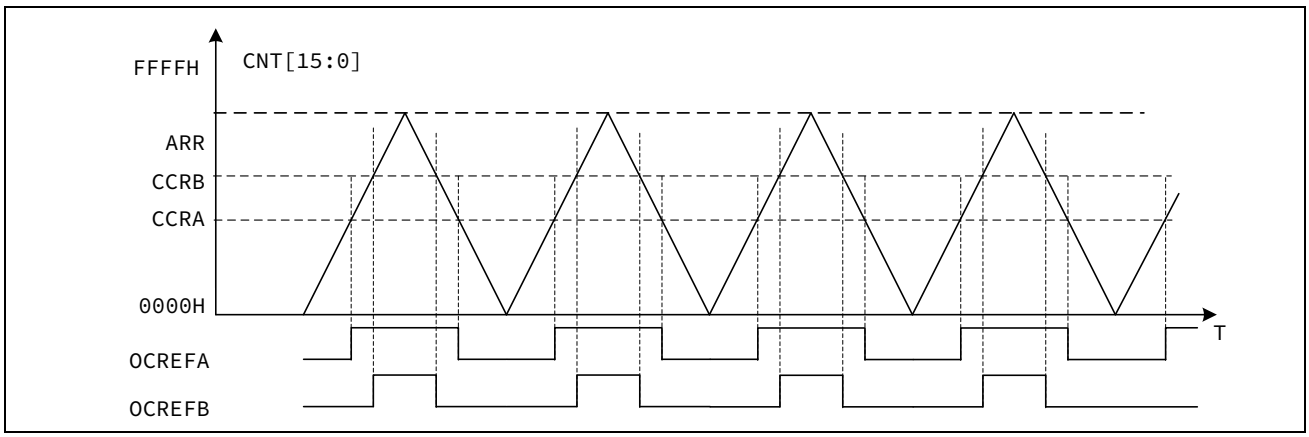


图 14-33　三角波计数单点比较 OCREF 输出波形（OCMx=111）

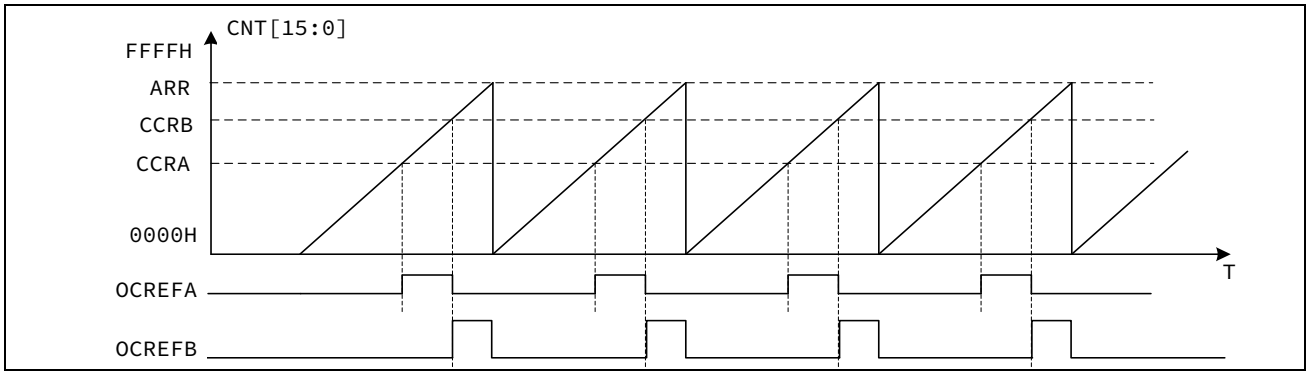


图 14-34　锯齿波计数双点比较 OCREF 输出（OCMx=111）

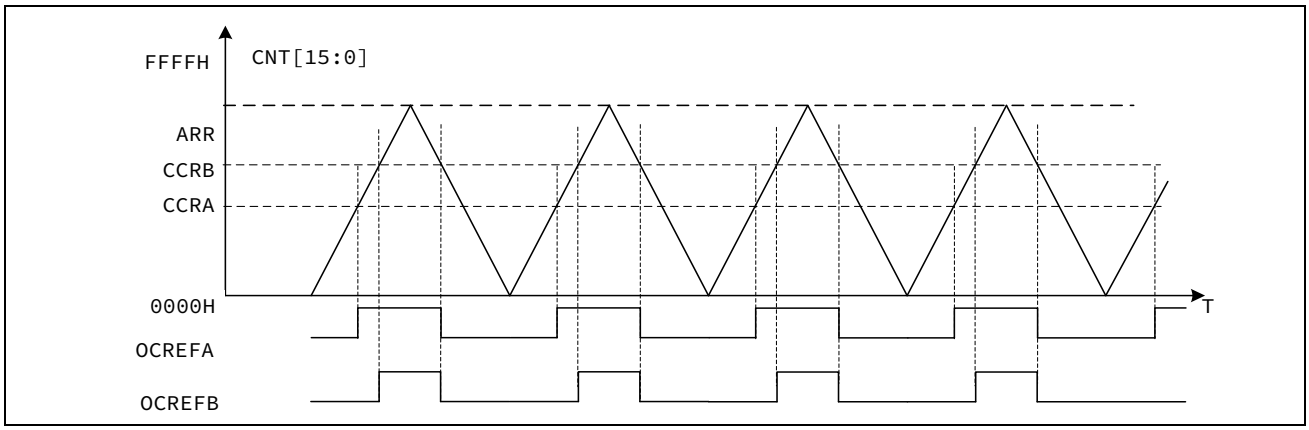


图 14-35　三角波计数双点比较 OCREF 输出（OCMx=111）

14.2.6.6 独立 PWM 输出

由 OCREFA 控制 CHA 的输出，OCREFB 控制 CHB 的输出。通过 CRCHx.CCPA，CRCHx.CCPB 可以控制 CHA，CHB 输出的反向。

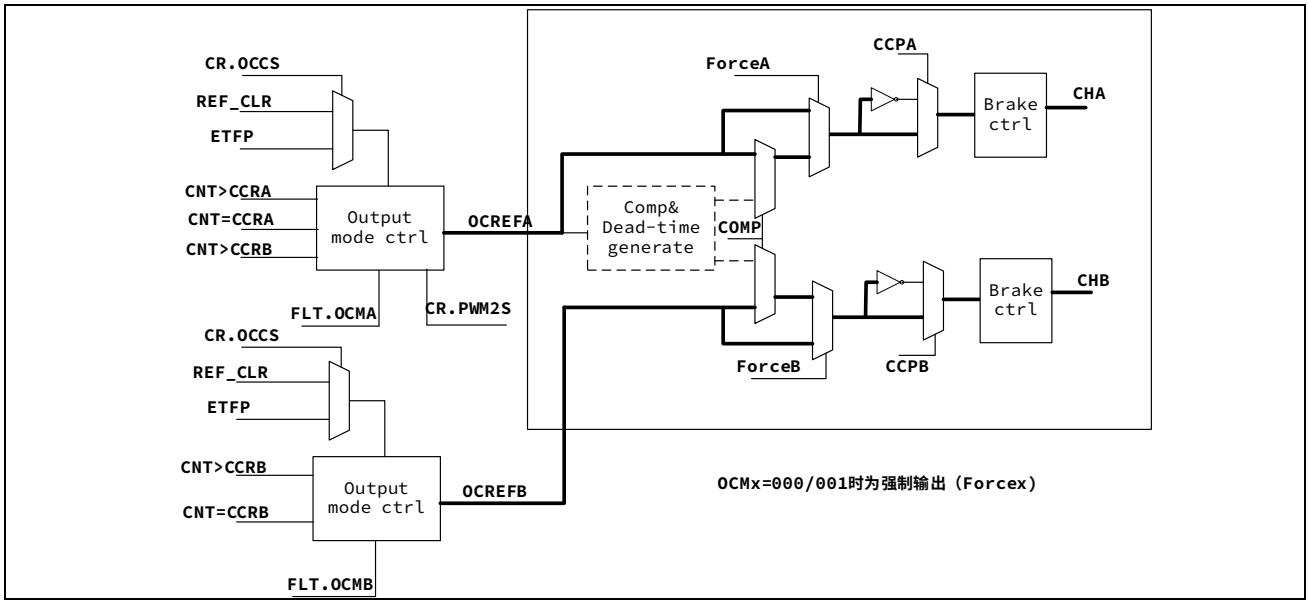


图 14-36　独立 PWM 输出框图

PWM 输出与 OCREF 关系

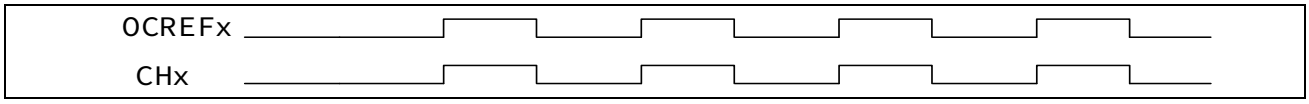


图 14-37　CCPx=0 时 PWM 输出波形

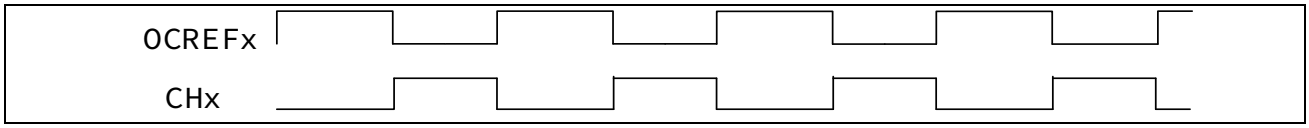


图 14-38　CCPx=1 时 PWM 输出波形

14.2.6.7 互补 PWM 输出

由 OCREFA 控制 CHA 的输出，OCREFA 同时控制 CHB 的输出。比较寄存器 CCRxB 可以作为专用比较控制 ADC 触发。

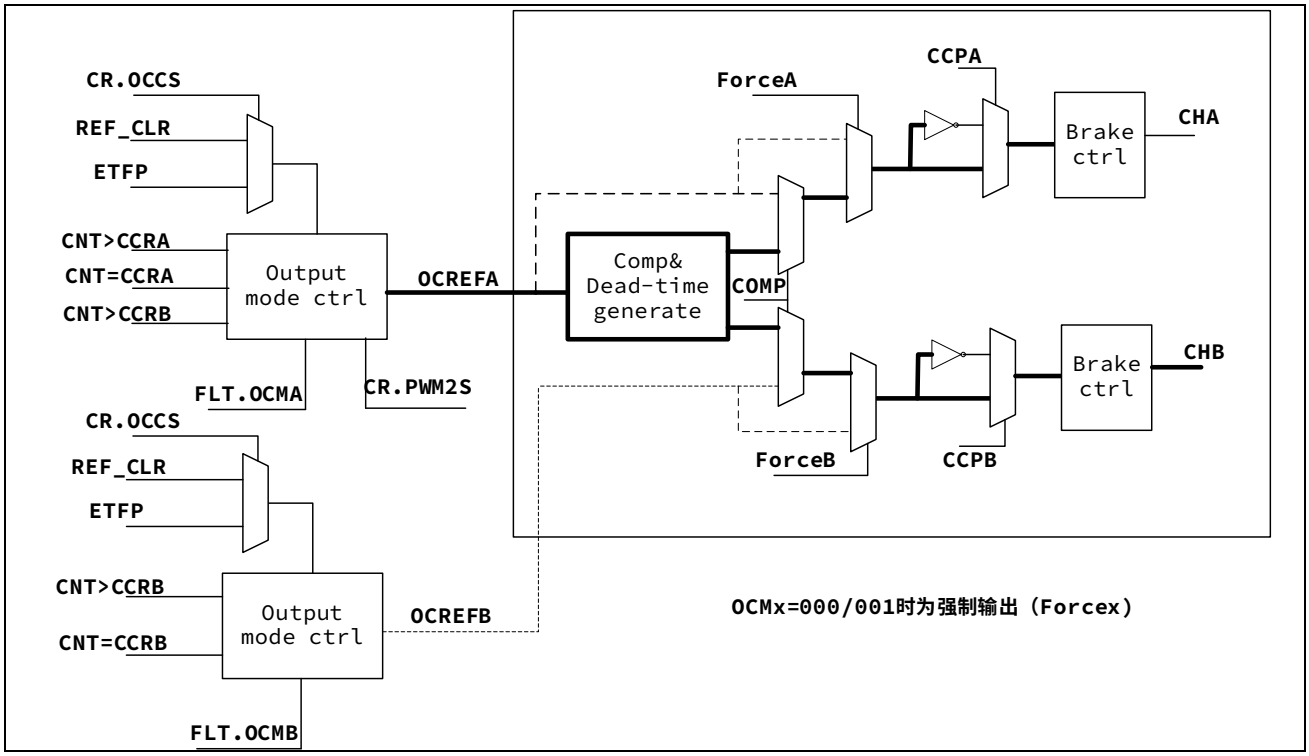


图 14-39 互补 PWM 输出框图

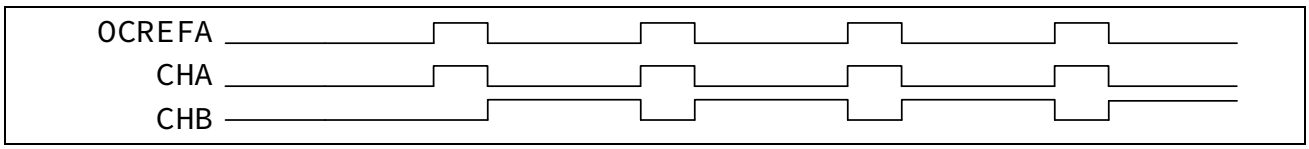


图 14-40 互补 PWM 输出波形图

14.2.6.8 有死区的 PWM 输出

在互补 PWM 输出模式下可以设置死区功能。

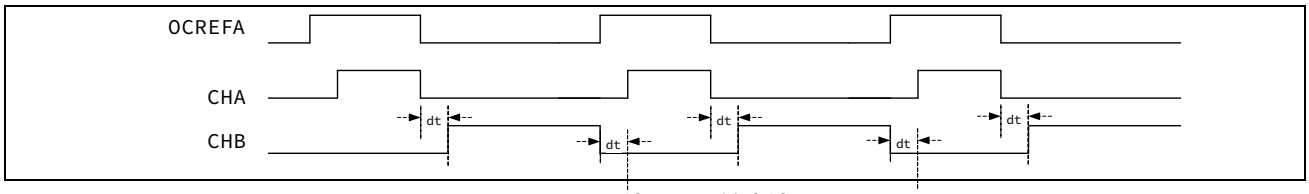
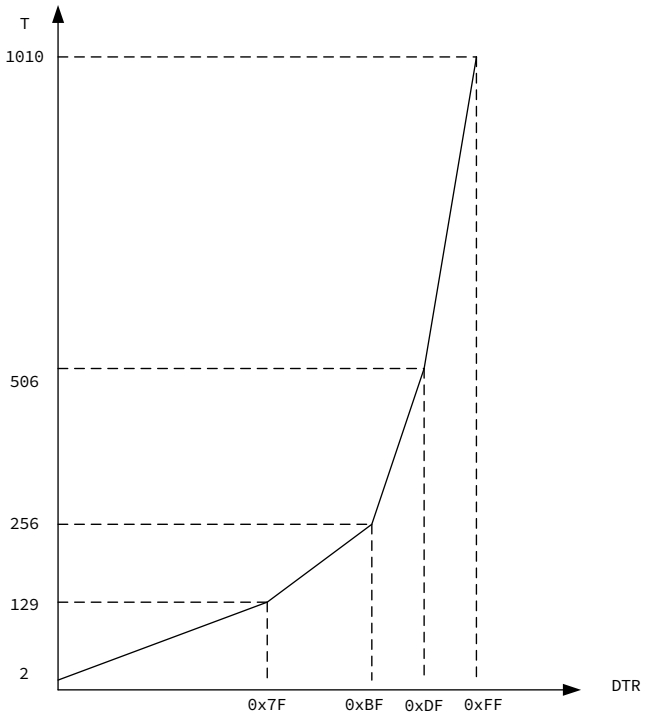


图 14-41 互补 PWM 输出波形图

死区时间使用 8 位 DTR 控制，死区时间 dt 与 DTR 的关系如下

DTR[7] =0	T=DTR[6:0]+2	2-129	step=1
DTR[7:6] =10	T={DTR[5:0]+64}*2 +2	130-256	step=2
DTR[7:5] =110	T={DTR[4:0]+32}*8 +2	258-506	step=8
DTR[7:5] =111	T={DTR[4:0]+32}*16 +2	514-1010	step=16



100M clock 死区时间由2.56us调整为10.1us

图 14-42 死区时间

14.2.6.9 单脉冲输出

单脉冲模式（ONE SHOT）是上述 PWM 模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，产生一个脉宽可编程的脉冲，如下图所示。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将

TIMx_M23CR 寄存器中的 ONESHOT 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

单脉冲模式在锯齿波下计数模式计数器初值不要设置为 0，上计数模式计数值不要设置到大于等于 ARR。

只有当比较值与计数器初始值不同时，才能正确产生输出脉冲。启动前（定时器等待触发时），必须进行如下配置：

- 递增计数时：CNT<CCRxy≦ ARR（特别注意，0<CCRxy），
- 递减计数时：CNT>CCRxy。

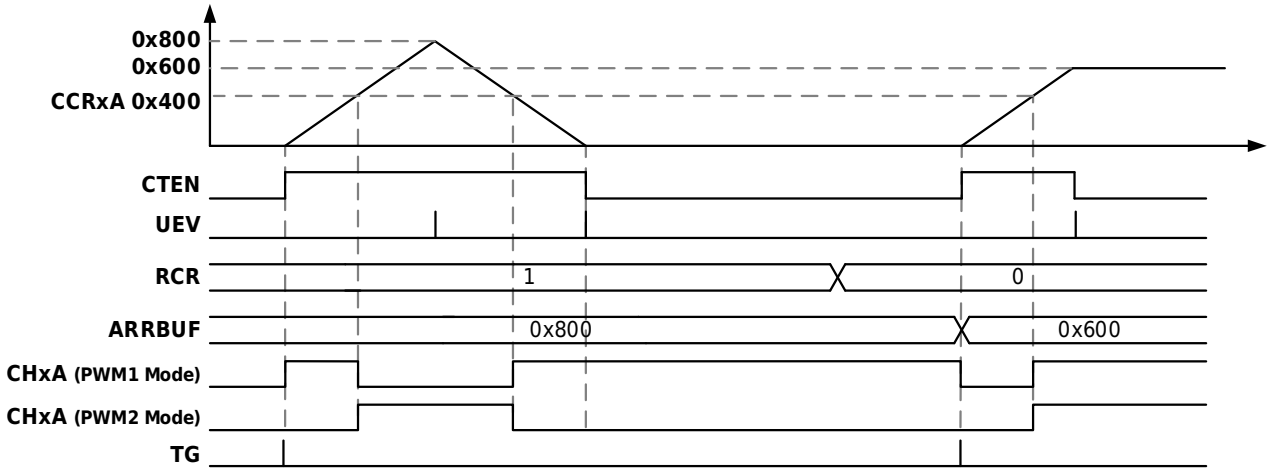


图 14-43 三角波模式单脉冲计数

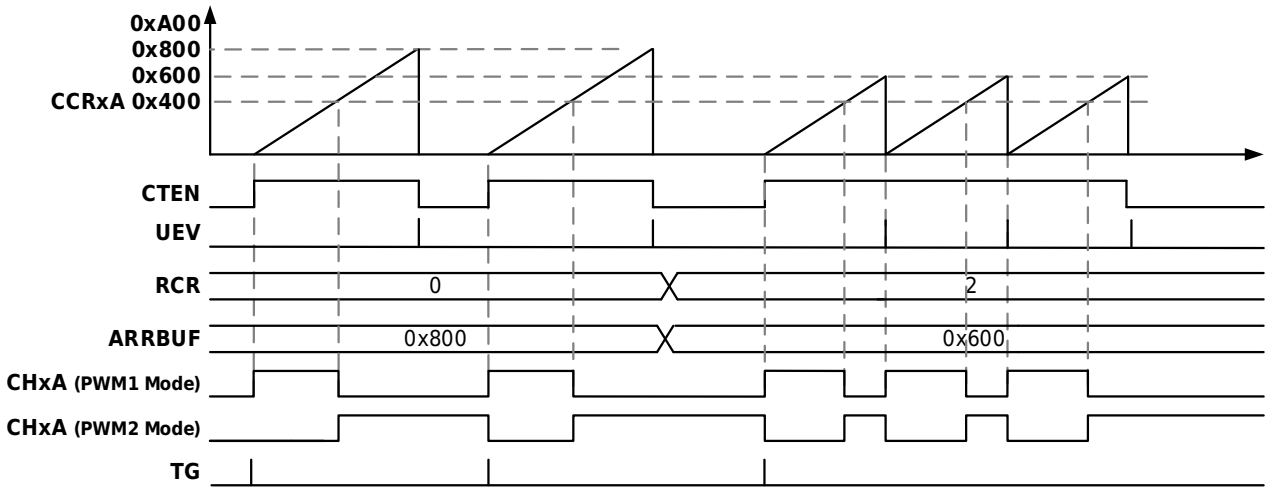


图 14-44 锯齿波上计数单脉冲模式

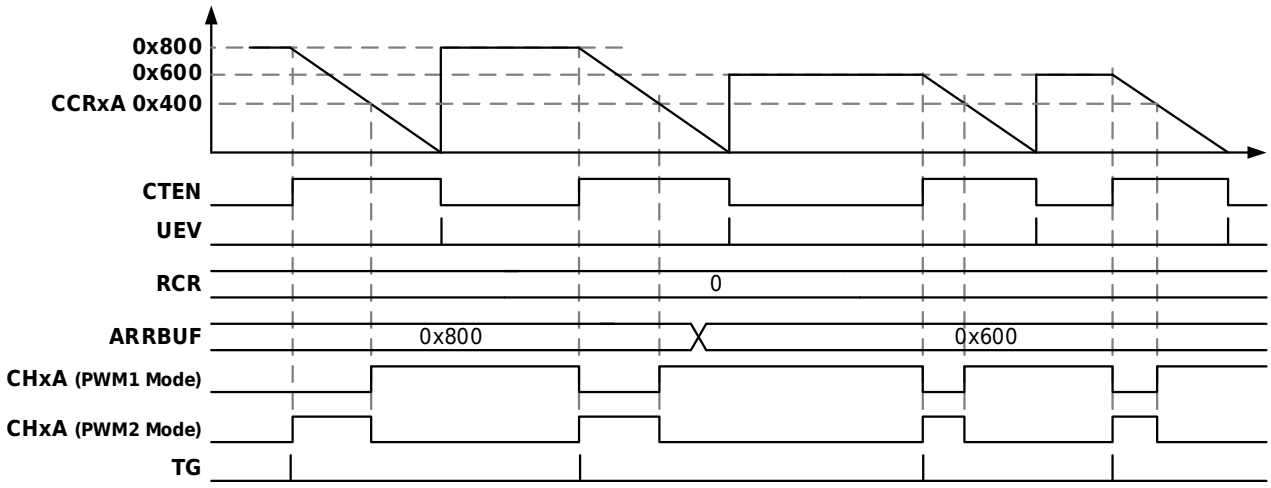


图 14-45 锯齿波下计数单脉冲模式

注意：上计数时，配置为 PWM2 模式时，由于通道初始化输出为低电平，所以第一次触发会产生一个上升沿并保持高电平，待到第二次触发时，可以输出一个脉宽可编程的低电平脉冲；下计数时，配置为 PWM1 模式时，由于通道初始化输出为低电平，所以第一次触发会产生一个上升沿并保持高电平，待到第二次触发时，可以输出一个脉宽可编程的低电平脉冲。

14.2.6.10 比较中断

锯齿波比较匹配会将 IFR 寄存器相应的标志置 1，如果中断使能 CRCHx.CIEy (x=0,1,2;y=A,B)将会触发中断。

三角波比较匹配可单独选择上升计数比较匹配，下降计数比较匹配或者两者比较都匹配。

比较 A 比较匹配当计数值与 CCRxA 相等时，统一使用 CR.CIS 控制。

CR.CIS=2’b00 时 比较匹配无输出

CR.CIS=2’b01 时 上计数时比较匹配

CR.CIS=2’b10 时 下计数时比较匹配

CR.CIS=2’b11 时 上下计数时比较都匹配

比较 B 比较匹配当计数值与 CCRxB 相等时，不同通道可以使用 CRCHx.CISB 单独控制。

CRCHx.CISB=2’b00 时 通道 x 比较匹配无输出

CRCHx.CISB=2’b01 时 通道 x 上计数时比较匹配

CRCHx.CISB=2’b10 时 通道 x 下计数时比较匹配

CRCHx.CISB=2’b11 时 通道 x 上下计数时比较都匹配

B 通道比较匹配单独控制为了更灵活的触发 ADC。详见 Timer 触发 ADC 章节。

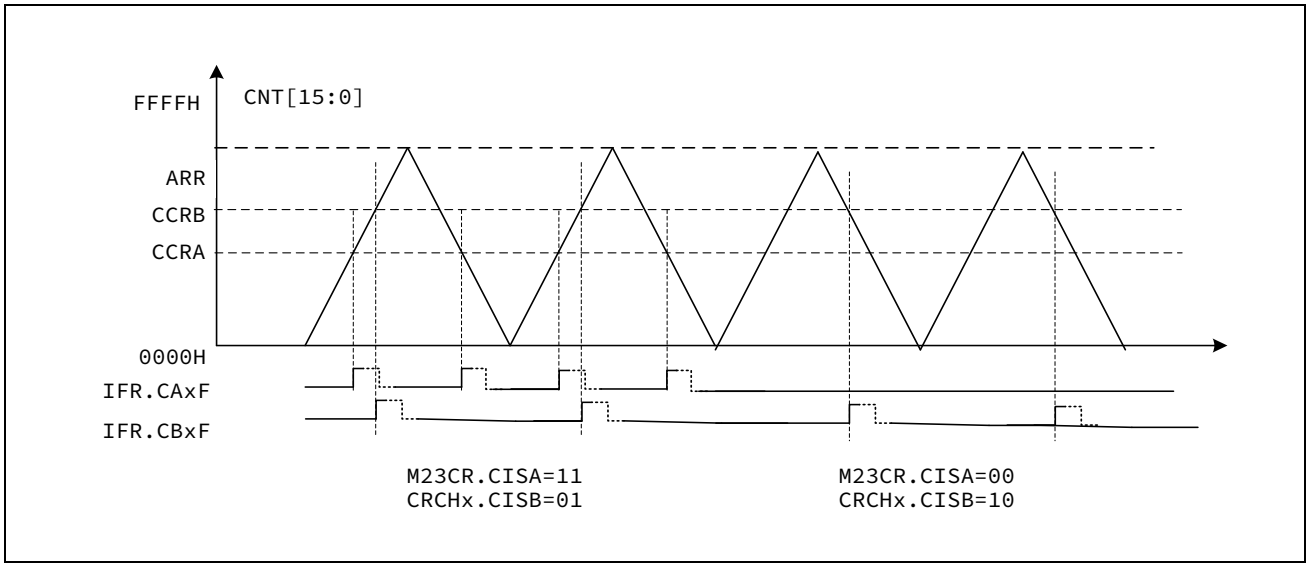


图 14-46 中断示意图

14.2.6.11 捕获输入

CR.MODE=2/3

在三角波计数或者锯齿波计数模式下都可以设置捕获功能，可以设置捕获的电平边沿，当发生捕获时捕获的值存入比较捕获寄存器并产出捕获中断。

每个通道的比较捕获功能可以单独设置，通过寄存器 CRCHx.CSA/CSB 选择。

每个通道的捕获边沿可以单独设置，通过寄存器 CRCHx.CFy/CRy(x=0/1/2;y=A/B)选择捕获触发的边沿。

当捕获发生后捕获标志未清除前再次发生捕获动作，会产生捕获数据覆盖标志。

定时器未启动时如果有效的捕获边沿也会产生捕获标志及捕获动作。

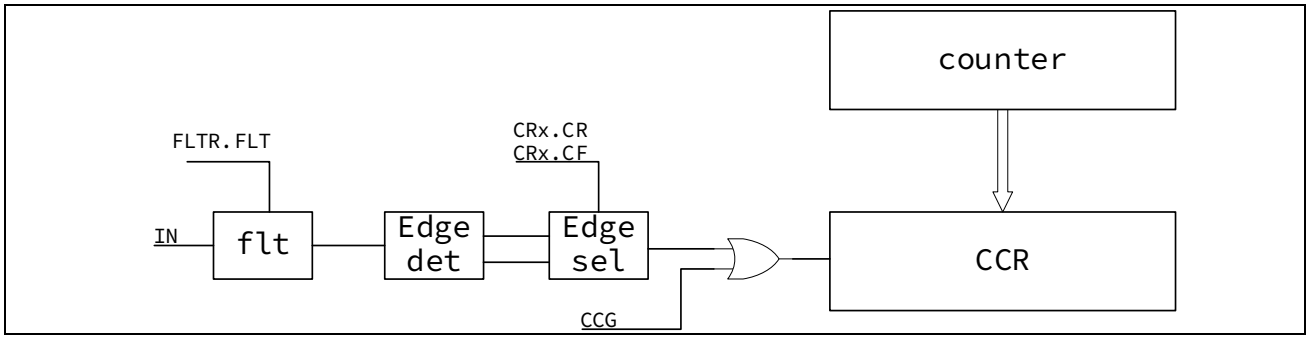


图 14-47 捕获功能框图

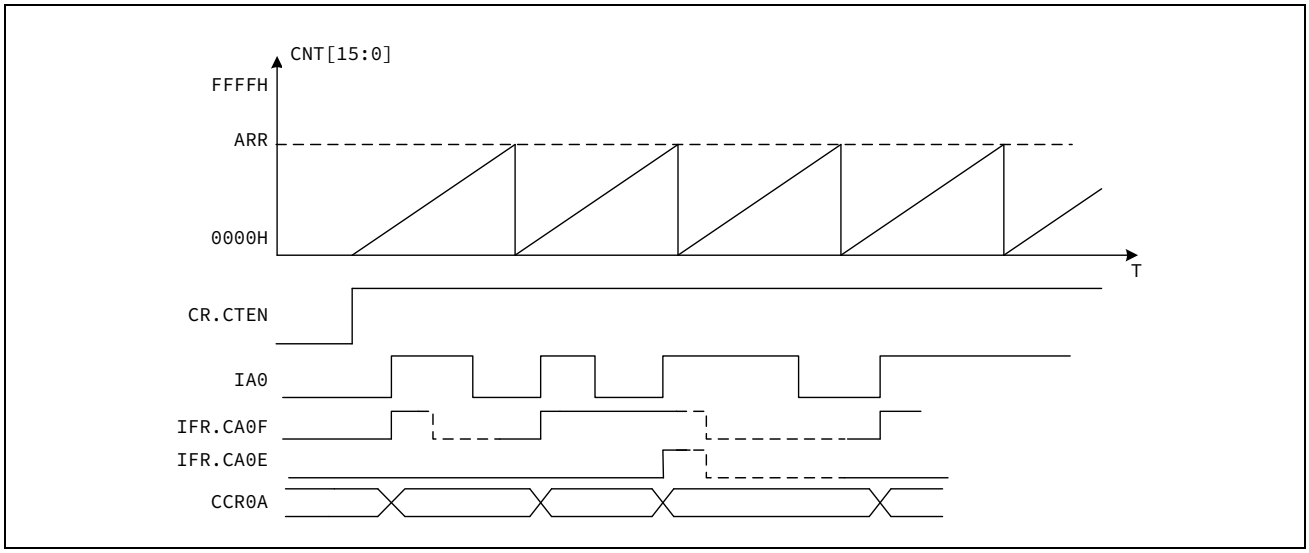


图 14-48 捕获时序图

CH0A 的捕获输入通过 MSCR.IA0S 选择 CH0A 输入还是 CH0A，CH1A，CH2A 的异或输入。

IA0S	0	1
Timer0/1/2	CHA0	CHA0 ETR GATE异或输入
Timer3	CHA0	CHA0 CHA CHA2异或输入

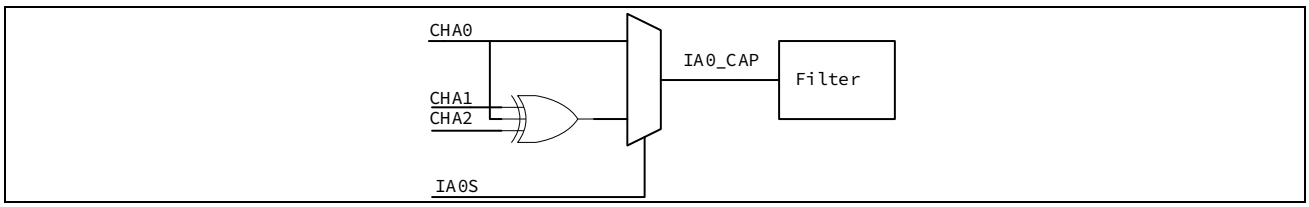


图 14-49 CHA 端口选择

CH0B 的捕获输入通过 MSCR. IB0S 选择 CH0B 输入还是内容 MSCR.TS 选择的信号。

<table>
<tr><th>IB0S</th><th>0</th><th>1</th></tr>
<tr><td>Timer0/1/2/3</td><td>CHB0</td><td>内部触发 MSCR.TS 选择信号
<table>
<tr><td>MCSCR.TS</td><td>通道 B 捕获输入</td></tr>
<tr><td>000</td><td>ETR 滤波相位选择输出信号，滤波相位选择可配</td></tr>
<tr><td>001</td><td>内部互联 ITR0</td></tr>
<tr><td>010</td><td>内部互联 ITR1</td></tr>
<tr><td>011</td><td>内部互联 ITR2</td></tr>
<tr><td>100</td><td>内部互联 ITR3</td></tr>
<tr><td>101</td><td>CH0A 边沿</td></tr>
<tr><td>110</td><td>CH0A 滤波输出信号，滤波功能可配</td></tr>
<tr><td>111</td><td>Reserved</td></tr>
</table></td></tr>
</table>

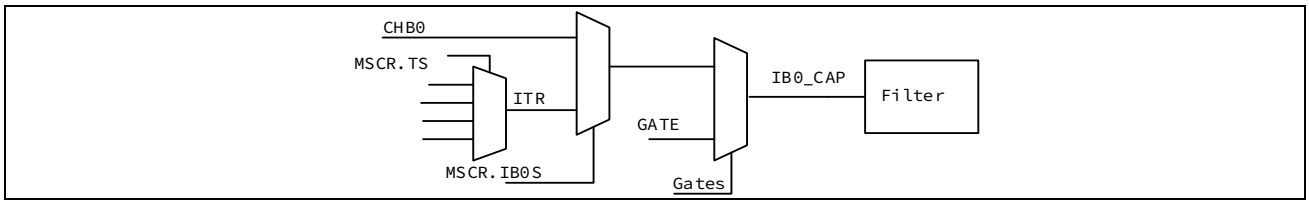


图 14-50 CHB 端口选择

Gates 信号为选择 PWM 互补输出功能生成的信号，当 PWM 互补输出时，比较捕获寄存器 CCR0B 没有使用，自动切换到使用 GATE 作为捕获/比较输出。

注意：当 GATE 配置为比较输出时，GATE 的输出模式仅支持配置为强制输出为 0 或 1 和 PWM 模式有效（即 OCMB0 配置为 0/1/6/7 时，GATE 输出有效）。

14.2.6.12 设置示例

边沿对齐独立 PWM 输出设置

1. 设置模式 CR.MODE=2
2. 设置锯齿波计数方向 CR.DIR
3. 设置 PWM 周期值 ARR
4. 设置计数器初值（初值必须小于周期值）
5. 设置 PWM 比较值 CCRxA,CCRxB
6. 设置 PWM 输出比较模式 FLT.OCMA FLT.OCMB 为 6 或 7
7. 清除相关中断标志
8. 使能相应的中断
9. 设置输出极性 FLT.CCPAx FLT.CCPBx
10. 使能输出 DTR.MOE

11. 使能定时器 CR.CTEN

中心对齐互补 PWM 输出设置

1. 设置模式 CR.MODE=3
2. 设置互补输出 CR.COMP
3. 设置 PWM 周期值 ARR
4. 设置计数器初值（初值必须小于周期值）
5. 设置 PWM 比较值 CCRxA(CCRxB 不需要设置，PWM 输出与这个寄存器无关)
6. 设置 PWM 输出比较模式 FLT.OCMA FLT.OCMB 为 6 或 7
7. 清除相关中断标志
8. 使能相应的中断
9. 设置输出极性 FLT.CCPAx FLT.CCPBx
10. 使能输出 DTR.MOE
11. 使能定时器 CR.CTEN

三角波非中心对齐带死区互补 PWM 输出设置

1. 设置模式 CR.MODE=3
2. 设置互补输出 CR.COMP
3. 设置两个比较使能 CR.PWM2S
4. 设置 PWM 周期值 ARR
5. 设置计数器初值（初值必须小于周期值）
6. 设置 PWM 比较值 CCRxA,CCRxB,上计数比较点为 CCRxA,下计数比较点为 CCRxB
7. 设置 PWM 输出比较模式 FLT.OCMA FLT.OCMB 为 6 或 7
8. 清除相关中断标志
9. 使能相应的中断
10. 设置输出极性 FLT.CCPAx FLT.CCPBx
11. 设置死区使能 DTR.DTEN
12. 设置死区时间 DTR.DT
13. 使能输出 DTR.MOE
14. 使能定时器 CR.CTEN

捕获功能设置

设置 CH0B 为上升沿捕获功能

1. 选择计数器计数方式，设置 mode=2
2. 选择 CH0B 为捕获模式 CRCHx.CSB=1
3. 根据需要选择输入滤波 FLT.FLTB0
4. 选择捕获的边沿 CRCHx.CRB=1
5. 设置 ARR 改变周期值
6. 启动定时器 CR.CTEN
7. 清除相关中断标志
8. 清除捕获标志，使能相应中断
9. 查询到捕获标志后，读取 CCR0B 获得捕获值

注：定时器未启动时如果有效的捕获边沿也会产生捕获标志及捕获动作。

互补 PWM 输出时使用 GATE 作为 PWM 输出功能

1. 设置为 PWM 互补输出模式，参考“中心对齐互补 PWM 输出设置”
2. 设置 CCR0B 设置 GATE PWM 比较值
3. 设置 FLT.OCMB0=6/7 设置 PWM 输出模式
4. 使能 PWM 输出 DTR.MOE

互补 PWM 输出时使用 GATE 下降沿作为捕获输入功能

1. 设置为 PWM 互补输出模式，参考“中心对齐互补 PWM 输出设置”
2. 设置 GATE 作为捕获输入 CR.CSG
3. 选择 GATE 捕获输入边沿设置 CR.CFG(GATE 作为捕获输入 B 通道滤波设置无效)
4. 启动定时器 CR.CTEN
5. 清除相关中断标志
6. 清除捕获标志，使能相应中断
7. 查询到捕获标志后，读取 CCR0B 获得捕获值

14.2.7 模式 2/3 从模式

定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式和触发模式。

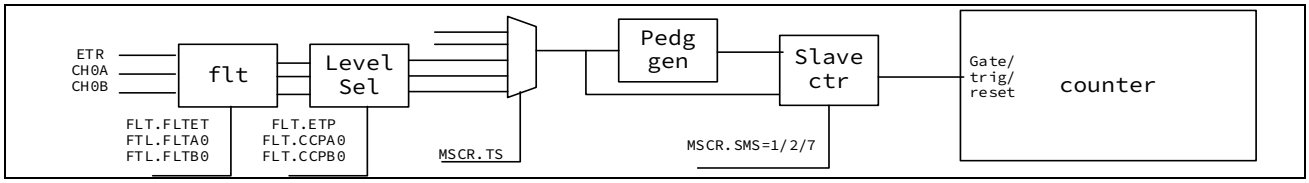


图 14-51 从模式示意图

从模式功能选择通过 MSCR.SMS 选择；

MSCR.SMS	从模式功能选择 001： 复位功能; 010： 触发模式; 111： 门控功能
MSCR.TS	触发选择 000：端口ETR的滤波相位选择后的信号ETFP，滤波功能可配置 001：内部互联信号 ITR0 010：内部互联信号 ITR1; 011：内部互联信号 ITR2; 100：内部互联信号 ITR3; 101：端口CH0A的边沿信号; 110：端口CH0A的滤波相位选择后的信号IAFP，滤波功能可配置 111：端口CH0B的滤波相位选择后的信号IBFP，滤波功能可配置

14.2.7.1 门控计数

按照选中的输入端电平使能计数器。在如下的例子中，计数器只在 CH0A 为低电平时向上计数：

- 配置通道 CH0A 以检测 CH0A 上的低电平。配置输入滤波器带宽(本例中，不需要滤波，所以保持 FLT.FLTA0=000)。置 FLTR 寄存器中 CCPA0=1 以确定极性(只检测低电平)。
- 置 SMCR 寄存器中 SMS=111，配置定时器为门控模式；置 SMCR 寄存器中 TS=110，选择 CH0A 作为输入源。
- 置 CR 寄存器中 CTEN=1，启动计数器。在门控模式下，如果 CTEN=0，则计数器不能启动，不论触发输入电平如何。只要 CH0A 为低，计数器开始依据内部时钟计数，一旦 CH0A 变高则停止计数。

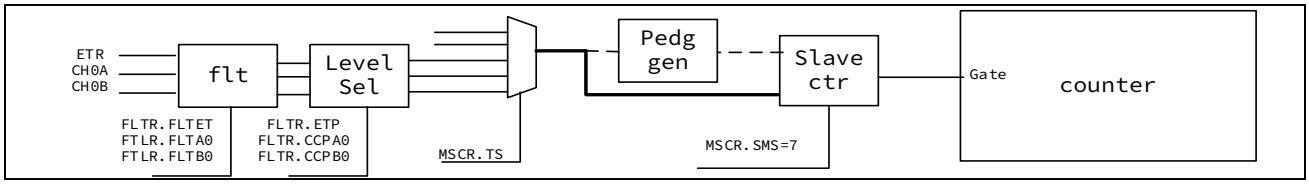


图 14-52 门控功能

14.2.7.2 触发功能

使用外部触发（CH0A、CH0B、ETR）可以使定时器同步启动，使用定时器内部互联信号结合 MSCR.MSM 也可以配置定时器同步启动。触发信号为输入信号的上升沿。也可以使用软件写 CR.TG 启动软件触发功能。

输入端上选中的事件使能计数器。在下面的例子中，计数器在 CH0B 输入的上升沿开始向上计数：

- 配置通道 CH0B 检测 CH0B 的上升沿。配置输入滤波器带宽(本例中，不需要任何滤波器，保持 FLT.FLTB0=000)。设 FLTR 寄存器中 CCPB0=0 以确定极性(不反向)。
- 置 SMCR 寄存器中 SMS=010，配置定时器为触发模式；置 SMCR 寄存器中 TS=111，选择 CH0B 作为输入源。当 CH0B 出现一个上升沿时，计数器开始在内部时钟驱动下计数，同时设置 TIF 标志。CH0B 上升沿和计数器启动计数之间的延时，取决于 CH0B 输入端的重同步电路。
  注意：如果使用下降沿触发，先选择触发极性，然后再选择模式，否则会产生误触发。

14.2.7.3 复位计数

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 CR 寄存器的 URS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器(ARR， CCRx)都被更新了。

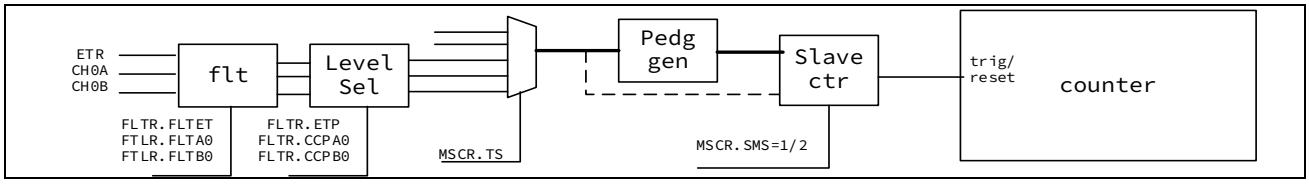


图 14-53 触发和复位功能

14.2.8 正交编码计数功能

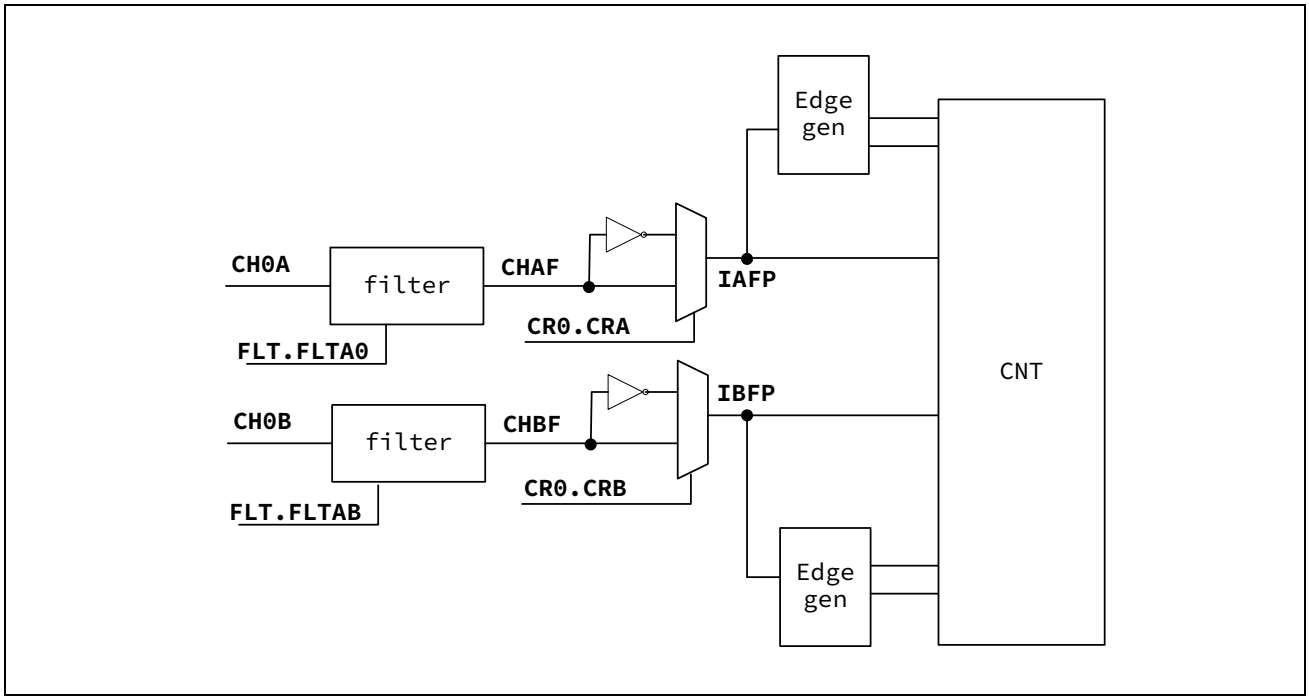
MSCR.SMS=4/5/6，对应正交编码模式的模式 1/2/3。这时计数器根据 IAFP, IBFP 的相位关系进行编码计数。IAFP,IBFP 为端口输入 CH0A，CH0B 的滤波相位选择的信号。

模式 1 使用 CH0A 的边沿计数。模式 2 使用 CH0B 的边沿计数。模式 3 使用 CH0A,CH0B 的边沿都计数。

为了保证计数相位的正确性，需要保证 A/B 输入的相位大于一个脉冲宽度的相位差，A/B 输入脉冲宽度需要大于两个脉冲宽度。

			IAFP		IBFP	
	IBFP	**IAFP**	**Rising**	**falling**	**Rising**	**falling**
MOD1	High		Down	Up	-	-
	Low		up	Down	-	-
MOD2		High	-	-	up	Down
		Low	-	-	Down	Up
MOD3	High	High	Down	Up	up	Down
	Low	Low	up	Down	Down	Up

CHAF/CHBF 为端口 CH0A/CH0B 滤波的信号，IAFP/IBFP 为端口滤波相位选择后的信号。



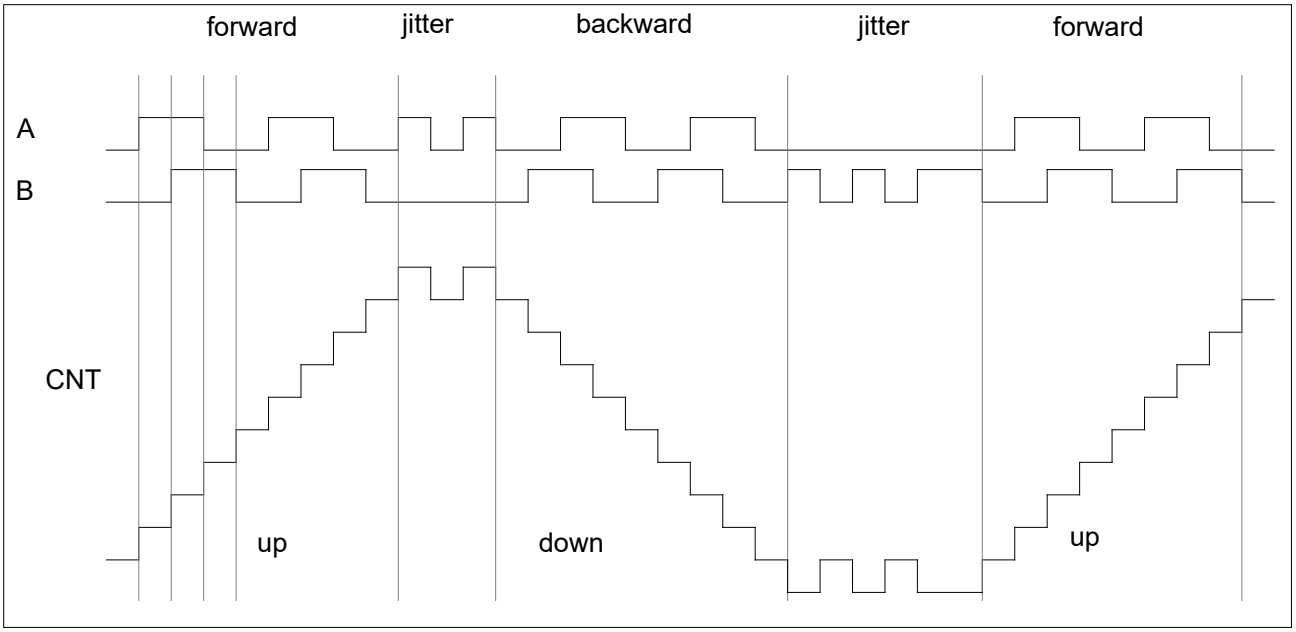


图 14-54　编码计数

14.2.9 Timer 触发 ADC

CCMR0A,CCMR1A,CCMR2A 比较匹配可以配置触发 ADC，中心对齐 PWM 时只能通过控制寄存器 CR.CIS 可以选择在上升匹配触发或者下降匹配触发。

CCMR0B,CCMR1B,CCMR2B 比较匹配可以配置触发ADC，中心对齐PWM时，可以通过寄存器CRx.CISB 分别控制三个匹配触发点（上升，下降，上升下降）。

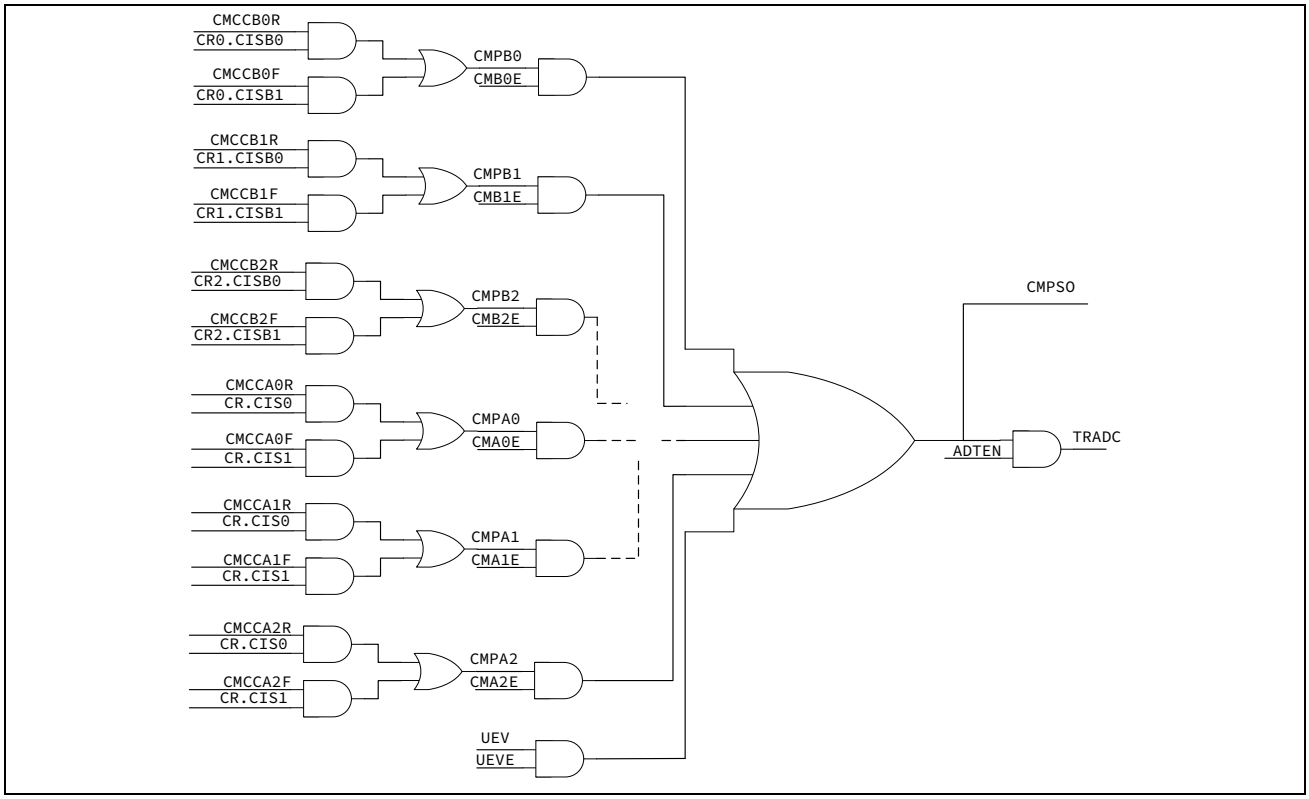


图 14-55 ADC 触发

互补 PWM 输出时 B 比较作为 ADC 触发功能

1. 设置为 PWM 互补输出模式，参考“中心对齐互补 PWM 输出设置”
2. 设置 CCRxB 设置 ADC 触发比较点
3. 设置 CRCHx.CISB 选择上计数下计数比较匹配（锯齿波不需要选择）
4. 选择 ADC 触发比较的源 ADTR
5. 使能 ADC 触发 ADTR.ADTE

14.2.10 刹车控制

VC 比较输出可以控制刹车功能，外部 BK 端口输入可控制刹车，系统 fail 可以控制刹车功能。通过 CR.BG 可以实现软件刹车功能，控制输出端口到设定的状态。

两通道的输出的 Tim0/1/2 可以选择使用 Tim0 的刹车端口控制 TIM1/2 的刹车功能，也可以使用各自的刹车输入控制各自的刹车功能。当 DTR.BKSEL 设置为 1 时，TIM1/2 共用 TIM0 的刹车输入。

14.2.11 Timer 互联

TRGO 输出信号可以连接到其他定时器的 ITR 信号。连接关系如下：

	ITR0	ITR1	ITR2	ITR3
Timer0	-	TIM1_TRGO	TIM2_TRGO	TIM3_TRGO
Timer1	TIM0_TRGO	-	TIM2_TRGO	TIM3_TRGO
Timer2	TIM0_TRGO	TIM1_TRGO	-	TIM3_TRGO
Timer3	TIM0_TRGO	TIM1_TRGO	TIM2_TRGO	-

定时器 TIM0/1/2 同时启动 PWM 输出设置（使用 TIM0 的 CTEN 触发 TIM1/2）示例

1. 参考 PWM 输出设置，设置定时器 0/1/2 的脉冲调整输出
2. 设置 TIM0 为主模式 MSCR.MSM=1
3. 设置 TIM0 的 CTEN 触发另外两个定时器 MSCR.MMS=1
4. 保持 TIM1/2 的 MSCR.MSM=0
5. 设置 TIM1/2 为触发模式 MSCR.SMS=2
6. 选择 TIM1/2 的触发源为 TIM0 的 TRGO, MSCR.TS=1
7. 最后使能定时器 TIM0，启动定时器 CR.CTEN=1

定时器 TIM0/1/2 同时启动 PWM 输出设置（使用 TIM1 的 UG 触发 TIM0/2）示例

1. 参考 PWM 输出设置，设置定时器 0/1/2 的脉冲调整输出
2. 设置 TIM1 为触发主模式 MSCR.MSM=1
3. 设置 TIM1 的 UG 触发另外两个定时器 MSCR.MMS=0
4. 保持 TIM0/2 的 MSCR.MSM=0
5. 设置 TIM0/2 为触发模式 MSCR.SMS=2
6. 选择 TIM0/2 的触发源为 TIM0 的 TRGO, MSCR.TS=2
7. 最后使能定时器 TIM1，启动定时器 CR.UG=1

14.2.12 GATE 输入互联

GATE 输入可以从端口通过 PX_SEL 选择直接输入，也可通过端口功能寄存器 GPIO_TIMGS 选择可以连通到其他模块或端口。

当 TIMx_G=0x0 时，门控 GATE 输入为 PX_SEL 选择的端口输入，当 TIMx_G=0x1~0x7 时，连接其他模块的输入或输出。

	TIM0_g	TIM1_g	TIM2_g	TIM3_g
000	PX_SEL	PX_SEL	PX_SEL	PX_SEL
001	UART0_RXD	LPUART0_RXD	UART0_RXD	UART0_RXD
010	UART1_RXD	LPUART1_RXD	UART1_RXD	UART1_RXD
011	VC0_OUT	VC0_OUT	VC0_OUT	LPUART0
100	VC1_OUT	VC1_OUT	VC1_OUT	LPUART1
101	PA03	PA08	PA10	VC0_OUT
110	PB08	PB03	PB04	PA06
111	PB15	PB13	PB11	PA11

14.2.13 ETR 输入互联

ETR 输入可以从端口直接输入，也可通过端口功能寄存器 GPIO_TIMES 选择可以连通到其他模块或端口。

当 TIMx_E=0x0 时，外部时钟 EXT 输入为 PX_SEL 选择的端口输入，当 TIMx_E=0x1~0x7 时，连接其他模块的输入或输出。

	TIM0_e	TIM1_e	TIM2_e	TIM3_e
000	PX_SEL	PX_SEL	PX_SEL	PX_SEL
001	LPUART0_RXD	UART0_RXD	LPUART0_RXD	UART0_RXD
010	LPUART1_RXD	UART1_RXD	LPUART1_RXD	UART1_RXD
011	VC0_OUT	VC1_OUT	VC0_OUT	VC1_OUT
100	LVD_OUT	LVD_OUT	Reserved	Reserved
101	PA00	PA01	PA04	PA00
110	PA05	PC09	PC04	PA12
111	PA15	PD02	PC08	PA13

14.2.14 CHx 捕获输入互联

Timer0/1/2 的 CHA，Timer3 的 CH0A/CH0B 输入可以从端口直接输入，也可通过端口功能寄存器 GPIO_TIMCPS 选择可以连通到其他模块或端口。

当 TIMx_CHy=0x0 时，捕获输入是 PX_SEL 选择的端口输入，当 TIMx_CHy=0x1~0x7 时，连接其他模块的输入或输出。

	TIM0_CHA	TIM1_CHA	TIM2_CHA	TIM3_CH0A	TIM3_CH0B
000	PX_SEL	PX_SEL	PX_SEL	PX_SEL	PX_SEL
001	UART0_RXD	UART1_RXD	LPUART0_RXD	LPUART1_RXD	UART0_RXD
010	PA00	PA00	VC0_OUT	LPUART0_RXD	UART1_RXD
011	PA02	PA02	PA02	Reserved	Reserved
100	PA05	PA06	PA07	VC0_OUT	VC1_OUT
101	PA15	PB08	PB08	PA08	PA07
110	PB06	PB10	PB09	PB03	PB04
111	PB14	PB13	PC06	PB06	PB13

14.2.15 DMA

Timer 支持软件和硬件触发 DMA 进行数据传输。支持数据从其他位置写入定时器，或从定时器读出写入其他位置。可应用于数据捕获后数据的自动搬运和更改周期值或占空比的脉冲宽度的自动调整。每个定时器有两个 DMA 请求，A 请求触发源可选择比较捕获 A，外部触发，B 请求触发源可选择比较捕获 B，事件更新。

IDREQ	Interrupt Signal of Peripheral
18	TIM0A
19	TIM0B
20	TIM1A
21	TIM1B
22	TIM2A
23	TIM2B
24	TIM3A
25	TIM3B

TIMxA 的触发源可选择比较捕获 A,ETRIG

TIMxB 的触发源可选择比较捕获 B,UEV

14.2.15.1 设置示例

捕获数据 DMA 数据传输

1. 使能 DMA
2. 使能 DMA 通道使能
3. 选择定时器 DMA 的通道
4. 设置传输类型、传输长度、传输方式
5. 设置源起始地址，目标起始地址
6. 设置源地址、目标地址的递增方式
7. 根据需要使能 DMA 中断
8. 参考数据捕获流程设置数据捕获
9. 使能 CRCHx.CDy 使能捕获数据触发 DAM 传输。

Note：源地址设置为捕获通道寄存器，源地址固定，目标地址加。

脉冲调整 DMA 数据传输

1. 使能 DMA
2. 使能 DMA 通道使能
3. 选择定时器 DMA 的通道
4. 设置传输类型、传输长度、传输方式
5. 设置源起始地址，目标起始地址
6. 设置源地址、目标地址的递增方式
7. 根据需要使能 DMA 中断
8. 参考脉冲调整输出设置脉冲调制输出
9. 根据需要选择硬件触发 DMA 的条件（UEV 触发或者比较匹配触发）

Note：目标地址设置为捕获通道寄存器，源地址变化，目标地址固定。

14.3Timer 寄存器描述

表 14-1 Timer 寄存器列表

Timer	基地址	描述
Timer0	0x40000C00	Timer0基地址
Timer1	0x40000D00	Timer1基地址
Timer2	0x40000E00	Timer2基地址
Timer3	0X40005800	Timer3基地址

寄存器	偏移地址	描述
TIMx_ARR	0X000	Timer重载寄存器/周期
TIMx_CNT	0X004	Timer 16位模式计数寄存器
TIMx_CNT32	0X008	Timer 32位模式计数寄存器 注：TIMx_CNT32是TIMx_ARR与TIMx_CNT的重映射寄存器。为了方便32位定时器写操作。
TIMx_M0CR	0X00C	Timer模式0控制寄存器（按不同模式进行描述）
TIMx_M1CR	0X00C	Timer模式1控制寄存器（按不同模式进行描述）
TIMx_M23CR	0X00C	Timer模式23控制寄存器（按不同模式进行描述）
TIMx_IFR	0X010	Timer中断标志
TIMx_ICLR	0X014	Timer中断清除寄存器
TIMx_MSCR	0X018	主从模式控制
TIMx_FLTR	0X01c	滤波控制
TIMx_ADTR	0X020	ADC触发控制
TIMx_CRCH0	0x024	比较单元0控制寄存器
TIMx_CRCH1	0X028	比较单元1控制寄存器
TIMx_CRCH2	0x02C	比较单元2控制寄存器
TIMx_DTR	0X030	死区寄存器
TIMx_RCR	0X034	重复计数寄存器
TIMx_ARRDM	0X038	Timer重载寄存器/周期映射地址
TIMx_CR0	0x024	控制寄存器
TIMx_CCR0A	0X03C	比较0A寄存器
TIMx_CCR0B	0x040	比较0B寄存器
TIMx_CCR1A	0X044	比较1A寄存器
TIMx_CCR1B	0X048	比较1B寄存器
TIMx_CCR2A	0X04C	比较2A寄存器
TIMx_CCR2B	0X050	比较2B寄存器

14.4模式 0 定时器寄存器描述

14.4.1 16 位模式重载寄存器（TIMx_ARR）

偏移地址：0x000

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	ARR	16位重载定时器重载值寄存器

14.4.2 16 位模式计数寄存器（TIMx_CNT）

偏移地址：0x004

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	16位重载定时器 计数值寄存器

14.4.3 32 位模式计数寄存器（TIMx_CNT32）

偏移地址：0x008

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT32[31:16]															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT32[15:0]															
RW															

位	符号	描述
31:0	CNT32	32位定时器 计数值寄存器 注：仅在模式0 32位定时器自由计数模式下有效，其他模式禁止写该寄存器

14.4.4 控制寄存器（TIMx_M0CR）

偏移地址：0x00C

复位值： 0x0060 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8
Resvered		Mode		Resvered	UIE	GATEP	GATE
		RW			RW	RW	RW

7	6	5	4	3	2	1	0
Resvered	PRS			TOGEN	CT	MD	CTEN
	RW			RW	RW	RW	RW

位	符号	描述
31:14	Reserved	保留位
13:12	MODE	工作模式 00 定时器模式0; 01 PWC模式 10 锯齿波模式; 11三角波模式
11	Reserved	保留位
10	UIE	中断使能控制，写1后使能中断
9	GATEP	端口 GATE 极性控制 0：端口GATE高电平有效 1：端口GATE低电平有效
8	GATE	定时器门控使能 0：无门控，CTEN=1 时定时器工作; 1：门控使能，端口 GATE 有效并且CTEN=1时定时器才工作;
7	Reserved	保留位
6:4	PRS	内部时钟分频选择 000：1; 001：2; 010：4; 011：8; 100：16; 101：32; 110：64; 111：256;
3	TOGEN	模式0下翻转输出使能 1：翻转输出使能 0：翻转输出关闭CHA,CHB输出为低电平
2	CT	计数时钟选择 0： 内部计数时钟TCLK 1： 外部计数时钟ETR;

1	MD	模式选择32定时/16定时模式选择 0： 32位自由计数 1： 16位重载计数
0	CTEN	定时器使能 0： 定时器停止； 1： 定时器使能

14.4.5 中断标志寄存器（TIMx_IFR）

偏移地址：0x010

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															UIF
															RO

位	符号	描述
31:1	Res.	保留位
0	UIF	溢出中断

14.4.6 中断标志清除寄存器（TIMx_ICLR）

偏移地址：0x014

复位值： 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															UIF
															R1W0

位	符号	描述
31:1	REV	保留位
0	UIF	溢出中断清除，写0清除

14.4.7 死区时间寄存器（TIMx_DTR）

偏移地址：0x030

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			MOE	Resvered											
			RW												

位	符号	描述
31:13	Reserved	保留位
12	MOE	翻转输出使能 0：翻转输出为输入状态 1：翻转端口为输出状态
11:0	Reserved	保留位

14.5脉冲宽度测量 PWC 寄存器描述

14.5.1 16 位模式计数寄存器（TIMx_CNT）

偏移地址：0x004

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	计数值寄存器

14.5.2 控制寄存器（TIMx_M1CR）

偏移地址：0x00C

复位值： 0x0060 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8
Resvered	Oneshot	Mode		Resvered	UIE	Edg2nd	Edg1st
	RW	RW			RW	RW	RW

7	6	5	4	3	2	1	0
Resvered	PRS			Resvered	CT	Resvered	CTEN
	RW				RW		RW

位	符号	描述
31:15	Reserved	保留位
14	Oneshot	单次触发模式选择 1：完成一次脉冲测量自动结束，再次测量需要重新使能CTEN 0：循环测量
13:12	MODE	工作模式 00 定时器模式0; 01 PWC模式 10 锯齿波模式; 11三角波模式
11	Reserved	保留位
10	UIE	中断使能控制，写1后使能中断 计数到0XFFFF才会溢出并产生溢出标志
9	Edg2nd	脉冲宽度测量结束边沿选择
8	Edg1st	脉冲宽度测量开始边沿选择 edg2nd Edg1st: 00 上沿-上沿周期; 01 低电平宽度; 10 高电平宽度; 11 下沿-下沿周期 测量
7	Reserved	保留位
6:4	PRS	内部时钟分频选择 000：1; 001：2; 010：4; 011：8; 100：16; 101：32; 110：64; 111：256;
3	Reserved	保留位
2	CT	计数时钟选择 0： 内部计数时钟TCLK 1： 外部计数时钟ETR;
1	Reserved	保留位
0	CTEN	脉冲宽度测量使能

		0：禁止； 1：使能

14.5.3 中断标志寄存器（TIMx_IFR）

偏移地址：0x010

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CA0F	Res.	UIF
													RO		RO

位	符号	描述
31:3	Res	保留位
2	CA0F	脉冲宽度测量中断标志
1	Res.	保留
0	UIF	溢出中断标志

14.5.4 中断标志清除寄存器（TIMx_ICLR）

偏移地址：0x014

复位值： 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CA0F	Res.	UIF
													R1W0		R1W0

位	符号	描述
31:3	Res.	保留位
2	CA0F	脉冲宽度测量中断标志清除，写0清除
1	Res.	保留
0	UIF	溢出中断清除，写0清除

14.5.5 主从模式控制寄存器（TIMx_MSCR）

偏移地址：0x018

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resverved			IB0S	IA0S	Resvered			TS			Resvered				
			RW	RW				RW							

位	符号	描述
31:13	Res.	保留位
12	IB0S	CH0B 输入选择 0： CH0B; 1：内部触发TS选择信号； 注：当PWM互补输出时自动选择GATE端口作为CH0B的输入
11	IA0S	IA0 输入选择 0： CH0A; 1： CH0A CH1A CH2A XOR(TIM3) 0： CH0A; 1： CH0A ETR GATE XOR(TIM0) 注：设置为1后端口的任意一个端口变化将导致输入变化
10:8	Res.	保留位
7:5	TS	触发选择 000：端口ETR的滤波相位选择后的信号ETFP ； 001：内部互联信号 ITR0 010：内部互联信号 ITR1; 011：内部互联信号 ITR2; 100：内部互联信号 ITR3; 101：端口CH0A的边沿信号; 110：端口CH0A的滤波相位选择后的信号IAFP 111：端口CH0B的滤波相位选择后的信号IBFP;
4:0	Res.	保留位

14.5.6 输出控制滤波（TIMx_FLTR）

偏移地址：0x01C

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETP	FLTET			Resvered											
RW	RW														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resvered									FLTB0			Resvered	FLTA0		
									RW				RW		

位	符号	描述
31	ETP	ETR 输入相位选择 0：同相位; 1：反向输入；
30:28	FLTET	ETR滤波控制 滤波设置 0xx：滤波无效 100：pclk 3个连续有效; 101：pclk/4 3个连续有效 110：pclk/16 3个连续有效; 111：pclk/64 3个连续有效
27:7	Resvered	保留位
6:4	FLTB0	CHB 输入滤波控制; 滤波设置 0xx：滤波无效 100：pclk 3个连续有效; 101：pclk/4 3个连续有效 110：pclk/16 3个连续有效; 111：pclk/64 3个连续有效
3	Resvered	保留位
2:0	FLTA0	CHA 输入滤波控制 滤波设置 0xx：滤波无效 100：pclk 3个连续有效; 101：pclk/4 3个连续有效 110：pclk/16 3个连续有效; 111：pclk/64 3个连续有效

14.5.7 控制寄存器（TIMx_CR0）

偏移地址：0x024

复位值：0x0000 3000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							CIEA	Reserved		CSB	CSA	Reserved			
							RW			RW	RW				

位	符号	描述
31:9	Resvered	保留位
8	CIEA	CIEA 脉冲宽度测量完成中断使能 0：禁止 1：使能
7:6	Resvered	保留位
5	CSB	B通道捕获/比较功能选择，当脉冲宽度测量使用通道B滤波时，需要设置CSB为1 0：比较模式 1：捕获模式
4	CSA	A通道捕获/比较功能选择，当脉冲宽度测量使用通道A滤波时，需要设置CSA为1 0：比较模式 1：捕获模式
3:0	Resvered	保留位

14.5.8 比较捕获寄存（TIMx_CCR0A ）

偏移地址：0x03C

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR0A															
R															

位	符号	描述
31:16	Reserved	保留位
15:0	CCR0A	脉冲宽度测量结果

14.6模式 2,3 寄存器描述

14.6.1 16 位模式重载寄存器（TIMx_ARR）

偏移地址：0x000(0X038 dummy address)

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	ARR	重载寄存器/周期寄存器，具有缓存功能 在计数器未使能或者缓存没有使能时，缓存寄存器可以立刻更新

14.6.2 16 位模式计数寄存器（TIMx_CNT）

偏移地址：0x004

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	重载定时器 计数值寄存器

注：使用 PWM 比较输出时，初始化 CNT 值需要小于 ARR 的值。

14.6.3 控制寄存器（TIMx_M23CR）

偏移地址：0x00C

复位值： 0x0060 0008

31	30	29	28	27	26	25	24
Resvered				DIR	BG	UG	TG
				RW/RO	W1	W1	W1

23	22	21	20	19	18	17	16
OCCE	CIS		BIE	TIE	TDE	URS	OCCS
RW	RW		RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8
CSG	Oneshot	Mode		UDE	UIE	CFG	CRG
RW	RW	RW		RW	RW	RW	RW

7	6	5	4	3	2	1	0
BUFPEN	PRS			Pwm2s	CT	Comp	CTEN
RW	RW			RW	RW	RW	RW

位	符号	描述
31:28	Reserved	保留位
27	DIR	计数方向，只有在锯齿波模式下可以写。其他模式下只读，写无效 0：上计数 1：下计数 注：从其他模式切换到中心对齐模式DIR自动清0。软件事件更新和从模式外部触发复位模式DIR自动清零。
26	BG	软件刹车，自动清零 写1产生软件刹车； 写0无效
25	UG	软件更新，自动清零 写1产生软件更新； 写0无效 初始化计数器并更新缓存寄存器到相应寄存器（缓存使能），预除频计数器也会被清零。
24	TG	软件触发，自动清零,需要在触发模式SMS=2且mode=2/3下都可以触发。 写1产生软件触发； 写0无效
23	OCCE	OCREF 清除使能 1：OCREF_CLR信号可以清除OCREF输出 0：OCREF输出不受OCREF_CLR影响

22:21	CIS	中心对齐A比较中断模式(B比较中断单独控制在CRx寄存器中CISB) 00：无中断， 01：上沿中断， 10：下沿中断， 11：上下沿都中断
20	BIE	刹车中断使能 1：中断使能 0：中断禁止
19	TIE	触发中断使能 1：中断使能 0：中断禁止
18	TDE	触发DMA使能 1：中断使能 0：中断禁止
17	URS	更新源 0：上溢出/下溢出/软件更新UG/从模式复位; 1：上溢出/下溢出
16	OCCS	OCREF 清除源选择 0：电压比较器VC输出，VC选择在VCx_OUTCFG寄存器设置 1：ETR端口滤波相位选择后的信号 当OCCE有效时，OC_clr可以清除OCREF的比较输出信号为零，（当OCMx>1时有效），下一个uev事件后继续比较输出
15	CSG	GATE 在PWM互补模式下捕获/比较选择; （只有在PWM互补输出时有效） 使用CCR0B作为GATE的比较或捕获通道 1：捕获; 0：比较
14	Oneshot	单次触发模式选择 1：发生事件更新后定时器停止。 0：循环计数
13:12	MODE	工作模式 00 定时器模式0; 01 PWC模式 10 锯齿波模式; 11三角波模式
11	UDE	更新DMA使能 1：使能更新触发DMA 0：禁止更新触发DMA
10	UIE	UIE 更新中断使能 1：使能更新中断 0：禁止更新中断
9	CFG	GATE作为捕获输入时，下沿捕获有效控制（只有在PWM互补输出时有效）

		1：下沿捕获有效 0：下沿捕获无效
8	CRG	GATE作为捕获输入时，上沿捕获有效控制（只有在PWM互补输出时有效） 1：上沿捕获有效 0：上沿捕获无效
7	BUFPEN	重载缓存使能 1：周期缓存使能，写入后下个事件更新后才影响到周期值。 0：周期缓存无效，写入后立刻影响周期值
6:4	PRS	内部时钟分频选择 000：1；　001：2；　010：4；　011：8； 100：16；　101：32；　110：64；　111：256；
3	PWM2S	OCREFA双点比较选择(缺省值为1) 0：双点比较使能，使用CCRA,CCRB比较控制OCREFA输出 1：单点比较使能，只使用CCRA比较控制OCREFA输出 注：OCREFB不受影响，仍然使用CCRB控制OCREFB输出
2	CT	计数时钟选择 0：　内部计数时钟TCLK 1：　外部计数时钟ETR;
1	Comp	PWM互补输出模式选择 0：独立PWM输出 1：互补PWM输出
0	CTEN	定时器使能 0：禁止； 1：使能 可以外部触发使能，在oneshot模式下结束时该位自动清零

14.6.4 中断标志寄存器（TIMx_IFR）

偏移地址：0x010

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIF	BIF	CB2E	CB1E	CB0E	CA2E	CA1A	CA0E	CB2F	CB1F	CB0F	CA2F	CA1F	CA0F	Res.	UIF
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO		RO

位	符号	描述
31:16	Reserved	保留位
15	TIF	触发中断标志
14	BIF	刹车中断标志
13	CB2E	通道CH2B捕获数据丢失标志 0：无数据丢失；1：数据丢失（仅TIM3存在）
12	CB1E	通道CH1B捕获数据丢失标志 0：无数据丢失；1：数据丢失（仅TIM3存在）
11	CB0E	通道CH0B捕获数据丢失标志 0：无数据丢失；1：数据丢失
10	CA2E	通道CH2A捕获数据丢失标志 0：无数据丢失；1：数据丢失（仅TIM3存在）
9	CA1E	通道CH1A捕获数据丢失标志 0：无数据丢失；1：数据丢失（仅TIM3存在）
8	CA0E	通道CH0A捕获数据丢失标志 0：无数据丢失；1：数据丢失
7	CB2F	通道CH2B发生捕获/比较匹配标志 0：没有发生 1：发生（仅TIM3存在）
6	CB1F	通道CH1B发生捕获/比较匹配标志 0：没有发生 1：发生（仅TIM3存在）
5	CB0F	通道CH0B发生捕获/比较匹配标志 0：没有发生 1：发生
4	CA2F	通道CH2A发生捕获/比较匹配标志 0：没有发生 1：发生（仅TIM3存在）
3	CA1F	通道CH1A发生捕获/比较匹配标志 0：没有发生 1：发生（仅TIM3存在）
2	CA0F	通道CH0A发生捕获/比较匹配标志 0：没有发生 1：发生
1	Res.	保留位

0	UIF	事件更新中断标志 0：没有发生　1：发生

14.6.5 中断标志清除寄存器（TIMx_ICLR）

偏移地址：0x014

复位值： 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIF	BIF	CB2E	CB1E	CB0E	CA2E	CA1A	CA0E	CB2F	CB1F	CB0F	CA2F	CA1F	CA0F	Res.	UIF
R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0		R1W0

位	符号	描述
31:16	REV	保留位
15	TIF	触发中断标志清除，写0清除
14	BIF	刹车中断标志清除，写0清除
13	CB2E	通道CH2B捕获数据丢失标志清除，写0清除（仅TIM3存在）
12	CB1E	通道CH1B捕获数据丢失标志清除，写0清除（仅TIM3存在）
11	CB0E	通道CH0B捕获数据丢失标志清除，写0清除
10	CA2E	通道CH2A捕获数据丢失标志清除，写0清除（仅TIM3存在）
9	CA1E	通道CH1A捕获数据丢失标志清除，写0清除（仅TIM3存在）
8	CA0E	通道CH0A捕获数据丢失标志清除，写0清除
7	CB2F	通道CH2B捕获/比较匹配标志清除，写0清除（仅TIM3存在）
6	CB1F	通道CH1B捕获/比较匹配标志清除，写0清除（仅TIM3存在）
5	CB0F	通道CH0B捕获/比较匹配标志清除，写0清除
4	CA2F	通道CH2A捕获/比较匹配标志清除，写0清除（仅TIM3存在）
3	CA1F	通道CH1A捕获/比较匹配标志清除，写0清除（仅TIM3存在）
2	CA0F	通道CH0A捕获/比较匹配标志清除，写0清除
1	Res.	保留
0	UIF	事件更新中断清除，写0清除

14.6.6 主从模式控制寄存器（TIMx_MSCR）

偏移地址：0x018

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resverved			IB0S	IA0S	SMS			TS			MSM	CCDS	MMS		
			RW	RW	RW			RW			RW	RW	RW		

位	符号	描述
31:13	Res	保留位
12	IB0S	CH0B 输入选择 0：CH0B; 1：内部触发TS选择信号； 注：当PWM互补输出时自动选择GATE端口作为CH0B的输入
11	IA0S	IA0 输入选择 0： CH0A; 1： CH0A CH1A CH2A XOR(TIM3) 0： CH0A; 1： CH0A ETR GATE XOR(TIM0) 注：设置为1后端口的任意一个端口变化将导致输入变化
10:8	SMS	从模式功能选择 000： 使用内部时钟; 001： 复位功能; 010： 触发模式; 011： 外部时钟模式 100： 正交编码计数模式1 ; 101： 正交编码计数模式2; 110： 正交编码计数模式3; 111： 门控功能
7:5	TS	触发选择 000：端口ETR的滤波相位选择后的信号ETFP ； 001：内部互联信号 ITR0 010：内部互联信号 ITR1; 011：内部互联信号 ITR2; 100：内部互联信号 ITR3; 101：端口CH0A的边沿信号; 110：端口CH0A的滤波相位选择后的信号IAFP 111：端口CH0B的滤波相位选择后的信号IBFP;
4	MSM	主从选择 0：无延时

<table>
<tr><td></td><td></td><td>1：延时使能，使主送计数器同时启动.
注：使用触发模式时，从模式设置为0，主模式设置为1，可以使主从计数同时启动</td></tr>
<tr><td>3</td><td>CCDS</td><td>比较模式下 DMA 比较触发选择；
0 ：比较匹配触发DMA;
1： 比较匹配不触发DMA，事件更新代替比较匹配触发DMA</td></tr>
<tr><td>2:0</td><td>MMS</td><td>主模式输出选择，用于内部互联，连接到其他定时器的ITRx
<table>
<tr><td>定时器0/1/2</td><td>定时器3</td></tr>
<tr><td>000：软件更新UG，写CR.UG
001：定时器使能CTEN
010：定时器事件更新UEV;
011：比较匹配选择输出CMPSO;
100：定时器比较参数输出OCREF0A
101：定时器比较参数输出OCREF0B
110：定时器比较参数输出OCREF0B
111：定时器比较参数输出OCREF0B</td><td>000：软件更新UG，写CR.UG
001：定时器使能CTEN
010：定时器事件更新UEV;
011：比较匹配选择输出CMPSO;
100：定时器比较参数输出OCREF0A
101：定时器比较参数输出OCREF1A
110：定时器比较参数输出OCREF2A
111：定时器比较参数输出OCREF0B</td></tr>
</table></td></tr>
</table>

14.6.7 输出控制/输入滤波（TIMx_FLTR）

偏移地址：0x01C

复位值：0x00000000

31	30 29 28	27	26 25 24	23	22 21 20	19	18 17 16
ETP	FLTET	BKP	FLTBK	CCPB2	OCMB2	CCPA2	OCMA2
-	-	-	-	-	FLTB2	-	FLTA2
RW	RW	RW	RW	RW	RW	RW	RW

15	14 13 12	11	10 9 8	7	6 5 4	3	2 1 0
CCPB1	OCMB1	CCPA1	OCMA1	CCPB0	OCMB0	CCPA0	OCMA0
-	FLTB1	-	FLTA1	CCPB0	FLTB0	CCPA0	FLTA0
RW	RW	RW	RW	RW	RW	RW	RW

位	符号	描述
31	ETP	ETR 输入相位选择 0：同相位; 1：反向输入；
30:28	FLTET	ETR滤波控制 滤波设置 0xx：滤波无效 100：pclk 3个连续有效; 101：pclk/4　3个连续有效 110：pclk/16　3个连续有效; 111：pclk/64　3个连续有效
27	BKP	刹车BK输入相位选择 0：同相位; 1：反向输入；
26:24	FLTBK	刹车输入滤波控制 滤波设置 0xx：滤波无效 100：pclk 3个连续有效; 101：pclk/4　3个连续有效 110：pclk/16　3个连续有效; 111：pclk/64　3个连续有效 注：为了保证OCMB0的PWM输出设置，GATE 在PWM互补模式下作为捕获输入，滤波设置无效。
23	CCPB2	比较功能：CH2B通道比较输出相位控制 0：正常输出; 1：反向输出

22:20	OCMB2 FLTB2	比较功能：CH2B通道比较控制;参考OCMB0 捕获功能：CH2B输入通道滤波设置，参考FLTBK
19	CCPA2	比较功能：CH2A通道比较输出相位控制 0：正常输出; 1：反向输出
18:16	OCMA2 FLTA2	比较功能：CH2A通道比较控制;参考OCMB0 捕获功能：CH2A输入通道滤波设置，参考FLTBK
15	CCPB1	比较功能：CH1B通道比较输出相位控制 0：正常输出; 1：反向输出
14：12	OCMB1 FLTB1	比较功能：CH1B通道比较控制;参考OCMB0 捕获功能：CH1B输入通道滤波设置，参考FLTBK
11	CCPA1	比较功能：CH1A通道比较输出相位控制 0：正常输出; 1：反向输出
10:8	OCMA1 FLTA1	比较功能：CH1A通道比较控制;参考OCMB0 捕获功能：CH1A输入通道滤波设置，参考FLTBK
7	CCPB0	比较功能：输出比较模式 CCPBx比较输出CHBx端口极性控制 0：正常输出; 1：反向输出 编码计数与从模式门控功能：输入相位控制 CCPB0从模式门控，复位，外部触发，外部时钟使用CH0B端口输入极性控制 0：正常输入; 1：反向输入
6:4	OCMB0 FLTB0	比较功能：CH0B通道比较控制 000：强制为0 001：强制为1 010：比较匹配时强制为0 011：比较匹配时强制为1 100：比较匹配时翻转 101：比较匹配时输出一个计数周期的高电平 110：PWM 模式1 单点比较： 上计数时CNT<CCRxy输出高，下计数时CNT>CCRxy输出为低电平 双点比较： 1）锯齿波上计数 CCRxA<CNT≤CCRxB输出为低电平 2）锯齿波下计数 CCRxA<CNT≤CCRxB输出为高电平 3）三角波上计数CNT<CCRxA输出高，下计数CNT>CCRxB为低电平 111 PWM 模式2

		单点比较： 上计数时CNT<CCRxy输出低，下计数时CNT>CCRxy输出为高电平 双点比较： 1）锯齿波上计数 CCRxA≤CNT<CCRxB输出为高电平 2）锯齿波下计数 CCRxA≤CNT<CCRxB输出为低电平 3）三角波上计数CNT<CCRxA输出低，下计数CNT>CCRxB为高电平 捕获功能：CH0B输入通道滤波设置，参考FLTBK
3	CCPA0	比较功能：CCPAx比较输出CHAx端口极性控制 0：正常输出; 1：反向输出 编码计数与从模式门控功能：输入相位控制 CCPA0从模式门控，复位，外部触发，外部时钟使用CH0A端口输入极性控制 0：正常输入; 1：反向输入
2:0	OCMA0 FLTA0	比较功能：A通道比较控制；参考OCMB0 捕获功能：CH0A输入通道滤波设置，参考FLTBK

14.6.8 ADC 触发控制寄存器（TIMx_ADTR）

偏移地址：0x020

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ADTE	CMB2E	CMB1E	CMB0E	CMA2E	CMA1E	CMA0E	UEVE
								RW	RW	RW	RW	RW	RW	RW	RW

位	符号	描述
31:8	Resvered	保留位
7	ADTE	使能ADC触发全局控制 1：使能 0：禁止
6	CMB2E	通道2B比较匹配触发ADC使能，只有TIM3存在 1：使能 0：禁止
5	CMB1E	通道1B比较匹配触发ADC使能，只有TIM3存在 1：使能 0：禁止
4	CMB0E	通道0B比较匹配触发ADC使能 1：使能 0：禁止
3	CMA2E	通道2A比较匹配触发ADC使能，只有TIM3存在 1：使能 0：禁止
2	CMA1E	通道1A比较匹配触发ADC使能，只有TIM3存在 1：使能 0：禁止
1	CMA0E	通道0A比较匹配触发ADC使能 1：使能 0：禁止
0	UEVE	事件更新触发ADC使能 1：使能 0：禁止

14.6.9 通道 0 控制寄存器（TIMx_CRCH0）

偏移地址：0x024

复位值： 0x0000 3000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCGB	CCGA	CISB		CDEB	CDEA	CIEB	CIEA	-	-	CSB	CSA	CFB	CRB	CFA	CRA
CCGB	CCGA	CISB		CDEB	CDEA	CIEB	CIEA	BUFEB	BUFEA	CSB	CSA	bksb		bksa	
RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW		RW	

位	符号	描述
31:16	Resvered	保留位
15	CCGB	捕获比较B软件触发，硬件自动清零。 在比较模式下只产生中断； 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
14	CCGA	捕获比较A软件触发，硬件自动清零。 在比较模式下只产生中断； 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
13:12	CISB	B通道比较匹配设置 00 无匹配；01上升匹配；10 下降匹配；11双匹配
11	CDEB	B捕获比较触发DMA使能 0：禁止 1：使能
10	CDEA	A捕获比较触发DMA使能 0：禁止 1：使能
9	CIEB	B捕获比较触发中断使能 0：禁止 1：使能
8	CIEA	A捕获比较触发中断使能 0：禁止 1：使能
7	BUFEB	比较功能：B比较缓存使能控制 0：禁止 1：使能
6	BUFEA	比较功能：A比较缓存使能控制 0：禁止 1：使能

5	CSB	B通道捕获/比较功能选择 0：比较模式 1：捕获模式
4	CSA	A通道捕获/比较功能选择 0：比较模式 1：捕获模式
3	CFB	B通道下降沿捕获使能 0：禁止 1：使能
2	CRB	B通道上升沿捕获使能 0：禁止 1：使能
3:2	BKSB	B通道比较功能输出刹车电平控制 00：高阻输出 01：对输出无影响 10：强制输出低电平 11：强制输出高电平;
1	CFA	A通道下降沿捕获使能 0：禁止 1：使能
0	CRA	A通道上升沿捕获使能 0：禁止 1：使能
1:0	BKSA	A通道比较功能输出刹车电平控制 00：高阻输出 01：对输出无影响 10：强制输出低电平 11：强制输出高电平;

14.6.10 通道 1/2 控制寄存器（TIM3_CRCH1/2）（仅 TIM3 存在）

偏移地址：

TIM3_CRCH1：0x028；

TIM3_CRCH2：0x02C

复位值： 0x0000 3000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCGB	CCGA	CISB		CDEB	CDEA	CIEB	CIEA	-	-	CSB	CSA	CFB	CRB	CFA	CRA
CCGB	CCGA	CISB		CDEB	CDEA	CIEB	CIEA	BUFEB	BUFEA	CSB	CSA	bksb		bksa	
RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW		RW	

位	符号	描述
31:16	Resvered	保留位
15	CCGB	捕获比较B软件触发，硬件自动清零。 在比较模式下只产生中断； 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
14	CCGA	捕获比较A软件触发，硬件自动清零。 在比较模式下只产生中断； 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
13:12	CISB	B通道比较匹配设置 00 无匹配；01上升匹配；10 下降匹配；11双匹配
11	CDEB	B捕获比较触发DMA使能 0：禁止 1：使能
10	CDEA	A捕获比较触发DMA使能 0：禁止 1：使能
9	CIEB	B捕获比较触发中断使能 0：禁止 1：使能
8	CIEA	A捕获比较触发中断使能 0：禁止 1：使能
7	BUFEB	比较功能：B比较缓存使能控制 0：禁止 1：使能

6	BUFEA	比较功能：A比较缓存使能控制 0：禁止 1：使能
5	CSB	B通道捕获/比较功能选择 0：比较模式 1：捕获模式
4	CSA	A通道捕获/比较功能选择 0：比较模式 1：捕获模式
3	CFB	B通道下降沿捕获使能 0：禁止 1：使能
2	CRB	B通道上升沿捕获使能 0：禁止 1：使能
3:2	BKSB	B通道比较功能输出刹车电平控制 00：高阻输出 01：保持之前输出I 10：强制输出低电平 11：强制输出高电平;
1	CFA	A通道下降沿捕获使能 0：禁止 1：使能
0	CRA	A通道上升沿捕获使能 0：禁止 1：使能
1:0	BKSA	A通道比较功能输出刹车电平控制 00：高阻输出 01：保持之前输出I 10：强制输出低电平 11：强制输出高电平;

14.6.11 死区时间寄存器（TIMx_DTR）

偏移地址：0x030

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	VCE	Safeen	MOE	AOE	BKE	DTEN	Bksel	DTR							
	RW	RW	RW	RW	RW	RW	RW	RW							

位	符号	描述
31:15	Reserved	保留位
14	VCE	VC Brake enable 0：禁止 1：使能
13	Safeen	Safety 刹车使能（osc fail,brown down,lockup） 0：禁止 1：使能
12	MOE	PWM输出使能 0：禁止 1：使能
11	AOE	PWM输出自动使能 0：禁止 1：使能
10	BKE	刹车使能 0：禁止 1：使能
9	DTEN	死区控制使能 0：禁止 1：使能
8	bksel	刹车选择 0：使用自己的刹车控制； 1：TIM1/2使用 TIM0的的刹车控制 注：TIM0/TIM3选择无效。
7:0	DTR	死区时间寄存器 DTR[7] =0 T=DTR[6:0]+2 2-129 step=1 DTR[7:6] =10 T={DTR[5:0]+64}*2 +2 130-256 step=2 DTR[7:5] =110 T={DTR[4:0]+32}*8 +2 258-506 step=8 DTR[7:5] =111 T={DTR[4:0]+32}*16 +2 514-1010 step=16

14.6.12 重复周期设置值（TIMx_RCR）

偏移地址：0x034

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RCR							
								RW							

位	符号	描述
31:8	Reserved	保留位
7:0	RCR	重复周期计数值 设置RCR+1个周期个上溢出下溢出后产生事件更新，计数器上溢出或下溢出时内部RCR_CNT减1，当计数到零后RCR_CNT重载RCR的值，并且产生事件更新UEV信号

14.6.13 通道 0 比较捕获寄存器（TIMx_CCR0A/B）

偏移地址：

TIMx_CCR0A ： 0x03C；

TIMx_CCR0B ： 0x040

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR0y															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CCR0y	比较捕获寄存器，比较具有缓存功能(y=A,B)

14.6.14 通道 1/2 比较捕获寄存器（TIM3_CCR1/2 A/B）（仅 TIM3 存在）

偏移地址：

TIM3_CCR1A：0x044

TIM3_CCR1B：0x048

TIM3_CCR2A：0x04C

TIM3_CCR2B：0x050

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCRxy															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CCRxy	比较捕获寄存器，比较具有缓存功能(x=1,2；y=A,B)

15 可编程计数阵列（PCA）

15.1PCA 简介

PCA(可编程计数器阵列 Programmable Counter Array)支持最多 5 个 16 位的捕获/比较模块。该定时/计数器可用作为一个通用的时钟计数/事件计数器的捕获/比较功能。PCA 的每个模块都可以进行独立编程，以提供输入捕捉，输出比较或脉冲宽度调制。另外模块 4 有额外的看门狗定时器模式。

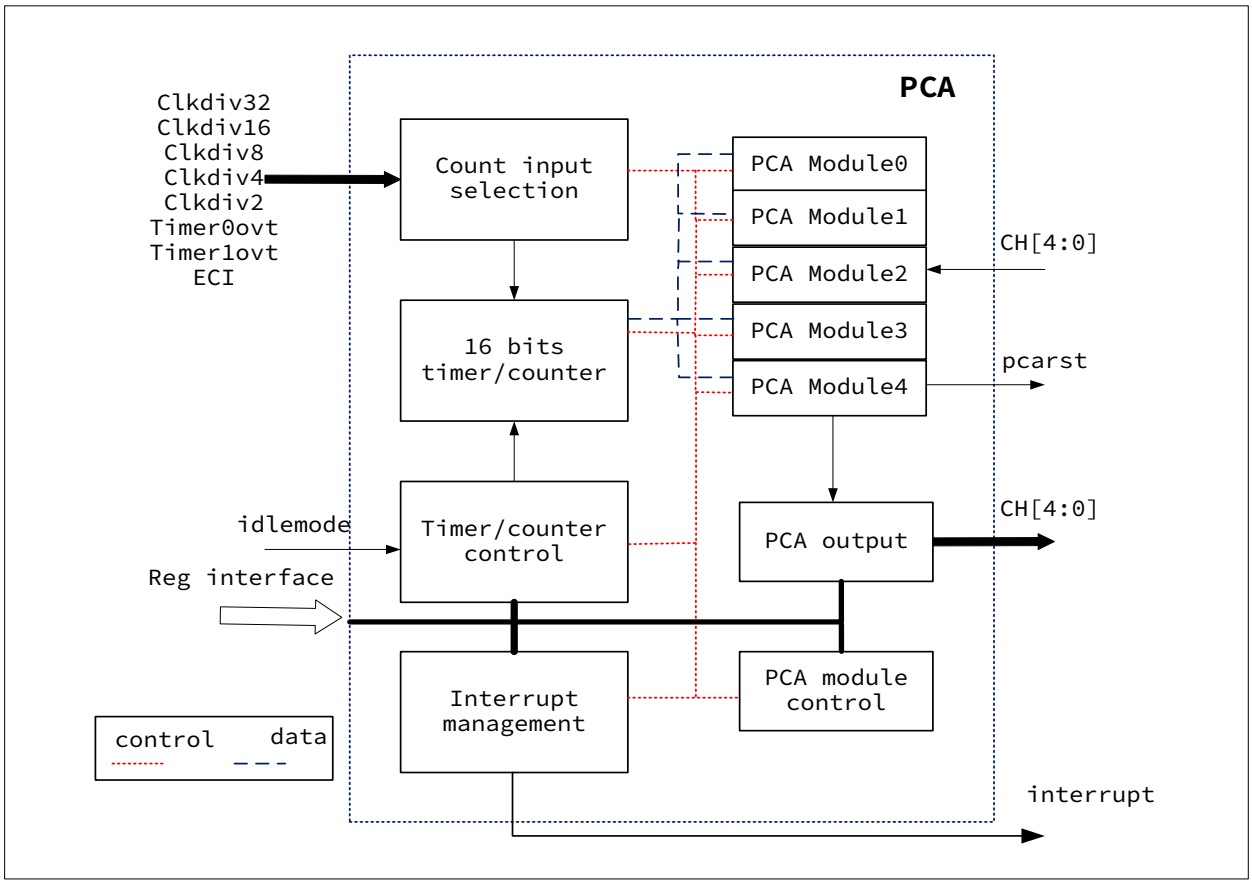


图 15-1 PCA 整体框图

15.2PCA 功能描述

五个模块都可被配置为独立工作，有三种工作方式：边沿触发捕捉、输出比较、8/16 位脉宽调制。每个模块在系统控制器中都有属于自己的功能寄存器，这些寄存器用于配置模块的工作方式和与模块交换数据。

每组比较/捕获模块是由一个比较/捕获寄存器组(CCAPx)，以及 1 个 16 位比较器，和各种逻辑门控制组成。寄存器组用来存储时间或次数，针对外部触发捕获条件，或内部触发比较条件。在 8 位 PWM 模式下,寄存器(CCAPxL)用来控制输出波形的占空比,CCAPxH 为 8 位比较缓存。16 位 PWM 模式下 CARR 用于控制 PWM 输出的周期，CCAPx 寄存器控制占空比。

每个模块都可以独立编程的操作在任何以下模式：

- 16 位捕获模式的上升沿，下降沿或任意沿触发。
- 比较模式：16 位高速输出，16 位看门狗定时器（模块 4）或 16/8 位脉冲宽度调制。
- 关闭状态。

比较/捕获模块模式寄存器（CCAPMx）确定相应的工作模式。对于比较/捕获模块进行编程时，他们是基于共同的时间计数。定时器/计数器打开和关闭通过 CCON.CR 位即可控制 PCA 定时/计数器的运行。在一个比较/捕获模块捕获，软件定时器，高速输出，设置模块的比较/捕获标志（CCON.CCFx），并产生 PCA 中断请求，如果相应的使能位在 CCAPMx 寄存器设置。CPU 可以在任何时候读写 CCAPx 寄存器。

15.2.1 PCA 定时/计数器

CNT 的这组特殊功能寄存器可用作为一个 16 位定时器/计数器。这是一个 16 位向上计数的计数器。如果 CMOD.CFIE 位被置“1”时，当 CNT 溢出时硬件自动设置 PCA 溢出标志（CCON.CF）并产生 PCA 中断请求。CMOD.CPS[2：0]三位选择八个信号输入到定时器/计数器。

- 系统时钟 PCLK 的 32 分频。
- 系统时钟 PCLK 的 16 分频。
- 系统时钟 PCLK 的 8 分频。
- 系统时钟 PCLK 的 4 分频。
- 系统时钟 PCLK 的 2 分频。
- 定时器 0 的溢出。每次定时器 0 计数溢出后，CNT 就递增，这样提供了 PCA 的可变编程频率输入。
- 定时器 1 的溢出。每次定时器 1 计数溢出后，CNT 就递增，这样提供了 PCA 的可变编程频率输入。
- ECI。CPU 每过 4 个 PCLK 时钟周期就对 PCA ECI 进行采样，当每次采样结果从高变低时，CL 自动加 1，因此最高的 ECI 输入频率不能高于系统时钟 PCLK 的 1/8,以满足采样需求。

设置运行控制器（CCON.CR）启动 PCA 定时/计数器。当 CMOD.CIDL 置“1”后，PCA 定时器/计数器可以继续运行在空闲模式下。CPU 可以随时读取 CNT 的数值，但当计数启动后（CCON.CR=1）时，为了防止计数错误，CNT 是禁止写入的。

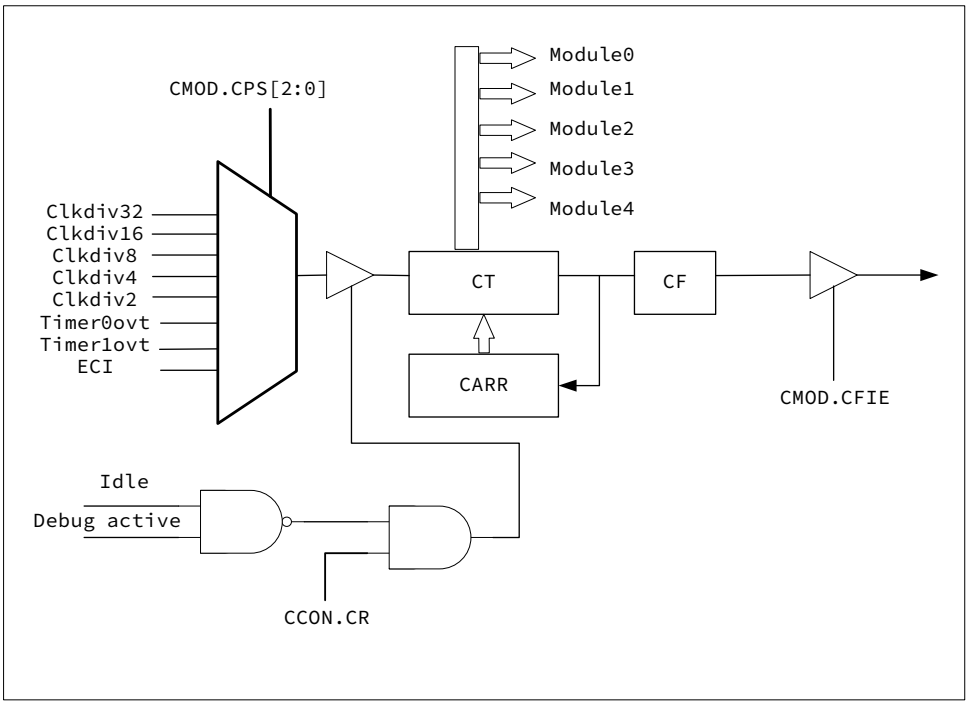


图 15-2 PCA 计数器框图

15.2.1.1 16 位自由计数模式

计数器计数到最大值 0XFFFF 溢出后计数器的值变为 0，重新开始向上计数，如果需要更改计数周期值可以停止 PCA 更改计数器的初值然后继续计数。可用于 PCA 捕获模式，8 位 PWM 模式。

设置流程

1. 保持 PCA_EPWM 的 EPMW 为 0
2. 设置 PCA_CMOD.CPS 选择计数时钟
3. 根据需要设置 PCA_CMOD.CFIE 设置计数溢出中断使能
4. 时钟 PCA_CCON.CR 启动 PCA 计数器
5. 溢出更改计数器初值需要停止 PCA 计数器

15.2.1.2 16 位重载计数模式

计数器计数到与寄存器 CARR 的值相同时溢出后计数器的值变为 0，继续开始向上计数，可用于 PCA 捕获模式，16 位 PWM 模式。

设置流程

1. 设置 PCA_EPWM 的 EPMW 为 1
2. 设置 PCA_CARR 设置计数周期值
3. 设置 PCA_CMOD.CPS 选择计数时钟
4. 根据需要设置 PCA_CMOD.CFIE 设置计数溢出中断使能

5． 时钟 PCA_CCON.CR 启动 PCA 计数器

15.2.2 PCA 捕获功能

PCA 捕获模式提供了 5 路 PCA 测量脉冲周期，脉冲宽度，占空比和相位差的功能。

引脚上出现的电平跳变导致 PCA 捕捉 PCA 计数器/定时器的值并将其装入到对应模块的 16 位捕捉/比较寄存器（CCAPx）。CCAPMx.CAPP 以及 CCAPMx.CAPN 位用于选择触发捕捉的电平变化类型：低电平到高电平（正沿）、高电平到低电平（负沿）或任何变化（正沿或负沿）。当捕捉发生时，CCON 中的捕捉/比较标志（CCFn）被置为逻辑‘1’并产生一个中断请求（如果 CCF 中断被允许）。当 CPU 转向中断服务程序时，CCFn 位不能被硬件自动清除，用户软件写 INTCL 寄存器清除此标志位。如果 CCPMx.CAPP 以及 CCAPMx.CAPN 位都被设置为逻辑‘1’，可以通过直接读对应端口引脚的状态来确定本次捕捉是由上升沿触发还是由下降沿触发。

分辨率等于定时器/计数器的时钟。输入信号必须在高电平或低电平期间至少保持 2 个时钟周期，以保证输入信号能够被硬件识别。

CPU 可以在任何时候读取或写入 CCAPx 的寄存器。

捕获设置：

- 当需要在外部上升沿进行捕获，CCPMx.CAPP =“1”以及 CCAPMx.CAPN = “0”
- 当需要在外部下降沿进行捕获，CCPMx.CAPP =“0”以及 CCAPMx.CAPN = “1”
- 当需要在外部上升、下降沿进行捕获，CCPMx.CAPP =“1”以及 CCAPMx.CAPN = “1”
- 根据需要配置捕获中断及中断处理程序。
- 按照定时/计数器配置 PCA 计数器启动。

注意：

- 随后由同一模块的捕获值会覆盖现有捕获的值。为了保持捕获的值，在中断服务程序中将它保存在 RAM 里面，这个操作必须在下一次事件出现之前完成，否则就会丢失前面一次捕获采样值。

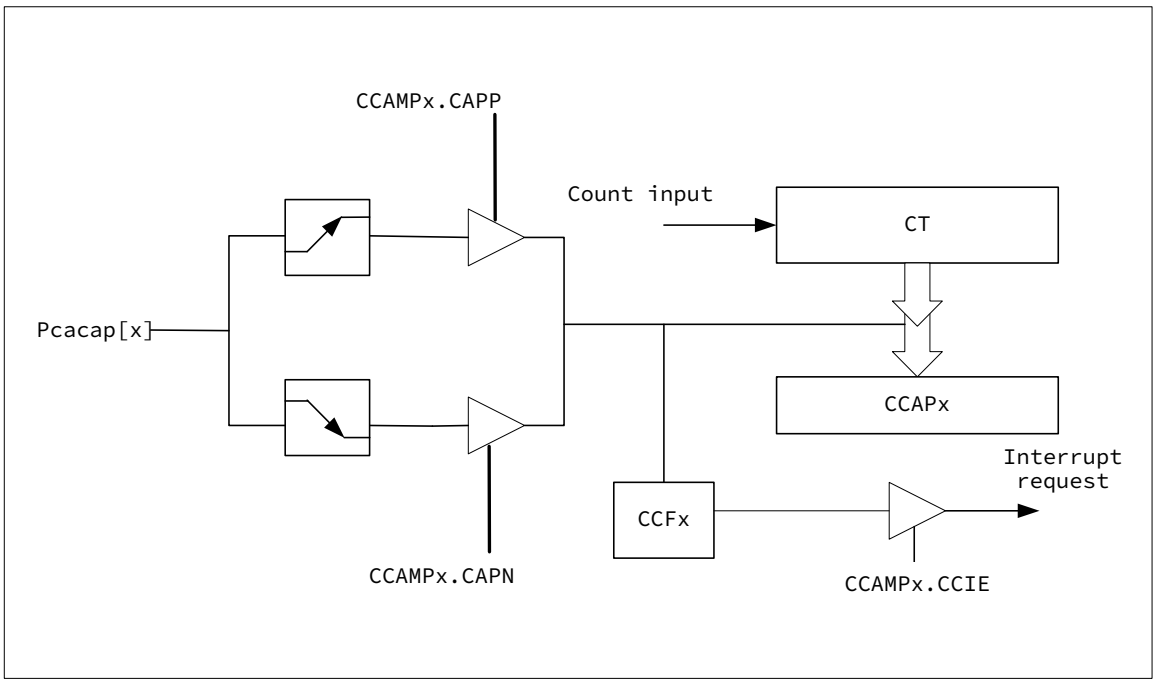


图 15-3 PCA 捕获功能框图

15.2.3 PCA 比较功能

比较功能提供如下功能，高速输出模式，WDT 模式，16 为 PWM 模式和 8 位 PWM 模式。在前三个功能中，比较/捕获模块比较 16 位 PCA 定时器/计数器的值与预先加载到该模块的 CCAPx 寄存器中的 16 位值。在 8 位 PWM 模式下，PCA 模块不断地将 PCA 定时器/计数器低字节寄存器（CNT）与一个在 CCAPxL 模块寄存器 8 位的值进行比较。每 4 个时钟周期比较一次，即与最快的 PCA 定时器/计数器的时钟速率相匹配。

设置 CCAPMx.ECOM 位选择该模块的比较功能。

若要正确使用在比较模式下的模块，请遵守以下的一般程序：

- 选择 PCA 模块的操作模式。
- 选择 PCA 定时器/计数器的输入信号。
- 比较值加载到模块的比较/捕获寄存器对。
- 设置 PCA 定时器/计数器运行控制位。
- 匹配后产生中断，清除模块的比较/捕获标志。

15.2.3.1 比较翻转输出模式

在比较翻转输出模式，每当 PCA 计数器内的值与模块的 16 位捕捉/比较寄存器（CCAPx）发生匹配时，模块 PCA 的 CH[x]引脚上的逻辑电平将发生变化。这可以提供比切换 IO 输出有更高精度，因为这个输出不会被中断响应而影响输出频率，靠 CPU 来切换 IO 输出的话，功耗，精度都有所欠缺。

要设定一个比较/捕获模块的比较翻转输出模式，设置 CCAPMx.ECOM，CCAPMx.MAT 和 CCAPMx.TOG 位。PCA 定时器/计数器和比较/捕获的寄存器（CCAPx）之间的匹配切换 PCA 的 CH[x]信号，并设置模块的比较/捕获标志（CCON.CCFx）。通过软件设置或清除 PCA 的 CH[x]信号，用户可以选择匹配切换信号从低到高或高到低。

用户也可以选择产生一个中断请求，通过设置相应的中断使能位（CCAPMx.CCIE）当匹配发生时，即可产生中断请求。由于硬件无法清除的比较/捕获标志中断，用户必须在软件中清除这个标志位。如果用户在中断程序中不去改变比较/捕获寄存器，PCA 并重新计数比较值，如相匹配则发生下一次翻转。在中断服务程序中，一个新的 16 位比较值可以被写入比较/捕获的寄存器（CCAPx）。

注意：

- 为了防止无效的匹配，而更新这些寄存器，用户软件应该写 CCAPxL 的首先然后 CCAPxH。写到 CCAPxL 清除禁用比较功能 ECOM 位，而写到 CCAPxH 设置的 ECOM 位，重新启用比较功能。

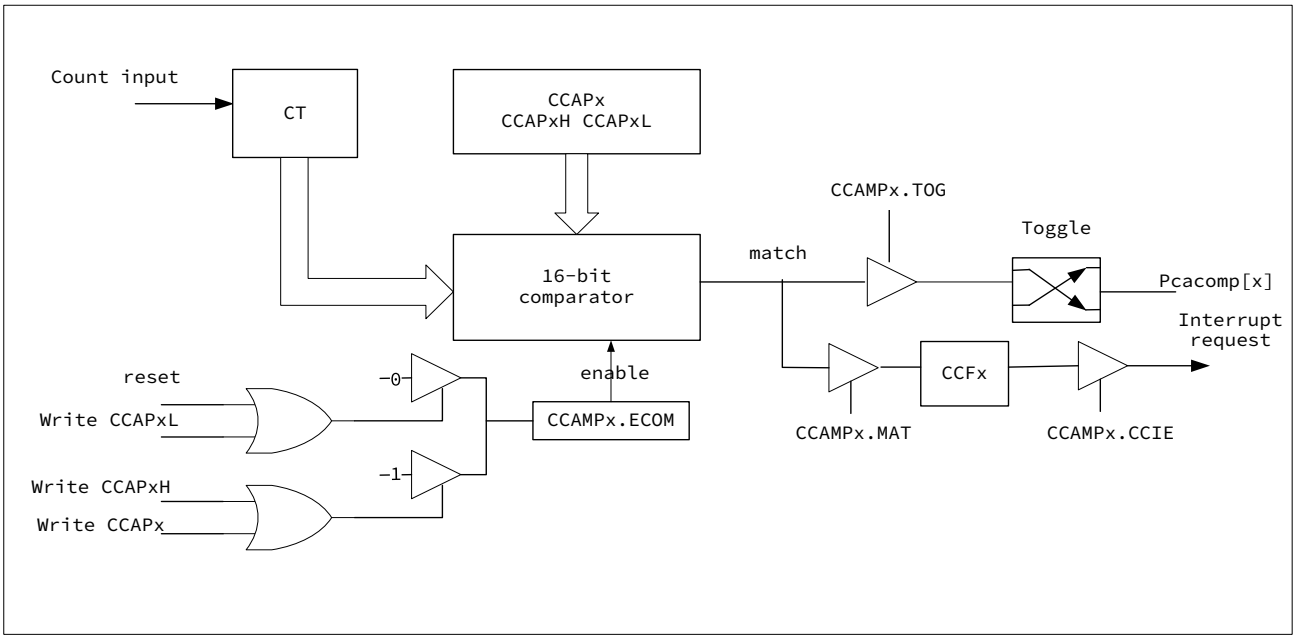


图 15-4 PCA 比较功能框图

15.2.3.2 PCA 16 位 PWM 功能

在比较输出 PWM 模式，每当 PCA 计数器内的值与模块的 16 位捕捉/比较寄存器（CCAPx）发生匹配时，模块 PCA 的 CH[x]引脚上的逻辑电平将发生变化。计数器溢出时，CH[x]引脚上的逻辑电平将清零。这可以提供一组 16 位的 PWM 输出。

要设定一个比较/捕获模块的 PWM 模式，设置 CCAPMx.ECOM，CCAPMx.MAT 和 CCAPMx.TOG 位及 EPWM 寄存器。 PCA 定时器/计数器和比较/捕获的寄存器（CCAPx）之间的匹配切换 PCA 的 CH[x]信号，并设置模块的比较/捕获标志（CCON.CCFx）。

用户也可以选择产生一个中断请求，通过设置相应的中断使能位（CCAPMx.CCIE）当匹配发生时，即可产生中断请求。由于硬件无法清除的比较/捕获标志中断，用户必须在软件中清除这个标志位。

注：使用 16 位 PWM 时建议不要使用 PCA 的分频，否则占空比输出最大会差别一个周期。

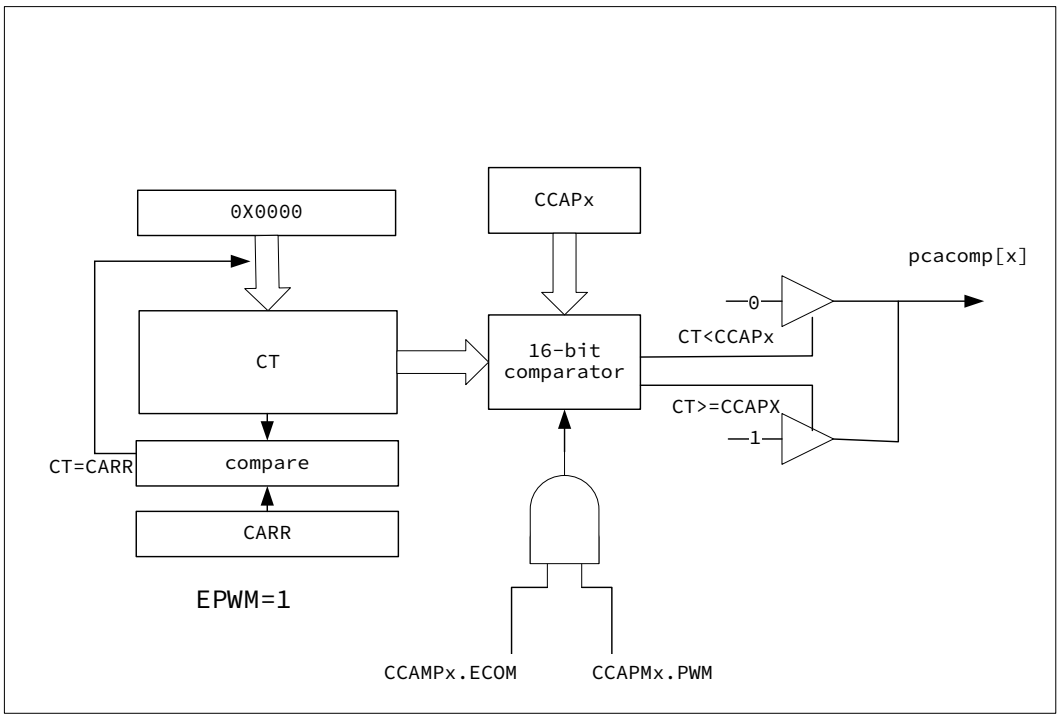


图 15-5 PCA 16 位 PWM 功能框图

15.2.3.3 PCA 模块 4 的 WDT 功能

本系列除了一个 WDT 硬件模块，PCA 的模块 4，还提供一个可编程频率的 16 位 WDT。当 PCA 定时器/计数器的计数值与模块 4 中存储的值比较/捕获寄存器（CCAP4）相匹配时，这种模式产生复位信号。PCA 的 WDT 复位信号做为一个独立的复位信号。与外部复位（RST），硬件看门狗复位（WDTRST）和 LVD 低电压复位，POR 上电下电复位相结合。用户可以自由结合或单独使用它们。模块 4 是具有 WDT 模式唯一 PCA 的模块。当不设置为 WDT 时，它可以在其它模式中独立使用。

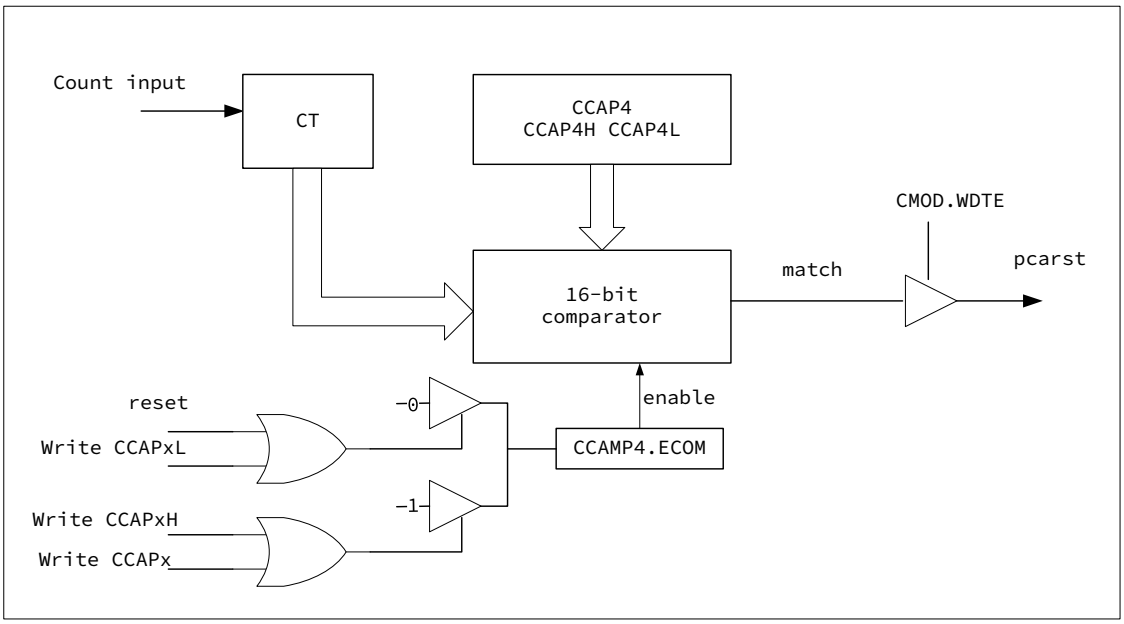


图 15-6 PCA WDT 功能框图

当把 PCA 模块 4 当 WDT 使用时，必须设置 CCAPM4.ECOM4，CCAPM4.MAT4 以及 CMOD.WDTE。另外 PCA 定时器/计数器可以设置 CMOD.CPS 来选择不同的输入计数频率。

在比较/捕获寄存器（CCAP4）输入一个 16 位的比较值。在 PCA 定时器/计数器（CNT），输入一个 16 位的初始值或使用复位值（0000h）。这些值乘以的 PCA 输入脉冲率之间的差额确定的 WDT 匹配运行时间。设置定时器/计数器运行控制位（CCON.CR）启动 PCA WDT。每次匹配时，PCA 的 WDT 产生复位信号。要防止一个 PCA WDT 复位，用户有三种选择。

- 定期的比较值 CCAP4 的改变，所以匹配永远不会发生。
- 定期更改 PCA 定时器/计数器值（CNT）所以匹配永远不会发生。
- 通过在匹配前清除的 CMOD.WDTE 位来禁用模块复位输出信号，后来重新启用它。

前两个选项是更可靠的，因为 WDT 在第三个选项没有被禁用。

第二个选项是不推荐，如果其他 PCA 模块都在使用，因为五个模块共享一个共同的时间基。因此，在大多数应用中的第一个选项是最好的。

PCA WDT 配置流程

1) 配置 WDT 比较/捕获寄存器 PCA_CCAP4
2) 配置 PCA 计数寄存器 PCA_CNT
3) 配置 PCA_CCAMP4 选择比较匹配功能
4) 配置 PCA_CMOD 选择输入时钟，使能 WDT 功能
5) 启动 PCA
6) 选择清除 PCA WDT 清除方式在 PCA WDT 复位前清除 PCA WDT

15.2.3.4 PCA 8 位 PWM 功能

脉宽调制是一种使用程序来控制波形占空比，周期，相位的技术。5 个 PCA 模块都可以被独立地用于在对应 PCA 的 CH[x]引脚产生脉宽调制（PWM）输出，脉冲宽度为 8 位分辨率。PWM 输出的频率取决于 PCA 计数器/定时器的时基。使用模块的捕捉/比较寄存器 CCAPxL 来改变 PWM 输出信号的占空比。当 PCA 计数器/定时器的低字节（CL）与 CCAPxL 中的值相等时，PCA 的 CH[x]引脚上的输出被置“1”；当 CL 中的计数值溢出时，PCA 的 CH[x]输出被复位“0”。当计数器/定时器的低字节 CL 溢出时（从 0xFF 到 0x00），保存在 CCAPxH 中的值被自动装入到 CCAPxL，不需软件干预。

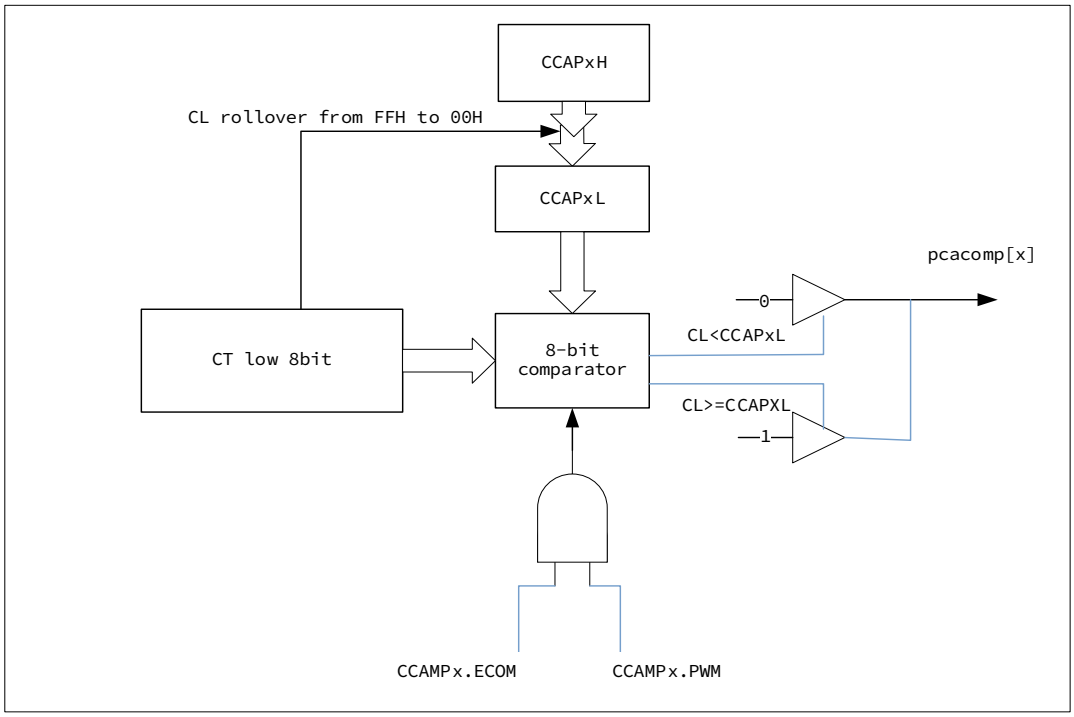


图 15-7 PCA PWM 功能框图

在这种模式下，PCA 定时器/计数器（CL）的低字节中的值是不断在低字节比较/捕获寄存器（CCAPxL）的值相比。当 CL < CCAPxL，输出波形为低。当两者匹配时（CL= CCAPxL），输出波形去到高，直到 CL 溢出从 FFH 到 00H，结束期间仍然很高。在溢出时，在 CCAPxH 的值自动装载到 CCAPxL 内，一个新的周期的开始。

在 CCAPxL 的值决定当前波形的占空比。在 CCAPxH 的值确定下一个波形的占空比。改变 CCAPxL 中的值即可更改的脉冲宽度调制。正如图所示，8 位值在 CCAPxL 可以从 0（100%占空比），到 255（0.4%占空比）。要改变 CCAPxL 值而不会产生毛刺，需要在高字节寄存器（CCAPxH）写入一个新值。当 CL 超过 FFH 滚动到 00h，这个值是由硬件自动加载到 CCAPxL。

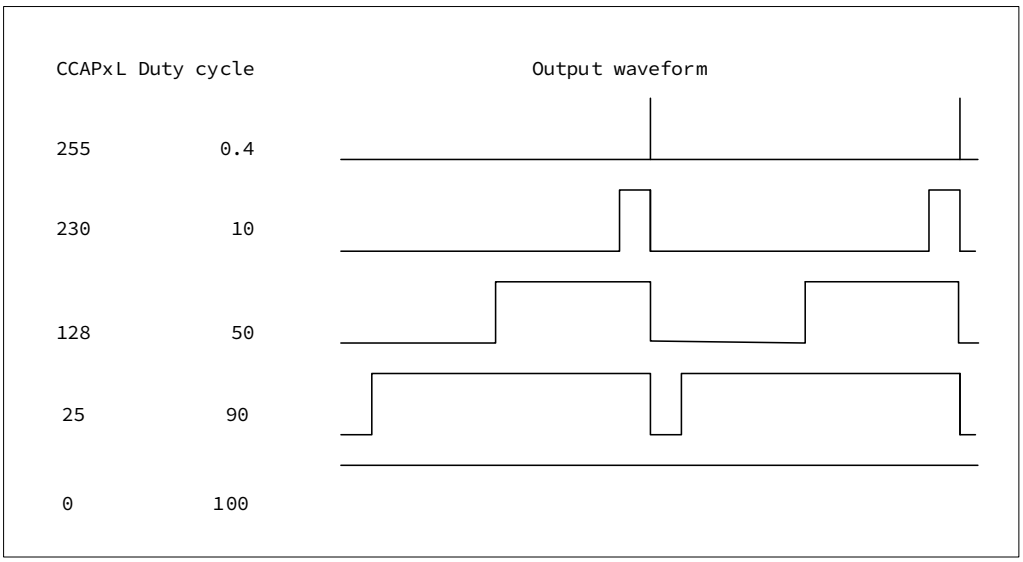


图 15-8 PCA PWM 输出波形

要设定一个比较/捕获模块在 PWM 模式下，需要设置 CCAPMx.ECOM 和 CCAPMx.PWM 位。另外 PCA 定时器/计数器由编程 CMOD.CSP[2:0]可以选择输入计数信号频率。在 CCAPxL 输入一个 8 位的值指定第一个 PWM 波形的占空比。在 CCAPxH 输入一个 8 位的值会指定第二个 PWM 波形的占空比。设置定时器/计数器运行控制位（CCON.CR）启动 PCA 定时器/计数器。

表 15-1 PCA 比较捕获功能模块设置

ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	EPWM	工作方式
X	1	0	0	0	0	X	0	用正沿触发捕获
X	0	1	0	0	0	X	0	用负沿触发捕获
X	1	1	0	0	0	X	0	用跳变沿触发捕获
1	0	0	1	0	0	X	0	软件定时器
1	0	0	1	1	0	X	0	高速输出
1	0	0	0	0	1	X	0	8 位脉冲宽度调制器
1	0	0	1	1	0	X	1	16 位脉冲宽度调制器

15.3PCA 模块与其他模块互连及控制

PCA 通过端口功能选择可以连通到其他模块或端口。

		0	1	2	3	4	5	6	7
GPIO_PCAS[2:0]	PCA_ECI	PX_SEL	Reserved	LVD	VC0	VC1	PA05	PB02	PD02
GPIO_PCAS[5:3]	PCA_CH0	PX_SEL	Reserved	Reserved	LVD	VC1	PA06	PB04	PC06

当 GPIO_PCAS[2:0]=0x0 时，PCA_ECI 输入为 PX_SEL 选择的端口输入，当 GPIO_PCAS[2:0]=0x1~0X7 时，连接其他模块的输入或输出。

当 GPIO_PCAS[5:3]=0x0 时，PCA_CH0 捕获输入为 PX_SEL 选择的端口输入，当 GPIO_PCAS[5:3]=0x1~0X7 时，连接其他模块的输入或输出。

15.4PCA 寄存器描述

基地址 0X40001000

表 15-2 PCA 寄存器列表

寄存器	偏移地址	描述
PCA_CCON	0X000	PCA 控制寄存器
PCA_CMOD	0X004	PCA 模式寄存器
PCA_CNT	0X008	PCA 计数寄存器
PCA_ICLR	0X00C	PCA 中断清除寄存器
PCA_CCAPM0	0x010	PCA 比较/捕获模块0模式寄存器
PCA_CCAPM1	0x014	PCA 比较/捕获模块1模式寄存器
PCA_CCAPM2	0x018	PCA 比较/捕获模块2模式寄存器
PCA_CCAPM3	0x01C	PCA 比较/捕获模块3模式寄存器
PCA_CCAPM4	0x020	PCA 比较/捕获模块4模式寄存器
PCA_CCAP0H	0X024	PCA 比较/捕获模块0高8位寄存器
PCA_CCAP0L	0X028	PCA 比较/捕获模块0低8位寄存器
PCA_CCAP1H	0X02C	PCA 比较/捕获模块1高8位寄存器
PCA_CCAP1L	0X030	PCA 比较/捕获模块1低8位寄存器
PCA_CCAP2H	0X034	PCA 比较/捕获模块2高8位寄存器
PCA_CCAP2L	0X038	PCA 比较/捕获模块2低8位寄存器
PCA_CCAP3H	0X03C	PCA 比较/捕获模块3高8位寄存器
PCA_CCAP3L	0X040	PCA 比较/捕获模块3低8位寄存器
PCA_CCAP4H	0X044	PCA 比较/捕获模块4高8位寄存器
PCA_CCAP4L	0X048	PCA 比较/捕获模块4低8位寄存器
PCA_CCAPO	0X04C	PCA PWM与高速输出标志寄存器
PCA_CCAP0	0X050	PCA 比较/捕获模块0 的16位寄存器
PCA_CCAP1	0X054	PCA 比较/捕获模块1 的16位寄存器
PCA_CCAP2	0X058	PCA 比较/捕获模块2 的16位寄存器
PCA_CCAP3	0X05C	PCA 比较/捕获模块3 的16位寄存器
PCA_CCAP4	0X060	PCA 比较/捕获模块4 的16位寄存器
PCA_CARR	0X064	PCA 周期装载寄存器
PCA_EPWM	0X068	PCA PWM增强寄存器

15.4.1 控制寄存器（PCA_CCON）

偏移地址：0x000

复位值: 0x0000 0000h

31-8	7	6	5	4	3	2	1	0
Reserved	CF	CR	Reserved	CCF4	CCF3	CCF2	CCF1	CCF0
	RO	RW		RO	RO	RO	RO	RO

位	符号	描述
31:8	Reserved	保留位
7	CF	PCA 计数器溢出标志（写无效） 当PCA计数溢出时，CF由硬件置位，如果CMOD寄存器的CFIE位为1，则CF标志可以产生中断 1：发生计数器溢出；0：无溢出；
6	CR	PCA计数器运行控制位 1：启动PCA计数器计数 0：关闭PCA计数器计数
5	Reserved	保留位
4	CCF4	PCA计数器模块4 比较/捕获标志位 当出现匹配或捕获时，该位由硬件置位。（写无效） 当CCAPM4.CCIE置位时，这个标志位会产生一个PCA中断
3	CCF3	PCA计数器模块3 比较/捕获标志位 当出现匹配或捕获时，该位由硬件置位。（写无效） 当CCAPM3.CCIE置位时，这个标志位会产生一个PCA中断
2	CCF2	PCA计数器模块2 比较/捕获标志位 当出现匹配或捕获时，该位由硬件置位。（写无效） 当CCAPM2.CCIE置位时，这个标志位会产生一个PCA中断
1	CCF1	PCA计数器模块1 比较/捕获标志位 当出现匹配或捕获时，该位由硬件置位。（写无效） 当CCAPM1.CCIE置位时，这个标志位会产生一个PCA中断
0	CCF0	PCA计数器模块0 比较/捕获标志位 当出现匹配或捕获时，该位由硬件置位。（写无效） 当CCAPM0.CCIE置位时，这个标志位会产生一个PCA中断

15.4.2 模式寄存器（PCA_CMOD）

偏移地址：0x004

复位值: 0x0000 0000h

31-8	7	6	5	4	3	2	1	0
Reserved	CIDL	WDTE	Reserved		CPS			CFIE
	RW	RW			RW			RW

位	符号	描述
31:8	Reserved	保留位
7	CIDL	空闲模式IDLE下，PCA是否停止工作 1：休眠模式（sleep）下，PCA停止工作 0：休眠模式（sleep）下，PCA继续工作
6	WDTE	PCA WDT功能使能控制位 1：启动PCA模块4 WDT功能 0：关闭PCA模块4 WDT功能
5:4	Reserved	保留位
3:1	CPS[2:0]	时钟分频选择及时钟源选择 000：PCLK/32 001：PCLK/16 010：PCLK/8 011：PCLK/4 100：PCLK/2 101：timer0 overflow 110：timer1 overflow 111：ECI 外部时钟，时钟PCLK四分频采样
0	CFIE	PCA计数器中断使能控制信号 1：使能中断 0：关闭中断

15.4.3 计数寄存器（PCA_CNT）

偏移地址：0x008

复位值: 0x0000 0000h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	定时器计数器的值 只有在PCA停止状态CNT才可以写入。否则写入无效

15.4.4 中断清除寄存器（PCA_ICLR）

偏移地址：0x00C

复位值: 0x0000 009Fh

31-8	7	6	5	4	3	2	1	0
Reserved	CF	Reserved		CCF4	CCF3	CCF2	CCF1	CCF0
	R1W0			R1W0	R1W0	R1W0	R1W0	R1W0

位	符号	描述
31:8	Reserved	保留位
7	CF	PCA 计数器溢出标志清除（软件写0清零，写1无效），读出值为1
6:5	RSV	保留位
4	CCF4	PCA计数器模块4 比较/捕获标志位清除（软件写0清零，写1无效），读出值为1
3	CCF3	PCA计数器模块3 比较/捕获标志位清除（软件写0清零，写1无效），读出值为1
2	CCF2	PCA计数器模块2 比较/捕获标志位清除（软件写0清零，写1无效），读出值为1
1	CCF1	PCA计数器模块1 比较/捕获标志位清除（软件写0清零，写1无效），读出值为1
0	CCF0	PCA计数器模块0 比较/捕获标志位清除（软件写0清零，写1无效），读出值为1

15.4.5 比较捕获模式寄存器（PCA_CCAPM0~4）

偏移地址：

CCAPM0：0x010;　　CCAPM1：0x014;　　CCAPM2：0x018;

CCAPM3：0x01C;　　CCAPM4：0x020;

复位值: 0x0000 0000h

31-8	7	6	5	4	3	2	1	0
Reserved		ECOM	CAPP	CAPN	MAT	TOG	PWM	CCIE
		RW	RW	RW	RW	RW	RW	RW

位	符号	描述
31:7	Reserved	保留位
6	ECOM	允许比较器功能控制位 1：允许比较器功能；0：禁止比较强功能； 当PCA用于软件计数器，高速输出，PWM模式，WDT模式，要置位ECOM 写CCAMPHx或CCAMPx寄存器会自动置位ECOM；写CCAMPLx寄存器会自动清除ECOM位
5	CAPP	正沿捕获控制位 1：允许上升沿捕获；0：禁止上升沿捕获
4	CAPN	负沿捕获控制位 1：允许下降沿捕获；0：禁止下降沿捕获
3	MAT	允许匹配控制位 1：PCA计数值与模块的比较/捕获寄存器的值一旦匹配，将置位CCON寄存的中断标志CCFx(x=0-4) 0：禁止匹配功能
2	TOG	翻转控制位 1：工作在PCA高速输出模式，PCA计数器的值与模块的比较/捕获寄存器的值一旦匹配，CCPx引脚翻转 0：禁止翻转功能
1	PWM	脉宽调制控制位 1：允许CCPx引脚作为PWM输出 0：禁止PWM脉宽调制功能 **只有CCAPMx[6:0]=100_0010时，PWM功能才有效**
0	CCIE	PCA使能中断 1：使能比较/捕获中断 0：PCA比较/捕获功能中断禁止

15.4.6 比较捕获数据寄存器高 8 位（PCA_CCAP0~4H）

偏移地址

CCAP0H：0x024;　　CCAP1H：0x02C;　　CCAP2H：0x034;

CCAP3H：0x03C;　　CCAP4H：0x044;

复位值: 0x0000 0000h

31:8	7	6	5	4	3	2	1	0
Reserved	CCAPx[15：8]							
	RW							

位	符号	描述
31:8	Reserved	保留位
7:0	CCAPx[15:8]	比较/捕获模式高8位寄存器 当PCA模式用于比较/捕获模式时，用于保存16位捕获计数值的高8位；写CCAPxH寄存器会自动置位寄存器CCAPMx的ECOM位。 当PCA模式用于PWM模式时，用于控制输出占空比装载寄存器，在计数器低8位溢出时，装载寄存器会自动更新到PWM比较寄存器

15.4.7 比较捕获数据寄存器低 8 位（PCA_CCAP0~4L）

偏移地址

CCAP0L：0x028;　　CCAP1L：0x030;　　CCAP2L：0x038;

CCAP3L：0x040;　　CCAP4L：0x048;

复位值: 0x0000 0000h

31:8	7	6	5	4	3	2	1	0
Reserved	CCAPx[7：0]							
	RW							

位	符号	描述
31:8	Reserved	保留位
7:0	CCAPx[7:0]	比较/捕获模式低8位寄存器 当PCA模式用于比较/捕获模式时，用于保存16位捕获计数值的低8位；写CCAPxL寄存器会自动清除寄存器CCAPMx的ECOM位。 当PCA模式用于PWM模式时，用于控制输出占空比比较寄存器，在PWM模式，计数器的低8位的值小于CCAPx[7:0]的值PWM输出低电平，否则PWM输出高电平。

15.4.8 比较捕获 16 位寄存器（PCA_CCAP0~4）

偏移地址

CCAP0：0x050;　　CCAP1：0x054;　　CCAP2：0x058;

CCAP3：0x05C;　　CCAP4：0x060;

复位值: 0x0000 0000h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	符号	描述
31:16	Reserved	保留位
15:0	CCAPx	比较/捕获模式16位寄存器 当PCA模式用于比较/捕获模式时，用于保存16位捕获计数值；写CCAPx寄存器会置位寄存器CCAPMx的ECOM位。 写CCAPX寄存器相当于写CCAPxL及CCAPxH这两个8位寄存器。在比较/捕获模式下可以直接读写这个寄存器，在PWM模式下，使用CCAPxL及CCAPxH寄存器

15.4.9 比较高速输出标志寄存器（PCA_CCAPO）

偏移地址：0x04C

复位值: 0x0000 0000h

31:8	7	6	5	4	3	2	1	0
Reserved				CCAPO4	CCAPO3	CCAPO2	CCAPO1	CCAPO0
				RW	RW	RW	RW	RW

位	符号	描述
31:5	Reserved	保留位
4	CCAPO4	比较模块4的输出值
3	CCAPO3	比较模块3的输出值
2	CCAPO2	比较模块2的输出值
1	CCAPO1	比较模块1的输出值
0	CCAPO0	比较模块0的输出值

15.4.10 周期寄存器（PCA_CARR）

偏移地址：0x064

复位值: 0x0000 0000h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	符号	描述
31:16	Reserved	保留位
15:0	CARR	计数周期重载寄存器

15.4.11 增强 PWM 控制（PCA_EPWM）

偏移地址：0x068

复位值: 0x0000 0000h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															EPWM
															RW

位	符号	描述
31:1	Reserved	保留位
0	EPWM	16 bit PWM 使能

16 高级定时器（TIM4/5/6）

16.1Advanced Timer 简介

Advanced Timer 是一个包含三个定时器 Timer4/5/6。Timer4/5/6 功能相同的高性能计数器，可用于计数产生不同形式的时钟波形，1 个定时器可以产生互补的一对 PWM 或者独立的 2 路 PWM 输出，可以捕获外界输入进行脉冲宽度或周期测量。

Advanced Timer 基本的功能及特性如表所示。

表 16-1 Advanced Timer 基本特性

波形模式	锯齿波、三角波
基本功能	• 递加、递减计数方向
	• 软件同步
	• 硬件同步
	• 缓存功能
	• 正交编码计数
	• 通用 PWM 输出
	• 保护机制
	• AOS 关联动作
中断类型	计数比较匹配中断
	计数周期匹配中断
	死区时间错误中断

表 16-2 Advanced Timer 端口列表

端口名	方向	功能
TIMx_CHA	输入/输出	正交编码计数时钟输入端口或捕获输入端口或比较输出端口（x=4~6）
TIMx_CHB		2）硬件启动、停止、清零条件输入端口
TIMTRIA	输入	硬件计数时钟输入端口或捕获输入端口 硬件启动、停止、清零条件输入端口
TIMTRIB		
TIMTRIC		
TIMTRID		

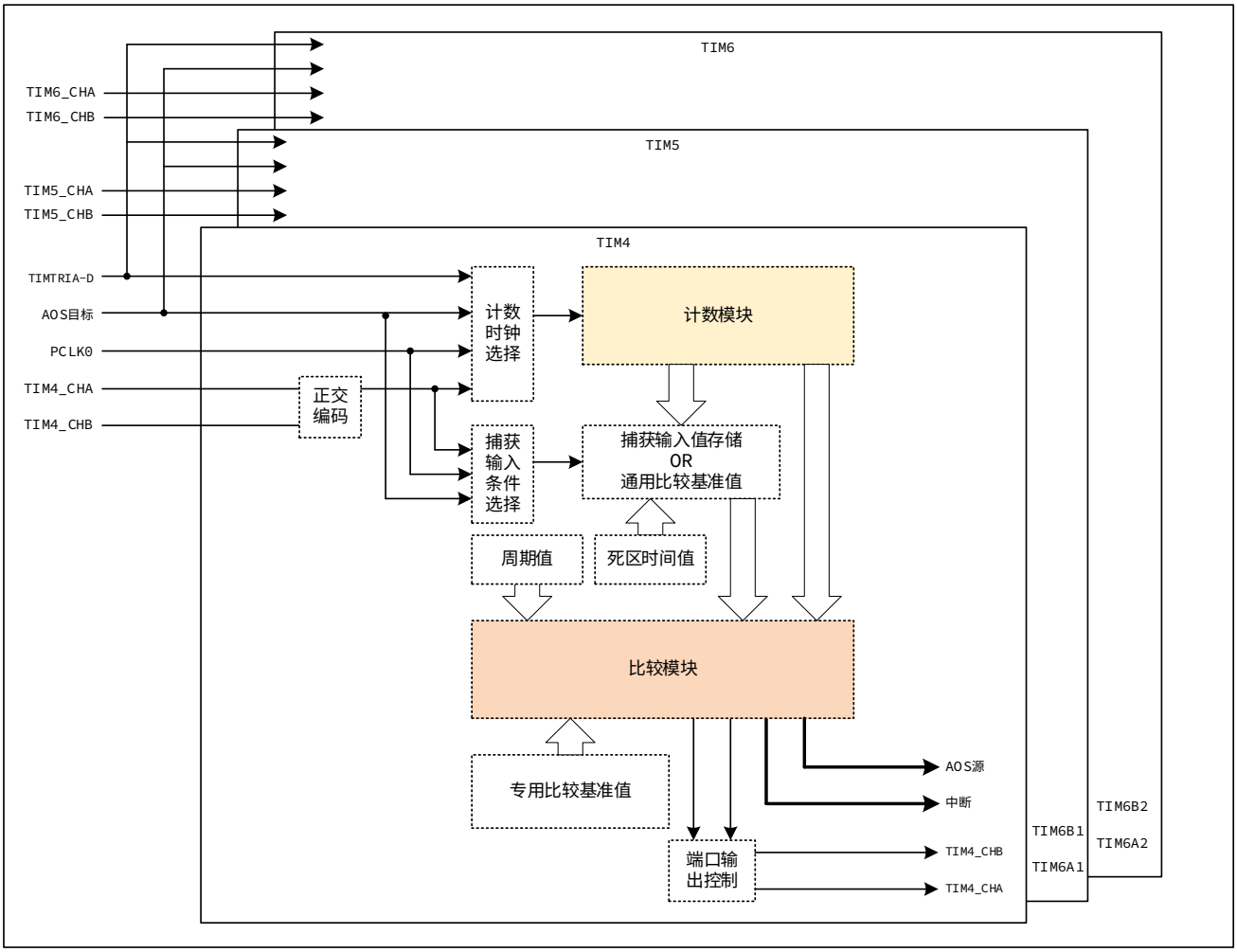


图 16-1 Advanced Timer 框图

16.2Advanced Timer 功能描述

16.2.1 基本动作

16.2.1.1 基本波形模式

Timer4/5/6 有 2 种基本计数波形模式，锯齿波模式和三角波模式。波形模式又由于不同的内部计数动作有所细分，三角波模式分为三角波 A 模式、三角波 B 模式。锯齿波和三角波的基本波形如图 16-2 图 16-3 所示。三角波 A 模式与三角波 B 模式区别在于缓存传送有差别，三角波 A 模式一个周期只发生一次缓存传送（谷点），而三角波 B 模式一个周期发生两次缓存传送（峰点和谷点）。

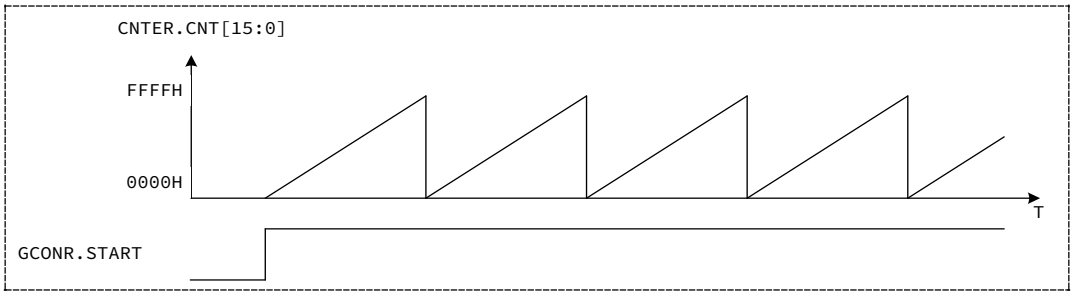


图 16-2 锯齿波波形（递加计数）

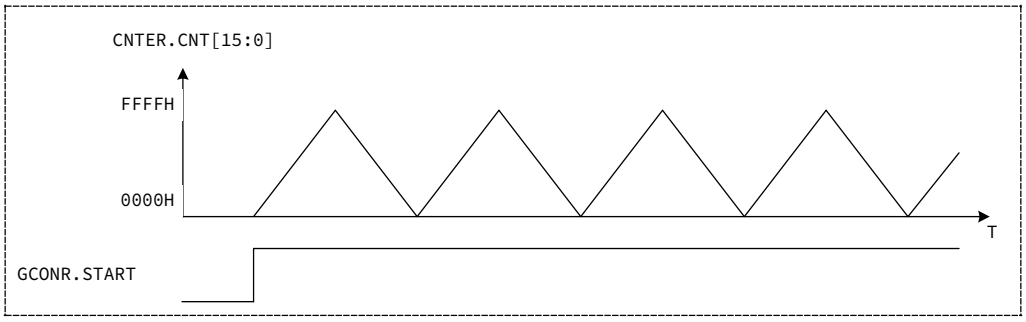


图 16-3 三角波波形

16.2.1.2 比较输出

Timer4/5/6 一个定时器有 2 个比较输出端口（CHxA、CHxB），可在计数值与计数基准值比较匹配时输出指定的电平。GCMAR、GCMBR 寄存器分别对应了 CHxA、CHxB 的计数比较基准值。当计数器的计数值和 GCMAR 相等时，CHxA 端口输出指定的电平；当计数器的计数值和 GCMBR 相等时，CHxB 端口输出指定电平。

CHxA、CHxB 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、PCONR.STPCA、PCONR.STASTPSA、PCONR.CMPCA[1:0]、PCONR.PERCA[1:0] 和 PCONR.STACB、PCONR.STPCB、PCONR.STASTPSB、PCONR.CMPCB[1:0]、PCONR.PERCB[1:0] 位设定。图 16-4 为比较输出的动作例。

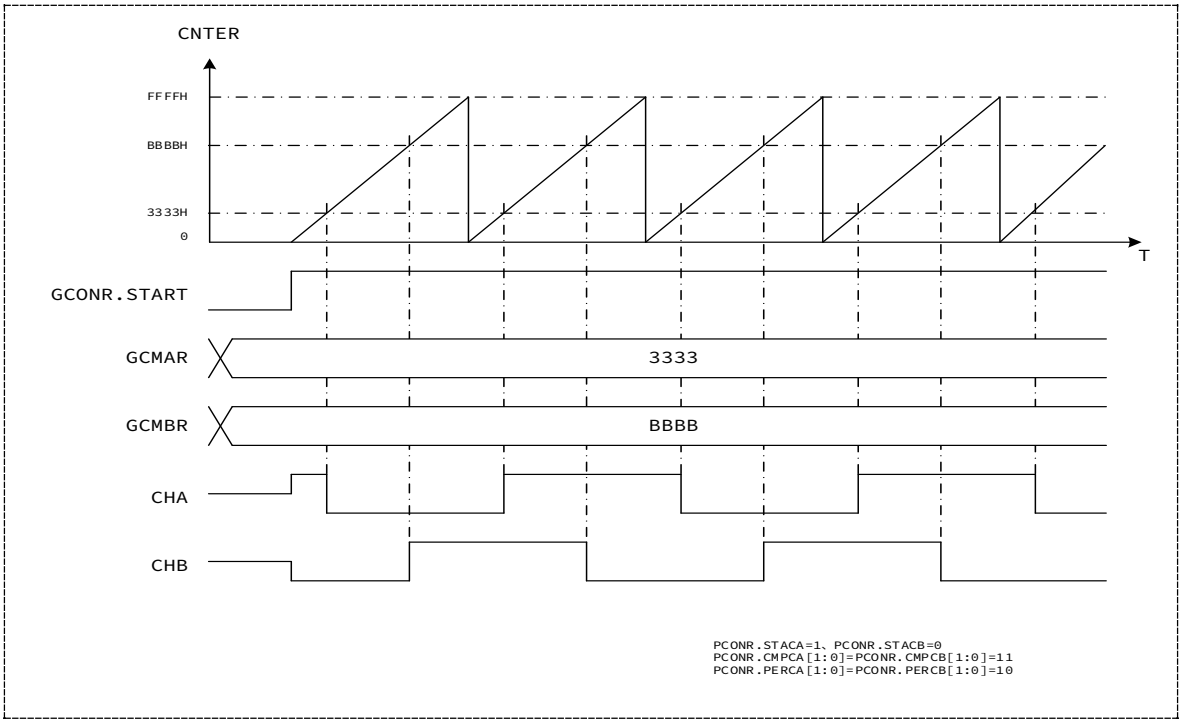


图 16-4 比较输出动作

16.2.1.3 捕获输入

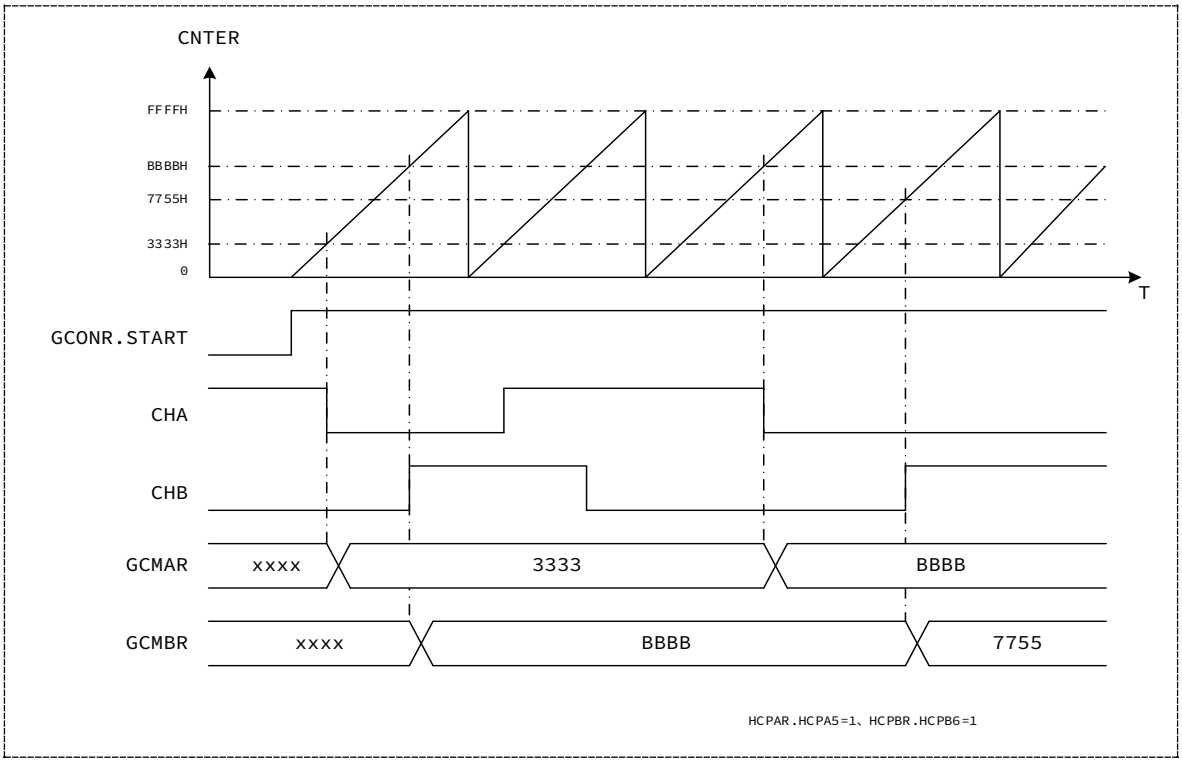


图 16-5 捕获输入动作

Timer4/5/6 都具有捕获输入功能，具备 2 组捕获输入寄存器（GCMAR、GCMBR），用于保存捕获到的计数值。设定端口控制寄存器（PCONR）的 PCONR.CAPCA、PCONR.CAPCB 位为 1，捕获输入功能就有效了。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寄存器（GCMAR、GCMBR）中。

每组捕获输入的条件可以是 AOS 事件触发、TIMTRIA-TIMTRID 输入、CHxA 或 CHxB 的输入等，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 16-5 为捕获输入的动作例。

16.2.2 时钟源选择

Timer4/5/6 的计数时钟可以有以下几种选择：

a. PCLK 及 PCLK 的 2、4、8、16、64、256、1024 分频（GCONR.CKDIV[2:0] 设定）
b. AOS 事件触发输入（HCUPR.HCUP [19:16] 或 HCDOR.HCDO [19:16] 设定）
c. CHxA 和 CHxB 的正交编码输入（HCUPR.HCUP[7:0] 或 HCDOR.HCDO [7:0] 设定）
d. TIMTRIA-TIMTRID 的端口输入（HCUPR.HCUP [15:8] 或 HCDOR.HCDO [15:8] 设定）

从上述描述可以看到，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。

16.2.3 计数方向

Timer4/5/6 的计数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。

16.2.3.1 锯齿波计数方向

锯齿波模式时，计数方向可在计数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），则计数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），则计数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。则计数开始后直至上溢或下溢时，GCONR.DIR 的设定才会反映到计数中。

16.2.3.2 三角波计数方向

三角波模式时，计数方向只能在计数器停止时设定。在计数中设定计数方向无效。

在计数停止时，设定 GCONR.DIR 位。则计数开始后直至上溢或下溢时，GCONR.DIR 的设定才会反映到计数中。

16.2.4 数字滤波

Timer4/5/6 的 CHxA、CHxB、TIMTRIA~D 端口输入都有数字滤波功能。可通过设定滤波控制寄存器（FCONR）的相关使能位开启对应端口的滤波功能。滤波用的基准时钟也通过滤波控制寄存器（FCONR）设定。

在滤波采样基准时钟采样到端口上 3 次一致的电平时，该电平被当作有效电平传送到模块内部；小于 3 次一致的电平会被当作外部干扰滤掉，不传送到模块内部。其动作例如图 16-6 所示。

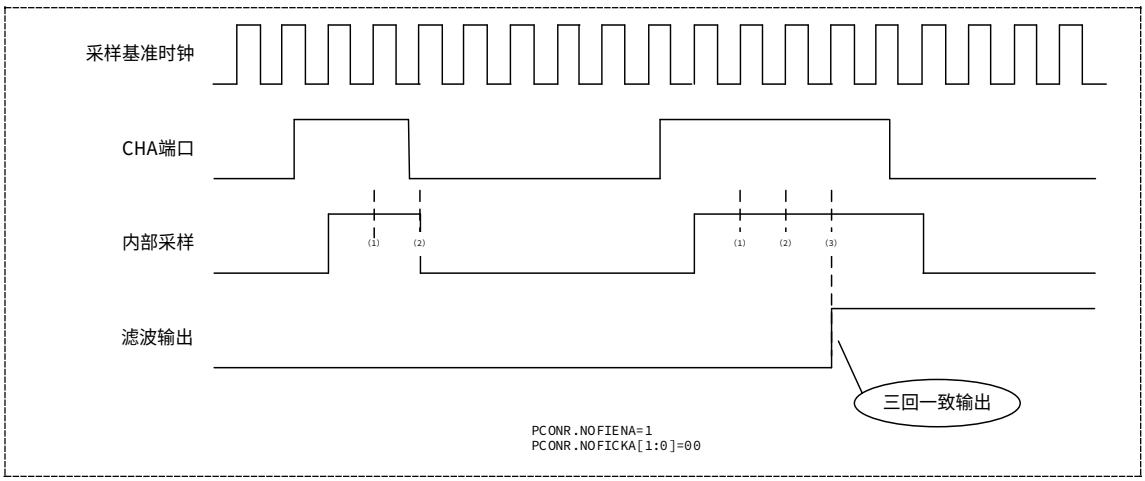


图 16-6 捕获输入端口的滤波功能

TIMTRIA~D 端口是一组 Timer4/5/6 间共用的端口，该组端口的数字滤波功能在只在 Timer4 实现，其他定时器 Timer5/6 对该组端口的数字滤波功能设定无效。

16.2.5 软件同步

16.2.5.1 软件同步启动

Timer4/5/6 可通过设定软件同步启动寄存器（SSTAR）的相关位，实现目标 Timer4/5/6 的同步启动。

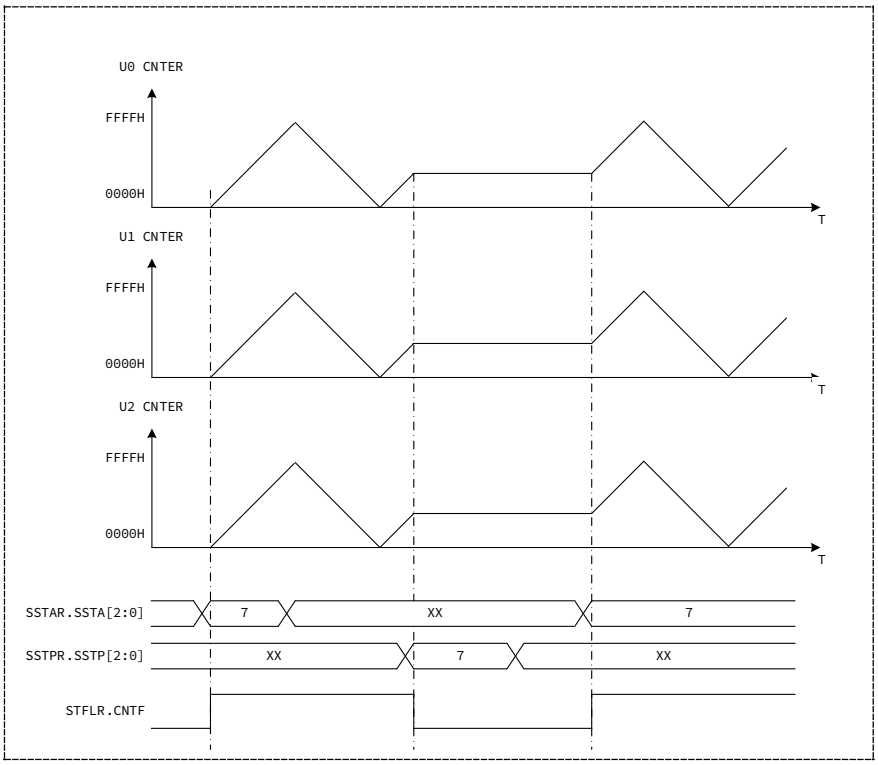


图 16-7 软件同步动作

16.2.5.2 软件同步停止

Timer4/5/6 可通过设定软件同步停止寄存器（SSTPR）的相关位，实现目标 Timer4/5/6 的同步停止。

16.2.5.3 软件同步清零

Timer4/5/6 可通过设定软件同步清零寄存器（SCLRR）的相关位，实现目标 Timer4/5/6 的同步清零。

如图 16-7 所示、若设定 Timer4 的 SSTAR.SSTA0=SSTAR.SSTA1=SSTAR.SSTA2, 即可实现 Timer4/5/6 的软件同步启动。

软件同步动作相关寄存器（SSTAR、SSTPR、SCLRR）是一组独立于 Timer4/5/6 外、各个 TIM 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，会读出各个定时器的计数器状态，在读取 SSTPR 或 SCLRR 时，会读出 0。

16.2.6 硬件同步

每个定时器除独立拥有 2 个通用输入端口（CHxA、CHxB）外，还共同拥有 4 个外部通用输入端口（TIMTRIA、TIMTRIB、TIMTRIC、TIMTRID）及 4 个 AOS 目标，可实现定时器间的硬件同步动作。

16.2.6.1 硬件同步启动

各 Timer4/5/6 均可选择用硬件方式启动计数器，选择相同硬件启动条件的定时器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。

16.2.6.2 硬件同步停止

各 Timer4/5/6 均可选择用硬件方式停止计数器，选择相同硬件停止条件的定时器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。

16.2.6.3 硬件同步清零

各 Timer4/5/6 均可选择用硬件方式清零计数器，选择相同硬件清零条件的定时器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。

16.2.6.4 硬件同步捕获输入

各 Timer4/5/6 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定时器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。

16.2.6.5 硬件同步计数

Timer4/5/6 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定时器即可在硬件计数 CLOCK 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，不影响计数器的启动、停止、清零动作。计数器的启动、停止、清零等还需要单独设定。

图 16-8 所示、Timer4/5/6 的硬件同步动作例。

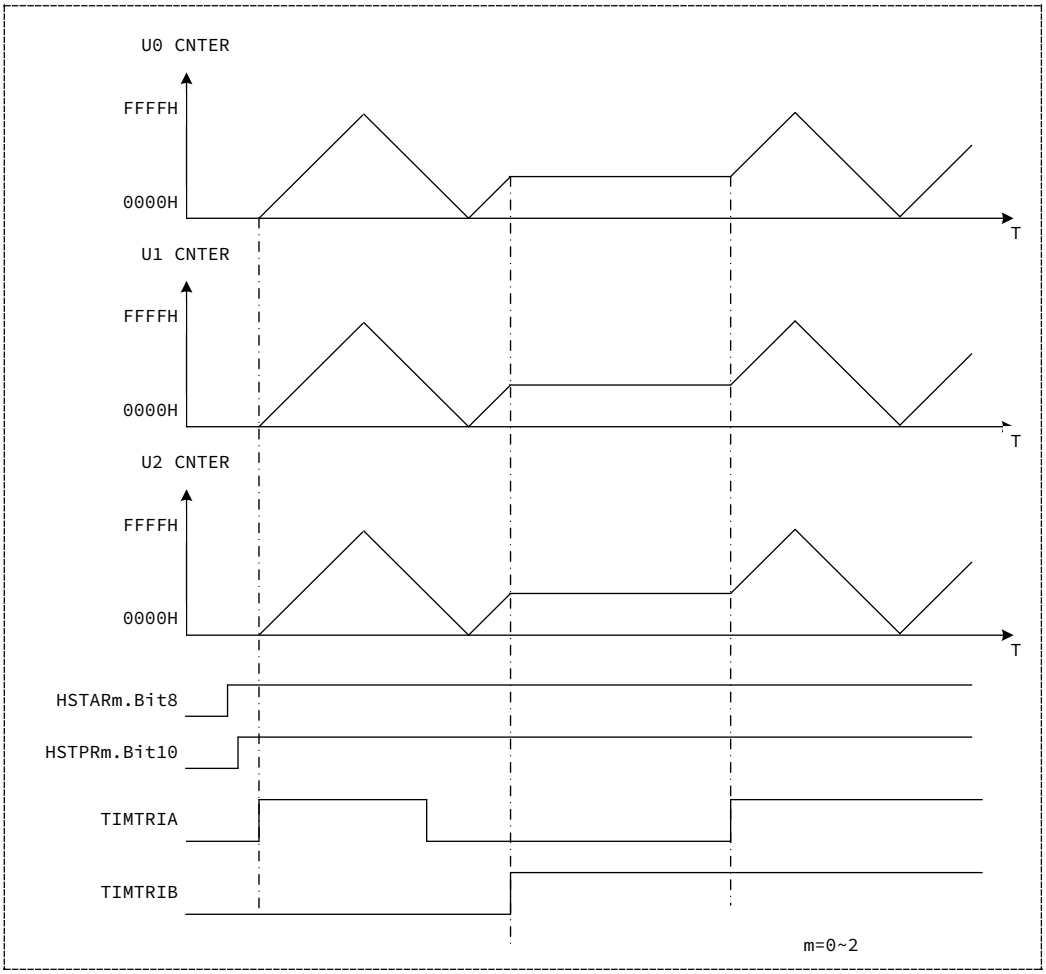


图 16-8 硬件同步动作

16.2.7 缓存功能

缓存动作是指通过设定缓存控制寄存器（BCONR），在缓存传送时间点，选择发生以下事件：

a. 通用周期基准值缓存寄存器（PERBR）的值自动传送到通用周期基准值寄存器（PERAR）中
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）

图 16-9 所示，是比较输出动作时、通用比较基准值寄存器的单缓存方式的时序图。从图中可以看到，在计数期间改变通用比较基准值寄存器（GCMAR）的值可以调整输出占空比，改变通用周期基准值寄存器（PERAR）的值可以调整输出周期。

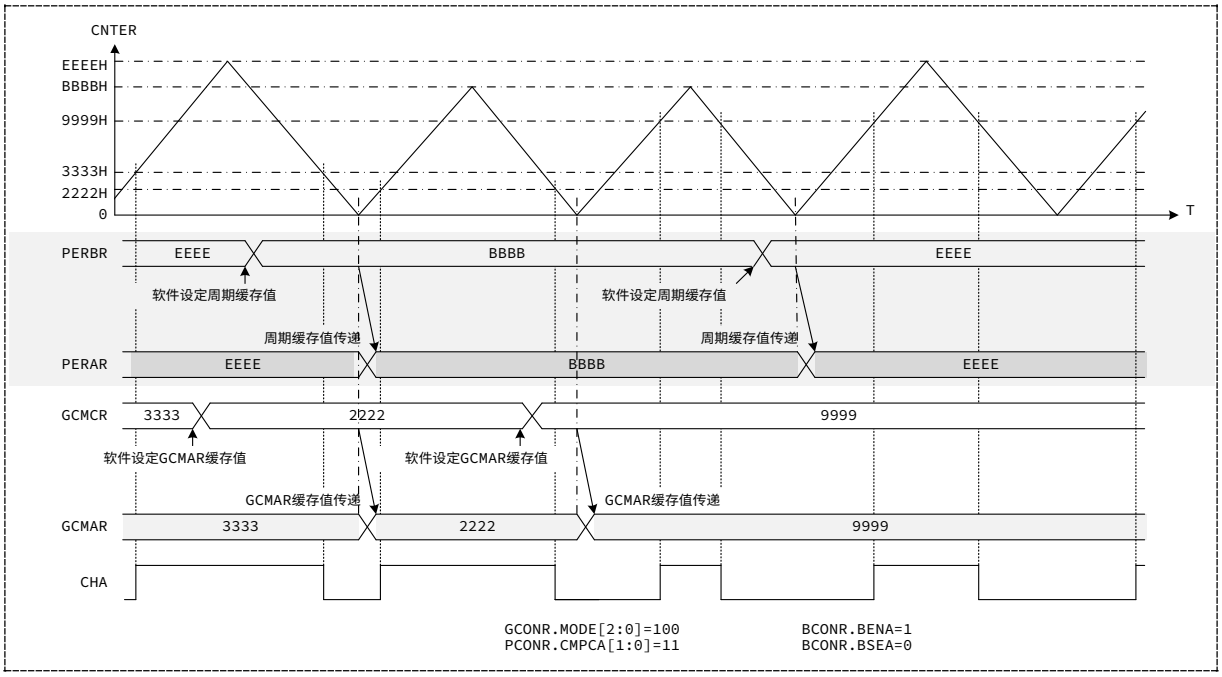


图 16-9 单缓存方式比较输出时序

16.2.7.1 缓存传送时间点

通用周期基准值缓存传送时间点

周期基准值缓存传送时间点为锯齿波时递加计数上溢点或递减计数下溢点、三角波时计数谷点。

通用比较基准值缓存传送时间点

锯齿波 A 模式时，设定 BCONR.BENA=1 或 BCONR.BNEB=1，缓存动作有效。缓存传送发生在上溢点或下溢点。

三角波 A 模式时，设定 BCONR.BENA=1 或 BCONR.BNEB=1，缓存动作有效。缓存传送发生在计数谷点。

三角波 B 模式时，设定 BCONR.BENA=1 或 BCONR.BNEB=1，缓存动作有效。缓存传送发生在计数谷点和计数峰点。

捕获输入值缓存传送时间点

捕获输入动作缓存传送时间点为捕获输入动作时。

清零动作时缓存传送

在锯齿波计数模式或硬件计数模式时，正常的比较输出动作期间若有清零动作产生，通用周期基准值、通用比较基准值、等会根据相应的缓存动作设定状况发生一次缓存传送。

16.2.8 通用 PWM 输出

16.2.8.1 PWM 展频输出

为了降低 PWM 输出对外部的干扰，在 PWM 输出级有展频配置。每个 PWM 输出周期会微调 PWM 输出的相位。

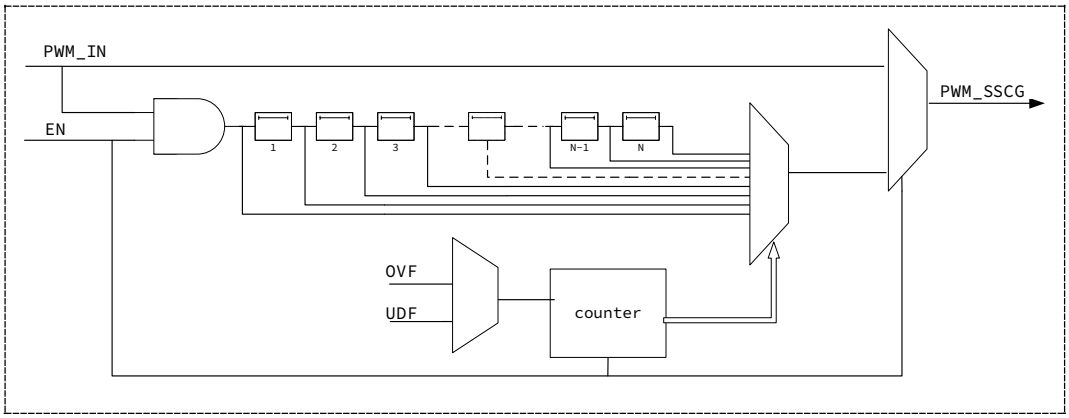


图 16-10 PWM 展频输出示意图

16.2.8.2 独立 PWM 输出

每个定时器的 2 个端口 CHxA、CHxB 能独立的输出 PWM 波。如图 16-11，定时器 Timer6 的 CHA 端口输出 PWM 波。

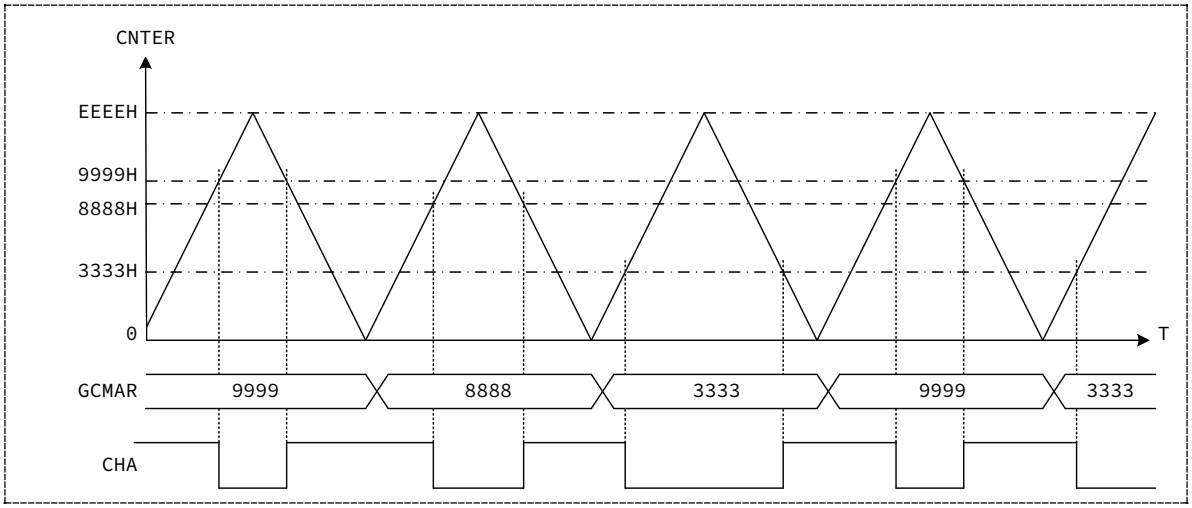


图 16-11 CHA 输出 PWM 波

16.2.8.3 互补 PWM 输出

CHxA 端口和 CHxB 端口，在不同的模式下可组合输出互补 PWM 波形。

软件设定 GCMBR 互补 PWM 输出

软件设定 GCMBR 互补 PWM 输出是指在锯齿波模式和三角波 A 模式、三角波 B 模式下，用于 CHxB 端口波形输出的通用比较基准值寄存器（GCMBR）的值由寄存器直接设定，与通用比较基准值寄存器（GCMAR）的值没有直接关系。

图 16-12 为软件设定 GCMBR 互补 PWM 波的输出例。

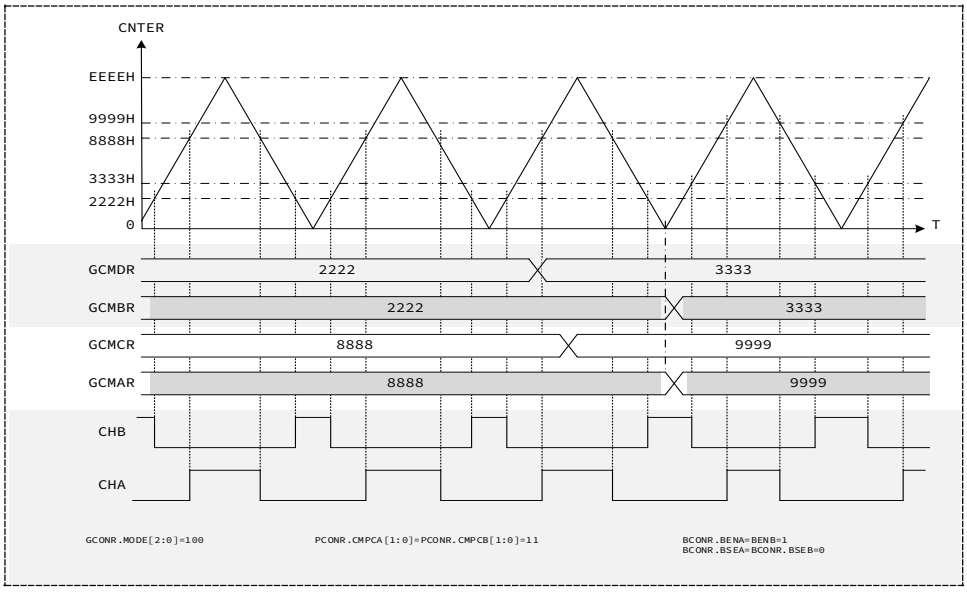


图 16-12 三角波 A 模式时软件设定 GCMBR 互补 PWM 波输出

硬件设定 GCMBR 互补 PWM 输出

硬件设定 GCMBR 互补 PWM 输出是指在三角波 A 模式、三角波 B 模式下，用于 CHxB 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（DTUAR、DTDAR）的值运算决定。

图 16-13 为硬件设定 GCMBR 互补 PWM 波输出例。

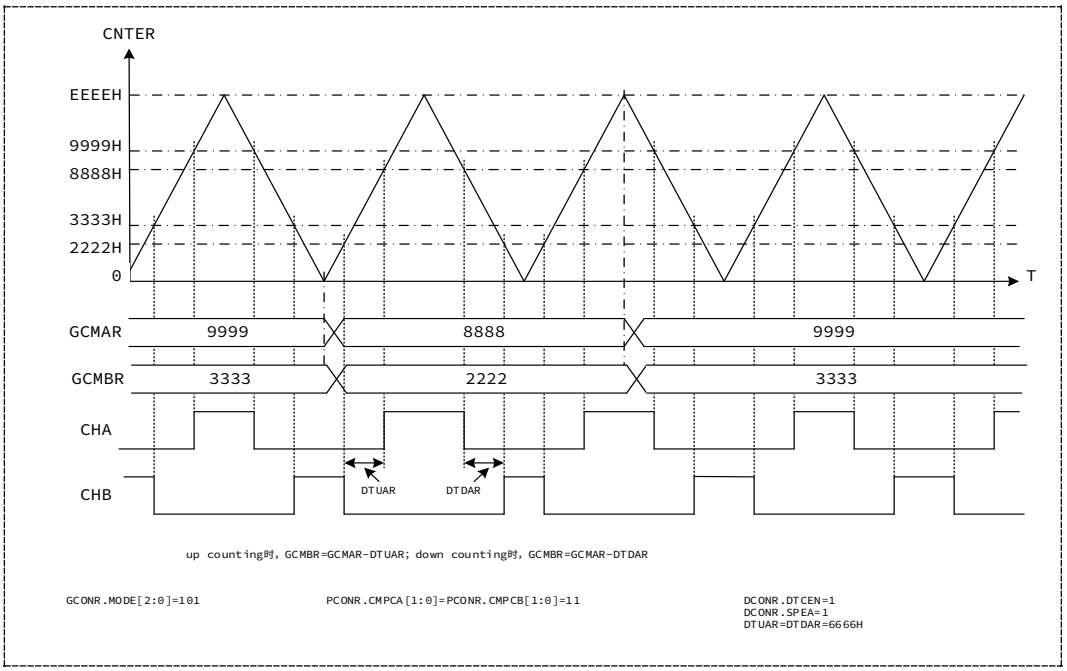


图 16-13 三角波 B 模式时硬件设定 GCMBR 互补 PWM 波输出（对称死区）

16.2.8.4 多相 PWM 输出

每个定时器的 CHxA、CHxB 端口都能输出 2 相独立的 PWM 波或一组互补 PWM 波，多个定时器间组合，同时结合软件、硬件同步动作就可实现多相 PWM 波输出。如图 16-14，Timer4，Timer5，Timer6 组合输出 6 相 PWM 波；如图 16-15，Timer4，Timer5，Timer6 组合输出 3 相互补 PWM 波。

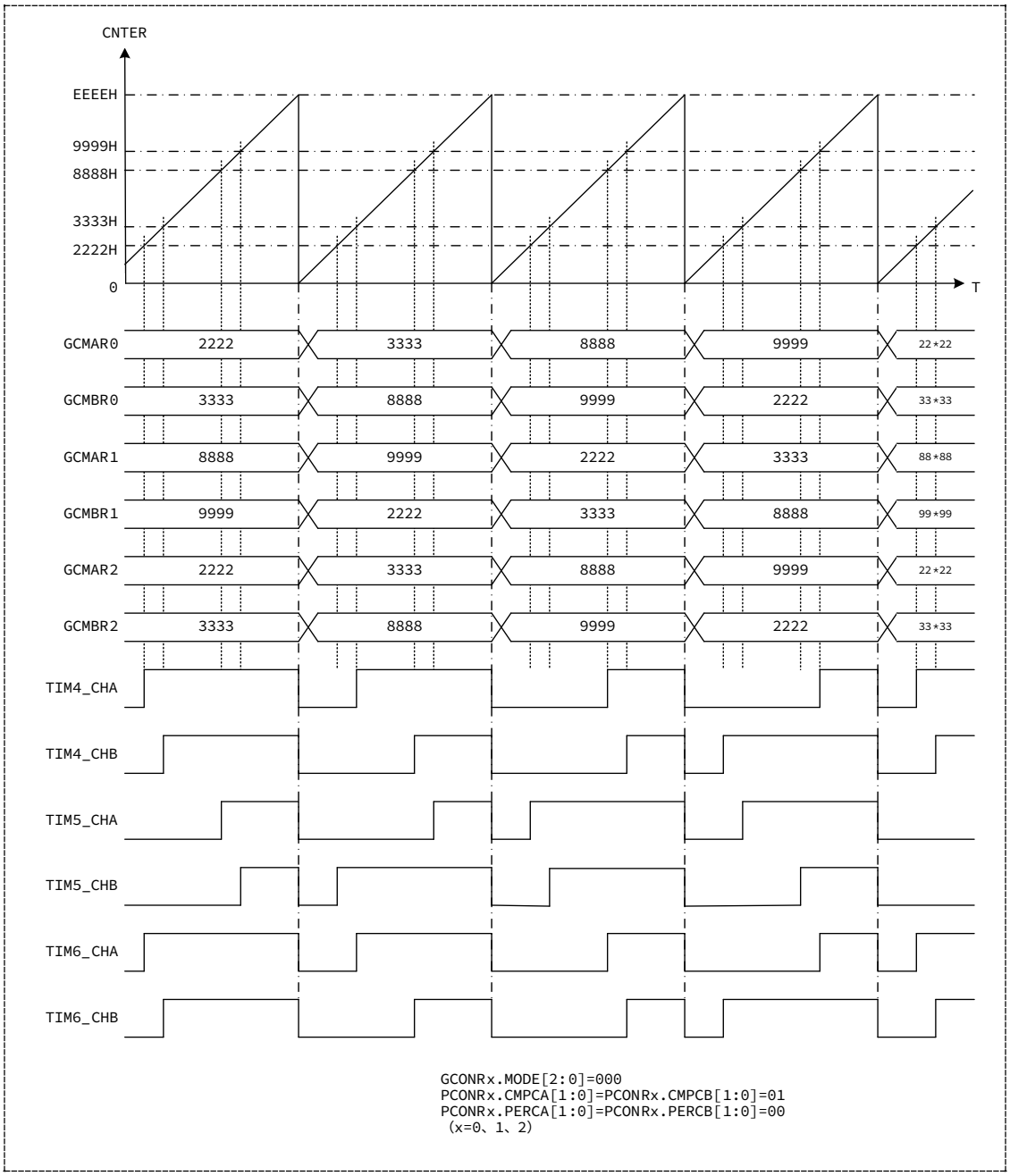


图 16-14 6 相 PWM 波

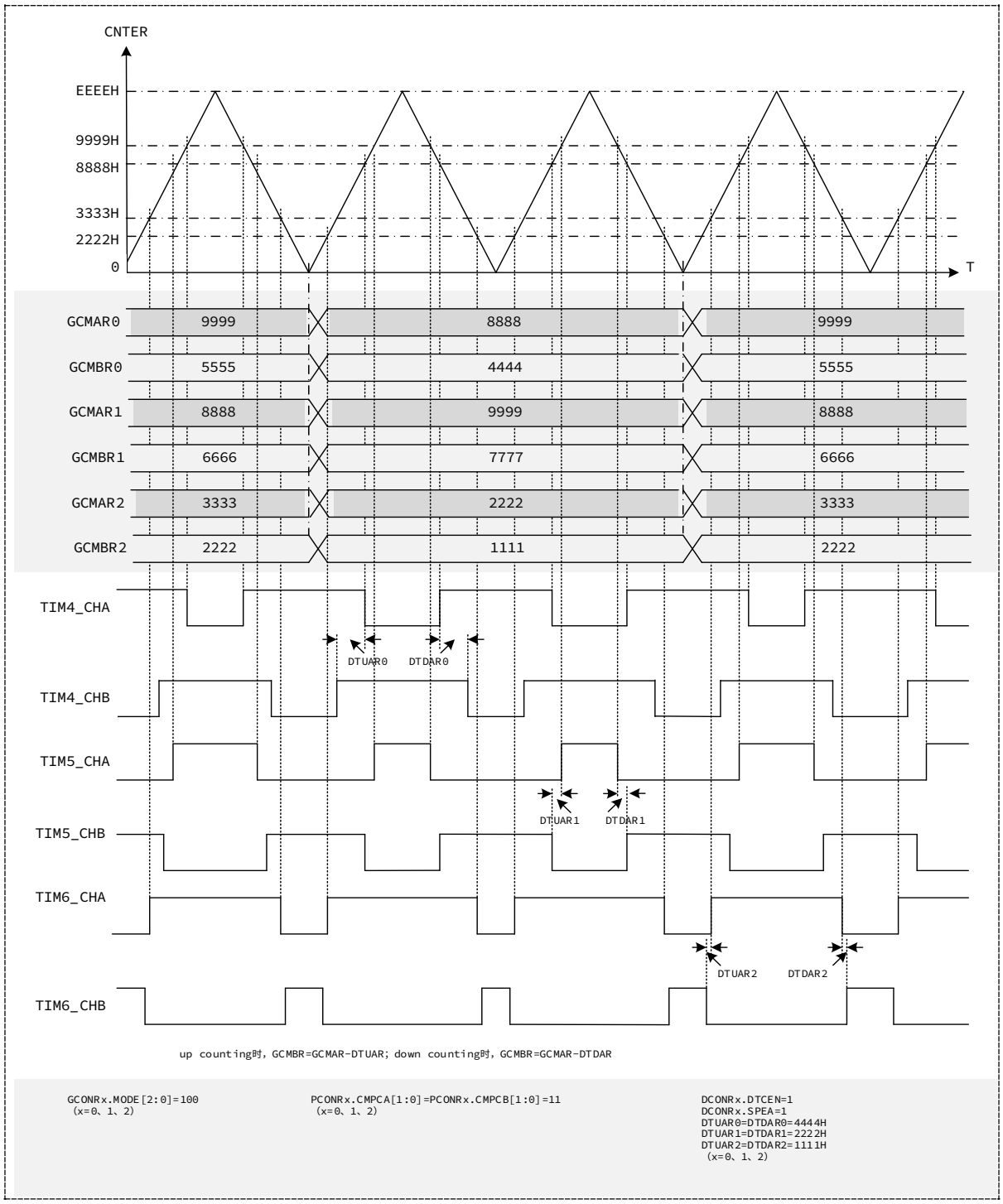


图 16-15　三角波 A 模式时带死区时间三相互补 PWM 波输出

16.2.9 正交编码计数

将 CHxA 输入看作 AIN 输入、CHxB 输入看作 BIN 输入、TIMTRIA-D 中的任意一个输入看作 ZIN 输入，Advanced Timer 就可以实现三路输入的正交编码计数。

一个定时器的 AIN、BIN 单独动作可以实现位置计数模式；两个定时器的 AIN、BIN、ZIN 组合动作可以实现公转计数模式，一个定时器用于位置计数，一个定时器用于公转计数。

公转计数模式时，每两个定时器组合（定时器 4、5 组合，定时器 4 作为位置计数单元，定时器 5 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 CHxA 和 CHxB 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计数器清零、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计数器计数。

16.2.9.1 位置计数模式

正交编码位置模式，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

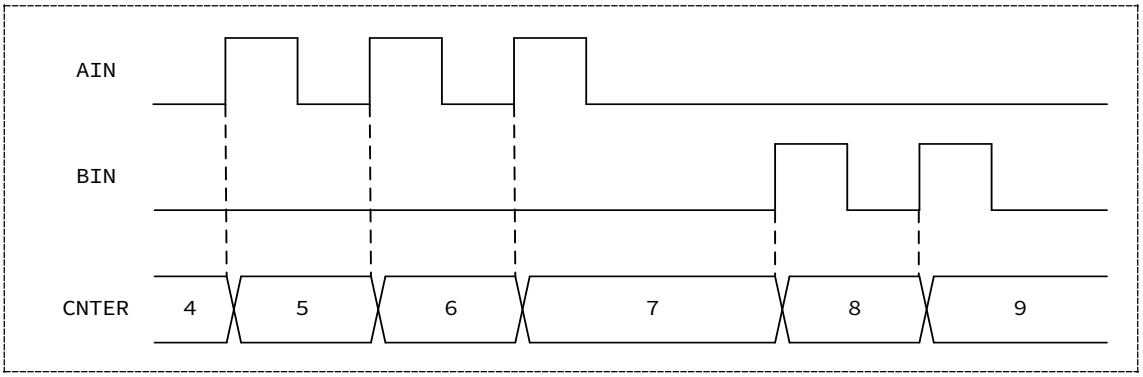


图 16-16 位置模式时基本计数动作

通过设定 HCUPR 与 HCDOR 寄存器，可以灵活的实现各种方式的相位差计数。

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，如下图图 16-17~图 16-19 所示。

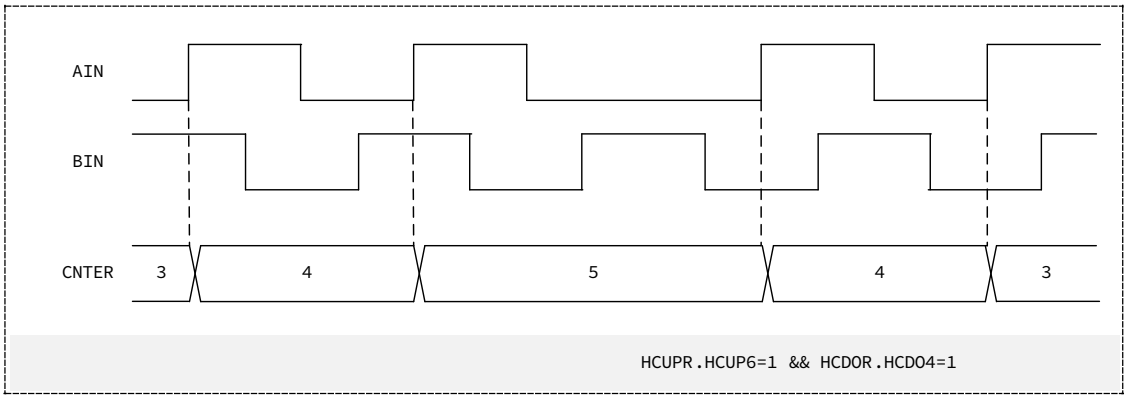


图 16-17 位置模式时相位差计数动作设定(1 倍)

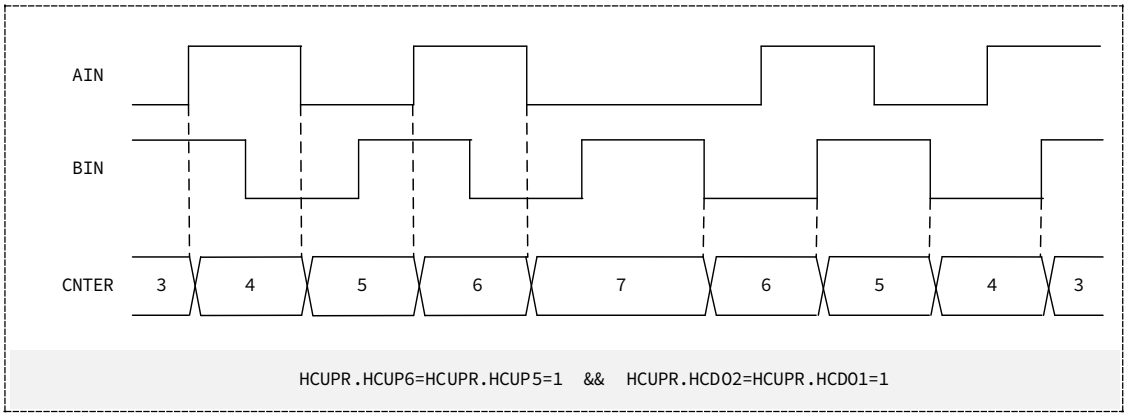


图 16-18 位置模式时相位差计数动作设定(2 倍)

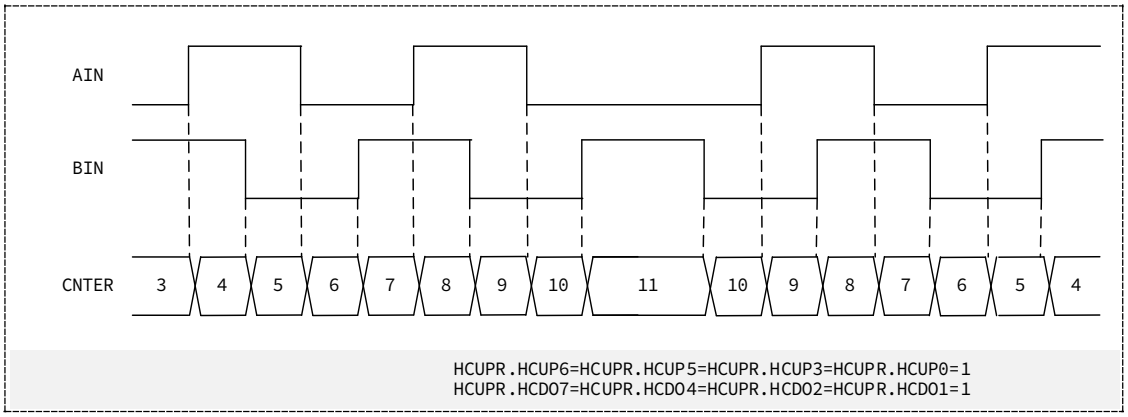


图 16-19 位置模式时相位差计数动作设定(4 倍)

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，如图 16-20 所示。

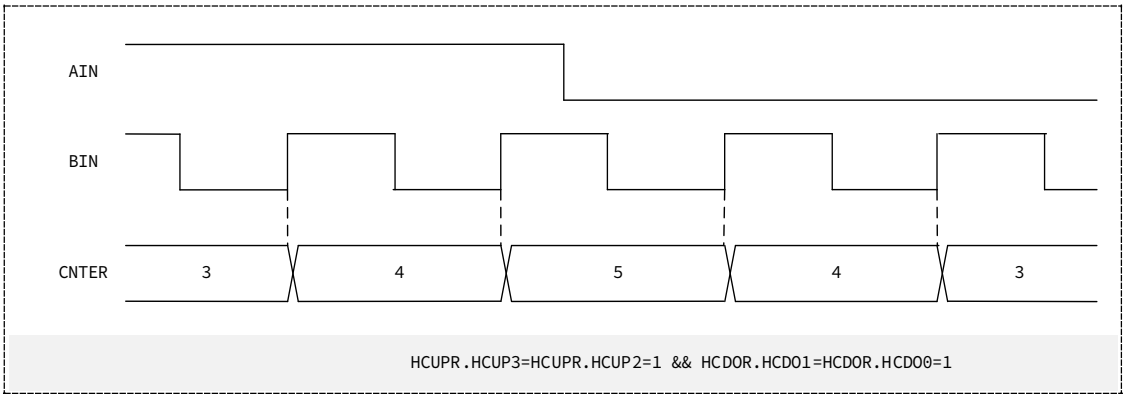


图 16-20 位置模式时方向计数动作

16.2.9.2 公转模式

正交编码公转模式，是指在 AIN、BIN 计数的基础上，加入 ZIN 的输入事件以实现对公转圈数等的判断。公转模式时根据公转计数器的计数方式，可实现 Z 相计数功能、位置计数器输出计数功能和 Z 相计数与位置计数器输出混合计数功能。即使用两个 Advanced Timer 实现此功能。

Z 相计数

Z 相计数是指根据 ZIN 的输入，公转计数单元进行计数，同时将位置计数单元清零的计数动作。

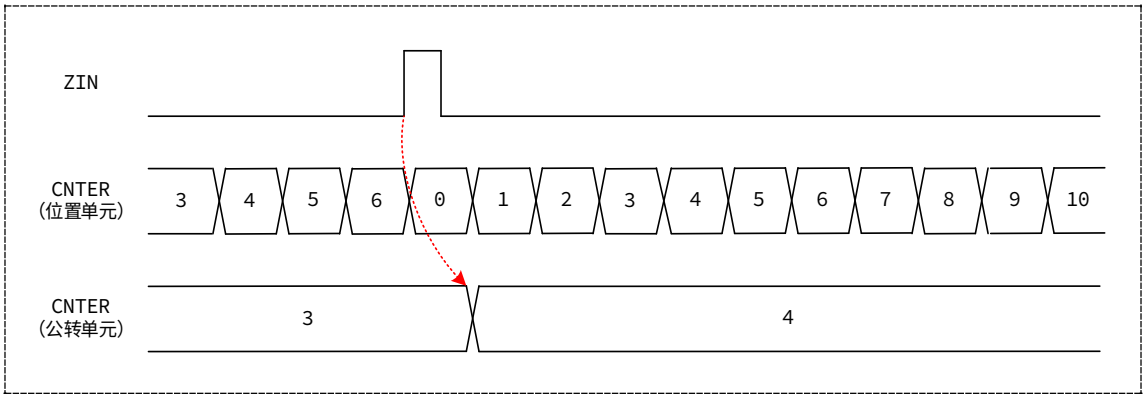


图 16-21 公转模式时 Z 相计数动作

位置溢出计数

位置溢出计数是指位置计数单元计数发生上溢或下溢时，产生一个溢出事件，从而触发公转计数单元的计数器进行一次计数（在该计数方式时 ZIN 的输入不进行公转计数单元的计数动作和位置计数单元的清零动作）。

位置计数单元的溢出事件通过 AOS 模块的联动选通实现公转计数单元计数，即可实现位置溢出计数。公转计数单元的硬件递加（递减）事件选择寄存器（HCUPR 或 HCDOR）的递加（递减）事件选择

Bit16:Bit19 中的 1 位，同时 AOS 模块设定对应的递加（递减）事件的事件源为位置计数单元的计数溢出事件，具体请参考 AOS 章节。如图 16-22 所示。

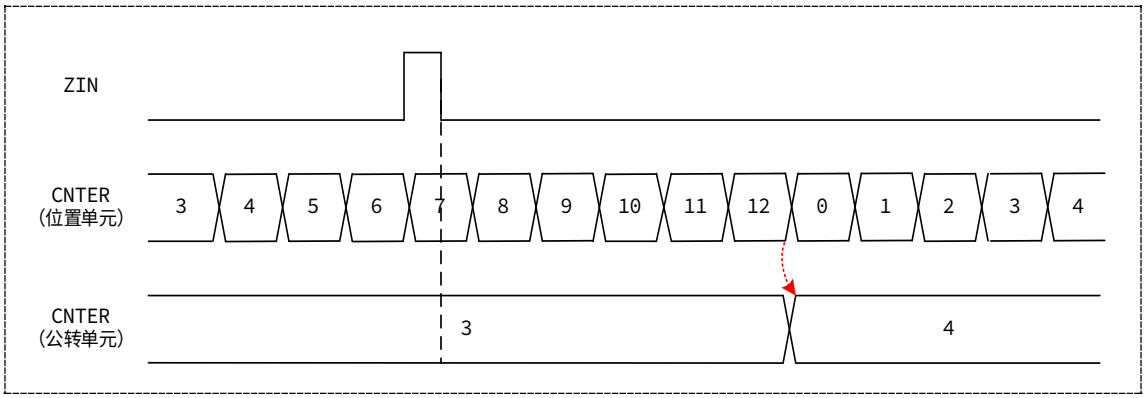


图 16-22 公转模式时位置计数器输出计数动作

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。如图 16-23 所示。

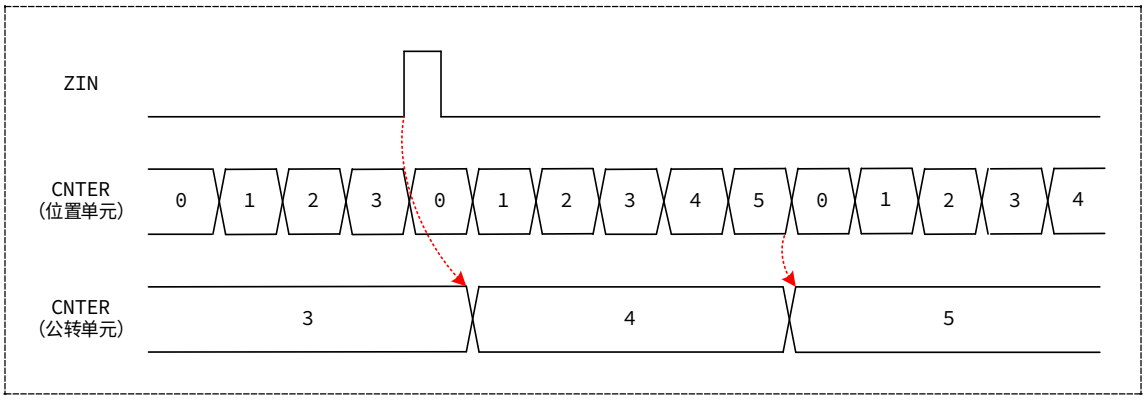


图 16-23 公转模式时 Z 相计数和位置计数器输出混合计数动作

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，可以设定在位置计数器的上溢点或下溢点后的几个周期内（GCONR.ZMSK[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.ZMSK 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 16-24 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。

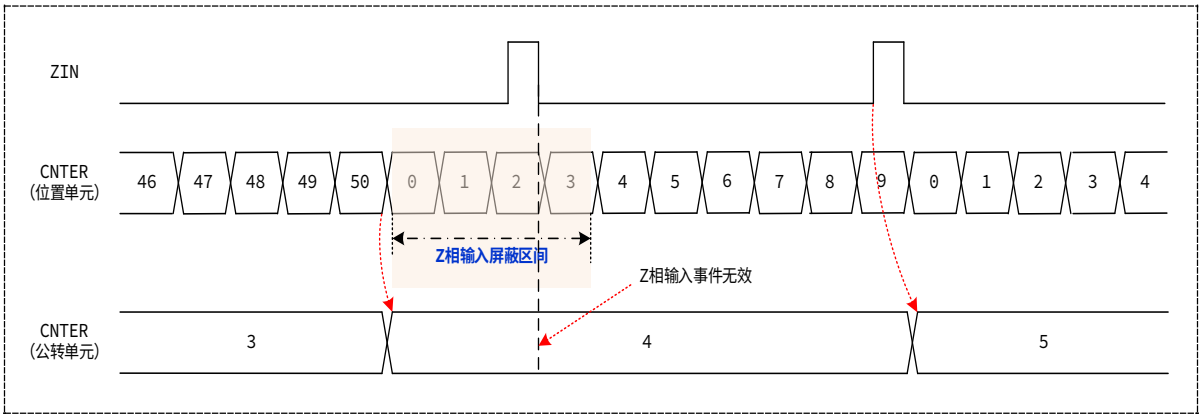


图 16-24 公转计数模式-混合计数 Z 相屏蔽动作例 1

图 16-25 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。

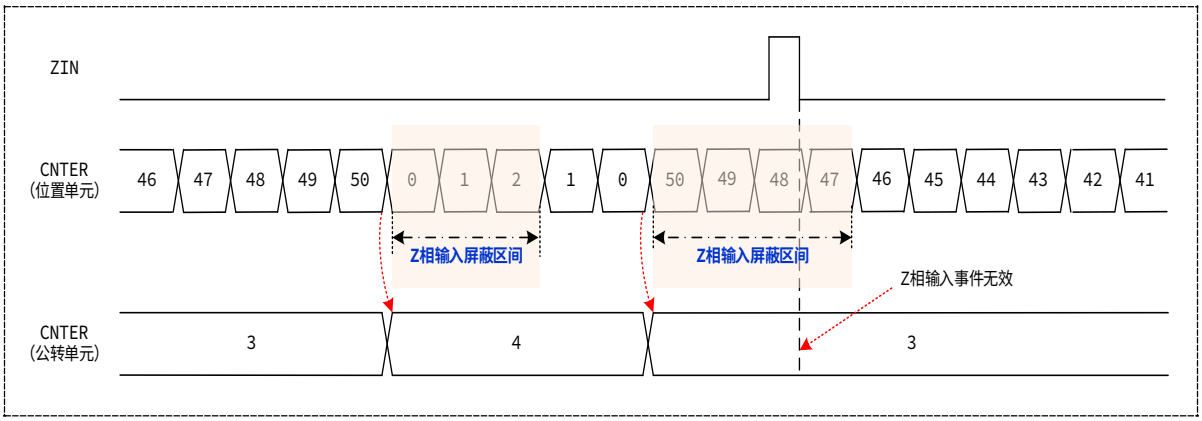


图 16-25 公转计数模式-混合计数 Z 相屏蔽动作例 2

16.2.10 周期间隔响应

Timer4/5/6 的通用比较基准值寄存器（GCMAR~GCMDR），在计数比较匹配时可分别产生专用有效请求信号，送到 AOS 模块中用于和其它模块关联动作。

该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。通过设定有效周期寄存器（VPERR）的 VPERR.PCNTS 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比较基准值寄存器 GCMAR 或 GCMBR 的值相等，也不会输出有效的请求信号。

如果在使用周期间隔响应功能时停止并重新启动定时器，请在停止定时器前配置 VPERR.PCNTE[1:0]=00，否则在重新启动后第一次产生周期间隔有效请求信号的时刻可能出现偏差。

图 16-26 所示是周期间隔有效请求信号的动作例。

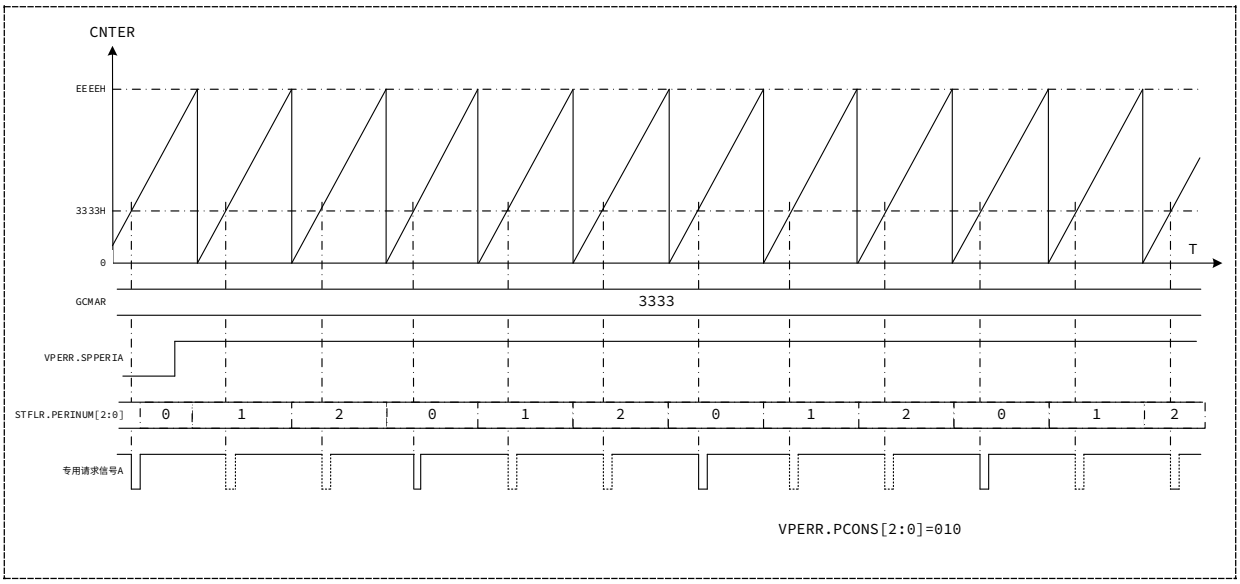


图 16-26 周期间隔有效请求信号动作

16.2.11 保护机制

Advanced Timer 可以对端口的输出状态进行保护控制。

Advanced Timer 有 4 个共用的端口输出无效事件接口，这 4 个接口连接刹车控制模块输出的 4 组刹车事件。每个接口上选通的异常状况事件可从刹车控制设定，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，若监测到从刹车控制过来的刹车事件，则端口的输出状态可变为预先设定好的状态。通用 PWM 输出端口在刹车控制异常事件发生时，端口状态可以变为输出高阻态、输出低电平或输出高电平（PCONR.DISVALA、PCONR.DISVALB 的设定决定）。

例如，若 PCONR.DISSELA[1:0]=01&PCONR.DISVALA=01 设定时，则在 CHxA 端口正常输出期间，若输出无效条件 1 上产生刹车事件，则 CHxA 端口上输出变为高阻态。

16.2.12 中断说明

Timer4/5/6 各含有 3 类共计 9 个中断。分别是 4 个通用计数比较匹配中断（含 2 个捕获输入中断）、2 个计数周期匹配中断、1 个死区时间错误中断。

16.2.12.1 计数比较匹配中断

通用比较基准值寄存器（GCMAR-GCMDR）共计 4 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，状态标志寄存器（STFLR）中的 STFLR.CMAF~STFLR.CMDF 位分别会被置为 1。此时若设定中断控制寄存器（ICONR）的 ICONR.INTENA~ICONR.INTEND 中相应位为 1 使能中断，则对应的中断请求也会被触发。

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 ICONR.INTENA 或 ICONR.INTENB 位为 1 使能中断，则对应的中断请求被触发。

16.2.12.2 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。此时若设置中断控制寄存器（ICONR）的 ICONR.INTENOVF 位与 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。

16.2.12.3 死区时间错误中断

将死区时间基准值寄存器（DTUAR、DTDAR）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 ICONR.INTENDE 位使能中断，则会在该时刻触发死区时间错误中断。

16.2.13 DMA

Timer 支持软件和硬件触发 DMA 进行数据传输。支持数据从其他位置写入定时器，或从定时器读出写入其他位置。可应用于数据捕获后数据的自动搬运和更改周期值或占空比的脉冲宽度的自动调整。每个定时器有两个 DMA 请求，A 请求触发源可选择通用比较捕获 A、C、专用比较 A、计数上溢出，B 请求触发源可选择通用比较捕获 B、D、专用比较 B、计数下溢出。

IDREQ	Interrupt Signal of Peripheral
26	TIM4A
27	TIM4B
28	TIM5A
29	TIM5B
30	TIM6A
31	TIM6B

TIMxA 的 触发源可选择比较捕获 A，C，计数上溢出，专用比较 A

TIMxB 的 触发源可选择比较捕获 B，D，计数下溢出，专用比较 B

16.2.14 刹车保护

当可以设定无效条件 0~3，配置 PCONR.DISVALA, PCONR.DISVALB。无效条件有效时硬件自动将端口状态改变为预设状态（高电平，低电平，高阻态，保持正常输出）。

16.2.14.1 端口刹车与软件刹车

端口经过极性选择控制，有效使能后，经过数字滤波，同步，产生端口刹车标志；端口刹车标志做为 Advanced Timer 的无效条件 3。端口刹车标志需要软件清除。

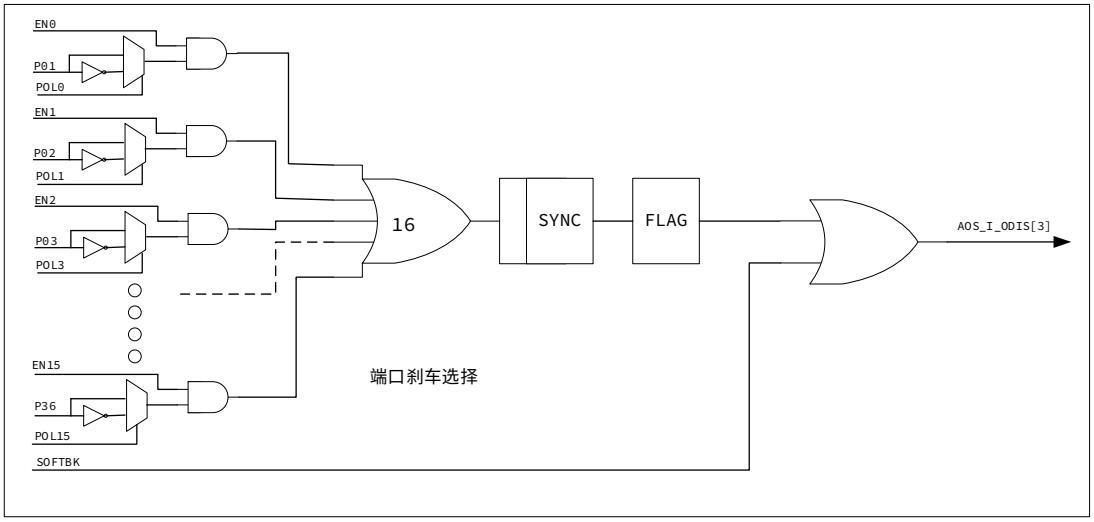


图 16-27 端口刹车与软件刹车示意图

16.2.14.2 低功耗模式自动刹车

系统进入低功耗模式，时钟停止后 PWM 将不能正常工作。低功耗模式作为 Advanced Timer 的无效条件 2 控制 PWM 刹车。

16.2.14.3 输出电平同高同低刹车

输出电平经过电平监测，有效使能后，经过同步，产生同高同低刹车标志；端口刹车标志做为 Advanced Timer 的无效条件 1。同高同低刹车标志需要软件清除。

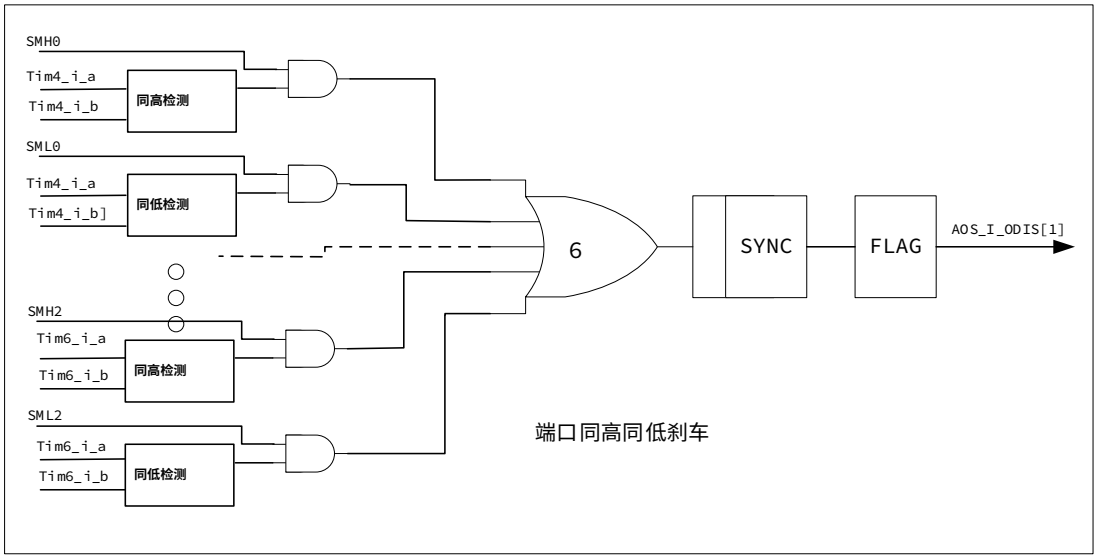


图 16-28 输出同高同低刹车示意图

16.2.14.4 VC 刹车

VC0，VC1，VC2 中断标志经过使能后作为 Advanced Timer 的无效条件 0。

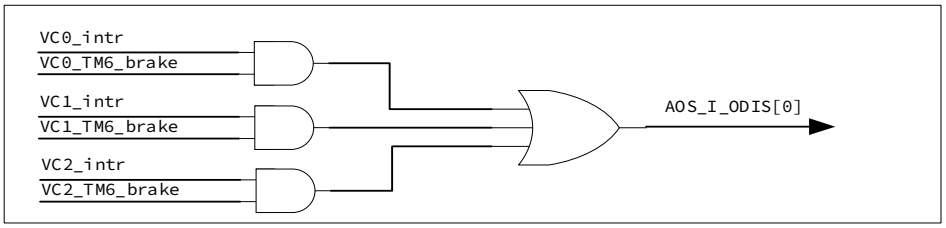


图 16-29 VC 刹车控制示意图

16.2.15 内部互连

16.2.15.1 中断触发输出

由于 Timer4/5/6 的一个中断包含多个中断源。控制触发 ADC 与控制 AOS 的中断信号有单独控制可以选择不同的源，可以选择上溢出、下溢出，4 个比较匹配共 6 个 TIMx 的中断源的任意中断源作为触发条件。

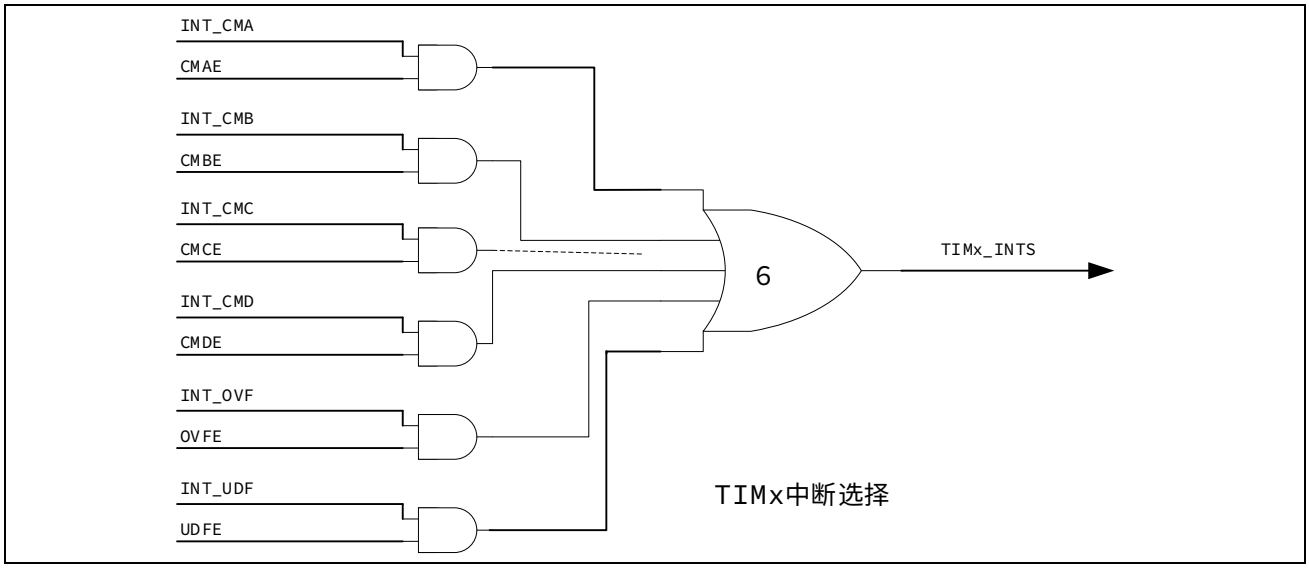


图 16-30 Timer4/5/6 中断选择

16.2.15.2 AOS 触发

AOS 是系统的内部信号，通过选择控制后可以触发 Advanced Timer 的计数器的开始，停止，清零，加 1，减 1 等功能。Advanced Timer 有 4 路 AOS 触发，每路触发可以选择不同模块的中断源。选择后的信号产生单脉冲触发输入到 Advanced Timer，控制 Advanced Timer 的计数器的开始，停止，清零。

Timer4/5/6 内部使用寄存器选择不同的 AOS_I_TRIG 作为自己的触发信号。如可使用 HSTAR 寄存器可以使用中断触发相应定时器的硬件启动。

表 16-3 AOS 源选择

	AOS_i_trig0	AOS_i_trig1	AOS_i_trig2	AOS_i_trig3
选择控制信号	ITRIG.IAOS0S	ITRIG.IAOS1S	ITRIG.IAOS2S	ITRIG.IAOS3S
0000	TIM0_INT	TIM0_INT	TIM0_INT	TIM0_INT
0001	TIM1_INT	TIM1_INT	TIM1_INT	TIM1_INT
0010	TIM2_INT	TIM2_INT	TIM2_INT	TIM2_INT
0011	Reserved	Reserved	Reserved	Reserved
0100	TIM4_INTS	TIM4_INTS	TIM4_INTS	TIM4_INTS
0101	TIM5_INTS	TIM5_INTS	TIM5_INTS	TIM5_INTS
0110	TIM6_INTS	TIM6_INTS	TIM6_INTS	TIM6_INTS
0111	UART0_INT	UART0_INT	UART0_INT	UART0_INT
1000	UART1_INT	UART1_INT	UART1_INT	UART1_INT
1001	LPUART0_INT	LPUART0_INT	LPUART0_INT	LPUART0_INT
1010	VC0_INT	VC0_INT	VC0_INT	VC0_INT
1011	VC1_INT	VC1_INT	VC1_INT	VC1_INT
1100	Reserved	Reserved	Reserved	Reserved
1101	PCA_INT	PCA_INT	PCA_INT	PCA_INT
1110	SPI_INT	SPI_INT	SPI_INT	SPI_INT
1111	ADC_INT	ADC_INT	ADC_INT	ADC_INT

16.2.15.3 端口触发 TRIGA-TRIGD

端口触发可以控制 Advanced Timer 的硬件启动、停止、清零、捕获、计数器加减计数等功能，有数字滤波功能可选，端口可以配置为芯片的任意一个端口上。

表 16-4 端口触发选择

选择控制信号分别独立控制	TRIGA	TRIGB	TRIGC	TRIGD
0000	PA3	PA3	PA3	PA3
0001	PB3	PB3	PB3	PB3
0010	PC3	PC3	PC3	PC3
0011	PD3	PD3	PD3	PD3
0100	PA7	PA7	PA7	PA7
0101	PB7	PB7	PB7	PB7
0110	PC7	PC7	PC7	PC7
0111	PD7	PD7	PD7	PD7
1000	PA11	PA11	PA11	PA11
1001	PB11	PB11	PB11	PB11
1010	PC11	PC11	PC11	PC11
1011	PD1	PD1	PD1	PD1
1100	PA15	PA15	PA15	PA15
1101	PB15	PB15	PB15	PB15
1110	PC5	PC5	PC5	PC5
1111	PD5	PD5	PD5	PD5

16.2.15.4 比较输出 VC 与 Advanced Timer 互连

VC 内部可以互连到 Advanced Timer 的捕获输入端，可以对 VC 输出的边沿进行捕获；VC0 与 CHA 连接，VC1 与 CHB 连接；控制在 VC 控制寄存器。

16.3寄存器描述

CH0 基地址 0x40003000

CH1 基地址 0x40003400

CH2 基地址 0x40003800

表 16-5 Advanced Timer 寄存器列表

寄存器	偏移地址	描述
TIMx_CNTER	0x000	通用计数基准值寄存器
TIMx_PERAR	0x004	通用周期基准值寄存器
TIMx_PERBR	0x008	通用周期基准值缓存寄存器
TIMx_GCMAR	0x010	通用比较 A 基准值寄存器
TIMx_GCMBR	0x014	通用比较 B 基准值寄存器
TIMx_GCMCR	0x018	通用比较 C 基准值寄存器
TIMx_GCMDR	0x01C	通用比较 D 基准值寄存器
TIMx_SCMAR	0x028	专用比较 A 基准值寄存器
TIMx_SCMBR	0x02C	专用比较 B 基准值寄存器
TIMx_DTUAR	0x040	死区时间基准值寄存器
TIMx_DTDAR	0x044	死区时间基准值寄存器
TIMx_GCONR	0x050	通用控制寄存器
TIMx_ICONR	0x054	中断控制寄存器
TIMx_PCONR	0x058	端口控制寄存器
TIMx_BCONR	0x05C	缓存控制寄存器
TIMx_DCONR	0x060	死区控制寄存器
TIMx_FCONR	0x068	滤波控制寄存器
TIMx_VPERR	0x06C	有效周期寄存器
TIMx_STFLR	0x070	状态标志寄存器
TIMx_HSTAR	0x074	硬件启动事件选择寄存器
TIMx_HSTPR	0x078	硬件停止事件选择寄存器
TIMx_HCELR	0x07C	硬件清零事件选择寄存器
TIMx_HCPAR	0x080	硬件捕获事件选择寄存器
TIMx_HCPBR	0x084	硬件捕获事件选择寄存器
TIMx_HCUPR	0x088	硬件递减事件选择寄存器
TIMx_HCDOR	0x08C	硬件递减事件选择寄存器
TIMx_IFR	0x100	中断标志寄存器
TIMx_ICLR	0x104	中断清除寄存器
TIMx_CR	0x108	展频及中断触发选择寄存器
TIMx_AOSSR	0x110	AOS 选择寄存器，三个通道共用

寄存器	偏移地址	描述
TIMx_AOSCL	0x114	AOS 刹车标志清除寄存器，三个通道共用
TIMx_PTBKS	0x118	端口刹车控制寄存器，三个通道共用
TIMx_TTRIG	0x11C	端口触发控制寄存器，三个通道共用
TIMx_ITRIG	0x120	AOS 触发控制寄存器，三个通道共用
TIMx_PTBKP	0x124	端口刹车极性控制寄存器，三个通道共用
TIMx_SSTAR	0x3F4	软件同步启动寄存器
TIMx_SSTPR	0x3F8	软件同步停止寄存器
TIMx_SCLRR	0x3FC	软件同步清零寄存器

16.3.1 通用计数基准值寄存器（TIMx_CNTER）

地址偏移量：0x000

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	符号	功能描述
31:16	Reserved	-
15:0	CNT[15:0]	当前计数器的计数值

16.3.2 通用周期基准值寄存器（TIMx_PERAR）

地址偏移量：0x004

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	符号	功能描述
31:16	Reserved	-
15:0	PERA[15:0]	计数周期值，设定每轮计数的计数周期值

16.3.3 通用周期缓存寄存器（TIMx_PERBR）

地址偏移量：0x008

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	符号	功能描述
31:16	Reserved	-
15:0	PERB[15:0]	缓存计数周期值，计数周期的缓存值

16.3.4 通用比较基准值寄存器（TIMx_GCMAR-GCMDR）

地址偏移量：0x0010, 0x0014, 0x0018, 0x001C

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GCMA-D [15:0]															
RW															

位	符号	功能描述
31:16	Reserved	-
15:0	GCMA-D [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

16.3.5 专用比较基准值寄存器（TIMx_SCMAR-SCMBR）

地址偏移量：0x0028, 0x002C

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCMA-B[15:0]															
RW															

位	符号	功能描述
31:16	Reserved	-
15:0	SCMA-B [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

16.3.6 死区时间基准值寄存器（TIMx_DTUAR- DTDAR）

地址偏移量：0x040, 0x044

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTUA/DTDA [15:0]															
RW															

位	符号	功能描述
31:16	Reserved	-
15:0	DTUA/DA [15:0]	死区时间值，死区时间设定值

16.3.7 通用控制寄存器（TIMx_GCONR）

地址偏移量：0x050

复位值: 0x00000100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												ZMSK[1:0]		ZMSK POS	ZMSK REV
												RW		RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							DIR	Res.	CKDIV[2:0]			MODE[2:0]			START
							RW		RW			RW			RW

位	标记	功能描述
31:20	Reserved	-
19:18	ZMSK[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
17	ZMSKPOS	Z相输入位置计数器选择 0：Z相输入时该定时器作为位置计数器，在屏蔽周期期间内位置计数器清零功能正常动作 1：Z相输入时该定时器作为位置计数器，在屏蔽周期期间内位置计数器清零功能被屏蔽
16	ZMSKREV	Z相输入公转计数器选择 0：Z相输入时该定时器作为公转计数器，在屏蔽周期期间内公转计数器计数功能正常动作 1：Z相输入时该定时器作为公转计数器，在屏蔽周期期间内公转计数器计数功能被屏蔽
15:9	Reserved	-
8	DIR	计数方向 0：递减计数；1：递加计数
7	Reserved	-
6:4	CKDIV[2:0]	计数时钟选择 000：PCLK0　001：PCLK0/2　010：PCLK0/4　011：PCLK0/8 100：PCLK0/16　101：PCLK0/64　110：PCLK0/256　111：PCLK0/1024
3:1	MODE[2:0]	计数模式 000：锯齿波A模式　100：三角波A模式　101：三角波B模式 请不要设定其它值
0	START	计数器启动　0：计数器关闭；1：计数器启动 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*

16.3.8 中断控制寄存器（TIMx_ICONR）

地址偏移量：0x054

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												INTENSBD	INTENSBU	INTENSAD	INTENSAU
												RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSAMH	INTENSAML	Reserved					INTENDE	INTENUDF	INTENOVF	Reserved		INTEND	INTENC	INTENB	INTENA
RW	RW						RW	RW	RW			RW	RW	RW	RW

位	标记	功能
31:20	Reserved	-
19	INTENSBD	专用向下计数触发ADC使能B
18	INTENSBU	专用向上计数触发ADC使能B
17	INTENSAD	专用向下计数触发ADC使能A
16	INTENSAU	专用向上计数触发ADC使能A
15	INTENSAMH	同高中断使能
14	INTENSAML	同低中断使能
13:9	Reserved	-
8	INTENDE	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
7	INTENUDF	下溢中断使能 0：锯齿波时下溢发生或三角波时计数到谷点，该中断无效 1：锯齿波时下溢发生或三角波时计数到谷点，该中断使能
6	INTENOVF	上溢中断使能 0：锯齿波时上溢发生或三角波时计数到峰点，该中断无效 1：锯齿波时上溢发生或三角波时计数到峰点，该中断使能
5:4	Reserved	-
3	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
2	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
1	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
0	INTENA	计数匹配中断使能A 0：GCMAR 寄存器与计数值相等时，或者发生捕获输入事件时，该中断无效 1：GCMAR 寄存器与计数值相等时，或者发生捕获输入事件时，该中断使能

16.3.9 端口控制寄存器（TIMx_PCONR）

地址偏移量：0x058

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			DISVALB		DISSELB		OUTENB	PERCB		CMPCB		STASTPSB	STPCB	STACB	CAPCB
			RW		RW		RW	RW		RW		RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DISVALA		DISSELA		OUTENA	PERCA		CMPCA		STASTPSA	STPCA	STACA	CAPCA
			RW		RW		RW	RW		RW		RW	RW	RW	RW

位	标记	功能
31:29	Reserved	-
28:27	DISVALB	CHxB 输出状态控制 00：强制输出无效条件0~3中被选择的条件成立时，CHxB 端口正常输出 01：强制输出无效条件0~3中被选择的条件成立时，CHxB 端口输出高阻态 10：强制输出无效条件0~3中被选择的条件成立时，CHxB 端口输出低电平 11：强制输出无效条件0~3中被选择的条件成立时，CHxB 端口输出高电平
26:25	DISSELB	强制输出无效条件选择B 00：选择强制输出无效条件0；01：选择强制输出无效条件1 10：选择强制输出无效条件2；11：选择强制输出无效条件3
24	OUTENB	输出使能B 0：Advanced Timer 功能时的 CHxB 端口输出无效 1：Advanced Timer 功能时的 CHxB 端口输出有效
23:22	PERCB	周期值匹配时端口状态设定B 00：计数器计数值与周期值相等时，CHxB 端口输出保持为低电平 01：计数器计数值与周期值相等时，CHxB 端口输出设定为高电平 10：计数器计数值与周期值相等时，CHxB 端口输出设定为先前状态 11：计数器计数值与周期值相等时，CHxB 端口输出设定为反转电平
21:20	CMPCB	比较值匹配时端口状态设定B 00：计数器计数值与 GCMBR 相等时，CHxB 端口输出保持为低电平 01：计数器计数值与 GCMBR 相等时，CHxB 端口输出设定为高电平 10：计数器计数值与 GCMBR 相等时，CHxB 端口输出设定为先前状态 11：计数器计数值与 GCMBR 相等时，CHxB 端口输出设定为反转电平
19	STASTPSB	计数开始停止端口状态选择B 0：计数开始或停止时，CHxB 端口输出由 STACB、STPCB 决定 1：计数开始或停止时，CHxB 端口输出设定为先前状态 注：此处的计数开始是指初始计数开始或停止再开始；计数停止是指初始时停止或计数开始后再停止

18	STPCB	计数停止端口状态设定B 0：计数停止时，CHxB 端口输出设定为低电平 1：计数停止时，CHxB 端口输出设定为高电平
17	STACB	计数开始端口状态设定B 0：计数开始时，CHxB 端口输出设定为低电平 1：计数开始时，CHxB 端口输出设定为高电平
16	CAPCB	功能模式选择B 0：比较输出功能；1：捕获输入功能
15:13	Reserved	-
12:11	DISVALA	CHxA 输出状态控制 00：强制输出无效条件0~3中被选择的条件成立时，CHxA 端口正常输出 01：强制输出无效条件0~3中被选择的条件成立时，CHxA 端口输出高阻态 10：强制输出无效条件0~3中被选择的条件成立时，CHxA 端口输出低电平 11：强制输出无效条件0~3中被选择的条件成立时，CHxA 端口输出高电平
10:9	DISSELA	强制输出无效条件选择A 00：选择强制输出无效条件0；01：选择强制输出无效条件1 10：选择强制输出无效条件2；11：选择强制输出无效条件3
8	OUTENA	输出使能A 0：Advanced Timer 功能时的 CHxA 端口输出无效 1：Advanced Timer 功能时的 CHxA 端口输出有效
7:6	PERCA	周期值匹配时端口状态设定A 00：计数器计数值与周期值相等时，CHxA 端口输出保持为低电平 01：计数器计数值与周期值相等时，CHxA 端口输出设定为高电平 10：计数器计数值与周期值相等时，CHxA 端口输出设定为先前状态 11：计数器计数值与周期值相等时，CHxA 端口输出设定为反转电平
5:4	CMPCA	比较值匹配时端口状态设定A 00：计数器计数值与 GCMAR 相等时，CHxA 端口输出保持为低电平 01：计数器计数值与 GCMAR 相等时，CHxA 端口输出设定为高电平 10：计数器计数值与 GCMAR 相等时，CHxA 端口输出设定为先前状态 11：计数器计数值与 GCMAR 相等时，CHxA 端口输出设定为反转电平
3	STASTPSA	计数开始停止端口状态选择A 0：计数开始或停止时，CHxA 端口输出由 STACA、STPCA 决定 1：计数开始或停止时，CHxA 端口输出设定为先前状态 注：此处的计数开始是指初始计数开始或停止再开始；计数停止是指初始时停止或计数开始后再停止
2	STPCA	计数停止端口状态设定A 0：计数停止时，CHxA 端口输出设定为低电平； 1：计数停止时，CHxA 端口输出设定为高电平
1	STACA	计数开始端口状态设定A 0：计数开始时，CHxA 端口输出设定为低电平

		1：计数开始时，CHxA 端口输出设定为高电平
0	CAPCA	功能模式选择A 0：比较输出功能；1：捕获输入功能

16.3.10 缓存控制寄存器（TIMx_BCONR）

地址偏移量：0x05C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							BENP	Reserved					BENB	Res.	BENA
							RW						RW		RW

位	标记	功能
31:9	Reserved	-
8	BENP	周期值缓存传送 0：缓存传送无效 1：缓存传送使能（PERBR->PERAR）
7:3	Reserved	-
2	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 比较输出功能时：（GCMDR->GCMBR）；捕获输入功能时：（GCMBR->GCMDR）
1	Reserved	-
0	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 比较输出功能时：（GCMCR->GCMAR）；捕获输入功能时：（GCMAR->GCMCR）

16.3.11 死区控制寄存器（TIMx_DCONR）

地址偏移量：0x060

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							SEPA	Reserved							DTCEN
							RW								RW

位	标记	功能
31:9	Reserved	-
8	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
7:1	Reserved	-
0	DTCEN	死区功能 0：死区功能无效 1：死区功能有效

16.3.12 滤波控制寄存器（TIMx_FCONR）

地址偏移量：0x068

复位值: 0x0000 0000

31	30 29	28	27	26 25	24	23	22 21	20	19	18 17	16
Res.	NOFICKTD	NOFIENTD	Res.	NOFICKTC	NOFIENTC	Res.	NOFICKTB	NOFIENTB	Res.	NOFICKTA	NOFIENTA
	RW	RW		RW	RW		RW	RW		RW	RW

15 14 13 12 11 10 9 8 7	6 5	4	3	2 1	0
Reserved	NOFICKGB	NOFIENGB	Res.	NOFICKGA	NOFIENGA
	RW	RW		RW	RW

位	标记	功能
31	Reserved	-
30:29	NOFICKTD	TRID 端口滤波采样基准时钟选择 00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64
28	NOFIENTD	TRID 端口捕获输入滤波使能，0无效；1使能
27	Reserved	-
26:25	NOFICKTC	TRIC 端口滤波采样基准时钟选择 00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64
24	NOFIENTC	TRIC 端口捕获输入滤波使能，0无效；1使能
23	Reserved	-
22:21	NOFICKTB	TRIB 端口滤波采样基准时钟选择 00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64
20	NOFIENTB	TRIB 端口捕获输入滤波使能，0无效；1使能
19	Reserved	-
18:17	NOFICKTA	TRIA 端口滤波采样基准时钟选择 00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64
16	NOFIENTA	TRIA 端口捕获输入滤波使能，0无效；1使能
15:7	Reserved	-
6:5	NOFICKGB	CHxIB 端口滤波采样基准时钟选择 00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64
4	NOFIENGB	CHxIB 端口捕获输入滤波使能，0无效；1使能
3	Reserved	-
2:1	NOFICKGA	CHxIA 端口滤波采样基准时钟选择 00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64
0	NOFIENGA	CHxIA 端口捕获输入滤波使能，0无效；1使能

注意：

- TRIGA-D 滤波设置只有在 TIM4 中设置有效，在 Timer5/6 设置无效。

16.3.13 有效周期寄存器（TIMx_VPERR）

地址偏移量：0x06C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											PCNTS			PCNTE	
											RW			RW	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												GEPERID	GEPERIC	GEPERIB	GEPERIA
												RW	RW	RW	RW

位	标记	功能
31:21	Reserved	-
20:18	PCNTS	有效周期选择 000：有效周期选择功能无效 001：每隔1个周期有效一次 010：每隔2个周期有效一次 011：每隔3个周期有效一次 100：每隔4个周期有效一次 101：每隔5个周期有效一次 110：每隔6个周期有效一次 111：每隔7个周期有效一次
17:16	PCNTE	有效周期计数条件选择 00：有效周期选择功能无效 01：锯齿波计数上、下溢点或三角波波谷做为计数条件 10：锯齿波计数上、下溢点或三角波波峰做为计数条件 11：锯齿波计数上、下溢点或三角波波谷、波峰做为计数条件
15:4	Reserved	-
3	GEPERID	通用信号有效周期选择D 0：有效周期选择功能无效；1：有效周期选择功能使能
2	GEPERIC	通用信号有效周期选择C 0：有效周期选择功能无效；1：有效周期选择功能使能
1	GEPERIB	通用信号有效周期选择B 0：有效周期选择功能无效；1：有效周期选择功能使能
0	GEPERIA	通用信号有效周期选择A 0：有效周期选择功能无效；1：有效周期选择功能使能

16.3.14 状态标志寄存器（TIMx_STFLR）

地址偏移量：0x070

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIRF	Reserved							VPERNUM			Reserved				
R								R							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CMSBDF	CMSBUF	CMSADF	CMSAUF	DTEF	UDFF	OVFF	Reserved		CMDF	CMCF	CMBF	CMAF
			RW	RW	RW	RW	RW	RW	RW			RW	RW	RW	RW

位	标记	功能
31	DIRF	计数方向 0：递减计数 1：递加计数
30:24	Reserved	-
23:21	VPERNUM	周期次数 有效周期选择功能使能时，计数后的周期次数
20:13	Reserved	-
12	CMSBDF	向下计数专用比较基准值匹配B
11	CMSBUF	向上计数专用比较基准值匹配B
10	CMSADF	向下计数专用比较基准值匹配A
9	CMSAUF	向上计数专用比较基准值匹配A
8	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
7	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
6	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
5:4	Reserved	-
3	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
2	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
1	CMBF	计数匹配B 0：GCMBR 寄存器的值与计数值不相等，且未发生 CHxB 捕获完成动作 1：GCMBR 寄存器的值与计数值相等，或发生 CHxB 捕获完成动作

0	CMAF	计数匹配A 0：GCMAR 寄存器的值与计数值不相等，且未发生 CHxA 捕获完成动作 1：GCMAR 寄存器的值与计数值相等，或发生 CHxA 捕获完成动作

16.3.15 硬件启动事件选择寄存器（TIMx_HSTAR）

地址偏移量：0x074

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STARTS	Reserved														
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSTA15	HSTA14	HSTA13	HSTA12	HSTA11	HSTA10	HSTA9	HSTA8	HSTA7	HSTA6	HSTA5	HSTA4	HSTA3	HSTA2	HSTA1	HSTA0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能
31	STARTS	硬件启动使能 0：硬件启动无效 1：硬件启动有效 注：硬件启动有效时，*SSTAR* 的设定无效
30:16	Reserved	-
15	HSTA15	硬件启动条件15：TIMTRID 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
14	HSTA14	硬件启动条件14：TIMTRID 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
13	HSTA13	硬件启动条件13：TIMTRIC 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
12	HSTA12	硬件启动条件12：TIMTRIC 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
11	HSTA11	硬件启动条件11：TIMTRIB 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
10	HSTA10	硬件启动条件10：TIMTRIB 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
9	HSTA9	硬件启动条件9：TIMTRIA 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

8	HSTA8	硬件启动条件8：TIMTRIA 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
7	HSTA7	硬件启动条件7：CHxB 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
6	HSTA6	硬件启动条件6：CHxB 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
5	HSTA5	硬件启动条件5：CHxA 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
4	HSTA4	硬件启动条件4：CHxA 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
3	HSTA3	硬件启动条件3：从 AOS 来的事件触发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
2	HSTA2	硬件启动条件2：从 AOS 来的事件触发2有效 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效
1	HSTA1	硬件启动条件1：从 AOS 来的事件触发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
0	HSTA0	硬件启动条件0：从 AOS 来的事件触发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

16.3.16 硬件停止事件选择寄存器（TIMx_HSTPR）

地址偏移量：0x078

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STOPS	Reserved														
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSTP15	HSTA14	HSTP13	HSTP12	HSTP11	HSTP10	HSTP9	HSTP8	HSTP7	HSTP6	HSTP5	HSTP4	HSTP3	HSTP2	HSTP1	HSTP0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能
31	STOPS	硬件停止使能 0：硬件停止无效 1：硬件停止有效 注：硬件停止有效时，软件停止的设定无效
30:16	Reserved	-
15	HSTP15	硬件停止条件15：TIMTRID 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
14	HSTP14	硬件停止条件14：TIMTRID 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
13	HSTP13	硬件停止条件13：TIMTRIC 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
12	HSTP12	硬件停止条件12：TIMTRIC 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
11	HSTP11	硬件停止条件11：TIMTRIB 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
10	HSTP10	硬件停止条件10：TIMTRIB 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
9	HSTP9	硬件停止条件9：TIMTRIA 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

8	HSTP8	硬件停止条件8：TIMTRIA 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
7	HSTP7	硬件停止条件7：CHxB 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
6	HSTP6	硬件停止条件6：CHxB 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
5	HSTP5	硬件停止条件5：CHxA 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
4	HSTP4	硬件停止条件4：CHxA 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
3	HSTP3	硬件停止条件3：从 AOS 来的事件触发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
2	HSTP2	硬件停止条件2：从 AOS 来的事件触发2有效 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效
1	HSTP1	硬件停止条件1：从 AOS 来的事件触发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
0	HSTP0	硬件停止条件0：从 AOS 来的事件触发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

16.3.17 硬件清零事件选择寄存器（TIMx_HCELR）

地址偏移量：0x07C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLEARS	Reserved														
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCEL15	HCEL14	HCEL13	HCEL12	HCEL11	HCEL10	HCEL9	HCEL8	HCEL7	HCEL6	HCEL5	HCEL4	HCEL3	HCEL2	HCEL1	HCEL0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能
31	CLEARS	硬件清零使能 0：硬件清零无效 1：硬件清零有效 注：硬件清零有效时，软件清零的设定无效
30:16	Reserved	-
15	HCEL15	硬件清零条件15：TIMTRID 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
14	HCEL14	硬件清零条件14：TIMTRID 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
13	HCEL13	硬件清零条件13：TIMTRIC 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
12	HCEL12	硬件清零条件12：TIMTRIC 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
11	HCEL11	硬件清零条件11：TIMTRIB 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
10	HCEL10	硬件清零条件10：TIMTRIB 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
9	HCEL9	硬件清零条件9：TIMTRIA 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

8	HCEL8	硬件清零条件8：TIMTRIA 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
7	HCEL7	硬件清零条件7：CHxB 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
6	HCEL6	硬件清零条件6：CHxB 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
5	HCEL5	硬件清零条件5：CHxA 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
4	HCEL4	硬件清零条件4：CHxA 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
3	HCEL3	硬件清零条件3：从 AOS 来的事件触发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
2	HCEL2	硬件清零条件2：从 AOS 来的事件触发2有效 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效
1	HCEL1	硬件清零条件1：从 AOS 来的事件触发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
0	HCEL0	硬件清零条件0：从 AOS 来的事件触发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

16.3.18 硬件捕获 A 事件选择寄存器（TIMx_HCPAR）

地址偏移量：0x080

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCPA15	HCPA14	HCPA13	HCPA12	HCPA11	HCPA10	HCPA9	HCPA8	HCPA7	HCPA6	HCPA5	HCPA4	HCPA3	HCPA2	HCPA1	HCPA0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能
31:16	Reserved	-
15	HCPA15	硬件捕获A条件15：TIMTRID 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
14	HCPA14	硬件捕获A条件14：TIMTRID 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
13	HCPA13	硬件捕获A条件13：TIMTRIC 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
12	HCPA12	硬件捕获A条件12：TIMTRIC 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
11	HCPA11	硬件捕获A条件11：TIMTRIB 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
10	HCPA10	硬件捕获A条件10：TIMTRIB 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
9	HCPA9	硬件捕获A条件9：TIMTRIA 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
8	HCPA8	硬件捕获A条件8：TIMTRIA 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
7	HCPA7	硬件捕获A条件7：CHxB 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
6	HCPA6	硬件捕获A条件6：CHxB 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
5	HCPA5	硬件捕获A条件5：CHxA 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
4	HCPA4	硬件捕获A条件4：CHxA 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
3	HCPA3	硬件捕获A条件3：从 AOS 来的事件触发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
2	HCPA2	硬件捕获A条件2：从 AOS 来的事件触发2有效 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效
1	HCPA1	硬件捕获A条件1：从 AOS 来的事件触发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
0	HCPA0	硬件捕获A条件0：从 AOS 来的事件触发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

16.3.19 硬件捕获 B 事件选择寄存器（TIMx_HCPBR）

地址偏移量：0x084

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCPB15	HCPB14	HCPB13	HCPB12	HCPB11	HCPB10	HCPB9	HCPB8	HCPB7	HCPB6	HCPB5	HCPB4	HCPB3	HCPB2	HCPB1	HCPB0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能
31:16	Reserved	-
15	HCPB15	硬件捕获B条件15：TIMTRID 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
14	HCPB14	硬件捕获B条件14：TIMTRID 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
13	HCPB13	硬件捕获B条件13：TIMTRIC 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
12	HCPB12	硬件捕获B条件12：TIMTRIC 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
11	HCPB11	硬件捕获B条件11：TIMTRIB 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
10	HCPB10	硬件捕获B条件10：TIMTRIB 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
9	HCPB9	硬件捕获B条件9：TIMTRIA 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
8	HCPB8	硬件捕获B条件8：TIMTRIA 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
7	HCPB7	硬件捕获B条件7：CHxB 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
6	HCPB6	硬件捕获B条件6：CHxB 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
5	HCPB5	硬件捕获B条件5：CHxA 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
4	HCPB4	硬件捕获B条件4：CHxA 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
3	HCPB3	硬件捕获B条件3：从 AOS 来的事件触发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
2	HCPB2	硬件捕获B条件2：从 AOS 来的事件触发2有效 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效
1	HCPB1	硬件捕获B条件1：从 AOS 来的事件触发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
0	HCPB0	硬件捕获B条件0：从 AOS 来的事件触发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

16.3.20 硬件递加事件选择寄存器（TIMx_HCUPR）

地址偏移量：0x088

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												HCUP19	HCUP18	HCUP17	HCUP16
												RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCUP15	HCUP14	HCUP13	HCUP12	HCUP11	HCUP10	HCUP9	HCUP8	HCUP7	HCUP6	HCUP5	HCUP4	HCUP3	HCUP2	HCUP1	HCUP0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能
31:20	Reserved	-
19	HCUP19	硬件递加条件：从 AOS 来的事件触发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
18	HCUP18	硬件递加条件：从 AOS 来的事件触发2有效 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
17	HCUP17	硬件递加条件：从 AOS 来的事件触发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
16	HCUP16	硬件递加条件：从 AOS 来的事件触发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
15	HCUP15	硬件递加条件：TIMTRID 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
14	HCUP14	硬件递加条件：TIMTRID 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
13	HCUP13	硬件递加条件：TIMTRIC 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
12	HCUP12	硬件递加条件：TIMTRIC 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

11	HCUP11	硬件递加条件：TIMTRIB 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
10	HCUP10	硬件递加条件：TIMTRIB 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
9	HCUP9	硬件递加条件：TIMTRIA 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
8	HCUP8	硬件递加条件：TIMTRIA 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
7	HCUP7	硬件递加条件：CHxB 端口为高电平时，CHxA 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
6	HCUP6	硬件递加条件：CHxB 端口为高电平时，CHxA 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
5	HCUP5	硬件递加条件：CHxB 端口为低电平时，CHxA 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
4	HCUP4	硬件递加条件：CHxB 端口为低电平时，CHxA 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
3	HCUP3	硬件递加条件：CHxA 端口为高电平时，CHxB 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
2	HCUP2	硬件递加条件：CHxA 端口为高电平时，CHxB 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
1	HCUP1	硬件递加条件：CHxA 端口为低电平时，CHxB 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
0	HCUP0	硬件递加条件：CHxA 端口为低电平时，CHxB 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

16.3.21 硬件递减事件选择寄存器（TIMx_HCDOR）

地址偏移量：0x08C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												HCDO19	HCDO18	HCDO17	HCDO16
												RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCDO15	HCDO14	HCDO13	HCDO12	HCDO11	HCDO10	HCDO9	HCDO8	HCDO7	HCDO6	HCDO5	HCDO4	HCDO3	HCDO2	HCDO1	HCDO0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能
31:20	Reserved	-
19	HCDO19	硬件递减条件：从 AOS 来的事件触发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
18	HCDO18	硬件递减条件：从 AOS 来的事件触发2有效 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
17	HCDO17	硬件递减条件：从 AOS 来的事件触发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
16	HCDO16	硬件递减条件：从 AOS 来的事件触发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
15	HCDO15	硬件递减条件：TIMTRID 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
14	HCDO14	硬件递减条件：TIMTRID 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
13	HCDO13	硬件递减条件：TIMTRIC 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
12	HCDO12	硬件递减条件：TIMTRIC 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

11	HCDO11	硬件递减条件：TIMTRIB 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
10	HCDO10	硬件递减条件：TIMTRIB 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
9	HCDO9	硬件递减条件：TIMTRIA 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
8	HCDO8	硬件递减条件：TIMTRIA 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
7	HCDO7	硬件递减条件：CHxB 端口为高电平时，CHxA 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
6	HCDO6	硬件递减条件：CHxB 端口为高电平时，CHxA 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
5	HCDO5	硬件递减条件：CHxB 端口为低电平时，CHxA 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
4	HCDO4	硬件递减条件：CHxB 端口为低电平时，CHxA 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
3	HCDO3	硬件递减条件：CHxA 端口为高电平时，CHxB 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
2	HCDO2	硬件递减条件：CHxA 端口为高电平时，CHxB 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
1	HCDO1	硬件递减条件：CHxA 端口为低电平时，CHxB 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
0	HCDO0	硬件递减条件：CHxA 端口为低电平时，CHxB 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

16.3.22 软件同步启动寄存器（TIMx_SSTAR）

地址偏移量：0x3F4

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													SSTA2	SSTA1	SSTA0
													RW	RW	RW

位	标记	功能
31:3	Reserved	
2	SSTA2	Timer6 软件启动 0：软件启动无效 1：软件启动使能
1	SSTA1	Timer5 软件启动 0：软件启动无效 1：软件启动使能
0	SSTA0	Timer4 软件启动 0：软件启动无效 1：软件启动使能

16.3.23 软件同步停止寄存器（TIMx_SSTPR）

地址偏移量：0x3F8

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													SSTP2	SSTP1	SSTP0
													RW	RW	RW

位	标记	功能
31:3	Reserved	
2	SSTP2	Timer6 软件停止 0：软件停止无效 1：软件停止使能
1	SSTP1	Timer5 软件停止 0：软件停止无效 1：软件停止使能
0	SSTP0	Timer4 软件停止 0：软件停止无效 1：软件停止使能

16.3.24 软件同步清零寄存器（TIMx_SCLRR）

地址偏移量：0x3FC

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													SCLR2	SCLR1	SCLR0
													RW	RW	RW

位	标记	功能
31:3	Reserved	
2	SCLR2	Timer6 软件清零 0：软件清零无效 1：软件清零使能
1	SCLR1	Timer5 软件清零 0：软件清零无效 1：软件清零使能
0	SCLR0	Timer4 软件清零 0：软件清零无效 1：软件清零使能

16.3.25 中断标志寄存器（TIMx_IFR）

地址偏移量：0x100

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAMHF	SAMLF	Reserved					DTEF	UDFF	OVFF	Reserved		CMDF	CMCF	CMBF	CMAF
RO	RO						RO	RO	RO			RO	RO	RO	RO

位	标记	功能
31:16	Reserved	-
15	SAMHF	CHxA/B 端口高状态中断标志 0：CHxA 和 CHxB 端口上没有同时出现高电平 1：CHxA 和 CHxB 端口上同时出现高电平
14	SAMLF	CHxA/B 端口低状态中断标志 0：CHxA 和 CHxB 端口上没有同时出现低电平 1：CHxA 和 CHxB 端口上同时出现低电平
13:9	Reserved	-
8	DTEF	死区时间错误中断标志 0：未发生死区时间错误；1：发生死区时间错误
7	UDFF	下溢匹配中断标志 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
6	OVFF	上溢匹配中断标志 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
5:4	Reserved	-
3	CMDF	计数匹配D中断标志 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
2	CMCF	计数匹配C中断标志 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
1	CMBF	计数匹配B中断标志 0：GCMBR 寄存器的值与计数值不相等，且未发生 CHxB 捕获完成动作 1：GCMBR 寄存器的值与计数值相等，或发生 CHxB 捕获完成动作
0	CMAF	计数匹配A中断标志 0：GCMAR 寄存器的值与计数值不相等，且未发生 CHxA 捕获完成动作 1：GCMAR 寄存器的值与计数值相等，或发生 CHxA 捕获完成动作

16.3.26 中断标志清除寄存器（TIMx_ICLR）

地址偏移量：0x104

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAMHC	SAMLC	Reserved					DTEC	UDFC	OVFC	Reserved		CMDC	CMCC	CMBC	CMAC
R1W0	R1W0						R1W0	R1W0	R1W0			R1W0	R1W0	R1W0	R1W0

位	标记	功能
31:16	Reserved	-
15	SAMHC	CHxA/B 端口高状态中断标志清除，写1无效，写0清除对应中断
14	SAMLC	CHxA/B 端口低状态中断标志清除，写1无效，写0清除对应中断
13:9	Reserved	-
8	DTEC	死区时间错误中断标志清除，写1无效，写0清除对应中断
7	UDFC	下溢匹配中断标志清除，写1无效，写0清除对应中断
6	OVFC	上溢匹配中断标志清除，写1无效，写0清除对应中断
5:4	Reserved	-
3	CMDC	计数匹配D中断标志清除，写1无效，写0清除对应中断
2	CMCC	计数匹配C中断标志清除，写1无效，写0清除对应中断
1	CMBC	计数匹配B中断标志清除，写1无效，写0清除对应中断
0	CMAC	计数匹配A中断标志清除，写1无效，写0清除对应中断

16.3.27 展频及中断触发选择（TIMx_CR）

地址偏移量：0x108

复位值: 0x0000 0300

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									Dma_s_cmb	Dma_s_cma	Dma_g_udf	Dma_g_ovf	Reserved		Dma_g_cmd
									RW	RW	RW	RW			RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dma_g_cmc	Dma_g_cmb	Dma_g_cma	CMSBE	CMSAE	DITENS	DITENB	DITNA	UDFE	OVFE	Reserved		CMDE	CMCE	CMBE	CMAE
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW			RW	RW	RW	RW

位	标记	功能
31:23	Reserved	-
22	Dma_s_cmb	专用比较DMA使能
21	Dma_s_cma	专用比较DMA使能
20	Dma_g_udf	下溢出DMA使能
19	Dma_g_ovf	上溢出DMA使能
18:17	Reserved	
16	Dma_g_cmd	通用比较DMA使能
15	Dma_g_cmc	通用比较DMA使能
14	Dma_g_cmb	通用比较DMA使能
13	Dma_g_cma	通用比较DMA使能
12	CMSBE	专用比较基准值匹配B使能触发ADC
11	CMSAE	专用比较基准值匹配A使能触发ADC
10	DITENS	PWM 展频计数选择 0: 选择下溢出，1：选择上溢出
9	DITENB	PWM 通道B展频使能 0：使能无效，1：使能有效，每个周期改变 PWM 的输出延时
9	DITENA	PWM 通道A展频使能 0：使能无效，1：使能有效，每个周期改变 PWM 的输出延时
7	UDFE	下溢匹配使能触发 ADC 0：使能无效，1：使能有效，这个匹配可以控制 ADC/AOS_i_tirg
6	OVFE	上溢匹配使能触发 ADC 0：使能无效，1：使能有效，这个匹配可以控制 ADC/AOS_i_tirg
5:4	Reserved	-
3	CMDE	计数匹配D使能触发 ADC 0：使能无效，1：使能有效，这个匹配可以控制 ADC/AOS_i_tirg

2	CMCE	计数匹配C使能触发 ADC 0：使能无效，1：使能有效，这个匹配可以控制 ADC/AOS_i_tirg
1	CMBE	计数匹配B使能触发 ADC 0：使能无效，1：使能有效，这个匹配可以控制 ADC/AOS_i_tirg
0	CMAE	计数匹配A使能触发 ADC 0：使能无效，1：使能有效，这个匹配可以控制 ADC/AOS_i_tirg

16.3.28 AOS 选择控制寄存器（TIMx_AOSSR）

地址偏移量：0x110

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器，任意一个定时器其更改后，在另外两个定时器的值会同时更改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		SMH2	SMH1	SMH0	SML2	SML1	SML0	SOFTBK	Reserved		BFILTEN	BFILTS		FSAME	FBRAKE
		RW	RW	RW	RW	RW	RW	RW			RW	RW		R	R

位	标记	功能
31:14	Reserved	-
13	SMH2	通道2同高选择 0：选择无效，1：选择有效，出现同高时 AOS_i_odis[1]
12	SMH1	通道1同高选择 0：选择无效，1：选择有效，出现同高时 AOS_i_odis[1]
11	SMH0	通道0同高选择 0：选择无效，1：选择有效，出现同高时 AOS_i_odis[1]
10	SML2	通道2同低选择 0：选择无效，1：选择有效，出现同低时 AOS_i_odis[1]
9	SML1	通道1同低选择 0：选择无效，1：选择有效，出现同低时 AOS_i_odis[1]
8	SML0	通道0同低选择 0：选择无效，1：选择有效，出现同低时 AOS_i_odis[1]
7	SOFTBK	软件刹车：写1实现软件刹车
13	Reserved	-
4	BFILTEN	端口刹车滤波使能
3:2	BFILTS	端口刹车滤波时钟选择
1	FSAME	同高同低刹车标志，只读
0	FBRAKE	端口刹车标志，只读

16.3.29 AOS 选择控制寄存器标志清除（TIMx_AOSCL）

地址偏移量：0x114

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器，任意一个定时器其更改后，在另外两个定时器的值会同时更改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														FSAME	FBRAKE
														RO	RO

位	标记	功能
31:2	Reserved	-
1	FSAME	同高同低刹车标志清除，写0清除，写1无效，读恒为1
0	FBRAKE	端口刹车标志清除，写0清除，写1无效，读恒为1

16.3.30 端口刹车控制寄存器（TIMx_PTBKS）

地址偏移量：0x118

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器，任意一个定时器其更改后，在另外两个定时器的值会同时更改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能
31:16	Reserved	-
15	EN15	PD5 刹车端口使能：1 选择，0无效
14	EN14	PC15 刹车端口使能：1 选择，0无效
13	EN13	PB15 刹车端口使能：1 选择，0无效
12	EN12	PA15 刹车端口使能：1 选择，0无效
11	EN11	PD1 刹车端口使能：1 选择，0无效
10	EN10	PC11 刹车端口使能：1 选择，0无效
9	EN9	PB11 刹车端口使能：1 选择，0无效
8	EN8	PA11 刹车端口使能：1 选择，0无效
7	EN7	PD7 刹车端口使能：1 选择，0无效
6	EN6	PC7 刹车端口使能：1 选择，0无效
5	EN5	PB7 刹车端口使能：1 选择，0无效
4	EN4	PA7 刹车端口使能：1 选择，0无效
3	EN3	PD3 刹车端口使能：1 选择，0无效
2	EN2	PC3 刹车端口使能：1 选择，0无效
1	EN1	PB3 刹车端口使能：1 选择，0无效
0	EN0	PA3 刹车端口使能：1 选择，0无效

16.3.31 端口触发控制寄存器（TIMx_TTRIG）

地址偏移量：0x11C

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器，任意一个定时器其更改后，在另外两个定时器的值会同时更改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIGDS				TRIGCS				TRIGBS				TRIGAS			
RW				RW				RW				RW			

位	标记	功能
31:16	Reserved	-
15:12	TRIGDS	TIMx 触发D端口选择
11:8	TRIGCS	TIMx 触发C端口选择
7:4	TRIGBS	TIMx 触发B端口选择
3:0	TRIGAS	TIMx 触发A端口选择

控制信号与端口选择如下

0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
PA3	PB3	PC3	PD3	PA7	PB7	PC7	PD7	PA11	PB11	PC11	PD1	PA15	PB15	PC15	PD5

16.3.32 AOS 触发控制寄存器（TIMx_ITRIG）

地址偏移量：0x120

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器，任意一个定时器其更改后，在另外两个定时器的值会同时更改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IAOS3S				IAOS2S				IAOS1S				IAOS0S			
RW				RW				RW				RW			

位	标记	功能
31:16	Reserved	-
15:12	IAOS3S	TIMx AOS3 触发源选择
11:8	IAOS2S	TIMx AOS2 触发源选择
7:4	IAOS1S	TIMx AOS1 触发源选择
3:0	IAOS0S	TIMx AOS0 触发源选择

控制信号（IAOSxS）与中断源选择如下（x=0,1,2,3）

0000	0001	0010	0011	0100	0101	0110	0111
TIM0_INT	TIM1_INT	TIM2_INT	Reserved	TIM4_INTS	TIM5_INTS	TIM6_INTS	UART0_INT
1000	1001	1010	1011	1100	1101	1110	1111
UART1_INT	LPUART0_INT	VC0_INT	VC1_INT	Reserved	PCA_INT	SPI_INT	ADC_INT

16.3.33 端口刹车极性控制寄存器（TIMx_PTBKP）

地址偏移量：0x124

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器，任意一个定时器其更改后，在另外两个定时器的值会同时更改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POL15	POL14	POL13	POL12	POL11	POL10	POL9	POL8	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能
31:16	Reserved	-
15	POL15	PD5 刹车端口极性选择：1 低电平有效，0 高电平有效
14	POL14	PC15 刹车端口极性选择：1 低电平有效，0 高电平有效
13	POL13	PB15 刹车端口极性选择：1 低电平有效，0 高电平有效
12	POL12	PA15 刹车端口极性选择：1 低电平有效，0 高电平有效
11	POL11	PD1 刹车端口极性选择：1 低电平有效，0 高电平有效
10	POL10	PC11 刹车端口极性选择：1 低电平有效，0 高电平有效
9	POL9	PB11 刹车端口极性选择：1 低电平有效，0 高电平有效
8	POL8	PA11 刹车端口极性选择：1 低电平有效，0 高电平有效
7	POL7	PD7 刹车端口极性选择：1 低电平有效，0 高电平有效
6	POL6	PC7 刹车端口极性选择：1 低电平有效，0 高电平有效
5	POL5	PB7 刹车端口极性选择：1 低电平有效，0 高电平有效
4	POL4	PA7 刹车端口极性选择：1 低电平有效，0 高电平有效
3	POL3	PD3 刹车端口极性选择：1 低电平有效，0 高电平有效
2	POL2	PC3 刹车端口极性选择：1 低电平有效，0 高电平有效
1	POL1	PB3 刹车端口极性选择：1 低电平有效，0 高电平有效
0	POL0	PA3 刹车端口极性选择：1 低电平有效，0 高电平有效

17 看门狗定时器（WDT）

17.1WDT 简介

WDT 可用来检测和解决由软件错误引起的故障。当 WDT 计数器达到设定的溢出时间后，会触发中断或产生系统复位。WDT 由专用的 10KHz 片内振荡器驱动。

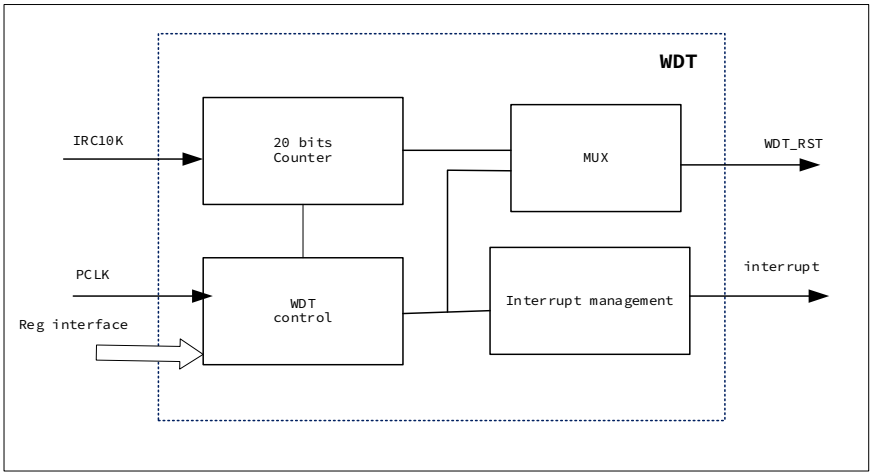


图 17-1 WDT 整体框图

17.2WDT 功能描述

- 20Bit 自由运行的递增计数器，溢出时间可配置为 1.6ms – 50s。
- 溢出后的动作可配置为中断或复位。
- WDT 时钟由独立的 RC 振荡器提供，可在 Sleep 和 DeepSleep 模式下工作。
- WDTCON 寄存器只有在 WDT 未被启动时才能修改，以防止启动后无意之间修改 WDT 的配置。

17.2.1 WDT 溢出后产生中断

在本模式下，WDT 将按所设定的时间周期性地产生中断。在中断服务程序中需要清除 WDT 中断标志。

配置方法如下所示：

Step1：配置 WDT_CON. WOV，选择 WDT 计时溢出时间。

Step2：设置 WDT_CON. WINT_EN 为 1，选择 WDT 溢出后产生中断。

Step3：使能 NVIC 中断向量表中的 WDT 中断。

Step4：向 WDT_RST 寄存器依次写入 0x1E、0xE1，启动 WDT 定时器。

Step5：在中断服务程序中向 WDT_RST 寄存器依次写入 0x1E、0xE1 以清除中断标志。

17.2.2 WDT 溢出后产生复位

在本模式下，WDT 计数器溢出后会产生 Reset 信号，该信号会复位 MCU。用户程序需要在 WDT 溢出前清零 WDT 计数器，从而避免产生 WDT 复位。

配置方法如下所示：

Step1：配置 WDT_CON. WOV，选择 WDT 计数器溢出时间。

Step2：设置 WDT_CON. WINT_EN 为 0，选择 WDT 溢出后产生复位。

Step3：向 WDT_RST 寄存器依次写入 0x1E、0xE1，启动 WDT 定时器。

Step4：在 WDT 溢出前向 WDT_RST 寄存器依次写入 0x1E、0xE1 以清零 WDT 计数器。

注意：

- 由于 WDT 振荡器是低精度的 RC 振荡器，强烈建议在 WDT 计数器计数值到达溢出值的一半之前对 WDT 进行清零。

17.3WDT 寄存器描述

基地址 0X40000F00

表 17-1 WDT 寄存器列表

寄存器	偏移地址	描述
WDT_RST	0X080	WDT清除控制寄存器
WDT_CON	0X084	WDT控制寄存器

17.3.1 WDT 清除控制寄存器（WDT_RST）

偏移地址：0x080

复位值: 0x0000 0000

31:8	7	6	5	4	3	2	1	0
Reserved	WDTRST							
	WO							

位	符号	描述
31:8	Reserved	保留位，读为0
7:0	WDTRST	看门狗启动/清零控制 当看门狗未启动时，向该寄存器依次写入0x1E、0xE1，启动WDT定时器。 当看门狗已启动时，向该寄存器依次写入0x1E、0xE1，清零WDT定时器及中断标志。

17.3.2 WDT_CON 寄存器

偏移地址：0x084

复位值: 0x0000 000F

注：该寄存器只有在 WDT 未运行时才可以写入。

31:16	15:8	7	6	5	4	3	2	1	0
Reserved	WCNTL	WDTINT	Res.	WINT_EN	WDTR	WOV			
	RO	RO		RW	RO	RW			

位	符号	描述
31:16	Reserved	保留位，读为0
15:8	WCNTL	WDT计数器低8位
7	WDTINT	WDT中断标志 1：已发生WDT中断，向WDT_RST寄存器依次写入0x1E、0xE1以清除该中断标志。 0：未发生WDT中断 。
5	WINT_EN	WDT溢出后的动作配置 1：WDT溢出后产生中断。 0：WDT溢出后产生复位。
4	WDTR	WDT运行标志 1：WDT正在运行 0：WDT停止
3:0	WOV[3:0]	WDT计时溢出时间配置 0000：1.6ms / 1000：500ms 0001：3.2ms / 1001：820ms 0010：6.4ms / 1010：1.64s 0011：13ms / 1011：3.28s 0100：26ms / 1100：6.55s 0101：51ms / 1101：13.1s 0110：102ms / 1110：26.2s 0111：205ms / 1111：52.4s

18 通用同步异步收发器（UART）

18.1简介

通用同步异步收发器（UART）能够灵活地与外部设备进行全双工数据交换，它支持同步单向通信、单线半双工通信以及多处理器通信。常用于短距离、低速的串行通信中。UART 通过可编程波特率发生器提供了多种波特率。UART 支持多种工作模式。

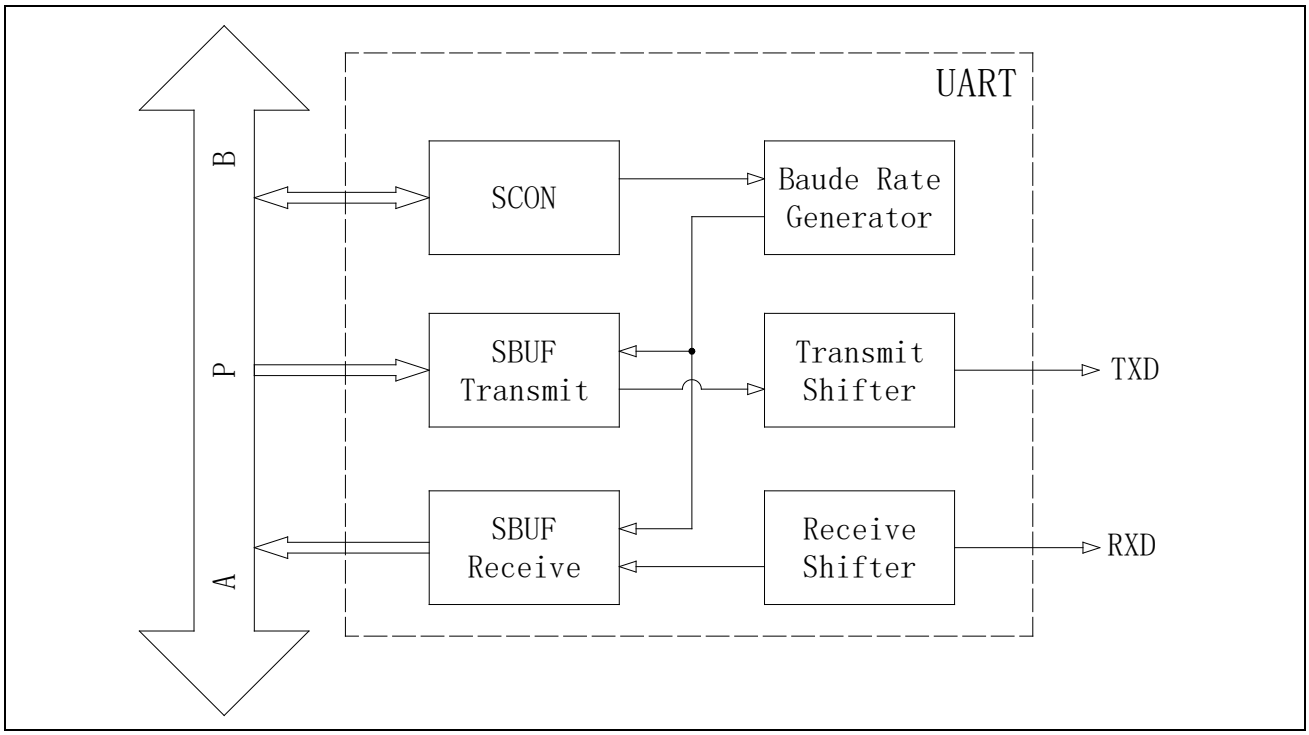


图 18-1 结构框图

18.2主要特性

通用 UART 模块支持以下基本功能：

- 全双工传输、半双工传输、单线半双工传输
- 可编程串行通信功能
  - 两种字符长度：8 比特、 9 比特
  - 三种校验方式：无检验、奇校验、偶校验
  - 三种停止长度：1 比特、 2 比特、 1.5 比特
- 16 比特波特率发生器
- 多机通讯
- 硬件地址识别
- 硬件流控
- DMAC 硬件传输握手

18.3功能描述

18.3.1 工作模式

UART 支持多种工作模式：同步半双工模式、异步全双工模式、单线半双工模式。通过 UARTx_SCON.SM 和 UARTx_SCON. HDSEL 搭配，即可配置出所需要的各种工作模式。

18.3.1.1 Mode0~Mode3 功能对比

配置 UARTx_SCON.SM 可选择不同的传输模式：Mode0~Mode3。这四种工作模式的主要功能对比如下表所示：

表 18-1 Mode0/1/2/3 数据结构

工作模式		传输位宽	数据组成	波特率
Mode0	同步模式 半双工	8bit	Data(8bit)	$BaudRate=\frac{f_{PCLK}}{12}$
Mode1	异步模式 全双工	10~11bit	Start (1bit) + Data(8bit) + Stop(1~2bit)	$BaudRate=\frac{f_{PCLK}}{OVER*SCNT}$
Mode2	异步模式 全双工	11~12bit	Start (1bit) + Data(8bit) + B8(1bit) + Stop(1~2bit)	$BaudRate=\frac{f_{PCLK}}{OVER}$
Mode3	异步模式 全双工	11~12bit	Start (1bit) + Data(8bit) + B8(1bit) + Stop(1~2bit)	$BaudRate=\frac{f_{PCLK}}{OVER*SCNT}$

注：

- Mode0 只能作为主机发送 UART 同步移位时钟，不可以作为从机接收外部输入的 UART 同步移位时钟。
- f_{PCLK}代表当前 PCLK 的频率。
- OVER 的定义详见 UARTx_SCON。
- SCNT 的定义详见 UARTx_SCNT。
- B8 数据位比较特殊，在不同应用下具有不同的含义，请参考以下表格：

表 18-2 B8 数据含义

应用场景	UARTx_SCON. ADRDET	UARTx_SCON. B8CONT[1:0]	B8 数据含义
奇偶校验	--	01/10	接收时，B8 是所收到的 8-Bit 数据的奇偶校验位； 发送时，B8 是待发送的 8-Bit 数据的奇偶校验位；
多机通讯	1	--	B8=1，代表当前是地址帧； B8=0，代表当前是数据帧；
其他	0	00/11	接收/发送数的第 8 比特

注意：

- 当开启多机通讯模式，接收数据奇偶校验自动关闭；发送数据奇偶校验仍受 B8CONT 控制。

18.3.1.2 Mode0 数据收发说明

发送数据时，清除 UARTx_SCON.REN 位，并将数据写入 UARTx_SBUF 寄存器中。此时，发送数据从 RXD 输出（低位在先，高位在后），同步移位时钟从 TXD 输出。

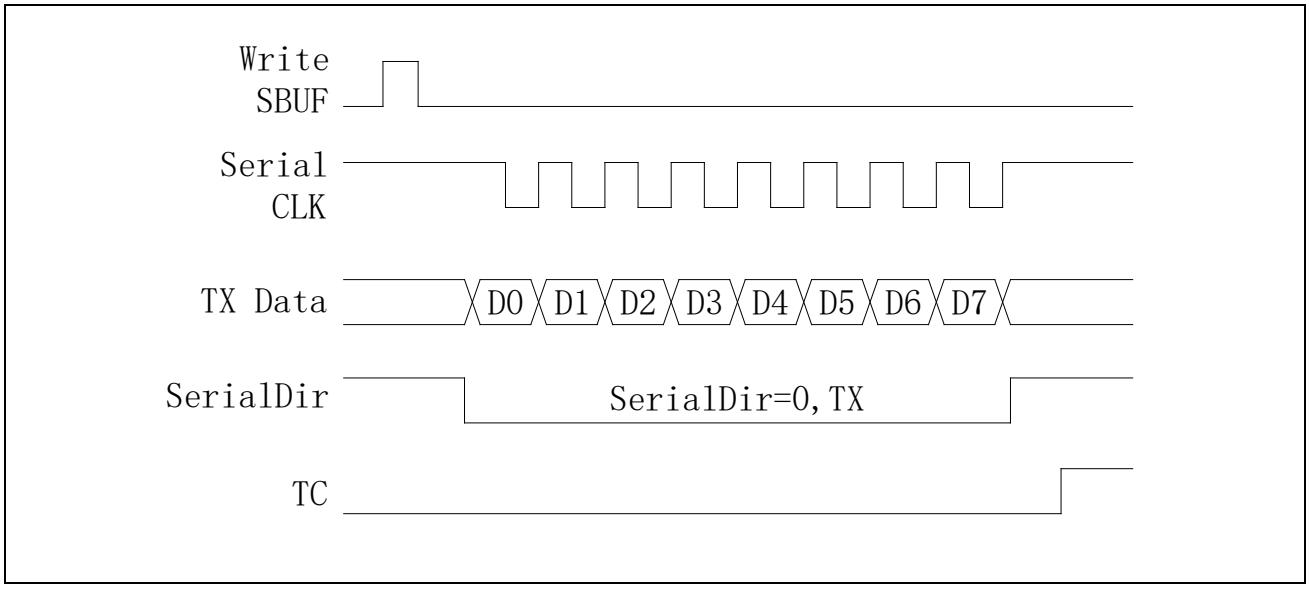


图 18-2 Mode0 发送数据

接收数据时，将 UARTx_SCON.REN 位置 1，并将 UARTx_ISR.RC 位清零。当接收结束，数据可从 UARTx_SBUF 寄存器读取。此时，接收数据从 RXD 输入（低位在先，高位在后），同步移位时钟从 TXD 输出。

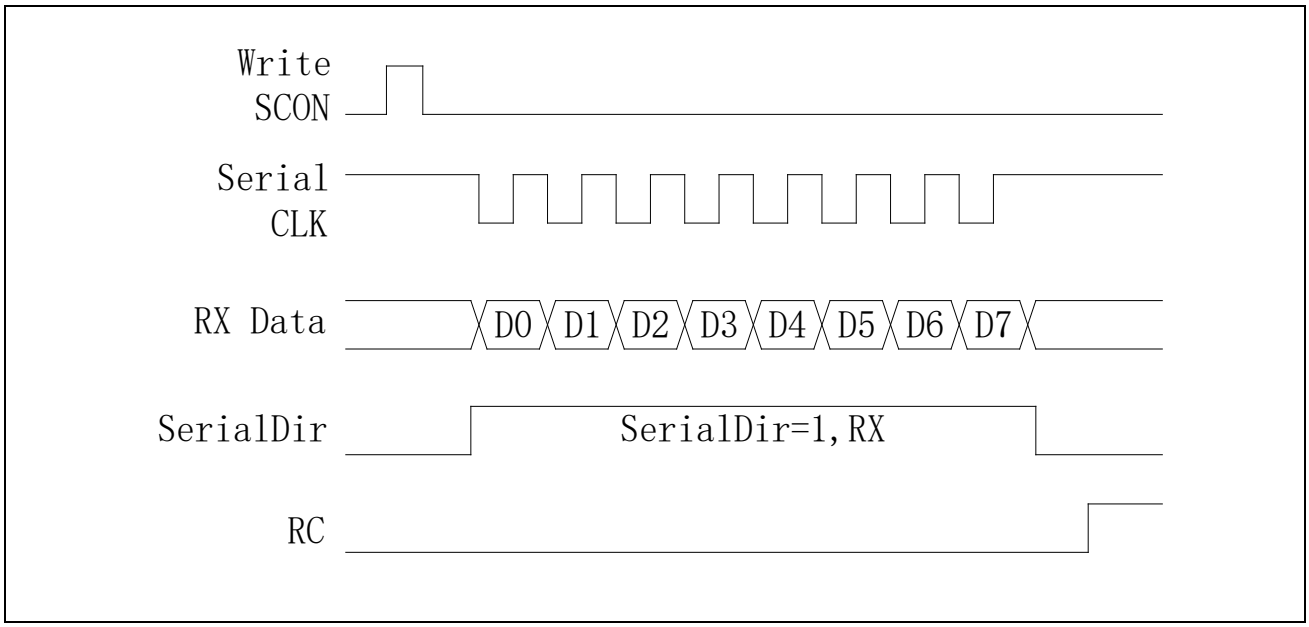


图 18-3 Mode0 接收数据

18.3.1.3 Mode1 数据收发说明

发送数据时，与 UARTx_SCON.REN 的值无关，将所发送数据写入 UARTx_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

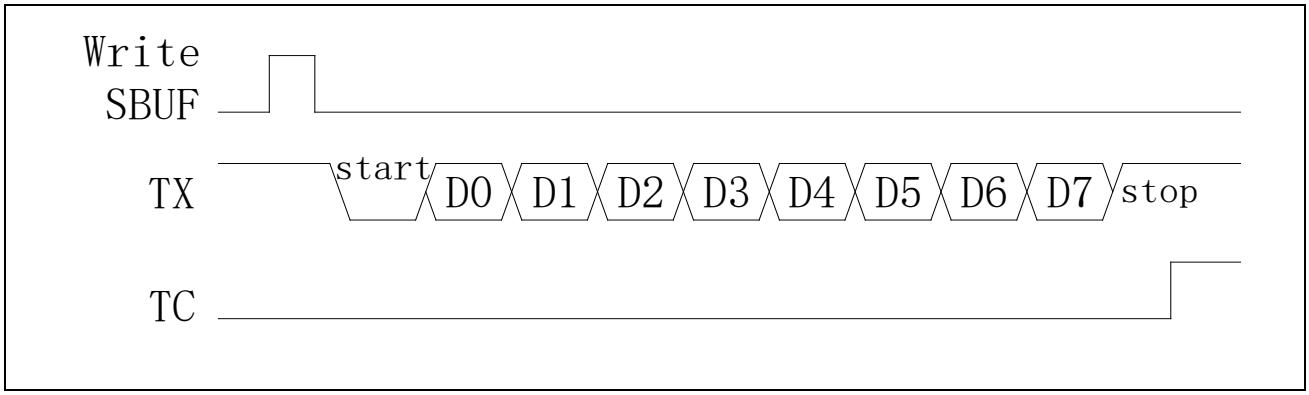


图 18-4 Mode1 发送数据

接收数据时，将 UARTx_SCON.REN 位置 1，并将 UARTx_ISR.RC 位清 0。开始接收 RXD 上数据（低位在先，高位在后），当接收完毕，可以从 UARTx_SBUF 寄存器读出。

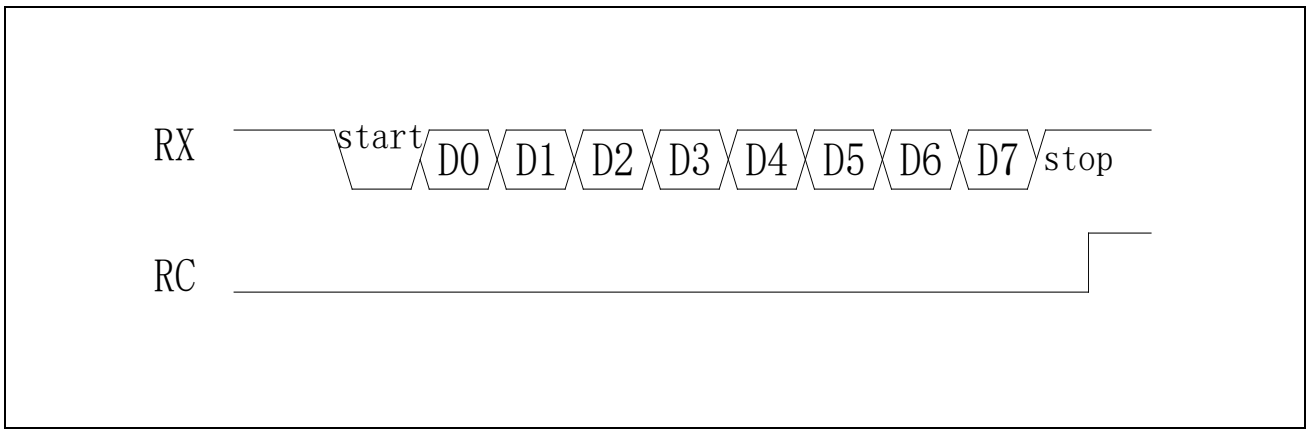


图 18-5 Mode1 接收数据

18.3.1.4 Mode2 数据收发说明

发送数据时，与 UARTx_SCON.REN 的值无关，将所发送数据写入 UARTx_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

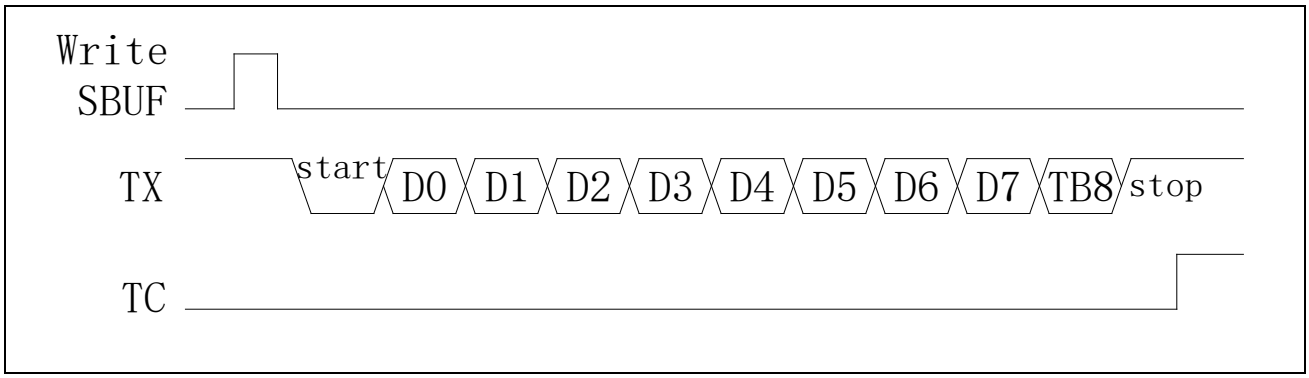


图 18-6 Mode2 发送数据

接收数据时，需将 UARTx_SCON.REN 位置 1，并将 UARTx_ISR.RC 位清 0。开始接收 RXD 上的数据（低位在先，高位在后），当接收完毕，可以从 UARTx_SBUF 寄存器读取。

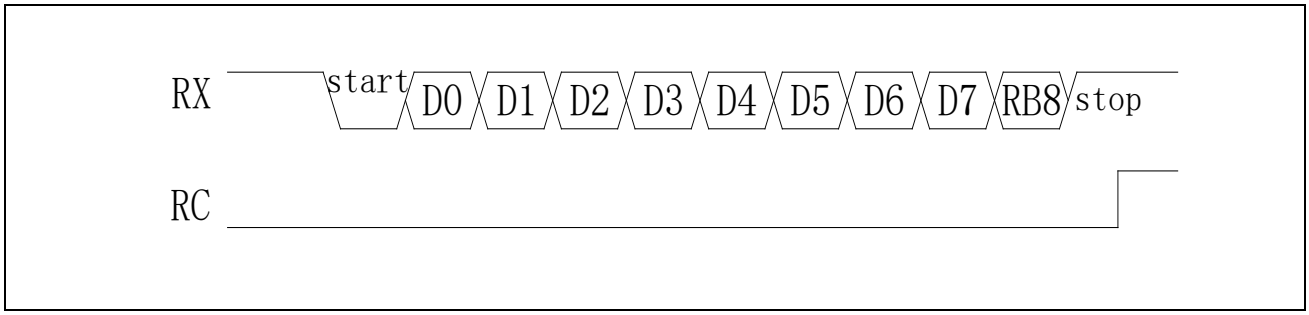


图 18-7 Mode2 接收数据

18.3.1.5 Mode3 数据收发说明

发送数据时，与 UARTx_SCON.REN 的值无关，将所发送数据写入 UARTx_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

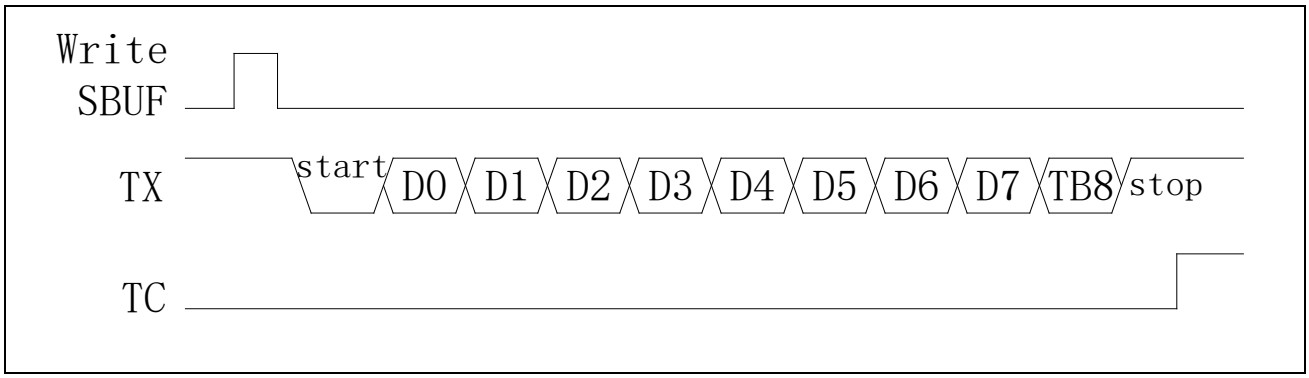


图 18-8 Mode3 发送数据

接收数据时，将 UARTx_SCON.REN 位置 1，并将 UARTx_ISR.RC 位清 0。开始接收 RXD 上数据（低位在先，高位在后），当接收完毕，可以从 UARTx_SBUF 寄存器读出。

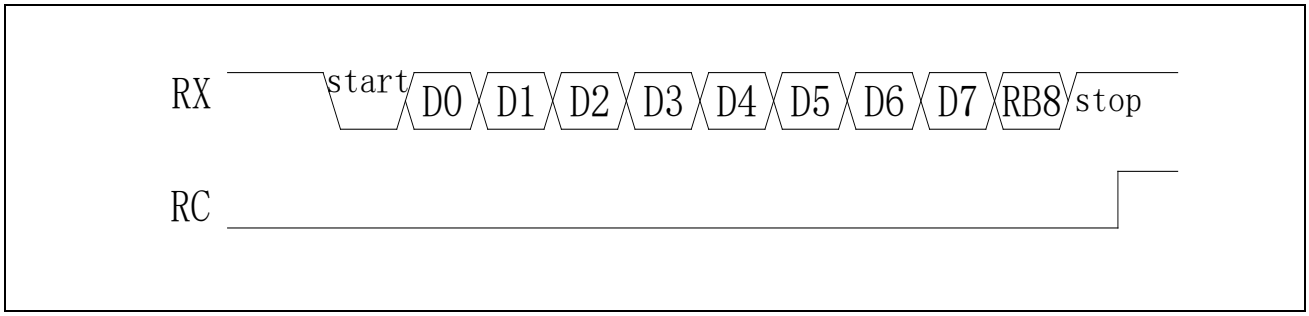


图 18-9 Mode3 接收数据

18.3.1.6 单线半双工数据收发说明

当 UARTx_SCON.HDSEL=1'b1，可进入单线半双工传输模式：

- TX 与 RX 信号线在模块内部相连。
- RX 信号不再使用，接收发送的数据都经由 TX 信号完成。TX 信号的方向控制由硬件逻辑完成，无需软件控制。

- 当发送缓存为空时，TX 信号始终为输入（接收状态），一旦向发送缓存填入一个数据，TX 信号变为输出（发送状态），当发送完成后，发送缓存变为空，TX 信号又回到输入。
- 无数据传输时，TXD 引脚始终处于输入状态。用户程序需配置该引脚为开漏输出并外接上拉电阻。

注意：

1．即使模块处于接收过程中，只要发送缓存被填入数据，TX 信号立即变为输出（发送状态），接收过程将被打断，并得到一个错误的接收数据。

2．单线半双工模式与 Mode0 模式的异同：

同：两者都是用一根信号分时复用地完成数据收发，

异：Mode0 为同步传输，有同步时钟，而单线半双工模式为异步传输，所有的波特率计算还是遵照 Mode1-2-3 的波特率公式，仅仅是收发功能复用到了 TX 信号上。

18.3.2 波特率生成

Mode0~Mode3 生成波特率的公式不尽相同，详见下方所示。

Mode0 波特率生成工式：$BaudRate=\frac{f_{PCLK}}{12}$

Mode1 波特率生成工式：$BaudRate=\frac{f_{PCLK}}{OVER*SCNT}$

Mode2 波特率生成工式：$BaudRate=\frac{f_{PCLK}}{OVER}$

Mode3 波特率生成工式：$BaudRate=\frac{f_{PCLK}}{OVER*SCNT}$

注：

- f_{PCLK}代表当前 PCLK 的 频率。
- OVER 的定义详见 UARTx_SCON。
- SCNT 的定义详见 UARTx_SCNT

18.3.2.1 Mode1/Mode3 波特率设置示例

表 18-3 PCLK=4MHz 波特率计算表

波特率	PCLK = 4 MHz					
	OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	208	2403.85	0.16%	104	2403.85	0.16%
4800	104	4807.69	0.16%	52	4807.69	0.16%
9600	52	9615.38	0.16%	26	9615.38	0.16%
19200	26	19230.77	0.16%	13	19230.77	0.16%
38400	13	38461.54	0.16%	7	35714.29	-6.99%
57600	9	55555.56	-3.55%	4	62500.00	8.51%
76800	7	71428.57	-6.99%	3	83333.33	8.51%
115200	4	125000.00	8.51%	2	125000.00	8.51%
128000	4	125000.00	-2.34%	2	125000.00	-2.34%
250000	2	250000.00	0.00%	1	250000.00	0.00%

表 18-4 PCLK=8MHz 波特率计算表

波特率	PCLK = 8 MHz					
	OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	417	2398.08	-0.08%	208	2403.85	0.16%
4800	208	4807.69	0.16%	104	4807.69	0.16%
9600	104	9615.38	0.16%	52	9615.38	0.16%
19200	52	19230.77	0.16%	26	19230.77	0.16%
38400	26	38461.54	0.16%	13	38461.54	0.16%
57600	17	58823.53	2.12%	9	55555.56	-3.55%
76800	13	76923.08	0.16%	7	71428.57	-6.99%
115200	9	111111.11	-3.55%	4	125000.00	8.51%
128000	8	125000.00	-2.34%	4	125000.00	-2.34%
256000	4	250000.00	-2.34%	2	250000.00	-2.34%
500000	2	500000.00	0.00%	1	500000.00	0.00%

表 18-5 PCLK=16MHz 波特率计算表

波特率	PCLK = 16 MHz					
	OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	833	2400.96	0.04%	417	2398.08	-0.08%
4800	417	4796.16	-0.08%	208	4807.69	0.16%
9600	208	9615.38	0.16%	104	9615.38	0.16%
19200	104	19230.77	0.16%	52	19230.77	0.16%
38400	52	38461.54	0.16%	26	38461.54	0.16%
57600	35	57142.86	-0.79%	17	58823.53	2.12%
76800	26	76923.08	0.16%	13	76923.08	0.16%
115200	17	117647.06	2.12%	9	111111.11	-3.55%
128000	16	125000.00	-2.34%	8	125000.00	-2.34%
256000	8	250000.00	-2.34%	4	250000.00	-2.34%
500000	4	500000.00	0.00%	2	500000.00	0.00%
1000000	2	1000000.00	0.00%	1	1000000.00	0.00%

表 18-6 PCLK=24MHz 波特率计算表

波特率	PCLK = 24 MHz					
	OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	1250	2400.00	0.00%	625	2400.00	0.00%
4800	625	4800.00	0.00%	313	4792.33	-0.16%
9600	313	9584.66	-0.16%	156	9615.38	0.16%
19200	156	19230.77	0.16%	78	19230.77	0.16%
38400	78	38461.54	0.16%	39	38461.54	0.16%
57600	52	57692.31	0.16%	26	57692.31	0.16%
76800	39	76923.08	0.16%	20	75000.00	-2.34%
115200	26	115384.62	0.16%	13	115384.62	0.16%
128000	23	130434.78	1.90%	12	125000.00	-2.34%
256000	12	250000.00	-2.34%	6	250000.00	-2.34%
1000000	3	1000000.00	0.00%	2	750000.00	-25.00%
1500000	2	1500000.00	0.00%	1	1500000.00	0.00%

表 18-7 PCLK=32MHz 波特率计算表

波特率	PCLK = 32 MHz					
	OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	1667	2399.52	-0.02%	833	2400.96	0.04%
4800	833	4801.92	0.04%	417	4796.16	-0.08%
9600	417	9592.33	-0.08%	208	9615.38	0.16%
19200	208	19230.77	0.16%	104	19230.77	0.16%
38400	104	38461.54	0.16%	52	38461.54	0.16%
57600	69	57971.01	0.64%	35	57142.86	-0.79%
76800	52	76923.08	0.16%	26	76923.08	0.16%
115200	35	114285.71	-0.79%	17	117647.06	2.12%
128000	31	129032.26	0.81%	16	125000.00	-2.34%
256000	16	250000.00	-2.34%	8	250000.00	-2.34%
1000000	4	1000000.00	0.00%	2	1000000.00	0.00%
2000000	2	2000000.00	0.00%	1	2000000.00	0.00%

表 18-8 PCLK=48MHz 波特率计算表

波特率	PCLK = 48 MHz					
	OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	2500	2400.00	0.00%	1250	2400.00	0.00%
4800	1250	4800.00	0.00%	625	4800.00	0.00%
9600	625	9600.00	0.00%	313	9584.66	-0.16%
19200	313	19169.33	-0.16%	156	19230.77	0.16%
38400	156	38461.54	0.16%	78	38461.54	0.16%
57600	104	57692.31	0.16%	52	57692.31	0.16%
76800	78	76923.08	0.16%	39	76923.08	0.16%
115200	52	115384.62	0.16%	26	115384.62	0.16%
128000	47	127659.57	-0.27%	23	130434.78	1.90%
256000	23	260869.57	1.90%	12	250000.00	-2.34%
1000000	6	1000000.00	0.00%	3	1000000.00	0.00%
2000000	3	2000000.00	0.00%	2	1500000.00	-25.00%
3000000	2	3000000.00	0.00%	1	3000000.00	0.00%

18.3.3 帧错误检测

当工作在 Mode1/2/3 时，UART 具有帧错误检测功能，如果接收数据时硬件未在预期时间内识别出停止位，从而出现同步失效或过度的噪声，则会检测到帧错误。检测到帧错误时，UARTx_ISR.FE 置 1。UARTx_ISR.FE 要及时软件清零。

18.3.4 多机通讯

当工作在 Mode2/3 时，将 UARTx_SCON.ADRDET 位置“1”，可开启多机通讯功能。

主机可以通过 UARTx_SBUF[8]来区分当前的发送帧是地址帧（UARTx_SBUF[8]=1）还是数据帧（UARTx_SBUF[8]=0）。

- 当为数据帧时，该帧数据不会存入到从机的 UARTx_SBUF 寄存器中，从机也不会产生接收中断。
- 当为地址帧时，由于多机通讯中自动地址识别功能已开启，使得从机可以检测接收到的地址与其自身地址是否相符合。
  - 如果地址相符合，从机会对 UARTx_ISR.RC 置“1”，UARTx_SBUF[8]置“1”，同时将该地址帧存入到 UARTx_SBUF 寄存器中。从机软件看到 UARTx_SBUF[8]=1 并且 UARTx_ISR.RC=1 后，将 UARTx_SCON.ADEDET 位清“0”，接受数据帧。
  - 如果地址不符合，表明主机并不是寻址该从机，从机硬件保持 UARTx_SBUF[8]和 UARTx_ISR.RC 为“0”，软件保持 UARTx_SCON.ADRDET 位为“1”，继续处于地址监听状态。

注：如果有需要，也可以在 Mode1 下开启多机通讯位，此时 TB8 位由 stop 位代替。当从机接收到匹配的地址帧和有效的 stop 位时，UARTx_ISR.RC 会被置“1”。

从机地址配置

从机地址由广播地址和用户配置地址组成：

- 广播地址：UART_SADDR | UART_SADEN
- 用户配置地址：UART_SADDR & UART_SADEN

收到地址帧时，检测到从总线收到的地址与 UART_SADEN 的值相与后与用户配置地址相同，或收到的地址与广播地址相与后与广播地址相同，则认为收到了与从机地址相匹配的地址帧。

例：UART_SADDR=0x55，UART_SADEN=0xF0，则从机可以匹配的地址为 0xF5、0xF7、0xFD、0xFF（广播地址）和 0x50~0x5F（用户配置地址）。

18.3.5 DMAC 硬件握手

UART 模块支持 DMAC 的硬件握手逻辑。

- 将 UARTx_SCON.DMACTXEN 设置为 1，可以打开 UART TX 的 DMAC 硬件握手逻辑。当发送缓存为空时，UART 会向 DMAC 发出数据搬运请求 TX REQ。DMAC 收到该信号，则从 DMAC 源地址搬

运一个帧的发送数据到 UARTx_SBUF 中。上述步骤重复发生，直到 DMAC 中所配置的数据长度全部搬运完毕。

- 将 UARTx_SCON.DMACRXEN 设置为 1，可以打开 UART RX 的 DMAC 硬件握手逻辑。当一帧接收完成，UART 会向 DMAC 发出数据搬运请求 RX REQ。DMAC 收到该信号，则从 UARTx_SBUF 中把接收数据搬运至 DMAC 的目标地址中。上述步骤重复发生，直到 DMAC 中所配置的数据长度全部搬运完毕。

18.3.6 硬件流控

通过增加 nCTS 和 nRTS 信号可以实现 UART 硬件流控的功能，即 UART 硬件模块根据 nCTS 和 nRTS 的高低电平自动控制数据的收发，而无需通过软件来判断。两个 UART 模块之间的硬件流控示意图如下所示：

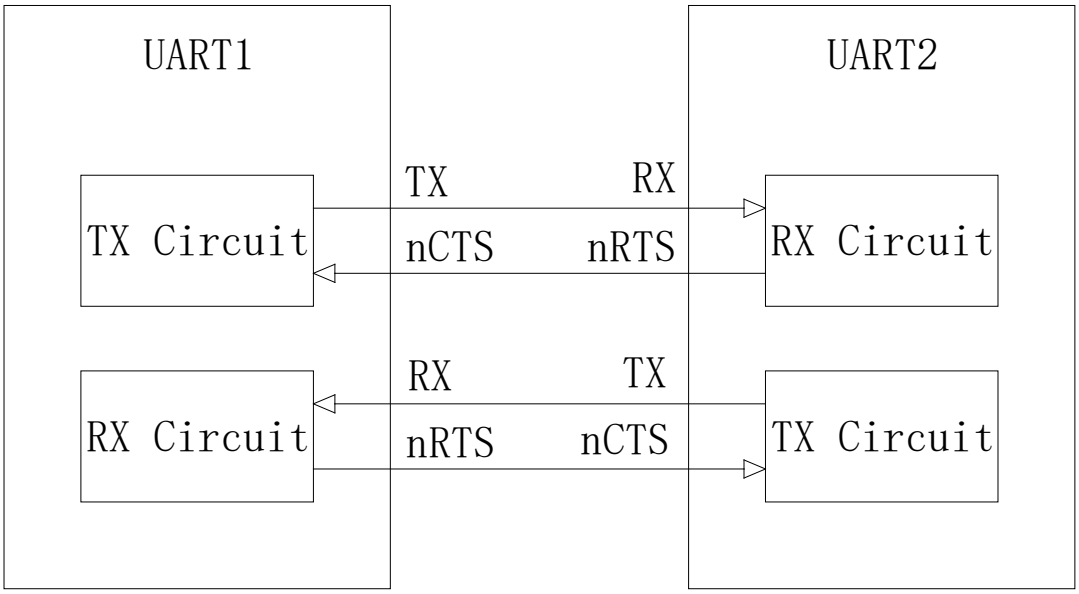


图 18-10 UART 硬件流控

- nRTS 流控

nRTS 流控使能时（UARTx_SCON.RTSEN 设置为 1）：

- 当 UART 接收缓存空时，会将 nRTS 变为有效（连接到低电平）。
- 当接收缓存满时，会将 nRTS 变为无效（连接到高电平），表明发送过程会在当前帧结束后停止。

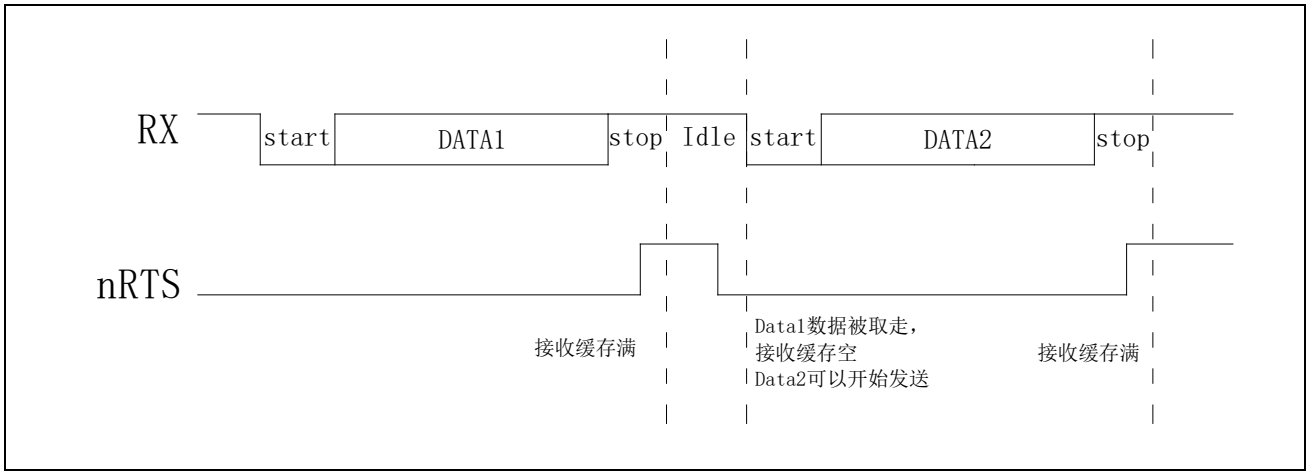


图 18-11　nRTS 硬件流控信号

■　CTS 流控

nCTS 流控使能时（UARTx_SCON.CTSEN 设置为 1），在 UART 发送下一帧数据之前，先判断 nCTS 的高低电平：

- 如果 nCTS 为有效（连接到低电平），则 UART 发送下一帧数据。
- 如果 nCTS 为无效（连接到高电平），则 UART 当前帧发送完成后，暂停发送下一帧数据。

当 nCTS 流控使能时，一旦 nCTS 信号发生翻转，UARTx_SFLAG.CTSIE 便会硬件置 1。如果 UARTx_SCON.CTSIE 置 1，则会产生中断，同时，nCTS 信号的高低电平会记录在 UARTx_SFLAG.CTS 标志位中。

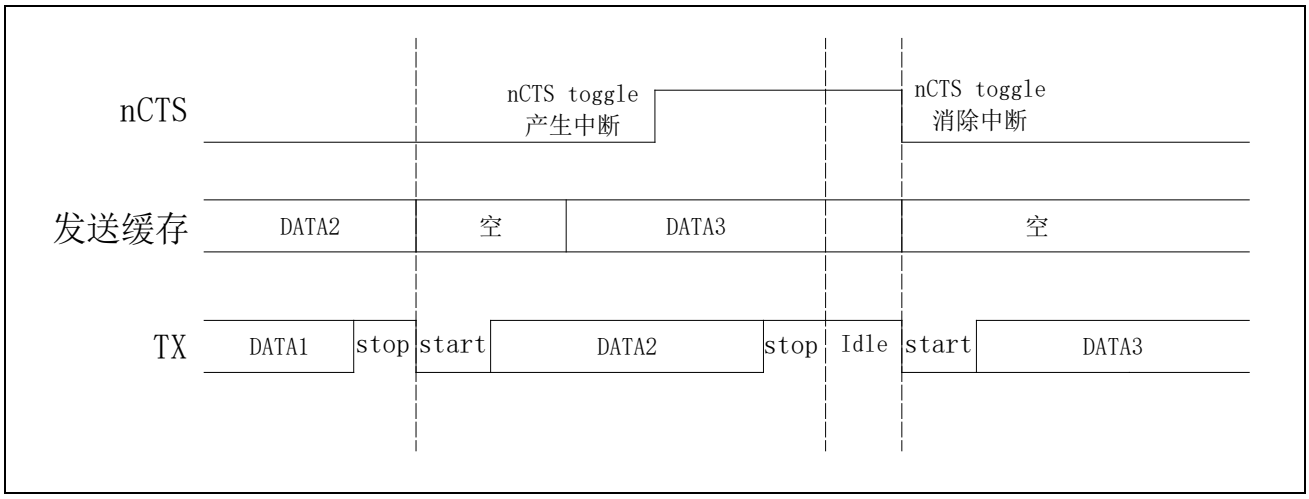


图 18-12　nCTS 硬件流控信号

18.3.7 收发端缓存

■ 接收缓存

UART 模块接收端有一个帧（8/9-Bit）的接收缓存，即保存接收的数据帧直至下一帧数据的 Stop 位被接收才更新该数据帧。

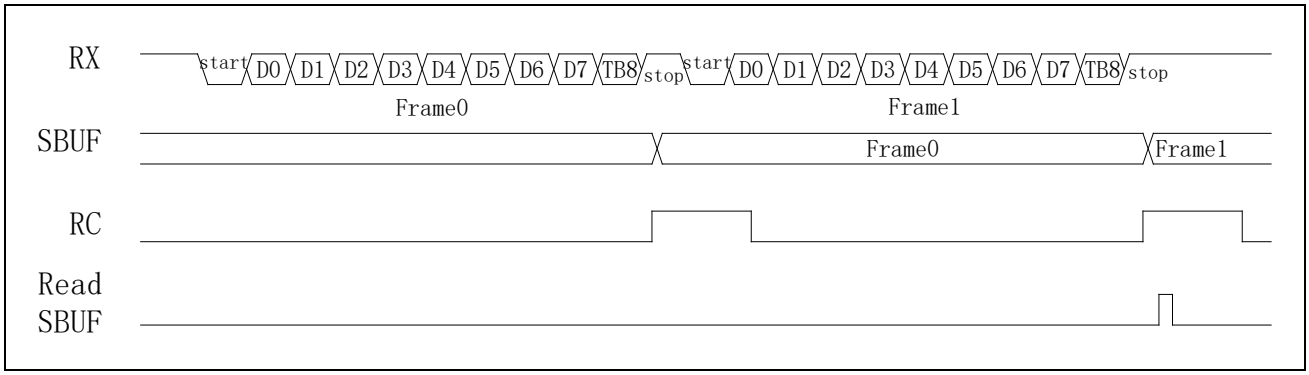


图 18-13 接收缓存

■ 发送缓存

UART 模块发送端有一个帧(8/9-Bit)的发送缓存，当 UART 在发送当前帧时，下一个发送数据软件写入 UARTx_SBUF。

当 UARTx_ISR.TXE=0 时，表明当前发送缓存满，UARTx_SBUF 不能写入下一个发送数据。否则该数据会硬件丢弃。

当 UARTx_ISR.TXE=1 时，表明当前发送缓存空，UARTx_SBUF 可以写入下一个发送数据，在完成当前数据传输后，硬件自动把发送缓存中的数据装载入移位寄存器中发送出去。

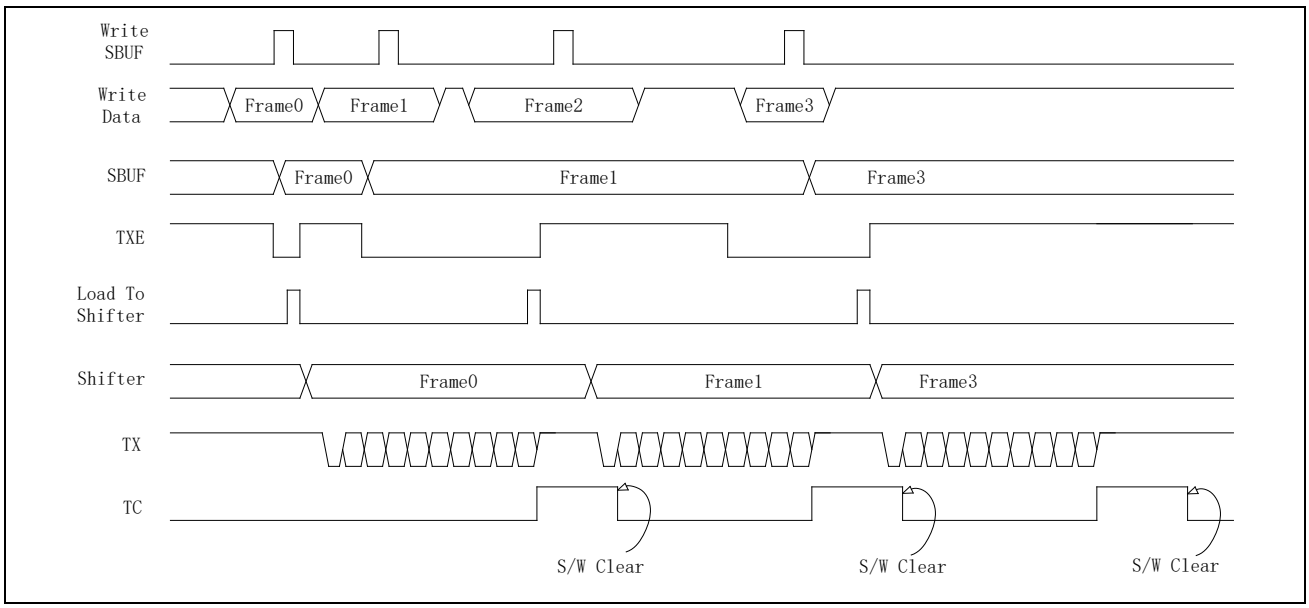


图 18-14 发送缓存

18.4寄存器

UART0 基地址：0x4000 0000

UART1 基地址：0x4000 0100

UART2 基地址：0x4000 6000

UART3 基地址：0x4000 6400

寄存器	偏移地址	描述
UARTx_SBUF	0x00	数据寄存器
UARTx_SCON	0x04	控制寄存器
UARTx_SADDR	0x08	地址寄存器
UARTx_SADEN	0x0C	地址掩码寄存器
UARTx_ISR	0x10	中断标志位寄存器
UARTx_ICR	0x14	中断标志位清除寄存器
UARTx_SCNT	0x18	波特率寄存器

18.4.1 数据寄存器（UARTx_SBUF）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							DATA[8]	DATA[7:0]							
							RW	RW							

位	标记	功能描述
31:9	Reserved	
8	DATA[8]	在Mode0/1下，读取该位为0，写入该位无效； 在Mode2/3下，该位表示Bit8数据位，分以下两种情况： （1）当硬件奇偶校验位开启时，接收时该位为接收数据奇偶校验位，校验由硬件进行，如校验出错，校验错误标志位PE置1；发送时该位无效，发送数据奇偶校验位由硬件计算并发送； （2）当硬件奇偶校验位关闭时，接收时该位为接收数据Bit8；发送时该位为发送数据Bit8； 注意：当开启多机通讯模式，接收数据奇偶校验自动关闭；发送数据奇偶校验仍受B8CONT控制；
7:0	DATA[7:0]	发送数据时，将发送数据写入该寄存器；接收数据时，数据接收完毕后，从该寄存器中读取。

18.4.2 控制寄存器（UARTx_SCON）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									HDSEL	FEIE	CTSIE	CTSEN	RTSEN	DMATXEN	DMARXEN
									RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOPBIT		PEIE	Reserved			OVER	TXEIE	SM		ADRDET	REN	B8CONT		TCIE	RCIE
RW		RW				RW	RW	RW		RW	RW	RW		RW	RW

位	标记	功能描述
31:23	Reserved	
22	HDSEL	单线半双工模式使能位；0: 关闭单线半双工模式；1：打开单线半双工模式；
21	FEIE	帧错误中断使能位；0：关闭中断；1：打开中断；
20	CTSIE	CTS信号翻转中断使能位；0：关闭中断；1：打开中断；
19	CTSEN	硬件flow control信号使能位；0：关闭flow control信号；1：打开flow control信号；
18	RTSEN	
17	DMATXEN	TX DMAC的硬件握手信号使能位；0：关闭硬件握手信号；1：打开硬件握手信号；
16	DMARXEN	RX DMAC的硬件握手信号使能位；0：关闭硬件握手信号；1：打开硬件握手信号；
15:14	STOPBIT	stop bit长度选择；00:1-bit；01:1.5-bit；10:2-bit；11：reserved； 注意：Mode0时虽然没有Stop Bit，但仍需把STOPBIT[1:0]保持为2’b00；
13	PEIE	奇偶校验错误中断使能位；0：关闭奇偶校验错误中断；1：打开奇偶校验错误中断； 数据接收标志产生于数据停止位接收完成，硬件奇偶校验与数据接收中断由于停止位设置不同，奇偶校验错误中断会提前于数据接收中断。使用此中断使能时请注意。 建议如下： 方法1：关闭PEIE中断使能，接收数据中断后软件判断ISR.PE奇偶校验是否正确。 方法2：关闭PEIE中断使能，接收数据中断通过接收SBUF.BIT8软件判断奇偶校验是否正确。
12:10	Reserved	
9	OVER	Mode0：　无效； Mode1/3：　0：16采样分频；　1：8采样分频； Mode2:　　0：32采样分频；　1：16采样分频；
8	TXEIE	TX空中断使能位；0：TX Buffer空中断关闭；1：TX Buffer空中断打开；
7:6	SM	工作模式；00：mode0；01：mode1；10：mode2；11：mode3；
5	ADRDET	多机通讯地址自动识别使能位；0：关闭；1：打开；
4	REN	Mode0：　0：发送；1：接收； Mode1/2/3：0：发送；1：接收/发送；
3:2	B8CONT	Bit8数据控制位；

		00：由软件读写SBUF[8]来决定；01：硬件偶校验； 10：硬件奇校验；　　11：保留；
1	TCIE	发送中断使能位；0：发送中断关闭；1：发送中断打开；
0	RCIE	接收中断使能位；0：接收中断关闭；1：接收中断打开；

18.4.3 地址寄存器（UARTx_SADDR）

偏移地址：0x08

复位值：0x0000 0000

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

位	标记	功能描述
31:8	Reserved	
7:0	SADDR	从机设备地址寄存器

18.4.4 地址掩码寄存器（UARTx_SADEN）

偏移地址：0x0C

复位值：0x0000 0000

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

位	标记	功能描述
31:8	RESERVED	
7:0	SADEN	从机设备地址掩码寄存器

18.4.5 标志位寄存器（UARTx_ISR）

偏移地址：0x10

复位值：0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									CTS	CTSIF	PE	TXE	FE	TC	RC
									RO	RO	RO	RO	RO	RO	RO

位	符号	描述
31:7	Reserved	
6	CTS	CTS信号标志位；硬件置1；硬件清零；0：CTS信号为低电平；1：CTS信号为高电平；
5	CTSIF	CTS中断标志位；硬件置1；软件清零；0：CTS信号没有发生反转；1：CTS信号发生反转；
4	PE	奇偶校验错误标志位；硬件置1；软件清零；0：无奇偶校验错误；1：奇偶校验错误；
3	TXE	Tx Buffer空标志位；硬件置1；硬件清零；0：Tx Buffer非空；1：Tx Buffer空
2	FE	帧错误标志位；0：硬件置1；软件清零；
1	TC	发送完毕中断标志位；硬件置1；软件清零；0：发送未完成；1：发送完毕；
0	RC	接收完毕中断标志位；硬件置1；软件清零；0：接收未完成；1：接收完成；

18.4.6 标志位清除寄存器（UARTx_ICR）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										CTSIFCF	PECF	Res.	FECF	TCCF	RCCF
										R1W0	R1W0		R1W0	R1W0	R1W0

位	标记	功能描述
31:6	Reserved	
5	CTSIFCF	CTSIF标志清楚位；写0清除；写1无效；
4	PECF	PE标志清楚位；写0清除；写1无效；
3	Reserved	
2	FECF	FE标志清除位；写0清零；写1无效；
1	TCCF	TC标志清除位；写0清零；写1无效；
0	RCCF	RC标志清除位；写0清零；写1无效；

18.4.7 波特率寄存器（UARTx_SCNT）

偏移地址：0x18

复位值：0x0000 0000

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

位	标记	功能描述
31:16	Reserved	
15:0	SCNT	波特率计数器

19 低功耗同步异步收发器（LPUART）

19.1简介

低功耗同步异步接收器（LPUART）是一种 UART，允许有限功耗下进行全双工 UART 通信。即使当微控制器处于停止模式，能耗极低时，LPUART 也会等待 UART 帧的到来。LPUART 包含所有必要的硬件支持，使在最小功耗下可以进行异步串行通信。

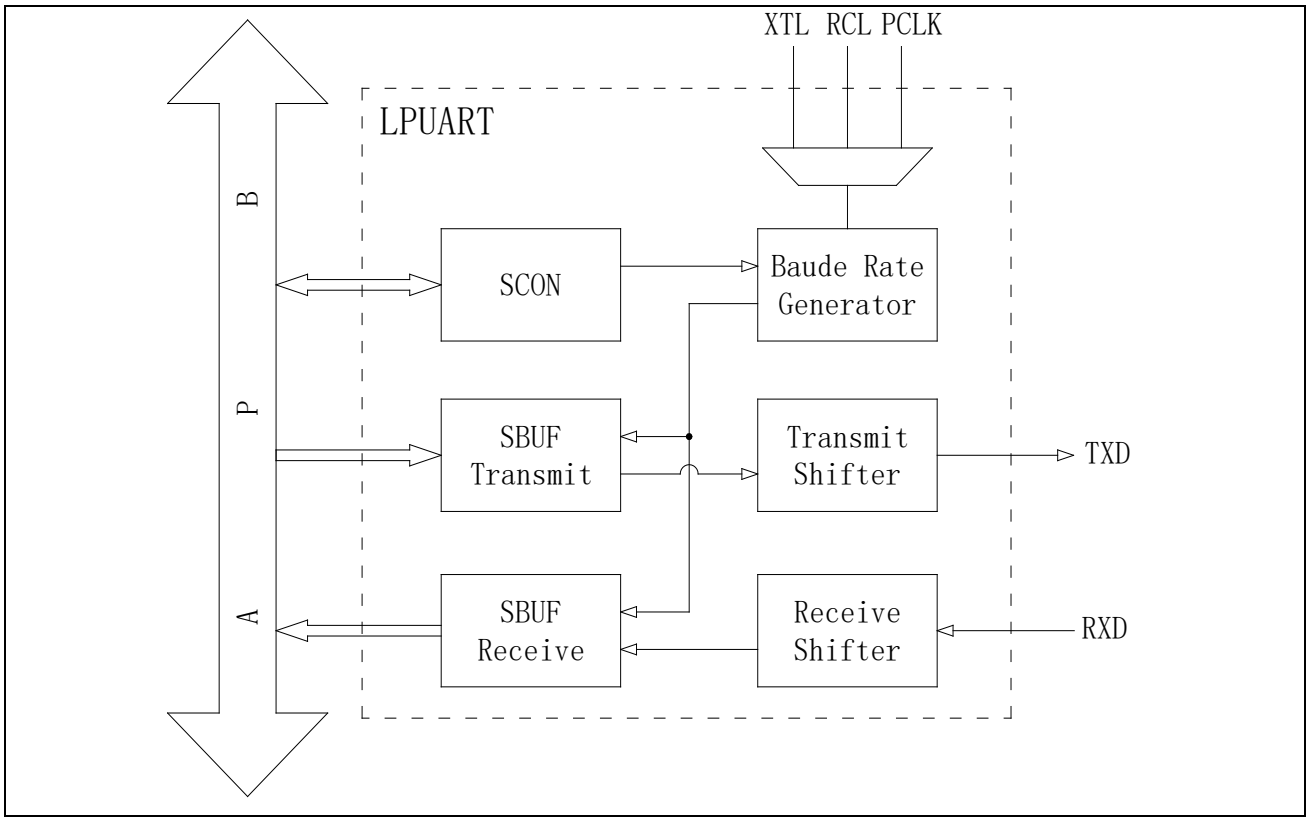


图 19-1 结构框图

19.2主要特性

LPUART 模块（LPUART0/1）支持以下基本功能：

- 配置时钟 PCLK
- 传输时钟 SCLK（SCLK 可选择 XTL、RCL 以及 PCLK）
- 系统低功耗模式下收发数据
- 全双工传输、半双工传输、单线半双工传输
- 可编程串行通信功能
  - 两种字符长度：8 比特、 9 比特
  - 三种校验方式：无检验、奇校验、偶校验
  - 三种停止长度：1 比特、 2 比特、 1.5 比特
- 16 比特波特率计数器
- 多机通讯
- 硬件地址识别
- 硬件流控
- DMAC 硬件传输握手

19.3功能描述

19.3.1 配置时钟和传输时钟

LPUART 模块有两个时钟：配置时钟 PCLK 和传输时钟 SCLK。

- 配置时钟

配置时钟（PCLK）用于系统 APB 总线对 LPUART 模块进行寄存器配置，配置时钟固定为 APB 总线时钟 PCLK。当系统进入深度休眠（DeepSleep）模式，PCLK 时钟将会停止。

- 传输时钟

传输时钟（SCLK）用于 LPUART 数据收发逻辑工作，可选择外部低速晶振时钟（XTL）、内部低速 RC 时钟（RCL）以及 PCLK 时钟。

当系统进入深度休眠（DeepSleep）模式，如果 SCLK 选择外部低速晶振时钟（XTL）或者内部低速 RC 时钟（RCL）。LPUART 仍旧可以进行正常的数据收发，而不受系统深度休眠（DeepSleep）模式的影响。

19.3.2 工作模式

LPUART 支持多种工作模式：同步半双工模式、异步全双工模式、单线半双工模式。通过 LPUARTx_SCON.SM 和 LPUARTx_SCON.HDSEL 搭配，即可配置出所需要的各种工作模式。

19.3.2.1 Mode0~Mode3 功能对比

配置 LPUARTx_SCON.SM 可选择不同的传输模式：Mode0~Mode3。这四种工作模式的主要功能对比如下表所示：

表 19-1 Mode0/1/2/3 数据结构

工作模式		传输位宽	数据组成	波特率
Mode0	同步模式 半双工	8-Bit	Data(8bit)	$BaudRate=\frac{f_{sclk}}{12}$
Mode1	异步模式 全双工	10-Bit	Start(1bit) + Data(8bit) + Stop (1~2bit)	$BaudRate=\frac{f_{sclk}}{OVER*SCNT}$
Mode2	异步模式 全双工	11-Bit	Start (1bit) + Data(8bit) + B8(1bit) + Stop (1~2bit)	$BaudRate=\frac{f_{sclk}}{OVER}$
Mode3	异步模式 全双工	11-Bit	Start (1bit) + Data(8bit) + B8(1bit) + Stop (1~2bit)	$BaudRate=\frac{f_{sclk}}{OVER*SCNT}$

注：

- Mode0 只能作为主机发送 LPUART 同步移位时钟，不可以作为从机接收外部输入的 LPUART 同步移位时钟；
- f_{sclk}为 SCLK 的时钟频率；
- OVER 的定义详见 LPUARTx_SCON；
- SCNT 的定义详见 LPUARTx_SCNT；
- B8 数据位比较特殊，在不同应用下具有不同的含义，请参考以下表格：

表 19-2 B8 数据含义

应用场景	LPUARTx_SCON. ADRDET	LPUARTx_SCON. B8CONT[1:0]	B8 数据含义
奇偶校验	--	01/10	接收时，B8 是所收到的 8-Bit 数据的奇偶校验位； 发送时，B8 是所发送的 8-Bit 数据的奇偶校验位；
多机通讯	1	--	B8=1，代表当前是地址帧； B8=0，代表当前是数据帧；
其他	0	00/11	接收/发送的 DATA[8]

注意：

- 当开启多机通讯模式，接收数据奇偶校验自动关闭；发送数据奇偶校验仍受 B8CONT 控制。

19.3.2.2 Mode0 数据收发说明

发送数据时，清除 LPUARTx_SCON.REN 位，并将数据写入 LPUARTx_SBUF 寄存器中。此时，发送数据将从 RXD 输出（低位在先，高位在后），同步移位时钟从 TXD 输出。

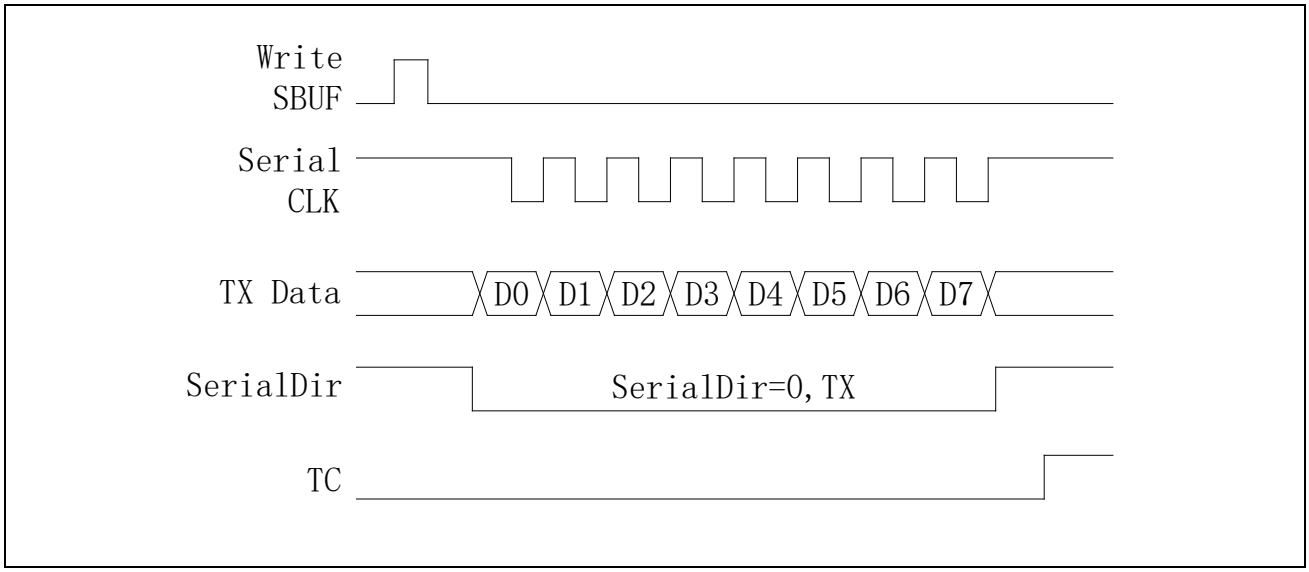


图 19-2 Mode0 发送数据

接收数据时，将 LPUARTx_SCON.REN 位置 1，并将 LPUARTx_ISR.RC 位清零。当接收结束，数据可从 LPUARTx_SBUF 寄存器读取。此时，接收数据从 RXD 输入（低位在先，高位在后），同步移位时钟从 TXD 输出。

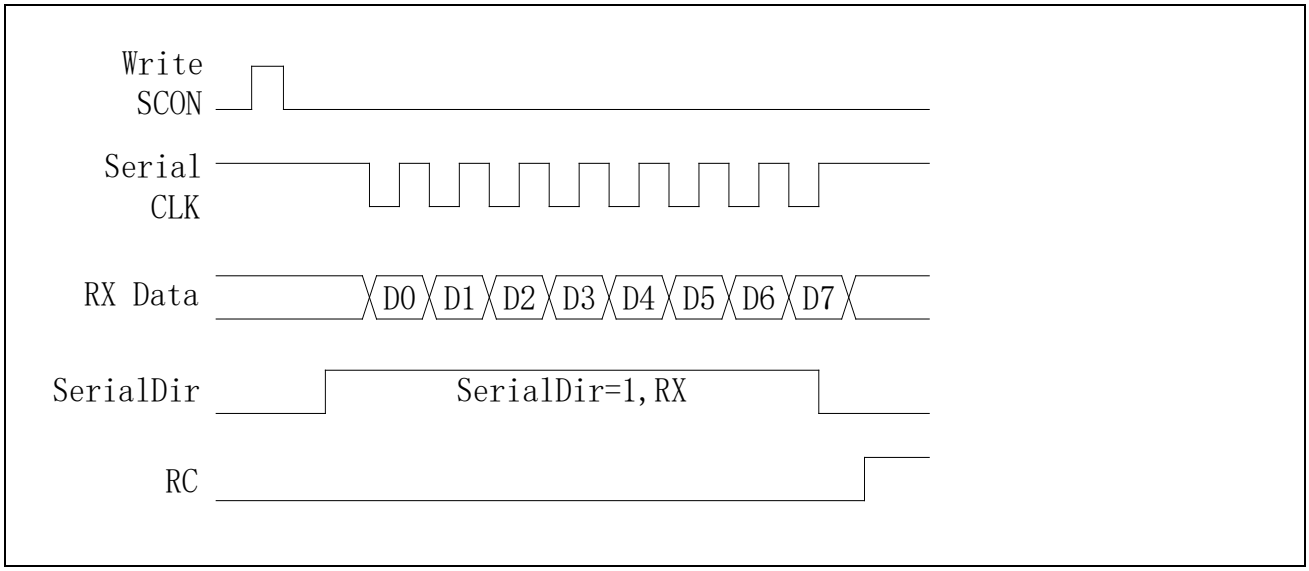


图 19-3 Mode0 接收数据

19.3.2.3 Mode1 数据收发说明

发送数据时，与 LPUARTx_SCON.REN 的值无关，将所发送数据写入 LPUARTx_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

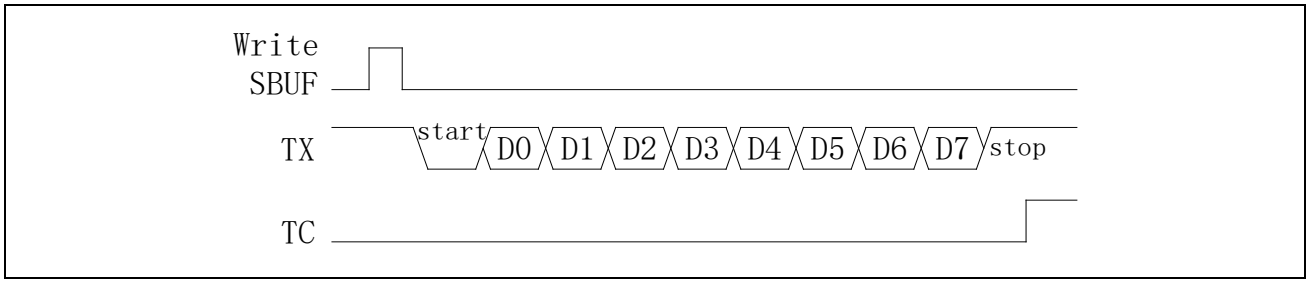


图 19-4 Mode1 发送数据

接收数据时，将 LPUARTx_SCON.REN 位置 1，并将 LPUARTx_ISR.RC 位清 0。开始接收 RXD 上数据（低位在先，高位在后），当接收完毕，可以从 LPUARTx_SBUF 寄存器读出。

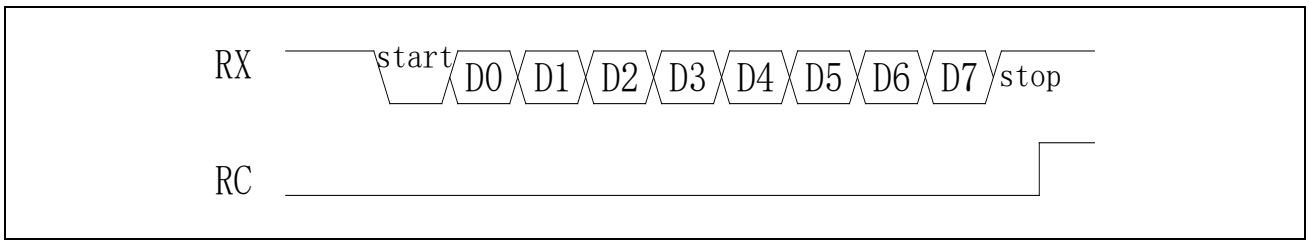


图 19-5 Mode1 接收数据

19.3.2.4 Mode2 数据收发说明

发送数据时，与 LPUARTx_SCON.REN 的值无关，将所发送数据写入 LPUARTx_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

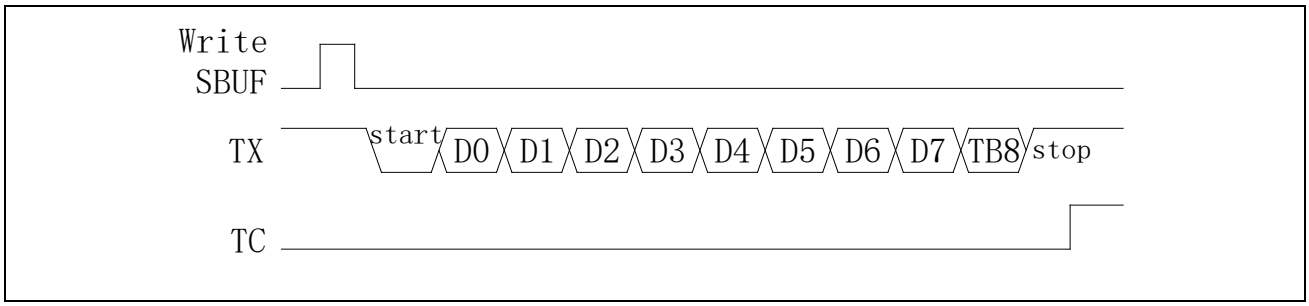


图 19-6 Mode2 发送数据

接收数据时，需将 LPUARTx_SCON.REN 位置 1，并将 LPUARTx_ISR.RC 位清 0。开始接收 RXD 上的数据（低位在先，高位在后），当接收完毕，可以从 LPUARTx_SBUF 寄存器读取。

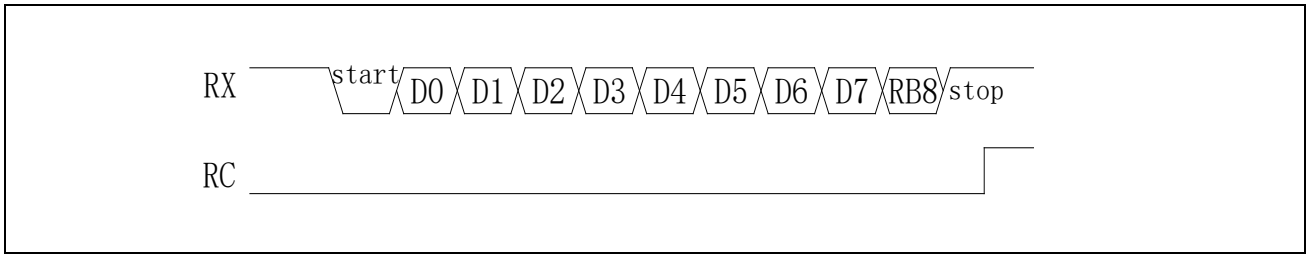


图 19-7 Mode2 接收数据

19.3.2.5 Mode3 数据收发说明

发送数据时，与 LPUARTx_SCON.REN 的值无关，将所发送数据写入 LPUARTx_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

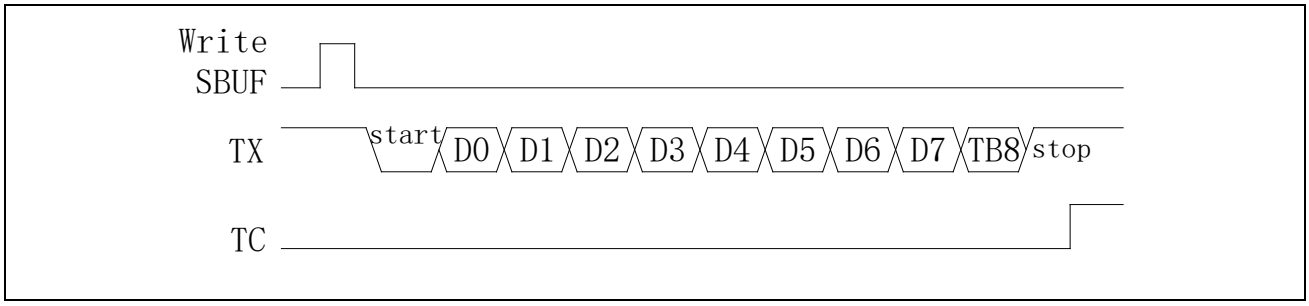


图 19-8 Mode3 发送数据

接收数据时，将 LPUARTx_SCON.REN 位置 1，并将 LPUARTx_ISR.RC 位清 0。开始接收 RXD 上数据（低位在先，高位在后），当接收完毕，可以从 LPUARTx_SBUF 寄存器读出。

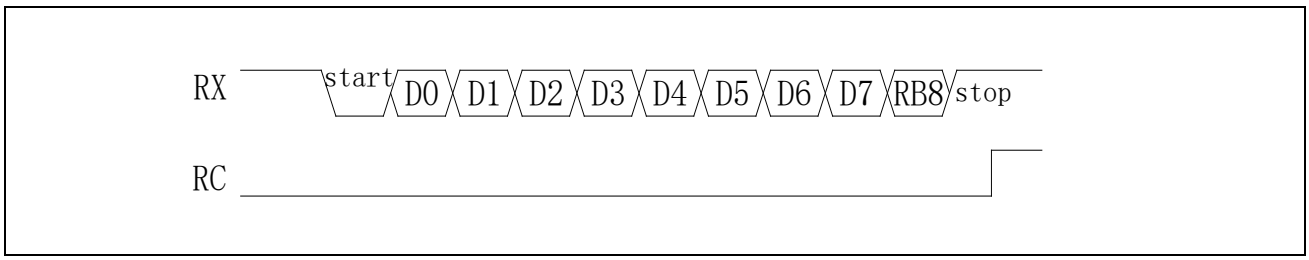


图 19-9 Mode3 接收数据

19.3.2.6 单线半双工数据收发说明

当 LPUARTx_SCON.HDSEL=1'b1，可进入单线半双工传输模式：

- TX 与 RX 信号线在模块内部相连。
- RX 信号不再使用，接收发送的数据都经由 TX 信号完成。TX 信号的方向控制由硬件逻辑完成，无需软件控制。
- 配置 LPUART_SCON.HDSEL=0b1 后，要等待一位数据的发送时长（即 1/BaudRate 秒后），才能正常接收。
- 当发送缓存为空时，TX 信号始终为输入（接收状态）。一旦向发送缓存填入一个数据，TX 信号变为输出（发送状态）。当发送完成，发送缓存变为空，TX 信号又回到输入（接收状态）。
- 无数据传输时，TXD 引脚始终处于输入状态。用户程序需配置该引脚为开漏输出并外接上拉电阻。

注意：

1. 即使模块处于接收过程中，只要发送缓存被填入数据，TX 信号立即变为输出（发送状态）。接收过程将被打断，并得到一个错误的接收数据。
2. 单线半双工模式与 Mode0 模式的异同：
   同：两者都是用一根信号分时复用地完成了数据收发，

异：Mode0 为同步传输，有同步时钟。而单线半双工模式为异步传输，所有的波特率计算还是遵照 Mode1~Mode3 的波特率公式，仅仅是收发功能复用到了 TX 这一根信号上面。

19.3.3 波特率生成

Mode0~Mode3 生成波特率的公式不尽相同，详见下方所示。

Mode0 波特率生成工式：$BaudRate=\frac{f_{SCLK}}{12}$

Mode1 波特率生成工式：$BaudRate=\frac{f_{SCLK}}{OVER*SCNT}$

Mode2 波特率生成工式：$BaudRate=\frac{f_{SCLK}}{OVER}$

Mode3 波特率生成工式：$BaudRate=\frac{f_{SCLK}}{OVER*SCNT}$

注：

- f_{SCLK}代表当前 SCLK 的 频率；
- OVER 的定义详见 LPUARTx_SCON；
- SCNT 的定义详见 LPUARTx_SCNT。

19.3.3.1 Mode1/Mode3 波特率设置示例

表 19-3 SCL 为 4MHz 波特率计算表

波特率	SCLK = 4 MHz								
	OVER4			OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	417	2398.08	-0.08%	208	2403.85	0.16%	104	2403.85	0.16%
4800	208	4807.69	0.16%	104	4807.69	0.16%	52	4807.69	0.16%
9600	104	9615.38	0.16%	52	9615.38	0.16%	26	9615.38	0.16%
19200	52	19230.77	0.16%	26	19230.77	0.16%	13	19230.77	0.16%
38400	26	38461.54	0.16%	13	38461.54	0.16%	7	35714.29	-6.99%
57600	17	58823.53	2.12%	9	55555.56	-3.55%	4	62500.00	8.51%
76800	13	76923.08	0.16%	7	71428.57	-6.99%	3	83333.33	8.51%
115200	9	111111.11	-3.55%	4	125000.00	8.51%	2	125000.00	8.51%
128000	8	125000.00	-2.34%	4	125000.00	-2.34%	2	125000.00	-2.34%
256000	4	250000.00	-2.34%	2	250000.00	-2.34%	1	250000.00	-2.34%
1000000	1	1000000.00	0.00%	1	500000.00	-50.00%	0	/	/

表 19-4 SCLK 为 8MHz 波特率计算表

波特率	SCLK = 8 MHz								
	OVER4			OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	833	2400.96	0.04%	417	2398.08	-0.08%	208	2403.85	0.16%
4800	417	4796.16	-0.08%	208	4807.69	0.16%	104	4807.69	0.16%
9600	208	9615.38	0.16%	104	9615.38	0.16%	52	9615.38	0.16%
19200	104	19230.77	0.16%	52	19230.77	0.16%	26	19230.77	0.16%
38400	52	38461.54	0.16%	26	38461.54	0.16%	13	38461.54	0.16%
57600	35	57142.86	-0.79%	17	58823.53	2.12%	9	55555.56	-3.55%
76800	26	76923.08	0.16%	13	76923.08	0.16%	7	71428.57	-6.99%
115200	17	117647.06	2.12%	9	111111.11	-3.55%	4	125000.00	8.51%
128000	16	125000.00	-2.34%	8	125000.00	-2.34%	4	125000.00	-2.34%
256000	8	250000.00	-2.34%	4	250000.00	-2.34%	2	250000.00	-2.34%
1000000	2	1000000.00	0.00%	1	1000000.00	0.00%	1	500000.00	-50.00%
2000000	1	2000000.00	0.00%	1	1000000.00	-50.00%	0	/	/

表 19-5 SCLK 为 16MHz 波特率计算表

波特率	SCLK = 16 MHz								
	OVER4			OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	1667	2399.52	-0.02%	833	2400.96	0.04%	417	2398.08	-0.08%
4800	833	4801.92	0.04%	417	4796.16	-0.08%	208	4807.69	0.16%
9600	417	9592.33	-0.08%	208	9615.38	0.16%	104	9615.38	0.16%
19200	208	19230.77	0.16%	104	19230.77	0.16%	52	19230.77	0.16%
38400	104	38461.54	0.16%	52	38461.54	0.16%	26	38461.54	0.16%
57600	69	57971.01	0.64%	35	57142.86	-0.79%	17	58823.53	2.12%
76800	52	76923.08	0.16%	26	76923.08	0.16%	13	76923.08	0.16%
115200	35	114285.71	-0.79%	17	117647.06	2.12%	9	111111.11	-3.55%
128000	31	129032.26	0.81%	16	125000.00	-2.34%	8	125000.00	-2.34%
256000	16	250000.00	-2.34%	8	250000.00	-2.34%	4	250000.00	-2.34%
1000000	4	1000000.00	0.00%	2	1000000.00	0.00%	1	1000000.00	0.00%
2000000	2	2000000.00	0.00%	1	2000000.00	0.00%	1	1000000.00	-50.00%
4000000	1	4000000.00	0.00%	1	2000000.00	-50.00%	0	/	/

表 19-6 SCLK 为 24MHz 波特率计算表

波特率	SCLK = 24 MHz								
	OVER4			OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	2500	2400.00	0.00%	1250	2400.00	0.00%	625	2400.00	0.00%
4800	1250	4800.00	0.00%	625	4800.00	0.00%	313	4792.33	-0.16%
9600	625	9600.00	0.00%	313	9584.66	-0.16%	156	9615.38	0.16%
19200	313	19169.33	-0.16%	156	19230.77	0.16%	78	19230.77	0.16%
38400	156	38461.54	0.16%	78	38461.54	0.16%	39	38461.54	0.16%
57600	104	57692.31	0.16%	52	57692.31	0.16%	26	57692.31	0.16%
76800	78	76923.08	0.16%	39	76923.08	0.16%	20	75000.00	-2.34%
115200	52	115384.62	0.16%	26	115384.62	0.16%	13	115384.62	0.16%
128000	47	127659.57	-0.27%	23	130434.78	1.90%	12	125000.00	-2.34%
256000	23	260869.57	1.90%	12	250000.00	-2.34%	6	250000.00	-2.34%
1000000	6	1000000.00	0.00%	3	1000000.00	0.00%	2	750000.00	-25.00%
2000000	3	2000000.00	0.00%	2	1500000.00	-25.00%	1	1500000.00	-25.00%
6000000	1	6000000.00	0.00%	1	3000000.00	-50.00%	0	/	/

表 19-7 SCLK 为 32MHz 波特率计算表

波特率	SCLK = 32 MHz								
	OVER4			OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	3333	2400.24	0.01%	1667	2399.52	-0.02%	833	2400.96	0.04%
4800	1667	4799.04	-0.02%	833	4801.92	0.04%	417	4796.16	-0.08%
9600	833	9603.84	0.04%	417	9592.33	-0.08%	208	9615.38	0.16%
19200	417	19184.65	-0.08%	208	19230.77	0.16%	104	19230.77	0.16%
38400	208	38461.54	0.16%	104	38461.54	0.16%	52	38461.54	0.16%
57600	139	57553.96	-0.08%	69	57971.01	0.64%	35	57142.86	-0.79%
76800	104	76923.08	0.16%	52	76923.08	0.16%	26	76923.08	0.16%
115200	69	115942.03	0.64%	35	114285.71	-0.79%	17	117647.06	2.12%
128000	63	126984.13	-0.79%	31	129032.26	0.81%	16	125000.00	-2.34%
256000	31	258064.52	0.81%	16	250000.00	-2.34%	8	250000.00	-2.34%
1000000	8	1000000.00	0.00%	4	1000000.00	0.00%	2	1000000.00	0.00%
2000000	4	2000000.00	0.00%	2	2000000.00	0.00%	1	2000000.00	0.00%
4000000	2	4000000.00	0.00%	1	4000000.00	0.00%	1	2000000.00	-50.00%
8000000	1	8000000.00	0.00%	1	4000000.00	-50.00%	0	/	/

表 19-8 SCLK 为 48MHz 波特率计算表

波特率	SCLK = 48 MHz								
	OVER4			OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	5000	2400.00	0.00%	2500	2400.00	0.00%	1250	2400.00	0.00%
4800	2500	4800.00	0.00%	1250	4800.00	0.00%	625	4800.00	0.00%
9600	1250	9600.00	0.00%	625	9600.00	0.00%	313	9584.66	-0.16%
19200	625	19200.00	0.00%	313	19169.33	-0.16%	156	19230.77	0.16%
38400	313	38338.66	-0.16%	156	38461.54	0.16%	78	38461.54	0.16%
57600	208	57692.31	0.16%	104	57692.31	0.16%	52	57692.31	0.16%
76800	156	76923.08	0.16%	78	76923.08	0.16%	39	76923.08	0.16%
115200	104	115384.62	0.16%	52	115384.62	0.16%	26	115384.62	0.16%
128000	94	127659.57	-0.27%	47	127659.57	-0.27%	23	130434.78	1.90%
256000	47	255319.15	-0.27%	23	260869.57	1.90%	12	250000.00	-2.34%
1000000	12	1000000.00	0.00%	6	1000000.00	0.00%	3	1000000.00	0.00%
2000000	6	2000000.00	0.00%	3	2000000.00	0.00%	2	1500000.00	-25.00%
4000000	4	3000000.00	0.00%	2	3000000.00	0.00%	1	3000000.00	0.00%
6000000	3	4000000.00	0.00%	2	3000000.00	/	1	3000000.00	/
12000000	1	12000000.00	0.00%	1	6000000.00	/	0	/	/

19.3.4 帧错误检测

当工作在 Mode1/2/3 时，LPUART 具有帧错误检测功能，如果接收数据时未在预期时间内识别出停止位，从而出现同步失效或过度的噪声，则会检测到帧错误。检测帧错误时，LPUARTx_ISR.FE 置 1。LPUARTx_ISR.FE 要及时软件清零。

19.3.5 多机通讯

当工作在 Mode2/3 时，将 LPUARTx_SCON.ADRDET 位置“1”，可开启多机通讯功能。

主机可以通过 LPUARTx_SBUF[8]来区分当前的发送帧是地址帧（LPUARTx_SBUF[8]=1）还是数据帧（LPUARTx_SBUF[8]=0）。

- 当为数据帧时，该帧数据不会存入到从机的 LPUARTx_SBUF 寄存器中，从机也不会产生接收中断。
- 当为地址帧时，由于多机通讯中自动地址识别功能已开启，使得从机可以检测接收到的地址与其自身地址是否相符合。
  - 如果地址相符合，从机会对 LPUARTx_ISR.RC 置“1”， LPUARTx_SBUF[8]置“1”，同时将该地址帧存入到 LPUARTx_SBUF 寄存器中。从机软件看到 LPUARTx_SBUF[8]=1 并且 LPUARTx_ISR.RC=1 后，将 LPUARTx_SCON.ADEDET 位清“0”，接受数据帧。

- 如果地址不符合，表明主机并不是寻址该从机，从机硬件保持 LPUARTx_SBUF[8]和 LPUARTx_ISR.RC 为“0”，软件保持 LPUARTx_SCON.ADRDET 位为“1”，从机继续处于地址监听状态。

注：

- 如果有需要，也可以在 Mode1 下开启多机通讯位，此时 TB8 位由 stop 位代替。当从机接收到匹配的地址帧和有效的 stop 位时，LPUARTx_ISR.RC 会被置“1”。

从机地址配置

从机地址由广播地址和用户配置地址组成：

- 广播地址：LPUART_SADDR | LPUART_SADEN
- 用户配置地址：LPUART_SADDR & LPUART_SADEN

收到地址帧时，检测到从总线收到的地址与 LPUART_SADEN 的值相与后与用户配置地址相同，或收到的地址与广播地址相与后与广播地址相同，则认为收到了与从机地址相匹配的地址帧。

例：LPUART_SADDR=0x55，LPUART_SADEN=0xF0，则从机可以匹配的地址为 0xF5、0xF7、0xFD、0xFF（广播地址）和 0x50~0x5F（用户配置地址）。

19.3.6 DMAC 硬件握手

LPUART 模块支持 DMAC 的硬件握手逻辑。

- 将 LPUARTx_SCON.DMACTXEN 设置为 1，可以打开 LPUART TX 的 DMAC 硬件握手逻辑。当发送缓存为空时，LPUART 会向 DMAC 发出数据搬运请求 TX REQ。DMAC 收到该信号，则从 DMAC 源地址搬运一个帧的发送数据到 LPUARTx_SBUF 中。上述步骤重复发生，直到 DMAC 中所配置的数据长度全部搬运完毕。
- 将 LPUARTx_SCON.DMACRXEN 设置为 1，可以打开 LPUART RX 的 DMAC 硬件握手逻辑。当一帧接收完成，LPUART 会向 DMAC 发出数据搬运请求 RX REQ。DMAC 收到该信号，则从 LPUARTx_SBUF 中把接收数据搬运至 DMAC 的目标地址中。上述步骤重复发生，直到 DMAC 中所配置的数据长度全部搬运完毕。

19.3.7 硬件流控

通过增加 nCTS 和 nRTS 信号可以实现 LPUART 硬件流控的功能，即 LPUART 硬件模块根据 nCTS 和 nRTS 的高低电平自动控制数据的收发，而无需通过软件来判断。两个 LPUART 模块之间的硬件流控示意图如下所示：

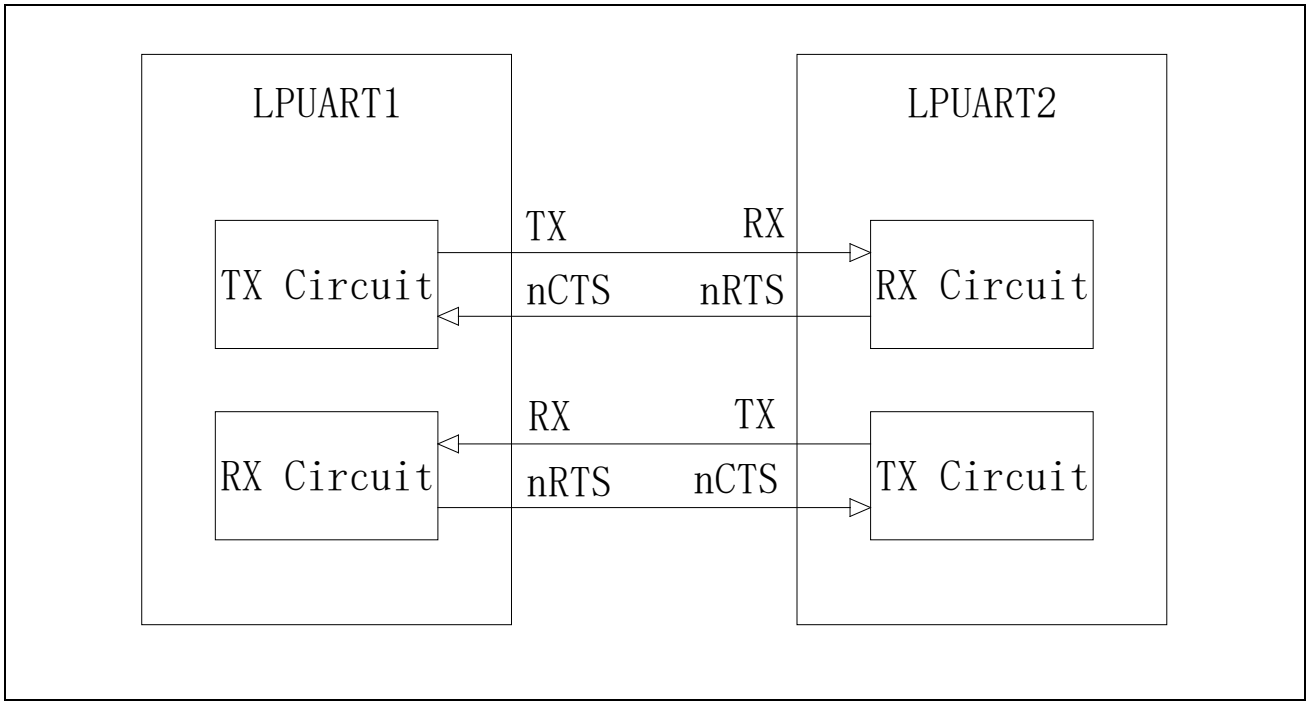


图 19-10 LPUART 硬件流控

■ nRTS 流控

nRTS 流控使能时（LPUARTx_SCON.RTSEN 设置为 1）：

- 当 LPUART 接收缓存空时，会将 nRTS 变为有效（连接到低电平）。
- 当接收缓存满时，会将 nRTS 变为无效（连接到高电平），表明发送过程会在当前帧结束后停止。

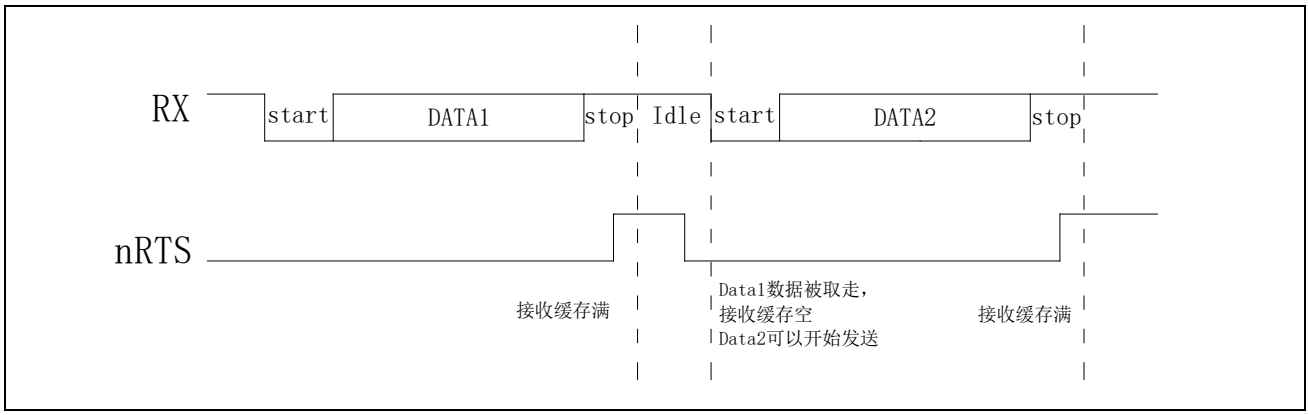


图 19-11 nRTS 硬件流控信号

■ CTS 流控

当 nCTS 流控使能时（LPUARTx_SCON.CTSEN 设置为 1），在 LPUART 发送下一帧数据之前，先判断 nCTS 的高低电平：

- 如果 nCTS 为有效（连接到低电平），则 LPUART 发送下一帧数据。
- 如果 nCTS 为无效（连接到高电平），则 LPUART 当前帧发送完成后，暂停下一帧的发送

当 nCTS 流控使能时，一旦 nCTS 信号发生翻转，LPUARTx_SFLAG.CTSIF 便会硬件置 1。如果 LPUARTx_SCON.CTSIE 置 1,则会产生中断。同时,nCTS 信号的高低电平会记录在 LPUARTx_SFLAG.CTS 标志位中。

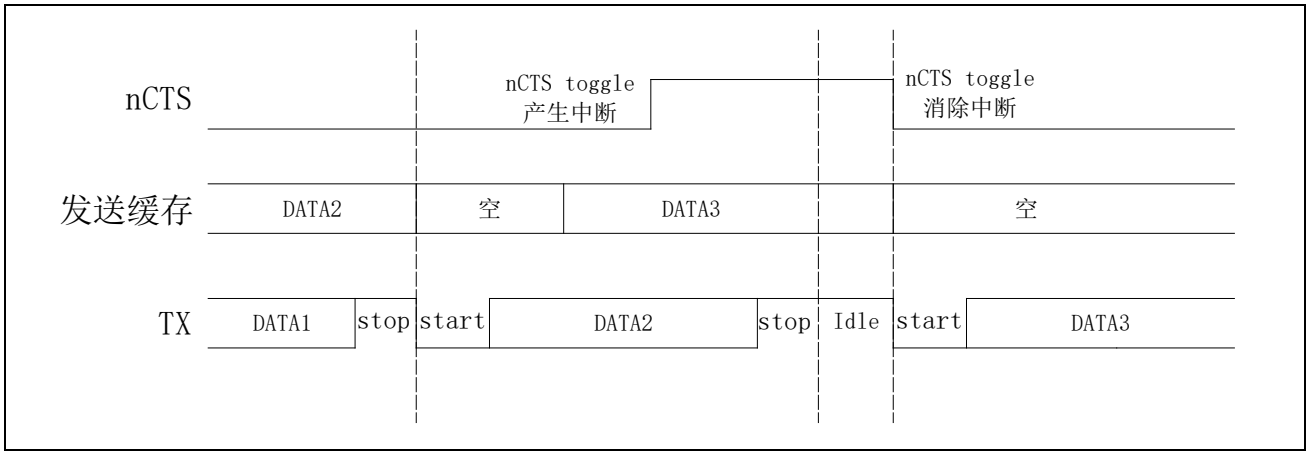


图 19-12 nCTS 硬件流控信号

19.3.8 收发端缓存

■ 接收缓存

LPUART 模块接收端有一个帧（8/9-Bit）的接收缓存，即保持接受的数据帧直至下一帧数据的 Stop 位被接才更新该数据帧。

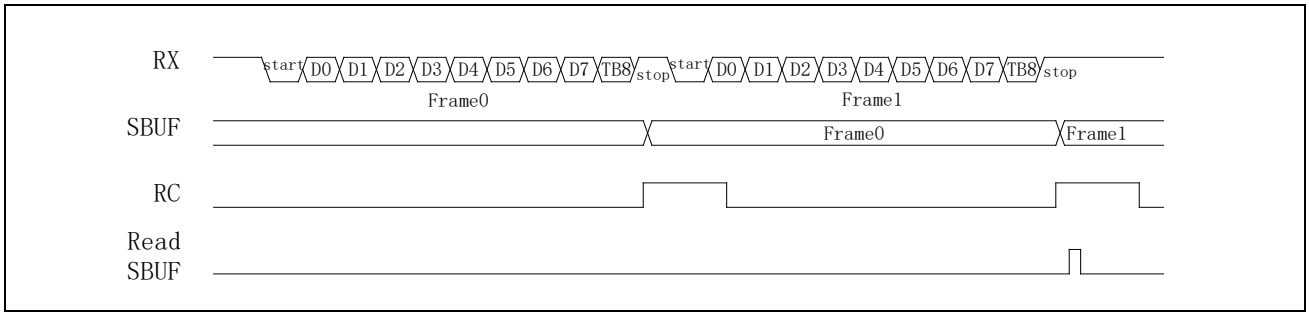


图 19-13 接收缓存

■ 发送缓存

LPUART 模块发送端有一个帧(8/9-Bit)的发送缓存，当 LPUART 在发送当前帧时，软件写入下一个发送数据到 LPUARTx_SBUF。

当 LPUARTx_ISR.TXE=0 时，表明当前发送缓存满， LPUARTx_SBUF 不能写入下一个发送数据。否则该数据会被硬件丢弃。

当 LPUARTx_ISR.TXE=1 时，表明当前发送缓存空，LPUARTx_SBUF 可以写入下一个发送数据，在完成当前数据传输后，硬件自动把发送缓存中的数据装载入移位寄存器中发送出去。

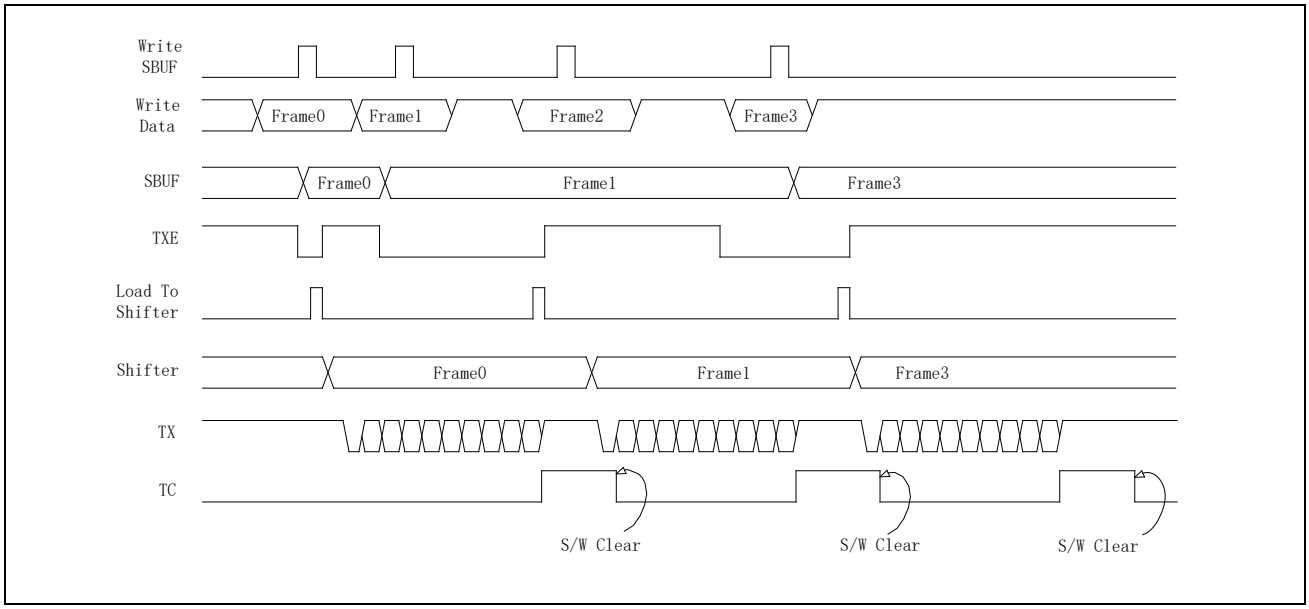

Write SBUF
Write Data
Frame0
Frame1
Frame2
Frame3
SBUF
Frame0
Frame1
Frame3
TXE
Load To Shifter
Shifter
Frame0
Frame1
Frame3
TX
TC
S/W Clear
S/W Clear
S/W Clear

图 19-14　发送缓存

19.4寄存器

LPUART0 基地址：0x4000 0200

LPUART1 基地址：0x4000 4000

寄存器	偏移地址	描述
LPUARTx_SBUF	0x00	数据寄存器
LPUARTx_SCON	0x04	控制寄存器
LPUARTx_SADDR	0x08	地址寄存器
LPUARTx_SADEN	0x0C	地址掩码寄存器
LPUARTx_ISR	0x10	中断标志位寄存器
LPUARTx_ICR	0x14	中断标志位清除寄存器
LPUARTx_SCNT	0x18	波特率寄存器

19.4.1 数据寄存器（LPUARTx_SBUF）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							DATA[8]	DATA[7:0]							
							RW	RW							

位	标记	功能描述
31:9	Reserved	
8	DATA[8]	在Mode0/1下，读取该位为0，写入该位无效； 在Mode2/3下，该位表示Bit8数据位，分以下两种情况： （1）当硬件奇偶校验位开启时，接收时该位为接收数据奇偶校验位，校验由硬件进行，如校验出错，校验错误标志位PE置1；发送时该位无效，发送数据奇偶校验位由硬件计算并发送； （2）当硬件奇偶校验位关闭时，接收时该位为接收数据Bit8；发送时该位为发送数据Bit8； 注意：当开启多机通讯模式，接收数据奇偶校验自动关闭；发送数据奇偶校验仍受B8CONT控制；
7:0	DATA[7:0]	发送数据时，将发送数据写入该寄存器；接收数据时，数据接收完毕后，从该寄存器中读取。

19.4.2 控制寄存器（LPUARTx_SCON）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									HDSEL	FEIE	CTSIE	CTSEN	RTSEN	DMATXEN	DMARXEN
									RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOPBIT		PEIE	SCLKSEL			OVER	TXEIE	SM		ADRDET	REN	B8CONT		TCIE	RCIE
RW		RW	RW			RW	RW	RW		RW	RW	RW		RW	RW

位	标记	功能描述
31:23	Reserved	
22	HDSEL	单线半双工模式是能；0: 关闭单线半双工模式；1：打开单线半双工模式；
21	FEIE	帧错误中断使能；0：关闭中断；1：打开中断；
20	CTSIE	CTS信号翻转中断使能；0：关闭中断；1：打开中断；
19	CTSEN	硬件flow control信号使能位；0：关闭flow control信号；1：打开flow control信号；
18	RTSEN	
17	DMATXEN	TX DMAC的硬件握手信号使能位；0：关闭硬件握手信号；1：打开硬件握手信号；
16	DMARXEN	RX DMAC的硬件握手信号使能位；0：关闭硬件握手信号；1：打开硬件握手信号；
15:14	STOPBIT	stop bit长度选择；00:1-bit；01:1.5-bit；10:2-bit；11：reserved； 注意：Mode0时虽然没有Stop Bit，但仍需把STOPBIT[1:0]保持为2’b00；
13	PEIE	奇偶校验错误中断使能位；0：奇偶校验错误中断关闭；1：奇偶校验错误中断打开； 在系统时钟使用高速时钟与LPUART使用低速时钟情况下，由于需要不同时钟域的同步，硬件奇偶校验与数据接收中断由于停止位设置不同，奇偶校验错误中断会提前或落后于数据接收中断。使用奇偶校验中断使能时请注意。 建议如下： 关闭PEIE中断使能，接收数据中断后通过接收的数据位SBUF.BIT8软件判断是否出现奇偶校验错误。
12:11	SCLKSEL	传输时钟选择位：00，01：PCLK；10：XTL；11：RCL；
10:9	OVER	Mode0：　无效； Mode1/3：　00：16采样分频；　01：8采样分频；　10：4采样分频；11：保留； Mode2：　00：32采样分频；　01：16采样分频；10：8采样分频；11：保留；
8	TXEIE	TX空中断使能位；0：TX Buffer空中断关闭；1：TX Buffer空中断打开；
7:6	SM	工作模式；00：mode0；01：mode1；10：mode2；11：mode3；
5	ADRDET	多机通讯地址自动识别使能位；0：关闭；1：打开；
4	REN	Mode0：　0：发送；1：接收； Mode1/2/3：0：发送；1：接收/发送；

3:2	B8CONT	Bit8数据控制位； 00：由软件读写SBUF[8]来决定；01：硬件偶校验； 10：硬件奇校验；　　11：保留；
1	TCIE	发送中断使能位；0：发送中断关闭；1：发送中断打开；
0	RCIE	接收中断使能位；0：接收中断关闭；1：接收中断打开；

19.4.3 地址寄存器（LPUARTx_SADDR）

偏移地址：0x08

复位值：0x0000 0000

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

位	标记	功能描述
31:8	Reserved	
7:0	SADDR	从机设备地址寄存器

19.4.4 地址掩码寄存器（LPUARTx_SADEN）

偏移地址：0x0C

复位值：0x0000 0000

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

位	标记	功能描述
31:8	Reserved	
7:0	SADEN	从机设备地址掩码寄存器

19.4.5 标志位寄存器（LPUARTx_ISR）

偏移地址：0x10

复位值：0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									CTS	CTSIF	PE	TXE	FE	TC	RC
									RO	RO	RO	RO	RO	RO	RO

位	符号	描述
31:7	Reserved	
6	CTS	CTS信号标志位；硬件置1；硬件清零；0：CTS信号为低电平；1：CTS信号为高电平；
5	CTSIF	CTS中断标志位；硬件置1；软件清零；0：CTS信号没有发生反转；1：CTS信号发生反转；
4	PE	奇偶校验错误标志位；硬件置1；软件清零；0：无奇偶校验错误；1：奇偶校验错误；
3	TXE	Tx Buffer空标志位；硬件置1；硬件清零；0：Tx Buffer非空；1：Tx Buffer空
2	FE	帧错误标志位；0：硬件置1；软件清零；
1	TC	发送完毕中断标志位；硬件置1；软件清零；0：发送未完成；1：发送完毕；
0	RC	接收完毕中断标志位；硬件置1；软件清零；0：接收未完成；1：接收完成；

19.4.6 标志位清除寄存器（LPUARTx_ICR）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										CTSIFCF	PECF	Res.	FECF	TCCF	RCCF
										R1W0	R1W0		R1W0	R1W0	R1W0

位	标记	功能描述
31:6	Reserved	
5	CTSIFCF	CTSIF标志清楚位；写0清除；写1无效；
4	PECF	PE标志清楚位；写0清除；写1无效；
3	Reserved	
2	FECF	FE标志清除位；写0清零；写1无效；
1	TCCF	TC标志清除位；写0清零；写1无效；
0	RCCF	RC标志清除位；写0清零；写1无效；

19.4.7 波特率寄存器（LPUARTx_SCNT）

偏移地址：0x18

复位值：0x0000 0000

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

位	标记	功能描述
31:16	Reserved	
15:0	SCNT	波特率计数器

20 循环冗余校验（CRC）

20.1概述

循环冗余校验（CRC）计算单元将数据流或数据块作为输入，在生成多项式的控制下生成一个输出数。该输出数常用于验证数据传输或存储的正确性和完整性。本模块支持计算 CRC 值和检验 CRC 值。

20.2主要特性

- 一种执行标准：ISO/IEC13239
- 两种编码方式：CRC-16，CRC-32
- 三种写入位宽：8bit，16bit，32bit
- 两种工作模式：CRC 编码模式、CRC 校验模式
- CRC-16 多项式：$x^{16}+x^{12}+x^5+1$
- CRC-32 多项式：$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

20.3功能描述

20.3.1 工作模式

本模块支持两种工作模式：CRC 编码模式、CRC 校验模式。

CRC 编码模式是指向 CRC 模块输入一定数量的原始数据，获取 CRC 模块生成的输出值（CRC_RESULT）。CRC 校验模式是指向 CRC 模块输入一定数量的原始数据+CRC 校验值，验证原始数据与 CRC 校验值是否匹配（CRC_CR.FLAG）。

20.3.2 编码方式

本模块支持两种编码方式 CRC-16、CRC-32，其计算结果分别为 16 比特和 32 比特。通过 CRC_CR.CR 配置需要使用的编码方式。

CRC-16 多项式：$x^{16}+x^{12}+x^5+1$。

CRC-32 多项式：$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$。

20.3.3 写入位宽

本模块支持三种写入位宽：8bit，16bit，32bit。不同位宽的写入需要符合“位宽一致、先低后高”的原则，即“每次写入数据，都必须写入到与本次有效数据位宽相等的寄存器中，并且较低位数据的写入先于较高位数据”。

下方展示了同一序列数据采用三种位宽进行写入的方法，其输出结果相同。

- 8bit 位宽写入：0x00, 0x11, 0x22 , 0x33, 0x44, 0x55, 0x66, 0x77

- 16bit 位宽写入：0x1100, 0x3322, 0x5544, 0x7766
- 32bit 位宽写入：0x33221100, 0x77665544

20.4编程示例

20.4.1 CRC-16 编码模式

Step 1： 向 CRC_CR.CR 写入 0x00，选择 CRC-16。

Step 2： 向 CRC_RESULT 写入 0xFFFF，初始化 CRC 计算。

Step 3： 将待编码的原始数据依次写入 CRC_DATA 寄存器，写入位宽可选择 8bit、16bit、32bit。

Step 4： 读取 CRC_RESULT[15:0]以获取 CRC 值。

20.4.2 CRC-16 检验模式

Step 1： 向 CRC_CR.CR 写入 0x00，选择 CRC-16。

Step 2： 向 CRC_RESULT 写入 0xFFFF，初始化 CRC 计算。

Step 3： 将已编码的数据序列依次写入 CRC_DATA 寄存器，写入位宽可选择 8bit、16bit、32bit。

Step 4： 根据 CRC_CR.FLAG 的值判定已编码的数据序列是否被篡改。

20.4.3 CRC-32 编码模式

Step 5： 向 CRC_CR.CR 写入 0x01，选择 CRC-32。

Step 6： 向 CRC_RESULT 写入 0xFFFFFFFF，初始化 CRC 计算。

Step 7： 将待编码的原始数据依次写入 CRC_DATA 寄存器，写入位宽可选择 8bit、16bit、32bit。

Step 8： 读取 CRC_RESULT[31:0]以获取 CRC 值。

20.4.4 CRC-32 检验模式

Step 5： 向 CRC_CR.CR 写入 0x01，选择 CRC-32。

Step 6： 向 CRC_RESULT 写入 0xFFFFFFFF，初始化 CRC 计算。

Step 7： 将已编码的数据序列依次写入 CRC_DATA 寄存器，写入位宽可选择 8bit、16bit、32bit。

Step 8： 根据 CRC_CR.FLAG 的值判定已编码的数据序列是否被篡改。

20.5寄存器描述

20.5.1 寄存器列表

基地址：0x4002 0900

寄存器	偏移地址	描述
CRC_CR	0x00	CRC 控制寄存器
CRC_RESULT	0x04	CRC 结果寄存器
CRC_DATA	0x80	CRC 数据寄存器

20.5.2 控制寄存器（CRC_CR）

偏移地址：0x00

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														FLAG	CR
R														RO	RW

位	符号	功能描述
31:2	Reserved	
1	FLAG	CRC校验结果 0：当前CRC校验错误 1：当前CRC校验正确
0	CR	CRC编码方式选择 0：CRC-16编码 1：CRC-32编码

20.5.3 结果寄存器（CRC_RESULT）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESULT[31:16]															
RW															

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

位	符号	描述
31:0	RESULT	CRC计算结果 读取RESULT[15:0]以获取CRC-16的计算结果 读取RESULT[31:0]以获取CRC-32的计算结果 向RESULT[15:0]写入0xFFFF以初始化CRC-16计算 向RESULT[31:0]写入0xFFFFFFFF以初始化CRC-32计算

20.5.4 数据寄存器（CRC_DATA）

偏移地址：0x80

复位值：0x0000 0000

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

位	符号	功能描述
31:0	DATA	本寄存器用于写入需要运算的数据，支持3种写入位宽 8bit写入方式： * ((uint8_t *)0x40020980) = 0xXX 16bit写入方式： * ((uint16_t *)0x40020980) = 0xXXXX 32bit写入方式： * ((uint32_t *)0x40020980) = 0xXXXXXXXX

21 真随机数发生器（TRNG）

21.1概述

真随机数模块产生 64 位真随机数。

21.2功能框图

以下示意了 TRNG 模块的数据流：

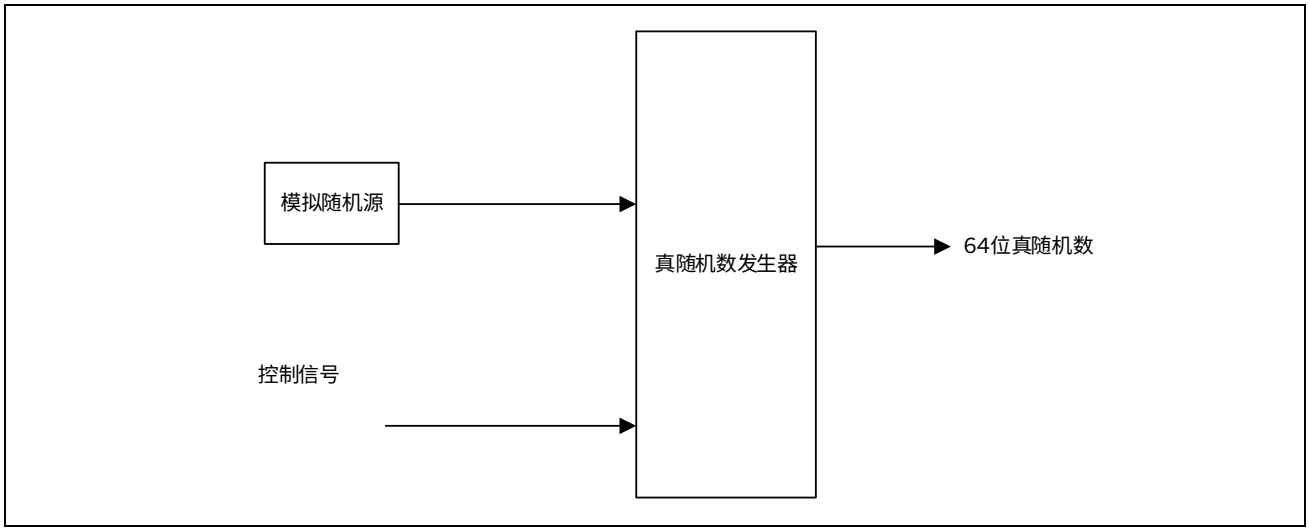


图 21-1 TRNG 数据流

21.3功能描述

本模块采用内部的模拟随机源，每次启动都可以产生 64bits 真随机数。此外，还可以对真随机数生成的方式进行软件配置，详细内容可查看寄存器描述章节。生成的 64 位真随机数分别存放在 DATA0 和 DATA1 寄存器中。

21.4寄存器

基地址：0x4000 4C00

寄存器	偏移地址	描述
TRNG_CR	0x00	控制寄存器
TRNG_MODE	0x04	模式寄存器
TRNG_DATA0	0x0C	数据寄存器 0
TRNG_DATA1	0x10	数据寄存器 1

21.4.1 控制寄存器（TRNG_CR）

偏移地址：0x00

复位值：0x0000 0000

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

位	标记	功能描述
31:2	Reserved	
1	TRNG_RUN	软件写入“1”，开始产生新的64bits随机数；运行完毕后，硬件清零； 0：随机数产生完成； 1：写1启动随机数产生，读1表示随机数正在产生；
0	TRNGcir_EN	随机源电路使能位： 0：关闭随机源； 1：打开随机源；

21.4.2 模式寄存器（TRNG_MODE）

偏移地址：0x04

复位值：0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											TRNG_CNT			TRNG_FDBK	TRNG_LOAD
											RW			RW	RW

位	标记	描述
31:5	Reserved	
4:2	TRNG_CNT	64bits TRNG的反馈移位次数 3’b000:移位0次（即输出随机源的采样值） 3’b001:移位8次 3’b010:移位16次 3’b011:移位32次 3’b100:移位64次 3’b101:移位128次 3’b110:移位256次 3’b111:Reserved
1	TRNG_FDBK	在移位操作时，64bits TRNG的反馈信号是否与随机源进行异或操作 0：不进行异或操作； 1：进行异或操作；
0	TRNG_LOAD	在产生新的随机数时，64bits TRNG是否从随机源获得新的初始值 0：不装载新的初始值（产生伪随机数）； 1：装载新的初始值（产生真随机数）；

21.4.3 数据寄存器 0（TRNG_DATA0）

偏移地址：0x0C

复位值：---

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA0[31:16]															
RO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA0[15:0]															
RO															

位	标记	功能描述
31:0	DATA0	软件对本寄存器读取将得到低32位的随机数

21.4.4 数据寄存器 1（TRNG_DATA1）

偏移地址：0x10

复位值：---

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA1[31:16]															
RO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[15:0]															
RO															

位	标记	功能描述
31:0	DATA1	软件对本寄存器读取将得到高32位的随机数

21.5软件基本操作

21.5.1 生成 64bits 真随机数的操作流程（上电第一次）

上电第一次生成 64bits 真随机数，需要经过以下操作：

Step1：打开随机源电路：对真随机数控制寄存器的 Bit0（TRNG_CR.TRNGcir_En）写入“1”，启动随机源电路，开始输出串行真随机数。

Step2：选择重新装载初始值：将真随机数模式寄存器的 Bit0（TRNG_MODE. TRNG_LOAD）设置为“1”，使新产生的真随机数的初始值从随机源获得。

Step3：选择 PRNG64 的直接反馈的方式：将真随机数模式寄存器的 Bit1（TRNG_MODE. TRNG_FDBK）设置为“1”，将反馈信号与随机源异或后输入 PRNG 中。

Step4：选择 PRNG64 的移位次数：设置真随机数模式寄存器的 Bit4－Bit2（TRNG_MODE. TRNG_CNT）为“110”，选择移位 256 次。

Step5：生成真随机数：软件将“1”写入真随机数控制寄存器的 Bit1（TRNG_CR. TRNG_RUN），硬件根据真随机数生成配置进行操作，在操作完成后，硬件自动将 Bit1 清为“0”。

Step6：选择不重新装载初始值：将真随机数模式寄存器的 Bit0（TRNGModeReg. TRNG_Load）设置为“0”。

Step7：选择 PRNG64 的直接反馈的方式：将真随机数模式寄存器的 Bit1（TRNG_MODE. TRNG_FDBK）设置为“0”，将反馈信号直接输入 PRNG 中。

Step8：选择 PRNG64 的移位次数：设置真随机数模式寄存器的 Bit4－Bit2（TRNG_MODE.TRNG_CNT）为“100”，选择移位 64 次。

Step9：生成真随机数：软件将“1”写入真随机数控制寄存器的 Bit1（TRNG_CR.TRNG_RUN），硬件根据真随机数生成配置进行操作，在操作完成后，硬件自动将 Bit1 清为“0”。

Step10：读取真随机数：软件在查询到真随机数控制寄存器的 Bit1（TRNG_CR.TRNG_RUN）变为“0”后，通过读取真随机数数据寄存器 0(TRNG_DATA0)和真随机数数据寄存器 1(TRNG_DATA1)，得到 64Bits 真随机数。

Step11：完成真随机数的生成后，推荐选择关闭随机源电路，节省功耗：对真随机数控制寄存器的 Bit0（TRNG_CR.TRNGcir_En）写入“0”，关闭随机源电路。

21.5.2 生成 64bits 真随机数的操作流程（非上电第一次生成）

非上电第一次生成 64bits 真随机数，需要经过以下操作：

Step1：打开随机源电路：对真随机数控制寄存器的 Bit0（TRNG_CR.TRNGcir_En）写入“1”，启动随机源电路，开始输出串行真随机数。

Step2：选择不重新装载初始值：将真随机数模式寄存器的 Bit0（TRNG_MODE.TRNG_LOAD）设置为“0”。

Step3：选择 PRNG64 的直接反馈的方式：将真随机数模式寄存器的 Bit1（TRNG_MODE.TRNG_FDBK）设置为“1”，将反馈信号与随机源异或后输入 PRNG 中。

Step4：选择 PRNG64 的移位次数：设置真随机数模式寄存器的 Bit4－Bit2（TRNG_MODE.TRNG_CNT）为”110”，选择移位 256 次。

Step5：生成真随机数：软件将“1”写入真随机数控制寄存器的 Bit1（TRNG_CR.TRNG_RUN），硬件根据真随机数生成配置进行操作，在操作完成后，硬件自动将 Bit1 清为“0”。

Step6:选择 PRNG64 的直接反馈的方式：将真随机数模式寄存器的 Bit1（TRNG_MODE.TRNG_FDBK）设置为“0”，将反馈信号直接输入 PRNG 中。

Step7：选择 PRNG64 的移位次数：设置真随机数模式寄存器的 Bit4－Bit2（TRNG_MODE.TRNG_CNT）为”100”，选择移位 64 次。

Step8：读取真随机数：软件在查询到真随机数控制寄存器的 Bit1（TRNG_CR.TRNG_RUN）变为“0”后，通过读取真随机数数据寄存器 0（TRNG_Data0）和真随机数数据寄存器 1（TRNG_Data1），得到 64Bits 真随机数。

如果需要继续生成新的真随机数，那么回到 Step2，直到满足要求。

Step9：完成真随机数的生成后，推荐选择关闭随机源电路，节省功耗：对真随机数控制寄存器的 Bit0（TRNG_CR.TRNGcir_En）写入“0”，关闭随机源电路。

22 高级加密标准模块（AES）

22.1功能定义

22.1.1 AES 算法简述

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 22-1 所示：

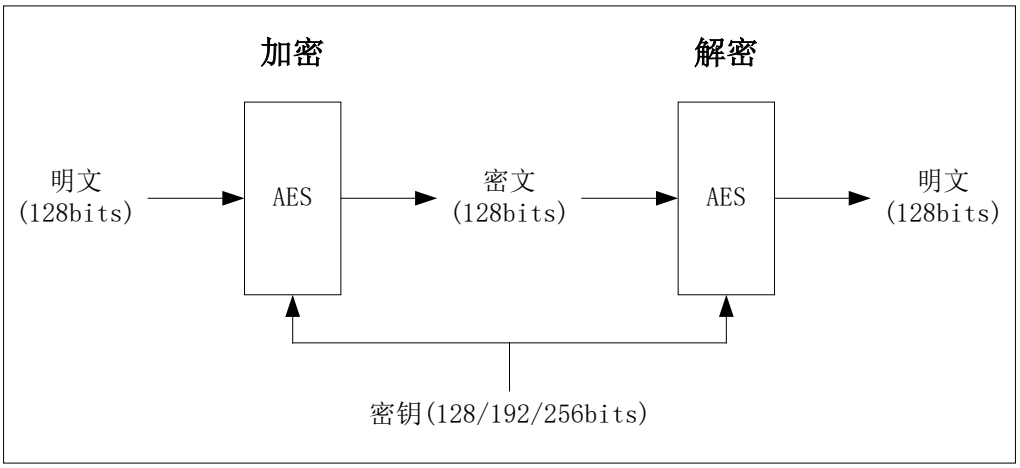


图 22-1 AES 的加解密示意图

AES 算法处理的基本单位是字节，128 位信息被分成 16 个字节，按顺序复制到一个 4×4 的矩阵中，称为状态（state），AES 的所有变换都是基于状态矩阵的变换，该矩阵上保存着计算的中间结果。

AES 是一个密钥迭代分组密码，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 22-2 所示：

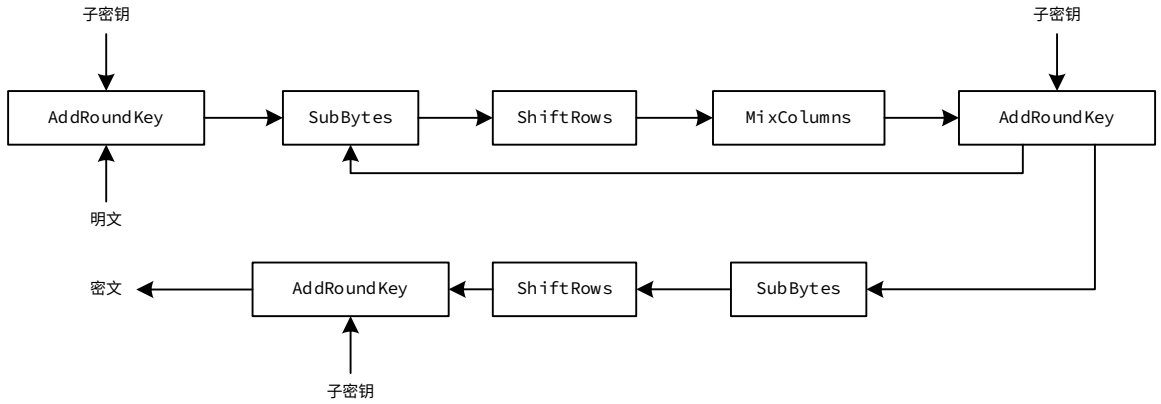


图 22-2 AES 的加密流程图

图中所用子密钥需要由初始密钥扩展而来，且密钥的扩展过程和加密过程是同步进行的。

由于明文固定为 128 位，加密过程运行的轮数就取决于密钥的长度。比如，密钥为 128 位时，运行轮数为 10 轮；密钥为 192 位时，运行轮数为 12 轮；密钥为 256 位时，运行轮数为 14 轮。除了最后一轮缺少 MixColumns 变换，其余各轮均进行完整的轮变换操作。

解密流程与加密流程有所区别，首先必须完成所有密钥的扩展，解密过程从扩展的最后一轮子密钥往回使用；然后是轮变换的四个操作变成了相应的逆运算：InvSubBytes、InvShiftRows、InvMixColumns、AddRoundKey。InvSubBytes 中的模逆运算仍然保持，但仿射变换改为逆变换；InvShiftRows 和 InvMixColumns 变成相应的逆变换；AddRoundKey 保持不变。

直接解密流程的轮变换对四个操作的调用顺序为：InvShiftRows、InvSubBytes、AddRoundKey、InvMixColumns，与加密流程的调用顺序不一致，但使用的密钥与加密流程一致；等价解密流程的轮变换对四个操作的调用顺序为：InvSubBytes、InvShiftRows、InvMixColumns、AddRoundKey，与加密流程的调用顺序完全一致，只是每一轮的子密钥需要进行 InvMixColumns 运算。

关于详细的算法表述，可以参见标准《FIPS PUB 197》。

22.1.2 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。

22.2模块寄存器说明

AES 基地址 0x40021400

表 22-1 寄存器列表

寄存器	偏移地址	描述
AES_CR	0x00 或 0x40	控制寄存器
AES_Data	0x10~0x1C	数据寄存器
AES_Key	0x20~0x3C	密钥寄存器

22.2.1 控制寄存器（AES_CR）

偏移地址：0x00 或 0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											Keysize		Reserved	Mode	Start
											RW			RW	RW

位	符号	描述
31:5	Reserved	保留位，读为0
4:3	Keysize	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
2	Reserved	保留位，读为0
1	Mode	0：加密运算 1：解密运算
0	Start	0：本模块运算结束或未被启动 1：启动本模块进行运算

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。

22.2.2 数据寄存器（AES_Data）

偏移地址：0x10~0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Data[31:16]															
RW															

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

位	符号	描述
31:0	Data	存放AES算法的128比特明文/密文

说明：

1. 数据寄存器由四个 32 位的寄存器组成 128 位数据，用于在模块运算前存放需要被加密的明文或者需要被解密的密文，并且运算完成后存放加密后的密文或者解密后的明文。

加密运算		解密运算	
运算前	运算后	运算前	运算后
128位明文	128位密文	128位密文	128位明文

四个 32 位寄存器连接在一起组成一个 128 位的数据，读写操作时需要分别对四个寄存器进行操作。数据寄存器对应的操作顺序如下：

数据举例：0xFFEEDDCCBBAA99887766554433221100

偏移地址	寄存器名称	填入数据
0x10	AES_Data0	0x33221100
0x14	AES_Data1	0x77665544
0x18	AES_Data2	0xBBAA9988
0x1C	AES_Data3	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。

22.2.3 密钥寄存器（AES_Key）

偏移地址：0x20~0x3C

复位值：0x0000 0000

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

位	符号	描述
31:0	Key	存放AES算法的128、192、256比特密钥

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_Key0	0x03020100
0x24	AES_Key1	0x07060504
0x28	AES_Key2	0x0B0A0908
0x2C	AES_Key3	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_Key0	0x03020100
0x24	AES_Key1	0x07060504
0x28	AES_Key2	0x0B0A0908
0x2C	AES_Key3	0x0F0E0D0C
0x30	AES_Key4	0x13121110
0x34	AES_Key5	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_Key0	0x03020100
0x24	AES_Key1	0x07060504
0x28	AES_Key2	0x0B0A0908
0x2C	AES_Key3	0x0F0E0D0C
0x30	AES_Key4	0x13121110
0x34	AES_Key5	0x17161514
0x38	AES_Key6	0x1B1A1918
0x3C	AES_Key7	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。

22.3异常机制

- 只支持 32 位访问，其它位宽的访问会导致系统异常，进入硬件异常中断。
- 访问 AES 模块的偏移地址大于等于 0x80 的地址，会导致系统异常，进入硬件异常中断。

22.4本模块操作说明

本模块共有两个功能：加密、解密。对两个功能的操作有一些共同的特点，下面先介绍其共同点，再分别介绍每个功能的标准操作流程。

22.4.1 IP 操作的共同点

1. 在 AES 加解密过程中，数据寄存器会改变，如果下次运算的被操作数据就是本次运算的结果，那么就无需重新写入数据了。
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。

22.4.2 加密操作流程

Step 1：将待加密的 128 位数据写入数据寄存器（AES_DATA）中。

Step 2：将加密密钥写入密钥寄存器（AES_KEY）中。

Step 3：根据密钥长度设置 AES_CR.Keysize

Step 4：将 AES_CR.Mode 设置为 0，启动加密模式。

Step 5：向控制寄存器中的 AES_CR.Start 写入 1，启动模块进行运算。

Step 3、Step 4 和 Step 5 可同时进行。

Step 6：等待 AES_CR.Start 的值恢复为 0，模块运算结束。

Step 7：读取数据寄存器（AES_DATA），获得 128 位密文。

22.4.3 解密操作流程

Step 1：将待解密的 128 位数据写入数据寄存器（AES_DATA）中。

Step 2：将解密密钥写入密钥寄存器（AES_KEY）中。

Step 3：根据密钥长度设置 AES_CR.Keysize

Step 4：将 AES_CR.Mode 设置为 1，启动解密模式。

Step 5：向控制寄存器中的 AES_CR.Start 写入 1，启动模块进行运算。

Step 3、Step 4 和 Step 5 可同时进行。

Step 6：等待 AES_CR.Start 的值恢复为 0，模块运算结束。

Step 7：读取数据寄存器（AES_DATA），获得 128 位明文。

22.4.4 数据示例

128 位明文：0xFFEEDDCCBBAA99887766554433221100

128 位密钥：0x0F0E0D0C0B0A09080706050403020100

128 位密文：0x5AC5B47080B7CDD830047B6AD8E0C469

表 22-2 128 位操作寄存器示例

加密前			
寄存器	值（密钥）	寄存器	值（明文）
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC
加密后			
寄存器	值（密钥）	寄存器	值（密文）
Key0	0x03020100	Data0	0xD8E0C469
Key1	0x07060504	Data1	0x30047B6A
Key2	0x0B0A0908	Data2	0x80B7CDD8
Key3	0x0F0E0D0C	Data3	0x5AC5B470

128 位明文：0xFFEEDDCCBBAA99887766554433221100

192 位密钥：0x17161514131211100F0E0D0C0B0A09080706050403020100

128 位密文：0x5AC5B47080B7CDD830047B6AD8E0C469

表 22-3 192 位操作寄存器示例

加密前

寄存器	值（密钥）	寄存器	值（明文）
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC
Key4	0x13121110		
Key5	0x17161514		

加密后

寄存器	值（密钥）	寄存器	值（密文）
Key0	0x03020100	Data0	0xA47CA9DD
Key1	0x07060504	Data1	0xE0DF4C86
Key2	0x0B0A0908	Data2	0xA070AF6E
Key3	0x0F0E0D0C	Data3	0x91710DEC
Key4	0x13121110		
Key5	0x17161514		

128 位明文：0xFFEEDDCCBBAA99887766554433221100

256 位密钥：

0x1F1E1D1C1B1A19181716151413121110F0E0D0C0B0A09080706050403020100

128 位密文：0x5AC5B47080B7CDD830047B6AD8E0C469

表 22-4 192 位操作寄存器示例

加密前

寄存器	值（密钥）	寄存器	值（明文）
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC
Key4	0x13121110		
Key5	0x17161514		
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	0x9049FCAE
Key3	0x0F0E0D0C	Data3	0x8960494B
Key4	0x13121110		
Key5	0x17161514		
Key6	0x1B1A1918		
Key7	0x1F1E1D1C		

22.5运行时间说明

本模块从启动一次运算（AES_CR.Start 写入 1）到该次运算结束（AES_CR.Start 恢复到 0）所需时间如表 22-5 所示：

表 22-5 AES 加解密运行时间

	128位密钥	192位密钥	256位密钥
加密	220 cycles	260 cycles	300 cycles
解密	290 cycles	332 cycles	398 cycles

23 Crystal-less USB 时钟校准器 （CTS）

时钟校准器可以调整校准 RCH48M 时钟频率，也可以调整校准其他 RC 振荡的时钟频率，还可以作为一个通用定时器来使用。本模块主要功能为 RCH48M 自动校准，其他功能为不使用 48M 自动校准时的附加功能。

调整校准 RCH48M 通过可选基准参考信号进行测量待校准振荡器的频率，可以根据测得频率误差自动调整振荡器的频率，也可以使用手动调整。

RCH48M 为 USB 外设提供精准时钟，在这种情况下，基准参考信号可由 USB 总线上的帧起始(SOF)数据包信号提供，USB 主机以 1ms 的时间间隔发送该信号。USB 模块会产生一个 2ms 为周期的翻转信号，CTS 系统时钟基准参考信号也可以由 XTL 振荡器输出或者外部管脚提供，或者由用户软件产生。

RCH24M 振荡校准不支持自动校准，只支持手动调整。参考时钟源可以灵活选择。

定时器是重载模式计数定时器，可以产生捕获中断。支持低功耗唤醒。

23.148M 时钟校准

23.1.1 系统特性

具有可编程预分频和极性的可选基准参考源

- USB SOF 数据包接收
- XTL 振荡器输出
- 外部引脚

可由软件生成基准参考脉冲

振荡器自动微调功能，无需 CPU 参与

手动软件控制选项，可加快校准启动

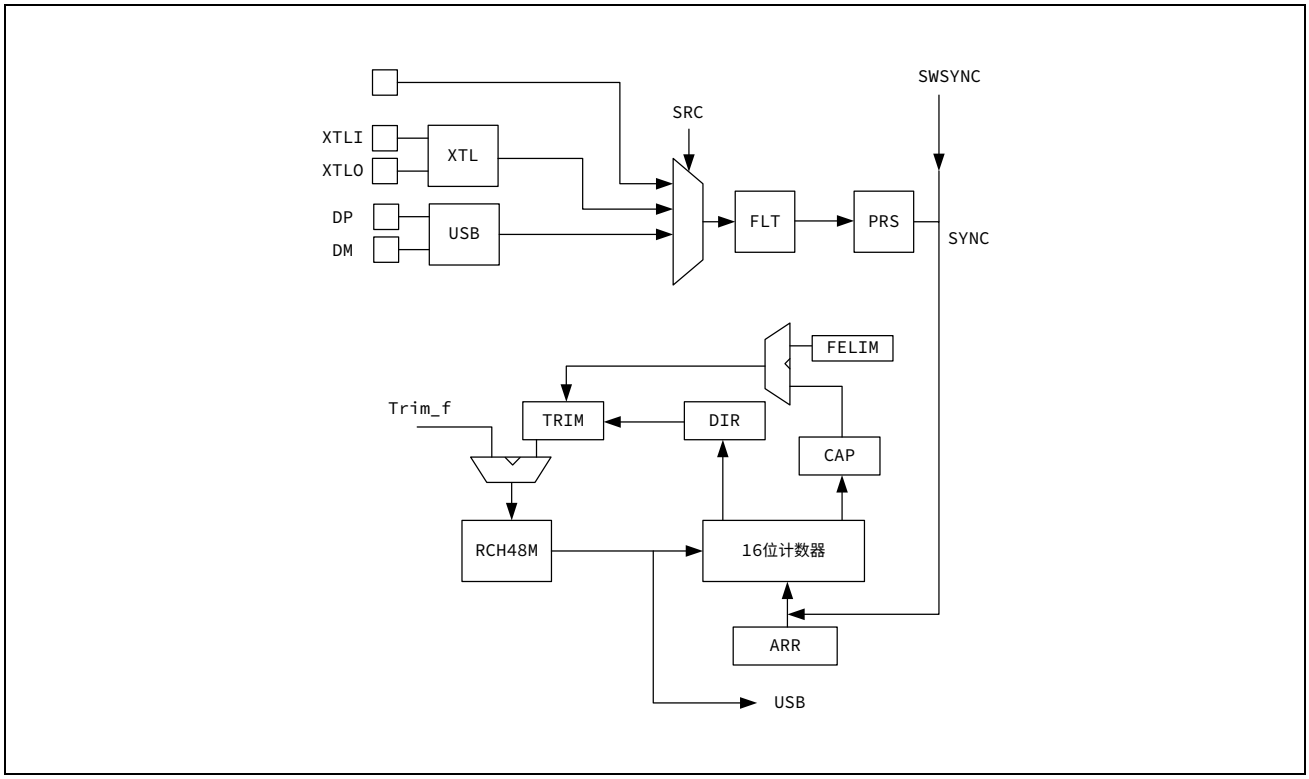
16 位频率误差计数器，用于自动误差捕获和重载

可编程误差限值，用于自动频率误差值和状态标志

可屏蔽中断

- 计数器计数下溢出（UDF）
- 时钟校准正常（OK）
- 时钟校准报警（WARN）
- 时钟校准错误（ERR）

23.1.2 框图



23.1.3 基准参考源同步输入

时钟校准系统基准参考源可通过寄存器 CTS_CFGR 选择，它可以是来自外部的引脚信号、XTL 时钟、USB_SOF 信号。为使基准参考信号输入稳定，可配置是否使用一个简单的数字滤波器来滤除任何干扰。该信号源还可以配置极性，可由预分频器分频，以获取合理频率范围内的同步信号（通常为 1mS 左右）。

还可以通过软件将 CTS_CR 寄存器中的 SW_SYNC 位置 1 的方法生成同步信号。使用 SW_SYNC 时，将参考时钟选择为引脚信号，并且要保证引脚输入电平为低电平。

23.1.4 频率误差测量

频率误差计数器是一个 16 位递减/递增计数器，会在每个基准参考源的 SYNC 事件发生时重载 ARR 值。它开始递减计数，直至达到零值，此时会生成 UDF（预期下溢出）事件。随后它将递增计数到 OUTRANGE 限值，这种情况下最终将停止计数（如果未接收到 SYNC 事件）并生成 MISS 事件。OUTRANGE 限值定义为频率误差限值（CTS_CFGR 寄存器的 FELIM 字段）乘以 128。

当检测到 SYNC 事件时，频率误差计数器的实际值及其方向存储在 CTS_ISR 寄存器的 FECAP（频率误差捕捉）字段和 FEDIR（频率误差方向）位中。当在递减计数阶段（达到零值之前）期间检测到 SYNC 事件时，意味着实际频率小于目标频率（因此，TRIM 值应递增）；当在递增计数期间检测到 SYNC 事件时，意味着实际频率大于目标频率（因此，TRIM 值应递减）。

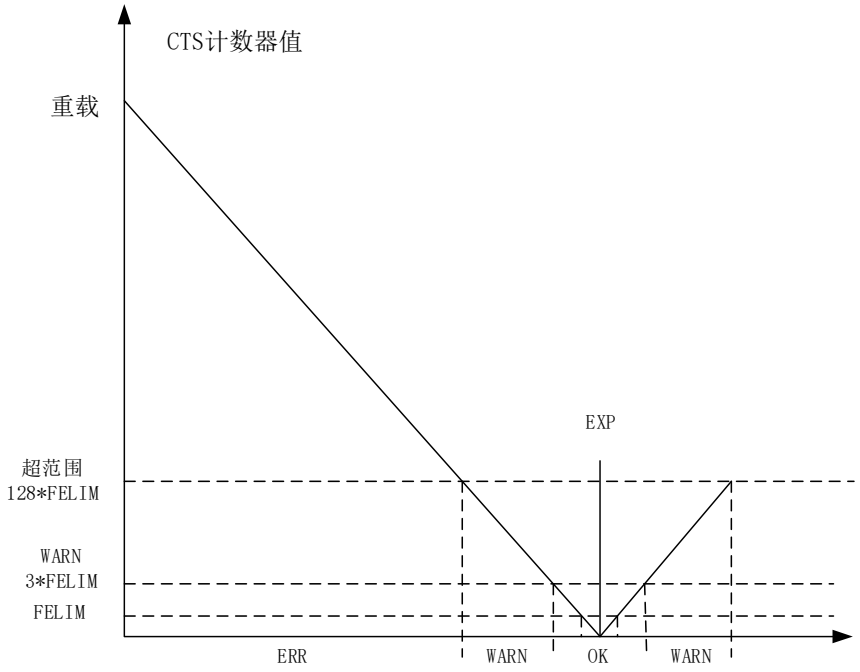


图 23-1 CTS 计数器

23.1.5 频率误差微调

测得的频率误差通过将频率值与一组限值进行比较进行估算：

- TOLERANCE LIMIT（容差限值），在 CTS_CFGR 寄存器的 FELIM 字段中直接给出
- WARNING LIMIT（警告限值），定义为 3 * FELIM 值
- OUTRANGE（误差限值），定义为 128 * FELIM 值

该比较结果用于生成状态指示以及控制自动微调，自动微调通过将 CTS_CR 寄存器中的 AUTO_TRIMEN 位置 1 来使能。

- 当频率误差低于容差限值时，意味着 TRIM 字段中的实际微调值是最优值，因此无需任何微调操作。
  - 指示 OK 状态
  - AUTOTRIM 模式下的 TRIM 值无变化
- 当频率误差低于警告限值但高于或等于容差限值时，意味着需要某种微调操作，但只需一步微调便可达到最优 TRIM 值。
  - 指示 OK 状态
  - 在 AUTOTRIM 模式下通过±1 微调来调整 TRIM 值
- 当频率误差高于或等于警告限值但低于误差限值时，意味着需要更强的微调操作，并且存在下一周期无法达到最优 TRIM 值的风险。
  - 指示 WARN 状态

- 在 AUTOTRIM 模式下通过±2 微调来调整 TRIM 值

■ 当频率误差高于或等于误差限值时，意味着频率超出微调范围。当 SYNC 输入不干净或某个 SYNC 脉冲丢失时（例如，当一个 USB SOF 被破坏时）。也会发生这种情况。

- 指示 ERR 或 MISS 状态
- AUTOTRIM 模式下的 TRIM 值无变化

注意：

- 如果 TRIM 字段的实际值十分接近其限值，如果需要向其限值方向调整，只会调整到其限值，之后 TRIM 值保持为限值，此时将指示 OVF 状态。
- 在 AUTOTRIM 模式（CTS_CR 寄存器中的 AUTO_TRIMEN 位置 1）下，CTS_CR 的 TRIM 字段由硬件调整并且是只读的。

23.1.6 CTS 初始化和配置

ARR 值

ARR 值应根据目标频率与预分频后参考同步源频率之比进行选择。该值随后会减 1，以在零值时达到预期同步。具体公式如下：

ARR = (fTARGET / fSYNC) -1

ARR 字段的复位值对应于 48 MHz 的目标频率和 1 kHz 的同步信号频率（来自 USB 的 SOF 信号）。

FELIM 值

FELIM 值的选择与 RCH48M 振荡器的特性及其典型微调步长紧密相关。最优值对应于微调步长的一半，以 RCH48M 振荡器时钟节拍数表示。可使用以下公式：

FELIM = (fTARGET / fSYNC) * STEP[%] / 100% / 2

结果应始终舍入为最接近的整数值以获得最佳微调响应。如果应用中不需要频繁的微调操作，可通过稍微增大 FELIM 值来增加微调滞后。

FELIM 字段的复位值对应于 (fTARGET / fSYNC) = 48000 以及 0.14% 的典型微调步长。

注意：

- 错误配置 ARR 和 FELIM 字段无法实现硬件保护，从而导致不定微调响应。预期工作模式需要正确设置 ARR 值（根据同步源频率），该值还大于 128 * FELIM 值（OUTRANGE 限值）。

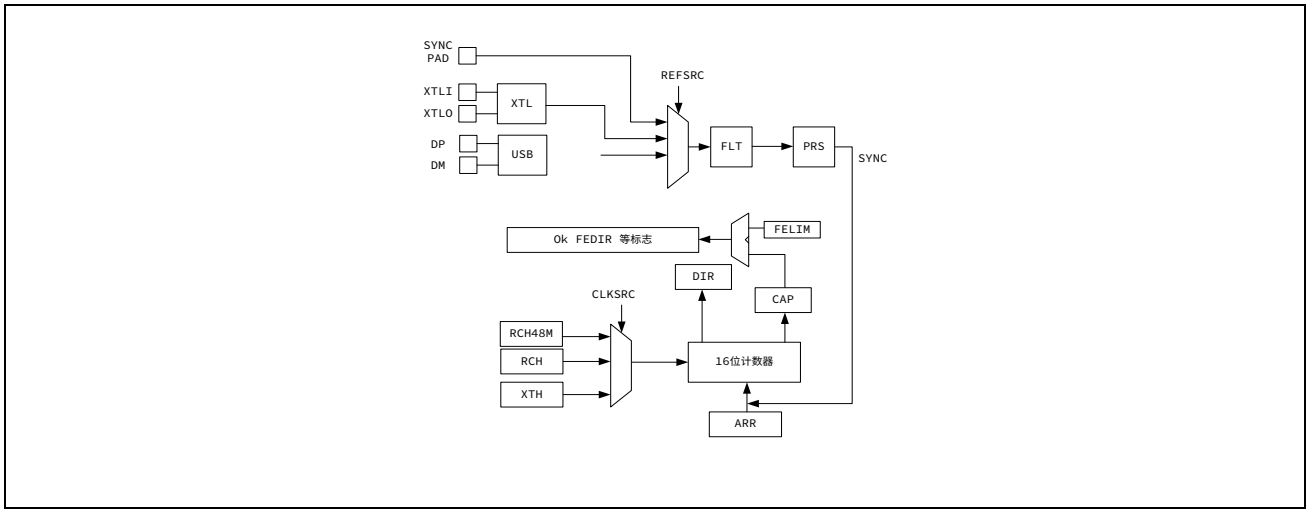
23.2其他振荡校准

23.2.1 系统特性

校准高速时钟 RCH 参考时钟可以选择端口输入。

校准低速时钟 RCL 时，将 RCL 作为参考时钟，计数时钟选择为精准的内部时钟或外部输入精准时钟。

23.2.2 框图



23.2.3 校准说明

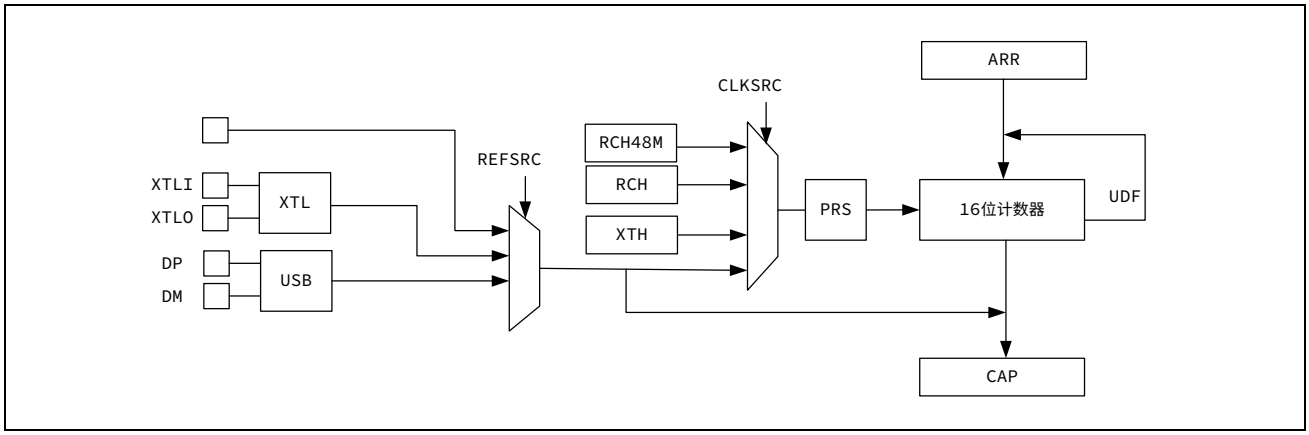
RCH 振荡校准：

1. 设置参考时钟为端口输入或者精准的 XTL，设置合适的参考时钟分频
2. 设置校准时钟为 RCH
3. 设置 CFGR 设置要校准的目标频率与误差范围
4. 在中断服务程序中判断 FEDIR,如果为 1，将频率值提高，增大 TRIM 值；反之减小

TRIM 值初始设置为中值，下次更改的值的上次更改值的 1/2，即 2 分法。9 位的 TRIM 值 8 次即可校准。

23.3定时器功能说明

23.3.1 框图



定时器特性

- 16 位下计数自动重载低功耗定时器
- 多种时钟源可配

- 1/2/4/8/16/32/64/256 预分频可配置
- 输入捕获功能
- 定时器唤醒功能
- 支持溢出中断，捕获中断

23.3.2 定时器时钟选择

定时器可以选择 RCH48M,RCH,XTH 等内部高速时钟，也可以选择外部端口输入，低速 XTL 做为计数时钟。当选择高速时钟时，外部输入可以实现上沿或下沿的边沿捕获功能。

23.3.3 定时器

定时器计数值从 ARR 减计数，当计数到 0 后产生下溢出，如果中断使能会产生下溢出中断，计数周期为 ARR+1。

TIM_EN 只有在 CEN 为 0 时才可以更改，TIM_EN 为 1 并且 CEN 写 1 时定时器会重新装载重载值到定时器。

23.4CTS 寄存器

基地址 0x40005000

表 23-1 CTS 寄存器

寄存器	偏移地址	描述
CTS_CR	0x000	控制寄存器
CTS_CFGR	0x004	配置寄存器
CTS_ISR	0x008	中断和状态寄存器
CTS_ICR	0x00C	中断标志清除寄存器

23.4.1 CTS 控制寄存器（CTS_CR）

偏移地址 0x00

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM_EN	TRIM							SW_SYNC	AUTO_TRIMEN	CEN	CAPIE	UDFIE	ERRIE	WARNIE	OKIE
RW	RW							RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:16	Reserved	保留
15	TIM_EN	定时器功能（只有在CEN为0时，才可以改写）
14:8	TRIM	RCH48M 振荡微调值，当AUTO_TRIMEN为1时，由硬件控制且只读;为有符号数，0x40最小，0x3F最大，0x00是中值；自动校准时值有上下限保护。 手动TRIM时，这个值可以读写。
7	SW_SYNC	软件同步，软件写1，硬件清零。 注：如果trim的目标时钟频率小于系统时钟，不可以使用软件同步
6	AUTO_TRIM_EN	自动微调使能 1：使能；0：禁止 注：使能时会将写入的TRIM值同步到硬件自动TRIM值中，如果需要保持自动TRIM值为中值，必须在手动TRIM模式下将TRIM值清零。
5	CEN	频率误差计数器/定时器使能
4	CAPIE	定时器模式捕获中断使能
3	UDFIE	计数器下溢出中断使能
2	ERRIE	校准错误中断使能
1	WARNIE	校准警告中断使能
0	OKIE	校准正常中断使能

23.4.2 CTS 配置寄存器（CTS_CFGR）

偏移地址 0x04

复位值 0x0022BB7F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLKSRC		REFSRC		POL	DIV			FELIM							
RW		RW		RW	RW			RW							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR															
RW															

位	标记	功能描述
31:30	CLKSRC	校准模式下校准时钟选择 定时器模式下计数时钟选择 保留 默认RCH48M 00： RCH48M 01：RCH 10：XTH 11：RFFCLK（仅限定时器模式下使用）
29:28	REFSRC	校准同步信号源选择，定时器模式时钟选择或捕获源选择 00：USB SOF 01：XTL 10：GPIO 11：Reserved
27	POL	同步极性选择 0：上升沿 1：下降沿 注：使用USB_SOF时无效，USB_SOF为周期2ms的翻转信号，需要将其转换为周期1ms的信号
26:24	PRS	TRIM同步信号分频器 /定时器时钟预分频 000：不分频 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: 256分频
23:16	FELIM	频率误差限值

15:0	ARR	计数器重载值

注：这个寄存器只有在 *CEN* 为 *0* 时才可以更改。

23.4.3 CTS 中断和状态寄存器（CTS_ISR）

偏移地址 0x08

复位值 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FECAP															
RO															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIR	Reserved				OVF	MISS	ERR	Reserved				UDFF	ERRF	WARNF	OKF
RO					RO	RO	RO					RO	RO	RO	RO

位	标记	功能描述
31:16	FECAP	频率误差捕获值
15	DIR	频率误差方向，显示实际频率低于目标频率还是高于目标频率 0：递增计数方向，实际频率高于目标频率 1：递减计数方向，实际频率低于目标频率
14:11	Reserved	保留
10	OVF	TRIM值上溢或下溢时，该标志由硬件置1
9	MISS	校准参考同步丢失(频率过快并超范围)，计数到0后计数值递增计数到FELIM*128没有检测到同步信号。
8	ERR	校准错误(频率过慢并超范围)，参考同步信号在计数器溢出之前且误差大于FELIM*128,
7:5	Reserved	保留
4	CAPF	捕获标志
3	UDFF	计数器下溢出标志，频率误差计数器计数到0时，该位硬件置1。
2	ERRF	校准错误标志，OVF,MISS,ERR逻辑或的结果
1	WARNF	校准警告标志，频率误差大于等于FELIM*3且小于FELIM*128
0	OKF	校准正常标志。测量频率误差小于FELIM*3时，该标志硬件置1。

23.4.4 CTS 中断状态清除寄存器（CTS_ICR）

偏移地址 0x0C

复位值 0x0000001F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											CAPC	UDFC	ERRC	WARNC	OKC
											R1W0	R1W0	R1W0	R1W0	R1W0

位	标记	功能描述
31:5	Reserved	保留
4	CAPC	定时器模式捕获标志清除，写0清除，写1无效。
3	UDFC	预期同步标志清除。写0清除
2	ERRC	校准错误标志清除，同时清除OVF,MISS,ERR标志。写0清除
1	WARNC	校准警告标志清除。写0清除
0	OKC	校准正常标志清除。写0清除

24 音频接口（I2S）

I2S 特性

- 支持 Philip / MSB / LSB / PCM 模式
- 支持 MCK 输出
- 支持 48K, 44.1K, 32 K, 16K, 8K 等不同的音频采样率
- 支持 数据长度 16 位,24 位,32 位
- 支持帧长度 16 位/32 位
- 支持 DMA 数据传输
- 支持全双工收发（2 个 I2S 配合）
- 支持 master 发送、接收
- 支持 slave 发送、接收
- 与 SPI 共用中断与端口

24.1功能说明

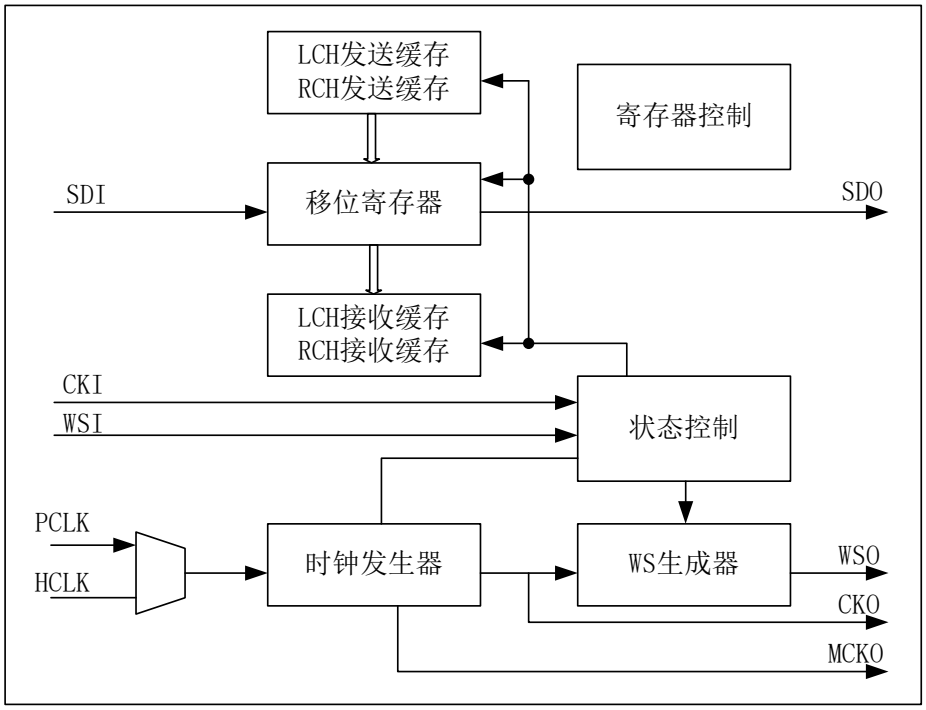


图 24-1 功能框图

24.1.1 管脚复用

I2S 与 SPI IO 复用，复用关系如下：

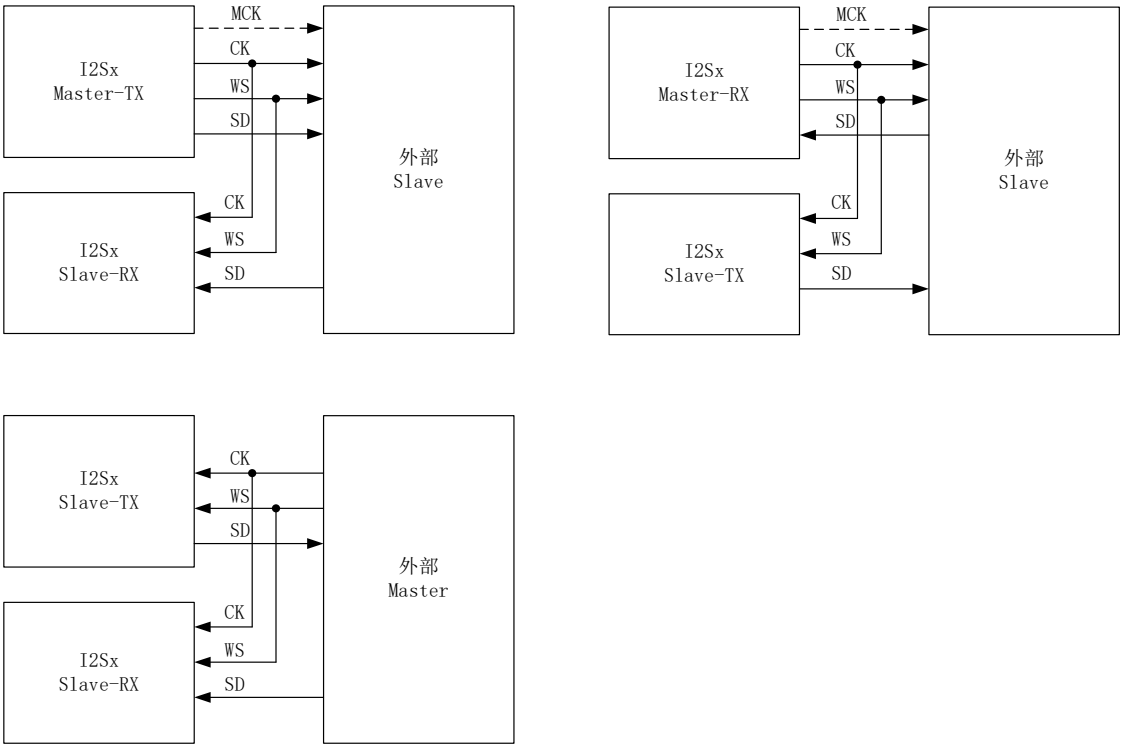
当使能 I2S 功能（将 SYSCTRL_PERI 寄存器中的 I2S 位置 1 并且 SPI 无效）后，SPI 与 I2S 复用端口用作音频 I2S 接口。

I2S	SPI	I2S 管脚说明
SD	MOSI	串行数据（映射到 MOSI 引脚），用于发送或接收两个时分复用的数据通道上的数据（仅半双工模式）
WS	NSS	字选择（映射到 NSS 引脚），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。
CK	SCK	串行时钟（映射到 SCK 引脚），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。
MCK	MISO	主时钟输出（MCK 映射到 MISO 引脚），当 I2S 配置为主模式（并且 I2Sx_PR 寄存器中的 MCKOE 位置 1）时，使用主时钟（单独映射）输出此附加时钟，该时钟以 256 × fs 的预配置频率生成，其中 fs 为音频信号采样频率。

I2S 另外有一组独立的 IO 设置，客户可以同时使用 I2S 与 SPI。在这种模式下中断与 SPI 共用同一个中断入口地址，需要根据各自标志判断处理相应的中断服务程序。

24.1.2 I2S 全双工

每个 I2S 都支持主模式与从模式，支持半双工的发送与接收，如果需要全双工模式，需要使用两个 I2S 进行配合。



24.2I2S 音频协议说明

I2S 需要处理通常在左右两个通道上时分复用的音频数据。但是，有两个 16 位寄存器进行发送或接收。所以，需由软件将与相应通道对应的值写入通道数据寄存器，以及从相应通道数据寄存器中读取相应数据。始终先发送左通道数据，而后再发送右通道数据（对于 PCM 协议来说，右通道数据没有意义）。

数据和帧格式组合有四种，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，只需要读写 16 位有效数据，16 位无效数据被强制清零，无需任何软件操作或 DMA 请求（只需一个读/写操作）。

24 位和 32 位数据帧需要对 I2Sx_DR 寄存器执行两次 CPU 读取或写入操作，或者需要两次 DMA 操作。对于 24 位数据帧，硬件会在自动填充 8 个 0 扩展到 32 位。

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 接口支持四种音频标准，可使用 I2Sx_CFGR 寄存器中的 I2SSTD[1:0] 和 PCMSYNC 位对其进行配置。

24.2.1 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。

发送方在时钟信号 (CK) 的下降沿改变数据，接收方在上升沿读取数据。WS 信号也在 CK 的下降沿变化。

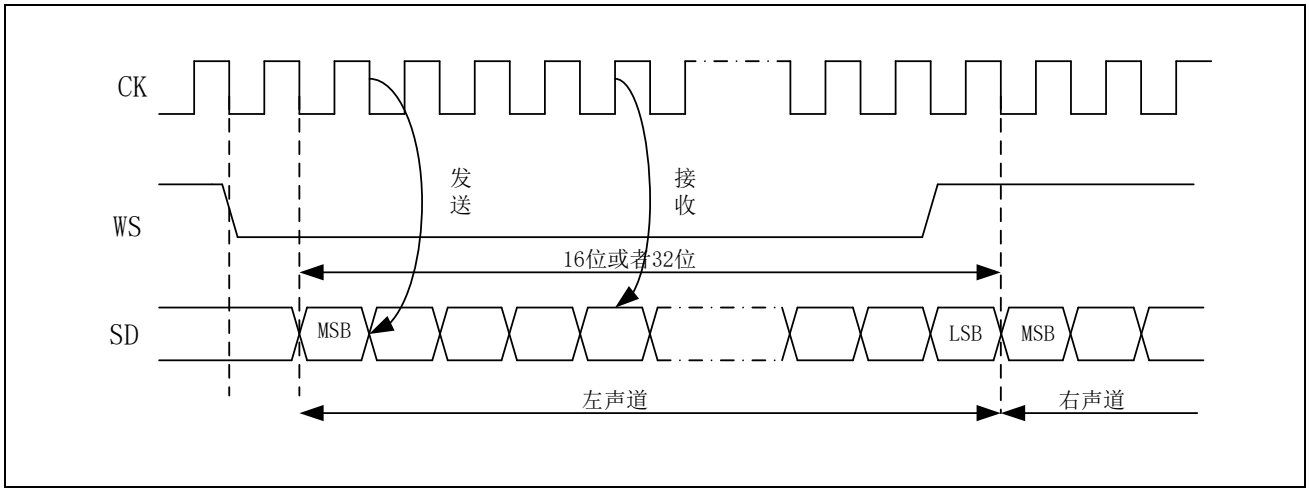


图 24-2 Philip 协议波形（16/32 位全精度）

该模式需要对 I2Sx_DRL/ I2Sx_DRR 寄存器执行两次写入或读取操作。

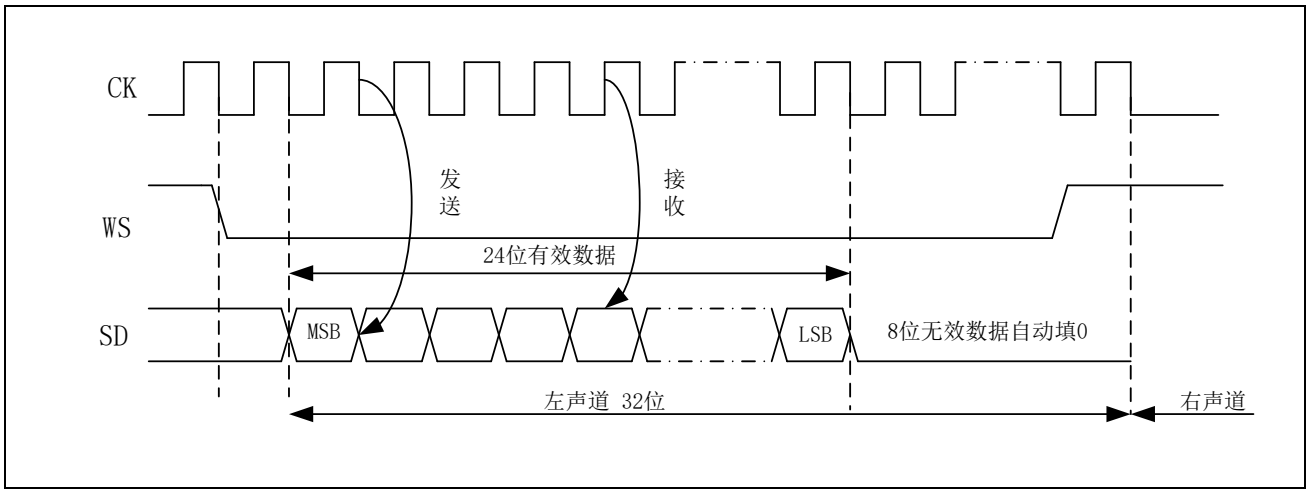


图 24-3 Philip 协议波形（24 位帧）

该模式需要对 I2Sx_DRL/ I2Sx_DRR 寄存器执行两次写入或读取操作。

在发送模式下，如果需要发送 0x89ABCD,0x8FEDCA（24 位）：

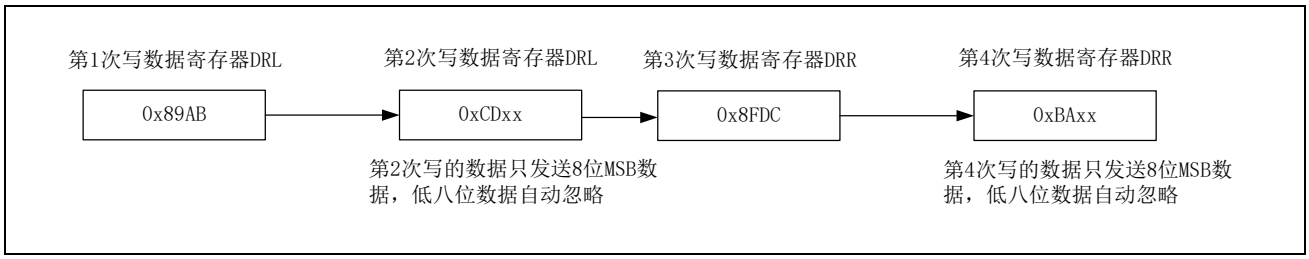


图 24-4　24 位数据发送写操作

在接收模式下，如果接收数据 0x123456,0x89ABCD：

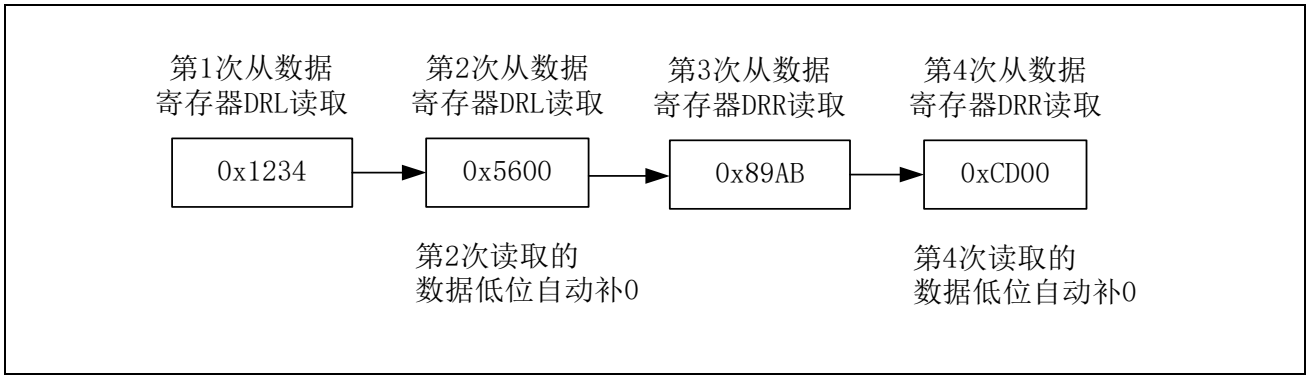


图 24-5　24 位数据模式接收读操作

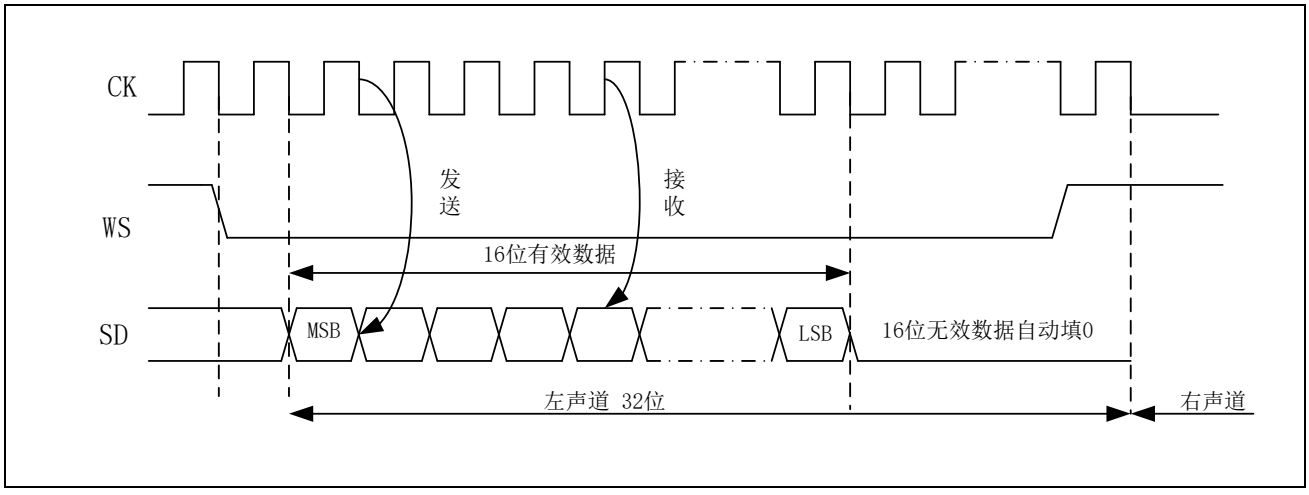


图 24-6　Philip 协议 16 位扩展到 32 位数据帧

如果在 I2S 配置阶段选择将 16 位数据帧扩展到 32 位通道帧，则只需要访问一次 I2Sx_DR 寄存器。另外 16 位将有硬件自动填 0，不需要软件参与。

如果要发送的数据或已接收的数据为 0xCDEF（0xCDEF0000 扩展为 32 位），则需要执行下图中显示的操作。



图 24-7 16 位数据读写操作

发送时，每次将 MSB 写入数据寄存器 I2Sx_DRL/R，数据转移到移位寄存器后，相应通道的 TXE 标志就会置 1，并在中断使能的情况下触发中断，以将要发送的新数据加载到 I2Sx_DR 寄存器。即使硬件填充的低 16 位 0x0000 还未发送，也会如此，因为低 16 位是由硬件发送。

接收时，接收到第一个半字（高 16 位），则硬件将相应通道的 RXNE 标志置 1，并在中断使能的情况下触发中断。

24.2.2 MSB 对齐标准

此标准同时生成 WS 信号和第一个数据位，WS 左声道对应高电平，右声道对应高电平。

发送方在时钟信号（CK）的下降沿改变数据，接收方在上升沿读取数据。WS 信号也在 CK 的下降沿变化。

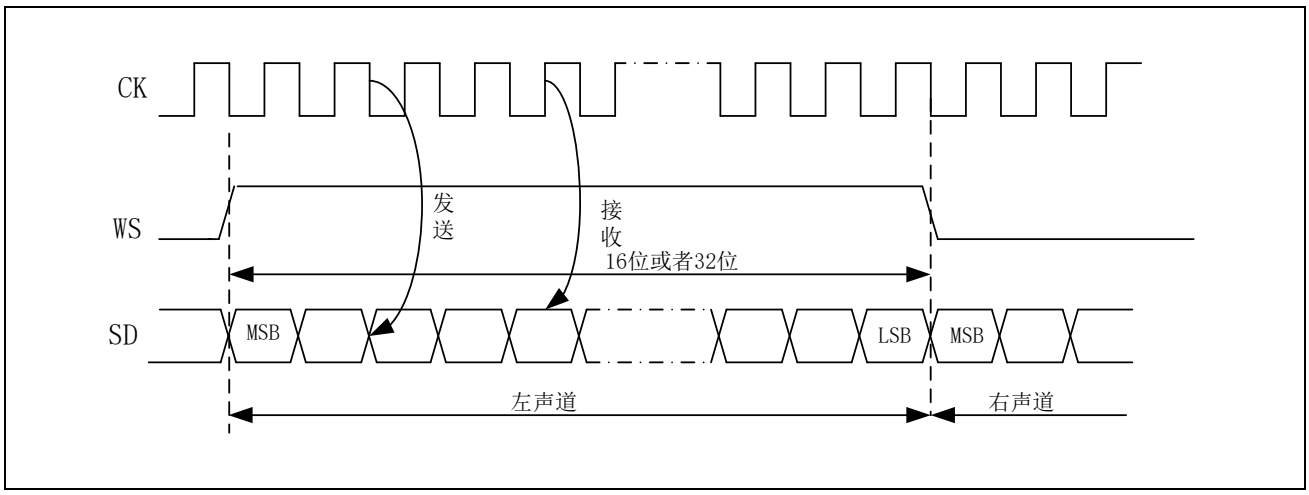


图 24-8 MSB 协议波形（16/32 位全精度）

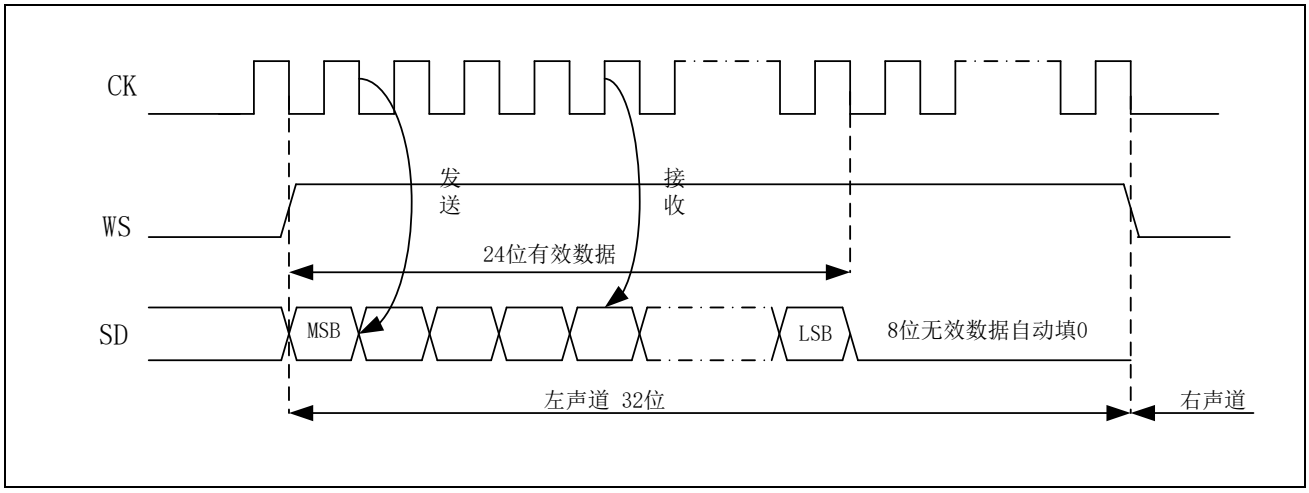


图 24-9 MSB 协议波形（24 位帧）

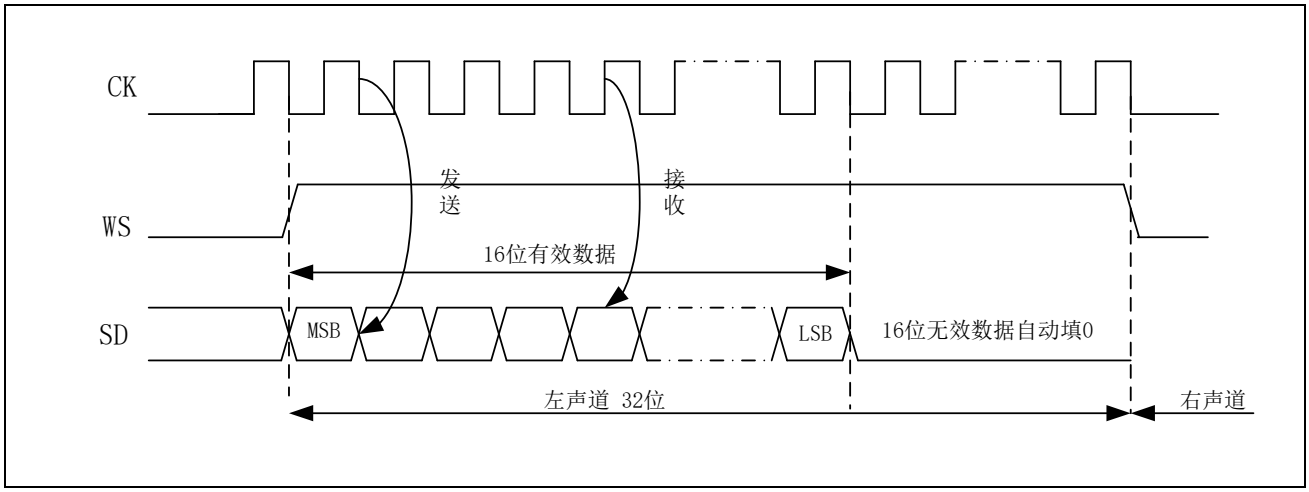


图 24-10 MSB 协议 16 位扩展到 32 位数据帧

24.2.3 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）。

发送方在时钟信号（CK）的下降沿改变数据，接收方在上升沿读取数据。WS 信号也在 CK 的下降沿变化。

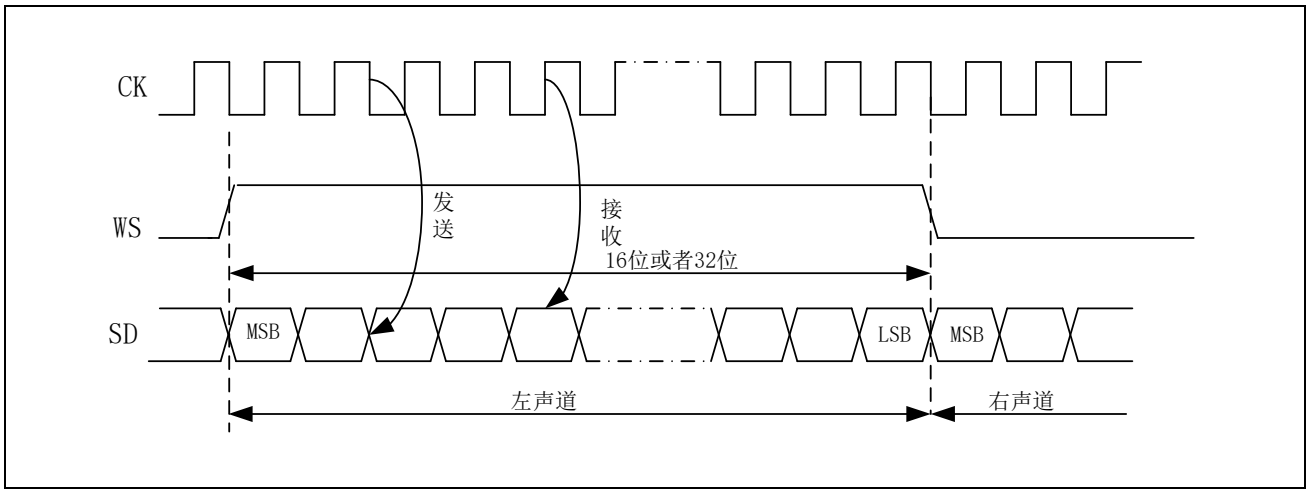


图 24-11 LSB 协议波形（16/32 位全精度）

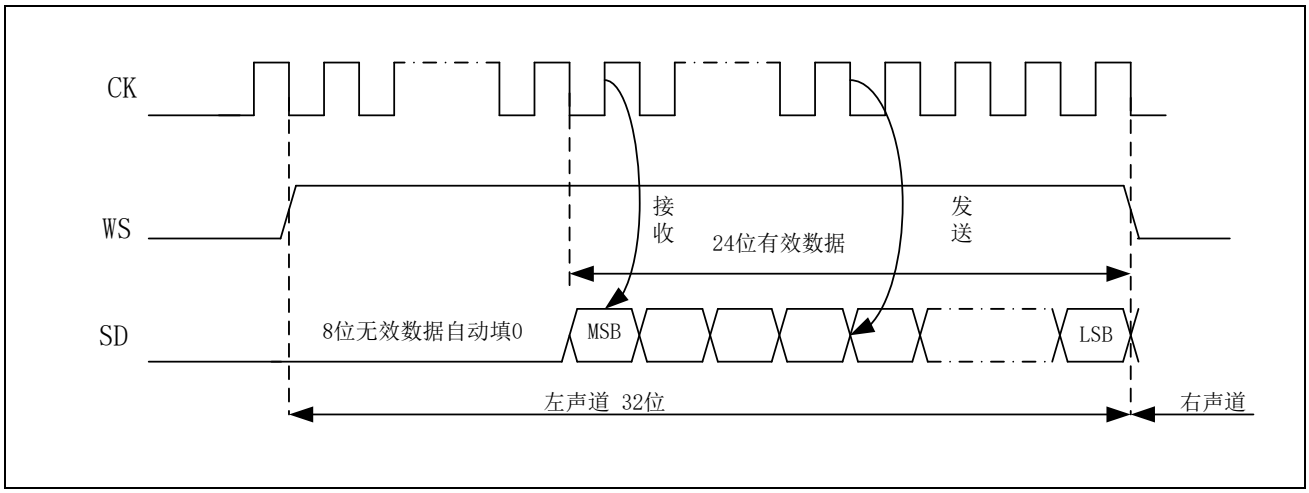


图 24-12 LSB 协议波形（24 位帧）

在发送模式下，如果需要发送数据 0x89ABCD，则需要通过软件或 DMA 对 I2Sx_DRL/ I2Sx_DRR 寄存器执行两次写入操作。下面给出了这些操作。

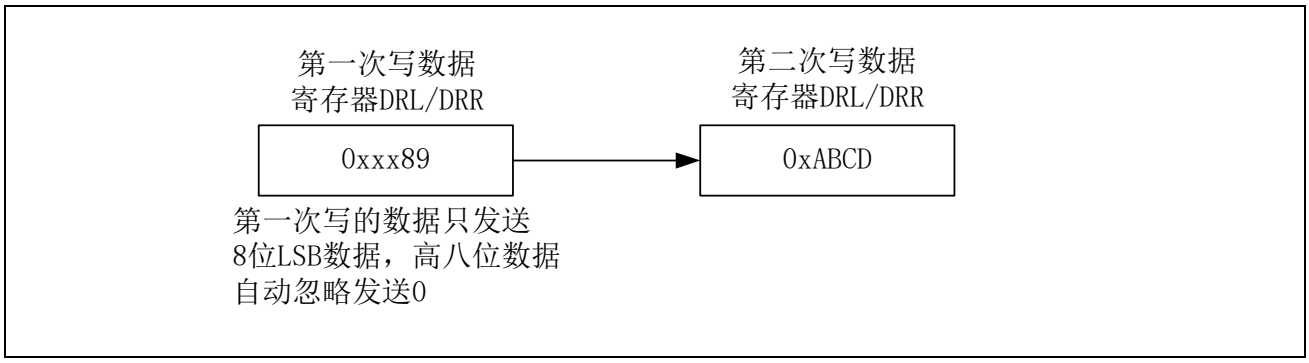


图 24-13　LSB 24 位数据发送写操作

在接收模式下，如果接收到数据 0x123456，则在每个 RXNE 事件时需要对 I2Sx_DR 寄存器执行两次连续的读取操作。

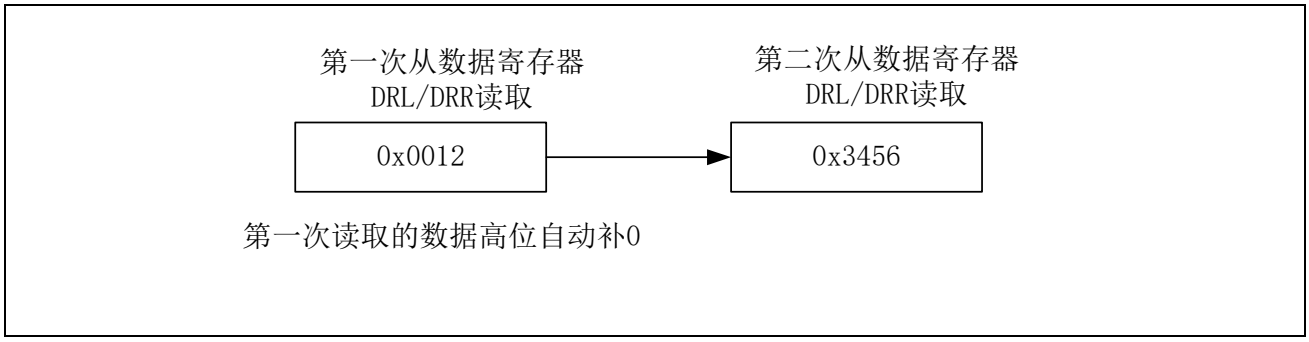


图 24-14　LSB 24 位数据模式接收读操作

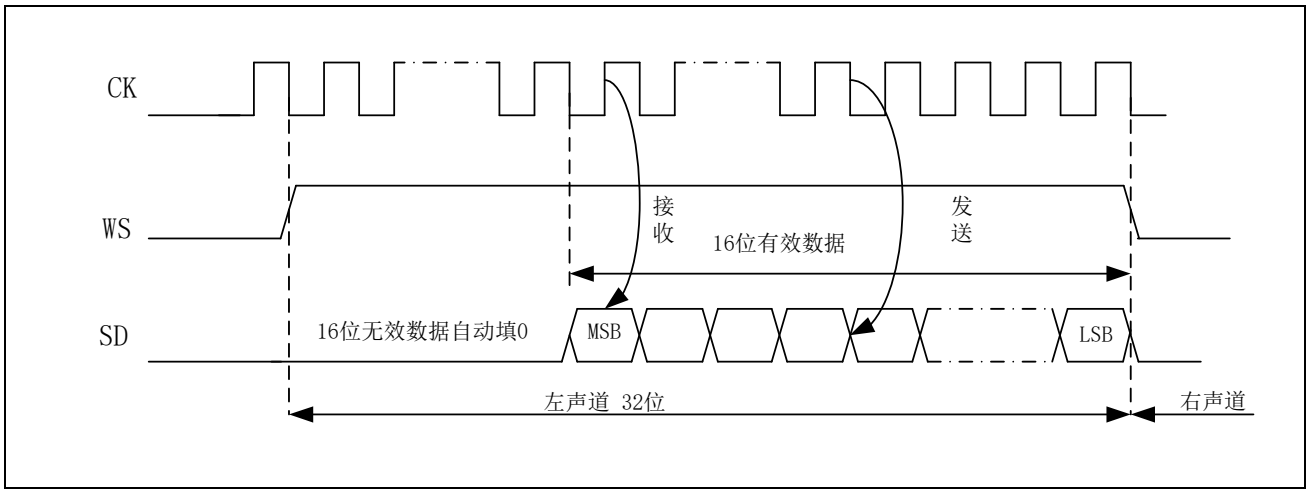


图 24-15　LSB 协议 16 位扩展到 32 位数据帧

如果在 I2S 配置阶段选择将 16 位数据帧扩展到 32 位通道帧，则只需要访问一次 I2Sx_DRL/I2Sx_DRR 寄存器。扩展到 32 位中，高半字（16 位 MSB）被硬件置为 0x0000。

如果要发送的数据或已接收的数据为 0x7893（0x0000 7983 是扩展 32 位的数据），则需要执行下中显示的操作。



图 24-16 16 位数据读写操作

在发送模式下，相应通道发生 TXE 事件时，应用程序需要在相应通道写入要发送的数据（此例中，为 0x7893）。

首先发送 0x0000 字段（扩展到 32 位）。相应通道的有效数据（0x7893）转移到移位寄存器后，相应通道的 TXE 标志会被再次置 1。

在接收模式下，当接收到有效半字后（而非 0x0000 字段），即会置位相应通道的 RXNE。

24.2.4 PCM 标准

对于 PCM 标准，无需使用通道信息。可使用两种 PCM 模式（短帧和长帧），并且可使用 I2Sx_CFGR 寄存器中的 PCMSYNC 位来配置。PCM 发送数据使用上升沿，接收数据使用下降沿。

发送数据与接收数据只需要使用 I2Sx_DRL 数据寄存器，I2Sx_DRR 数据寄存器在这种模式下无效。

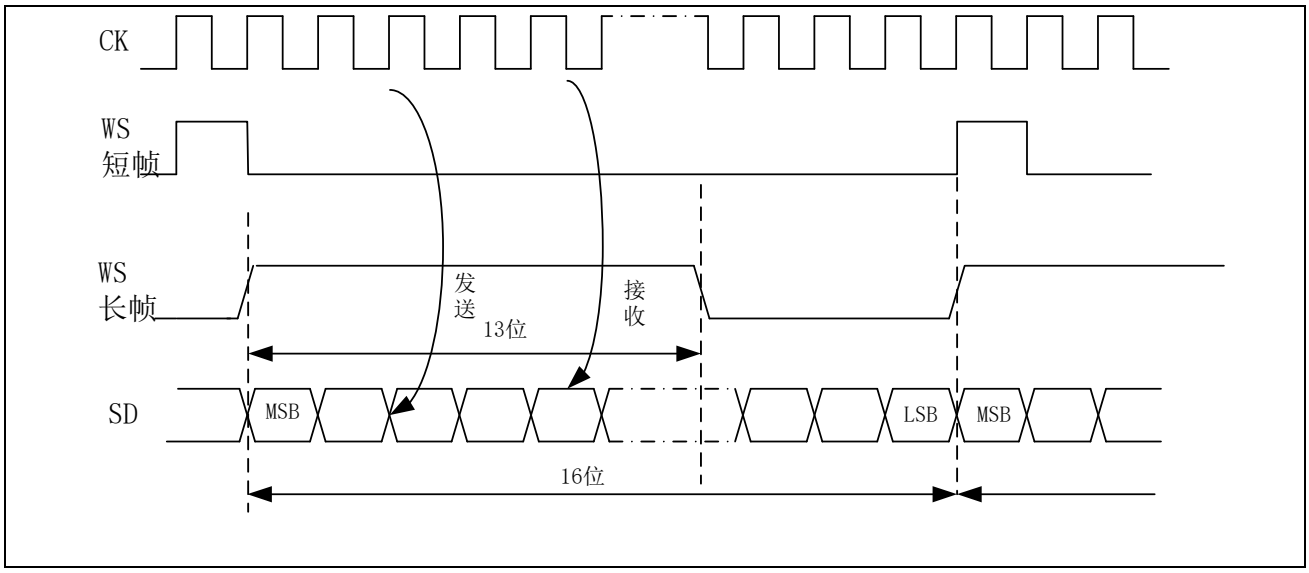


图 24-17 PCM 协议波形（16 位）

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。

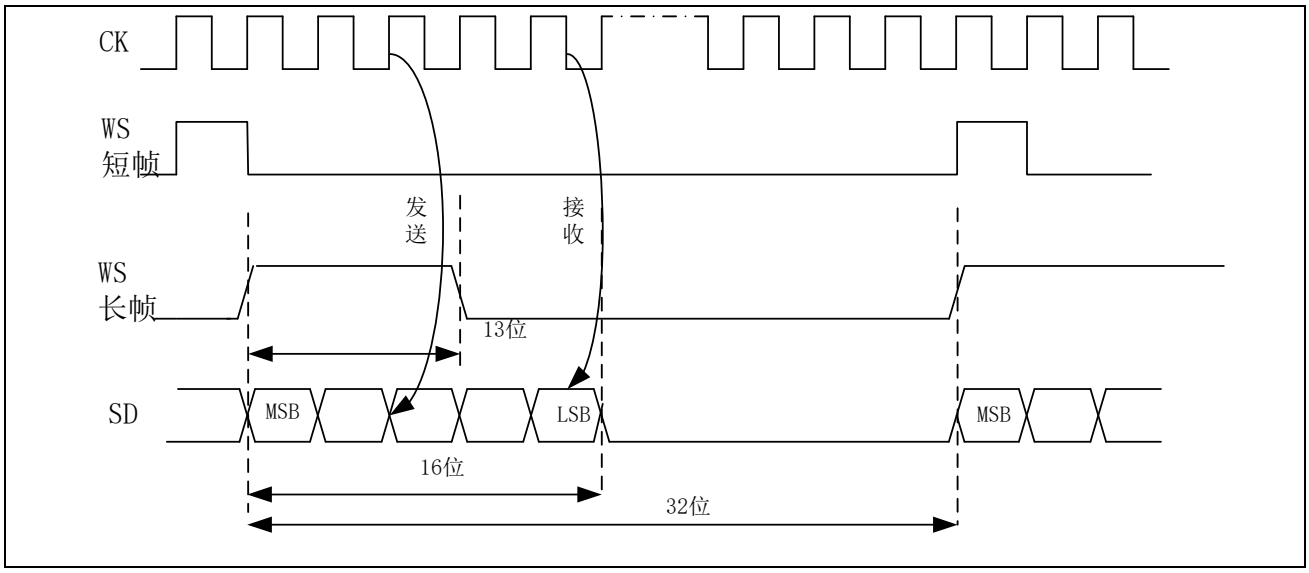


图 24-18　PCM 协议波形（16 位数据扩展到 32 位）

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（I2Sx_CFGR 寄存器中的 DATLEN 位和 CHLEN 位）。

24.3I2S 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 = 32 x 2 x fS。

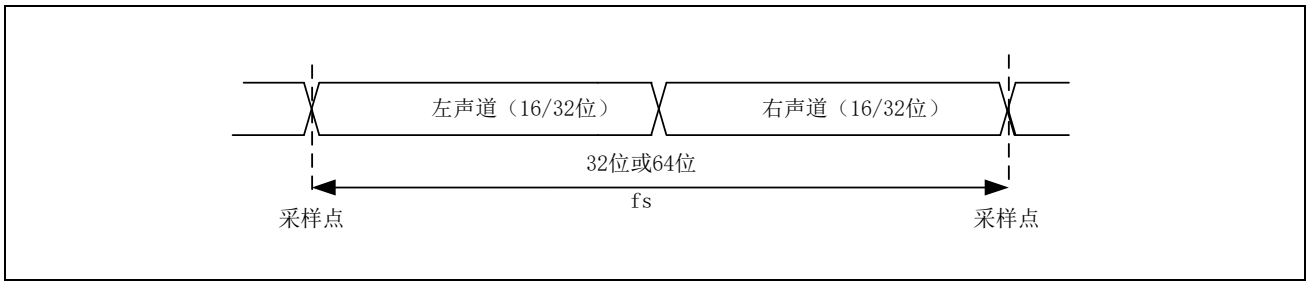


图 24-19 音频采样频率定义

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。

下图展示了通信时钟架构。I2Sx 时钟可选择 PCLK 或者 HCLK。

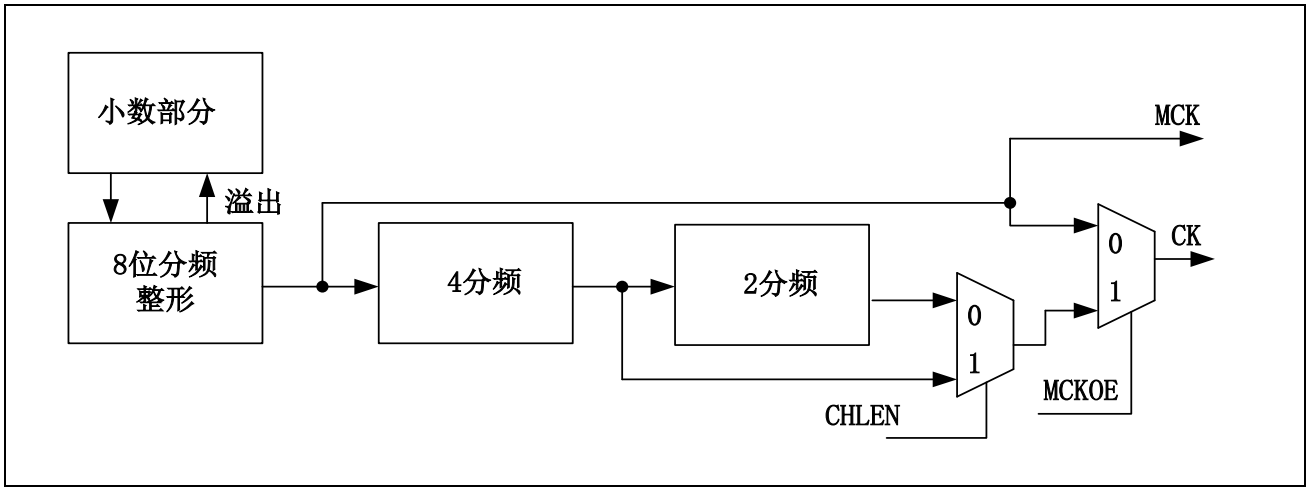


图 24-20 时钟产生

音频采样频率可以是 96 kHz、48 kHz、44.1 kHz、32 kHz、22.05 kHz、16 kHz、11.025 kHz 或 8 kHz（或此范围内的任何其它值）。为达到所需频率，需要根据以下公式对线性分频器进行编程：

小数分频 FRACT 等于 0x00 时：

输出主时钟（I2Sx_PR 寄存器中的 MCKOE 置 1)时：

fS = I2SxCLK / [(16*2)*((2*I2SDIV)+ODD)*8)]（通道帧宽度为 16 位时）

fS = I2SxCLK / [(32*2)*((2*I2SDIV)+ODD)*4)]（通道帧宽度为 32 位时）

禁止主时钟输出（MCKOE 位清零）时：

fS = I2SxCLK / [(16*2)*((2*I2SDIV)+ODD)]（通道帧宽度为 16 位时）

fS = I2SxCLK / [(32*2)*((2*I2SDIV)+ODD)]（通道帧宽度为 32 位时）

小数分频 FRACT 不等于 0x00 时：

输出主时钟（I2Sx_PR 寄存器中的 MCKOE 置 1)时：

fS = I2SxCLK /{ (16*2)* [2*(I2SDIV+FRACT/64)]*8}（通道帧宽度为 16 位时）

fS = I2SxCLK /{ [(32*2)*[2*(I2SDIV+FRACT/64)]*4}（通道帧宽度为 32 位时）

禁止主时钟输出（MCKOE 位清零）时：

fS = I2SxCLK /{ (16*2)* [2*(I2SDIV+FRACT/64)]}（通道帧宽度为 16 位时）

fS = I2SxCLK /{ (32*2)* [2*(I2SDIV+FRACT/64)]}（通道帧宽度为 32 位时）

下表提供了时钟 48M 时钟针对不同采样率的设置及精度值（不使用小数分频）。

I2S CLK	通道长度	I2SDIV	I2SODD	MCLK	采样率	实际频率	误差%	freq(mck)
48000000	16	8	0	无输出	96000	93750	-2.34	0
48000000	32	4	0	无输出	96000	93750	-2.34	0
48000000	16	15	1	无输出	48000	48387	0.81	0
48000000	32	8	0	无输出	48000	46875	-2.34	0
48000000	16	17	0	无输出	44100	44118	0.04	0
48000000	32	8	1	无输出	44100	44118	0.04	0
48000000	16	23	1	无输出	32000	31915	-0.27	0
48000000	32	11	1	无输出	32000	32609	1.9	0
48000000	16	34	0	无输出	22050	22059	0.04	0
48000000	32	17	0	无输出	22050	22059	0.04	0
48000000	16	47	0	无输出	16000	15957	-0.27	0
48000000	32	23	1	无输出	16000	15957	-0.27	0
48000000	16	68	0	无输出	11025	11029	0.04	0
48000000	32	34	0	无输出	11025	11029	0.04	0
48000000	16	94	0	无输出	8000	7979	-0.26	0
48000000	32	47	0	无输出	8000	7979	-0.26	0
48000000	16	2	0	输出	48000	46875	-2.34	12000000
48000000	32	2	0	输出	48000	46875	-2.34	12000000
48000000	16	2	0	输出	44100	46875	6.29	12000000
48000000	32	2	0	输出	44100	46875	6.29	12000000
48000000	16	3	0	输出	32000	31250	-2.34	8000000
48000000	32	3	0	输出	32000	31250	-2.34	8000000
48000000	16	4	1	输出	22050	20833	-5.52	5333248
48000000	32	4	1	输出	22050	20833	-5.52	5333248
48000000	16	6	0	输出	16000	15625	-2.34	4000000
48000000	32	6	0	输出	16000	15625	-2.34	4000000
48000000	16	8	1	输出	11025	11029	0.04	2823424
48000000	32	8	1	输出	11025	11029	0.04	2823424
48000000	16	11	1	输出	8000	8152	1.9	2086912
48000000	32	11	1	输出	8000	8152	1.9	2086912

下表提供了时钟 48M 时钟使用**小数分频**针对不同采样率的设置及精度值。

I2S CLK	通道长度	I2SDIV	FRACT	MCLK	采样率	实际频率	误差%	freq(mck)
48000000	16	7	52	无输出	96000	96000	0	0
48000000	32	3	58	无输出	96000	96000	0	0
48000000	16	15	40	无输出	48000	48000	0	0
48000000	32	7	52	无输出	48000	48000	0	0
48000000	16	17	0	无输出	44100	44118	0.04	0
48000000	32	8	32	无输出	44100	44118	0.04	0
48000000	16	23	28	无输出	32000	32000	0	0
48000000	32	11	46	无输出	32000	32000	0	0
48000000	16	34	1	无输出	22050	22049	0	0
48000000	32	17	0	无输出	22050	22059	0.04	0
48000000	16	46	56	无输出	16000	16000	0	0
48000000	32	23	28	无输出	16000	16000	0	0
48000000	16	68	2	无输出	11025	11024	-0.01	0
48000000	32	34	1	无输出	11025	11024	-0.01	0
48000000	16	93	48	无输出	8000	8000	0	0
48000000	32	46	56	无输出	8000	8000	0	0
48000000	16	1	61	输出	48000	48000	0	12288000
48000000	32	1	61	输出	48000	48000	0	12288000
48000000	16	2	8	输出	44100	44118	0.04	11294208
48000000	32	2	8	输出	44100	44118	0.04	11294208
48000000	16	2	60	输出	32000	31915	-0.27	8170240
48000000	32	2	60	输出	32000	31915	-0.27	8170240
48000000	16	4	16	输出	22050	22059	0.04	5647104
48000000	32	4	16	输出	22050	22059	0.04	5647104
48000000	16	5	55	输出	16000	16000	0	4096000
48000000	32	5	55	输出	16000	16000	0	4096000
48000000	16	8	32	输出	11025	11029	0.04	2823424
48000000	32	8	32	输出	11025	11029	0.04	2823424
48000000	16	11	46	输出	8000	8000	0	2048000
48000000	32	11	46	输出	8000	8000	0	2048000

24.4I2S 工作模式

24.4.1 I2S 主模式

I2S 可配置为主模式。这意味着将在 CK 引脚输出串行时钟，在 WS 引脚生成字选信号。主时钟（MCK）可以输出，也可以不输出，具体由 I2Sx_PR 寄存器中的 MCKOE 位控制。

步骤：

1. 设置 I2Sx_PR 寄存器的 I2SDIV[7:0] 位，以定义串行时钟波特率，从而达到相应的音频采样频率。I2Sx_PR 寄存器的 ODD 与 FRACT 位根据需要进行设置。
2. 如果需要为外部 DAC/ADC 音频组件提供主时钟 MCK，则将 I2Sx_PR 寄存器的 MCKOE 位置 1（I2SDIV 和 ODD （FRACT）值应根据 MCK 输出的状态进行计算。有关详细信息，请参见时钟发生器。
3. 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。此外，通过 I2Sx_CFGR 寄存器的 I2SCFG[1:0] 位选择 I2S 主模式和方向（发送器或接收器）。
4. 如果需要，通过对 I2Sx_CR2 寄存器执行写操作来选择所有可能的中断源和 DMA 功能。
5. I2Sx_CFGR 寄存器的 I2SE 位必须置 1。

WS 和 CK 的 IO 配置为输出模式。如果 I2Sx_PR 的 MCKOE 位置 1，则 MCK 也需要配置为输出。

发送序列

将半字写入发送缓冲区后，发送序列随即开始。

假设写入发送缓冲区的左声道第一个数据写入数据寄存器 L; 写入发送缓冲区的右声道第一个数据写入数据寄存器 R。数据从发送缓冲区传输到移位寄存器时，相应通道的 TXE 置 1，并且必须将对应于相应通道的数据写入相应通道的发送缓冲区。

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，数据按半字并行加载到 16 位移位寄存器中，然后以串行方式移位并输出到 MOSI/SD 引脚（MSB 在前）。每次数据从发送缓冲区传输到移位寄存器后，相应通道的 TXE 标志都将置 1，如果 I2Sx_CR2 寄存器的 TXEIE 位置 1，将产生中断。

有关各种 I2S 标准模式的写操作的更多详细信息，请参见 I2S 音频协议说明。

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的数据写入 I2Sx_DRL/DRR 寄存器。

要通过将 I2SE 清零来关闭 I2S，必须等待 TXE = 1 且 BSY = 0。

接收序列

此工作模式与发送模式相同，只有第 3 点存在不同（请参见 I2S 主模式所述的步骤），即通过 I2SCFG[1:0] 位设置主器件接收模式。

无论数据或通道长度如何，音频数据始终按 16 位数据包进行接收。这意味着，每当接收相应通道缓冲区有接收到数据时,相应通道的 RXNE 标志即置 1,并且如果 I2Sx_CR2 寄存器的 RXNEIE 位置 1，还将产生中断。所接收的右通道或左通道的音频值可通过一次或两次接收操作进入相应通道的接收缓冲区，具体取决于数据和通道长度配置。

读取 I2Sx_DRL/DRR 寄存器即会使相应通道的 RXNE 位清零。

有关各种 I2S 标准模式中读操作的更多详细信息，请参见 I2S 音频协议说明。

如果在先前收到的数据尚未读取时又接收到新数据，将产生上溢错误并将 OVR 标志置 1。数据接收寄存的值保持先前的值，当前接收的数据丢失。

如果 I2Sx_CR2 寄存器的 ERRIE 位置 1，将产生中断以指示该错误。

要关闭 I2S，需要执行特定操作来确保 I2S 正确完成传输周期而不启动新的数据传输。该序列取决于数据和通道长度的配置，以及所选的音频协议模式。在以下情况下：

32 位通道长度上扩展的 16 位数据长度（DATLEN = 00 且 CHLEN = 1），使用 LSB 对齐模式 (I2SSTD = 10)

a) 等待倒数第二个 RXNE = 1 (n – 1)
b) 然后等待 17 个 I2S 时钟周期（使用软件循环）
c) 禁止 I2S (I2SE = 0)

32 位通道长度上扩展的 16 位数据长度（DATLEN = 00 且 CHLEN = 1），使用 MSB 对齐、I2S 或 PCM 模式（分别为 I2SSTD = 00、I2SSTD = 01 或 I2SSTD = 11）

a) 等待最后一个 RXNE
b) 然后等待 1 个 I2S 时钟周期（使用软件循环）
c) 禁止 I2S (I2SE = 0)

对于 DATLEN 和 CHLEN 的所有其它组合，无论通过 I2SSTD 位选择何种音频模式，都将执行以下序列来关闭 I2S：

a) 等待倒数第二个 RXNE = 1 (n – 1)
b) 然后等待一个 I2S 时钟周期（使用软件循环）
c) 禁止 I2S (I2SE = 0)

注：传输期间，BSY 标志保持低电平。

24.4.2 I2S 从模式

对于从配置而言，I2S 可配置为发送模式或接收模式。

此工作模式所遵循的规则与 I2S 主模式配置基本相同。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。

应遵循如下配置步骤：

1. 通过 I2SSTD[1:0] 位选择 I2S 标准，通过 DATLEN[1:0] 位选择数据长度并通过配置 CHLEN 位选择帧中每个通道的位数。此外，通过 I2Sx_CFGR 寄存器的 I2SCFG[1:0] 位选择从器件的模式（发送或接收）。
2. 如果需要，通过对 I2Sx_CR2 寄存器执行写操作来选择所有可能的中断源和 DMA 功能。
3. I2Sx_CFGR 寄存器的 I2SE 位必须置 1。

发送序列

当外部主器件发送时钟并且通过 I2Sx_WS 信号请求传输数据时，发送序列开始。必须首先使能从器件，然后外部主器件才能开始通信。主器件开始通信前，还必须加载 I2S 数据寄存器。

对于 I2S、MSB 对齐和 LSB 对齐模式，要发送的第一个数据为左声道数据，分别在左右声道数据寄存器写入相应通道的数据。通信开始时，数据从发送缓冲区传输到移位寄存器。相应通道 TXE 标志随即置 1，以请求将相应通道的下一个数据写入 I2S 相应通道数据寄存器。

从器件需要首先为发送第一个数据做好准备，然后主器件才能产生时钟。WS 置位意味着首先发送左通道数据。

注：必须要在主器件发出的第一个时钟出现在 CK 线上至少 2 个 PCLK 周期之前置位 I2SE。

首位发送期间，数据按半字从内部总线并行加载到 16 位移位寄存器中，然后以串行方式移位并输出到 MOSI/SD 引脚（MSB 在前）。每次数据从发送缓冲区传输到移位寄存器后，相应通道的 TXE 标志都将置 1，如果 I2Sx_CR2 寄存器的 TXEIE 位置 1，将产生中断。

请注意，仅当 TXE 标志为 1 时，才可以尝试向发送缓冲区写入数据。

有关各种 I2S 标准模式中写操作的更多详细信息，请参见 I2S 音频协议说明。

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 I2Sx_DRL/R 寄存器。如果在数据尚未写入 I2Sx_DR 寄存器时下一个数据通信的首个时钟边沿到来，下溢标志将置 1 并可能产生中断。通过这种方式，软件可以获知所传输的数据不正确。如果 I2Sx_CR2 寄存器的 ERRIE 位置 1，则当 I2Sx_SR 寄存器中的 UDR 标志变为 1 时，将产生中断。这种情况下，必须关闭 I2S 并从左通道开始重新启动数据传输。

要通过将 I2SE 位清零来关闭 I2S，必须等待 TXE = 1 且 BSY = 0。

接收序列

此工作模式与发送模式相同，只有第 1 点存在不同（请参见 I2S 从模式所述的步骤），即通过 I2Sx_CFGR 寄存器的 I2SCFG[1:0] 位设置从器件接收模式。

无论数据长度或通道长度如何，音频数据始终按 16 位数据包进行接收。这意味着，每当接收缓冲区有数据时,I2Sx_SR 寄存器中的相应通道的 RXNE 标志即置 1,并且如果 I2Sx_CR2 寄存器的 RXNEIE 位置 1，还将产生中断。所接收的右通道或左通道的音频值可能通过一次或两次接收操作进入相应通道的接收缓冲区，具体取决于数据长度和通道长度配置。

每次接收的数据要从相应通道的数据寄存器读取，读取 I2Sx_DR 寄存器即会使 RXNE 位清零。

有关各种 I2S 标准模式的读操作的更多详细信息，请参见 I2S 音频协议说明。

如果在先前收到的数据尚未读取时又接收到新数据，将产生上溢错误，OVR 标志将置 1。如果 I2Sx_CR2 寄存器的 ERRIE 位置 1，将产生中断以指示该错误。

要在接收模式下关闭 I2S，必须在接收到最后一个 RXNE = 1 后立即将 I2SE 清零。

注：外部主器件应能够通过音频通道以 16 位或 32 位数据包发送/接收数据。

24.5I2S 状态标志

应用程序可通过三个状态标志来全面监视 I2S 总线的状态。

24.5.1 忙标志（BSY)

BSY 标志由硬件置 1 和清零（写入此标志没有任何作用)。该标志表示 I2S 通信层的状态。

BSY 置 1 时，表示 I2S 正在忙于通信。在主接收模式 (I2SCFG = 11) 中，BSY 标志的情况例外，该标志在接收期间仍保持低电平。

如果软件需要禁止 I2S，可使用 BSY 标志检测传输是否结束。这可以避免损坏最后传输的数据。为此，必须严格遵循下述步骤。

在传输开始时（I2S 处于主接收器模式时除外)，将 BSY 标志置 1

出现以下情况时，BSY 标志被硬件清零：

- 传输完成时（主发送模式除外，在该模式下通信是连续的)
- 禁止 I2S 时

当通信连续时：

- 在主发送模式下，BSY 标志在所有传输期间均保持高电平
- 在从模式下，BSY 标志在每次传输之间变为低电平并持续一个 I2S 时钟周期

注：请勿使用 BSY 标志处理每次数据发送或接收，最好改用 TXE 标志和 RXNE 标志。

24.5.2 发送缓冲区为空（TXE_L TXE_R）

如果此标志置 1，表示发送缓冲区为空，可将要发送的个数据加载到其中。发送缓冲区已包含要发送的数据时，TXE 标志复位。禁止 I2S（I2SE 位复位）时，该标志也会复位。

24.5.3 接收缓冲区非空（RXNE_L RXNE_R）

此标志置 1 时，表示接收缓冲区中存在有效的已接收数据。读取 I2Sx_DRL/R 寄存器时，该标志复位。

24.5.4 下溢标志（UDR）

在发送模式下，如果在软件尚未将任何值加载到 I2Sx_DR 之前出现第一个数据发送时钟，此标志将置 1。如果 I2Sx_CR2 寄存器中的 ERRIE 位置 1，可产生中断。

UDR 位通过 I2Sx_ICLR 进行清零。

24.5.5 上溢标志（OVR）

如果在尚未从 I2Sx_DR 寄存器读取上一个数据时又接收到新数据，此标志将置 1。因此，传入的数据将丢失。如果 I2Sx_CR2 寄存器中的 ERRIE 位置 1，可产生中断。

这种情况下，将不会用接收到的新数据更新接收缓冲区的内容。对 I2Sx_DR 寄存器执行的

读操作将返回先前正确接收的数据。主器件后续发送的所有其它半字都将丢失。

要将 OVR 位清零，应首先对 I2Sx_DR 寄存器执行读操作，然后再对 I2Sx_SR 寄存器进行读访问。

24.5.6 帧错误标志（FRE）

仅当 I2S 配置为从模式时，此标志才可由硬件置 1。如果外部主器件没有按照从器件期望的那样改变 WS 线，则此标志将置 1。如果失去同步，要从此状态中恢复并将外部主器件与

I2S 从器件重新同步，需执行下列步骤：

1. 禁止 I2S。
2. 在 WS 线上检测到正确的电平时将其重新使能（WS 线在 I2S 模式下为高电平，在 MSB 对齐、LSB 对齐或 PCM 模式下为低电平）。

主器件与从器件之间的同步失效可能是由于 SCK 通信时钟或 WS 帧同步信号线上存在噪音干扰。如果 ERRIE 位置 1，可产生错误中断。同步失效标志（FRE）由软件操作 I2Sx_ICLR 进行清零。

24.6I2S 中断 DMA

下表为 I2S 中断列表

中断事件	事件标志	使能控制
发送缓存空	TXE_L TXE_R	TXEIE
接收缓存非空	RXNE RXNE_R	RXNEIE
上溢错误	OVR	ERRIE
下溢错误	UDR	
帧错误	FRE	

DMA 特性

在 I2S 模式下，发送数据和接收的数据可以使用 DMA 进行搬运。

24.7I2S 寄存器

基地址

I2S0：0x4000 2800

I2S1：0x4000 2C00

表 24-1 I2Sx 寄存器

寄存器	偏移地址	描述
I2Sx_CR	0x004	控制寄存器
I2Sx_SR	0x008	中断和状态寄存器
I2Sx_CFGR	0x01C	配置寄存器
I2Sx_PR	0x020	主模式分频寄存器
I2Sx_ICR	0x024	中断标志清除寄存器
I2Sx_DRL	0x028	左声道发送数据与接收数据寄存器
I2Sx_DRR	0x02C	右声道发送数据与接收数据寄存器

24.7.1 I2Sx 控制寄存器（I2Sx_CR）

偏移地址 0x04

复位值 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								TXEIE	RXNEIE	ERRIE	Res.			RDMAEN	LDMAEN
								RW	RW	RW				RW	RW

位	标记	功能描述
31:8	Reserved	保留
7	TXEIE	发送缓冲区空中断使能（Tx buffer empty interrupt enable) 0：屏蔽 TXE 中断 1：使能 TXE 中断。TXE 标志置 1 时产生中断请求。
6	RXNEIE	接收缓冲区非空中断使能（RX buffer not empty interrupt enable) 0：屏蔽 RXNE 中断 1：使能 RXNE 中断 RXNE 标志置 1 时产生中断请求。
5	ERRIE	错误中断使能（Error interrupt enable) 此位用于在出现错误条件(UDR、OVR 和 FRE）时控制中断的生成。 0：屏蔽错误中断 1：使能错误中断
4:2	Reserved	保留
1	RDMAEN	右声道缓冲区 DMA 数据发送接使能（Right buffer DMA enable) 当此位置 1 时，每当右声道有数据发送接收标志置 1 时，即产生 DMA 请求。 0：禁止右声道缓冲区 DMA 1：使能右声道缓冲区 DMA
0	LDMAEN	左声道缓冲区 DMA声道数据发送接使能（Left buffer DMA enable) 当此位置 1 时，每当左声道有数据发送接标志置 1 时，即产生 DMA 请求。 0：禁止左声道缓冲区 DMA 1：使能左声道缓冲区 DMA

24.7.2 I2Sx 状态寄存器（I2Sx_SR）

偏移地址 0x08

复位值 0x0000 8002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Resereved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXE_R	RXNE_R	OVR_R	Res				FRE	BSY	OVR_L	Res.		UDR_R	UDR_L	TXE_L	RXNE_L
RO	RO	RO					RO	RO	RO			RO	RO	RO	RO

位	标记	功能描述
31:16	Reserved	保留
15	TXE_R	右通道发送缓冲区为空（Transmit buffer empty） 0：发送缓冲区非空 1：发送缓冲区为空
14	RXNE_R	右通道接收缓冲区非空（Receive buffer not empty） 0：发送缓冲区为空 1：接收缓冲区非空
13	OVR_R	右通道上溢标志（Overrun flag） 0：未发生上溢 1：发生上溢 此标志由硬件置 1，由软件清零
12:9	Reserved	保留
8	FRE	帧错误（Frame Error） 0：无帧错误 1：发生帧错误 该位由硬件置 1，由软件清零。 无论选择哪种音频协议，它会在从模式下的非预期时间检测WS 线的变化，通知外部主器件和从器件之间的同步失效情况。
7	BSY	BSY：忙标志（Busy flag） 0： I2S不繁忙 1： I2S忙于通信或者发送缓冲区不为空 此标志由硬件置 1 和清零。
6	OVR_L	左通道上溢标志（Overrun flag） 0：未发生上溢 1：发生上溢 此标志由硬件置 1，由软件清零
5:4	Reserved	保留
3	UDR_R	右通道下溢标志（Underrun flag）

		0：未发生下溢 1：发生下溢 此标志由硬件置 1，由软件清零
2	UDR_L	左通道下溢标志 (Underrun flag) 0：未发生下溢 1：发生下溢 此标志由硬件置 1，由软件清零
1	TXE_L	左通道发送缓冲区为空 (Transmit buffer empty) 0：发送缓冲区非空 1：发送缓冲区为空
0	RXNE_L	左通道接收缓冲区非空 (Receive buffer not empty) 0：发送缓冲区为空 1：接收缓冲区非空

24.7.3 I2Sx 配置寄存器（I2Sx_CFGR）

偏移地址 0x1C

复位值 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Resereved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resereved					I2SE	I2SCFG		PCMSYNC	CKSEL	I2SSTD		Res	DATLEN		CHLEN
					RW	RW		RW	RW	RW			RW		RW

位	标记	功能描述
31:11	Reserved	保留
10	I2SE	I2S 使能（I2S Enable） 0：禁止 I2S 1：使能 I2S
9:8	I2SCFG	I2S 配置模式（I2S configuration mode） 00：从模式 - 发送 01：从模式 - 接收 10：主模式 - 发送 11：主模式 - 接收
7	PCMSYNC	PCM 帧同步（PCM frame synchronization） 0：短帧同步 1：长帧同步 注： 只有在 I2SSTD = 11（使用 PCM 标准）时，此位才有意义
6	CKSEL	主模式下I2S时钟选择 0 PCLK 1 HCLK
5:4	I2SSTD	I2S 标准选择（I2S standard selection） 00：I2S Philips 标准。 01：MSB 对齐标准（左对齐） 10：LSB 对齐标准（右对齐） 11：PCM 标准 注： 为确保正确运行，应在 I2S 禁止时配置这些位
3	Reserved	保持值为0
2:1	DATLEN	要传输的数据长度（Data length to be transferred） 00：16 位数据长度 01：24 位数据长度 10：32 位数据长度 11：不允许 注： 为确保正确运行，应在 I2S 禁止时配置这些位。

0	CHLEN	CHLEN：通道长度（每个音频通道的位数）(Channel length (number of bits per audio channel)) 0：16 位 1：32 位 只有在 DATLEN = 00 时，此位的写0操作才有意义，在其他数据长度下此位被强制写1。 注： 为确保正确运行，应在 I2S 禁止时配置此位。

24.7.4 I2Sx 预分频寄存器（I2Sx_PR）

偏移地址 0x20

复位值 0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Resereved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRACT						MCKOE	ODD	I2SDIV							
RW						RW	RW	RW							

位	标记	功能描述
31:16	Reserved	保留
15:10	FRACT	预分频器小数分频设置，0x00时ODD设置有效，小数分频为0，非零时，ODD设置无效 实际分频值为 = (I2SDIV +FRACT/64)*2 注： 应在 I2S 禁止时配置此位。
9	MCKOE	主时钟输出使能 (Master clock output enable) 0：禁止主时钟输出 1：使能主时钟输出 注： 应在 I2S 禁止时配置此位。
8	ODD	预分频器的奇数因子 (Odd factor for the prescaler) 0：实际分频值为 = I2SDIV *2 1：实际分频值为 = (I2SDIV * 2)+1 详细参见预分频设置 注： 应在 I2S 禁止时配置此位。
7:0	I2SDIV	I2SDIV：I2S 线性预分频器 (I2S Linear prescaler) I2SDIV [7:0] = 0 为禁用值。 详细参见预分频设置 注： 应在 I2S 禁止时配置这些位。

24.7.5 I2Sx 中断状态清除寄存器（I2Sx_ICR）

偏移地址 0x24

复位值 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							FRE	Res	OVR	Res		UDF	Res		
							W0		W0			W0			

位	标记	功能描述
31:9	Reserved	保留
8	FRE	帧错误 (Frame Error)标志清除 写0清除 写1无效
7	Reserved	保留
6	OVR	上溢标志 (Overrun flag) 写0清除 写1无效
5:4	Reserved	保留
3	UDF	下溢标志 (Underrun flag) 写0清除 写1无效
2:0	Reserved	保留

24.7.6 I2Sx 数据寄存器 L（I2Sx_DRL）

偏移地址 0x28

复位值 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRL															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	DRL	数据寄存器（Data register） 已接收或者要发送的左声道数据与PCM的数据。 数据寄存器分为 2 个缓冲区，一个用于写入（发送缓冲区），一个用于读取（接收缓冲区）。对数据寄存器执行写操作时，数据将写入发送缓冲区，从数据寄存器执行读取时，将返回接收缓冲区中的值。

24.7.7 I2Sx 数据寄存器 R（I2Sx_DRR）

偏移地址 0x2C

复位值 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRR															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	DRR	数据寄存器（Data register） 已接收或者要发送的右声道数据。PCM模式下无效。 数据寄存器分为 2 个缓冲区，一个用于写入（发送缓冲区），一个用于读取（接收缓冲区）。对数据寄存器执行写操作时，数据将写入发送缓冲区，从数据寄存器执行读取时，将返回接收缓冲区中的值。

25 USB2.0 全速模块（USBFS）

25.1 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 控制器支持设备模式，且芯片内部集成全速 PHY。设备模式下支持全速（FS，12Mb/s）收发器。USBFS 控制器支持 USB2.0 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）。

25.2 USBFS 主要特性

主要分为两类：通用特性和设备模式特性。

25.2.1 通用特性

- 内置片上 USB2.0 全速 PHY
- 支持设备模式
- 模块内嵌 DMA，并可软件配置 AHB 突发传输类型
- 具备省电功能，例如 USB 挂起，停止 RAM 时钟，停止 PHY 域时钟
- 具有采用高级 FIFO 控制的 1.25KB 专用 RAM
- 可以将 RAM 空间划分为不同的 FIFO，以便灵活有效的使用 RAM
- 每个 FIFO 可存储多个数据包
- 动态分配存储区
- FIFO 的大小可配置成为非 2 的幂次方值，以便连续使用存储单元
- 一帧之内可以不需要应用程序干预，以达到最大 USB 带宽

25.2.2 设备模式特性

- 设备模式支持 USB 2.0 全速（FS，12Mb/s）传输。
- 1 个双向控制端点 0
- 4 个 IN 端点（1，3，5，7），可以配置为批量传输、中断传输或同步传输
- 4 个 OUT 端点（2，4，6，8），可以配置为批量传输、中断传输或同步传输
- 8 个端点只能选择一个配置为同步传输（周期传输）模式
- 包含一个非周期发送 FIFO（由所有非周期 IN 端点共享），一个周期发送 FIFO（由周期 IN 端点独享）和一个接收 FIFO（由所有 OUT 端点共享）
- 支持远程唤醒功能。
- 支持软断开功能

25.3USBFS 系统框图

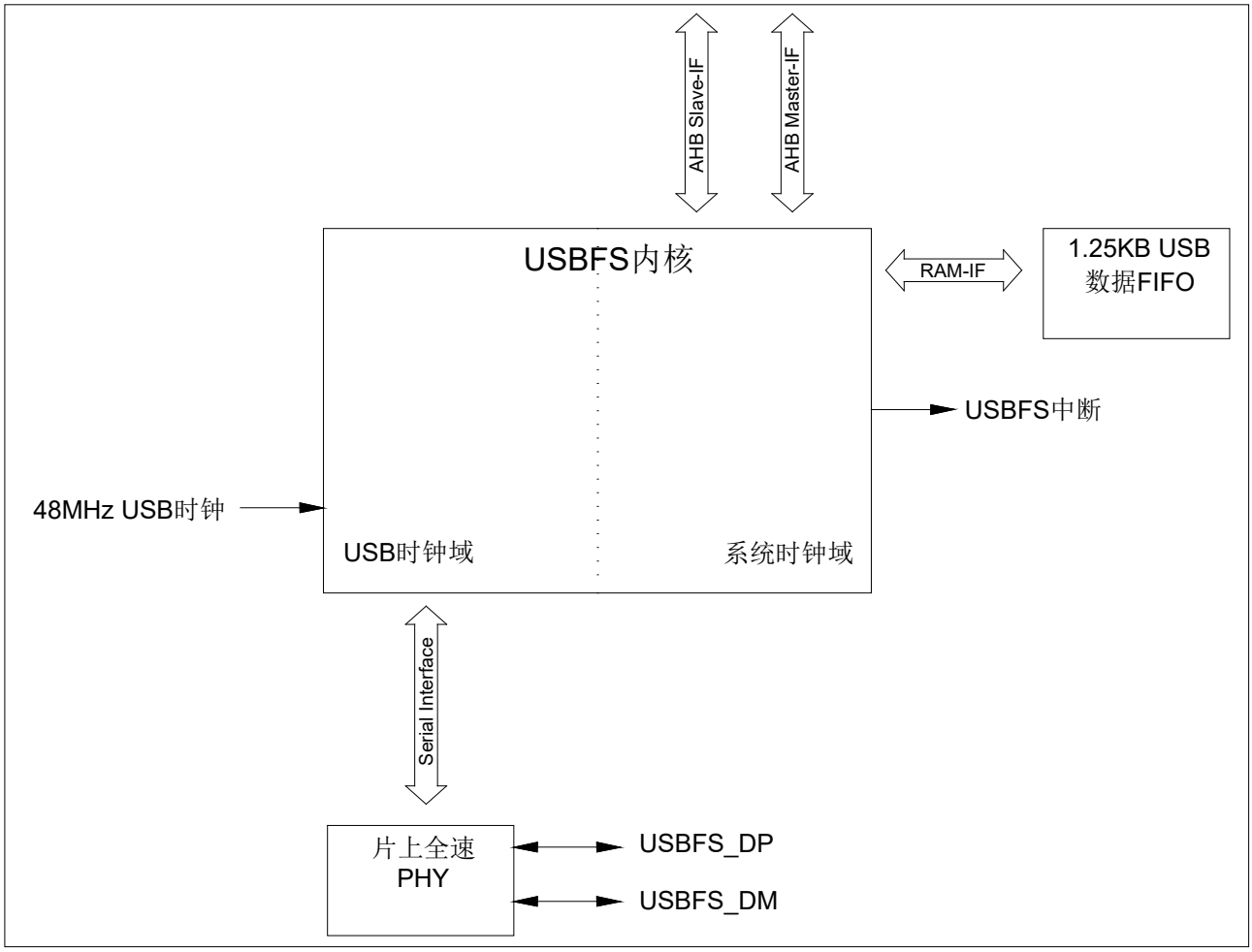


图 25-1 USBFS 系统框图

25.4USBFS 管脚说明

表 25-1 USBFS 管脚说明

管脚名	方向	功能描述
USBFS_DP	输入/输出	差分数据D+信号
USBFS_DM	输入/输出	差分数据D-信号

由于 USBFS_DP 和 USBFS_DM 管脚和通用 GPIO 复用，在使用 USBFS 时，建议关闭其对应管脚的数字功能，具体请参考端口控制器（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。

25.5USBFS 功能说明

25.5.1 USBFS 时钟以及工作模式

USBFS 所使用的时钟有 PHY 时钟和系统时钟，其中 PHY 时钟频率需要配置为 48MHz，该 48MHz 时钟可由内部 PLL 电路产生，也可由内部高速 48M RC 时钟产生，使用 USBFS 模块前，需要在系统控制器模块内配置好 USBFS 时钟。

USBFS 作为设备使用，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。

25.5.2 USBFS 设备功能

25.5.2.1 设备功能简介

典型的 USB 设备模式系统构建图如下：

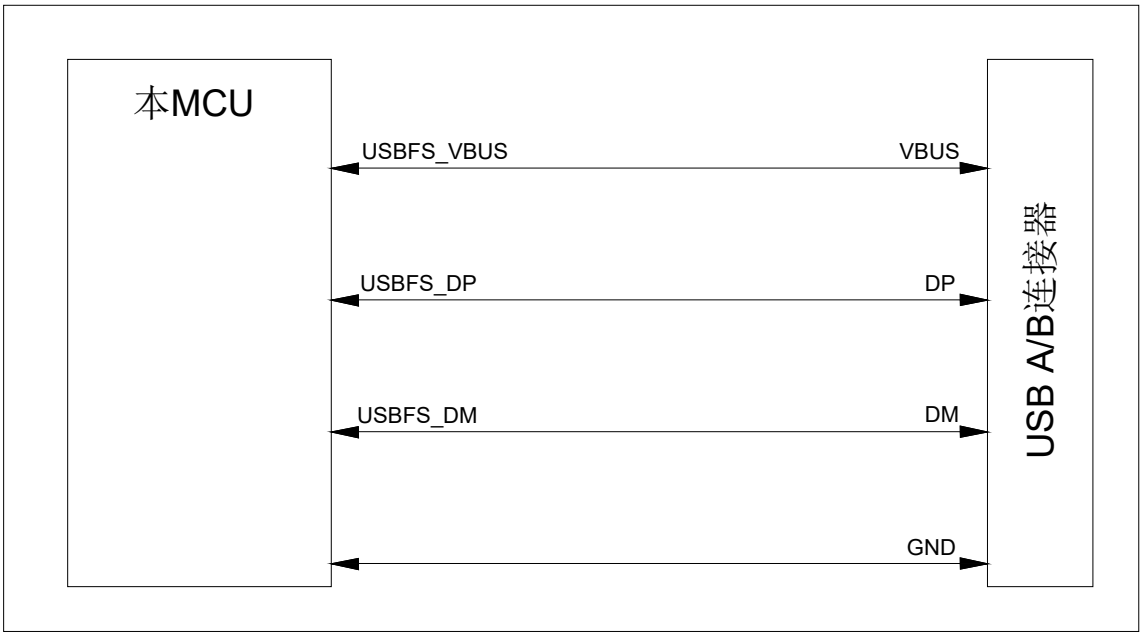


图 25-2 USBFS 设备模式系统构建图

25.5.2.2 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（请参见 USB2.0 第 9.1 节）。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。

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。

25.5.2.3 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。

25.5.2.4 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 后将其清零。

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。

25.5.2.5 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。

25.5.2.6 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
  - 使用一组单独的寄存器来处理 IN 和 OUT 事务
  - 专用控制 (USBFS_DIEPCTL0/USBFS_DOEPCTL0) 寄存器、传输配置 (USBFS_DIEPTSIZ0/USBFS_DOEPTSIZ0) 寄存器和状态中断 (USBFS_DIEPINT0/USBFS_DOEPINT0)寄存器。控制和传输配置寄存器中可用的位组与其它端点中稍有不同

- 4 个 IN 端点（n = 1,3,5,7）
  - 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_DIEPCTLn）寄存器、传输配置（USBFS_DIEPTSIZn)寄存器和状态中断(USBFS_DIEPINTn）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
- 4 个 OUT 端点(n = 2,4,6,8)
  - 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_DOEPCTLn）寄存器、传输配置(USBFS_DOEPTSIZn）寄存器和状态中断（USBFS_DOEPINTn）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 n IN/OUT 控制寄存器（DIEPCTLn/DOEPCTLn）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 n 传输尺寸寄存器（DIEPTSIZn/DOEPTSIZn）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 n 中断寄存器（DIEPINTn/DOPEPINTn）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 n 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）
- 同步传输类型的数据包未产生中断而丢失

25.5.3 USBFS 功耗控制

不使用 USBFS 模块时，可以通过系统控制器模块停止 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。

25.5.4 USBFS 数据 FIFO

USBFS 系统具有 1.25KB 专用 RAM，采用高效的 FIFO 控制机制。USBFS 模块中的数据包 FIFO 控制器模块将 RAM 空间划分为一个非周期 TxFIFO，一个周期 TxFIFO（USB 传输前，应用程序将数据压入其中进行短暂存储）和一个 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）。

其中非周期 TxFIFO 由所有的非周期 IN 端点共享，RxFIFO 由所有的 OUT 端点共享。FIFO 的大小均由软件配置，以更好地满足应用要求。

25.5.5 USBFS 设备 FIFO 架构

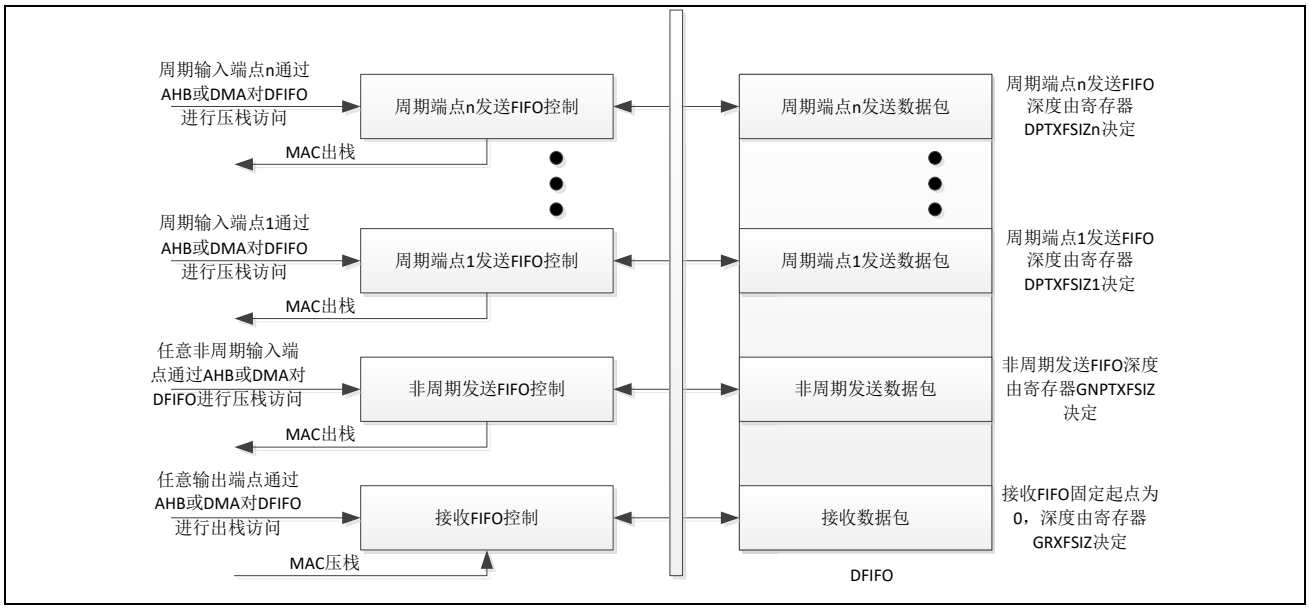


图 25-3 USBFS 设备模式下 FIFO 架构示意图

25.5.5.1 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。

25.5.5.2 设备 TxFIFO

所有的非周期 IN 端点共享一个 TxFIFO。周期 IN 端点独享一个 TxFIFO。应用程序通过非周期发送 FIFO 大小寄存器（USBFS_GNPTXFSIZ)为非周期 IN 端点配置 FIFO 大小；通过周期 IN 端点发送 FIFOn 大小寄存器（DPTXFSIZn)为周期 IN 端点 n 配置 FIFO 大小。

25.5.6 USBFS FIFO RAM 分配

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。

25.5.7 USBFS 系统性能

凭借大容量 RAM 缓冲区、高度可配置的 FIFO 大小、通过 AHB 压栈/出栈寄存器进行 32 位 FIFO 快速访问，尤其是高级 FIFO 控制机制可获得最佳 USB 和系统性能。实际上，无论当前 USB 序列如何，USBFS 均可通过该机制高效填充可用的 RAM 空间。借助这些特性：

- 应用程序有足够的裕量来计算并校正 CPU 的负载，从而优化 CPU 带宽利用率：
  - 应用程序可先积累大量发送数据，再通过 USB 发送出去
  - 可带来足够的时间裕量，以从接收 FIFO 读取数据
- USB 模块能够保持全速工作状态，也就是提供最大的全速带宽（尽量多的硬件自动运行，尽量少的软件参与）
  - USB 模块可提前积累大量发送数据供其支配，从而可对 USB 数据发送进行自主管理
  - 接收缓冲区中有大量空白空间，可通过 USB 中的数据自动填满

由于 USBFS 模块能够高效填充 1.25KB RAM 缓冲区且 1.25KB 发送/接收数据足以满足一个全速帧所能容纳的数据量，因此 USB 系统在一帧之内可以无需应用程序干预达到最大 USB 带宽。

25.5.8 USBFS 中断和事件

全局中断是软件需要处理的主要中断，全局中断的标志位可在 USBFS_GINTSTS 寄存器读取。

表 25-2 USBFS 中断事件表

中断标志	描述
WKUPINT	远程唤醒中断
DATAFSUSP	数据获取挂起
INCOMPISOOUT	未完成OUT同步传输
IISOIXFR	未完成IN同步传输
OEPINT	OUT端点中断
IEPINT	IN端点中断
EOPF	周期性帧结束中断
ISOODRP	丢弃同步OUT数据包中断
ENUMDNE	枚举完成
USBRST	USB复位中断
USBSUSP	USB挂起中断
ESUSP	早期挂起中断
GONAKEFF	全局OUT NAK有效中断
GINAKEFF	全局非周期性IN NAK有效中断
RXFNE	RxFIFO非空中断
SOF	帧起始中断
MMIS	模式不匹配中断

25.6USBFS 编程模型

25.6.1 USBFS 模块初始化

USBFS 模块工作需要两个时钟，系统时钟和 PHY 时钟，其中 PHY 时钟频率必须配置为 48Mhz。PHY 时钟可以选择 PLL 时钟或者内部高速 RC48M 时钟。

1. 配置 PLL 时钟或者 RC48M 时钟，等待时钟输出稳定（操作流程请参考系统控制器（SYSCTRL）章节）。
2. 配置系统控制器寄存器 SYSCTRL1.USB48MSEL，置 0 选择 RC48M 时钟，置 1 选择 PLL 时钟。
3. 配置系统控制器寄存器 PERI_CLKEN1.USB 置 1 使能 USBFS 模块时钟。

25.6.2 USBFS 设备初始化

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. Slave 模式下，USBFS_GINTMSK.NPTxFEmpMsk 和 USBFS_GINTMSK.RxFLvlMsk 置 1，DMA 模式下，USBFS_GINTMSK.NPTxFEmpMsk 和 USBFS_GINTMSK.RxFLvlMsk 置 0
2. 在 USBFS_DCFG 寄存器中编程以下字段：
   - 设备速度（DevSpd）
   - 非零长度状态输出握手信号（NZStsOUTHShk）
   - 周期帧间隔（PerFrInt）
3. 清除 USBFS_DCTL.SftDiscon 位，让控制器执行连接到主机。
4. 编程 USBFS_GINTMSK 寄存器以取消屏蔽以下中断：
   - USB 复位（USBRstMsk）
   - 枚举完成(EnumDoneMsk)
   - 早期挂起(ErlySuspMsk)
   - USB 挂起(USBSuspMsk)
   - SOF(SofMsk)
5. 等待 USBFS_GINTSTS.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。应用程序必须执行章节 USB 复位时的端点初始化中所列的步骤。
6. 等待 USBFS_GINTSTS.EnumDone 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 USB 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。

25.6.3 USBFS 设备软断开操作

应用程序可以置位寄存器 USBFS_DCTL.SftDiscon 执行一个软断开操作。当设备处于数据传输状态时，执行以下步骤：

1. 置位寄存器 USBFS_DCTL.SftDiscon
2. 置位寄存器 USBFS_GRSTCTL.CSftRst
3. 等待寄存器 USBFS_GRSTCTL.CSftRst 位被控制器清零
4. 重新执行 USBFS 设备初始化步骤

25.6.4 USB 复位时的端点初始化

1. 把所有输出端点 NAK 位置 1

   USBFS_DOEPCTLn .SNAK = 1（对于所有输出端点）
2. 取消对以下中断位的屏蔽

   USBFS_DAINTMSK.INEP0=1（控制 0 输入端点）

   USBFS_DAINTMSK.OUTEP0=1（控制 0 输出端点）
   USBFS_DOEPMSKn.SetUPMsk=1
   USBFS_DOEPMSKn.XferComplMsk=1
   USBFS_DIEPMSKn.XferComplMsk=1
   USBFS_DIEPMSKn.TimeOUTMsk=1
3. 为每个 FIFO 设置数据 FIFO RAM

   编程 USBFS_GRXFSIZ 寄存器，以能够接收控制传输的输出数据和 SETUP 数据。该寄存器值必须至少等于控制端点 0 的 1 个最大数据包大小+2 个字（用于控制输出数据包的状态）+10 个字（用于 SETUP 数据包）。

   编程 USBFS_GNPTXFSIZ 寄存器，以能够发送控制输入数据。该寄存器值必须至少等于控制端点 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，以使控制输出端点 0 接收 SETUP 数据包。

   – USBFS_DOEPTSIZ0.SUPCnt=3（接收最多 3 个连续的 SETUP 数据包）

   – DMA 模式下，USBFS_DOEPDMA0 寄存器指定内存地址存放收到的 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。

25.6.5 USB 枚举完成时的端点初始化

1. 在枚举完成中断（USBFS_GINTSTS.EnumDone）发生后，读取寄存器 USBFS_DSTS.EnumSpd 以确定设备的枚举速度。
2. 对 USBFS_DIEPCTL0 中的 MPS 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。
3. DMA 模式下，编程 USBFS_DOEPCTL0 寄存器使能控制输出端点 0。
   USBFS_DOEPCTL0.EPEna = 1
   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。

25.6.6 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输入数据包。

25.6.7 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。必须不激活这些无效端点。
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输入数据包。
   此时，设备模块已可以接收和发送任何类型的数据包

25.6.8 端点激活(Active)

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_DIEPCTLn 寄存器（对于输入端点）或 USBFS_DOEPCTLn 寄存器（对于输出端点）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（MPS）
   - USB 活动端点位（USBActEP）置 1
   - 端点初始数据同步位（DPID）（对于中断和批量端点）
   - 端点类型（EPType）
   - TxFIFO 编号(TxFNum)

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。

25.6.9 端点不激活(Deactive)

本节介绍不激活现有端点所需的步骤。

1. 在要不激活的端点中，将 USBFS_DIEPCTLn 寄存器（对于输入端点）或 USBFS_DOEPCTLn 寄存器（对于输出端点）中的 USB 活动端点位（USBActEP）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。

25.6.10 禁止(Disable)输出端点

应用程序必须按照以下步骤禁止一个已经使能的输出端点。

1. USBFS_DCTL.SGOUTNak=1，进入全局输出端点 NAK 模式
2. 等待中断 USBFS_GINTSTS.GOUTNakEff
3. USBFS_DOEPCTLn.EPDisable=1，USBFS_DOEPCTLn.SNAK=1
4. 等待中断 USBFS_DOEPINTx.EPDisbld，

   当中断发生时，控制器会清除以下寄存器
   USBFS_DOEPCTLx.EPDis = 0
   USBFS_DOEPCTLn.EPEna = 0
5. 程序必须退出全局输出端点 NAK 模式
   USBFS_DCTL.CGOUTNak = 1
   USBFS_DCTL.SGOUTNak=0

25.6.11 暂停(Stall)非同步输出端点

应用程序必须按照以下步骤暂停非同步输出端点。

1. USBFS_DCTL.SGOUTNak=1，进入全局输出 NAK 模式
2. 等待中断 USBFS_GINTSTS.GOUTNakEff
3. USBFS_DOEPCTLn.EPDisable=1，USBFS_DOEPCTLn.Stall=1
4. 等待中断 USBFS_DOEPINTx.EPDisbld，

   当中断发生时，控制器会清除以下寄存器
   USBFS_DOEPCTLx.EPDis = 0
   USBFS_DOEPCTLn.EPEna = 0
5. 程序必须退出全局输出端点 NAK 模式
   USBFS_DCTL.CGOUTNak = 1
   USBFS_DCTL.SGOUTNak=0

25.6.12 全局输出端点 NAK 模式

程序必须进入全局输出端点 NAK 模式以停止接收接收 FIFO 里的任何数据。

1. USBFS_DCTL.SGOUTNak=1，进入全局输出 NAK 模式。
2. 等待中断 USBFS_GINTSTS.GOUTNakEff，中断发生表示控制器已停止接收除 SETUP 数据包以外的数据。
3. 程序可以写寄存器屏蔽这个中断 USBFS_GINTMSK.GOUTNakEffMsk=0。
4. 如果程序要退出全局输出 NAK 模式，执行以下操作同时可以清除中断。
   USBFS_GINTSTS.GOUTNakEff
   USBFS_DCTL.CGOUTNak = 1
   USBFS_DCTL.SGOUTNak=0

25.6.13 输出端点传输停止

1. 使能所有的输出端点 USBFS_DOEPCTLn.EPEna = 1。
2. 等待 USBFS_GRSTCTL.AHBIdle 置位，表示控制器的 AHB 总线控制器空闲。
3. USBFS_GRSTCTL.RxFFlsh =1
4. 等待 USBFS_GRSTCTL.RxFFlsh 清零,表示接收 FIFO 成功清空。
5. 执行章节禁止(Disable)输出端点。

25.6.14 配置中断输入端点为周期或非周期端点（共享 FIFO 模式）

共享 FIFO 模式下，可以通过设置 USBFS_DIEPCTLn.TxFNum 值，把中断输入端点配置成周期输入端点或者非周期输入端点。

当配置成非周期输入端点时，程序必须遵照非周期输入端点的操作流程。

当配置成周期输入端点时，程序必须遵照周期输入端点的操作流程。

25.6.15 全局非周期输入端点 NAK 模式（共享 FIFO 模式）

1. 设置 USBFS_DCTL.SGNPInNak=1 ，通知控制器停止向非周期输入端点发送数据，进入全局非周期输入端点 NAK 模式。
2. 等待中断标志位 USBFS_GINTSTS.GINNakEff 置位，该标志位置位表示控制器已停止向非周期输入端点发送数据，但是在置位前控制器依然可以向非周期输入端点发送数据。
3. 中断标志位 USBFS_GINTSTS.GINNakEff 可以由
   USBFS_GINTMSK.GINTNakEffMsk 屏蔽，
   设置 USBFS_GINTMSK.GINTNakEffMsk=0，
   屏蔽中断标志位 USBFS_GINTSTS.GINNakEff
   设置 USBFS_GINTMSK.GINTNakEffMsk=1，
   取消屏蔽中断标志位 USBFS_GINTSTS.GINNakEff

4. 编程 USBFS_DCTL.CGNPInNak=1
   USBFS_DCTL.SGNPInNak=0 ,
   退出全局非周期输入端点 NAK 模式，
   同时清除中断标志位 USBFS_GINTSTS.GINNakEff。

25.6.16 发生输入端点 NAK

1. 停止向一个特别的输入端点发送数据，程序必须置位 NAK 位
   USBFS_DIEPCTLn.SNAK=1，进入输入端点 NAK 模式。
2. 等待中断标志位 USBFS_DIEPINTn.NAKIntrpt 置位，表示控制器已经停止向这个端点发送数据。
3. USBFS_DIEPMSK.NAKMsk = 0，可以屏蔽中断 USBFS_DIEPINTn.NAKIntrpt，
   USBFS_DIEPMSK.NAKMsk = 1，取消屏蔽中断 USBFS_DIEPINTn.NAKIntrpt。
4. 如果要退出输入端点 NAK 模式
   USBFS_DIEPCTLn.CNAK=1

25.6.17 禁止非周期输入端点（共享 FIFO 模式）

禁止端点操作必须在该端点被使能之后。禁止一个非周期输入端点，程序必须禁止所有非周期端点。

1. 设置 USBFS_DCTL.SGNPInNak=1，进入全局非周期输入端点 NAK 模式。
2. 等待中断标志位 USBFS_GINTSTS.GINNakEff 置位。
3. 设置 USBFS_DIEPCTLn.EPDis=1, USBFS_DIEPCTLn.SNAK=1，一个接一个，直到所有非周期端点被禁止。
4. 等待中断标志位 USBFS_DIEPINTn.EPDisbld 置位，该标志位置位表示控制器已经禁止对应的端点，同时清除寄存器位 USBFS_DIEPCTLn.EPEna 和 USBFS_DIEPCTLn.EPDis。依次判断所有的非周期端点已经被禁止。
5. 程序必须读出所有被禁止的非周期输入端点寄存器 USBFS_DIEPTSIZn 的值，计算出每个端点已经传输的数据量。
6. 程序必须清除非周期发送 FIFO，操作如下：
   USBFS_GRSTCTL.TxFNum=0
   USBFS_GRSTCTL.TxFFlsh=1
   等待控制器清除 USBFS_GRSTCTL.TxFFlsh， 表示非周期发送 FIFO 清除已结束。
7. 编程 USBFS_DCTL.CGNPInNak=1 USBFS_DCTL.SGNPInNak=0 ，退出全局非周期输入端点 NAK 模式。

25.6.18 禁止周期输入端点（共享 FIFO 模式）

禁止端点操作必须在该端点被使能之后。

1. 设置 USBFS_DIEPCTLn.SNAK=1,
   等待中断标志位 USBFS_DIEPINTn.INEPNakEff 置位,

该标志位置位表示控制器已停止向该端点发送数据。

2. 设置 USBFS_DIEPCTLn.EPDis=1，
   等待中断标志位 USBFS_DIEPINTn.EPDisbld 置位，
   该标志位置位表示控制器已经禁止对应的端点，同时清除寄存器位
   USBFS_DIEPCTLn.EPEna 和 USBFS_DIEPCTLn.EPDis。
3. 程序必须读出被禁止的周期输入端点寄存器 USBFS_DIEPTSIZn 的值，计算出该端点已经传输的数据量。
4. 程序必须清除非周期发送 FIFO，操作如下：
   USBFS_GRSTCTL.TxFNum= Periodic FIFO Number
   USBFS_GRSTCTL.TxFFlsh=1
   等待控制器清除 USBFS_GRSTCTL.TxFFlsh， 表示周期发送 FIFO 清除已结束。

25.6.19 非周期输入数据传输超时（共享 FIFO 模式）

1. 当某个非周期输入端点在传输数据时，
   中断标志位 USBFS_DIEPCTLn.TimeOut 置位，该标志位置位表示该端点数据传输超时。
2. 按照禁止非周期输入端点（共享 FIFO 模式）步骤执行操作。
3. 重新使能端点并传送超时的数据包。

25.6.20 非同步输入端点暂停（共享 FIFO 模式）

1. 设置 USBFS_DCTL.SGNPInNak=1，进入全局非周期输入端点 NAK 模式。
2. 等待中断标志位 USBFS_GINTSTS.GINNakEff 置位。
3. 设置 USBFS_DIEPCTLn.EPDis=1，USBFS_DIEPCTLn.Stall=1，一个接一个，直到所有非周期端点被禁止。
4. 等待中断标志位 USBFS_DIEPINTn.EPDisbld 置位，该标志位置位表示控制器已经禁止对应的端点，同时清除寄存器位 USBFS_DIEPCTLn.EPEna 和 USBFS_DIEPCTLn.EPDis。依次判断所有的非周期端点已经被禁止。
5. 程序必须读出所有被禁止的非周期输入端点寄存器 USBFS_DIEPTSIZn 的值，计算出每个端点已经传输的数据量。
6. 程序必须清除非周期发送 FIFO，操作如下：
   USBFS_GRSTCTL.TxFNum=0
   USBFS_GRSTCTL.TxFFlsh=1
   等待控制器清除 USBFS_GRSTCTL.TxFFlsh，表示非周期发送 FIFO 清除已结束。
7. 设置 USBFS_DCTL.SGNPInNak=0，退出全局非周期输入端点 NAK 模式。
8. 重新使能其他不需要暂停的端点
9. 如果要退出暂停模式，设置 USBFS_DIEPCTLn.Stall=0。

25.6.21 输入端点传输停止（共享 FIFO 模式）

1. 设置 USBFS_DIEPCTLn.EPDis=1。
2. 等待中断标志位 USBFS_DIEPINTn.EPDisbld 置位，该标志位置位表示控制器已经禁止对应的端点，同时清除寄存器位 DIEPCTLn.EPEna 和 DIEPCTLn.EPDis。
3. 程序必须清除非周期发送 FIFO，操作如下：
   USBFS_GRSTCTL.TxFNum=0
   USBFS_GRSTCTL.TxFFlsh=1
   等待控制器清除 USBFS_GRSTCTL.TxFFlsh， 表示非周期发送 FIFO 清除已结束。

25.6.22 端点不匹配（共享 FIFO 模式）

共享 FIFO 模式下，所有的非周期输入端点共用一个发送 FIFO,程序必须预测主机访问非周期输入端点的顺序，当顺序预测错误时，就会发生端点不匹配情况，中断标志位 USBFS_DIEPINTn.INTknEPMis 置位。

25.6.23 控制传输

控制传输有三种类型：

- 控制写传输
- 控制读传输
- 两阶段控制传输

每一个控制传输包含两个或者三个阶段：

- Setup 阶段
- 数据阶段
- 状态阶段

25.6.23.1 控制写传输编程流程

控制写传输 Setup 阶段

1. 在系统内存里创建一个缓存用来保存 Setup 数据包
2. 编程寄存器 USBFS_DOEPDMA0 来配置缓存列表的基地址
3. 编程寄存器 USBFS_DOEPTSIZ0 的 XferSize,PktCnt 和 SUPCnt 字段
4. 编程寄存器 USBFS_DOEPCTL0.MPS 来配置控制端点最大数据包大小，编程寄存器 USBFS_DOEPCTL0.EPEna 置 1 使能控制输出端点
5. 等待输出端点中断
6. 当控制器收到 Setup 令牌包时，会置位 USBFS_DOEPINT0 中断

a) 如果 USBFS_DOEPINT0.XferComp=1 并且 USBFS_DOEPINT0.SetUp=0，程序必须读寄存器 USBFS_DOEPINT0 的 StupPktRcvd 位。如果 StupPktRcvd=1，表示 Setup 令牌包已经收到。

程序必须解码主机发送的控制传输类型。然后程序必须检查 USBFS_DOEPINT0.StsPhaseRcvd 是否置 1，如果置 1，表示进入状态阶段，如果没有置 1，表示进入数据阶段。

b) 如果 USBFS_DOEPINT0.XferComp=1 并且 USBFS_DOEPINT0.SetUp=1，程序必须读 USBFS_DOEPINT0 寄存器的 StupPktRcvd 位。如果 StupPktRcvd=1，表示 Setup 令牌包已经收到。程序必须解码主机发送的控制传输类型。然后程序必须进入数据阶段。USBFS_DOEPINT0.SetUp=1 表示主机在控制写传输的数据阶段发送了输出令牌包。

控制写传输数据阶段

1. 创建 n 个缓存接收`wlength`字节数据确保每个缓存包含一个数据包 [n=wlength/MPS+wlengthMOD(MPS)]
2. 编程寄存器 USBFS_DOEPDMA0 来配置描述符列表的基地址
3. 编程寄存器 USBFS_DOEPTSIZ0 的 XferSize,PktCnt 和 SUPCnt 字段
4. 编程寄存器 USBFS_DOEPCTL0.MPS 来配置控制端点最大数据包大小，编程寄存器 USBFS_DOEPCTL0.EPEna 置 1 使能控制输出端点
5. 等待输出端点中断
6. 根据收到令牌的类型，控制器会产生下面的中断

a) 如果 USBFS_DOEPINT0.XferComp=0 并且 USBFS_DOEPINT0.SetUp=1,控制器对主机发送的第一个输出令牌回复 NAK。然后编程 USBFS_DOEPCTL0.CNAK=1 清除 NAK，然后跳转到第 4 步。

b) 如果 USBFS_DOEPINT0.XferComp=1 并且 USBFS_DOEPINT0.SetUp=0，读寄存器 USBFS_DOEPINT0 的 StupPktRcvd 位如果 StupPktRcvd=0，表示输出令牌已收到，如果这个令牌不是最后一个，就跳转到第 4 步接收下一个令牌。如果这是最后一个令牌，配置 USBFS_DOEPCTL0.Stall=1,这样控制器会暂停接收主机发送的输出令牌包。数据阶段已完成，程序将跳转到控制写传输的状态阶段。如果 StupPktRcvd=1，表示一个新的控制传输 Setup 令牌已收到。程序必须解码主机发送的控制传输类型。

c) 如果 USBFS_DOEPINT0.XferComp=1 并且 USBFS_DOEPINT0.StsPhseRcvd=1，中断标志位 USBFS_DOEPINT0.XferComp 表示控制器在数据阶段收到了输出令牌而中断标志位 USBFS_DOEPINT0.StsPhseRcvd 表示状态阶段开始。在处理完最后一个输出令牌后，程序进入控制写传输的状态阶段。

控制写传输状态阶段

1. 在系统内存里创建缓存列表包含一个输入缓存
2. 编程寄存器 USBFS_DIEPDMA0 来配置缓存列表的基地址
3. 编程寄存器 USBFS_DIEPTSIZ0 的 XferSize,PktCnt 字段
4. 编程寄存器 USBFS_DIEPCTL0.MPS 来配置控制端点最大数据包大小，编程寄存器 USBFS_DIEPCTL0.EPEna 置 1 使能控制输入端点

5. 在系统内存里创建缓存列表包含一个输出缓存
6. 编程寄存器 USBFS_DOEPDMA0 来配置缓存列表的基地址
7. 编程寄存器 USBFS_DOEPTSIZ0 的 XferSize,PktCnt 和 SUPCnt 字段
8. 编程寄存器 USBFS_DOEPCTL0.MPS 来配置控制端点最大数据包大小，编程寄存器 USBFS_DOEPCTL0.EPEna 置 1 使能控制输出端点
9. 等待控制端点输入和输出中断同时发生
10. 根据收到令牌的类型，控制器会产生下面的中断

    如果 USBFS_DIEPINT0.XferComp=1，只有输入端点产生中断，表示 DMA 已经处理输入描述符的零长度数据包并压栈到对应的发送 FIFO。当前控制传输的状态阶段已完成，程序可以处理下一个控制传输。

    如果 USBFS_DOEPINT0.StsPhseRcvd=1，只有输出端点产生中断，编程寄存器 USBFS_DIEPCTL0.CNAK=1，接受输入状态令牌，并跳转到第 8 步。

    如果 USBFS_DOEPINT0.XferComp=1，只有输出端点产生中断，表示主机提前发送一个新的 Setup 令牌。

    如果 USBFS_DOEPINT0.SetUp=0 并且 USBFS_DOEPINT0.StupPktRcvd=1，表示新的控制传输 Setup 令牌已经收到。程序必须解码主机发送的控制传输类型。然后程序必须进入下一个阶段。

    如果 USBFS_DOEPINT0.SetUp=1 且 USBFS_DOEPINT0.StupPktRcvd=1，表示新的控制传输 Setup 令牌已经收到。

    DOEPINT0.SetUp=1 表示主机可能发送以下令牌包：

    - 控制写传输数据阶段输出令牌包
    - 控制读传输数据阶段输入令牌包
    - 两阶段控制传输状态阶段输入令牌包

25.6.23.2 控制读传输编程流程

控制读传输 Setup 阶段

1. 在系统内存里创建一个缓存列表包含一个输出缓存
2. 编程寄存器 USBFS_DOEPDMA0 来配置缓存列表的基地址
3. 编程寄存器 USBFS_DOEPTSIZ0 的 XferSize,PktCnt 和 SUPCnt 字段
4. 编程寄存器 USBFS_DOEPCTL0.MPS 来配置控制端点最大数据包大小，编程寄存器 USBFS_DOEPCTL0.EPEna 置 1 使能控制输出端点
5. 等待输出端点中断
6. 当控制器收到 Setup 数据包时，会置位 USBFS_DOEPINT0 中断

a) 如果 USBFS_DOEPINT0.XferComp=1 并且 USBFS_DOEPINT0.SetUp=0，程序必须读 USBFS_DOEPINT0 寄存器的 StupPktRcvd 位。如果 StupPktRcvd=1，表示 Setup 令牌已经收

到。程序必须解码主机发送的控制传输类型。然后程序必须进入控制传输的数据阶段。

b) 如果 USBFS_DOEPINT0.XferComp=1 并且 USBFS_DOEPINT0.SetUp=1，程序必须读 USBFS_DOEPINT0 寄存器的 StupPktRcvd 位。如果 StupPktRcvd=1，表示 Setup 令牌已经收到。USBFS_DOEPINT0.SetUp=1 表示主机在控制读传输的数据阶段发送了输入令牌包。然后程序必须进入下一个阶段。

控制读传输数据和状态阶段

1. 创建 n 个缓存接收`wlength`字节数据确保每个缓存包含一个数据包 [n=wlength/MPS+wlengthMOD(MPS)]
2. 编程寄存器 USBFS_DIEPTSIZ0 的 XferSize,PktCnt 字段
3. 编程寄存器 USBFS_DIEPDMA0 来配置缓存列表的基地址
4. 编程寄存器 USBFS_DIEPCTL0.MPS 来配置控制端点最大数据包大小，编程控制输入端点寄存器 USBFS_DIEPCTL0.EPEna 置 1 使能控制输入端点
5. 编程寄存器 USBFS_DOEPTSIZ0 的 XferSize,PktCnt 和 SUPCnt 字段来接收一个提前的状态阶段或者一个新的 Setup 令牌包。
6. 编程寄存器 USBFS_DOEPDMA0 来配置缓存列表的基地址
7. 编程寄存器 USBFS_DOEPCTL0.MPS 来配置控制端点最大数据包大小，编程寄存器 USBFS_DOEPCTL0.EPEna 置 1 使能控制输出端点编程寄存器 USBFS_DOEPCTL0.CNAK=1 清除 NAK
8. 等待控制端点的输入输出中断。
9. 根据控制器接收到的令牌包的类型，会产生以下中断的组合：

   如果 USBFS_DOEPINT0.SetUp=1，并且只有输出端点产生中断，表示主机发送输入令牌开始这次传输的数据阶段。

   编程 USBFS_DIEPCTL0.CNAK=1 清除 NAK 并接收这次传输数据阶段的输入令牌包。编程 USBFS_DOEPCTL0.CNAK=1 清除 NAK 并接收这次传输的状态阶段的输出令牌包或者 Setup 阶段的 SETUP 令牌包。跳转到第 9 步。

   如果 USBFS_DIEPINT0.XferComp=1，并且只有输入端点产生中断，表示 DMA 已经处理输入缓存数据包并压栈到对应的发送 FIFO。当最后一个缓存的 XferCompl 标志位产生，编程 USBFS_DIEPCTL0.Stall=1 当发送 FIFO 里所有的数据已经发送给主机并且发送 FIFO 为空时暂停响应主机发送的额外的输入令牌包。跳转到第 9 步，等待输出端点中断。如果 USBFS_DOEPINT0.XferComp=1，并且只有输出端点产生中断，表示主机发送控制传输的状态阶段或者一个新的控制传输的 Setup 令牌包。

a) 如果 USBFS_DOEPINT0.XferComp=1 并且 USBFS_DOEPINT0.SetUp=0，程序必须读 USBFS_DOEPINT0 寄存器的 StupPktRcvd 位。如果 StupPktRcvd=1，表示 Setup 令牌已经收到。程序必须解码主机发送的控制传输类型。如果 StupPktRcvd=0，表示这次传输的状态阶段的输出令牌已经收到。程序必须为下一次控制传输准备一个缓存。

b) 如果 USBFS_DOEPINT0.XferComp=1 并且 USBFS_DOEPINT0.SetUp=1，程序必须读 USBFS_DOEPINT0 寄存器的 StupPktRcvd 位，如果 StupPktRcvd=1，表示 Setup 令牌已经收到。程序必须解码主机发送的控制传输类型。

25.6.23.3 两阶段控制传输

两阶段控制传输 Setup 阶段

1. 在系统内存里创建一个缓存列表包含一个输出缓存
2. 编程寄存器 USBFS_DOEPTSIZ0 的 XferSize,PktCnt 和 SUPCnt 字段
3. 编程寄存器 USBFS_DOEPDMA0 来配置缓存列表的基地址
4. 编程寄存器 USBFS_DOEPCTL0.MPS 来配置控制端点最大数据包大小，编程寄存器 USBFS_DOEPCTL0.EPEna 置 1 使能控制输出端点
5. 等待输出端点中断
6. 当控制器收到 Setup 令牌包时，会置位 USBFS_DOEPINT0 中断

a) 如果 USBFS_DOEPINT0.XferComp=1 并且 USBFS_DOEPINT0.SetUp=0，程序必须读 USBFS_DOEPINT0 寄存器的 StupPktRcvd 位。如果 StupPktRcvd=1，表示 Setup 令牌已经收到。程序必须解码主机发送的控制传输类型。根据 Wlength 字段，程序可以判断当前传输是否是一个两阶段控制传输。如果是两阶段控制传输，程序必须进入这次传输的状态阶段。

b) 如果 USBFS_DOEPINT0.XferComp=1 并且 USBFS_DOEPINT0.SetUp=1，程序必须读 USBFS_DOEPINT0 寄存器的 StupPktRcvd 位。如果 StupPktRcvd=1，表示 Setup 令牌已经收到。USBFS_DOEPINT0.SetUp=1 表示主机在这次控制读传输的数据阶段发送了输入令牌包。

两阶段控制传输状态阶段

1. 在系统内存里创建一个缓存列表包含一个输入缓存。
2. 编程寄存器 USBFS_DIEPTSIZ0 的 XferSize,PktCnt 字段。
3. 编程寄存器 USBFS_DIEPDMA0 来配置缓存列表的基地址。
4. 编程寄存器 USBFS_DIEPCTL0.MPS 来配置控制端点最大数据包大小，编程寄存器 USBFS_DIEPCTL0.EPEna 置 1 使能控制输出端点。
5. 编程输出端点，在系统内存里创建一个缓存列表包含一个输出缓存保证在状态阶段可以接收一个新的 Setup 令牌包。
6. 编程寄存器 USBFS_DOEPTSIZ0 的 XferSize,PktCnt 和 SUPCnt 字。
7. 编程寄存器 USBFS_DOEPDMA0 来配置缓存列表的基地址。
8. 编程寄存器 USBFS_DOEPCTL0.MPS 来配置控制端点最大数据包大小，编程寄存器 USBFS_DOEPCTL0.EPEna 置 1 使能控制输出端。
9. 等待控制端点的输入输出中断。
10. 根据控制器接收到的令牌包的类型，会产生以下中断的组合：

如果 USBFS_DOEPINT0.Setup=1，并且只有输出端点产生中断，编程 USBFS_DIEPCTL0.CNAK=1 清除 NAK 并接收这次传输状态阶段的输入令牌包。跳转到第 8 步。

如果 USBFS_DIEPINT0.XferComp=1，只有输入端点产生中断，表示 DMA 已经处理输入缓存的零长度数据包并压栈到对应的发送 FIFO。两阶段控制传输完成，程序可以处理新的控制传输。

如果 USBFS_DOEPINT0.XferComp=1，并且只有输出端点产生中断。

a) 如果 USBFS_DOEPINT0.XferComp=1 并且 USBFS_DOEPINT0.SetUp=0，程序必须读 USBFS_DOEPINT0 寄存器的 StupPktRcvd 位。如果 StupPktRcvd=1，表示新的控制传输的 Setup 令牌已经收到。程序必须解码主机发送的控制传输类型。程序必须处理新的控制传输。

b) 如果 USBFS_DOEPINT0.XferComp=1 并且 USBFS_DOEPINT0.SetUp=1，程序必须读 USBFS_DOEPINT0 寄存器的 StupPktRcvd 位。如果 StupPktRcvd=1，表示新的控制传输的 Setup 令牌已经收到。

USBFS_DOEPINT0.SetUp=1 表示主机可能发送以下令牌包：

- 控制写传输数据阶段输出令牌包
- 控制读传输数据阶段输入令牌包
- 两阶段控制传输状态阶段输入令牌包

25.6.24 输出数据传输

25.6.24.1 控制 Setup 传输

1. 编程寄存器 USBFS_DOEPTSIZ0
   USBFS_DOEPTSIZ0.SUPCnt = 3
2. 根据端点特性编程寄存器 USBFS_DOEPDMA0 和寄存器 USBFS_DOEPCTL0，并使能端点 USBFS_DOEPCTL0.EPEna = 1
3. 等待 USBFS_DOEPINT0.SETUP 中断标志位置位表示 SETUP 数据传输完成。
4. 程序必须读寄存器 USBFS_DOEPTSIZ0 判定已收到的 SETUP 数据包个数并处理最后收到的 SETUP 数据包。
5. 在 DMA 模式下，程序必须判定寄存器 USBFS_DOEPINT0.Back2BackSETup 是否置位。如果置位表示控制器收到超过 3 个背靠背 SETUP 数据包。在这种情况下，程序必须忽略 USBFS_DOEPTSIZ0.SUPCnt 值，而使用寄存器 USBFS_DOEPDMA0 直接读出最后一个收到的 SETUP 数据包。

25.6.25 非周期批量输入数据传输无阈值

1. 编程寄存器 USBFS_DIEPTSIZn 的 XferSize 和 PktCnt 字段，编程寄存器 USBFS_DIEPDMAn。
2. 根据端点特性编程寄存器 USBFS_DIEPCTLn，置位 CNAK 位和 EPEna 位。
3. 在 DMA 模式，编程 NextEp 字段以确保控制器按照正确的顺序抓取输入端点数据。
4. USBFS_DIEPINTn.TimeOut 中断标志位置位表示在这个端点检测到一个超时情况。处理方法请

参考章节非周期输入数据传输超时（共享 FIFO 模式）。

25.6.26 非周期批量输入数据传输有阈值

1. 编程寄存器 USBFS_DIEPTSIZn 的 XferSize 和 PktCnt 字段，编程寄存器 USBFS_DIEPDMAn。
2. 根据端点特性编程寄存器 USBFS_DIEPCTLn，置位 CNAK 位和 EPEna 位，编程 USBFS_DIEPCTLn.TXFNum 字段。
3. USBFS_DIEPINTn.XferCompl 中断标志位置位表示非周期输入传输完成。读取寄存器 USBFS_DIEPTSIZn 的 XferSize 和 PktCnt 字段，值都为 0，表示所有的数据已经传输完成。

25.6.27 非同步输出数据传输无阈值

1. 编程寄存器 USBFS_DOEPTSIZn 的 XferSize 和 PktCnt 字段，编程寄存器 USBFS_DOEPDMAn。
2. 根据端点特性编程寄存器 USBFS_DOEPCTLn，置位 CNAK 位和 EPEna 位。
3. USBFS_DOEPINTn.XferCompl 中断标志位置位表示非同步输出传输完成。读取寄存器 USBFS_DOEPTSIZn 来判定收到的数据包大小。

25.6.28 非同步输出数据传输有阈值

操作流程请参考章节非同步输出数据传输无阈值。

25.6.29 同步输出数据传输有阈值

操作流程请参考章节非同步输出数据传输无阈值。

25.6.30 未完成同步输出数据传输

1. 中断标志位 USBFS_GINTSTS.incompISOIN 置位表示当前帧至少有一个同步输出端点有一个未完成的传输。
2. 如果该端点未完成的传输是因为同步输出数据没有完全被清空，程序必须清空接收 FIFO 内所有同步输出数据。
3. 当接收 FIFO 内的所有数据被清空，程序可以检测到 USBFS_DOEPINTn.XferCompl 中断。在这种情况下，程序必须重新使能端点在下一帧接收同步输出数据。
4. 当中断标志位 USBFS_GINTSTS.incompISOIN 置位，程序必须读所有同步输出端点的控制寄存器(USBFS_DOEPCTLn)判定哪一个端点在当前帧产生了未完成传输。
5. 前面的步骤必须在 USBFS_GINTSTS.SOF 中断标志位置位前完成以保证当前帧数字没有改变。
6. 当同步输出端点发生了未完成传输，程序必须丢弃内存的数据并且禁止该端点。USBFS_DOEPCTLn.EPDis = 1。
7. 等待中断标志位 USBFS_DOEPINTn.EPDisbld 置位，然后重新使能端点接收下一帧的新数据。

25.6.31 周期输入数据传输无阈值

1. 编程寄存器 USBFS_DIEPTSIZn 的 XferSize 和 PktCnt 字段，编程寄存器 USBFS_DIEPDMAn。
2. 根据端点特性编程寄存器 USBFS_DIEPCTLn，置位 CNAK 位和 EPEna 位。
3. USBFS_DIEPINTn.INTknTXFEmp 中断标志位置位表示 DMA 还没有把所有的数据传输到发送 FIFO。
4. 如果中断端点早已使能，忽略第 3 步里的中断。如果同步端点早已使能，而发生第 3 步里的中断。
5. 当中断标志位 USBFS_DIEPINTn.XferCompl 置位，而 USBFS_DIEPINTn.INTknTXFEmp 中断标志位没有置位时，表示同步输入传输成功完成。读寄存器 USBFS_DIEPTSIZn 的 XferSIze 和 PktCnt 字段，值都为 0，表示所有的数据已经传输完成。
6. 当中断标志位 USBFS_DIEPINTn.XferCompl 置位，而 USBFS_DIEPINTn.INTknTXFEmp 中断标志位置位或者没有置位时，表示中断输入传输成功完成。读寄存器 USBFS_DIEPTSIZn 的 XferSIze 和 PktCnt 字段，值都为 0，表示所有的数据已经传输完成。
7. 中断标志位 USBFS_GINTSTS.incompISOIN 置位表示控制器在当前帧没有收到至少一个周期输入令牌包。

25.6.32 周期输入数据传输有阈值

1. 编程寄存器 USBFS_DIEPTSIZn 的 XferSIze 和 PktCnt 字段，编程寄存器 USBFS_DIEPDMAn。
2. 根据端点特性编程寄存器 USBFS_DIEPCTLn，置位 CNAK 位和 EPEna 位。同时在寄存器 USBFS_DIEPCTLn.TXFNum 字段指定发送 FIFO 数字。
3. USBFS_DIEPINTn.INTknTXFEmp 中断标志位置位表示 DMA 还没有把所有的数据传输到发送 FIFO。
4. 如果中断端点早已使能，忽略第 3 步里的中断。如果同步端点早已使能，而发生第 3 步里的中断。
5. 当中断标志位 USBFS_DIEPINTn.XferCompl 置位，而 USBFS_DIEPINTn.INTknTXFEmp 中断标志位没有置位时，表示同步输入传输成功完成。读寄存器 USBFS_DIEPTSIZn 的 XferSize 和 PktCnt 字段，值都为 0，表示所有的数据已经传输完成。
6. 当中断标志位 USBFS_DIEPINTn.XferCompl 置位，而 USBFS_DIEPINTn.INTknTXFEmp 中断标志位置位或者没有置位时，表示中断输入传输成功完成。读寄存器 USBFS_DIEPTSIZn 的 XferSize 和 PktCnt 字段，值都为 0，表示所有的数据已经传输完成。
7. 中断标志位 USBFS_GINTSTS.incompISOIN 置位表示控制器在当前帧没有收到至少一个周期输入令牌包。
8. 中断标志位 USBFS_DIEPINTn.TxFifoUndrn 置位表示当前帧的同步传输出现低流量情况，程序可以选择忽略这个中断，因为这个中断经常由于 USBFS_GINTSTS. incompISOIN 在周期帧的结尾引发。

25.7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：

- 模块全局寄存器
- 设备模式寄存器
- 电源和时钟门控控制寄存器

USBFS 模块寄存器列表以及基地址请参考表 25-3 USBFS 寄存器一览表。

USBFS 模块寄存器基地址：0x40040000

表 25-3 USBFS 寄存器一览表

偏移量	寄存器名称	访问	寄存器描述
0x08	USBFS_GAHBCFG	RW	USBFS AHB控制寄存器
0x0c	USBFS_GUSBCFG	RW	USBFS USB配置寄存器
0x10	USBFS_GRSTCTL	RW	USBFS复位寄存器
0x14	USBFS_GINTSTS	RW	USBFS模块中断寄存器
0x18	USBFS_GINTMSK	RW	USBFS中断屏蔽寄存器
0x1c	USBFS_GRXSTSR	RW	USBFS接收状态调试读取寄存器
0x20	USBFS_GRXSTSP	RW	USBFS接收状态读取和出栈寄存器
0x24	USBFS_GRXFSIZ	RW	USBFS接收FIFO大小寄存器
0x28	USBFS_GNPTXFSIZ	RW	USBFS非周期发送FIFO大小寄存器
0x2c	USBFS_GNPTXSTS	R	USBFS非周期发送FIFO/队列状态寄存器
0x3c	USBFS_CID	RW	USBFS模块ID寄存器
0x104	USBFS_DPTXFSIZ	RW	USBFS设备周期IN端点发送FIFO大小寄存器
0x800	USBFS_DCFG	RW	USBFS设备配置寄存器
0x804	USBFS_DCTL	RW	USBFS设备控制寄存器
0x808	USBFS_DSTS	R	USBFS设备状态寄存器
0x810	USBFS_DIEPMSK	RW	USBFS设备IN端点通用中断屏蔽寄存器
0x814	USBFS_DOEPMSK	RW	USBFS设备OUT端点通用中断屏蔽寄存器
0x818	USBFS_DAINT	R	USBFS设备全体端点中断寄存器
0x81c	USBFS_DAINTMSK	RW	USBFS设备全体端点中断屏蔽寄存器
0x820	USBFS_DTKNQR1	R	USBFS设备输入令牌顺序队列读寄存器1
0x824	USBFS_DTKNQR2	R	USBFS设备输入令牌顺序队列读寄存器2
0x830	USBFS_DTKNQR3	R	USBFS设备输入令牌顺序队列读寄存器3
0x834	USBFS_DTKNQR4	R	USBFS设备输入令牌顺序队列读寄存器4
0x900	USBFS_DIEPCTL0	RW	USBFS设备IN端点0控制寄存器

偏移量	寄存器名称	访问	寄存器描述
0x900+n*0x20(n=1,3,5,7)	USBFS_DIEPCTLn	RW	USBFS设备IN端点n控制寄存器
0x908+n*0x20(n=0,1,3,5,7)	USBFS_DIEPINTn	RW	USBFS设备IN端点n中断寄存器
0x910	USBFS_DIEPSIZ0	RW	USBFS设备IN端点0传输大小寄存器
0x910+n*0x20(n=1,3,5,7)	USBFS_DIEPSIZn	RW	USBFS设备IN端点n传输大小寄存器
0x914+n*0x20(n=1,3,5,7)	USBFS_DIEPDMAn	RW	USBFS设备IN端点n DMA地址寄存器
0xb00	USBFS_DOEPCTL0	RW	USBFS设备OUT端点0控制寄存器
0xb00+n*0x20(n=2,4,6,8)	USBFS_DOEPCTLn	RW	USBFS设备OUT端点n控制寄存器
0xb08+n*0x20(n=0,2,4,6,8)	USBFS_DOEPINTn	RW	USBFS设备OUT端点n中断寄存器
0xb10	USBFS_DOEPTSIZ0	RW	USBFS设备OUT端点0传输大小寄存器
0xb10+n*0x20(n=2,4,6,8)	USBFS_DOEPTSIZn	RW	USBFS设备OUT端点n传输大小寄存器
0xb14+n*0x20(n=2,4,6,8)	USBFS_DOEPDMAn	RW	USBFS设备OUT端点n DMA地址寄存器
0xe00	USBFS_GCCTL	RW	USBFS电源和门控时钟控制寄存器

25.7.1 USBFS 全局寄存器

25.7.1.1 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							Reserved	Reserved	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
b8	Reserved	-	必须保持复位值。	R
b7	Reserved	-	必须保持复位值。	R
b6	Reserved	-	必须保持复位值。	R
b5	DMAEN	DMA 使能	DMA 使能（DMA enable） 0：模块以从模式运行 1：模块以DMA模式运行	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 0011:INCR4 0101:INCR8 0111:INCR16 其它值：保留	R
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	R/W

25.7.1.2 USBFS USB 配置寄存器（USBFS_GUSBCFG)

USBFS USB configuration register

偏移地址：0x00C

复位值：0x0000 0A00

该寄存器可对模块进行配置。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															
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~b14	Reserved	-	必须保持复位值。	R
b13~b10	TRDT	USB周转时间	USB周转时间（USB turnaround time) 以PHY时钟数为单位设置周转时间。 要计算TRDT的值，请使用如下公式： TRDT = 4 × 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。	R/W
b9~b7	Reserved	-	必须保持复位值。	R
b6	PHYSEL	全速系列收发器选择	全速系列收发器选择（Full Speed serial transceiver select) 此位为只写位且始终为1。	W
b5~b3	Reserved	-	必须保持复位值。	R
b2~b0	TOCAL	FS 超时校准	FS 超时校准（FS timeout calibration) PHY引入的额外延迟包括应用程序在该字段中设置的PHY时钟数，以及模块的全速数据包间超时间隔。不同PHY引入的延迟对数据线状态的影响是不同的。全速操作的USB标准超时值为16到18（含）个位时间。应用程序必须根据枚举速度编程该字段。每个PHY时钟增加的位时间数为0.25个位时间。	

25.7.1.3 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	Reserved	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。	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。	R
b29~b11	Reserved	-	读出时为“0”,写入时写“0”	R
b10~b6	TXFNUM	TxFIFO编号	TxFIFO编号(TxFIFO number) 使用TxFIFO刷新位进行FIFO刷新的FIFO编号。只有在模块将TxFIFO刷新位清零后，方可更改此字段。 ● 00000： 刷新Tx FIFO 0 ● 00001： 刷新TXFIFO 1 ● 00010：刷新TXFIFO 2 … ● 00101：刷新TXFIFO 15 ● 10000：刷新所有的发送FIFO	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	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，	R/W

			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）。	
b3~b2	Reserved	-	必须保持复位值。	R
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。	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	R/W

			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。	

25.7.1.4 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

该寄存器用于借助系统级别的事件来中断应用程序。

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	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	DATAFSUSP	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	GOUTNAKEFF	GINAKEFF	Reserved	RXFLNE	SOF	Reserved	Reserved	Reserved

位	标记	位名	功能	读写
b31	WKUPINT	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） 当USB总线上检测到恢复信号时，将触发该中断通过软件对该位写1清零。	R/W
b30	VBUSVINT	VBUS有效中断	VBUSy有效中断（VBUS valid interrupt） 设备模式下，当检测到USBFS_VBUS管脚由低变高时，将触发该中断。 通过软件对该位写1清零。	R/W
b29~b23	Reserved	-	必须保持复位值。	R
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 – 重新使能端点	R/W

			– 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。	
b21	INCOMPISOOUT	未完成OUT同步传输	INCOMPISOOUT: 未完成OUT同步传输 (Incomplete isochronous OUT transfer) 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。	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。	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。	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。	R
b17~b16	Reserved	-	必须保持复位值。	R
b15	EOPF	周期性帧结束中断	周期性帧结束中断 (End of periodic frame interrupt) 指示当前帧已达到USBFS_DCFG 寄存器中周期性帧间隔字段（USBFS_DCFG 中的PFIVL位）所指定的周期。 通过软件对该位写1清零。	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。	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。	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。	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。	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。	R/W
b9~b8	Reserved	-	必须保持复位值。	R
b7	GONAKEFF	全局OUT NAK有效中断	全局OUT NAK有效中断(Global OUT NAK effective interrupt) 指示USBFS_DCTL寄存器中由应用程序设置的“将全局OUT NAK置 1”位（USBFS_DCTL中的SGONAK位）已在模块中生效。通过写入 USBFS_DCTL 寄存器中的“将全局 OUT NAK 清零”位（USBFS_DCTL中的CGONAK位），可将该位清零。	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。	R
b5	Reserved	-	必须保持复位值。	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。	R
b3	SOF	帧起始中断	帧起始中断 (Start of frame interrupt) 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	R/W

			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。	
b2~b0	Reserved	-	必须保持复位值。	R

25.7.1.5 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	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	DATAFSUSPM	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	GOUTNAKEFFM	GINAKEFFM	Reserved	RXFNEM	SOFM	Reserved	Reserved	Reserved

位	标记	位名	功能	读写
b31	WKUPIM	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	R/W
b30	VBUSVM	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	R/W
b29~b23	Reserved	-	必须保持复位值。	R
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	R/W
b21	INCOMPISOOUTM	未完成OUT同步传输中断屏蔽	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	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	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	R/W

			1：使能中断	
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	R/W
b17~b16	Reserved	-	必须保持复位值。	R
b15	EOPFM	周期性帧结束中断屏蔽	周期性帧结束中断屏蔽(End of periodic frame interrupt mask) 0：屏蔽中断 1：使能中断	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	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	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	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	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	R/W
b9~b8	Reserved	-	必须保持复位值。	R
b7	GONAKEFFM	全局OUT NAK有效中断屏蔽	全局OUT NAK有效中断屏蔽(Global OUT NAK effective interrupt mask) 0：屏蔽中断 1：使能中断	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	R/W
b5	Reserved	-	必须保持复位值。	R
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	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	R/W

b2~b0	Reserved	-	必须保持复位值。	R

25.7.1.6 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。

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。

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
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

25.7.1.7 USBFS 接收 FIFO 大小寄存器（USBFS_GRXFSIZ）

USBFS Receive FIFO size register

偏移地址：0x024

复位值：0x0000 0140

此应用程序可以对必须分配给 RxFIFO 的 RAM 大小进行编程。

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

位	标记	位名	功能	读写
b31~b11	Reserved	-	必须保持复位值。	R
b10~b0	RXFD	RxFIFO深度	RXFD：RxFIFO深度（RxFIFO depth） 以32位字为单位。 最小值为16 最大值为256 上电复位值为最大Rx数据FIFO深度。	R/W

25.7.1.8 USBFS 非周期发送 FIFO 大小寄存器 (USBFS_GNPTXFSIZ)

USBFS Non-Periodic Transmit FIFO size register

偏移地址：0x028

复位值：可配置

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
NPTxFDep[15:0]															
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~b16	NPTxFDep	TxFIFO深度	非周期发送FIFO大小设定 最小值16 最大值32768	R/W
b15~b0	Reserved	-	必须保持复位值。	R

25.7.1.9 USBFS 非周期发送 FIFO/队列状态寄存器（USBFS_GNPTXSTS）

USBFS Non-Periodic Transmit FIFO/Queue Status register

偏移地址：0x02c

复位值：可配置

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res.	NPTxQTop[6:0]							NPTxQSpcAvail[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
NPTxFSpcAvail															

位	标记	位名	功能	读写
b31	Reserved		必须保持复位值。	R
b30~b24	NPTxQTop	非周期发送请求队列	MAC正在处理的非周期发送请求队列	R
b23~b16	NPTxQSpcAvail	非周期发送请求队列可用空间	表示非周期发送请求队列可用空间值 0 ： 非周期发送请求队列满 1 ： 1个位置可用 2 ： 2个位置可用 N ： n个位置可用(n <= 8)	R
b15~b0	NPTxFSpcAvail	-非周期发送FIFO可用空间	表示非周期发送FIFO可用空间值 0 ： 非周期发送FIFO满 1 ： 1个字可用 2 ： 2个字可用 N ： n个字可用(n <= 1024)	R

25.7.1.10 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

25.7.1.11 USBFS 设备周期 IN 端点发送 FIFO 大小寄存器（USBFS_DPTXFSIZx）（x = 1..4）

USBFS Device Periodic Transmit FIFO size register

偏移地址：0x104+(x-1)*0x4

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
DPTxFSize[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DPTxFStAddr[11:0]															

位	标记	位名	功能	读写
b31~b16	DPTxFSize	周期IN端点TxFIFO大小	设备周期IN端点TxFIFO深度 (Device Period Transmit TxFIFO size) 以32位字为单位。 最小值为4 最大值为256	R
b15~b0	DPTxFStAddr	周期IN端点TxFIFOx RAM起始地址	周期IN端点TxFIFOx RAM起始地址 (Period IN endpoint FIFOx transmit RAM start address) 此字段包含周期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/R_W

25.7.2 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。

除非特别说明，否则寄存器描述中的位值以二进制表示。

25.7.2.1 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
b12~b11	PFIVL	周期性帧间隔	周期性帧间隔(Periodic frame interval) 指示一帧内必须使用周期性帧中断通知应用程序的时间点。此功能可用于确定该帧的所有同步通信是否完成。 00：80%帧间隔 01：85%帧间隔 10：90%帧间隔 11：95%帧间隔	R/W
b10~b4	DAD	设备地址	设备地址(Device address) 应用程序必须在执行每个SetAddress控制命令后根据命令参数对该字段进行设置。	R/W
b3	Reserved	-	必须保持复位值。	R
b2	NZLSOHSK	非零长度状态OUT握手信号	非零长度状态OUT握手信号 (Non-zero-length status OUT handshake) 在控制传输状态阶段的OUT事务期间，当模块收到非零长度数据包后，应用程序可以使用此字段选择要发送的握手信号。 1：收到非零长度状态OUT事务时，回复STALL握手信号，收到的OUT 数据包不发送给应用程序。 0：将收到的OUT数据包（零长度或非零长度）发送给应用程序，并基于设备端点控制寄存器中端点的NAK和STALL位回复握手信号。	R/W

b1~b0	DSPD	设备速度	设备速度（Device speed） 指示应用程序要求模块进行枚举所采用的速度，或应用程序支持的最大速度。但是，实际总线速度只有在完成chirp序列后才能确定，同时此速度基于与模块连接的USB主机的速度。 00：保留 01：保留 10：保留 11：全速（USB 2.0收发器时钟为48 MHz）	R/W

25.7.2.2 USBFS 设备控制寄存器(USBFS_DCTL)

USBFS Device control register

偏移地址：0x804

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				POPRGDNE	CGONAK	SGONAK	CGINAK	SGINAK	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
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 NA (Clear global OUT NAK) 对此位执行写操作会将全局OUT NAK清零。	W
b9	SGONAK	置位全局OUT NAK	置位全局OUT NAK(Set global OUT NAK) 对此位执行写操作会将全局OUT NAK置 1。 应用程序使用此位在所有OUT端点发送NAK握手信号。 应用程序只有确定模块中断寄存器中全局OUT NAK有效位（USBFS_GINTSTS中GONAKEFF位）已清零时，才可以将此位置 1。	W
b8	CGINAK	清零全局IN NAK	清零全局IN NAK (Clear global IN NAK) 对此位执行写操作会将全局 IN NAK 清零。	W
b7	SGINAK	置位全局IN NAK	置位全局IN NAK (Set global IN NAK) 对此字段执行写操作会将全局非周期性IN NAK置 1。应用程序使用此位使所有非周期性IN端点发送NAK握手信号。 应用程序只有确定模块中断寄存器中全局IN NAK有效位（USBFS_GINTSTS中GINAKEFF位）已清零时，才可以将此位置1。	W
b6~b4	Reserved	-	必须保持复位值。	R
b3	GONSTS	全局OUT NAK状态	全局OUT NAK状态(Global OUT NAK status) 0：将根据FIFO状态和NAK和STALL位设置发送握手信号。 1：无论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)	R

			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。	
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

25.7.2.3 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										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~b22	Reserved	-	必须保持复位值。	R
b21~b8	FNSOF	接收SOF的帧编号	接收SOF的帧编号（Frame number of the received SOF）	R
b7~b4	Reserved	-	必须保持复位值。	R
b3	EERR	不定错误	不定错误(Erratic error) 模块将该位置1以报告任何不定错误。 由于不定错误，USBFS控制器会进入挂起状态，并且会USBFS_GINTSTS寄存器的早期挂起位（USBFS_GINTSTS中的ESUSP 位）生成一个中断。如果早期挂起中断是由不定错误触发，则应用程序只能执行软断开以恢复通信。	R
b2~b1	ENUMSPD	枚举速度	枚举速度(Enumerated speed) 指示USBFS控制器通过chirp序列检测速度后被枚举成的速度。 01：保留 10：保留 11：全速（PHY 时钟运行频率为 48 MHz） 其它值：保留	R
b0	SUSPSTS	挂起状态	挂起状态(Suspend status) 在设备模式下，只要在USB上检测到挂起状态，该位就会置1。当 USB 总线上的空闲状态保持3ms，模块便会进入挂起状态。出现以下情况时，模块会退出挂起状态： – USB 数据线上有活动 – 应用程序对USBFS_DCTL寄存器的远程唤醒信号位（USBFS_DCTL中的RWUSIG位）执行写操作。	R

25.7.2.4 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									INEPNEM	INEPNMM	ITTXFEMSK	TOM	Reserved	EPDM	XFRCM

位	标记	位名	功能	读写
b31~b7	Reserved	-	必须保持复位值。	R
b6	INEPNEM	IN端点NAK有效中断屏蔽	IN端点NAK有效中断屏蔽（IN endpoint NAK effective mask) 0：屏蔽中断 1：使能中断	R/W
b5	INEPNMM	EP不匹配时接收到IN令牌中断屏蔽	EP不匹配时接收到IN令牌中断屏蔽 (IN token received with EP mismatch mask) 0：屏蔽中断 1：使能中断	R/W
b4	ITTXFEMSK	TxFIFO为空时接收到IN令牌中断屏蔽	TxFIFO为空时接收到IN令牌中断屏蔽 (IN token received when TxFIFO empty mask) 0：屏蔽中断 1：使能中断	R/W
b3	TOM	超时中断屏蔽（非同步端点）	超时中断屏蔽（非同步端点） (Timeout condition mask (Non-isochronous endpoints)) 0：屏蔽中断 1：使能中断	R/W
b2	Reserved	-	必须保持复位值。	R
b1	EPDM	端点禁止中断屏蔽	端点禁止中断屏蔽(Endpoint disabled interrupt mask) 0：屏蔽中断 1：使能中断	R/W
b0	XFRCM	传输完成中断屏蔽	传输完成中断屏蔽(Transfer completed interrupt mask) 0：屏蔽中断 1：使能中断	R/W

25.7.2.5 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
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
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

25.7.2.6 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4对应位20。	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4对应位4。	R/W

25.7.2.7 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端点中断平局位(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4对应位20。 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4对应位4。 0：屏蔽中断 1：使能中断	R/W

25.7.2.8 USBFS 设备 IN 指令顺序学习队列读寄存器 1(USBFS_DTKNQR1)

USBFS Device IN Token Sequence Learning Queue Read register 1

偏移地址：0x820

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EPTkn[23:8]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EPTkn[7:0]								WrapBit	Reserved		INTknWPtr[4:0]				

位	标记	位名	功能	读写
b31~b8	EPTkn	端点指令	每4位表示指令的端点值 [31:28] 指令5的端点值 [27:24] 指令4的端点值 [23:20] 指令3的端点值 [19:16] 指令2的端点值 [15:12] 指令1的端点值 [11:8] 指令0的端点值	R
b7	WrapBit	缠绕位	这位置1表示写指针缠绕。这位会在学习队列清除的时候被清零。	R
b6~b5	Reserved		必须保持复位值。	R
b4~b0	INTknWPtr	IN指令队列写指针	IN指令队列写指针	R

25.7.2.9 USBFS 设备 IN 指令顺序学习队列读寄存器 2(USBFS_DTKNQR2)

USBFS Device IN Token Sequence Learning Queue Read register 2

偏移地址：0x824

复位值：0x0000 0000

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

位	标记	位名	功能	读写
b31~b0	EPTkn	端点指令	每4位表示指令的端点值 [31:28] 指令13的端点值 [27:24] 指令12的端点值 [23:20] 指令11的端点值 [19:16] 指令10的端点值 [15:12] 指令9的端点值 [11:8] 指令8的端点值 [7:4] 指令7的端点值 [3:0] 指令6的端点值	R

25.7.2.10 USBFS 设备 IN 指令顺序学习队列读寄存器 3(USBFS_DTKNQR3)

USBFS Device IN Token Sequence Learning Queue Read register 3

偏移地址：0x830

复位值：0x0000 0000

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

位	标记	位名	功能	读写
b31~b0	EPTkn	端点指令	每4位表示指令的端点值 [31:28] 指令21的端点值 [27:24] 指令20的端点值 [23:20] 指令19的端点值 [19:16] 指令18的端点值 [15:12] 指令17的端点值 [11:8] 指令16的端点值 [7:4] 指令15的端点值 [3:0] 指令14的端点值	R

25.7.2.11 USBFS 设备 IN 指令顺序学习队列读寄存器 4(USBFS_DTKNQR4)

USBFS Device IN Token Sequence Learning Queue Read register 4

偏移地址：0x834

复位值：0x0000 0000

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

位	标记	位名	功能	读写
b31~b0	EPTkn	端点指令	每4位表示指令的端点值 [31:28] 指令29的端点值 [27:24] 指令28的端点值 [23:20] 指令27的端点值 [19:16] 指令26的端点值 [15:12] 指令25的端点值 [11:8] 指令24的端点值 [7:4] 指令23的端点值 [3:0] 指令22的端点值	R

25.7.2.12 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： – 端点禁止 – 传输完成	R/W
b30	EPDIS	端点禁止	端点禁止(Endpoint disable) 即使在该端点上的传输完成之前，应用程序也可将此位置1，以停止端点上的数据发送。应用程序必须等到发生端点禁止中断后，才能将端点视为禁止端点。在端点禁止中断位置1前，模块会将此位清零。只有在该端点的端点使能位置1后，应用程序才可将该位置1。	R/W
b29~b28	Reserved	-	必须保持复位值。	R
b27	SNAK	置位NAK位	置位NAK位(Set NAK) 对此位进行写操作会将端点的NAK位置1。 通过此位，应用程序可以控制端点上NAK握手信号的发送。模块也可在端点接收到SETUP数据包后将该端点的此位置1。	R/W
b26	CNAK	清零NAK位	清零NAK位(Clear NAK) 对此位进行写操作会将端点的NAK位清零。	R/W
b25~b22	TXFNUM	TxFIFO编号	TxFIFO编号(TxFIFO number) 该值设置为分配给IN端点0的FIFO编号。	R/W
b21	STALL	STALL握手	STALL握手(STALL handshake) 应用程序只能将此位置1，端点接收到SETUP令牌时，模块会将此位清零。如果NAK位、全局IN NAK或全局OUT NAK与此位均置 1，则STALL位优先。	R/W
b20	Reserved	-	必须保持复位值。	R

b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 硬件设置为‘00’，表示控制类型的端点。	R
b17	NAKSTS	NAK状态	NAK状态(NAK status) 指示以下结果： 0：模块根据FIFO状态回复非NAK握手。 1：模块在此端点上回复NAK握手。 当此位置1时（无论是被应用程序还是被模块），即使TxFIFO中仍有数据可用，模块也会停止发送数据。无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R
b16	Reserved	-	必须保持复位值。	R
b15	USBAEP	USB活动端点	USB活动端点(USB active endpoint) 此位总是置1，指示在所有配置和接口中控制端点0始终处于激活状态。	R
b14~b2	Reserved	-	必须保持复位值。	R
b1~b0	MPSIZ	最大数据包大小	最大数据包大小(Maximum packet size) 应用程序必须将此字段编程为当前逻辑端点的最大数据包大小。 00：64字节 01：32字节 10：16字节 11：8字节	R/W

25.7.2.13 USBFS 设备 IN 端点 x 控制寄存器(USBFS_DIEPCTLx)(x=1..4)

USBFS Device IN endpoint x control register

偏移地址：0x900 + (端点编号 × 0x20)

复位值：0x0000 0080

应用程序使用此寄存器控制各个逻辑端点（端点 0 除外）的行为。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM[3:0]				STALL	Reserved	EPTYP[1:0]		NAKSTS	EONUM/DPID
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
USBAEP	Reserved					MPSIZ[10:0]									

位	标记	位名	功能	读写
b31	EPENA	端点使能	端点使能(Endpoint enable) 应用程序将此位置1以在端点上启动数据发送。 在此端点上触发以下任一中断之前，模块会将此位清零： – SETUP阶段完成 – 端点禁止 – 传输完成	R/W
b30	EPDIS	端点禁止	端点禁止(Endpoint disable) 即使在该端点上的传输完成之前，应用程序也可将此位置1，以停止端点上的数据发送。应用程序必须等到发生端点禁止中断后，才能将端点视为禁止端点。在端点禁止中断位置1前，模块会将此位清零。只有在该端点的端点使能位置1后，应用程序才可将该位置1。	R/W
b29	SODDFRM	设置奇数帧	设置奇数帧(Set odd frame) 仅适用于同步IN和OUT端点。 对此字段进行写操作会将偶数/奇数帧（EONUM）字段设置为奇数帧。	R/W
b28	SD0PID/SEVNFRM	设置DATA0 PID/SEVNFRM	设置DATA0 PID (Set DATA0 PID) 仅适用于中断/批量IN 端点。 对此字段进行写操作会将此寄存器中的端点数据PID(DPID)字段设置为 DATA0。 SEVNFRM：设置偶数帧（Set even frame) 仅适用于同步IN端点。 对此字段进行写操作会将偶数/奇数帧(EONUM)字段设置为偶数帧。	R/W

b27	SNAK	置位NAK位	置位NAK位(Set NAK) 对此位进行写操作会将端点的NAK位置 1。 通过此位，应用程序可以控制端点上NAK握手信号的发送。发生传输完成中断时或端点上 接收到SETUP后，模块也可以将OUT端点的这个位置 1	R/W
b26	CNAK	清零NAK位	清零NAK位(Clear NAK) 对此位进行写操作会将端点的NAK位清零。	R/W
b25~b22	TXFNUM	TxFIFO编号	TxFIFO编号(TxFIFO number) 这些位用于指定与此端点相关联的 FIFO 编号。必须为每个有效的IN端点设置单独的 FIFO编号。 此字段仅针对IN端点有效。	R/W
b21	STALL	STALL握手	STALL握手(STALL handshake) 应用程序将此位置1使得设备对来自 USB 主机的所有令牌都回复STALL。如果NAK位、全局IN NAK或全局OUT NAK与此位同时置1，则STALL位优先。只有应用程序能够将此位清零，而模块则不能。	R/W
b20	Reserved	-	必须保持复位值。	R
b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 以下是这个逻辑端点支持的传输类型。 00：控制 01：同步 10：批量 11：中断	R
b17	NAKSTS	NAK状态	NAK状态(NAK status) 指示以下结果： 0：模块根据FIFO状态回复非NAK握手。 1：模块在此端点上回复NAK握手。 当应用程序或模块将此位置1时： 对于非同步IN端点：即使TxFIFO中存在可用数据，模块也会停止通过 IN端点发送任何数据。 对于同步IN端点：即使TxFIFO中存在可用数据，模块也会发送长度为零的数据包。 无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R
b16	EONUM/ DPID	偶数/奇数帧/ 端点数据PID	偶数/奇数帧 (Even/odd frame) 仅适用于同步IN端点。 指示模块为此端点发送/接收同步的数据所在的帧的编号。应用程序必须通过此寄存器中的SEVNFRM和SODDFRM字段对偶数/奇数帧编号进行编程，以便此端点发送/接收同步数据。 0：偶数帧 1：奇数帧	R/W

			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	
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。	

25.7.2.14 USBFS 设备 IN 端点 x 中断寄存器（USBFS_DIEPINTx) (x=0..4)

USBFS Device IN endpoint x interrupt register

偏移地址：0x908 + (端点编号 × 0x20)

复位值：0x0000 0000

此寄存器指示端点在出现 USB 和 AHB 相关事件时的状态。当模块中断寄存器中的 IN 端点中断位（USBFS_GINTSTS 中的 IEPINT 位）置 1 时，应用程序必须读取此寄存器。在应用程序能够读取此寄存器之前，必须先读取设备全体端点中断(USBFS_DAINT)寄存器，以获取设备端点 x 中断寄存器的准确端点编号。应用程序必须将此寄存器中的相应位清零，才能将 USBFS_DAINT 和 USBFS_GINTSTS 寄存器中的对应位清零。

b31	b30	b29	b28	b27	b26	b25	b24	b23	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	NEPNE	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
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
b4	ITTXFE	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。	
b3	TO	超时	超时条件(Timeout condition)	R/W

			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。	
b2	Reserved	-	必须保持复位值。	R
b1	EPDISD	端点禁止中断	端点禁止中断（Endpoint disabled interrupt） 此位指示该端点已经由应用程序禁止掉。 通过软件写1清零。	R/W
b0	XFRC	传输完成中断	传输完成中断（Transfer completed interrupt） 此字段指示在此端点上设置的传输已经在USB和AHB上传输完成。 通过软件写1清零。	R/W

25.7.2.15 USBFS 设备 IN 端点 0 传输大小寄存器(USBFS_DIEPTSIZ0)

USBFS Device IN endpoint 0 transfer size register

偏移地址：0x910

复位值：0x0000 0000

在使能端点 0 之前，应用程序必须修改此寄存器。通过设备控制端点 0 控制寄存器中的端点使能位（USBFS_DIEPCTL0 中的 EPENA）使能端点 0 后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

非零端点使用端点 1~5 的寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved											PKTCNT[1:0]		Reserved		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved									XFRSIZ[6:0]						

位	标记	位名	功能	读写
b31~b21	Reserved	-	必须保持复位值。	R
b20~b19	PKTCNT	数据包计数	数据包计数（Packet count） 指示端点0的一次数据传输包含的数据包个数。 每次从TxFIFO读取数据包（最大大小或短数据包）时，此字段将递减。	R/W
b18~b7	Reserved	-	必须保持复位值。	R
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

25.7.2.16 USBFS 设备 IN 端点 x 传输大小寄存器(USBFS_DIEPTSIZx)(x=1..4)

USBFS Device IN endpoint x transfer size register

偏移地址：0x910 + (端点编号 × 0x20)

复位值：0x0000 0000

在使能该端点之前，应用程序必须修改此寄存器。通过 USBFS_DIEPCTLx 寄存器中的端点使能位（USBFS_DIEPCTLx 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved			PKTCNT[9:0]										XFRSIZ[18:16]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XFRSIZ[15:0]															

位	标记	位名	功能	读写
b31~b29	Reserved	-	必须保持复位值。	R
b28~b19	PKTCNT	数据包计数	数据包计数（Packet count） 指示该端点上的一次数据传输包含的数据包个数。 每次从TxFIFO读取数据包（最大大小或短数据包）时，此字段将递减。	R/W
b18~b0	XFRSIZ	传输大小	传输大小(Transfer size) 此字段包含当前端点的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。 每次向TxFIFO写入来自外部存储器的数据包时，模块会使此字段递减。	R/W

25.7.2.17 USBFS 设备 IN 端点 x DMA 地址寄存器（USBFS_DIEPDMAx)(x=0..4)

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

注意：

- 当 USBFS_GAHBCFG.DMAEN 为 1 时，该寄存器的值应小于 0x20002000（RAM）/0x00010000（FLASH）。

25.7.2.18 USBFS 设备 IN 端点发送 FIFO 状态寄存器（USBFS_DTXFSTSx)(x=0..4)

USBFS Device IN endpoint transmit FIFO status register

偏移地址：0x918 + (端点编号 × 0x20)

复位值：0x0000 0000

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

位	标记	位名	功能	读写
b31~b16	Reserved	-	必须保持复位值。	R
b15~b0	INEPTFSAV	IN端点TxFIFO可用空间	IN端点TxFIFO可用空间（IN endpoint TxFIFO space available） 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

25.7.2.19 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) 应用程序无法禁止控制OUT端点0。	R
b29~b28	Reserved	-	必须保持复位值。	R
b27	SNAK	置位NAK位	置位NAK位(Set NAK) 对此位进行写操作会将端点的NAK位置1。 通过此位，应用程序可以控制端点上NAK握手信号的发送。模块也可在端点接收到SETUP数据包后将该端点的此位置1。	R/W
b26	CNAK	清零NAK位	清零NAK位(Clear NAK) 对此位进行写操作会将端点的NAK位清零。	R/W
b25~b22	Reserved	-	必须保持复位值。	R
b21	STALL	STALL握手	STALL握手(STALL handshake) 此端点接收到SETUP令牌时，应用程序只能将此位置1，而模块会将其清零。如果NAK位、全局OUT NAK与此位同时置1，则STALL位优先。无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R/W
b20	SNPM	监听模式	监听模式 (Snoop mode) 此位用于将端点配置为监听模式。在监听模式下，模块不会在将OUT数据包传输到应用存储区前检查其是否正确。	R/W
b19~b18	EPTYP	端点类型	端点类型(Endpoint type)	R/W

			硬件设置为‘00’，表示控制类型的端点。	
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
b15	USBAEP	USB活动端点	USB活动端点(USB active endpoint) 此位总是置1，指示在所有配置和接口中控制端点0始终处于激活状态。	R
b14~b2	Reserved	-	必须保持复位值。	R
b1~b0	MPSIZ	最大数据包大小	最大数据包大小(Maximum packet size) 控制OUT端点0的最大数据包大小与在控制IN端点0中进行编程的值相同。 00：64 字节 01：32 字节 10：16 字节 11：8 字节	R/W

25.7.2.20 USBFS 设备 OUT 端点 x 控制寄存器(USBFS_DOEPCTLx)(x=1..4)

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)	R/W

			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。	
b26	CNAK	清零NAK位	清零NAK位(Clear NAK) 对此位进行写操作会将端点的NAK位清零。	R/W
b25~b22	Reserved	-	必须保持复位值。	R
b21	STALL	STALL握手	STALL握手(STALL handshake) 此端点接收到SETUP令牌时，应用程序只能将此位置1，而模块会将其清零。如果NAK位、全局OUT NAK与此位同时置1，则STALL位优先。只有应用程序能够将此位清零，而模块则不能。	R/W
b20	SNPM	监听模式	监听模式（Snoop mode) 此位用于将端点配置为监听模式。在监听模式下，模块不会再检查接收数据的正确性。	R/W
b19~b18	EPTYP	端点类型	端点类型(Endpoint type) 以下是这个逻辑端点支持的传输类型。 00：控制 01：同步 10：批量 11：中断	R/W
b17	NAKSTS	NAK状态	NAK状态（NAK status) 指示以下结果： 0：模块根据FIFO状态回复非NAK握手。 1：模块在此端点上回复NAK握手。 当应用程序或模块将此位置1时： 即使RxFIFO存在空间可容纳传入数据包，模块也会停止在OUT端点上接收任何数据。 无论此位如何设置，模块总是通过ACK握手响应SETUP数据包。	R
b16	EONUM/ DPID	偶数/奇数帧/ 端点数据PID	偶数/奇数帧（Even/odd frame) 仅适用于同步OUT 端点。 指示模块为此端点发送/接收同步的数据所在的帧的编号。应用程序必须通过此寄存器中的SEVNFRM和SODDFRM字段对偶数/奇数帧编号进行编程，以便此端点发送/接收同步 数据。 0：偶数帧 1：奇数帧 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	R/W

			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	
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。	

25.7.2.21 USBFS 设备 OUT 端点 x 中断寄存器（USBFS_DOEPINTx) (x=0..4)

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
b6	B2BSTUP	接收到连续的SETUP数据包	接收到连续的SETUP数据包 (Back-to-back SETUP packets received) 仅适用于控制OUT端点。此位指示该端点已接收到三个以上的连续SETUP数据包。软件写1也可将此位清零。	R/W
b5	Reserved	-	必须保持复位值。	R
b4	OTEPDIS	端点禁止时接收到OUT令牌	端点禁止时接收到OUT令牌 (OUT token received when endpoint disabled) 仅适用于控制OUT端点。指示在尚未使能端点时接收到OUT令牌，从而产生中断。通过软件写1清零。	R/W
b3	STUP	SETUP阶段完成	SETUP 阶段完成 (SETUP phase done) 仅适用于控制OUT端点。指示控制端点的SETUP阶段已完成，当前控制传输中不再接收到连续的 SETUP数据包。在此中断上，应用程序可以对接收到的SETUP数据包进行解码。 通过软件写1清零。	
b2	Reserved	-	必须保持复位值。	R
b1	EPDISD	端点禁止中断	端点禁止中断 (Endpoint disabled interrupt) 此位指示该端点已经由应用程序禁止掉。 通过软件写1清零。	R/W
b0	XFRC	传输完成中断	传输完成中断 (Transfer completed interrupt) 此字段指示在此端点上设置的传输已经在USB和AHB上传输完成。 通过软件写1清零。	R/W

25.7.2.22 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 1~4 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
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	
b28~b20	Reserved	-	必须保持复位值。	R
b19	PKTCNT	数据包计数	数据包计数（Packet count） 一次传输中应该接收到数据包数量。 在端点使能前，软件设置该位，在传输开始后，每当数据包接收到后，该域数值自动减少。	R/W
b18~b7	Reserved	-	必须保持复位值。	R
b6~b0	XFRSIZ	传输大小	传输大小(Transfer size) 指示端点0的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。 每次从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

25.7.2.23 USBFS 设备 OUT 端点 x 传输大小寄存器(USBFS_DOEPTSIZx)(x=1..4)

USBFS Device OUT endpoint x transfer size register

偏移地址：0xB10 + (端点编号 × 0x20)

复位值：0x0000 0000

在使能该端点之前，应用程序必须修改此寄存器。通过 USBFS_DOEPCTLx 寄存器中的端点使能位（USBFS_DOEPCTLx 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved			PKTCNT[9:0]										XFRSIZ[18:16]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XFRSIZ[15:0]															

位	标记	位名	功能	读写
b31~b29	Reserved	-	必须保持复位值。	R
b28~b19	PKTCNT	数据包计数	数据包计数（Packet count） 指示该端点上的一次数据传输包含的数据包个数。 每次向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

25.7.2.24 USBFS 设备 OUT 端点 x DMA 地址寄存器（USBFS_DOEPDMAx)(x=0..4)

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

注意：

- 当 USBFS_GAHBCFG.DMAEN 为 1 时，该寄存器的值应小于 0x20002000（RAM）/0x00010000（FLASH）。

25.7.3 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。

25.7.3.1 USBFS 时钟门控控制寄存器(USBFS_GCCTL)

偏移地址：0xE00

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														GATEHCLK	STPPCLK

位	标记	位名	功能	读写
b31~b2	Reserved	-	必须保持复位值。	R
b1	GATEHCLK	门控HCLK	门控HCLK (Gate HCLK) 当USB通信挂起或会话无效时，应用程序会将此位置 1，以停止对除AHB总线从接口、主接口和唤醒逻辑之外的模块提供时钟。当USB恢复通信或新会话启动时，应用程序将此位清零。	R/W
b0	STPPCLK	停止PHY时钟	停止PHY时钟(Stop PHY clock) 当USB通信挂起、会话无效或设备断开连接时，应用程序将此位置1以停止PH 时钟。当USB恢复通信时，应用程序将此位清零。	R/W

26 控制器局域网（CAN）

26.1简介

CAN（Controller Area Network）总线是一种可以在无主机情况下实现微处理器或者设备之间相互通信的总线标准。本模块遵循 CAN 总线协议 2.0A 和 2.0B 协议并向上兼容 CAN-FD。CAN 总线控制器可以处理总线上的数据收发，在本产品中，CAN 具有 8 组筛选器。筛选器用于为应用程序选择要接收的消息。

应用程序通过 1 个高优先级的主发送缓冲器（Primary Transmit Buffer，以下简称 PTB）和 4 个辅发送缓冲器（Secondary Transmit Buffer，以下简称 STB）将发送数据送至总线，由发送调度器决定邮箱发送顺序。通过 10 个接收缓冲器（Receive Buffer，以下简称 RB）获取总线数据。4 个 STB 以及 10 个 RB 可以理解为一个 4 级 FIFO 和一个 10 级 FIFO，FIFO 完全由硬件控制。

CAN 总线控制器同时也可以支持时间触发 CAN 通信（Time-trigger communication）。

CAN 主要特性：

- 完全支持 CAN2.0A/CAN2.0B 协议。
- 向上兼容 CAN-FD 协议。
- 支持最高通信波特率 1Mbit/s
- 支持 1~1/256 的波特率预分频，灵活配置波特率。
- 10 个接收缓冲器
  - FIFO 方式
  - 错误或者不被接收的数据不会覆盖存储的消息
- 1 个高优先主发送缓冲器 PTB
- 4 个副发送缓冲器 STB
  - FIFO 方式
  - 优先级仲裁方式
- 8 组独立的筛选器
  - 支持 11 位标准 ID 和 29 位扩展 ID
  - 可编程 ID CODE 位以及 MASK 位
- PTB/STB 均支持支持单次发送模式
- 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

26.2CAN 系统框图

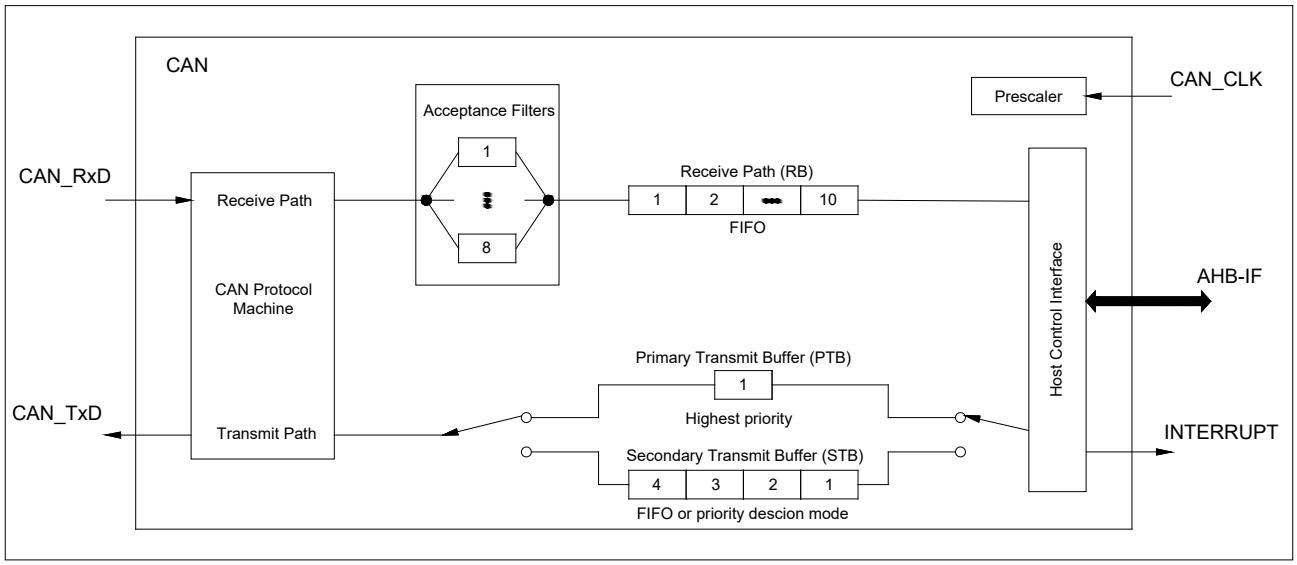


图 26-1 CAN 系统框图

26.3管脚说明

表 26-1 CAN 管脚说明

管脚名	方向	功能描述
CAN_RxD	输入	CAN接收数据信号
CAN_TxD	输出	CAN发送数据信号

26.4功能说明

本章将对 CAN 功能详细说明。

26.4.1 波特率设定

CAN 通信使用时钟 can_clk 的时钟源为外部高速振荡器，使用 CAN 模块之前，需要在 CMU 章节设定 CAN 通信时钟。

下图给出 CAN 位时间定义图，虚线上部分为 CAN 协议规定的位时间，虚线下部分为本 CAN 控制器 CAN-CTRL 定义的位时间。其中 segment1 和 segment2 可以通过寄存器 BT 设定。BT 寄存器只能在 CFG_STAT.RESET=1 即 CAN 软件复位时设定。

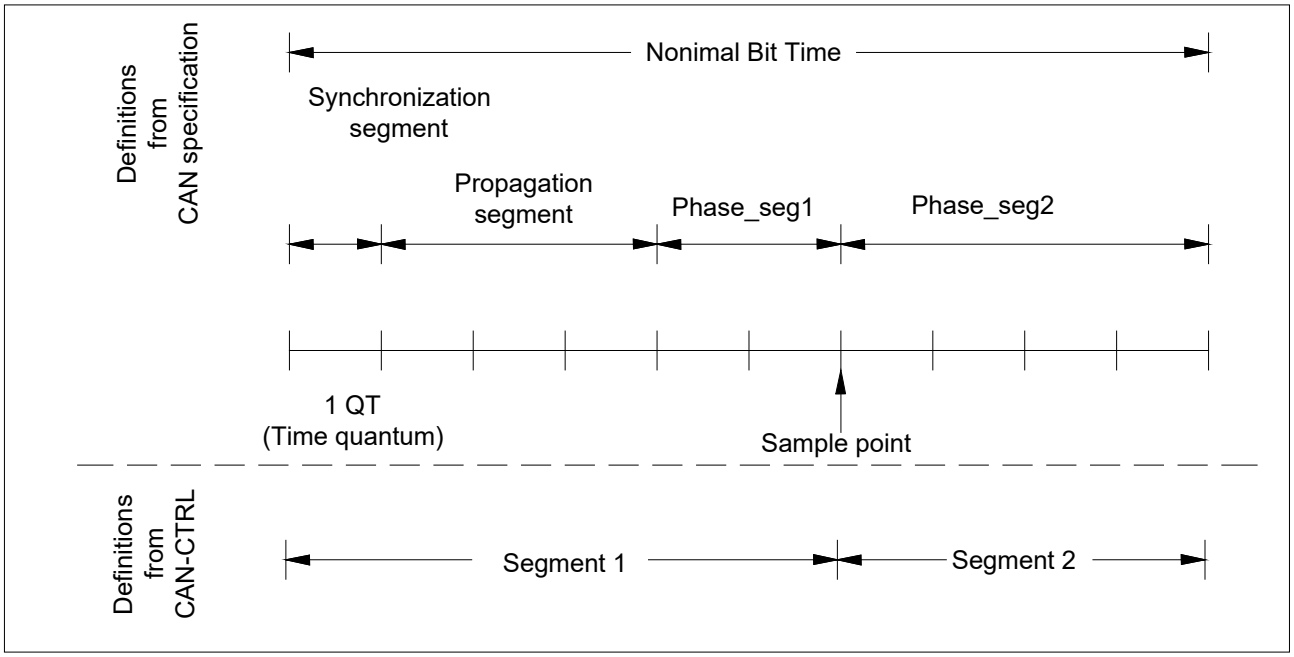


图 26-2 CAN 位时间定义图

TQ 计算方法请参考以下公式，其中 PRESC 通过 BT 寄存器的 PRESC 位设定。f_{can_clk}为 CAN 通信时钟频率。

$$TQ=\frac{PRESC+1}{f_{can\_clk}}$$

采样点的计算方法请参考下列说明：

1）若 BT 寄存器的 PRESC 位设定≥1 时，参考图 30-2 的 sample point 所示，采样点位于 segment1 和 segment2 的分界点上；

2）若 BT 寄存器的 PRESC 位设定为 0 时，采样点在 sample point，也即 segment1 和 segment2 分界点前面 2 个 TQ 的位置。

建议将 BT 寄存器的 PRESC 位设定值≥1。

位时间计算方法请参考以下公式，其中 SEG_1 和 SEG_2 通过 BT 寄存器的 SEG_1 位和 SEG_2 位设定。

$$BT=t_{SEG1}+t_{SEG2}=\left((SEG\_1+2)+(SEG\_2+1)\right)\times TQ$$

表 26-2 CAN 位时间设定规则

位	设定范围	规则
BT寄存器的SEG_1位	0~63	SEG_1 ≥ SEG_2 + 1 SEG_2 ≥ SJW
BT寄存器的SEG_2位	0~7	
BT寄存器的SJW位	0~7	

26.4.2 发送缓冲器

CAN_CTRL 提供两种发送缓冲器用于发送数据，主发送数据缓冲器 PTB 和副发送缓冲器 STB。PTB 具有最高的优先级，但只能缓冲一帧数据。STB 优先级比 PTB 低，但可以缓冲 4 帧数据，且 STB 内 4 帧数据可以工作在 FIFO 模式或者优先级仲裁模式。

STB 中的 4 帧说数据可以通过 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 中的一帧数据需占用 4 个字，可以通过 TBUF 寄存器进行访问。通过 TCMD 寄存器的 TBSEL 位选择 PTB 或者 STB，TBSEL=0，选择 PTB，TBSEL=1，选择 STB。通过 TCTRL 寄存器的 TSNEXT 位选择 STB 中的下一个 SLOT。对应关系如下图所示：

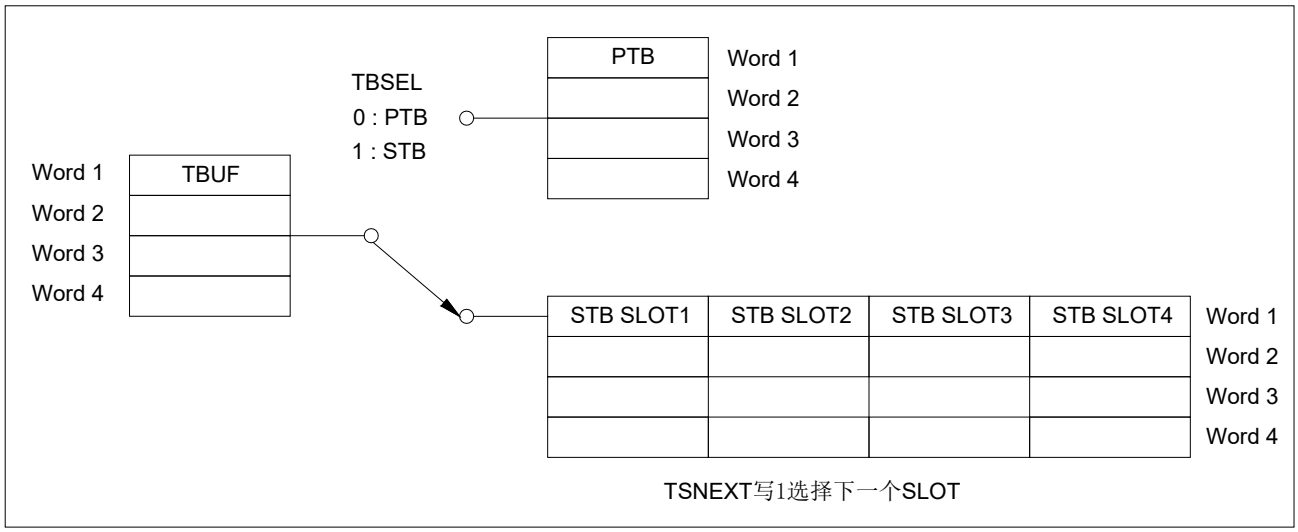


图 26-3 CAN TBUF 寄存器写发送缓冲器和示意图

26.4.3 接收缓冲器

CAN_CTRL 提供 10 个 SLOT 的接收缓冲器用于存储接收到的数据，该 10SLOT 的接收缓冲器工作在 FIFO 模式。每个 RB SLOT 需占用 4 个字，通过 RBUF 寄存器来读取接收到的数据，总是最先读取最早接收到数据，并通过 RCTRL 寄存器的 RREL 设置为 1 释放已经读取的 RB SLOT，并指向下一个 RB SLOT。

通过 RBUF 读取 RB SLOT 示意图如下。



图 26-4 CAN RBUF 寄存器读接收缓冲器示意图

26.4.4 接收筛选寄存器组

CAN_CTRL 提供 8 组 32 位筛选器用于过滤接收到的数据从而降低 CPU 负荷，筛选器可以支持标准格式 11 位 ID 或者扩展格式 29 位 ID。每组筛选器有一个 32 位 ID CODE 寄存器和一个 32 位 ID MASK 寄存器，ID CODE 寄存器用于比较接收到 CAN ID，而 ID MASK 寄存器用于选择比较的 CAN ID 位。对应的 ID MASK 位为 1 时，不比较该位的 ID CODE。

接收到的数据只要通过 8 组筛选器的任意一组，则被接收，接收到的数据存储在 RB 中，否则数据不被接收，也不被存储。

每组筛选器通过 ACFEN 寄存器使能或者禁止。ID CODE 和 ID MASK 通过 ACFCTRL 寄存器的 SELMASK 位设定，SELMASK=0 时，指向 ID CODE，SELMASK=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。

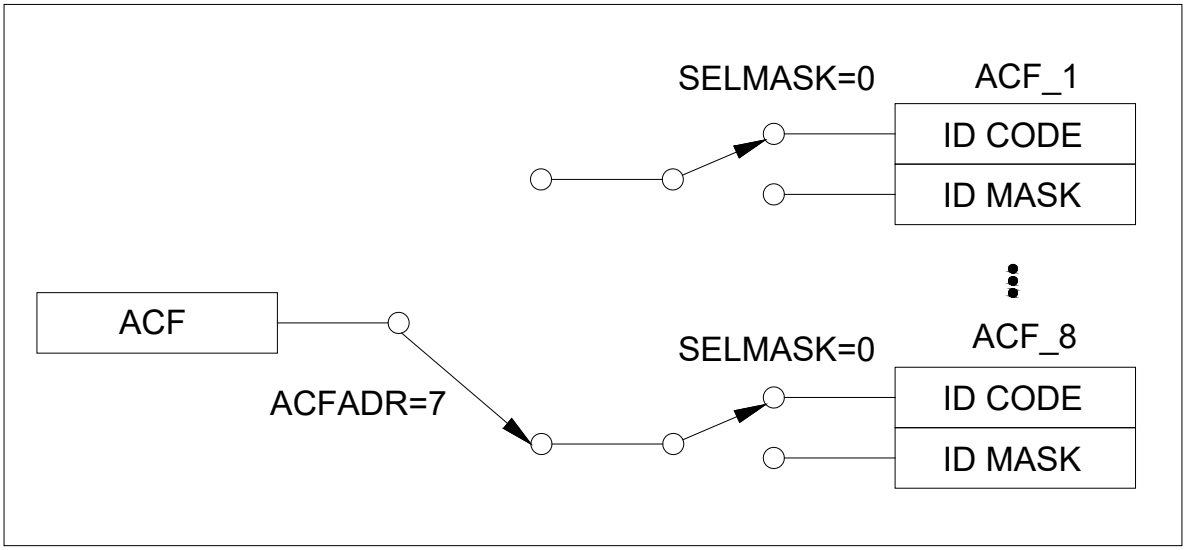


图 26-5 CAN ACF 寄存器访问筛选器组示意图

26.4.5 数据发送

在开始发送前必须保证 PTB 或者 STB 中至少有一帧数据已被装载，PTB 发送过程中 TPE 被锁定，STB 的填充情况可以通过 TSSTAT 位确认。发送数据设定步骤如下：

1. 设定 TBSEL 从 PTB 和 STB 中选择发送 BUF
2. 通过 TBUF 寄存器写需要发送的数据。
3. 如果选择的时 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

26.4.6 单次数据发送

不需要使用自动重新发送功能时，可以通过寄存器设定为单次发送模式，CFG_STAT 寄存器的 TPSS 位用于 PTB 的单次发送模式设定，TSSS 位用于 STB 的单次发送模式。数据成功发送时单次发送和正常发送模式时动作相同。但是数据没有成功发送时会出现以下结果：

- TPIF 置位（TPIE=1），对应的 BUF SLOT 数据会被清除。
- 有错误发送时，KOER 更新，BEIF 置位（BEIE=1）。
- 仲裁失败，ALIF 置位（ALIE=1）。

单次发送模式，不能单独依靠 TPIF 来判断发送完成，需要同 BEIF 和 ALIF 一起判断发送是否完成。

26.4.7 取消数据发送

可以通过 TPA 或者 TSA 取消已请求但还没有被执行的数据发送。取消数据发送会出现以下几种情况：

- 仲裁中
  - 节点仲裁失败，则取消数据发送。
  - 节点仲裁成功，则继续发送。
- 数据发送中
  - 成功发送数据且收到 ACK，对应的标志和状态正常置位。数据发送不取消。
  - 成功发送数据但没有收到 ACK，数据发送取消，错误计数器增加。
  - TSALL=1 设定的发送数据，正在发送的 STB SLOT 数据正常发送，没有开始发送的 STB SLOT 被取消。

取消数据发送的结果有以下两种情况。

- TPA 释放 PTB，且使 TPE=0。
- TSA 释放一个 STB SLOT 或者全部 STB SLOT 取决是 TSONE 还是 TSALL 使能的发送。

26.4.8 数据接收

接收筛选器组可以过滤掉不需要的接收数据，减少中断的发生和 RB 的读取，从而降低 CPU 负荷。接收数据设定步骤如下：

1. 设定筛选器组。
2. 设定 RFIE，RAIE 和 AFWL。
3. 等待 RFIF 或者 RAIF。
4. 通过 RBUF 从 RB FIFO 中读取最早接收到的数据。
5. 设置 RREL=1，选择下一个 RB SLOT。
6. 重复 4，5 直到通过 RSTAT 确认 RB 为空。

26.4.9 错误处理

CAN_CTRL 一方面可以自动处理部分错误，比如自动重发数据或者丢弃接收到含有错误的帧，另一方面通过中断将错误向 CPU 报告。

CAN 节点有以下三种错误状态：

- 错误主动：节点检测到错误时自动发送主动错误标志。
- 错误被动：节点检测到错误时自动发送被动错误标志。
- 节点关闭：关闭状态下此节点不再影响整个 CAN 网络。

CAN_CTRL 提供 TECNT 和 RECNT 两个计数器用于计数错误。TECNT 和 RECNT 计数器按照 CAN2.0B 协议规定的规则进行增减。另外提供可编程的 CAN 错误警告 LIMIT 寄存器用于产生错误中断通知 CPU。

CAN 通信过程中有以下 5 种错误类型，错误类型可以通过 EALCAP 寄存器的 KOER 位识别。

- 位错误
- 形式错误
- 填充错误
- 应答错误
- CRC 错误

26.4.10 节点关闭

当发送错误数大于 255 时，CAN 节点自动进入节点关闭状态从而不参与 CAN 通信，直到返回到错误主动状态。可以通过 CFG_STAT 寄存器的 BUSOFF 位确认 CAN 节点关闭状态。BUSOFF 被置位的同时 EIF 中断产生。

CAN 从节点关闭状态状态恢复到错误主动状态有以下两种方法：

- 上电复位
- 接收到连续 128 个 11 位的隐形位序列（恢复序列）

节点关闭状态下，TECNT 值保持不变，RECNT 用于计数恢复序列。从节点关闭状态恢复后，TECNT 和 RECNT 被复位为 0。

节点关闭标志 BUSOFF 置位的同时，CFG_STAT 寄存器的 RESET 位也被置位。

26.4.11 仲裁失败位置捕捉

CAN_CTRL 能够精确捕捉到仲裁失败位的位置并反映到 ALC 寄存器中。ALC 寄存器中保存着最近一次仲裁失败位的位置，如果节点赢得仲裁，则 ALC 位不更新。

ALC 值定义如下：

SOF 位后，第一个 ID 数据位 ALC 为 0，第二个 ID 数据位 ALC 为 1，依次类推。因为仲裁只发生在仲裁场内，所以 ALC 的最大值为 31。比如一个标准格式远程帧和一个扩展帧仲裁，扩展帧在 IDE 位失败，则 ALC=12。

26.4.12 回环模式

CAN_CTRL 支持以下两种回环模式：

- 内部回环
- 外部回环

两种回环模式都可以接收自己发出的数据帧，主要用于测试用途。

内部回环模式，模块内部将接收数据线连接到发送数据线，并且发送数据不输出。内部回环模式下，节点会生成自应答信号以避免 ACK 错误。

外部回环模式保持和收发器的连接因此发送的数据仍能出现在 CAN 总线上，在收发器的帮助下，CAN 能收到自己发送的数据。外部回环模式可以通过 RCTRL 寄存器的 SACK 位来决定是否生成自应答信号，SACK=0 时，不生成自应答信号，SACK=1 时，生成自应答信号。

外部回环模式，SACK=0 时，会出现以下两种情况：

- 其它节点也收到本节点发送的数据帧并发送应答信号，该情况下本节点能够成功收发数据。
- 如果没有其它节点返回应答信号，则会产生应答错误，会重新发送数据并增加错误计数器。此时推荐采用单次发送模式。

从回环模式返回到正常模式时，除了清除模式位以外，还需要软件复位 CAN_CTRL。

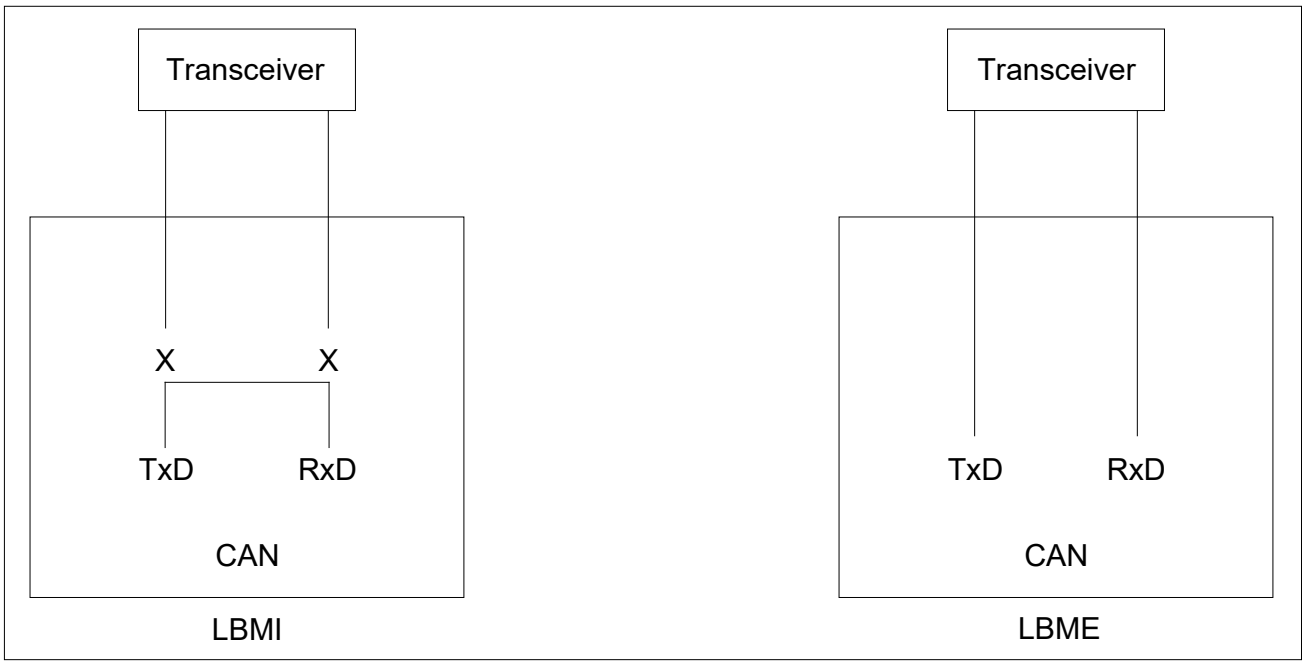


图 26-6 CAN LBMI 和 LBME 示意图

26.4.13 静默模式

静默模式可以用来监控 CAN 网络数据。在静默模式下，可以从 CAN 总线接收数据，不向总线发送任何数据。将 TCMD 寄存器中的 LOM 置 1，使 CAN 总线控制器进入静默模式，将其清 0 可以离开静默模式。

外部回环模式可以和静默模式组合成外部回环静默模式，此时 CAN 可以认为一个安静的接收者，但在有必要的时候可以发送数据。外部回环静默模式下，帧包含自应答信号允许被发送，但是该节点不会产生错误标志和过载帧。

26.4.14 软件复位功能

通过设定寄存器 CFG_STAT 寄存器的 RESET 位为 1,实现软件复位功能，软件复位功能的复位范围如下表所示。

表 26-3 软件复位范围表

寄存器位名	软件复位	备注	寄存器位名	软件复位	备注
ACFADR	否	-	EWL	是	-
ACODE	否	只能在软件复位时可以写	KOER	是	-
AE_x	否	-	LBME	是	-
AFWL	否	-	LBMI	是	-
AIF	是	-	RACTIVE	是	接收立即停止，并不生成ACK
ALC	是	-	RAFIE	否	-
ALIE	否	-	RAFIF	是	-
ALIF	是	-	RBALL	是	-
AMASK	否	只能在软件复位时可以写	RBUF	是	所有RB被标记为空，数值不定
BEIE	否	-	REF_ID	否	-
BEIF	是	-	REF_IDE	否	-
BUSOFF	否	通过写1清除	RFIE	否	-
EIEF	否	-	RFIF	是	-
EIF	否	-	RIE	否	-
EPASS	否	-	RIF	是	-
EPIE	否	-	ROIE	否	-
EPIF	是	-	ROIF	是	-
EWARN	否	否	ROM	否	-

寄存器位名	软件复位	备注	寄存器位名	软件复位	备注
ROV	是	-	TSNEXT	是	-
RREL	是	-	TSONE	是	-
PRESC	否	只能在软件复位时可以写	TPIE	否	-
RSTAT	是	-	TPIF	是	-
SACK	是	-	TPSS	是	-
SELMASK	否	-	TSFF	是	所有STB SLOT被标记为空
SEG_1	否	只能在软件复位时可以写	TSIE	否	-
SEG_2	否	只能在软件复位时可以写	TSIF	是	-
SJW	否	只能在软件复位时可以写	TSSS	是	-
TACTIVE	是	发送立即停止	TSSTAT	是	所有STB SLOT被标记为空
TBE	是	-	TTEN	是 是	-
TBF	否	-	TTIF	是	-
TBPTR	否	-	TTIE	否	-
TBSEL	是	-	TTPTR	否	-
TBUF	是	所有的STB被标记为空，指向PTB	TTTBM	否	-
TECNT	否	通过BUSOFF=1清除	TTYPE	否	-
TEIF	是	-	TT_TRIG	否	-
TPA	是	-	TT_WTRIG	否	-
TPE	是	-	T_PRESC	否	-
TSA	是	-	WTIE	否	-
TSALL	是	-	WTIF	是	
TSMODE	否	-			

26.4.15 向上兼容 CAN-FD 功能

CAN-CTRL 即使在包含 CAN-FD 网络中收到 CAN-FD 的帧，接收器会自动忽略这些帧，不返回 ACK，等到总线空闲时，再发送或者接收下一个 CAN2.0B 的帧。

26.4.16 时间触发 TTCAN

CAN-CTRL 为 ISO11898-4 规定的时间触发通信方式提供部分(lever 1)硬件支持。本章节从以下 5 个部分介绍 TTCAN 功能。

26.4.16.1 TTCAN 模式下的 TBUF 行为

TTTBM=1

TTTBM=1 时，PTB 和 STB SLOT 一样组成 TB SLOT，通过 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。

26.4.16.2 TTCAN 功能描述

上电后，Time Master 需要根据 ISO 11898-4 协议进行初始化。一个 CAN 网络中，最多可以有 8 个潜在的 Time Master。每一个 Time Master 都具有自己的参考消息 ID（ID 最后 3 位）。这些潜在的 Time Master 根据自己的优先级发送各自的参考消息。

TTEN=1 后，16 位的计数器开始工作，当参考消息被成功接收或者 Time Master 成功发送参考消息时，CAN 控制器将 Sync_Mark 拷贝给 Ref_Mark，Ref_Mark 将 cycle time 设置为 0。成功接收参考消息置位 RIF 标志而成功发送参考消息置位 TPIF 标志或者 TSIF 标志。此时主机需要准备下一个动作的触发条件。

触发条件可以是接收触发。该触发仅触发中断可用于检测期待的消息没有被收到。

触发条件也可以是发送触发。该触发开始发送通过 TTPTR 寄存器指定的 TBUF SLOT 里的数据。如果选定的 TBUF SLOT 被标记为空，则不开始发送，但置位中断标志。

26.4.16.3 TTCAN 时序

CAN_CTRL 支持 ISO11898-4 level 1。包含的一个 16 位计数器工作在为 PRESC，SEG_1，SET_2 定义的位时间下。如果 TTEN=1，则有一个额外的预分频器 T_PRESC。

一帧数据的 SOF 时，计数器的值为 Sync_Mark。如果该帧数据为参考消息，则将 Sync_Mark 拷贝给 Ref_Mark。cycle time 等于计数器的值减去 Ref_Mark。该时间用作为接收消息的时间戳或者发送消息的触发时间基准。

26.4.16.4 TTCAN 触发方式

通过 TTYPE 寄存器定义 TTCAN 的触发方式，TTPTR 寄存器指定发送 SLOT，而 TT_TRIG 指定触发器的 cycle time。

包含以下五种触发方式：

- 立即触发
- 时间触发
- 单次发送触发
- 发送开始触发
- 发送停止触发

除了立即触发方式外，所有的触发器都使用 TTIF 标志。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。

26.4.16.5 TTCAN 触发看门时间

TTCAN 触发看门时间功能类似于看门狗功能，在 TTTBM=1 时使用。用来看门从上次成功接收到参考消息开始的时间。参考消息可以在周期 cycle time 中或者一个事件后被接收，应用程序应该根据具体情况设定合适的看门时间。

如果 cycle count 等于 TT_WTRIG，则置位 WTIF。通过 WTIE 写 0，关闭看门触发。

如果 TT_WTRIG 比实际的 cycle time 小，则 TEIF 置位。

26.4.17 中断

表 26-4 CAN 中断表

中断标志	描述
RIF	接收中断
ROIF	接收上溢中断
ROIF	接收BUF满中断
RAFIF	接收BUF将满中断
TPIF	PTB发送中断
TSIF	STB发送中断
EIF	错误中断
AIF	取消发送中断
EPIE	错误被动中断
ALIF	仲裁失败中断
BEIF	总线错误中断
WTIF	触发看门中断
TEIF	触发错误中断
TTIF	时间触发中断

26.5寄存器说明

CAN_BASE_ADDR: 0x4003_0000

表 26-5 CAN 寄存器一览表

寄存器名	符号	偏移地址	位宽	复位值
CAN接收BUF寄存器	CAN_RBUF	0x00~0x0F	128	0xXXXX XXXX
CAN发送BUF寄存器	CAN_TBUF	0x50~0x5F	128	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器	CAN_RTIE	0xA4	8	0xFE
CAN接收和发送中断标志寄存器	CAN_RTIF	0xA5	8	0x00
CAN错误中断使能和标志寄存器	CAN_ERRINT	0xA6	8	0x00
CAN警告限定寄存器	CAN_LIMIT	0xA7	8	0x1B
CAN位时序寄存器	CAN_BT	0xA8	32	0x0102 0203
CAN错误和仲裁失败捕捉寄存器	CAN_EALCAP	0xB0	8	0x00
CAN接收错误计数器寄存器	CAN_RECNT	0xB2	8	0x00
CAN发送错误计数器寄存器	CAN_TECNT	0xB3	8	0x00
CAN筛选器组控制寄存器	CAN_ACFCTRL	0xB4	8	0x00
CAN筛选器组使能寄存器	CAN_ACFEN	0xB6	8	0x01
CAN筛选器组code和mask寄存器	CAN_ACF	0xB8	32	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_TRG_TRIG	0xC6	16	0x0000
TTCAN 触发看门时间寄存器	CAN_TRG_WTRIG	0xC8	16	0x0000

表 26-6 CAN 寄存器 BYTE/HALFWORD/WORD 访问排布表

地址	BYTE访问	HALFWORD访问		WORD访问			
0x00~0x0F	CAN_RBUF	CAN_RBUF		CAN_RBUF			
0x50~0x5F	CAN_TBUF	CAN_TBUF		CAN_TBUF			
0xA0	CAN_CFG_STAT	CAN_TCMD	CAN_CFG_STAT	CAN_RCTRL	CAN_TCTRL	CAN_TCMD	CAN_CFG_STAT
0xA1	CAN_TCMD	-		-			
0xA2	CAN_TCTRL	CAN_RCTRL	CAN_TCTRL	-			
0xA3	CAN_RCTRL	-		-			
0xA4	CAN_RTIE	CAN_RTIF	CAN_RTIE	CAN_LIMIT	CAN_ERRINT	CAN_RTIF	CAN_RTIE
0xA5	CAN_RTIF	-		-			
0xA6	CAN_ERRINT	CAN_LIMIT	CAN_ERRINT	-			
0xA7	CAN_LIMIT	-		-			
0xA8	CAN_BT[7:0]	CAN_BT[15:0]		CAN_BT			
0xA9	CAN_BT[15:8]			-			
0xAA	CAN_BT[23:16]	-		-			
0xAB	CAN_BT[31:24]	CAN_BT[31:16]		-			
0xB0	-	-		CAN_TECNT	CAN_RECNT	-	
0xB1	-	-		-			
0xB2	CAN_RECNT	CAN_TECNT	CAN_RECNT	-			
0xB3	CAN_TECNT	-		-			
0xB4	CAN_ACFCTRL[7:0]	CAN_ACFCTRL		CAN_ACFEN		CAN_ACFCTRL	
0xB5	CAN_ACFCTRL[15:8]	-		-			
0xB6	CAN_ACFEN[7:0]	CAN_ACFEN		-			
0xB7	CAN_ACFEN[15:8]	-		-			
0xB8	CAN_ACF	CAN_ACF		CAN_ACF			
0xBC	-	-		CAN_TTCFG	CAN_TBSLOT	-	-
0xBD	-	-		-			
0xBE	CAN_TBSLOT	CAN_TTCFG	CAN_TBSLOT	-			
0xBF	CAN_TTCFG			-			
0xC0	CAN_REF_MSG[7:0]	CAN_REF_MSG[15:0]		CAN_REF_MSG			
0xC1	CAN_REF_MSG[15:8]	-		-			
0xC2	CAN_REF_MSG[23:16]	CAN_REF_MSG[31:16]		-			
0xC3	CAN_REF_MSG[31:24]	-		-			
0xC4	CAN_TRG_CFG[7:0]	CAN_TRG_CFG		CAN_TRG_TRIG		CAN_TRG_CFG	
0xC5	CAN_TRG_CFG[15:8]	-		-			
0xC6	CAN_TRG_TRIG[7:0]	CAN_TRG_TRIG		-			
0xC7	CAN_TRG_TRIG[15:8]	-		-			
0xC8	CAN_TRG_WTRIG[7:0]	CAN_TRG_WTRIG		-		CAN_TRG_WTRIG	
0xC9	CAN_TRG_WTRIG[15:8]	-		-			

26.5.1 CAN 接收 BUF 寄存器（CAN_RBUF）

CAN Receive Buffer Registers

偏移地址：0x00~0x0F

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邮箱。

CYCLE_TIME 位仅在 TTCAN 模式时有效，表示 SOF 开始时的 cycle time。

CAN 接收邮箱的数据格式如下：

表 26-7 标准格式 CAN 接收邮箱格式

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
RBUF	ID[7:0]								ID
RBUF+1	-					ID[10:8]			ID
RBUF+2	-								ID
RBUF+3	-								ID
RBUF+4	IDE=0	RTR	0	0	DLC[3:0]				Control
RBUF+5	KOER[2:0] TX			TX	-				Status
RBUF+6	CYCLE_TIME[7:0]								TTCAN
RBUF+7	CYCLE_TIME[15:8]								TTCAN
RBUF+8	DATA1								Data
RBUF+9	DATA2								Data
RBUF+10	DATA3								Data
RBUF+11	DATA4								Data
RBUF+12	DATA5								Data
RBUF+13	DATA6								Data
RBUF+14	DATA7								Data
RBUF+15	DATA8								Data

表 26-8 扩展格式 CAN 接收邮箱格式

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
RBUF	ID[7:0]								ID
RBUF+1	ID[15:8] ID[10:8]								ID
RBUF+2	ID[23:16]								ID
RBUF+3	-			ID[28:24]					ID
RBUF+4	IDE=1	RTR	0	0	DLC[3:0]				Control
RBUF+5	KOER[2:0] TX			TX	-				Status
RBUF+6	CYCLE_TIME[7:0]								TTCAN
RBUF+7	CYCLE_TIME[15:8]								TTCAN
RBUF+8	DATA1								Data
RBUF+9	DATA2								Data
RBUF+10	DATA3								Data
RBUF+11	DATA4								Data
RBUF+12	DATA5								Data
RBUF+13	DATA6								Data
RBUF+14	DATA7								Data
RBUF+15	DATA8								Data

控制位含义如下：

IDE(IDentifier Extension):

0：标准格式

1：扩展格式

RTR(Remote Transmission Request)

0：数据帧

1：远程帧

DLC(Data Length Code):

数据长度码，设定范围为 0~8，对应数据长度为 0Byte~8Byte

26.5.2 CAN 发送 BUF 寄存器（CAN_TBUF）

CAN Transmit Buffer Registers

偏移地址：0x50~0x5F

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：

表 26-9 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[10:8]			ID
TBUF+2	-								ID
TBUF+3	- -								ID
TBUF+4	IDE=0	RTR	0	0	DLC[3:0]				Control
TBUF+5	- TX TX -								-
TBUF+6	-								-
TBUF+7	-								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
TBUF+11	DATA4								Data
TBUF+12	DATA5								Data
TBUF+13	DATA6								Data
TBUF+14	DATA7								Data
TBUF+15	DATA8								Data

表 26-10 扩展格式 CAN 发送邮箱格式

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
TBUF	ID[7:0]								ID
TBUF+1	ID[15:8] ID[10:8]								ID
TBUF+2	ID[23:16]								ID
TBUF+3	-			ID[28:24]					ID
TBUF+4	IDE=0	RTR	0	0	DLC[3:0]				Control
TBUF+5	- TX TX -								-
TBUF+6	-								-
TBUF+7	-								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
TBUF+11	DATA4								Data
TBUF+12	DATA5								Data
TBUF+13	DATA6								Data
TBUF+14	DATA7								Data
TBUF+15	DATA8								Data

控制位含义如下：

IDE(IDentifier Extension):

0：标准格式

1：扩展格式

RTR(Remote Transmission Request)

0：数据帧

1：远程帧

DLC(Data Length Code):

数据长度码，设定范围为 0~8，对应数据长度为 0Byte~8Byte

26.5.3 CAN 配置和状态寄存器（CAN_CFG_STAT）

CAN Configuration and Status Register

偏移地址：0xA0

复位值：0x80

b7	b6	b5	b4	b3	b2	b1	b0
RESET	LBME	LBMI	TPSS	TSSS	RACTIVE	TACTIVE	BUSOFF

位	标记	位名	功能	读写
b7	RESET	复位请求	复位请求位 0：不请求局部复位 1：请求局部复位 部分寄存器只能在RESET=1时进行写操作，具体请参考软件复位功能，当该节点进入BUS OFF状态时，硬件自动将RESET位置1。请注意，当RESET=0后需要11个CAN bit time该节点才能参与通信。	R/W
b6	LBME	外部回环模式使能位	外部回环模式使能位 0：禁止外部回环模式 1：使能外部回环模式 注意：通信中禁止设定该位。	R/W
b5	LBMI	内部回环模式使能位	内部回环模式使能位 0：禁止内部回环模式 1：使能内部回环模式 注意：通信中禁止设定该位。	R/W
b4	TPSS	PTB单次传输模式	PTB单次传输模式 0：禁止PTB单次传输模式 1：使能PTB单次传输模式	R/W
b3	TSSS	STB单次传输模式	STB单次传输模式 0：禁止STB单次传输模式 1：使能STB单次传输模式	R/W
b2	RACTIVE	接收中状态信号	接收中状态信号 0：非接收中 1：接收中	R
b1	TACTIVE	发送中状态信号	发送中状态信号 0：非发送中 1：发送中	R
b0	BUSOFF	总线关闭状态	总线关闭状态 0：总线有效状态 1：总线关闭状态 注意：写1可以清零TECNT和RECNT寄存器，但仅限用于调试用途。	R/W

26.5.4 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	STBY	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	STBY	STBY设定位	STBY设定位 0：清除STBY 1：置位STBY (置位前必须保证TPE,TSONE,TSALL都为0)	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

26.5.5 CAN 发送控制寄存器（CAN_TCTRL）

CAN Transmit Control Register

偏移地址：0xA2

复位值：0x90

b7	b6	b5	b4	b3	b2	b1	b0
-	TSNEXT	TSMODE	TTTBM	-	-	TSSTAT[1:0]	

位	标记	位名	功能	读写
b7	Reserved	-	必须保持复位值。	R
b6	TSNEXT	下一个STB 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

位	标记	位名	功能	读写
b1~b0	TSSTAT	STB状态	STB状态（Transmission Secondary STATus bits） TTEN=0 00：STB空 01：STB小于等于半满 10：STB大于半满 11：STB满 TTEN=0&TTTBM=1 00：PTB和STB空 01：PTB和STB非满 10：保留 11：PTB和STB满	R

26.5.6 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
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

26.5.7 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	

26.5.8 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：未发生接收上溢 1：发生接收上溢 上溢时ROIF和RFIF同时置1。 通过应用程序写1清0。	R/W
b5	RFIF	接收BUF满中断标志	接收BUF满中断标志（RB Full Interrupt Flag） 0：RB FIFO未满 1：RB FIFO满 通过应用程序写1清0。	R/W
b4	RAFIF	接收BUF将满中断标志	接收BUF将满中断标志（RB Almost Full Interrupt Flag） 0：被填充的RB SLOT数目小于AFWL设定值 1：被填充的RB SLOT数目大于等于AFWL设定值 通过应用程序写1清0。	R/W
b3	TPIF	PTB发送中断标志	PTB发送中断标志（Transmission Primary Interrupt Flag） 0：没有PTB发送完成 1：请求的PTB发送成功完成 通过应用程序写1清0。 注意：TTCAN模式时，TPIF无效，仅适用TSIF标志	R/W
b2	TSIF	STB发送中断标志	STB发送中断标志（Transmission Secondary Interrupt Flag） 0：没有STB发送完成 1：请求的STB发送成功完成 通过应用程序写1清0。 注意：TTCAN模式时，TPIF无效，仅使用TSIF标志	R/W

位	标记	位名	功能	读写
b1	EIF	错误中断标志	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

26.5.9 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	WPIE	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

26.5.10 CAN 位时序寄存器（CAN_BT）

CAN Bit Timing Register

偏移地址：0xA8

复位值：0x0102 0203

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PRESC[7:0]								-	SJW[6:0]						
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	SEG_2[6:0]							SEG_1[7:0]							

位	标记	位名	功能	读写
b31~b24	PRESC	预分频设定	预分频设定（Prescaler） 此寄存器设定值时将模块BCLK设定位（PRESC+1）分频作为TQ的时钟。	R/W
b23	Reserved	-	必须保持复位值。	R
b22~b16	SJW	再同步补偿宽度时间设定	再同步补偿宽度时间设定（Bit Timing Segment 2） 再同步补偿宽度时间=(SJW+1)*TQ	R/W
b15	Reserved	-	必须保持复位值。	R
b14~b8	SEG_2	位段2时间设定	位段2时间单元设定（Bit Timing Segment 2） 位段2时间=(SEG_2+1)*TQ	R/W
b7~b0	SEG_1	位段1时间设定	位段1时间单元设定（Bit Timing Segment 1） 位段1时间=(SEG_1+2)*TQ	R/W

26.5.11 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

26.5.12 CAN 警告限定寄存器（CAN_LIMIT）

CAN Warning Limits Register

偏移地址：0xA7

复位值：0x1B

b7	b6	b5	b4	b3	b2	b1	b0
AFWL[3:0]				EWL[3:0]			

位	标记	位名	功能	读写
b7~b4	AFWL	接收BUF将满Warning Limit	接收BUF将满Warning Limit（receive buffer Almost Full Warning Limit） 设定值范围为1~10。 AFWL=0无意义，当做AFWL=1处理。	R/W
b3~b0	EWL	Error Waring Limit编程值	Error Waring Limit编程值（Programmable Error Warning Limit） Error Waring Limit=（EWL+1）*8。 该寄存器设定值影响EIF标志。	R/W

26.5.13 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

26.5.14 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

26.5.15 CAN 筛选器组控制寄存器（CAN_ACFCTRL）

CAN Acceptance Filter Control Register

偏移地址：0xB4

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
-		SELMASK	-	ACFADR			

位	标记	位名	功能	读写
b7~b6	Reserved	-	必须保持复位值。	R
b5	SELMASK	选择筛选器的屏蔽寄存器	选择筛选器的屏蔽寄存器（SELect acceptance MASK） 0：ACF指向筛选器ID寄存器 1：ACF指向筛选器MASK寄存器 通过ACFADR选择具体的筛选寄存器组	R/W
b4	Reserved	-	必须保持复位值。	R
b3~b0	ACFADR	筛选器地址	筛选器地址（acceptance filter address） ACFADR指向具体的筛选器，通过SELMASK去区分ID和MASK。 0000：指向ACF_1 0001：指向ACF_2 0010：指向ACF_3 0011：指向ACF_4 0100：指向ACF_5 0101：指向ACF_6 0110：指向ACF_7 0111， 1000， 1001， 1010， 1011， 1100， 1101， 1110， 1111：指向ACF_8	

26.5.16 CAN 筛选器组使能寄存器（CAN_ACFEN）

CAN Acceptance Filter Enable Register

偏移地址：0xB6

复位值：0x01

b7	b6	b5	b4	b3	b2	b1	b0
AE_8	AE_7	AE_6	AE_5	AE_4	AE_3	AE_2	AE_1

位	标记	位名	功能	读写
b7	AE_8	ACF_8使能	ACF_8使能（Acceptance Filter 8 Enable） 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

26.5.17 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
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

26.5.18 TTCAN TB slot 指针寄存器（CAN_TBSLOT）

TTCAN TB Slot Pointer Register

偏移地址：0xBE

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
TBE	TBF	-	-	-	TBPTR[2:0]		

位	标记	位名	功能	读写
b7	TBE	设置TB为空	设置TB为空（set TB slot to “empty”） 0：无操作 1：被TBPTR选择的SLOT被标记为空 当SLOT被标记为空并且TSFF=0时，TBE自动复位为0。 如果设定此位为1时，被选定的SLOT中存在数据正在发送状态则TBE=1，则等到发送完成、发送错误或者发送取消后TBE复位为0。 TBE优先级高于TBF。	R/W
b6	TBE	设置TB为已填充	设置TB已填充（set TB slot to “Filled”） 0：无操作 1：被TBPTR选择的SLOT被标记为已填充 当SLOT被标记为已填充并且TSFF=1时，TBE自动复位为0。	R/W
b5~b3	Reserved	-	必须保持复位值。	R
b2~b0	TBPTR	TB SLOT指针	TB SLOT指针（Pointer to a TB message slot） 000：指向PTB 001：指向STB SLOT1 010：指向STB SLOT2 011：指向STB SLOT3 100：指向STB SLOT4 其他：设定禁止 被指向的TB SLOT可以通过TBUF进行读写访问，并且可以通过TBE和TBF来标记是否已经被填充。 TTCAN模式时，TBSEL和TSNEXT寄存器无效。 注意：仅可以在TSFF=0时对该位进行写操作。	R/W

26.5.19 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：BT寄存器设定的位时间的1分频 01：BT寄存器设定的位时间的2分频 10：BT寄存器设定的位时间的4分频 11：BT寄存器设定的位时间的8分频 注意：T_PRESC可在TTEN=0时进行写操作或者写TTEN=1时同时操作。	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

26.5.20 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
b28~b0	REF_ID	参考消息的ID位	参考消息的ID位（REFerence message IDentifier） REF_ID=0：REF_ID[28:0]有效 REF_ID=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

26.5.21 TTCAN 触发配置寄存器（CAN_TRG_CFG）

TTCAN Reference Message Register

偏移地址：0xC4

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TEW[3:0]				-	TTYPE[2:0]			-					TTPTR[2:0]		

位	标记	位名	功能	读写
b15~b12	TEW	发送使能窗口	发送使能窗口（Transmit Enable Window） 用于TTCAN的单次发送触发模式（Single Shot Transmit Trigger）,可以设定TEW+1个cycle time的窗口，发送仅在此窗口内被允许。	R/W
b11	Reserved	-	必须保持复位值。	R
b10~b8	TTYPE	触发类型	触发类型（Trigger Type） 000：立即触发（Immediate Trigger for immediate transmission） 001：时间触发（Time Trigger for receive triggers） 010：单次发送触发（Single Shot Transmit Trigger for exclusive time windows） 011：发送开始触发（Transmit Start Trigger for merged arbitrating time windows） 100：发送停止触发（Transmit Stop Trigger for merged arbitrating time windows） 其他：保留 触发时间通过TT_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
b2~b0	TTPTR	发送触发器TB slot指针	发送触发器TB slot指针（Transmit Trigger TB slot Pointer） 000：指向PTB 001：指向STB SLOT1 010：指向STB SLOT2 011：指向STB SLOT3 100：指向STB SLOT4 其他：设定禁止 如果指向的TB SLOT被标记为空，当到达触发时间后，TEIF置位。	R/W

26.5.22 TTCAN 触发时间寄存器（CAN_TRG_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

26.5.23 TTCAN 触发看门时间寄存器（TRG_WTRIG）

TTCAN Watch Trigger Time Register

偏移地址：0xC8

复位值：0xFFFF

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

位	标记	位名	功能	读写
b15~b0	TT_TRIG	触发时间	触发时间（Trigger Time） 用于指定看门触发器的cycle time。 当TT_WTRIG的最高字节写操作后，TT_WTRIG值开始向CAN时钟域的传递。因此如果BYTE操作，需先写低字节再写高字节。	R/W

26.6使用注意事项

26.6.1 CAN 总线抗干扰措施

CAN 总线广泛应用于汽车、工业控制等行业，如果 CAN 应用现场电磁环境比较恶劣，存在电路不平衡性、空间电磁场、电网进线等因素，会导致 CAN 总线因辐射、传导干扰而产生大量通信噪声，致使总线错误帧增加、重发频繁，正确数据不能及时到达等情况，严重影响数据通信质量。所以实际应用中应该致力于消除噪声干扰，保障 CAN 总线网络稳定工作。

以下是几类常用的 CAN 总线抗干扰措施（包括但不仅限于）

1） 增加 CAN 总线接口电气隔离
2） 共收发器的信号地
3） 使用屏蔽双绞线缆并正确接地
4） 提高 CAN 传输线双绞程度
5） 增加信号保护器
6） 改进网络拓扑
7） 应用层软件抗干扰机制

26.6.2 CAN 控制器噪声制约

在 CAN 总线网络中应确保通信的位时间满足标准协议的要求，若引入不满足位时间宽度的噪声干扰，可能引起 CAN 控制器异常动作。

27 模数转换器（ADC）

27.1模块简介

外部的模拟信号需要转变成数字信号才能由 MCU 进一步处理。内部集成了一个 12 位高精度、高转换速率的逐次逼近型模数转换器(SAR ADC)模块。具有以下特性：

- 12 位转换精度；
- 1M SPS 转换速度；
- 40 个输入通道，包括 36 路外部管脚输入、1 路内部温度传感器电压、1 路 1/3 AVCC 电压、2 路 DAC 输出；
- 4 种参考源：AVCC 电压、ExRef 引脚、内置 1.5v 参考电压、内置 2.5v 参考电压；
- ADC 的电压输入范围：0~Vref；
- 4 种转换模式：单次转换、顺序扫描连续转换、插队扫描连续转换、连续转换累加；
- 输入通道电压阈值监测；
- 软件可配置 ADC 的转换速率；
- 内置信号跟随器，可转换高阻信号；
- 支持片内外设自动触发 ADC 转换，有效降低芯片功耗并提高转换的实时性。

27.2ADC 框图

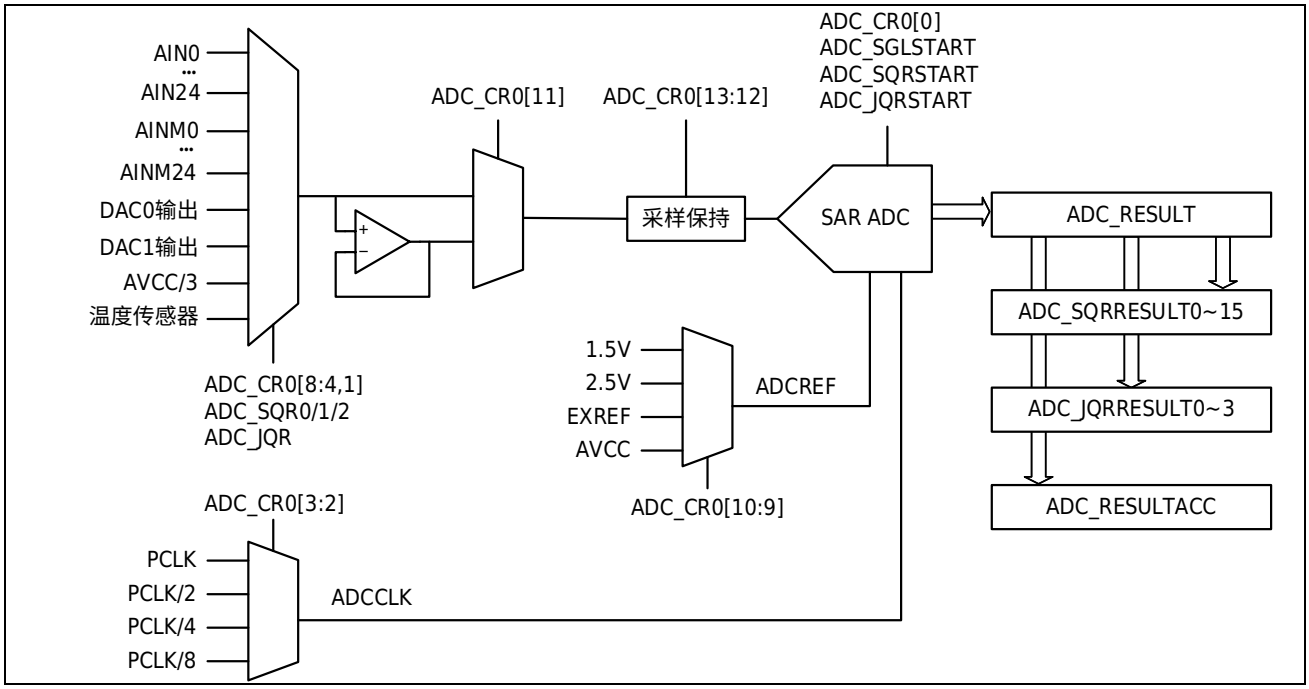


图 27-1 ADC 示意框图

27.3转换时序及转换速度

ADC 的转换时序如下图所示：一次完整的 ADC 转换由转换过程及逐次比较过程组成。其中转换过程需要 4~12 个 ADCCLK，由 ADC_CR0.SAM 配置；逐次比较过程需要 16 个 ADCCLK。所以，一次 ADC 转换共需要 20~28 个 ADCCLK。

ADC 转换速度的单位为 SPS，即每秒进行多少次 ADC 转换。ADC 转换速度的计算方法为：ADCCLK 的频率 / 一次 ADC 转换所需要的 ADCCLK 的个数。

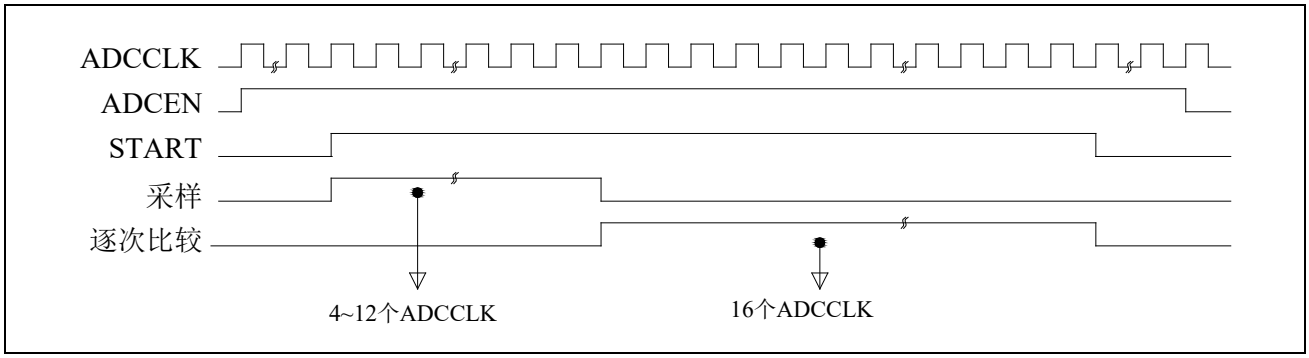


图 27-2 ADC 转换时序图

ADC 转换速度与 ADC 参考电压及 AVCC 电压相关，最高转换速度如下表所示：

ADC 参考电压	AVCC 电压	最高转换速度	最大 ADCCLK 频率
内部 1.5V	1.8V~5.5V	200K SPS	4MHz
内部 2.5V	2.8V~5.5V	200K SPS	4MHz
AVCC / ExRef	1.8V~2.4V	200K SPS	4MHz
AVCC / ExRef	2.4V~2.7V	500K SPS	16MHz
AVCC / ExRef	2.7V~5.5V	1M SPS	24MHz

27.4单次转换模式

在单次转换模式下，ADC 启动后只执行一次转换，可对所有的 30 路 ADC 通道进行转换。该模式既可通过设置 ADC_SglStart.Start 位启动也可通过设置 ADC_ExtTrigger0 的外部触发启动。一旦选定通道的 ADC 转换完成，ADC_IFR.SGLIF 位会自动置 1，转换结果保存在 ADC_Result 寄存器中。

通过 ADC_SglStart.Start 位启动 ADC 单次转换操作流程：

Step1：配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2：设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3：设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step6：设置 ADC_CR1.Mode 为 0，选择单次转换模式。

Step7：配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step8：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step9：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step10：配置 ADC_CR0.SGLMux，选择待转换的通道。

Step11：设置 ADC_ICR.SGLIC 为 0，清除 ADC_IFR.SGLIF 标志。

Step12：设置 ADC_SglStart.Start 为 1，启动 ADC 单次转换。

Step13：等待 ADC_IFR. SGLIF 变为 1，读取 ADC_Result 寄存器以获取 ADC 转换结果。

Step14：如需对其它通道进行转换，重复执行 Step10~Step13。

Step15：设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

通过外部触发启动 ADC 单次转换操作流程：

Step1：配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2：设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3：设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step6：设置 ADC_CR1.Mode 为 0，选择单次转换模式。

Step7：设置 ADC_CR0.IE 为 1，使能 ADC 中断。

Step8：使能 NVIC 中断向量表中的 ADC 中断。

Step9：配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step10：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step11：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step12：配置 ADC_CR0.SGLMux，选择待转换的通道。

Step13：设置 ADC_IFR 为 0x0，清除 ADC 中断标志。

Step14：配置 ADC_ExtTrigger0，选择外部触发条件。

Step15：当外部触发条件触发 ADC 完成转换时，ADC 模块会产生中断。用户可在 ADC 中断服务程序中读取 ADC_Result 寄存器以获取 ADC 转换结果。

Step16：如需对其它通道进行转换，重复执行 Step12~Step15。

Step17：设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

27.5扫描转换模式

扫描转换模式分为顺序扫描转换和插队扫描转换两种模式。两种模式各自单独工作时，均可连续对多个通道进行多次转换；当两种模式同时工作时，则优先对插队扫描配置的通道进行转换。

27.5.1 顺序扫描转换模式

顺序扫描转换模式最多可进行 16 次连续转换，转换的总次数由 ADC_SQR2.CNT 进行配置；可配置所有通道进行转换，待转换通道由 ADC_SQRx.CHxMux 进行配置。该模式既可通过设置 ADC_SqrStart.Start 位启动也可通过设置 ADC_ExtTrigger0 的外部触发启动。启动转换后，ADC 模块依次转换 CHxMux~CH0Mux 中配置的通道直到总转换次数完成。ADC 模块完成总转换次数后，ADC_IFR.SQRIF 位会自动置 1，转换结果保存在转换通道所对应的 ADC_SqrResultx~ADC_SqrResult0 寄存器中。下图演示了对 AIN0、AIN1、AIN5 进行 8 次转换的顺序扫描转换过程。其中顺序扫描转换通道 7，4，1 配置为 AIN1，转换通道 6，3，0 配置为 AIN0，转换通道 5，2 配置为 AIN5。ADC_SqrStart.Start 置 1 后，ADC 模块会依次对顺序扫描转换通道 7~0 进行转换。

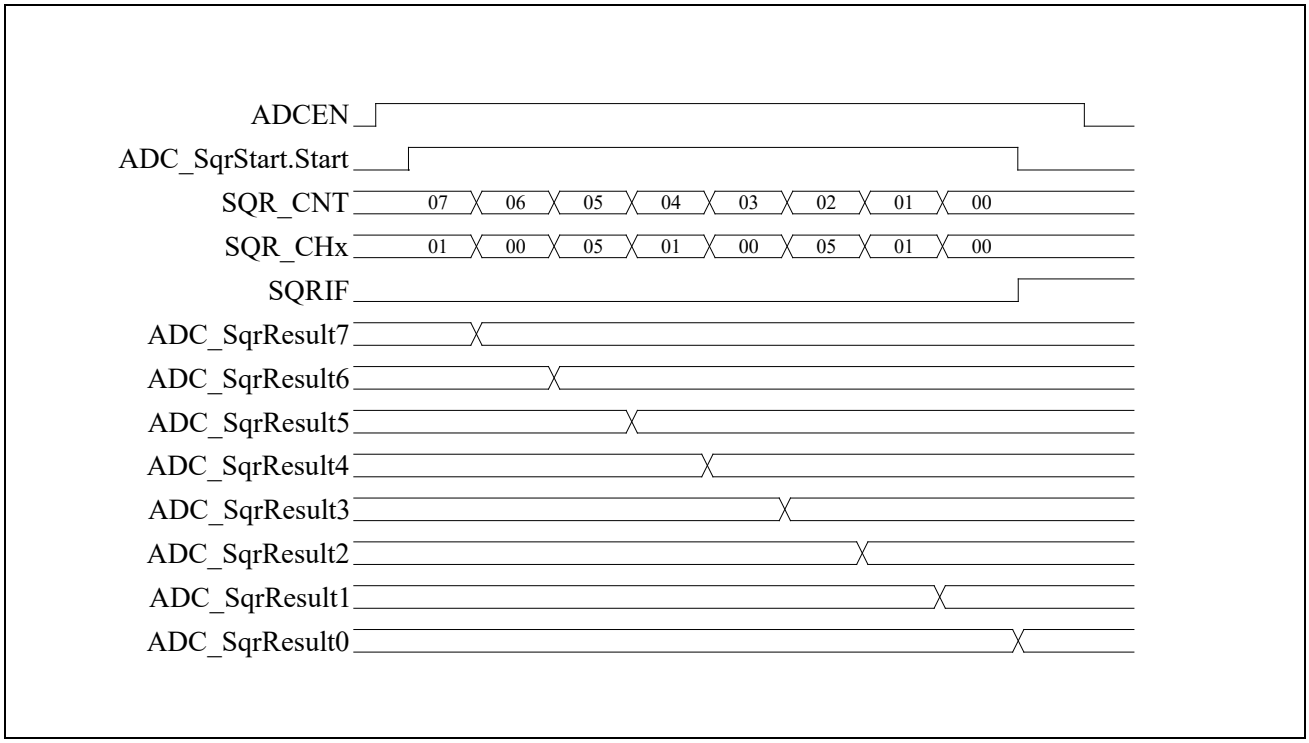


图 27-3 ADC 顺序扫描转换过程示例

通过 ADC_SqrStart.Start 位启动 ADC 顺序扫描转换操作流程：

Step1：配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2：设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3：设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step6：设置 ADC_CR1.Mode 为 1，选择扫描转换模式。

Step7：配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step8：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step9：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step10：配置 ADC_SQRx.CHxMux，选择顺序扫描转换通道。

Step11：配置 ADC_SQR2.CNT，选择顺序扫描转换的总转换次数。

注：总转换次数需与 Step9 中配置的转换通道数保持一致。

Step12：设置 ADC_ICR.SQRIC 为 0，清除 ADC_IFR.SQRIF 标志。

Step13：设置 ADC_SqrStart.Start 为 1，启动 ADC 顺序扫描转换。

Step14：等待 ADC_IFR.SQRIF 变为 1，读取 ADC_SqrResultx ~ ADC_SqrResult0 寄存器以获取相应通道的转换结果。

Step15：如需对其它通道进行转换，重复执行 Step10~Step14。

Step16：设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

通过外部触发启动 ADC 顺序扫描转换操作流程：

Step1：配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2：设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3：设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step6：设置 ADC_CR1.Mode 为 1，选择扫描转换模式。

Step7：设置 ADC_CR0.IE 为 1，使能 ADC 中断。

Step8：使能 NVIC 中断向量表中的 ADC 中断。

Step9：配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step10：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step11：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step12：配置 ADC_SQRx.CHxMux，选择顺序扫描转换通道。

Step13：配置 ADC_SQR2.CNT，选择顺序扫描转换的总转换次数。

注：总转换次数需与 Step11 中配置的转换通道数保持一致。

Step14：设置 ADC_IFR 为 0x0，清除 ADC 中断标志。

Step15：配置 ADC_ExtTrigger0，选择外部触发条件。

Step16：当外部触发条件触发 ADC 完成转换时，ADC 模块会产生中断。用户可在 ADC 中断服务程序中读取 ADC_SqrResultx ~ ADC_SqrResult0 寄存器以获取相应通道的转换结果。

Step17：如需对其它通道进行转换，重复执行 Step12~Step16。

Step18：设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

27.5.2 插队扫描转换模式

插队扫描转换模式最多可进行 4 次连续转换，转换的总次数由 ADC_JQR.CNT 进行配置；可配置所有通道进行转换，待转换通道由 ADC_JQR.CHxMux 进行配置。该模式既可通过设置 ADC_JqrStart.Start 位启动也可通过设置 ADC_ExtTrigger1 的外部触发启动。启动转换后，ADC 模块依次转换 CHxMux~CH0Mux 中配置的通道直到总转换次数完成。ADC 模块完成总转换次数后，ADC_IFR.JQRIF 位会自动置 1，转换结果保存在转换通道所对应的 ADC_JqrResultx~ ADC_JqrResult0 寄存器中。下图演示了对 AIN0、AIN1、AIN5 进行 4 次转换的插队扫描转换的过程。其中插队扫描转换通道 3，2，1，0 分别设置为 AIN5，AIN0，AIN1，AIN5。ADC_JqrStart.Start 置 1 后，ADC 模块会依次对插队扫描转换通道 3~0 进行转换。

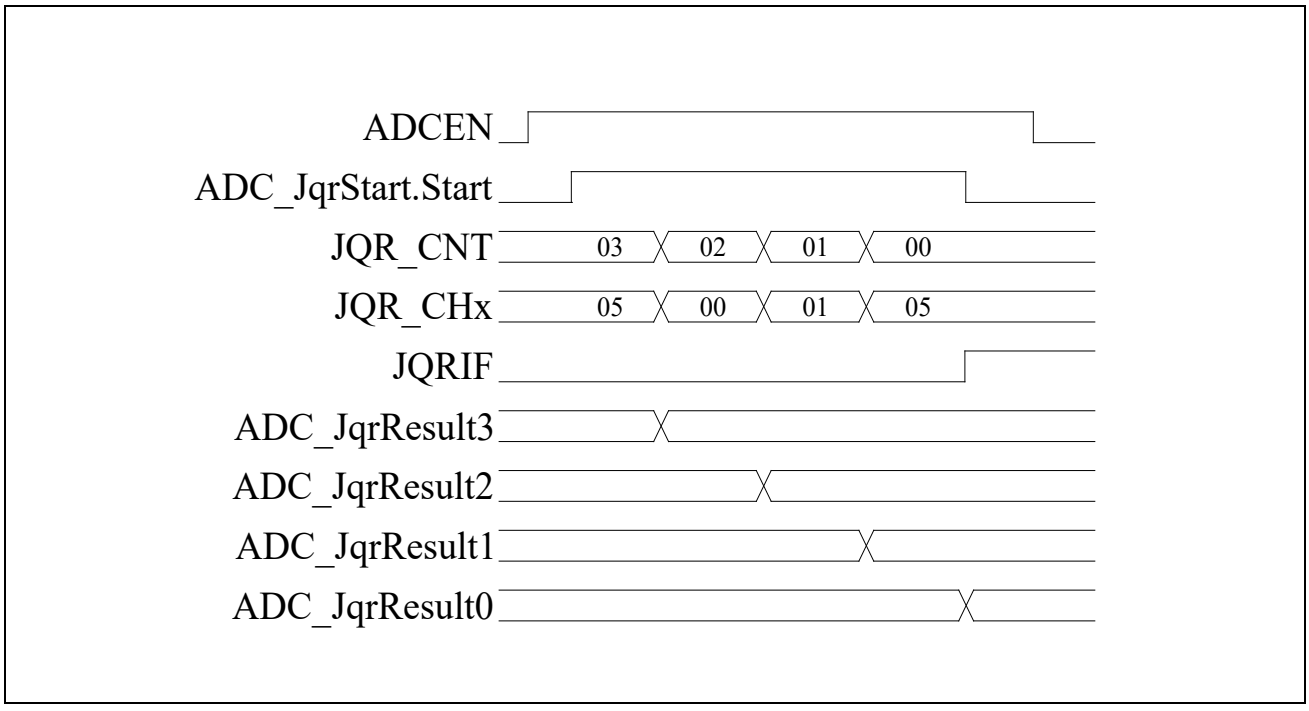


图 27-4 ADC 插队扫描转换过程示例

通过 ADC_JqrStart.Start 位启动 ADC 插队扫描转换操作流程：

Step1：配置 PAADS~PCADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2：设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3：设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step6：设置 ADC_CR1.Mode 为 1，选择扫描转换模式。

Step7：配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step8：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step9：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step10：配置 ADC_JQR.CHxMux，选择插队扫描转换通道。

Step11：配置 ADC_JQR.CNT，选择插队扫描转换的总转换次数。

注：总转换次数需与 Step9 中配置的转换通道数保持一致。

Step12：设置 ADC_ICR.JQRIC 为 0，清除 ADC_IFR.JQRIF 标志。

Step13：设置 ADC_JqrStart.Start 为 1，启动 ADC 插队扫描转换。

Step14：等待 ADC_IFR.JQRIF 变为 1，读取 ADC_JqrResultx~ ADC_JqrResult0 寄存器以获取相应通道的转换结果。

Step15：如需对其它通道进行转换，重复执行 Step10~Step14。

Step16：设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

通过外部触发启动 ADC 插队扫描转换操作流程：

Step1：配置 PAADS~PCADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2：设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3：设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step6：设置 ADC_CR1.Mode 为 1，选择扫描转换模式。

Step7：设置 ADC_CR0.IE 为 1，使能 ADC 中断。

Step8：使能 NVIC 中断向量表中的 ADC 中断。

Step9：配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step10：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step11：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step12：配置 ADC_JQR.CHxMux，选择插队扫描转换通道。

Step13：配置 ADC_JQR.CNT，选择插队扫描转换的总转换次数。

注：总转换次数需与 Step11 中配置的转换通道数保持一致。

Step14：设置 ADC_IFR 为 0x0，清除 ADC 中断标志。

Step15：配置 ADC_ExtTrigger1，选择外部触发条件。

Step16：当外部触发条件触发 ADC 完成转换时，ADC 模块会产生中断。用户可在 ADC 中断服务程序中读取 ADC_JqrResultx~ ADC_JqrResult0 寄存器以获取相应通道的转换结果。

Step17：如需对其它通道进行转换，重复执行 Step12~Step16。

Step18：设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

插队扫描转换的执行优先级高于顺序扫描转换，当顺序扫描转换正在进行时，如果插队扫描转换启动，则顺序扫描在完成当前通道的转换后暂停，等到插队扫描转换完成后，再继续执行剩下的通道转换。下图演示了对 AIN0，AIN1，AIN5，AIN8 进行顺序扫描转换时，启动对 AIN2，AIN6 进行插队扫描转换的过程。其中顺序扫描转换通道 3，2，1，0 分别设置为 AIN1，AIN0，AIN5，AIN8，插队扫描转换通道 1，0 分别设置为 AIN6，AIN2。在顺序扫描进行 AIN0 转换的过程中，启动了插队扫描，顺序扫描会将当前 AIN0 的转换完成然后暂停，等到插队扫描完成 AIN6 和 AIN2 的转换后，顺序扫描再进行 AIN5 和 AIN8 的转换。

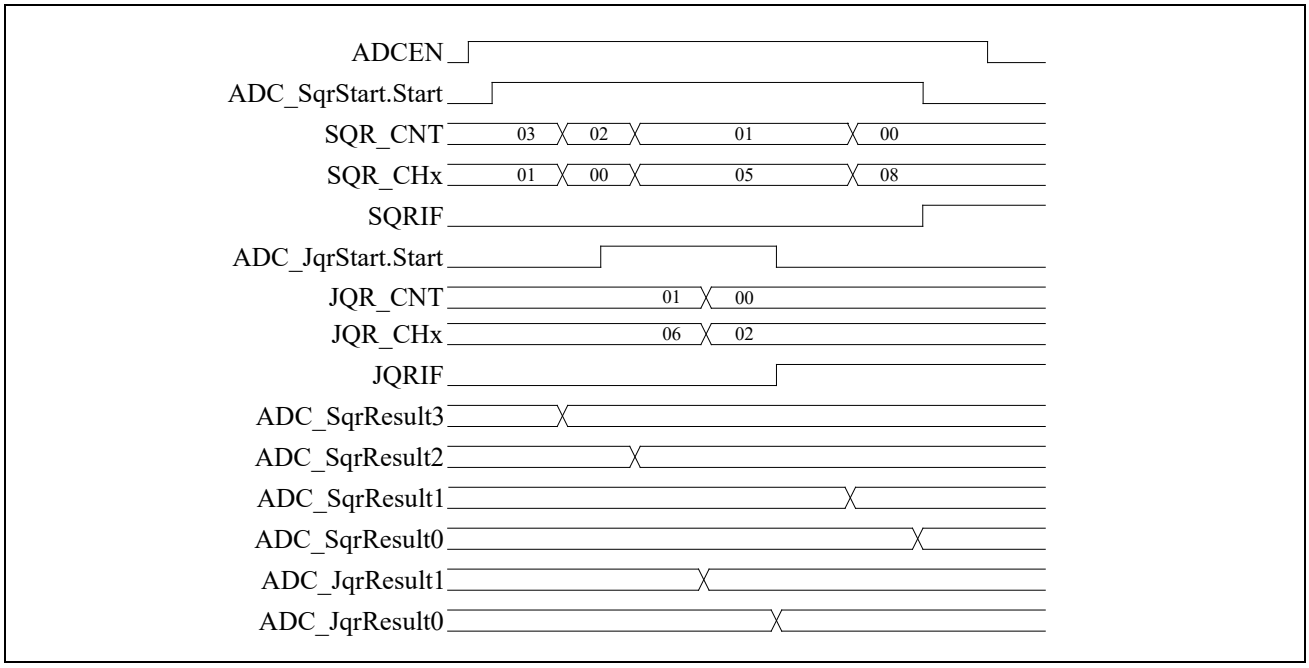


图 27-5 ADC 顺序扫描过程中进行插队扫描示例

27.5.3 扫描转换触发 DMA 读取

顺序扫描和插队扫描转换完成后，可自动触发 DMA 读取转换结果。顺序扫描模式下，通过配置 ADC_CR1. DmaSqr 为 1 使能该功能；插队扫描模式下，通过配置 ADC_CR1. DmaJqr 为 1 使能该功能。

27.6连续转换累加模式

在连续转换累加模式下，启动一次 ADC 可对多个通道进行多次转换并对每次转换的结果进行累加；可配置所有通道进行转换。ADC 转换的总次数由 ADC_SQR2.CNT 进行配置；待转换通道由 ADC_SQRx.CHxMux 进行配置。该模式既可通过设置 ADC_SqrStart.Start 位启动也可通过设置 ADC_ExtTrigger0 的外部触发启动。启动连续转换后，ADC 模块依次转换 CHxMux~CH0Mux 中配置的通道直到总转换次数完成。ADC 模块完成总转换次数后，ADC_IFR.SQRIF 位会自动置 1，转换结果的累加值保存在 ADC_ResultAcc 寄存器中。

下图演示了对 AIN0、AIN1、AIN5 进行 10 次连续转换累加的过程。顺序扫描转换通道 9，6，3，0 配置为 AIN0，转换通道 8，5，2 配置为 AIN1，转换通道 7，4，1 配置为 AIN5。ADC_SqrStart.Start 置 1 后，ADC 模块会按照顺序扫描转换通道配置依次进行转换，直到 SQR_CNT 的计数值变为 0。每次转换完成时，ADC_ResultAcc 寄存器都会自动进行累加。图中给定的 AIN0、AIN1、AIN5 的转换结果依次为 0x010、0x020、0x040。

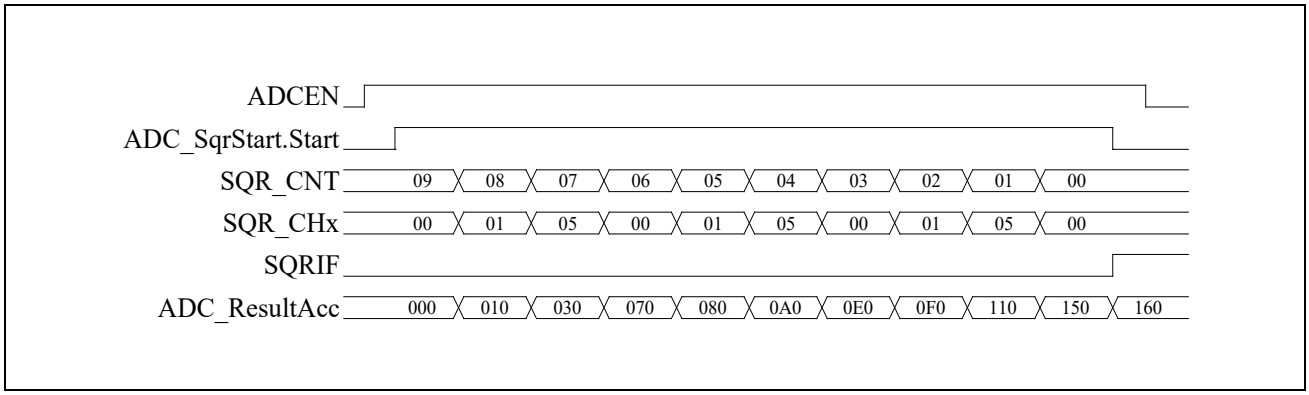


图 27-6 ADC 连续转换累加过程示例

通过 ADC_SqrStart.Start 位启动 ADC 连续转换累加操作流程：

Step1：配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2：设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3：设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step6：设置 ADC_CR1.Mode 为 1，选择扫描转换模式。

Step7：设置 ADC_CR1.RAccEn 为 1，使能 ADC 转换自动累加功能。

Step8：配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step9：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step10：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step11：配置 ADC_SQRx.CHxMux，选择顺序扫描转换通道。

Step12：配置 ADC_SQR2.CNT，选择顺序扫描转换的总转换次数。

注：总转换次数需与 Step10 中配置的转换通道数保持一致。

Step13：设置 ADC_ICR.SQRIC 为 0，清除 ADC_IFR.SQRIF 标志。

Step14：设置 ADC_CR1.RAccClr 为 0，清零 ADC_ResultAcc 寄存器。

Step15：设置 ADC_SqrStart.Start 为 1，启动 ADC 顺序扫描转换。

Step16：等待 ADC_IFR.SQRIF 变为 1，读取 ADC_ResultAcc 寄存器以获取转换结果累加值。

Step17：如需对其它通道进行转换，重复执行 Step11~Step16。

Step18：设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

通过外部触发启动 ADC 连续转换累加操作流程：

Step1：配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2：设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3：设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step6：设置 ADC_CR1.Mode 为 1，选择扫描转换模式。

Step7：设置 ADC_CR0.IE 为 1，使能 ADC 中断。

Step8：使能 NVIC 中断向量表中的 ADC 中断。

Step9：设置 ADC_CR1.RAccEn 为 1，使能 ADC 转换自动累加功能。

Step10：配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step11：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step12：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step13：配置 ADC_SQRx.CHxMux，选择顺序扫描转换通道。

Step14：配置 ADC_SQR2.CNT，选择顺序扫描转换的总转换次数。

注：总转换次数需与 Step12 中配置的转换通道数保持一致。

Step15：设置 ADC_IFR 为 0x0，清除 ADC 中断标志。

Step16：设置 ADC_CR1.RAccClr 为 0，清零 ADC_ResultAcc 寄存器。

Step17：配置 ADC_ExtTrigger0，选择外部触发条件。

Step18：当外部触发条件触发 ADC 完成转换时，ADC 模块会产生中断。用户可在 ADC 中断服务程序中读取 ADC_ResultAcc 寄存器以获取转换结果累加值。

Step19：如需对其它通道进行转换，重复执行 Step13~Step18。

Step20：设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

27.7ADC 转换外部触发源

ADC 转换既可通过软件配置启动，也可通过外部触发启动。

配置 ADC_ExtTrigger0 寄存器可设置 ADC 单次转换或顺序扫描转换的外部触发源。

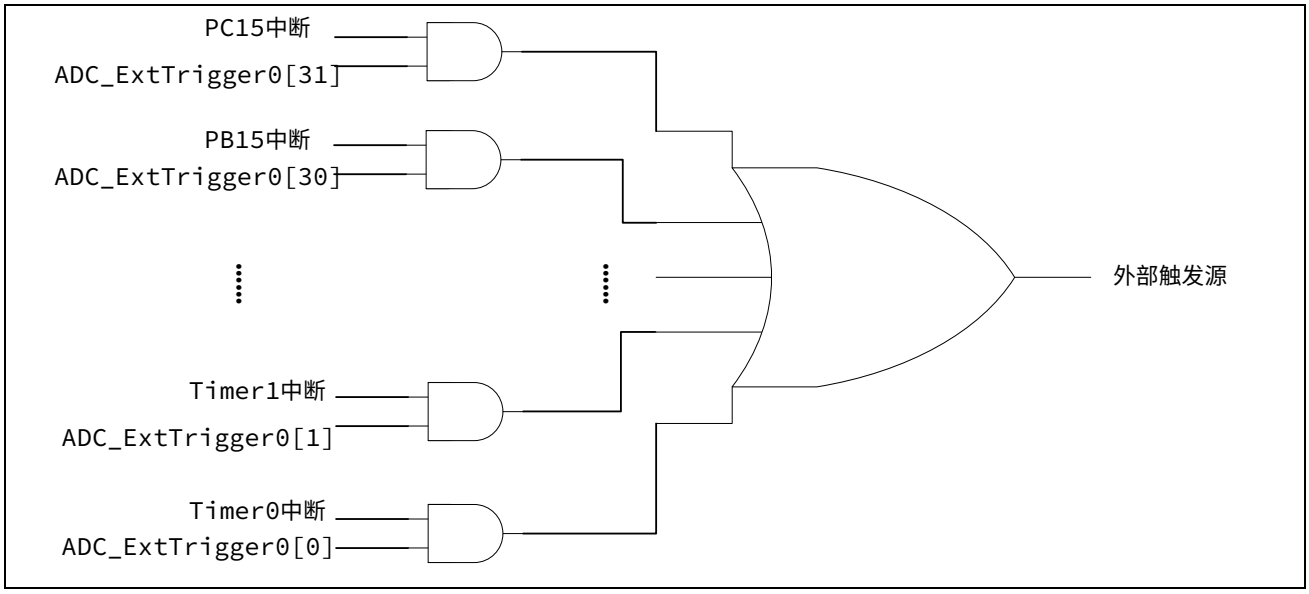


图 27-7　ADC 单次转换或顺序扫描转换外部触发源示意图

配置 ADC_ExtTrigger1 寄存器可设置 ADC 插队扫描转换的外部触发源。

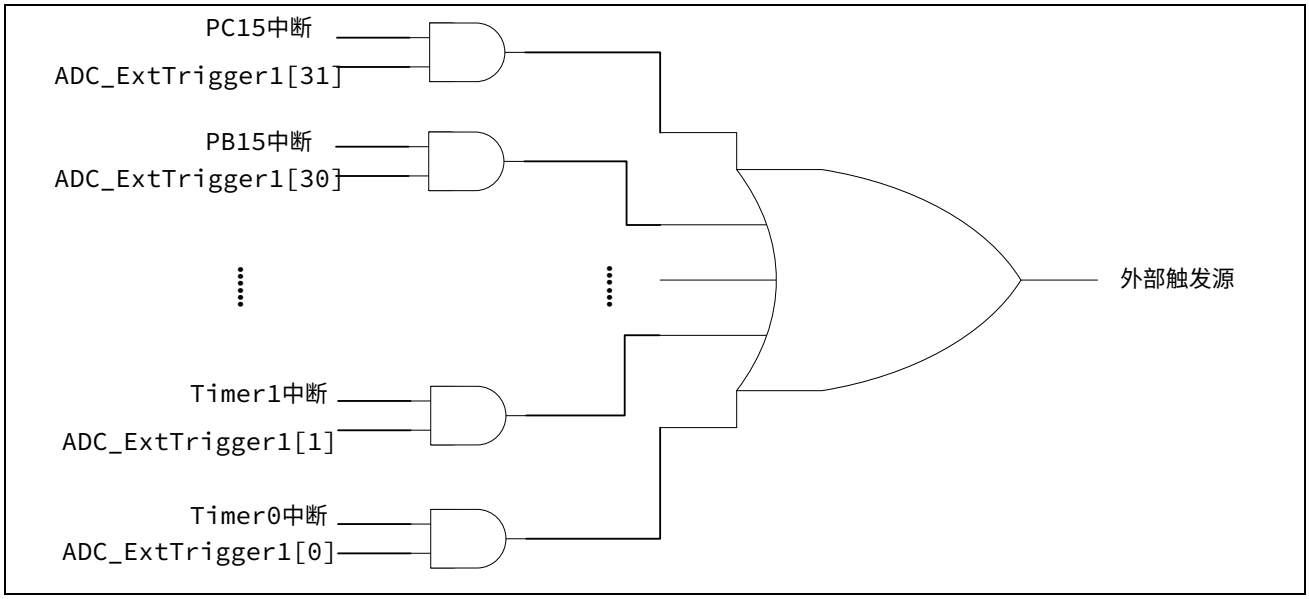


图 27-8　ADC 插队扫描转换外部触发源示意图

27.8ADC 转换结果比较

ADC 转换完成时，ADC 转换结果可以与用户设定的阈值进行比较，支持上阈值比较、下阈值比较、区间值比较。该功能需要将相应的控制位 HtCmp、LtCmp、RegCmp 置 1。该功能可实现对模拟量的自

动监测，直到 ADC 转换结果符合用户预期时才产生中断申请用户程序界入。监测通道选择通过 ADC_CR1.ThCh 进行配置。

上阈值比较：当 ADC 转换结果位于[ADC_HT，4095]区间内则 ADC_IFR.HTIF 置 1；向 ADC_ICR.HTIC 写入 0 则清零 ADC_IFR.HTIF。

下阈值比较：当 ADC 转换结果位于[0,ADC_LT)区间内则 ADC_IFR.LTIF 置 1；向 ADC_ICR.LTIC 写入 0 则清零 ADC_IFR.LTIF。

区间值比较：当 ADC 转换结果位于[ADC_LT，ADC_HT)区间内则 ADC_IFR.REGIF 置 1；向 ADC_ICR.REGIC 写入 0 则清零 ADC_IFR.REGIF。

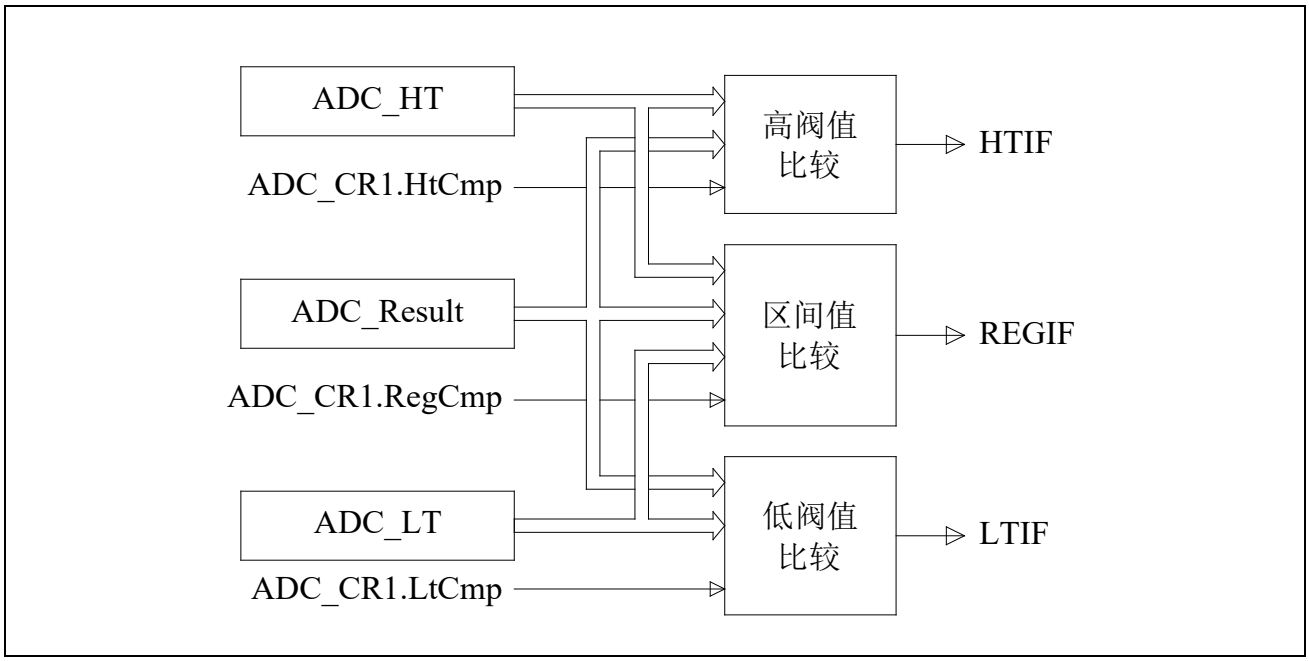


图 27-9 ADC 转换结果

27.9ADC 中断

ADC 中断请求如下表所示：

中断源	中断标志	中断使能
ADC 插队扫描转换完成	ADC_IFR.JQRIF	ADC_CR0.IE
ADC 顺序扫描转换完成	ADC_IFR.SQRIF	
ADC 转换结果位于区间值区域（大于等于下阈值小于上阈值）	ADC_IFR.REGIF	
ADC 转换结果位于上阈值区域(大于等于上阈值)	ADC_IFR.HTIF	
ADC 转换结果比较下阈值区域（小于下阈值）	ADC_IFR.LTIF	
ADC 单次转换完成	ADC_IFR.SGLIF	

27.10 使用温度传感器测量环境温度

温度传感器的输出电压会随环境温度的改变而变化，故根据温度传感器的输出电压即可计算出相应的环境温度。当 ADC 模块的测量通道选择温度传感器的输出电压时，即可测量环境温度。

计算公式如下：

环境温度 = 25 + 0.0854 × Vref × (AdcValue − Trim)

其中：Vref 为当前 ADC 模块的参考电压，取值为 1.5 或 2.5。

AdcValue 为 ADC 模块测量温度传感器输出电压的结果，取值为 0~4095。

Trim 为 16Bits 的校准值，计算时需要从 Flash 存储器中读出，其存放地址详见下表。

ADC 参考电压	校准值存放地址	校准值精度
内部 1.5V	0x00100C34	±3℃
内部 2.5V	0x00100C36	±3℃

计算示例如下：

条件 1：Vref=2.5、AdcValue=0x7E5、Trim=0x76C：

温度 1：25 + 0.0854 × 2.5 × (0x7E5 − 0x76C) = 50.8℃。

条件 2：Vref=1.5、AdcValue=0x72D、Trim=0x76C：

温度 2：25 + 0.0854 × 1.5 × (0x72D − 0x76C) = 16.9℃。

通过 ADC 测量环境温度操作流程：

Step1：设置 BGR_CR 为 3，使能 BGR 模块和温度传感器模块。

Step2：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step3：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step4：设置 ADC_CR1.Mode 为 0，选择单次转换模式。

Step5：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

Step6：配置 ADC_CR0.Ref，选择 ADC 的参考电压为内部 1.5V 或内部 2.5V。

Step7：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step8：设置 ADC_CR0.SGLMux 为 0x1C，选择待转换的通道为温度传感器的输出。

Step9：设置 ADC_CR0.Buf 为 1，使能输入信号放大器。

Step10：设置 ADC_ICR.SGLIC 为 0，清除 ADC_IFR.SGLIF 标志。

Step11：设置 ADC_SglStart.Start 为 1，启动 ADC 单次转换。

Step12：等待 ADC_IFR.SGLIF 变为 1，读取 ADC_Result 寄存器以获取 ADC 转换结果。

Step13：设置 ADC_CR0.En 及 BGR_CR 为 0，关闭 ADC 模块、BGR 模块、温度传感器模块。

Step14：读取温度传感器校准值，根据公式计算当前的环境温度。

27.11 ADC 模块寄存器

基地址 0x40002400

表 27-1 ADC 寄存器

寄存器	偏移地址	描述
ADC_CR0	0x004	ADC 基本配置寄存器 0
ADC_CR1	0x008	ADC 基本配置寄存器 1
ADC_SQR0	0x040	ADC 顺序扫描转换通道配置寄存器 0
ADC_SQR1	0x044	ADC 顺序扫描转换通道配置寄存器 1
ADC_SQR2	0x048	ADC 顺序扫描转换通道配置寄存器 2
ADC_JQR	0x04C	ADC 插队扫描转换通道配置寄存器
ADC_SqrResult0	0x050	ADC 顺序扫描转换通道 0 转换结果
ADC_SqrResult1	0x054	ADC 顺序扫描转换通道 1 转换结果
ADC_SqrResult2	0x058	ADC 顺序扫描转换通道 2 转换结果
ADC_SqrResult3	0x05C	ADC 顺序扫描转换通道 3 转换结果
ADC_SqrResult4	0x060	ADC 顺序扫描转换通道 4 转换结果
ADC_SqrResult5	0x064	ADC 顺序扫描转换通道 5 转换结果
ADC_SqrResult6	0x068	ADC 顺序扫描转换通道 6 转换结果
ADC_SqrResult7	0x06C	ADC 顺序扫描转换通道 7 转换结果
ADC_SqrResult8	0x070	ADC 顺序扫描转换通道 8 转换结果
ADC_SqrResult9	0x074	ADC 顺序扫描转换通道 9 转换结果
ADC_SqrResult10	0x078	ADC 顺序扫描转换通道 10 转换结果
ADC_SqrResult11	0x07C	ADC 顺序扫描转换通道 11 转换结果
ADC_SqrResult12	0x080	ADC 顺序扫描转换通道 12 转换结果
ADC_SqrResult13	0x084	ADC 顺序扫描转换通道 13 转换结果
ADC_SqrResult14	0x088	ADC 顺序扫描转换通道 14 转换结果
ADC_SqrResult15	0x08C	ADC 顺序扫描转换通道 15 转换结果
ADC_JqrResult0	0x090	ADC 插队扫描转换通道 0 转换结果
ADC_JqrResult1	0x094	ADC 插队扫描转换通道 1 转换结果
ADC_JqrResult2	0x098	ADC 插队扫描转换通道 2 转换结果
ADC_JqrResult3	0x09C	ADC 插队扫描转换通道 3 转换结果
ADC_Result	0x0A0	ADC 转换结果
ADC_ResultAcc	0x0A4	ADC 转换结果累加值
ADC_HT	0x0A8	ADC 比较上阈值
ADC_LT	0x0AC	ADC 比较下阈值
ADC_IFR	0x0B0	ADC 中断标志寄存器
ADC_ICR	0x0B4	ADC 中断清除寄存器
ADC_ExtTrigger0	0x0B8	ADC 单次转换或顺序扫描转换外部中断触发源配置寄存器

寄存器	偏移地址	描述
ADC_ExtTrigger1	0x0BC	ADC 插队扫描转换外部中断触发源配置寄存器
ADC_SglStart	0x0C0	ADC 单次转换启动控制寄存器
ADC_SqrStart	0x0C4	ADC 顺序扫描转换启动控制寄存器
ADC_JqrStart	0x0C8	ADC 插队扫描转换启动控制寄存器
ADC_allstart	0x0CC	ADC 一直转换启动停止控制寄存器

27.11.1 ADC 基本配置寄存器 0（ADC_CR0）

偏移地址 0x004

复位值 0x000067F0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE	InRefEn	SAM		Buf	Ref		SGLMux					CkDiv		CHMAP	En
RW	RW	RW		RW	RW		RW					RW		RW	RW

位	标记	功能描述	
31:16	Reserved	保留	
15	IE	ADC中断控制 1：使能中断 0：禁止中断	
14	InRefEn	ADC内部参考电压使能 1：使能内部参考电压 0：禁止内部参考电压	
13:12	SAM	ADC采样周期选择 00：4个转换周期 01：6个转换周期 10：8个转换周期 11：12个转换周期	
11	Buf	内置跟随器使能控制 0：关闭跟随器，外部输入信号与ADC直接相连。 1：打开跟随器，外部输入信号通过跟随器放大后与ADC相连，用于测量高阻信号。 注1：使能该功能时，最大速率为200Ksps 注2：以下条件必须使能该功能 待转换的信号输出阻抗较高 待转换的信号为：AVCC/3、内置温度传感器输出电压、DAC输出	
10:9	Ref	ADC参考电压选择 00：内部1.5V 01：内部2.5V 10：外部参考电压ExRef（PB01） 11：AVCC电压	
8:4	SGLMux	单次转换模式转换通道选择 (ADC_CR0.CHMAP=0) 00000：AIN0（PA00）	单次转换模式转换通道选择 (ADC_CR0.CHMAP=1) 00000：AIN0（PD08）

		00001：AIN1（PA01） 00010：AIN2（PA02） 00011：AIN3（PA03） 00100：AIN4（PA04） 00101：AIN5（PA05） 00110：AIN6（PA06） 00111：AIN7（PA07） 01000：AIN8（PB00） 01001：AIN9（PB01） 01010：AIN10（PC00） 01011：AIN11（PC01） 01100：AIN12（PC02） 01101：AIN13（PC03） 01110：AIN14（PC04） 01111：AIN15（PC05） 10000：AIN16（PB02） 10001：AIN17（PB10） 10010：AIN18（PB11） 10011：AIN19（PB12） 10100：AIN20（PB13） 10101：AIN21（PB14） 10110：AIN22（PB15） 10111：AIN23（PE15） 11000：AIN24（PC07）	00001：AIN1（PD09） 00010：AIN2（PD10） 00011：AIN3（PD11） 00100：AIN4（PA04） 00101：AIN5（PA05） 00110：AIN6（PE08） 00111：AIN7（PE09） 01000：AIN8（PE10） 01001：AIN9（PB01） 01010：AIN10（PE11） 01011：AIN11（PE12） 01100：AIN12（PE13） 01101：AIN13（PE14） 01110：AIN14（PC04） 01111：AIN15（PC05） 10000：AIN16（PB02） 10001：AIN17（PB10） 10010：AIN18（PB11） 10011：AIN19（PB12） 10100：AIN20（PB13） 10101：AIN21（PB14） 10110：AIN22（PB15） 10111：AIN23（PE15） 11000：AIN24（PC07）
		11001：DAC0输出　　注：ADC_CR0.Buf必须为1 11010：DAC1输出　　注：ADC_CR0.Buf必须为1 11011：AVCC/3　　注：ADC_CR0.Buf必须为1 11100：内置温度传感器输出电压　　注：ADC_CR0.Buf必须为1 11101：Reserved	
3:2	CkDiv	ADC时钟选择 00：PCLK时钟 01：PCLK时钟2分频 10：PCLK时钟4分频 11：PCLK时钟8分频	
1	CHMAP	OPA到DAC 输入映射 0：使用SGLMUX选择的端口通道 1：部分通道映射到OPA输出的相关端口	
0	En	ADC使能控制 1：使能ADC 0：禁止ADC	

CHMAP 为 1 时，以下 ADC 通道端口映射到其他端口。

AIN6M	PE08
AIN7M	PE09
AIN8M	PE10
AIN10M	PE11
AIN11M	PE12
AIN12M	PE13
AIN13M	PE14
AIN0M	PD08
AIN1M	PD09
AIN2M	PD10
AIN3M	PD11

27.11.2 ADC 基本配置寄存器 1（ADC_CR1）

偏移地址 0x008

复位值 0x00008000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAccClr	RegCmp	HtCmp	LtCmp	RAccEn	Mode	DmaJqr	DmaSqr	ThCh					Align	Reserved	
WO	RW	RW	RW	RW	RW	RW	RW	RW					RW		

位	标记	功能描述
31:16	Reserved	保留
15	RAccClr	ADC转换结果累加寄存器清零 1：无作用； 0：ADC转换结果累加寄存器（ADC_ResultAcc）清零。
14	RegCmp	ADC区间比较控制 1：使能区间比较 0：禁止区间比较
13	HtCmp	ADC高阈值比较控制 1：使能高阈值比较 0：禁止高阈值比较
12	LtCmp	ADC低阈值比较控制 1：使能低阈值比较 0：禁止低阈值比较
11	RAccEn	ADC转换结果自动累加控制 1：使能ADC转换结果自动累加功能 0：禁止ADC转换结果自动累加功能
10	Mode	ADC转换模式选择 1：扫描转换模式 0：单次转换模式
9	DmaJqr	插队扫描触发DMA读取控制 1：使能插队扫描转换触发DMA读取 0：禁止插队扫描转换触发DMA读取
8	DmaSqr	顺序扫描触发DMA读取控制 1：使能顺序扫描转换触发DMA读取 0：禁止顺序扫描转换触发DMA读取
7:3	ThCh	阈值比较通道选择 00000：选择通道0进行阈值比较

		00001：选择通道1进行阈值比较 00010：选择通道2进行阈值比较 …… 11101：选择通道29进行阈值比较
2	Align	转换结果对齐控制 1：转换结果16Bits左对齐存储 0：转换结果16Bits右对齐存储
1:0	Reserved	保留

27.11.3 ADC 顺序扫描转换通道配置寄存器 0（ADC_SQR0）

偏移地址 0x040

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CH5Mux					CH4Mux					CH3Mux			
		RW					RW					RW			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH2Mux					CH1Mux					CH0Mux				
	RW					RW					RW				

位	标记	功能描述
31:30	Reserved	保留
29:25	CH5Mux	顺序扫描转换通道5选择，设置参见ADC_CR0.SGLMux
24:20	CH4Mux	顺序扫描转换通道4选择，设置参见ADC_CR0.SGLMux
19:15	CH3Mux	顺序扫描转换通道3选择，设置参见ADC_CR0.SGLMux
14:10	CH2Mux	顺序扫描转换通道2选择，设置参见ADC_CR0.SGLMux
9:5	CH1Mux	顺序扫描转换通道1选择，设置参见ADC_CR0.SGLMux
4:0	CH0Mux	顺序扫描转换通道0选择，设置参见ADC_CR0.SGLMux

27.11.4 ADC 顺序扫描转换通道配置寄存器 1（ADC_SQR1）

偏移地址 0x044

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CH11Mux					CH10Mux					CH9Mux			
		RW					RW					RW			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH8Mux					CH7Mux					CH6Mux				
	RW					RW					RW				

位	标记	功能描述
31:30	Reserved	保留
29:25	CH11Mux	顺序扫描转换通道11选择，设置参见ADC_CR0.SGLMux
24:20	CH10Mux	顺序扫描转换通道10选择，设置参见ADC_CR0.SGLMux
19:15	CH9Mux	顺序扫描转换通道9选择，设置参见ADC_CR0.SGLMux
14:10	CH8Mux	顺序扫描转换通道8选择，设置参见ADC_CR0.SGLMux

9:5	CH7Mux	顺序扫描转换通道7选择，设置参见ADC_CR0.SGLMux
4:0	CH6Mux	顺序扫描转换通道6选择，设置参见ADC_CR0.SGLMux

27.11.5 ADC 顺序扫描转换通道配置寄存器 2（ADC_SQR2）

偏移地址 0x048

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CNT				CH15Mux			
								RW				RW			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH14Mux					CH13Mux					CH12Mux				
	RW					RW					RW				

位	标记	功能描述
31:24	Reserved	保留
23:20	CNT	顺序扫描转换次数
19:15	CH15Mux	顺序扫描转换通道15选择，设置参见ADC_CR0.SGLMux
14:10	CH14Mux	顺序扫描转换通道14选择，设置参见ADC_CR0.SGLMux
9:5	CH13Mux	顺序扫描转换通道13选择，设置参见ADC_CR0.SGLMux
4:0	CH12Mux	顺序扫描转换通道12选择，设置参见ADC_CR0.SGLMux

27.11.6 ADC 插队扫描转换通道配置寄存器（ADC_JQR）

偏移地址 0x04C

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										CNT		CH3Mux			
										RW		RW			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH2Mux					CH1Mux				CH0Mux					
	RW					RW				RW					

位	标记	功能描述
31:24	Reserved	保留
21:20	CNT	插队扫描转换次数
19:15	CH3Mux	插队扫描转换通道3选择，设置参见ADC_CR0.SGLMux
14:10	CH2Mux	插队扫描转换通道2选择，设置参见ADC_CR0.SGLMux
9:5	CH1Mux	插队扫描转换通道1选择，设置参见ADC_CR0.SGLMux
4:0	CH0Mux	插队扫描转换通道0选择，设置参见ADC_CR0.SGLMux

27.11.7 ADC 顺序扫描转换通道 x 转换结果（ADC_SqrResult0 - 15）

偏移地址 0x050 ~ 0x8C

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				Result											
				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	Result	ADC 顺序扫描转换通道x转换结果

27.11.8 ADC 插队扫描转换通道 x 转换结果（ADC_JqrResult0 - 3）

偏移地址 0x090 ~ 0x9C

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				Result											
				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	Result	ADC 插队扫描转换通道x转换结果

27.11.9 ADC 转换结果（ADC_Result）

偏移地址 0x0A0

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				Result											
				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	Result	ADC转换结果

27.11.10 ADC 转换结果累加值（ADC_ResultAcc）

偏移地址 0x0A4

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												ResultAcc[19:16]			
												RO			

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

位	标记	功能描述
31:20	Reserved	保留
19:0	ResultAcc	ADC转换累加值

27.11.11 ADC 比较上阈值（ADC_HT）

偏移地址 0x0A8

复位值 0x00000FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				HT											
				RW											

位	标记	功能描述
31:12	Reserved	保留
11:0	HT	ADC转换结果比较上阈值

27.11.12 ADC 比较下阈值（ADC_LT）

偏移地址 0x0AC

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	标记	功能描述
31:12	Reserved	保留
11:0	LT	ADC转换结果比较下阈值

27.11.13 ADC 中断标志寄存器（ADC_IFR）

偏移地址 0x0B0

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										JQRIF	SQRIF	REGIF	HHT_INTF	LLT_INTF	S_INTF
										RO	RO	RO	RO	RO	RO

位	标记	功能描述
31:6	Reserved	保留
5	JQRIF	ADC插队扫描转换完成标志 1：ADC插队扫描转换完成 0：ADC插队扫描转换未完成
4	SQRIF	ADC顺序扫描转换完成标志 1：ADC顺序扫描转换完成 0：ADC顺序扫描转换未完成
3	REGIF	ADC转换结果比较区间标志 1：ADC转换结果位于[ADC_LT ，ADC_HT)区间内 0：ADC转换结果位于[ADC_LT ，ADC_HT)区间外
2	HTIF	ADC转换结果比较上阈值标志 1：ADC转换结果位于[ADC_HT，4095]区间内 0：ADC转换结果位于[ADC_HT，4095]区间外
1	LTIF	ADC转换结果比较下阈值标志 1：ADC转换结果位于[0，ADC_LT)区间内 0：ADC转换结果位于[0，ADC_LT)区间外
0	SGLIF	ADC单次转换完成标志 1：ADC单次转换完成 0：ADC单次转换未完成

27.11.14 ADC 中断清除寄存器（ADC_ICR）

偏移地址 0x0B4

复位值 0x0000003F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										JQRIC	SQRIC	REGIC	HHT_INTC	LLT_INTC	S_INTC
										R1W0	R1W0	R1W0	R1W0	R1W0	R1W0

位	标记	功能描述
31:6	Reserved	保留
5	JQRIC	写0清除ADC插队扫描转换完成标志 写1无作用
4	SQRIC	写0清除ADC顺序扫描转换完成标志 写1无作用
3	REGIC	写0清除ADC转换结果比较区间标志 写1无作用
2	HTIC	写0清除ADC转换结果比较上阈值 写1无作用
1	LTIC	写0清除ADC转换结果比较下阈值标志 写1无作用
0	SGLIC	写0清除ADC单次转换完成标志 写1无作用

27.11.15 ADC 单次转换或顺序扫描转换外部中断触发源配置寄存器（ADC_ExtTrigger0）

偏移地址 0x0B8

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC15	PB15	PA15	PC11	PB11	PA11	PD07	PC07	PB07	PA07	PD03	PC03	PB03	PA03	DMA	SPI1
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI0	PCA	Res.	VC1	VC0	LPUART1	LPUART0	UART1	UART0	TIM6	TIM5	TIM4	TIM3	TIM2	TIM1	TIM0
RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31	PC15	PC15中断触发ADC转换
30	PB15	PB15中断触发ADC转换
29	PA15	PA15中断触发ADC转换
28	PC11	PC11中断触发ADC转换
27	PB11	PB11中断触发ADC转换
26	PA11	PA11中断触发ADC转换
25	PD07	PD07中断触发ADC转换
24	PC07	PC07中断触发ADC转换
23	PB07	PB07中断触发ADC转换
22	PA07	PA07中断触发ADC转换
21	PD03	PD03中断触发ADC转换
20	PC03	PC03中断触发ADC转换
19	PB03	PB03中断触发ADC转换
18	PA03	PA03中断触发ADC转换
17	DMA	DMA中断触发ADC转换
16	SPI1	SPI1中断触发ADC转换
15	SPI0	SPI0中断触发ADC转换
14	PCA	PCA中断触发ADC转换
13	Reserved	保留位
12	VC1	VC1中断触发ADC转换
11	VC0	VC0中断触发ADC转换
10	LPUART1	LPUART1中断触发ADC转换
9	LPUART0	LPUART0中断触发ADC转换
8	UART1	UART1中断触发ADC转换

7	UART0	UART0中断触发ADC转换
6	TIM6	Timer6中断触发ADC转换
5	TIM5	Timer5中断触发ADC转换
4	TIM4	Timer4中断触发ADC转换
3	TIM3	Timer3中断触发ADC转换
2	TIM2	Timer2中断触发ADC转换
1	TIM1	Timer1中断触发ADC转换
0	TIM0	Timer0中断触发ADC转换

Notes:

1) TIM4/5/6 中断触发 ADC 自动转换，除了需要使能 TIM4/5/6 的相应中断外，还需要配置 Advanced Timer 的展频及中断触发选择寄存器 TIMX_CR 选择可以触发 ADC 的中断源。
2) 触发 ADC 使用的是各中断标志位的上升沿。如果需要重复触发，需要清除中断标志。如果不需要进入中断服务程序，请不要使能 NVIC 的中断使能。

27.11.16 ADC 插队扫描转换外部中断触发源配置寄存器（ADC_ExtTrigger1）

偏移地址 0x0BC

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC15	PB15	PA15	PC11	PB11	PA11	PD07	PC07	PB07	PA07	PD03	PC03	PB03	PA03	DMA	SPI1
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI0	PCA	Res.	VC1	VC0	LPUART1	LPUART0	UART1	UART0	TIM6	TIM5	TIM4	TIM3	TIM2	TIM1	TIM0
RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31	PC15	PC15中断触发ADC转换
30	PB15	PB15中断触发ADC转换
29	PA15	PA15中断触发ADC转换
28	PC11	PC11中断触发ADC转换
27	PB11	PB11中断触发ADC转换
26	PA11	PA11中断触发ADC转换
25	PD07	PD07中断触发ADC转换
24	PC07	PC07中断触发ADC转换
23	PB07	PB07中断触发ADC转换
22	PA07	PA07中断触发ADC转换
21	PD03	PD03中断触发ADC转换
20	PC03	PC03中断触发ADC转换
19	PB03	PB03中断触发ADC转换
18	PA03	PA03中断触发ADC转换
17	DMA	DMA中断触发ADC转换
16	SPI1	SPI1中断触发ADC转换
15	SPI0	SPI0中断触发ADC转换
14	PCA	PCA中断触发ADC转换
13	Reserved	保留位
12	VC1	VC1中断触发ADC转换
11	VC0	VC0中断触发ADC转换
10	LPUART1	LPUART1中断触发ADC转换
9	LPUART0	LPUART0中断触发ADC转换
8	UART1	UART1中断触发ADC转换

7	UART0	UART0中断触发ADC转换
6	TIM6	Timer6中断触发ADC转换
5	TIM5	Timer5中断触发ADC转换
4	TIM4	Timer4中断触发ADC转换
3	TIM3	Timer3中断触发ADC转换
2	TIM2	Timer2中断触发ADC转换
1	TIM1	Timer1中断触发ADC转换
0	TIM0	Timer0中断触发ADC转换

Notes:

1) TIM4/5/6 中断触发 ADC 自动转换，除了需要使能 TIM4/5/6 的相应中断外，还需要配置 Advanced Timer 的展频及中断触发选择寄存器 TIMX_CR 选择可以触发 ADC 的中断源。
2) 触发 ADC 使用的是各中断标志位的上升沿。如果需要重复触发，需要清除中断标志。如果不需要进入中断服务程序，请不要使能 NVIC 的中断使能。

27.11.17 ADC 单次转换启动控制寄存器（ADC_SglStart）

偏移地址 0x0C0

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															Start
															RW

位	标记	功能描述
31:1	Reserved	保留
0	Start	ADC单次转换启动控制 1：启动ADC单次转换 0：停止ADC单次转换

27.11.18 ADC 顺序扫描转换启动控制寄存器（ADC_SqrStart）

偏移地址 0x0C4

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															Start
															RW

位	标记	功能描述
31:1	Reserved	保留
0	Start	ADC顺序扫描转换启动控制 1：启动ADC顺序扫描转换 0：停止ADC顺序扫描转换

27.11.19 ADC 插队扫描转换启动控制寄存器（ADC_JqrStart）

偏移地址 0x0C8

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															Start
															RW

位	标记	功能描述
31:1	Reserved	保留
0	Start	ADC插队扫描转换启动控制 1：启动ADC插队扫描转换 0：停止ADC插队扫描转换

27.11.20 ADC 一直转换启动停止控制寄存器（ADC_allStart）

偏移地址 0x0CC

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															Start
															RW

位	标记	功能描述
31:1	Reserved	保留
0	Start	ADC插队扫描转换启动控制 1：启动ADC一直转换 0：停止ADC一直转换,需要等待到当次转换结束 读0无转换；1正在转换

28 数模转换器（DAC）

DAC 模块是 12 位电压输出数模转换器。DAC 可以按 8 位或 12 位模式进行配置，并且可与 DMA 控制器配合使用。在 12 位模式下，数据可以采用左对齐或右对齐。可使用输入参考电压 VREF+（与 ADC 共享）。可以有选择地缓冲输出以实现更高的电流驱动。

28.1DAC 特性

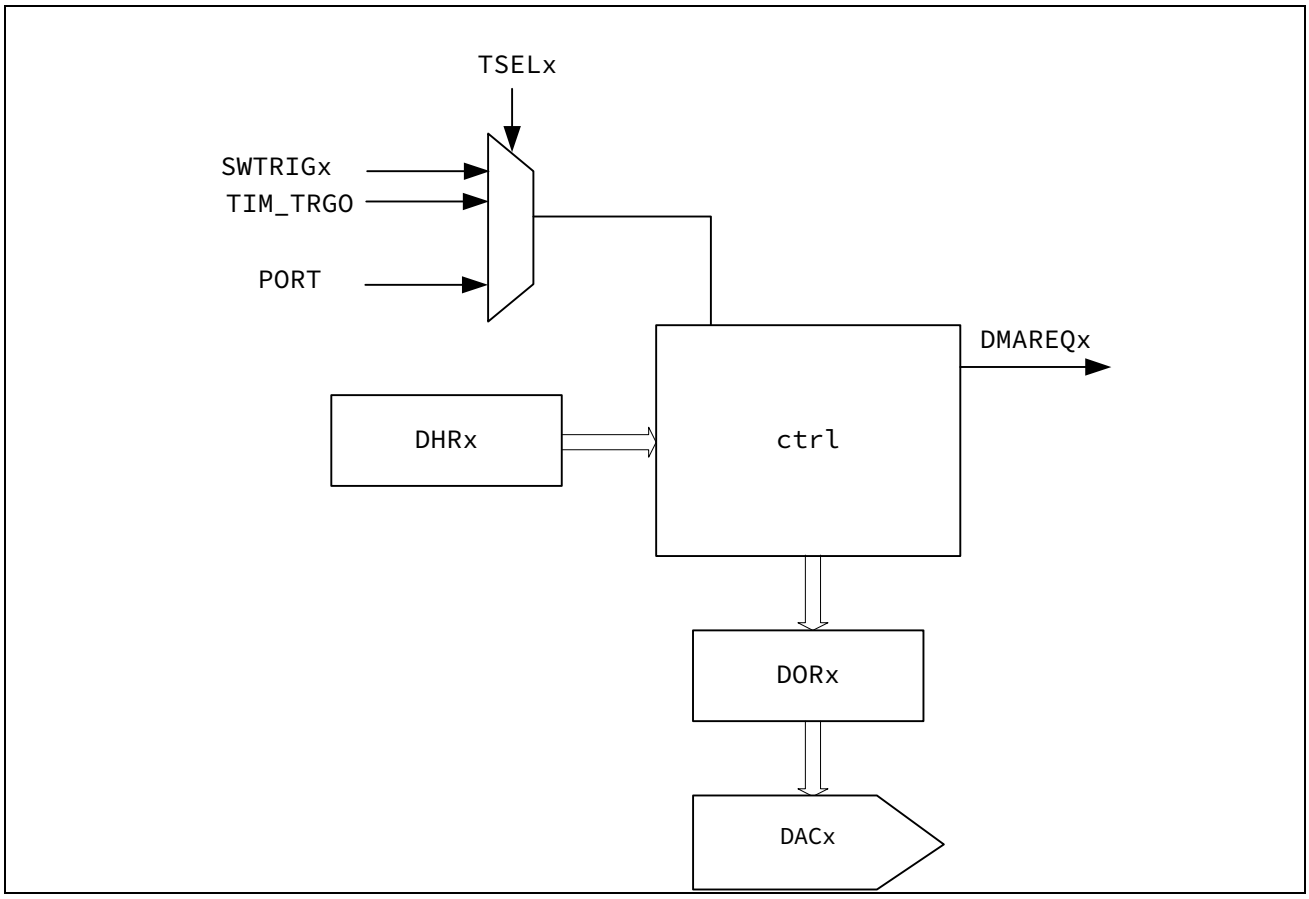
器件集成了两个 DAC 转换器，各具有一个输出通道：DAC0_OUT 和 DAC1_OUT。

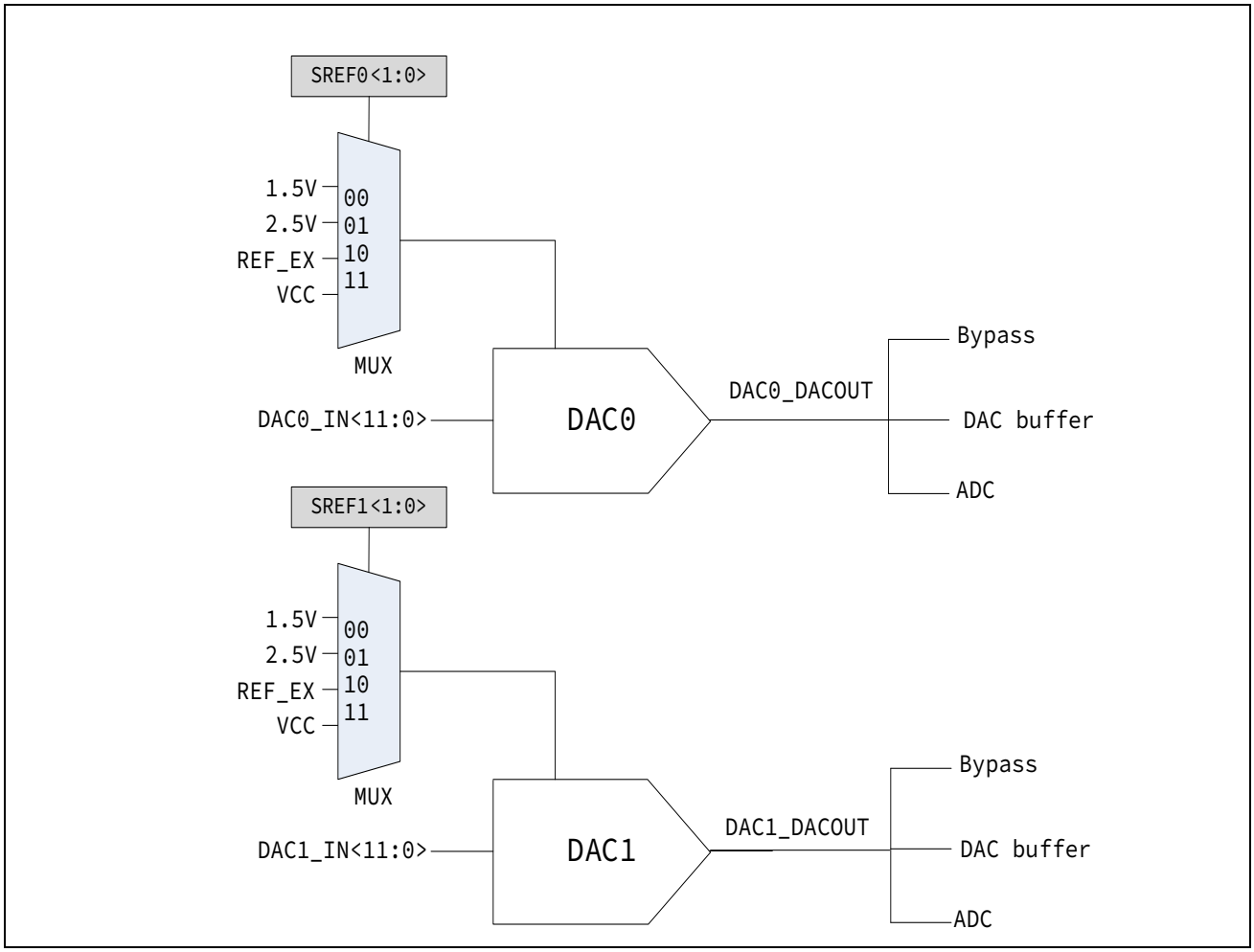
DAC 主要特性如下：

- 一个数据保持寄存器
- 12 位模式下数据采用左对齐或右对齐
- 同步更新功能
- 生成噪声波
- 生成三角波
- 双 DAC 通道，支持单独或同时转换
- DMA 功能（包括下溢检测）
- 通过外部触发信号进行转换
- 4 种参考源：AVCC 电压、ExRef 引脚、内置 1.5v 参考电压、内置 2.5v 参考电压；

28.2功能说明

28.2.1 框图





28.2.2 DAC 输出缓冲器使能

DAC 可以使用输出缓冲器，可用来降低输出阻抗并在不增加外部运算放大器的情况下直接驱动外部负载。关于输出缓存器的设置参考<运算放大器 OPA>章节

通过 DAC_CR 寄存器中的 BOFF1 位，可使能或禁止 DAC 通道输出缓冲器。

28.2.3 DAC 通道使能

将 DAC_CR 寄存器中的相应 ENx 位置 1，即可使能对应 DAC 通道。经过一段启动时间 t_{WAKEUP} 后，各 DAC 通道被真正使能。

ENx 位只会使能模拟 DAC 通道 x 单元。即使 ENx 位复位，DAC 通道 x 数字接口仍处于使能状态。

28.2.4 DAC 输出电压

经过线性转换后，数字输入会转换为 0 到 VREF+ 之间的输出电压。各 DAC 通道引脚的模拟输出电压通过以下公式确定：

$$DAC_{out} = V_{ref} * DOR/4096$$

28.2.5 DAC 触发选择

如果 TENx 控制位置 1，可通过外部事件（定时计数器、外部中断线）或软件触发转换。TSELx[2:0] 控制位将决定哪个可能事件将触发转换，如下表所示。

TSEL	000	001	010	011	100	101	110	111
触发源	TIM0_ TRADC	TIM1_ TRADC	TIM2_ TRADC	TIM3_ TRADC	TIM4_ TRADC	TIM5_ TRADC	SW TRIG	EXTI

每当 DAC 所选定时器 TRGO 输出或所选外部端口触发上检测到上升沿时，DAC_DHRx 寄存器中存储的最后一个数据即会转移到 DAC_DORx 寄存器中。发生触发后再经过三个 APB1 周期，DAC_DORx 寄存器将会得到更新。

如果选择软件触发，一旦 SWTRIG 位置 1，转换即会开始。DAC_DHRx 寄存器内容加载到 DAC_DORx 寄存器中后，SWTRIG 即由硬件复位。

注意：

- ENx 位置 1 时，无法更改 TSELx[2:0] 位。如果选择软件触发，DAC_DHRx 寄存器的内容只需一个 APB1 时钟周期即可转移到 DAC_DORx 寄存器。

28.2.6 单模式功能说明

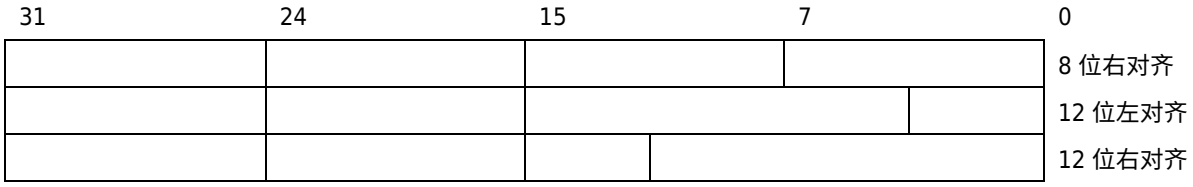
28.2.6.1 DAC 数据格式

根据所选配置模式，数据必须按如下方式写入指定寄存器：

存在三种可能：

- 8 位右对齐：软件必须将数据加载到 DAC_DHR8Rx [7:0] 位（存储到 DHRx[11:4] 位）
- 12 位左对齐：软件必须将数据加载到 DAC_DHR12Lx [15:4] 位（存储到 DHRx[11:0] 位）
- 12 位右对齐：软件必须将数据加载到 DAC_DHR12Rx [11:0] 位（存储到 DHRx[11:0] 位）

根据加载的 DAC_DHRyyyx 寄存器，用户写入的数据将移位并存储到相应的 DHRx（数据保持寄存器 x，即内部非存储器映射寄存器）。之后，DHRx 寄存器将被自动加载，或者通过软件或外部事件触发加载到 DORx 寄存器。



28.2.6.2 DAC 转换

DAC_DORx 无法直接写入，任何数据都必须通过加载 DAC_DHRx 寄存器（写入 DAC_DHR8Rx、DAC_DHR12Lx、DAC_DHR12Rx）才能传输到 DAC 通道 x。

如果未选择触发（DAC_CR 寄存器中的 TENx 位复位），那么经过一个 APB1 时钟周期后，DAC_DHRx 寄存器中存储的数据将自动转移到 DAC_DORx 寄存器。但是，如果选择硬件触发（置位 DAC_CR 寄存器中的 TENx 位）且触发条件到来，将在三个 PCLK1 时钟周期后进行转移。

当 DAC_DORx 加载了 DAC_DHRx 内容时，模拟输出电压将在一段时间 tSETTLING 后可用，具体时间取决于电源电压和模拟输出负载。

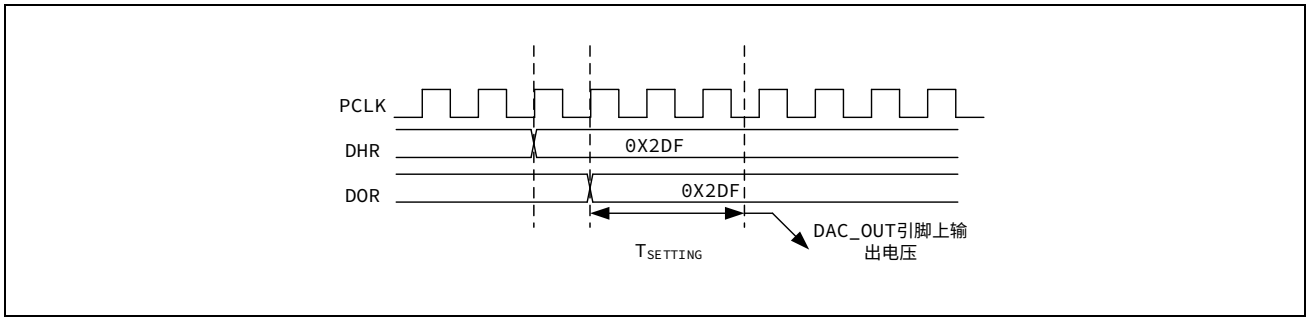


图 28-1 触发不使能时 DAC 转换时序图

生成单个 LFSR（线性反馈移位寄存器）的独立触发

要将 DAC 配置为此转换模式（请参见 28.2.8 噪音生成），需要遵循以下顺序：

1. 将 DAC 通道触发使能位 TENx 置 1。
2. 通过设置 TSELx[2:0] 位配置触发源。
3. 将 DAC 通道的 WAVEx[1:0] 位配置为“01”，并在 MAMPx[3:0] 位中配置相同的 LFSR 掩码值
4. 将 DAC 通道数据加载到所需 DAC_DHRx 寄存器（DHR12RD、DHR12LD 或 DHR8RD）。

DAC 通道 x 触发信号到达时，LFSRx 计数器内容（使用相同的掩码）与 DHRx 寄存器内容相加，所得总和转移到 DAC_DORx 中（三个 APB 时钟周期之后）。LFSRx 计数器随即更新。

生成单个三角波的独立触发

要将 DAC 配置为此转换模式（请参见 28.2.9 三角波生成），需要遵循以下顺序：

1. 将 DAC 通道 x 触发使能位 TENx 置 1。
2. 通过设置 TSELx[2:0] 位配置触发源。
3. 将 DAC 通道 x 的 WAVEx[1:0] 位配置为“1x”，并在 MAMPx[3:0] 位中配置相同的最大振幅值
4. 将 DAC 通道 x 数据加载到所需 DAC_DHRx 寄存器（DHR12RD、DHR12LD 或 DHR8RD）。

DAC 通道 x 触发信号到达时，DAC 通道 x 三角波计数器内容（使用相同的三角波振幅）与 DHRx 寄存器内容相加，所得总和转移到 DAC_DORx 中（三个 APB 时钟周期之后）。DAC 通道 x 三角波计数器随即更新。

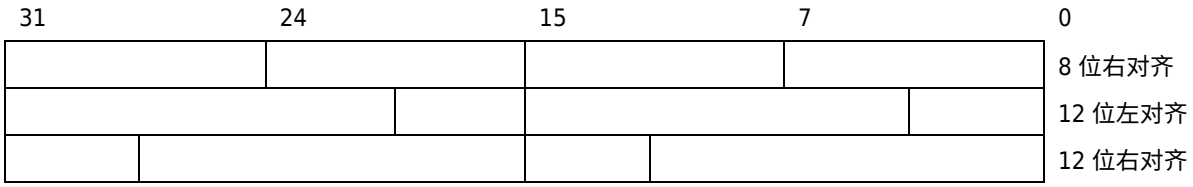
28.2.7 双模式功能说明

28.2.7.1 DAC 数据格式

在 DAC 双通道模式下，有三种可能的方式：

- 8 位右对齐：将 DAC0 通道的数据加载到 DAC_DHR8RD [7:0] 位（存储到 DHR0[11:4]位），将 DAC1 通道的数据加载到 DAC_DHR8RD [15:8] 位（存储到 DHR1[11:4] 位）
- 12 位左对齐：将 DAC0 通道的数据加载到 DAC_DHR12RD [15:4] 位（存储到 DHR0[11:0]位），将 DAC1 通道的数据加载到 DAC_DHR12RD [31:20] 位（存储到 DHR1[11:0] 位）
- 12 位右对齐：将 DAC0 通道的数据加载到 DAC_DHR12RD [11:0] 位（存储到 DHR0[11:0]位），将 DAC1 通道的数据加载到 DAC_DHR12RD [27:16] 位（存储到 DHR1[11:0] 位）

根据加载的 DAC_DHRyyyD 寄存器，用户写入的数据将移位并存储到 DHR0 和 DHR1（数据保持寄存器，即内部非存储器映射寄存器）。之后，DHR0 和 DHR1 寄存器将被自动加载，或者通过软件或外部事件触发分别被加载到 DOR0 和 DOR1 寄存器。



28.2.7.2 DAC 转换

双模式下的 DAC 通道转换执行方式与单模式下基本相同，必须通过写入 DAC_DHR8Rx、DAC_DHR12Lx、DAC_DHR12Rx、DAC_DHR8RD、DAC_DHR12LD 或 DAC_DHR12RD 加载数据。

28.2.7.3 DAC 双转换模式说明

为了在同时需要两个 DAC 通道的应用中有效利用总线带宽，DAC 模块有三个双寄存器可供操作：DHR8RD、DHR12RD 和 DHR12LD。这样，只需一个寄存器访问即可同时驱动两个 DAC 通道。

通过两个 DAC 通道和这三个双寄存器可以实现 11 种转换模式。但如果需要，所有这些转换模式也都可以通过单独的 DHRx 寄存器来实现。

下面几段内容将介绍所有这些模式。

独立触发（不产生波形）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

1. 将两个 DAC 通道触发使能位 TEN0 和 TEN1 置 1
2. 将 TSEL1[2:0] 和 TSEL0[2:0] 设置为不同的值，以配置不同的触发源
3. 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）

DAC0 通道触发信号到达时，DHR0 寄存器的内容转移到 DAC_DOR9（三个 APB 时钟周期之后）。

DAC 1 通道触发信号到达时，DHR1 寄存器的内容转移到 DAC_DOR1（三个 APB 时钟周期之后）。

独立触发（生成单个 LFSR）

要将 DAC 配置为此转换模式（请参见 28.2.8 噪音生成），需要遵循以下顺序：

1. 将两个 DAC 通道触发使能位 TEN1 和 TEN0 置 1
2. 将 TSEL1[2:0] 和 TSEL0[2:0] 设置为不同的值，以配置不同的触发源
3. 将两个 DAC 通道的 WAVEx[1:0] 设置为“01”，并在 MAMPx[3:0] 位中配置相同的 LFSR 掩码值
4. 将 DAC 双通道数据加载到所需 DHR 寄存器（DHR12RD、DHR12LD 或 DHR8RD）

DAC 通道 0 触发信号到达时，LFSR0 计数器内容（使用相同的掩码）与 DHR0 寄存器内容相加，所得总和转移到 DAC_DOR0 中（三个 APB 时钟周期之后）。LFSR0 计数器随即更新。

DAC 通道 1 触发信号到达时，LFSR1 计数器内容（使用相同的掩码）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB 时钟周期之后）。LFSR1 计数器随即更新。

独立触发（生成不同 LFSR）

要将 DAC 配置为此转换模式（请参见 28.2.8 噪音生成），需要遵循以下顺序：

1. 将两个 DAC 通道触发使能位 TEN1 和 TEN0 置 1
2. 将 TSEL1[2:0] 和 TSEL0[2:0] 设置为不同的值，以配置不同的触发源
3. 将两个 DAC 通道的 WAVEx[1:0] 设置为“01”，并在 MAMP1[3:0] 和 MAMP0[3:0] 位中设置不同的 LFSR 掩码值
4. 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）

DAC 通道 0 触发信号到达时，LFSR0 计数器内容（使用 MAMP0[3:0] 配置的掩码）与 DHR0 寄存器内容相加，所得总和转移到 DAC_DOR0 中（三个 APB 时钟周期之后）。LFSR0 计数器随即更新。

DAC 通道 1 触发信号到达时，LFSR1 计数器内容（使用 MAMP1[3:0] 配置的掩码）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB 时钟周期之后）。LFSR1 计数器随即更新。

独立触发（生成单个三角波）

要将 DAC 配置为此转换模式（请参见 28.2.9 三角波生成），需要遵循以下顺序：

1. 将 DAC 通道 x 触发使能位 TENx 置 1。
2. 通过在 TSELx[2:0] 位中设置不同的值，以配置不同的触发源
3. 将 DAC 通道 x 的 WAVEx[1:0] 位配置为“1x”，并在 MAMPx[3:0] 位中配置相同的最大振幅值
4. 将 DAC 通道 x 数据加载到所需 DAC_DHRx 寄存器

有关为 DAC 转换计时的 AHB 总线（APB 或 APB1）的详细信息，请参见 第 15.5.2 节：DAC 通道转换。

DAC 通道 x 触发信号到达时，DAC 通道 x 三角波计数器内容（使用相同的三角波振幅）与 DHRx 寄存器内容相加，所得总和转移到 DAC_DORx 中（三个 APB 时钟周期之后）。DAC 通道 x 三角波计数器随即更新。

独立触发（生成不同三角波）

要将 DAC 配置为此转换模式（请参见 28.2.9 三角波生成），需要遵循以下顺序：

1. 将 DAC 通道 x 触发使能位 TENx 置 1。
2. 通过在 TSELx[2:0] 位中设置不同的值，以配置不同的触发源
3. 将 DAC 通道 x 的 WAVEx[1:0] 位设置为“1x”，并在 MAMPx[3:0] 位中设置不同的最大振

幅值

4. 将 DAC 通道 x 数据加载到所需 DAC_DHRx 寄存器

DAC 通道 x 触发信号到达时，DAC 通道 x 三角波计数器内容（使用 MAMPx[3:0] 配置的三角波振幅）与 DHRx 寄存器内容相加，所得总和转移到 DAC_DORx 中（三个 APB 时钟周期之后）。DAC 通道 x 三角波计数器随即更新。

同步软件启动

要将 DAC 配置为此转换模式，需要遵循以下顺序：

1. 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）

该配置用一个 APB 时钟周期。

同步触发（不产生波形）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

1. 将两个 DAC 通道触发使能位 TEN1 和 TEN0 置 1
2. 将 TSEL1[2:0] 和 TSEL0[2:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源
3. 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）

当触发信号到达时，DHR1 和 DHR0 寄存器内容将分别转移到 DAC_DOR1 和 DAC_DOR0 中（三个 APB 时钟周期之后）。

同步触发（生成单个 LFSR）

要将 DAC 配置为此转换模式（请参见 28.2.8 噪音生成），需要遵循以下顺序：

1. 将两个 DAC 通道触发使能位 TEN1 和 TEN0 置 1
2. 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源
3. 将两个 DAC 通道的 WAVEx[1:0] 设置为“01”，并在 MAMPx[3:0] 位中配置相同的 LFSR 掩码值
4. 将 DAC 双通道数据加载到所需 DHR 寄存器（DHR12RD、DHR12LD 或 DHR8RD）触发信号到达时，LFSR1 计数器内容（使用相同的掩码）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB 时钟周期之后）。LFSR1 计数器随即更新。同时，LFSR0 计数器内容（使用相同的掩码）与 DHR0 寄存器内容相加，所得总和转移到 DAC_DOR0 中（三个 APB 时钟周期之后）。LFSR0 计数器随即更新。

同步触发（生成不同 LFSR）

要将 DAC 配置为此转换模式（请参见 28.2.8 噪音生成），需要遵循以下顺序：

1. 将两个 DAC 通道触发使能位 TEN1 和 TEN0 置 1
2. 将 TSEL1[2:0] 和 TSEL0[2:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源
3. 将两个 DAC 通道的 WAVEx[1:0] 设置为“01”，并在 MAMP1[3:0] 和 MAMP2[3:0] 位中设置不同的 LFSR 掩码值
4. 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）触发信号到达时，LFSR1 计数器内容（使用 MAMP1[3:0] 配置的掩码）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB 时钟周期之后）。LFSR1 计数器随即更新。

   同时，LFSR0 计数器内容（使用 MAMP0[3:0] 配置的掩码）与 DHR0 寄存器内容相加，所得总和转移到 DAC_DOR0 中（三个 APB 时钟周期之后）。LFSR0 计数器随即更新。

同步触发（生成单个三角波）

要将 DAC 配置为此转换模式（请参见 28.2.9 三角波生成），需要遵循以下顺序：

1. 将 DAC 通道 x 触发使能位 TEN1x 置 1。
2. 在 TSELx[2:0] 位中设置相同的值，以便为两个 DAC 通道配置相同的触发源。
3. 将 DAC 通道 x 的 WAVEx[1:0] 位配置为“1x”，并使用 MAMPx[3:0] 位配置相同的最大振幅值
4. 将 DAC 通道 x 数据加载到所需 DAC_DHRx 寄存器。

触发信号到达时，DAC 通道 x 三角波计数器内容（使用相同的三角波振幅）与 DHRx 寄存器内容相加，所得总和转移到 DAC_DORx 中（三个 APB 时钟周期之后）。DAC 通道 x 三角波计数器随即更新。

同步触发（生成不同三角波）

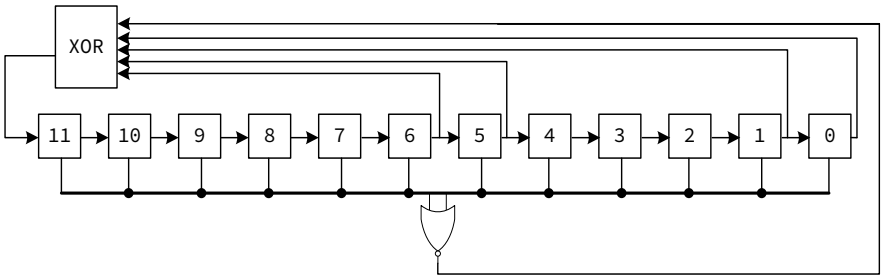
要将 DAC 配置为此转换模式（请参见 28.2.9 三角波生成），需要遵循以下顺序：

1. 将 DAC 通道 x 触发使能位 TENx 置 1。
2. 在 TSELx[2:0] 位中设置相同的值，以便为 DAC 通道 x 配置相同的触发源。
3. 将 DAC 通道 x 的 WAVEx[1:0] 位配置为“1x”，并在 MAMPx[3:0] 位中设置不同的最大振幅值。
4. 将 DAC 通道 x 数据加载到所需 DAC_DHRx 寄存器。

触发信号到达时，DAC 通道 x 三角波计数器内容（使用 MAMPx[3:0] 配置的三角波振幅）与 DHRx 寄存器内容相加，所得总和转移到 DAC_DORx 中（三个 APB 时钟周期之后）。DAC 通道 x 三角波计数器随即更新。

28.2.8 噪音生成

为了生成可变振幅的伪噪声，可使用 LFSR（线性反馈移位寄存器）。将 WAVEx[1:0] 置为“01”即可选择生成噪声。LFSR 中的预加载值为 0xAAA。在每次发生触发事件后，经过三个 APB 时钟周期，该寄存器会依照特定的计算算法完成更新。



LFSR 值可以通过 DAC_CR 寄存器中的 MAMPx[3:0] 位来部分或完全屏蔽，在不发生溢出的情况下，该值将与 DAC_DHRx 的内容相加，然后存储到 DAC_DORx 寄存器中。

如果 LFSR 为 0x0000，将向其注入“1”（防锁定机制）。

可以通过复位 WAVEx[1:0] 位来将 LFSR 波形产生功能关闭。

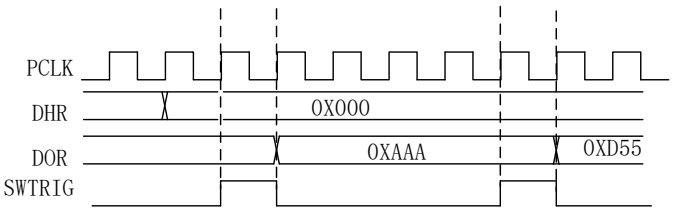


图 28-2 LFSR 产生波形的 DAC 转换（使用软件触发）

28.2.9 三角波生成

可以在直流电流或慢变信号上叠加一个小幅三角波。将 WAVEx[1:0] 置为“10”即可选择 DAC 生成三角波。振幅通过 DAC_CR 寄存器中的 MAMPx[3:0] 位进行配置。每次发生触发事件后，经过三个 APB 时钟周期，内部三角波计数器将会递增。在不发生溢出的情况下，该计数器的值将与 DAC_DHRx 寄存器内容相加，所得总和将存储到 DAC_DORx 寄存器中。只要小于 MAMPx[3:0] 位定义的最大振幅，三角波计数器就会一直递增。一旦达到配置的振幅，计数器将递减至零，然后再递增，以此类推。

可以通过复位 WAVEx[1:0] 位来将三角波产生功能关闭。

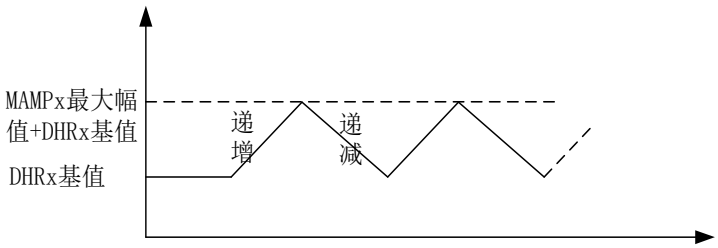


图 28-3 生成 DAC 三角波

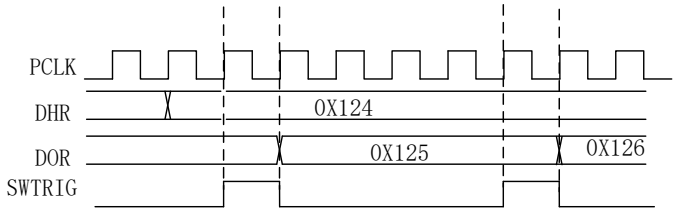


图 28-4 生成三角波波形的 DAC 转换（使能软件触发）

28.2.10 DMA 请求

每个 DAC 通道都具有 DMA 功能。两个 DMA 通道用于处理 DAC 通道的 DMA 请求。

当 DMAENx 位置 1 时，如果发生触发，则将产生 DAC DMA 请求。DAC_DHRx 寄存器的值随后转移到 DAC_DORx 寄存器。

在双通道模式下，如果两个 DMAENx 位均置 1，则将产生两个 DMA 请求。如果只需要一个 DMA 请求，用户应仅将相应 DMAENx 位置 1。这样，应用程序可以在双通道模式下通过一个 DMA 请求和一个特定 DMA 通道来管理两个 DAC 通道。

DMA 下溢

DAC DMA 请求没有缓冲队列。这样，如果第二个外部触发到达时尚未收到第一个外部触发的应答，将不会发出新的请求，并且 DAC_SR 寄存器中的 DAM 通道下溢标志 DMAUDRx 将置 1，以报告这一错误状况。DMA 数据传输随即禁止，并且不再处理其他 DMA 请求。DAC 通道仍将继续转换旧有数据。

软件应通过写入“0”来将 DMAUDRx 标志清零，将所用 DMA 数据流的 DMAEN 位清零，并重新初始化 DMA 和 DAC 通道，以便正确地重新开始 DMA 传输。软件应修改 DAC 触发转换频率或减轻 DMA 工作负载，以避免再次发生 DMA 下溢。最后，可通过使能 DMA 数据传输和转换触发来继续完成 DAC 转换。

对于各 DAC 通道，如果使能 DAC_CR 寄存器中相应的 DMAUDRIEx 位，还将产生中断。

28.3DAC 寄存器

基地址 0x40002500

表 28-1 DAC 寄存器

寄存器	偏移地址	描述
DAC_CR0	0x000	控制寄存器
DAC_SWTRIGR	0x004	软件触发寄存器
DAC_DHR12R0	0x008	DAC0 12位右对齐数据寄存器
DAC_DHR12L0	0x00C	DAC0 12位左对齐数据寄存器
DAC_DHR8R0	0X010	DAC0 8位右对齐数据寄存器
DAC_DHR12R1	0x014	DAC1 12位右对齐数据寄存器
DAC_DHR12L1	0x018	DAC1 12位左对齐数据寄存器
DAC_DHR8R1	0x01C	DAC1 8位右对齐数据寄存器
DAC_DHR12RD	0X020	DAC0/1 12位右对齐数据寄存器
DAC_DHR12LD	0x024	DAC0/1 12位左对齐数据寄存器
DAC_DHR8RD	0x028	DAC0/1 8位右对齐数据寄存器
DAC_DOR0	0x02C	DAC0数据输出寄存器
DAC_DOR1	0X030	DAC1数据输出寄存器
DAC_SR	0X034	状态寄存器
DAC_ETRS	0X038	DAC 测试寄存器/外部触发选择寄存器

28.3.1 DAC 控制寄存器（DAC_CR0）

偏移地址 0x00

复位值 0xC000 C000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SREF1		DMAUDRIE1	DMAEN1	MAMP1				WAVE1		TSEL1			TEN1	BOFF1	EN1
RW		RW	RW	RW				RW		RW			RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SREF0		DMAUDRIE0	DMAEN0	MAMP0				WAVE0		TSEL0			TEN0	BOFF0	EN0
RW		RW	RW	RW				RW		RW			RW	RW	RW

位	标记	功能描述
31:30	SREF1	DAC1参考电压选择 00：内部1.5V 01：内部2.5V 10：外部参考电压ExRef（PB01） 11：AVCC电压
29	DMAUDRIE1	DAC1 DMA 下溢中断使能 0:禁止DAC1通道DMA下溢中断； 1:使能DAC1通道DMA下溢中断
28	DMAEN1	DAC1 通道DMA使能 0:禁止DAC1通道DMA模式 1:使能DAC1通道DMA模式
27:24	MAMP1	DAC1通道掩码/振幅选择器 0000：不屏蔽LFSR的位0/三角波振幅等于1 0001：不屏蔽LFSR的位[1:0]/三角波振幅等于3 0010：不屏蔽LFSR的位[2:0]/三角波振幅等于7 0011：不屏蔽LFSR的位[3:0]/三角波振幅等于15 0100：不屏蔽LFSR的位[4:0]/三角波振幅等于31 0101：不屏蔽LFSR的位[5:0]/三角波振幅等于63 0110：不屏蔽LFSR的位[6:0]/三角波振幅等于127 0111：不屏蔽LFSR的位[7:0]/三角波振幅等于255 1000：不屏蔽LFSR的位[8:0]/三角波振幅等于511 1001：不屏蔽LFSR的位[9:0]/三角波振幅等于1023 1010：不屏蔽LFSR的位[10:0]/三角波振幅等于2047 ≥1011：不屏蔽LFSR的位[11:0]/三角波振幅等于4095

23:22	WAVE1	DAC1 通道噪声/三角波生成使能 00:禁止生成波 01:使能生成噪声波 1x 使能生成三角波
21:19	TSEL1	DAC1通道触发器选择 000 TIM0 TRADC触发 001 TIM1 TRADC触发 010 TIM2 TRADC触发 011 TIM3 TRADC触发 100 TIM4 TRADC触发 101 TIM5 TRADC触发 110 软件触发 111 外部端口触发
18	TEN1	DAC1通道触发使能 0：禁止 DAC1 通道触发，写入 DAC_DHR1 寄存器的数据在一个 APB1 时钟周期之后转移到 DAC_DOR1 寄存器 1：使能 DAC1 通道触发，DAC_DHR1 寄存器的数据在三个 APB1 时钟周期之后转移到DAC_DOR1 寄存器
17	BOFF1	DAC1 通道输出缓冲器禁止控制，使能缓冲器的BUFFEN在OPA控制寄存器中 0：使能 DAC1 通道输出缓冲器，DAC1通过缓存器到端口 1：禁止 DAC1 通道输出缓冲器，DAC1输出到端口 使用缓存作为输出到端口,需要使能OPA4 的BUFEN功能，且不能使能OPA4
16	EN1	DAC1 通道使能 0：禁止 DAC1 通道 1：使能 DAC1 通道
15:14	SREF0	DAC0参考电压选择 00：内部1.5V 01：内部2.5V 10：外部参考电压ExRef（PB01） 11：AVCC电压
13	DMAUDRIE0	DAC0 DMA 下溢中断使能 0:禁止DAC0通道DMA下溢中断； 1:使能DAC0通道DMA下溢中断
12	DMAEN0	DAC0 通道DMA使能 0:禁止DAC0通道DMA模式 1:使能DAC0通道DMA模式
11:8	MAMP0	DAC0通道掩码/振幅选择器 0000：不屏蔽LFSR的位0/三角波振幅等于1 0001：不屏蔽LFSR的位[1:0]/三角波振幅等于3 0010：不屏蔽LFSR的位[2:0]/三角波振幅等于7

		0011：不屏蔽LFSR的位[3:0]/三角波振幅等于15 0100：不屏蔽LFSR的位[4:0]/三角波振幅等于31 0101：不屏蔽LFSR的位[5:0]/三角波振幅等于63 0110：不屏蔽LFSR的位[6:0]/三角波振幅等于127 0111：不屏蔽LFSR的位[7:0]/三角波振幅等于255 1000：不屏蔽LFSR的位[8:0]/三角波振幅等于511 1001：不屏蔽LFSR的位[9:0]/三角波振幅等于1023 1010：不屏蔽LFSR的位[10:0]/三角波振幅等于2047 ≥1011：不屏蔽LFSR的位[11:0]/三角波振幅等于4095
7:6	WAVE0	DAC0 通道噪声/三角波生成使能 00:禁止生成波 01:使能生成噪声波 1x 使能生成三角波
5:3	TSEL0	DAC0通道触发器选择 000 TIM0 TRADC触发 001 TIM1 TRADC触发 010 TIM2 TRADC触发 011 TIM3 TRADC触发 100 TIM4 TRADC触发 101 TIM5 TRADC触发 110 软件触发 111 外部端口触发
2	TEN0	DAC0通道触发使能 0：禁止 DAC0 通道触发，写入 DAC_DHR0 寄存器的数据在一个 APB1 时钟周期之后转移到 DAC_DOR0 寄存器 1：使能 DAC0 通道触发，DAC_DHR0 寄存器的数据在三个 APB1 时钟周期之后转移到DAC_DOR0 寄存器
1	BOFF0	DAC0 通道输出缓冲器禁止控制，使能缓冲器的BUFFEN在OPA控制寄存器中 0：使能 DAC0 通道输出缓冲器，DAC0通过缓存器到端口 1：禁止 DAC0 通道输出缓冲器，DAC0输出到端口 使用缓存作为输出到端口,需要使能OPA3 的BUFEN功能，且不能使能OPA3
0	EN0	DAC0 通道使能 0：禁止 DAC0 通道 1：使能 DAC0 通道

28.3.2 DAC 软件触发寄存器（DAC_SWTRIGR）

偏移地址 0x04

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														SWTRIG1	SWTRIG0
														W1	W1

位	标记	功能描述
31:2	Reserved	保留
1	SWTRIG1	DAC1 软件触发 写1软件触发，自动清零，写0无动作
0	SWTRIG0	DAC0 软件触发 写1软件触发，自动清零，写0无动作

28.3.3 DAC0 12 位右对齐数据保持寄存器（DAC_DHR12R0）

偏移地址 0x08

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DHR0											
				RW											

位	标记	功能描述
31:12	Reserved	保留
11:0	DHR0	DAC 通道0 12 位右对齐数据 这些位由软件写入，用于为 DAC 通道0指定 12 位数据。

28.3.4 DAC0 12 位左对齐数据保持寄存器（DAC_DHR12L0）

偏移地址 0x0C

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DHR0												Reserved			
RW															

位	标记	功能描述
31:16	Reserved	保留
15:4	DHR0	DAC 通道0 12 位左对齐数据 这些位由软件写入，用于为 DAC 通道0指定 12 位数据。
3:0	Reserved	保留

28.3.5 DAC0 8 位右对齐数据保持寄存器（DAC_DHR8R0）

偏移地址 0x10

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DHR0							
								RW							

位	标记	功能描述
31:8	Reserved	保留
7:0	DHR0	DAC 通道0 8 位右对齐数据 这些位由软件写入，用于为 DAC 通道0指定 8 位数据。写入的为DHR的高八位。

28.3.6 DAC1 12 位右对齐数据保持寄存器（DAC_DHR12R1）

偏移地址 0x14

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DHR1											
				RW											

位	标记	功能描述
31:12	Reserved	保留
11:0	DHR1	DAC 通道1 12 位右对齐数据 这些位由软件写入，用于为 DAC 通道1指定 12 位数据。

28.3.7 DAC1 12 位左对齐数据保持寄存器（DAC_DHR12L1）

偏移地址 0x18

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DHR1												Reserved			
RW															

位	标记	功能描述
31:16	Reserved	保留
15:4	DHR1	DAC 通道1 12 位左对齐数据 这些位由软件写入，用于为 DAC 通道1指定 12 位数据。
3:0	Reserved	保留

28.3.8 DAC1 8 位右对齐数据保持寄存器（DAC_DHR8R1）

偏移地址 0x1C

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DHR1							
								RW							

位	标记	功能描述
31:8	Reserved	保留
7:0	DHR1	DAC 通道1 8 位右对齐数据 这些位由软件写入，用于为 DAC 通道1指定 8 位数据。写入的为DHR的高八位。

28.3.9 双 DAC 12 位右对齐数据保持寄存器（DAC_DHR12RD）

偏移地址 0x20

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				DHR1											
				RW											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DHR0											
				RW											

位	标记	功能描述
31:28	Reserved	保留
27:16	DHR1	DAC 通道1 12 位左对齐数据 这些位由软件写入，用于为 DAC 通道1指定 12 位数据。
15:12	Reserved	保留
11:0	DHR0	DAC 通道0 12 位右对齐数据 这些位由软件写入，用于为 DAC 通道0指定 12 位数据。

28.3.10 双 DAC 12 位左对齐数据保持寄存器（DAC_DHR12LD）

偏移地址 0x24

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DHR1												Reserved			
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DHR0												Reserved			
RW															

位	标记	功能描述
31:20	DHR1	DAC 通道1 12 位左对齐数据 这些位由软件写入，用于为 DAC 通道1指定 12 位数据。
19:16	Reserved	保留
15:4	DHR0	DAC 通道0 12 位左对齐数据 这些位由软件写入，用于为 DAC 通道0指定 12 位数据。
3:0	Reserved	保留

28.3.11 双 DAC 8 位右对齐数据保持寄存器（DAC_DHR8RD）

偏移地址 0x28

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DHR1								DHR0							
RW								RW							

位	标记	功能描述
31:16	Reserved	保留
15:8	DHR1	DAC 通道1 8 位右对齐数据 这些位由软件写入，用于为 DAC 通道0指定 8 位数据。写入的为DHR的高八位。
7:0	DHR0	DAC 通道0 8 位右对齐数据 这些位由软件写入，用于为 DAC 通道0指定 8 位数据。写入的为DHR的高八位。

28.3.12 DAC0 数据输出寄存器（DAC_DOR0）

偏移地址 0x2C

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DOR0											
				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	DOR0	DAC 0 通道数据输出 DAC 0 通道的数据输出

28.3.13 DAC1 数据输出寄存器（DAC_DOR1）

偏移地址 0x30

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DOR1											
				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	DOR1	DAC 1 通道数据输出 DAC 1 通道的数据输出

28.3.14 DAC 状态寄存器（DAC_SR）

偏移地址 0x34

复位值 0x00004000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DMAUDR1	Reserved												
		RW													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		DMAUDR0	Reserved												
		RW													

位	标记	功能描述
31:30	Reserved	保留
29	DMAUDR1	DAC 1 通道 DMA 下溢标志 0：DAC 1 通道未发生 DMA 下溢错误状况 1：DAC 1 通道发生 DMA 下溢错误状况（当前所选触发源以高于 DMA 服务能力的频率驱动 DAC 1 通道转换）
28:14	Reserved	保留
13	DMAUDR0	DAC 0 通道 DMA 下溢标志 0：DAC0 通道未发生 DMA 下溢错误状况 1：DAC 0 通道发生 DMA 下溢错误状况（当前所选触发源以高于 DMA 服务能力的频率驱动 DAC 0 通道转换）
12:0	Reserved	保留

28.3.15 DAC 触发选择寄存器（DAC_ETRS）

偏移地址 0x038

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									PTIRGSEL			RSV			
									RW			RW			

位	标记	功能描述
31:7	Reserved	保留位
6:4	PTRIGSEL	端口触发选择 000 PA9 001 PB9 010 PC9 011 PD9 100 PE9 101 PF9 11X 保留
3:0	RSV	保留位

29 模拟比较器（VC）

29.1模拟电压比较器 VC 简介

模拟电压比较器 VC 用于比较两个输入模拟电压的大小，并根据比较结果输出高/低电平。当“+”输入端电压高于“−”输入端电压时，电压比较器输出为高电平；当“+”输入端电压低于“−”输入端电压时，电压比较器输出为低电平。本产品内部集成的模拟电压比较器 VC 具有以下特性：

- 支持电压比较功能；
- 支持内部 64 阶 VCC 分压（使用分压来源电压需要大于 1.8V）；
- 支持 16 个外部输入端口作为电压比较器的输入；
- 支持三种软件可配置的中断触发方式：高电平触发/上升沿触发/下降沿触发；
- 电压比较器的输出可以作为 通用定时器 和 低功耗定时器控制输入；
- 电压比较器的输出可以作为 高级定时器和通用定时器的刹车输入或者捕获输入；
- 支持在超低功耗模式下工作，电压比较器的中断输出可以将芯片从超低功耗模式下唤醒；
- 提供软件可配置的滤波时间以增强芯片的抗干扰能力。

注意：使用 VC 需要使能 BGR,参考 BGR 寄存器描述

29.2电压比较器框架图

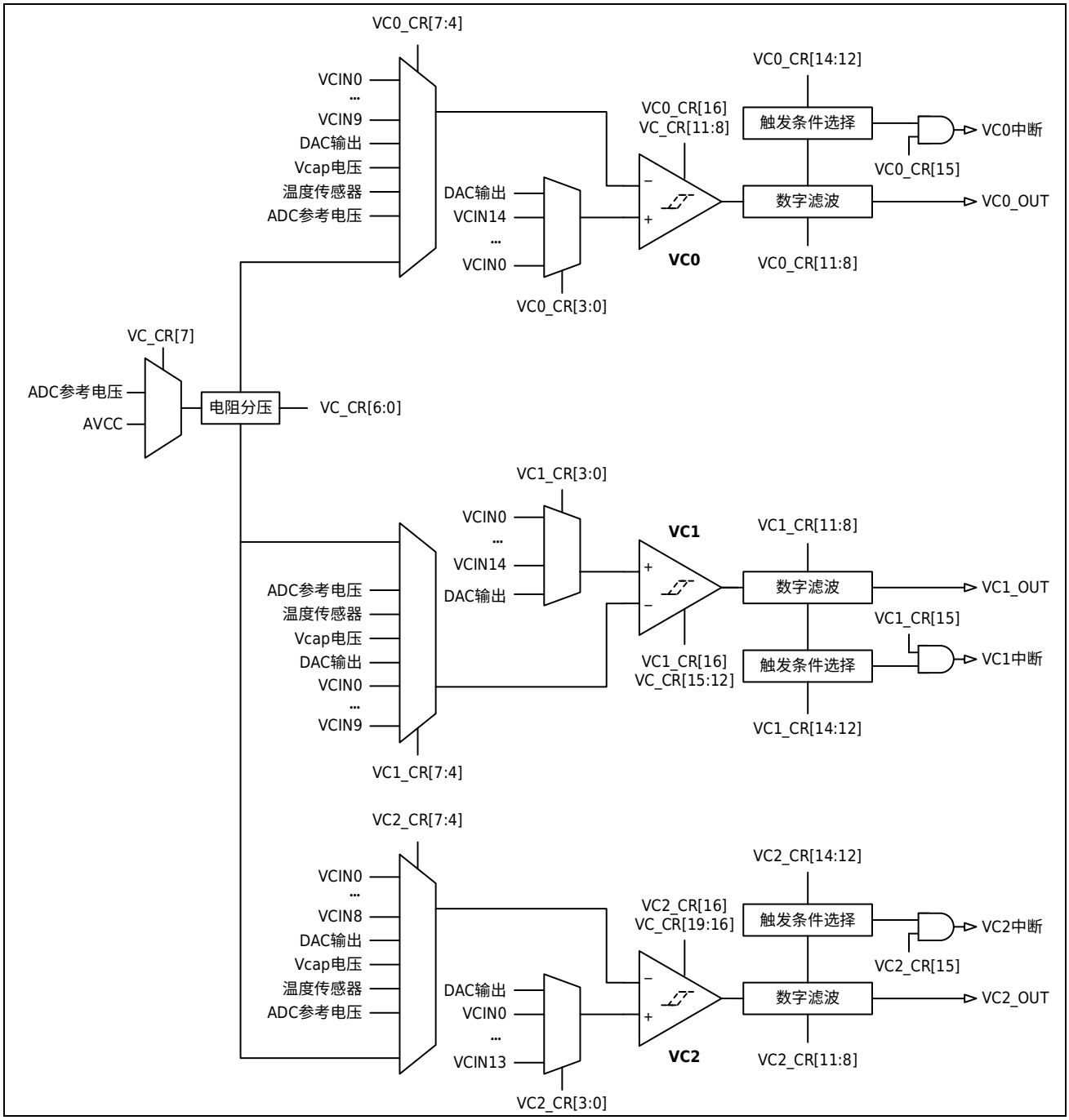


图 29-1 VC 框架图

29.3建立/响应时间

当使用电压比较器时，从 VC 使能或者 VC 的两端输入电压变化到输出正确的结果的时间由 VC 控制寄存器（VC_CR）中的 BIAS_SEL 控制位决定，电流越大 VC 响应越快，典型值从 200ns 到 20us 四档可调。

如果需要内部 BGR 提供的参考电压（1.5V 或 2.5V）作为比较基准电压。内部 BGR 的启动时间大约为 20us，电压比较器需要等待内部 BGR 稳定后才能正常输出。

29.4滤波时间

在电压比较器固有的建立/响应时间之外，用户可以设置更长的滤波时间来过滤掉系统噪声，比如马达停止时的大电流噪声。

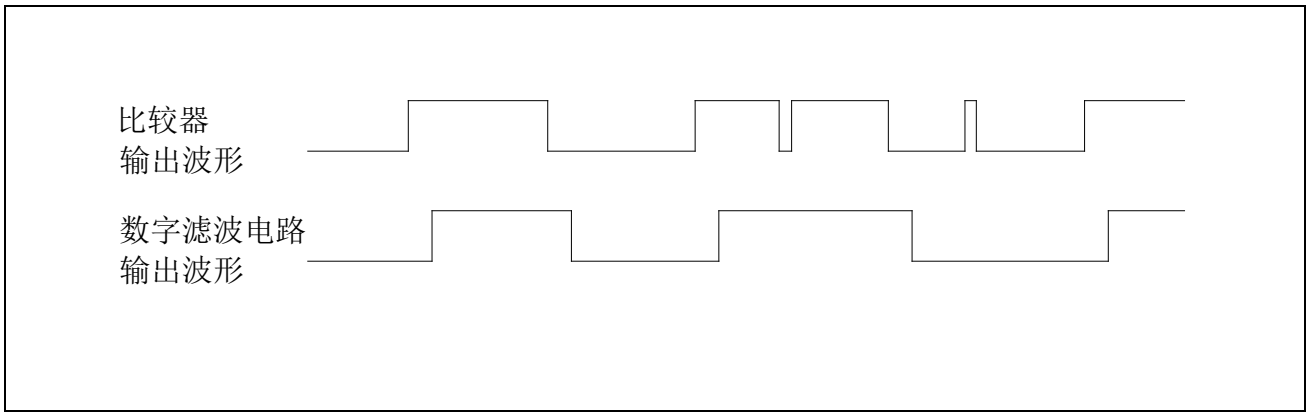


图 29-2 VC 滤波响应时间

29.5迟滞功能

电压比较器可以选择迟滞功能，迟滞功能使能后的图示如下：

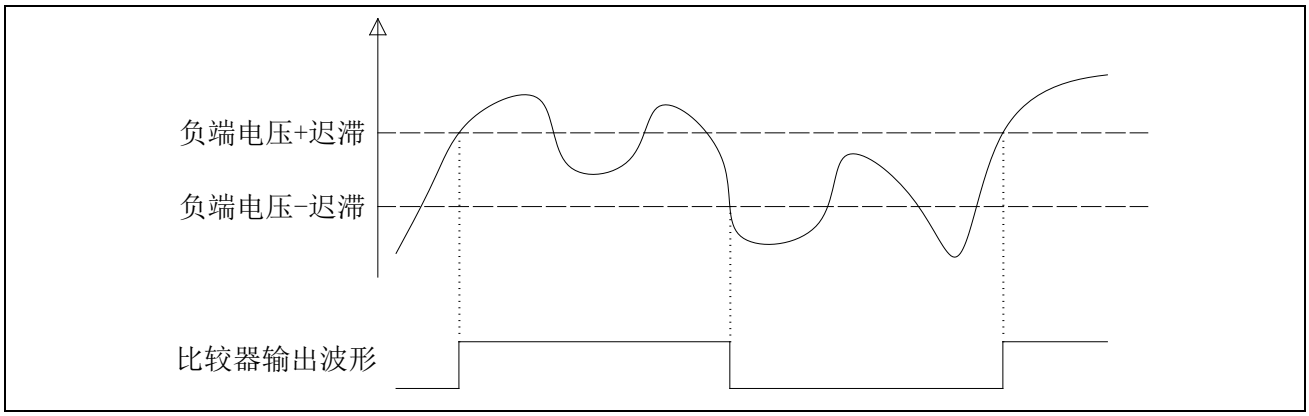


图 29-3 VC 迟滞功能

29.6VC 寄存器

基地址 0x40002400

表 29-1 VC 寄存器

寄存器	偏移地址	描述
VC_CR	0x010	VC0/1 配置寄存器 0
VC0_CR	0x014	VC0 配置寄存器
VC1_CR	0x018	VC1 配置寄存器
VC0_ OUT_CFG	0x01C	VC0 输出配置寄存器
VC1_ OUT_CFG	0x020	VC1 输出配置寄存器
VC_IFR	0x024	VC 中断寄存器
VC2_CR	0x150	VC2 配置寄存器
VC2_ OUT_CFG	0x154	VC2 输出配置寄存器

29.6.1 VC 配置寄存器（VC_CR）

偏移地址 0x010

复位值 0x00000020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												VC2_HYS_SEL		VC2_BIAS_SEL	
												R/W		R/W	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VC1_HYS_SEL		VC1_BIAS_SEL		VC0_HYS_SEL		VC0_BIAS_SEL		VC_REF2P5_SEL	VC_DIV_EN	VC_DIV					
R/W		R/W		R/W		R/W		R/W	R/W	R/W					

位	标记	功能描述
31:16	Reserved	保留
19:18	VC2_HYS_SEL	VC2 迟滞选择: 00:没有迟滞 01:迟滞电压大约10mV 10:迟滞电压大约20mV 11:迟滞电压大约30mV
17:16	VC2_BIAS_SEL	VC2 功耗选择（功耗越大，响应速度越快） 00:300nA 01:1.2uA 10:10uA(需要开启BGR，BGR启动时间大约20us) 11:20uA(需要开启BGR，BGR启动时间大约20us)
15:14	VC1_HYS_SEL	VC1 迟滞选择: 00:没有迟滞 01:迟滞电压大约10mV 10:迟滞电压大约20mV 11:迟滞电压大约30mV
13:12	VC1_BIAS_SEL	VC1 功耗选择（功耗越大，响应速度越快） 00:300nA 01:1.2uA 10:10uA(需要开启BGR，BGR启动时间大约20us) 11:20uA(需要开启BGR，BGR启动时间大约20us)
11:10	VC0_HYS_SEL	VC0 迟滞选择: 00:没有迟滞 01:迟滞电压大约10mV 10:迟滞电压大约20mV 11:迟滞电压大约30mV
9:8	VC0_BIAS_SEL	VC0 功耗选择（功耗越大，响应速度越快） 00:300nA

		01:1.2uA 10:10uA(需要开启BGR，BGR启动时间大约20us) 11:20uA(需要开启BGR，BGR启动时间大约20us)
7	VC_REF2P5_SEL	VC_DIV 参考电压Vref选择 0:VCC 1:ADC_CR0.SREF所选择的参考电压
6	VC_DIV_EN	6位 DAC 使能 1：使能
5:0	VC_DIV	6位 DAC 配置 000000: 1/64 Vref 000001:2/64 Vref 000010:3/64 Vref 000011:4/64 Vref … 111110:63/64 Vref 111111: Vref

29.6.2 VC0 配置寄存器（VC0_CR）

偏移地址 0x014

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															EN
															RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE	level	rising	falling	debounce_time			FLTEN	n_sel				p_sel			
RW	RW	RW	RW	RW			RW	RW				RW			

位	标记	功能描述
31:17	Reserved	保留
16	EN	电压比较器使能 1：使能电压比较器 0：关闭电压比较器
15	IE	VC 中断使能 1：使能；0：禁止
14	level	VC 输出信号触发中断选择 1：使能高电平触发 INT flag 0：禁止高电平触发 INT flag
13	rising	VC 输出信号触发中断选择 1：使能上升沿触发 INT flag 0：禁止上升沿触发 INT flag
12	falling	VC 输出信号触发中断选择 1：使能下降沿触发 INT flag 0：禁止下降沿触发 INT flag
11:9	debounce_time	VC 输出滤波时间配置 111：滤波时间大约为28.8ms 110：滤波时间大约为7.2ms 101：滤波时间大约为1.8ms 100：滤波时间大约为450us 011：滤波时间大约为112us 010：滤波时间大约为28us 001：滤波时间大约为14us 000：滤波时间大约为7us 注意：滤波时间的配置只有在 FLTEN=1 时才有效。
8	FLTEN	1：启动 VC 滤波

		0：VC 无滤波
7:4	N_SEL	电压比较器“-”端输入选择 0000: select channel 0 input PA0 0001: select channel 1 input PA1 0010: select channel 2 input PA2 0011: select channel 3 input PA3 0100: select channel 4 input PA4 0101: select channel 5 input PA5 0110: select channel 6 input PA6 0111: select channel 7 input PA7 1000: select channel 8 input PC4 1001: select channel 9 input PC5 1010:DAC0 1011:电阻分压输出电压 1100:内置温度传感器输出电压 1101:Reserved 1110:ADC的REF (使用需要使能ADC) 1111:VCAP的输出电压
3:0	P_SEL	电压比较器“+”端输入选择 0000: select channel 0 input PC0 0001: select channel 1 input PC1 0010: select channel 2 input PC2 0011: select channel 3 input PC3 0100: select channel 4 input PA0 0101: select channel 5 input PA1 0110: select channel 6 input PA2 0111: select channel 7 input PA3 1000: select channel 8 input PA4 1001: select channel 9 input PA5 1010: select channel 10 input PA6 1011: select channel 11 input PA7 1100: select channel 12 input PB4 1101: select channel 13 input PB5 1110: select channel 14 input PB6 1111: DAC0

29.6.3 VC1 配置寄存器（VC1_CR）

偏移地址 0x018

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															EN
															RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE	level	rising	falling	debounce_time			FLTEN	n_sel				p_sel			
RW	RW	RW	RW	RW			RW	RW				RW			

位	标记	功能描述
31:17	Reserved	保留
16	EN	电压比较器使能 1：使能电压比较器 0：关闭电压比较器
15	IE	VC 中断使能 1：使能；0：禁止
14	level	VC 输出信号触发中断选择 1：使能高电平触发 INT flag 0：禁止高电平触发 INT flag
13	rising	VC 输出信号触发中断选择 1：使能上升沿触发 INT flag 0：禁止上升沿触发 INT flag
12	falling	VC 输出信号触发中断选择 1：使能下降沿触发 INT flag 0：禁止下降沿触发 INT flag
11:9	debounce_time	VC 输出滤波时间配置 111：滤波时间大约为28.8ms 110：滤波时间大约为7.2ms 101：滤波时间大约为1.8ms 100：滤波时间大约为450us 011：滤波时间大约为112us 010：滤波时间大约为28us 001：滤波时间大约为14us 000：滤波时间大约为7us 注意：滤波时间的配置只有在 FLTEN=1 时才有效。
8	FLTEN	1：启动 VC 滤波

		0：VC 无滤波
7:4	N_SEL	电压比较器“-”端输入选择 0000: select channel 0 input PC0 0001: select channel 1 input PC1 0010: select channel 2 input PC2 0011: select channel 3 input PC3 0100: select channel 4 input PA0 0101: select channel 5 input PA1 0110: select channel 6 input PB0 0111: select channel 7 input PB1 1000: select channel 8 input PB2 1001: select channel 9 input PB3 1010: DAC1 1011:电阻分压输出电压 1100:内置温度传感器输出电压 1101:Reserved 1110:ADC的REF (使用需要使能ADC) 1111:VCAP引脚电压
3:0	P_SEL	电压比较器“+”端输入选择 0000: select channel 0 input PA0 0001: select channel 1 input PA1 0010: select channel 2 input PA2 0011: select channel 3 input PA3 0100: select channel 4 input PA4 0101: select channel 5 input PA5 0110: select channel 6 input PB1 0111: select channel 7 input PB2 1000: select channel 8 input PB10 1001: select channel 9 input PB12 1010: select channel 10 input PB13 1011: select channel 11 input PB14 1100: select channel 12 input PB4 1101: DAC1 1110: select channel 14 input PB6 1111: select channel 15 input PB7

29.6.4 VC0 输出配置寄存器（VC0_OUT_CFG）

偏移地址 0x01C

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
brake	TIM6	INV_TIM6	TIM5	INV_TIM5	TIM4	INV_TIM4	Reserved			TIMBK	TIM3RCLR	TIM2RCLR	TIM1RCLR	TIM0RCLR	INV_Timer
RW	RW	RW	RW	RW	RW	RW				RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:16	Reserved	保留
15	brake	VC0 作为 Advanced Timer 刹车控制 1：使能； 0：禁止。
14	TIM6	VC0 filter 结果输出到 TIM6 捕获输入CHA使能 1：使能； 0：禁止。
13	INV_TIM6	VC0 filter 结果输出到 TIM6 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
12	TIM5	VC0 filter 结果输出到 TIM5 捕获输入CHA使能 1：使能； 0：禁止。
11	INV_TIM5	VC0 filter 结果输出到 TIM5 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
10	TIM4	VC0 filter 结果输出到 TIM4 捕获输入CHA使能 1：使能； 0：禁止。
9	INV_TIM4	VC0 filter 结果输出到 TIM4 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
8:6	Res.	保留
5	TIMBK	VC0 filter 结果输出到Timer0/1/2/3 刹车控制 1：使能； 0：禁止。
4	TIM3RCLR	VC0 filter 结果输出到 TIM3 REFCLR 使能控制 1：使能； 0：禁止。
3	TIM2RCLR	VC0 filter 结果输出到 TIM2 REFCLR 使能控制 1：使能； 0：禁止。
2	TIM1RCLR	VC0 filter 结果输出到 TIM1 REFCLR 使能控制 1：使能； 0：禁止。
1	TIM0RCLR	VC0 filter 结果输出到 TIM0 REFCLR 使能控制 1：使能； 0：禁止。

0	INV_Timer	VC0 filter 结果输出反向到各 TIM0/1/2/3/REFCLR 1：使能反向； 0：禁止反向，输入与 VC 输出同向。

29.6.5 VC1 输出配置寄存器（VC1_ OUT_CFG）

偏移地址 0x020

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
brake	TIM6	INV_TIM6	TIM5	INV_TIM5	TIM4	INV_TIM4	Reserved			TIMBK	TIM3RCLR	TIM2RCLR	TIM1RCLR	TIM0RCLR	INV_Timer
RW	RW	RW	RW	RW	RW	RW				RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:16	Reserved	保留
15	brake	VC1 作为 Advanced Timer 刹车控制 1：使能； 0：禁止。
14	TIM6	VC1 filter 结果输出到 TIM6 捕获输入CHB使能 1：使能； 0：禁止。
13	INV_TIM6	VC1 filter 结果输出到 TIM6 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
12	TIM5	VC1 filter 结果输出到 TIM5 捕获输入CHB使能 1：使能； 0：禁止。
11	INV_TIM5	VC1 filter 结果输出到 TIM5 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
10	TIM4	VC1 filter 结果输出到 TIM4 捕获输入CHB使能 1：使能； 0：禁止。
9	INV_TIM4	VC1 filter 结果输出到 TIM4 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
8:6	Res.	保留
5	TIMBK	VC1 filter 结果输出到Timer0/1/2/3 刹车控制 1：使能； 0：禁止。
4	TIM3RCLR	VC1 filter 结果输出到 TIM3 REFCLR 使能控制 1：使能； 0：禁止。
3	TIM2RCLR	VC1 filter 结果输出到 TIM2 REFCLR 使能控制 1：使能； 0：禁止。
2	TIM1RCLR	VC1 filter 结果输出到 TIM1 REFCLR 使能控制

		1：使能； 0：禁止。
1	TIM0RCLR	VC1 filter 结果输出到 TIM0 REFCLR 使能控制 1：使能； 0：禁止。
0	INV_Timer	VC1 filter 结果输出反向到各 TIM0/1/2/3/REFCLR 1：使能反向； 0：禁止反向，输入与 VC 输出同向。

29.6.6 VC 中断寄存器（VC_IFR）

偏移地址 0x024

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										VC2_Filter	VC2_INTF	VC1_Filter	VC0_Filter	VC1_INTF	VC0_INTF
										RO	RW0	RO	RO	RW0	RW0

位	标记	功能描述
31:4	Reserved	保留
5	VC2_Filter	VC2 Filter 后的状态
4	VC2_INTF	VC2 中断标志，1发生 VC2 中断；0未发生中断；写0清除中断标志，写1无效
3	VC1_Filter	VC1 Filter 后的状态
2	VC0_Filter	VC0 Filter 后的状态
1	VC1_INTF	VC1 中断标志，1发生 VC1 中断；0未发生中断；写0清除中断标志，写1无效
0	VC0_INTF	VC0 中断标志，1发生 VC0 中断；0未发生中断；写0清除中断标志，写1无效

29.6.7 VC2 配置寄存器（VC2_CR）

偏移地址 0x150

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															EN
															R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE	level	rising	falling	debounce_time			FLTEN	n_sel				p_sel			
R/W	R/W	R/W	R/W	R/W			R/W	R/W				R/W			

位	标记	功能描述
31:17	Reserved	保留
16	EN	电压比较器使能 1：使能电压比较器 0：关闭电压比较器
15	IE	VC 中断使能 1：使能；0：禁止
14	level	VC 输出信号触发中断选择 1：使能高电平触发 INT flag 0：禁止高电平触发 INT flag
13	rising	VC 输出信号触发中断选择 1：使能上升沿触发 INT flag 0：禁止上升沿触发 INT flag
12	falling	VC 输出信号触发中断选择 1：使能下降沿触发 INT flag 0：禁止下降沿触发 INT flag
11:9	debounce_time	VC 输出滤波时间配置 111：滤波时间大约为28.8ms 110：滤波时间大约为7.2ms 101：滤波时间大约为1.8ms 100：滤波时间大约为450us 011：滤波时间大约为112us 010：滤波时间大约为28us 001：滤波时间大约为14us 000：滤波时间大约为7us 注意：滤波时间的配置只有在 FLTEN=1 时才有效。
8	FLTEN	1：启动 VC 滤波

		0：VC 无滤波
7:4	N_SEL	电压比较器“-”端输入选择 0000: select channel 0 input PA5 0001: select channel 1 input PB1 0010: select channel 2 input PE11 0011: select channel 3 input PE15 0100: select channel 4 input PB11 0101: select channel 5 input PB14 0110: select channel 6 input PD10 0111: select channel 7 input PD11 1000: select channel 8 input PC7 1001: DAC0 1010: DAC1 1011:NC 1100:内置温度传感器输出电压 1101:Reserved 1110:ADC的REF (使用需要使能ADC) 1111:LDO的输出电压
3:0	P_SEL	电压比较器“+”端输入选择 0000: select channel 0 input PA5 0001: select channel 1 input PB1 0010: select channel 2 input PE9 0011: select channel 3 input PE10 0100: select channel 4 input PE11 0101: select channel 5 input PE13 0110: select channel 6 input PE14 0111: select channel 7 input PE15 1000: select channel 8 input PB11 1001: select channel 9 input PB14 1010: select channel 10 input PD9 1011: select channel 11 input PD10 1100: select channel 12 input PD11 1101: select channel 13 input PC7 1110: DAC0 1111: DAC1

29.6.8 VC2 输出配置寄存器（VC2_ OUT_CFG）

偏移地址 0x154

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
brake	TIM6	INV_Timer6	TIM5	INV_Timer5	TIM4	INV_Timer4	Reserved			TIMBK	TIM3RCLR	TIM2RCLR	TIM1RCLR	TIM0RCLR	INV_Timer
R/W	R/W	R/W	R/W	R/W	R/W	R/W				R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述
31:16	Reserved	保留
15	brake	VC2 作为 Advanced Timer 刹车控制 1：使能； 0：禁止。
14	TIM6	VC2 filter 结果输出到 TIM6 捕获输入CHB使能 1：使能； 0：禁止。
13	INV_TIM6	VC2 filter 结果输出到 TIM6 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
12	TIM5	VC2 filter 结果输出到 TIM5 捕获输入CHB使能 1：使能； 0：禁止。
11	INV_TIM5	VC2 filter 结果输出到 TIM5 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
10	TIM4	VC2 filter 结果输出到 TIM4 捕获输入CHB使能 1：使能； 0：禁止。
9	INV_TIM4	VC2 filter 结果输出到 TIM4 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
8:6	Res.	保留
5	TIMBK	VC2 filter 结果输出到Timer0/1/2/3 刹车控制 1：使能； 0：禁止。
4	TIM3RCLR	VC2 filter 结果输出到 TIM3 REFCLR 使能控制 1：使能； 0：禁止。
3	TIM2RCLR	VC2 filter 结果输出到 TIM2 REFCLR 使能控制 1：使能； 0：禁止。
2	TIM1RCLR	VC2 filter 结果输出到 TIM1 REFCLR 使能控制 1：使能； 0：禁止。
1	TIM0RCLR	VC2 filter 结果输出到 TIM0 REFCLR 使能控制 1：使能； 0：禁止。

0	INV_Timer	VC2 filter 结果输出反向到各 TIM0/1/2/3/REFCLR 1：使能反向； 0：禁止反向，输入与 VC 输出同向。

30 低电压检测器（LVD）

30.1LVD 简介

LVD 可用于监测 VCC 及芯片管脚的电压。当被监测电压与 LVD 阈值的比较结果满足触发条件时，LVD 会产生中断或复位信号，用户可根据该信号执行一些紧急任务。

LVD 具有以下特性：

- 4 路监测源，AVCC、PC13、PB08、PB07；
- 16 阶阈值电压，1.8V~3.3V 可选；
- 8 种触发条件，高电平、上升沿、下降沿组合；
- 2 种触发结果，复位、中断；
- 8 阶滤波配置，防止误触发；
- 具备迟滞功能，强力抗干扰。

30.2LVD 框图

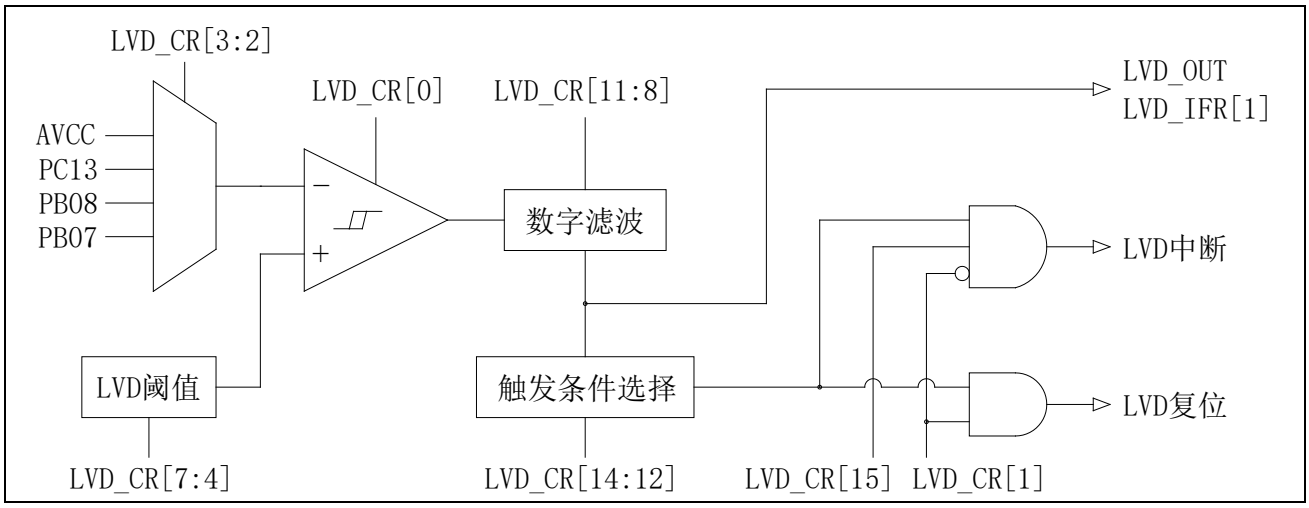


图 30-1 LVD 框图

30.3迟滞功能

LVD 内置的电压比较器具有迟滞功能，可以增强芯片的抗干扰能力。LVD 的输出信号会等到输入信号高于或低于阈值电压 20mV 后才发生翻转，其输入输出信号示意如下：

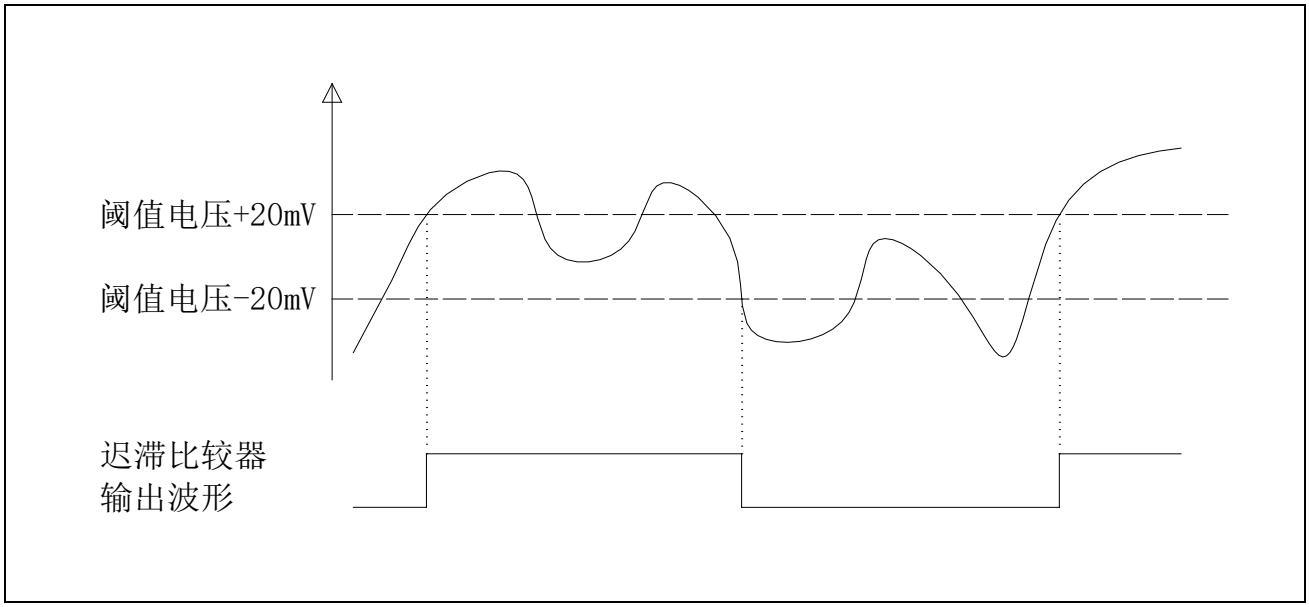


图 30-2 LVD 迟滞响应

30.4数字滤波

如果芯片的工作环境恶劣，迟滞比较器的输出会出现噪声信号。使能数字滤波模块，则迟滞比较器的输出波形中脉宽小于 LVD_CR.Debounce_time 的噪声信号都可以被滤除。禁止数字滤波模块，则数字滤波模块的输入输出信号相同。

从寄存器 LVD_IFR[1]可以读出数字滤波后的信号电平；当 GPIO 功能配置为 LVD_OUT 时，数字滤波后的信号可以从 GPIO 输出以方便测量。

使能数字滤波模块，滤波示意如下所示：

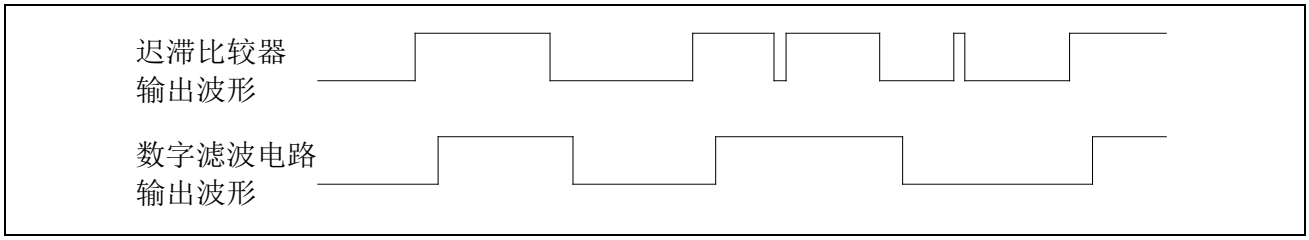


图 30-3 LVD 滤波输出

30.5配置示例

30.5.1 LVD 配置为低电压复位

在本模式下，监测电压低于阈值电压时复位 MCU。

配置方法如下所示：

Step1：配置 LVD_CR.Source_sel，选择待监测的电压来源。

Step2：配置 LVD_CR. VTDS，选择 LVD 的阈值电压。

Step3：配置 LVD_CR. Debounce_time，选择 LVD 滤波时间。

Step4：配置 LVD_CR. FLTEN，使能 LVD 滤波。

Step5：设置 LVD_CR. HTEN 为 1，选择高电平触发 LVD 动作。

Step6：设置 LVD_CR.ACT 为 1，选择 LVD 动作为复位。

Step7：设置 LVD_CR.LVDEN 为 1，使能 LVD。

30.5.2 LVD 配置为电压变化中断

在本模式下，监测电压高于或低于阈值电压时产生中断。

配置方法如下所示：

Step1：配置 LVD_CR.Source_sel，选择待监测的电压来源。

Step2：配置 LVD_CR. VTDS，选择 LVD 的阈值电压。

Step3：配置 LVD_CR. Debounce_time，选择 LVD 滤波时间。

Step4：配置 LVD_CR. FLTEN，使能 LVD 滤波。

Step5：设置 LVD_CR. RTEN 和 LVD_CR. FTEN 为 1，选择电平变化触发 LVD 动作。

Step6：设置 LVD_CR.ACT 为 0，选择 LVD 动作为中断。

Step7：设置 LVD_CR.IE 为 1，使能 LVD 中断。

Step8：使能 NVIC 中断向量表中的 LVD 中断。

Step9：设置 LVD_CR.LVDEN 为 1，使能 LVD。

Step10：在 LVD 的中断服务程序中执行用户需要进行的操作；退出中断服务程序前向 LVD_IFR 写入 0x00 以清除中断标志。

30.6LVD 寄存器

基地址 0x40002400

表 30-1 LVD 寄存器

寄存器	偏移地址	描述
LVD_CR	0x028	LVD 配置寄存器
LVD_IFR	0x02C	LVD 中断标志寄存器

30.6.1 LVD 配置寄存器（LVD_CR）

偏移地址 0x028

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE	HTEN	RTEN	FTEN	Debounce_time			FLTEN	VTDS				Source_sel		ACT	LVDEN
RW	RW	RW	RW	RW			RW	RW				RW		RW	RW

位	标记	功能描述
31:16	Reserved	保留
15	IE	LVD中断使能 1：使能； 0：禁止。
14	HTEN	高电平触发使能 （被监测电压低于阈值电压) 1：使能； 0：禁止。
13	RTEN	上升沿触发使能 （被监测电压从高于阈值电压变为低于阈值电压) 1：使能； 0：禁止。
12	FTEN	下降沿触发使能 （被监测电压从低于阈值电压变为高于阈值电压) 1：使能； 0：禁止。
11:9	Debounce_time	数字滤波时间配置 111：滤波时间大约为28.8ms 110：滤波时间大约为7.2ms 101：滤波时间大约为1.8ms 100：滤波时间大约为450us 011：滤波时间大约为112us 010：滤波时间大约为28us 001：滤波时间大约为14us 000：滤波时间大约为7us 注：滤波时间仅在FLTEN为1时有效。
8	FLTEN	数字滤波使能配置 1: 使能数字滤波 0：禁止数字滤波
7:4	VTDS	LVD 阈值电压选择

		1111：3.3v 1110：3.2v 1101：3.1v 1100：3.0v 1011：2.9v 1010：2.8v 1001：2.7v 1000：2.6v 0111：2.5v 0110：2.4v 0101：2.3v 0100：2.2v 0011：2.1v 0010：2.0v 0001：1.9v 0000：1.8v
3:2	Source_sel	LVD 监测来源选择 11：PB07端口输入电压 10：PB08端口输入电压 01：PC13端口输入电压 00：AVCC 电源电压
1	ACT	LVD触发动作选择 1：系统复位 0：NVIC中断
0	LVDEN	LVD使能控制 1：使能LVD 0：禁止LVD

30.6.2 LVD 中断寄存器（LVD_IFR）

偏移地址 0x02C

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														LVD_Filter	INTF
														RO	RW0

位	标记	功能描述
31:2	Reserved	保留
1	LVD_Filter	LVD Filter 后的状态
0	INTF	LVD中断标志： 1：发生LVD中断； 0：未发生中断； 写0清除中断标志，写1无效。

31 运算放大器（OPA）

OPA 模块适用于简易放大器和电压跟随器应用。内部的五个运放可以使用外部电阻进行级联。OPA 的输入范围是 0V 到 AVCC，输出范围是 0.1V 到 AVCC-0.2V。

31.1OPA 特性

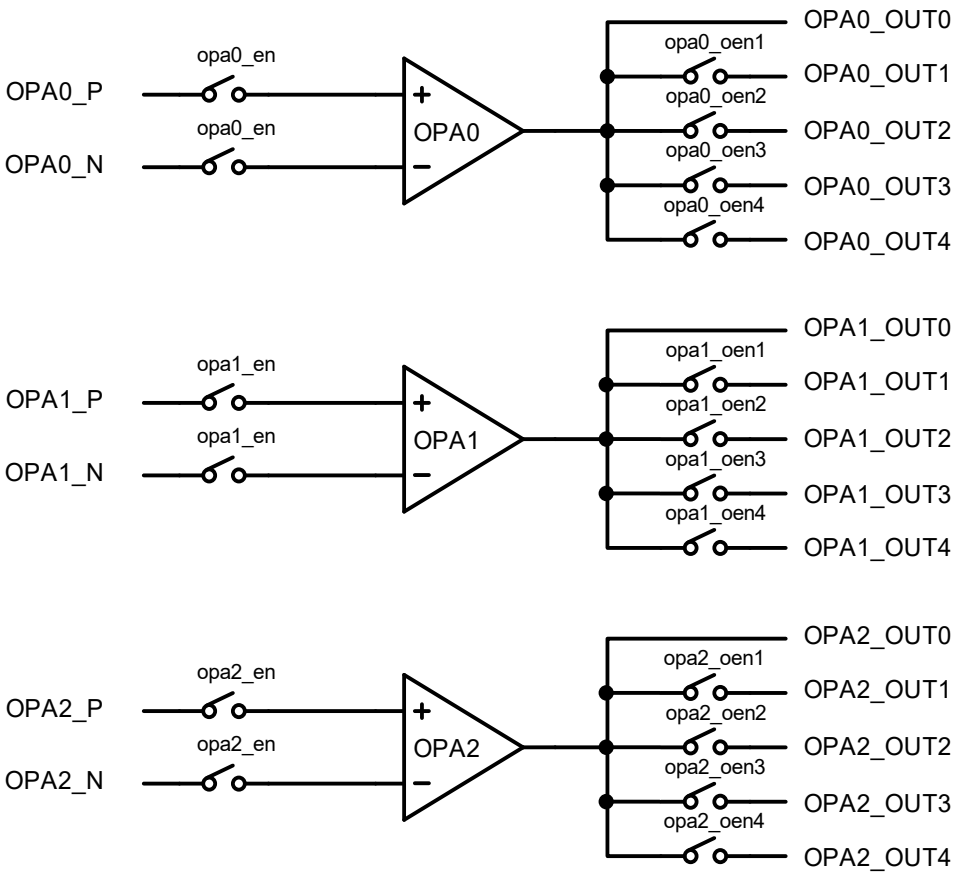
- 5 个独立配置运放
- OPA 的输入范围是 0 到 AVCC，输出范围是 0.1V 到 AVCC-0.2V 可编程增益
- 可配置为以下模式
  - 通用运放模式（general purpose OPA）
  - DAC 电压跟随器

31.2OPA 功能描述

5 个 OPA 可以通过使用外部元件组成放大器来放大小信号模拟输入信号，输出为放大后的信号。

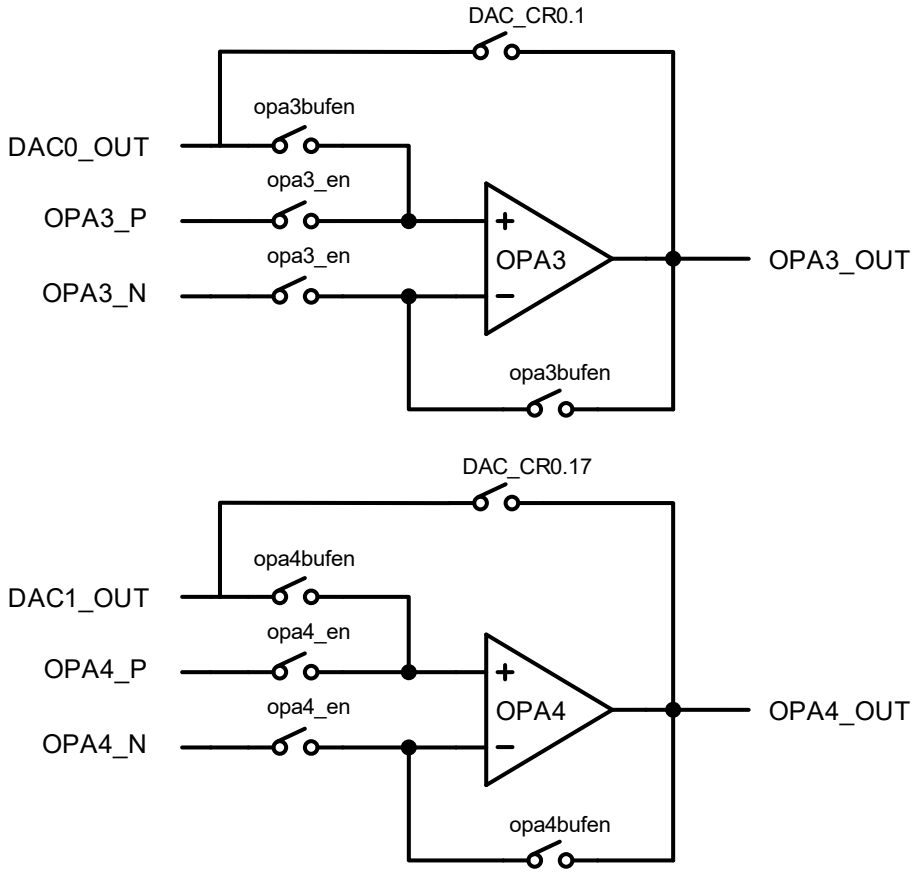
31.2.1 OPA0/1/2 （通用 OPA）

OPA0/1/2 的示意图如下：



31.2.2 OPA3/4 --- (可复用为 DAC buffer*2)

OPA3/4 的示意图如下：



31.3OPA 寄存器

基地址 0x40002400

表 31-1 OPA 寄存器

寄存器	偏移地址	描述
OPA_CR0	0x030	OP0 控制寄存器
OPA_CR1	0x034	OP1 控制寄存器
OPA_CR2	0x038	OP2 控制寄存器

31.3.1 OPA 输出使能寄存器（OPA_CR0）

偏移地址 0x030

复位值 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			OP2OEN4	OP2OEN4	OP2OEN3	OP2OEN2	OP2OEN1	OP1OEN4	OP1OEN3	OP1OEN2	OP1OEN1	OP0OEN4	OP0OEN3	OP0OEN2	OP0OEN1
			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:12	Reserved	保留
11	OP2OEN4	OP2输出4使能
10	OP2OEN3	OP2输出3使能
9	OP2OEN2	OP2输出2使能
8	OP2OEN1	OP2输出1使能
7	OP1OEN4	OP1输出4使能
6	OP1OEN3	OP1输出3使能
5	OP1OEN2	OP1输出2使能
4	OP1OEN1	OP1输出1使能
3	OP0OEN4	OP0输出4使能
2	OP0OEN3	OP0输出3使能
1	OP0OEN2	OP0输出2使能
0	OP0OEN1	OP0输出1使能

31.3.2 OPA 控制寄存器（OPA_CR1）

偏移地址 0x034

复位值 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				OP4BUFEN	OP3BUFEN	EN4	EN3	EN2	EN1	EN0	Reserved				
				RW	RW	RW	RW	RW	RW	RW					

位	标记	功能描述
31:12	Reserved	保留
11	OP4BUFEN	DAC1使用OP4单位增加缓存使能，与OPA4使能不能同时使能
10	OP3BUFEN	DAC0使用OP3单位增加缓存使能，与OPA3使能不能同时使能
9	EN4	OPA4使能
8	EN3	OPA3使能
7	EN2	OPA2使能
6	EN1	OPA1使能
5	EN0	OPA0使能
4:0	Reserved	保留

31.3.3 OPA 配置寄存器（OPA_CR2）

偏移地址 0x038

复位值 0x0000 36DB

不需要更改配置，使用缺省值就可以

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	BIASSEL4			BIASSEL3			BIASSEL2			BIASSEL1			BIASSEL0		
	RW			RW			RW			RW			RW		

位	标记	功能描述
31:15	Reserved	保留
14:12	BIASSEL4	OPA4 偏置电流选择 客户不需要更改配置，使用缺省值就可以
11:9	BIASSEL3	OPA3 偏置电流选择 客户不需要更改配置，使用缺省值就可以
8:6	BIASSEL2	OPA2 偏置电流选择 客户不需要更改配置，使用缺省值就可以
5:3	BIASSEL1	OPA1 偏置电流选择 客户不需要更改配置，使用缺省值就可以
2:0	BIASSEL0	OPA0 偏置电流选择 客户不需要更改配置，使用缺省值就可以

32 模拟其它寄存器

基地址 0x40002400

寄存器	偏移地址	描述
BGR_CR	0x000	BGR 控制寄存器

32.1BGR 配置寄存器（BGR_CR）

偏移地址 0x000

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														TS_EN	BGR_EN
														RW	RW

位	标记	功能描述
31:2	Reserved	保留
1	TS_EN	内置温度传感器使能控制 1：使能内部温度传感器 0：禁止内部温度传感器 注意：温度传感器使能 20us 后，才能输出稳定信号。
0	BGR_EN	BGR使能控制 1：使能 BGR 0：禁止 BGR 注意： 1) Peri_clken0.ADC为1时，才可以操作此寄存器。 2) BGR使能 20us 后，才能输出稳定的高精度参考电压。BGR稳定后才可以被其它模块使用，故用户操作中应加入等待BGR稳定的步骤。 3) 当使用ADC/OPA/PLL时，必须使能BGR。 4) 当使用VC时，需根据VC寄存器的配置决定是否使能BGR。

33 SWD 调试接口

本系列使用 ARM Cortex-M0+内核，该内核具有硬件调试模块 SWD，支持复杂的调试操作。硬件调试模块允许内核在取指(指令断点)或访问数据(数据断点)时停止。内核停止时，内核的内部状态和系统的外部状态都可以在 IDE 中进行查询。完成查询后，内核和外设可以被复原，程序将继续执行。当本系列微控制器连接到调试器并开始调试时，调试器将使用内核的硬件调试模块进行调试操作。

注意：

- SWD 在 DeepSleep 模式下不能工作，请在 Active 和 Sleep 模式下进行调试操作。

33.1SWD 调试附加功能

本产品使用了 ARM Cortex-M0+ CPU，该内核包含用于高级调试功能的硬件扩展，因此本产品所拥有的调试功能与 Cortex-M0+一致。调试扩展允许内核可以在取指（指令断点）或取访问数据（数据断点）时停止内核。内核停止时，可以查询内核的内部状态和系统的外部状态。查询完成后，将恢复内核和系统并恢复程序执行。

当调试主机与 MCU 相连并进行调试时，将使用调试功能。

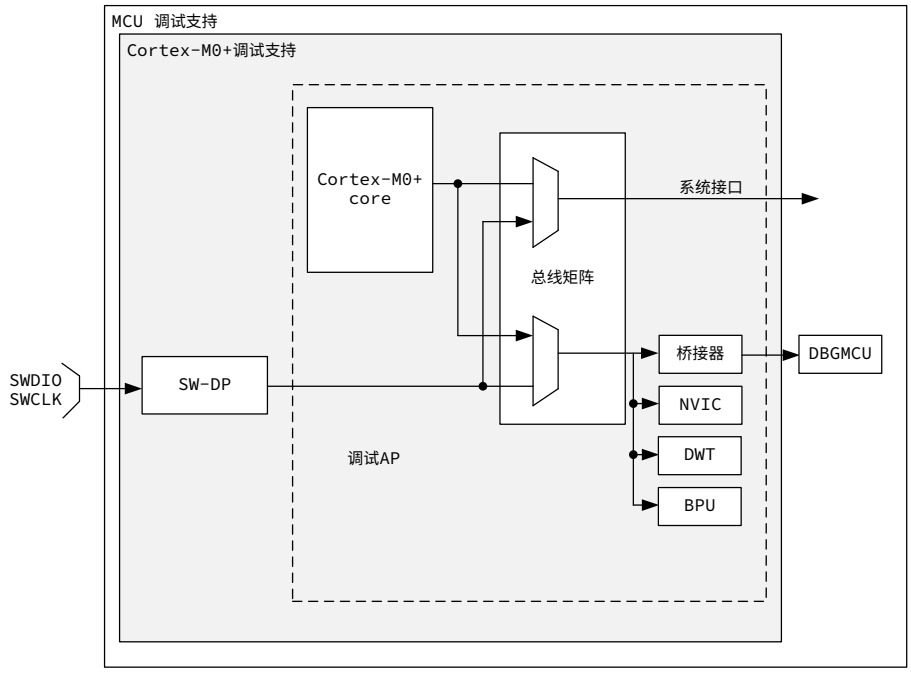


图 33-1 调试支持框图

Cortex®-M0+ 内核中内置的调试功能是 ARM® CoreSight 设计套件的一部分。

ARM® Cortex®-M0+内核提供集成片上调试支持。它包括：

- SW-DP：串行线
- BPU：断点单元
- DWT：数据观察点触发

注：

- 有关 ARM® Cortex®-M0+ 内核支持的调试功能的详细信息，请参见 Cortex®-M0+ 技术参考手册。

33.2ARM® 参考文档

- Cortex®-M0+技术参考手册（TRM)

可从 www.infocenter.arm.com 获取。

- ARM® 调试接口 V5
- ARM® CoreSight 设计套件版本 r1p1 技术参考手册

33.3调试端口引脚

33.3.1 SWD 端口引脚

本系列的 SWD 接口，需要用到 2 个引脚，如下表所示。

SWD 端口名称	调试功能	引脚分配
SWCLK	串行时钟	PA14
SWDIO	串行数据输入/输出	PA13

33.3.2 SW-DP 引脚分配

如果烧录程序时使能了【加密芯片】选项，则上电后 SWD 调试功能被禁止。如果烧录程序时没有使能【加密芯片】选项，则上电后 PA13/PA14 引脚均被初始化为可被调试器使用的专用引脚。用户可设置 SYSCTRL1 .SWD_USE_IO 寄存器来禁止 SWD 引脚的调试功能，SWD 引脚将被释放以用作普通 GPIO。SWD 引脚的配置与功能小结如下表所示：

【加密芯片】选项	SWD_USE_IO 配置	PA13/PA14 功能
加密	0	NA
加密	1	GPIO
不加密	0	SWD
不加密	1	GPIO

33.3.3 SWD 引脚上的内部上拉

用户软件释放 SW I/O 后，GPIO 控制器便会控制这些引脚。GPIO 控制寄存器的复位状态会将 I/O 置于等效的状态：

- SWDIO：输入上拉
- SWCLK：输入上拉

由于内置上拉和下拉电阻，因此无需添加外部电阻。

33.4SWD 端口

33.4.1 SWD 协议简介

此同步串行协议使用两个引脚：

- SWCLK：从主机到目标的时钟
- SWDIO：双向

利用该协议，可以同时读取和写入两组寄存器组（DPACC 寄存器组和 APACC 寄存器组）。传输数据时，LSB 在前。

对于 SWDIO 双向管理，必须在电路板上对线路进行上拉（ARM® 建议采用 100 k）。这些上拉电阻可在内部配置。无需外部上拉电阻。

每次在协议中更改 SWDIO 的方向时，都会插入转换时间，此时线路即不受主机驱动也不受目标驱动。默认情况下，此转换时间为一位时间，但可以通过配置 SWCLK 频率来调整。

33.4.2 SWD 协议序列

每个序列包括三个阶段：

1. 主机发送的数据包请求（8 位）
2. 目标发送的确认响应（3 位）
3. 主机或目标发送的数据传输阶段（33 位）

位	名称	说明
0	启动	必须为 1
1	APnDP	0：DP 访问；1：AP 访问
2	RnW	0：写请求；1：读请求
4:3	A[3:2]	DP 或 AP 寄存器的地址字段
5	奇偶校验	前面几位的单位奇偶校验
6	停止	0
7	驻留	不受主机驱动。由于存在上拉，因此必须由目标读为 1

有关 DPACC 和 APACC 寄存器的详细说明，请参见 Cortex®-M0+ TRM。

数据包请求后面始终为转换时间（默认 1 位），此时主机和目标都不会驱动。

位	名称	说明
0	ACK	001：FAULT 010：WAIT 100：OK

仅当发生 READ 事务或者接收到 WAIT 或 FAULT 确认时，ACK 响应后才必须是转换时间。

位	名称	说明
0:31	WDATA 或 RDATA	写入或读取数据
32	奇偶校验	32 个数据位的单奇偶校验

仅当发生 READ 事务时，DATA 传输后才必须是转换时间。

33.4.3 SW-DP 状态机（复位、空闲状态、ID 代码）

SW-DP 的状态机有一个用于标识 SW-DP 的内部 ID 代码。该代码符合 JEP-106 标准。此 ID 代码是默认的 ARM®代码，设置为 **0x0BC11477**（相当于 Cortex®-M0+）。

注意：

- 在目标读取此 ID 代码前，SW-DP 状态机是不工作的。

- 在上电复位后或者线路处于高电平超过 50 个周期后，SW-DP 状态机处于复位状态。
- 如果在复位状态后线路处于低电平至少两个周期，SW-DP 状态机处于空闲状态。
- 复位状态后，该状态机必须首先进入空闲状态，然后对 DP-SW ID CODE 寄存器执行读访问。否则，目标将在另一个事务上发出 FAULT 确认响应。

有关 SW-DP 状态机的更多详细信息，请参见 Cortex®-M0+ TRM 和 CoreSight 设计套件 r1p0TRM。

33.4.4 DP 和 AP 读/写访问

- 不延迟对 DP 的读访问：可以立即发送目标响应（如果 ACK=OK），也可以延迟发送目标响应（如果 ACK=WAIT）。
- 延迟对 AP 的读访问。这意味着会在下次传输时返回访问结果。如果要执行的下次访问不是 AP 访问，则必须读取 DP-RDBUFF 寄存器来获取结果。
- 每次进行 AP 读访问或 RDBUFF 读请求时都会更新 DP-CTRL/STAT 寄存器的 READOK 标志，以便了解 AP 读访问是否成功。
- SW-DP 有写缓冲区（用于 DP 或 AP 写入），这样即使在其它操作仍未完成时，也可以接受写入操作。如果写缓冲区已满，则目标确认响应为 WAIT。但 IDCODE 读取、CTRL/STAT 读取或 ABORT 写入除外，这几项操作在写缓冲区已满时也会被接受。

- 由于存在异步时钟域 SWCLK 和 HCLK，因此写操作后（奇偶校验位后）还需要两个额外的 SWCLK 周期，以使写入操作在内部生效。应在将线路驱动为低电平时（空闲状态）应用这些周期。

在写 CTRL/STAT 寄存器以提出一个上电请求时，这一点特别重要。否则下一个操作（在内核上电后才有效的操作）会立即执行，这将导致失败。

33.4.5 SW-DP 寄存器

当 APnDP=0 时能够访问这些寄存器：

A[3:2]	RW	SELECT 寄存器的 CTRLSEL 位	寄存器	注释
00	读取		IDCODE	制造商代码设置为 Cortex®-M0+ 的默认 ARM® 代码。 0x0BC11477（标识 SW-DP）
00	写		ABORT	
01	读/写	0	DP-CTRL/STAT	目的： – 请求系统或调试上电 – 配置 AP 访问的传输操作 – 控制比较和验证操作 – 读取一些状态标志（上溢和上电确认）
01	读/写	1	WIRE CONTROL	用于配置物理串行端口协议（如转换时间的 持续时间）
10	读取		READ RESEND	允许从已损坏的调试软件传输中恢复读取数 据，无需重复执行原始 AP 传输。
10	写		SELECT	用于选择当前访问端口和活动的 4 字寄存器 窗口
11	读/写		READ BUFFER	由于已发出 AP 访问，因此该读缓冲区非常有 用（在执行下个 AP 事务时提供读取 AP 请求 的结果）。 此读取缓冲区捕获 AP 中的数据，显示为

前一次读取的结果，无需启动新操作。

33.4.6 SW-AP 寄存器

当 APnDP=1 时能够访问这些寄存器：

有多个 AP 寄存器，这些寄存器按以下组合进行寻址：

- 移位值 A[3:2]
- DP SELECT 寄存器的当前值

地址	A[3:2] 值	说明
0x0	00	保留，必须保持复位值。
0x4	01	DP CTRL/STAT 寄存器。用于： – 请求系统或调试上电 – 配置 AP 访问的传输操作 – 控制比较和验证操作 – 读取一些状态标志（上溢和上电确认）
0x8	10	DP SELECT 寄存器：用于选择当前访问端口和活动的 4 字寄存器窗口。 – 位 31:24：APSEL：选择当前 AP (select the current AP) – 位 23:8：保留 – 位 7:4：APBANKSEL：在当前 AP 上选择活动的 4 字寄存器窗口 – 位 3:0：保留
0xC	11	DP RDBUFF 寄存器：用于通过调试器在执行一系列操作后获取最后结果 （无需请求新的 JTAG-DP 操作）

33.5内核调试

通过内核调试寄存器调试内核。通过调试访问端口调试访问这些寄存器。它由四个寄存器组成：

寄存器	说明
DHCSR	32 位调试停止控制和状态寄存器 此寄存器提供有关处理器状态的信息，能够使内核进入调试停止状态并提供处理器步进功能。
DCRSR	17 位调试内核寄存器选择器寄存器： 此寄存器选择需要进行读写操作的处理器寄存器。
DCRDR	32 位调试内核寄存器数据寄存器： 此寄存器保存在寄存器与 DCRSR（选择器）寄存器选择的处理器之间读取和写入的 数据。
DEMCR	32 位调试异常和监视控制寄存器： 此寄存器提供向量捕获和调试监视控制。

这些寄存器在系统复位时不复位。它们只能通过上电复位来复位。有关更多详细信息，请参见 Cortex®-M0+ TRM。

为了在复位后立即使内核进入调试停止状态，必须：

- 使能调试和异常监视控制寄存器的位 0 (VC_CORRESET)
- 使能调试停止控制和状态寄存器的位 0 (C_DEBUGEN)

33.6BPU（断点单元）

Cortex®-M0+ BPU 实现提供四个断点寄存器。

33.6.1 BPU 功能

处理器断点实现了基于 PC 的断点功能。

有关 BPU CoreSight 标识寄存器及其地址和访问类型的更多信息，请参见 ARMv6-M ARM®和 ARM® CoreSight 组件技术参考手册。

33.7DWT（数据观察点）

Cortex®-M0+ DWT 实现提供了两个观察点寄存器组。

33.7.1 DWT 功能

处理器观察点实现了数据地址和基于 PC 的观察点功能（即 PC 采样寄存器），并支持比较器地址掩码，如 ARMv6-M ARM® 中所述。

33.7.2 DWT 程序计数器采样寄存器

实现数据观察点单元的处理器还实现了 ARMv6-M 可选 DWT 程序计数器采样寄存器(DWT_PCSR)。此寄存器允许调试程序定期采样 PC，无需停止处理器。这可提供粗略分析。有关更多信息，请参见 ARMv6-M ARM®。

Cortex®-M0+ DWT_PCSR 记录通过条件代码和指令以及未通过条件代码的指令。

33.8MCU 调试组件（DBG)

MCU 调试组件帮助调试器为以下各项提供支持：

- 低功耗模式
- 断点期间的定时器、看门狗的时钟控制

33.8.1 对低功耗模式的调试支持

要进入低功耗模式，必须执行指令 WFI 或 WFE。

MCU 支持多个低功耗模式，这些模式可以禁止 CPU 时钟或降低 CPU 功耗。

内核不允许在调试会话期间关闭 FCLK 或 HCLK。由于调试期间需要使用它们进行调试连接，因此其必须保持激活状态。MCU 集成了特殊方法，允许用户在低功耗模式下调试软件。

33.8.2 对定时器、看门狗的调试支持

断点期间，必须选择定时器和看门狗的计数器的行为方式：

- 在产生断点时，计数器继续计数。例如，当 PWM 控制电机时，通常需要这种方式。
- 在产生断点时，计数器停止计数。用于看门狗时需要这种方式。

33.9调试模式模块工作状态控制（DEBUG_ACTIVE）

复位值 0x00001FFF (仅在 SWD 调试模式下，此寄存器设置才起作用)

偏移地址：0x4000_2038

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			LPTIM1	TIM3	Res.		WDT	PCA	TIM6	TIM5	TIM4	LPTIM0	TIM2	TIM1	TIM0
			RW	RW			RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:13	Reserved	保留
12	LPTIM1	调试时， LpTimer1 计数功能配置 1：在 SWD 调试界面下，暂停 LPTimer1 计数功能 0：在 SWD 调试界面下，LPTimer1正常计数功能
11	TIM3	调试时， Timer3 计数功能配置 1：在 SWD 调试界面下，暂停 Timer3计数功能 0：在 SWD 调试界面下，Timer3正常计数功能
10:9	Reserved	保留
8	WDT	调试时， WDT 计数功能配置 1：在 SWD 调试界面下，暂停 WDT 计数功能 0：在 SWD 调试界面下，WDT 正常计数功能
7	PCA	调试时， PCA 计数功能配置 1：在 SWD 调试界面下，暂停 PCA 计数功能 0：在 SWD 调试界面下，PCA 正常计数功能
6	TIM6	调试时， Timer6 计数功能配置 1：在 SWD 调试界面下，暂停 Timer 计数功能，暂停后再次运行需要初始化计数器计数值，否则PWM输出相位可能不对。 0：在 SWD 调试界面下，Timer 正常计数功能
5	TIM5	调试时， Timer5 计数功能配置 1：在 SWD 调试界面下，暂停 Timer 计数功能，暂停后再次运行需要初始化计数器计数值，否则PWM输出相位可能不对。 0：在 SWD 调试界面下，Timer 正常计数功能
4	TIM4	调试时， Timer4 计数功能配置 1：在 SWD 调试界面下，暂停 Timer 计数功能，暂停后再次运行需要初始化计数器计数值，否则PWM输出相位可能不对。 0：在 SWD 调试界面下，Time r正常计数功能
3	LPTIM0	调试时， LpTimer0 计数功能配置

		1：在 SWD 调试界面下，暂停 LPTimer0 计数功能 0：在 SWD 调试界面下，LPTimer0正常计数功能
2	TIM2	调试时， Timer2 计数功能配置 1：在 SWD 调试界面下，暂停 Timer 计数功能 0：在 SWD 调试界面下，Timer 正常计数功能
1	TIM1	调试时， Timer1 计数功能配置 1：在 SWD 调试界面下，暂停Timer计数功能 0：在 SWD 调试界面下，Timer正常计数功能
0	TIM0	调试时， Timer0 计数功能配置 1：在 SWD 调试界面下，暂停 Timer 计数功能 0：在 SWD 调试界面下，Timer 正常计数功能

34 器件电子签名

电子签名存放在闪存存储器模块的系统存储区域，可以通过 SWD 或者 CPU 读取。它包含的芯片识别信息在出厂时编写，用户固件或者外部设备可以读取电子签名，用以自动匹配不同配置的 HC32Fxxx / HC32Lxxx 微控制器。

34.1产品唯一身份标识（UID）寄存器（80 位）

唯一身份标识符典型应用场景：

- 用作序列号
- 在对内部 Flash 进行编程前将 UID 与软件加密原语和协议结合使用时用作安全密钥以提高 Flash 中代码的安全性
- 激活安全自举过程等

80 位的唯一设备标识符提供了一个对于任何设备和任何上下文都唯一的参考号码。用户永远不能改变这些位。80 位的唯一设备标识符也可以以单字节/半字/字等不同方式读取，然后使用自定义算法连接起来。

基址：0x0010 0E74

偏移地址	描述	UID Bits（80 bits）							
		7	6	5	4	3	2	1	0
0	Lot Number	UID[7:0]							
1		UID[15:8]							
2		UID[23:16]							
3		UID[31:24]							
4		UID[39:32]							
5		UID[47:40]							
6	X Coordinate on the wafer	UID[55:48]							
7	Y Coordinate on the wafer	UID[63:56]							
8	Wafer Number	UID[71:64]							
9	Rev ID	UID[79:72]							

34.2产品型号寄存器

0x0010 0C60 ~ 0x0010 0C6F 存储了产品型号的 ASCII 码。如产品型号不足 16 字节，则以 0x00 进行填充。

例：484333324C3133364B38544100000000 所代表的产品型号为 HC32L136K8TA。

34.3FLASH 容量寄存器

基地址：0x0010 0C70

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FlashSize[31:16]															
R															

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

位	标记	功能描述
31:0	FlashSize	产品内置Flash的容量，以字节为单位 0x00008000代表Flash容量为32K Byte

34.4RAM 容量寄存器

基地址：0x0010 0C74

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RamSize[31:16]															
R															

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

位	标记	功能描述
31:0	RamSize	产品内置RAM的容量，以字节为单位 0x00000800代表RAM容量为2K Byte

34.5管脚数量寄存器

基地址：0x0010 0C7A

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

位	标记	功能描述
15:0	PinCount	产品管脚的数量，以只为单位 0x0020代表产品管脚数量为32

35 附录 A SysTick 定时器

35.1SysTick 定时器简介

OS 要想支持多任务，就需要周期执行上下文切换，这样就需要有定时器之类的硬件资源打断程序执行。当定时器中断产生时，处理器就会在异常处理中进行 OS 任务调度，同时还会进行 OS 维护的工作。Cortex-M0 处理器中有一个称为 SysTick 的简单定时器，用于产生周期性的中断请求。

SysTick 为 24 位的定时器，并且向下计数。定时器的计数减到 0 后，就会重新装载一个可编程的数值，并且同时产生 SysTick 异常（异常编号为 15），该异常事件会引起 SysTick 异常处理的执行，这个过程是 OS 的一部分。

对于不需要 OS 的系统，SysTick 定时器也可以用作其他用途，比如定时、计时或者为需要周期执行的任务提供中断源。SysTick 异常的产生是可控的，如果异常被禁止，仍然可以用轮询的方法使用 SysTick 定时器，比如检查当前的计数值或者轮询计数标志。

35.2设置 SysTick

由于 SysTick 定时器的重载值和当前值在复位时都是未定义的，为了防止产生异常结果，对 SysTick 的配置需要遵循一定的流程：

Step1：配置 SysTick->CTRL. ENABLE 为 0，禁止 SysTick。

Step2：配置 SysTick->CTRL. CLKSOURCE，选择 SysTick 的时钟源。

Step3：配置 SysTick->LOAD，选择 SysTick 的溢出周期。

Step4：向 SysTick->VAL 写入任意值，清零 SysTick->VAL 及 SysTick->CTRL. COUNTFLAG。

Step5：配置 SysTick->CTRL. TICKINT 为 1，使能 SysTick 中断。

Step6：配置 SysTick->CTRL. ENABLE 为 1，使能 SysTick。

Step7：在中断服务程序中读取 SysTick->CTRL 以清除溢出标志。

注意：Systick 溢出周期为 SysTick->LOAD+1，配置示例如下：

时钟源	SysTick->LOAD	溢出周期
HCLK 4MHz	3999	1ms
XTL 32.768KHz	327	10.01ms

35.3SysTick 寄存器

地址	名称	CMSIS 符号	全名
0XE000E010	SYS_CSR	SysTick->CTRL	SysTick 控制和状态寄存器
0XE000E014	SYS_RVR	SysTick->LOAD	SysTick 重装载寄存器
0XE000E018	SYS_CVR	SysTick->VAL	SysTick 当前值寄存器
0XE000E01C	SYS_CALIR	SysTick->CALIB	SysTick 校准值寄存器

35.3.1 SysTick 控制和状态寄存器（CTRL）

位	符号	功能描述	类型	复位值
31:17	Reserved	-	-	-
16	COUNTFLAG	SysTick定时器溢出标志 1：SysTick定时器发生下溢出 0：SysTick定时器未发生溢出 读该寄存器，可清除COUNTFLAG标志	RO	0
15:3	Reserved	-	-	-
2	CLKSOURCE	SysTick时钟源选择 1：使用内核时钟(HCLK) 0：使用参考时钟(XTL) 注：当选择参考时钟作为SysTick的时钟源时，需要同步使能XTL时钟源及SysCtrl.PERI_CLKEN0.TICK	RW	0
1	TICKINT	SysTick中断使能 1：使能中断 0：禁止中断	RW	0
0	ENABLE	SysTick定时器使能 1：使能SysTick 0：禁止SysTick	RW	0

35.3.2 SysTick 重载寄存器（LOAD）

位	符号	功能描述	类型	复位值
31:24	Reserved	-	-	-
23:0	RELOAD	SysTick 定时器重载值	RW	未定义

35.3.3 SysTick 当前值寄存器（VAL）

位	符号	功能描述	类型	复位值
31:24	Reserved	-	-	-
23:0	CURRENT	读取该寄存器，获取SysTick定时器的当前计数值 写任意值到该寄存器，清零该寄存器及COUNTFLAG	RW	未定义

35.3.4 SysTick 校准值寄存器（CALIB）

位	符号	功能描述	类型	复位值
31	NOREF	SysTick当前计数时钟标志 1：当前计数时钟为内核时钟 0：当前计数时钟为外部参考时钟（XTL）	RO	-
30	SKEW	TENMS精度指示 1：TENMS值代表粗略的10ms 0：TENMS值代表精确的10ms	RO	
29:24	Reserved	-	-	
23:0	TENMS	10毫秒校准值	RO	-

36 附录 B　文档约定

36.1寄存器相关缩写词列表

寄存器说明中使用以下缩写词：

RW：读写，软件可以读写这些位。

RO：只读，软件只能读取这些位。

WO：只写，软件只能写入该位。读取该位时将返回无效数据。

W1：只写 1，硬件自动清 0，写 0 无效。

R0W1：软件读取该位为 0，写入 1 将该位清零。写入 0 对该位的值无影响。

RW0：软件可以读写该位，写 1 无效，写 0 清除。

R1W0：软件读取该位为 1,，写入 0 将该位清零。写入 1 对该位的值无影响。

RC：软件可以读取该位。读取该位时，将自动清零。写入“0”对该位的值无影响。

Res, Reserverd：保留位，必须保持复位值。

36.2词汇表

本节简要介绍本文档中所用首字母缩略词和缩写词的定义：

Word：32 位数据。

Half Word：16 位数据。

Byte：8 位数据。

IAP（在应用中编程）：IAP 是指可以在用户程序运行期间对微控制器的 Flash 进行重新编程。

ICP（在线编程）：ICP 是指可以在器件安装于用户应用电路板上时使用 JTAG 协议、SWD 协议或自举程序对微控制器的 Flash 进行编程。

AHB：高级高性能总线。

APB：低速外设总线。

DMA：直接存储器访问。

TIM：定时器。

版本修订记录

版本号	修订日期	修订内容
Rev1.00	2019/11/20	初稿发布。
Rev1.10	2019/12/31	更新以下信息：①增加QFN32封装；②典型应用电路图；③器件电子签名；④附录A SysTick 定时器；⑤高速外部时钟XTH和低速外部时钟XTL中配图和注意事项；⑥通用工作条件描述；⑦系统控制寄存器1（SYSCTRL1）增加注意项；⑧使用温度传感器测量环境温度；⑨DMA控制器（DMAC）和循环冗余校验（CRC）章节描述；⑩I2C配置寄存器(I2Cx_CR)的0位。
Rev1.20	2020/04/10	更新以下信息：①引脚功能说明；②29.7.2.17和29.7.2.24增加注意项；③7.5.1和7.5.2的Step8；④封装尺寸中QFN32；⑤3.2.7和3.2.8中增加Step；⑥ADC特性中增加AVCC/3精度；⑦HC32L073KATA的44/45/47/48pin变更；⑧HC32L073JATA的32/33/35/36pin变更。
Rev1.30	2020/05/29	更新以下信息：①11.3.2删除注意项；②外部输入低速时钟中修正笔误；③增加LCD控制器；④内部RCL振荡器中RCL振荡器精度；⑤产品特性 超低功耗 MCU中增加低功耗定时器描述；⑥7.5中增加Step2和Step3。
Rev1.40	2020/06/30	更新以下信息：①引脚功能说明中增加I2S信息；②LPTIM更正为LPTIM0，LPTIMx_ETR更正为LPTIMx_EXT；③统一引脚功能名称，统一部分寄存器名称；④更新21.3.1.6和22.3.2.6描述。
Rev1.50	2020/07/31	更新以下信息：①输入特性——端口 PA，PB，PC，PD，PE，PF中V_{IH}值和V_{IL}的值；②增加TIM定时器特性和通信接口节；③EFT特性等级；④调整29.7.1和29.7.2寄存器。
Rev1.60	2020/09/30	更新以下信息：①功能模块中功能框图；②增加SPI 特性和I2S 特性；③11.8.4、11.8.14和35.4.2更新笔误；④时钟系统描述；⑤RESETB引脚特性的V_{IL}和V_{IH}；⑥增加输入特性——USB_DP，USB_DM；⑦28.7.3的bit3。
Rev1.61	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。
Rev1.62	2021/09/17	更换为清晰度高的矢量图。
Rev1.63	2022/03/09	公司Logo更新。
Rev1.64	2022/08/13	更新以下信息：①“I2C 总线”章节，I2C 操作流程勘误；②“外部高速晶振时钟XTH”章节，增加建议使用有源晶振的说明；③“端口数字复用功能”章节，删除TIM4_CHB的PF01功能映射；④“周期间隔响应”章节，增加此功能的初始化方法描述及注意事项；⑤“波特率设定”章节，增加采样点的说明；⑥“控制器局域网”章节，增加CAN使用注意事项。

版本号	修订日期	修订内容
Rev1.65	2023/06/21	“脉冲宽度测量PWC寄存器描述”章节，控制寄存器（TIMx_CR0）增加未开放的功能描述。
Rev1.70	2024/05/23	更新以下信息：①3.5.2 SYSCTRL1寄存器的EXTL_EN和EXTH_EN功能描述的注意事项修改；②7.5 编程示例操作步骤更新；③9.3.2.2 时钟监测硬件结构中被监测时钟源添加PLL；④15.2.3.2 PCA 16位PWM功能中添加注意事项；⑤16.3.30 TIMx_PTBKS与16.3.33 TIMx_PTBKP中PC5改为PC15；⑥更新26.5.9 CAN_ERRINT寄存器的BEIF功能说明；⑦模数转换器（ADC）章节中删除1.2V相关内容；⑧模拟电压比较器（VC）章节中删除1.2V和片内BGR输出的参考电压相关内容。
Rev1.71	2025/01/14	3.5.2 SYSCTRL1寄存器的EXTL_EN和EXTH_EN功能描述的注意事项修改。
Rev1.72	2025/03/28	删除8.2 SPI主要特性中关于主从机最高通信速率的描述。
Rev1.73	2026/02/11	更新以下信息：①“模式2/3 比较捕获模式-单脉冲输出”章节脉冲计数时序图细节优化，新增相关说明；②模式2/3 比较捕获模式-捕获输入”章节新增注意“当GATE配置为比较输出时，GATE的输出模式仅支持配置为强制输出为0或1和PWM模式有效（即OCMB0配置为0/1/6/7时，GATE输出有效）”；③“单线半双工数据收发说明”章节新增描述“配置LPUART_SCON.HDSEL=0b1后，要等待一位数据的发送时长（即1/BaudRate秒后），才能正常接收”；④“OPA校零”章节删除，OPA_CR1寄存器中删除相关OPA校零描述，删除OPA_CR寄存器。
Rev1.80	2026/08/19	更新以下信息：①“13.4.4 DMA硬件Block传输模式”章节，修改硬件Block传输示意图；②“14.2.6 模式2/3比较捕获模式-捕获输入”章节，修改CH0B捕获输入信号选择使用约束；③“18.3.4 多机通讯”“19.3.5 多机通讯”章节，修改从机地址检测发现广播地址描述；④“33.4 SWD端口”章节，修改SW-DP core ID“0x0BB11477”为“0x0BC11477”，”33.9 调试模式模块工作状态控制（DEBUG_ACTIVE）”章节，删除RTC内容。